EECS 582 TEAM 14

FINAL PROJECT DESIGN

ElectWise

February 7, 2020

Zach Davis
Tim Fox
Hirsh Guha
Benjamin Streit
Tanner Strickler

SYNOPSIS

Web-based voting application providing complete anonymity and security, as well as real-time results and analysis. ElectWise is to be used in any polling scenario where transparent and trusted results are vital.

DESCRIPTION

This project is addressing a direct need that our election process currently faces. Our current voting procedures are outdated and inefficient. ElectWise fixes this problem by leveraging modern technology. Our election processes also deal with the issue of voter turnouts. ElectWise will deliver ease of access and dramatically increases voter turnouts. Our product delivers a robust security stack to ensure election integrity.

Creating such a platform is as versatile as it gets, from private groups to the federal elections, ElectWise provides speed, efficiency, usability, and ease of access at every stage in the election process - assigning candidates, voting, etc.

The deliverable should be a fully developed platform that allows a user enter their voter credentials and interact with all available elections, secured by their hashed personal information to create a unique voter ID. Their vote information will be hashed and entered into an immutable and secure database, upon which the voter will receive a unique and private vote reference number. As accuracy is of the utmost importance, it is vital that voters are capable of ensuring that their vote was counted correctly. In order to ensure a “paper trail”, a voter can query an election’s results using their provided vote reference number. Votes will be processed and able to be visualized periodically as votes are collected.

MILESTONES

Fall

(i) (November 23) Administrator and Voting Dashboard Mock-ups.

(ii) (November 23) Configure MySQL Database.

Spring

(i) (February 6) Instance MySQL DB, required tables, and begin connecting to front-end.

(ii) (February 15) Implement unique voter hash.

(iii) (March 15) Implement vote reference hash.

(iv) (April 15) Implement Spark MySQL.

(v) (May 1) Implement Statistics Dashboard.

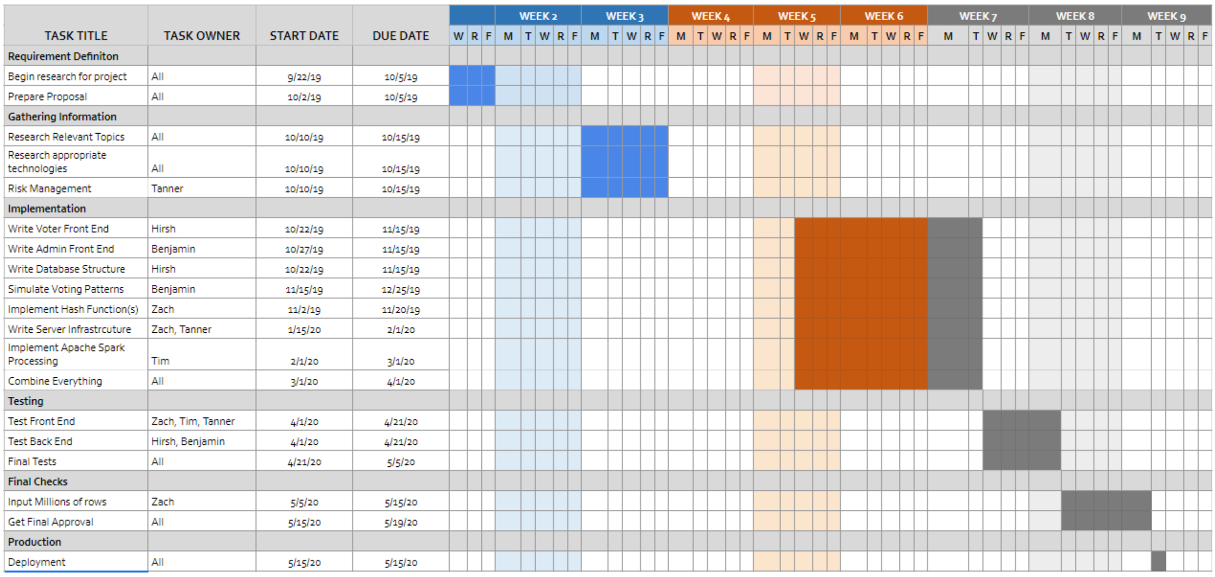
BUDGET

Item	Projected Cost
Amazon Web Services (AWS) Free Tier	$0.00 USD per month of uptime

WORK PLAN

Member	Role
Zach Davis	Server Infrastructure, Data Processing, Documentation
Tim Fox	Server Infrastructure, Data Processing
Hirsh Guha	Front-end services, Database Management, API Management
Benjamin Streit	Front-end services, Server Infrastructure, API Management
Tanner Strickler	Security, Data Processing

Gantt Chart



TASK TITLE	TASK OWNER	START DATE	DUE DATE
Requirement Definiton			
Begin research for project	All	9/22/19	10/5/19
Prepare Proposal	All	10/2/19	10/5/19
Gathering Information			
Research Relevant Topics	All	10/10/19	10/15/19
Research appropriate technologies	All	10/10/19	10/15/19
Risk Management	Tanner	10/10/19	10/15/19
Implementation			
Write Voter Front End	Hirsh	10/22/19	11/15/19
Write Admin Front End	Benjamin	10/27/19	11/15/19
Write Database Structure	Hirsh	10/22/19	11/15/19
Simulate Voting Patterns	Benjamin	11/15/19	12/25/19
Implement Hash Function(s)	Zach	11/2/19	11/20/19
Write Server Infrastrcuture	Zach, Tanner	1/15/20	2/1/20
Implement Apache Spark Processing	Tim	2/1/20	3/1/20
Combine Everything	All	3/1/20	4/1/20
Testing			
Test Front End	Zach, Tim, Tanner	4/1/20	4/21/20
Test Back End	Hirsh, Benjamin	4/1/20	4/21/20
Final Tests	All	4/21/20	5/5/20
Final Checks			
Input Millions of rows	Zach	5/5/20	5/15/20
Get Final Approval	All	5/15/20	5/19/20
Production			
Deployment	All	5/15/20	5/15/20

Preliminary Project Design

Our software will allow administrators to create accounts, load voter databases, and create elections. In creating an election, a plurality of options will control the extent and details that voters will have to input. For instance, the United States federal government will provide secure voting registration data, against which we would validate voting attempts, along with candidates and candidate data. From there, the system will take care of the rest, allowing administrators to sit back, relax, and watch real-time statistics roll in from voters. Then, if the administrator has predetermined a start and end date, as well as voting times during the day, these will automatically be taken into consideration, with the election going offline during off hours or before and after the election has been selected to begin and end.

At a federal level, ElectWise's public facing landing page will have voters enter whatever personal information is required for a particular election, such as their first and last name, social security number, and date of birth for a federal election, for example. A voter may optionally input their email address if they wish to be emailed a receipt of their recorded vote. ElectWise will hash their personal information and compare it to the hashed personal information in the registered voter database. If there is a match the voter will continue to the voting dashboard. If there is not a match the voter will be redirected to a page where they can gain information regarding how to register to vote.

The voting dashboard will display the election information supplied by the administrator. This will result in the voter seeing a formatted list of election cards containing the election's information, as well as a search bar. The search bar allows for queries on any vote or election reference number to observe the requested vote information or the results of an election. When an election card is selected, the voter will be presented with a vote card containing the possible options for the selected election, as well as information regarding each possible option. Bucketed real-time statistics and analytics regarding the selected election will also populate on the election card should the election be in an online and active state. Upon submission of a vote card the voter will be presented with a review card where they will review their voting selections with options to confirm or cancel the vote. Selecting 'cancel' will return the voter to the voting dashboard. Selecting 'confirm' will present a confirmation card containing their voting receipt. The confirmation card will be emailed to the voter if they have provided their email address, otherwise it will be encouraged that the voter print out the voting receipt as the voting receipt contains the vote's reference number associated with the voter's private key. This reference number acts as a paper trail and allows the voter to look up their vote at a later time and observe their selection, while also preventing other voters from knowing the selections of other voters.

This front-end infrastructure will be built using React, a JavaScript library for building user interfaces using encapsulated components composed together to render complex user interfaces. The use of React will allow for the creation of an easily reusable and extendable code base that can be deployed across multiple platforms. By writing responsive user interfaces, ElectWise can be used comfortably whether displayed on a desktop monitor, laptop, or smartphone. In addition, should the roll-out of a native mobile application prove fruitful down the road, a native application can be deployed that simply displays the website scaled down to the smartphone's display using responsive CSS. Using this approach to the front-end software design of ElectWise results in a single code base that is written once and deployed anywhere, meaning that ElectWise does not need to worry about scaling up the front-end to meet increases in user demand.

A vote submission will constitute an insert into a MySQL database. This will be handled by an Express server. Express is a web application framework for Node.js designed for building server-side web applications and APIs. Additionally, Apache Spark, a general-purpose cluster-computing framework, will be used to increase the rate at which inserts can be performed by up to ten fold during times of intense load. The MySQL database will use the following row schema:

ID	Hash	Address	Flag	Etc.

To ensure the anonymity and security of ElectWise, absolutely no vote will have any identifying information that could be tied back to a voter without the privately provided vote reference number that is given solely to that specific voter. To achieve the lack of identifying information stored with a vote, a cryptographic hash will be provided the voter's required personal information, such as social security number and date of birth as inputs, and its output will be used as the reference hash stored with the vote. This reference hash can then only be traced back to a voter using the voter's private key.

Ref Hash	Votes	Info	Etc.

ElectWise will also provide bucketed real-time statistics and analytics on each election taking place on the platform. Our system will leverage Apache Spark to allow queries to be run on the voting data incredibly efficiently and return accurate and informative statistical analysis back to the election card's statistics dashboard on the front-end. The back-end infrastructure will be hosted on Amazon EC2 instances using Express to connect React with MySQL. Using such a back-end data manipulation configuration will allow ElectWise to provide real time election progress results and analysis, while also having the headroom necessary to scale up resources during times of heavy load and scale down resources during times of decreased load.

The end result of the use of the ElectWise application will be a database of entirely anonymous voting data regarding various elections as well as statistical analysis on said voting data, all made completely public in real-time to any valid voter in said elections. Providing a transparent and efficient means of capturing votes from populaces ranging from that of small cities to that of entire nations.

Ethical Issues

According to the National Conference of State Legislatures, the voter data in the State of Kansas can be accessed by anyone in the general public and only the date of birth and driver's license number are anonymized. As such, our system would actually be more secure than how the State of Kansas currently hides/provides voter registration information to the public. Because of this, the key ethical concern, that of revealing private voter information, is not something we need to be overly concerned with, as it would not be handled by ElectWise, but rather by the administrator of the election.

Intellectual Property Issues

There are no foreseeable intellectual property issues at this time with the technologies being used. Any data inputted into our system would belong to us, and would only be used for graphical and statistical modeling purposes.

Change Log

10/23/19, Project Name: Changing project name to ElectWise to better communicate the functionality of the project.

10/23/19, Project Synopsis: Blockchain technology will not be used, adding information regarding analysis of voting results.

10/23/19, Project Description: Voters will not login to an account, instead will enter voting credentials to ensure voting registration and will be directed to available elections for said voter. Votes will not be entered into a blockchain, opting for an immutable and secure database instead.

2/6/20, Project Description: Small changes in language to provide a smoother description of the project.

2/6/20, Project Milestones: Milestones modified to better reflect our current timelines.

2/6/20, Project Budget: Hadoop was the sole source of our initial predicted budget, so the removal of Hadoop eliminates these costs.

2/6/20, Project Design: After reevaluating the scope of our project, Hadoop and Blockchain were deemed unnecessary. As such, they were removed from the project design. Discussion of Express for back-end communication added.