**Publication Management**

Test report

Version *<3.0>*

*22/04/2018*

VERSION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Brief Description** |
| 1.0 | Qazi Sajid Azam,  Rahul Jindal | 03/04/2018 | *Publication management for journal papers.* |
| 2.0 | K. Rajasekhar,  N. Mourya Mithra | 10/04/2018 | *Cross testing* |
| 3.0 | Qazi Sajid Azam,  Rahul Jindal | 22/04/2018 | *Final test report after enhancement* |

**Table of Contents**

[1.0 Introduction](#_fa8zsp6uwk0s) 3

[2.0 Test Plan](#_tyjcwt) 3

3.0 [Details of unit testing](#_3dy6vkm) 3

1. Main 3

2. Login Screen 3

3. User UI 4

4. Admin UI 5

5. User level search 6

6.System level search 7

7. Admin level edit 7

8. System level save 8

9. User level statistics generator 9

10. System level database loader 9

11. Admin login 10

12. Admin logout 10

13. User level news 11

14. System level statistics generator 11

4.0 D[etails of System Testing](#_1ksv4uv) 12

5.0 [Unresolved test instances](#_z337ya) 12

# 1.0 Introduction

Purpose: The purpose of making a test report is to help in finding defects which may have occurred during the development phase. This is indispensable if we want to deliver an error-free product and also provides a means to ensure that our software contains all the functionalities that were stated in the SRS document.

# 2.0 Test Plan

This test report contains the unit test plan and system test plan.

# Details of unit testing

# Main function

## Test Items: main()

## Features To Be Tested: Program should start execution

## Item Pass/Fail Criteria: Test is passed if it successfully starts execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.1.1.1 | none | running | running | PASS |

**All test cases verified.**

* 1. Login screen

## Test Items: start()

## Features To Be Tested: It should print a welcome message, load the data and then ask for login until the program is exited. If login is successful, it shows options for admin otherwise asks user to retry login. If user chooses not to login, it shows the options for a guest user.

## Item Pass/Fail Criteria: Test is passed if it shows the welcome screen and login options, and then calls the appropriate functions for guest or admin UI.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.2.1.1 | ‘y’, “admin1”, “pass1” | Called adminOptions() | Called adminOptions() | PASS |
| 3.2.1.2 | ‘n’ | Called userOptions() | Called userOptions() | PASS |
| 3.2.1.3 | ‘y’, “wrongname”, “pass” | “Login failed”.  retry | “Login failed”.  retry | PASS |
| 3.2.1.4 | ‘y’, “admin”, “wrongpass” | “Login failed”.  retry | “Login failed”.  retry | PASS |

**All test cases verified.**

* 1. User UI function

## Test Items: userOptions()

## Features To Be Tested: Should display menu of options available to guest user and let them execute these options repeatedly until logout.

## Item Pass/Fail Criteria: Test is passed if menu is displayed and appropriate action is taken for each selection in the menu.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.3.1.1 | none | Menu shown | Menu shown | PASS |
| 3.3.1.2 | 1 | Called usearch() | Called usearch() | PASS |
| 3.3.1.3 | 2 | Called uStat() | Called ustat() | PASS |
| 3.3.1.4 | 3 | Called uNews() | Called uNews() | PASS |
| 3.3.1.5 | 4 | Return to login menu | Return to login menu | PASS |
| 3.3.1.6 | 8 | “Invalid response” | “Invalid response” | PASS |
| 3.3.1.7 | a | “Invalid response” | “Invalid response” | PASS |

**All test cases verified.**

# Admin UI function

## Test Items: adminOptions()

## Features To Be Tested: Should display menu of options available to admin user if they are logged in and let them execute these options repeatedly until logout. If they are not logged in, return to login menu.

## Item Pass/Fail Criteria: Test is passed if menu is displayed and appropriate action is taken for each selection in the menu.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.4.1.1 | none | Menu shown | Menu shown | PASS |
| 3.4.1.2 | 1 | Called usearch() | Called usearch() | PASS |
| 3.4.1.3 | 2 | Called uStat() | Called ustat() | PASS |
| 3.4.1.4 | 3 | Called uNews() | Called uNews() | PASS |
| 3.4.1.5 | 4 | Called edit() | Called edit() | PASS |
| 3.4.1.6 | 5 | Called saveData() | Called saveData() | PASS |
| 3.4.1.7 | 6 | Called rollBack() | called rollBack() | PASS |
| 3.4.1.8 | 7 | Return to login | Return to login | PASS |
| 3.4.1.9 | 8 | “Invalid response” | “Invalid response” | PASS |
| 3.4.1.10 | local login=0 | “Not logged in”  Return to login | “Not logged in”  Return to login | PASS |

**All test cases verified.**

# User-Level Search function

## Test Items: usearch()

## Features To Be Tested: Should show the filters menu by which the search can be performed. Then if the response is valid, it should take the keyword for search and pass it to the system search function.

## Item Pass/Fail Criteria: Test is passed if menu is displayed, filter type and keyword are input and the system search function is called.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.5.1.1 | none | Menu shown | Menu shown | PASS |
| 3.5.1.2 | 1, string | Called searchdata (string, 1) | Called searchdata (string, 1) | PASS |
| 3.5.1.3 | 2, string | Called searchdata (string, 2) | Called searchdata (string, 2) | PASS |
| 3.5.1.4 | 3, string | Called searchdata (string, 3) | Called searchdata (string, 3) | PASS |
| 3.5.1.5 | 4, string | Called searchdata (string, 4) | Called searchdata (string, 4) | PASS |
| 3.5.1.6 | 5, string | Called searchdata (string, 5) | Called searchdata (string, 5) | PASS |
| 3.5.1.7 | 6 | Return to previous | Return to previous | PASS |
| 3.5.1.8 | 8 | “Invalid response”  return to options menu | “Invalid response”  return to options menu | PASS |

**All test cases verified.**

# System-Level Search function

## Test Items: searchdata(string key, int type)

## Features To Be Tested: Should print the output table heading and call the correct specific search function according to the type variable.

## Item Pass/Fail Criteria: Test is passed if the correct specific search function is called.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.6.1.2 | string, 1 | Called searchname (string) | Called searchname (string) | PASS |
| 3.6.1.3 | string, 2 | Called searchauth (string) | Called searchname (string) | PASS |
| 3.6.1.4 | string, 3 | Called searchjournal (string) | Called searchjournal (string) | PASS |
| 3.6.1.5 | string, 4 | Called searchdoi (string) | Called searchdoi (string) | PASS |
| 3.6.1.5 | string, 5 | Called searchyear (string) | Called searchyear (string) | PASS |

**All test cases verified.**

# Admin-level Edit function

## Test Items: edit()

## Features To Be Tested: Should display a menu of the types of edit an admin can make. If the choice is valid, then take the corresponding input and pass it to the system-level specific edit functions.

## Item Pass/Fail Criteria: Test is passed if menu is shown and the input is taken and passed to the specific edit functions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.7.1.1 | none | Menu shown | Menu shown | PASS |
| 3.7.1.2 | 1, string | Called addPaper(string)  "Records updated" | Called addPaper(string)  "Records updated" | PASS |
| 3.7.1.3 | 2, int, y | Called deletePaper(int)  "Records updated" | Called deletePaper(x)  "Records updated" | PASS |
| 3.7.1.4 | 3, int, y, string | Called changePaper (string,int)  "Records updated" | Called changePaper (string,x)  "Records updated" | PASS |
| 3.7.1.5 | 2, wrong | “Index not found” | “Index not found” | PASS |
| 3.7.1.6 | 3, wrong | “Index not found” | “Index not found” | PASS |
| 3.7.1.7 | 4 | Return to options menu | Return to options menu | PASS |
| 3.7.1.7 | 5 | “Invalid response”  Return to options menu | “Invalid response”  Return to options menu | PASS |

**All test cases verified.**

# System-level Save function

## Test Items: saveData()

## Features To Be Tested: Write the data in running memory to the stored file.

## Item Pass/Fail Criteria: Test is passed if the database gets saved.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.8.1.1 | none | "Database saved" | "Database saved" | PASS |

**All test cases verified.**

# User-level Statistics Generator function

## Test Items: uStat()

## Features To Be Tested: Should display menu of grouping types for statistics, then pass the grouping type to system-level stats generator function.

## Item Pass/Fail Criteria: Test is passed if the system-level stats generator function is called.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.9.1.1 | int | Called getStat(int) | Called getStat(int) | PASS |
| 3.9.1.2 | not int | Called getStat(0) | Called getStat(int) | PASS |

**All test cases verified.**

# System-level Database Loader function

## Test Items: loadData()

## Features To Be Tested: Should display message confirming data load process.

## Item Pass/Fail Criteria: Test is passed if the data loading message is displayed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.10.1.1 | none | “Loading data…” | “Loading data…” | PASS |

**All test cases verified.**

# Admin Login function

## Test Items: adminLogin (string,string)

## Features To Be Tested: Should try to login with all elements of admin list using username and password. If any login is successful, return index of admin that is logged in. Otherwise print message that login failed.

## Item Pass/Fail Criteria: Test is passed if the login status is displayed and index returned.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.11.1.1 | username, password | “Logged in as: username” | “Logged in as: username” | PASS |
| 3.11.1.2 | wrong, password | “Login Failed” | “Login Failed” | PASS |
| 3.11.1.3 | username, wrong | “Login Failed” | “Login Failed” | PASS |
| 3.11.1.4 | wrong, wrong | “Login Failed” | “Login Failed” | PASS |

**All test cases verified.**

# Admin Logout function

## Test Items: logout ()

## Features To Be Tested: Database is saved or not saved based on user input and logout is done.

## Item Pass/Fail Criteria: Test is passed if the login is set to false and database is saved or paperlist is reloaded depending on user input.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.12.1.1 | ‘y’ | Called saveData()  login=false | Called saveData()  login=false | PASS |
| 3.12.1.2 | ‘n’ | Called loadData()  login=false | Called loadData()  login=false | PASS |

**All test cases verified.**

# User-level News function

## Test Items: uNews()

## Features To Be Tested: Should call the system-level news function.

## Item Pass/Fail Criteria: Test is passed if printNews() function is called.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.14.1.1 | none | called printNews() | called printNews() | PASS |

**All test cases verified.**

# System-level Statistics Generator function

## Test Items: getStat(int)

## Features To Be Tested: Display statistics based on value passed into it.

## Item Pass/Fail Criteria: Test is passed if correct type of stats are displayed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Input** | **Expected** | **Actual** | **Pass/Fail** |
| 3.14.1.1 | 1 | Display author stats | Display author stats | PASS |
| 3.14.1.2 | 2 | Display year stats | Display year stats | PASS |
| 3.14.1.3 | 0 | “Invalid Response” | “Invalid Response" | PASS |

**All test cases verified.**

# details of system testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Test step** | **Expected** | **Actual** | **Pass/Fail** |
| 1 | Login as Guest->Search | Search menu shown | Search menu shown | PASS |
| 2 | Login as Guest->Stats | Stats menu shown | Stats menu shown | PASS |
| 3 | Login as Guest->News | News shown | News shown | PASS |
| 4 | Login as Admin->Edit | Edit menu shown | Edit menu shown | PASS |
| 5 | Login as Admin->Logout | Returned to login screen | Returned to login screen | PASS |
| 6 | Login as Admin->Search | Search menu shown | Search menu shown | PASS |
| 7 | Login as Admin->Stats | Stats menu shown | Stats menu shown | PASS |
| 8 | Login as Admin->News | News shown | News shown | PASS |
| 9 | Login as Admin->Save Data | Database saved | Database saved | PASS |
| 10 | Load Data | Database loaded | Database loade | PASS |

**All test cases verified.**

**News is shown for Login as Admin->News shows incorrect details like some times it shows double details which implies already deleted data is again deleted which not possible.**

**News is shown for Login as Guest->News shows incorrect details like some times it shows double details which implies already deleted data is again deleted which not possible.**

**Above two test results are essentially the same as News function is same for both types of users.**

**Moreover, we have tested these results before giving the project for cross testing and already verified that this function works properly. The above error may have occurred as a result of carelessness on the part of the cross testing team.**

# Unresolved test instances

|  |  |  |
| --- | --- | --- |
| Test Case ID | Summary of Defect | Why not resolved |
| *<none>* |  |  |