|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team#10** | |
| **Test Stage: Unit \_\_ System \_\_** | **Test Date: 11/17/2019** |
| **Test Case ID#: A**udit**\_01** | **Name(s) of Testers: Amber Wong** |
| **Test Description:**  This test serves to verify that the AuditFile class’s constructor properly initializes in the PLVS’s. |  |
| **Automated: yes\_\_\_ no \_\_\_** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  The src/Audit.java class was tested using the src/TestAudit.java class, specifically the main() method. |
| **Results: Pass \_\_ X \_\_ Fail\_\_\_\_\_\_\_\_** |  |
|  |  |
| **Preconditions for Test:** There is a defined instance of Auditfile named “af”. f is not a constant and its current value is null. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
|  |  |  |  |  |  |
| 1 | Call af =new AuditFile() | af==null | False | False | This confirms that af was initialized but does not confirm af’s controls were initialized. |
| 2 | Check the af.path = the path we have | af.path = “/User/….” | True | True | We can change the path |

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

“af”, the new Auditput file, is initialized to an empty file along with all of its methods.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team#10** | |
| **Test Stage: Unit X\_ System \_\_** | **Test Date: 11/17/2019** |
| **Test Case ID#: Auditput \_02** | **Name(s) of Testers: Amber Wong** |
| **Test Description:**  This test serves to verify that the AuditFile class’s WriteHeader() method written in the created audit file. Note this is not visible in the terminal, it can be seen from the file. |  |
| **Automated: yes\_\_\_ no X \_\_\_** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  The src/AuditFile.java class was tested using the src/TestAudit.java class, specifically the WriteHeader() method. |
| **Results: Pass \_\_X\_\_\_ Fail\_\_\_\_\_\_\_\_** |  |
|  |  |
| **Preconditions for Test:** There is a defined instance of GUI named “g”. g has been initialized through its constructor. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | af.WriteHeader() | Null | The writeln window should write the "~~~~ Audit File ~~~~"to the outputfile. | The writeln window should write the "~~~~ Audit File ~~~~"to the outputfile. |  |

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

This WriteHeader() method write the header in every Auditfile “~~~~ Audit File ~~~~”

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team#10** | |
| **Test Stage: Unit X\_ System \_\_** | **Test Date: 11/17/2019** |
| **Test Case ID#: Auditput \_03** | **Name(s) of Testers: Amber Wong** |
| **Test Description:**  This test serves to verify that the AuditFile class’s WriteVote(string party) / WriteVote(string party, string candidate) method written in the created audit file. Note this is not visible in the terminal, it can be seen from the file. |  |
| **Automated: yes\_\_\_ no X \_\_\_** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  The src/AuditFile.java class was tested using the src/TestAudit.java class, specifically the WriteVote(string party) / WriteVote(string party, string candidate) method. |
| **Results: Pass \_\_X\_\_\_ Fail\_\_\_\_\_\_\_\_** |  |
|  |  |
| **Preconditions for Test:** There is a defined instance of GUI named “g”. g has been initialized through its constructor. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | af.WriteVote("D") | “D” | The writeln window should write the “Vote for D"to the outputfile. | The writeln window should write the " Vote for D"to the outputfile. |  |
| 2 | af.WriteVote("R") | “R” | The writeln window should write the “Vote for R"to the outputfile. | The writeln window should write the " Vote for R"to the outputfile. |
| 3 | af.WriteVote("G") | “G” | The writeln window should write the “Vote for G"to the outputfile. | The writeln window should write the " Vote for G"to the outputfile. |
| 4 | af.WriteVote("I") | “I” | The writeln window should write the “Vote for I"to the outputfile. | The writeln window should write the " Vote for I"to the outputfile. |
| 5 | af.WriteVote("D","Foster") | "D","Foster" | The writeln window should write the “Vote for Foster (D)"to the outputfile. | The writeln window should write the " Vote for Foster (D)"to the outputfile. |
| 6 | af.WriteVote("R","Deutsch") | "R","Deutsch" | The writeln window should write the “Vote for Deutsch (R)"to the outputfile. | The writeln window should write the " Vote for Deutsch (R)"to the outputfile. |
| 7 | af.WriteVote("G","Smith") | "G","Smith" | The writeln window should write the “Vote for Smith (G)"to the outputfile. | The writeln window should write the " Vote for Smith (G)"to the outputfile. |
| 8 | af.WriteVote("I","Perez") | "I","Perez" | The writeln window should write the “Vote for Perez (I)"to the outputfile. | The writeln window should write the " Vote for Perez (I)"to the outputfile. |

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

This WriteVote() method has two type, one only with party will write “Vote for …” and the one with party and candidates will write the ““Vote for candidate(party…)” in every Auditfile .

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.