**Hotel System**

**Software Test Description (STD)**

(Version 1.0)

**December 14, 2020**

**California State University Fullerton**

**Computer Science Department**

# Prepared for

**CPSC 463**

**Software Testing**

**Fall 2020**

# Prepared by

**Frank Mirando**

**Benjamin Baesu**

**Jeremy Viray**

**Tristan Bui**

# TABLE OF CONTENTS

Contents

[1.0](#_2et92p0) SCOPE 1

[1.1](#_tyjcwt) Identification 1

[1.2](#_3dy6vkm) System Overview 1

[1.3](#_1t3h5sf) Document Overview 1

[2.0](#_4d34og8) REFERENCED DOCUMENTS 2

[NOT REQUIRED 2](#_2s8eyo1)

[3.0](#_17dp8vu) TEST PREPARATIONS 3

[3.1](#_3rdcrjn) Hardware Preparation NOT REQUIRED 3

[3.1](#_26in1rg) Software Preparation NOT REQUIRED 3

[4.0](#_lnxbz9) TEST DESCRIPTIONS 4

[Table 2: Requirements Traceability Matrix - General 4](#_1y810tw)

[4.1](#_1ksv4uv) Display Fingerprint Images and Minutiae 4

[4.1.1](#_44sinio) Test Case 1: New Session with Image/CSV 4

[4.1.2](#_2jxsxqh) Test Case 2: New Session with LFFS 8

[5.0 REQUIREMENTS TRACEABILITY 17](#_z337ya)

# SCOPE

# Identification

Hotel System Version 1.0

# System Overview

The purpose of this system is to give HotelX a database with an operating UI that will help organize their concept of business. HotelX is an establishment that allows their customers to reside for however long planned. With the use of our hotel system, HotelX employees have an area to store the information of their guests. This includes the guest’s first and last name, their date of check-in and check-out, as well as their personal information. With this system, the HotelX employees have access to all of the reservations in the system along with who is staying in what room. This system has a feature that keeps track of the housekeeping service, as well as a search screen to search for guests. The system also has a daily report screen which reports the activity for the day, along with the information of the room and guest. This daily report screen also keeps track of the money made for the day, helping the budgeting for HotelX management. This system was built to allow hotel establishments to retain the data and information of their guests, while providing services and features that every hotel requires. This hotel system will be updated periodically to ensure that the features it provides continue to enhance the quality of the hotel establishment.

# Document Overview

The Software Test Description (STD) document describes the test preparations, test cases, and test procedures to be used to perform qualification testing of the software capabilities specified in the Software Requirements Descriptions

# REFERENCED DOCUMENTS

# NOT REQUIRED

# TEST PREPARATIONS

# Hardware Preparation NOT REQUIRED

This paragraph shall describe the procedures necessary to prepare the hardware for the test. Reference may be made to published operating manuals for these procedures. The following shall be provided, as applicable:

a. The specific hardware to be used, identified by name and, if applicable, number

b. Any switch settings and cabling necessary to connect the hardware

c. One or more diagrams to show hardware, interconnecting control, and data paths

d. Step-by-step instructions for placing the hardware in a state of readiness

EXAMPLE: The execution of the software does not require extensive hardware configuration or modification and can easily be loaded onto a standard desktop computer. The software requires that the computer have a minimum of 100 Megabytes (MB) of RAM and 200 MB of hard drive space. The software does not require network connectivity.

None of the test cases have additional hardware prerequisite conditions.

# Software Preparation NOT REQUIRED

This paragraph shall describe the procedures necessary to prepare the item(s) under test and any related software, including data, for the test. Reference may be made to published software manuals for these procedures. The following information shall be provided, as applicable:

a. The specific software to be used in the test

b. The storage medium of the item(s) under test (e.g., magnetic tape, diskette)

c. The storage medium of any related software (e.g., simulators, test drivers, databases)

d. Instructions for loading the software, including required sequence

e. Instructions for software initialization common to more than one test case

Example: The software is to be installed on a desktop computer running Microsoft (MS) Windows 7 operating system. The software package can be installed in any folder of the user’s choosing. The user must have write-to privileges at that folder. For test purposes, the software should be installed on the User Desktop, Programs folder, or My Documents folder.

SOFTWAREX requires MS .Net Framework 4 also be installed on the desktop prior to executing the software utility. An install package can be downloaded from MS at <http://www.microsoft.com/en-us/download/details.aspx?id=17851>.

Execution of SOFTWAREX is performed by double clicking “SOFTWAREXTool.exe” in the parent folder or right clicking and selection *Open* from the menu.

# TEST DESCRIPTIONS

Test cases for specific capabilities and features are included in later sections. System requirements not explicitly addressed by the test case, but satisfied through general installation and use of the system are included in the table below. More detail testing of the compatibility of the system with general office environments is not included in this STD.

# Display all rooms and their current status.

#### **Capability 1:** The software will allow the user to view all rooms in the hotel as clickable buttons. When clicking on a room, the user has the option to modify information regarding that room.

##### 4.2.1 Test Case 1: Check in guest

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SOFTWAREX System Version 0.1.0 Test Case** | | | | |
| **Test Case ID:** T-00 | | **Test Item: Check in guest** | | |
| **Requirements Addressed:** 1 | | | | |
| **Test Case Description:** When a room is clicked and if it is available, the window for capability 6 will display and will be able to check in a guest. | | | | |
| **Prerequisites** | | | | |
| **Test Environment:** Default | | | | |
| **Required Interfaces:** HotelX System | | | | |
| **Assumptions:** Chosen room will be available. | | | | |
| **Test Inputs:**   1. sample.py | | | | |
| **Step #** | **User Action** | | **Expected Result** | **Pass = P Fail = X** |
| 1. | Open HotelX software | | Program will display menu choices | P |
| 2. | Choose option “1’ | | All rooms will display | P |
| 3. | Click on a room with a “available” status | | Window for capability 6 will display and user can enter information for check in | P |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PASSED: FAILED: *<PASSED>* | | | | | | | |
| **Actual Results if Fail:** *<Enter the actual results if test fails>* | | | | | | | |
| **Tester’s Name:** | | | | **Date/Time:** | | **Actual time to complete:** | |
| Frank Mirando | | | | *2020-12-14*  *20:00* | | *3 minutes* | |
| **Comments:** *< Suggestions, description of possible improvements, etc.>* | | | | | | | |

# 

# Display all rooms and who is staying in each one for the next 7 days.

#### **Capability 2:** The software will display each room as clickable buttons and will display any guests that are staying in each room for the next 7 days. Clicking on each room will display information about the guest. The user can also input future reservations or check someone into a room if it is available.

##### 4.3.1 Test Case 1: Display information about a guest who is currently checked in

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SOFTWAREX System Version 0.1.0 Test Case** | | | | |
| **Test Case ID:** T-01 | | **Test Item: Display information about a guest who is currently checked in** | | |
| **Requirements Addressed:** | | | | |
| **Test Case Description: A room will be clicked and if there is a guest currently checked in then their information will be displayed.** | | | | |
| **Prerequisites** | | | | |
| **Test Environment:** Default | | | | |
| **Required Interfaces:** HotelX System | | | | |
| **Assumptions:** A guest is checked into the clicked room. | | | | |
| **Test Inputs:**   1. sample.py | | | | |
| **Step #** | **User Action** | | **Expected Result** | **Pass = P Fail = X** |
| 1. | Open HotelX software | | Menu will display | P |
| 2. | Enter “2” | | Window will display showing all the rooms and the guests that are staying in each room for the next 7 days | P |
| 3. | Click on room 1 | | The information for the guest staying in room 1 will display | P |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PASSED: FAILED: *<PASSED>* | | | | | | | |
| **Actual Results if Fail:** *<Enter the actual results if test fails>* | | | | | | | |
| **Tester’s Name:** | | | | **Date/Time:** | | **Actual time to complete:** | |
| Frank Mirando | | | | *2020-12-14*  *19:00* | | *2 minutes* | |
| **Comments:** *< Suggestions, description of possible improvements, etc.>* | | | | | | | |

# 

**4.4 Display all reservations currently in the system**

#### **Capability 3:** The software will display current reservations to the user. The user shall have the ability to Add, Delete, or convert a reservation into a check-in.

##### 4.4.1 Test Case 1: Display current reservations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hotel System Version 0.1.0 Test Case** | | | | |
| **Test Case ID:** T-02 | | **Test Item:** Display current reservations | | |
| Requirements Addressed: 3 | | | | |
| **Test Case Description:** Verify that the current reservations are displayed when the user chooses it from the menu. | | | | |
| **Prerequisites** | | | | |
| **Test Environment:** Default | | | | |
| **Required Interfaces:** HotelX System | | | | |
| **Assumptions:** N/A | | | | |
| **Test Inputs:** sample.py | | | | |
| **Step #** | **User Action** | | **Expected Result** | **Pass = P Fail = X** |
| 1. | Run the program | | Program displays menu | P |
| 2. | Enter “3” in the menu | | A pop-up window will appear displaying all current reservations | P |

|  |  |  |  |
| --- | --- | --- | --- |
| **Results** | | | |
| PASSED: 1, 2 | FAILED: | *Test Passed* | |
| **Actual Results if Fail:** *n/a* | | | |
| **Tester’s Name:** | | **Date/Time:** | **Actual time to complete:** |
| Jeremy Viray | | *2020-12-14* | *2 minutes* |
|  | | *09:00* |  |
|  | |  |  |
| **Comments:** *< Suggestions, description of possible improvements, etc.>* | | | |

##### 4.4.2 Test Case 2: Add New Reservation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hotel System Version 0.1.0 Test Case** | | | | |
| **Test Case ID:** T-03 | | **Test Item:** Add new reservation | | |
| **Requirements Addressed:** 3 | | | | |
| **Test Case Description:** Verify that a new session reservation can be created in they system | | | | |
| **Prerequisites** | | | | |
| **Test Environment:** Default | | | | |
| **Required Interfaces:** HotelX System | | | | |
| **Assumptions:** N/A | | | | |
| **Test Inputs:** sample.py | | | | |
| **Step #** | **User Action** | | **Expected Result** | **Pass = P Fail = X** |
| 1. | Run the program | | Program displays menu | P |
| 2. | Enter “3” in the menu | | A pop-up window will appear displaying all current reservations | P |
| 3. | Click *“Add Reservation”* | | User is prompted to enter reservation information in the console. | P |
| 4. | Type in required information | | System saves reservation information and instructs user to close window | P |
| 5. | Close window | | System updates all information and returns user to main menu | P |

|  |  |  |  |
| --- | --- | --- | --- |
| **Results** | | | |
| PASSED: 1,2,3,4 | FAILED: | *Test Passed* | |
| **Actual Results if Fail:** N/A | | | |
| **Tester’s Name:** | | **Date/Time:** | **Actual time to complete:** |
| Jeremy Viray | | *2020-12-14* | *20 minutes* |
|  | | *09:00* |  |
|  | |  |  |
| **Comments:** *< Suggestions, description of possible improvements, etc.>* | | | |

**4.5 Display housekeeping screen to manage housekeeping**

#### **Capability 4:** The software will display a list of all dirty and occupied rooms. The user shall have the ability to click on a room to update the cleaning list. The user shall also have the ability to change the status of the room.

##### **4.5.1 Test Case 1: Display all dirty and occupied rooms**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hotel System Version 0.1.0 Test Case** | | | | |
| **Test Case ID:** T-04 | | **Test Item:** Display all dirty and occupied rooms | | |
| Requirements Addressed: 4 | | | | |
| **Test Case Description:** Verify that all the dirty and occupied rooms are displayed when the user chooses it from the menu. | | | | |
| **Prerequisites** | | | | |
| **Test Environment:** Default | | | | |
| **Required Interfaces:** HotelX System | | | | |
| **Assumptions:** N/A | | | | |
| **Test Inputs:** sample.py | | | | |
| **Step #** | **User Action** | | **Expected Result** | **Pass = P Fail = X** |
| 1. | Run the program | | Program displays menu | P |
| 2. | Enter “4” | | A pop-up window will appear displaying all dirty or occupied rooms, the status of each cleaning list item, and overall status. | P |

|  |  |  |  |
| --- | --- | --- | --- |
| **Results** | | | |
| PASSED: 1, 2 | FAILED: | *Test Passed* | |
| **Actual Results if Fail:** N/A | | | |
| **Tester’s Name:** | | **Date/Time:** | **Actual time to complete:** |
| Jeremy Viray | | *2020-12-14* | *2 minute* |
|  | | *09:00* |  |
|  | |  |  |
| **Comments:** *< Suggestions, description of possible improvements, etc.>* | | | |

##### **4.5.2 Test Case 2: Update cleaning List and room status**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hotel System Version 0.1.0 Test Case** | | | | |
| **Test Case ID:** T-05 | | **Test Item: Update cleaning List and room status** | | |
| Requirements Addressed: 4 | | | | |
| **Test Case Description:** Verify that the user is able to update the cleaning checklist and room status | | | | |
| **Prerequisites** | | | | |
| **Test Environment:** Default | | | | |
| **Required Interfaces:** HotelX System | | | | |
| **Assumptions:** N/A | | | | |
| **Test Inputs:** sample.py | | | | |
| **Step #** | **User Action** | | **Expected Result** | **Pass = P Fail = X** |
| 1. | Run the program | | Program displays menu | P |
| 2. | Enter “4” | | A pop-up window will appear displaying all dirty or occupied rooms, the status of each cleaning | P |
| 3. | Click on a room number to edit cleaning list | | The user will be shown the information for that room and be prompted to check off cleaning list items | P |
| 4. | Check off cleaning list items or leave blank | | The System will record the values for each list. Then ask the user if they would like to change the room status | P |
| 5. | Change status or leave it as is | | The System will record any input and prompt user to close the window | P |
| 6. | Close window | | System updated all information and returns user to main menu | P |

|  |  |  |  |
| --- | --- | --- | --- |
| **Results** | | | |
| PASSED: 1,2,3,4,5,6 | FAILED: | *Test Passed* | |
| **Actual Results if Fail:** N/A | | | |
| **Tester’s Name:** | | **Date/Time:** | **Actual time to complete:** |
| Jeremy Viray | | *2020-12-14* | *25 minutes* |
|  | | *09:00* |  |
|  | |  |  |
| **Comments:** *< Suggestions, description of possible improvements, etc.>* | | | |

**4.6 Display guest profile screen to show guest information**

#### **Capability 5:** The software will display the profile information of a specified guest. The user shall have the ability to click on a guest to view their profile which includes their personal information and room number.

##### 4.6.1 Test Case 1: Display guest information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SOFTWAREX System Version 0.1.0 Test Case** | | | | |
| **Test Case ID:** T-06 | | **Test Item: Display guest information** | | |
| Requirements Addressed: 3 | | | | |
| **Test Case Description:** View and verify guest information of a specified guest | | | | |
| **Prerequisites** | | | | |
| **Test Environment:** Default | | | | |
| **Required Interfaces:** SOFTWAREX System | | | | |
| **Assumptions:** N/A | | | | |
| **Test Inputs:** Files located in the “\SOFTWAREXTool\TestData” folder   1. SoftwareX\_test\_G093T8U\_08.bmp, SoftwareX\_test\_G093T8U\_08.csv 2. SoftwareX\_test\_G095T6U\_06.bmp, SoftwareX\_test\_G095T6U\_06.csv 3. SoftwareX\_test\_G096T2U\_02.bmp, SoftwareX\_test\_G096T2U\_02.csv | | | | |
| **Step #** | **User Action** | | **Expected Result** | **Pass = P Fail = X** |
| 1. | Run the program | | Program displays menu | P |
| 2. | Enter “5” | | A pop-up window will appear titled “Guest Profile”. | P |
| 3. | Click  *Name.* | | A pop-up window will appear titled “Guest Profile”. | P |
| 4. | Enter new name | | Text box will fill up with the new typed name. | P |
| 5. | Press *Update* | | The edit window closes and name changes to edited name. | P |
| 6. | Press ID Photo | | A pop-up window opens and shows guest ID photo. | P |
| 7. | 1. Click *File* 🡪 *Close Window.* | | The image display clears and resets to the initial SOFTWAREX state. | P |

|  |  |  |  |
| --- | --- | --- | --- |
| **Results** | | | |
| PASSED: 1,2,3,4,5,6, 7 | FAILED: | *Test Passed* | |
| **Actual Results if Fail:** N/A | | | |
| **Tester’s Name:** | | **Date/Time:** | **Actual time to complete:** |
| Benjamin Baesu | | *2020-12-14* | *25 minutes* |
|  | | *09:00* |  |
|  | |  |  |

**4.7 Display stay screen to show a guest’s information**

#### **Capability 6:** The software will display the stay information of a specified guest. The user shall have the ability to click on an attribute to update the attribute. The user shall also be able to change check-in and check-out times of a guest.

##### 4.7.1 Test Case 1: Display stay information of specified guest

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SOFTWAREX System Version 0.1.0 Test Case** | | | | |
| **Test Case ID:** T-07 | | **Test Item: Display stay information of specified guest** | | |
| Requirements Addressed: 3 | | | | |
| **Test Case Description:** Verify that the current reservations are displayed when user chooses it from the menu. | | | | |
| **Prerequisites** | | | | |
| **Test Environment:** Default | | | | |
| **Required Interfaces:** SOFTWAREX System | | | | |
| **Assumptions:** N/A | | | | |
| **Test Inputs:** Files located in the “\SOFTWAREXTool\TestData” folder   1. SoftwareX\_test\_G093T8U\_08.bmp, SoftwareX\_test\_G093T8U\_08.csv 2. SoftwareX\_test\_G095T6U\_06.bmp, SoftwareX\_test\_G095T6U\_06.csv 3. SoftwareX\_test\_G096T2U\_02.bmp, SoftwareX\_test\_G096T2U\_02.csv | | | | |
| **Step #** | **User Action** | | **Expected Result** | **Pass = P Fail = X** |
| 1. | Run the program | | Program displays menu | P |
| 2. | Enter “6” | | A pop-up window will appear titled “Guest Stay Information”. | P |
| 3. | Click *up on check-out.* | | The check-out date will change by one day. | P |
| 4. | Click a name. | | A pop-up window will appear titled “Guest Stay Information”. | P |
| 5. | Enter a new name | | Name will appear in-text box. | P |
| 6. | Press *update*. | | Name will update in Guest Stay Information window and pop-up window will close. | P |
| 11. | Click *Next.* | | The window will advance, requesting that  the user review the session inputs. | P |
| 13. | 1. Click *File* 🡪 *Close Window.* | | The display clears and resets to the initial SOFTWAREX state. | P |

|  |  |  |  |
| --- | --- | --- | --- |
| **Results** | | | |
| PASSED: 1,2,3,4,5,6 | FAILED: | *Test Passed* | |
| **Actual Results if Fail:** N/A | | | |
| **Tester’s Name:** | | **Date/Time:** | **Actual time to complete:** |
| Benjamin Baesu | | *2020-12-14* | *25 minutes* |
|  | | *09:00* |  |
|  | |  |  |

**4.8 Search screen to search for guests**

#### **Capability 7:** The software will display a search screen to search for guests. The user shall have the ability to search for a guest through this search screen. The user shall also have the ability to select the guests name which displays the screen of the guest information.

##### **4.8.1 Test Case 1: Search for a guest**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **HotelX System Version 0.1.0 Test Case** | | | | |
| **Test Case ID:** T-08 | | **Test Item: Search for a guest** | | |
| Requirements Addressed: | | | | |
| **Test Case Description:** Verify that the user is able to view the search screen and use it to access guest information | | | | |
| **Prerequisites** | | | | |
| **Test Environment:** Default | | | | |
| **Required Interfaces:** HotelX System | | | | |
| **Assumptions:** N/A | | | | |
| **Test Inputs:** | | | | |
| **Step #** | **User Action** | | **Expected Result** | **Pass = P Fail = X** |
| 1. | Run the program | | Program displays menu | P |
| 2. | Enter “7” | | Program displays menu on which option to search for a guest | P |
| 3. | Enter “1” | | Program will allow user to search by “First Name” | P |
| 4. | Enter “2” | | Program will allow user to search by “Last Name” | P |
| 5. | 1. Enter “3” | | Program will allow user to search by “Room Number” | P |
| 6. | 1. Enter “4” | | Program will allow user to search by “Phone Number” | P |
| 7. | Enter “5” | | Program will allow user to search by “Street Address” | P |
| 8. | Enter “6” | | Program will allow user to search by “Check In Date” | P |
| 9. | Enter “7” | | Program will allow user to search by “Check Out Date” | P |
| 10. | Enter search value based on parameter chosen | | Program will display result of search value with a list of guests matching the value | P |
| 11. | Enter numerical value of the search result | | Pop-up window will appear displaying Guest Information | P |
| 12. | Select the guest’s name | | The program will display the screen from Capability 5 (Shows all guest information) | P |
| 13. | Close the pop-up window | | Program will return user back to the menu where user can choose from options 1-8 | P |

**4.9 Display a daily report screen**

#### **Capability 8:** The software will display a daily report screen that reports the day’s activity. The user shall have the ability to view the list of rooms rented that day and have the room information as well as the guest information. It will also display the amount of total dollars paid for rooms that day.

##### **4.9.1 Test Case 1: View daily report screen**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **HotelX System Version 0.1.0 Test Case** | | | | |
| **Test Case ID:** T-09 | | **Test Item: View daily report screen** | | |
| Requirements Addressed: 3 | | | | |
| **Test Case Description:** View and verify that the current reservations are displayed when the user chooses it from the menu. | | | | |
| **Prerequisites** | | | | |
| **Test Environment:** Default | | | | |
| **Required Interfaces:** HotelX System | | | | |
| **Assumptions:** N/A | | | | |
| **Test Inputs:** | | | | |
| **Step #** | **User Action** | | **Expected Result** | **Pass = P Fail = X** |
| 1. | Run the program | | Program displays menu | P |
| 2. | Enter “8” | | A pop-up window will appear displaying the daily report for HotelX | P |
| 3. | User views the report for the day | | User can record how much money made total within the day | P |
| 4. | Click the “Quit back to Main Menu” button | | Pop-up window will close and will navigate user back to program menu where user can choose options 1-8. | P |

|  |  |  |  |
| --- | --- | --- | --- |
| **Results** | | | |
| PASSED:  1, 2, 3, 4 | FAILED: | *Test Passed* | |
| **Actual Results if Fail:** *<Enter the actual results if test fails>* | | | |
| **Tester’s Name:** | | **Date/Time:** | **Actual time to complete:** |
| *Tristan Bui* | | *2020-12-14* | *15 minutes* |
|  | | *09:30* |  |
|  | |  |  |
| **Comments:** *< Suggestions, description of possible improvements, etc.>* | | | |

# 5.0 REQUIREMENTS TRACEABILITY

**Table 5: Requirements to Test Case**

|  |  |  |
| --- | --- | --- |
| **Req ID**  **No.** | **Requirement** | **Test Case** |
| 1 | Screen that shows all rooms and their current status. | T-00 |
| 2 | Screen showing a list of the rooms and who is staying in the room for each day for the next 7 days. | T-01 |
| 3 | A reservation screen showing a list of all reservations currently in the system | T-02, T-03 |
| 4 | A housekeeping screen to manage housekeeping | T-04, T-05 |
| 5 | A guest profile screen to show guest information | T-06 |
| 6 | Current stay screen showing a guest’s information for their current stay. | T-07 |
| 7 | A search screen to search for guests | T-08 |
| 8 | A daily report screen | T-09 |

**Table 6: Test Case to Requirements**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Test Case Name** | **Req. ID**  **No.** |
| T-00 | Check in guest | 1 |
| T-01 | Display information about a guest who is currently checked in | 2 |
| T-02 | Display current reservations | 3 |
| T-03 | Add New Reservation | 3 |
| T-04 | Display all dirty and occupied rooms | 4 |
| T-05 | Update cleaning List and room status | 4 |
| T-06 | Display guest information | 5 |
| T-07 | Display stay information of specified guest | 6 |
| T-08 | **Search for a guest** | 7 |
| T-09 | **View daily report screen** | 8 |