|  |  |
| --- | --- |
| **Project Name: Project 2: Voting System Agile Scrum Team#25** | |
| **Test Stage: Unit \_\_ System ✓** | **Test Date: 12/12/2018** |
| **Test Case ID#: Extreme\_test\_3** | **Name(s) of Testers: Yizhe Wang** |
| **Test Description:**  **This system test is designed to run the testinvalidcheck3.csv from the directory /Project2/testing/extreme/** **testinvalidcheck3.csv**  **I will use the main application to test the testinvalidcheck3.csv** |  |
| **Automated: yes\_\_\_ no ✓** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.** |
| **Results: Pass ✓ Fail\_\_\_\_\_\_\_\_** |  |
|  |  |
| **Preconditions for Test:**  **The system that is used for testing must be CSELabs machines and user successfully access into the application system.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Go to the directory /Project2/src | null | null | null | The first step will be going to the application directory |
| 2 | compile all file with javac \*.java | null | null | null | The second step will be compiling all existing java file |
| 3 | do java Tester\_main | null | null | null | The third step will be open the application |
| 4 | Login using 3WUser as Username and 5801 as password | null | null | null | Log in into the application |
| 5 | Click the button “Open an input file” and select the test file which is located /Project2/testing/systemtesting/extreme/ testinvalidcheck3.csv | Testinvalidcheck3.csv | Winner: Rosen (D)  And the  corresponding invalid ballot collection file appears | unkown | Select the file |
| 6 | Click start and there will be four button available.  1.Export Audit file  2.On screen result  3.Export Media file  4.Back | Testinvalidcheck3.csv | Winner: Rosen (D) | unkown | At this stage, tester could select four button. Further result will be shown in the following steps |
|  |  |  | And the  corresponding invalid ballot collection file appears |  | This is the result of the audit file |
| 7 | Click Export Audit file button | Testinvalidcheck3.csv | Winner: Rosen (D)  And the  corresponding invalid ballot collection file appears | CandidatesName: Rosen (D) Kleinberg (R) Chou (I) Royce(L)  FirstChoice: 2 0 0 0  Winner: Rosen (D)  Found file in the directory | You could find the audit file in the src directory (or your current working directory) |
| 8 | Click On Screen button | Testinvalidcheck3.csv | Winner: Rosen (D) | CandidatesName: Rosen (D) Kleinberg (R) Chou (I) Royce(L)  FirstChoice: 2 0 0 0  Winner: Rosen (D) | This is the result of the on screen file |
| 9 | Click Export Media file Button | Testinvalidcheck3.csv | Winner: Rosen (D)  And the  corresponding invalid ballot collection file appears | CandidatesName: Rosen (D) Kleinberg (R) Chou (I) Royce(L)  FirstChoice: 2 0 0 0  Winner: Rosen (D)  Found file in the directory | You could find the media file in the src directory (or your current working directory) |



**Post condition(s) for Test:**



After you click any of the export file button, the corresponding file which include the audit or media file will be generate into the src directory. The page will remain unchanged. And if you want to go back to load another test file or actual ballot file, there is a back button that tester could click. If this file contain the invalid ballot, you could also find them in the src directory (or your current directory).



**Project Name:**  The project #, name of your system, and the team#

**Test Stage:** Indicate whether it is a unit test or a system test.

**Test Date:**  The date the test was performed.

**Test Case ID#:**  A unique ID is required. Decide on a naming convention and use numbering. Example: Ballot\_Shuffle\_1

**Name(s) of Testers:** List the names of anyone involved in running this test case.

**Test Description:**  Describe briefly the test objective.

**Automated:**  Indicate if the test is completely automated or being checked manually. (If you have methods running the tests and checking results, select “yes”. If you are manually checking results, indicate manual by selecting the “no.”)

**Results:** Indicate if the test passed or failed.

**Step #:** You will be listing the test steps in order. This number is the step number in the process.

**Test Step Description:** Details of the test step.

**Test Data:** What the test data will be for this step. Be clear on what the input data will be. If using a specific file, be clear on the name.

**Expected Result:** What result are you expecting from the program component or system.

**Actual Result:** What result were returned based on the test.

**Post condition for Test:** What will be true after the test has been run? Has the state of the system changed in any way?

**Notes:** Comments and notesfor you and your team members.