Project Report on

**“SIMULATING A VOTING SYSTEM IN THE UNITED STATES”**

Lamar University

College of Arts and Sciences

Computer Science Department

CPSC 4360 Software Engineering

Fall Semester 2022

**SUBMIITED BY:**

Charles Blair

William Bohn

Parker Clemmons

Cameron Devenport

Oluwatomisin Egbewale

**Under the Guidance of:**

Dr. Sinan Kockara

**Table of Contents**

1. **Introduction**
   1. Purpose …………………………………………………………………………………………… 4
   2. Intended Audience and Reading Suggestions ……………………………………………………. 4
   3. Product Scope ……………………………………………………………………………………. 4
   4. References ……………………………………………………………………………………...... 5
2. **Overall Description**
   1. Product Perspective ……………………………………………………………………………… 5
   2. Product Functions ………………………………………………………………………………... 5
   3. User Classes and Characteristics ………………………………………………………………… 6
   4. Operating Environment …………………………………………………………………………... 6
   5. Design and Implementation ……………………………………………………………………… 6
   6. Assumptions and Dependencies …………………………………………………………………. 6
3. **Functional Requirements**
   1. Minimum Viable Product …………………………………………………………………...……. 6
   2. System Features …………………………………………………………………………... 7

3.3. Requirements ……………………………………………………………………………………… 7

1. **Other Non-Functional Requirements**
   1. Performance Requirements ……………………………………………………………...………... 8
   2. Safety Requirements …………………………………………………………………………… 8
   3. Security Requirements …………………………………………………………………………… 8
   4. Software Quality Attributes ……………………………………………………………………… 8
2. **UML Diagrams Analysis Models**
   1. Activity Diagram ………………………………………………………………………………… 9
   2. Class Diagram ………………………………………………………………………………….... 10
   3. State Diagram ………………………………………………………………………………….... 10
   4. Use-Case Diagram …………………………………………………………………………….... 11-12
3. **Implementation and Testing Screenshots** …………………………………………………………
4. **Project Outcome** ……………………………………………………………………………………
5. **Conclusion and Future Scope** ……………………………………………………………………...
6. **Introduction**

Voting is a fundamental process in a democratic system. The Merriam-Webster’s definition for voting is described as a formal expression of opinion as an indication of approval or disapproval of a proposal, motion, or candidate for office. The process of voting in the United States is straightforward. An eligible citizen registers to vote, studies the candidates and issues, looks up their polling location, then casts their ballot during the election. As technology advances so does the tools in our everyday life. Voting has become more automated with machines that have user interfaces to guide voters through their selection process. Yet with all new technology there will be some drawback and growing pains. If the machine/software are too sophisticated it can confuse a person with minimal computer skills. Also, security is another concern to make sure the machine is not hacked or manipulated compromising the integrity of the election.

* 1. **Purpose**

The purpose of this document is to inform the reader of the methodology the team went through during the traditional Software Development Lifecycle (SDLC) in designing this simulator. This report evaluates the feasibility of developing an open-source voting system using Python and highlights the risks, development options and the capabilities the team experienced to carry out its development.

* 1. **Intended Audience and Reading Suggestions**

This project is only a model for a voting system. It is meant to be used by students and faculty within the CS department for educational purposes only under the guidance of the course’s professor. This project is a useful case study to students and design tool for faculty.

How to navigate the main chapters of the document:

* **Section 1** summarizes the objectives, scope, and methodology of the feasibility of the project.
* **Sections 2 – 4** explains the team’s considerations from a system requirement, system build, and product delivery perspective.
* **Sections 5** provides context of the structure and logic of the design through diagrams with a focus on explaining functionality.
* **Section 6** is composed of screenshots of implementation of the software
* **Section 7-8** gives the results of the project through testing and recommendations offered after the assessment.
  1. **Product Scope**

This project is small scale voting system that is meant to replicate the voting system in the United States. The main purpose of the software is to ensure no voter fraud such as a registered voter attempting to vote more than once. It also aims to make the voting experience easier to understand by making function clear to end users.

* 1. **References**

AWS Public Sector Blog. “Spotlight on Elections: Voting, Cloud, and Alexa.” *AWS Public Sector Blog*, 5 Nov. 2018, https://aws.amazon.com/blogs/publicsector/spotlight-on-elections-voting-cloud-and-alexa/?did=psr\_card&trk=psr\_card.

AWS Public Sector. “How do political campaigns use AWS to improve voter experience Amazon Web Services.” YouTube,28 Sep. 2020, youtu.be/1Eh-i6AalHw.

Bell, Susan, et al. “STAR-Vote: A Secure, Transparent, Auditable, and Reliable Voting System.” *Microsoft Research*, USENIX Journal of Election Technology and Systems (JETS), Aug. 2013, https://www.microsoft.com/en-us/research/publication/star-vote-secure-transparent-auditable-reliable-voting-system/.

Porutiu, Horea, et al. “Build a Secure e-Voting App.” *IBM Developer*, 10 July 2019, https://developer.ibm.com/patterns/how-to-create-a-secure-e-voting-application-on-hyperledger-fabric/?mhsrc=ibmsearch\_a&mhq=voting.

1. **Overall Description**
   1. **Product Perspective**

The idea is to build a software system that provides a voting experience that has a high level of accessibility and usability. This must be accomplished to ensure the voting system is viewed as a trustworthy application that maintains the accuracy of the election. To execute this task, we must develop a prototype that is transparent and auditable end-to-end throughout the voting process. The simulation will use a command line user interface for simplicity. The programming language selected was Python, which the development team was the most familiar with. Also, SQL will be used for database purposes.

* 1. **Product Functions**

The two modes of operation administrator and voter. The administrator will manage candidates of the election and have access to voter profiles for security purposes. Voters will input their information to generate a unique ID and cast their vote from the listed options.

The “voter mode” process will work as follows:

* User approaches an Accessible Voting Device and inputs their name and address
* The system will then generate a unique voting ID number
* The voter will then input their ID for eligibility verification
* The appropriate ballot is then provisioned and presented to the user
* User makes selections and is asked to verify choices
* Upon completion the specific voter is removed from further eligibility and their votes recorded

The “administrator mode” process will work as follows:

* Admin logs into the Accessible Voting Device with their credentials
* Has the ability to view all voter information records for verification
* Reviews eligibility status, voter activity, and election reports
  1. **User Classes and Characteristics**

Admin class: Admin can get logged into the application by entering valid credentials. Admin

can add candidates. Admin can view the details of the votes and result. Voter class: Voters can need to get registered with the application by entering all the details in the registration form. The voter can get logged in to the application by entering unique username and password. The voter can fill all the details to give his/her vote.

* 1. **Operating Environment**

To use the setup software, the environment needs to have

* Operating system: windows XP and above.
* Coding language: Python
* Tool:
* Database: SQL
* GUI will be used for user input and output display.
  1. **Design and Implementation Limitations**

The information handled by the “Simulated Voting System” is of a sensitive nature. It will be using information like names and addresses. Also, who an individual chose to vote for will be recorded as well. So, it is of the utmost importance that voter identity is secure, and the votes recorded cannot be manipulated so the integrity of the voting process is maintained. With that in mind specific design constraints will be taken into consideration. The system will be intended to have the ability to only write vote data. Finally ballot information cannot be changed once a vote has been cats. The software will be constructed to take each of these possible design controls into consideration.

* 1. **Assumptions and Dependencies**

It is the developments team belief that the end user of this software will possess the bare minimum computer literacy skills. To successfully navigate the GUI, input their information, and making a candidate selection the user should at the least be capable of operating a smartphone. Also, it will be understood that it’s the responsibility of the voters to be familiar with the candidates on the ballot seeking election.

1. **Functional Requirements and System features**
   1. **Minimum Viable Product**

The software’s primary goal is to simulate a voting system in the United States. It will need to provide a

proper representation of the database corresponding to the candidates running for office and the U.S. citizens who voted. The other objective of the software is to make sure that only a single vote can be cast. The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

* Entities denote specific real-world items in application.
* Attributes indicate properties of an entity and relationships.
* Relationships connect entities and represent meaningful dependencies between them.
* Normalization is to reduce redundancy which means that information is to be stored only once.

\*See Entity-Relationship diagram below

Diagram

Description automatically generated

* 1. **System Features**

A well-designed voting system will allow accessibility, integrity, and security during the voting process. Administrators access to information necessary during the voting process such as reports and display results and maintain/produce an audit trail. For the voters, the system should be able to record their address, create a unique ID, define the ballot, and cast votes.

* 1. **Requirements**
* Display list of available candidates
* Generate a unique voter ID number
* Store voter input into a database
* Administrator access to databases for validation
* Generate a report quantifying voting results
* Maintain voter credibility

1. **Other Non-Functional Requirements**
   1. **Performance Requirements**

The program is expected to have a sensible short-term response. The end-user should receive a reaction from their interaction with the software within 3-5 seconds. The performance of the systems varies depending on the mode its operating in.

* 1. **Safety Requirements**

If the Admin finds any security issues in the system, it should be able to shut down shut down the system and block all server communication immediately to save votes have already been cast.

* 1. **Security Requirements**

All passwords and unique ID numbers generated or received must be stored in a database in an encrypted form. The software will only accommodate one user at a time. No simultaneous use of the system by multiple voters, administrators, or a combination of the two shall be allowed.

* 1. **Software Quality Attributes**
* Usefulness: a clear and easy to navigate interface for operator usability.
* Availability: an online system allowing for votes to be cast from any place.
* Correctness: accurately maintain voter eligibility record and record voter submissions.
* Efficiency: track election updates and generate precise reports of results.

1. **UML Diagrams and Analysis Models**

These diagrams will help explain the though process behind the functionality of the system as it was being developed. UML or Unified Modeling Language is a standard approach to visualize the design of a system. These diagrams offer a way to visualize a system's architectural blueprints in a diagram, including elements such as:

* Any activities
* Individual components of the system
* How components and or interfaces interact with other components
* How the system will operate

There are 14 types of different UML style diagrams to help highlight the aspects of a developer’s design. For this project, we will be displaying a UML class, activity, state, and use-case diagrams. A class diagram is a drawing used to construct and visualize object-oriented systems. It is static structure diagram that describes the structure of a system by showing the system's classes, operations, methods, and their relationship among objects. Activity diagrams are interaction diagrams that detail how operations are carried out. It captures high-level interactions between user of the system and the system, between the system and other systems, or between subsystems. State diagrams show the various states of an entity. State machine diagrams can also show how an object responds to various events by changing from one state to another. A use case diagram helps us design a system from the end user's perspective. It is an valuable method for communicating system behavior in the user's terms by specifying all externally visible system behavior.

* 1. **Activity Diagram**

Diagram

Description automatically generated

* 1. **Class Diagram**

**Diagram

Description automatically generated**

* 1. **State Diagram**

**Diagram

Description automatically generated**

* 1. **Use-Case Diagrams**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

1. **Implementation and Testing Screenshots**

**(\*add files)**

1. **Project Outcome**

The development team successfully created a working software application offering a functional polling system to assist in in the voting process during U.S. elections. This project was written in the Python programming language, utilized SQL for the database and employed (GUI) graphical user interface to present and record user input. There was a registration page created for new voters only. A separate page for both registered voters and administrators to log in for there on individual purposes. Voters use the unique number ID generated to verify voter eligibility and to access the candidate list and cast their votes. Administrators will use their credentials to open registered voter lists, candidates, and view voting results. All votes cast are being stored in a table in a database. Once all votes are collected, they are calculated for accuracy, and simple report displaying the results are generated.

1. **Conclusion and Future Scope**
   1. **Conclusion**

This project satisfies all the necessary requirements to be considered a viable system for voting during elections. The software operated based on the principal of a machine marked paper ballot record casting and counting process. Additionally with voter fraud prevention in the form of unique ID’s it further builds trust within the voters about the systems integrity. Having a separate GUI and log in credentials proves valuable in displaying the security and versatility of the simulator. Overall, the voting simulation system presented offers a low cost and high feasibility alternative to other voting methods.

* 1. **Future Scope**

Although this is just a college level project only to be used in a controlled environment it shows that the system is a modest but practical option for real world applications given the opportunity for further development. This prototype is currently running on a single computer. It could be expanded upon by using web-framework to develop a dynamic web application. Given the time the use of CSS or HTML languages could be used to create a more elaborate web-based GUI. Other improvements could be focused on design a system which can be easy to use and will provide security and privacy of votes on acceptable level by proper authentication and processing section. With time the and further development this application can provide more accessibility, accuracy, and lower human or mechanical error during the voting process.