­­­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.

Email Class

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID | Function Name | Test Aim (From perspective of method/how method handles this) | Pre-Requisites | Inputs (+ examples) | Expected Output(s) | Actual Output | Test Created By | Source: Spec or code inspection | Pass / Fail | Date (Completed by) | Notes (Questions/Assumptions) |
| 100 | Setter for Email Message Body  **(setMessage)** | Test the method correctly sets the email message and then compare that to what is retrieved from getMessage method | Email object with with sender, receiver and subject via the constructor with emailMessage left blank | “This is a test email for using unit testing”  Stored in kEMAILBODY1 Constant | “This is a test email for using unit testing”  After calling the getMessage method once the message has been set | Stackoverflow error | Aidan Reed |  | FAIL | 26.04.2018 | Unable to determine if message was set correctly as when trying to retrieve the message using emailMessage method received a stack overflow error. |
| 101 | Setter for Email Message Body  **(setMessage)** | Test the method correctly sets the email message to an empty string “” and then compare that to what is retrieved from getMessage method | Email object with with sender, receiver and subject via the constructor with emailMessage left blank | “”  Stored in kEMAILBODY2 Constant | “”  After calling the getMessage method once the message has been set | Stackoverflow error | Aidan Reed |  | FAIL | 26.04.2018 | Unable to determine if message was set correctly as when trying to retrieve the message using emailMessage method received a stack overflow error. |
| 102 | Check Validity of Email Function  **(isValid)** | Tests the email object with no attributes set in the constructor meaning they are null | New Company Email Object | N/A | False | False | Aidan Reed |  | PASS | 26.04.2018 | The worst case where no values are set |
| 103 | Check Validity of Email Function  **(isValid)** | Tests the email object with All attributes set in the constructor | New Company Email Object | Sender:  [nottingam@me.co.uk](mailto:nottingam@me.co.uk)  (**kFROMADDR1)**  To:  [londong@me.co.uk](mailto:londong@me.co.uk)  (**kTOADDR1)**  Subject:  “New Email System Requirements Document”  **(kSUBJECT1]**  Email Message:  “Test Email”  **(kEMAILBODY3)** | True | True | Aidan Reed |  | PASS | 26.04.2018 | The best case where all values are set |
| 104 | Check Validity of Email Function  **(isValid)** | Tests the email object with one attribute set – the from address | New Company Email Object | Sender:  [nottingham@me.co.uk](mailto:nottingham@me.co.uk)  **(kFROMADDR1)** | False | False | Aidan Reed |  | PASS | 26.04.2018 | The following 4 tests of similar nature test the individual cases for each attribute to ensure all attributes are checked in the isValid Function |
| 105 | Check Validity of Email Function  **(isValid)** | Tests the email object with one attribute set – the To address | New Company Email Object | To:  [london@me.co.uk](mailto:nottingham@me.co.uk)  **(kTOADDR1)** | False | False | Aidan Reed |  | PASS | 26.04.2018 |  |
| 106 | Check Validity of Email Function  **(isValid)** | Tests the email object with one attribute set – the Subject | New Company Email Object | Subject:  “New Email System Requirements”  **(kSUBJECT1)** | False | False | Aidan Reed |  | PASS | 26.04.2018 |  |
| 107 | Check Validity of Email Function  **(isValid)** | Tests the email object with one attribute set – the message body | New Company Email Object | Email Message:  “This is a test email for unit testing”  **(kEMAILBODY1)** | False | False | Aidan Reed |  | PASS | 26.04.2018 |  |
| 108 | toString() override  **(toString)** | Checks the toString override method returns a value of type string | New Company Email Object | N/A | (String) | (String) | Aidan Reed |  | PASS | 26.04.2018 |  |
| 109 | toString() override  **(toString)** | Checks the correct subject is returned when set during the test | New Company Email Object | Subject:  “New Email System Requirements”  **(kSUBJECT1)** | “New Email System Requirements” | “New Email System Requirements” | Aidan Reed |  | PASS | 26.04.2018 |  |
| 110 | toString() override  **(toString)** | Checks the correct subject is returned when set during the test to an empty string | New Company Email Object | Subject:  “” | “no subject” | “no subject” | Aidan Reed |  | PASS | 26.04.2018 |  |
| 111 | toString() override  **(toString)** | Checks the correct subject is returned when not set i.e null | New Company Email Object | Subject:  “” | “no subject” | Null pointer exception | Aidan Reed |  | FAIL | 26.04.2018 | This fails as the method tries to return null as type string and causes a null pointer exception. Although the class description does not include this behavior the test was included to find potential bugs that cause the system to crass |

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 | Stack over flow error caused after setting the email Message using setMessage method and trying to compare the message to the one set using the emailMethod Message | 40 CompanyProject.class | 101/102 |  | It appears instead of returning the email message it calls the emailMessage function again remove the () for the method call and look to rename the method to “getEmailMessage” | High | 26.04.2018 |  |  | Aidan Reed |  | The emailMessage method bug was found when performing tests for setMessage |
| 2 | Null Pointer exception when trying to print the subject of a email that has not been set | 73 CompanyEmail.class | 111 |  | Perform a check to see if the member variable is null and return the no subject string | High1 | 26.04.2018 |  |  | Aidan Reed |  | Although the class description does not include this behavior the test was included to find potential bugs that cause the system to crass |