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** |
| 1603625C | Ch’ng Wai Kit Wesley |
| 1601705E | Qamarul Fattah Bin Hamdan |
| 1601553I | Yip Xiu Han |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 22/11/17 | 1.0 | Initial Draft | ALL |
| 6/12/2017 | 1.1 | Updated Workload Distribution | ALL |
| 12/12/2017 | 1.2 | Vetted Document | Xiu Han |
| 13/12/2017 | 1.3 | Compiled Document | Wesley |

**Table of Contents**

[**DISTRIBUTION OF WORKLOAD**](#_uc5ao1l7btxg)4

[**MODULE DEVELOPMENT AND UNIT TESTING**](#_xwmgxwfuw26i)4

[Room Availability and Booking Module](#_2et92p0) 5

[Housekeeping and Staff Management Module](#_8uf4gywl221o) 5

[Reporting Module](#_wvnksoxpnp9x) 5

[**SYSTEM INTEGRATION**](#_865n1tz89sfl)5

[**TEST LOG**](#_3kwgbykjhh91)6

[Room Availability and Booking Module](#_pbhdi6467q53) 7

[4.1.1 - Login (Empty Fields)](#_dgshjfoqc6vh) 7

[4.1.2 - Login (Non Existing Data)](#_tky821obzmnc) 7

[4.1.3 - Login (Existing Data, Correct Username, Incorrect Password)](#_wr2owp8u9eo3) 8

[4.1.4 - Login (Existing Data, Correct Password, Incorrect Username)](#_vwpuyi8q4jdl) 8

[4.1.4 - Login (Existing Data, Correct Password, Correct Username)](#_nv4kbietmo6z) 8

[4.1.5 - Generate Room Category (Valid Inputs)](#_domqs1b3bmgi) 9

[4.1.6 - Generate Room Category (Invalid Input)](#_ebzcb9lqwm3l) 9

[4.1.7 - Retrieve Room Category](#_1zkmth3n79zw) 10

[4.1.8 - Update/Edit Room Category](#_ttl04nq4zv7l) 11

[4.1.9 - Booking (Book a Room)](#_3y3q41bc4ly0) 12

[4.1.10 - Booking (Retrieve Booking Information)](#_e2t8lrotyf5j) 13

[4.1.11 - Booking (Edit Booking Information)](#_txac08x88u9h) 13

[4.1.12 - Room Availability](#_ghekbk9a6k4h) 14

[Housekeeping and Staff Management Module](#_l3m11tx4s1q7) 15

[Reporting Module](#_uo2nxafrr86s) 16

# DISTRIBUTION OF WORKLOAD

|  |  |
| --- | --- |
| **Construction & Testing** | **Members** |
| Module Development And Unit Testing | Wesley |
| System Integration | Xiu Han |
| Room Availability and Booking Module Test Log | Wesley |
| Housekeeping and Staff Management Module Test Log | Qamarul |
| Reporting Module Test Log | Xiu Han |
| Video Compilation and Editing | Qamarul |

# 

# MODULE DEVELOPMENT AND UNIT TESTING

The naming convention used in the programming of our software follows the “CamelCase” style of Java language. Further clarification is given in the table below.

|  |  |  |
| --- | --- | --- |
| **Identifier type** | **Rules for naming** | **Examples** |
| Classes | Class names should be nouns in UpperCamelCase, with the first letter of every word capitalised. | class Raster;  class ImageSprite; |
| Methods | Methods should be verbs in lowerCamelCase or a multi-word name that begins with a verb in lowercase; that is, with the first letter lowercase and the first letters of subsequent words in uppercase. | run();  runFast();  getBackground(); |
| Variables | Local variables, instance variables, and class variables are also written in lowerCamelCase. Variable names should not start with underscore (\_) or dollar sign ($) characters, even though both are allowed. This is in contrast to other coding conventions that state that underscores should be used to prefix all instance variables. | int i;  char c;  float myWidth; |
| Constants | Constants should be written in uppercase characters separated by underscores. Constant names may also contain digits if appropriate, but not as the first character. | static final int MAX\_PARTICIPANTS = 10; |

We have used our prevailing knowledge from the Dynamic Web Application Development subject to write this program, making use of .html .css .php programming languages to code this. We will using XAMPP with “Apache” and “MySQL” as a means of storing user information.

## Room Availability and Booking Module

For the Room Availability and Booking Module, I have gone online and carried out the process flow of unit testing by trying out other reputable established businesses like “Resorts World Sentosa”, taking note of the UI design as well as the flow of test logs. Module development was done by utilizing the knowledge of SWEN 1 and 2, which helped us get past the planning phase. We have also referred to the lecture notes provided to us on LMS to ensure that we were on the right track, as well as consulted our lecturer frequently.

## Housekeeping and Staff Management Module

To test and see if the Housekeeping module is working, we will enter a data into a form which will add inventory into the database. We will also enter a data into a form which will add staff into the database.

## Reporting Module

To test for the Reporting Module, a basic form will be submitted into the software’s database to see if the form input has been successfully registered without any errors. The print function of the report will also be tested as well.

# SYSTEM INTEGRATION

For our software, system integration has been carried out in the form of horizontal integration during the development process, whereby a specialized subsystem is developed and established specifically as the foundation and means of communication between all the other developed modules. This makes it so that each of the modules that we develop are essentially made with the same framework so that variables and methods remain uniform when the connected modules interact via this subsystem.

Additionally, all modules of the software have also been coded in the same language of html, css and php. This is done to avoid any naming collisions that may arise from working with similar pertaining data within the software, as well as to enhance the clarity of variables and methods of the modules in cases of potential ambiguity when combining the modules during system integration.

Furthermore, integration testing has also been carried out for the overall software after all modules of the software have been developed and tested for any major errors. Integration testing refers to the phase in system integration where the individual software modules are combined and tested as a group after unit testing has been carried out for each module, as elaborated in the Test Log below.

# TEST LOG

## Room Availability and Booking Module

## 4.1.1 - Login (Empty Fields)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | User keys in login information and selects “Submit” button. | Username/Email:  -NIL-  Password: -NIL- | Alert appears referring to the input fields to “Please Fill Out This Field” | | Basic Flow, Successful  Data of user accounts must be present in the database of hotel.sql in order for login to function |
| 2 | Username/Email:  -NIL-  Password:  1234 |
| 3 | Username/Email:  admin  Password:  -NIL- |

## 

## 4.1.2 - Login (Non Existing Data)

## 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | User keys in login information and selects “Submit” button. | Username/Email:  testing  Password:  1234 | Alert with red text appears referring to the input fields to “Wrong username or password” | | Basic Flow, Successful  Data of user accounts must be present in the database of hotel.sql in order for login to function |

## 

## 4.1.3 - Login (Existing Data, Correct Username, Incorrect Password)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | User keys in login information and selects “Submit” button. | Username/Email:  admin  Password:  12345 | Alert with red text appears referring to the input fields to “Wrong username or password” | | Basic Flow, Successful  Data of user accounts must be present in the database of hotel.sql in order for login to function |

## 

## 4.1.4 - Login (Existing Data, Correct Password, Incorrect Username)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | User keys in login information and selects “Submit” button. | Username/Email:  testing  Password:  1234 | Alert with red text appears referring to the input fields to “Wrong username or password” | | Basic Flow, Successful  Data of user accounts must be present in the database of hotel.sql in order for login to function |

## 

## 4.1.4 - Login (Existing Data, Correct Password, Correct Username)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | User keys in login information and selects “Submit” button. | Username/Email:  admin  Password:  1234 | User is redirected to the “Admin” tab whereby he/she can carry out admin privileged test cases. | | Basic Flow, Successful  Data of user accounts must be present in the database of hotel.sql in order for login to function. |

## 

## 4.1.5 - Generate Room Category (Valid Inputs)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Assuming user is logged in, user clicks on “Add Room Category” button in the “Admin” tab. | -NIL- | Information is filled and submitted successfully to the database. Notification pops up saying “Room has been added successfully!”.  This can be checked by navigating to “localhost/phpmyadmin > hotel > room\_category”. | | Main Flow, Successful  This will update the hotel.sql database with whatever values have been inputted under the “room\_category” tab. |
| 2 | User is redirected to the “addroom.php” file, where the user keys in specified information. | Room Type Name: Deluxe  No.of Rooms: 1  No. of Beds: 2  Bed Type: Single  Facility: Testing  Price Per Night: 1337 |
| 3 | User clicks the “Add” button on the bottom right. |

## 

## 4.1.6 - Generate Room Category (Invalid Input)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Assuming user is logged in, user clicks on “Add Room Category” button in the “Admin” tab. | -NIL- | Information is filled and submitted successfully to the database. Notification pops up saying “Room has been added successfully!”.  This can be checked by navigating to “localhost/phpmyadmin > hotel > room\_category”.  Note that if a String is entered into the “Price Per Night” field, that field will automatically reset the price to 0. | | Main Flow, Successful  This will update the hotel.sql database with whatever values have been inputted under the “room\_category” tab. |
| 2 | User is redirected to the “addroom.php” file, where the user keys in specified information. Deliberately writing a String in the “Price” field. | Room Type Name: Deluxe  No.of Rooms: 1  No. of Beds: 2  Bed Type: Single  Facility: Testing  Price Per Night: String instead of integer |
| 3 | User clicks the “Add” button on the bottom right. |

## 

## 4.1.7 - Retrieve Room Category

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Assuming user is logged in, user clicks on “Show All Room Category” button in the “Admin” tab. | -NIL- | Information already present in the database will be displayed. | | Main Flow, Successful  This will display whatever information is already presented in the hotel.sql database room\_category tab. |
| 2 | User is redirected to the “show\_room\_cat.php” file, where the user is able to edit the data. | Room Type Name: Deluxe  No.of Rooms: 1  No. of Beds: 2  Bed Type: Single  Facility: Testing  Price Per Night: 1337 |

## 

## 4.1.8 - Update/Edit Room Category

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Assuming user is logged in, user clicks on “Edit Room Category” button in the “Admin” tab. | -NIL- | Information is updated and submitted successfully to the database. Notification pops up saying “Updated Successfully!”.  This can be checked by navigating to “localhost/phpmyadmin > hotel > room\_category”. | | Main Flow, Successful  This will update the hotel.sql database with whatever values have been inputted under the “room\_category” tab. |
| 2 | User is redirected to the “show\_room\_cat” file, where the user is able to edit the data. | Room Type Name: Deluxe  No.of Rooms: 1  No. of Beds: 2  Bed Type: Single  Facility: Testing  Price Per Night: 420 |
| 3 | User clicks the “Edit” button on the right side of the screen pertaining to the room category they wish to edit. |
| 4 | User fills in the information which he/she wishes to alter. In this example he/she updates the price of the room from “1337” to “420” and clicks “Update” button on the bottom right. |

## 

## 4.1.9 - Booking (Book a Room)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Assuming user is logged in, user clicks on “Room & Facilities” tab under the navigation bar. | -NIL- | Information is updated and submitted successfully to the database. Notification pops up saying “Your Room has been booked!”.  This can be checked by navigating to “localhost/phpmyadmin > hotel > rooms”. | | Main Flow, Successful  This will update the hotel.sql database with whatever values have been inputted under the “room” tab. |
| 2 | User is redirected to the “room.php” file, where the user is able to book the rooms. | Check-In Date: 12/12/2017  Check-Out Date:  13/12/2017  Full Name:  John Wick  Phone Number:  82138248 |
| 3 | User clicks the “Book Now” button on the right side of the screen pertaining to the room category they wish to book a room for. |
| 4 | User fills in the information which he/she require to provide and clicks “Book Now”. |

## 

## 4.1.10 - Booking (Retrieve Booking Information)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Assuming user is logged in, user clicks on “Show all Booked Rooms” tab under the “Bookings” title in “Admin”. | -NIL- | Information is displayed from the database.  This can be checked by navigating to “localhost/phpmyadmin > hotel > rooms”. | | Main Flow, Successful  This will display whatever booked room information is present in the hotel.sql database.. |
| 2 | User is redirected to the “show\_all\_room.php” file, where the user is able to retrieve all bookings that have been made by staff. | Check-In Date: 12/12/2017  Check-Out Date:  13/12/2017  Full Name:  John Wick  Phone Number:  82138248 |

## 4.1.11 - Booking (Edit Booking Information)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Assuming user is logged in, user clicks on “Edit Booked Room” tab under the “Bookings” title in “Admin”. | -NIL- | Information is updated and successfully submitted to the database. Notification pops up saying “Your changes have been saved!”  This can be checked by navigating to “localhost/phpmyadmin > hotel > rooms”. | | Main Flow, Successful  This will update whatever booked room information is present in the hotel.sql database.. |
| 2 | User is redirected to the “show\_all\_room.php” file, where the user is able to edit all bookings that have been made by staff. | Check-In Date: 12/12/2017  Check-Out Date:  13/12/2017  Full Name:  John Wick  Phone Number:  82138248 |
| 3 | User clicks on “Edit” button on the right side of the screen pertaining to whichever booking he/she wishes to edit. |
| 4 | User fills in whatever information he/she wishes to alter and clicks the “Update” button on the bottom right of the page. |  |

## 4.1.12 - Room Availability

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Assuming user is logged in, user clicks on “Online Reservation” tab in the navigation bar. | -NIL- | Information is displayed pertaining to what room category and what room date the booking is available on. | | Main Flow, Unsuccessful  The information entered does not coincide with hotel.sql and displays rooms that are already booked, as rooms that are available for booking. |
| 2 | User is redirected to the “reservation.php” file, where the user is able check with the system on the available dates for booking. | Check-In Date: 12/12/2017  Check-Out Date:  13/12/2017 |
| 3 | User fills in the stipulated Check-In and Check-Out dates and clicks on “Check Availability” button at the bottom of the screen. |

## Housekeeping and Staff Management Module

* + 1. **Add Item**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1.0 | The system shall record an item and the number of stock of items. | Toilet Rolls  11 | Toilet Rolls  11 | Toilet Rolls  100 | Main Flow, Unsuccessful  The information entered does not coincide with hotel.sql and displays stock of items as 100. |
| 1.1 | The system shall display the item and the number of stock of items | NIL | Information is displayed pertaining to the item and the quantity. | Only the correct item is displayed but the quantity is wrong. |

## 

* + 1. **Add Staff**

## 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1.0 | The system shall register a staff | testing | testing | testing | Main Flow, Successful  The information entered coincides with hotel.sql |
| 1.1 | The system shall display the staff list | NIL | Information is displayed pertaining to the name of the staff entered. | |

## Reporting Module

**4.3.1 - Generate Room Report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected/Actual Results** | **Remarks** |
| 1.0 | The user logs in to access the admin.php page through the admin tab. | - | The user is able to login successfully. | More details on the login process can be found above. |
| 1.1 | The user selects the add report function under the report generator header. | - | The user is redirected to the addreport.php page. | - |
| 1.2 | The user keys in all of the required information to generate the report. | Room Type Name: Template  No. Of Guests in Room: 2  No. Of Beds: 2  Room Type: Double  Facilities Available: Toilet  Price: 20 | The user fills in each component of the form with the correct variable types (E.g. only integers in the quantity input). | If the price field is inputted with a variable type other than an integer, the system will register it as an integer of 0 instead of invalidating the entire form. |
| 1.3 | The user clicks on the add button to submit the report form data into the SQL database. | - | The report generation is successful and is confirmed with a browser notification. | The inputted information of the report generation can be found in the SQL database under the hotel table in localhost/phpmydamin. |

## 

**4.3.2 - Print Room Report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected/Actual Results** | **Remarks** |
| 1.0 | The user logs in to access the admin.php page through the admin tab. | - | The user is able to login successfully. | More details on the login process can be found above. |
| 1.1 | The user accesses the administration tools by going into localhost/phpmyadmin. | - | The user is able to navigate to the administration page successfully. | The user should only be able to access this page as the hotel owners with the right privileges. |
| 1.2 | The user clicks on the hotel table and selects the print button under the table that is to be printed. | - | A preview of the report is presented to the user. | - |
| 1.3 | A print confirmation model is brought up and the user selects print again to confirm printing. | - | The print request is successfully sent to the local printer and the report is printed. | The user can select the print destination, pages to print, layout as well as color. |

## 