Temasek Polytechnic

School of Informatics and IT

**Diploma in Information Technology (IT)**

Software Test Specifications (STS)

**Project Particulars**

|  |  |
| --- | --- |
| **Tutor** | Mr Qi Yutao |
| **Class** | P04 |
| **Project Title** | Hotel Management System |

**Project Team’s Particulars**

|  |  |
| --- | --- |
| **Matric Number** | **Student Name** |
| **1704762B** | **Haziq Asyraaf** |
| **1705035B** | **Pradhun Raj** |
| **1701635C** | **Frederick Kang** |
|  |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 17/01/2019 | 0.1 | Copied template | Haziq |
| 20/01/2019 | 1.0 | Added 2.1, 4.1 and 4.2 | Frederick |
| 23/01/2019 | 1.1 | Added 2.2, 4.3, 4.4 and 4.5 | Haziq |
| 07/02/2019 | 1.2 | Added 2.3,4.6 and 4.7 | Raj |
| 08/02/2019 | 1.3 | Final editing and corrections | Raj |
| 08/02/2019 | 1.3 | Added Data integration | Haziq |

**Table of Contents**

[**DISTRIBUTION OF WORKLOAD**](#_vgwdkl51bzti) **4**

[**MODULE DEVELOPMENT AND UNIT TESTING**](#_2n8lyii7ht1n) **4**

[2.1 Room Availability and Booking](#_2et92p0) 4

[2.2 Report module](#_e41nnub2svo6) 4

[2.3 Signup and Login Module](#_75w1gafr65qy) 5

[**3 SYSTEM INTEGRATION**](#_r283kmmvrsuv) **6**

[**4 TEST LOG**](#_dklrzty49rrj) **7**

[Room Availability and Booking Module](#_l3xi4ovejslm) 7

[4.1 Room availability and booking Testing To check if can see rooms available](#_f7iv46tv9rxs) 7

[4.2 Room availability and booking testing to see if can open up a form to fill information to book a room](#_6cpx0ojvdymi) 7

[Reporting Module](#_trw3sa8n2bwe) 8

[4.3 Room status](#_yxypskkmsgib) 8

[4.4 All Room listing](#_70aj4m8e1b6r) 8

[4.5 All Guest listing](#_gn9rk8f101k4) 9

[Signup and Login Module](#_vjg9bs2m5s22) 9

# DISTRIBUTION OF WORKLOAD

|  |  |
| --- | --- |
| **Construction & Testing** | **Members** |
| 2.1 Room Availability and Booking Module  4.1 Room Availability and Booking Module | Frederick |
| 2.2 Reporting Module  3 System Integration  4.2 Reporting Module | Haziq |
| 2.3 Signup and Login Module  4.3 Signup and Login Module | Raj |
|  |  |

# 

# MODULE DEVELOPMENT AND UNIT TESTING

## 2.1 Room Availability and Booking

Ensured quality software development and testing by running and building the web application and testing out if the room availability booking module is working. We would try to see if the module is able to do it’s assigned functions. For the First Function is checking Room availability, we test by clicking on the button that direct us to the room availability page where we will be able to see the rooms available. The second function which is the booking we test out by clicking a button that opens up a form that allows us to fill in whatever information needed in order to book, after filling out the required information, we click on the button submit, and there we booked the room and the function has been tested out.

## 2.2 Report module

There are total of 6 testing methodologies that I researched and applied into report module development and unit testing.

I ensure that every line of code executes properly with unit testing. I use function coverage instead of statement and path coverage because function coverage means that each function of codes to be at least in one test case. In this one use case, I use it to check every function such as clicking on the button that link to the next page to view my list of customer that is currently staying in the hotel. Thus, there's no need to check every code instead deliver the function.

Afterwards I ensure that every function produces its expected outcome with functional testing this also commonly referred to black box testing, which requires no knowledge of the underlying implementation. This testing is created from requirement use cases that belonging to hotel owner of the functions that he required. In this case, he wants to views the list of customer that is currently staying in the hotel by having a button to view and redirect to the page of listing.

After blackbox testing, I then ensure that all of my delivered functions combined delivers the desired business result with system testing. All of the use cases of all I made combined are to meet the end-to-end results such as executing the system testing are to deliver the desired business result. In this case, having 3 use cases to deliver requirements of 3 reports which are room status, all room listing and all guest listing.

I use regression testing to ensure that new changes did not adversely affect other parts of the system. Any code modifications in Regression Testing would not inadvertently introduced bugs into the system. Thus goals in this, is to frequently ensure a baseline software quality is maintained. You can take a look at report module function codes does not affect other modules, and only running on my page.

I ensure the system angular of my project integrates with and does not adversely affect other enterprise systems with System Integration Testing. This means it can easily test interaction between systems that will not collaborate once the developed system is installed. This can be done by looking at the html pages, and typescript codes are involved in navigating pages back and forth. This can be easily install in other systems.

Lastly, I ensure that the customer is satisfied with the system with Acceptance Testing. In order to run this, i make example of my lecturer to test the system that i designed and give feedback.

## 2.3 Signup and Login Module

Tested login and signup functions of the app using silk test. Ensured that the data is validated for login and the app differentiates user roles which are admin and user. If an admin ID or user ID and its password is used for login, the web app is navigated to user home page. But, when admins login to the app, the admin tab is kept open for access. If a user logs in, and clicks on admin tab, the will be directed to login page again by ending the user’s login session. Using various test data, I tested my part of the web app to ensure the robustness of login function.

The Sign Up button in the login page will bring the user to a signup form. Here the user or admin can create their account by filling up the form. Once the user fills up the form including the user role, the data is sent to the database and this data is used for login validation. This function is also tested using silk and by posting various test data.

# 3 SYSTEM INTEGRATION

In order to efficiently integrate our system, we use GitHub as it allows us to easily improve and automate our workflow. We can easily sell or share our apps in GitHub Marketplace. It easily shared our codes and request to solve our problems.

In integrating the system with github, we can first look at the version control of github. It is a system that can records the changes into a file or set of files over time so that we can recall the specific versions later. In this example, we commit or push changes we can take a look at their descriptions of the file.

It is important to have github as a IT related jobs such as graphic or web designer that we want to keep every version of an layout or image. Thus Version Control System(VCS) allows us to revert files back to a previous state,revert the entire project back to a previous start,look at the preview changes made over time, look who had last modified the changes that may or not cause the problem, who introduced an issue and when, and many more. Using the VCS means that if we actually lose files or perhaps cause an error, we can easily recover it easily.

During the integration phase, we ensure that each of us check the file status and log before updating the master or branch feature. In this case we each have 3 branch, after the group members have make commit changes to the their respective branch. I then checkout into the master branch.pull and merge the recent changes and lastly merge all three feature branch into master branch. Afterwards to end our project, we delete the three mini feature branch as all of our compiled codes had combined into one in master branch. Now what if we had made a mistake during integrating all the system together such as not checking the logs and checking the status. I would do a full git reset hard commit command to revert back to the prior commit.

In general in github, we first made git clone repository before starting to work on the system. Where we first checkout the branch we want to work on, and to ensure that we are on local version of master is up to date we pull the base from master. Afterwards to remote branch we push back to the origin branch. Now they just have to submit a pull request for the remote branch, and the administrator accept it. In order to solve conflicts due to making changes to their local repositories, their code may conflict with that of other contributors once those changes are actually pushed to a central repository. We must always pull all new changes from central repository. In other conflicts, we can refer back to the master branch project history.

We felt that it is important to know that Git is the currently most known and popular for their version control system. It should be noted that it is not the only version control system. However, it is important to at least understand version control and some version control system.

# 

# 4 TEST LOG

## Room Availability and Booking Module

## 4.1 Room availability and booking Testing To check if can see rooms available

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1. | click on room availability and booking tab |  | directed to room availability and booking tab | directed to room availability and booking tab |  |
| 2. | click on check room available button |  | directed to page that shows the rooms that are available | directed to page that shows the rooms that are available |  |
|  |  |  |  |  |  |

## 

## 4.2 Room availability and booking testing to see if can open up a form to fill information to book a room

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1. | click on room availability and booking tab |  | directed to room availability and booking tab | directed to room availability and booking tab |  |
| 2. | click on book room button |  | open up form that shows information fields to fill in information to book a room | open up form that shows information fields to fill in information to book a room |  |
|  |  |  |  |  |  |

## 

## Reporting Module

## 4.3 Room status

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1. | click on report module tab button |  | directed to report module tab |  |  |
| 2. | click on button that indicates Room status |  | directed to page that shows the list the status of all the rooms that shows whether its available,occupy or scheduled for cleaning |  |  |

## 

## 4.4 All Room listing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1. | click on report module tab button |  | directed to report module tab |  |  |
| 2. | click on the button that indicates All Room listing |  | directed to page that shows the list of all the rooms, these allows the report to show that name or number of occupants which also show whether its adult and children |  |  |

## 

## 4.5 All Guest listing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1. | click on report module tab button |  | directed to report module tab |  |  |
| 2. | click on the button that indicates All Guest listing |  | directed to page that shows the list of all the guest of all the rooms |  |  |

## 

## Signup and Login Module

**4.6 Sign Up**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1. | Enter the required data to sign up | Name: Test  Password: Test  Confirm password: Test  Age: 20  Address: test  Email: [test.@mail.com](about:blank)  User/Admin: Admin |  |  |  |
| 2. | Click on Sign up button |  | Navigates to login page | Navigates to login page |  |

## 

**4.7 Log In**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/No** | **Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Remarks** |
| 1. | Enter the required data to log in | Name: Test  Password: Test |  |  |  |
| 2. | Click on Login button |  | Navigates to users home page | Navigates to users home page |  |

## 