1. Input ballot from GUI

As an election official	
I want to input ballot data on the screen directly into the voting system instead of just load data through file.	
So that I could continually provide new data and see the real-time voting updates	
Acceptance Criteria	1. The system needs to provide a GUI for user to input Ballot 2. Before input ballot, the system prompt request to user to know the election type, number of candidates and etc. 3. System allow user to fulfill all related data of a ballot through that GUI 4. The input ballot should be loaded into ballot dataset properly and correctly
Definition of Done	1. Pass all regression test and ensure the GUI could prompt normally 2. Pass testing per acceptance criteria 3. Use different input to fully test the functionality, make sure the system could proceed normally no matter the input is correct or not. 4. Pass the unit test to make sure that the user is able to select and type the input
Efforts	Medium 4 hours

Note:
This user story is unfinished because for this assignment, we do not need to do the last three bullet on page 2 from the Document: Project2 Agile Scrum. This bullet is belonged to the last bullet. Which is: The election officials would like to have the ballots input on the screen directly into the program. They would like to be able to do this for one machine originally and eventually have the voting machines send the information directly to the system. They want a graphical user interface that looks nice. This will mean that a file would become another option for controlling the election and this would provide real-time voting updates. The election official would need a way to tell the program what type of election it is, the number of candidates, etc

2. Work on a client machine

As an election official	
I prefer the system allows me to input ballot on a client machine and the ballot inputted are automatically transferred to the remote voting system server.	
So that we could collect the voting data much more efficiently and see the real-time voting process	
Acceptance Criteria	1. The system needs to provide a client version for user to collect data remotely 2. The client need to encode the inputted ballot into a file and automatically transfer that file to server. 3. System need to guarantee every ballot successfully transferred to the remote server without lost any package.
Definition of Done	1. Pass unit testing for collecting data remotely 2. Pass unit testing for a single file upload 3. Pass the testing for correctness of data 4. Pass the testing for lost of package 5. Pass the system testing for a single machine run
Efforts	Large 8 hours

Note:
This user story is unfinished because for this assignment, we do not need to do the last three bullet on page 2 from the Document: Project2 Agile Scrum. This bullet is belonged to the last bullet. Which is:The election officials would like to have the ballots input on the screen directly into the program. They would like to be able to do this for one machine originally and eventually have the voting machines send the information directly to the system. They want a graphical user interface that looks nice. This will mean that a file would become another option for controlling the election and this would provide real-time voting updates. The election official would need a way to tell the program what type of election it is, the number of candidates, etc

3. Work on multiple client machines simultaneously

As an election official	
I want multiple clients could interact with the main voting system simultaneously and all machines would eventually sending updated voting information to the system without lost any ballot	
So that all users could share the updated voting result	
Acceptance Criteria	1. The system needs to provide a client version for user to collect data remotely 2. The system needs to have the ability to handle multiple clients' file updates simultaneously without any conflicts. 3. System need to guarantee every ballot successfully transferred to the remote server without lost any package. 4. System need to guarantee every client will receive update result from remote server without any lost package.
Definition of Done	1. Pass unit testing for collecting data remotely 2. Pass unit testing for a single file upload 3. Pass unit testing for multiple file upload 4. Pass unit testing for correctness checking of data 5. Pass unit testing for package lose checking 6. Pass system testing for a single machine run 7. Pass system testing for multiple machines run
Efforts	Large 8 hours

Note:
This user story is unfinished because for this assignment, we do not need to do the last three bullet on page 2 from the Document: Project2 Agile Scrum. This bullet is belonged to the last bullet. Which is:The election officials would like to have the ballots input on the screen directly into the program. They would like to be able to do this for one machine originally and eventually have the voting machines send the information directly to the system. They want a graphical user interface that looks nice. This will mean that a file would become another option for controlling the election and this would provide real-time voting updates. The election official would need a way to tell the program what type of election it is, the number of candidates, etc

Beside the three PBIs above, we did not detect any obvious bugs or unfinished parts so far.

PBI OPL & IRV works correctly is accepted by the product owner.

PBI Check and remove invalid ballots from row data is accepted by the product owner.

PBI Provide short report file is accepted by the product owner.

PBI Display result table is accepted by the product owner.

PBI Efficiently process tons of data is accepted by the product owner.

PBI Auto detect info and format from CSV is accepted by the product owner.

PBI Provide two ways to specify filename is accepted by the product owner.

PBI Provide GUI for filename request is accepted by the product owner.