­­­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.

Company Email System (Main Class)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID | Function Name | Test Aim | Prerequisite | Inputs | Expected Output(s) | Actual Output | Test Created By | Source: Spec or code inspection | Pass / Fail | Date (Completed by) | Notes |
| 301 | Main Method | Testing the user sees the correct menus when not in a project. | Set Current Project Variable to 0 | No inputs | A list of user options:   * See a list of projects * Add a new project * Viewing a project * Exit the system |  | Aidan Reed | Specification - Class Description |  |  |  |
| 302 | Main Method | Tests the user cannot see the should not be able to see the commands offered by the loop. | Create a project for the test | Select project created in prerequisite stage | The user should have access to the project and no menus. |  | Aidan Reed | Specification – Class Description |  |  |  |
| 303 | Main Method | Tests the ArrayList can grow to large size | Create 10,000 Projects | N/A | Program continues to run displaying the user command menu |  | Aidan Reed | Specification – Class Description |  |  | Tests the program does not crash or thrown an exception when handling a large number of projects. Also tests to ensure the array is not statically set. |
| 304 | List Project | Checks that the function returns the correct number of emails in the current phase. | Add 10 emails to the current project phase | Select the test project | A string with the count in the format :  New Project [Design] – 10 Emails |  | Aidan Reed | Specification – Class Description |  |  |  |
| 305 | List Project | Checks that the function can handle large email counts. | Create 100,000 emails for current project phase | Select the test project | A string with the count in the format:  New Project [Design] – 10 Emails  The execution time should be below 5 seconds. |  | Aidan Reed | Specification – Class Description |  |  | This test includes a performance test for the function to ensure the function is efficient when handling large amounts of data. |
| 306 | Add Project | Ensures the users title added as input is correctly added to the project | Set the Current Project Variable to 0 | Select the new project command and enter the Marketing Campaign title | The project should be created and display the name Marketing Campaign |  | Aidan Reed | Specification – Class Description |  |  |  |
| 307 | Add Project | Checks that the project is created if the user does not enter a name | Set the Current Project Variable to 0. | Select the new project command and do not enter title | The project should be created with no name but state project ID. |  | Aidan Reed | Specification – Class Description |  |  | The Class Description Function 3 does not state what should be displayed if the user omits to enter the name. The test ensures the Project ID is currently presented. |
| 308 | List Emails | Tests that the function returns the correct phase ID for project | Create project Orion and set phase ID TO 2 | Phase ID 1 | Returns the Display the emails for current project at phase ID 1 |  | Aidan Reed | Specification – Class Description |  |  |  |
| 309 | List Emails | Tests the function with an invalid phase ID | Create project Orion and set Phase ID to 2 | Phase ID 20000 | Should not return any emails. |  | Aidan Reed | Specification – Class Description |  |  |  |
| 310 | List Emails | Test the function with no phase ID | Create Project Orion and set Phase ID to 2 | N/A | Should return emails for the current project at phase ID 2 |  | Aidan Reed | Specification – Class Description |  |  |  |
| 311 | List Phases | Call the function from the user interface | Create Project Orion and Select Project and set phase ID.  Add 100 emails phase 1  Add 200 emails phase 2  Add 20 emails phase 3 | N/A gets current project from global variable | Returns the phases for the current project with email counts:  Add 100 emails phase 1  Add 200 emails phase 2  Add 20 emails phase 3 |  | Aidan Reed | Specification – Class Description |  |  |  |
| 312 | List Contacts | The function is called from user interface in the user context | Create Project Alpha and add 20 users | Select the project and list contacts button | Returns the list of 20 users set in the prerequisite stage |  | Aidan Reed | Specification – Class Description |  |  |  |
| 313 | List Contacts | Call function when not in project | Current project value set to 0 | N/A | The program should not crash and thrown an exception |  | Aidan Reed | Specification – Class Description |  |  | It’s important to check that the class implantation avoids error prone constructs and does not crash if the project variable is not set correctly. |
| 314 | Change Project Phase | Ensures the next phase function is called correctly | Create Project Beta and set project phase 1 | Select Next phase button | Calls the next phase function and prints out a message to say it was successful |  | Aidan Reed | Specification – Class Description |  |  |  |
| 315 | Change Project Phase | Tests to check the last phase message is displayed | Create Project November and set project phase to last phase | Select Next phase button | Calls the next phase function and prints out message informing user already in last phase |  | Aidan Reed | Specification – Class |  |  |  |

Email Class

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID (class/function/test) | Function Name | Test Aim (From perspective of system/how system handles this) | Pre-Requisites | Inputs | Expected Output(s) | Actual Output | Test Created By | Source: Spec or code inspection | Pass / Fail | Date (Completed by) | Notes |
| 101 | Default Constructor | To see how the system will react when no variables are passed to the constructor. | None | None | Create an empty email ready for populating |  | Ram Raja | Specification Document – Class Document |  |  |  |
| 102 | Main Constructor | To see how the system |  |  |  |  |  |  |  |  |  |
| 102 | Get Sender’s Email Address | The sender’s email must be |  |  |  |  |  |  |  |  |  |
| 102 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 102 | Get Receiver’s Email Address |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

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 | |
| 201 | Default Constructor | | Testing to see if it prepares the project with the initial variables | | | None | | None | | Should increment project counter, get default title, set project phase to the first stage and prepare private variables | |  | Justin Ng | Specification Document – Class Document | |  |  | |  | |
| 202 | Main Constructor | | Tests to see if it prepares the project with the initial variables and the defined project title | | | Default Constructor | | Title parameter | | Should initialise the project as in the default constructor but should set the title as the parameter passed | |  | Christian Stubbs | Specification Document – Class Document | |  |  | |  | |
| 203 | Get Project ID | | Checks if the Project ID is returned | | | Project ID | | None | | Should return the Project ID as an int | |  | Justin Ng | Specification Document – Class Document | |  |  | |  | |
| 204 | Get Project Title | | See if the function checks if the title is a valid length (Over 10 chars) and then sets it to the class variable | | | Project Title | | None | | Should return the Project Title as a string | |  | Christian Stubbs | Specification Document – Class Document | |  |  | |  | |
| 205 | Set Project Title | | Tests to take a valid project title as a parameter | | | None | | Project Title | | Should set the project title parameter and set it to the class variable if valid | |  | Justin Ng | Specification Document – Class Document | |  |  | |  | |
| 206 | Check Contact Function | | Tests to see if contact is in the contact array list | | | Email Address | | None | | Should return true if email address is in list and false if it is not | |  | Christian Stubbs | Specification Document – Class Document | |  |  | |  |
| 207 | Add Contact function | | Checks if the email address is added to the Project Contacts ArrayList via a given parameter | | | Project Contacts ArrayList | | Valid Email Address | | Adds valid email address to the Project Contacts ArrayList | |  | Justin Ng | Specification Document – Class Document | |  |  | |  |
| 208 | | Add Email Function | Checks if the ArrayList of CompanyEmails is fetched from the **current** phase, and then adds the email. The function should check the validity of the email and should display an error. Testing will also be included to add new recipients to contacts. | ArrayListof CompanyEmails from Project Emails array | CompanyEmail Object | | Adds email address to the CompanyEmails ArrayListif the email is valid, and throws an error if it is not valid |  | Christian Stubbs | | Specification Document – Class Document |  |  | |  | | |
| 209 | | Get Emails for current phase | Test to see if this function returns the emails in the correct phase of the project (current) | None | None | |  |  | Justin Ng | | Specification Document – Class Document |  |  | |  | | |
| 210 | | Get emails for specific phase | Test to see if this function returns the emails in the correct phase of the project (with a parameter specifying the desired phase) | Int – phase ID |  | |  |  | Christian Stubbs | | Specification Document – Class Document |  |  | |  | | |
| 211 | | Next Phase | This will check if the function moves the project to the next phase. Tests if the correct value is returned | Current Phase | None | |  |  | Justin Ng | | Specification Document – Class Document |  |  | | Will there be a "change phase" function to allow the phase to be changed to a desired stage? 09/03 | | |
| 212 | | Get Phase Name Function | Will see if the correct name of function is returned |  | Phase ID | |  |  | Christian Stubbs | | Specification Document – Class Document |  |  | |  | | |
| 213 | | Get Phase ID | Checks if the correct phase ID is returned |  | None | |  |  | Justin Ng | | Specification Document – Class Document |  |  | |  | | |
| 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 |  |  | |  | | |

Big/Fix List

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 included an **ID** for the problem so that it is easier to refer back to. Next we included a column that states the **line of the program** which the bug was found in. This will make it easier to see which part of the program contains errors. Alongside this, stating the line of code will make the code easier to trace to, as a developer can easily look at the table and identify where the bug is in the code. We have also included a column that contains the **Test ID**. This is so that even after fixing the bug, a test might be needed to check that part of the code again. If this is the case, the test created should be listed so that there is evidence for the bugged code properly being tested. Alongside this, we have also included the **date the problem was identified**. This will show the developers how recent the bug has been discovered. This could indicate how relevant or important it is to fix the bug. We have also included a column that states whether the bug has been **fixed** or not. This will show the developers if they need to be concerned about the bug or if it has been resolved already. We then added another 2 columns that states the **date** that the problem was fixed on and the **name** of the person who identified/fixed the bug. This will be essential as it provides proper documentation towards the project. It shows how the project is being developed and hold the developer accountable for the bug fixed, so if there are any enquiries in the future concerning the bug, the developer can be contacted. We also have a column that states whether the bugged line **is linked to any other functions** or parts of code. For example, if a function is called on a bugged code, the error might actually be in the function. Hence it is a good idea to take a note of this so that the function can be tested. A section is also provided for additional **notes**.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Problem ID | Problem Description | Line of code | Test ID (if created) | 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 | Limit of 7 emails in the new ArrayList object. This seems like a small amount for a project that will likely have a lot of back and forth communication. | CompanyProject, 8 |  | Increase the value to something more feasible | Med | 09/03 |  |  | Christian(Identified) |  |  |
| 2 | No error thrown if email invalid in Company Project Class | CompanyProject, 58 |  | Return an error if email is invalid | Low | 09/03 |  |  | Justin (Identified) |  | Print error message necessary? |
| 3 | ListContacts method does not return email address in CompanyEmailSystem Class | CompanyEmailSystem, 151 |  |  | Med | 09/03 |  |  |  |  |  |
| 4 | No validation on validity of project title - should be minimum of 10 chars | CompanyEmailsystem, 110 |  | Return an error if the title is not valid | Med | 09/03 |  |  | Christian |  |  |
| 5 |  | CompanyEmail,39&45 |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |