**TEST PLAN**

**G6T6**

# Introduction

This test plan has been created to communicate the test approach and procedures to the team members to standardise testing done by different team members. This document will identify the test deliverables and what is in scope and out of scope.

# Purpose

This document covers the testing that the team will be conducting for the Merlion University Bidding Online System (BIOS) project. It defines the testing requirements for the various functionalities that the client has requested for and the proper procedures for testing them. The following document entails:

* Functionalities to be tested
* Testing procedures
* Pass/Fail criteria
* Resources needed for testing
* Risk and mitigation plans

# Testing Summary

## Testing Scope

The scope of the testing will be defined into the three following parts:

* Manual Testing, this will entail the functional requirements specified in <https://wiki.smu.edu.sg/is212/Project#Functional_Requirements>
* Automated Testing, this will entail the web service requirements specified in <https://wiki.smu.edu.sg/is212/Project#Web_Service_Requirements>
* Usability Testing, this will be done in tandem with the manual testing, this will entail the usability (number of clicks, UI design) of the website.

## Test Plan

The testing will be separated into 5 parts, being conducted at the end of each iteration of the project timeline. The testing for each part will be conducted in the following manner:

Unit Test 🡪 Integration Test 🡪 Regression Test

|  |  |
| --- | --- |
| Unit Test | Testing the new functionalities implemented in the iteration as individual units. |
| Integration Test | Testing the new functionalities together as a whole. |
| Regression Test | Testing previously implemented functionalities to ensure that new changes do not adversely affect them. |

## Creation & maintenance of Testcases

The testcases created will be done specifically to the functionalities implemented in the iteration which will determine by what needs to be checked. These testcases will be added to the testcase.xlsx file found in project-g6t6/testing folder. Maintenance of testcases after the iteration will be done by the iterations PM according to changes in functionality requirements or testcase. Additional test results be appended to the new columns and will not replace test results from previous testing sessions.

## Testcase file format

### Manual Testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S/N | Functionality | Description | Test Input | Test Procedure | Expected Results | Actual Results | Pass/Fail |

**S/N** – Serial Number of the testcase

**Functionality** – Functionality tested for the testcase

**Description** – Description of the testcase

**Test Input** – Input of the testcase

**Test Procedure** – Testing procedure, when to input and actions to take

**Expected Results** – Results expected as of client functionality requirements

**Actual Results** – Results of the testing

**Pass/Fail** – Does the actual results match the expected Results

### Automated Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S/N | Description | Test Inputs | Expected Results | Actual Results | Pass/Fail |

**S/N** – Serial Number of the testcase

**Description** – Description of the testcase with functionality tested clearly indicated

**Test Input** – file name of the testcase

**Expected Results** – Results expected as of client functionality requirements

**Actual Results** – Results of the testing

**Pass/Fail** – Does the actual results match the expected Results

**For both manual and automated testing, any additional testing sessions should be documented the new columns: actual results and pass/fail of the new testing sessions to be documented.**

# Bugs

## Bug Metrics

Any bugs found during the testing will be documented in the bm.xlsx file found in project-g6t6/metrics. The bugs should be updated with the severity of low, high and critical, with the score of 1, 5 and 10 respectively. When the score or unresolved bugs hits 10, the PM should call for the development of new functionalities to be put on hold and focus the team’s attention to fixing the bugs.

# Testing Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Iter | Functionalities | Plan date | Pass/Fail criteria |
| 1 | Login (student) | 9/25/2019 | Password and User ID validation |
| 2 | Bid for section (UI) for round 1 | 10/1/2019 | Validate in UI:  1. insufficient edollar for bid amount;  2. class timetable and exam timetable clash;  3. incomplete prerequisites;  4. course completed;  5. section limit reached;  6. not own school;  7. no vacancy; |
| 2 | Drop a Bid (UI) | 10/5/2019 | Common validation; logical validations ; |
| 2 | Login (Admin) | 9/25/2019 | Password and Admin ID validation |
| 2 | Starting and clearing logic for round 1 | 10/5/2019 | logic validation, sorting , UI |
| 2 | View Bidding results (UI) for round 1 | 10/5/2019 | Info updated can correspond with update a bid |
| 2 | Bootstrap with validation (UI) | 9/29/2019 | Common validation; number of rows are loaded and correct error messages |
| 3 | Bid for section with real-time information for round 2 (UI) | 10/14/2019 | Validate in UI:  1. insufficient edollar for bid amount;  2. class timetable and exam timetable clash;  3. incomplete prerequisites;  4. course completed;  5. section limit reached;  6. not own school;  7. no vacancy;  8. minimum bid amount  9. real-time status for the bid |
| 3 | Starting and clearing logic for round 2 | 10/16/2019 | logic validation, sorting , UI |
| 3 | Authenticate web service | 10/10/2019 | token; validation |
| 4 | Bootstrap web services | 10/17/2019 | load data correctly or give correct error message and in correct order; validate token |
| 4 | Start round web service | 10/18/2019 | token validation; start round successfully |
| 4 | Stop round web service | 10/18/2019 | token validation; stop a round successfully |
| 4 | Update Bid web service | 10/21/2019 | token validation, validate logic; bid update successfully |
| 4 | Delete Bid web service | 10/25/2019 | token validation, logic validation; bid delete successfully |
| 4 | Drop section web service | 10/25/2019 | token validation; logic validation; section delete successfully |
| 4 | Dump (user) web service | 10/29/2019 | token validation, logic validation, userid validation |
| 4 | Dump(bid) web service | 10/20/2019 | course info check; token validation; logic validation |
| 4 | Dump(section) web service | 10/29/2019 | section info check; token validation; logic validation |
| 4 | Dump table web service | 10/31/2019 | token validation; dump correct information for each table |
| 5 | Bid status web service | 11/15/2019 | token validation; bid status info check |
| 5 | Extra functionalities (course filter / class timetable / exam timetable)(UI) | 11/15/2019 | class time display in correct time slot; exam time display; |