

## Deliverable 1

By Brandi Durham, Austin Hollis, Justin Willis

**Vision:** Our vision for this project is to make a secure, easy to use e-election software application with the following features:

- Easy to use and interact with
- Keeps information and votes secure
- Allows for immediate voting tally at any time
- Unbiased candidate display
- Easy to maintain
- Quick and easy to understand instructions

We understand each stakeholder has unique requirements and specification for the election software, and we hope to meet each of these. While there may be some speculation around election software, we hope that by outlining exactly how we plan to implement the software we can show confidence in our product. In each of the use cases below, we have outlined what we think each stakeholder will be interested in. These may change after meeting with the client.

**Use Case 1:** User arrives to vote, already registered to vote

**Scope:** Election Software application

**Level:** user goal

**Primary Actor:** User

**Stakeholders and Interests:**

- User: Want accurate results, the person they votes for to receive a vote, fast and easy-to-use interface, privacy of personal information and who they votes for, quick turnaround on results.
- Candidates: Want accurate results, equal representation (alphabetical order, no bold, same fonts), fast and easy to use interface process, quick turnaround on results.
- Government/Business: Want inexpensive in terms of resources and management, easy to maintain, accurate results, quick turnaround on results.
- Field Admin/Proctor: Want smooth setup, easy to maintain, proper working application, easy to follow instructions, quick process for each user.

**Preconditions:** Proctor has application up and running, user is registered before current day.

**Success Guarantee:** Vote is counted, accurate vote, information is secure.

**Main Success Scenario (or Basic Flow):**

1. User arrives and logs in at an available computer
2. Vote \*repeat for number of categories
3. System shows review page of choices
4. User clicks confirm button on review pages
5. System logs choices in database, or adds to queue to add to database
6. "Thank you" page appears indicating user can leave and vote has been processed
7. User leaves

**Extensions (or Alternative Flows):**

- User is not registered
  - Send user to registration page

- Resume voting process
- Wrong login credentials
  - Tell user information is incorrect and have them try again (set number of attempts)
  - If user cannot login, screen locks and waits for admin
- Trying to vote before time
  - Tells user that it's too early
  - Admin would reset screen to home
- Application freezes
  - Databases store after each vote
  - When the users re-logs in, jumps to unanswered category

**Special Requirements:**

- Computer with keyboard and mouse
- Processing time for vote to register in database is with 5 mins.

**Technology and Data Variations List:**

- Admin override by keyboard code
- Vote by clicking
- Log in by entering name and social security number

**Frequency of Occurrence:** Could be nearly continuous.

**Open Issues:**

- When is voting open?

**Use Case 2:** User wants to register to vote.

**Scope:** Election Software application

**Level:** user goal

**Primary Actor:** User

**Stakeholders and Interests:**

- **User:** Fast and easy to use interface, privacy of personal informations, clear registration information, secure database.
- **Candidates:** Want voters to have a fast and easy registration process, voters given clear registration information so they can vote.
- **Government/Business:** Want Inexpensive in terms of resources and management , easy to maintain, secure database.
- **Field Admin/Proctor:** Want smooth setup, easy to maintain, proper working application, easy to follow instructions, quick process for each user.

**Preconditions:** Proctor has application up and running, and user is not registered to vote already.

**Success Guarantee:** User is able to register to vote, vote is counted, accurate vote, information is secure.

**Main Success Scenario (or Basic Flow) :**

1. User arrives
2. User chooses to register at an available computer
3. User inputs personal identification (Ex: first name, last name, DOB, SSN, etc.)
4. User clicks submit to confirm registration

5. Application tells user they are registered
6. User is notified that they are not allowed to vote on the same day as registration but is told when they can vote along with when voting ends
7. User returns to vote on appropriate day

**Extensions (or Alternative Flows):**

- Incorrect Registration Information
  - User may be asked to re-enter information is not correct
- Application freezes
  - Application doesn't store registration data until user clicks submit
  - If application freezes while user is registering then data is not saved and user is prompted to enter information again

**Special Requirements:**

- Computer with keyboard and mouse
- Processing time for vote to register in database is within 5 mins.

**Technology and Data Variations List:**

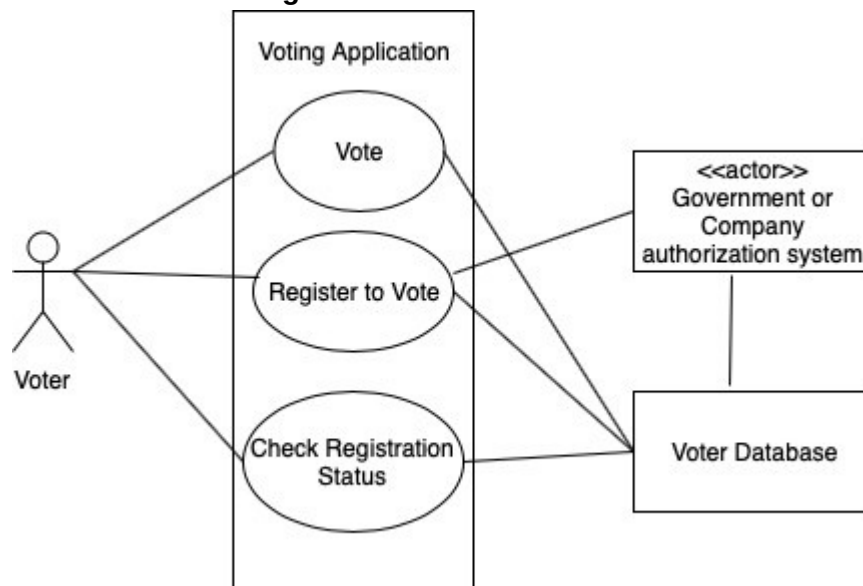
- Admin override by keyboard code

**Frequency of Occurrence:** Could be nearly continuous.

**Open Issues:**

- When is voting open?
- Incorrect information?

**Use Cases UML diagram:**



**Supplementary Specifications:**

**Functionality:** The user should be able to cast their vote as mentioned in the use cases.

**Logging and Error Handling:** The software will handle all errors and will require an admin in

some cases to resume.

**Pluggable Rules:** The system will be customizable to do certain features bases on the use cases.

**Security:** The user will have to identify themselves in order to vote, identification methods to be decided on later. There will be no chance to over-vote and voter information as well as choices will be kept secure.

**Usability:** The screen will have font that is easy to read for user. The interface will be easy to use and understand. There will be no colors associated with color blindness. User will be asked to confirm choices many times, and have the option to change vote.

**Reliability:** There should be a way for the user to start over or continue if there is a problem with the software and a reset is required.

**Performance:** The user should not have to wait longer than 5 min for their vote to be processed and counted in the database.

**Supportability:** The system could be configured for any type of voting needs and with any number of categories or canadetes.

**Implementation Constraints:** The software is implemented using Java and JavaFX.

**Purchased Components:** This would be the computer the application is run on and possibly the database being used.

**Free Open Source Components:** Will be using the free Java software, Java 8 and JavaFX along with Scene Builder to build the software application.

**Interfaces:** Computer with keyboard and mouse.

**Application-Specific Domain Rules:**

- User must register to vote
- User cannot vote on the same day that you register
- User can only vote during specified voting periods

Legal Issues: All voting rules in the state must be followed.

Information in Domains of Interest: Each state has their own rules for voter registration. Each users has to have identifiable information in order to register to vote.

## **Glossary:**

**Ballot** - the piece of paper used to record someone's vote.

**Candidate** - a person who applies for a job or is nominated for election.

**Category** - The classification of the candidates, tells the user who or what they are voting for.

**Election** - a formal and organized choice by vote of a person for a political office or other position.

**Election Proctor** - the person that will be overseeing the voting procedures.

**Policy** - a course or principle of action adopted or proposed by a government, party, business, or individual

**Vote** - a formal indication of a choice between two or more candidates or courses of action, expressed typically through a ballot or a show of hands or by voice.

**Voter** - a person who votes or has the right to vote at an election

## **Risk List:**

**Time** -There could be a latency between submitting the vote and processing it in the database.

This could cause user wait times.

**Hardware** - There needs to be a decent amount of computers available so the wait time between users voting is not unacceptable.

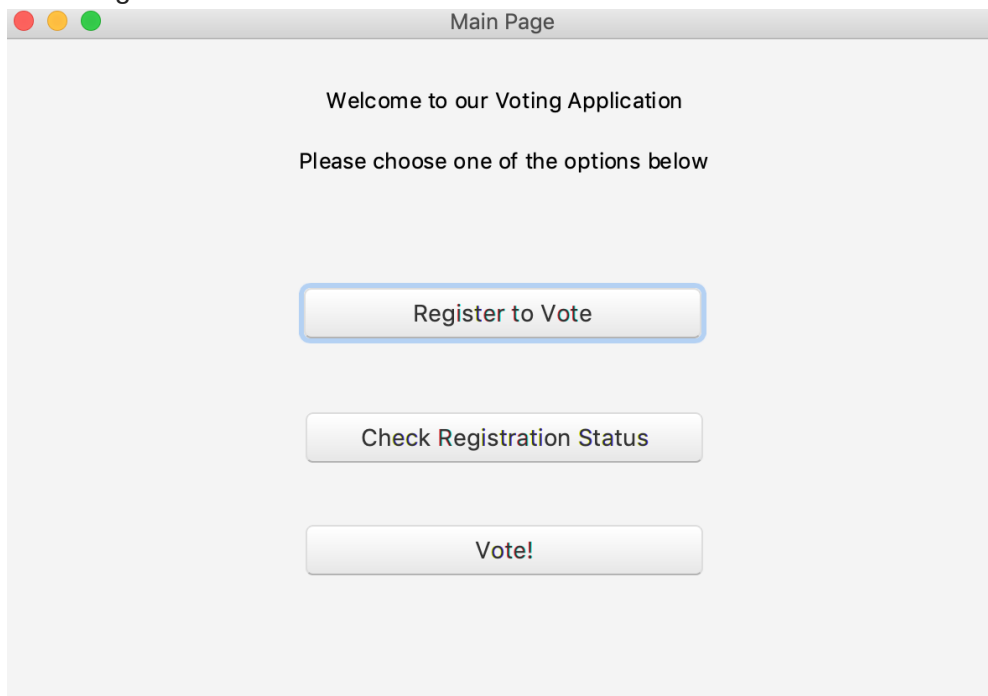
**Labor** - The application should be easy for the system admin to set up and should be able to override the system if it freezes or can't connect to database.

**Conversions** - The users information, especially their social security numbers, should be kept in the database in plain text. There should be some amount of effective hashing done before information is stored.

**Legal** - The users information and voting choices should be kept private. There would be a legal issue if other people could access the database.

GUI Prototype:

Main Page:



Voter Registration Page:

Registration Page

Registration Page

First Name

Last Name

Date of Birth

Social Security Number

Password

Check Registration Status Page:

Check Status Page

Check Registration Status

First Name

Last Name

Date of Birth

Social Security Number

Voting Page:

## CSCI 360 Elections

You may only have one candidate selected for each category before submitting ballot.

### ▼ President

☒ Candidate 1      Description for Candidate 1

☐ Candidate 2      Description for Candidate 1

☐ Candidate 3      Description for Candidate 1

### ▶ Vice President

### ▶ Secretary

### ▶ Treasurer

Cancel

Submit