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**)  
**companyProjectForth\_Phases** (CompanyProject Object) – Used for phase testing adding emails and moving between stages

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID | Function Name | Test Aim | Source: Spec or code inspection | Test Created By | Pre-Requisites | Inputs | Expected Output(s) | Actual Output | Pass / Fail | Date (Completed) | Tested By | Notes |
| 201 | Company Project Constructor | Checks if default constructor sets up correctly with array lists initialized | Class Document | Justin Ng | 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 (Int) - PID  **kDEFAULT TITLE** (String)  True (Boolean) -Empty Contacts  1 (Int) - PhaseID  0 (Int) – Number Emails | FAIL | 27.04.2018 | Justin Ng | The test failed as the output for the phase id was initialized to 1 not the start of the array bug created– JGN |
| Same as above with Phase ID = 0 (INT) | PASS | 02.05.2018 | Aidan Reed | The test ow produces the correct output after the BUG201 and CHNGE201 were acted up on. AR – 02.05.2018 |
| 202 | Company Project Constructor | Checks if main constructor sets up correctly with title parameter passed | Class Document | Justin Ng | Uses **companyFirst** **\_Empty** object no values have been set | N/A | 3 (Int) - PID  **kPTITLE1 TITLE** (String)  False (Boolean) -Empty Contacts  0 (Int) - PhaseID  0 (Int) – Number Emails | 3 (Int) - PID  **kPTITLE1 TITLE** (String)  False (Boolean) -Empty Contacts  1 (Int) - PhaseID  0 (Int) – Number Emails | FAIL | 30.04.2018 | Justin NG | The test failed as the output for the phase id was initialized to 1 not the start of the array bug created– JGN |
| Same out put as above apart from Phase ID was 0 (Int | PASS | 02.05.2018 | Aidan Reed | The test ow produces the correct output after the BUG201 and CHNGE201 were acted up on. AR – 02.05.2018 |
| 203 | getPID | Checks if the correct project ID is returned when a project is 1st created | Class Document | Justin Ng | Uses **companyFirst** **\_Empty** object no values have been set | N/A | 1 (Int) | 1 (Int) | PASS | 25.04.2018 | Justin Ng |  |
| 204 | getPID | Checks if the project ID increments correctly with the addition of two new projects | Class Document | Justin Ng | Uses **companyThird** **\_Complete** object | N/A | 3 (Int) | 3 (Int) | PASS | 27.04.2018 | Justin Ng | Objects for these tests are created using the before class when the unit test is initiated |
| 205 | getPTitle | Checks if the correct project Title is returned when a project is 1st created | Class Document | Justin Ng | Uses **companyFirst** **\_Empty** object | N/A | **kPDEFAULTTITLE** (String) | **kDEFAULT TITLE** (String) | PASS | 25.04.2018 | Justin Ng |  |
| 206 | setPTitle | Checks if it is possible to set a project title which is less than 10 characters long | Class Document | Justin Ng | Uses **companyFirst** **\_Empty** object set title method and **kPTITLE1** | **kPTITLE1**  (String) | **kPDEFAULTTITLE** (String) | **kPDEFAULT TITLE** (String) | PASS | 25.04.2018 | Justin Ng |  |
| 207 | setPTitle | Checks if it is possible to set a project title which is more than 10 characters long | Class Document | Justin Ng | Uses **companyFirst** **\_Empty** object set title method and **kPTITLE2** | **kPTITLE2** (String) | **kPTITLE2** (String) | **kPTITLE2**  (String) | PASS | 25.04.2018 | Justin Ng |  |
| 208 | isContact | Checks if specified contact exists in project | Class Document | Justin Ng | Uses **companyFirst** **\_Empty** object | **kCONTACT1** (String) | False (Boolean) | False (Boolean) | PASS | 25.04.2018 | Justin Ng |  |
| 209 | isContact | Should add new contact then check if it can be read back | Class Document | Justin Ng | Uses **companyFirst** **\_Empty** object and add contact method | **kCONTACT1** (String) | True (Boolean) | True (Boolean) | PASS | 25.04.2018 | Justin Ng |  |
| 210 | addContact | Should be able to add two new contacts and use isContact to check if they exist in the Array List | Class Document | Justin Ng | Uses **companyFirst** **\_Empty** object and add contact method | **kCONTACT2** (String)  **kCONTACT3** (String) | True  (Boolean)  True  (Boolean) | True (Boolean) True (Boolean) | PASS | 25.04.2018 | Justin Ng |  |
| 211 | getEmailForPhase | Tests to see if the function returns the emails in the current phase of a new project with no emails. | Class Document | Christian Subbs | Uses **companyForth** **\_Phases** | N/A | 0 (Int) | 0 (Int) | PASS | 27.04.2018 | Christian Stubbs |  |
| 212 | getEmailForPhase | Tests to see if the function returns the emails in the current phase of a new project after adding one email | Class Document | Christian Stubbs | Uses **companyForth** **\_Phases** and adds email with **kSENDER, kRECPIENT, kSUBJECT, kBODY** | N/A | 1 (Int) | 1 (Int) | PASS | 27.04.2018 | Christian Stubbs |  |
| 213 | getEmailForPhase | Tests when moving to next stage email count is 0 when moving to next stage after adding 1000 emails at the previous stage | Class Document | Christian Stubbs | Uses **companyForth** **\_Phases** and adds 1000 email with **kSENDER, kRECPIENT, kSUBJECT, kBODY** | N/A | 0 (Int) | Error: null pointer | FAIL | 02.05.2018 | Christian Stubbs | This fails due to a bug in the program where the array list is not initialized at the next stage. See BUG204 |
|  |  |  |  |  |
| 214 | getEmailForPhase | Iterates through the phases creating emails for each phase checking the size is equal to the number created and cycling through all stages – each stage has different number of emails | Class Documents | Christian Stubbs | Uses **companyForth** **\_Phases** and adds 0 – first stage 1 – second – 2 third – 3 forth – 4 fifth and 5 in 6th STAGE | Each stage should have the number of emails outlined in pre requisite column | 0  1 2 3 4 5 | Error: Null pointer exception | FAIL | 02.05.2018 | Christian Stubbs | This fails due to a bug in the program where the array list is not initailised at the next stage. See BUG204 |
|  |  |  |  |  |
| 215 | getEmailsForPhase(**int** thePhase) | Test to see if this function returns a value of 0 when accessing a new project in the first phase | Class Documents | Christian Stubbs | Uses **companyForth** **\_Phases**  with no emails in first phase (0) | N/A | 0 (Int) | Error: Null pointer exception | FAIL | 02.05.2018 | Christian Stubbs | Failed initially because the constructor initializes the Emails array at value 1 not 0 – see BUG201 |
| 0 (Int) | PASS | 03.05.2018 | Aidan Reed | The test passes after the BUG201 was rectified. See Change ID for bug for futher details. |
| 216 | getEmailsForPhase(**int** thePhase) | Iterates through all phases adding emails at each stage and moving to the next phase in the outer loop moves to the next stage and checks the previous stage | Class Documents | Christian Stubbs | Uses **companyForth** **\_Phases**  adds 0 – first stage 1 – second – 2 third – 3 forth – 4 fifth and 5 in 6th STAGE | Each stage should have the number of emails outlined in pre requisite column | 0  1 2 3 4 5 | Error: Null pointer exception | FAIL | 02.05.2018 | Christian Stubbs | This fails due to a bug in the program where the array list is not initailised at the next stage. See BUG204 |
|  |  |  |  |  |
| 217 | getEmailsForPhase(**int** thePhase) | Adds email to first phase and then moves to next stage and checks the previous stage size is equal to 1 | Class Documents | Christian Stubbs | Uses **companyForth** **\_Phases**  and adds email with **kSENDER, kRECPIENT, kSUBJECT, kBODY** | N/A | 1 (Int) | Error: Null pointer exception | FAIL | 02.05.2018 | Christian Stubbs | Failed initially because the constructor initializes the Emails array at value 1 not 0 – see BUG201 |
| 1 (Int) | PASS | 03.05.2018 | Aidan Reed | The test passes after the BUG201 was rectified. See Change ID for bug for futher details. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 218 | nextPhase() | Tests to ensure the function moves a project from it’s initial stage to the next phase once | Class Documents | Christian Stubbs | Uses **companyForth** **\_Phases** | N/A | 1 (Int) | 2 (Int) | FAIL | 02.05.2018 | Christian Stubbs | Failed initially because the constructor initializes the ProjectPhase to 1 |
| 1 (Int) | PASS | 03.05.2018 | Aidan Reed | This test further identified issues that were found in constructor tests. Issue was fixed by fixing BUG201 |
| 219 | nextPhase() | Test moves a project from the initial stage until the last stage and ensures it’s phase id is 5 -phases 0 - 5 | Class Documents | Christian Stubbs | Uses **companyForth** **\_Phases** | N/A | 5 (Int) | 7 (Int) | FAIL | 02.05.2018 | Christian Stubbs | Failed because the PhaseID is initialized to 1 not 0 – Arrays go from 0 – 5 in this implementation and because the project phase is incremented before checking |
| 5 (Int) | PASS | 03.05.2018 | Aidan Reed | Test now produces the successful result as a result of the issue been fixed in the constructor in BUG201 and BUG202 the issue was fixed |
| 220 | nextPhase() | Test moves project to final stage and tries to go past the final stage | Class Documents | Christian Stubbs | Uses **companyForth** **\_Phases** | N/A | 5 (Int) | 7 (Int) | FAIL | 02.05.2018 | Christian Stubbs | The reason this failed is because the project phase is incremented before the conditional check |
| 5 (INT) | PASS | 03.05.2018 | Aidan Reed | After the implementation of changes in BUG202 the projects cannot move past the last phase. AR – 03.05.2018 |
| 221 | getPhaseName | Will see if the correct name of the first phase is returned | Code Inspection | Christian Stubbs | Uses **companyForth** **\_Phases**  object with no emails added in initial stage | N/A | Feasibility | Design | FAIL | 02.05.2018 | Christian Stubbs | The test fails as it appears to miss the first stage of names form the project phases array – CS 02.05.2018 |
| Fesability | PASS | 03.05.2018 | Aidan Reed | The initialization of project set the phase to 1 instead of 0 – as such the fixes in BUG 201 have rectified this issue – AR 03.05.2018 |
| 222 | getPhaseName | Will see if the correct name of phase is returned on the second phase after moving along 1 phase | Class Documents | Christian Stubbs | Uses **companyForth** **\_Phases**  object with no emails moved to next stage | N/A | Design | Implementation | FAIL | 02.05.2018 | Christian Stubbs | The test fails as it appears to miss the first stage of names form the project phases array – CS 02.05.2018 |
| Design | PASS | 03.05.2018 | Aidan Reed | The initialization of project set the phase to 1 instead of 0 – as such the fixes in BUG 201 have rectified this issue – AR 03.05.2018 |
| 223 | getPhaseName | Will see if the last name of phase is returned even though attempted moving past the last phase | Class Documents | Christian Stubbs | Uses **companyForth** **\_Phases**  object with no emails moved to next stage | N/A | Completed | Null | FAIL | 02.05.2018 | Christian Stubbs | The test fails as it appears to miss the first stage of names form the project phases array – CS 02.05.2018 |
| Completed | PASS | 03.05.2018 | Aidan Reed | The initialization of project set the phase to 1 instead of 0 and the next phase method allowed the phase id to move past the end of the project phases array. When BUG201 and BUG202 were fixed the system produced the correct result. AR – 03.05.2018 |
| 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 |  |  |  |  |  | 1 (Int) | FAIL | 25.04.2018 |  | This is because the constructor initialises it to 1 instead of 0 |
| 214 | Get Phase ID | Checks if the correct phase ID is returned when project is moved from the 1st stage to the next | Class Document | Aidan Reed | Uses **companyFirst** **\_Empty** object | N/A | 1 (Int) | 2 (Int) | 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 | Specification Document – Class Document | Aidan Reed | Uses **companySecond** **\_Phases** object | N/A | 5 (Int) | 7 (Int) | 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 | Class Document | Aidan Reed | Uses **companySecond** **\_Phases** object | N/A | 0,1,2,3,4,5 (Int) | 1,2,3,4,5,6 (Int) | 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 | Class Document | Aidan Reed | New Project Object | N/A | 4 (Int) | 7 (Int) | 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 | Class Document | Aidan Reed | Uses **companyFirst** **\_Empty** object | N/A | (Array List) |  | 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 | Class Document | Aidan Reed | Uses **companyThird** **\_Complete** object which has 3 contacts kCONTACT 1/2/3 | **kCONTACT1** (String) | **kCONTACT1** (String) | **kCONTACT1** (String) | 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 | Class Document | Aidan Reed | Uses **companyThird** **\_Complete** object which has 3 contacts kCONTACT 1/2/3 | **kCONTACT1** (String) | (String) | (String) | PASS | 25.04.2018 |  |  |
| 225 | toString Overide | Checks the type of the return value is of type string | Class Document | Aidan Reed | Uses **companyFirst** **\_Empty** object | N/A | (String) | (String) | PASS | 25.04.2018 |  |  |
| 226 | toString Overide | Checks the toString is in the format Project Title [Phase Name] | Class Document | Aidan Reed | Uses **companyThird** **\_Complete** object with title **kPTITLE1** with the phase in the initial stage | N/A | “kPTITLE1 [Feasibility]”  (String) | “kPTITLE1 [Design]” | 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 | Class Document | Aidan Reed | 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 | FAIL | 25.04.2018 |  |  |

Change Log

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Change ID | Bug ID | GIT Commit ID | Description of Change | Files Changed | Date Received | Date Changed | Initiator By (Who Changed) | Change Checked By | Notes |
| CHGE201 | Bug 001 | f51dc08 | The ProjectPhase is to be set to 0 in both CompanyProject constructors so that “Feasibility” is not skipped in CompanyEmailSystem | CompanyProjcet.java (line 23) | 02.05.2018 | 02.05.2018 | Justin Ng | Aiden Reed |  |
| CHGE 202 | Bug 202 | f51dc08 | The comparison statement is changed to be compared to the length of project phases - 1 and the project phase incrementor was moved inside the else statement | CompanyProjcet.java (line 91 & line 94) | 02.05.2018 | 02.05.2018 | Justin Ng | Aidan Reed | The change didn’t fully resolve the issue as the project could still go past the end of the array because the conditional was not >= - AR 02.05.2018 |
| fbea9fd | The conditional only checked when the phase was greater than which resulted in the phase going 1 past the length of the array. The conditional should include a = parameter for greater or equals. | CompanyProjcet.java (line 91 & line 94) | 02.05.2018 | 02.05.2018 | Justin Ng | Aidan Reed | This now passes the test and ensures the phase does not go past the end of the projects array. - AR 02.05.2018 |
| CHGE203 | Bug 204 |  | Add a statement to initialize the next phase of the ProjectEmail array list after the phase has been checked and incremented | CompanyProjcet.java (line 94-95) | 02.05.2018 |  |  |  |  |

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** |
| **Bug 201** | Project Phase Name set to Design when created instead of Feasibility - This also impacts the default constructor and Phase ID tests | 24: Company Project.class | 201  202 | CHGE201 | Set the Project phase to 0 more details in CHGE201 | Med | 25.04.2018 | YES | 02.05.2018 | Justin NG | N/A | The tests listed in test id are tests that first outlined this bug. |
| **Bug 202** | Project Phase ID goes over the number of different phases when going to next stage. | 90-98: Company Project.class | 219 | CHGE202 | Move the phase++ inside the else clasue and change condition to length -1 | Med | 25.04.2018 | YES | 02.05.2018 | Aidan Reed / Christian Stubbs | N/A | The initial fix put in place did not fully fix the issue as the project phase was still incremented before the condition. After the second amendment the code was correct. – AR 02.05.2018 |
| **Bug 203** | 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. |
| **BUG204** | When moving the project to the next stage the program fails with a null pointer exception when adding an email. This is because the constructor only initializes the first part of the Projects emaila array. | 90: CompanyProject. Class | 213 | CHGE203 | Add a statement to initialize the current phase after it has moved | High | 02.05.2018 |  |  | Christian Stubbs |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |