Testing Logs for Project 2 – Agile Scrum

Team #24: Matthew Graba, Julian Heyman, Dahir Ali, Muhsin Mohamed

# Log 1:

| **The PBI, the Task Description (from Sprint Log) with Unique Testing Number:** | PBI: Display PO stats  Task: Implement audit in PO Class so user can see election results  Testing Number: 1 |
| --- | --- |
| **Team Member(s) Responsible:** | Matt Graba |
| **Inputs:** | 1. Name for election 2. Names for candidates 3. Value for the number of candidates 4. Value for the number of ballots 5. Ballot array in CSV format |
| **Tests:** | 1. Aduit produces an election name that equals the election name input 2. Aduit produces candidate names that equals the inputted candidate names 3. Audit produces a value for number of candidates that equal the number of candidate inputted 4. Audit produces a value for the number of ballots that equal the number of ballots inputted 5. Audit produces the correctly distributed votes for candidates |
| **Outputs:** | 1. Audit.txt: 2. Election Name: Test Election 3. Candidate Names: Candidate A, Candidate B, Candidate C, Candidate D, Candidate E, Candidate F 4. Number of Candidates: 6 5. Number of Ballots: 9 6. Vote Counts: 7. Winner: Candidate A 8. Candidate F: 1 9. Candidate E: 1 10. Candidate D: 2 11. Candidate C: 0 12. Candidate B: 2 13. Candidate A: 3 |
| **Passed or Failed:** | Passed |
| **Date:** | 4/27 |

# Log 2:

| **The PBI, the Task Description (from Sprint Log) with Unique Testing Number:** | PBI: Implement PO election  Task: Count votes  Testing Number: 2 |
| --- | --- |
| **Team Member(s) Responsible:** | Matt Graba |
| **Inputs:** | 1. Name for election 2. Names for candidates 3. Value for the number of candidates 4. Value for the number of ballots 5. Ballot array in CSV format |
| **Tests:** | 1. expectedCandidateVotes.put("Candidate A", 3); 2. expectedCandidateVotes.put("Candidate B", 2); 3. expectedCandidateVotes.put("Candidate C", 0); 4. expectedCandidateVotes.put("Candidate D", 2); 5. expectedCandidateVotes.put("Candidate E", 1); 6. expectedCandidateVotes.put("Candidate F", 1); |
| **Outputs:** | 1. *assertEquals*(expectedCandidateVotes, electionData.getCandidateVotes());   System returns true |
| **Passed or Failed:** | Passed |
| **Date:** | 4/27 |

# Log 3:

| **The PBI, the Task Description (from Sprint Log) with Unique Testing Number:** | PBI: Implement PO election  Task: Determine winner with no ties  Testing Number: 3 |
| --- | --- |
| **Team Member(s) Responsible:** | Matt Graba |
| **Inputs:** | 1. candidateVotes.put("Candidate A", 5); 2. candidateVotes.put("Candidate B", 3); 3. candidateVotes.put("Candidate C", 4); 4. candidateVotes.put("Candidate D", 2); 5. candidateVotes.put("Candidate E", 1); 6. candidateVotes.put("Candidate F", 0); |
| **Tests:** | 1. Candidates receive different values of votes with one receiving the most |
| **Outputs:** | 1. Candidate with most votes declared winner 2. Winner properly contains the number of votes as the candidate with the most votes |
| **Passed or Failed:** | Passed |
| **Date:** | 4/27 |

# Log 4:

| **The PBI, the Task Description (from Sprint Log) with Unique Testing Number:** | PBI: Implement Coin Toss for PO  Task: Coin toss, Determine winner with ties  Testing Number: 4 |
| --- | --- |
| **Team Member(s) Responsible:** | Matt Graba |
| **Inputs:** | 1. candidateVotes.put("Candidate A", 5); 2. candidateVotes.put("Candidate B", 3); 3. candidateVotes.put("Candidate C", 5); 4. candidateVotes.put("Candidate D", 2); 5. candidateVotes.put("Candidate E", 1); 6. candidateVotes.put("Candidate F", 0); |
| **Tests:** | 1. Candidates receive different values of votes except with two receiving the same number of votes |
| **Outputs:** | 1. One of the candidates with the most votes is declared winner 2. Winner properly contains the number of votes as the candidate with the most votes |
| **Passed or Failed:** | Passed |
| **Date:** | 4/27 |

# Log 5:

| **The PBI, the Task Description (from Sprint Log) with Unique Testing Number:** | PBI: Create a graphical user interface to select files  Task:  Testing Number 5: |
| --- | --- |
| **Team Member(s) Responsible:** | Muhsin, Dahir |
| **Inputs:** |  |
| **Tests:** | 1. Once we run the electionprocessor file, a GUI should pop up and allow us to select files |
| **Outputs:** |  |
| **Passed or Failed:** | Passed |
| **Date:** | 4/28 |

# Log 6:

| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | PBI: Process all file headers for an election  Task: Process the headers  Testing Number:6 |
| --- | --- |
| Team Member(s) Responsible: | Muhsin, Dahir |
| Inputs: |  |
| Tests: | 1. Test to see if the files have valid headers: IR,CPL,PO and if not it prints the election type is null |
| Outputs: | 1. If the ballot file has a valid header it proceeds to process its respective algorithm and if not it print that the election type is null |
| Passed or Failed: | Passed |
| Date: | 4/27 |

# Log 7:

| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | PBI: Process all file headers for an election  Task: Re-implement IR and CPL  Testing Number: 7 |
| --- | --- |
| Team Member(s) Responsible: | Dahir, Muhsin |
| Inputs: |  |
| Tests: | 1. The cpl class needs to print the correct distribution of seats to each party and produce the correct audit file 2. The IR class needs to produce the correct winner of the election and produce the correct audit file |
| Outputs: | CPL:        IR: |
| Passed or Failed: | Test 1: Passed  Test 2: Passed |
| Date: | 4/27 |

# Log 8:

| **The PBI, the Task Description (from Sprint Log) with Unique Testing Number:** | PBI: Have closed party listing election  Task: Produces correct winner  Testing Number: 8 |
| --- | --- |
| **Team Member(s) Responsible:** | Dahir, Muhsin |
| **Inputs:** |  |
| **Tests:** | 1.File goes through CPL, the prints party and distributed seats |
| **Outputs:** |  |
| **Passed or Failed:** | Passed |
| **Date:** | 4/30 |

# Log 9:

| **The PBI, the Task Description (from Sprint Log) with Unique Testing Number:** | PBI: Have closed party listing election  Task: Produces audit file  Testing Number: 9 |
| --- | --- |
| **Team Member(s) Responsible:** | Dahir |
| **Inputs:** |  |
| **Tests:** | 1. File ran through the CPL, and gives winner, should produce audit file |
| **Outputs:** |  |
| **Passed or Failed:** | Passed |
| **Date:** | 4/29 |

# Log 10:

| **The PBI, the Task Description (from Sprint Log) with Unique Testing Number:** | PBI: Have closed party listing election  Task: Displays winner to screen  Testing Number: 10 |
| --- | --- |
| **Team Member(s) Responsible:** | Dahir |
| **Inputs:** |  |
| **Tests:** | File goes through CPL algorithm, the prints party and distributed seats to the screen |
| **Outputs:** |  |
| **Passed or Failed:** | Passed |
| **Date:** | 4/30 |

# Log 11:

| **The PBI, the Task Description (from Sprint Log) with Unique Testing Number:** | PBI: Run instant runoff election  Task: Produces correct winner  Testing Number: 11 |
| --- | --- |
| **Team Member(s) Responsible:** | Muhsin, Dahir |
| **Inputs:** |  |
| **Tests:** | 1. File goes through IRV algorithm, the prints party and distributed seats to the screen |
| **Outputs:** |  |
| **Passed or Failed:** | Passed |
| **Date:** | 4/30 |

# Log 12:

| **The PBI, the Task Description (from Sprint Log) with Unique Testing Number:** | PBI: Run instant runoff election  Task: Produces audit file displaying election information  Testing Number: 12 |
| --- | --- |
| **Team Member(s) Responsible:** | Dahir |
| **Inputs:** |  |
| **Tests:** | 1. Test to see if it produces a audit file with correct information |
| **Outputs:** |  |
| **Passed or Failed:** | Passed |
| **Date:** | 4/30 |

# Log 13:

| **The PBI, the Task Description (from Sprint Log) with Unique Testing Number:** | PBI: Run instant runoff election  Task: Display the correct winner to screen  Testing Number: 13 |
| --- | --- |
| **Team Member(s) Responsible:** | Muhsin, Dahir |
| **Inputs:** |  |
| **Tests:** | 1. Check to see if the screen produced the right winner |
| **Outputs:** |  |
| **Passed or Failed:** | Passed |
| **Date:** | 4/30 |