BugTracker

Test Cases

Version 1.3

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 21/10/2016 | 1.0 | Begin Test Cases, Add some test cases for login, register, edit profile and view profile. | Xinwei Jiang |
| 22/10/2016 | 1.1 | Added test cases for Add Bug, View Bug, Search User, Edit Bug, Search Bug, Filter Bug | Tessa Marelic |
| 25/10/2016 | 1.2 | Add more test cases about register | Xinwei Jiang |
| 25/10/2016 | 1.3 | Added non-functional performance test cases and User Interface test cases | Tessa Marelic |

**Functional requirement: The system will provide a user of any type with a GUI to log in**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system will provide a user of any type with a GUI to log in | Test case 1 |
| Input | Enter valid E-mail and password which are stored in the database  E-mail/Username: xj323@email.com  Password: 123 | |
| Expected output | The main screen which is search bug interface will be display. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system will provide a user of any type with a GUI to log in | Test case 2 |
| Input | Enter valid Username and password which are stored in the database  E-mail/Username: xj323  Password: 123 | |
| Expected output | The main screen which is search bug interface will be display. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system will provide a user of any type with a GUI to log in | Test case 3 |
| Input | Enter invalid E-mail which cannot be found in the database and valid password which is stored in the database  E-mail/Username: xj323@  Password: 123 | |
| Expected output | A warning message “email or password is wrong” will be displayed. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system will provide a user of any type with a GUI to log in | Test case 4 |
| Input | Enter invalid Username which cannot be found in the database and valid password which is stored in the database  E-mail/Username: abcd  Password: 123 | |
| Expected output | A warning message “email or password is wrong” will be displayed. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system will provide a user of any type with a GUI to log in | Test case 5 |
| Input | Enter valid E-mail/Username which is stored in the database and invalid password which is not matched with the right one.  E-mail/Username: xj323  Password: 1234567 | |
| Expected output | A warning message “email or password is wrong” will be displayed. | |

**Functional requirement: The system should provide a guest to register an account in the system.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a guest to register an account in the system. | Test case 6 |
| Input | Enter the valid last name, first name, username, E-mail and password.  Last name: Black  First name: Lucy  Email: lucy@email.com  User name: lucy  Password: 12345  Verify password: 12345 | |
| Expected output | A message “The new account created!” will be displayed, and the data are stored in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a guest to register an account in the system. | Test case 7 |
| Input | Enter the valid last name, first name, username, E-mail and password. But the passwords are not matched  Last name: Black  First name: Lucy  Email: lucy@email.com  User name: lucy  Password: 12345  Verify password: 123 | |
| Expected output | A warning message “The passwords are not matched.” will be displayed. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a guest to register an account in the system. | Test case 8 |
| Input | Enter the valid last name, first name, username and password. But the Email already existed.  Last name: Black  First name: Lucy  Email: xj323@email.com  User name: lucy  Password: 12345  Verify password: 12345 | |
| Expected output | A warning message “The user already exists” will be displayed | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a guest to register an account in the system. | Test case 9 |
| Input | Enter the valid first name, username, E-mail and password. But leave the line of last name empty.  Last name:  First name: Lucy  Email: xj323@email.com  User name: lucy  Password: 12345  Verify password: 12345 | |
| Expected output | A warning message “Error: 1 or more fields are empty” will be displayed | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a guest to register an account in the system. | Test case 10 |
| Input | Enter the valid last name, username, E-mail and password. But leave the line of first name empty.  Last name: Black  First name:  Email: xj323@email.com  User name: lucy  Password: 12345  Verify password: 12345 | |
| Expected output | A warning message “Error: 1 or more fields are empty” will be displayed | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a guest to register an account in the system. | Test case 11 |
| Input | Enter the valid first name, first name, E-mail and password. But leave the line of user name empty.  Last name: Black  First name: Lucy  Email: xj323@email.com  User name:  Password: 12345  Verify password: 12345 | |
| Expected output | A warning message “Error: 1 or more fields are empty” will be displayed | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a guest to register an account in the system. | Test case 12 |
| Input | Enter the valid first name, first name, E-mail and user name. But leave the line of password and verify password empty.  Last name: Black  First name: Lucy  Email: xj323@email.com  User name: lucy  Password:  Verify password: | |
| Expected output | A warning message “Error: 1 or more fields are empty” will be displayed | |

**Functional requirement: The system should provide any user with a GUI to modify their own profile details, name, E-mail, profile picture and subscriptions.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide any user with a GUI to modify their own profile details, name, E-mail, profile picture and subscriptions. | Test case 13 |
| Input | Edit the first name.  Lucy changes to Amy | |
| Expected output | A message “Edit successfully” will be displayed. The first name of this user should be changed to Amy in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide any user with a GUI to modify their own profile details, name, E-mail, profile picture and subscriptions. | Test case 14 |
| Input | Edit the last name.  Black changes to White | |
| Expected output | A message “Edit successfully” will be displayed. The last name of this user should be changed to White in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide any user with a GUI to modify their own profile details, name, E-mail, profile picture and subscriptions. | Test case 15 |
| Input | Edit the Email.  lucy@email.com changes to lucy@gmail.com | |
| Expected output | A message “Edit successfully” will be displayed. The Email of this user should be changed to lucy@gmail.com in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide any user with a GUI to modify their own profile details, name, E-mail, profile picture and subscriptions. | Test case 16 |
| Input | Edit the gender.  Click the check box Female. | |
| Expected output | A message “Edit successfully” will be displayed. The gender of this user should be changed to female in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide any user with a GUI to modify their own profile details, name, E-mail, profile picture and subscriptions. | Test case 17 |
| Input | Edit the profile picture.  Upload a link of the profile picture | |
| Expected output | A message “Edit successfully” will be displayed. The profile picture of this user should be changed to the new one in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide any user with a GUI to modify their own profile details, name, E-mail, profile picture and subscriptions. | Test case 18 |
| Input | Edit the subscriptions.  Delete some of the subscriptions | |
| Expected output | A message “Edit successfully” will be displayed. The subscriptions of this user should be changed in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide any user with a GUI to modify their own profile details, name, E-mail, profile picture and subscriptions. | Test case 19 |
| Input | User click edit profile the button in view profile interface which displays the same user’s profile. | |
| Expected output | The edit profile interface will be displayed. | |

**Functional requirement: The system should provide any user with a GUI to modify their own password.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide any user with a GUI to modify their own password | Test case 20 |
| Input | Enter the valid old password, new password and verify password  Old password: 123456  New password: 123abc  Verify password: 123abc | |
| Expected output | The password which is stored in the database should be changed. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide any user with a GUI to modify their own password | Test case 21 |
| Input | Enter the invalid old password, valid new password and verify password  Old password: 1234  New password: 123abc  Verify password: 123abc | |
| Expected output | A warning message “The password is wrong” will be displayed. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide any user with a GUI to modify their own password | Test case 22 |
| Input | Enter the valid old password, new password and invalid verify password  Old password: 123456  New password: 123abc  Verify password: 123ab | |
| Expected output | A warning message “The passwords are not matched” will be displayed. | |

**Function requirement: The system should provide a System Admin with a GUI to edit an existing non-administrator user’s profile in the Bug Tracker System.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a System Admin with a GUI to edit an existing non-administrator user’s profile in the Bug Tracker System. | Test case 23 |
| Input | Edit the profile as the admin. | |
| Expected output | The reputation, status and role are not read only. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a System Admin with a GUI to edit an existing non-administrator user’s profile in the Bug Tracker System. | Test case 24 |
| Input | Edit the first name.  Lucy changes to Amy | |
| Expected output | A message “Edit successfully” will be displayed. The first name of this user should be changed to Amy in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a System Admin with a GUI to edit an existing non-administrator user’s profile in the Bug Tracker System. | Test case 25 |
| Input | Edit the last name.  Black changes to White | |
| Expected output | A message “Edit successfully” will be displayed. The last name of this user should be changed to White in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a System Admin with a GUI to edit an existing non-administrator user’s profile in the Bug Tracker System. | Test case 26 |
| Input | Edit the Email.  lucy@email.com changes to lucy@gmail.com | |
| Expected output | A message “Edit successfully” will be displayed. The Email of this user should be changed to lucy@gmail.com in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a System Admin with a GUI to edit an existing non-administrator user’s profile in the Bug Tracker System. | Test case 27 |
| Input | Edit the gender.  Click the check box Female. | |
| Expected output | A message “Edit successfully” will be displayed. The gender of this user should be changed to female in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a System Admin with a GUI to edit an existing non-administrator user’s profile in the Bug Tracker System. | Test case 28 |
| Input | Edit the profile picture.  Upload a link of the profile picture | |
| Expected output | A message “Edit successfully” will be displayed. The profile picture of this user should be changed to the new one in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a System Admin with a GUI to edit an existing non-administrator user’s profile in the Bug Tracker System. | Test case 29 |
| Input | Edit the subscriptions.  Delete some of the subscriptions | |
| Expected output | A message “Edit successfully” will be displayed. The subscriptions of this user should be changed in the database. | |

**Function requirement: The system should provide a user with a GUI to view any non-administrator user profile.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI to view any non-administrator user profile. | Test case 30 |
| Input | User via search user interface to view their own profile or other user’s profile. | |
| Expected output | The view profile interface will be displayed with the user’s information which is stored in the database. | |

**Functional requirement: The system should provide a user with a GUI form to enter new bugs. The form should include:**

* **Application the reporter is reporting a bug on**
* **Title**
* **Text area to enter description of bug**

**Optional information: Version, file uploads, repeatable walkthrough, platform, severity, priority**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI to enter new bugs. | Test case 31 |
| Input | Select Add Bug button on the main window | |
| Expected output | The add bug gui will open displaying the compulsory information text input boxes only which are writable. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI to enter new bugs. | Test case 32 |
| Input | Enter dummy compulsory information only - Title, Select an Application from drop down list, enter description, click submit | |
| Expected output | A message is displayed – “your bug has been added, thank you. Your bug ID number is – display new ID number”. New bug data should be entered in the bug table with the optional information columns blank. Text boxes on ui should now be blank. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI to enter new bugs. | Test case 33 |
| Input | Click on optional information button | |
| Expected output | The gui expands to provide the user with writable text boxes to add optional information: These are:  Version  Platform  Component  Priority  Severity  Walkthrough  File uploads | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI to enter new bugs. | Test case 34 |
| Input | Enter dummy data into each text box including the compulsory data section and upload a file. Press submit button. | |
| Expected output | A message is displayed – “your bug has been added, thank you. Your bug ID number is – display new ID number”. All data entered should be in the bug table in the database. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI to enter new bugs. | Test case 35 |
| Input | Upload a file then press the “remove selected file” button | |
| Expected output | The file should disappear from the drop down list of files | |

**Functional requirement: The system should provide a user with a GUI form to View a Bug. Compulsory information to display includes:**

**ID**

**Status**

**Application**

**Version**

**Description**

**Date Submitted**

**Assigned to**

**Identified by**

**Priority**

**Desirable information includes:**

**Platform**

**Severity**

**CC List**

**Files**

**Comments**

**History**

**Functional requirement: The system should provide a user of any type (Reporter, Developer, Triager, and Reviewer) with a GUI form to subscribe to the bug.**

**Functional requirement: The system should provide a user of any type with a gui to add a comment to an existing bug.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI to view a bug, subscribe to a bug and add a comment. | Test case 36 |
| Input | Search bug by entering bug ID 1 in the search field via the main window then double click on the bug to view | |
| Expected output | A gui will display showing the compulsory information specific to that bug | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI to view a bug, subscribe to a bug and add a comment. | Test case 37 |
| Input | Select the “view more” button | |
| Expected output | Desirable information specific to that bug will be displayed | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI to view a bug, subscribe to a bug and add a comment. | Test case 38 |
| Input | Select “add to CC List“ button. Enter your username in the text area, click add button. | |
| Expected output | The CC list text area becomes editable so the user can enter their username. Once add button is clicked a message informs the user they are now subscribed to the bug ID number. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI to view a bug, subscribe to a bug and add a comment. | Test case 39 |
| Input | Select “add to CC List“ button to verify only valid user names can be added to the list. Enter an invalid user name. Eg 1234. | |
| Expected output | A message displays informing user that an invalid user name has been entered. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI to view a bug, subscribe to a bug and add a comment. | Test case 40 |
| Input | Enter a comment in the “add comment” text box and click “Add Comment” | |
| Expected output | A thank you message will display similar to: “Your comment has been added to Bug ID – id number, thank you. The new comment will be added to the bug Id’s comment field in the database. Check this by searching for the bug, view it again and your comments should display in comments field. | |

**Functional requirement: The system should provide a user with a GUI form to search for users of any type.**

**Search criteria include: First Name, Last Name, Email, Role, Account**

**Status and Key word.**

**Functional requirements: The system should allow a user to filter users of any type by First Name, Last Name and Account Status.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI form to search for users of any type. The system should allow a user to filter users of any type by name, account status. | Test case 41 |
| Input | From main window click on user -> search user. | |
| Expected output | Search User gui displays. It displays all users in the system in the form of a table as well as a search facility to search by username, name, email, country, role, status. It also has a facility to filter users by status, role, country and date. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI form to search for users of any type. The system should allow a user to filter users of any type by account status, Role, Country, Member Since Date. | Test case 42 |
| Input | Search for one valid user by entering their data and select each search criteria. Eg enter username of user then click search, then enter email and select search …etc. | |
| Expected output | The only user displayed in the table will be the searched user for each criteria. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI form to search for users of any type. The system should allow a user to filter users of any type by account status, Role, Country, Member Since Date. | Test case 43 |
| Input | Search for one invalid user by entering their data and select each search criteria. Eg enter invalid username of user then click search, then enter invalid email and select search …etc. | |
| Expected output | Message displays informing the user that the searched user does not exist. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI form to search for users of any type. The system should allow a user to filter users of any type by account status, Role, Country, Member Since Date. | Test case 44 |
| Input | Filter all users by selecting “Status Type” | |
| Expected output | Only those users with specified status type will display in the table. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI form to search for users of any type. The system should allow a user to filter users of any type by account status, Role, Country, Member Since Date. | Test case 45 |
| Input | Filter all users by selecting “Role” | |
| Expected output | Only those users of specified role will display in the table. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI form to search for users of any type. The system should allow a user to filter users of any type by account status, Role, Country, Member Since Date. | Test case 46 |
| Input | Filter all users by selecting “Country” | |
| Expected output | Only those users with specified country type will display in the table. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a user with a GUI form to search for users of any type. The system should allow a user to filter users of any type by account status, Role, Country, Member Since Date. | Test case 47 |
| Input | Filter all users by selecting “Member Since Dates” | |
| Expected output | Only those users who have been members between these dates will display in the table. | |

**Functional Requirement: The system should allow a Reporter, Developer, Reviewer to clock up reputation points to be displayed in their user profile and whenever their name is used in commenting or bug reporting or bug solving.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should allow a Reporter, Developer, Reviewer to clock up reputation points to be displayed in their user profile and whenever their name is used in commenting or bug reporting or bug solving. | Test case 48 |
| Input | Add a bug as a reporter. Then open profile to check reputation points. | |
| Expected output | Reputation points have increased after correct submission of 1 bug. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should allow a Reporter, Developer, Reviewer to clock up reputation points to be displayed in their user profile and whenever their name is used in commenting or bug reporting or bug solving. | Test case 49 |
| Input | Reviewer clicks verify developer after reviewing a bug fix classed as good. Enter system as a Developer and check reputation points in profile. | |
| Expected output | Reputation points have increased after correct fix of a bug and it has been verified by a reviewer. | |

**Functional Requirement: The system should provide a Triager with a GUI to Assign bugs to developers. This GUI shows all currently un-assigned bugs and all developers.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Triager with a GUI to Assign bugs to developers. | Test case 50 |
| Input | Double click bug, edit bug and assign bug button | |
| Expected output | A gui will display showing a drop down list of developer user names. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Triager with a GUI to Assign bugs to developers. This GUI shows all currently un-assigned bugs and all developers. | Test case 51 |
| Input | Click the developer’s name, then submit | |
| Expected output | Nothing will be displayed in the main screen. | |

**Functional Requirement: The system should provide a Triager with a GUI to view current bugs assigned to each developer.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Triager with a GUI to view current bugs assigned to each developer. | Test case 52 |
| Input | Enter developer name in search field from the main window, click search | |
| Expected output | All bugs assigned to that developer will display | |

**Functional Requirement: The system should provide a facility for the developer to change the status of the bug. A developer can only change a bug assigned to them.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a facility for the developer to change the status of the bug. A developer can only change a bug assigned to them | Test case 53 |
| Input | Log in as a developer, double click a bug on the main window, then press edit. Change status to a different status. Click submit. | |
| Expected output | A message displays acknowledging the change in bug status. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a facility for the developer to change the status of the bug. A developer can only change a bug assigned to them | Test case 54 |
| Input | Search for the bug you previously changed the status on from main window. Double click. | |
| Expected output | Check the bugs status is changed. | |

**Functional Requirements: The system should provide users a GUI form to search for bugs of any type. Search criteria include: Application, ID, Description, Product, User, Developer.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Gui form to search for bugs of any type. | Test case 55 |
| Input | Enter the bug search system via the main window, enter a valid bug ID in the search text box. Select ID check box and click on Search button | |
| Expected output | The bug with the ID entered will display in the table on the main window. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Gui form to search for bugs of any type. | Test case 56 |
| Input | Enter an invalid bug ID, (ie one that is not in the database) into search text box via the main window. Select ID check box and click on Search button. | |
| Expected output | Nothing will be displayed in the main screen. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Gui form to search for bugs of any type. | Test case 57 |
| Input | Enter a description into Search text field, click Description check box then click Search button | |
| Expected output | The bugs with descriptions similar to the description entered will display in the table on the main window. If there is not similar description a message is displayed informing the user that a bug of this description does not exist. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Gui form to search for bugs of any type. | Test case 58 |
| Input | Enter Product/application name in search field. Select Search button | |
| Expected output | The bugs related to that product/application name will be displayed in the table on the main window. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Gui form to search for bugs of any type. | Test case 59 |
| Input | Enter an invalid Product/application name (ie one that is not in the database) in search field. Select Search button | |
| Expected output | Nothing will be displayed in the main screen. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Gui form to search for bugs of any type. | Test case 60 |
| Input | Enter the name of the user who reported the bug into search field. Select Search button | |
| Expected output | Bugs reported by that user will display in the bug table | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Gui form to search for bugs of any type. | Test case 61 |
| Input | Enter an invalid user name (ie one that is not in the database) in search field. Select Search button | |
| Expected output | Nothing will be displayed in the main screen. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Gui form to search for bugs of any type. | Test case 62 |
| Input | Enter the developer name assigned to the bug in the search text box. Click Search | |
| Expected output | Bugs assigned to this developer will display in the table on the main window. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Gui form to search for bugs of any type. | Test case 63 |
| Input | Enter an invalid developer name (ie one that is not in the database) in search field. Select Search button | |
| Expected output | Nothing will be displayed in the main screen. | |

**Functional Requirement: The system should allow a user to filter bugs of any type by Application, Developer and Status.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should allow a user to filter bugs of any type by Application, Developer and Status | Test case 64 |
| Input | Select a status from drop down list on the Filter Bugs panel on the Main Window. Select Search | |
| Expected output | Bugs with the selected status will display in the table on the Main Window. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should allow a user to filter bugs of any type by Application, Developer and Status | Test case 65 |
| Input | Select an application from drop down list on the Filter Bugs panel on the Main Window. Select Search | |
| Expected output | Bugs related to the selected application will display in the table on the Main Window. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should allow a user to filter bugs of any type by Application, Developer and Status | Test case 66 |
| Input | Select an item from each filter drop down list. Select Search | |
| Expected output | Bugs that include all the filter requirements will be displayed. | |

**Functional Requirement: The system should provide a Triager with a gui to edit an existing bug.**

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a triager with a gui to edit an existing bug. | Test case 67 |
| Input | Log in using a Triager’s user name and email Double click a bug in the bug table on the main window. Select edit. | |
| Expected output | View bug text fields become writable. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Triager with a gui to edit an existing bug. | Test case 68 |
| Input | Change the status, application name and remove a subscribed user from the CC list. Click Apply button | |
| Expected output | Message displays informing the Triager the bug has been modified. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Triager with a gui to edit an existing bug. | Test case 69 |
| Input | Search for the modified bug via the search bug facility on the main window using the bug ID number. Then double click it. | |
| Expected output | The modified bug should display with modified details. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Triager with a gui to edit an existing bug. | Test case 70 |
| Input | Click edit bug then select “Delete Bug” button | |
| Expected output | A message should display asking if you are sure you want to delete the bug. Choose ok. | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | The system should provide a Triager with a gui to edit an existing bug. | Test case 71 |
| Input | Search for the Bug via the main window Search Bug facility using it’s ID number. | |
| Expected output | Nothing will be displayed in the main screen. | |

**Non-Functional: The system will take no longer than 12 seconds to load**

|  |  |  |
| --- | --- | --- |
| requirement | The system will take no longer than 12 seconds to load | Test case 72 |
| Input | Click program executable or enter from login form within QT | |
| Expected output | Once log-in complete, the main window will display with all the bug data within 12 seconds. | |

**Non-Functional: The system should generate a report within no more than 3 seconds.**

|  |  |  |
| --- | --- | --- |
| requirement | The system will take no longer than 3 seconds to generate a report | Test case 73 |
| Input | Log in as manager, click report, generate report | |
| Expected output | Results will be displayed within 12 seconds. | |

**User Interface Test: Walk through of the main interfaces of the program to discover any faults in the design and make sure the program works as one.**

|  |  |  |
| --- | --- | --- |
| requirement | Walk through of the main interfaces of the program to discover any faults in the interface design and make sure the program works as one. | Test case 74 |
| Input | Open program and login as a reporter.  From main window open and close the following interfaces:  Add Bug  View Bug  Edit Profile  View Profile  Generate Report | |
| Expected output | * The interfaces should be consistent design with submit and cancel buttons placed in the same locations. * Forms look intuitive and easy to understand how to fill them out. * All interfaces available to the current logged in user should be easily accessible from the main window. | |

|  |  |  |
| --- | --- | --- |
| requirement | Walk through of the main interfaces of the program to discover any faults in the interface design and make sure the program works as one. | Test case 75 |
| Input | Open program and login as a Manager/Triager.  From main window open and close the following interfaces:  Assign Bug | |
| Expected output | * The interfaces should be consistent design with submit and cancel buttons placed in the same locations. * Forms look intuitive and easy to understand how to fill them out. * All interfaces available to the logged in user type should be easily accessible from the main window. | |

|  |  |  |
| --- | --- | --- |
| requirement | Walk through of the main interfaces of the program to discover any faults in the interface design and make sure the program works as one. | Test case 76 |
| Input | Open program and login as a Developer.  From main window open and close the following interfaces:  ChangeBugStatus | |
| Expected output | * The interfaces should be consistent design with submit and cancel buttons placed in the same locations. * Forms look intuitive and easy to understand how to fill them out. * All interfaces available to the logged in user type should be easily accessible from the main window. | |

|  |  |  |
| --- | --- | --- |
| requirement | Walk through of the main interfaces of the program to discover any faults in the interface design and make sure the program works as one. | Test case 77 |
| Input | Open program and login as an Administrator.  From main window open and close the following interfaces:  AddUser  Import bug/user  Export bug/user | |
| Expected output | * The interfaces should be consistent design with submit and cancel buttons placed in the same locations. * Forms look intuitive and easy to understand how to fill them out. * All interfaces available to the logged in user type should be easily accessible from the main window. | |

**Security and Access Control Testing: A user can only access those functions for which their user type is provided permissions**

|  |  |  |
| --- | --- | --- |
| requirement | A user can only access those functions for which their user type is provided permissions | Test case 78 |
| Input | Log-in as a Reporter and click add bug | |
| Expected output | * Access should be granted to this interface | |

|  |  |  |
| --- | --- | --- |
| requirement | A user can only access those functions for which their user type is provided permissions | Test case 79 |
| Input | Enter as a guest and click add Bug | |
| Expected output | * Access should be denied | |

|  |  |  |
| --- | --- | --- |
| requirement | A user can only access those functions for which their user type is provided permissions | Test case 80 |
| Input | Log-in as a Developer. Click edit Bug and try to change the status of the bug. | |
| Expected output | * Access is granted. | |

|  |  |  |
| --- | --- | --- |
| requirement | A user can only access those functions for which their user type is provided permissions | Test case 81 |
| Input | Log-in as a Reporter. Click edit Bug and try to change the status of the bug. | |
| Expected output | * Access is denied | |

|  |  |  |
| --- | --- | --- |
| requirement | A user can only access those functions for which their user type is provided permissions | Test case 82 |
| Input | Log-in as a Triager. Click AssignBug. | |
| Expected output | * Access is granted | |

|  |  |  |
| --- | --- | --- |
| requirement | A user can only access those functions for which their user type is provided permissions | Test case 83 |
| Input | Log-in as a Developer. Click AssignBug. | |
| Expected output | * Access is denied | |

|  |  |  |
| --- | --- | --- |
| requirement | A user can only access those functions for which their user type is provided permissions | Test case 84 |
| Input | Log-in as Reviewer. Click AssignBug. | |
| Expected output | * Access is denied | |

|  |  |  |
| --- | --- | --- |
| requirement | A user can only access those functions for which their user type is provided permissions | Test case 85 |
| Input | Log-in as Reviewer. Click Verify bug. | |
| Expected output | * Access is granted | |

**Unit test: Test functions to make sure they can work**

|  |  |  |
| --- | --- | --- |
| Functional requirement | Test GUI of AddBug | Test case 86 |
| Input | Run a QLineEdit object with Hello World String | |
| Expected output | Pass the Qt test | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | Test GUI of Editprofile | Test case 87 |
| Input | Run a QLineEdit object with Hello World String | |
| Expected output | Pass the Qt test | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | Test GUI of login | Test case 88 |
| Input | Run a QLineEdit object with Hello World String | |
| Expected output | Pass the Qt test | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | Test GUI of Register | Test case 89 |
| Input | Run a QLineEdit object with Hello World String | |
| Expected output | Pass the Qt test | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | Test GUI of viewbug | Test case 90 |
| Input | Run a QLineEdit object with Hello World String | |
| Expected output | Pass the Qt test | |

|  |  |  |
| --- | --- | --- |
| Functional requirement | Test function of AddBug add() | Test case 91 |
| Input | Input QString into the function and check the retrun value. | |
| Expected output | Pass the Qt test | |