<div align="center">**Deliverable 1**</div>

<div align="center">*Team Members: Levi Hagan, Richard Marshall and Hannah Posch*</div>

<div align="center">**Vision**</div>

Revision History

Version	Date	Description	Authors
Deliverable 1	Sep 09, 2018	First draft. To be refined during deliverable 2.	Hannah Posch Levi Hagan Richard Marshall

Introduction

We envision an electronic voting system with the reliability to cast and store citizens' votes, and the usability to ensure those votes can be accounted for, while remaining secure.

Positioning

Business Opportunity

Current voting systems are not entirely reliable in the storage of votes and the security of the system as seen with many recent elections that may have been hacked. This ineffective, non-democratic way of voting has created a need for a system that is reliable and accurate and provides the honest results of the election. Our electronic voting system is needed to rectify these problems previously mentioned as the system is built to remain secure and provide precise results.

Problem Statement

Traditional voting systems are subject to voter tampering, inaccuracy and corruption. This causes problems in voting results that can have alternative effects of government and local affairs depending on the candidate elected. These issues do not reflect the will of the people and affect voters, voting assistants, candidates, and even administrators.

Product Position Statement

The system is for any voter to be able to complete their civic duty and cast their vote, as well as, provide a secure, reliable system that prevents over-voting creating a much better experience while supporting its purpose: democracy.

Alternatives and Competition

One major alternative to our system is the traditional paper ballot system.

Stakeholder Descriptions

Market Demographics

Any citizen wishing to fulfill their civic duty in a democratic manner.

Stakeholder(Non-User) Summary
- Government: This stakeholder includes the state officials (or another level of official) that oversees the entire voting process and promotes the use of a democratic election.
- Voting Service: This stakeholder is the main business in charge of providing the government with a voting system and would be hired to complete the voting poll setup.
- Candidate: This stakeholder is the political candidate that is included on the electronic ballot.

User Summary
- Voter: This user includes the everyday citizen who wishes to cast their vote in the election.
- Administrator: This user is the designated person that has access to the unofficial results from the election.
- Voting assistant: This user includes those who have been assigned the duty of helping voters and may need to operate the system depending on the scenario.

Key High-Level Goals

High-Level Goal	Priority	Problems and Concerns	Current Solutions
Fast, reliable secure vote submission	High	-Secure system not susceptible to hacking and voter fraud -System bugs that prevent accurate voting results or cause inability to vote -Reduced submission speed as more systems come online -Inability to authorize voter	Current voting systems supply basic voting needs, but do not prevent all problems described.
Easy to use interface, and able to locate voting results	High	-User interface is too complicated and user is unable to complete task -Voting results are not provided when requested -Administrator not able to authenticate identity to access results	Current voting systems do meet most of these needs, however, if we could make our system more efficient and easy to use, then that is our current goal.

User-Level Goals

The users require a system that can meet these needs:
- Voter: cast vote, submit vote, authenticate identity
- Administrator: check unofficial vote tally
- Voting assistant: Start up, shut down, restart system

● Electronic voting system: cast and count votes electronically

User Environment

Voting poll location which will include machines with voting system installed.

Product Overview

Product Perspective

The electronic voting system will be located in a voting poll location. Due to the individual machines being used by the voter, the machines will be in close proximity to the network, if not hard wired to the network. It will provide a service to voters, and administrators as indicated by Figure Vision-1.

Figure Vision-1. Voting system context diagram.
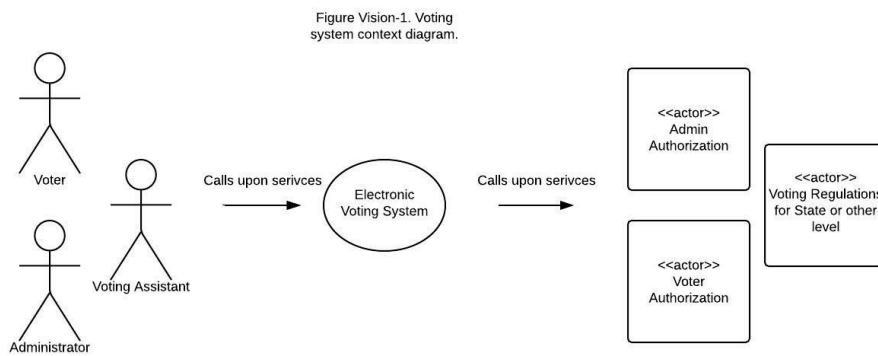
Summary of Benefits

Supporting Feature	Stakeholder Benefit
The system will supply the necessary services to complete a vote, including submission of vote and unofficial tally.	Automated, and easy to use voting services.
Automatic voter and admin authorization will be required for all functionality.	Secure system that prevents over-voting and vote tampering.
Real-time voting submissions using state or specific voting level regulations	Reliable and regulated system that is comparable to any paper ballot system.

Summary of System Features
  ● Vote submission
  ● Voter authorization

- Admin authorization
- Real-time unofficial voting tally

Other Requirements and Constraints

See the Supplemental Specification and two use cases for additional details on design constraints, reliability, usability, performance, documentation, packaging, supportability and more.

Use Case UC1: Cast Vote

Scope: Voting Application

Level: User-goal

Primary Actor: Voter

Stakeholders and Interests:

-Voter: Wants accurate, fast voting experience with no system errors, and vote to be submitted correctly.

-Government: Wants to calculate votes correctly from every voting period. May include multiple agencies, such as national, state and local officials.

-Candidates: Want accurate, secure voting system with no errors in calculations.

-Voting service: Want reliable, fast voting system that works efficiently and is fair in results.

-Admin: Want an easy to use system that is secure and easy to locate results.

Preconditions: System is up and running with a voting period in place.

Success Guarantee (or Postconditions):

- Individual vote is saved, vote tally is correctly calculated

- System resets to handle next voter

Main Success Scenario (or Basic Flow):

1. Voter arrives at voting booth.
2. Voter logs in/enters identification.
3. System validates identification.
4. Voter chooses candidate.
5. Voter verified candidate selection.
6. System saves vote.
7. System verifies vote submission.
8. System saves voter identification.
9. System returns to main menu and is ready for next voter.

Extensions (or Alternative Flows):

*a. At any time, System fails:

1. System displays an error.

2. Voting attendant attempts to restarts system.

3. If system unable to start, voter completes vote on paper.

2a. Vote fails to input valid input identification information:

1. System Displays Error

2. Voter is found unable to vote

3a. Voter identification not valid:

3. System Displays Error

4. Voter is found unable to vote

3b. Voter has already voted:

1. System Displays Error

2. Voter is found unable to vote

4a. Voter chooses to write in a candidate:

1. Systems displays field to enter the name of the candidate.

2. Voter enters name of candidate.

5a. System unable to verify candidate selection:

1. System displays error.

2. Voting assistant assists voter in filling out a paper ballot.

6a. System unable to save vote:

1. System displays error

2. Voting assistant assist voter in filling out paper ballot.

7a. System unable to save voter identification:

1. Voting attendants write down voter's identification by hand.

Special Requirements:

- UI on a large flat panel monitor. Text must be visible from 1 meter.

- UI color pallet must be color blind friendly

- Voter authorization response within 1 minute 90% of the time.

- System vote submission confirmation response within 1 minute 90% of the time.

- System must be secure.

Technology and Data Variations List:

- System must support keyboard and mouse or touch input.

Frequency of Occurrence:

- Nearly continuous during voting period.

Open Issues:

- Variations in Local, State, and Federal Elections?
- How to deal with a large number of candidates?
- How to handle voter verification?

Use Case UC2: Tallying Votes

Scope: Voting Application

Level: User-goal

Primary Actor: Admin

Stakeholders and Interests:

-Voter: Wants accurate, fast voting experience with no system errors, and vote to be submitted correctly.

-Government: Want to calculate votes correctly from every voting period. May include multiple agencies, such as national, state and local officials.

-Candidates: Want accurate, secure voting system with no errors in calculations.

-Voting service: Want reliable, fast voting system that works efficiently and is fair in results.

-Admin: Want an easy to use system that is secure and easy to locate results.

Preconditions:

- Voting period is closed.
- System is still running.
- Votes have been saved.

Success Guarantee (or Postconditions):

- System displays vote totals.

Main Success Scenario (or Basic Flow):

1. Administrator logs into system.
2. System verifies administrator's identity
3. Administrator selects voting results.
4. System displays voting results.

Extensions (or Alternative Flows):

*a. If at any time system fails:

1. System displays an error.

1a. Administrator is not able to log into system.

1. System displays an error.

2a. System unable to verify administrator identity.

1. System displays an error.

4a. System unable to display voter results.

1. System displays an error.

Special Requirements:

- UI on a large flat panel monitor. Text must be visible from 1 meter.

- UI color pallet must be color blind friendly

- Voting results response within 1 minute 90% of the time.

- Administrator confirmation response within 1 minute 90% of the time.

- System must be secure.

Technology and Data Variations List:

- System must support keyboard and mouse or touch input.

Frequency of Occurrence:

- At least once or more per voting period.

Open Issues:

- How to handle administration log in?

- How to address incorrect results?

<div align="center">**Supplemental Specification**</div>

Introduction

This document is the repository of all voting system requirements not captured in the use cases.

Functionality
- ***Error Handling:*** All errors will be followed by a message to the voter, voting assistant or administrator.
- ***Security:*** All usage requires identification and authorization to use the system.

Usability
- ***Human Factors:*** The voter will be able to see a large-monitor display of the voting system within the voting booth.
    - Avoid a color palette associated with most forms of color blindness.
    - System text should also be visible from 1 meter away.
- Speed, accuracy, and reliability are monumental in a voting system, as the voter wants to cast their vote quickly and ensure that is was submitted correctly.
- Message boxes signaling errors should be used to ensure that if a voter or administrator is using the system, they can be made aware of the error.

Reliability
- ***Recoverability:*** If the system begins to malfunction, try to solve it with a paper solution (cast vote on paper), or alert voting assistant to check the system. It is imperative that the voting assistant is given instruction via the system of how to check and reboot system if needed. More analysis will be needed in this portion as the system is developed.

Performance
- Voters want to submit their votes quickly, however, one issue that may slow this down is voter authorization. Our goal as mentioned in the use cases is to have the voter authorization within 1 minute, 90% of the time.
- The administrator also want to access the voting results quickly, as well as, ensure that the votes have been counted accurately. Another goal we have is to have the administrator authorization within 1 minute, 90% of the time. We would also like the system to display the voting results to the administrator within 1 minute, 90% of the time.

Supportability
- ***Adaptability:*** Although most of the voting scenarios should be similar because each voter is simply casting our vote, we do want our system to remain adaptable to any issues that may arise. Also we want our system to allow write in candidates which will require the system to adapt as it counts votes.
- ***Configurability:*** Our voting system will need to be configurable to the voting election type it is being used for (local, state, federal). This support mechanism allows the system to be useful in a variety of scenarios, and apply the correct voting rules and vote calculations when necessary. This portion will need more analysis as we begin the design and development of our system.

Implementation Constraints (Hardware and Software)
- Our team will be using Java and JavaFX as the base of our system, as our team is familiar with these chosen software constraints which will help the development of our project run smoother.

Purchased Components
- During this phase of development, we do not have any purchased components for our system.

Free Open Source Components
- At this time, we don't have definite components in mind, but we would also like to use our own developed components as much as possible due to the nature of the project.
- Since the security of this project remains at the forefront, we would also be weary to use free open source code with known security issues as it could lead to an unsecure system.

Documentation
- *Voter Instructions:* Voters will be provided with instructions on how to verify their identity before continuing and whether their ability to continue voting is valid. Voters will also be given clear instruction on how to cast their vote throughout the system. The voter will also be asked to confirm their vote before submitting.
- *Administrator Instructions:* The administrator will be given instruction on how to login to the system, as well as, complete an unofficial tally at the specific precinct.
- *Help:* In the event of system failure, instructions will be provided to the voting assistant on ways to troubleshoot or reboot the system if necessary.

Internationalization concerns
- Due to the nature of the voting system, we remain confident that our system is capable of being internationalized. To ensure the system is effective in another company, the voting rules would need to be changed and the written instruction throughout the UI would need to be translated. However these changes would not be extremely difficult and allow the developer to transition our system to another location.

Interfaces

Hardware Interfaces
- Touch screen monitor

Software Interfaces
- At the time we are not using any software interfaces (this is subject to change)

Risks

Hardware
- Software may be deployed on a variety of hardware depending on the client which may lead to compatibility and security issues

Time
- Production time
- Run time

- Time constraints based on the election

Labor
- Voter assistant and Government Administrators are required for this system operate as intended

Conversions
- Different types of elections require different numbers of candidates and election types

Legal Issues
- Inaccurate results could cause severe issues
- Software must be compliant with all voting laws which vary from place to place

Application-Specific Domain(Business) Rules

Rule Number	Rule	Changeability	Source
Rule 1	Proper Identification must be used and be able to be verified	States voter ID laws change as newer forms of identification are produced such as SC's new "REAL ID"	Law
Rule 2	Time constraints	Some States Require voting booths to shut off after the election period is over and not allow new voters.	Law

Information in Domains of Interest

Voter identification
- With recent events voter fraud is a big issue so our team must have an effective system to identify voters and prevent voter fraud. Because different states require different levels of identification to vote our system must be able to recognize all of them, such as driver's license numbers, social security numbers, or green cards for California.

Reports
- Inception document created.
- UI mockup created

Concerns (Environment and Operational)
- Must use JavaFX (requirement)
- Must use Java (team familiarity)

Glossary

Definitions

Term	Definition and Information	Format	Validation Rules	Aliases
System	The voting application.	N/A	N/A	None
Voter	Person casting a vote that will have a voter account tied to their identification which allows them to cast their vote for a specific candidate once in each voting period. Elements include a name, voter ID, and social security number.	N/A	N/A	None
Voting Assistant	Staff at the election facility who assists the voter.	N/A	N/A	None
Administrator	The person in charge of checking the results from the system, with a special admin account which allows them to check the results of a voting period on the machine, as well as to begin and end a voting period.	N/A	N/A	Admin
Voting Period	Time in which the election occurs and the system takes votes.	N/A	N/A	None
Administrator Authorization	Validation by an external system to ensure that the admin	N/A	N/A	Admin Authorization

	has permission to view the voting results.			
Voter Authorization	Validation within the system to ensure the voter is not a duplicate voter.	N/A	N/A	None

Use Case UML Diagram