|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 9** | |
| **Test Stage: Unit System** | **Test Date: 2019-11-18** |
| **Test Case ID#: 1** | **Name(s) of Testers:**  *Kai Wang, wang8739*  *Jingfan Guo, guo00109*  *Ruoyan Kong, kong0135*  *Yuan Yao, yaoxx340* |
| **Test Description:**  Test the units (Menu, Candidate, Party, Statistics, Election, Audit) of voting system. |  |
| **Automated: yes** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  **./testing**  **can be run by ./run\_unit\_test.sh** |
| **Results: Pass** |  |
|  |  |
| **Preconditions for Test: jdk 11, Junit4** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Test Menu – runElection | String runElection\_command = "runElection";  String Setup\_command = "setup";  String file\_path = "./sampleBallot1.txt"; | "Please setup your election first." | "Please setup your election first." |  |
| 2 | Test Candidate – constructor | String partyID = "A";  public String name = "John";  int ranks = 1; | PartyID = "A" | "A" |  |
| 3 | Test Party – constructor | Candidate cand1 = new Candidate("A", "John", 1);  Candidate cand2 = new Candidate("A", "Mac", 2);  public CopyOnWriteArrayList<Candidate> candidateList = new CopyOnWriteArrayList<Candidate>();  public String ID = "A"; | party.candidateList.get(0).name = "Mac" | “Mac” |  |
| 4 | Test Statistics - voteForParty | format = "CPL";  seatNumber = 7;  ballotNumber = 100;  Map<String, List<String>> parties = new HashMap<>() {{  put("D", Arrays.asList("Pike", "Foster", "Floyd", "Jones", "Mallory"));  put("R", Arrays.asList("Deutsch", "Wong", "Walters", "Keller", "Borg"));  put("G", Arrays.asList("Jones", "Smith", "Lewis", "Smith", "Li"));  put("I", Arrays.asList("Perez"));  }}; | When voteForParty(p) is called, the counter for Party p is incremented by 1 | When voteForParty(p) is called, the counter for Party p is incremented by 1 |  |
| 5 | Test Statistics – voteForCandidate | format = "OPL";  seatNumber = 7;  ballotNumber = 100;  Map<String, List<String>> parties = new HashMap<>() {{  put("D", Arrays.asList("Pike", "Foster", "Floyd", "Jones", "Mallory"));  put("R", Arrays.asList("Deutsch", "Wong", "Walters", "Keller", "Borg"));  put("G", Arrays.asList("Jones", "Smith", "Lewis", "Smith", "Li"));  put("I", Arrays.asList("Perez"));  }}; | When voteForCandidate (c) is called, the counter for Candidate c is incremented by 1 | When voteForCandidate (c) is called, the counter for Candidate c is incremented by 1 |  |
| 6 | Test Statistics - testGenerateAuditFile | format = "CPL";  seatNumber = 7;  ballotNumber = 100;  Map<String, List<String>> parties = new HashMap<>() {{  put("D", Arrays.asList("Pike", "Foster", "Floyd", "Jones", "Mallory"));  put("R", Arrays.asList("Deutsch", "Wong", "Walters", "Keller", "Borg"));  put("G", Arrays.asList("Jones", "Smith", "Lewis", "Smith", "Li"));  put("I", Arrays.asList("Perez"));  }}; | The generated Audit object is saved in auditList with the most recent timestamp | The generated Audit object is saved in auditList with the most recent timestamp |  |
| 7 | Test Statistics - testGenerateWinner | format = "CPL";  seatNumber = 7;  ballotNumber = 100;  Map<String, List<String>> parties = new HashMap<>() {{  put("D", Arrays.asList("Pike", "Foster", "Floyd", "Jones", "Mallory"));  put("R", Arrays.asList("Deutsch", "Wong", "Walters", "Keller", "Borg"));  put("G", Arrays.asList("Jones", "Smith", "Lewis", "Smith", "Li"));  put("I", Arrays.asList("Perez"));  }}; | The party that receives the most number of ballots is the winner | The party that receives the most number of ballots is the winner |  |
| 8 | Test Election and all get methods | None | No error report | No error |  |
| 9 | Test Election - setup | ballotSample1.txt (OPL case) | Integer seatNumberExpected = 3;  Integer ballotNumberExpected = 9;  Integer candNumberExpected = 6;  election.setup(oplFile);  assertEquals("OPL", election.getFormat());  assertEquals(oplFile, election.getFilePath());  assertEquals(seatNumberExpected, election.getSeatNumber());  assertEquals(ballotNumberExpected, election.getBallotNumber());  assertEquals(candNumberExpected, election.getCandidateNumber());  assertEquals(null, election.getPartyNumber());  assertEquals(6, election.getTotalCandidateList().size());  assertEquals(3, election.getPartyList().size()); | All pass |  |
| 10 | Test Election - setup | ballotSample2.txt (CPL case) | Integer seatNumberExpected = 7;  Integer ballotNumberExpected = 26;  Integer candNumberExpected = 16;  Integer partyNumberExpected = 4;  election.setup(cplFile);  assertEquals("CPL", election.getFormat());  assertEquals(cplFile, election.getFilePath());  assertEquals(seatNumberExpected, election.getSeatNumber());  assertEquals(ballotNumberExpected, election.getBallotNumber());  assertEquals(candNumberExpected, election.getCandidateNumber());  assertEquals(partyNumberExpected, election.getPartyNumber());  assertEquals(16, election.getTotalCandidateList().size());  assertEquals(4, election.getPartyList().size()); | All pass |  |
| 11 | Test Election - runElection | ballotSample1.txt (OPL case) | No any output (of error) to screen | No screen output |  |
| 12 | Test Election - runElection | ballotSample2.txt | NO output to screen | No screen output |  |
| 13 | System test\_open party | (base) ruoyan@ruoyan-Aspire-EK-571G:~/Documents/hw/repo-Team9/Project1$ ./make\_and\_run.sh  Welcome to Voting System!  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  setup  Please input your file path, press enter to finish:  ./testing/sampleBallot1.txt  Successly setup.  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  runElection  Begin running election.  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  party number NOT equal to length of party list!  generateAuditFile  Begin generating audit file.  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  getWinner  Begin generating winner.  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  shareToMedia  Begin sharing to media.  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  Type of Voting: open  Number of seats is 3  Number of parties is 3  Number of Candidates is 6  Number of current ballot cast is 9  Current Winner is D  Number of votes each candidate receives:  Jones : 1  Foster : 2  Smith : 1  Pike : 3  Borg : 2  Deutsch : 0  Number of votes each party receives:  R : 3  D : 5  I : 1  Number of seats each party receives:  D : 2  R : 1  I : 0  exit  Exit the VS. | Successly setup.  Begin running election.  Type of Voting: open  Number of seats is 3  Number of parties is 3  Number of Candidates is 6  Number of current ballot cast is 9  Current Winner is D  Number of votes each candidate receives:  Jones : 1  Foster : 2  Smith : 1  Pike : 3  Borg : 2  Deutsch : 0  Number of votes each party receives:  R : 3  D : 5  I : 1  Number of seats each party receives:  D : 2  R : 1  I : 0  exit | Successly setup.  Begin running election.  Type of Voting: open  Number of seats is 3  Number of parties is 3  Number of Candidates is 6  Number of current ballot cast is 9  Current Winner is D  Number of votes each candidate receives:  Jones : 1  Foster : 2  Smith : 1  Pike : 3  Borg : 2  Deutsch : 0  Number of votes each party receives:  R : 3  D : 5  I : 1  Number of seats each party receives:  D : 2  R : 1  I : 0  exit |  |
| 14 | System\_test close party | (base) ruoyan@ruoyan-Aspire-EK-571G:~/Documents/hw/repo-Team9/Project1$ ./make\_and\_run.sh  Welcome to Voting System!  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  setup  Please input your file path, press enter to finish:  ./testing/sampleBallot2.txt  Successly setup.  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  generateAuditFile  Please run your election first.  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  runElection  Begin running election.  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  generateAuditFile  Begin generating audit file.  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  getWinner  Begin generating winner.  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit  shareToMedia  Begin sharing to media.  -----Menu-----  setup  runElection  generateAuditFile  getWinner  shareToMedia  exit | Type of Voting: close  Number of seats is 7  Number of parties is 4  Number of Candidates is 16  Number of current ballot cast is 26  Current Winner is R  Number of votes each candidate receives:  Smith : 8  Smith : 8  Perez : 2  Mallory : 6  Jones : 6  Jones : 8  Foster : 6  Pike : 6  Wong : 10  Lewis : 8  Keller : 10  Li : 8  Floyd : 6  Borg : 10  Deutsch : 10  Walters : 10  Number of votes each party receives:  R : 10  D : 6  G : 8  I : 2  Number of seats each party receives:  R : 3  G : 2  D : 2  I : 0 | Type of Voting: close  Number of seats is 7  Number of parties is 4  Number of Candidates is 16  Number of current ballot cast is 26  Current Winner is R  Number of votes each candidate receives:  Smith : 8  Smith : 8  Perez : 2  Mallory : 6  Jones : 6  Jones : 8  Foster : 6  Pike : 6  Wong : 10  Lewis : 8  Keller : 10  Li : 8  Floyd : 6  Borg : 10  Deutsch : 10  Walters : 10  Number of votes each party receives:  R : 10  D : 6  G : 8  I : 2  Number of seats each party receives:  R : 3  G : 2  D : 2  I : 0 |  |

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

The winner and audit files of the election should be generated and shared to media.

The election should be exit.