­­­G51FSE Assessed Lab 4

|  |  |  |
| --- | --- | --- |
| **The Assemblers with Flip-Flops** |  | 09/03/2018 |

# 3. Test Plans

This section of the software specification document highlights the details of the system through the use of Unified Modelling Language and prototypes.

We have added some additional columns to the bug table. These are notes and a description for the changes are given in this paragrah.We have decided to have additional columns in the test table so we can provide more detail about the tests carried out. This will increase the traceability of the tests and make the changes easier to see. Firstly, we added **a Test ID** so that we can have a unique id for each test carried out. This will make it easier to refer back to the tests later on in the document. In addition to this, we added a column for prerequisite which will store detail about what other functions need to be performed before this test is carried out. Next, we included a column for the **actual output**. This is necessary so that we can compare if the expected output matches the actual output. If it doesn’t, a reason must be provided as to why and most likely there is an error/flow in the program. In addition to this, we added a column called ‘**test created by’**. This will hold the name of the person who created the tests and carried it out. This will increase accountability. Next, we added a column called ‘**pass or fail’**. This will make it easier to see the number of tests that has passed without the person having to read the details of each individual test. We then created a column which will store the **date** that the test has been carried out on. This will make it easier to see which version of a test is the latest if a test had been carried out multiple times. Finally, we added a column for **notes** which can be used to store additional details about the tests that does not fall into any other category. For example, if a test has failed, the notes could include a description as to why the test has failed.

Project Class

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID | Function Name | Test Aim | Pre-Requisites | Inputs | Expected Output(s) | Actual Output | Test Created By | Source: Spec or code inspection | Pass / Fail | Date (Completed by) | Notes |
| 213 | Get Phase ID | Checks if the correct phase ID is returned when a project is 1st created | New Project Object | N/A | 1 (Int) | 1 (Int) | Aidan Reed | Specification Document – Class Document | PASS | 25.04.2018 |  |
| 214 | Get Phase ID | Checks if the correct phase ID is returned when project is moved from the 1st stage to the next | Object from Test 213 | N/A | 2 (Int) | 2 (Int) | Aidan Reed | Specification Document – Class Document | PASS | 25.04.2018 |  |
| 215 | Get Phase ID | Checks if the correct phase ID is returned when project is moved from the 2nd stage to the next | Object from Test 214 | N/A | 3 (Int) | 3 (Int) | Aidan Reed | Specification Document – Class Document | PASS | 25.04.2018 |  |
| 216 | Get Phase ID | Checks if the correct phase ID is returned when project is moved from the 3rd stage to the next | Object from Test 215 | N/A | 4 (Int) | 4 (Int) | Aidan Reed | Specification Document – Class Document | PASS | 25.04.2018 |  |
| 217 | Get Phase ID | Checks if the correct phase ID is returned when project is moved from the 4th stage to the next | Object from Test 216 | N/A | 5 (Int) | 5 (Int) | Aidan Reed | Specification Document – Class Document | PASS | 25.04.2018 |  |
| 218 | Get Phase ID | Checks if the correct phase ID is returned when project is moved from the 5th stage to the next | Object from Test 217 | N/A | 6 (Int) | 6 (Int) | Aidan Reed | Specification Document – Class Document | PASS | 25.04.2018 |  |
| 219 | Get Phase ID | Checks if the correct phase ID is returned when project attempted to be moved passed 6th stage | Object from Test 218 | N/A | 6 (Int) | 7 (Int) | Aidan Reed | Specification Document – Class Document | FAIL | 25.04.2018 |  |
| 220 | Get Phase ID | Checks if the correct phase ID is returned when project is moved from the last stage to the previous | Object from Test 218 | N/A | 5 (Int) | 7 (Int) | Aidan Reed | Specification Document – Class Document | FAIL | 25.04.2018 | Currently no functionality to go to previous stage. |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 214 | Get Project Contacts | The test will see if the string array of contacts for the given project is returned | Project ID |  |  |  | Christian Stubbs | Specification Document – Class Document |  |  |  |
| 215 | toString() Override | Checks if the function creates the string of the template "[Project title] [[Project Phase]] " |  |  |  |  | Justin Ng | Specification Document – Class Document |  |  |  |

Change Log

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Change ID | GIT Commit ID | Bug ID | Description of Change | Files Changed | Date Received | Date Changed | Initiator By (Who Changed) | Change Checked By | Notes |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Bug Fix List

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Problem ID | Problem Description | Line of code | Test ID (if created) | Change ID | Proposed Fix | Priority  High, Med, Low | Date Problem identified | Fixed? (Yes/No) | Date Problem fixed | Who identified/fixed the test? | Does the bugged code relate to other functions? | Notes |
| 1 | Project Phase ID goes 1 over the number of different phases. |  | 219 |  |  | Med | 25.04.2018 |  |  | Aidan Reed |  |  |
| 2 | No method to go to previous phase inside the CompanyProject Class |  | 220 |  |  | Low | 25.04.2018 |  |  | Aidan Reed |  | Low priority as description does not directly say such functionality should exist but was flagged during testing as test failed. |