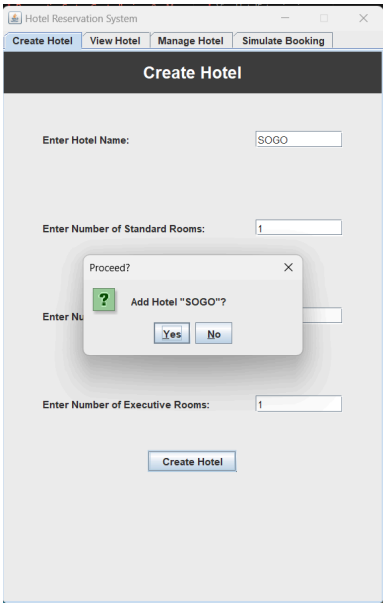
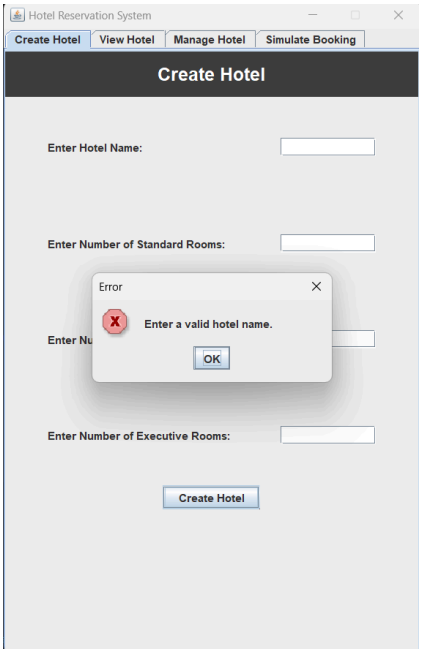
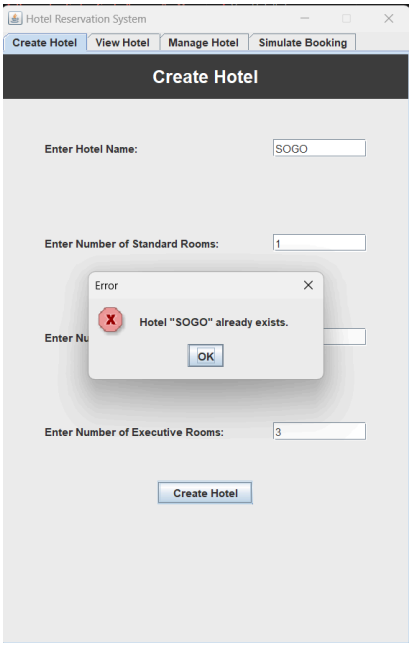
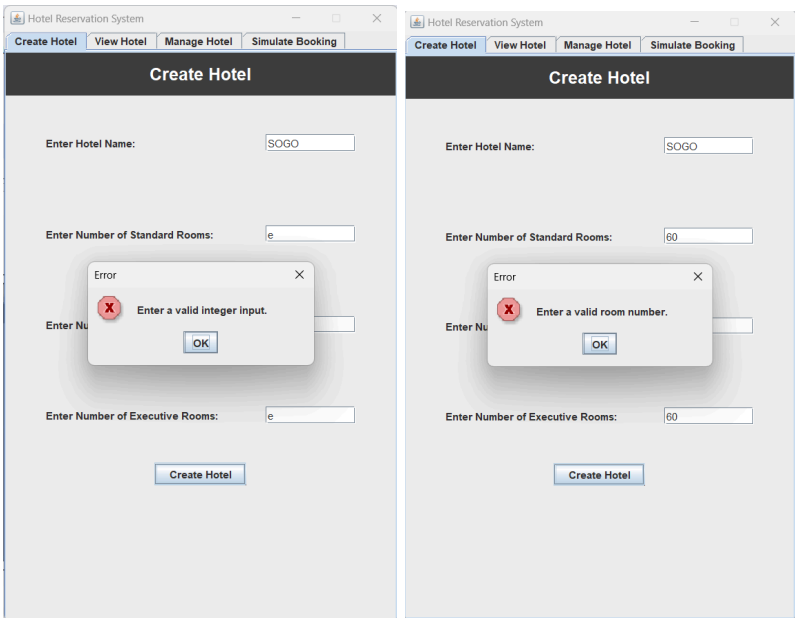
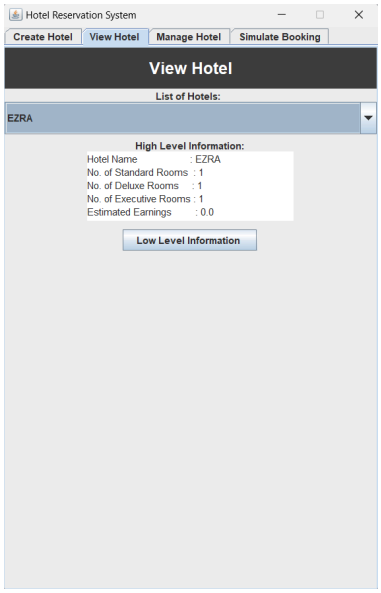
Group Members: Burayag, Ethan Axl S. - Del Rosario, Ezra Jeonadab G.
Group #: 4

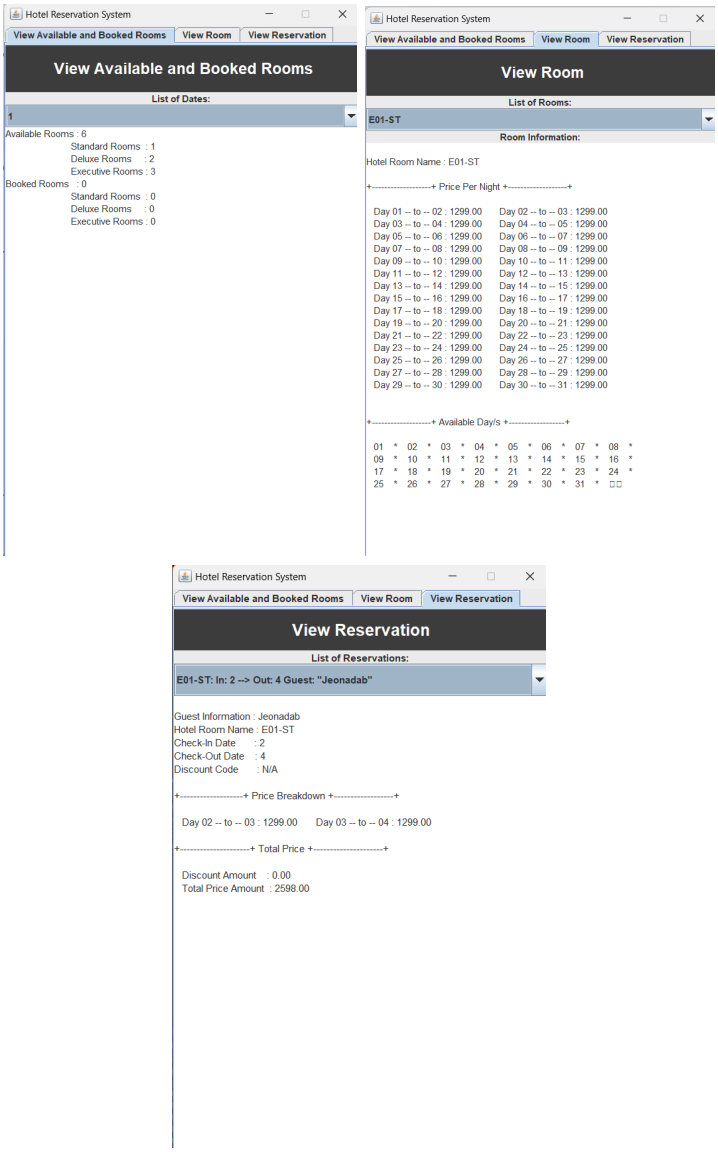
MCO2 - Test Script

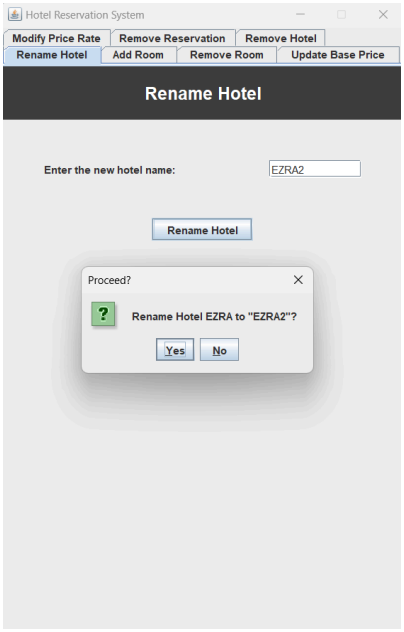
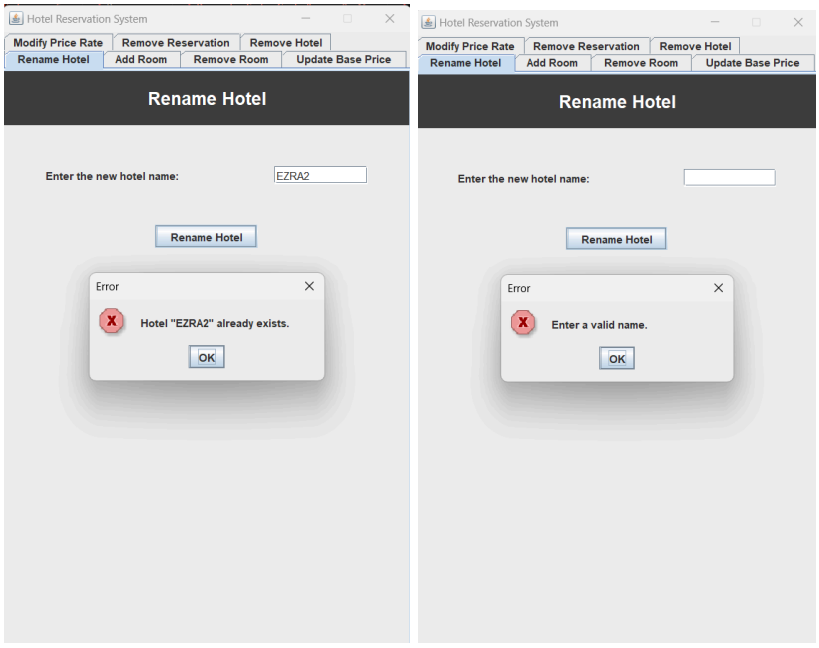
CCPROG3

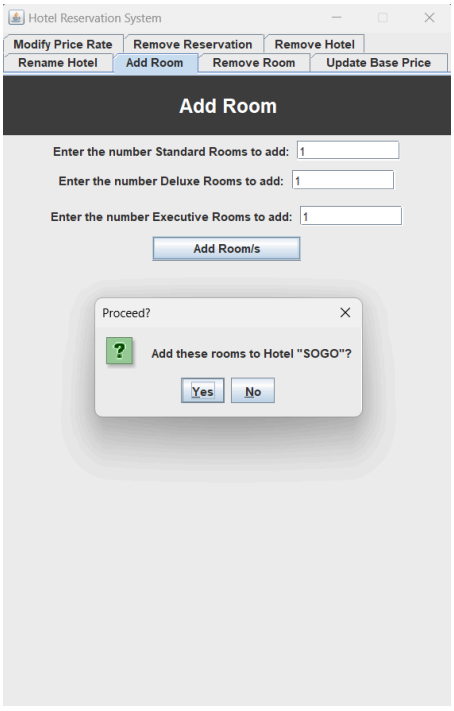
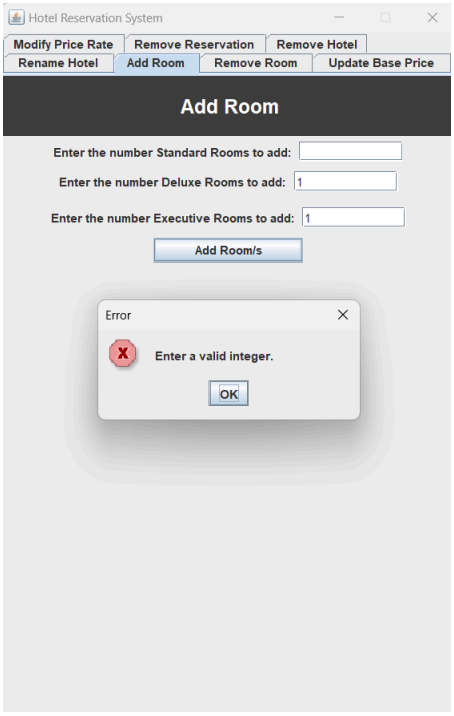
<table>
<tr><th colspan="4">View Classes : (ReservationSystemView, ViewHotelExtension, ManageHotelExtension)</th></tr>
<tr><th>Case</th><th>#</th><th></th><th>P/F</th></tr>
<tr><td>CreateHotel
Valid</td><td>1</td><td> 
</td><td>P</td></tr>
<tr><td>CreateHotel
Invalid Hotel Name</td><td>2</td><td> 

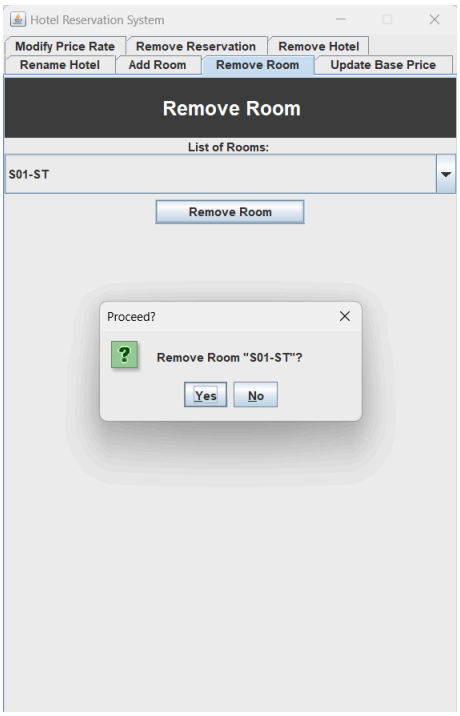
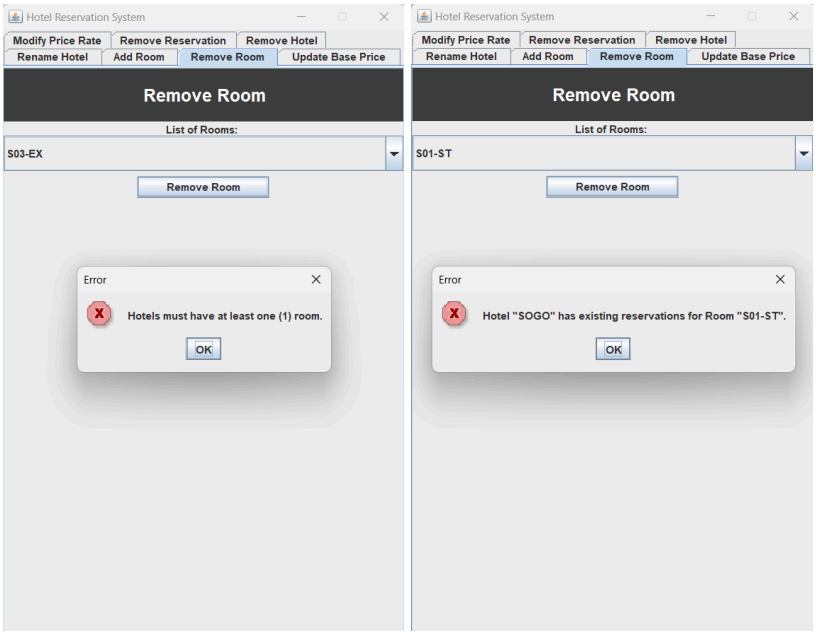

</td><td>P</td></tr>
</table>

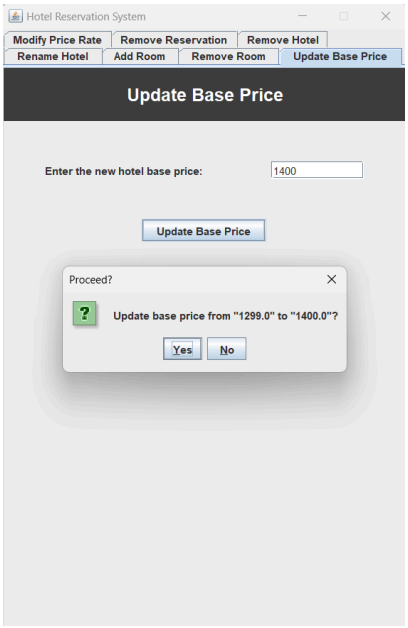
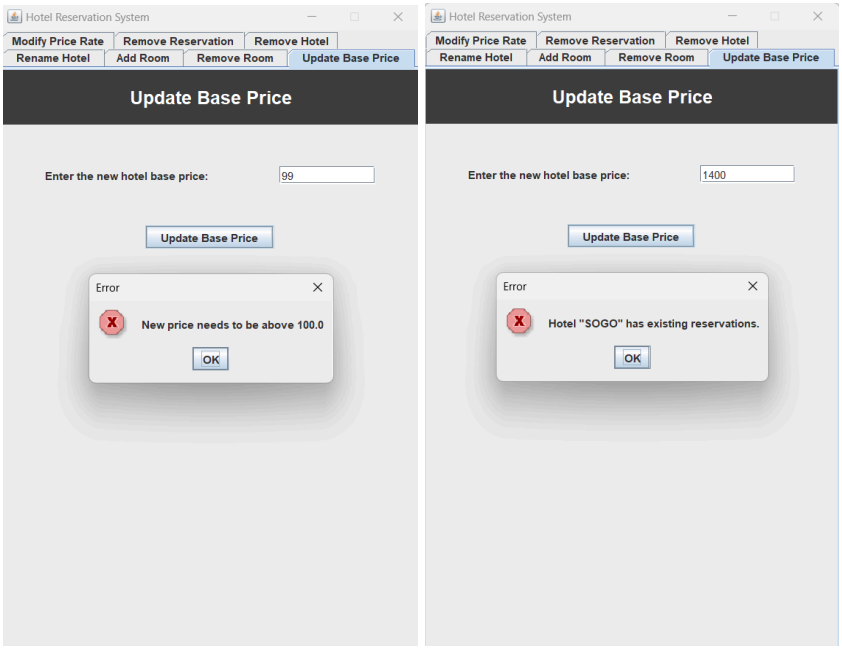
CreateHotel **Invalid Room Number Input**	3		P
View Hotel **High Level Information**	4		P

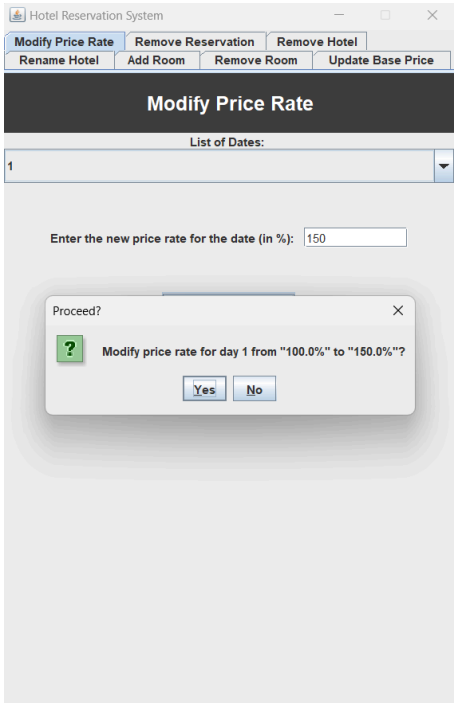
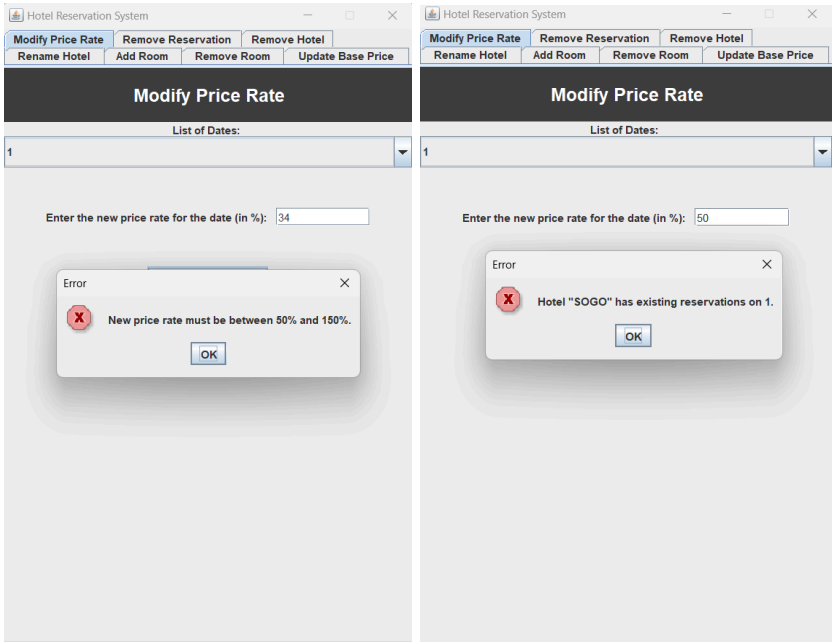
View Hotel **Low Level Information**	5		P

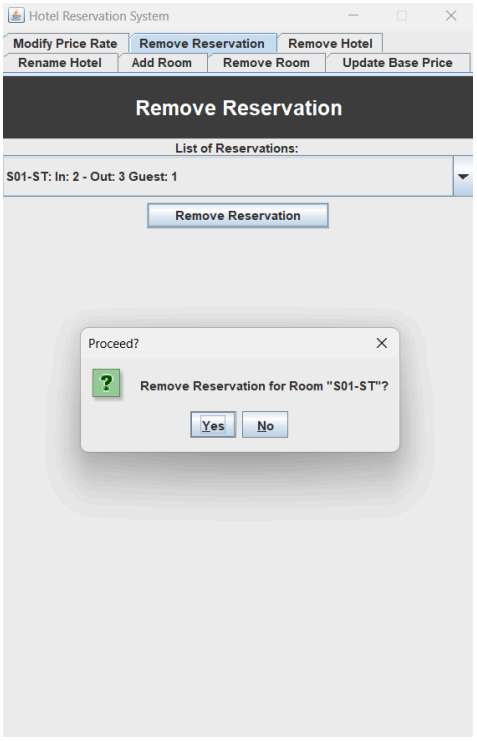
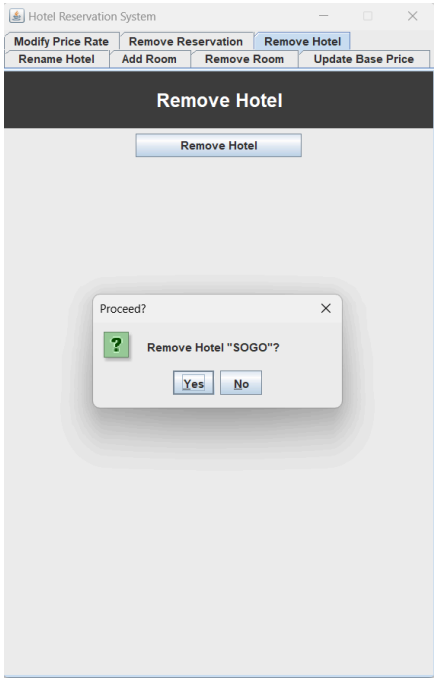
Manage Hotel **Valid Rename Hotel**	6		P
Manage Hotel **Invalid Rename Hotel**	7		P

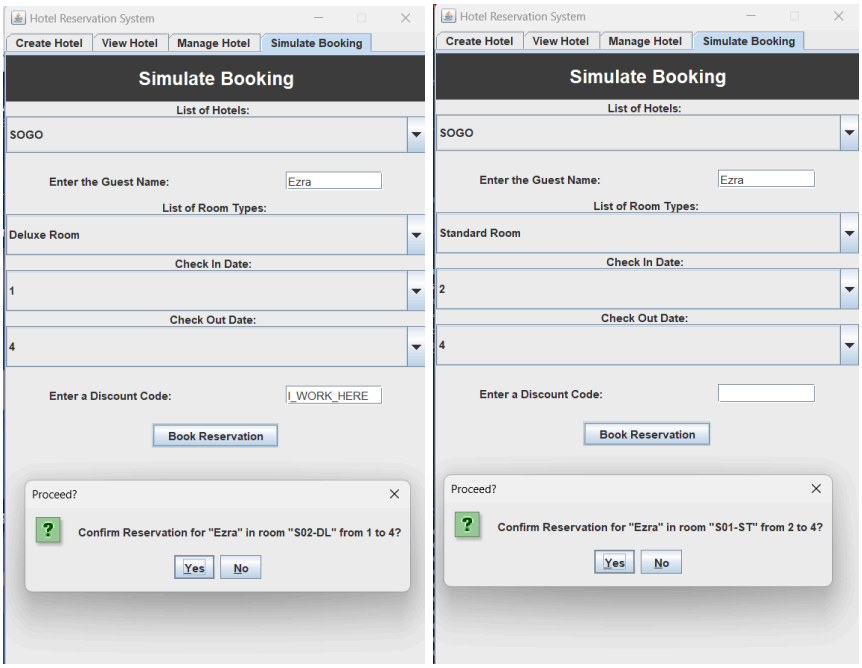
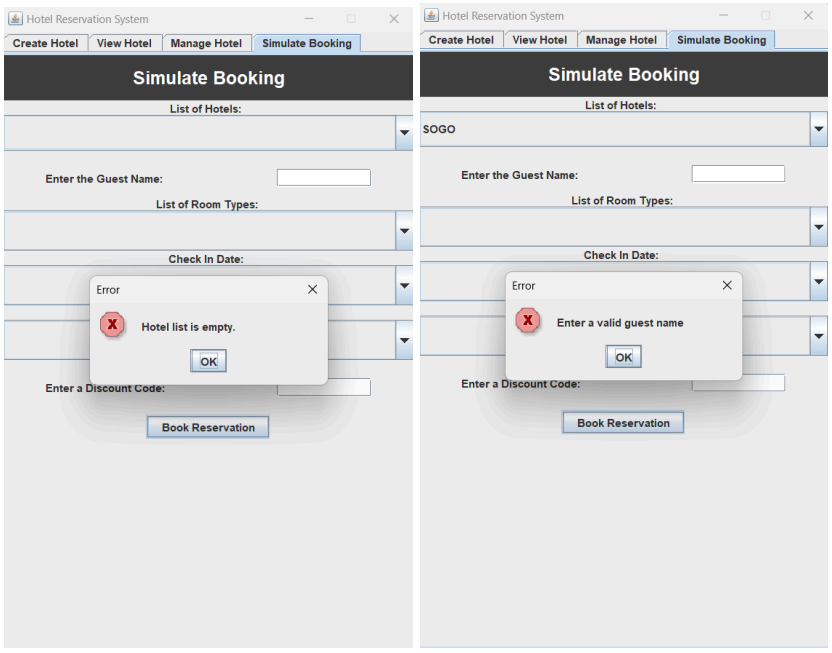
Manage Hotel **Valid Add Room**	8		P
Manage Hotel **Invalid Add Room**	9		P

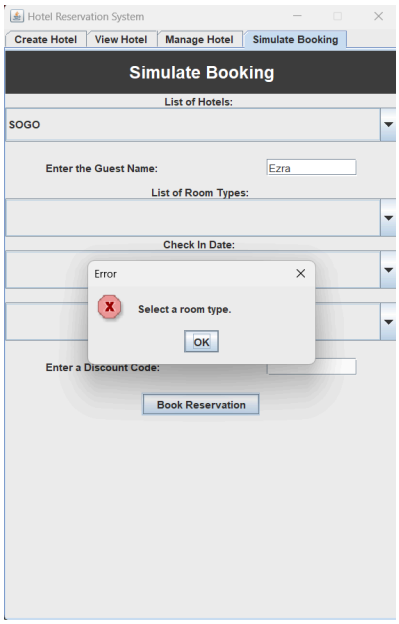
Manage Hotel **Valid Remove Room**	10		P
Manage Hotel **Invalid Remove Room**	11		P

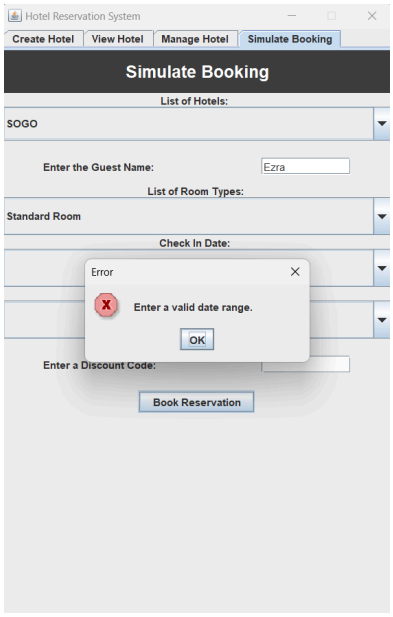
Manage Hotel **Valid Update Base Price**	12		P
Manage Hotel **Invalid Update Base Price**	13		P

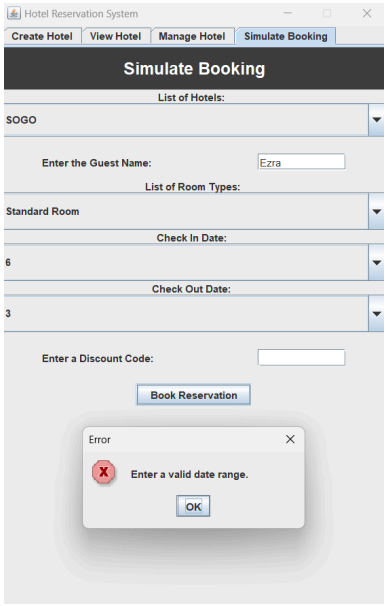
Manage Hotel **Valid Modify Price Rate**	14		P
Manage Hotel **Invalid Modify Price Rate**	15		P

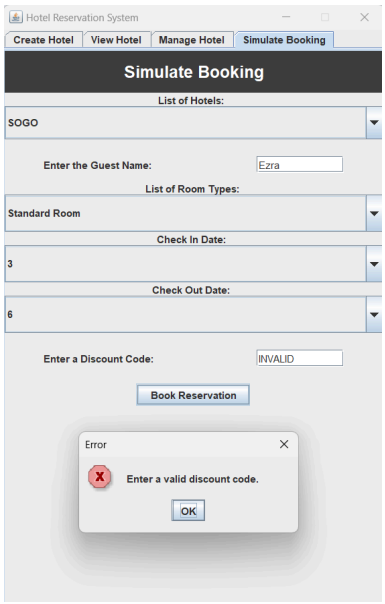
Manage Hotel **Valid Remove Reservation**	16		P
Manage Hotel **Valid Remove Hotel**	17		P

Simulate Booking **Valid Booking**	18		P
Simulate Booking **Invalid Booking**	19		P


Hotel Reservation System
Create Hotel
View Hotel
Manage Hotel
Simulate Booking
Simulate Booking
List of Hotels:
SOGO
Enter the Guest Name:
Ezra
List of Room Types:
Check In Date:
Error
Select a room type.
OK
Enter a Discount Code:
Book Reservation


Hotel Reservation System
Create Hotel
View Hotel
Manage Hotel
Simulate Booking
Simulate Booking
List of Hotels:
SOGO
Enter the Guest Name:
Ezra
List of Room Types:
Standard Room
Check In Date:
Error
Enter a valid date range.
OK
Enter a Discount Code:
Book Reservation


Hotel Reservation System
Create Hotel
View Hotel
Manage Hotel
Simulate Booking
Simulate Booking
List of Hotels:
SOGO
Enter the Guest Name:
Ezra
List of Room Types:
Standard Room
Check In Date:
6
Check Out Date:
3
Enter a Discount Code:
Book Reservation
Error
Enter a valid date range.
OK


Hotel Reservation System
Create Hotel
View Hotel
Manage Hotel
Simulate Booking
Simulate Booking
List of Hotels:
SOGO
Enter the Guest Name:
Ezra
List of Room Types:
Standard Room
Check In Date:
3
Check Out Date:
6
Enter a Discount Code:
INVALID
Book Reservation
Error
Enter a valid discount code.
OK

Class : ReservationSystem						
Method	**#**	**Test Description**	**Sample Input Data**	**Expected Output**	**Actual Output**	**P/F**
IsExisting	1	Hotel name is non-existent	Input: “INVALID” hotelList : “SOGO” “SOFITEL”	Result = -1	Result = -1	P
	2	Hotel name exists	Input: “SOGO” hotelList : “SOGO” “SOFITEL”	Result = 0	Result = 0	P
	3	Similar hotel names exist but with different lettercases	Input: “SOGO” hotelList : “soGo” “SOGO”	Result = 1	Result = 1	P
Note: The rest of the methods in the reservation system are information getters or methods that do a process but assume all inputs are correct and, therefore, involve no significant logic.						

Class : Hotel						
Method	**#**	**Test Description**	**Sample Input Data**	**Expected Output**	**Actual Output**	**P/F**
sortRoom List	1	Sorts room names that are in increasing order already based on room numbers	roomListNames: {“A01-ST”, “A02-DL”, “A03-EX”, “A04-ST”, “A05-DL”}	roomListNames: {“A01-ST”, “A02-DL”, “A03-EX”, “A04-ST”, “A05-DL”}	roomListNames: {“A01-ST”, “A02-DL”, “A03-EX”, “A04-ST”, “A05-DL”}	P
	2	Sorts room names that are in decreasing order based on room numbers	roomListNames: {“A05-DL”, “A04-ST”, “A03-EX”, “A02-DL”, “A01-ST”}	roomListNames: {“A01-ST”, “A02-DL”, “A03-EX”, “A04-ST”, “A05-DL””}	roomListNames: {“A01-ST”, “A02-DL”, “A03-EX”, “A04-ST”, “A05-DL”}	P
	3	Sorts room names that are in no particular order based on room numbers	roomListNames: {“A10-ST”, “A05-EX”, “A07-DL”, “A01-ST”, “A04-EX”}	roomListNames: {“A01-ST”, “A04-EX”, “A05-EX”, “A07-DL”, “A10-ST”}	roomListNames: {“A01-ST”, “A04-EX”, “A05-EX”, “A07-DL”, “A10-ST”}	P
check Availability	1	Checks availability of a room with no existing reservations for the entire month	Room “A01-ST” Reservation: None Check Availability: checkIn: 6 checkOut: 10	True	True	P
	2	Checks availability of a room for a different range of dates from an existing reservation	Room “A01-ST” Reservation 1: checkIn: 1 checkOut: 5 Check Availability: checkIn: 6 checkOut: 10	True	True	P
	3	Checks availability of a room for a similar range of dates from an existing reservation	Room “A01-ST” Reservation 1: checkIn: 1 checkOut: 5 Check Availability: checkIn: 3 checkOut: 5	False	False	P

	4	Checks availability of a room for an overlap range of dates from an existing reservation	Room “A01-ST” Reservation 1: checkIn: 1 checkOut: 5 Check Availability: checkIn: 4 checkOut: 10	False	False	P
checkRoom Availability IndexList	1	Checks availability for the entire month of the rooms represented by the indexes with all rooms available	roomList: {“A01-ST”, “A02-ST”, “A03-ST”} reservedRoom List: None roomIndexList: {0,1,2}	True	True	P
	2	Checks availability for the entire month of the rooms represented by the indexes with one room unavailable	roomList: {“A01-ST”, “A02-ST”, “A03-ST”} reservedRoom List: {“A01-ST”} roomIndexList: {0,1,2}	False	False	P
	3	Checks availability for the entire month of the rooms represented by the indexes with all rooms unavailable	roomList: {“A01-ST”, “A02-ST”, “A03-ST”} reservedRoom List: {“A01-ST”, “A02-ST”, “A03-ST”} roomIndexList: {0,1,2}	False	False	P
checkDate Availability	1	Checks return if there are no reserved rooms for the date input	reservedRoom List & reservedDate: {“A01-ST” - (1,5), “A02-EX” - (6,10), “A03-DL” - (11,15)}	True	True	P

			date: 20			
	2	Checks return if there are reserved rooms for the date input	reservedRoom List & reservedDate: {“A01-ST” - (1,5), “A02-EX” - (6,10), “A03-DL” - (11,15)} date: 8	False	False	P
	3	No significant test case to be conducted				
get Available RoomType	1	Checks return if there is a room available for the specified type and date	roomType: “Standard Room” checkInDate: 1 checkOutDate: 2 roomList: {“A01-ST”, “A02-DL”, “A03-EX”} fullyReserved RoomList: None	0	0	P
	2	Checks return if there is a room available for the date but not for the specified type	roomType: “Standard Room” checkInDate: 1 checkOutDate: 2 roomLisz: {“A02-DL”, “A03-EX”} fullyReserved Room List: None	-1	-1	P
	3	Checks return if there is a room available for the specified type but not for the date	roomType: “Standard Room” checkInDate: 1 checkOutDate: 2 roomList: {“A01-ST”, “A02-DL”, “A03-EX”}	-1	-1	P

			fullyReserved RoomList: {“A01-ST”}			
checkAll Room Availability	1	Checks all room availability of the hotel with no existing reservations	reservedRoom List: None	True	True	P
	2	Checks all room availability of the hotel with an existing reservation	reservedRoom List: {“A01-ST”}	False	False	P
	3	Checks all room availability of the hotel with multiple existing reservations	reservedRoom List: {“A01-ST”, “A02-DL”, “A03-EX”}	False	False	P
check Discount Applicable	1	Checks validity of discount code with unrecognized discount code	discountCode: “Hello_World!” checkInDate: 1 checkOutDate: 5	False	False	P
	2	Checks validity of discount code with recognized discount code (I_WORK_HERE)	discountCode: “I_WORK_ HERE” checkInDate: 1 checkOutDate: 5	True	True	P
	3	Checks validity of discount code with recognized discount code with non satisfying dates (STAY4_GET1)	discountCode: “STAY4_GET1” checkInDate: 1 checkOutDate: 4	False	False	P
	4	Checks validity of discount code with recognized discount code with non satisfying dates (STAY4_GET1)	discountCode: “STAY4_GET1” checkInDate: 1 checkOutDate: 5	True	True	P
	5	Checks validity of discount code with recognized discount code with non satisfying dates (PAYDAY)	discountCode: “PAYDAY” checkInDate: 1 checkOutDate: 5	False	False	P
	6	Checks validity of discount code with recognized discount code with satisfying dates (PAYDAY)	discountCode: “STAY4_GET1” checkInDate: 1 checkOutDate:	True	True	P

			16			
getRoom Index	1	Gets the index of an existing room name from the room list	roomListNames: {“A01-ST”, “A02-EX”, “A03-DL”, “A04-ST”, “A05-EX”} roomName: “A02-EX”	1	1	P
	2	Gets the index of a non-existing room name from the room list	roomListNames: {“A01-ST”, “A02-EX”, “A03-DL”, “A04-ST”, “A05-EX”} roomName: “A06-ST”	-1	-1	P
	3	Gets the index of an existing room name found at the end of the room list	roomListNames: {“A01-ST”, “A02-EX”, “A03-DL”, “A04-ST”, “A05-EX”} roomName: “A05-EX”	4	4	P
checkRoom IndexList Validity	1	Checks return if all room index values are valid	roomList: {“A01-ST”, “A02-EX”, “A03-DL”, “A04-ST”, “A05-EX”} roomIndexList: {0,2,4}	True	True	P
	2	Checks return if one room index value is invalid	roomList: {“A01-ST”, “A02-EX”, “A03-DL”, “A04-ST”, “A05-EX”} roomIndexList: {0,2,5}	False	False	P
	3	Checks return if all room index values are invalid	roomList: {“A01-ST”, “A02-EX”,	False	False	P

			“A03-DL”, “A04-ST”, “A05-EX”} roomIndexList: {5,6,7}			
addRoom	1	Creates rooms with room names that are sequential from the existing complete room list and adds to the room list	roomListNames: {“A01-ST”, “A02-ST”, “A03-DL”, “A04-DL”, “A05-EX”} numRoomList: {1,1,1}	roomListNames: {“A01-ST”, “A02-ST”, “A03-DL”, “A04-DL”, “A05-EX”, “A06-ST”, “A07-DL”, “A08-EX”}	roomListNames: {“A01-ST”, “A02-ST”, “A03-DL”, “A04-DL”, “A05-EX”, “A06-ST”, “A07-DL”, “A08-EX”}	P
	2	Creates rooms with room names that are sequential from the existing broken room list and adds to the room list	roomListNames: {“A01-ST”, “A03-ST”, “A04-DL”, “A06-DL”, “A07-EX”} numRoomList: {1,1,1}	roomListNames: {“A01-ST”, “A02-ST”, “A03-ST”, “A04-DL”, “A05-DL”, “A06-DL”, “A07-EX”, “A08-EX”}	roomListNames: {“A01-ST”, “A02-ST”, “A03-ST”, “A04-DL”, “A05-DL”, “A06-DL”, “A07-EX”, “A08-EX”}	P
	3	No significant test case to be conducted				
getRoom Info	1	Checks availability of room return if room has no reservations	index: 2 roomList (No reservations): {Room1, Room2, Room3, Room4, Room5}	Room Info of Room3… Availability: 1,2,...,31	Room Info of Room3… Availability: 1,2,...,31	P
	2	Checks availability of room return if room has full reservations	index: 2 roomList (Full reservations): {Room1, Room2, Room3, Room4, Room5}	Room Info of Room3… Availability: None	Room Info of Room3… Availability: None	P
	No significant methods to test for system logic					

Class : Reservation
No significant methods to test for system logic

Class : Room						
Method	**#**	**Test Description**	**Sample Input Data**	**Expected Output**	**Actual Output**	**P/F**
set Availability	1	Checks availability status of room if end date input is less than 31	startDate: 1 endDate: 30 available: false	availability = false, false,..., true	availability = false, false,..., true,	P
	2	Checks availability status of room if end date input is equal to 31	startDate: 1 endDate: 31 available: false	availability = false, false,..., false	availability = false, false,..., false	P
	3	No significant test case to be conducted				