

# Milestone 1 - Ballot Ai

*SW Engineering CSC648/848 Fall 2024 Section 01*

*9/24/2024*

## Team 03

Brenna Gilbert - [bgilbert3@sfsu.edu](mailto:bgilbert3@sfsu.edu) - Team Lead

Ian Knott - [iknott@sfsu.edu](mailto:iknott@sfsu.edu) - Lead Backend

Leonardo Segura - [lsegura1@mail.sfsu.edu](mailto:lsegura1@mail.sfsu.edu) -Frontend

Demetrio Ricardo Calimag - [dcalimag@sfsu.edu](mailto:dcalimag@sfsu.edu) - Frontend

Subhan Khan - [skhan20@sfsu.edu](mailto:skhan20@sfsu.edu) - Backend/Frontend

Adan Matute - [amatute1@sfsu.edu](mailto:amatute1@sfsu.edu) - Backend

Daniel Mitra - [dmitra@sfsu.edu](mailto:dmitra@sfsu.edu) - Github Master

| Date Submitted | Date Revised |
|----------------|--------------|
| 9/24/2024      |              |
|                |              |

## Executive Summary - Daniel Mitra

In today's political landscape, voters are often overwhelmed by the sheer volume of information available about candidates and their policies. With the rise of misinformation and biased sources, making informed voting decisions has become more difficult than ever. Ballot AI seeks to address this issue by offering a comprehensive platform that simplifies the voting process, providing users with reliable, personalized political information. Our goal is to empower voters to make educated decisions that align with their personal values and preferences, fostering a more informed electorate.

Ballot AI is designed to serve a broad audience, including college students, the general public, political officials, and election offices. The platform's AI-driven recommendation system is a key feature, helping users match their interests and concerns with candidates who share similar values. This personalized approach allows voters to cut through the noise and focus on the issues that matter most to them. Additionally, users can search for candidates based on filters like party affiliation, location, and policies, making it easier to find candidates who represent their interests on both local and national levels. For officials, Ballot AI offers tools to manage candidate profiles and events, helping them reach potential voters effectively.

One of the standout features of Ballot AI is the integration of an AI chatbot, which engages users in a conversational manner to gather their political preferences and guide them toward suitable candidates. This approach ensures that even users who may not be well-versed in political terminology can easily navigate the platform and find the information they need. The system also includes a rating feature, allowing voters to share their experiences and opinions about candidates, which in turn helps other users make more informed decisions.

Our team is composed of highly skilled backend and frontend developers, each bringing their expertise to the project alongside a strong lead to guide the team. Together, we will craft a system that not only offers an intuitive user experience but also ensures security and scalability. Using technologies such as Node.js, Amazon RDS, and Remix.js, we've built a platform that is both reliable, responsive, and capable of handling a high volume of users during peak election periods. Our competitive analysis has shown that, unlike other unnamed political information platforms, Ballot AI stands out by offering real-time updates, AI-driven recommendations, and customizable ballot previews.

As a student-led project, we are passionate about leveraging technology to enhance

engagement. Ballot AI has the potential to make a significant impact on voter participation by providing a more accessible and personalized way to navigate the complexities of modern elections. With its innovative features built through our highly skilled backend team and user-friendly interface strengthened by our highly skilled frontend team, Ballot AI is expected to become a valuable tool for voters seeking ease and clarity in a large sea of political information.

## Personas and Main Use Cases - Adan Matute

- Personas
  - a. College Students
    - summaries related to [VOTING]
      - characteristics
        - search for candidates and their past voting history
        - check on how candidates may have kept up with pledges made
        - preference on easily accessible information
      - goals
        - find a candidate that will best support their needs
        - find someone reliable who won't go back on their word and has some integrity
        - give some well-informed ratings for candidates
      - skills
        - years of finding reputable sources for classes applies for searching for reputable sources about candidates
        - quick access to peers to discuss topics of interest, such as candidates and what they do as well as any info in regards to voting
        - tend to be more technologically literate as of late
  - b. General Public
    - summaries related to [VOTING]
      - characteristics
        - pays more attention to clickbait news articles about candidates
        - quick access to information works better than needing to get through large news articles and OpEds
        - might not know every piece of necessary information needed to vote
      - goals
        - elect a candidate who best suits their need

- prevent candidates who they dislike from gaining support through public opinions
  - make sure their vote actually counted towards what they wanted it to
- skills
  - ability to search through countless articles about the vast number of politicians and their policies
  - ability to share information they find with others and compare as well to gain new perspectives
  - ability to rate candidates who may or may not always provide accurate information about how they work for their constituents
- c. Officials from the political parties and election office
  - summaries related to [VOTING]
    - characteristics
      - displaying the best side of party and candidates that their group supports
      - searching for what appeals to users for better marketing campaigns
      - keen on public opinions in regards to candidates, politicians, policies and the likes and how they are viewed among the general public
    - goals
      - put out information that would help their party candidate succeed the most
      - prevent their opponents from gaining public support
      - getting people to vote in their favor
    - skills
      - access to information that would be beneficial for voters
      - can post information relating to candidates
      - can see what users are in most support of and least critical of when it comes to information regarding political candidates and policies
- Use Cases:
  - a. **Accessibility of Information**
    - Students and the general public should be able to easily access any information from the site in regards to the candidates that are running. This could include, but is not limited to, their career history, voting history, public statements made, or articles that hold significance in regards to whomever they are searching. This information is important in making decisions that can have an effect on not just a national level, but a local one as well. Knowing what

you are voting for is important for each and every person.

**b. Addition of Information**

- Political officials serve their candidates and have the primary goal of gaining public support for their candidates. Using this tool, they are able to paint their candidate in the best light that they can, putting out only the necessary details to do so, omitting information that could call for criticisms and including the information that can draw positive attention from users. The opposite should be possible too, such as posting information that was omitted by the opposing party, allowing for some transparency rather than only allowing the good side to be shown to the public.

**c. Rating System**

- The ability for users to rate the candidates can be seen as both a pro and a con in this case as it is dependent on how the information is dealt with. Allowing for free range in ratings without any monitoring would likely not work out too well as there could be false remarks being made which would potentially sway opinions, however, allowing for a rating system that is built well allows the public to look at others experiences with the potential candidates and see whatever information was left out from the information provided by officials. Rating is a system that is full of potential issues that could make it collapse, but it helps with providing information that could be beneficial to users in understanding what their candidates are like.

**d. Search System**

- The primary purpose of the program should be to show information regarding political candidates. The ability to search through the information to provide what is necessary something users will be heavily reliant on, similar to how google populates items that relate to what you are in search of, there is the expectation that articles and information we search for will prompt more than only the input, with more details regarding candidates, policies, and so on being valuable to what users can take in and form their own opinions on. Experiencing situations where there is an information deficit or the search system does not provide you with what you are looking for can be frustrating and we should thus aim to provide whatever is searched for as long as it pertains to voting.

## List of Main Data Items and Entities -Brenna Gilbert

## Key Terms and Entities:

- **User**
  - **Name: User**
  - **Meaning:** Any individual or entity that will interact with the system. There shall be different categories of users with specified privileges and roles.
  - **Usage:** Defines the base attributes and behaviors that users have within the system.
  - **Attributes:** UserID, Username, UserType, Email, IssuesInterest, SearchHistory.
  - **Privileges: Depending on UserType -**
    - **Voter:** Full search access, filters candidates based on criteria, rate candidates, use AI chatbot, and message officials.
    - **General Public:** Limited search capabilities, read-only for certain features, basic interaction with AI chatbot, and restricted messaging.
    - **Official:** Full privileges to manage candidate and voter information, update polling locations, and respond to user messages.
- **Candidate:**
  - **Name: Candidate**
  - **Meaning:** A person running for public office at any level in the United States.
  - **Usage:** Core entity in the system. Candidates represent what users will search for and interact with, including candidate background, policies, and their electoral campaign information.
  - **Attributes:** CandidateID, FullName, PoliticalParty, RunningOffice (President, Governor, Mayor), Policies, Background, Location.
  - **Privileges:** Candidates do not directly interact but Campaign Official manage their profiles.

- **Issue:**
  - **Name: Issue**
  - **Meaning:** A political or social topic that is relevant in an upcoming election and which candidates have positions on.
  - **Usage:** Issues are used for filtering and organizing candidates, this allows users to search based on topics of interest.
  - **Attributes:** IssueID, IssueName, IssueDescription, AssociatedCandidates
- **Voter Information:**
  - **Name: VoterInfo**
  - **Meaning:** Contains details about voter registration, and polling locations.
  - **Usage:** Provides users with necessary information to participate in elections such as deadlines for registration and location of polling stations.
  - **Attributes:** VoterInfoID, RegistrationDate (deadline for registration), PollLocations, VotingInstructions.
- **AI Chatbot**
  - **Name: Chatbot**
  - **Meaning:** An AI powered virtual assistant within the application that helps users identify candidates by asking them about their political interests and guiding them to candidates that match their preferences.
  - **Usage:** Engages with users by gathering input about their associated issues and recommending candidates who align with their values.
  - **Attributes:** ChatbotID, UserInput, MatchCandidate, MatchingCriteria.
- **Search**
  - **Name: SearchResult**
  - **Meaning:** The outcome of a search containing a list of candidates, voter information, or other relevant data.
  - **Usage:** Used to display information based on user queries helping them to identify candidates and voting resources

relevant to their interests on issues.

- **Attributes:** Query, FilteredCandidates, FilteredVoterInfo, ResultsMap.

## List of Functional Requirements - Demetrio Calimag

### 1. General User/Users:

- 1.1. User Registration: A user shall be able to create at most one account.
- 1.2. Email Verification: A registered user shall be sent a verification email.
- 1.3. User Voting Preferences: Users shall be able to input voting preferences
- 1.4. User Feedback and Reporting: Users shall be able to report

### 2. Registered Users:

- 2.1. Login From Multiple Devices: A registered user shall be able to log into the system from multiple devices.
- 2.2. One Email Per Account: A registered user shall only have one email associated with their account.
- 2.3. Account Update: A registered user shall be able to update their account information.
- 2.4. Account Password: A registered user shall have a password.
- 2.5. Account Date of Birth: A registered user shall have a date of birth.
- 2.6. Account Age: A registered user shall have an age.
- 2.7. In-Site messaging: Registered users shall communicate with one another via a secure, in-site webmail system.



2.8. Roles: A registered user shall be a Official, Political Party, or Election Office

2.9. Username: Registered user shall have a username.

2.10. Name: Registered user shall have a name but are not verified.

### **3. Officials/Political Parties:**

3.1. Candidate Information Submission: Officials/Political Parties shall be able to submit and manage information.

3.2. Event Listings: Officials/Political Parties shall be able to post events.

3.3. Name and Photo: Official/Political Party shall have a verified full name and photo.

3.4. Background: Official/Political Party shall have a background

3.5. Policies: Official/Political Party shall have policies

3.6. County: Official/Political Party shall have a location and county

3.7. Registered User: Officials/Political Party shall be a registered and verified user

3.8. Political Party: Official shall have one political party

3.9. Date Of Birth: Officials shall have a Date of Birth

### **4. Elections Office:**

4.1. Election Information: Election offices shall be able to enter election information.

4.2. Registered User: Election offices shall be a registered and verified user

4.3. Election Office Name: Election office shall have a verified name

4.4. Location: Election Office shall have a verified location.

## List of Non-Functional Requirements - Subhan Khan

### 1. Portability

1.1 The site shall be deployable on different operating systems and cloud platforms with minimal configuration changes.

1.2 The code shall be written in a way that allows for easy migration to different web servers or cloud providers.

### 2. Security

2.1 All user data shall be protected through encryption to ensure secure data transmission.

2.2 The application shall implement authentication mechanisms, including email verification.

2.3 Regular security audits shall be conducted, ensuring that vulnerabilities like SQL injection, cross-site scripting, and cross-site request forgery are mitigated.

### 3. Maintainability

3.1 The codebase shall follow industry standards for readability, including consistent code formatting, detailed comments, and modular design to ensure easy maintenance and updates.

3.2 Comprehensive documentation shall be provided for all modules, including API endpoints and database schemas, to facilitate future

development and troubleshooting.

3.3 Bug tracking and version control shall be used to manage changes and updates efficiently.

#### **4. Reliability**

4.1 The application shall be highly reliable with a target uptime of all time, particularly during peak election periods.

4.2 Automated backups shall be performed daily, with a disaster recovery plan in place to restore the system within 4 hours of a major failure.

4.3 All key functions shall have error-handling mechanisms to ensure continuous operation even in case of partial system failures.

#### **5. Scalability**

5.1 The application shall be designed to handle an increase in user traffic without performance degradation, supporting many users.

5.2 The architecture shall be modular, allowing for the addition of new features or services without major changes to the existing codebase.

#### **6. Performance**

6.1 Page load times shall not exceed 5 seconds under normal usage conditions.

6.2 Database queries and AI-based searches shall return results quickly with most inquiries.

6.3 The system shall have a latency of less than 1 seconds for user interactions, such as clicking on links or submitting forms.

#### **7. Reusability**

7.1 The application shall be developed using reusable components, such as UI elements, APIs, and backend services, to minimize redundant code and facilitate future projects.

7.2 Core functionalities, such as the search and rating systems, shall be built as modular components that can be reused in other similar applications.

## 8. Flexibility

8.1 The system architecture shall be flexible to accommodate future changes, such as adding new features or adapting to other countries' elections.

8.2 The application shall allow for the easy addition of new user roles, functionalities, or changes in the user interface without extensive refactoring.

8.3 The system shall support configuration changes without requiring significant modifications to the core code.

## Competitive Analysis - Leonardo Segura

| Features                           | VOTE411 | Ballotpedia | Voterly | Our Voting Guide /w AI |
|------------------------------------|---------|-------------|---------|------------------------|
| AI-Based candidate recommendations | No      | No          | No      | Yes                    |
| Personalized voting suggestions    | Yes     | No          | Yes     | Yes                    |
| Candidate Policy Analysis          | Yes     | Yes         | Yes     | Yes                    |
| Easy to use interface              | Yes     | Yes         | Yes     | Yes                    |

|                              |    |    |    |          |
|------------------------------|----|----|----|----------|
| Customizable ballot previews | No | No | No | Planning |
| Real-time election updates   | No | No | No | Planning |

Our product will try to improve past our competitors by offering AI-based candidate recommendations and real time elections updates, which are features that the three VOTE411, Ballotpedia, and Voterly lack in. We will also include Customizable ballot previews which will allow users to input their location and generate a preview of their ballot tailored to their area. With the mix of AI, real time updates, and customizable ballots positions our product will be a more personalized and dynamic option for people to use to make informed voting decisions.

## High-level System Architecture and Technologies Used- Ian

The API's provided utilize data that if we were to try to replicate ourselves would require a massive amount of resources as we would need to pull candidate information from each of their provided webpages. And thanks to the following apis we will easily be able to pull information in about candidates, election times, and polling places. This will give us the ability to store less information in our own database which reduces cost.

### Frameworks:

Express.js(<https://expressjs.com/>), Bootstrap(<https://getbootstrap.com/>),  
Remix.js(<https://remix.run/>)

### APIs:

Financial Backing Candidate information // free

(<https://api.open.fec.gov/developers/>)

Candidate information and Polling Locations //Not Free

([https://developers.democracy.works/api/v1?utm\\_source=dwwebsite&utm\\_medium=footer](https://developers.democracy.works/api/v1?utm_source=dwwebsite&utm_medium=footer))

Candidate and election information //free

(<https://developers.google.com/civic-information/>)

**Database:**

Amazon Relational Database Service(RDS) managed through MySQL v. 8.0

(<https://aws.amazon.com/rds/?p=ft&c=db&z=3> )

**Version Control:**

Git CLI and Github

**APP Server:**

EC2 AWS

**Server-Side Language:**

Node.js (<https://nodejs.org/en> )

**Additional Technologies:**

IDE VSCode, Docker extensions, Live Share, GitLens, Google Analytics, coPilot, IntelliJ.

**Server Host:**

AWS 1vCPU 2GB RAM Operating System: Ubuntu 16.04 Server

## Team and Roles

Brenna Gilbert - bgilbert3@sfsu.edu - Team Lead

Ian Knott - Iknott@sfsu.edu - Lead Backend

Leonardo Segura - lsegura1@mail.sfsu.edu - Frontend

Demetrio Ricardo Calimag - dcalimag@sfsu.edu - Frontend

Subhan Khan - skhan20@sfsu.edu - Backend/Frontend

Adan Matute - amatute1@sfsu.edu - Backend

Daniel Mitra - dmitra@sfsu.edu - Github Master

## Checklist

- Team found a time slot to meet outside of the class (Done)
- Github master chosen (Done)
- Team decided and agreed together on using the listed SW tools and deployment server (Done)
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and Practicing (Done & On Track)
- Team lead ensured that all team members read the final M1 and agree/understand it before submission (On Track)
- Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc. (On track)