**Software Engineering Principles and Techniques (2178)**

**CMIS 330 6380**

**University of Maryland University College**

**29-Sep 2017**

**Software Test Specification for B&B System- Assignment #3**

**To**

**Pro. Dave Gabello**

**By**

**Binyam Woldesenbet & Bisrat Tadesse**

**Table of Content**

1. Introduction
2. Purpose
3. Detailed Test Environment Description
   1. Hardware
   2. Software
   3. Others
4. White-Box Test Cases
   1. Test Case # 1
      1. Test Objective
      2. Test Items
      3. Input Specification
      4. Output Specification
      5. Environmental Needs
         1. Hardware
         2. Software
         3. Others
      6. Special Procedural Requirements
      7. Execution Procedure Steps
      8. Inter-case Dependencies
   2. Test Case # 2
      1. Test Objective
      2. Test Items
      3. Input Specification
      4. Output Specification
      5. Environmental Needs
         1. Hardware
         2. Software
         3. Others
      6. Special Procedural Requirements
      7. Execution Procedure Steps
      8. Inter-case Dependencies

5. Black Box Test Cases

5.1 Test Case # 1

5.1.1 Test Objective

5.1.2 Test Items

5.1.3 Input Specification

5.1.4 Output Specification

5.1.5 Environmental Needs

5.1.5.1 Hardware

5.1.5.2 Software

5.1.5.3 Others

5.1.6 Special Procedural Requirements

5.1.8 Execution Procedure Steps

5.1.7 Inter-case Dependencies

5.2 Test Case # 2

5.2.1 Test Objective

5.2.2 Test Items

5.2.3 Input Specification

5.2.4 Output Specification

5.2.5 Environmental Needs

5.2.5.1 Hardware

5.2.5.2 Software

5.2.5.3 Others

5.2.6 Special Procedural Requirements

5.2.7 Execution Procedure Steps

5.2.8 Inter-case Dependencies

6. Traceability Matrix

1. **Introduction**

This Software Test Specification document is produced for the B&B system. The major components from the Architectural Context Diagram(ACD) are selected based on the use cases and software requirements set in the SRS and SDD produced earlier. It contains the component set of Associated white and black box test case definitions.

1. **Purpose**

This Software Test Specification is written to guide the B&B system test operation to ensure that all the specifications in the requirements document are followed, and to validate the software product.

1. **Detailed test Environment**

**3.1 Hardware**

For this software test, there is no need of using any CRT software, since the B&B system is used as a desktop application. The PC that is going to be used for this test is the same machine that is going to be used as the B&B system platform. The cash register, the card reader, the printer, the fax machine and other support hardware must be present and connected to the platform.

**3.2 Software**

The PC is assumed to run on windows 10 operating system, and must be configured with all drivers for each connected hardware. The PC must have a Tomcat server installed, which in turn must be linked to a Spring Tool Suit to run the application.

* 1. **Others**

The test does not need a unique environment but some trained personnel, who have a thorough understanding of OOP, and MVC framework are required. The personnel must also be familiar with the logic and implementation of the software.

1. **White-Box Test Cases**

**4.1 Test Case #1**

**4.1.1 Test Objective**

To test the instruction logic in the getName method of the Customer object function properly.

**4.1.2 Test Items**

The Customer class, its constructor, and the getter that returns the customer full name are all the items to be tested in this test case. The requirements and interface of the Customer object can be found in figure 4.1 of the SDD.

**4.1.3 Input Specifications**

The input required to execute this test case is of type String. No databases, files, terminal messages, or values passed by the operating system is required.

**4.1.4 Output Specification**

The output must be the customer full name. The type is String, and must be displayed on the console. For no inputs, the program should display “Customer not available!” on the console. Table 4.1 depicts the white-box test case #1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | **Expected Result** | **Actual Result** |
| **#** | **Input** |
| 1 | "Customer Name" | Customer Name |  |
| 1.1 | "" | "Not Valid" |  |

**4.1.5 Environmental needs**

**4.1.5.1 Hardware**

For this test, the main application platform or PC is used. No other hardware components

Required.

**4.1.5.2 Software**

Windows 10 operating system is assumed to be installed on the PC. Tools or IDEs like

Eclipse, or Spring Tool Suite(STS) must be installed on the PC to run the program.

**4.1.5.3 Others**

There is no need for a unique environmental facility, but trained personal who understand Object oriented programming, and the logic and implementation of the B&B system are

Required.

**4.1.6 Special Procedural Requirements**

Before this test is conducted, the customer class must have a driver or main method, or a

Junit test cases should be developed. No special set up or warm up is needed.

**4.1.7 Execution Procedure Steps**

An external excel program is used to log the results of the test executions. The customer

Object must be created, before this test procedure is carried out. The Customer object

must be created by running the program either on eclipse or spring. The setName()

method of the customer object then should be invoked before the getName() method.

**4.1.8 inter-case Dependencies**

Test case #2 must be executed before this test case. Test case #2 is used to test the logic

Of the setName() method of the Customer object. Until this method is validated, all the

results of test case #1 are inconclusive.

* 1. **Test case #2**

**4.1.1 Test Objective**

To test the instruction logic in the setName method of the Customer object function properly.

**4.1.2 Test Items**

The Customer class, its constructor, and the setter that sets the customer full name are all the items to be tested in this test case. The requirements and interface of the Customer object can be found in figure 4.1 of the SDD.

**4.1.3 Input Specifications**

The input required to execute this test case is of type String. No databases, files, terminal messages, or values passed by the operating system is required.

**4.1.4 Output Specification**

The output must be the customer full name. The type is String, and must be displayed on the console. For an empty argument, the program should display “Null value is not allowed!” on the console. Table 4.1 depicts the white-box test case #2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | **Expected Result** | **Actual Result** |
| **#** | **Input** |
| 2 | "Customer Name" | Customer Name |  |
| 2.1 | "" | "Null value is not accepted" |  |

**4.1.5 Environmental needs**

**4.1.5.1 Hardware**

For this test, the main application platform or PC is used. No other hardware components

Required.

**4.1.5.2 Software**

Windows 10 operating system is assumed to be installed on the PC. Tools or IDEs like

Eclipse, or Spring Tool Suite(STS) must be installed on the PC to run the program.

**4.1.5.3 Others**

There is no need for a unique environmental facility, but trained personal who understand Object oriented programming, and the logic and implementation of the B&B system are

Required.

**4.1.6 Special Procedural Requirements**

Before this test is conducted, the customer class must have a driver or main method, or a

Junit test cases should be developed. No special set up or warm up is needed.

**4.1.7 Execution Procedure Steps**

An external excel program is used to log the results of the test executions. The customer

Object must be created, before this test procedure is carried out. The Customer object

must be created by running the program either on eclipse or spring. The setName()

method of the customer object then should be invoked.

**4.1.8 Inter-case Dependencies**

This test case has no inter-case dependencies, but it requires that the Customer object

To created before invoking the setName() method.

1. **Black-Box Test Cases**

**5.1 Black-Box Test Case #1**

**5.1.1 Test Objective**

To test login function of the system when a user is trying to use the B&B application.

**5.1.2 Test Items**

The functionality of the login process for the **user component** of the ACD (Figure 1 of the SDD) presented previously is to be tested in this case. This is also discussed in the **Use Case 1.0 and User scenario** 1 in table 1 and 2 respectively of the SRS.

**5.1.3 Input Specifications**

The inputs required to execute this test case are of two type Strings. These inputs are reference to the user name and password of a user to login. No databases, files, terminal messages, or values passed by the operating system is required except an indication of input names as in user name and password when the UI is displayed.

**5.1.4 Output Specification**

The output must be the transfer (opening) to the operations option page of the B&B system up on the input of the correct user name and password combinations. For wrong combinations, the program should display “Wrong username/ password” message. If either field is left blank, it displays "username/password cannot be blank".

**Table 5.1**

**Black-Box Test Case 1 for B&B System**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | | | **Expected result** | **Actual result** |
| **#** | **User Name** | **Password** |
| 1 | "correct user name" | "correct pw" | Display options |  |
| 1.1 | "wrong user name" | "correct pw" | Error message |  |
| 1.2 | "correct user name" | "Wrong pw" | Error message |  |
| 1.3 | "user name" | " " | Error message |  |
| 1.4 | " " | "pass word" | Error message |  |
| 1.5 | " " | " " | Error message |  |

**5.1.5 Environmental needs**

**5.1.5.1 Hardware**

For this test, the main application platform or PC is used. No other hardware components Required.

**5.1.5.2 Software**

Windows 10 operating system is assumed to be installed on the PC. Tools or IDEs like Eclipse, or Spring Tool Suite(STS) must be installed on the PC to run the program.

**5.1.5.3 Others**

There is no need for a unique environmental facility for this test case. Just a person with a given privilege of a user name and password so as to be able to test the case.

**5.1.6 Special Procedural Requirements**

For this test case , the user class should establish a database connectivity with the database so as to check the user privileges. This connectivity should be first established using JDBC(Java Database Connectivity) procedures.

**5.1.7 Execution Procedure Steps**

The GUI for input of username and the password will be displayed for the user to enter. The user can put in any kind of combinations of username and password including the correct ones given to him/her by the administrator. Then, the tester records the inputs and outputs on an external table template that is prepared to log the results of the test executions.

In order to allow a more exhaustive test, the limit of log in trial that is implemented in the system shall be disabled for this test case.

**5.1.8 inter-case Dependencies**

There is no inter-case dependencies in order to proceed with this test case. However, this test case is helpful in order to test other procedures of the system like session control.

**5.2 Black--Box Test Case #2**

**5.2.1 Test Objective**

To test the Calendar functionality of the system before the user starts a new reservation

**5.2.2 Test Items**

The functionality of the Calendar class of the system so as to be able to determine if the system correctly identify open days for a room to be reserved. This is part of both the **Use Case 1.0 and User scenario** 1 in table 1 and 2 respectively of the SRS.

**5.2.3 Input Specifications**

The inputs required to execute this test case is a click on the Calendar button of the GUI displayed after new reservation option is chosen.

**5.2.4 Output Specification**

The output must be the display of the calendar GUI for the current month with the room numbers listed with the days of the month. The days that the room is already booked for will be grayed out and should not be clickable. The days the room is available are clickable for further processing in the reservation. If a day for which a room is already booked is clicked on the system should display an error message. Otherwise the day will be included for reservation.

**Table 5.2**

**Black-Box Test Case 2 for B&B System**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | **Expected result** | **Actual result** |
| **#** | **Calendar** |
| 2 | Calendar button clicked | Display current month calendar |  |
| 2.1 | Grayed button/booked button clicked | "Room is already reserved" message |  |
| 2.2 | Blue button/open for booking button | "Room is open for reservation" message |  |

**5.2.5 Environmental needs**

**5.2.5.1 Hardware**

For this test, the main application platform or PC is used. No other hardware components Required.

**5.2.5.2 Software**

Windows 10 operating system is assumed to be installed on the PC. Tools or IDEs like Eclipse, or Spring Tool Suite(STS) must be installed on the PC to run the program.

**5.2.5.3 Others**

There is no need for a unique environmental facility for this test case. Just a person with a given privilege of a user name and password so as to be able to test the case.

**5.2.6 Special Procedural Requirements**

For this test case , the Calendar class should have a driver to run up on invoking and display the GUI.

**5.2.7 Execution Procedure Steps**

A clickable button with a value of "calendar" should be displayed. Then, the tester records the displays and messages on an external table template that is prepared to log the results of the test executions.

**5.2.8 inter-case Dependencies**

The inter-case dependency for this test case is a case with status showing for the room object(Class). In order to be able to check the rooms availability, the Calendar class shall have a Room object that checks the status of each room.

**6. Traceability Matrix**

The following two tables show the traceability matrix for the White-Box (Table 6.1) and the Black-Box(Table 6.2)

**Table 6.1  
Traceability Matrix for the White-Box test cases of the B&B System**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Category** | **Description** | **System Req. #** | **Use Case #** | **Software Req. #** | **Test Case #** | **Pass/Fail** |
| Customer class | getName | - | 1.0 | 1.1 | 1.1 |  |
| Customer class | setName | - | 1.0 | 1.1 | 2.1 |  |

**Table 6.2  
Traceability Matrix for the Black-Box test cases of the B&B System**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Category** | **Description** | **System Req. #** | **Use Case #** | **Software Req. #** | **Test Case #** | **Pass/Fail** |
| New Login | Log in | 3.1 | 1.0 | 3.1.3 | 1 |  |
| New Reservation | Reservation | 3.1 | 1.0 | 3.1 | 2 |  |