Temasek Polytechnic

School of Informatics and IT

**Diploma in Information Technology (IT)**

Software Test Specifications (STS)

**Project Particulars**

|  |  |
| --- | --- |
| **Tutor** | Qi Yutao |
| **Class** | P01 |
| **Project Title** | Delonix Regia Hotel Management System |

**Project Team’s Particulars**

|  |  |
| --- | --- |
| **Matric Number** | **Student Name** |
| **1605809A** | **Zafrulla** |
| **1605894C** | **Zulhilmi** |
| **1602691F** | **Myron Low** |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 04/12/17 | 1.0 | Started Module Development | Myron |
| 06/12/17 | 1.1 | Started Unit Testing | Zul |
| 07/12/17 | 1.2 | Finalized Module Development | Zaf |
| 09/12/17 | 1.3 | Finalized Unit Testing | Zul |
| 10/12/17 | 1.4 | Started System Integration | Myron |
| 11/12/17 | 1.5 | Finalized System Integration | Zaf |
| 12/12/17 | 1.6 | Started and Finalized Test Log | Zul |

**Table of Contents**

[1. DISTRIBUTION OF WORKLOAD 4](#_Toc500880767)

[2. MODULE DEVELOPMENT AND UNIT TESTING 4](#_Toc500880768)

[2.3 Room Availability and Booking 5](#_Toc500880769)

[2.4 Housekeeping and staff management 6](#_Toc500880782)

[2.5 User account and login creation 10](#_Toc500880783)

[SYSTEM INTEGRATION 13](#_Toc500880784)

[TEST LOG 13](#_Toc500880785)

[Login 13](#_Toc500880786)

[Add Employee 13](#_Toc500880787)

[Edits Employee 15](#_Toc500880789)

[Add Housekeeping 16](#_Toc500880791)

[Room Reservation 17](#_Toc500880792)

[Reference 19](#_Toc500880793)

# DISTRIBUTION OF WORKLOAD

|  |  |
| --- | --- |
| **Construction & Testing** | **Members** |
| **Unit Testing**  Black-Box  **Module Development**  User login and account creation  Staff management  **Test Log**  Login  Add Employee  Edits Employee  **System Integration** | Zulhilmi |
| **Unit Testing**  White-Box  **Module Development**  Housekeeping  **Test Log**  Add Housekeeping  Edits Housekeeping | Zafrulla |
| **Module Development**  Room Availability and Booking  **Test Log**  Room Reservation | Myron |

# MODULE DEVELOPMENT AND UNIT TESTING

**2.1 Black-Box Testing**

A software testing method in which the internal structure (program source code) is not known to the front-end user. This testing will treat the website as a black-box, meaning it isn't transparent, thus the front-end user will not be able to see the internal source codes of the website. This method will help find errors in the User Interface, functions, performance and data structures of the website. The testing will be done at the front-end user point of view.

With black-box testing, the tester does not require extensive knowledge. However, every aspect of the system should be tested. This means that we have to test the login page, and all the other forms in the system. The test will also check if system also matches with the specifications stated in the first and second assignment.

The tester can also make guesses as to where an error will occur. For example, in a form, if label is “Contact Number”, the tester can check if the input of a string will be successful or an error message is provided. If the string is successfully submitted it is not an error that disrupts the system and use flow but it is still a huge error in using the proper validation techniques.

Black-box testing might also lead to unidentified pages. There will also be cases where bugs are not found out using this method of testing. With this reason, it is important to also implement white box testing.

**2.2** **White-Box Testing**

A software testing method in which the tester tests through the actual line of codes instead of testing through the User Interface. For this to be tested, the tester has to be study and understand the programming style and languages used. Also, the tester can see clearly which specific codes are being performed and which ones are not if it needs to be.

With white-box testing, the tester can test to see if certain logic is being executed properly. An example of why this is advantageous is whereby an if statements consist of a parameter for comparison. Checking the “&&” and “||” operators being used appropriately can help negate huge problems that might occur. White-box testing also allows our group to check if the SQL codes are being used correctly.

## 2.3 Room Availability and Booking

For room availability and booking, we ensured the system should accept the data inputs like first name, last name, number of adult guest and children, phone number email and address, mailing address include street block house number, postal code and country. Additional remarks such as whether king or queen sized is needed and a smoking room or non-smoking room. Check in details like check in date and time and desired check out date and time are also included. As for payment, the system should accept details like credits card number, cardholder name and expiration date. Payment by cash is also an available option.

|  |  |
| --- | --- |
| Test Case | Login function throws SQLException (an example of exception case handling) |
| Test Unit | Room Availability and Booking |
| Method | Click btn Cfm payment |
| Input Initialization | Both textBox1 Text and textBox2 text are carrying the credits card number and pin object |
| Expected Output | Return error message and to re-enters details |
| Testing Procedure | Disable the SQL database on the server and simulate a payment confirmation calling the payment method. |

## 2.4 Housekeeping and staff management

**Housekeeping**

The housekeeping management was the biggest improvement made on the system. Housekeeping module was not a part of the original system. This means that we set more time on understanding and studying the codes first before implementing the various refinements.

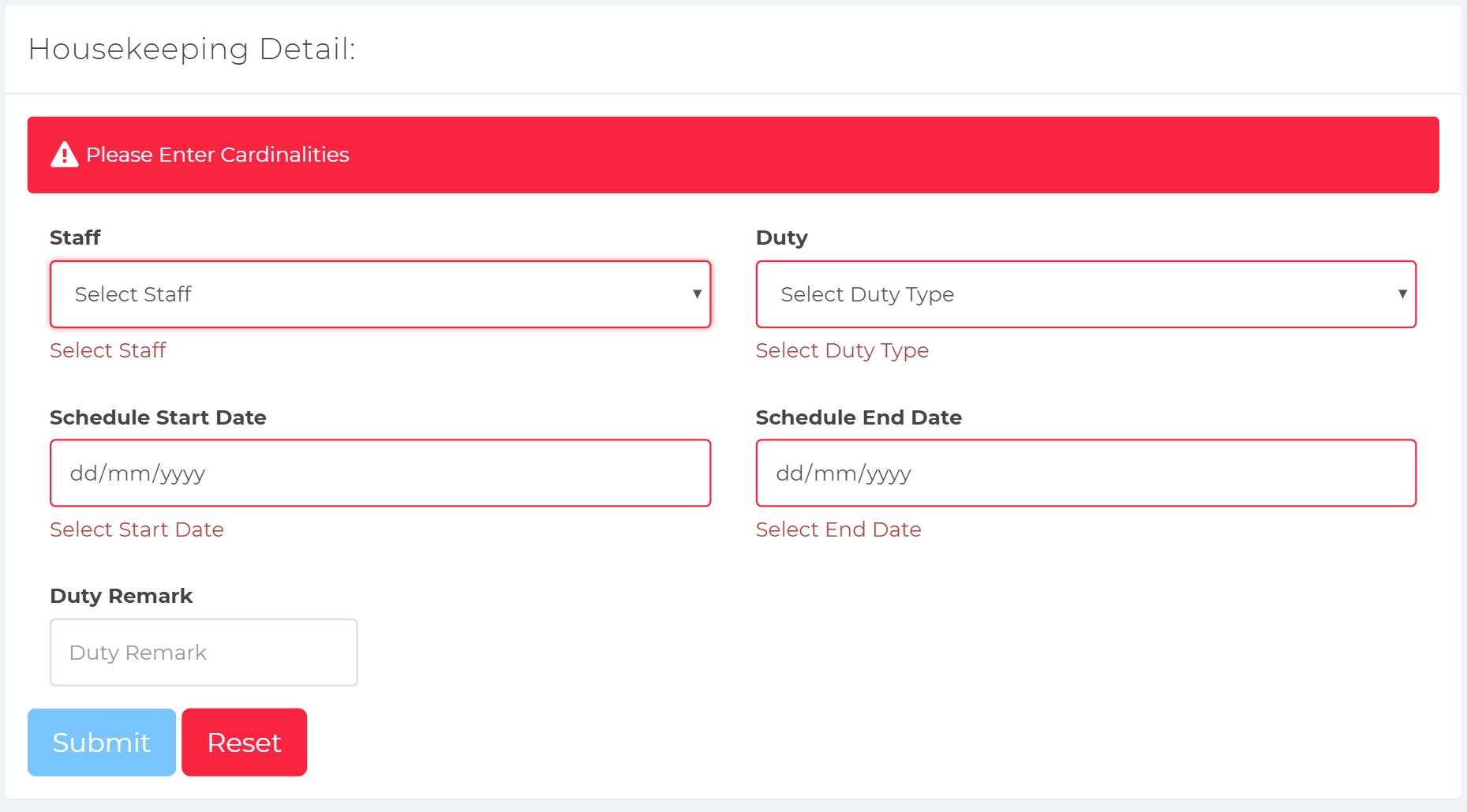
For housekeeping management, we ensured the database table of “housekeeping” would have an ID, schedule start date, schedule end date, duty remarks and foreign keys. The foreign keys would be the employee ID from the staff table and a duty ID from the duty table. The duty table consists of the 4 different types of duties; general maintenance, room maintenance, estate maintenance and security which will be shown in a drop-down list.

Unit Testing: **Black-Box**

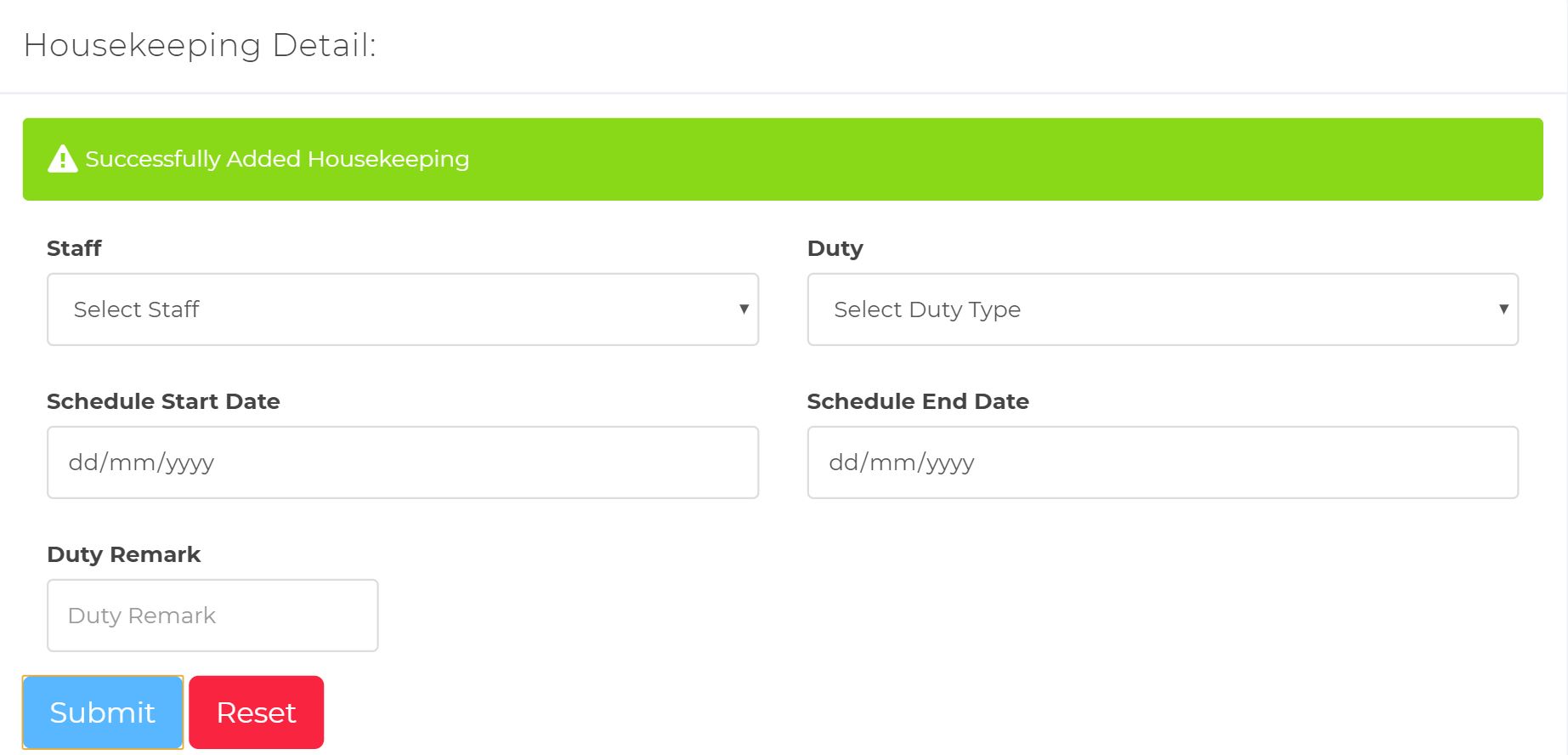
**Add Housekeeping**

The files used to perform this function are mainly add\_housekeep.php, ajax.js and ajax.php.

If the user were to leave the fields blank in this form, the text boxes are highlighted and error messages are shown together with an alert box in red. An exception of this is the “Duty Remark” as it is an optional data to input. The submit button will also be faded out to indicate that the form cannot be submitted.



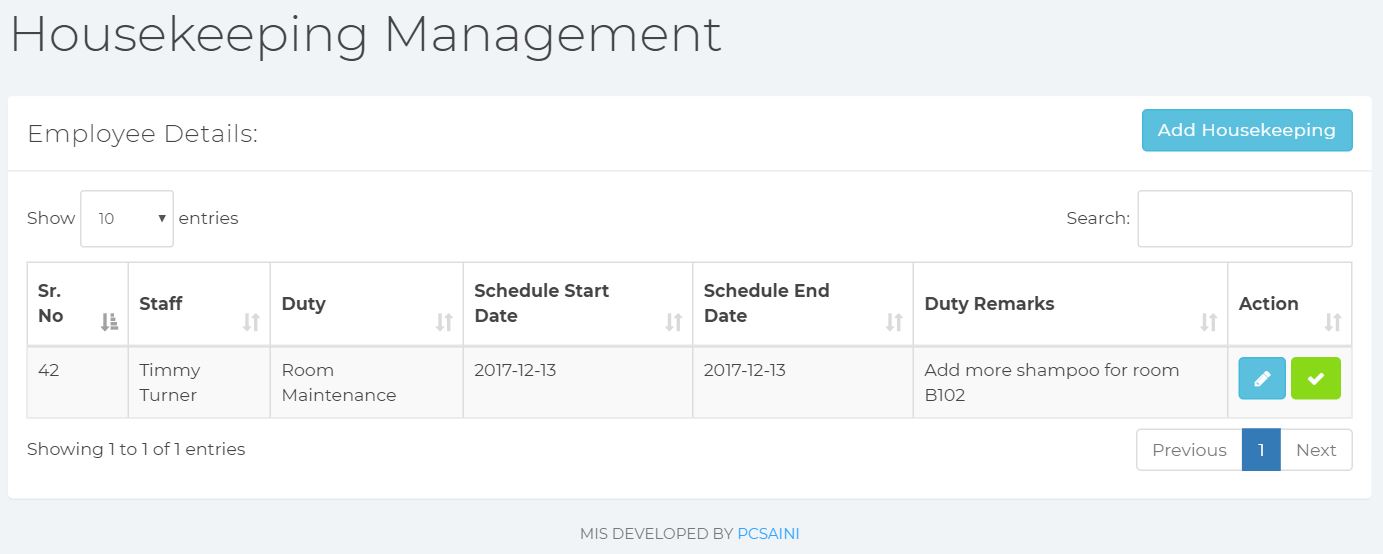
Vice versa, when the user selectss and includes the data in the textboxes, the submit button will become active, which means that the button can be clicked. If the form is successfully submitted, a green alert box will be shown to let the user know of it.



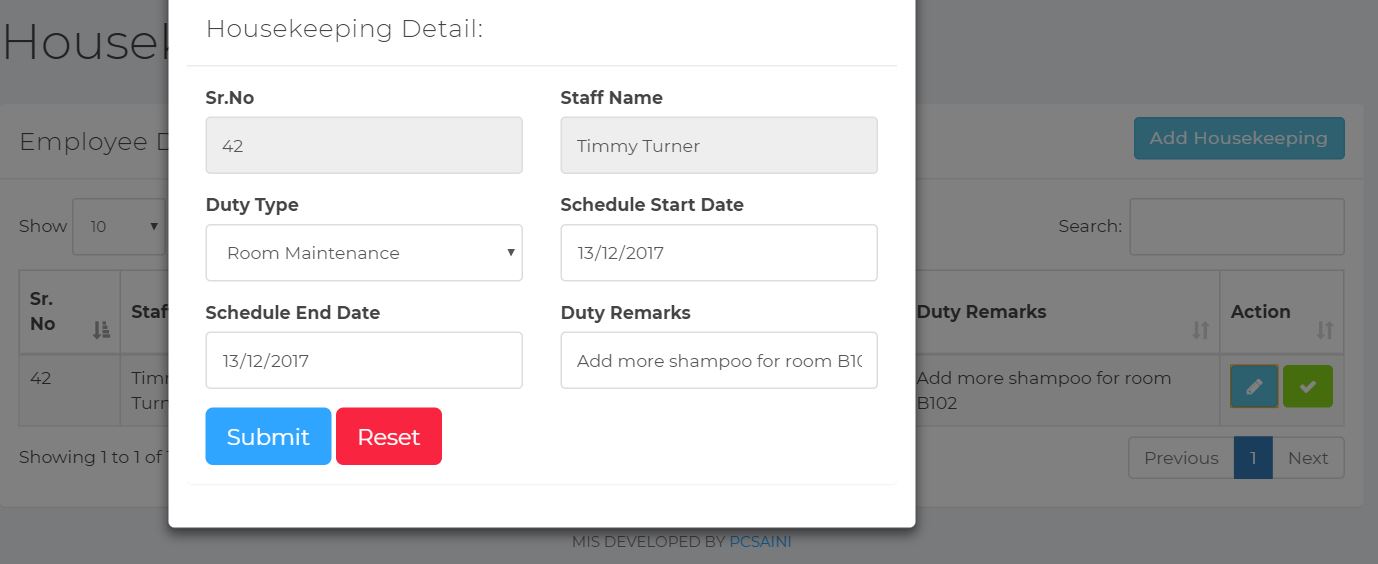
**View or Edits Housekeeping**

The files used to perform this function are mainly housekeeping.php and functionmis.php.

The users are can view the added housekeeping details here in a table format. The Manager users can edit or complete the duty with the buttons in the “Action” column. The Cleaning users will not be able to view the “Action” column.



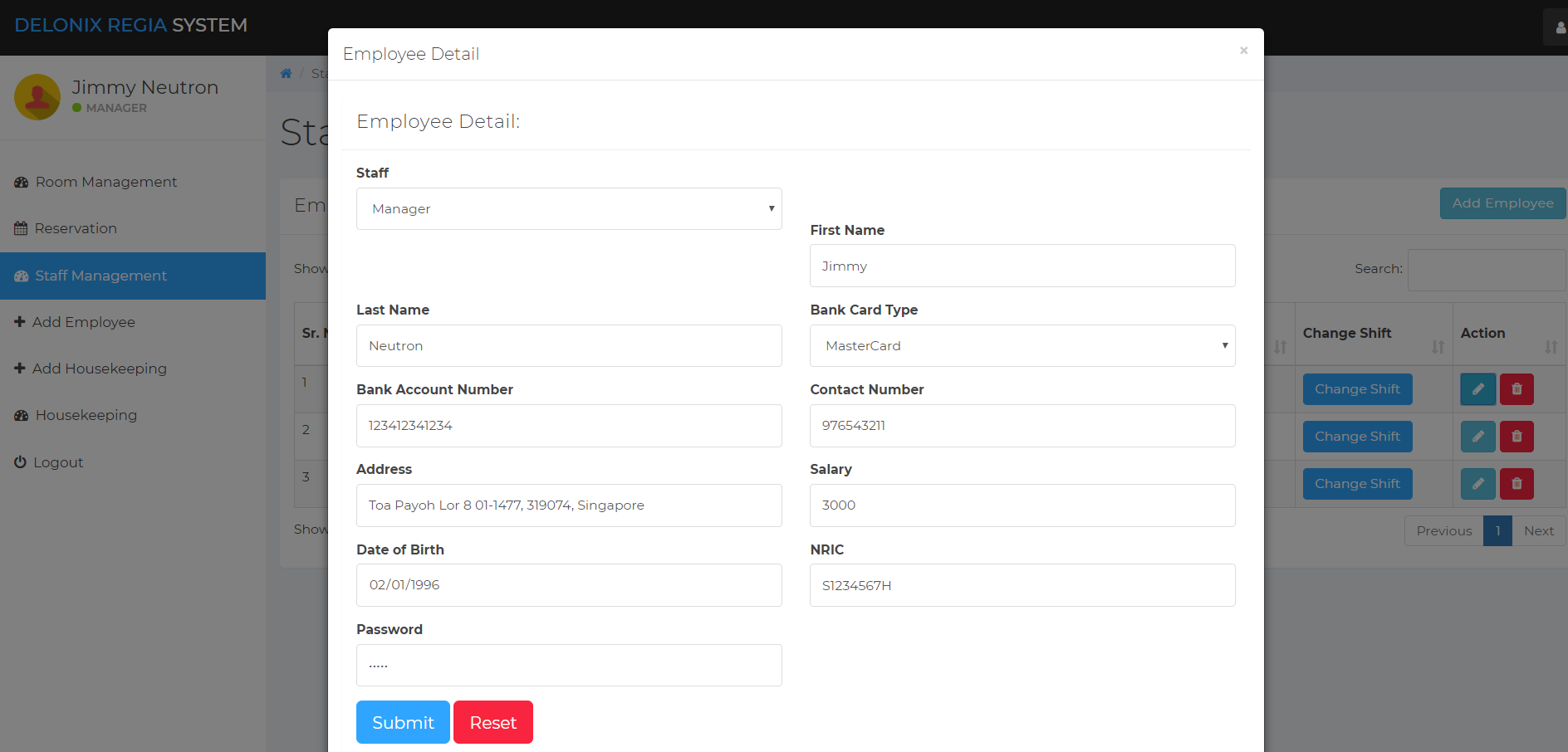
When the user selectss on the edits icon in the “Action” column, a modal appears and the user can edit and update the details in a form. The “Sr. No” and “Staff Name” textboxes are disabled as updating these will cause an error in the database. The input fields cannot be left as empty as the other required textboxes are in a drop-down list and a date format. The reset button will reset the form to its original data.



**Staff Management**

Files used to perform this module are mainly ajax.php. ajax.js, staff\_mang.php, functionmis.php.

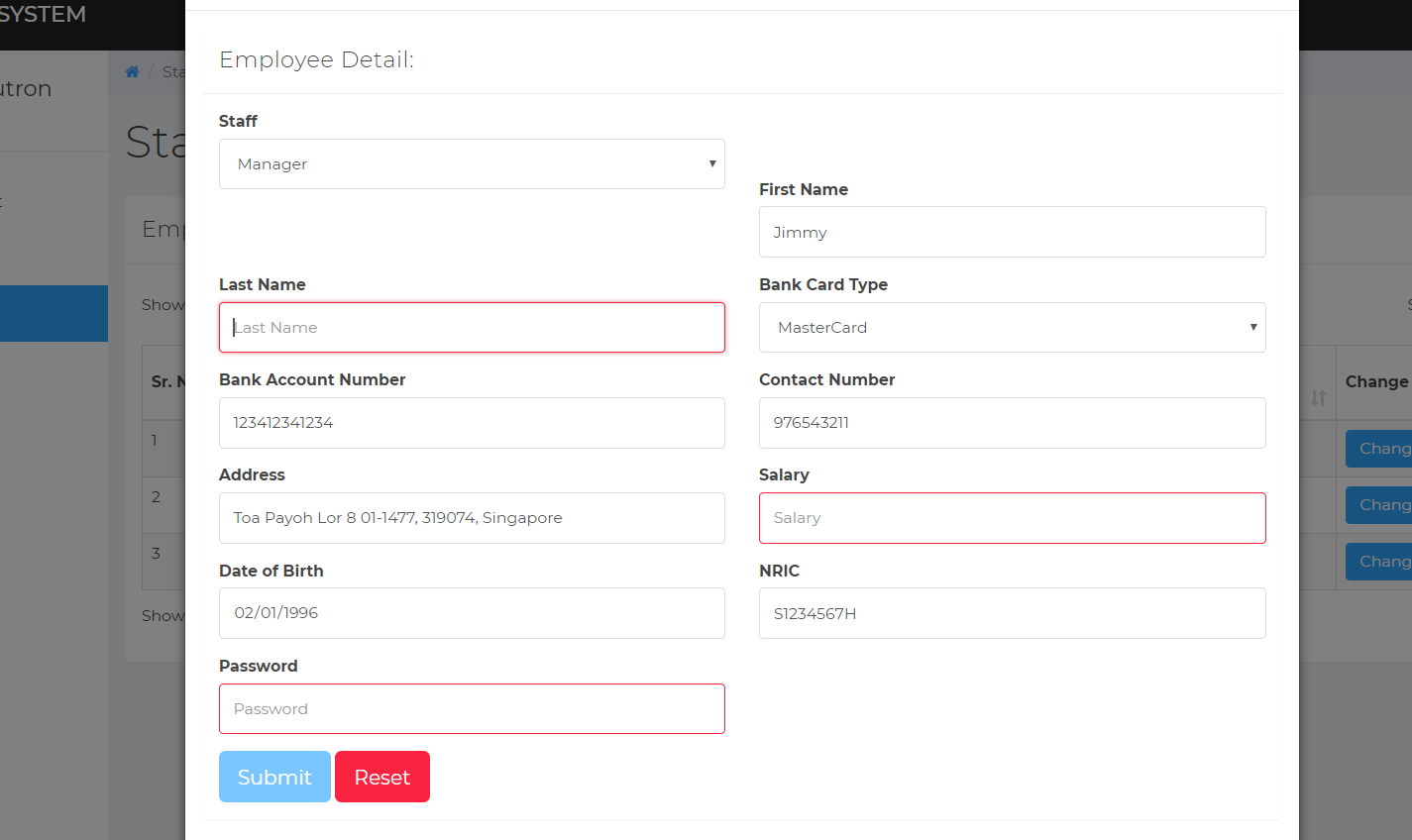
We have improved the Staff management module from the original system by allowing the Manager to edit more details about the Staff when they press the edits button. The edits button will pop up a modal which is a box consisting of all the details that the Manager can edits.



Unit Testing: **Black-Box** and **White-box**

**Updating Staff Details**

If the user were to leave any text field blank and click on the Submit button, an error will occur and the text field that is blank will be highlighted in red, indicating that the user will have to input something in it.



## 2.5 User account and login creation

Files used to perform this module are mainly ajax.php, ajax.js, login.php, add\_emp.php.

We have improved the account creation module of the original system by adding in the feature to restrict certain access rights based on the type of staff they are.

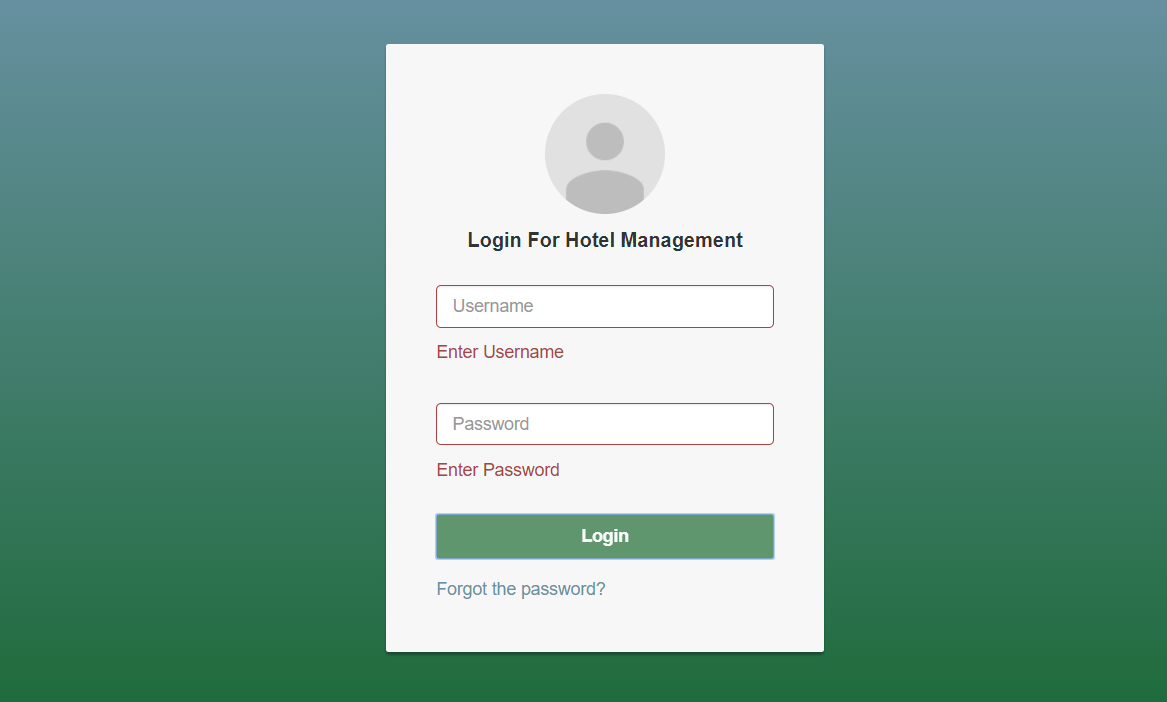
We have 3 types of staff that uses the system and they are the, Manager, Cleaning and Reception users. Only the Manager will be able to create an account for the staffs of the hotel.

For Manager, they will have access rights to all 3 modules and the ability to modify the data inside those modules (Room Availability and booking, Housekeeping and staff management, User account and login creation). For Cleaning, they will only have access rights to 1 module (Housekeeping), however they will not be able to modify the data stored in that module. For Reception, they will also only have access rights to 1 module (Room Availability and booking) but they are able to modify the data stored in that module.

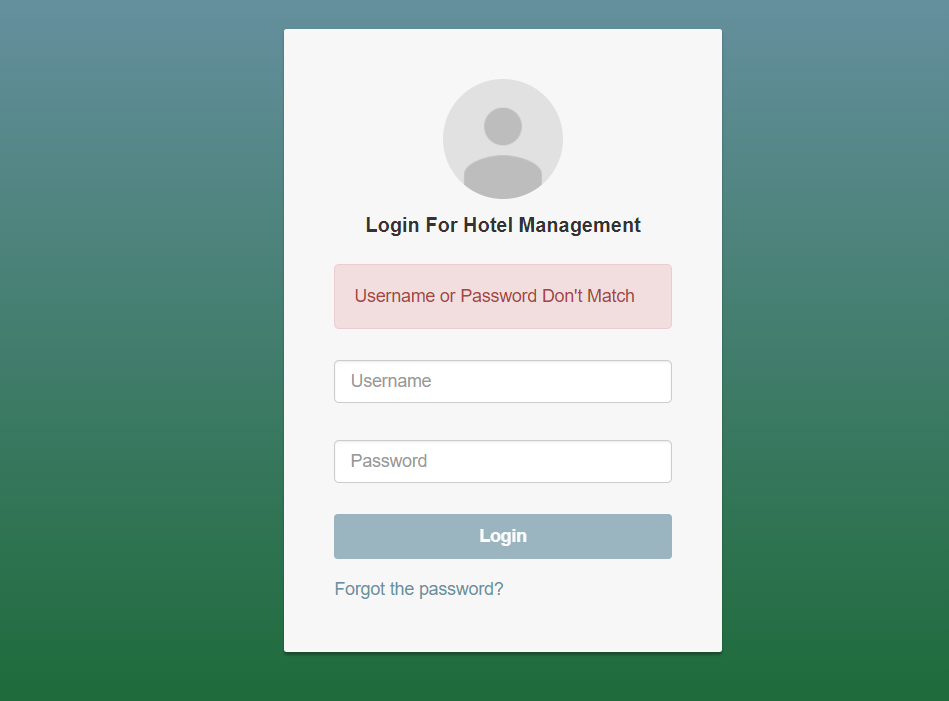
Unit Testing: **Black-Box** and **White-box**

**Login**

At the login screen, if the user were to leave the NRIC and Password text field empty, and click the submit button, they will not be able to get through due to the validation that are included in the text field which will not perform any action until the text fields are not empty.

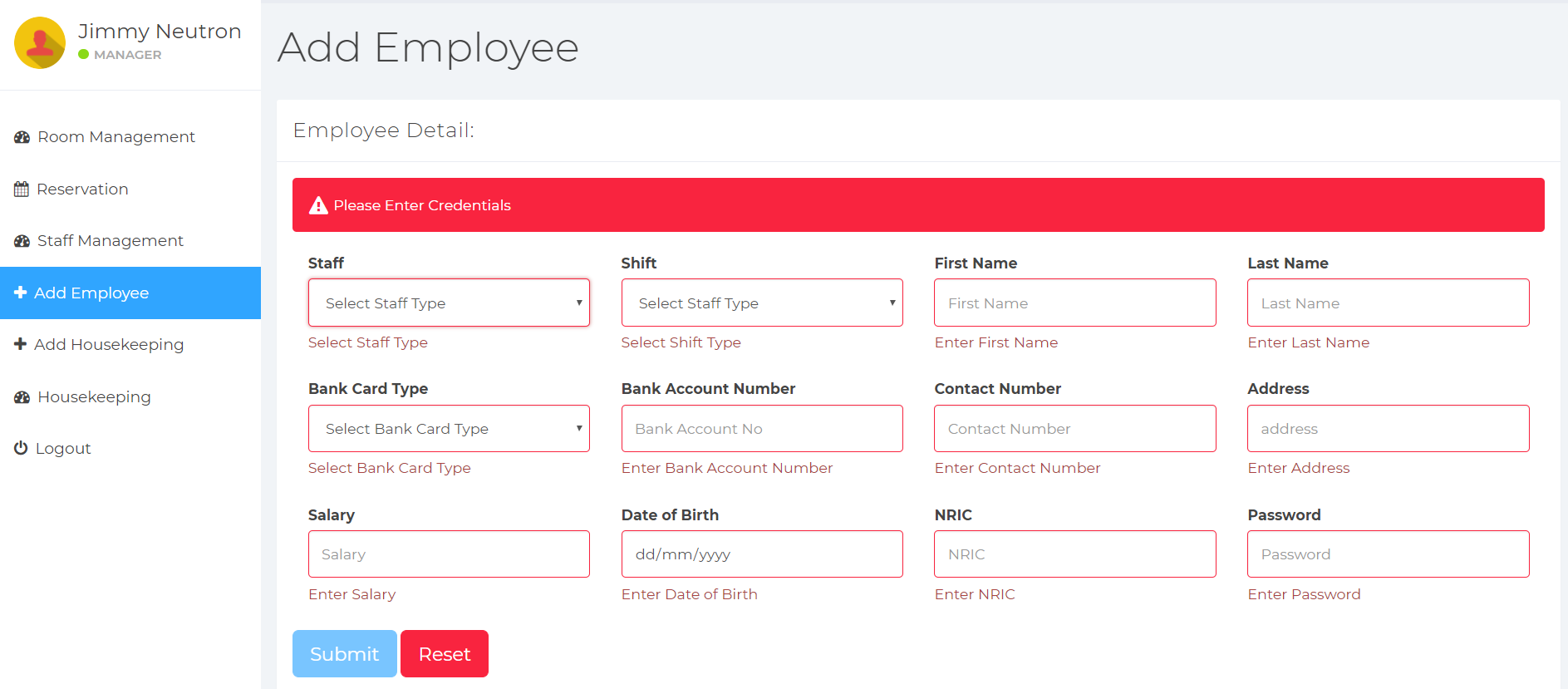


If the user were to input some login credentials that does not matched the ones found in the database, an error message will pop up at the top and the user will not be able to get through to the website.



**Account Creation**

When creating an account for the staffs, the user will have to input data such as (Field name here). However, if the user were to leave those text fields empty and click submit, they will encounter an error due to the validation that are included in the text fields and an error message will show up indicating that the user should enters the credentials.



# SYSTEM INTEGRATION

We have decided to distribute the workload of programming the website by separating the User Interface (UI) and Data Structure. UI will provide the front-end users the views and interface needed to create or modify the data that will be stored in the database of the Data Structure.

The integration will be completed by working on the Data Structure only after adding in the User Interface. This way we can know which data type needs to be used, hence, allowing us to create a proper Data Structure.

We make use of our Github repository to share the program codes with each other to add/update the necessary functions needed.

# TEST LOG

## Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S/No | Steps | Test Data | Expected Results | Actual Results |
| 1 | The user enters the NRIC of the staff | 1) S1234567H  2) dummytext  3) NULL | 1) Login success  2) Username or Password Don’t Match  3) Enters NRIC, Enters Password | 1) Login success  2) Username or Password Don’t Match  3) Enters NRIC, Enters Password |
| 2 | The user enters the Password of the staff | - jimmy  - dummytext  - NULL | 1) Login success  2) Username or Password Don’t Match  3) Enters NRIC, Enters Password | 1) Login success  2) Username or Password Don’t Match  3) Enters NRIC, Enters Password |

## Add Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S/No | Steps | Test Data | Expected Results | Actual Results |
| 1 | The user selects Staff Type | 1) Manager 2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |
| 2 | The user selects Shift type | 1) Morning  2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |
| 3 | The user enterss First Name | 1) Test  2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |
| 4 | The user enters Last Name | 1) Test  2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |
| 5 | The user selects Bank Card Type | 1) MasterCard  2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |
| 6 | The user enters Bank Account Number | 1) 1234123412341234  2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |
| 7 | The user enters Contact Number | 1) 99988876  2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |
| 8 | The user enters Address | 1) Pasir ris blk 123  2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |
| 9 | The user enters Salary | 1) 1000  2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |
| 10 | The user selects Date of Birth | 1) 08/06/1997  2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |
| 11 | The user enters NRIC | 1) S6789054Y  2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |
| 12 | The user enters Password | 1) password  2) NULL | 1)Employee Successfully added  2) Please Enters Credentials | 1)Employee Successfully added  2) Please Enters Credentials |

## Edits Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S/No | Steps | Test Data | Expected Results | Actual Results |
| 1 | The user edits Staff Type | 1) Cleaning 2) NULL | 1) Success  2) Error | 1) Success  2) Error |
| 2 | The user edits Shift type | 1) Evening  2) NULL | 1) Success  2) Error | 1) Success  2) Error |
| 3 | The user edits First Name | 1) Test2  2) NULL | 1) Success  2) Error | 1) Success  2) Error |
| 4 | The user edits Last Name | 1) Test2  2) NULL | 1) Success  2) Error | 1) Success  2) Error |
| 5 | The user edits Bank Card Type | 1) Visa  2) NULL | 1) Success  2) Error | 1) Success  2) Error |
| 6 | The user edits Bank Account Number | 1) 1111222233334444  2) NULL | 1) Success  2) Error | 1) Success  2) Error |
| 7 | The user edits Contact Number | 1) 911822733  2) NULL | 1) Success  2) Error | 1) Success  2) Error |
| 8 | The user edits Address | 1) Tampines blk 123  2) NULL | 1) Success  2) Error | 1) Success  2) Error |
| 9 | The user edits Salary | 1) 1500  2) NULL | 1) Success  2) Error | 1) Success  2) Error |
| 10 | The user edits Date of Birth | 1) 01/02/1990  2) NULL | 1) Success  2) Error | 1) Success  2) Error |
| 11 | The user edits NRIC | 1) S9647632Y  2) NULL | 1) Success  2) Error | 1) Success  2) Error |
| 12 | The user edits Password | 1) password2  2) NULL | 1) Success  2) Error | 1) Success  2) Error |

## Add Housekeeping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S/No | Steps | Test Data | Expected Results | Actual Results |
| 1 | The user selects Staff | 1) Timmy  2) NULL | 1) Successfully Added Housekeeping  2) Please Enters Cardinalities | 1) Successfully Added Housekeeping  2) Please Enters Cardinalities |
| 2 | The user selects Duty | 1) General Maintenance  2) NULL | 1) Successfully Added Housekeeping  2) Please Enters Cardinalities | 1) Successfully Added Housekeeping  2) Please Enters Cardinalities |
| 3 | The user selects Schedule Start Date | 1) 10/12/2017 2) NULL | 1) Successfully Added Housekeeping  2) Please Enters Cardinalities | 1) Successfully Added Housekeeping  2) Please Enters Cardinalities |
| 4 | The user selects Schedule End Date | 1) 11/12/2017 2) NULL | 1) Successfully Added Housekeeping  2) Please Enters Cardinalities | 1) Successfully Added Housekeeping  2) Please Enters Cardinalities |
| 5 | The user selects Duty Remark | 1) Change bed sheets 2) NULL | 1) Housekeeping Successfully added  2) Housekeeping Successfully added | 1) Housekeeping Successfully added  2) Housekeeping Successfully added |

**Edits Housekeeping**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S/No | Steps | Test Data | Expected Results | Actual Results |
| 1 | The user selects Duty | 1) Room Maintenance | 1) Success | 1) Success |
| 2 | The user selects Schedule Start Date | 1) 11/12/2017 | 1) Success | 1) Success |
| 3 | The user selects Schedule End Date | 1) 12/12/2017 | 1) Success | 1) Success |
| 4 | The user selects Duty Remark | 1) Add shampoo | 1) Success | 1) Success |

## 

## Room Reservation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** |
| 1 | The system will record the check-in-date | 4/12/2017 | 04/12/2017 | 04 12 2017 |
| 2 | The system will record the check-out date | 6/12/2017 | 06/12/2017 | 06 12 2017 |
| 3 | The system will record the customer’s first name | John | John | John |
| 4 | The System will record the customer’s last name | Tan | Tan | Tan |
| 5 | The system will record number of adults or children | 2adult/1child | 02/01 | 02 01 |
| 6 | The system will record the customer’s email | johntan@gmail.com | johntan@gmail.com | johntan@gmail.com |
| 7 | The system will record the customer’s phone number | 12345678 | 12345678 | 12345678 |
| 8 | The system will record customer’s address | 12 abc road | 12 abc road | 12 abc road |
| 9 | The system will record king or queen size bed (if requested) | 1 king | 1 king | 01k |
| 10 | The system will record smoking or non-smoking | Smoking | S | S |

## 

# 

# Reference

What is black box testing? Black box testing types and techniques. (n.d.). Retrieved from <http://www.softwaretestinghelp.com/black-box-testing/>

How to Perform White Box Testing – Explained with a Simple Example. (n.d.). Retrieved from <http://www.softwaretestinghelp.com/white-box-testing-techniques-with-example/>

P. (2017, December 12). Pcsaini/hotel\_management\_system. Retrieved from <https://github.com/pcsaini/hotel_management_system>

## 

*…*