Test Plan

Assignment

# **Document Control**

|  |  |
| --- | --- |
| **Document Name** | Test Plan |
| **Reference Number** | - |
| **Version** | V1.0 |
| **Project Code** | UECS2354 Software Testing Assignment |
| **Status** | [to be updated] |
| **Date Released** | [to be updated] |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Position** | **Signature** |
| Prepared By: |  | **Test Manager** |  |
| Reviewed By: |  | **Quality Assurance Team** |  |
| Approved By: |  | **Operation Manager** |  |
| Approved By: |  | **Project Manager** |  |

# **Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Release Date** | **Section** | **Amendments** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Content

[Document Control 1](#_Toc127346001)

[Version History 2](#_Toc127346002)

[1.0 Introduction 1](#_Toc127346003)

[1.1 Objective 1](#_Toc127346004)

[1.2 Scope 1](#_Toc127346005)

[1.3 References 1](#_Toc127346006)

[2.0 Test Plans 2](#_Toc127346007)

[2.1 Test Items 2](#_Toc127346008)

[2.2 Features to be tested. 2](#_Toc127346009)

[2.3 Features not to be tested. 2](#_Toc127346010)

[2.4 Test Basis 2](#_Toc127346011)

[2.5 Test Conditions 2](#_Toc127346012)

[2.6 Entry Criteria 2](#_Toc127346013)

[2.7 Exit Criteria 2](#_Toc127346014)

# **Introduction**

## 1.1 Objective

This test plan supports the following objectives:

1. To make sure the hotel management booking system can function according to the requirement.
2. To define the correct test tools and condition to conduct the software testing.
3. To analyse the requirement and designing the test case accordingly.

## 1.2 Scope

The scope of testing covers the test of categorizing users into VIP member, normal member, or non-member. The test plan is to test exclusive reward system that affects room allocation for normal members. It will focus on testing the rules for room allocation based on member type. The Printer and Room classes does not ready for testing in this test plan. The test plan covers unit test and integration test.

## 1.3 References

1. IEEE 829-1998 Standard for Software and System Test Documentation

# **2.0 Test Plans**

## 2.1 Test Items

UserTest:

* testGetName() : to verify getName() can return the correct name.
* testGetMemberType(): to verify getMemberTyper() can return the correct member type.
* testGetExclReward(): to verify getExclReward() can return the correct exclusice reward status.
* testSetName(String name, String memberType, boolean exclReward): to verify setName() method can set the name.
* testSetMemberType(String name, String memberType, boolean exclReward): to verify setMemberType sets the member typer.
* testSetExclReward(String name, String memberTyper, Boolean exclReward): to verify setExclReward() sets the exclusive reward status.
* illegalTestSetName(String name, String memberTyper, Boolean exclReward): to verify that system throws illegalArgumentexception.
* illegalTestSetMemberType(String name, string memberTyper, Boolean excReward): to verify invalid member type throws illegalArgumentException.
* illegalTestSetExclReward(String name, String memberType, Boolean exclReward): to verify invalid exclusive reward throws illegalArgumentException.

WaitingListTest:

* testAddWaiting(String user1, String user2, String listType): to verify users are added to the waitingListUser.txt.
* testGetWaitingList(String user1, String user2, String listType): to verify users can be retrieved from the waitingListUser.txt.
* testRemoveWaiting(String user, String listType): to verify users can be removed from the waitingListUser.txt.
* illegalTestRemoveWaiting(String user, String listType): to verify that attempting to remove user not on the waitingListUser.txt throws illegalArgumentException.

BookingTest:

* testSetBooking(String name, String memberTyper, boolean exclReward,int numberOfRooms): to verify setbooking() is handled correctly based on user attributes.
* testSetBookingVipRoomForVipUser(String name, String member Type, boolean exclReward,int numberOfRooms): to verify VIP room will be allocate to a VIP member if a VIP member requests one room when VIP room is available..
* testSetBookingDeluxeRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify Deluxe room is allocate to a VIP member if a VIP member requests room but there is no VIP room is available.
* testSetBookingStandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Standard room is allocate to a VIP member if a VIP member requests one room but no VIP or Deluxe room is available.
* testSetBookingWaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify VIP member will be add to VIPList if a VIP member request three room but no VIP, Deluxe or Standard room is available.
* testSetBooking1VipAnd1DeluxeRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and one Deluxe room will be allocate to a VIP member if a VIP member request two room but there is only one VIP room available.
* testSetBooking1DeluxeAnd1StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room and one Standard room will be allocate to a VIP member if a VIP member request two room but there is not VIP room available and only one Deluxe room available.
* testSetBooking1VipAnd1StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and one Standard room will be allocate to a VIP member if a VIP member request two room but there is only one VIP room available and there is no Deluxe room available.
* testSetBooking1VipRoomAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request two room but there is only one VIP room available but no other room available.
* testSetBooking1DeluxeAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request two room but there is only one Deluxe room available but no other room available.
* testSetBooking1StandardAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to one Standard room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request two room but there is only one Standard room available but no other room available.
* testSetBooking2VipAnd1DeluxeRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two VIP room and one Deluxe room will be allocate to a VIP member if a VIP member request two room but there is only two VIP room available.
* testSetBooking2VipAnd1StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two VIP room and one Standard room will be allocate to a VIP member if a VIP member request three room but there is only two VIP room available and there is no Deluxe room available.
* testSetBooking2VipRoomAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two VIP room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only two VIP room available but no other room available.
* testSetBooking1VipAnd2DeluxeRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and two Deluxe room will be allocate to a VIP member if a VIP member request three room but there is only one VIP room available.
* testSetBooking2DeluxeAnd1StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two Deluxe room and one Standard room will be allocate to a VIP member if a VIP member request three room but there is not VIP available and only two Deluxe room available.
* testSetBooking2DeluxeAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two Deluxe room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only two Deluxe room available but no other room available.
* testSetBooking1VipAnd2StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and two Standard room will be allocate to a VIP member if a VIP member request three room but there is only one VIP room available and there is no Deluxe room available.
* testSetBooking1DeluxeAnd2StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room and two Standard room will be allocate to a VIP member if a VIP member request three room but there is not VIP available and only one Deluxe room available.
* testSetBooking2StandardAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two Standard room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only two Standard room available but no other room available.
* testSetBooking1VipRoomAnd2WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only one VIP room available but no other room available.
* testSetBooking1DeluxeAnd2WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only one Deluxe room available but no other room available.
* testSetBooking1StandardAnd2WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Standard room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only one Standard room available but no other room available.
* testSetBookingVipDeluxeAndStandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room, one Deluxe room and one Standard room will be allocated to a VIP member if a VIP member request three room but there is only one VIP room available and only one Deluxe room available.
* testSetBooking1Vip1DeluxeAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to To verify one Vip room and one Standard room will be allocate to a VIP member he/she will be added to VIPList if a VIP member request three room but there is only one Deluxe room available and only one Standard room available.
* testSetBooking1Vip1StandardAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and one Standard room will be allocate to a VIP member he/she will be added to VIPList if a VIP member request three room but there is only one VIP room available and only one Standard room available.
* testSetBooking1Deluxe1StandardAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room and one Standard room will be allocate to a VIP member he/she will be added to VIPList if a VIP member request three room but there is only one Deluxe room available and only one Standard room available.
* illegalTestSetBookingForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify invalid number of room throw illegalArgumentException.
* testSetBooking1DeluxeRoomForMemberUserWithoutExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room will be allocate to a member if a member without exclusive reward requests one room when a Deluxe room is available.
* testSetBooking1StandardRoomForMemberUserWithoutExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Standard room is allocate to a member if a member without exclusive reward requests one room but there is no Deluxe room is available.
* testSetBooking2DeluxeRoomForMemberUserWithAndWithoutExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two Deluxe room will be allocate to a member if a member requests two room when two Deluxe room is available.
* testSetBookingDeluxeAndStandardRoomForMemberUserWithAndWithoutExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room and one Standard room will be allocate to a member if a member request two room but there is only one Deluxe room available.
* testSetBooking2StandardRoomForMemberUserWithAndWithoutExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two Standard room is allocate to a member if a member with exclusive reward requests two room but there is no Deluxe room is available.
* testSetBooking1DeluxeAnd1WaitingListForMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room will be allocate to a member and he/she will be added to memberList if a member with exclusive reward request two room but there is only one Deluxe room available but VIP room available.
* testSetBooking1StandardAnd1WaitingListForMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Standard room will be allocate to a member and he/she will be added to memberList if a member with exclusive reward request two room but there is only one Deluxe room available but VIP room available.
* testSetBookingWaitingListForMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify member will be add to memberList if a member request two room with exclusive reward but no VIP, Deluxe or Standard room is available.
* testSetBooking1VIPRoomForMemberUserWithExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room will be allocate to a member if a member requests one room with exclusive reward when a VIP room is available.
* testSetBookingVIPAndStandardRoomForMemberUserWithExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and Standard room will be allocate to a member if a member requests two room with exclusive reward when a VIP room is available.
* testSetBookingVIPAndDeluxeRoomForMemberUserWithExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and Deluxe room will be allocate to a member if a member requests two room with exclusive reward when a VIP room is available.
* testSetBooking1VipAnd1WaitingListForMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room will be allocate to a member and he/she will be added to memberList if a member requests two room with exclusive reward when a VIP room is available and no other room available.
* illegalTestSetBookingForMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify invalid number of rooms throw illegalArgumentException.
* testSetBookingStandardRoomForNonMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify non member can book for 1 Standard room.
* testSetBookingWaitingListForNonMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify a non-member will be added to nonMemberList if a non-member requests one room when a no Standard room is available.
* illegalTestSetBookingForNonMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify System will throw illegalArgumentException if the value is invalid.
* testCancelBookingWithBookedRooms(String name, String memberType, boolean exclReward,String roomType1,String roomType2): to verify user can cancel booking.
* testCancelBookingWithUserOnWaitingListOnly(String name, String memberType, Boolean exclReward, String roomType1, String roomType2): to verify user on waiting list can cancel booking by removing.
* illegalTestCancelBookingWithNoBookingsOrWaitingList(String name, String memberType, boolean exclReward,String roomType1,String roomType2): to verify invalid input will throw illegalArgumentException.

BookingIntegrationTest:

* testSetBooking(String name, String memberTyper, boolean exclReward,int numberOfRooms): to verify setbooking() is handled correctly based on user attributes.
* testSetBookingVipRoomForVipUser(String name, String member Type, boolean exclReward,int numberOfRooms): to verify VIP room will be allocate to a VIP member if a VIP member requests one room when VIP room is available..
* testSetBookingDeluxeRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify Deluxe room is allocate to a VIP member if a VIP member requests room but there is no VIP room is available.
* testSetBookingStandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Standard room is allocate to a VIP member if a VIP member requests one room but no VIP or Deluxe room is available.
* testSetBookingWaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify VIP member will be add to VIPList if a VIP member request three room but no VIP, Deluxe or Standard room is available.
* testSetBooking1VipAnd1DeluxeRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and one Deluxe room will be allocate to a VIP member if a VIP member request two room but there is only one VIP room available.
* testSetBooking1DeluxeAnd1StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room and one Standard room will be allocate to a VIP member if a VIP member request two room but there is not VIP room available and only one Deluxe room available.
* testSetBooking1VipAnd1StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and one Standard room will be allocate to a VIP member if a VIP member request two room but there is only one VIP room available and there is no Deluxe room available.
* testSetBooking1VipRoomAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request two room but there is only one VIP room available but no other room available.
* testSetBooking1DeluxeAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request two room but there is only one Deluxe room available but no other room available.
* testSetBooking1StandardAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to one Standard room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request two room but there is only one Standard room available but no other room available.
* testSetBooking2VipAnd1DeluxeRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two VIP room and one Deluxe room will be allocate to a VIP member if a VIP member request two room but there is only two VIP room available.
* testSetBooking2VipAnd1StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two VIP room and one Standard room will be allocate to a VIP member if a VIP member request three room but there is only two VIP room available and there is no Deluxe room available.
* testSetBooking2VipRoomAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two VIP room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only two VIP room available but no other room available.
* testSetBooking1VipAnd2DeluxeRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and two Deluxe room will be allocate to a VIP member if a VIP member request three room but there is only one VIP room available.
* testSetBooking2DeluxeAnd1StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two Deluxe room and one Standard room will be allocate to a VIP member if a VIP member request three room but there is not VIP available and only two Deluxe room available.
* testSetBooking2DeluxeAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two Deluxe room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only two Deluxe room available but no other room available.
* testSetBooking1VipAnd2StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and two Standard room will be allocate to a VIP member if a VIP member request three room but there is only one VIP room available and there is no Deluxe room available.
* testSetBooking1DeluxeAnd2StandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room and two Standard room will be allocate to a VIP member if a VIP member request three room but there is not VIP available and only one Deluxe room available.
* testSetBooking2StandardAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two Standard room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only two Standard room available but no other room available.
* testSetBooking1VipRoomAnd2WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only one VIP room available but no other room available.
* testSetBooking1DeluxeAnd2WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only one Deluxe room available but no other room available.
* testSetBooking1StandardAnd2WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Standard room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only one Standard room available but no other room available.
* testSetBookingVipDeluxeAndStandardRoomForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room, one Deluxe room and one Standard room will be allocated to a VIP member if a VIP member request three room but there is only one VIP room available and only one Deluxe room available.
* testSetBooking1Vip1DeluxeAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to To verify one Vip room and one Standard room will be allocate to a VIP member he/she will be added to VIPList if a VIP member request three room but there is only one Deluxe room available and only one Standard room available.
* testSetBooking1Vip1StandardAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and one Standard room will be allocate to a VIP member he/she will be added to VIPList if a VIP member request three room but there is only one VIP room available and only one Standard room available.
* testSetBooking1Deluxe1StandardAnd1WaitingListForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room and one Standard room will be allocate to a VIP member he/she will be added to VIPList if a VIP member request three room but there is only one Deluxe room available and only one Standard room available.
* illegalTestSetBookingForVipUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify invalid number of room throw illegalArgumentException.
* testSetBooking1DeluxeRoomForMemberUserWithoutExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room will be allocate to a member if a member without exclusive reward requests one room when a Deluxe room is available.
* testSetBooking1StandardRoomForMemberUserWithoutExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Standard room is allocate to a member if a member without exclusive reward requests one room but there is no Deluxe room is available.
* testSetBooking2DeluxeRoomForMemberUserWithAndWithoutExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two Deluxe room will be allocate to a member if a member requests two room when two Deluxe room is available.
* testSetBookingDeluxeAndStandardRoomForMemberUserWithAndWithoutExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room and one Standard room will be allocate to a member if a member request two room but there is only one Deluxe room available.
* testSetBooking2StandardRoomForMemberUserWithAndWithoutExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify two Standard room is allocate to a member if a member with exclusive reward requests two room but there is no Deluxe room is available.
* testSetBooking1DeluxeAnd1WaitingListForMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Deluxe room will be allocate to a member and he/she will be added to memberList if a member with exclusive reward request two room but there is only one Deluxe room available but VIP room available.
* testSetBooking1StandardAnd1WaitingListForMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one Standard room will be allocate to a member and he/she will be added to memberList if a member with exclusive reward request two room but there is only one Deluxe room available but VIP room available.
* testSetBookingWaitingListForMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify member will be add to memberList if a member request two room with exclusive reward but no VIP, Deluxe or Standard room is available.
* testSetBooking1VIPRoomForMemberUserWithExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room will be allocate to a member if a member requests one room with exclusive reward when a VIP room is available.
* testSetBookingVIPAndStandardRoomForMemberUserWithExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and Standard room will be allocate to a member if a member requests two room with exclusive reward when a VIP room is available.
* testSetBookingVIPAndDeluxeRoomForMemberUserWithExclReward(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room and Deluxe room will be allocate to a member if a member requests two room with exclusive reward when a VIP room is available.
* testSetBooking1VipAnd1WaitingListForMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify one VIP room will be allocate to a member and he/she will be added to memberList if a member requests two room with exclusive reward when a VIP room is available and no other room available.
* illegalTestSetBookingForMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify invalid number of rooms throw illegalArgumentException.
* testSetBookingStandardRoomForNonMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify non member can book for 1 Standard room.
* testSetBookingWaitingListForNonMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify a non-member will be added to nonMemberList if a non-member requests one room when a no Standard room is available.
* illegalTestSetBookingForNonMemberUser(String name, String memberType, boolean exclReward,int numberOfRooms): to verify System will throw illegalArgumentException if the value is invalid.
* testCancelBookingWithBookedRooms(String name, String memberType, boolean exclReward,String roomType1,String roomType2): to verify user can cancel booking.
* testCancelBookingWithUserOnWaitingListOnly(String name, String memberType, Boolean exclReward, String roomType1, String roomType2): to verify user on waiting list can cancel booking by removing.
* illegalTestCancelBookingWithNoBookingsOrWaitingList(String name, String memberType, boolean exclReward,String roomType1,String roomType2): to verify invalid input will throw illegalArgumentException.

## 2.2 Features to be tested.

* A user can be categorized as VIP, normal member or non-member.
* VIP member:
  1. VIP members allocate to VIP room.
  2. If Deluxe room is fully booked, Standard room will be allocated.
  3. A VIP can book up to 3 rooms at a time. If only one VIP room is available upon requested, the Deluxe or Standard rooms will be allocated to VIP member.
  4. If there is no rooms available, the member is placed into VIP waiting list
* Normal member:

1. normal members allocate to Deluxe room
2. If Deluxe room is fully booked, normal member will allocate to standard room without exclusive reward.
3. Normal member with exclusive reward will be allocated VIP room if there is VIP room available. If VIP room is not available, standard room will be allocated and exclusive reward remains valid for other redemptions.
4. If Standard room is fully booked too, member will be placed to member waiting list.

* Non-member:

1. A non-member can only book one room at a time.
2. If Standard room is fully booked, non-member will be placed into normal waiting list.

## 2.3 Features not to be tested.

* Printing the booking information
* Check the room availability using room classes
* Updating the room status after customer check out

## 2.4 Test Basis

## Project Requirement Document: to analyse the requirement for a user can be a VIP member, normal member or non-member. There is rule for room allocation based on member type.

## Class Diagrams: showing relationship between User class, Booking class, WaitingList class, Room class and Printer class.

## 2.5 Test Conditions

1. To verify one VIP room will be allocate to a VIP member if a VIP member requests one room when a VIP room is available.
2. To verify one Deluxe room is allocate to a VIP member if a VIP member requests one room but there is no VIP room is available.
3. To verify one Standard room is allocate to a VIP member if a VIP member requests one room but no VIP or Deluxe room is available.
4. To verify VIP member will be add to VIPList if a VIP member request a room but no VIP, Deluxe or Standard room is available.
5. To verify two VIP rooms will be allocate to a VIP member if a VIP member requests two rooms when two VIP rooms is available.
6. To verify two Deluxe room will be allocate to a VIP member if a VIP member requests two room but there is no VIP room is available.
7. To verify two Standard room will be allocate to a VIP member if a VIP member requests two room there is no VIP room or Deluxe room is available.
8. To verify one VIP room and one Deluxe room will be allocate to a VIP member if a VIP member request two room but there is only one VIP room available.
9. To verify one Deluxe room and one Standard room will be allocate to a VIP member if a VIP member request two room but there is not VIP room available and only one Deluxe room available.
10. To verify one VIP room and one Standard room will be allocate to a VIP member if a VIP member request two room but there is only one VIP room available and there is no Deluxe room available.
11. To verify one VIP room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request two room but there is only one VIP room available but no other room available.
12. To verify one Deluxe room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request two room but there is only one Deluxe room available but no other room available.
13. To verify one Standard room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request two room but there is only one Standard room available but no other room available.
14. To verify VIP member will be add to VIPList if a VIP member request two room but no VIP, Deluxe or Standard room is available.
15. To verify three VIP rooms will be allocate to a VIP member if a VIP member requests three rooms when three VIP rooms is available.
16. To verify three Deluxe room will be allocate to a VIP member if a VIP member requests three room but there is no VIP room is available.
17. To verify three Standard room will be allocate to a VIP member if a VIP member requests three room there is no VIP room or Deluxe room is available.
18. To verify VIP member will be add to VIPList if a VIP member request three room but no VIP, Deluxe or Standard room is available.
19. To verify two VIP room and one Deluxe room will be allocate to a VIP member if a VIP member request two room but there is only two VIP room available.
20. To verify two VIP room and one Standard room will be allocate to a VIP member if a VIP member request three room but there is only two VIP room available and there is no Deluxe room available.
21. To verify two VIP room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only two VIP room available but no other room available.
22. To verify one VIP room and two Deluxe room will be allocate to a VIP member if a VIP member request three room but there is only one VIP room available.
23. To verify two Deluxe room and one Standard room will be allocate to a VIP member if a VIP member request three room but there is not VIP available and only two Deluxe room available.
24. To verify two Deluxe room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only two Deluxe room available but no other room available.
25. To verify one VIP room and two Standard room will be allocate to a VIP member if a VIP member request three room but there is only one VIP room available and there is no Deluxe room available.
26. To verify one Deluxe room and two Standard room will be allocate to a VIP member if a VIP member request three room but there is not VIP available and only one Deluxe room available.
27. To verify two Standard room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only two Standard room available but no other room available.
28. To verify one VIP room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only one VIP room available but no other room available.
29. To verify one Deluxe room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only one Deluxe room available but no other room available.
30. To verify one Standard room will be allocate to a VIP member and he/she will be added to VIPList if a VIP member request three room but there is only one Standard room available but no other room available.
31. To verify one VIP room, one Deluxe room and one Standard room will be allocated to a VIP member if a VIP member request three room but there is only one VIP room available and only one Deluxe room available.
32. To verify one Vip room and one Standard room will be allocate to a VIP member he/she will be added to VIPList if a VIP member request three room but there is only one Deluxe room available and only one Standard room available.
33. To verify one VIP room and one Standard room will be allocate to a VIP member he/she will be added to VIPList if a VIP member request three room but there is only one VIP room available and only one Standard room available.
34. To verify one Deluxe room and one Standard room will be allocate to a VIP member he/she will be added to VIPList if a VIP member request three room but there is only one Deluxe room available and only one Standard room available.
35. To verify system will throw illegalArgumentException when VIP member requested zero room.
36. To verify system will throw illegalArgumentException when VIP member requested four room.
37. To verify system will throw illegalArgumentException when VIP member enter string instead of integer when requesting for room.
38. To verify one Deluxe room will be allocate to a member if a member requests one room when a Deluxe room is available.
39. To verify one Standard room is allocate to a member if a member requests one room but there is no Deluxe room is available.
40. To verify member will be add to memberList if a member request a room but no Deluxe or Standard room is available.
41. To verify one VIP room will be allocate to a member if a member requests one room with exclusive reward when a VIP room is available.
42. To verify member will be add to memberList if a member request a room with exclusive reward but no VIP, Deluxe or Standard room is available.
43. To verify two Deluxe room will be allocate to a member if a member requests two room when two Deluxe room is available.
44. To verify two Standard room is allocate to a member if a member requests two room but there is no Deluxe room is available.
45. To verify member will be add to memberList if a member request two room but no Deluxe or Standard room is available.
46. To verify one Deluxe room and one Standard room will be allocate to a member if a member request two room but there is only one Deluxe room available.
47. To verify one Deluxe room will be allocate to a member and he/she will be added to memberList if a member request two room but there is only one Deluxe room available but no other room available.
48. To verify one Standard room will be allocate to a member and he/she will be added to memberList if a member request two room but there is only one Standard room available but no other room available.
49. To verify two Standard room is allocate to a member if a member with exclusive reward requests two room but there is no Deluxe room is available.
50. To verify two Deluxe room will be allocate to a member if a member with exclusive reward requests two room when two Deluxe room is available.
51. To verify member will be add to memberList if a member request two room with exclusive reward but no VIP, Deluxe or Standard room is available.
52. To verify one VIP room and Standard room will be allocate to a member if a member requests two room with exclusive reward when a VIP room is available.
53. To verify one VIP room and Deluxe room will be allocate to a member if a member requests two room with exclusive reward when a VIP room is available.
54. To verify one VIP room will be allocate to a member and he/she will be added to memberList if a member requests two room with exclusive reward when a VIP room is available and no other room available.
55. To verify one Deluxe room and one Standard room will be allocate to a member if a member with exclusive reward request two room but there is no VIP room available.
56. To verify one Standard room will be allocate to a member and he/she will be added to memberList if a member with exclusive reward request two room but there is only one Deluxe room available but VIP room available.
57. To verify one Deluxe room will be allocate to a member and he/she will be added to memberList if a member with exclusive reward request two room but there is only one Deluxe room available but VIP room available.
58. To verify system will throw illegalArgumentException when member requested zero room.
59. To verify system will throw illegalArgumentException when member requested four room.
60. To verify system will throw illegalArgumentException when member enter string instead of integer when requesting for room.
61. To verify system will throw illegalArgumentException when member without exclusive reward request for VIP room.
62. To verify one Standard room will be allocate to a non-member if a non-member requests one room when a Standard room is available.
63. To verify a non-member will be added to nonMemberList if a non-member requests one room when a no Standard room is available.
64. To verify system will throw illegalArgumentException when non-member requested zero room.
65. To verify system will throw illegalArgumentException when non-member requested four room.
66. To verify system will throw illegalArgumentException when non-member enter string instead of integer when requesting for room.
67. To verify Vip member with exclusive reward and booked room can cancel bookings.
68. To verify member without exclusive reward and booked room can cancel bookings.
69. To verify non-member without exclusive reward and booked room can cancel bookings.
70. To verify Vip member on waiting list with exclusive reward with no booked room can cancel bookings.
71. To verify member on waiting list without exclusive reward with no booked room can cancel bookings.
72. To verify non-member on waiting list with exclusive reward with no booked room can cancel bookings.
73. To verify system throw illegalArgumentException if Vip member try to cancel booking without book room and not on waiting list.
74. To verify system throw illegalArgumentException if member try to cancel booking without book room and not on waiting list.
75. To verify system throw illegalArgumentException if non-member try to cancel booking without book room and not on waiting list.
76. To verify getter method retrieve a Vip member with a name and exclusive reward.
77. To verify getter method a member with a name and exclusive reward.
78. To verify getter method a non-member with a name without an exclusive reward.
79. To verify system will throw illegalArgumentException when creating name with null value.
80. To verify system will throw illegalArgumentException when creating name with empty value.
81. To verify system will throw illegalArgumentException when creating name with integer.
82. To verify system will throw illegalArgumentException when creating name with double.
83. To verify system will throw illegalArgumentException when creating member type with null value.
84. To verify system will throw illegalArgumentException when creating member type with empty value.
85. To verify system will throw illegalArgumentException when creating member type with integer.
86. To verify system will throw illegalArgumentException when creating member type with double.
87. To verify system will throw illegalArgumentException when creating member type with String other than VIP, member or non-member.
88. To verify system will throw illegalArgumentException when create user’s exclusive reward with null value.
89. To verify system will throw illegalArgumentException when create user’s exclusive reward with integer.
90. To verify system will throw illegalArgumentException when create user’s exclusive reward with double.
91. To verify adding Vip member to the list.
92. To verify adding Vip member who is already on the list.
93. To verify adding member to the list.
94. To verify adding member who is already on the list.
95. To verify adding non-member to the list.
96. To verify removing a Vip member from the list.
97. To verify removing a member from the list.
98. To verify removing non-member from the list.
99. To verify system throw illegalArgumentException when removing member who is not on the list.
100. To verify system throw illegalArgumentException when removing non-member who is not on the list.
101. To verify system throw illegalArgumentException when removing Vip member who is not on the list.

## 2.6 Entry Criteria

The entry criteria for the test plan are:

* Completion of the hotel room booking system development.
* Completion of the test plan and test cases.
* Availability of test data to simulate various booking scenarios.
* Availability of testing tools and libraries such as JUnit and Mockito.

## 2.7 Exit Criteria

The exit criteria for the test plan are:

* All conditions written in test plan are tested and validated.
* All test cases have been executed, passed and result are documented.
* All Bugs identified during testing are resolved and retested to make sure the bugs are eliminated.
* The hotel room booking system fulfill the functional and performance requirements.
* The Hotel Room Booking System is ready for deployment following the successful completion of the User Acceptance Testing (UAT).