Washington State University

Electronic Voting System

Deliverable 1-2

Tengyang Zhang, Riley Gusa, Tianxiang Xu, Wei-Lun Huang, Chien-Min Lee

CPTS 583

Professor Haipeng Cai

February 08, 2022

Requirements

A. Use case diagram

B. Use case elaboration

Use case	Register voter
Actors	administrator database

Goal	Add new voters to database
Preconditions	Voter doesn’t exist
Scenarios	Voter information is provided Voter is created
Exceptions	Lack of permissions

Use case	Login
Actors	Voter
Goal	Login to voting portal
Preconditions	Voter has an account
Scenarios	Voters provide id and password
Exceptions	Invalid id or password

Use case	Search for candidate information
Actors	Voters
Goal	Search for a candidate’s information in the voting system
Preconditions	Voters know nothing about any candidates
Scenarios	The voter searches candidate information according to his/her candidate’s number/name

Exceptions	The candidate doesn’t exist in the system

Use case	Add candidates information
Actors	Administrator database
Goal	Add new candidate information to database
Preconditions	Candidate doesn’t exist
Scenarios	Candidate information is provided, Candidate information is added
Exceptions	Duplicate candidate name, fake name (ie. Mickey Mouse, Harambe, etc.)

Use case	Vote for candidate
Actors	Voter database
Goal	Let candidate’s vote number successfully increase 1
Preconditions	Voter successfully login to system
Scenarios	Select candidate’s number/name, Click voting button

Exceptions	The candidate doesn’t exist in the system

Use case	Votes number for each candidate calculation
Actors	Administrator database
Goal	Figure out each candidate’s vote number
Preconditions	All voting finished
Scenarios	Sum up all votes with the same candidate number
Exceptions	Invalid votes

Use case	Stat winning candidate
Actors	Database
Goal	Figure out which candidate has largest vote number
Preconditions	All voting finished
Scenarios	Figure out each candidate’s vote number Compare vote number and find the

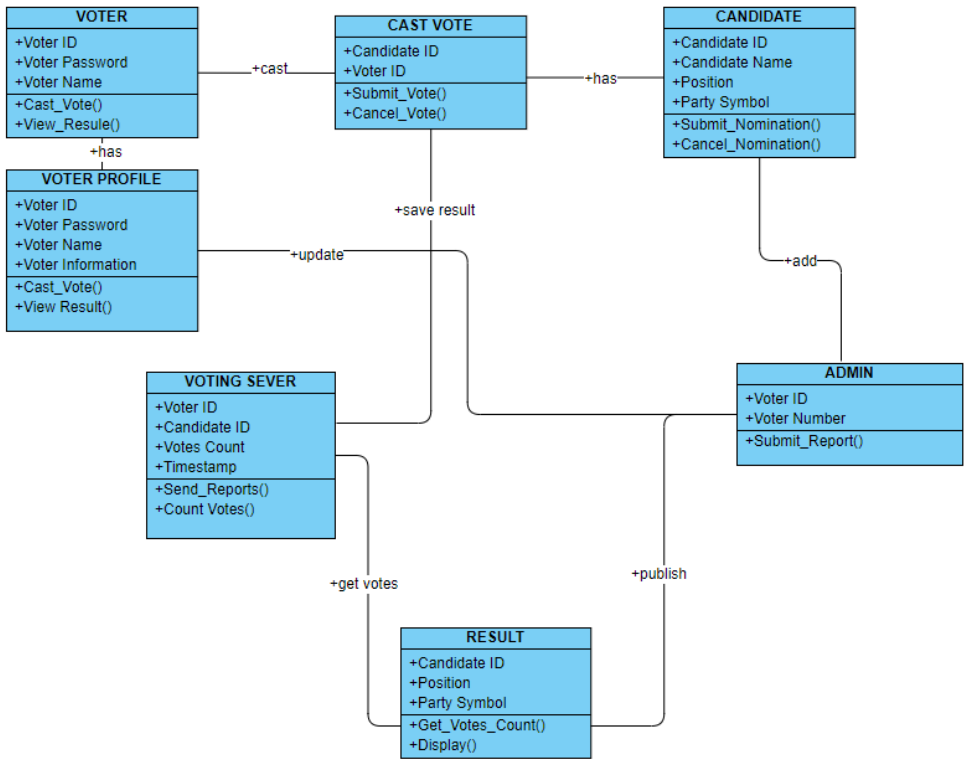
	largest one
Exceptions	Same vote number with greater or equal to 2 candidates

Use case	Total votes number calculation
Actors	Administrator database
Goal	Figure out total vote number
Preconditions	All voting finished Figure out all candidates' corresponding votes number
Scenarios	Sum up every candidate's vote number
Exceptions	The information on the vote is incomplete

Use case	Vote statistics by geography or political region
Actors	Database
Goal	Draw voting data diagram by geography or political region
Preconditions	Figure out every candidate's vote

	number Accurate candidate information
Scenarios	Draw rectangular coordinate system The x axis is geography or political region The y axis is the number of candidate
Exceptions	Candidates' information is incomplete

C. UML Class Diagram



2. Quality Plan

A. Quality goals and metrics

Product Quality	Quality Goals	Quality Metrics	Strategy
Availability	System should always be available for the all end user to use (rather than centralized polling sites).	System should be available to the user 99% of the time.	Downloadable .exe with user login and creation screens to enter program from any Windows Desktop. Proper server maintanence and balancing (stretch goal).
Reliability	System should allow users to successfully cast their votes for any candidate.	System should 100% correctly interpret user input and convert to a vote.	Write test cases to test edge cases.
Robustness	System should allow users to successfully cast their votes for any candidate.	System should 100% correctly interpret user input and convert to a vote.	Write test cases to test edge cases
Learnability	Users should be able to simple cast their vote	A user should be able to cast a vote in a few minutes.	Minimized the amount of clicks the user needs to make to cast a vote.
Usability	All users should be able to cast votes. Assuming system works properly	Have a user-friendly interface	Perform usability tests.
Efficiency	System should give feedback to users when voting and registering to vote.	Should return all candidates info in less than 1 sec	Perform unit tests to analysis retrieval time of candidates' info.
Security	System use only tally votes from verified users	Enforce a location rule that only allows votes to be casted if location requirement is met.	Integrate argon2 into password save/retrieve functions (hash function).

			Implement 2FA login for reduction of bot/repeat votes.
Portability	Service should be on a microservice architecture so individual components of the system can be upgraded individually	Individual services can be updated and switched without affecting other services	Put gui, backend, database, and other services on a .exe for simple installation and portability to all Windows systems.

Process Quality	Quality Goals	Quality Metrics	Strategy
Maintainability	The system should be easy to maintain	The backend API server will follow the MVC design patterns. Code needs to be documented well and follow design patterns.	Implementing MVC patterns for the backend API. Use Doxygen and comments to document code. And then follow React best practices for developing frontend.
Testability	System should be easy to add additional tests.	achieve 90% code coverage and 95% branch coverage	perform black and white box testing for frontend and backend. Perform integration testing for microservices. Also, Use mutation tools

3. Quality Priorities

1. **Reliability and Testability:** Is the most important quality feature because if the system is taking input from the voter but is taking the representative vote count wrong then the system is corrupt. This defeats the whole purpose of the project. This can be addressed by testing the system input.

2. **Security:** Security is the second most important quality component because we want to ensure that only USA citizens/green card holders are allowed to vote for presidential candidates and only in state people are allowed to vote local government. These requirements can be accomplished by integrating a microservice Auth0 that ensures secure access to voting and has functionality to enable login based on location and device registered.

3. **Availability:** Third most important quality feature will be availability. It's important to have the voting site up at all times to ensure fair voting. This is hard to accomplish setting up backup servers that load balance on the cloud to address blackouts, storms, to many users at a time and act.

4. **Useability and learnability:** This is important as well because all different age groups will be using the platform to cast their vote. So a user-friendly UI will be important to keep voters on the platform. This can be accomplished by minimizing the number of clicks it requires a user before they cast their vote. Also having a landing page that describes the who app will be essential to inform and train a novice user.3

5. **Portability and Maintainability:** This is a very important quality aspect because it can cause the most money to fix if overlooked. But in this application security and reliability have high importance. Nevertheless, having a microservice architecture will make the application more portable and easier to maintain if just a specific component of the website is going down. This can be accomplished by containerizing the application as a microservice architecture with an API gateway and following software design patterns when writing code.

4. Cost of Quality

Task name	Estimated effort	Implementation	Evaluation	Prevention	Rework
Project Planning	20	10	3	2	5
Database Implementation	35	18	4	4	9
UI Design/Implementation	65	40	10	5	10
Initial Project Development and Unit	90	45	15	5	25

Testing					
Quality Assurance-Integration/E2E testing	25	8	4	4	9
Acceptance Testing	25	13	6	0	6
Software Bug Correction	20	8	4	0	8
Total	280	142	46	20	72