Temasek Polytechnic

School of Informatics and IT

**Diploma in Information Technology (IT)**

Software Test Specifications (STS)

**Project Particulars**

|  |  |
| --- | --- |
| **Tutor** | Mr Mel Goh |
| **Class** | P03 |
| **Project Title** | Delonix Regia Hotel Management System |

**Project Team’s Particulars**

|  |  |
| --- | --- |
| **Matric Number** | **Student Name** |
| **1505065A** | **Ho Wei Jian** |
|  | **Tuitus Sim** |
| **1506538F** | **Sim Zhan An** |
| **1506659J** | **Kiew Yao Hong** |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 23/05/17 | 0.1 | Added required interface design for modules | Zhan An |
| 26/05/17 | 1.0 | Added compiled database required for program | Zhan An |
| 27/05/17 | 1.1 | Added CRUD functions to room status, booking, guest management, staff management to program | Zhan An |
| 28/05/17 | 1.2 | Updated CRUD features to program after trial test | Zhan An |
| 30/05/17 | 1.3 | Added required interface design for modules | Zhan An |
| 31/05/17 | 1.4 | Added login CRUD feature to program | Zhan An |
| 02/06/17 | 1.5 | Added retrieve function for reporting module | Zhan An |
| 05/06/17 | 2.0 | Updated with system test to final program | Zhan An |
|  |  |  |  |

**Table of Contents**

[1. DISTRIBUTION OF WORKLOAD 4](#_Toc484599518)

[2. MODULE DEVELOPMENT AND UNIT TESTING 6](#_Toc484599519)

[2.1 <Room Availability & Booking Module> 6](#_Toc484599520)

[2.2 <Housekeeping & Staff Management Module> 8](#_Toc484599521)

[2.3 <Login/Register Module> 9](#_Toc484599522)

[3. SYSTEM INTEGRATION 10](#_Toc484599523)

[4. TEST LOG 10](#_Toc484599524)

[4.1 <Scenario / Function 1 by Wei Jian> 11](#_Toc484599525)

[4.2 <Scenario / Function 2 by Tuitus > 12](#_Toc484599526)

[4.3 <Scenario / Function 3 by Yao Hong> 14](#_Toc484599527)

[4.4 <Scenario / Function 4 by Zhan An> 15](#_Toc484599528)

[5. Configuration Management 24](#_Toc484599529)

[6. References 25](#_Toc484599530)

# DISTRIBUTION OF WORKLOAD

|  |  |
| --- | --- |
| **Construction & Testing** | **Members** |
| * User Management   + CRUD of accounts   + Login/ Register (Authentication System) * Housekeeping & Staff Management Module   + (Housekeeping Module) CRUD for the 4 different duty types * Reporting Module   + Room Occupancy Statistics report (#4) * Software Testing Specification * Module 2 * System Integration * Test Log (Scenario /Function 2) * Configuration Management | Wei Jian |
| * Housekeeping & Staff Management Module   + Staff Management (CRUD) * Reporting Module   + Housekeeping report * Software Testing Specification * Module 2 * System Integration * Test Log (Scenario /Function 2) * Configuration Management | Tuitus |
| * Room Availability & Booking Module (Program)   + Booking (CRUD)   + Guest Management (CRUD) * Reporting Module   + Room Occupancy Report (R) * Software Testing Specification * Room Availability & Booking Module * System Integration * Test Log (Scenario /Function 4)   + Possible Scenarios / Alternate Flows (Booking)   + Possible Scenarios / Alternate Flows (Guest)   + Booking Module (CRUD)   + Guest Management (CRUD) * Configuration Management * Formatting & Compilation of Report | Zhan An |
| * Room Availability & Booking Module   + Room Status (CRUD) * Reporting Module   + Room report * Software Testing Specification * Module 1 * System Integration * Test Log (Scenario /Function 1) * Configuration Management | Yao Hong |

# MODULE DEVELOPMENT AND UNIT TESTING

## 2.1 <Room Availability & Booking Module>

In this module, it contains 2 areas of functionality that are required in the **module development** process. The first function is to manage room status of rooms in the hotel while the second function is to manage bookings as well as guest details.

In addition, the task is split into 2 parts where Yao Hong does the room status management part and Zhan An does the guest management part. This will then increase our efficiency as we will be able to merge those 2 together much easily and faster rather than 2 of us focusing on the same parts and afraid it might overlap.

Room Status Management

The room status management shows the different type of rooms available and what kind of rooms there are such as the number of occupants the room is available to, whether it is available or scheduled for cleaning up, is a smoking room when a guest / customer wants a smoking room. When the rooms are no longer available, the admin or management can delete the room or edit when it is not available for the time being.

Guest Management

In the guest management, users of the system can view and add details of guests to the system. Details of the guest includes their first name, last name, contact, email, address and date of birth. Apart from the creation and retrieval of guests in the system, the guest management allow users to edit the guest records if any unintentional human errors had been made. And lastly, users of the system can remove guest records of guest if they had left the hotel. Through this, we had derived that a create, retrieve, update and delete is required for the coding in the Visual Studio MVC Project and necessary details required in the SQL server.

Booking Management

In the booking management, users of the system can view existing or add new booking records to the system. Details of bookings include check-in date & time, check-out date & time, additional requests and check-out extension. Apart from creation and retrieval of bookings in the system, the booking management also allows users to edit booking records if any incorrect details were added. And finally, users of the system can also remove any booking records if the guest of a particular room had left the hotel. Through this, we had derived that a create, retrieve, update and delete is required for the coding in the Visual Studio MVC Project and necessary details required in the SQL server.

To ensure quality software and development and testing, we have used good programming styles such as naming convention and code refactoring.

1. Naming Convention

For the naming convention, the team members in charge of this module decided to use the lower camel case in developing. Lower camel case refers to the joining of 2 or more words as a compound word, and each word is capitalized in the compound with the word starting with a lower case.

Examples used in database:

Room:

Guest:

"firstName", "lastName", "contact", "email", "address", "dateOfBirth"

Booking:

"checkInDate", "checkOutDate", "additionalRequest", "checkoutExtension"

1. Code Refactoring

Code Refactoring is one of the effective ways improve the design of the code. And factoring allows our coding structure to be easily understandable and read by our group members when we have passed on the work to them.

In addition, it does not change the output of what is desired for the room status management.

One approach used for refactoring in guest management is by parameterizing several methods into 1 single method where they do similar things but each uses different values. For an instance, the guest table originally the fields consist of address street, address town, address postal code and address country and can be used into a single parameter called address instead. Though assumptions like records are to be added by receptionist, management or administrator only since they would know the format of the address field.

After implementing the good programming styles like lower camel case and code refactoring to our database in our SQL server database and codes in Visual Studio Project, we had carried out unit testing to ensure quality of software.

Unit Testing

Unit testing also known as component testing refers to the single testing of codes. A unit can be also referred as a single segment of code –a class, method or module and developers are responsible for it.

Unit testing involves the driver and stubs concept where it is used to replace the missing software and simulate the interface between the software components in a simple manner.

For an instance, if Zhan An were to developed the booking module, it would definitely require the IDs of room in the hotel so that guest could make booking. However, the room database which contains the room ID where Yao Hong is doing is not completed. To solve this, the stub known as dummy data is used. Since the booking module does not require multiple data except the room IDs, a separate test table containing dummy room ID data could be created in the meantime for unit testing before the actual integration of software.

**Black Box Testing**

**Input data validation**

**Illegal Data**

**Abnormal Data**

**Normal Data**

## 2.2 <Housekeeping & Staff Management Module>

For the housekeeping and staff management module, we decided to split the 2 modules and develop individually, the housekeeping module is done by Wei Jian and the Staff management module is done by Tuitus.

Housekeeping Module

I have used the MVC (Model, View, Controller) approach to develop the housekeeping module and the software Microsoft visual studio for development. Firstly, I created the staffs table in the database. This are fields in the table: First Name, Last Name, Duty Type. Secondly, once the database is done, I generate the models. Lastly, I change the methods in the controllers and views. My module allows the user to assign the different duty types (General Maintenance, Room Maintenance, Estate Maintenance and Security Maintenance) according to the employee, using a dropdown list.

I have also ensured that before I move on to the next phase of developing the program, I run the program to check whether there are any errors, if there is an error I fix it first before moving on. To ensure the module I develop has fewer bugs, as much as I can, I only change the necessary parts of the file generated by visual studio, as visual studio helps me to automatically generate the controller and the view according to the model.

To ensure the naming convention is standardized throughout the module, I have used the camel case method of naming my files, for example: Duty Type, First Name. The starting of a new word is always capitalized. To make sure the codes I used for developing the module is clear and easy for debugging, I removed all the unnecessary methods and gave meaningful name for the files in the module/

For unit testing, the only part that are different is the “Duty Type” field. The “Duty Type” field uses a drop down list instead of textbox. The objective of my unit testing is to make sure once the user selects the “Duty Type” and press the “Submit” button. The system is able to capture the “Duty Type” accurately. In the end it manage to work

Staff Management Module

The staff management contains personal details of the staff such as “First name, last name, bank account number and others. So basically, the staff management will allow specific people to do the 4 basic functions to the staff records. Create, the higher ups can create a specific staff and enter different type of details from staff id to phone numbers and others. Mr Wang can also Update the staff details in case the staff has change any personal details such as address or even phone number. Mr Wang can also read the details by viewing the staff page and look for different staffs. Mr Wang can delete staff details in case any of his staff decides to leave or is no longer working there.

Firstly, the database of the staff is created which will link to other tables, which are booking and housekeeping. In the database, the required details will be recorded through the staff form.

Also, the staff management module will also have the 4 basic functions like mentioned before therefore when creating a new staff, the admin must input the mandatory details such as staffID, first\_name, last\_name, date of birth, bank\_acc\_no, contact\_number, duty\_type. After which the details will be entered the system. For the unit testing of this module, dummy data will be added using the add “new staff” function.

Unit Testing

During the creation of staff process, all fields are mandatory and there is a required field called duty types which is linked to housekeeping which is currently not developed. However, since I would like to test if the creating function is working I would need the data of the housekeeping types from the housekeeping management function which is not created. Without that data or option, I would not be able to test my function which is the creating of staff.

For an instance, if Tuitus were to develop the staff management, it would require the duty types in the hotel so that admin can create the staff. However, the housekeeping database which contains the duty types where Wei Jian is doing is not completed. To solve this, the stub known as dummy data is used. Since the staff management module does not require multiple data except the duty types, a separate test table containing dummy duty type data could be created in the meantime before the actual integration of software.

Black Box Testing

Black box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings.

By using black

## 2.3 <Login/Register Module>

This module is an additional feature, as Mr Wang has mentioned in the video only administrator level and management users are able to access certain modules in the system. Hence, an authentication system needs to be in place for the system to verify the user. The register module can only be access by the system administrator, the system administrator gets to decide the type of access the user has in the system and is in charge of helping to register a new user.

Once I am done with the page view for the register page, I tried registering for an account. I made sure that fields of the user is being recorded in the database. Then after once the login page is done, I entered the username and password of the account I registered, to make sure the authentication system is working.

# SYSTEM INTEGRATION

After much discussion with everyone in the group we have decided for system integration, we are going to use the improved big bang integration method. To put it simply, improved big bang integration testing is a method whereby individual units are first test in unit testing before bringing all the testing units together to test the complete system.

In our case, one module can be further split into two, for example housekeeping and staff management has two parts. We decide who do which part and individually we move on to develop the module and do unit testing. So, once student A and B is done with developing the housekeeping and staff management module, they integrate it first. Finally, once everyone is done with the different modules (Reporting, Room Availability & Booking Management, Housekeeping and Staff Management) then we start integrating the system.

Besides the use of big bang integration, the platform of GitHub allows us to share and integrate our projects together for consistency in our work. As GitHub allows us to upload our work into their repository, it provides much more efficiency as it also sync our work progression. Since we can get our friend’s part to test or whether it works with the current part we are being assigned to do. Hence, it helps a lot during our work phases.

System Testing

System testing is a level of the software testing where a complete and integrated software is tested. The importance of system testing is so that the developers can evaluate the system’s compliance with the specified requirements.

After conducting the Big Bang integration method, the team decided to finalize off by conducting a system testing to make sure that all our functions were properly usable. We check that our basic CRUD for each management were usable by creating dummy data to see if they were functional and getting recorded. After which we checked out the database to see if it has taken in the details and information that we have placed during the creation of the dummy data.

User Acceptance Testing

User acceptance testing is the final phase of the software testing process. During UAT, the actual software users test the software to make sure it can handle required tasks in real-world scenarios, accordingly to the required specifications.

After completing the system testing, the team will conduct a user acceptance testing with Mr. Wang to ensure that the software is developed fully to the required specification proposed by him. However, testing in this phase not only involves Mr. Wang but as well as the developers on our team. Feedbacks received during this phase will used to address certain bugs they may occur in the system as well as to improve the system in future.

# TEST LOG

## 4.1 <Scenario / Function 1 by Wei Jian>

**Title:** Staff Management Page - Successfully create new staff and assign a duty type

**Description:** The user should be able successfully create a new staff and assign a duty type

**Precondition:** The user should be registered as a management level staff with a valid login credentials

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Click on "Staff Management" | - | Go to Staffs Page | Pass | - |
| 2 | Click on "Create New" |  | Go to Create New Page | Pass | - |
| 3 | Enter Staff's Details | First\_Name: John  Last\_Name:Pearseed  Date\_Of\_Birth: 05/11/1997  Bank\_Acc\_No: 928 3482  Address: 35 jalan tua kong road 520102  Contact\_Number: 9283 2812  Duty\_Type:Security Maintenance | Staff created successfully | Pass | New staff has been added to the database table |

**Title:** Register Page - Successfully register for an user account

**Description:** The user should be able to successfully register for a new account

**Precondition:** The "register for an account" view should be mask from the system only the system administrators is able to register a new account for users and assign the type of access the user has

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Click on "Register" | - | Goes to register page | Pass | Only System admin can access this page |
| 2 | Enter Required fields | First\_Name: Jack  Last\_Name:Apple  Email:  [JackApple@gmail.com.sg](mailto:JackApple@gmail.com.sg)  Username: JackApple  Password: Helloworld123@  Confirm Password:  Helloworld123@ | New user created | Pass | The password field will be hash in real life |
| 3 | Goes to Login Page | - | Automatically redirect to Login page | Pass | - |

**Title: Login** Page - User authenticated successfully by the system

**Description:** The user should be able to successfully login

**Precondition:** The user already has the required login credentials from the system administrator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Click on "Login" page | - | Goes to login page | Pass | - |
| 2 | Enter Login credentials | Email: [JackApple@gmail.com](mailto:JackApple@gmail.com)  Password: Helloworld123@ | User is able to login | Pass | - |
| 3 | Go to the home page of the system | - | Automatically redirects users to home page | Pass | - |

## 4.2 <Scenario / Function 2 by Tuitus >

Add

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Click Staff Management |  | Go to Staff page | Pass | - |
| 2 | Click Create New |  | Go to create staff page | Pass | - |
| 3 | Enter staff details in the boxes. | First\_name: Tuitus  Last\_name: Sim  Date\_Of\_Birth:  8/6/2012  Bank\_Acc\_No  1234 4567 8912 3456  Address: Punggol Rd 2933  Contact\_Number:  659683712  DutyType: General Maintence | New staff created | Pass | The new staff created will now be added into the database. |
| 4 | Click Back |  | Go back to staff page | Pass |  |

View Staff details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Click staff management |  | Go to staff management page | Pass | - |
| 2 | Click details |  | Go to staff details page | Pass | - |
| 3 | Click back to list |  | Go to staff management page | Pass | - |

Update Staff details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Click staff management |  | Go to staff management page | Pass | - |
| 2 | Click edit |  | Display staff details of selected staff. | Pass | - |
| 3 | Edit the selected details | Contact\_number changed from 98745449 to 92458786 |  | Pass |  |
| 4 | Click save | Contact\_number changed from 98745449 to 92458786 | Record is saved | Pass | Contact number changed |
| 6 | Click back to list |  | Go back to staff management | Pass |  |

Delete staff record

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Click staff management |  | Go to staff management page | Pass | - |
| 2 | Click on delete on selected staff records |  | Go to staff deletion page. | Pass | - |
| 3. | Click on delete |  | Record is deleted. | Pass | - |

## 4.3 <Scenario / Function 3 by Yao Hong>

Add

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Click list of rooms |  | Direct user to list of rooms available | Pass |  |
| 2 | Click on Create New rooms | Size:4  IsCleaning: No  IsSmoking: No  Numbed: 2  CheckInDate:  CheckOutDate: | Pass  Room data will be created below | Pass |  |
|  | Return to list |  | Direct back to list of rooms | Pass | Overall PASS |
|  |  |  |  |  |  |

View

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Click list of rooms |  | Direct to list of rooms page and their statuses | Pass |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

Update

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Click update at list of rooms | Change isAvailable to Yes | Room isAvailable status become Yes | Pass |  |
| 2 | Return to room views |  | Direct back to room view list and the status of isAvailable is Yes | Pass | Must update for it to pass |
| 3 | Update other details of the room status |  |  |  |  |
| 4 | Click other status to update | Change them accordingly | Pass | Pass | Select the category of the room status to update and save it so it will be updated |

Delete

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Delete room button |  | Room is deleted | Pass | Delete specific room when room is no longer available |
| 2 |  |  |  |  |  |
|  |  |  |  |  |  |

## 4.4 <Scenario / Function 4 by Zhan An>

Add New Booking Record

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Select Booking |  | Go to booking module page | Pass | - |
| 2 | Select “Create” button in the Booking Module Page |  | Go to add new booking page | Pass | - |
| 3 | Enter details required for all the fields in booking | Enter a booking start date time of “05/06/17 12:00:00 PM”, end date time of “08/06/17 12:00:00 PM”, additional request of “a single king size bed”, empty checkout extension, select guest under the first name of “Melvin” and RoomId of “1001” | New booking record added to the system | Pass | New booking records will be added to the database. Checkout Extension and Additional Request are non-mandatory fields. Checkout Extension is only used when last minute changes are made by guest who had already made payment in advance. |
| 4 | Return to booking list |  | Go to the booking list page | Pass | - |

View Existing Booking Records

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Select Booking |  | Go to booking module page | Pass | - |
| 2 | Search for a particular booking record and select “details” | Search for booking record under the first name of guest “Melvin” | Go to details page under the first name of guest Melvin | Pass | - |
| 3 | Return to booking list |  | Go to the booking list page | Pass | - |

Update Booking Records

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Select Booking |  | Go to booking module page | Pass | - |
| 2 | Search and find for a booking record at a date & time under the name of a guest | Search booking record of guest’s first name “Melvin” | Details of booking record under the guest with the first name of “Melvin” displayed | Pass | - |
| 3 | Click “edit” record and select “yes” when system prompts for confirmation |  | Go to edit page of booking | Pass | - |
| 4 | Search for the appropriate field for data to be edited | Search for the end date time of “05/06/17 12:00:00 PM” under the first name of guest called “Melvin” | Details of end date time under the guest with the first name of “Melvin” displayed | Pass | - |
| 5 | Edit the end date time | Change the end date time of “05/06/17 12:00:00 PM” to “05/06/17 2:00:00 PM” | End date time is updated to “05/06/17 2:00:00 PM” | Pass | - |
| 4 | Return to booking list |  | Go to the booking list page | Pass |  |

Delete Booking Records

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Select Booking |  | Go to booking module page | Pass | - |
| 2 | Search and find for a booking record at a date & time under the name of a guest | Search booking record of guest’s first name “Melvin” | Details of booking record under the guest with the first name of “Melvin” displayed | Pass | - |
| 3 | Click “delete” record and select “yes” when system prompts for confirmation | Deletes booking record under the guest’s first name “Melvin” | Data records of booking under the first name of guest called “Melvin” is successfully deleted. | Pass | - |
| 4 | Return to booking list |  | Go to the booking list page | Pass | - |

Add New Guest Record

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Select Guest Management |  | Go to Guest page | Pass | - |
| 2 | Select “Create” button in Guest Management |  | Go to add new Guest page | Pass | - |
| 3 | Enter guest details required for all fields | Guest with the first name of “Melvin”, last name of “Koh”, contact of “81234321”, email of [melkoh@yahoo.com](mailto:melkoh@yahoo.com)”, address of “Punggol Rd Singapore 653123”, date of birth of “05/05/89” | New guest created | Pass | The new guest created will now be added into the database. |
| 4 | Return to guest management list |  | Go to the guest management list page | Pass | - |

View Guest details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Select guest management |  | Go to Guest Management page | Pass | - |
| 2 | Search for a particular guest record and select “details” | Search for guest record under the first name of guest “Melvin” | Go to details page under the first name of guest Melvin | Pass | - |
| 3 | Return to guest management list |  | Go to the guest management list page | Pass | - |

Update Guest details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Select guest management |  | Go to guest management page | Pass | - |
| 2 | Search and find for a guest record under the name of a guest | Search booking record of guest’s first name “Melvin” | Details of guest record under the guest with the first name of “Melvin” displayed | Pass | - |
| 3 | Click “edit” record and select “yes” when system prompts for confirmation |  | Go to edit page of guest management | Pass | - |
| 4 | Search for the appropriate field for data to be edited | Search for the email “[melkoh@yahoo.com](mailto:melkoh@yahoo.com)” under the first name of guest called “Melvin” | Details of end date time under the guest with the first name of “Melvin” displayed | Pass | - |
| 6 | Edit the end date time | Change the end date time of email of “[melkoh@yahoo.com](mailto:melkoh@yahoo.com)” to “[melkoh@gmail.com](mailto:melkoh@gmail.com)” | Email is updated to “[melkoh@gmail.com](mailto:melkoh@gmail.com)” under the guest with the first name of “Melvin” | Pass | - |
| 7 | Return to guest management list |  | Go to the guest management list page | Pass | - |

Delete Guest record

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1 | Click guest management |  | Go to guest management page | Pass | - |
| 2 | Search and find for a guest record under the name of a guest | Search guest record of guest’s first name “Melvin” | Details of booking record under the guest with the first name of “Melvin” displayed | Pass | - |
| 3. | Click “delete” record and select “yes” when system prompts for confirmation | Deletes guest record under the guest’s first name “Melvin” | Data records of guest under the first name of guest called “Melvin” is successfully deleted. | Pass | - |
| 4 | Return to guest management list |  | Go to the guest management list page | Pass | - |

Possible Scenarios in Booking Module (Zhan An)

|  |  |  |
| --- | --- | --- |
| Scenario Name | Starting Flow | Alternate |
| Scenario 1:  Successful Booking | Basic Flow |  |
| Scenario 2:  Guest unverified and not displayed | Basic Flow | A1 |
| Scenario 3:  User exits the booking module | Basic Flow | A2 |
| Scenario 4:  Booking failed due to room occupied | Basic Flow | A3 |
| Scenario 5:  Booking failed due to incorrect data format | Basic Flow | A4 |

Possible Scenarios in Staff Module (Tuitus)

|  |  |  |
| --- | --- | --- |
| Scenario Name | Starting Flow | Alternate |
| Scenario 1:  Successful creation of staff | Basic Flow |  |
| Scenario 2:  Staff details enter with the wrong data format | Basic Flow | A1 |
| Scenario 3:  User exits the staff management module | Basic Flow | A2 |
| Scenario 4:  Missing Mandatory fields | Basic Flow | A3 |
| Scenario 5: | Basic Flow | A4 |

Possible Scenarios in Guest Module (Zhan An)

|  |  |  |
| --- | --- | --- |
| Scenario Name | Starting Flow | Alternate |
| Scenario 1:  Successful creation of guest | Basic Flow |  |
| Scenario 2:  Creation of guest failed due to missing mandatory fields | Basic Flow | A1 |
| Scenario 3:  User exits the guest management module | Basic Flow | A2 |
| Scenario 4:  Creation of guest failed due to room occupied | Basic Flow | A3 |
| Scenario 5:  Creation of new guest failed due to incorrect data format | Basic Flow | A4 |

Possible Scenarios in Register Account Module (Wei Jian)

|  |  |  |
| --- | --- | --- |
| Scenario Name | Starting Flow | Alternate |
| Scenario 1:  Successful registering of account | Basic Flow |  |
| Scenario 2:  Creation of account failed due to missing mandatory fields | Basic Flow | A1 |
| Scenario 3:  User exits the register account module | Basic Flow | A2 |
| Scenario 4:  Creation of account failed as the email address entered is already used by someone else | Basic Flow | A3 |
| Scenario 5:  Creation of account failed due to incorrect format entered for password | Basic Flow | A4 |
| Scenario 6:  Creation of account failed as "Confirm Password" field does not match the "Password" field | Basic Flow | A5 |

Possible Scenarios in Login Module (Wei Jian)

|  |  |  |
| --- | --- | --- |
| Scenario Name | Starting Flow | Alternate |
| Scenario 1:  Successful login of account | Basic Flow |  |
| Scenario 2:  Login failed due to incorrect Username or Password entered | Basic Flow | A1 |
| Scenario 3:  User exits the login module | Basic Flow | A2 |
| Scenario 4:  Login failed due to incorrect data type being entered in the username field | Basic Flow | A3 |

# 

# Configuration Management

# References

**GitHub Repository:**

<https://github.com/WeiJian123/SWEN-Wei-Jian-Tuitus-Zhan-An-Yao-Hong->