

Project Name: Project 1: Voting System Software

Team 4

Test Stage: Unit 1 System 1

Test Date: 26 March 2023

Test Case ID#: TESTCASE_1

**Name(s) of Testers: Bryan Yen Sheng Lee
Cedric Tan Yee Shuen
Sherryl Ooi Shi Tyng**

Test Description: All the unit tests for fileSystem.java is written inside the fileSystemTest.java file where the fileSystemTest.java file is then stored under /repo-Team4/Project1/src/. The unit tests check the functionality of all the method / functions in the fileSystem.java. The method / functions being used to test for unit testing are as follows:

- (1) public fileSystem() - class constructor
- (2) public static void openFile(File filename)
- (3) public static boolean checkFileFormat (File filename)
- (4) public static void readFile (File filename)
- (5) public static void getFileExtension(File fullname)

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes X no ____

Results: Pass 15 Fail 1

Preconditions for Test:

PC1: The system is started in the normal mode.

PC2: The file is successfully being brought into the system and is in the same directory as the program.

PC3: The file is successfully opened by the system.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	The test1_getFileExtension() checks whether getFileExtension() returns the correct file extension.	testfile.csv	csv	csv	Success
2	The test2_getFileExtension() checks whether getFileExtension() returns the correct file extension.	testfile.pdf	pdf	pdf	Success
3	The test3_getFileExtension() checks whether getFileExtension() returns the correct file extension.	testfiledocs	""	""	Success
4	The test4_getFileExtension() checks whether getFileExtension() returns the correct file extension.	""	""	""	Success
5	The test5_getFileExtension() checks whether getFileExtension() returns the correct file extension.	testfile.csv.csv	csv	csv.csv	Fail
6	The test6_getFileExtension() checks whether getFileExtension() returns the correct file extension.	testfile..	""	""	Success
7	The test7_checkFileFormat() checks whether checkFileFormat() returns boolean indicating correct file format.	testfile.csv	true	true	Success
8	The test8_checkFileFormat() checks	testfile.pdf	false	false	Success

	whether checkFileFormat() returns boolean indicating correct file format.				
9	The test9_checkFileFormat() checks whether checkFileFormat() returns boolean indicating correct file format.	testfiledocs	false	false	Success
10	The test10_checkFileFormat() checks whether checkFileFormat() returns boolean indicating correct file format.	testfile..	false	false	Success
11	The test11_checkFileFormat() checks whether checkFileFormat() returns boolean indicating correct file format.	testfile.csv.csv	true	true	Success
12	The test12_readFile() checks whether readFile() returns correct information about the election type, number of candidates , and list of candidates.	Project1/testing/ CPL_18-3-2023. csv	CPL 6 new String[]{"Democra tic", "Republican", "New Wave", "Reform", "Green", "Independent"} new String[]{"Foster, Volz, Pike", "Green, Xu, Wang", "Jacks, Rosen", "McClure, Berg", "Zheng, Melvin", "Peters"} 3	CPL 6 new String[]{"Democra tic", "Republican", "New Wave", "Reform", "Green", "Independent"} new String[]{"Foster, Volz, Pike", "Green, Xu, Wang", "Jacks, Rosen", "McClure, Berg", "Zheng, Melvin", "Peters"} 3	Success

			11	11	
13	The test13_readFile() checks whether readFile() returns correct information about the election type, number of candidates , and list of candidates.	Project1/testing/I R_2-3-2023.csv	IR 4 new String []{"Rosen(D), "Kleinberg(R), Chou(I), Royce(L)"} 6	IR 4 new String []{"Rosen(D), "Kleinberg(R), Chou(I), Royce(L)"} 6	Success
14	The test14_openFile() checks whether openFile() returns correct information about the election type, number of candidates , and list of candidates.	Project1/testing/ CPL_18-3-2023. csv	true true	true true	Success
15	The test15_openFile() checks whether openFile() returns correct information about the election type, number of candidates, and list of candidates.	Project1/testing/ CPL_18-3-2023. pdf	false	false	Success

Post condition(s) for Test:

1. The file is identified and the type of election is determined by the system.
-

Project Name: Project 1: Voting System Software

Team 4

Test Stage: Unit 2 System 1

Test Date: 26 March 2023

Test Case ID#: TESTCASE_2

**Name(s) of Testers: Bryan Yen Sheng Lee
Cedric Tan Yee Shuen
Sherryl Ooi Shi Tyng**

Test Description: All the unit tests for countBallot.java is written inside the countBallotTest.java file where the countBallotTest.java file is then stored under /repo-Team4/Project1/src/. The unit tests check the functionality of all the method/functions in the countBallot.java. The method/functions being used for unit testing are follows:
(1) public countBallot() - class constructor

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes X no ____

Results: Pass 0 Fail 3

Preconditions for Test:

PC1: The system is started in the normal mode.

PC2: The file is successfully being brought into the system and is in the same directory as the program.

PC3: The file is successfully opened by the system.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
--------	-----------------------	-----------	-----------------	---------------	-------

1	The test1_countBallot() checks whether countBallot() returns correct total ballots information for CPL.	fileSystem.electionType, fileSystem.numOfCandidates, fileSystem.numOfVotes, fileSystem.ballot	CPL 3 5 new int[][] {{1,2,3}, {1,2,3}, {2,1,3}, {2,3,1}, {3,2,1}} new int[][] {{2,3},{2,2},{1 ,3}}	CPL 3 5 new int[][] {{1,2,3}, {1,2,3}, {2,1,3}, {2,3,1}, {3,2,1}} new int[][] {{2,3},{2,2},{ 1,3}}	Fail
2	The test2_countBallot() checks whether countBallot() returns correct total ballots information for IR.	fileSystem.electionType, fileSystem.numOfCandidates, fileSystem.numOfVotes, fileSystem.ballot	IR 3 5 new int[][] {{1,2,3}, {1,2,3}, {2,1,3}, {2,3,1}, {3,2,1}} new int[][] {{2,0,0,0},{2,2, 1,0},{1,3,4,1}}	IR 3 5 new int[][] {{1,2,3}, {1,2,3}, {2,1,3}, {2,3,1}, {3,2,1}} new int[][] {{2,0,0,0},{2, 2,1,0},{1,3,4,1}}	Fail
3	The test3_countBallot() checks whether countBallot() returns correct total ballots information for invalid election type.	fileSystem.electionType, fileSystem.numOfCandidates, fileSystem.numOfVotes, fileSystem.ballot	“This election type is not recognized.”	“This election type is not recognized.”	Fail

Post condition(s) for Test:

1. The election is ready to be run based on the election type.
-

Project Name: Project 1: Voting System Software

Team 4

Test Stage: Unit 3 System 1

Test Date: 26 March 2023

Test Case ID#: TESTCASE_3

**Name(s) of Testers: Bryan Yen Sheng Lee
Cedric Tan Yee Shuen
Sherryl Ooi Shi Tyng**

Test Description: All the unit tests for finalRanking.java is written inside the finalRankingTest.java file where the finalRankingTest.java file is then stored under /repo-Team4/Project1/src/. The unit tests check the functionality of all the method/functions in the finalRanking.java. The method/functions being used for unit testing are as follows:

- (1) public finalRanking() - class constructor
- (2) public boolean checkForTie()
- (3) public int fairCoinToss()
- (4) public int poolCoinToss()

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes X no ____

Results: Pass 0 Fail 9

Preconditions for Test:

1. The election is being run based on the election type.
2. The input must be valid.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	The test1_checkForTie() checks whether checkForTie() performs a fair coin toss if the election type is CPL.	fileSystem.electionType, fileSystem.numOfSeats, rankings.ranking	false	false	Fail
2	The test2_checkForTie() checks whether checkForTie() performs a pool coin toss if the election type is IR.	fileSystem.electionType, fileSystem.numOfSeats, rankings.ranking	true	true	Fail
3	The test3_checkForTie() checks whether checkForTie() performs a coin toss if the election type is invalid.	fileSystem.electionType, fileSystem.numOfSeats, rankings.ranking	false	false	Fail
4	The test4_checkForTie() checks whether checkForTie() performs a coin toss is necessary if the election type is IR.	fileSystem.electionType, fileSystem.numOfSeats, rankings.ranking	false	false	Fail

5	The test5_fairCoinToss() checks whether fairCoinToss() performs a coin toss if necessary for the election type CPL.	fileSystem.electionType, fileSystem.numOfSeats	Random number 0 or 1	Random number 0 or 1	Fail
6	The test6_fairCoinToss() checks whether fairCoinToss() performs a coin toss if necessary for the election type IR.	fileSystem.electionType	Random number 0 or 1	Random number 0 or 1	Fail
7	The test7_poolCoinToss() checks whether poolCoinToss() performs a coin toss if necessary for the election type CPL.	fileSystem.electionType, fileSystem.numOfSeats	Random number from 0 to 2	Random number from 0 to 2	Fail
8	The test8_poolCoinToss() checks whether poolCoinToss() performs a coin toss if necessary for the election type IR.	fileSystem.electionType	Random number from 0 to 4	Random number from 0 to 4	Fail
9	The test9_poolCoinToss() checks whether poolCoinToss() performs a coin toss if necessary for an invalid election type.	fileSystem.electionType	0	0	Fail

Post condition(s) for Test:

1. The winner is declared in an unbiased manner.
-

Project Name: Project 1: Voting System Software

Team 4

Test Stage: Unit 4 System 1

Test Date: 26 March 2023

Test Case ID#: TESTCASE_4

**Name(s) of Testers: Bryan Yen Sheng Lee
Cedric Tan Yee Shuen
Sherryl Ooi Shi Tyng**

Test Description: All the unit tests for rankings.java is written inside the rankingsTest.java file where the rankingsTest.java file is then stored under /repo-Team4/Project1/src/. The unit tests check the functionality of all the method/functions in the rankings.java. The method/functions being used for unit testing are as follows:

- (1) public rankings() - class constructor
- (2) public int [] checkRanking (int [][] newBallots, int initialRank)
- (3) public void checkMajority (int [][] totalBallots)

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes X no ____

Results: Pass 0 Fail 2

Preconditions for Test:

PC1: When the system runs either Instant Runoff Voting or Closed Party List Voting.
PC2: When a CSV file is processed for either Instant Runoff or Closed Party List.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	The test1_checkRankingsTest() checks whether checkRanking() sorts each candidate based on votes.	fileSystem.electionType, fileSystem.numOfCandidates, fileSystem.numOfVotes, fileSystem.totalBallots	new int[] {2,1,0}	new int[] {2,1,0}	Fail
2	The test2_checkMajority() checks whether checkMajority() sorts each candidate based on votes.	fileSystem.electionType, fileSystem.numOfCandidates, fileSystem.numOfVotes, fileSystem.totalBallots	new int[] {0,1,2}	new int[] {0,1,2}	Fail

Post condition(s) for Test:

1. The rankings of each candidate is determined.
 2. The winner(s) are being calculated based on the number of seats and the order of the candidates.
-

Project Name: Project 1: Voting System Software

Team 4

Test Stage: Unit 5 System 1

Test Date: 26 March 2023

Test Case ID#: TESTCASE_5

**Name(s) of Testers: Bryan Yen Sheng Lee
Cedric Tan Yee Shuen
Sherryl Ooi Shi Tyng**

Test Description: All the unit tests for displayResults.java is written inside the displayResultsTest.java file where the displayResultsTest.java file is then stored under /repo-Team4/Project1/src/. The unit tests check the functionality of all the method/functions in the rankings.java. The method/functions being used for unit testing are as follows:

- (1) public displayResults() - class constructor
- (2) public void generateAuditFile ()
- (3) public void showResults ()

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes X no ____

Results: Pass 0 Fail 2

Preconditions for Test:

PC1: The rankings of each candidate is determined.

PC2: The winner(s) are being calculated based on the number of seats and the order of the candidates.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	The test1_generateAuditFile() checks whether generateAuditFile() exports an audit file.	fileSystem.electionType, fileSystem.numOfCandidates, fileSystem.allCandidates, fileSystem.candidates, fileSystem.numOfSeats, fileSystem.numOfVotes, rankings.ranking	true true true	true true true	Fail
2	The test2_showResults() checks whether showResults() display the information correctly..	fileSystem.electionType, fileSystem.numOfCandidates, fileSystem.numOfVotes, fileSystem.totalBallots	expectedMessage	expectedMessage	Fail

Post condition(s) for Test:

1. Audit File is being downloaded.
 2. Results based on each election type are displayed.
-