**Bed & Breakfast (B&B)**

**Management System**

**Software Test Specification (STS)**

Test Plan Identifier: *BBMS-TS-0001*

**Team Delta:**

*Qeturah Jackson, Roger Lee, Paul Minniti, William Foy*

University of Maryland University College

CMIS 330 Section 7980 Software Engineering Principles and Techniques

Professor: Lauren King

April 23, 2017

**Contents**

[1. Introduction 1](#_Toc480719359)

[1.1 Objectives 1](#_Toc480719360)

[1.2 Background 1](#_Toc480719361)

[1.3 Scope 1](#_Toc480719362)

[1.4 References 1](#_Toc480719363)

[1.5 Test Environment 2](#_Toc480719364)

[1.5.1 Hardware 2](#_Toc480719365)

[1.5.2 Software 2](#_Toc480719366)

[1.5.2.1 Operating System 2](#_Toc480719367)

[1.5.2.2 BBMS Version 1.0 2](#_Toc480719368)

[1.5.2.3 Microsoft Access 2016 2](#_Toc480719369)

[1.5.2.4 Java SE 8 Update 131 2](#_Toc480719370)

[1.5.3 Data 3](#_Toc480719371)

[1.5.4 Security 3](#_Toc480719372)

[2. Architectural Design 3](#_Toc480719373)

[2.1 Program Modules 3](#_Toc480719374)

[2.1.1 User Interface 4](#_Toc480719375)

[2.1.2 Services 4](#_Toc480719376)

[2.1.3 Domain Objects 5](#_Toc480719377)

[2.1.4 Database 6](#_Toc480719378)

[3. Traceability Matrix 7](#_Toc480719379)

[4. Test Specifications 8](#_Toc480719380)

[4.1 Black Box Test Cases 8](#_Toc480719381)

[4.1.1 Test Case Specification BBMS-BB-0001 8](#_Toc480719382)

[4.1.2 Test Case Specification BBMS-BB-0002 9](#_Toc480719383)

[4.1.3 Test Case Specification BBMS-BB-0003 11](#_Toc480719384)

[4.1.4 Test Case Specification BBMS-BB-0004 12](#_Toc480719385)

[4.2 White Box Test Cases 13](#_Toc480719386)

[4.2.1 Test Case Specification BBMS-WB-0001 13](#_Toc480719387)

[4.2.2 Test Case Specification BBMS-WB-0002 14](#_Toc480719388)

[4.2.3 Test Case Specification BBMS-WB-0003 15](#_Toc480719389)

[4.2.4 Test Case Specification BBMS-WB-0004 16](#_Toc480719390)

[4.2.5 Test Case Specification BBMS-WB-0005 17](#_Toc480719391)

[References 18](#_Toc480719392)

# 1. Introduction

## 1.1 Objectives

This Software Test Specification (STS) was created for the software development team of the Bed and Breakfast (B&B) Management System software created for John and Jane Smith. This document will contain traceability of requirements to each design entity and describe the software structure, software components and interfaces. This document will be used by the ABC Software company’s test team to perform testing of the BBMS software.

## 1.2 Background

The B&B Management System (B&BMS) is an interactive total software solution for the Smith’s business. This system will be compromised of a terminal which will display the interactive User Interface (UI), a database which will store all the data which is input in the system as well as created data.

This system will allow front user to select a date and determine which rooms are available to for rent. It will allow the user to enter all the required customer information (name, address, phone number, and credit card number) into the guest database which will be stored. The system will allow users to process credit card payments and guarantee reservations all from the user interface. The reservation data will be stored in the reservation database.

In addition to the guest and reservation database functionality this system also will store and compute accounting data in the accounting database. The accounting database will allow privileged users access to financial reporting that include expenses and profits. This will provide the Smith’s with access to their profit and loss data at any time with the ease of a click of a button.

## 1.3 Scope

The Software Test Specification will be used by ABC Software’s test team to perform all required testing of the BBMS. The software Architectural Design of the BBMS is contained in Section 2 of this document. The test case specifications of both white box and black box are defined in *Section 4*. The STS provides a definition of all aspects of testing required for the BBBMS.

## 1.4 References

* The Institute of Electrical and Electronics Engineers (IEEE) Standard for Software Test Documentation (IEEE Std. 829-1998) was referenced in the creation of this document.
* BBMS Software Requirements Specification (SRS)
* BBMS Software Design Document (SDD)

## 1.5 Test Environment

The will be a simulated test environment located at ABC Software Company which will contain hardware and software which meets the guidelines contained in *Section 3.1 – 3.5*. I addition to these items it is assumed the BBMS will have access to a surge protected power supply and a highspeed internet connection via ethernet.

## 1.5.1 Hardware

A desktop computer with a 24” monitor with a resolution of at least 1920p x 1080p attached via HDMI connection. This monitor will have a built-in credit card reading allowing BBMS users the ability to process payments when presented with a physical credit card. The desktop computer should have at minimum a 5th Generation Intel i5 processor (2.4 GHz), 8GB of RAM and 1TB of hard drive space. A USB connected keyboard and mouse are required for the user to operate the BBMS.

## 1.5.2 Software

The BBMS requires certain software be installed on the desktop computer mentioned in *Section 3.1*, these software components are contained in the subsections below.

### 1.5.2.1 Operating System

Microsoft Windows 10 Professional Shall be used for testing purposes for the BBMS.

### 1.5.2.2 BBMS Version 1.0

The test system shall have BBMS Version 1.0 installed and configured which will allow transmission of data to and from the Database module 4.0 from the ACD in Figure 1.

### 1.5.2.3 Microsoft Access 2016

Microsoft Access is required as it is an integral part of the BBMS. All of the database functionality is contained in a database built with Microsoft Access.

### 1.5.2.4 Java SE 8 Update 131

Java SE 8 is required to be installed on the test system. The BBMS is written in Java and this software supports the Graphic User Interface of the BBMS.

## 1.5.3 Data

The data required for testing will be created in the White Box test cases. Therefore no additional data is required to perform testing of the BBMS software.

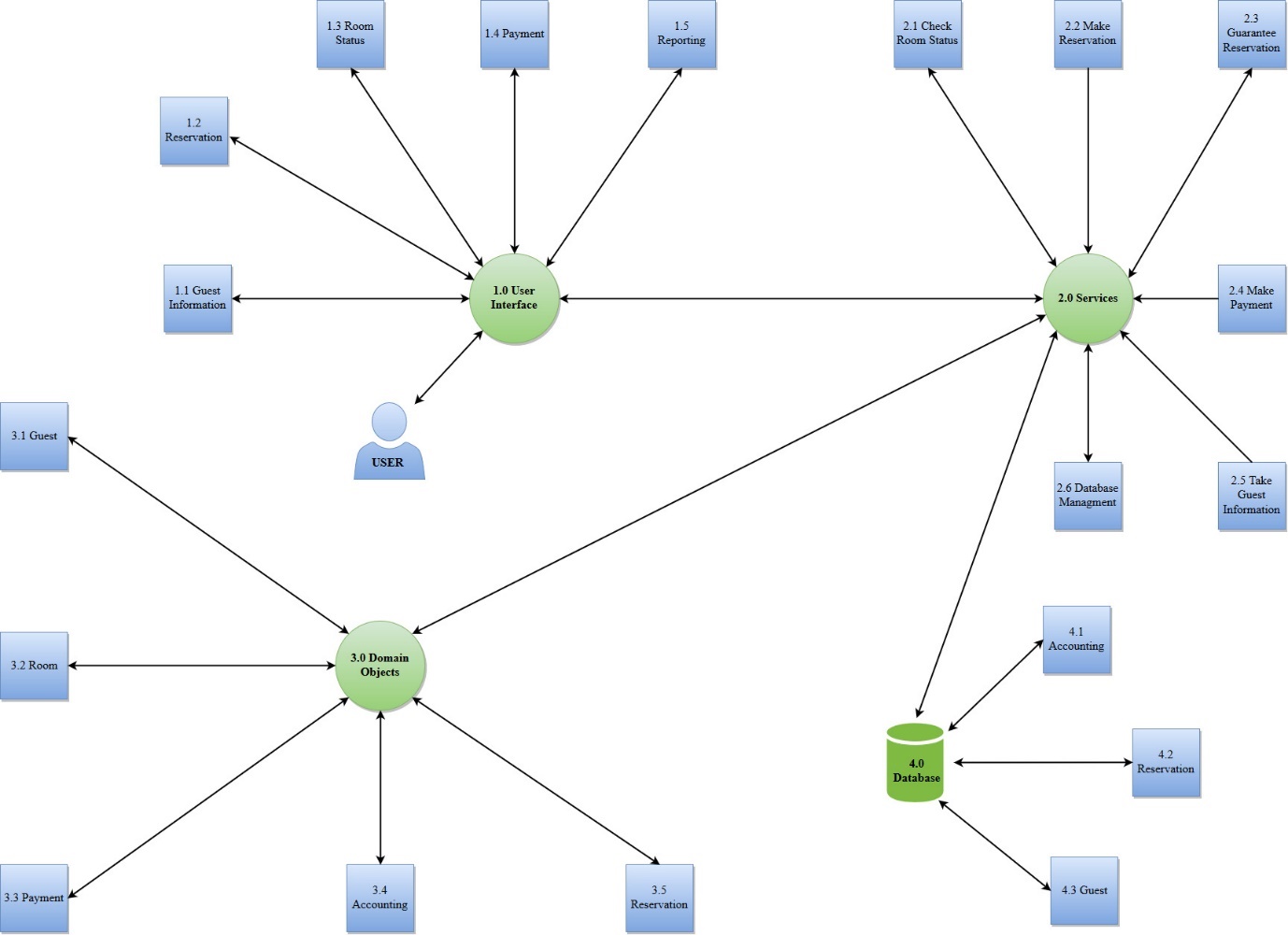
## 1.5.4 Security

The test environment will follow standard ABC Software Company security controls which will restrict access to the test environment to members of the testing team. This is accomplished by requiring swipe access with a pin to open the door to the test environment. Only members of ABC’s testing team has swipe access to this room.

# 2. Architectural Design

## 2.1 Program Modules

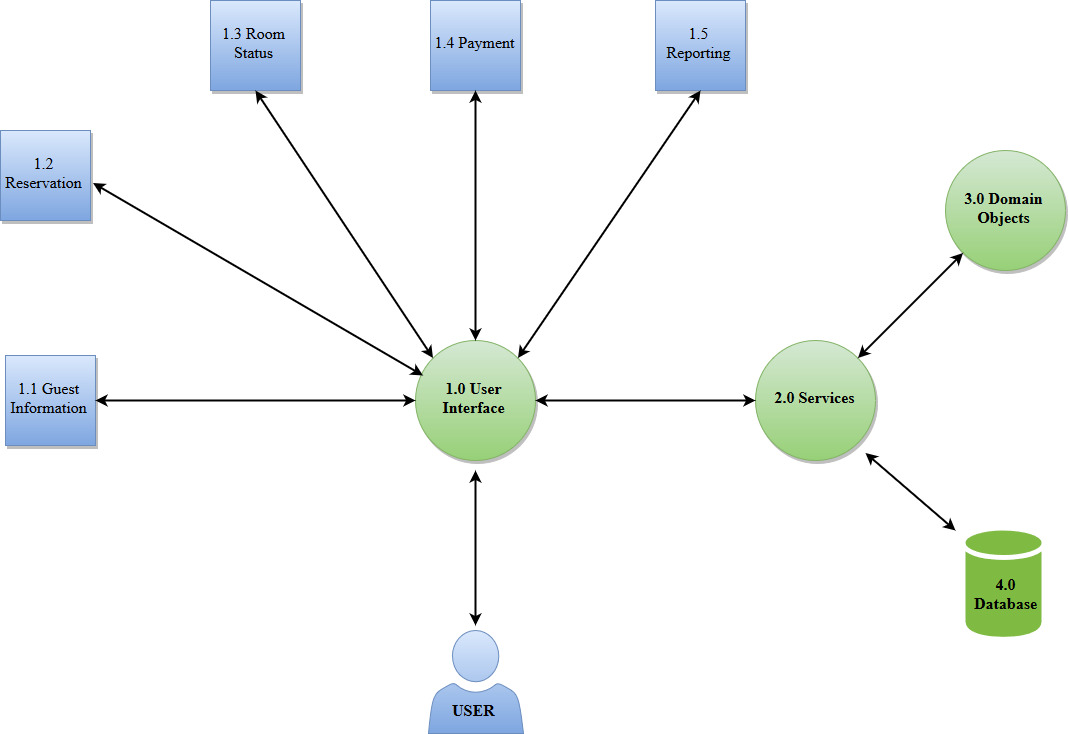
There are four different modules that comprise the B&BMS, which are shown below in Fig. 1. This Architectural Context Diagram (ACD) shows the User Interface, Services, Domain Objects, and Database modules. Descriptions are provided for each module in the proceeding sections.



**Fig. 1 ACD**

### 2.1.1 User Interface

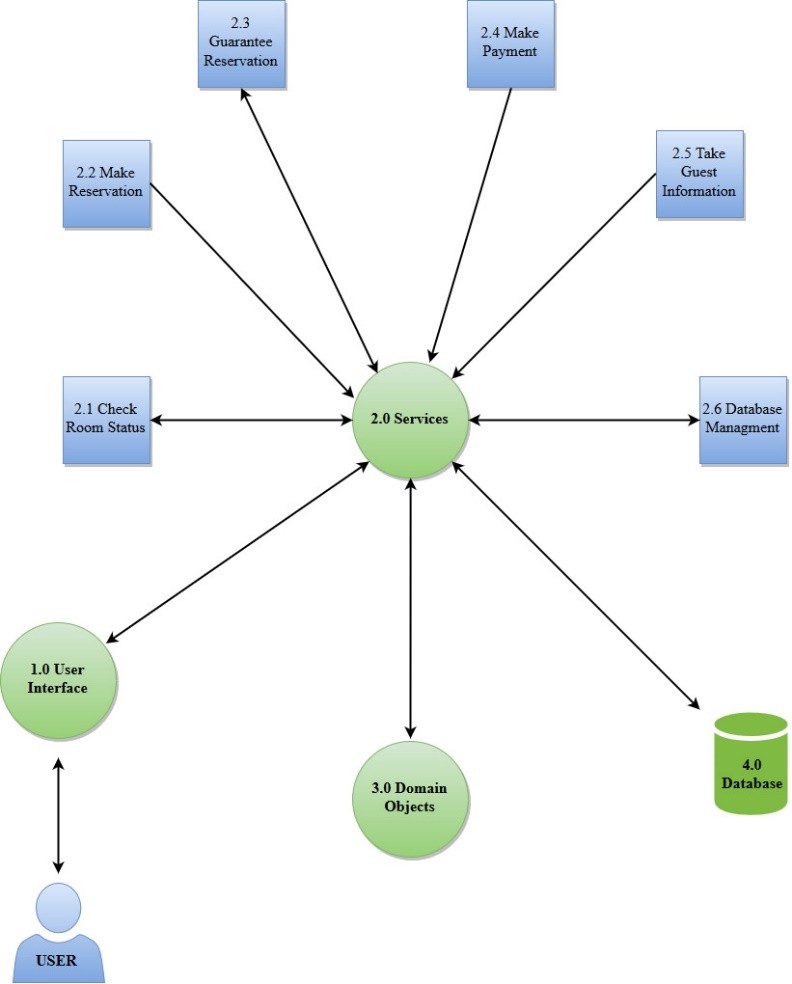
The UI is the only module that requires human interaction, which will be accessed only by B&B staff and/or management. The purpose of the UI is to provide one location for users to perform any of the B&BMS’s functions. This could be as simple as the user clicking the room status button to see if a room is available for rent on a particular day to a more complex function such as yearly accounting reports. Below in Fig. 2 you will see each of each functional process of the UI*.*



**Fig. 2 User Interface ACD**

### 2.1.2 Services

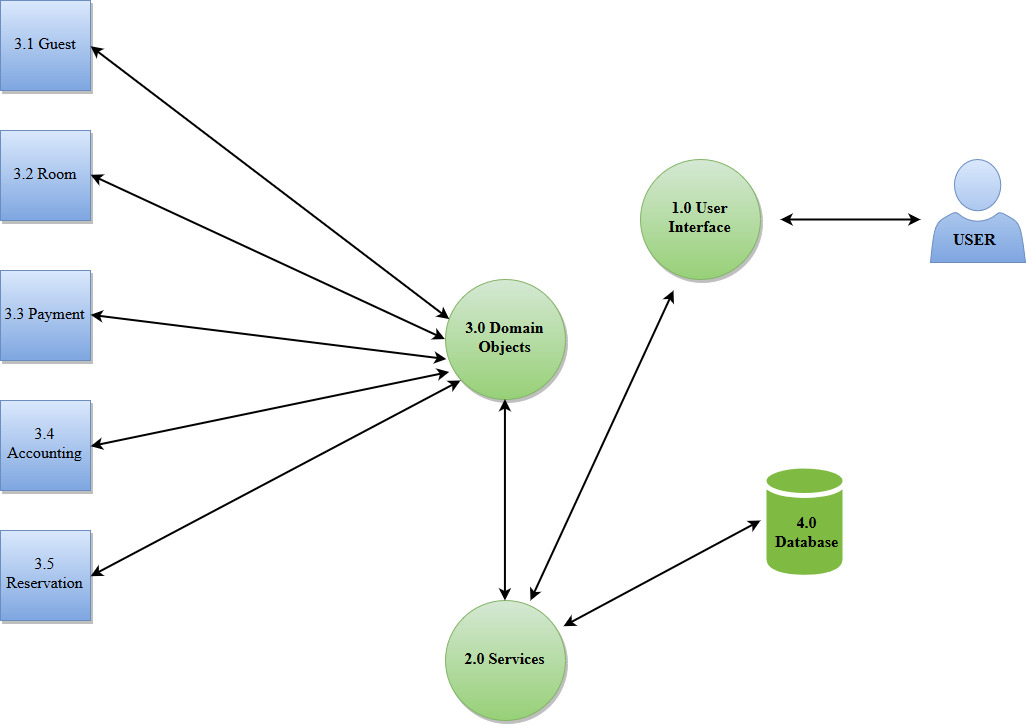
The Services module is the only module that interacts with all of the other modules (UI, Domain Objects, Database). The functions are shown below in Fig. 3



**Fig. 3 Services ACD**

### 2.1.3 Domain Objects

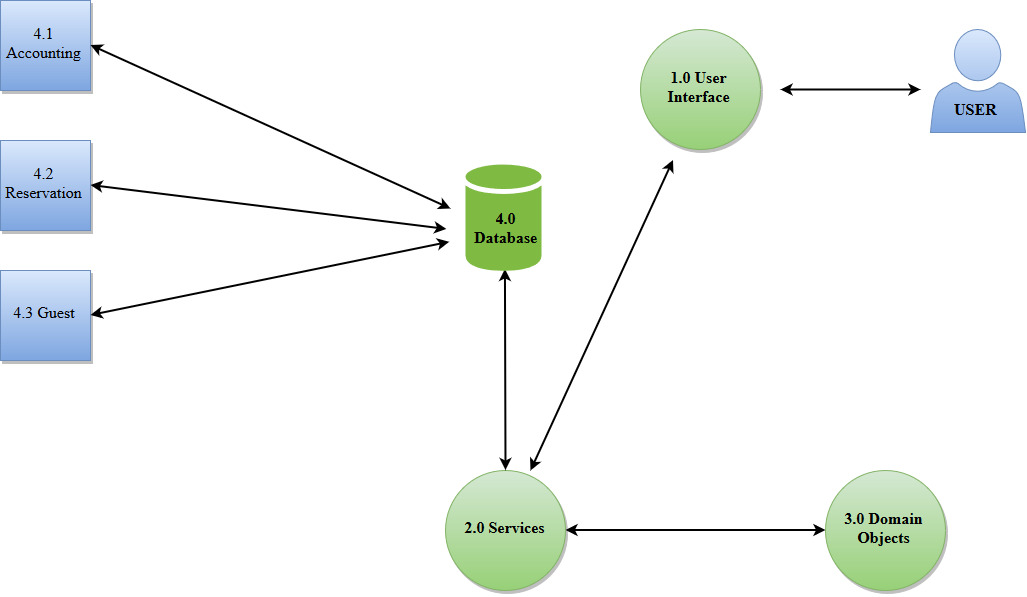
The Domain Objects module is represented in Fig. 4 as 3.1 – 3.5. Each object preforms a specific function, data inputs and exports are stored in the B&BMS database.

****

**Fig. 4 Domain Objects ACD**

### 2.1.4 Database

The Database module is used to store all data that is collected and generated for the B&BMS. There are three entities within the Database module (Accounting, Reservation, Guest) which are shown in Fig. 5.

****

**Fig. 5 Database ACD**

# 3. Traceability Matrix

This matrix contains the traceability from system requirements and use cases which are outlined in the Software Requirements Specification (SRS), and software requirements which are shown in *Section 4* of this document as well as the Software Design Document (SDD).



# 4. Test Specifications

## 4.1 Black Box Test Cases

### 4.1.1 Test Case Specification BBMS-BB-0001

**Test Case Identifier**: BBMS-BB-0001

**Description**

This is a Black Box test for creating a guest in the Guest Creation screen of the user interface. The subsequent appearance of guest’s information in the Guest Information screen is also tested.

**Input Specification**

1. Click Create Guest Button.
2. Enter “John Doe” in Name field.
3. Enter “123 Maple Drive, Rumney, NH 03274” in Address field.
4. Enter “1234567890” in Phone Number field.
5. Enter “1234123412341234” in Credit Card Number field.
6. Click “Create” button.
7. Click on guest drop down box.
8. Click scroll arrow until John Doe appears.
9. Click on John Doe.
10. Click Guest Information button.

**Output Specification**

1. Screen goes to Guest Creation screen.
2. “John Doe” appears in the Name field.
3. “123 Maple Drive, Rumney, NH 03274” appears in Address field.
4. “1234567890” appears in Phone Number field.
5. “1234123412341234” appears in Credit Card Number field.
6. Screen returns to Home screen.
7. List of guests appears.
8. John Doe appears in guest drop down box.
9. Drop down minimizes and John Doe is selected.
10. Guest Information screen appears. Name field has a value of “John Doe”; Address field has a value of “123 Maple Drive, Rumney, NH 03274”; Phone Number field has a value of 1234567890”; Credit Card Number field has a value of “1234123412341234”.

**Environmental Needs**

Requires working computer with BBMS software installed with a working keyboard and mouse. Database must be capable of storing guest information.

**Special Procedural Requirements**

Tester starts on the Home screen.

**Intercase Dependencies**: None

### 4.1.2 Test Case Specification BBMS-BB-0002

**Test Case Identifier**: BBMS-BB-0002

**Description**

This is a Black Box test case for making a reservation for a guest in the Reservations screen of the user interface. The appearance of the reservation in the Calendar is also tested.

**Input Specification**

1. Select a guest from guest drop down.
2. Click Guest Information button.
3. Click Reservations button.
4. Click Create Reservation.
5. Enter “2017-06-01” in Start Date field.
6. Enter “2017-06-02” in End Date field.
7. Enter “2017-05-31” in Guarantee Date field.
8. Enter “100.00” in Price field.
9. Click Check Vacancy button.
10. Click Room drop down box.
11. Select Room 1.
12. Click Create Reservation button.
13. Click Reservations button.
14. Click Calendar button.

**Output Specification**

1. A guest is selected in the drop down.
2. Guest Information screen appears along with guest information (name, address, phone number, credit card number).
3. Reservations screen appears. Reservations table is blank.
4. Create Reservation screen appears. Start Date, End Date, Guarantee Date, and Price fields are empty.
5. “2017-06-01” appears in Start Date field.
6. “2017-06-02” appears in End Date field.
7. “2017-05-31” appears in Guarantee Date field.
8. Message appears displaying that Rooms 1 through 4 are vacant.
9. “100.00” appears in Price field.
10. Rooms 1 through 4 appear in drop down.
11. Room 1 is selected in drop down box.
12. Reservation created message appears and Guest Information screen appears. Reservation is now selected.
13. A new reservation appears in the Reservations Table that displays Start Date, End Date, Guarantee Date, Price, Room Number, Payment Status. Start Date value is “June 1, 2017”; End Date value is “June 2, 2017”; Guarantee Date value is “May 31, 2017”; Room value is “Room 1”; Price value is “$100.00”; Status value is “Unconfirmed”; and Payment Status value is “Unpaid”.
14. Room 1 with status “Unconfirmed” appears for June 1st, 2017 and June 2nd, 2017.

**Environmental Needs**

Requires working computer with BBMS software installed with a working keyboard and mouse. Database must be capable of storing reservation information.

**Special Procedural Requirements**

Tester starts on Home screen. A guest must have already been created in the system prior to this test. The selected guest does not have any prior reservations.

**Intercase Dependencies**: A guest must be created in the system using the procedure detailed in BBMS-BB-0001 prior to this test.

### 4.1.3 Test Case Specification BBMS-BB-0003

**Test Case Identifier**: BBMS-BB-0003

**Description**

This is a Black Box test case tests for testing the functionality of the Calendar from the Reservations screen. A reserved room is tested to appear in the Calendar. The Cancel Reservation functionality is also tested.

**Input Specification**

1. Select guest from guest drop down.
2. Click Guest Information Button.
3. Click Reservations button.
4. Click reservation in Reservations Table
5. Click Cancel Reservation button
6. Click Yes button.
7. Click Calendar

**Output Specification**

1. Guest is selected.
2. Guest Information screen appears.
3. Reservation screen appears. Reservations Table is populated with a reservation.
4. The reservation is selected.
5. “Are you sure you want to cancel this reservation?” Yes/No prompt appears.
6. The reservation disappears from the Reservations Table.
7. The room number disappears from the calendar for the date ranges of the reservation.

**Environmental Needs**

Requires working computer with BBMS software installed with a working keyboard and mouse. Database must be capable of modifying reservation and Calendar information.

**Special Procedural Requirements**

Tester starts on Home screen. A guest must have already been created in the system prior to this test with a reservation.

**Intercase Dependencies**: A guest is previously created in the system with a reservation as detailed in BBMS-BB-0002.

### 4.1.4 Test Case Specification BBMS-BB-0004

**Test Case Identifier**: BBMS-BB-0004

**Description**

This Black Box test case tests the Payment portion of the user interface. Two payments are made. The first payment is a partial payment. The second payment fulfills the remainder of the amount owed by the guest.

**Input Specification**

1. Click the guest in the guest drop down.
2. Click Guest Information button.
3. Click Reservations button.
4. Click the unpaid reservation in the Reservations Table.
5. Click Make a Payment button.
6. Enter “40.00” in Payment Amount field.
7. Click Confirm Payment button.
8. Click the reservation in the Reservations Table.
9. Click Make a Payment button.
10. Enter “60.00” in Payment Amount field.
11. Click Confirm Payment button.
12. Click Calendar.

**Output Specification**

1. The guest is selected.
2. Guest Information screen appears.
3. Reservations screen appears.
4. The unpaid reservation is selected.
5. Payment screen appears. Payment Amount field is blank; Amount Paid field is “0.00”; and Total Price field is “100.00”.
6. “40.00” appears in Payment Amount field.
7. “Payment confirmed” message appears and Reservations screen appears. The reservation in the Reservations Table in unchanged.
8. The reservation is selected.
9. Payment screen appears. Payment Amount field is blank; Amount Paid field is “40.00”; and Total Price field is “100.00”.
10. “60.00” appears in Payment Amount field.
11. “Payment confirmed” message appears and Reservations screen appears. The Status and Payment Status for the reservation in the Reservations Table is respectively “Confirmed” and “Paid”.
12. The reservation room with status “Confirmed” appears for the dates of the reservation in the Calendar.

**Environmental Needs**

Requires working computer with BBMS software installed with a working keyboard and mouse. Database must be capable of storing payment information.

**Special Procedural Requirements**

Tester starts on Home screen. A guest must have already been created in the system prior to this test. The selected guest has a prior reservation with a price of $100.00. The guest has not made any payments.

**Intercase Dependencies**: A guest is previously created in the system with a reservation as detailed in BBMS-BB-0002.

## 4.2 White Box Test Cases

This section describes the test cases and what they are designed to test in the application programming interface. It lists the Objective of the test, input and output specifications, environment needs and special requirements.

### 4.2.1 Test Case Specification BBMS-WB-0001

**Test Case Identifier**: BBMS-WB-0001

**Description**

This section describes a white-box test case for the reservations portion of the software to assign new customers to rooms and reserve it for the specified length of their stay.

**Input Specification**

1. Enter date range or select a date on the calendar.
2. Select from available rooms
3. Enter reservation information
4. Click “Complete Reservation”

**Output Specification**

1. Reservation form will be displayed in a pop up window.
2. Accepted reservation details will display after user has entered correct information.

**Environmental Needs**

* Operating system with the software installed
* Mouse
* Keyboard
* Monitor

**Special Procedural Requirements**

When two or more rooms are available the system requires additional input from the user to advance in the reservation process.

**Inter-case Dependencies**

None

### 4.2.2 Test Case Specification BBMS-WB-0002

**Test Case Identifier**: BBMS-WB-0002

**Description**

This section describes a white-box test case for the guest information portion of the software to store guest information through the service.

**Input Specification**

1. Click “Guest Information Form” to input guest’s information
2. Perform check for guest reservation form. If reservation form is present, the software will move to the next step.
3. Input guests first name.
4. Input guests last name.
5. Input guests address.
6. Input guests phone number.
7. Input guest’s payment information.
8. Click “Submit” button.

**Output Specification**

1. If reservation form is not present an error message is shown and user is prompted for reservation form.
2. If successful, the user will be notified and the form will be closed. If not successful, the user will be notified on which fields are incorrect.

**Environmental Needs**

* Operating system with the software installed
* Mouse
* Keyboard
* Monitor

**Special Procedural Requirements**

The system expects valid input and must be at a point to accept the guest’s information

**Inter-case Dependencies**

BBMS-WB-0001 must be completed before this test case because the user must complete the reservation process before taking guest information.

### 4.2.3 Test Case Specification BBMS-WB-0003

**Test Case Identifier**: BBMS-WB-0003

**Description**

This section describes a white-box test case for the payment portion of the domain objects. It will test making processing a payment from a guest through the service.

**Input Specification**

1. Click “Process Payment form” to input payment information
2. Perform check for guest reservation form and guest information form. If forms are present software will move onto next step
3. Input total payment amount.
4. Click “Process payment”

**Output Specification**

1. If forms are not present an error message is shown and user is prompted for the forms.
2. Payment service will display “successful payment” or “payment error” and prompt the user for correct information.
3. If successful system will return to the previous screen.

**Environmental Needs**

* Operating system with the software installed
* Mouse
* Keyboard
* Monitor

**Special Procedural Requirements**

None

**Inter-case Dependencies**

BBMS-WB-0001 and BBMS-WB-0001 must be completed before this test case because the user must complete the reservation and guest information process before continuing to process payment.

### 4.2.4 Test Case Specification BBMS-WB-0004

**Test Case Identifier**: BBMS-WB-0004

**Description**

This section describes a white-box test case for the room domain object to assign a guest to a room.

**Input Specification**

1. Click “Room Selection” to assign a guest to a room
2. Perform check for guest reservation form and guest information form. If forms are present software will move onto next step
3. Select a room and click “assign to guest”.

**Output Specification**

1. If forms are not present an error message is shown and user is prompted for the forms.
2. Display all available rooms for applicable date on the forms
3. System will submit room status to database.
4. After completion system will return to main screen.

**Environmental Needs**

* Operating system with the software installed
* Mouse
* Keyboard
* Monitor

**Special Procedural Requirements**

None

**Inter-case Dependencies**

BBMS-WB-0001 and BBMS-WB-0001 must be completed before this test case because the user must complete the reservation and guest information process before continuing to assign a room.

### 4.2.5 Test Case Specification BBMS-WB-0005

**Test Case Identifier**: BBMS-WB-0004

**Description**

This section describes a white-box test case for the accounting domain object to keep track of room billing.

**Input Specification**

1. System will track any items order to the room or taken from mini fridge.
2. System will track total amount payed

**Output Specification**

1. Purchases will be calculated and added to total bill.
2. Calculate total purchases and subtract total payed
3. Display total amount owed before checkout.

**Environmental Needs**

* Operating system with the software installed
* Mouse
* Keyboard
* Monitor

**Special Procedural Requirements**

None

**Inter-case Dependencies**

None

# References

Institute of Electrical and Electronics Engineers 1998 IEEE recommended practice for software requirements specifications.The Institute of Electrical and Electronics Engineers. (1998). IEEE Standard for Software Test Documentation.

1-59. Retrieved from ISBN:0-7381- 1444-8.