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

The following tests use references to constants defined in the Junit test files at the start of the document. As such for easier comparison in the place the constants are defined below and are used in the Input, Expected Output and Actual Output columns to reduce repeating data and if the test inputs change in the future the document can be updated at a single point.

**kPTITLE1 =** “12345678”  
**kPTITLE2 =** “123456789000”  
**kDEFAULTTITLE =** “New Project”  
**kCONTACT1 = “**[test@gmail.com](mailto:test@gmail.com)”  
**kCONTACT2 =** “raiu9s@gmail.com”   
**kCONTACT3 =** [q39ikdf@outlook.com](mailto:q39ikdf@outlook.com)  
**companyProjectFirst\_Empty** (CompanyProject Object)  
**companyProjectSecond\_Phases** (CompanyProject Object) – Empty used for testing the phases   
**companyProjectThird\_Complete**(CompanyEmail Object) – Title: **kPTITLE1** Contacts: **kCONTACT1, kCONTACT2, kCONTACT3**)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| 201 | Company Project Constructor | Checks if default constructor sets up correctly with array lists initialized | Uses **companyFirst** **\_Empty** object no values have been set | N/A | 1 (Int) - PID  **kDEFAULT TITLE** (String)  True (Boolean) -Empty Contacts  0 (Int) - PhaseID  0 (Int) – Number Emails | 1  **kDEFAULT TITLE**  True  1  1 | Justin Ng | Specification Document – Class Document | FAIL | 27.04.2018 |  |
| 202 | Company Project Constructor | Checks if main constructor sets up correctly with title parameter passed | Uses **companyFirst** **\_Empty** object no values have been set | N/A | 3 (Int) - PID  **kPTITLE1 TITLE** (String)  True (Boolean) -Empty Contacts  0 (Int) - PhaseID  0 (Int) – Number Emails | 3 (Int) - PID  **kPTITLE1 TITLE** (String)  True  0 (Int) - PhaseID  0 (Int) – Number Emails | Justin Ng | Class Document | FAIL | 30.04.2018 |  |
| 203 | getPID | Checks if the correct project ID is returned when a project is 1st created | Uses **companyFirst** **\_Empty** object no values have been set | N/A | 1 (Int) | 1 (Int) | Justin Ng | Class Document | PASS | 25.04.2018 |  |
| 204 | getPID | Checks if the project ID increments correctly with the addition of two new projects | Uses **companyThird** **\_Complete** object | N/A | 3 (Int) | 3 (Int) | Justin Ng | Class Document | PASS | 27.04.2018 |  |
| 205 | getPTitle | Checks if the correct project Title is returned when a project is 1st created | Uses **companyFirst** **\_Empty** object | N/A | **kPDEFAULTTITLE** (String) | **kDEFAULT TITLE** (String) | Justin Ng | Class Document | PASS | 25.04.2018 |  |
| 206 | setPTitle | Checks if it is possible to set a project title which is less than 10 characters long | Uses **companyFirst** **\_Empty** object set title method and **kPTITLE1** | **kPTITLE1**  (String) | **kPDEFAULTTITLE** (String) | **kPDEFAULT TITLE** (String) | Justin Ng | Class Document | PASS | 25.04.2018 |  |
| 207 | setPTitle | Checks if it is possible to set a project title which is more than 10 characters long | Uses **companyFirst** **\_Empty** object set title method and **kPTITLE2** | **kPTITLE2** (String) | **kPTITLE2** (String) | **kPTITLE2**  (String) | Justin Ng | Class Document | PASS | 25.04.2018 |  |
| 208 | isContact | Checks if specified contact exists in project | Uses **companyFirst** **\_Empty** object | **kCONTACT1** (String) | False (Boolean) | False (Boolean) | Justin Ng | Class Document | PASS | 25.04.2018 |  |
| 209 | isContact | Should add new contact then check if it can be read back | Uses **companyFirst** **\_Empty** object and add contact method | **kCONTACT1** (String) | True (Boolean) | True (Boolean) | Justin Ng | Specification Document – Class Document | PASS | 25.04.2018 |  |
| 210 | addContact | Should be able to add two new contacts and use isContact to check if they exist in the Array List | Uses **companyFirst** **\_Empty** object and add contact method | **kCONTACT2** (String)  **kCONTACT3** (String) | True (Boolean) True (Boolean) | True (Boolean) True (Boolean) | Justin Ng | Specification Document – Class Document | PASS | 25.04.2018 |  |
| 213 | Get Phase ID | Checks if the correct phase ID is returned when a project is 1st created - should be initialized to represent start of array | Uses **companyFirst** **\_Empty** object | N/A | 0 (Int) | 1 (Int) | Aidan Reed | Class Document | FAIL | 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 | Uses **companyFirst** **\_Empty** object | N/A | 1 (Int) | 2 (Int) | Aidan Reed | Class Document | FAIL | 25.04.2018 |  |
| 219 | Get Phase ID | Checks if the correct phase ID is returned when project attempted to be moved passed 6th stage | Uses **companySecond** **\_Phases** object | N/A | 5 (Int) | 7 (Int) | Aidan Reed | Specification Document – Class Document | FAIL | 25.04.2018 |  |
| 221 | Get Phase ID | Checks if the correct phase ID is returned when moving the project FROM the first until last – expecting start phase to be 0 | Uses **companySecond** **\_Phases** object | N/A | 0,1,2,3,4,5 (Int) | 1,2,3,4,5,6 (Int) | Aidan Reed | 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 | New Project Object | N/A | 4 (Int) | 7 (Int) | Aidan Reed | Class Document | FAIL | 25.04.2018 | Currently no functionality to go to previous stage. |
| 222 | Get Project Contacts | Checks if the get project contacts method returns an object of type array list | Uses **companyFirst** **\_Empty** object | N/A | (Array List) |  | Aidan Reed | Class Document | PASS | 25.04.2018 |  |
| 223 | Get Project Contacts | Checks the get project contacts method returns a string equal to the first element input using the add contact method | Uses **companyThird** **\_Complete** object which has 3 contacts kCONTACT 1/2/3 | **kCONTACT1** (String) | **kCONTACT1** (String) | **kCONTACT1** (String) | Aidan Reed | Class Document | PASS | 25.04.2018 | Note the constant for the input is named : kCONTACT1 located in top of unit test |
| 224 | Get Project Contacts | Checks the type of the first element added into the contacts array list is of type string | Uses **companyThird** **\_Complete** object which has 3 contacts kCONTACT 1/2/3 | **kCONTACT1** (String) | (String) | (String) | Aidan Reed | Class Document | PASS | 25.04.2018 |  |
| 225 | toString Overide | Checks the type of the return value is of type string | Uses **companyFirst** **\_Empty** object | N/A | (String) | (String) | Aidan Reed | Class Document | PASS | 25.04.2018 |  |
| 226 | toString Overide | Checks the toString is in the format Project Title [Phase Name] | Uses **companyThird** **\_Complete** object with title **kPTITLE1** with the phase in the initial stage | N/A | “kPTITLE1 [Feasibility]”  (String) | “kPTITLE1 [Design]” | Aidan Reed | Class Document | FAIL | 25.04.2018 | kPROTITLE is a constant defined in the unit test. |
| 227 | toString Overide | Checks the toString is in the format Project Title [Phase Name] for all stages of the project lifecycle | Uses **companyThird** **\_Complete** object with title **kPTITLE1** with the phase in the initial stage but will be incremented to the last stage | N/A | “**kPTITLE1** [Feasibility]”  “**kPTITLE1** [Design]”  “**kPTITLE1** [Implementation]”  “**kPTITLE1** [Testing]”  “**kPTITLE1** [Deployment]”  “**kPTITLE1** [Completed]”  All of type  (String) | “**kPTITLE1** [Design]”  “**kPTITLE1** [Implementation]”  “**kPTITLE1** [Testing]”  “**kPTITLE1** [Deployment]”  “**kPTITLE1** [Completed]”  Missing the first stage | Aidan Reed | Class Document | FAIL | 25.04.2018 |  |

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 Name set to Design when created instead of Feasibility | 24: Company Project.class | 226 |  | Set the Project phase to 0 | Med | 25.04.2018 |  |  | Aidan Reed |  |  |
| **2** | Project Phase ID goes over the number of different phases when going to next stage. |  |  |  |  | Med | 25.04.2018 |  |  | Aidan Reed |  |  |
| **3** | No method to go to previous phase inside the CompanyProject Class |  |  |  |  | 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. |
|  |  |  |  |  |  |  |  |  |  |  |  |  |