­­­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 (All values specified are exemplary and indicative of the types of values required as inputs.) | Expected Output(s) | Actual Output | Test Created By | Source: Spec or code inspection | Pass / Fail | Completed by | Notes (Questions/Assumptions) |
| 101 | Default Constructor | To see how the method will react when no variables are passed to the constructor. | None | None | Initialises an email ready for populating. | “emailMessage()” method returns itself rather than the variable “emailMessage”. | Athullya Roy | Specification Document – Class Document | FAIL | Ram Raja 26/04/2018 |  |
| 102 | Main Constructor | To see how the method will react when receiving all four string parameters in correct form. | None | Sender’s Email: [psyra7@hotmail.co.uk](mailto:psyra7@hotmail.co.uk)  Recipient’s Email: [psyrrr1@hotmail.co.uk](mailto:psyrrr1@hotmail.co.uk)  Subject:  “New Project – New Project Concept”  Email Body:  “Hi,  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ullamcorper tellus id nisi mattis, consequat vulputate lorem eleifend. Sed sollicitudin odio eu elit porta, a dictum urna tempus.  Regards,  Athullya” | Initialises an email ready to be sent, with a sender’s email, recipient’s email, subject and body. | “emailMessage()” method returns itself rather than the variable “emailMessage”. | Ram Raja | Specification Document – Class Document | FAIL | Ram Raja 26/04/2018 |  |
| 103 | Main Constructor | To test interactivity between both the Default and Main Constructor when all parameters are not populated. In this case all but the recipient’s email has been given. | None | Recipient’s email is null. | Initialises an email with all but the recipient’s email given, ready to be populated. | “emailMessage()” method returns itself rather than the variable “emailMessage”. | Ram Raja | Code Inspection | FAIL | Ram Raja 26/04/2018 |  |
| 104 | Main Constructor | To test interactivity between both the Default and Main Constructor when only one parameter is given, in this case the sender’s email. | None | Sender’s Email: [psyar8@live.co.uk](mailto:psyar8@live.co.uk) | Initialises an email with the sender’s email; with recipient’s email, subject and body as null, ready to be populated. | “emailMessage()” method returns itself rather than the variable “emailMessage”. | Athullya Roy | Code Inspection | FAIL | Ram Raja 26/04/2018 |  |
| 105 | Getter for Sender’s email address | To ensure the getter outputs the sender’s email when specified. | None | Sender’s Email: [psyar8@live.co.uk](mailto:psyar8@live.co.uk) | Returns the sender’s email as a string. | The given sender’s email is returned. | Ram Raja | Specification Document – Class Document | PASS | Ram Raja 26/04/2018 |  |
| 106 | Getter for Sender’s email address | To test how the method will respond when the sender’s email is not set. | None | None | Returns sender’s email as null. | The sender’s email is returned as null. | Athullya Roy | Specification Document – Class Document | PASS | Ram Raja 26/04/2018 |  |
| 107 | Getter for Recipient’s Email Address | To ensure the getter outputs the recipient’s email when specified. | None | Recipient’s Email: [psyrrr1@hotmail.co.uk](mailto:psyrrr1@hotmail.co.uk) | Returns the recipient’s email as a string. | The given recipient’s email is returned. | Ram Raja | Specification Document – Class Document | PASS | Ram Raja 26/04/2018 |  |
| 108 | Getter for Recipient’s Email Address | To test how the method will respond when the recipient’s email is not set. | None | None | Return’s recipient’s email as null. | The recipient’s email is returned as null. | Athullya Roy | Specification Document – Class Document | PASS | Ram Raja 26/04/2018 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Could test setter functionality with a null parameter? |  |  |  |  |  |  |  |  |  |

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 |
| 1 | ? | BUG006 | Return statement of emailMessage() getter method changed from “emailMessage()” method to “emailMessage” variable. | CompanyEmail | 26/04/2018 |  | Ram Raja |  | Found whilst performing tests on default constructor. |
|  |  |  |  |  |  |  |  |  |  |

Bug Fix List

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Problem ID | Problem Description | Class, Method & Line Number | 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 |
| BUG005 | The setFrom() and setTo() email methods only check for a “@” being present in the entered string, not at any specific position. Therefore, an incorrect email address could be entered. i.e. “psyrr1nottinghamac.uk@” | CompanyEmail, setFrom() at line 39 & setTo() at line 45 |  |  | Specify clearly where the @ should be. | Med | 09/03/2018 |  |  | Ram Raja |  |  |
| BUG006 | emailMessage() method returns itself, not the value stored in the variable “emailMessage”. The method should be performing as a getter for the variable “emailMessage”. | CompanyEmail, emailMessage() at line 35 | ***Test ID to check if the bug fix works?***  ***Or test ID for where the bug was found?*** | 1 | Change return statement to the variable “emailMessage”, instead of the method “emailMessage()”. | High | 26/04/2018 |  |  | Ram Raja | No?  ***(Need clarification on “functions” – do you mean other methods within the code? Or other component/features of the software?*** | Found whilst performing tests on default constructor. |