Committed Backlog Items	Not Started	In Progress	Completed	
1. CPL multiple files			refactor/split countSeats function - **Ziyoda**	
As <**an Election Official**> I want <**CPL Algorithm to handle multiple files**> So that <**we can bring in different files from different balloting locations.**>			refactor main to take in multiple files - **Ziyoda**	
			refactor main "front-end", so use can type multiple files - **Andrei**	
Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete) - **Program can accept multiple files** - **It can read multiple files correctly** - **It will always produce the expected result when summing up votes**			refactor MAKE file so it can take multiple files - **Ziyoda**	
			write CPL files for multiple locations	
Definition of Done (what is required by the team before sending out for review) - **Program can accept multiple files via CMD line, or asks user** - **All files can be read** - **Votes are summed up correctly** - **One result/audit is produced**			add header (name, comment) to .h files - **Chork**	
			create more files for testings and testlogs - **Chork**	
			Testing - unit test newly split functions - **Mohamed**	
			Testing - Create new test logs - **Chork**	
Effort: Small, Medium, **Large**, Extra Large			Doxygen - Documentation for the header - **Mohamed**	
PBI Author(s) **Andrei Anicescu**			C++ documenation - **Andrei**	
2. OPL multiples files			refactor/split countSeats function - **Mohamed**	
As <**an Election Official**> I want <**OPL Algorithm to handle multiple files**> So that <**we can bring in different files from different balloting locations.**>			refactor main to take in multiple files - **Ziyoda**	
			write OPL files for multiple locations -**Andrei**	
			refactor MAKE file so it can take multiple files - **Andrei**	
Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete) - **Program can accept multiple files** - **It can read multiple files correctly** - **It will always produce the expected result when summing up votes**			Testing - ensure output is as expected - **Mohamed**	
			Testing - unit test newly split functions - **Chork**	
Definition of Done (what is required by the team before sending out for review) - **Program can accept multiple files via CMD line, or asks user** - **All files can be read** - **Votes are summed up correctly** - **One result/audit is produced**			Testing - Create new test logs - **Chork**	
			Doxygen - Documentation for the header - **Andrei**	
			C++ documenation - **Andrei**	
Effort: Small, **Medium**, Large, Extra Large				
PBI Author(s) **Andrei Anicescu**				
3. MPO files (in general)			allow for MPO files to be read **in main -Andrei**	
As <**an Election Official**> I want <**the program to handle one MPO file**> So that <**so that MPO elections can be processed**>			Create .csv files for testing - **Chork**	
			Testing - Create Unit Tests For new functions - **Mohamed**	
Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete) - **MPO header is read, and file can be opened and is processed accordingly**			Testing - Create new test logs - **Ziyoda**	
			Doxygen - Documentation for the header - **Ziyoda**	
Definition of Done (what is required by the team before sending out for review) - **When header is read correctly, objects instantiated, tests passing**			C++ documenation - **Ziyoda**	
			Create MPO files - **Ziyoda**	
Effort: Small, **Medium**, Large, Extra Large				
PBI Author(s) **Ziyoda Mamatkulova**				
4. MPO multiple files			write MPO files for multiple locations - **Andrei**	
As <**an Election Official**> I want <**the program to handle multiple MPO files**> So that <**so that MPO elections from multiple locations can be processed**>			Testing - ensure output is as expected - **Mohamed**	
			C++ documenation - **Andrei**	
			Testing - Create new test logs - **Ziyoda**	
Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete) - **MPO header is read, and file can be opened and is processed accordingly** - **System will read files one by one**			Testing - System Testing - **Ziyoda**	
			Doxygen - Documentation for the header - **Chork**	
Definition of Done (what is required by the team before sending out for review) - **When first header is read correctly, objects instantiated, tests passing** - **Each following file the # of votes are read and tallied**				
Effort: Small, **Medium**, Large, Extra Large				
PBI Author(s) **Ziyoda Mamatkulova**				

Committed Backlog Items	Not Started	In Progress	Completed	
5. MPO Election			Unit tests (allocateSeats()) - **Ziyoda**	
As <**an Election Official**> I want <**an MPO election to be processed correctly**> So that <**so that all votes are counted, and seats allocated for each candidate**>			Create allocateSeats() method - **Andrei**	
			Unit tests (handling ties) - **Mohamed**	
			Testing - System testing - **Ziyoda**	
Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete) - **Votes are tallied correctly from one or many files** - **Seats are allocated fairly, and ties are handled**			Testing - Create new test logs - **Mohamed**	
			Doxygen - Documentation for the header- **Chork**	
			C++ documenation - **Chork**	
Definition of Done (what is required by the team before sending out for review) - **All votes are counted correctly, & all seats allocated correctly and fairly**			create more files for testing - **Chork**	
Effort: Small, **Medium**, Large, Extra Large				
PBI Author(s) **Ziyoda Mamatkulova**				
6. MV files (in general)			Testing - Create Unit Tests For new functions - **Mohamed**	
As <**an Election Official**> I want <**the program to handle one MV file**> So that <**so that MV elections can be processed**>			Testing - Create new test logs - **Chork**	
			Implement System Testing - **Ziyoda**	
Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete) - **MPO header is read, and file can be opened and is processed accordingly**			Doxygen - Documentation for the header - **Andrei**	
			C++ documenation - **Andrei**	
Definition of Done (what is required by the team before sending out for review) - **When header is read correctly, objects instantiated, tests passing**				
Effort: Small, **Medium**, Large, Extra Large				
PBI Author(s) **Chork Hieng**				
7. MV multiple files			Testing - ensure output is as expected - **Chork**	
As <**an Election Official**> I want <**the program to handle one MV file**> So that <**so that MV elections from multiple locations can be processed**>			write MV files for multiple locations - **Mohamed, Ziyoda**	
			in audit/display mention # of locations? - **Andrei**	
			Implementing System Testing - **Mohamed**	
Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete) - **MV header is read, and file can be opened and is processed accordingly** - **System will read files one by one**			Doxygen - Documentation for the header - **Chork**	
			C++ documenation - **Andrei**	
Definition of Done (what is required by the team before sending out for review) - **When first header is read correctly, objects instantiated, tests passing** - **Each following file the # of votes are read and tallied**			Testing - Create new test logs - **Ziyoda**	
Effort: Small, **Medium**, Large, Extra Large				
PBI Author(s) **Chork Hieng**				
8. MV Election				
As <**an Election Official**> I want <**an MV election to be processed correctly**> So that <**so that all votes are counted, and seats allocated for each candidate**>				
Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete) - **Votes are tallied correctly from one or many files** - **Seats are allocated fairly, and ties are handled**				
Definition of Done (what is required by the team before sending out for review) - **All votes are counted correctly, & all seats allocated correctly and fairly**				
Effort: Small, **Medium**, Large, Extra Large				
PBI Author(s) **Chork Hieng**				
9. MPO stats to screen				
As <**an Election Official**> I want <**the results of the election to be displayed on the screen and/or in an Audit file**> So that <**so the results can be easily shared, winners are easily seen, and the number of votes each candidate received is recorded**>				
Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete) - **All stats are shown accurately** - **Winners and non-winners are easily differentiated**				

Committed Backlog Items	Not Started	In Progress	Completed	
Definition of Done (what is required by the team before sending out for review) - **Testing that all of the displayed or audited data is correct and accurate** - **The layout is easy to read**				
Effort: Small, **Medium**, Large, Extra Large				
PBI Author(s) **Mohamed Ali**				
10. MV stats to screen				
As <**an Election Official**> I want <**the results of the election to be displayed on the screen and/or in an Audit file**> So that <**so the results can be easily shared, winners are easily seen, and the number of votes each candidate received is recorded**>				
Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete) - **All stats are shown accurately** - **Winners and non-winners are easily differentiated**				
Definition of Done (what is required by the team before sending out for review) - **Testing that all of the displayed or audited data is correct and accurate** - **The layout is easy to read**				
Effort: Small, **Medium**, Large, Extra Large				
PBI Author(s) **Mohamed Ali**				