**HOTEL RESERVATION SYSTEM**

**System**

**Hotel Reservation System [HRS]**

|  |
| --- |
| **Name:**Hotel Reservation System |
| **Authors:**Dharshini, Dipanjan, Mahesh and Sanhita |
| **Description:** The Hotel Reservation system has user(Customer) as a participant in most of the use cases. All the communication between any user and the HRS is assumed to occur via authenticated and secure channel. Activities such as login and registration are therefore not included in the use cases. The cost of the menu items available in the menu list are included within the price of room. |
| **Organizations:**Northeastern University |
| **Creation Date:**February 5, 2017 |

**System Actors**

**HRS Admin [sys]**

|  |
| --- |
| **Description:**The singular authority that supervises the overall functionality of the Hotel Reservation system |

**Customer [user]**

|  |
| --- |
| **Description:**The entity that books and uses the hotel services |

**Use Cases**

**Add Customer**

|  |
| --- |
| **Description:**HRS admin creates a new customer record in HRS  **Post condition**: A new customer record is created in HRS |
| **Step-by-step Description:**   1. [#sys] – Creates a new customer record in HRS 2. [HRS] - returns the customer id of the newly created record |

**Add Room**

|  |
| --- |
| **Description:**HRS admin creates a new room record in HRS  **Post condition**: A new room record is created in HRS |
| **Step-by-step Description:**   1. [#sys] – Creates a new room record in HRS 2. [HRS] – returns the room id of the newly created record |

**Add Menu Item**

|  |
| --- |
| **Description:**HRS admin creates a new menu item in HRS  **Post condition**: A new menu item is added in the HRS menu-list if it already doesn’t exist in HRS menu-list |
| **Step-by-step Description:**   1. [#sys] – Creates a new menu item record in HRS 2. [HRS] – Returns the menu id for the newly created item 3. Alternative Path- [#Menu Item Exists] |

**Menu Item Exists**

|  |
| --- |
| **Description:**HRS admin tries to create a new menu item in HRS  **Post condition**: A new menu item is not added in the HRS menu-list as it already exists |
| **Step-by-step Description:**   1. [#sys] – Tries to creates a new menu item record in HRS 2. [HRS] – informs the HRS admin that the item already exists in menu list |

**Delete Room**

|  |
| --- |
| **Description:**HRS admin deletes a room record in HRS  **Post condition**: A room record is deleted from HRS |
| **Step-by-step Description:**   1. [#sys] – Deletes a room record from HRS 2. [HRS] – marks the room status as deleted (this is a soft-delete) |

**Search All Room**

|  |
| --- |
| **Description:**HRS admin searches for all rooms available in HRS  **Post condition**: The list of all room records is displayed |
| **Step-by-step Description:**   1. [#sys] – searched HRS database for all rooms 2. [#HRS] – returns the list of all rooms with details to the HRS admin |

**Search All Customer**

|  |
| --- |
| **Description:**HRS admin wants to search the customer database  **Post condition**: All customer record available in HRS are returned |
| **Step-by-step Description:**   1. [#sys] – Searches HRS database for list of all registered customers 2. [#HRS] – returns the list of all customers in the HRS system |

**Room Search**

|  |
| --- |
| **Description:**User checks for availability of room in HRS  **Post condition**: The HRS provides the user with the list of available rooms based on search criteria |
| **Step-by-step Description:**   1. [#user] - A user searches for the room 2. [#HRS] - The system searches the database for the availability based on search criteria and returns a list of available rooms 3. Alternative Flow: [#Non-Available Room] |

**Book Room**

|  |
| --- |
| **Description:**User views the list of available rooms and books one of them  **Post condition**: The HRS returns the booking details on successful booking |
| **Step-by-step Description:**   1. [#user] - A user reviews the available rooms and books the room 2. [#user]- Makes a partial or full payment during booking 3. [#HRS]-The system notifies the customer about successful booking with a booking details id |

**Non-Available Room**

|  |
| --- |
| **Description:**User tries to search a room in HRS but there are no rooms available matching user search criteria  **Post condition**: The HRS returns a message informing that no rooms are available |
| **Step-by-step Description:**   1. [#user] - A user tries to book a room in HRS 2. [#HRS]-The system notifies the user that no rooms are available for given search criteria |

**Make Payment**

|  |
| --- |
| **Description:**  Users makes a partial or full payment for the booking  **Post condition**: The HRS updates the payment record for the user and returns the payment id |
| **Step-by-step Description:**   1. [#user]- Makes the partial or full payment 2. [#HRS]- Payment record is updated with the payment details. 3. [#HRS]- The system notifies the customer about successful payment by providing the payment details. 4. Alternative Path: [#Payment Failed] |

**Payment Failed**

|  |
| --- |
| **Description:**  Users tries to make a partial or full payment for the booking  **Post condition**: The HRS system couldn’t accept the payment due to some technical issue |
| **Step-by-step Description:**   1. [#user]- Tries to makes the partial or full payment 2. [#HRS]- Payment record is not updated for the user due to some technical issues 3. [#HRS]- The system notifies the customer about payment failure |

**Check Booking Details**

|  |
| --- |
| **Description:**User returns to the HRS to see his/her existing booking  **Post condition**: The HRS returns the details of the user’s booking for the specified booking id |
| **Step-by-step Description:**   1. [#user]- searched for existing booking details in HRS based on booking id 2. [#HRS]- system returns the details of the booking 3. Alternative Flow: [#No booking details] |

**No Booking Details**

|  |
| --- |
| **Description:**User tries to view a booking details with an invalid booking id  **Post condition**: The HRS returns a message to the user informing that no booking details available for the given booking id and customer id |
| **Step-by-step Description:**   1. [#user]- Tries to retrieve booking details with an invalid booking id 2. [#HRS]- system returns a message informing that booking details is not available for the given booking id and customer id |

**Cancel Booking**

|  |
| --- |
| **Description:**User cancels an existing booking  **Post condition**: The HRS provides the user a confirmation message of cancellation for a valid booking details and if not checked-in |
| **Step-by-step Description:**   1. [[#user]](http://www.ccs.neu.edu/home/kenb/ontologies/oor-usecase.xml#User) - A user places a request to cancel an existing booking 2. [#HRS]- The system cancels the booking and deletes the entry from booking details 3. [#HRS]- Change the status of the booked room to available 4. [#HRS]- Initiates a payment settlement by nullifying the total dues 5. Alternative Flow: [# No booking details] |

**Check In**

|  |
| --- |
| **Description:**User arrives at the hotel for check-in  **Post condition**: The user is checked in if user has a valid booking id |
| **Step-by-step Description:**   1. [# sys]- checks the valid details and initiates a check-in HRS 2. [#HRS] – changes the status of the room from booked to check-in 3. Alternative Flow: [#No booking details] |

**Food Order**

|  |
| --- |
| **Description:**User browses through the hotel food-menu and orders for food  **Post condition**: A food order is placed for the user and HRS returns the food order id |
| **Step-by-step Description:**   1. [[#user]](http://www.ccs.neu.edu/home/kenb/ontologies/oor-usecase.xml#Submitter) – checks the menu and places an order in HRS 2. [HRS] – update the food order details in the system 3. [HRS] – returns the user with the food order id |

**Checkout**

|  |
| --- |
| **Description:**User leaves the hotel after making final payment  **Post condition**: The vacated room is made available and payment status is settled |
| **Step-by-step Description:**   1. [[#user]](http://www.ccs.neu.edu/home/kenb/ontologies/oor-usecase.xml#Submitter) –  Initiates check-out from the hotel and clear all dues 2. [#HRS] – payment record for the user updated with no dues 3. [#HRS] – update the status of the room to available |