

E-Vote

**Version 6.0**

**Group 1**   
Clayton Snyder  
He Cao  
Hoi Kin Cheng   
Yingjun Fan  
Milo Hegamin  
Misha Ward

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 01/27/2019 | 1.0 | Added Business Proposal, SRS, and Domain Diagram | Hoi Kin Cheng, Milo Hegamin, Yingjun Fan, Clayton Snyder |
| 02/02/2019 | 1.01 | Renamed “Election Item” to “Race” in SRS to keep naming consistent with Domain Diagram | Hoi Kin Cheng, Clayton Snyder |
| 02/03/2019 | 2.0 | Added Use Cases which includes Use-Case Diagram, Use-Case Descriptions, and Miscellaneous Design Artifacts | Misha Ward, Clayton Snyder, Hoi Kin Cheng, Milo Hegamin, Yingjun Fan, He Cao |
| 02/17/2019 | 2.1 | Updated report for comments, added activity diagrams based on use cases. | Misha Ward, Clayton Snyder, Hoi Kin Cheng, Milo Hegamin, Yingjun Fan |
| 02/24/2019 | 3.0 | Added Business Proposal and Wireframes to report. | Misha Ward, Clayton Snyder, Hoi Kin Cheng, Milo Hegamin, Yingjun Fan, He Cao |
| 03/03/2019 | 4.0 | Added Robustness Diagrams and Sequence Diagrams | Misha Ward, Clayton Snyder, Hoi Kin Cheng, Milo Hegamin, Yingjun Fan, He Cao |
| 3/14/2019 | 5.0 | Added Test Case and E-Vote Logo | Misha Ward, Clayton Snyder, Hoi Kin Cheng, Milo Hegamin, Yingjun Fan, He Cao |
| 3/19/2019 | 6.0 | Final updates, response to professor and developer peer review, added Class diagrams and Sequence Diagrams, Developer Peer Review Approval. | Misha Ward, Clayton Snyder, Hoi Kin Cheng, Milo Hegamin, Yingjun Fan, He Cao |

**Table of Contents**

[**1 Business Proposal**](#_n31048acbg60) **5**

[1.1 Executive Summary](#_wdquiik6w4wd) 5

[1.2 Stakeholders](#_m03128a2bq95) 7

[1.3 The Problem](#_vd02x25g91vd) 8

[1.4 The Solution](#_wn2rtk7fmxle) 8

[1.5 Financial Summary](#_ggisjrvwpw5a) 9

[1.6 Risks](#_5dff40vmaapj) 10

[1.7 Conclusion](#_2pjwrmumusph) 11

[**2 Software Requirements Specification**](#_v9vycciz4c22) **12**

[2.1 Functional Requirements](#_p93bm8pvavrf) 12

[Portal Interface Requirements](#_zem369yi04cf) 12

[Landing Interface Requirements](#_p8j4409bo1u1) 14

[Election Interface Requirements](#_8jvkhjyq0r45) 14

[Voting Interface Requirements](#_4mb9hf9ooe7r) 15

[Back-End System Requirements](#_ky1qojid89mo) 16

[2.2 Non-Functional Requirements](#_7nkntok09fux) 16

[Usability](#_odch5v8bpsil) 17

[Reliability](#_n7q0psx0qz8p) 17

[Performance](#_ims452mshq5m) 17

[Availability](#_8p3pkcf6a3zw) 17

[2.3 Business Team Approval](#_6lbykchpeorx) 18

[**3 Architecture Design**](#_4prq3s2shc32) **18**

[3.1 Domain Modeling](#_q31lmr7smonq) 18

[**4 Use Cases**](#_3spk016h7ti6) **19**

[4.1 Use-Case Diagram](#_xe4oxw3u40db) 19

[4.2 Use-Case Descriptions](#_goc6i3zge59n) 19

[Start session (Voter login)](#_g5dar4rsxynw) 19

[Vote in a race](#_lmxzccogwgj5) 20

[View info about upcoming races](#_gfosil2g8cyv) 21

[View info about a specific candidate](#_hfke7vns63dt) 21

[Change password](#_hegdpu4y4fn5) 22

[Change email](#_z2y36v6fkfk6) 22

[**5 Activity Diagrams**](#_rebxzu9zif08) **23**

[5.1 Start session (Voter login)](#_8f2ikp4usy4) 23

[5.2 Vote in a race](#_nvupc2m5jm7x) 24

[5.3 View info about upcoming races](#_3uyn6y3z0ia) 25

[5.4 View info about a specific candidate](#_b0wm0oqwqwn9) 26

[5.5 Change password](#_gem737s71upm) 27

[**6 Wireframes**](#_p8j9qd9jhexp) **29**

[6.1a Login](#_yp0aio950fzc) 29

[6.1b Reset Password Page](#_75phjkntyxwi) 30

[6.1c Registration Page](#_b0twtey35kwo) 31

[6.2 Landing](#_p3jmselvxfru) 33

[6.3 Election](#_37ln6tm5en3o) 35

[6.4 Voting](#_gwdm52m0um8j) 37

[**7 Robustness Diagrams**](#_otpw9vvht43j) **39**

[7.1 Start session (Voter login)](#_jt3lqzxe7ppi) 39

[7.2 Vote in a race](#_gtt3us6xo05u) 40

[7.3 View info about upcoming races](#_65pvh245x4sw) 41

[7.4 View info about a specific candidate](#_y76i4eo66zda) 42

[7.6 Change email](#_yxn3ehz759xv) 44

[**8 Sequence Diagrams**](#_tazaybbbxp07) **45**

[8.1 Start session (Voter login)](#_vf4q3g2io387) 45

[8.2 Vote in a race](#_16j51slzearb) 46

[8.3 View info about upcoming races](#_ulbtlmjs3ih) 47

[8.4 View info about a specific candidate](#_sq05jvdouty6) 48

[8.5 Change Password](#_x5c7llfrte8r) 49

[8.6 Change email](#_pd6dqewen85) 50

[**9 Test Cases**](#_em46dcxq6tvx) **51**

[9.1 User login](#_b0tv63l6f166) 51

[**10 Class Diagram**](#_1tr7gigxkrc8) **53**

[10.1 Class](#_pphesxvk1k2n) 53

[**11 Developer Peer Review Approval**](#_omcl0znm2fvy) **54**

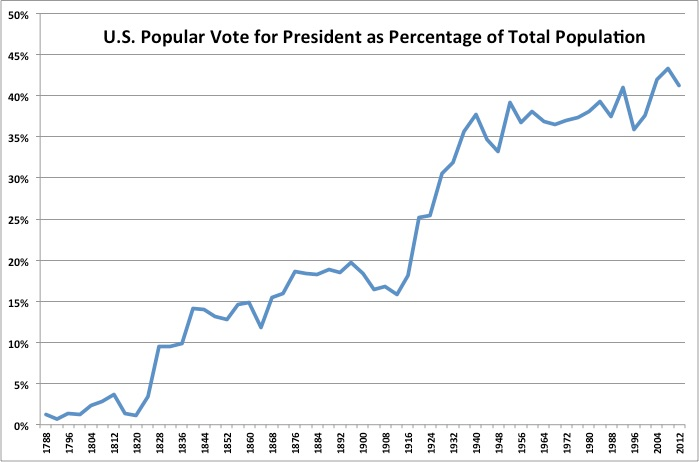
# 1 Business Proposal

## 1.1 Executive Summary

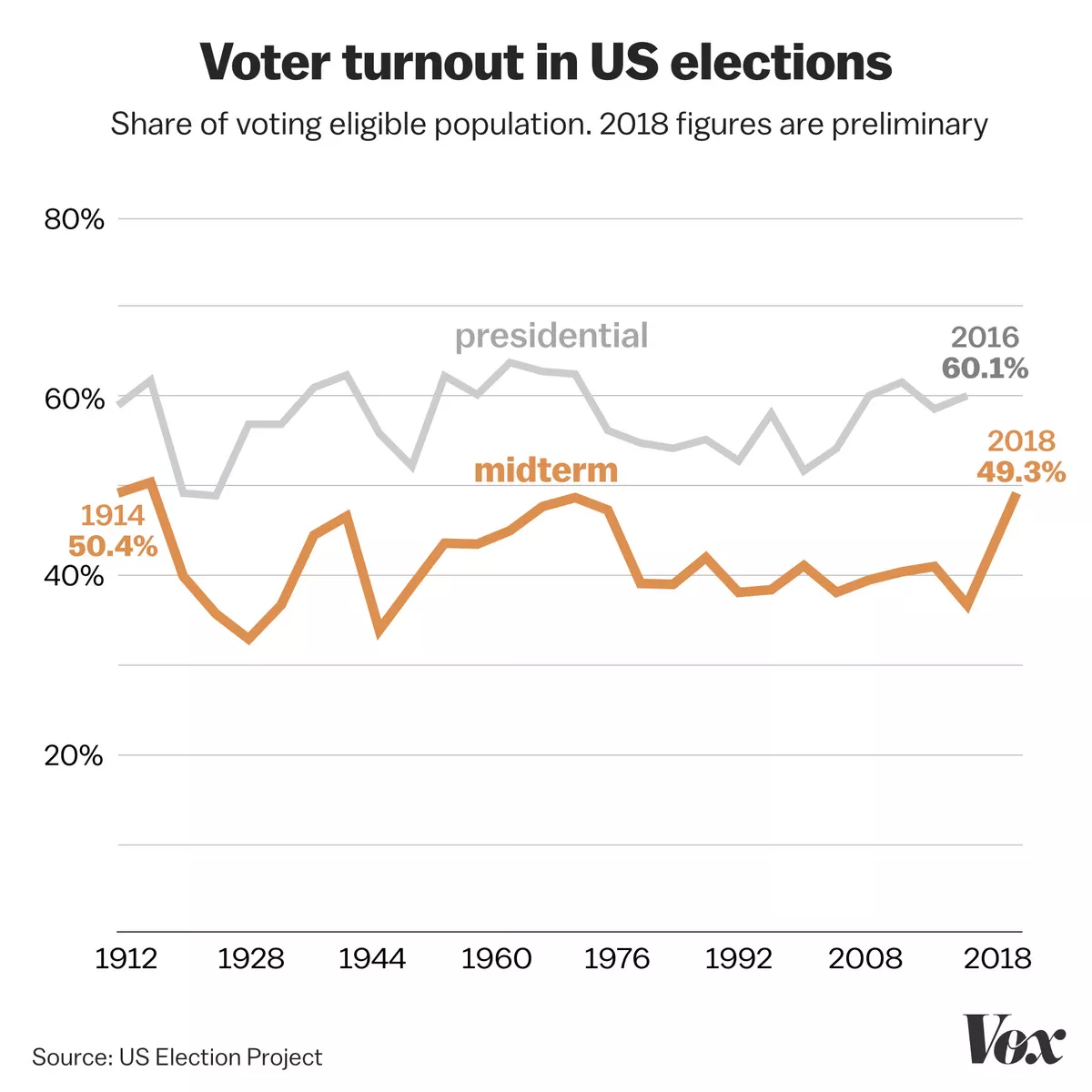
America is one of the largest democracies in the world and has continually evolved over the course of the past two centuries to expand voting rights and access for its citizens. With the development and adoption of the internet, it only makes sense to utilize this technology to allow for further voter access and improve turnout. This summary will break down the problem and why stakeholders (the state of Washington) are willing to pay for this solution, how the venture plans on solving the problem, why this venture is able to solve the problem, and brief economic and risk summary of the project.

Although the United States of America is one of the largest economies and influential democracies in the world, voter turnout has been at historical lows. Ironically, throughout American history, the amount of people eligible to vote over the total population has increased in percentage terms seen in *Figure 1.1*. Although many more Americans are voting as a percentage of total population, many eligible Americans are not voting. According to the news agency Vox, voter turnout in the United States has stagnated over the past few decades as shown in *Figure 1.2*. When comparing the United States to other developed countries that vote, the low turnout becomes even more evident. *Figure 1.3* below shows the United State’s position in the world in terms of low turnout.

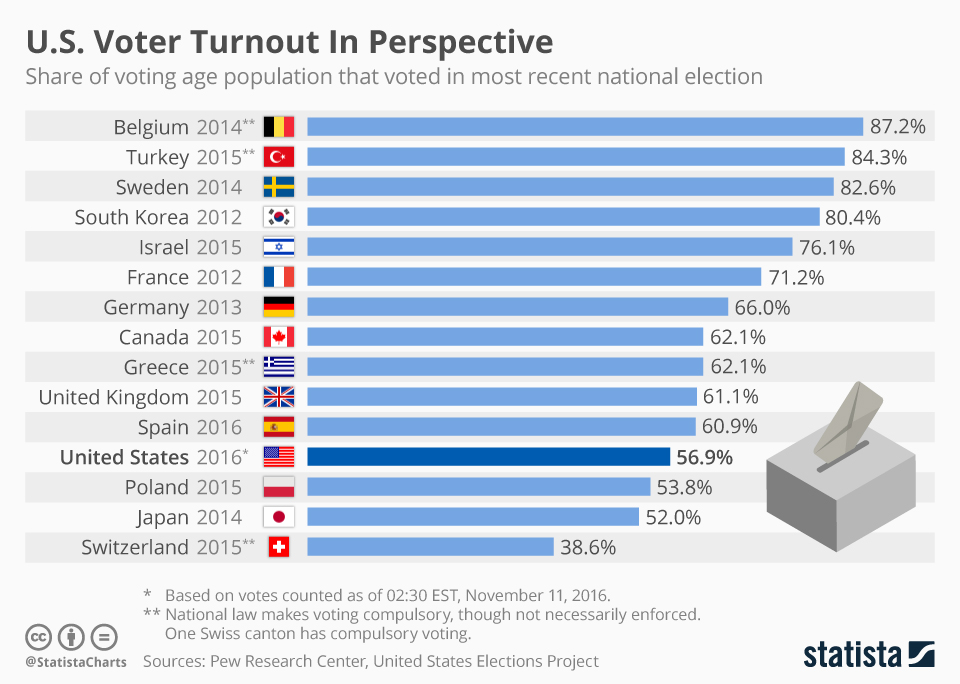
**Figure 1.1; U.S. Popular Vote as Percentage of Total Population**



**Figure 1.2: Voter Turnout in the U.S.**



**Figure 1.3: U.S. Voter Turnout Global Comparison**



Stagnating voter turnout poses a problem to a country where the political system is built on voter participation. Currently, the views of millions of people are not being heard including poorer populations, minorities, millennials (individuals born between 1985-1999), and generation Z (individuals born between 2000-present). By incorporating technology into the democratic process, the voter turnout will increase while also decreasing the cost of paper ballots and the environmental impact of elections. Stakeholders such as voters, government officials, and politicians should all recognize the importance of these benefits of electronic voting.

Specifically, the E-Vote website will be able to solve this problem by providing voters a way to vote online. By doing so, the voters would be able to conveniently access their ballot and vote for candidates from federal, state, local elections and local ballot initiatives. Studies have shown that by providing a more convenient way to vote, many younger and ethnic minority voters would participate and increase their turnout. This venture would solve the problem due to the scalability of modern cloud solutions and the security that the cloud provides.

Financially, the project will be funded by the State of Washington with a cost plus fixed fee contract. This contract will allow for low risks for development while maintaining a level of profitability that is needed to complete the task with the quality that is needed. Finally, the risks of the project are mostly around the security of the ballot voting, the information of the voters, and hacking or information leaks. E-Vote will field some of the most rigorous standards for encryption and work with leading experts to design and build a system that mitigates these risks.

## 1.2 Stakeholders

**Owner:** The State of Washington, for whom the E-Vote system is being developed. It will be a partner of the application. They will be responsible for making the E-Vote server(s) available for communication between their server and the voting server.

**Voters:** They are the main users of the application. Eligible voters can cast ballots through the E-Vote service. They will pass their information to the business team.

**Political candidates and organizations:** Individual candidates and the organizations supporting ballot measures will have information regarding their positions accessed and displayed. They are responsible for providing voters with some information about the candidates. They have an interest in E-Vote because it provides their information for votes, so voters don't have to constantly be asking candidates about their information.

**Client:** Communicates with the Product Manager to create and update product requirements and validate specifications. The client is the team working for the State of Washington who is in charge of communicating with the contracted architecture team to establish requirements, specifications, and acceptance criteria.

**Product manager:** They are responsible for ensuring the project development goes according to schedule and budget and that the project satisfies the client's requirements. Elicits requirements from the client. Creates product specifications in collaboration with the development team and validates those specifications with the client. Communicates with the development team (the system architect, UX developer, and back-end developers) to ensure that the product being developed is meeting the specification.

**System architect:** Designs the overall construction of the system (i.e., what services/modules the system is composed of and how they communicate). Communicates with the Product Manager to ensure that the architecture will succeed in meeting functional and non-functional requirements of the product.

**Fullstack developers:** Works on development of all parts of the product. Collaborates with the product manager to implement the front-end of the product to specification. Designs and implements the back-end components of the product as outlined by the system architect.

## 1.3 The Problem

**Primary problem low voter participation:**

Voter turnout is negatively affected by difficulty in voting. The voter turnout rate in the United States has been at a historical low and has stagnated over the past few decades. Many voters cannot go to in-person polls on election day or are not sufficiently motivated enough to fill out and mail a mail-in ballot. This project sets out to address this by moving the voting process to the web and to mobile devices, thus making it significantly easier to vote in any given election.

**Additional problems to be solved by this product:**

Lack of informed voting due to low voter exposure to candidate information. E-Vote can be designed to include the basic information on ballot measures already included in ballots, and also include candidate party information and links to external information sources for candidates. Excessive resource usage caused by current election systems (e.g., facilities/workers for polls, paper for ballots, etc.). As E-Vote becomes more popular, less of these resources are necessary.

## 1.4 The Solution

This project sets out to address this by moving the voting process to the web, thus making it significantly easier to vote in any given election. E-Vote will be designed to include the basic information on ballot measures already included in ballots, and also include candidate party information and links to external information sources for candidates.Studies have shown that by providing a more convenient way to vote, many younger and ethnic minority voters would participate and increase their turnout. This venture would solve the problem due to the scalability of modern cloud solutions and the security that the cloud provides. Financially, the project will be funded by the State of Washington with a cost plus fixed fee contract. See subsection 1.5 for details.

## 1.5 Financial Summary

Solving the problem of voting is a critical issue that E-Vote will solve in an economically feasible way. Projections show that the project will produce a strong profit margin and positive benefit to cost ratio. Cost estimates can be seen below in *Figure 1.4* while *Figure 1.5* details the cost-benefit analysis. A detailed analysis of each analysis can be found below.

**Financial Cost Analysis**

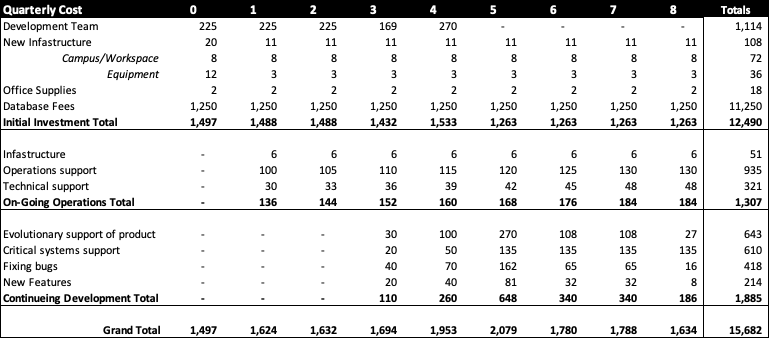
Due to a project this scope, *Initial Investment* costs were estimated based on the expectation of 6 full stack developers with the average total package for those developers being at $150,000 per year. Infrastructure such as office space and equipment were roughly $28,000 at the beginning due to move in costs, buying laptop and equipment and office deposits/renovation. After the first quarter, these cost become fixed at $11,000 per quarter.

A major expense is predicted to be various databases like the voter registration database that we will use and connect APIs to. This cost is expected to be $1.25 million a quarter for all the various databases that we will utilize. The final estimated cost for the initial investment for E-Vote is expected to total $12.49 million over 9 quarters.

*Ongoing Operational* costs start after the first quarter when code and prototypes are being developed. Infrastructure is included due to maintenance, electricity, water, and garbage. Operations and technical support begin accruing and cost roughly $116,000 and $40,000 respectfully. These services include management, testers, consultants, and other critical operational roles. Total estimated costs for Ongoing Operations are $1.31 million over the period of the project.

*Continuing Development* will occur roughly a year the initial product release with some work being transitioned into alpha, beta testing, and bug fixes. These costs start to build during the sixth quarter of the project but decrease afterwards as testing finishes and bugs are fixed. New product ideas are also introduced and worked on. Total costs for Continuing Development over the life of the project is $1.86 million.

Total estimated cost for the entire project is $15.7 million.

**Figure 1.4: Quarterly Estimated Cost Analysis (all figures in thousands)**  


**Cost-Benefit Analysis (Not included in real bid due to internal financials)**

Although the estimated total cost of the project is $15.6 million, the bid for this project is for a total $18.8 million assuming the estimated costs plus a fixed fee of 20%. Below is a detailed analysis of the costs and the assumed benefits from the E-Vote project. Overall, the project will produce a profit margin of 20% providing a reasonable profit and economic benefit.

**Figure 1.5: Quarterly Cost and Benefit Analysis (all figures in thousands)**



## 1.6 Risks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Risk Category** | **Risk Description** | **Impact** | **Frequency** | **Risk Control** |
| **R-1.0** | **User** | Some old people are not familiar with smartphones | Negligible | Remote | **Accept**  Still using mail voting |
| **R-2.0** | **User** | People may not trust automated voting | Serious | Occasional | **Avoid:** Run "test drives" with a specific sample of people to promote the app and gain trust among people. |
| **R-3.0** | **Networktivity** | Incorrect data during network transmission | Critical | Occasional | **Avoid:** Using stable network |
| **R-4.0** | **Security** | The database was invaded or modified | Catastrophic | Remote | **Avoid:** Backup database on time |
| **R-5.0** | **Expandability** | When expanding E-Vote to other states, those servers may not enough to run normally | Serious | Remote | **Avoid:** Requiring a rebuild of the servers to meet the higher operational requirements or databases. |
| **R-6.0** | **Security** | Internal misuse of information | Catastrophic | Remote | **Avoid:** Use background checks when hiring and have proper monitoring systems for suspicious conduct |
| **R-7.0** | **Security** | Foreign hacking or monitoring of systems | Catastrophic | Remote | **Avoid:** Hire security experts and consultants, use secure coding practices |
| **R-8.0** | **Reliability** | Speed of systems hinder or discourage use of system | Serious | Remote to Occasional | **Avoid:** Use cloud technologies to ensure high reliability |

## 1.7 Conclusion

E-Vote is a great opportunity to alleviate much of the frustrations of voting and bring the practice into the 21st century. The strengths of the proposal are key to the success of the project. From state of the art security and design; to reasonable estimated cost and the economic benefits, the project stands to help virtually all stakeholders in the State of Washington vote in a convenient and secure way. With this project completed, voter turnout should increase amongst every group and continue the historical trends that American Democracy has established. Ultimately, our plan offers the most cost efficient, secure, and cohesive plan to tackle this issue.

# 2 Software Requirements Specification

## 2.1 Functional Requirements

Four interfaces have been described in the functional requirements below:

* **Portal interface** -- Allows users to create and manage their account
* **Landing interface** -- Displays a list of upcoming elections in which a user can vote, with links to an “Election Interface” for each
* **Election Interface** -- Provides specific information about one given race
* **Voting interface** -- Allows the user to vote on one given race

Each interface has a corresponding section lists requirements about what the interface should display and related back-end system requirements necessary to support that interface.

A supplementary section, “Back-End System Requirements”, contains additional system requirements which are not specific to any particular interface.

### Portal Interface Requirements

|  |  |
| --- | --- |
| **Req-ID** | **Requirement** |
| **FR-PP-01** | User shall be able to create an account |
| *FR-PP-01.01* | User shall be able to select a username and password when creating an account by entering data into a text field. |
| *FR-PP-01.02* | Users shall be prevented from choosing a username less than three characters in length or exceeding fifteen characters in length. |
| *FR-PP-01.03* | Users shall be prevented from choosing a username containing non-alphanumeric characters (i.e., anything other than a-Z, 0-9). |
| *FR-PP-01.04* | Users shall be prevented from selecting a username that matches any existing username. |
| *FR-PP-01.05* | Users shall be prevented from choosing a password less than eight characters in length or exceeding twenty characters in length. |
| *FR-PP-01.06* | Users shall be prevented from choosing a password that does not contain at least one letter (a-Z), one number (0-9), and one of the following characters: ! @ # $ % ^ & \* ( ) - \_ + = , . ? < > ~ ` |
| *FR-PP-01.07* | Users shall be prevented from choosing a password that contains any of the characters not specified in *FR-PP-01.06*. |
| *FR-PP-01.08* | User shall be notified on screen via message after submission of any failed username/password validation checks incurred by their selection of a username/password. |
| *FR-PP-01.09* | User shall input their social security number, date of birth, last name, and first name to verify their identity when creating an account. |
| **FR-PP-02** | User shall be able to login to their account by providing account information: username, password, last four digits of social security number, and date of birth. |
| *FR-PP-02.01* | If the user is unable to successfully login within 5 attempts, the login option is locked and the user is offered to receive an email link to reset the password. |
| *FR-PP-02.02* | The user shall have the option of receiving an email link to reset their password from the login screen. |
| *FR-PP-02.03* | Notification of failed login attempts shall be sent to registered email account. |
| **FR-PP-03** | User should be able to optionally link a phone number to their account. |
| *FR-PP-03.01* | If a user attempts to link a phone number to their account, the user should be sent a text message with a verification code which they can enter on their account management page to complete the linking of that phone number to their account. |
| **FR-PP-04** | If a user has linked a phone number to their account, the user should be able to enable two-factor authentication for logging in to their account. |
| **FR-PP-05** | If a user has enabled two-factor authentication on their account, after completing the login process outlined in **FR-PP-02**, the user should be required to additionally enter a unique code sent to the phone number linked to their account via SMS message in order to log in. |
| **FR-PP-06** | User may be able to enable biometric login options to enter cached login data if their device allows for this. |
| **FR-PP-07** | User shall be able to change their password by providing their old password and a new password. |
| **FR-PP-08** | User shall be able to change the email address associated with their account from the account management page during while they are logged in by providing a new email address and clicking a verification link sent to that email address. |

### 

### Landing Interface Requirements

|  |  |
| --- | --- |
| **Req-ID** | **Requirement** |
| **FR-LP-01** | System shall pull user address information from the state’s provided voter registration database. |
| **FR-LP-02** | Landing interface shall inform user of their registered address. |
| **FR-LP-03** | System shall use legislative district information associated with the account to display a tailored landing page. |
| *FR-LP-03.01* | Landing interface shall display a list items to vote on for which the corresponding election is three or less months away. |
| *FR-LP-03.02* | Landing interface shall provide a link for each race which leads to the “Election Interface” for that item. |
| *FR-LP-03.03* | User shall be able to click on one of those links to go to an “Election Interface” specific to that item. |
| **FR-LP-04** | System shall display the date and number of days until the next election they are eligible to vote in. |
| **FR-LP-05** | Landing interface shall display the user’s current voter registration status (providing a warning and a link to register if the user is not registered). |
| **FR-LP-06** | Landing interface shall provide a link to account management services to view and edit account settings. |
| **FR-LP-07** | Landing interface may provide an option for the user to vote “ballot-style”, where the user is sequentially taken to each item for which they can currently vote (i.e., when they submit a vote for one item, they are automatically taken to the voting interface for the next item, until there are no remaining items to vote for). |

### Election Interface Requirements

|  |  |
| --- | --- |
| **Req-ID** | **Requirement** |
| **FR-EP-01** | Election Interface shall display the following information points specific to the corresponding race: description of the race; description of each selection for that race; date that voting for that race begins. |
| *FR-EP-01.01* | System shall pull information for the Election Interface for a specific item from the state-provided election information database. |
| **FR-EP-02** | Election Interface shall display a “vote button” which links to a voting interface for the item. |
| *FR-EP-02.01* | If voting has not opened for the item, the “vote” button shall be disabled, with a visual indication that it is disabled. |
| **FR-EP-03** | Election Interface shall contain a “Research” section that shall provide approved links to external resources regarding campaigns and issues. |
| *FR-EP-03.01* | Links to external resources shall be opened in a separate browser tab and be external to the E-Vote system. |

### Voting Interface Requirements

|  |  |
| --- | --- |
| **Req-ID** | **Requirement** |
| **FR-VP-01** | Voting Interface shall display the title of the corresponding race and the different voting options. |
| *FR-VP-01.01* | System shall pull information to be displayed for each voting option from the databases provided by the state. |
| **FR-VP-02** | Each option shall be individually selectable. |
| **FR-VP-03** | User shall only be able to select one option from the list of options. |
| **FR-VP-04** | User shall be able to submit their vote by selecting an option and then clicking the “Submit Vote” button. |
| **FR-VP-05** | Voting interface shall display a confirmation message to the user, asking for confirmation before their vote is submitted. |
| *FR-VP-05.01* | Voting interface shall attempt to submit the vote after user has confirmed submission in the confirmation message. |
| **FR-VP-06** | If a vote was successfully submitted, the Voting interface shall update to inform user their vote was submitted and display what their choice was. |
| **FR-VP-07** | If the submission of a vote fails as outlined in **FR-BES-01**, the user shall be notified that their vote was unsuccessful and prompted to vote again. |
| **FR-VP-08** | If a user is blocked from voting on an issue due to repeated failures as outlined in **FR-BES-03**, the user shall be notified that there was an issue and asked to try again in an hour. |
| **FR-VP-09** | Voting interface shall not allow a user to vote on an item they have already voted on. Information about which issues a voter has voted on is accessed as outlined in **FR-BES-05**. |
| **FR-VP-10** | System should automatically save what the user has selected during the voting process so they don’t have to repeat their selection if they become disconnected during their voting session. |
| *FR-VP-10.01* | System should notify the user upon login if they have a saved selection that is unsubmitted for any voting items. |

### Back-End System Requirements

|  |  |
| --- | --- |
| **Req-ID** | **Requirement** |
| **FR-BES-01** | If the submission of a vote fails, the entire transaction and any related transactions shall be discarded. |
| **FR-BES-02** | All failures to submit a vote shall be recorded. |
| **FR-BES-03** | If a user submits at least five votes which fail within an hour, the user shall be blocked from voting on that item for one hour. |
| *FR-BES-03.01* | If a user is blocked from voting due to repeated failures, the system shall send a notification to a system administrator. |
| **FR-BES-04** | The system shall connect to databases provided by the state for the purpose of user authentication. |
| **FR-BES-05** | The system shall connect to databases provided by the state for the purpose of tracking which items a user has voted on. |
| **FR-BES-06** | The system shall allow for the creation of “admin” accounts which can modify user account information and reset user account passwords to assist users who are unable to do so through the website. |

## 2.2 Non-Functional Requirements

This section describes non-functional requirements in terms of usability, reliability, performance, and availability. Specific functionality is not described here; for a list of requirements pertaining to the functions of the E-Vote system, refer to the “Functional Requirements” section.

### Usability

|  |  |
| --- | --- |
| **Req-ID** | **Requirement** |
| **NFR-U-01** | Users must be able to locate the “Submit Vote” button on the Voting Interface within five seconds on at least 95% of first visits to the interface. |
| **NFR-U-02** | Users must accurately identify what effect clicking any button on the website will have 95% of the time. |
| **NFR-U-03** | Website must be available in all languages that the state-wide voter’s pamphlet is in (currently these are English, Spanish, Chinese, and Vietnamese). |

### Reliability

|  |  |
| --- | --- |
| **Req-ID** | **Requirement** |
| **NFR-R-01** | Every vote from a user must be stored redundantly. |
| **NFR-R-02** | Any system failure does not result in the loss of previously committed votes. |

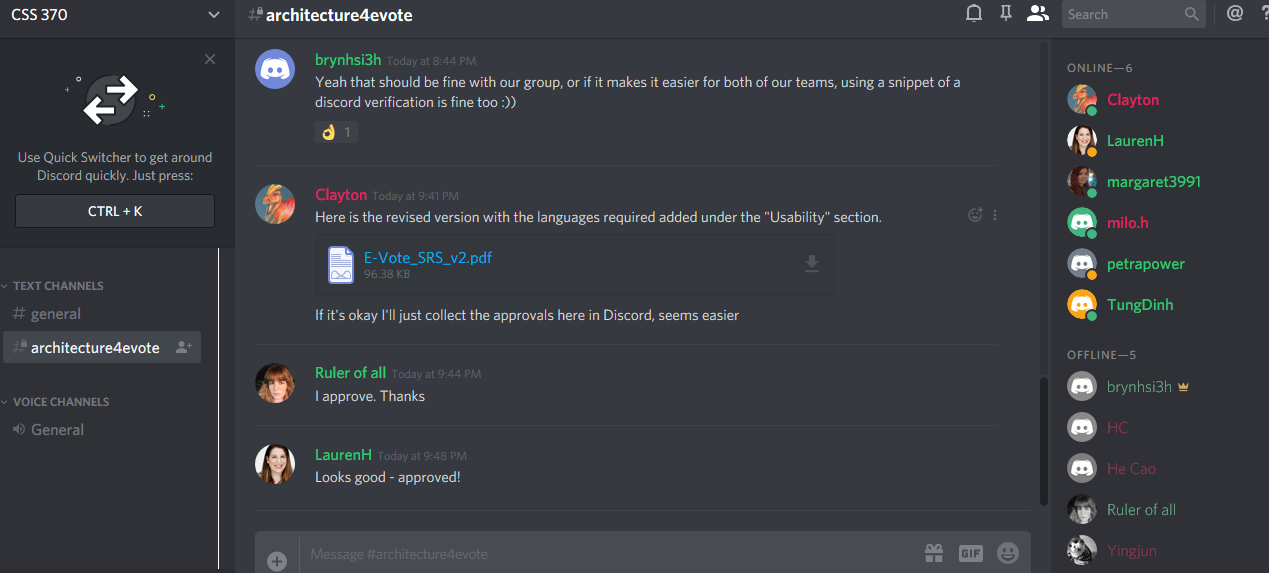
### Performance

|  |  |
| --- | --- |
| **Req-ID** | **Requirement** |
| **NFR-P-01** | Authentication of a user as outlined in *FR-PP-01.02* must not exceed 10 seconds in more than 1% of attempts. |
| **NFR-P-02** | Authentication of a user as outlined in *FR-PP-01.02* must not exceed 7 seconds on average across all attempts. |
| **NFR-P-03** | System must maintain 99.9% availability during high-traffic voting periods. |

### Availability

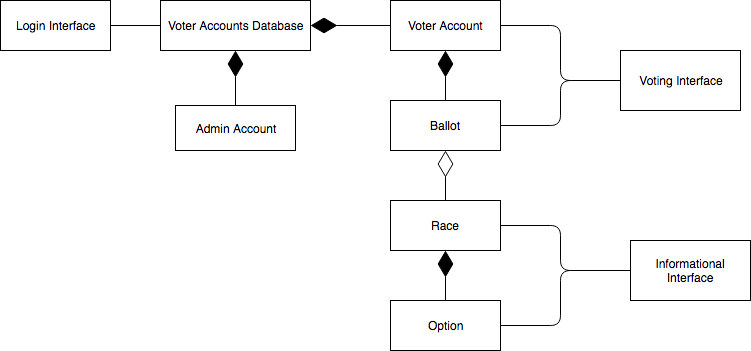
|  |  |
| --- | --- |
| **Req-ID** | **Requirement** |
| **NFR-A-01** | System must maintain 99% availability during high-traffic voting periods. |
| **NFR-A-02** | Any period of unavailability during high-traffic voting periods must not exceed 15 minutes in length. |
| **NFR-A-03** | System must maintain 99.99% availability during low-traffic voting periods. |

## 2.3 Business Team Approval



# 3 Architecture Design

## 3.1 Domain Modeling



# 4 Use Cases

## 4.1 Use-Case Diagram

### 

## 4.2 Use-Case Descriptions

### Start session (Voter login)

Scenarios:

1. Voter goes to the E-Vote website in their browser.
2. E-Vote website displays the “login” interface which prompts the Voter for their username and password.
3. E-Vote system validates the Voter’s information with the Account Database.
4. E-Vote system displays the “landing” interface.
5. Voter is then allowed to perform one or more account actions by clicking links on the landing interface.
6. Apply “Perform transaction” use case.
7. When a Voter is finished with their session, they can sign out by selecting “Sign Out” on the landing interface.

Alternatives:

**In step 2:**

If the Voter enters the incorrect account information, they will be notified of this, returned to the “login” interface, and prompted to re-enter their password. If they enter the incorrect password for an account three times within five minutes, the account will be locked and they will be sent an email to reset their account password.

**In step 7:** If the Voter closes the browser window without clicking “Sign Out”, their session will remain active until the user has been inactive for 5 minutes, at which point their session will be automatically ended.

### Vote in a race

Scenarios:

1. Election Interface displays description of the race, description of each selection for that race, and date that voting for that race begins.
2. Voter clicks on the “Vote” button on the race that they want to vote for, which takes them to the Voting Interface.
3. System pulls information to be displayed for each voting option provided by the state from the Election Database.
4. Voting Interface display the title of the corresponding race and the different voting options with brief information about each.
5. Voter selects one option for the race.
6. Voter clicks the “Submit Vote” button to confirm their decision.
7. System displays a confirmation message to the user, asking for confirmation before their vote is submitted.
8. Voter confirms their decision by clicking the “Yes” button.
9. System sends the Voter’s decision to the Election Database.
10. Voting interface updates to inform the Voter that their vote was submitted and display what their choice was.
11. System takes the Voter back to Election Interface.
12. Voter clicks on the “Exit” button, exiting the Election Interface.

Alternatives:

**In step 2:** If the Voter decides to leave the Election Interface without voting, they can click on the “Exit” button to return to the previous interface without voting.

**In step 2:** If voting has not opened for the item, the “vote” button shall be disabled, with a visual indication that it is disabled.

**In step 2:** If the Voter has already voted in that particular race, then the Voting Interface will inform the Voter that they have already voted in that particular race.

**In step 4:** If the Voter decides to vote at a later time or not vote on that particular race, they can click on the “Back” button to return to the Election Interface without voting.

**In step 5:** If the Voter selects more than one option, the Voting Interface will only recognizes the most recent option that the Voter has selected.

**In step 8**: If the Voter clicks the “No” button, go back to step 4.

**In step 9:** If the submission of a vote fails, the entire transaction and any related transactions shall be discarded, and the system notifies the Voter that their vote was unsuccessful and prompts them to vote again.

**In step 9:** If the Voter is blocked from voting on an issue due to the system failing to submit the vote five times in the last hour, the system notifies the Voter that there was an issue and asks the Voter to try again in an hour.

### View info about upcoming races

Scenarios:

1. The voter can perform a transaction of viewing the upcoming races by selecting “Upcoming races”.
2. The Election Race page pulls info from the Election Database for each race and displays a list of candidates.
3. The voter can click “Remind me” link to receive a reminder when the voting time is open.
4. The voter can view more details when clicking on a specific race such as local, state, federal, ballot measure, and provided political statement.
5. A voter can deselect a certain election race by selecting the previously selected election race.
6. A voter can view election race information about the election race by selecting the “Contact” link which would allow the user to request more information about the candidate like their website and mailbox.
7. The E-Vote system can update the election races when the data is updated in the Election Database.
8. The voter can search for election races by district, local, state, federal, or by ballot measure.

### View info about a specific candidate

Scenarios:

1. The voter can perform a transaction of viewing the upcoming candidates by selecting “View Candidates”.
2. The Candidates page pulls info from the Election Database for each race and displays a list of candidates.
3. The voter can click “Remind me” link to receive a reminder when the voting time is open.
4. The voter can view more details when clicking on a specific candidate such as their party, education, background, and provided political statement.
5. A voter can deselect a certain candidate by selecting the previously selected candidate.
6. A voter can view contact information about the candidate by selecting the “Contact” link which would allow the user to request more information about the candidate like their website and mailbox.
7. The E-Vote system can update the candidates when the data is updated in the Election Database.
8. The voter can search for candidates by district, name, or party.

### Change password

Scenarios:

1. System prompts user to enter current password.
2. Voter validates identity by correctly entering password, which is validated by the Account Database.
3. System prompts user to enter new password.
4. Voter enters new password.
5. System validates that new password conforms to system requirements.
6. System prompts user to verify the password.
7. Voter confirms new password by correctly re-entering the password.
8. The Account Database is updated with the new password.
9. System sends confirmation email to registered email address informing Voter of the change of password.

**Alternatives**

In **step 2**, In step 3, if the Voter incorrectly enters the password, prompt the Voter to re-enter their password, and repeat **step 1**.

In **step 2**, if the Voter incorrectly enters the password 5 times, notify Voter that their account has been locked and that they have been sent a recovery email to reset their password and unlock their account. Prevent any additional login attempts until the user has followed the instructions in the email.

In **step 5**, if the password does not conform to requirement, display message informing Voter of password requirements, and return to **step 3.**

In **step 7**, if the passwords do not match, display an error message and repeat **step 7**.

### Change email

Scenarios:

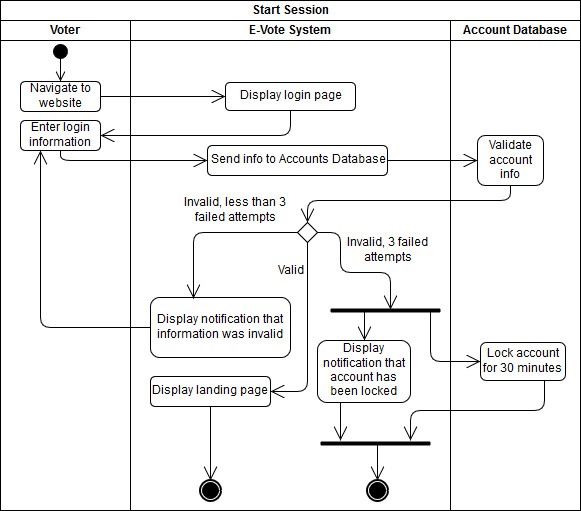
1. Voter selects “Change email” within “Update Account Information” use case.
2. Voter can then enter a new email address they would like to be associated with their account.
3. System sends an e-mail containing a verification link to the new address provided by the Voter.
4. Voter clicks on the verification link in the received email.
5. System permanently updates Voter’s account email address.

Alternatives:

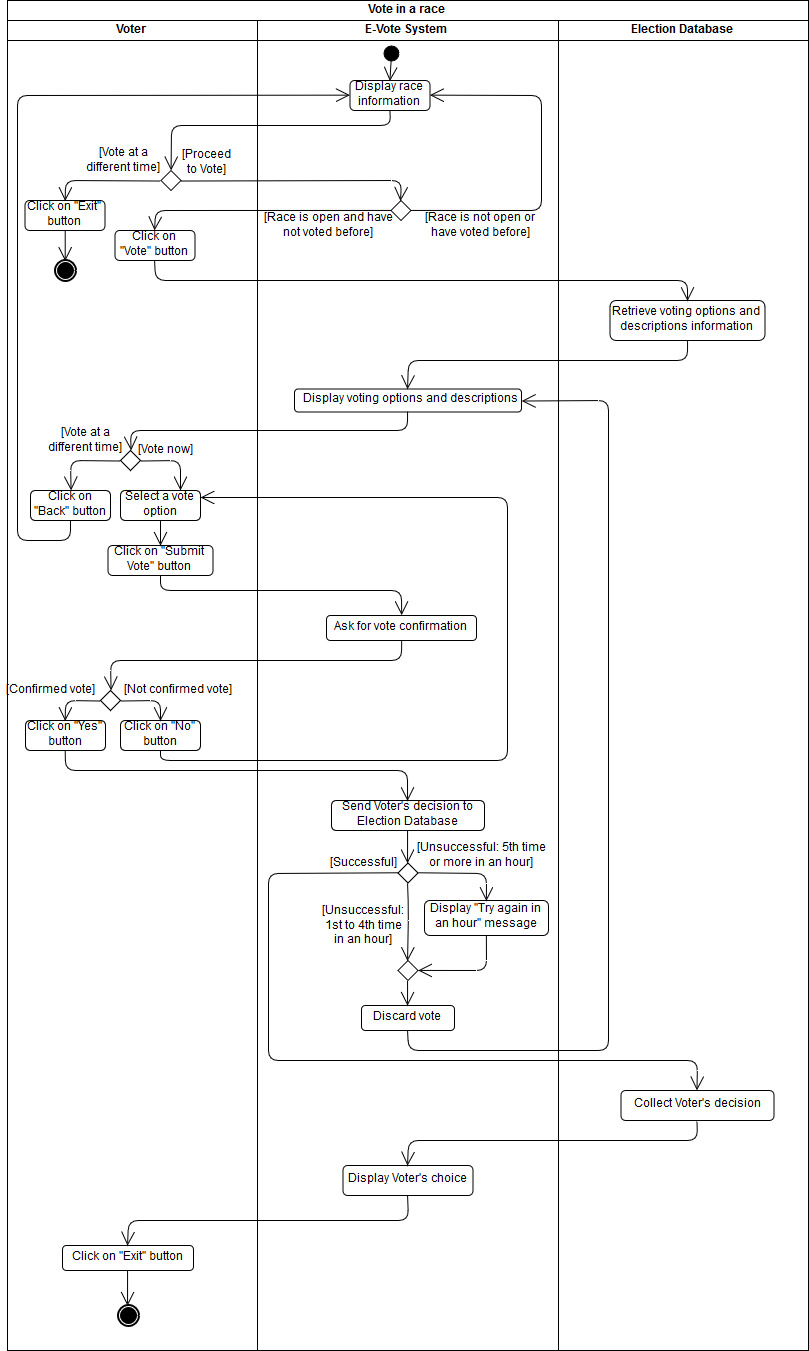
1. In **step 2**, if the entered email address does not conform to recognized email pattern, the System will notify the user and re-prompt for input.
2. In **step 4**, if the Voter never clicks the link in the verification email, the account’s email address is not changed.

# 5 Activity Diagrams

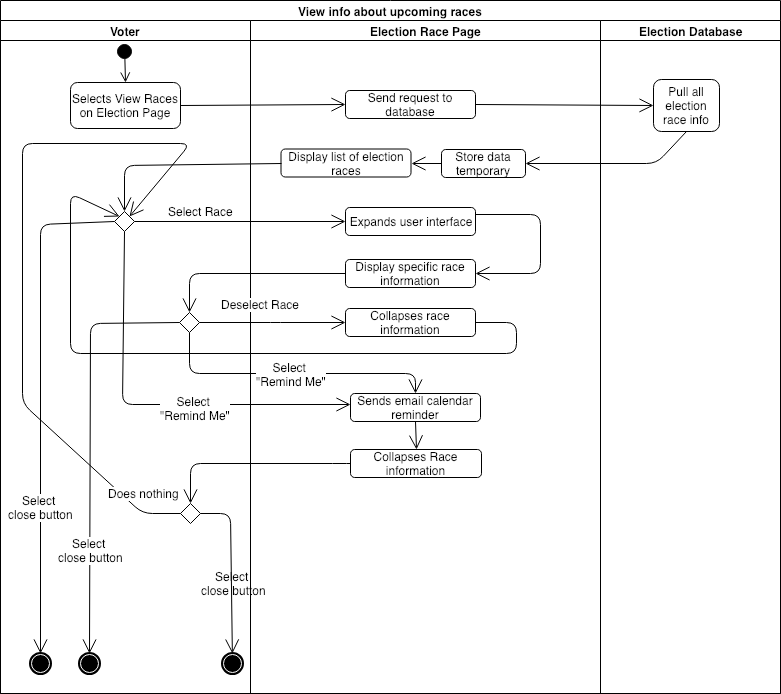
## 5.1 Start session (Voter login)



## 5.2 Vote in a race

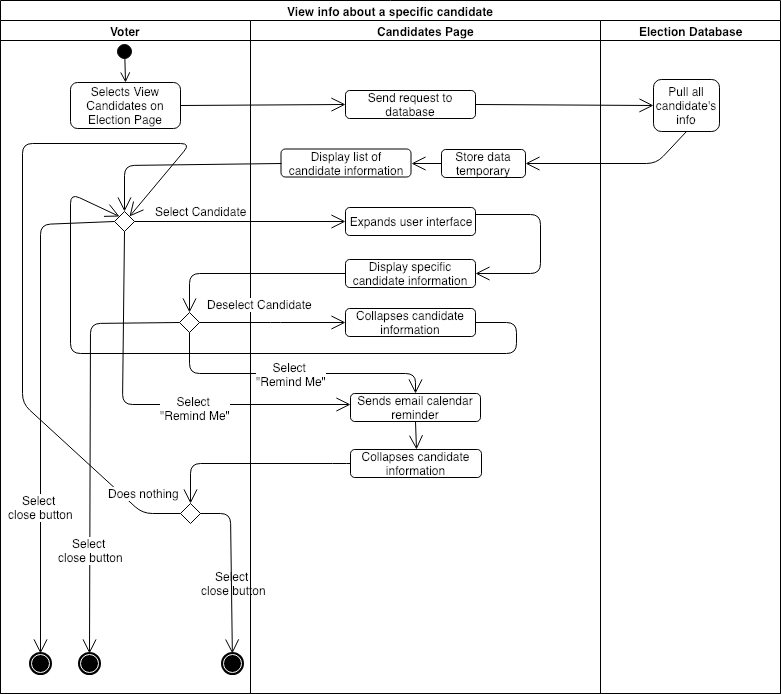


## 5.3 View info about upcoming races

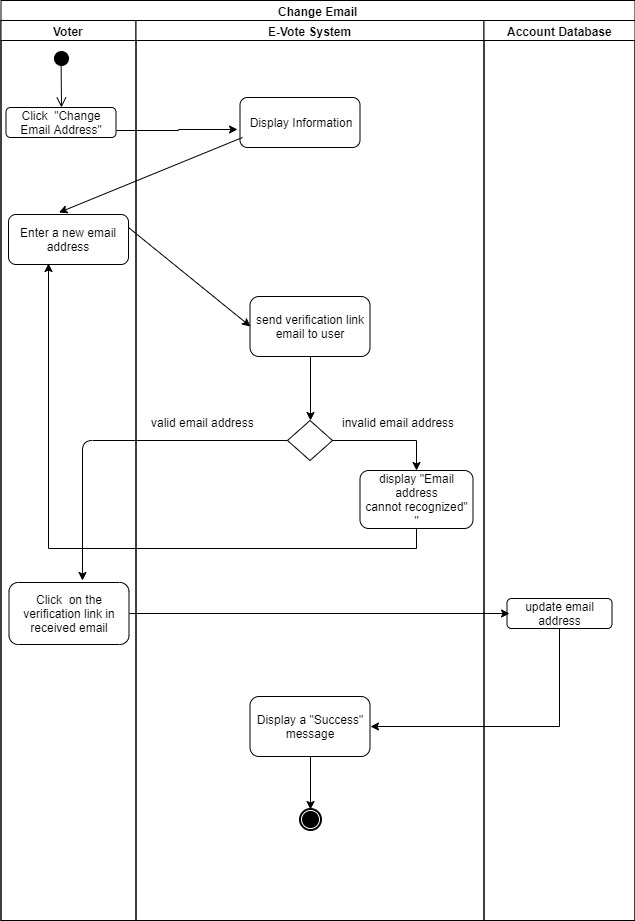


## 

## 5.4 View info about a specific candidate



## 5.5 Change password

5.6 Change email

# 6 Wireframes

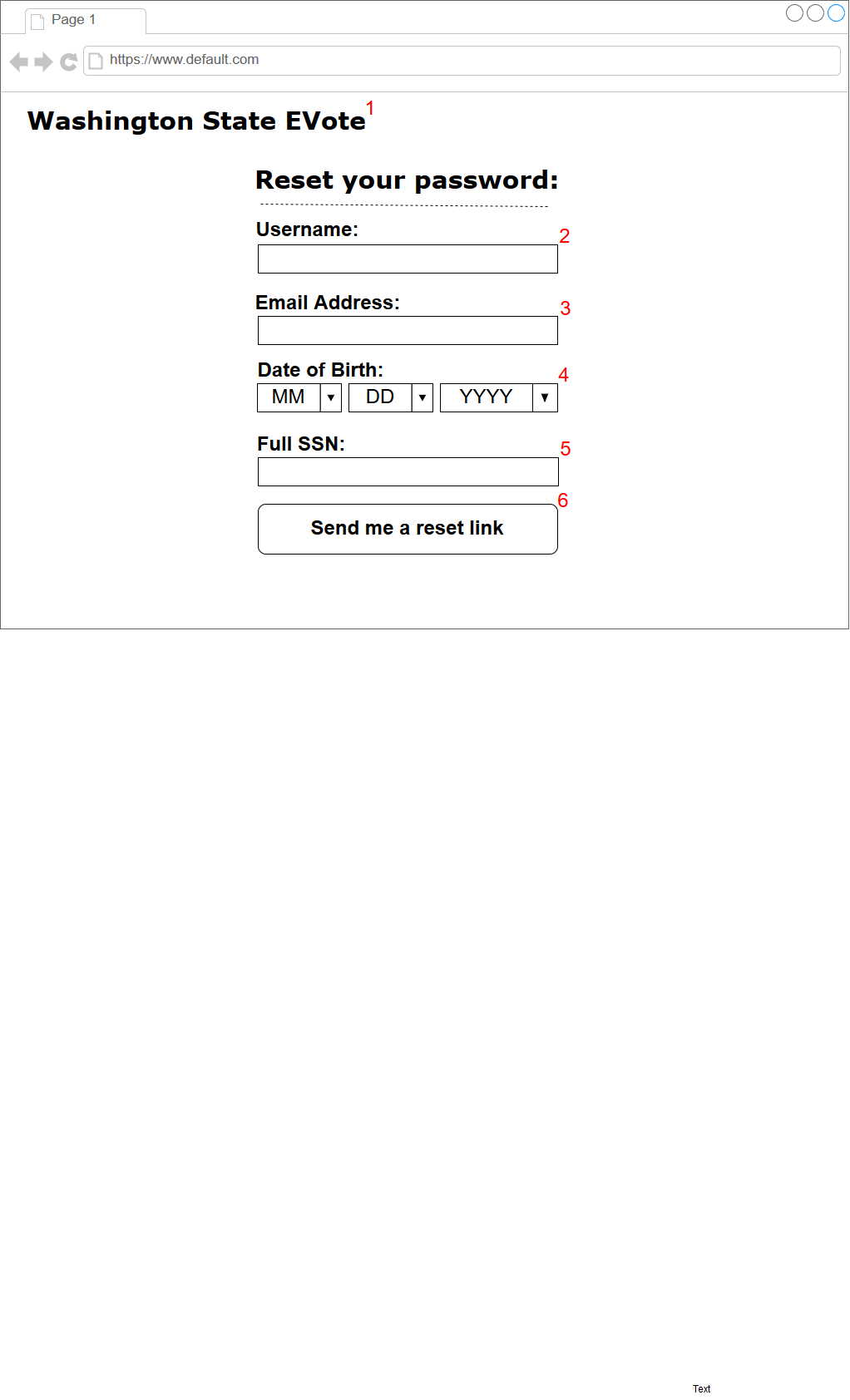
## 6.1a Login

## 

|  |  |
| --- | --- |
| **WF1a: Login Page** | |
| **Wireframe ID** | **Wireframe Description** |
| WF1a.1 | System Title/Identification |
| WF1a.2 | Username text field |
| WF1a.3 | Password text field |
| WF1a.4 | Drop-down menus for inputting user’s birthdate |
| WF1a.5 | Last four digits of SSN text field |
| WF1a.6 | Button to initiate login authorization |
| WF1a.7 | Link to “Reset Password” page |
| WF1a.8 | Link to “Registration” page |

|  |
| --- |
| **Navigation Table for Login Page** |
| WF1a.6 → WF2 (Landing Page) |
| WF1a.7 → WF1b (Password Reset Page) |
| WF1a.8 → WF1c (Registration Page) |

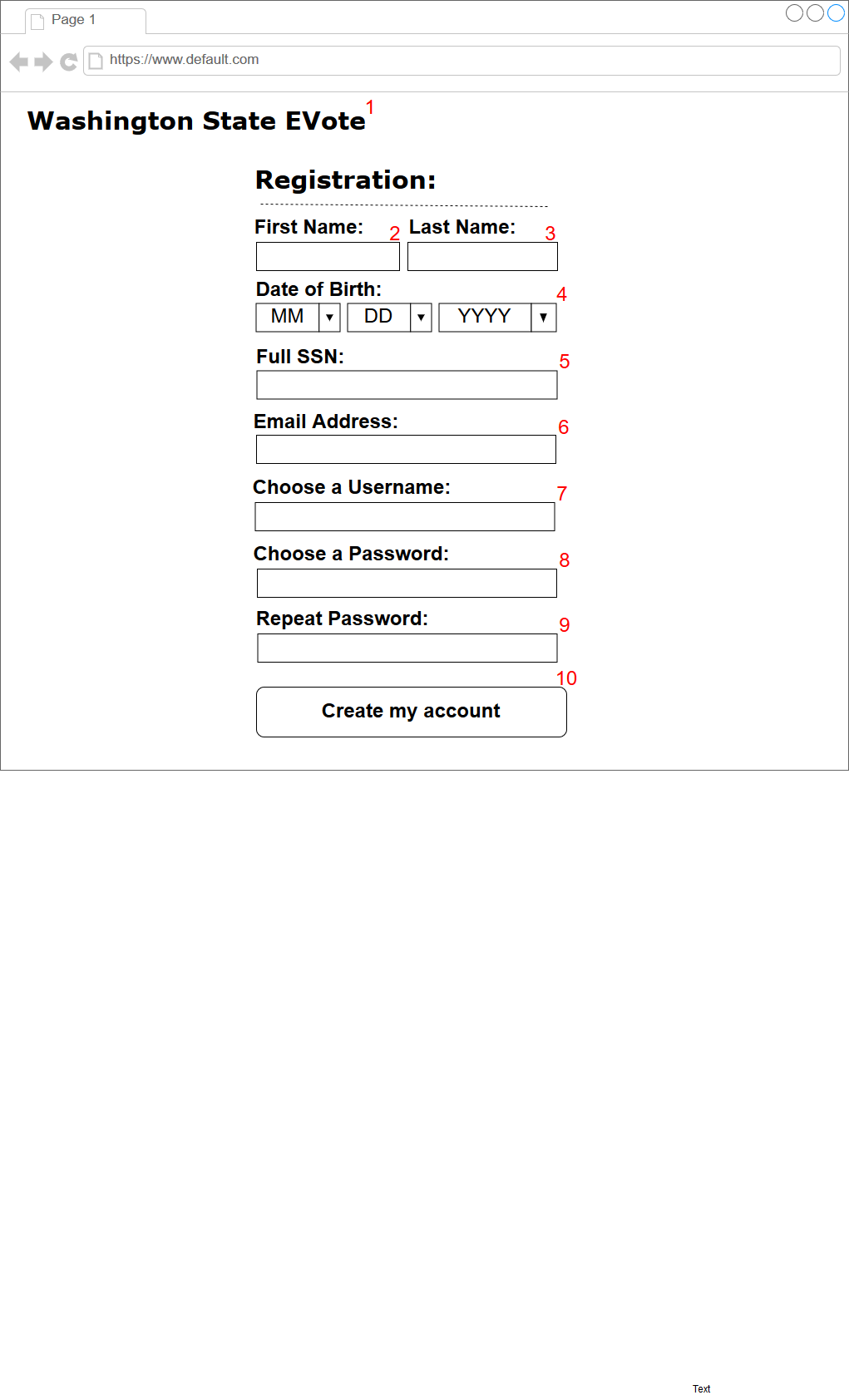
## 6.1b Reset Password Page



|  |  |
| --- | --- |
| **WF1b: Reset Password Page** | |
| **Wireframe ID** | **Wireframe Description** |
| WF1b.1 | System Title/Identification |
| WF1b.2 | Username text field |
| WF1b.3 | E-mail address text field |
| WF1b.4 | Drop-down menus for inputting user’s birthdate |
| WF1b.5 | Full SSN text field |
| WF1b.6 | Button to initiate password reset |

|  |
| --- |
| **Navigation Table for Reset Password Page** |
| WF1b.6 → WF1a (Login Page) |

## 6.1c Registration Page



|  |  |
| --- | --- |
| **WF1c: Registration Page** | |
| **Wireframe ID** | **Wireframe Description** |
| WF1c.1 | System Title/Identification |
| WF1c.2 | First name text field |
| WF1c.3 | Last name text field |
| WF1c.4 | Drop-down menus for inputting user’s birthdate |
| WF1c.5 | Full SSN text field |
| WF1c.6 | Email address text field |
| WF1c.7 | Username text field |
| WF1c.8 | Password text field |
| WF1c.9 | Password confirmation text field |
| WF1c.10 | Button to initiate account creation process |

|  |
| --- |
| **Navigation Table for Registration Page** |
| WF1c.10 → WF1a (Login Page) |

## 6.2 Landing



|  |  |
| --- | --- |
| **WF2: Landing Page** | |
| **Wireframe ID** | **Wireframe Description** |
| WF2.1 | System Title/Identification |
| WF2.2 | Link to Landing Page |
| WF2.3 | Link to Election Page |
| WF2.4 | Link to Voting Page |
| WF2.5 | Link to leave E-Vote system |
| WF2.6 | Link to Account Setting |
| WF2.7 | Information about the voters (Address and registration statue) |
| WF2.8 | Information about the election (Title, Date, and Days before the election) |
| WF2.9 | Link to the Election Page |
| WF2.10 | Link to the next Landing Page |

|  |
| --- |
| **Navigation Table for Landing Page** |
| WF2.2 → WF2 (Non-functional on this page, already on Landing Page) |
| WF2.3 → WF3 (Election Page) |
| WF2.4 → WF4 (Voting Page: last visited Voting Page) |
| WF2.5 → WF5 (Link to log user out, returns to Portal Page) |
| WF2.6 → WF6 (Link to account management services to view and edit account settings) |
| WF2.9 → WF4 (Election Page that corresponds to the specific election of the current Landing Page) |
| WF2.10 → WF10 (Link to next page for more election options) |

## 6.3 Election



|  |  |
| --- | --- |
| **WF3: Election Page** | |
| **Wireframe ID** | **Wireframe Description** |
| WF3.1 | System Title/Identification |
| WF3.2 | Link to Landing Page |
| WF3.3 | Link to Election Page |
| WF3.4 | Link to Voting Page |
| WF3.5 | Link to leave E-Vote system |
| WF3.6 | Information about the race (Title, Date, description) |
| WF3.7 | Information about the candidates (Name, Party, Biography, External Link) |
| WF3.8 | Link to Voting Page for that specific race |

|  |
| --- |
| **Navigation Table for Election Page** |
| WF3.2 → WF2 (Landing Page) |
| WF3.3 → WF3 Non-functional on this page, already on Election Page |
| WF3.4 → WF4 (Voting Page: last visited Voting Page) |
| WF3.5 → WF5 (Link to log user out, returns to Portal Page) |
| WF.8 → WF4 (Voting Page that corresponds to the specific race of the current Election Page) |

## 

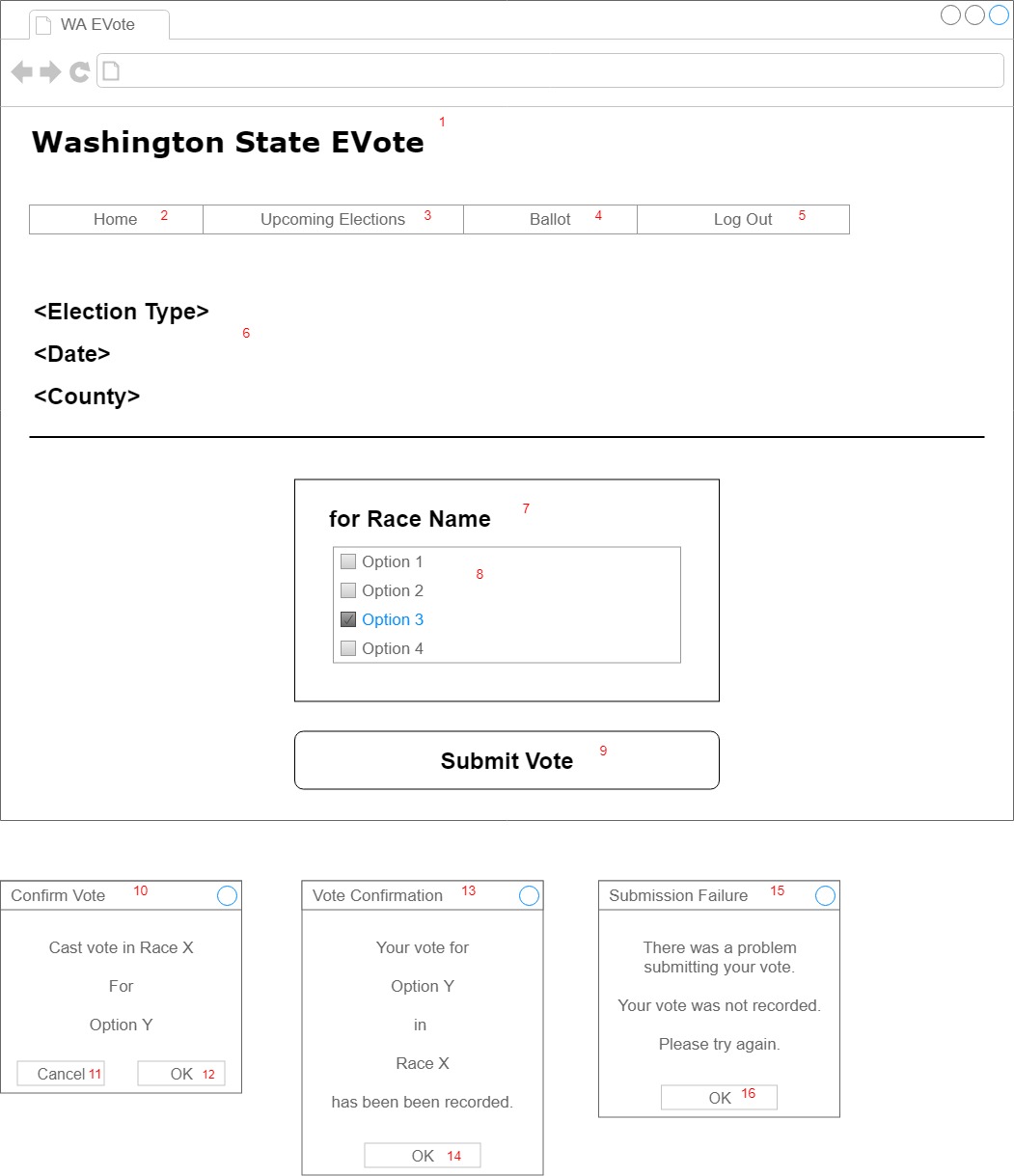
## 

## 

## 

## 

## 6.4 Voting

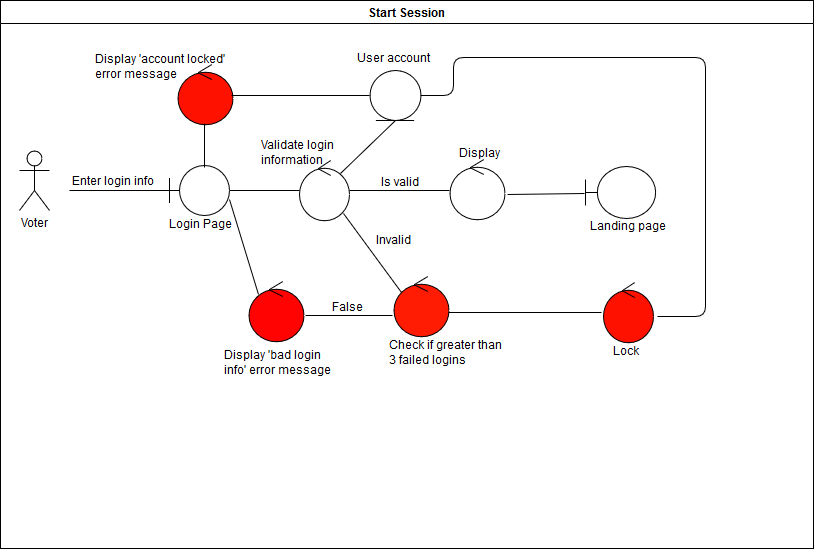


|  |  |
| --- | --- |
| **WF4: Voting Page** | |
| **Wireframe ID** | **Wireframe Description** |
| WF4.1 | Title |
| WF4.2 | Link to Landing Page |
| WF4.3 | Link to Election Page |
| WF4.4 | Link to Voting Page (inactive) |
| WF4.5 | Link to Portal Page, and logs user out of current session |
| WF4.6 | Election information for the current race |
| WF4.7 | Name of the current race |
| WF4.8 | List of options for the current race |
| WF4.9 | Button to submit vote |
| WF4.10 | Selection confirmation for the current race |
| WF4.11 | Cancel confirmation |
| WF4.12 | Confirm selection |
| WF4.13 | Notice to confirm vote was recorded |
| WF4.14 | Button to close confirmation notice |
| WF4.15 | Notice of vote failure |
| WF4.16 | Button to close failure notice |

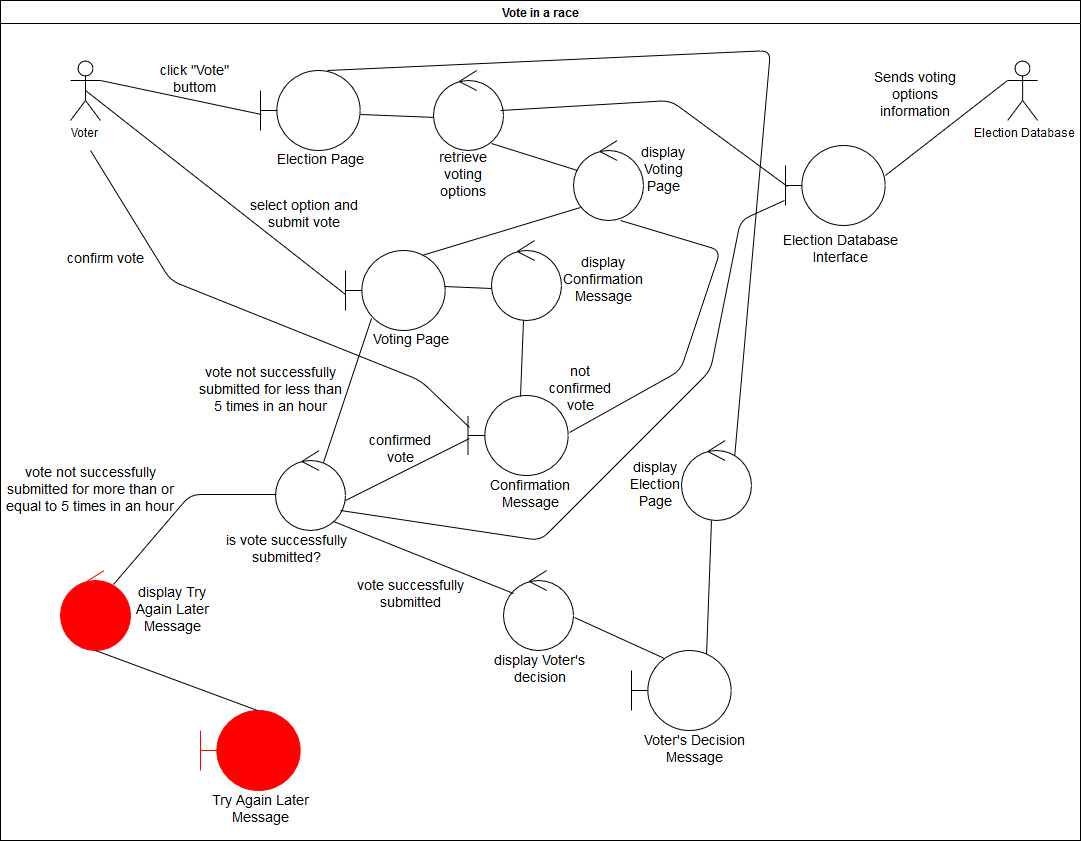
|  |
| --- |
| **Navigation Table for Voting Page** |
| WF4.2 → WF2 (Landing Page) |
| WF4.3 → WF3 (Election Page) |
| WF4.4 → inactive on this page (link to self) |
| WF4.5 → WF5 (Link to log user out, returns to Portal Page) |

# 7 Robustness Diagrams

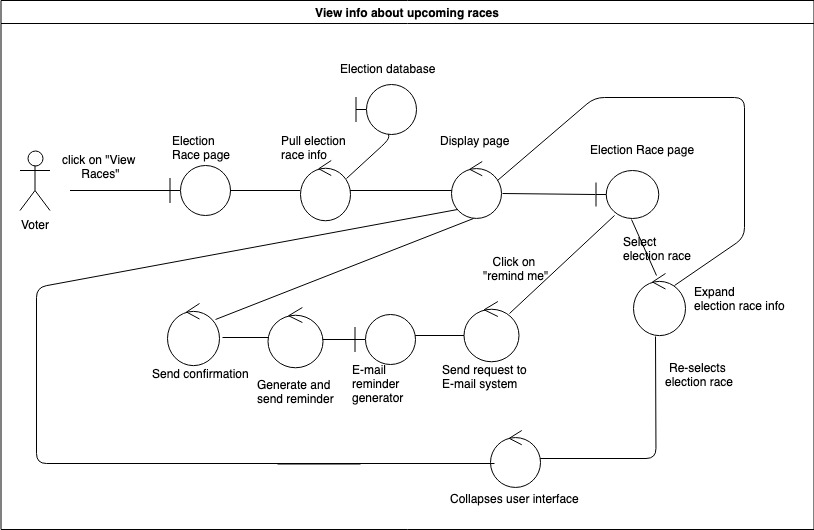
## 7.1 Start session (Voter login)



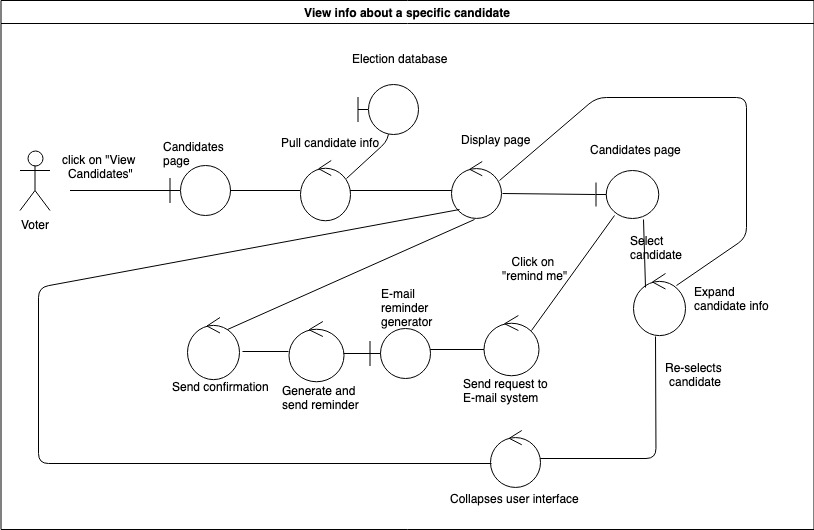
## 7.2 Vote in a race



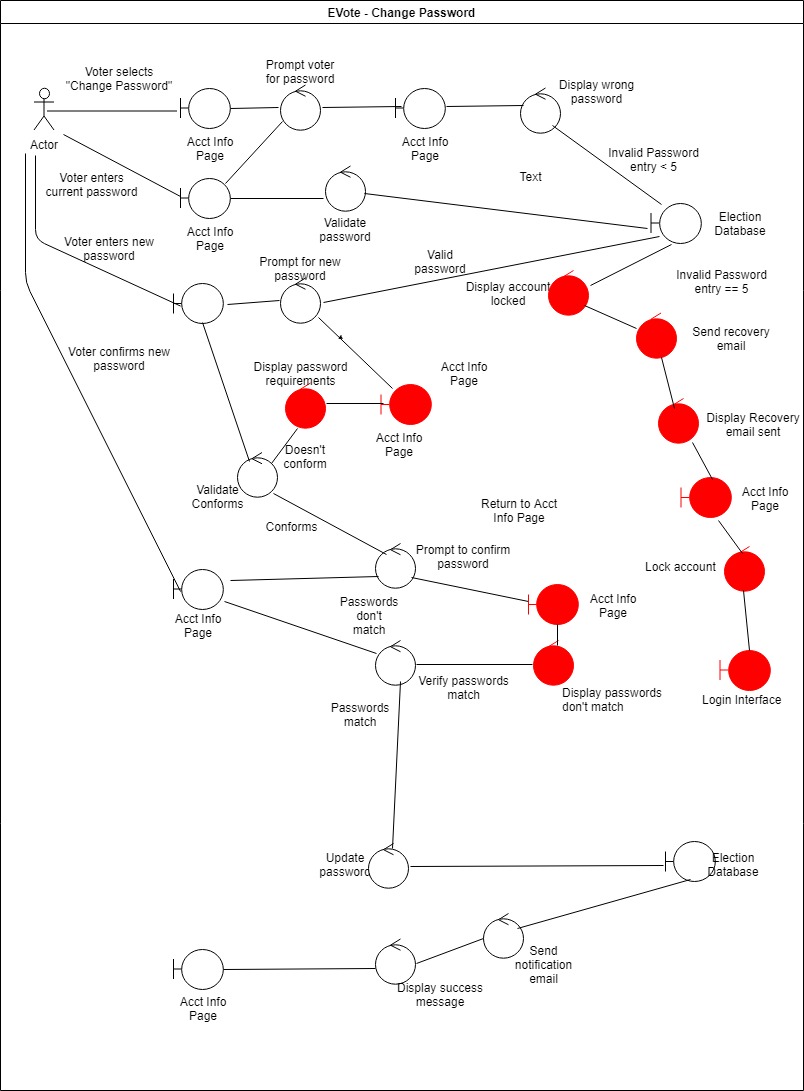
## 7.3 View info about upcoming races



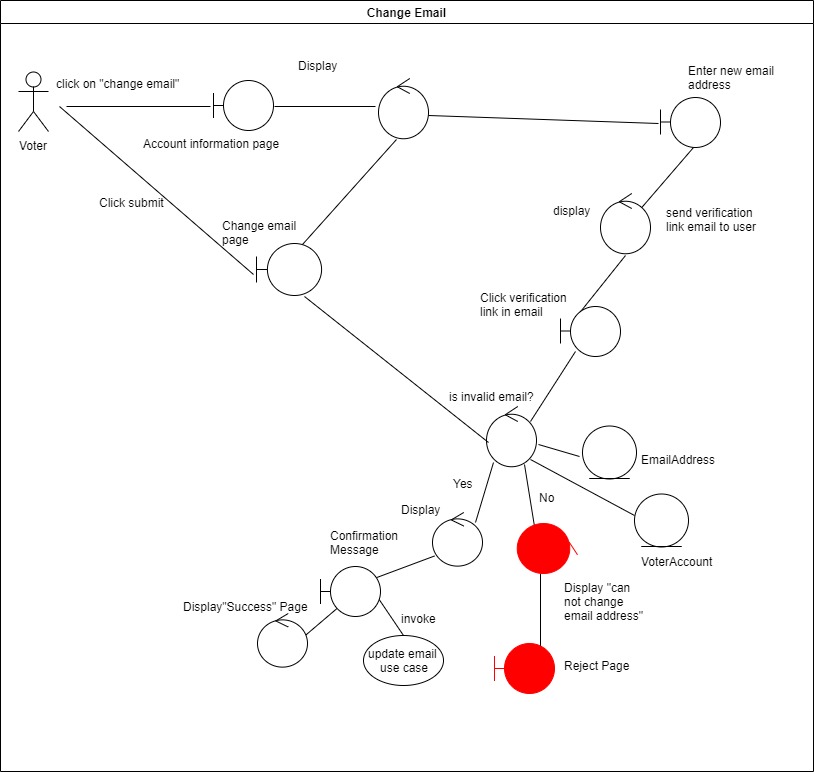
## 7.4 View info about a specific candidate



7.5 Change password

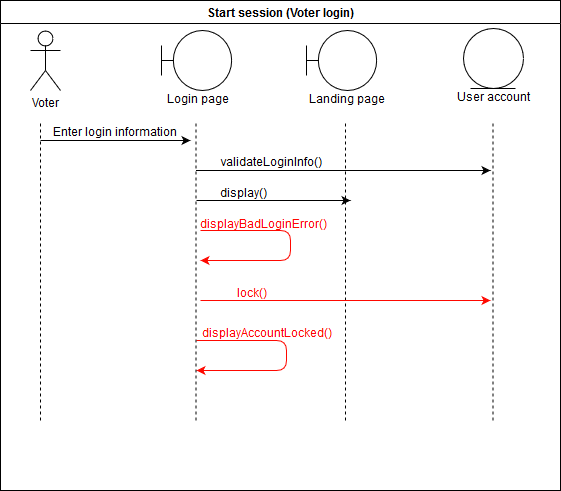


## 7.6 Change email

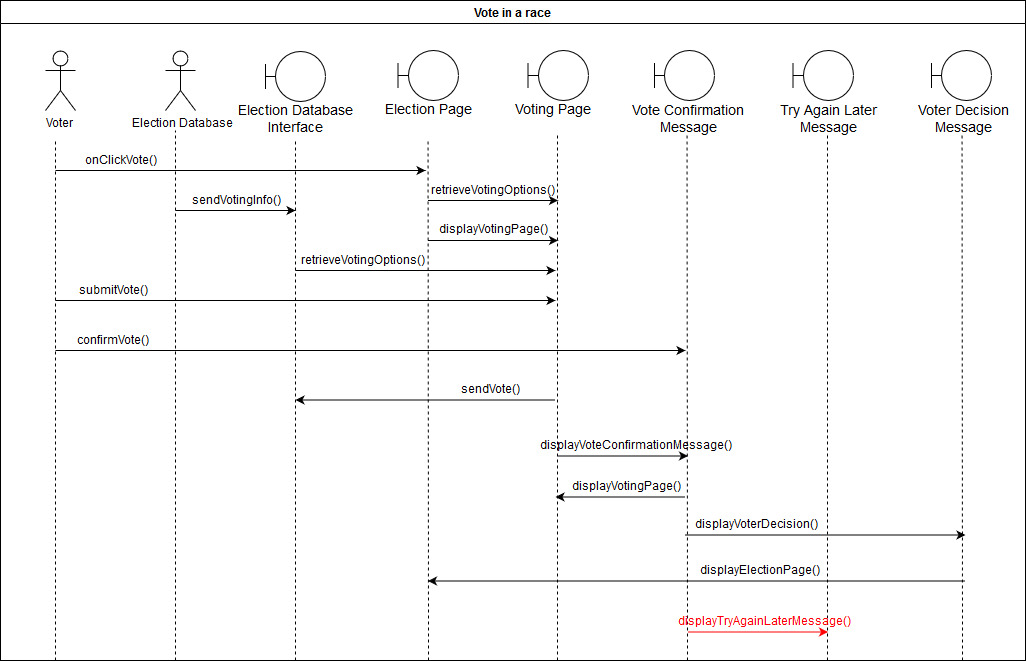


# 8 Sequence Diagrams

## 8.1 Start session (Voter login)



## 8.2 Vote in a race



## 8.3 View info about upcoming races

## 

## 

## 8.4 View info about a specific candidate

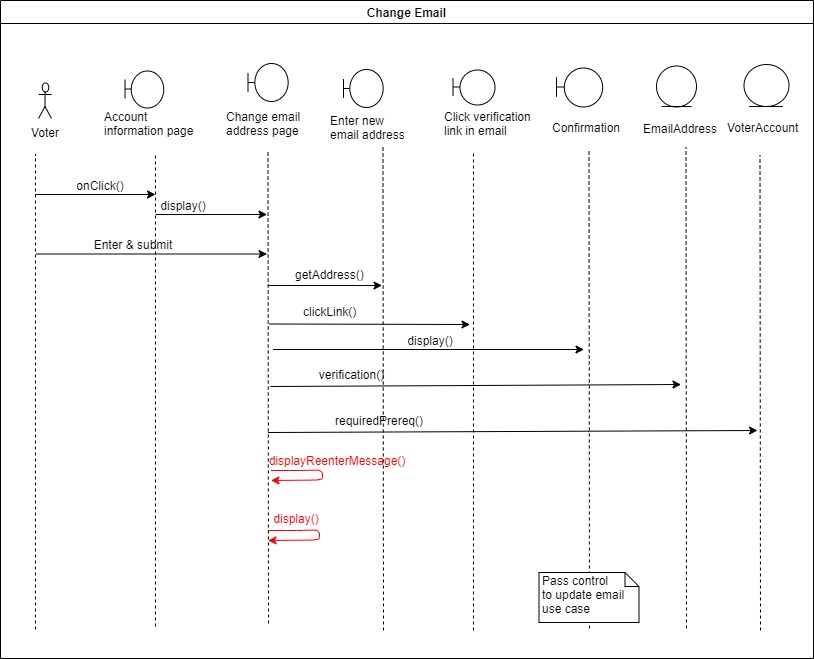
## 

## 

## 8.5 Change Password

## 

## 8.6 Change email



# 9 Test Cases

## 9.1 User login

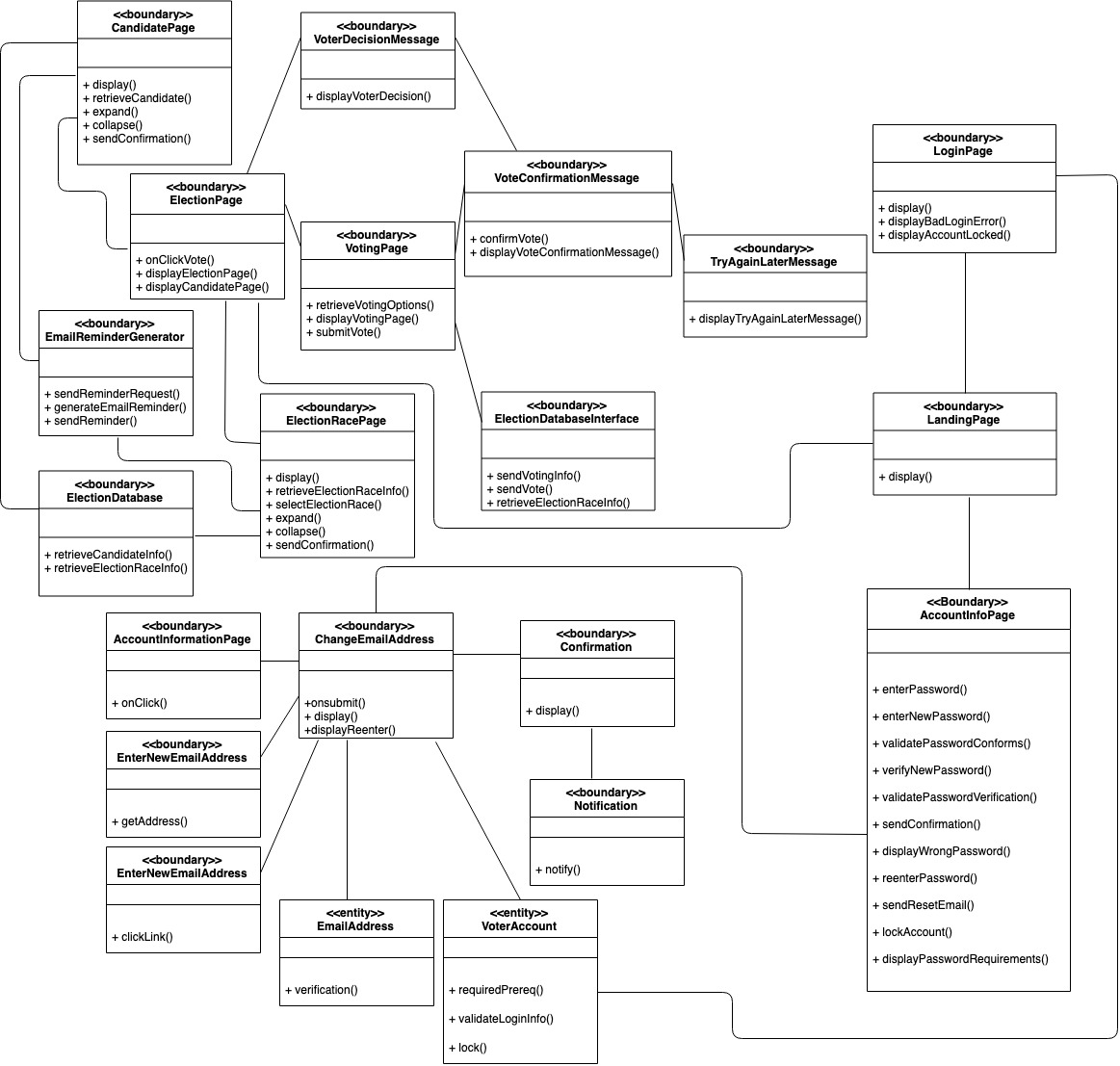
|  |  |
| --- | --- |
| **Test Case #:** 9.1 | **Test Case Name:** User login test case |
| **System:** E-Vote | **Subsystem:** User authentication |
| **Designed by:** Clayton Snyder | **Design Date:** March 13, 2019 |
| **Executed by:** | **Execution Date:** |
| **Description:** Test logging into a user account. | |

|  |
| --- |
| **Preconditions:**   1. User has successfully created an account. 2. User has navigated to the E-Vote website. 3. User is not logged in. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/**  **Fail** | **Comment** |
| **1** | Login page is opened. | Page is displayed as specified in wireframe 6.1a |  |  |
| **2** | Enter a valid username, password, and date of birth. Leave “SSN” field blank, and click Submit. | Page displays error text “This field is required” under the SSN field. User remains on the Login page. |  |  |
| **3** | Repeat step 2 for each username, password, and date of birth. | Page displays error text under the proper field. |  |  |
| **4** | Enter a correct username, password, date of birth, and SSN, and click Submit. | System logs in the user and begins a session. |  |  |
| **5** | Click the “Log Out” button in the header bar. | System displays the Login page. |  |  |
| **6** | Enter an incorrect username with valid password, date of birth, and SSN, and click Submit. | Page displays generic “Invalid login information” error text under Submit. |  |  |
| **7** | Enter a valid username, date of birth, and SSN with an incorrect password and click Submit. | Page displays generic “Invalid login information” error text under Submit. |  |  |
| **8** | Enter a valid username, password, and SSN, with an incorrect date of birth and click Submit. | Page displays generic “Invalid login information” error text under Submit. |  |  |
| **9** | Enter a valid username, password, and date of birth, with an incorrect SSN and click Submit. | Page displays generic “Invalid login information” error text under Submit. |  |  |
| **10** | Repeat step 4. | Page displays the error text “Your account has been locked for too many failed login attempts. Please try again in 30 minutes.” |  |  |
| **11** | Attempt to enter more than four characters into the SSN field. | Input field does not accept more than four characters. Any further input is not seen. |  |  |
| **12** | Attempt to enter more than twenty characters into the username field. | Input field does not accept more than twenty characters. Any further input is not seen. |  |  |
| **13** | Attempt to enter more than twenty characters into the password field. | Input field does not accept more than twenty characters. Any further input is not seen. |  |  |

# 10 Class Diagram

## 10.1 Class



# 11 Developer Peer Review Approval

## 

