Polling System

Final Project

By: Josh Scott

# SWOT

# Strengths:

* Accuracy – A polling system must take accurate measurements of all votes and candidates.
* Ease-of-use – A polling system must be easy to use so the users can vote quickly and accurately.
* Secure – An electronic polling system allows the government/Admins to keep a secure count of the votes that can’t be tampered with.
* Hacking/altering votes – An electronic polling system that was created with security in mind should be much safer than paper votes that could be ruined by something as simple as water.
* Vote count – votes get added up automatically.

# Weaknesses:

* Expenses – Creating an electronic software along with the hardware required will be expensive.
* Hacking – An electronic polling system would be hackable which could allow for voter fraud and fake elections.
* Tech trouble/Downtime – Software and hardware can have many unforeseen issues that pop up.
* Crazy people who think the votes don’t count – Just adding this one because it happened just a few years ago! It speaks for itself.

# Opportunity:

* Increased voter turnout – More people might be willing to join in on voting if they could vote from home.
* Efficiency – There would be no need to hand count paper ballets because the computer can count very well!

# Threats:

* Voter Fraud – People could attempt to change the vote in the system if they felt it was wrong.
* Resistance to change – As people have shown, we are typically afraid of change, so going from paper to electronic will spark fear for many people.
* Security threats – Enemy States or domestic threats may try to hack into the system since it may be connected to the internet.

## Water Fall Model

# Requirements Analysis

The purpose of our polling system is to gather the votes from a local government election. The municipality will be going around on June 1, 2023 to vote for their local mayor. We will need the ability to add up to 6 candidates. The results can be viewed by the voters and the admin staff. The system must be fast, responsive and provide no issues to the users.

# System Design

For this system, we are going to use the “C” language because it has excellent performance and reliability when implemented correctly. The system design phase will require multiple moving parts, such as Hardware, Software, Networking, Databases, and Admins. We will need to think about security internal and external. We will also need to prepare documentation for Admins and Voters so they’re able to use the machines.

# Implementation

During this phase, the team will begin coding and development of the polling system, conducting unit tests after every new feature and ensuring they’re functioning as intended with the rest of the system. Upon completion of the project, the team must conduct multiple system tests ensuring that everything works properly, and no bugs arise. Documentation is required while coding your project to ensure any developers who need to perform maintenance later can understand what’s going on.

# Integration Testing

During the Integration and Testing phase, we’ll go through the code testing each function to ensure security, and reliability. Some tests we may consider are, attempting to vote while logged in as an admin, a voter trying to add or change one or more of the candidate’s names, or have the admin be able to change the vote count.

# Deployment

After we’ve completed all other steps, we need to start thinking about deployment of the product. During deployment, we will need to train users, and admins on how the system works and what to do in case of mistakes or accidents. On the side of each Polling System, or if they’re voting at home, then on the web page, they must have the rules of voting clearly stated and the penalty for fraudulent activity.

# Maintenance

For Maintaining the systems, the company will be available Monday to Friday so assist with any issues. Outside of those hours, Admins will need to be trained on what to do in which situation and how to handle it.

# Project Planning

# Scope

The purpose of this project is to allow the local municipality to be more efficient with their voting allowing for online voting from within the municipality to select the next and ongoing mayors of the city. We expect 85% of the city to attend the voting, we are looking at around 25% of that will be using our new voting system. Users will need to log in to their accounts and cast one single vote. Admins will be required to add the candidates 12 hours before the election to set to begin while another admin is there to witness and authenticate the candidates.

# Objectives and Goals

Our main goal with this system is to increase voter turn out and ensure the vote count is accurate at all times. The previous few years have been tough which led to a major decrease in voter turn out of around 60%, we’re hoping to see voter turn out increase by 25% from the previous years, a large portion of these votes are expected because of this system. We plan to guarantee 99% accuracy, with a less then 1% false voter rate considering user error, system downtime, and public interference.

# Schedule

The development team will follow the wall fall SDLC laid out above, we will consider each step of the SDLC the next milestone with the final milestone being launch. Launch is scheduled for April 6th,2023. We should account for any issues that might pop up.

# Resources

With this project, we were given an unlimited budget for everything needed to complete the task. We will need to hire a project manager, 3 additional developers, 2 product testers, and the admins to run the system. On election day, there will need to be one developer on site for the vote in case of issues and we will need to hire 10 support staff to manage the voting event itself.

# Test Case Scenarios

# Admin tests

### 1.Add Candidates

## Input

Enter your role(admin/user): admin  
Enter your username: admin123  
Enter your password: password123

## Admin action

1: add candidate – Josh

## Output

Expected output: Candidate successfully added!

### 2.Adding existing candidates

## Input

Enter your role(admin/user): admin  
Enter your username: admin123  
Enter your password: password123

## Admin Action

1: add candidate – Josh

## Output

Expected output : Candidate already exists!

# User Tests

# 1.Attempting to vote twice

## A. Input

Enter your role(admin/user): user  
Enter your username: user123  
Enter your password: userpass123

## B. User action

1: vote for candidates – Josh

# User Action

1: vote for candidates – Josh

## Output

Expected out: You can’t vote for 2 different candidates in the same vote!

# Test Case Diagram

Diagram

Description automatically generated