|  |  |  |
| --- | --- | --- |
| **CalendarLookup Entity** | | |
| **Test Objective:**  This test case will emulate a user searching for an available room on specified date(s). | | |
| **Test Case #:** SERV1 |  | |
| **Test Items:** Reservation (Object), Reservation[] checkDates, assignReservation, “Make Reservation” button (See SDD: sec 5.1 pg. 15, sec 6.1.2 pg. 21) | | |
| **Input Specifications:**  Steps:   1. Click on a date on Calendar (UI) or enter a date range in text boxes. 2. Click on an available room. 3. Click on “Make Reservation” button. | | |
| **Output Specifications:**  If successful, the Guest Reservation form will be displayed and the user will be able to enter the reservation information. | | |
| **Environmental needs:** | *Hardware*   * Sufficient hard disk space to store increasing amounts of data. * Working computer with a processor capable of executing multiple simultaneous commands. * Mouse * Keyboard * Monitor | *Software*   * Operating system with BBMS installed. |
|  |
| **Special Procedural Requirements:**  More than one room may be available and would require the user to request additional preferences from the customer to advance to the reservation form. | | |
| **Intercase dependencies:**  None | | |

|  |  |  |
| --- | --- | --- |
| **TakeGuestInformation Entity** | | |
| **Test Objective:**  This test case will emulate a user entering and saving the customer’s information. | | |
| **Test Case #:** SERV2 |  | |
| **Test Items:** takeGuestInfo(date, date, double, bool, date), Guest[] guests, Guest Information form, “Save Guest button” (Guest Information Form) (See SDD: sec pg. 19, sec 6.1.5 pg. 23-24) | | |
| **Input Specifications:**  Steps:   1. Verify the Guest Reservation form is shown. 2. Click on “Guest Information” to access the Guest Information form. 3. Enter customer’s first name. 4. Enter customer’s last name. 5. Enter customer’s address. 6. Enter customer’s phone number. 7. Enter customer’s credit card number. 8. Click on “Save Guest” button. | | |
| **Output Specifications:**  If successful, the Guest Information form will automatically close and the system will return to the Guest Reservation form. | | |
| **Environmental needs:** | *Hardware*   * Sufficient hard disk space to store increasing amounts of data. * Working computer with a processor capable of executing multiple simultaneous commands. * Mouse * Keyboard * Monitor | *Software*   * Operating system with BBMS installed. |
|  |
| **Special Procedural Requirements:**  None | | |
| **Intercase dependencies:**  The CalendarLookup test case (SERV1) must be completed prior to the TakeGuestInformation test because the Guest Information form can only be accessed after the user selects “Make Reservation” on the Calendar Lookup form. | | |

|  |  |  |
| --- | --- | --- |
| **TakeGuestInformation Entity** | | |
| **Test Objective:**  This test case will emulate a user cancelling the Guest Information process. | | |
| **Test Case #:** SERV3 |  | |
| **Test Items:** takeGuestInfo(date, date, double, bool, date), Guest[] guests, “Cancel” button (See SDD: sec 5.5 pg.19, sec 6.1.5 pg. 23-24) | | |
| **Input Specifications:**  Steps:   1. Verify the Guest Reservation form is shown. 2. Click on “Guest Information” to access the Guest Information form. 3. Enter customer’s first name. 4. Enter customer’s last name. 5. Enter customer’s address. 6. Enter customer’s phone number. 7. Enter customer’s credit card number. 8. Click on “Cancel” button. | | |
| **Output Specifications:**  If successful, the Guest Information form will automatically close and the system will return to the Calendar form. | | |
| **Environmental needs:** | *Hardware*   * Sufficient hard disk space to store increasing amounts of data. * Working computer with a processor capable of executing multiple simultaneous commands. * Mouse * Keyboard * Monitor | *Software*   * Operating system with BBMS installed. |
|  |
| **Special Procedural Requirements:**  None | | |
| **Intercase dependencies:**  The CalendarLookup test case (SERV1) must be completed prior to the TakeGuestInformation test because the Guest Information form can only be accessed after the user selects “Make Reservation” on the Calendar Lookup form. | | |
| **MakePayment Entity** | | |
| **Test Objective:**  This test case will emulate a user processing a customer’s payment. | | |
| **Test Case #:** SERV4 |  | |
| **Test Items:** makePayment(Payment), assignGuest(Guest), “Make Payment” button (See SDD: sec 5.3 pg. 17, sec 6.1.3 pg. 22) | | |
| **Input Specifications:**  Steps:   1. Verify the Guest Reservation form is shown. 2. Click on “Make a Payment” to access the Make Payment form. 3. Enter a total. 4. Enter the payment amount. 5. Select the applicable guest. 6. Click on “Make Payment”. | | |
| **Output Specifications:**  If successful, the Make a Payment form will automatically close and the system will return to the Guest Reservation form. | | |
| **Environmental needs:** | *Hardware*   * Sufficient hard disk space to store increasing amounts of data. * Working computer with a processor capable of executing multiple simultaneous commands. * Mouse * Keyboard * Monitor | *Software*   * Operating system with BBMS installed. |
|  |
| **Special Procedural Requirements:**  None | | |
| **Intercase dependencies:**  The CalendarLookup test case (SERV1) must be completed prior to the MakePayment test case because the Make Payment form can only be accessed after the user selects “Make Reservation” on the Calendar Lookup form. The TakeGuestInformation test case must also be completed prior to this test because a payment must be assigned to a guest’s profile. | | |
| **MakePayment Entity** | | |
| **Test Objective:**  This test case will emulate a user cancelling the payment process. | | |
| **Test Case #:** SERV5 |  | |
| **Test Items:** MakePayment form, “Cancel” button (See SDD: sec 5.3 pg. 17, sec 6.1.3 pg. 22) | | |
| **Input Specifications:**  Steps:   1. Verify the Guest Reservation form is shown. 2. Click on “Make a Payment” to access the Make Payment form. 3. Click on “Cancel”. | | |
| **Output Specifications:**  If successful, the Make a Payment form will automatically close and the system will return to the Guest Reservation form. | | |
| **Environmental needs:** | *Hardware*   * Sufficient hard disk space to store increasing amounts of data. * Working computer with a processor capable of executing multiple simultaneous commands. * Mouse * Keyboard * Monitor | *Software*   * Operating system with BBMS installed. |
|  |
| **Special Procedural Requirements:**  None | | |
| **Intercase dependencies:**  The CalendarLookup test case (SERV1) must be completed prior to the MakePayment test case because the Make Payment form can only be accessed after the user selects “Make Reservation” on the Calendar Lookup form. | | |

|  |  |  |
| --- | --- | --- |
| **GuestReservation Entity** | | |
| **Test Objective:**  This test case will emulate a user confirming the guest’s reservation. | | |
| **Test Case #:** SERV6 |  | |
| **Test Items:** Reservation (Object), Reservation[] reservation, createReservation(Reservation), takeRoomReservation(void), RoomReservation roomReservation, “Save Reservation” button (See SDD: sec 5.4 pg. 18, sec 6.1.4 pg. 22-23) | | |
| **Input Specifications:**  Steps:   1. Verify the Guest Reservation form is shown and values are present for the reservation start and end dates. 2. Enter the appropriate price per day in text box. 3. Check (or uncheck if not paid) the guaranteed checkbox. 4. Enter the guarantee date the customer must pay by. 5. Click on “Save Reservation”. | | |
| **Output Specifications:**  If successful, the Guest Reservation form will close and the RoomReservation form will be displayed. | | |
| **Environmental needs:** | *Hardware*   * Sufficient hard disk space to store increasing amounts of data. * Working computer with a processor capable of executing multiple simultaneous commands. * Mouse * Keyboard * Monitor | *Software*   * Operating system with BBMS installed. |
|  |
| **Special Procedural Requirements:**  None | | |
| **Intercase dependencies:**  The CalendarLookup test case (SERV1) must be completed prior to the GuestReservation test case because the GuestReservation form can only be accessed after the user selects “Make Reservation” on the Calendar Lookup form. Test cases SERV2 through SERV6 must also be completed first so all of the information needed to save the reservation is entered. | | |
| **GuestReservation Entity** | | |
| **Test Objective:**  This test case will emulate a user deleting the guest’s reservation. | | |
| **Test Case #:** SERV7 |  | |
| **Test Items:** Reservation (Object), Reservation[] reservation, deleteReservation(Reservation), takeRoomReservation(void), RoomReservation roomReservation, “Delete Reservation” button (See SDD: sec 5.4 pg. 18, sec 6.1.4 pg. 22-23) | | |
| **Input Specifications:**  Steps:   1. Verify the Guest Reservation form is shown and values are present for the reservation start and end dates. 2. Click on “Delete Reservation”. | | |
| **Output Specifications:**  If successful, the Guest Reservation form will close and the system will return to the Calendar form. | | |
| **Environmental needs:** | *Hardware*   * Sufficient hard disk space to store increasing amounts of data. * Working computer with a processor capable of executing multiple simultaneous commands. * Mouse * Keyboard * Monitor | *Software*   * Operating system with BBMS installed. |
|  |
| **Special Procedural Requirements:**  None | | |
| **Intercase dependencies:**  The CalendarLookup test case (SERV1) must be completed prior to the GuestReservation test case because the GuestReservation form can only be accessed after the user selects “Make Reservation” on the Calendar Lookup form. | | |

|  |  |  |
| --- | --- | --- |
| **RoomReservation Entity** | | |
| **Test Objective:**  This test case will emulate a user updating the vacancy status of a room. | | |
| **Test Case #:** SERV8 |  | |
| **Test Items:** Room getRoom(int), createRoom(Room), Room[] rooms, “Save Room” button (See SDD: sec 5.2 pg. 16, sec 6.1.1 pg. 21) | | |
| **Input Specifications:**  Steps:   1. Verify the RoomReservation form is shown. 2. Enter a room number. 3. Check (or uncheck accordingly) the Vacant checkbox. 4. Click the “Save Room” button. | | |
| **Output Specifications:**  If successful, the RoomReservation form will close and the system will return to the Calendar form. | | |
| **Environmental needs:** | *Hardware*   * Sufficient hard disk space to store increasing amounts of data. * Working computer with a processor capable of executing multiple simultaneous commands. * Mouse * Keyboard * Monitor | *Software*   * Operating system with BBMS installed. |
|  |
| **Special Procedural Requirements:**  None | | |
| **Intercase dependencies:**  SERV1 through SERV7 test cases must be completed prior to the RoomReservation test case because the RoomReservation form can only be accessed after the user selects “Make Reservation” on the Calendar Lookup form, and all of the information has been added. | | |

|  |  |  |
| --- | --- | --- |
| **RoomReservation Entity** | | |
| **Test Objective:**  This test case will emulate a user updating the vacancy status of a room. | | |
| **Test Case #:** SERV9 |  | |
| **Test Items:** Room getRoom(int), deleteRoom(Room), Room[] rooms, “Delete Room” button (See SDD: sec 5.2 pg. 16, sec 6.1.1 pg. 21) | | |
| **Input Specifications:**  Steps:   1. Verify the RoomReservation form is shown. 2. Enter a room number. 3. Click the “Delete Room” button. | | |
| **Output Specifications:**  If successful, the RoomReservation form will close and the system will return to the Calendar form. | | |
| **Environmental needs:** | *Hardware*   * Sufficient hard disk space to store increasing amounts of data. * Working computer with a processor capable of executing multiple simultaneous commands. * Mouse * Keyboard * Monitor | *Software*   * Operating system with BBMS installed. |
|  |
| **Special Procedural Requirements:**  None | | |
| **Intercase dependencies:**  SERV1 through SERV7 test cases must be completed prior to the RoomReservation test case because the RoomReservation form can only be accessed after the user selects “Make Reservation” on the Calendar Lookup form, and all of the information has been added. | | |