Software Test Specification

Tiffany Wise

University of Maryland University College

CMIS 330 – Section 7980

Professor Abdulnasir Shash

Table of Contents

1. Introduction

The Test Specification Document details the features to be tested and the steps to test the features of the bed and breakfast system against the software requirements and software design documents.

2. References

IEEE 829 - 1998

3. Test plan

3.1. Purpose

The test plan describes the specific features to be tested in this specification.

3.2. Components to be tested

Testing for the Reservation system and Accounting subsystems of the bed and breakfast software will be provided.

3.2.1. Reservation System

The use cases in the initial Software Requirements Specification document have been expanded to include the use cases for the Reservation system, listed in Table 1.0.

Use Case	*Test case specification number*	*Description*
Search for Vacancies	TC 1.0	Query Reservation database for vacancies
	TC 1.1	Room Selection
	TC 1.2	Entering Customer Information
Process Reservation	TC 1.3	Create Reservation
	TC 1.4	Hold Reservation without Reservation Guarantee
	TC 1.5	Hold Reservation with Reservation Guarantee

Table 1.0. Use Cases for Reservation System

3.2.2. Accounting System

Additional use cases for the Accounting System are listed in Table 2.0

Use Case	*Test case specification number*	*Description*
Search for Financial Data	TC 2.0	Query Accounting database for financial records
	TC 2.1	View Financial Records
	TC 2.2	Print Financial Report
Financial Data Entry	TC 2.3	Enter Expense Information
	TC 2.4	Enter Income Information

3.3. Components not to be tested

The following components of the bed and breakfast software program will be tested by subcontractors:

<u>Payment System</u>

- Card reader
- Key pad reader
- Interface between card reader and reservation system
- Interface between key pad reader and reservation system

<u>Database System</u>

- Reservation database internal configuration
- Accounting database internal configuration
- Interface between the Reservation database and the Accounting database

<u>System Authentication</u>

- Logging into the Reservation system by authorized users
- Logging into the Accounting system by authorized users
- Database authentication

3.4. Approach
3.5. Test Deliverables
3.6. Detailed Test Environment

4. Test case specification

4.1. Purpose

The following test cases provide steps to ensure that the program meets its design and requirements specification

4.2. Test Case Outline

Each test case with have the following information:

- Objective
- Test case specification identifier
- Input specification
- Output specification
- Environment Conditions
- Procedural Requirements
- Procedure Steps
- Dependencies

4.3. Reservation System, Use Case 1.0: Search for Vacancies

4.3.1. Test Case 1.1 Query Reservation Database

Objective

The customer wants information on vacancies available from a starting to an ending date. The end-user enters these dates into the system and tells the customers which rooms are available.

Test case specification identifier

Test Case 1.0 Query Reservation database for vacancies

Input Specification

A starting date and ending date must be entered into the reservation software. For one-day reservations, the ending date must be the same as the starting date.

Output Specifications

The results from the Reservation database are displayed on the GUI for the bed and breakfast software. For the date(s) entered, only the rooms that are vacant are displayed. A checkbox appears next to each vacant room on the GUI for selection by the end-user at the customer’s request.

Environment Conditions

Dependencies

- The end-user must be authorized to use the software and logged into the system.
- The bed and breakfast software must be connected to an internal and external network that provides access to the Reservation database

Procedural Requirements

For Test Number TN 1.0 – 4, Creation of ReservationDate object, no additional input is required from the end-user. The ReservationDate object is created when two validated dates are entered into the GUI by the end-user.

For Test Number TN 1.0 – 5, Sending ReservationDate object to database, no additional input is required from the end-user. This step should occur automatically.

Procedural Steps

Procedural Steps for TC 1.0 **Query Reservation database for vacancies**			
White-Box Testing			
Test Number	**Description**	**Input**	**Expected Output**
TN 1.0 - 1	Enter start date into Start Date textfield	Numbers (acceptable format XX/XX/XXXX where X=number)	Date saved in Date object
TN 1.0 – 2	Enter incorrectly formatted start date into Start Date textfield	Incorrect format	Error message generated
TN 1.0 - 3	Enter start date into End Date textfield	Numbers (acceptable format XX/XX/XXXX where X=number)	Date saved in Date object
TN 1.0 – 4	Enter incorrectly formatted start date into End Date textfield	Incorrect format	Error message generated
TN 1.0 – 5	ReservationDate object	Two Date objects for starting and ending date. No input from	Creation of ReservationDate object
TN 1.0 – 6	ReservationDate object sent to Reservation database	End-user selects Search button	ReservationDate object sent to database
TN 1.0 – 7	Reservation database responds to query	Reservation data information	Vacant rooms and room guarantee

			prices returned from database

<table>
<tr><th colspan="4"><h1>Procedural Steps for TC 1.0</h1><b>Query Reservation database for vacancies</b></th></tr>
<tr><th colspan="4">Black-Box Testing</th></tr>
<tr><th>Test Number</th><th>Description</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>TN 1.0 - 1</td><td>End-user enters valid start date</td><td>Valid Start Date (XX/XX/XXX)</td><td>No error message on GUI</td></tr>
<tr><td>TN 1.0 – 2</td><td>End-user enters invalid start date</td><td>Invalid Start Date</td><td>Error message displays in GUI</td></tr>
<tr><td>TN 1.0 - 3</td><td>End-user enters valid end date</td><td>Valid End Date (XX/XX/XXX)</td><td>No error message on GUI</td></tr>
<tr><td>TN 1.0 – 4</td><td>End-user enters invalid end date</td><td>Invalid End Date</td><td>Error message displays in GUI</td></tr>
<tr><td>TN 1.0 – 5</td><td>End-user clicks Search button</td><td>Button click on Search button</td><td>Vacant rooms, reservation guarantee prices display in GUI with checkboxes for selection</td></tr>
</table>

4.3.2. Test Case 1.1 Room Selection

Objective

Customer chooses an available room for the results returned from the database. The end-user makes the selection in the software program and starts a new reservation

Test Case Specifier

Test Case 1.1 Room Selection

Input Specification

The end-users selects a checkbox next to a vacant room and clicks a Reservation button

Output Specification

A child of the Room object class is created depending on which room is selected. The child classes are RoomOne, RoomTwo, RoomThree, and RoomFour.

Dependencies

Test Case 1.1 is depended on the successful completion of the white- and black-box testing completing with Test Case 1.0.

Special Environmental Conditions

None

Procedural Requirements

None

Procedural Steps

<table>
<tr><th colspan="4">Procedural Steps for TC 1.1
Room Selection</th></tr>
<tr><th colspan="4">White-Box Testing</th></tr>
<tr><th>Test Number</th><th>Description</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>TN 1.1 – 1</td><td>End-user selects a vacant room</td><td>Checkbox selected for a vacant room</td><td>Room number is used to determine which child object of Room class to create</td></tr>
<tr><td>TN 1.1 – 2</td><td>End-user clicks Begin Reservation button</td><td>Button click for Begin Reservation button</td><td>Child Room object is created.</td></tr>
</table>

Procedural Steps for TC 1.1

Room Selection

Black-Box Testing

Test Number	Description	Input	Expected Output
TN 1.1 – 1	End-user selects a vacant room	Checkbox selected for a vacant room	Checkbox appears in checkbox next to selected room
TN 1.1 – 2	End-user clicks Begin Reservation button	Button click for Begin Reservation button	GUI enables a disabled textboxes to allow for entering customer information

4.3.3. Test Case 1.2 Entering Customer Information

Objective

The customer provides a first and last name, street address, city, state, and zip code, as well as a phone number. The end-user enters this information into the software. This information will be saved in a Customer object.

Test Case Specifier

Test Case 1.2

Input Specification

Input is received from the keyboard via Scanner objects.

Output Specification

Customer object is created containing the customer's information.

Dependencies

- End-user must enter valid input

Special Environmental Conditions

None

Procedural Requirements

None

Procedural Steps

Procedural Steps for TC 1.2 Entering Customer Information			
White-Box Testing			
Test Number	**Description**	**Input**	**Expected Output**
TN 1.2 – 1	End-user enters invalid customer information	Name, invalid customer address (i.e. incorrect zip code format), invalid phone number	Error message generated
TN 1.2 – 2	End-user enters valid customer information	Name, valid address, phone number	String objects created and added to Customer object
TN 1.2 – 3	End-user clicks Save Customer Data button	Button click	Button click event generated and string objects added to Customer object
TN 1.2 – 4	Information from Room and ReservationDate objects transferred	(None from end-user)	Reservation object created
TN 1.2 – 5	Reservation object sent to Reservation database	(None from end-user)	Reservation database accepts object and generates Reservation ID and Customer ID numbers
TN 1.2 – 6	Reservation and Customer ID numbers added to respective objects	(None from end-user)	Reservation ID added to Reservation object. Customer ID added to Customer object.

Procedural Steps for TC 1.2 Entering Customer Information Black-Box Testing			
Test Number	**Description**	**Input**	**Expected Output**
TN 1.2 – 1	End-user enters invalid customer information	Name, invalid customer address (i.e. incorrect zip code format), invalid phone number	Error message appears on GUI display.
TN 1.2 – 2	End-user enters valid customer information	Name, valid address, phone number	None visible to end-user
TN 1.2 – 3	End-user clicks Save Customer Data button	Button click	Button depresses

4.4. Reservation System, Use Case 1.1 Process Reservation

4.4.1. Test Case 1.3 Create Reservation

Objective

Customer information, reservation dates, and room number and pricing are used to create a reservation.

Test Case Specifier

Test Case 1.3

Input Specification

Output Specification

Dependencies

Special Environmental Conditions

Procedural Requirements

Procedural Steps

Procedural Steps for TC 1.3

Create Reservation

White-Box Testing

Test Number	Description	Input	Expected Output
TN 1.3 – 1	Information from Customer, Room, and ReservationDate objects transferred	(None from end-user)	Reservation object created
TN 1.3– 2	Reservation object sent to Reservation database	(None from end-user)	Reservation database accepts object and generates Reservation ID and Customer ID numbers
TN 1.3 – 3	Reservation and Customer ID numbers added to respective objects	(None from end-user)	Reservation ID added to Reservation object. Customer ID added to Customer object.

Procedural Steps for TC 1.3

Create Reservation

Black-Box Testing

Test Number	Description	Input	Expected Output
TN 1.3 – 1	Information from Customer, Room, and ReservationDate objects transferred	(None from end-user)	(none seen by end-user)
TN 1.3– 2	Reservation object sent to Reservation database	(None from end-user)	(none seen by end-user)
TN 1.3 – 3	Reservation and Customer ID numbers added to respective objects	(None from end-user)	GUI changes to Hold Reservation window.

4.4.2. Test Case 1.4 Hold Reservation without Reservation Guarantee

Objective

The customer chooses to hold a room for the selected dates without paying the reservation guarantee payment. The room will be held as reserved in the database until the reservation guarantee is paid or a chosen hold expiration date has passed.

Test Case Specifier

Test Case 1.4

Input Specification

End-user enters a hold date into the GUI.

Output Specification

Hold date is saved ReservationDate object and in Reservation database. Reservation status in database is set to awaiting payment.

Dependencies

Special Environmental Conditions

Procedural Requirements

Procedural Steps

<table>
<tr><th colspan="4">Procedural Steps for TC 1.4
Hold Reservation without Reservation Guarantee</th></tr>
<tr><th colspan="4">White-Box Testing</th></tr>
<tr><th>Test Number</th><th>Description</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>TN 1.4 – 1</td><td>End-user enters hold date in Reservation Hold Date textfield</td><td>Keyboard strokes</td><td>Scanner objects created</td></tr>
<tr><td>TN 1.4– 2</td><td>The entered date is valid</td><td>Valid date (XX/XX/XXXX)</td><td>String objects created</td></tr>
<tr><td>TN 1.4 – 3</td><td>The entered date is valid</td><td>Invalid date</td><td>Error message generated</td></tr>
<tr><td>TN 1.4 – 4</td><td>End-user selects Hold Reservation button</td><td>Button click event</td><td>String object added to ReservationDate object in reservationHold field and guaranteePaid boolean set to false</td></tr>
</table>

TN 1.4 – 5	Reservation sends hold date to ReservationDate object	(none from end-user)	Date object added to ReservationDate object

<table>
<tr><th colspan="4">Procedural Steps for TC 1.4
Hold Reservation without Reservation Guarantee</th></tr>
<tr><th colspan="4">Black-Box Testing</th></tr>
<tr><th>Test Number</th><th>Description</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>TN 1.4 – 1</td><td>End-user enters hold date in Reservation Hold Date textfield</td><td>Keyboard strokes</td><td>(none seen by end-user)</td></tr>
<tr><td>TN 1.4– 2</td><td>The entered date is valid</td><td>Valid date (XX/XX/XXXX)</td><td>(none seen by end-user)</td></tr>
<tr><td>TN 1.4 – 3</td><td>The entered date is valid</td><td>Invalid date</td><td>Error message displays on Error Message window</td></tr>
<tr><td>TN 1.4 – 4</td><td>End-user selects Hold Reservation button</td><td>Button click event</td><td>(none seen by end-user)</td></tr>
<tr><td>TN 1.4 – 5</td><td>Reservation sends hold date to ReservationDate object</td><td>(none from end-user)</td><td>(none seen by end-user)</td></tr>
</table>

4.5. Reservation System, Use Case 1.5 Hold Reservation without Reservation Guarantee

4.5.1. Test Case 1.5 Hold Reservation without Reservation Guarantee

Objective

The customer pays the reservation guarantee fee to hold the room. The room is held until the ending date of the reservation selected by the customer. The customer pays with a credit or debit card.

Test Case Specifier

Test Case 1.5

Input Specification

End-user enters a hold date into the GUI. Credit card information or debit card information and PIN number are input via the payment system.

Output Specification

The customer reservation is confirmed held.

Dependencies

The payment system must be operational

Special Environmental Conditions

- Credit card reader must be present and operational
- Key pad reader must be present and operational

Procedural Requirements

The steps in Test Case 1.4 have not been performed and the system GUI is waiting for input at Test Case 1.3 with the Hold Reservation Window on the display.

Procedural Steps

<table>
<tr><th colspan="4">Procedural Steps for TC 1.5
Hold Reservation with Reservation Guarantee</th></tr>
<tr><th colspan="4">White-Box Testing</th></tr>
<tr><th>Test Number</th><th>Description</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>TN 1.5 – 1</td><td>End-user enters the ending date of the reservation in the Hold Reservation Date textfield</td><td>Keyboard strokes</td><td>Scanner object</td></tr>
<tr><td>TN 1.4 – 2</td><td>Valid date entered</td><td>Valid date (XX/XX/XXXX)</td><td>String objects created</td></tr>
<tr><td>TN 1.4 – 3</td><td>Invalid date entered</td><td>Invalid date</td><td>Error message generated</td></tr>
<tr><td>TN 1.4 – 4</td><td>Customer swipes credit or debit card</td><td>Credit card barcode is readable</td><td>String object containing credit card number is added to Card object</td></tr>
<tr><td>TN 1.4 – 5</td><td>Customer swipes credit card with problematic barcode</td><td>Credit card barcode is unreadable</td><td>Error message generated</td></tr>
<tr><td>TN 1.4 – 6</td><td>Customer enters valid PIN number in key pad</td><td>Key pad strokes, 4 strokes total</td><td>Integer object</td></tr>
</table>

TN 1.4 – 7	Customer enters invalid PIN number in key pad	Key pad strokes, insufficient	Error message generated
TN 1.4 – 8	Card object is verified via Internet	Card object	Verified card with sufficient funds
TN 1.4 – 8	Card object is verified via Internet	Card object	Card stolen or insufficient funds. Error message generated.
TN 1.4 - 9	End-user selects Charge Card button	Button click	Card is charged the reservation guarantee amount • guaranteePaid boolean in Reservation object set to true • Reservation object sends cost of reservation to reservation database • Reservation database communications reservation guarantee cost to Accounting database • Reservation database returns Reservation confirmation number to Reservation object • Reservation object adds confirmation number to Card object

4.6. Accounting System, Use Case 2.0 Search for Financial Data

4.6.1. Test Case 2.1 Query Accounting database for financial records

Objective

The end-user is searching for financial records from a specified starting and ending date. All income and expenses during that date rate will be displayed in the display area of the GUI.

Test Case Specifier

Test Case 2.1

Input Specification

Starting and ending dates for the financial record search.

Output Specification

Financials records displayed on the GUI

Dependencies

The end-user is authorized to user the Accounting system and has a username and password that will allow access.

Special Environmental Conditions

Connection to an internal or external network that connects to the Accounting database.

Procedural Requirements

Procedural Steps

Procedural Steps for TC 2.0 Query Accounting database for financial records			
Black-Box Testing			
Test Number	**Description**	**Input**	**Expected Output**
TN 2.0 – 1	End-user enters valid start date	Valid Start Date (XX/XX/XXX)	None
TN 2.0 – 2	End-user enters invalid start date	Invalid Start Date	Error message displays in GUI
TN 2.0 – 3	End-user enters valid end date	Valid End Date (XX/XX/XXX)	None
TN 2.0 – 4	End-user enters invalid end date	Invalid End Date	Error message displays in GUI
TN 2.0 – 5	End-user clicks Search button	Button click on Search button	Search button depresses

1.1.1. Test Case 2.1 View Financial Reports

Objective

Search results are displayed in the GUI. The end-user can view the results or print a financial report.

Test Case Specifier

Test Case 2.1

Input Specification

End-user clicks Print Report button

Output Specification

Results displayed on GUI and/or on printed report

Dependencies

The end-user is authorized to user the Accounting system and has a username and password that will allow access.

Special Environmental Conditions

Connection to an internal or external network that connects to the Accounting database. Connection to a working printer with sufficient ink.

Procedural Requirements

None

Procedural Steps

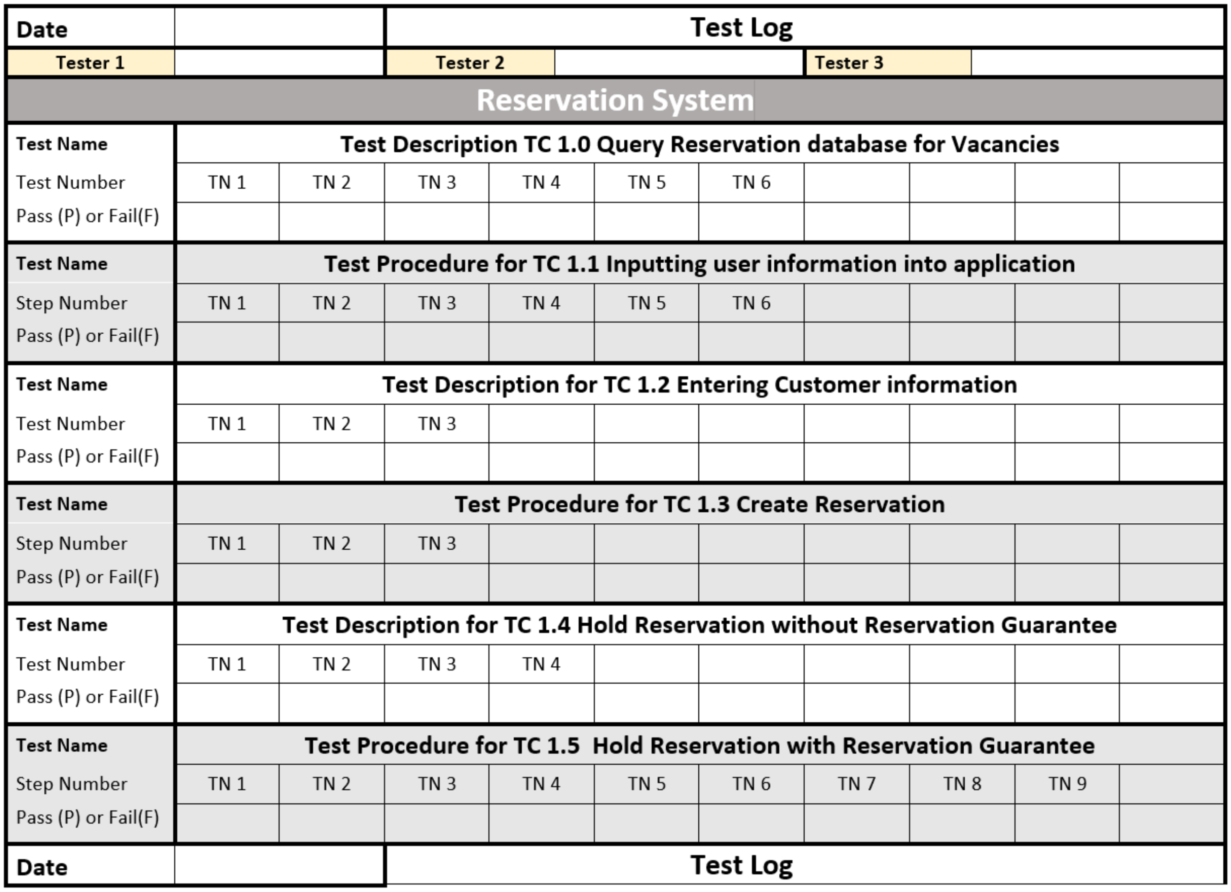
Procedural Steps for TC 2.1

View financial records

Black-Box Testing			
Test Number	**Description**	**Input**	**Expected Output**
TN 2.1 – 1	Accounting database returns financial records for display	None from end-user	Displayed results on GUI. Print report Button available for selection
TN 2.1 – 2	End-user selects Print report button	Button click	Records are sent to printer

5. Traceability Matrix

A Traceability Matrix and test log are shown in Figure 1.



Date		**Test Log**								
Tester 1		**Tester 2**				**Tester 3**				
Reservation System										
Test Name	**Test Description TC 1.0 Query Reservation database for Vacancies**									
Test Number	TN 1	TN 2	TN 3	TN 4	TN 5	TN 6				
Pass (P) or Fail(F)										
Test Name	**Test Procedure for TC 1.1 Inputting user information into application**									
Step Number	TN 1	TN 2	TN 3	TN 4	TN 5	TN 6				
Pass (P) or Fail(F)										
Test Name	**Test Description for TC 1.2 Entering Customer information**									
Test Number	TN 1	TN 2	TN 3							
Pass (P) or Fail(F)										
Test Name	**Test Procedure for TC 1.3 Create Reservation**									
Step Number	TN 1	TN 2	TN 3							
Pass (P) or Fail(F)										
Test Name	**Test Description for TC 1.4 Hold Reservation without Reservation Guarantee**									
Test Number	TN 1	TN 2	TN 3	TN 4						
Pass (P) or Fail(F)										
Test Name	**Test Procedure for TC 1.5 Hold Reservation with Reservation Guarantee**									
Step Number	TN 1	TN 2	TN 3	TN 4	TN 5	TN 6	TN 7	TN 8	TN 9	
Pass (P) or Fail(F)										
Date		**Test Log**								

Figure 1. Test Log and Traceability Matrix