eVoting System

Requirements Analysis Document

Team 5:

Rosalie Thobe

Donovan Todd

John Kim

Phi Nguyen

1. Introduction
   1. Purpose of the system
   2. Scope of the system
   3. Objectives and success criteria of the project
   4. Definitions, acronyms, and abbreviations
   5. References
2. Proposed System

2.1 Overview

2.2 Functional Requirements

2.3 Non-functional Requirements

2.3.1 Usability

2.3.2 Reliability

2.3.3 Performance

2.3.4 Supportability

2.3.5 Implementation

2.3.6 Interface

2.3.7 Packaging

2.3.8 Legal

2.4 System Models

2.4.1 Scenarios

2.4.2 Use Cases

2.4.3 Domain Object Model

2.4.4 Sequence Diagrams

2.4.5 User Interface

**1. Introduction**

* 1. Purpose of the system

Voting is a hassle. If you decide to cast your vote in person on the day of an election, then you first have to take time off your schedule to vote since elections always seem to fall on inconvenient days. Next, you have to drive out to the voting site and wait in line for your turn. Only then do you get to cast your vote. If you decide to mail your vote, then you have to actually use the postal system! You also have to hope that your vote arrives at the election office on time to be valid.

Not anymore! The eVote Electronic Voting Management System will be an online voting site that will allow voters easier access to the democratic process and election officers can run the elections with less resources. It will have all the standard voting essentials just with 21th century convenience!

* 1. Scope of the system

The system will have election officers to create elections and ballot items. These will be available to view for voters on a public website. Voters will then have the ability to register to vote on the website by entering their personal information. Once registered, voters will cast their votes. Once the voting is complete, the election officer will close the election and publish the results, so that voters can see the results.

* 1. Objectives and success criteria of the project

The system must successful achieve the following to be consider successful:

* Allow multiple users to access it over the Internet simultaneously
* Perform authentication on users before allowing any access to the system
* Create new user profiles divided into two groups: election officers and voters
* Permit election officers to create elections, ballots, etc.
* Permit election officers to save the above and edit/delete them later
* Let voters cast votes on the elections
* Permit election officers to publish the results
  1. Definitions, acronyms, and abbreviations

*The system*  Short hand for the eVote Electronic Voting Management System

*Election Officers* People with the power to create new elections, along with related things, and close elections.

*Voters* People who can cast votes on the elections created by the election officers. A person can be a voter and an election officer, but requires two separate profiles to do so.

*Profiles* A file that contains all relevant information related to the user. The profiles of voters and election officers are separate, so election officers will have two.

*Elections* Things that voters can cast votes on. Consists of ballots and referendums.

*Ballots/referendums* Issues that voters can cast votes to decide on. Makes up elections.

* 1. References
* CSCI 4050/6050 Term Project Description
* Answers User Stories

**2. Proposed System**

**2.1 Overview**

This describes the functionality of the proposed system. It is divided into several sections based on the function.

**2.2 Functional Requirements**

Voter’s User Stories

1. As the voter, I want to be able to login to the system so I can access the ballots and cast my vote.
   1. If I enter the correct username and password, I want to get into the eVote system
   2. If I enter the wrong username or password, a little message will appear about how either of the fields are incorrect
   3. If I forget my username/password, I want to have an option to change my password or retrieve my username. The new password or username will be sent to my email.
   4. 10 minutes of inactivity, I will get logged out for safety purposes
2. As the voter, I want to be able to logout of my account so that someone else cannot access it.
   1. After logging out, there are no other functions except to sign in again, register, or request new password or username.
3. As the voter, I want to be able to register or unregister to vote so that I am able to vote or not vote when there is an election.
   1. I must enter all the information (name, email, phone number, address, username, password, and age) correctly.
   2. If username is already in use, an error will pop up telling voter that the username is already in use
   3. Once all the forms are filled out appropriately, I have registered successfully and now am able to start the voting process on my new profile.
4. As the voter, I want to be able to unregister so that I am able to register as a voter in a different area, if I were to move.
   1. Voter is unable to log in and must create a new account in order to use the system functions again.
5. As the voter, I want to be able to maintain my voter profile so that it has the most current information.
   1. If I wanted to change any of my information to keep it up to date, all blanks must still be filled out correctly.
   2. All form items can be changed except for the voter ID.
6. As the voter, I want to be able to vote on various issues during elections so that my voice can be heard.
   1. A ballot that displays all the current elections and issues custom to the voter’s district and include all the candidates and alternate candidates.
   2. The identity of the voter is not included once having voted
   3. The voter has choice to vote on all or some of the items on the ballot.
7. As the voter, I want to be able to view a ballot in order to decide ahead of time on how to vote for when the election opens.
   1. The ballot must stay up to date as well as customized to the voter’s electoral district. Also the ballot must include issues and elections that have all the candidates and alternative candidates.
8. As the voter, I want to be able to see the results of the election in order to figure out who won the election.
   1. All ballots for all electoral districts are displayed
   2. The winner is clearly identified
   3. The results are displayed by the numbers of votes per candidate.
   4. The approval/disapproval for each issue is clearly identified
9. As the voter, I want to be able to have the option of resetting my password if I happen to forget it while trying to log in.
   1. An email will be sent to the email that the user puts in.
10. As the voter, I want to be able to have the option of retrieving my username if I happen to forget it while trying to log in.
    1. An email will be sent to the registered email with the current username.
11. As the voter, I want to be able to have the option of canceling the vote before hitting submit just in case I decide not to vote in that particular election/issue.
12. As the voter, I want to be able to know if I have entered my username or password wrong so I know that I need to retype the username/password.
    1. A message will appear that either username/password is incorrect, but not specify which one.
13. As the voter, I want a warning or error message if I click on an election or issue that I have already voted on.
14. As the voter, I want an error message or confirmation message that my vote went through.
15. As the voter, I want a confirmation message that I have logged out of my profile successfully.

Elections Officer User Stories

1. As the elections officer, I want to be able to login to the system so I can do various election-related things.
   1. When I enter the correct username and password, I will be able to log into the system
   2. I want the ability to request for a new password, which will be sent to the EO’s email.
   3. After 10 minutes of inactivity, I want the system to log me out automatically so that no one else can get into my account.
2. As the elections officer, I want to be able to logout of my account so that someone else cannot access it.
   1. Once I have logged out, I can no longer access the system’s functions and can only register, sign in, and request password/username.
3. As the elections officer, I want to be able to create elections, ballots, electoral districts, candidates, political parties and issues so that people have something to vote on when the time comes.
4. As the elections officer, I want to be able to create an election so that the voters are able to vote.
   1. The position of the office for election is identified
   2. The position is unique, else an error message is displayed
   3. Election is marked as either partisan or non partisan
   4. The election has an electoral district
   5. Candidates can be created if needed
5. As the elections officer, I want to be able to create an election alternate candidate so that the voters have the choice to vote on alternates.
   1. An election alternate candidate is created for an election (goes with creating an election)
   2. Name of alternate candidate is identified and is unique.
   3. Political party (if election is partisan) is selected.
   4. I can reorder the already defined alternate candidates.
6. As the elections officer, I want to be able to view an election in order to see if the election needs to be modified.
   1. The name of office position for the election is given and any matching elections.
   2. If no election is provided, all the elections are displayed
   3. If no elections match the search, a message is displayed
   4. Complete data for selected election is displayed.
7. As the elections officer, I want to be able to view an electoral district in order to see if the voter districts need to be adjusted.
   1. The name of district is shown
   2. The name of district is unique
   3. An error message is displayed if the name of district is not unique.
8. As the elections officer, I want to be able to view the elections candidates so that I can determine if the list needs to be updated as more or less candidates run.
   1. The full name of the candidate is shown and list of matching candidates is also shown.
   2. When nothing is entered in search box, the entire list of candidates is shown
   3. If candidate could not be found, an error message is shown
   4. Name, election office, district, and political party are displayed next to selected candidate.
9. As the elections officer, I want to be able to view the elections alternate candidates so that I can determine if the list needs to be updated as more or less candidates run.
   1. The full name of the alternate candidate is shown and list of matching candidates is also shown.
   2. When nothing is entered in search box, the entire list of candidates is shown
   3. If alternate candidate could not be found, an error message is shown
   4. Name, election office, district, and political party are displayed next to selected alternate candidate.
10. As the elections officer, I want to be able to view a political party in order to determine if the list of political parties needs to be updated.
    1. Political party’s full name is shown along with a list of matching electoral districts
    2. If nothing is typed in search box, a list of all political parties is shown
    3. If the political party cannot be found, an error message will appear
    4. Name of selected political party is displayed
11. As the elections officer, I want to be able to view an issue so that I may modify them as need be at a later time.
    1. Full or part of the question of issue is provided and list of any matching issues is also displayed
    2. When an empty question is provided, a list of all issues is visible.
    3. If no issues match the search, a message is provided that it can’t be found
    4. The full question and electoral district for the selected issue.
12. As the elections officer, I want to be able to view a ballot so that I may modify the current ballots as need be at a later time.
    1. A ballot can be chosen from a list of all ballots
    2. If no ballots are found, a message is displayed
    3. All issues and elections are displayed including the electoral districts, political parties, start and end times.
13. As the elections officer, I want to be able to update an election so that I can modify the existing elections if need be.
    1. The position of the office for election is identified
    2. The position is unique, else an error message is displayed
    3. Election is marked as either partisan or non partisan
    4. The election has an electoral district
    5. The candidates running can be updated if needed.
14. As the elections officer, I want to be able to update the election’s candidates so that I can modify the existing elections candidates if one decides to drop out or another decides to run.
    1. An election candidate is created for an election (goes with creating an election)
    2. Name of candidate is identified and is unique.
    3. Political party (if election is partisan) is selected.
    4. I can reorder the already defined candidates.
15. As the elections officer, I want to be able to update an electoral district so that I can modify the existing electoral districts, as needed.
    1. The name of district is shown
    2. The name of district is unique
    3. An error message is displayed if the name of district is not unique.
16. As the elections officer, I want to be able to update an election’s alternate candidates so that I can modify the existing alternate candidates if they decide to join or drop out.
    1. An election alternate candidate is created for an election (goes with creating an election)
    2. Name of alternate candidate is identified and is unique.
    3. Political party (if election is partisan) is selected.
    4. I can reorder the already defined alternate candidates.
17. As the elections officer, I want to be able to update a political party so that I can modify the existing political parties as need be.
    1. Name of political party is identified and unique
    2. If name is not unique, error message is displayed.
18. As the elections officer, I want to be able to update an issue so that the existing issues can be modified or add or remove issues as they change.
    1. An issue is for an election being created (goes along with an election)
    2. An issue’s question is identified and unique
    3. If issue’s question is not unique, an error message is given
19. As the elections officer, I want to be able to update a ballot in case I want to modify an existing ballot as needed.
    1. The open and close dates are identified
    2. The EO can create an election or issue
    3. The defined elections or issues can be rearranged.
20. As the elections officer, I want to be able to delete the election so that I can remove any existing elections that are no longer in process or relevant.
    1. The election is located and displayed
    2. EO must confirm deletion of election before actually removing it
    3. Candidates/ alternate candidates must be removed/deleted before the election can be deleted. An error message is displayed
    4. EO can undo the action immediately after deletion and can restore the election.
21. As the elections officer, I want to be able to delete an electoral district in case an electoral district needs to be removed.
    1. The electoral district is located and displayed
    2. EO must confirm deletion of electoral district before actually removing it
    3. Elections or issues must be removed/deleted before the election can be deleted. An error message is displayed
    4. EO can undo the action immediately after deletion and can restore the electoral district.
22. As the elections officer, I want to be able to delete the election candidates so that I can remove any candidates that no longer want to run.
    1. An election, along with it’s candidates, is located and displayed
    2. EO has to confirm the deletion before actual removal of the elections candidates
    3. An undo can occur immediately after deletion and can restore the candidate.
23. As the elections officer, I want to be able to delete an election alternate candidate just in case an alternate candidate drops out.
    1. An election, along with it’s alternate candidates, is located and displayed
    2. EO has to confirm the deletion before actual removal of the elections alternate candidates
    3. An undo can occur immediately after deletion and can restore the alternate candidate.
24. As the elections officer, I want to be able to delete a political party just in case the political party needs to be removed.
    1. Political party is located and displayed
    2. EO must confirm the deletion of political party before actually deleting the party
    3. If candidates/alternate candidates belong to political party exist, removal is not performed and an error message is displayed
    4. EO can undo removal immediately after deletion and can restore the party
25. As the elections officer, I want to be able to delete an issue just in case the existing issue is no longer relevant.
    1. An issue, is located and displayed
    2. EO has to confirm the deletion before actual removal of the issue.
    3. An undo can occur immediately after deletion and can restore the issue.
26. As the elections officer, I want to be able to delete a ballot just in case the existing ballot is no longer relevant.
    1. Ballot has been located and displayed.
    2. EO must confirm deletion before removal of ballot.
    3. If confirmed, deletion of ballot means removing all issues and elections, candidates/alternate candidates that are on the ballot.
    4. EO can undo of deleted ballot if it occurs immediately after.
27. As the election officer, I want to be able to assign any number of districts to a certain ballot item so that other districts may have a chance to vote on the election and/or issue.
    1. A vote that is casted by a voter on an issue or election must be consistent with the voter’s electoral district assignment.
28. As the elections officer, I want to be able to open and close the voting system in order to have a set time of when voters can vote on the issues/elections.
    1. At least one ballot exists
    2. The times and dates are correct as specified on the ballot of when to open and end ballot
    3. If the current date is not correct, an error message will display
    4. After system is closed for voting, no more votes are allowed
29. As the elections officer, I want to be able to validate and publish voting results so that the public is able to know the results of the election and/or issues.
    1. All votes casted by each voter on an issue or election is consistent with the voter’s electoral district assignment
    2. No one can vote twice
    3. Dates of all votes are cast within the certain time slot of the ballot they voted on.
30. As the elections officer, I want to be able to modify and manage my profile in order to keep it up to date.
    1. New values for the profile information can be updated as long as they are in the appropriate formats
    2. All attributes except for the login name can be changed
31. As the election officer, I want to be able to define the requirements of being a voter because not everyone should be able to vote such as little kids.
32. As the election officer, I want to be able to unregister voter profiles because they may be duplicates or unwanted voter profiles.
33. As the election officer, I want to be able to see the people who are registered under my district so that voters are organized according to district.
34. As the elections officer, I want to be able to have the option of resetting my password if I happen to forget it while trying to log in.
35. As the elections officer, I want to be able to have the option of retrieving my username if I happen to forget it while trying to log in.
36. As the election officer, I want to make sure that I log in as an election’s officer and not a voter by having unique username.

**2.3 Non-Functional Requirements**

2.3.1 Usability

* The system must have an easy-to-use user interface (UI).
* The UI must be a graphical user interface (GUI).
* The UI must have screens design for each part of the system’s functionality.
* The UI must be designed appropriately for election officers and voters.

2.3.2 Reliability

* The system must allow multiple users to access the system simultaneously without issues.

2.3.3 Performance

* The system must be online and responsive at all time with downtime kept at an absolute minimum.

2.3.4 Supportability

* The system must be easy to maintain and modify in the future.

2.3.5 Implementation

* The system must be implemented mainly with the Java programming language.
* The system must use accepted standards whenever possible (HTML, CGI, etc.).
* The system must use a persistent data store (MySQL RDMBS) for all data so that the system’s cost is low.

2.3.6 Interface

* Code must be able to run on the Linux operating system.
* The system must be accessible by the most common Web browsers such as Mozilla Firefox, Google Chrome, Safari, and Internet Explorer.

2.3.7 Packaging

* All code must be located and running on the server located at um2.cs.uga.edu.

2.3.8 Legal

* The system must comply all laws related to information security.
* System must use established standards so that it is maintainable.

**2.4 System Models**

2.4.1 Scenarios

|  |  |
| --- | --- |
| Scenario Name: | creatingElection |
| Actors: | Steve: Elections Officer |
| Flow of Events: | 1. Steve is an elections officer who needs to create an election for a seat in local government so that voters can vote on it. 2. Steve decides to use the evote system and opens the evote system on his desktop computer and enters his username and password for his previously made account before clicking login. 3. Steve navigates to the tools section on the homepage and hits create an election. 4. Steve then creates a political party for Republicans and another one for Democrats. 5. Steve then creates two election candidates and denotes one as a Republican and one as a Democrat. 6. Steve then creates an additional alternative candidate. 7. Steve then creates the electoral district for the election. 8. Steve then opens the election up for voting in the specified electoral district. 9. Voters vote in the election over a specified amount of time. 10. At any time Steve can access the system to view, delete and modify the election, electoral district, political parties or candidates. 11. Steve closes the election for voting once the time has expired and validates the election results. 12. Steve publishes the election results so that the public can view it and see who the winner is. 13. Steve can choose to create more elections or issues as needed. |

|  |  |
| --- | --- |
| Scenario Name: | creatingIssue |
| Actors: | Steve: Elections Officer |
| Flow of Events: | 1. Steve is an elections officer who needs to create an issue for an upcoming issue so that voters can vote on it and he decides to use the evote system. 2. Steve opens the evote system on his desktop computer and enters his username and password for his previously made account and then clicks login. 3. Steve navigates to the tools section on the homepage and clicks creates an issue. 4. Steve then creates the electoral district for the specified issue. 5. Steve then creates the issue and then writes a brief description of the issue and specifies that the voter can choose yes or no. 6. Steve then opens the system up for voting in the specified electoral district. 7. Voters vote in the election over a specified amount of time. 8. At any time Steve can access the system to view, delete and modify the issue or electoral district. 9. Steve closes the system for voting once the time has expired. 10. The issue is then either given the vote of yes or no depending on how the voters voted. 11. Steve can choose to create more elections or issues as needed. |

|  |  |
| --- | --- |
| Scenario Name: | voterRegistration |
| Actors: | Voter: Kyle |
| Flow of Events: | 1. Kyle, a current resident of Athens Clarke County, recently realized that he wanted to be able to vote on current issues and elections that are occurring in Athens. 2. He goes to register to vote on the online eVoting system, but realized he has not registered to vote yet. 3. Kyle signs up to vote and makes an account. 4. Kyle enters in his name, address, phone number, email, social security number, username, and password and presses enter 5. Kyle now has access to his profile of voting. This means he can update his username and password as well as delete his voter registration. 6. Kyle can also view ballots and elections that are occurring or will soon be occurring in his area. 7. Kyle logs out of his account until he wants to vote on an issue/election. |

|  |  |
| --- | --- |
| Scenario Name: | voter |
| Actors: | Voter: Kyle |
| Flow of Events: | 1. Kyle wants to vote on a current election that is happening in Athens Clarke County. 2. Kyle signs into his account that he previously made. 3. Kyle selects the election that he wants to vote on. 4. Kyle must wait until the election is open for voting. 5. Once the voting system has been opened for the particular election, Kyle can select his vote. 6. Once Kyle submits his vote, he is unable to change his vote. 7. Kyle must wait for the voting system to close and for the system to publish the results. 8. Once the results have been published, Kyle logs into his account and views the results. 9. Once Kyle sees the results, he logs out of his account. |

2.4.2 Use Cases

Use Cases:

|  |  |
| --- | --- |
| Name | LogIn |
| ID | LOGIN-1 |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | This use case details the login steps of an Elections Officer; it is necessary to use the functions of the system. |
| Basic Path | 1. The system displays a prompt for the username and password. The login button is also displayed by the system. 2. The ElectionsOfficer enters the username and password and then his the login button. 3. The system locates the ElectionsOfficer’s user object in the database. The system verifies the ElectionsOfficer’s credentials and access rights. Then the system displays the overview of the ElectionsOfficer’s account. |
| Alternative Paths | N/A |
| Exception Paths | If in step 3 the ElectionsOfficer’s user object cannot be located in the database, the system displays the error message SystemUnavailable and terminates the use case.  If in step 3 the system cannot verify ElectionOfficer’s access rights, the system displays the error message UnknownUser and terminates the use case. |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer decides to use the evote system. |
| Assumption | N/A |
| Pre-conditions | N/A |
| Post-conditions | The ElectionsOfficer’s user object is available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | LogOut |
| ID | LOGOUT-1 |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | This use case details the logout steps of an Elections Officer; it is necessary to exit the system. |
| Basic Path | 1. ElectionsOfficer selects the Logout button. 2. The system logs the ElectionOfficer out and displays the start page. |
| Alternative Paths | N/A |
| Exception Paths | N/A |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to logout. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. |
| Post-conditions | The user is no longer logged in. The application displays a prompt for the username and password. The login button is also displayed by the application. |

|  |  |
| --- | --- |
| Name | Create a new Election |
| ID | CreateElection |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | Use case to create a new Election for an Elections Officer. |
| Basic Path | 1. The system displays the prompt to provide the following attributes of the new Election: name; Done and Cancel buttons are visible. 2. The ElectionsOfficer provides the required attribute values and clicks Done. 3. The system creates a new Election with the provided data and the current ElectionsOfficer is set as the owner of the new Election. 4. The system displays a screen showing all of the Election attributes provided so far. The screen displays three buttons: CreatePoliticalParty, Done, and Cancel, as well. 5. The ElectionsOfficer clicks Done. 6. The system displays a screen showing all of the Election attributes provided so far, as well as PoliticalParties provided so far. The screen displays three buttons: CreateCandidates, Done, and Cancel, as well. 7. The ElectionsOfficer clicks Done. 8. The system displays a screen showing all of the Election attributes provided so far as well as PolitcalParties and Candidates provided so far. The screen displays three buttons: CreateElectoralDistrict, Done, and Cancel, as well. 9. The ElectionsOfficer clicks Done. 10. The system displays a screen showing all of the Election attributes provided so far as well as PoliticalParties, Candidates, and ElectoralDistricts provided so far. The screen displays two buttons: Done and Cancel, as well. 11. The ElectionsOfficer clicks Done. |
| Alternative Paths | 1. In steps 2, 5, 7, 9, 11 if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | 1. In step 3, if the system determines that an Election with the user inputted name already exists, the system displays the error message DuplicateElectionName and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points | 1. In step 4, if the user selects CreatePoliticalParty, the system executes the CreatePolitcalParty use case extension; at its completion, the system displays a screen showing all of the Election attributes and PoliticalParties provided so far and goes to step 5 in the Basic Path. 2. In step 6, if the user selects CreateCandidates, the system executes the CreateCandidates use case extension; at its completion, the system displays a screen showing all of the Election attributes, as well as PoliticalParties and Candidates provided so far and goes to step 7 in the Basic Path. 3. In step 6, if the user selects CreateElectoralDistrict, the system executes the CreateElectoralDistrict use case extension; at its completion, the system displays a screen showing all of the Election attributes, as well as PoliticalParties, Candidates, and ElectoralDistricts provided so far and goes to step 7 in the Basic Path. |
| Triggers | The ElectionsOfficer has elected to create a new Election. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. |
| Post-conditions | The new Election with the defined attributes is available for use. Its owner is the current ElectionsOfficer and its PoliticalParties, Candidates, and ElectoralDistricts are set as specified. |

|  |  |
| --- | --- |
| Name | Create a new Issue |
| ID | CreateIssue |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | Use case to create a new Issue for an Elections Officer. |
| Basic Path | 1. The system displays the prompt to provide the following attributes of the new Issue: name and description; Done and Cancel buttons are visible. 2. The ElectionsOfficer provides the required attribute values and clicks Done. 3. The system creates a new Issue with the provided data and the current ElectionsOfficer is set as the owner of the new Issue. 4. The system displays a screen showing all of the Issue attributes provided so far. The screen displays three buttons: CreateElectoralDistrict, Done, and Cancel, as well. 5. The ElectionsOfficer clicks Done. 6. The system displays a screen showing all of the Issue attributes provided so far as well as ElectoralDistricts provided so far. The screen displays two buttons: Done and Cancel, as well. 7. The ElectionsOfficer clicks Done. |
| Alternative Paths | 1. In steps 2, 5, 7, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | 1. In step 3, if the system determines that an Issue with the user inputted name already exists, the system displays the error message DuplicateIssueName and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points | 1. In step 4, if the ElectionsOfficer selects CreateElectoralDistrict, the system executes the CreateElectoralDistrict use case extension; at its completion, the system displays a screen showing all of the Issue attributes and ElectoralDistricts provided so far and goes to step 5 in the Basic Path. |
| Triggers | The ElectionsOfficer has elected to create a new Issue. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. |
| Post-conditions | The new Issue with the defined attributes is available for use. Its owner is the current ElectionsOfficer and its ElectoralDistricts are set as specified. |

|  |  |
| --- | --- |
| Name | Create a new Electoral District |
| ID | CreateElectoralDistrict |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to create a new Electoral District to an Issue or Election; used by CreateIssue and CreateElection use cases |
| Basic Path | 1. The system displays the prompt to enter: name, area; FindElectoralDistrict, Done and Cancel buttons are visible. 2. The ElectionsOfficer provides the required information and clicks Done. 3. The system creates a new ElectoralDistrict with the provided data or uses an existing one, if one was selected during the FindElectoralDistrict use case execution. The ElectoralDistrict is linked to the Issue or Election being created or modified in the referring use case. The system displays a new screen in which all the created ElectoralDistricts are shown. The screen displays 3 buttons: CreateElectoralDistrict, Done and Cancel as well. |
| Alternative Paths | 1. In step 2 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. 2. In step 2, if the user selects the FindElectoralDistrict button, the system executes the FindElectoralDistrict use case; at its resolution, the system returns to step 2 in the Basic Path, and if the ElectionsOfficer selected one of the already created ElectoralDitricts during the execution of the FindElectoralDistrict use case, all of the attributes of the selected ElectoralDistrict are preset as if entered by the user; otherwise no attributes are set in advance. 3. In step 3, if the ElectionsOfficer selects the CreateElectoralDistrict button, the system goes back to step 1 in the Basic Path. |
| Exception Paths | 1. In step 3, if the system determines that an ElectoralDistrict with the user inputted name already exists, the system displays the error message DuplicateElectoralDistrictName and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points | 1. In step 3, if the ElectionsOfficer selects FindElectoralDistrict, the system executes the FindElectoralDistrict use case extension; at its completion, the system displays a screen showing all of the ElectoralDistricts provided so far and goes to step 2 in the Basic Path. |
| Triggers | The ElectionsOfficer has elected to create a new Electoral District within the CreateIssue or CreateElection use cases. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Election or Issue from the extended use case is available. |
| Post-conditions | A new ElectoralDistrict is added to the Issue or Election. |

|  |  |
| --- | --- |
| Name | Create a new Political Party |
| ID | CreatePoliticalParty |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to create a new Political Party for an Election; used by CreateElection use cases |
| Basic Path | 1. The system displays the prompt to enter: name; FindPoliticalParty, Done and Cancel buttons are visible. 2. The ElectionsOfficer provides the required information and clicks Done. 3. The system creates a new PoliticalParty with the provided data or uses an existing one, if one was selected during the FindPoliticalParty use case execution. The PoliticalParty is linked to the Election being created or modified in the referring use case. The system displays a new screen in which all the created PoliticalParties are shown. The screen displays 3 buttons: CreatePoliticalParty, Done and Cancel as well. |
| Alternative Paths | 1. In step 2 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. 2. In step 2, if the user selects the FindPoliticalParty button, the system executes the FindPoliticalParty use case; at its completion, the system returns to step 2 in the Basic Path, and if the ElectionsOfficer selected one of the already created PoliticalParty during the execution of the FindPoliticalParty use case, all of the attributes of the selected PoliticalParty are preset as if entered by the user; otherwise no attributes are set in advance. 3. In step 3, if the ElectionsOfficer selects the CreatePoliticalParty button, the system goes back to step 1 in the Basic Path. |
| Exception Paths | 1. In step 3, if the system determines that a PoliticalParty with the user inputted name already exists, the system displays the error message DuplicatePoliticalPartyName and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points | 1. In step 3, if the ElectionsOfficer selects FindPoliticalParty, the system executes the FindPoliticalParty use case extension; at its completion, the system displays a screen showing all of the PoliticalParties provided so far and goes to step 2 in the Basic Path. |
| Triggers | The ElectionsOfficer has elected to create a new PoliticalParty within the CreateElection use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Election from the extended use case is available. |
| Post-conditions | A new PoliticalParty is added to the Election. |

|  |  |
| --- | --- |
| Name | Create a new Candidate |
| ID | CreateCandidate |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to create a new Candidate for an Election; used by CreateElection use case |
| Basic Path | 1. The system displays the prompt to enter: name and political party; FindCandidate, Done and Cancel buttons are visible. 2. The ElectionsOfficer provides the required information and clicks Done. 3. The system creates a new Candidate with the provided data or uses an existing one, if one was selected during the FindCandidate use case execution. The Candidate is linked to the Election being created or modified in the referring use case. The system displays a new screen in which all the created Candidates are shown. The screen displays 3 buttons: CreateCandidate, Done and Cancel as well. |
| Alternative Paths | 1. In step 2 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. 2. In step 2, if the user selects the FindCandidate button, the system executes the FindCandidate use case; at its completion, the system returns to step 2 in the Basic Path, and if the ElectionsOfficer selected one of the already created Candidate during the execution of the FindCandidate use case, all of the attributes of the selected Candidate are preset as if entered by the user; otherwise no attributes are preset. 3. In step 3, if the ElectionsOfficer selects the CreateCandidate button, the system goes back to step 1 in the Basic Path. |
| Exception Paths | 1. In step 3, if the system determines that a Candidate with the user inputted name already exists, the system displays the error message DuplicateCandidateName and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points | 1. In step 3, if the ElectionsOfficer selects FindCandidate, the system executes the FindCandidate use case extension; at its completion, the system displays a screen showing all of the Candidates provided so far and goes to step 2 in the Basic Path. |
| Triggers | The ElectionsOfficer has elected to create a new Candidate within the CreateElection use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Election from the extended use case is available. |
| Post-conditions | A new Candidate is added to the Election. |

|  |  |
| --- | --- |
| Name | Find an existing Electoral District |
| ID | FindElectoralDistrict |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to find an existing Electoral District; used by CreateElectoralDistrict use case |
| Basic Path | 1. The system displays all created ElectoralDistricts. Done and Cancel buttons are also visible. 2. The ElectionsOfficer selects a created ElectoralDistricts and clicks Done. |
| Alternative Paths | 1. In step 1, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to find an already created Electoral District within the CreateElectoralDistrict use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The CreateElectoralDistrict from the extended use case is available. |
| Post-conditions | An already created ElectoralDistrict is added to CreateElectoralDistrict. |

|  |  |
| --- | --- |
| Name | Find an existing Political Party |
| ID | FindPoliticalParty |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to find an existing Political Party; used by CreatePoliticalParty use case |
| Basic Path | 1. The system displays all created PoliticalParties. Done and Cancel buttons are also visible. 2. The ElectionsOfficer selects a created PoliticalParty and clicks Done. |
| Alternative Paths | 1. In step 1, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to find an already created Political Party within the CreatePoliticalParty use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The CreatePoliticalParty from the extended use case is available. |
| Post-conditions | An already created PoliticalParty is added to CreatePoliticalParty. |

|  |  |
| --- | --- |
| Name | Find an existing Candidates |
| ID | FindCandidates |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to find an existing Candidate; used by CreateCandidates use cases |
| Basic Path | 1. The system displays all created Candidates. Done and Cancel buttons are also visible. 2. The ElectionsOfficer selects a created Candidates and clicks Done. |
| Alternative Paths | 1. In step 1, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to find an already created Candidate within the CreateCandidates use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The CreateCandidates from the extended use case is available. |
| Post-conditions | An already created Candidates is added to CreateCandidates. |

|  |  |
| --- | --- |
| Name | View an ElectoralDistrict |
| ID | ViewElectoralDistrict |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | A use case to show all existing Electoral District. |
| Basic Path | 1. The system displays all created ElectoralDistricts. Each ElectoralDistrict has a visible modify and delete button. A Cancel button is also visible. |
| Alternative Paths | 1. In step 1, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | 1. In step 1, if the ElectionsOfficer selects modify, the system executes the ModifyElectoralDistrict use case extension; at its completion, the system displays a screen showing all of the ElectoralDistricts provided so far and goes to step 1 in the Basic Path. 2. In step 1, if the ElectionsOfficer selects delete, the system executes the DeleteElectoralDistrict use case extension; at its completion, the system displays a screen showing all of the ElectoralDistricts provided so far and goes to step 1 in the Basic Path. |
| Triggers | The ElectionsOfficer has elected to view an already created Electoral District. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The ElectoralDistrict user objects are available. |
| Post-conditions | N/A |

|  |  |
| --- | --- |
| Name | View a Candidate |
| ID | ViewCandidates |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | A use case to show all existing Candidates. |
| Basic Path | 1. The system displays all created Candidates. Each Candidate has a visible modify and delete button. A Cancel button is also visible. |
| Alternative Paths | 1. In step 1, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | 1. In step 1, if the ElectionsOfficer selects modify, the system executes the ModifyCandidate use case extension; at its completion, the system displays a screen showing all of the Candidates provided so far and goes to step 1 in the Basic Path. 2. In step 1, if the ElectionsOfficer selects delete, the system executes the DeleteCandidate use case extension; at its completion, the system displays a screen showing all of the Candidates provided so far and goes to step 1 in the Basic Path. |
| Triggers | The ElectionsOfficer has elected to view an already created Candidate. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Candidate user objects are available. |
| Post-conditions | N/A |

|  |  |
| --- | --- |
| Name | View a PoliticalParty |
| ID | ViewPoliticalParty |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | A use case to show all existing Political Parties. |
| Basic Path | 1. The system displays all created PoliticalParties. Each PoliticalParty has a visible modify and delete button. A Cancel button is also visible. |
| Alternative Paths | 1. In step 1, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | 1. In step 1, if the ElectionsOfficer selects modify, the system executes the ModifyPoliticalParty use case extension; at its completion, the system displays a screen showing all of the PoliticalParties provided so far and goes to step 1 in the Basic Path. 2. In step 1, if the ElectionsOfficer selects delete, the system executes the DeletePoliticalParty use case extension; at its completion, the system displays a screen showing all of the PoliticalParties provided so far and goes to step 1 in the Basic Path. |
| Triggers | The ElectionsOfficer has elected to view an already created Political Party. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The PoliticalParty user objects are available. |
| Post-conditions | N/A |

|  |  |
| --- | --- |
| Name | View an Election |
| ID | ViewElection |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | A use case to show all existing Elections. |
| Basic Path | 1. The system displays all created Elections. Each Election has a visible modify, delete, OpenSystem, CloseSystem, and PublishResults buttons. A Cancel button is also visible. |
| Alternative Paths | 1. In step 1, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | 1. In step 1, if the ElectionsOfficer selects modify, the system executes the ModifyElection use case extension; at its completion, the system displays a screen showing all of the Elections provided so far and goes to step 1 in the Basic Path. 2. In step 1, if the ElectionsOfficer selects delete, the system executes the DeleteElection use case extension; at its completion, the system displays a screen showing all of the Elections provided so far and goes to step 1 in the Basic Path. 3. In step 1, if the ElectionsOfficer selects OpenSystem, the system executes the OpenSystem use case extension; at its completion, the system displays a screen showing all of the Elections provided so far and goes to step 1 in the Basic Path. 4. In step 1, if the ElectionsOfficer selects CloseSystem, the system executes the CloseSystem use case extension; at its completion, the system displays a screen showing all of the Elections provided so far and goes to step 1 in the Basic Path. 5. In step 1, if the ElectionsOfficer selects PublishResults, the system executes the PublishResults use case extension; at its completion, the system displays a screen showing all of the Elections provided so far and goes to step 1 in the Basic Path. |
| Triggers | The ElectionsOfficer has elected to view an already created Election. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Election user objects are available. |
| Post-conditions | N/A |

|  |  |
| --- | --- |
| Name | View an Issue |
| ID | ViewIssue |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | A use case to show all existing Issues. |
| Basic Path | 1. The system displays all created Issues. Each Issue has a visible modify, delete, OpenSystem, CloseSystem, and PublishResults buttons. A Cancel button is also visible. |
| Alternative Paths | 1. In step 1, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | 1. In step 1, if the IssuesOfficer selects modify, the system executes the ModifyIssue use case extension; at its completion, the system displays a screen showing all of the Issues provided so far and goes to step 1 in the Basic Path. 2. In step 1, if the ElectionsOfficer selects delete, the system executes the DeleteIssue use case extension; at its completion, the system displays a screen showing all of the Issues provided so far and goes to step 1 in the Basic Path. 3. In step 1, if the ElectionsOfficer selects OpenSystem, the system executes the OpenSystem use case extension; at its completion, the system displays a screen showing all of the Issues provided so far and goes to step 1 in the Basic Path. 4. In step 1, if the ElectionsOfficer selects CloseSystem, the system executes the CloseSystem use case extension; at its completion, the system displays a screen showing all of the Issues provided so far and goes to step 1 in the Basic Path. 5. In step 1, if the ElectionsOfficer selects PublishResults, the system executes the PublishResults use case extension; at its completion, the system displays a screen showing all of the Issues provided so far and goes to step 1 in the Basic Path. |
| Triggers | The ElectionsOfficer has elected to view an already created Issue. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Issue user objects are available. |
| Post-conditions | N/A |

|  |  |
| --- | --- |
| Name | Modify an existing Electoral District |
| ID | ModifyElectoralDistrict |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to modify an existing Electoral District; used by ViewElectoralDistrict use case. |
| Basic Path | 1. The system displays the prompt to enter: name, area; Done and Cancel buttons are visible. 2. The ElectionsOfficer provides the required information and clicks Done. 3. The system modifies an existing ElectoralDistrict with the provided data, if one was selected during the FindElectoralDistrict use case execution. The system displays a new screen in which all the created ElectoralDistricts are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In step 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | 1. In step 3, if the system determines that an ElectoralDistrict with the user inputted name already exists, the system displays the error message DuplicateElectoralDistrictName and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points |  |
| Triggers | The ElectionsOfficer has elected to modify an existing Electoral District within the ViewElectoralDistrict use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The ElectoralDistricts from the extended use case is available. |
| Post-conditions | A modified ElectoralDistrict user objects is now available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | Modify an existing Political Party |
| ID | ModifyPoliticalParty |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to modify an existing Political Party; used by ViewPoliticalParty use case. |
| Basic Path | 1. The system displays the prompt to enter: name; Done and Cancel buttons are visible. 2. The ElectionsOfficer provides the required information and clicks Done. 3. The system modifies an existing PoliticalParty with the provided data, if one was selected during the FindPoliticalParty use case execution. The system displays a new screen in which all the created PoliticalParties are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In step 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | 1. In step 3, if the system determines that a PoliticalParty with the user inputted name already exists, the system displays the error message DuplicatePoliticalPartyName and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points |  |
| Triggers | The ElectionsOfficer has elected to modify an existing Political Party within the ViewPoliticalParty use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The PoliticalParties from the extended use case is available. |
| Post-conditions | A modified PoliticalParty user objects is now available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | Modify an existing Candidate |
| ID | ModifyCandidate |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to modify an existing Candidate; used by ViewCandidate use case. |
| Basic Path | 1. The system displays the prompt to enter: name, political party; Done and Cancel buttons are visible. 2. The ElectionsOfficer provides the required information and clicks Done. 3. The system modifies an existing Candidate with the provided data, if one was selected during the FindCandidate use case execution. The system displays a new screen in which all the created Candidates are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In step 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | 1. In step 3, if the system determines that a Candidate with the user inputted name already exists, the system displays the error message DuplicateCandidateName and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points |  |
| Triggers | The ElectionsOfficer has elected to modify an existing Candidate within the ViewCandidate use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Candidates from the extended use case is available. |
| Post-conditions | A modified Candidate user objects is now available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | Modify an existing Election |
| ID | ModifyElection |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to modify an existing Election; used by ViewElection use case. |
| Basic Path | 1. The system displays the prompt to enter: name, candidates and electoral districts; Done and Cancel buttons are visible. 2. The ElectionsOfficer provides the required information and clicks Done. 3. The system modifies an existing Election with the provided data, if one was selected during the FindElection use case execution. The system displays a new screen in which all the created Elections are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In step 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | 1. In step 3, if the system determines that an Election with the user inputted name already exists, the system displays the error message DuplicateElectionName and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points |  |
| Triggers | The ElectionsOfficer has elected to modify an existing Election within the ViewElection use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Elections from the extended use case is available. |
| Post-conditions | A modified Election user objects is now available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | Modify an existing Issue |
| ID | ModifyIssue |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to modify an existing Issue; used by ViewIssue use case. |
| Basic Path | 1. The system displays the prompt to enter: name, ElectoralDistricts; Done and Cancel buttons are visible. 2. The ElectionsOfficer provides the required information and clicks Done. 3. The system modifies an existing Issue with the provided data, if one was selected during the FindIssue use case execution. The system displays a new screen in which all the created Issues are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In step 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | 1. In step 3, if the system determines that an Issue with the user inputted name already exists, the system displays the error message DuplicateIssueName and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points |  |
| Triggers | The ElectionsOfficer has elected to modify an existing Issue within the ViewIssue use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Issues from the extended use case is available. |
| Post-conditions | A modified Issue user objects is now available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | Modify an the Elections Officer’s Profile |
| ID | ModifyProfile |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | This use case details modifying the ElectionsOfficer’s profile. |
| Basic Path | 1. The system displays the prompt to enter: name, email, password; Done and Cancel buttons are visible. 2. The ElectionsOfficer provides the required information and clicks Done. 3. The system modifies an existing ElectionsOfficer user object with the provided data. The system displays a new screen in which all the user info is shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In step 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points |  |
| Triggers | The ElectionsOfficer has elected to modify his profile. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. |
| Post-conditions | A modified ElectionsOfficer user objects is now available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | Delete an existing Electoral District |
| ID | DeleteElectoralDistrict |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to delete an existing Electoral District; used by ViewElectoralDistrict use case. |
| Basic Path | 1. The system displays the prompt with the buttons Done and Cancel. 2. The ElectionsOfficer clicks Done. 3. The system deletes the selected ElectoralDistrict, The system displays a new screen in which all the created ElectoralDistricts are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In step 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to delete an existing Electoral District within the ViewElectoralDistrict use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The ElectoralDistricts from the extended use case is available. |
| Post-conditions | The deleted ElectoralDistrict user objects is no longer available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | Delete an existing Political Party |
| ID | DeletePoliticalParty |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to delete an existing Political Party; used by ViewPoliticalParty use case. |
| Basic Path | 1. The system displays the prompt with the buttons Done and Cancel. 2. The ElectionsOfficer clicks Done. 3. The system deletes the selected PoliticalParty, The system displays a new screen in which all the created PoliticalParties are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In step 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to delete an existing Political Party within the ViewPoliticalParty use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The PoliticalParties from the extended use case is available. |
| Post-conditions | The deleted PoliticalParty user objects is no longer available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | Delete an existing Candidate |
| ID | DeleteCandidate |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to delete an existing Candidate; used by ViewCandidate use case. |
| Basic Path | 1. The system displays the prompt with the buttons Done and Cancel. 2. The ElectionsOfficer clicks Done. 3. The system deletes the selected Candidate, The system displays a new screen in which all the created Candidates are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In step 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to delete an existing Candidate within the ViewCandidate use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Candidates from the extended use case is available. |
| Post-conditions | The deleted Candidate user objects is no longer available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | Delete an existing Election |
| ID | DeleteElection |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to delete an existing Election; used by ViewElection use case. |
| Basic Path | 1. The system displays the prompt with the buttons Done and Cancel. 2. The ElectionsOfficer clicks Done. 3. The system deletes the selected Election, The system displays a new screen in which all the created Elections are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In step 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to delete an existing Election within the ViewElection use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Elections from the extended use case is available. |
| Post-conditions | The deleted Election user objects is no longer available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | Delete an existing Issue |
| ID | DeleteIssue |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/6/2016 |
| Summary | An extension use case to delete an existing Issue; used by ViewIssue use case. |
| Basic Path | 1. The system displays the prompt with the buttons Done and Cancel. 2. The ElectionsOfficer clicks Done. 3. The system deletes the selected Issue, The system displays a new screen in which all the created Issues are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In step 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | N/A |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to delete an existing Issue within the ViewIssue use case. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Issues from the extended use case is available. |
| Post-conditions | The deleted Issue user objects is no longer available to other use cases available to the same ElectionsOfficer. |

|  |  |
| --- | --- |
| Name | Open System For Voting |
| ID | OpenSystem |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/7/2016 |
| Summary | An extension use case to open an existing Issue or Election; used by ViewElections and ViewIssues use cases. |
| Basic Path | 1. The system displays the prompt with the buttons Done and Cancel. 2. The ElectionsOfficer clicks Done. 3. The system opens the selected Election or Issue for voting by voters, the system displays a new screen in which all the created Elections or Issues are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In steps 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | 1. In step 1, if the system determines that an Issue or Election is already open for voting, the system displays the error message AlreadyOpen and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to open an existing Election or Issue within the ViewElections or ViewIssues use cases. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Issues or Elections from the extended use cases are available. |
| Post-conditions | The selected Issue or Election is now open and can now be voted on by voters in the assigned electoral district. |

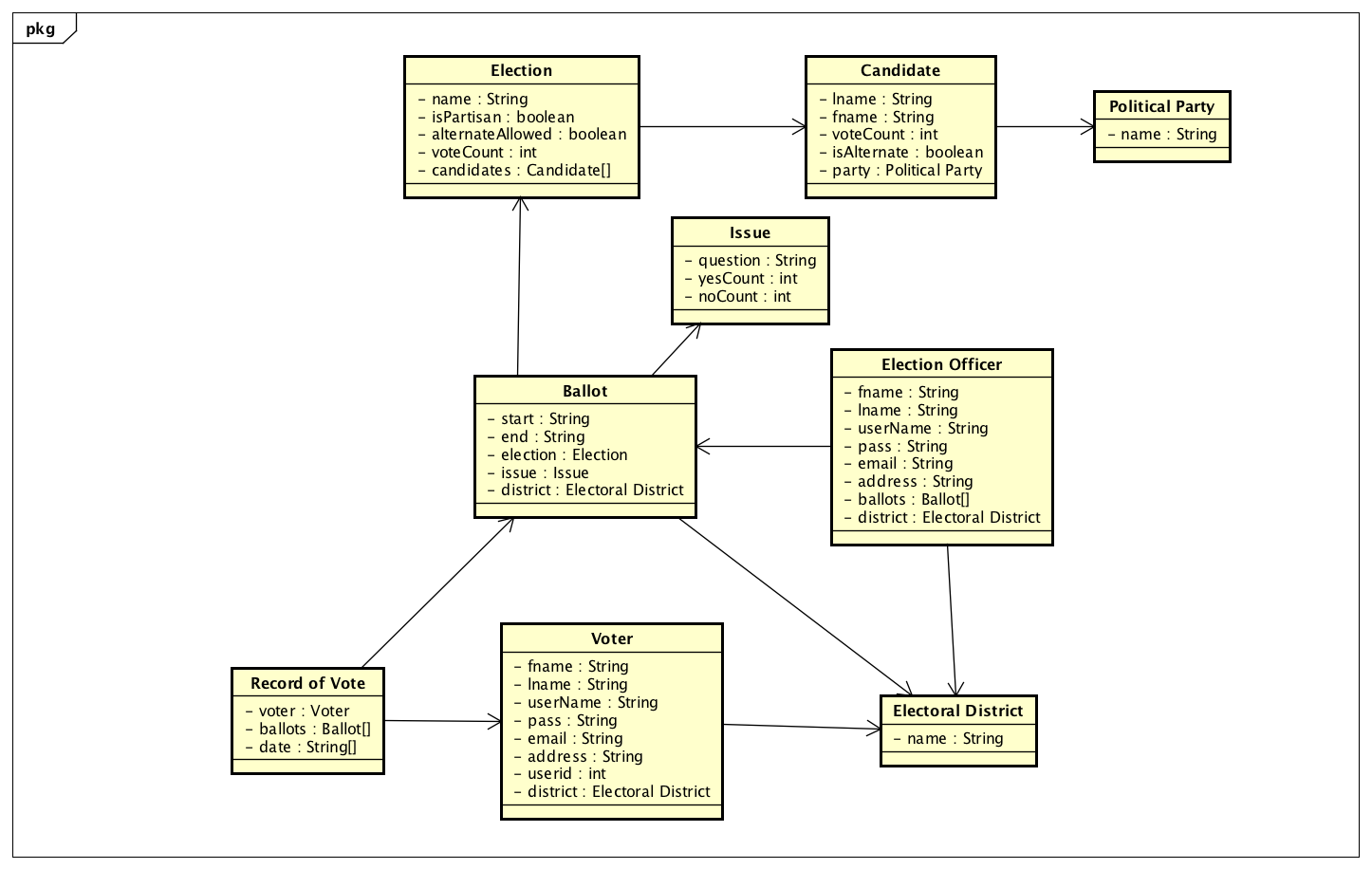
|  |  |
| --- | --- |
| Name | Close System For Voting |
| ID | CloseSystem |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/7/2016 |
| Summary | An extension use case to open an existing Issue or Election; used by ViewElections and ViewIssues use cases. |
| Basic Path | 1. The prompt displays the buttons Done and Cancel. 2. The ElectionsOfficer clicks Done. 3. The system closes the selected Election or Issue for voting by voters, the system displays a new screen in which all the created Elections or Issues are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In steps 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | 1. In step 1, if the system determines that an Issue or Election is already closed for voting, the system displays the error message AlreadyClosed and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to close an existing Election or Issue within the ViewElections or ViewIssues use cases. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Issues or Elections from the extended use cases are available. |
| Post-conditions | The selected Issue or Election is now closed and can now not be voted on by voters in the assigned electoral district. |

|  |  |
| --- | --- |
| Name | Publish Results |
| ID | PublishResults |
| Version | 1 |
| Author | Donovan Todd |
| Date | 9/7/2016 |
| Summary | An extension use case to publish the results of an Issue or Election; used by ViewElections and ViewIssues use cases. |
| Basic Path | 1. The system displays the results of the Issue or Election so that it can be validated by the ElectionsOfficer. The prompt also displays the buttons Done and Cancel. 2. The ElectionsOfficer clicks Done. 3. The system publishes the results of the selected Election or Issue by sending the results to all voters that voted in the selected Election or Issue, the system displays a new screen in which all the created Elections or Issues are shown. The screen displays 2 buttons: Done and Cancel as well. |
| Alternative Paths | 1. In steps 1 and 3, if the user presses the Cancel button, the use case is terminated and no changes to the system are made. |
| Exception Paths | 1. In step 1, if the system determines that an Issue or Election isnt closed for voting, the system displays the error message StillOpen and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. 2. In step 1, if the system determines that an Issue or Election was never opened for voting and the system displays the error message NoVotesAvailable and the system goes back to step 1 in the Basic Path, retaining all previously provided attribute values. |
| Extension Points | N/A |
| Triggers | The ElectionsOfficer has elected to publish the results of a closed Election or Issue within the ViewElections or ViewIssues use cases. |
| Assumption | N/A |
| Pre-conditions | The ElectionsOfficer is currently logged in and the session has not expired. The ElectionsOfficer’s user object is available. The Issues or Elections from the extended use cases are available. |
| Post-conditions | The selected Issue or Election’s results are now published to all voters that voted on it and can now not be viewed by voters in the assigned electoral district. |

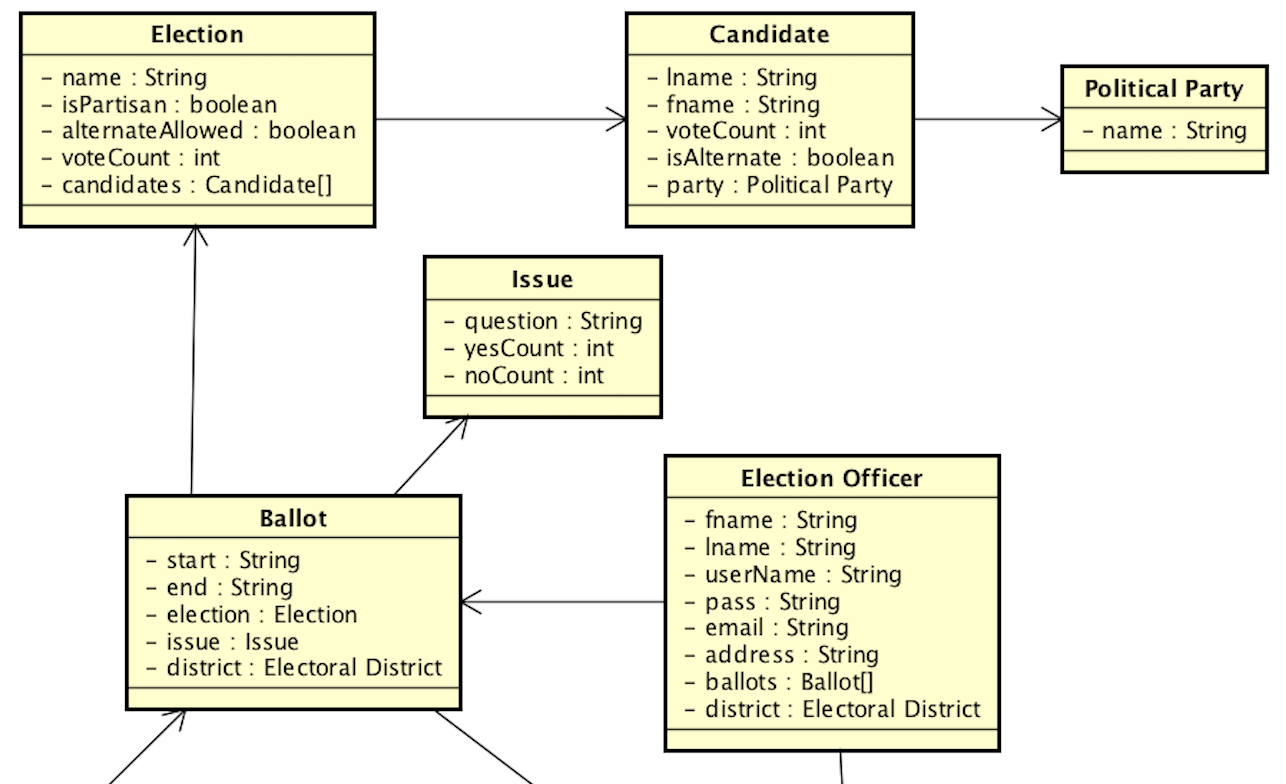
|  |  |
| --- | --- |
| Use case name: | voterRegistration |
| ID | voterRegistration |
| Version | 1 |
| Author | Rosalie Thobe |
| Date | 9/6/16 |
| Summary | This use case explains the process of signing up to become a registered voter so that he/she may vote in the future. |
| Basic Path | 1. Voter goes to eVoting system website where there is a side for voter to sign in with username and password or to create a new account. 2. Voter does not have username or password yet so they fill out the prompted sections under the “create a new account” column. 3. The system asks for: first name, last name, age, address, email, phone number, unique username and password. 4. Voter clicks on the “create account” button. 5. Website goes to the voter’s newly created profile where he/she can: view and manage their profile, view any elections/issues that are going to be or are occurring in their district, and the voter can also select an election/issue to vote on. |
| Alternative Paths | None |
| Exception Paths | If in step 3: the voter enters a username that is already taken by another voter, a message will appear. The message will say “username already taken, please select another”. |
| Extension Points |  |
| Triggers | The citizen wants become a voter in order to take part in voting on the next election/issue. |
| Assumption |  |
| Pre-conditions |  |
| Post-conditions | The voter becomes apart of the system and has the ability to now vote on open elections/issues. |

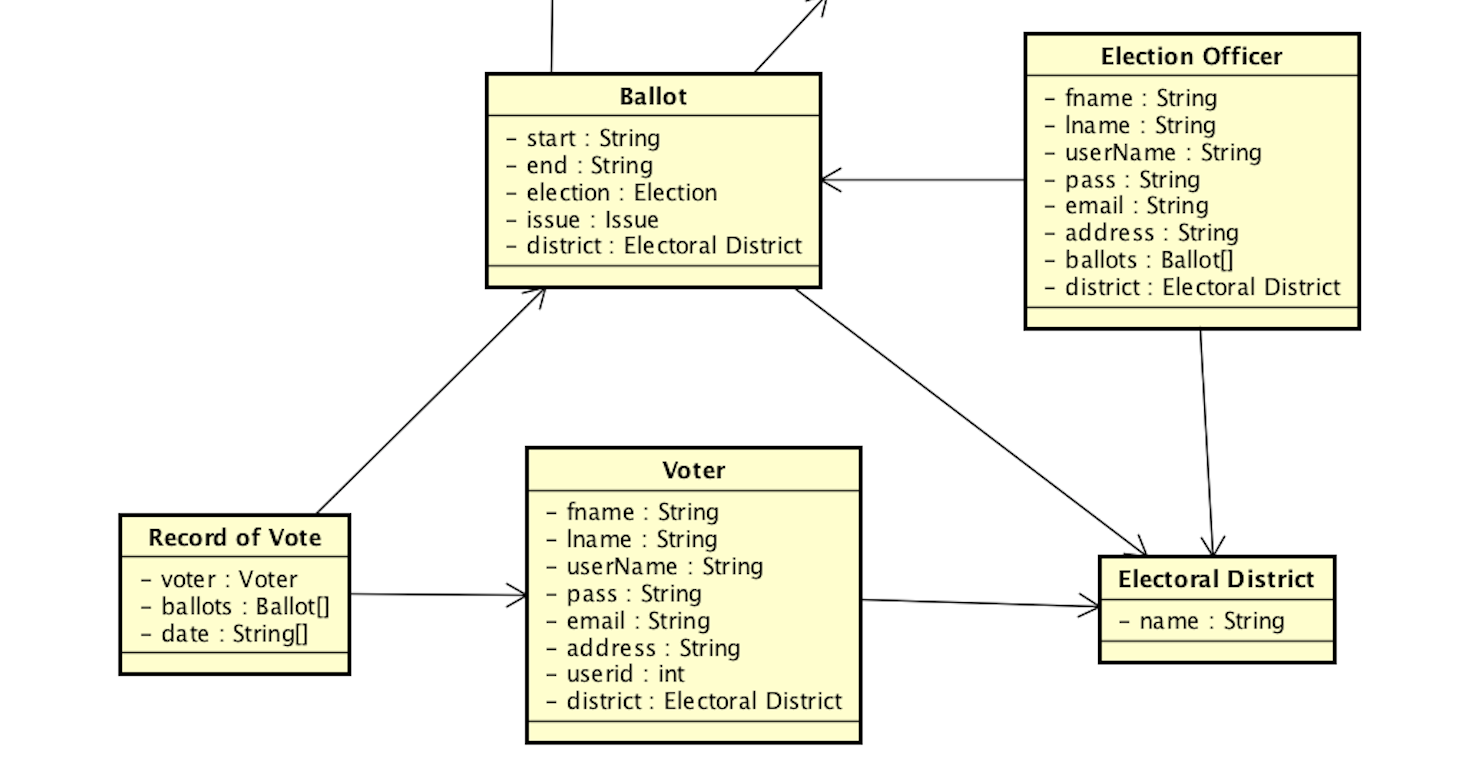
|  |  |
| --- | --- |
| Use case name: | voting |
| ID | voting |
| Version | 1 |
| Author | Rosalie Thobe |
| Date | 9/6/16 |
| Summary | This use case explains the process of voting on an issue or election that is about to open for voting or currently open for voting. |
| Basic Path | 1. The Voter logs into his/her account by entering username and password and pressing the “log in” button. 2. The system will bring up the home page of their account. 3. The voter has the choice of viewing the election/ballot that they want to participate in by clicking “view” on the particular election/issue. 4. The next screen pulls up the election/ballot along with the choices of “select” or “back”. 5. The voter choses the “Vote” button to vote on that issue/election. The system pulls up the selected election/issue along with the candidates or questions associated with that election/issue. 6. Voter has choice of selecting their choice of vote. There are buttons that say “vote” and “cancel”. Voter selects “vote”. 7. The system will display a “thank you for voting” page. And there will be a button “home page”. The voter clicks on that button to return to home page and may vote on another issue/election. 8. The voter is able to view the results of the election/issue they voted on. Once the election/issue closes for voting, there will appear a button “view results” next to the “view” button. 9. The voter clicks on “view results” to view the results of election/issue they voted on. 10. Once the voter is done voting, the voter clicks on the “log out” button. |
| Alternative Paths | If in step 4, the voter decides not to chose that election to vote in, the voter choses “back” button. This will bring them back to the home screen of his/her profile.  If in step 6, the voter decides not to vote on the election/issue, the vote choses the “cancel” button. This is will bring them back to the home screen of his/her profile. |
| Exception Paths | If in step 1, the voter enters his/her username or password wrong, a message will appear at the top of the page in red. The message will read: “Username/password is incorrect”.  If in step 1, the voter forgets his/her password, they can click on the “forgot password” text. The screen will then prompt the voter for their email and username. The system will then send them an email with a random temporary password for them to log in with.  If in step 1, the voter forgets his/her username, they can click on the “forgot username”. The screen will prompt the voter to enter their email. The system will then send an email to them containing their username.  If in step 4, the voter does not want to vote on the issue they can select the “back” button to go back to the list of possible issues/elections to vote on.  If in step 4, the election/issue has not been opened for voting, an error message will appear at the top. The message will say: “election/issue has not been opened yet for voting”.  If in step 7, the voter selects the same election/issue to vote on, an error message will appear. The message will say “Sorry, you already voted”.  If in step 7, the voter does select another election/issue to vote on, the same system rules 3-7 apply.  If in step 10, the voter can “delete account” by pressing on that button. The voter is no longer a registered voter for that district. |
| Extension Points |  |
| Triggers | There is an election or issue that the voter wants to participate in. |
| Assumption |  |
| Pre-conditions | The voter already is already registered to vote and their account is recognized by the system. |
| Post-conditions | The voter has voted on an issue or election and the vote has been accounted for. |

2.4.3 Domain Object Model



A closer view:

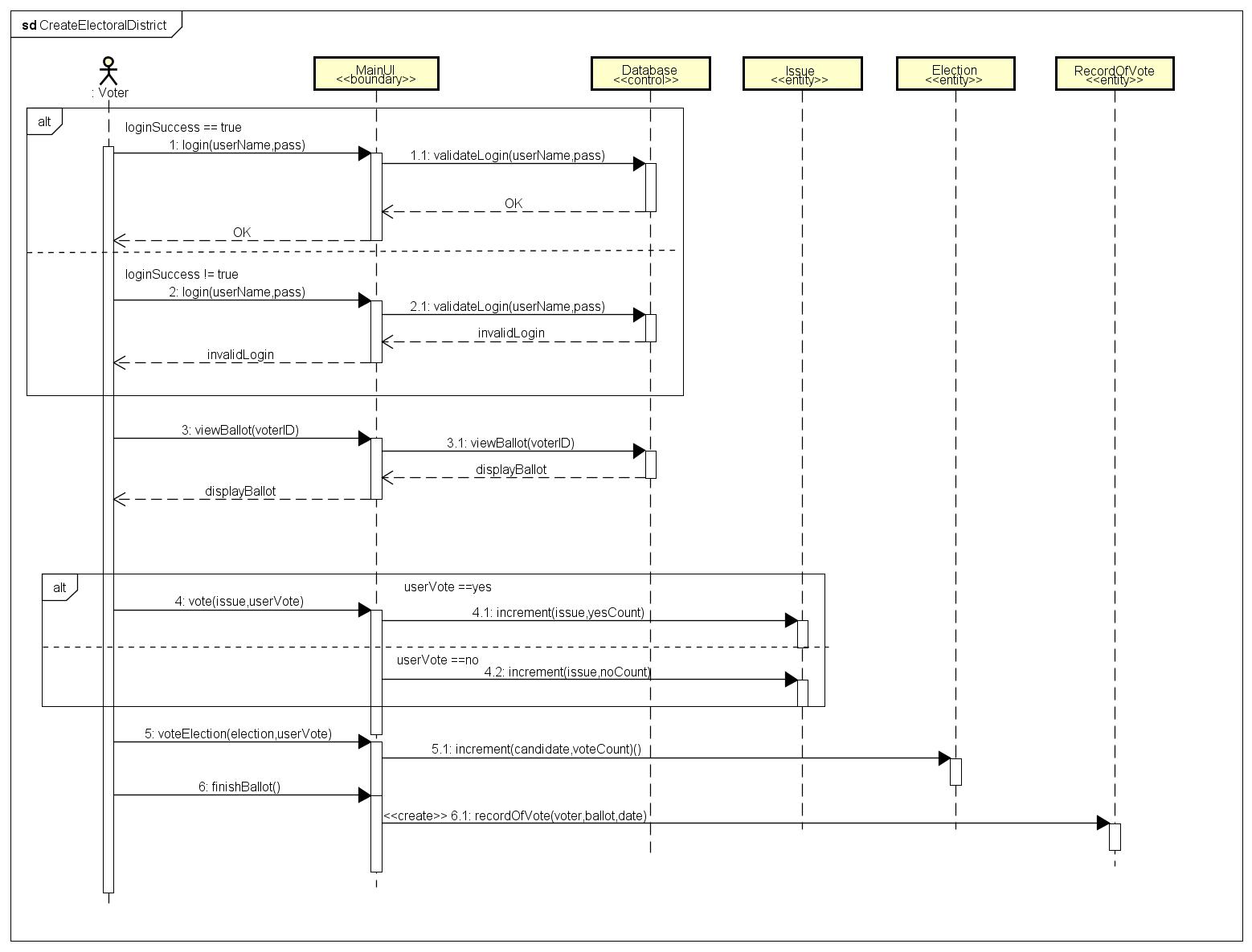




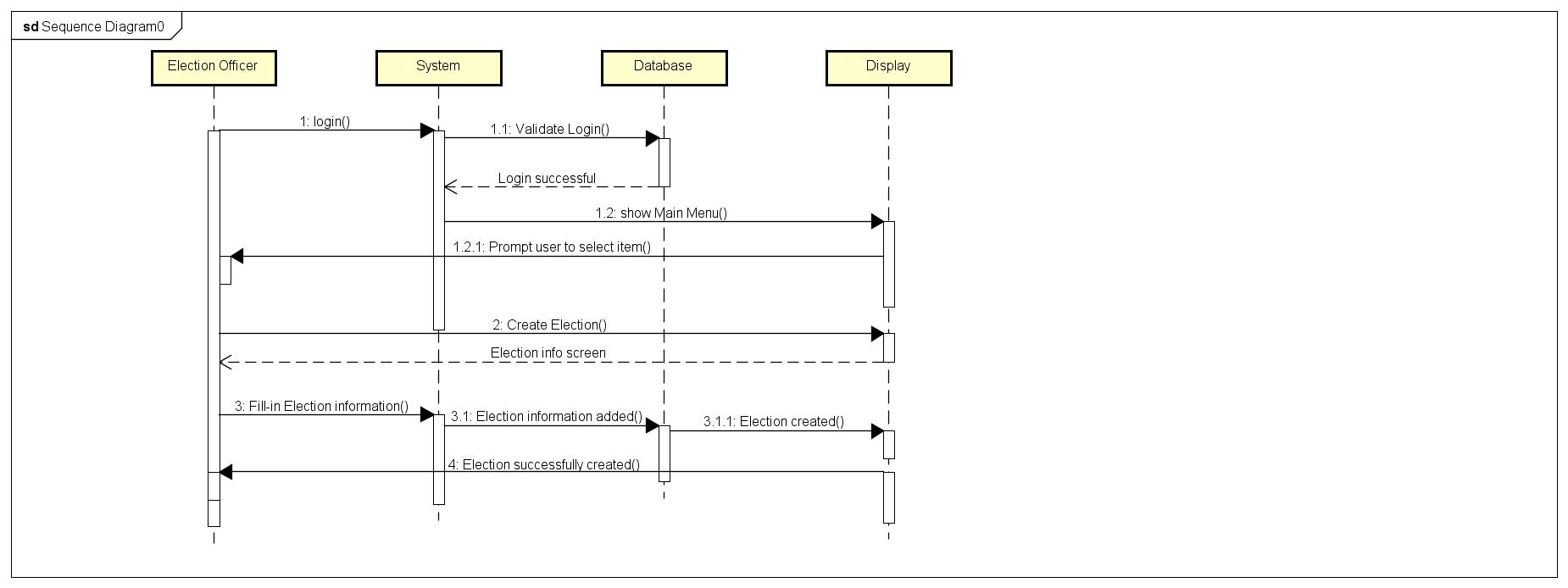
Object Model Details

1. Political Party
   1. Attributes
      1. name: String
2. Candidate
   1. Attributes
      1. lname: String
      2. fname: String
      3. voteCount: int
      4. isAlternate: boolean
   2. Static
      1. party: Political Party
         * has a political party associated with the candidate (can be NULL)
3. Election
   1. Attributes
      1. name: String
      2. isPartisan: boolean
      3. alternateAllowed: boolean
      4. voteCount: int
   2. Static
      1. candidates: Candidate[]
         * has a list of candidates involved in this election
4. Issue
   1. Attributes
      1. question: String
      2. yesCount: int
      3. noCount: int
5. Electoral District
   1. name: String
6. Ballot
   1. Attributes
      1. start: String
      2. end: String
   2. Static
      1. election: Election
         * A ballot for an election (can be NULL)
      2. issue: Issue
         * A ballot for an issue (can be NULL)
      3. district: Electoral District
         * The respective electoral district for this ballot
7. Record of Vote
   1. Attributes
      1. date: String[]
         * keeps a list of date and time of which the voter vote for a particular ballot
   2. Static
      1. voter: Voter
         * Corresponds to a particular voter
      2. ballots: Ballot[]
         * Keeps a list of ballots the voter votes on
8. Voter
   1. Attributes
      1. fname: String
      2. lname: String
      3. userName: String
      4. pass: String
      5. email: String
      6. address: String
      7. userid: int
   2. Static
      1. district: Electoral District
         * the district in which the voter belongs
9. Election Officer
   1. Attribute
      1. fname: String
      2. lname: String
      3. userName: String
      4. pass: String
      5. email: String
      6. address: String
   2. Static
      1. ballots: Ballot[]
         * the ballots under the election officer control
      2. district: Electoral District
         * the district in which the election officier control

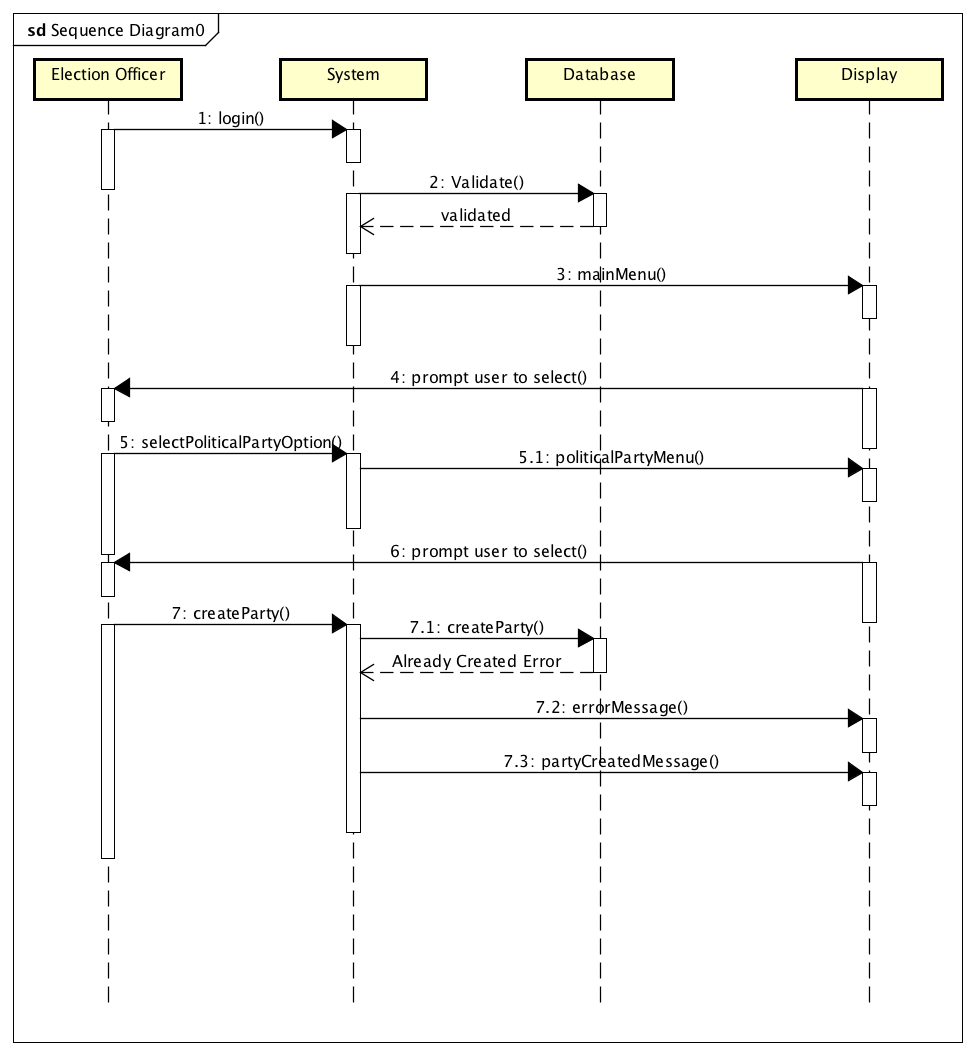
2.4.4 Sequence Diagram



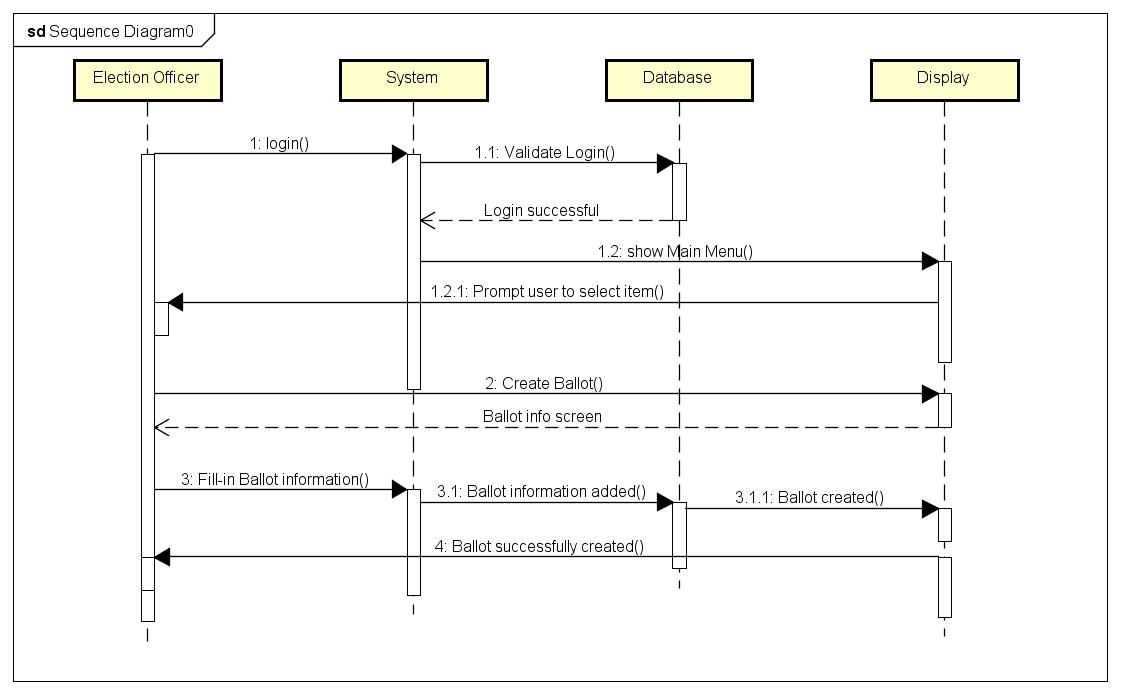
Create Electoral District (zoom in, in order to see clearly)



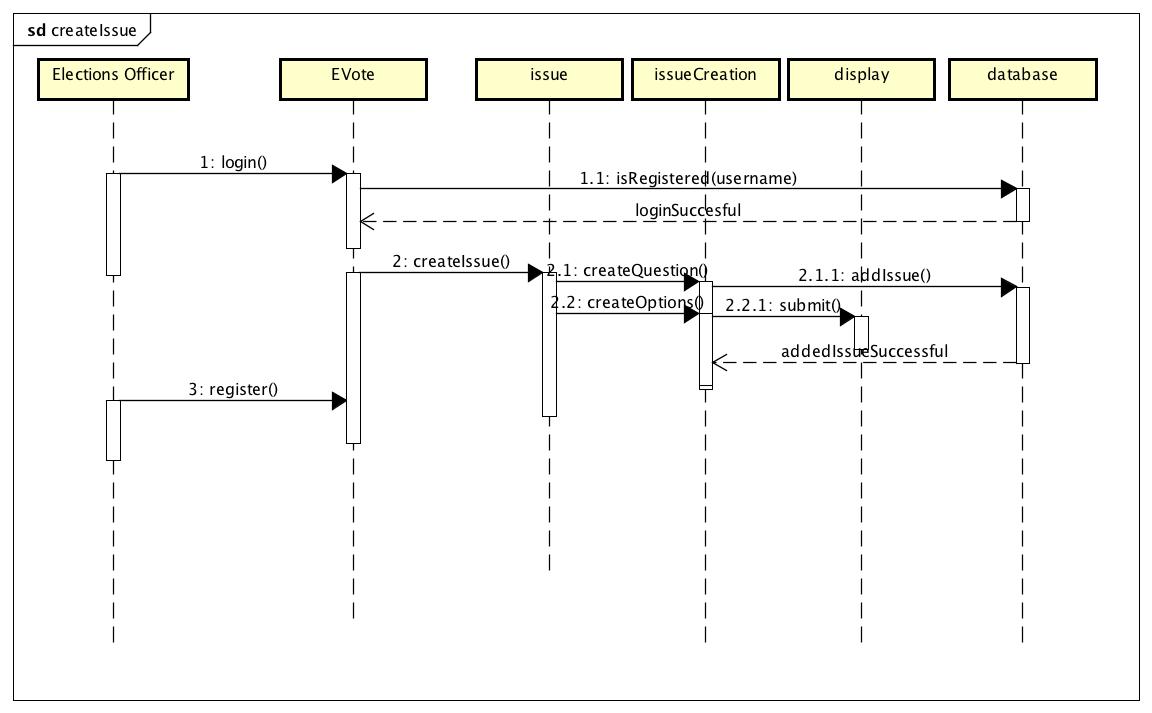
Create Election by EO



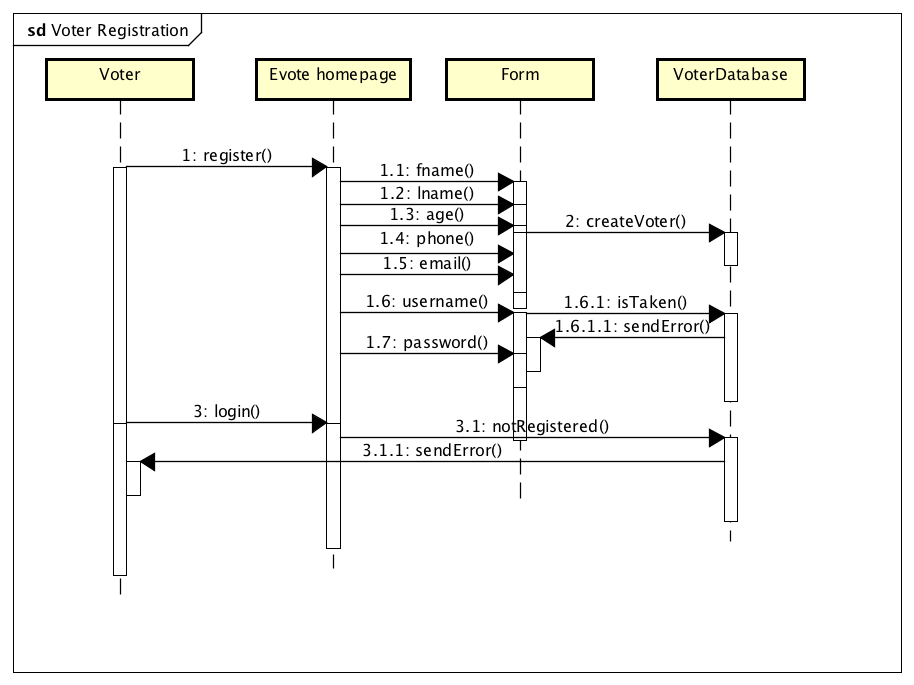
Create Political party



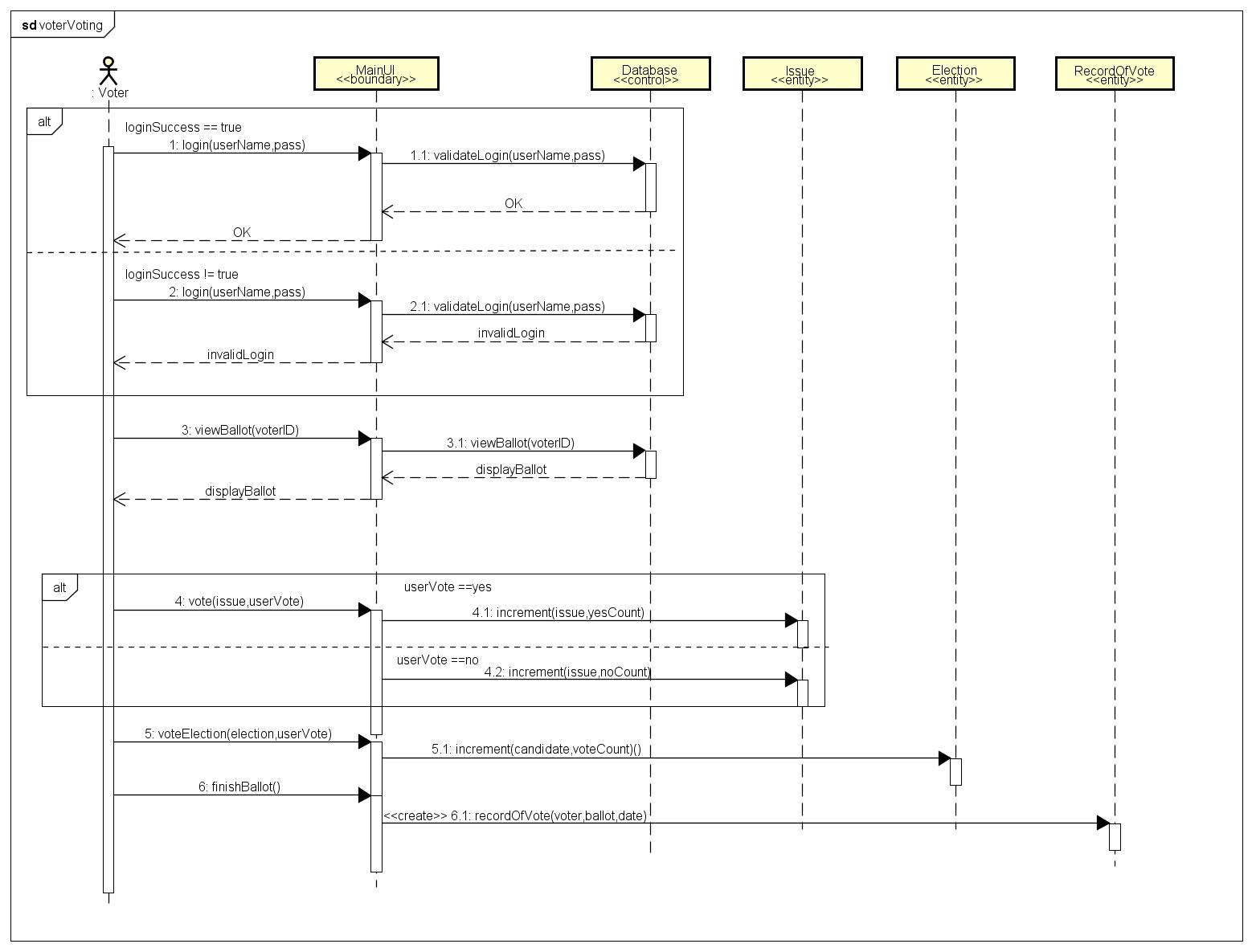
Create Ballot



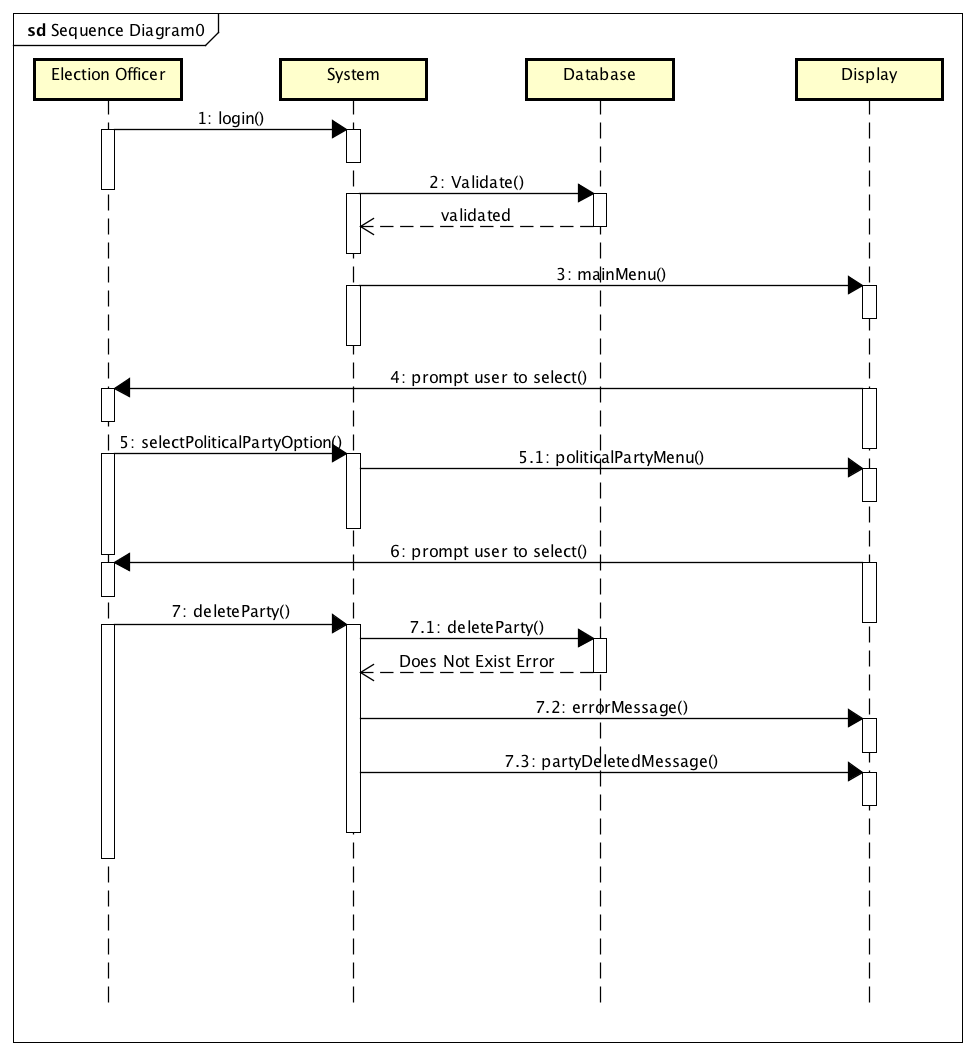
Create Issue



Voter Registration



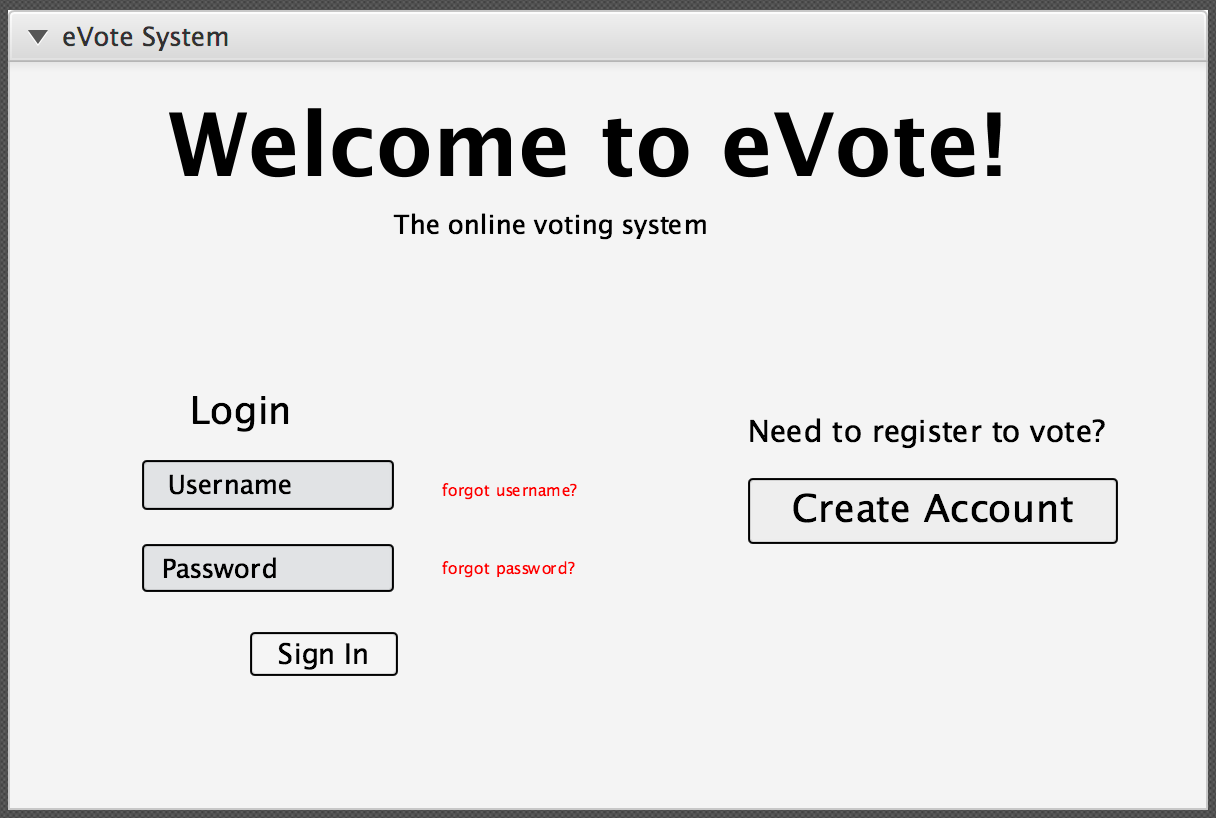
Voter Voting



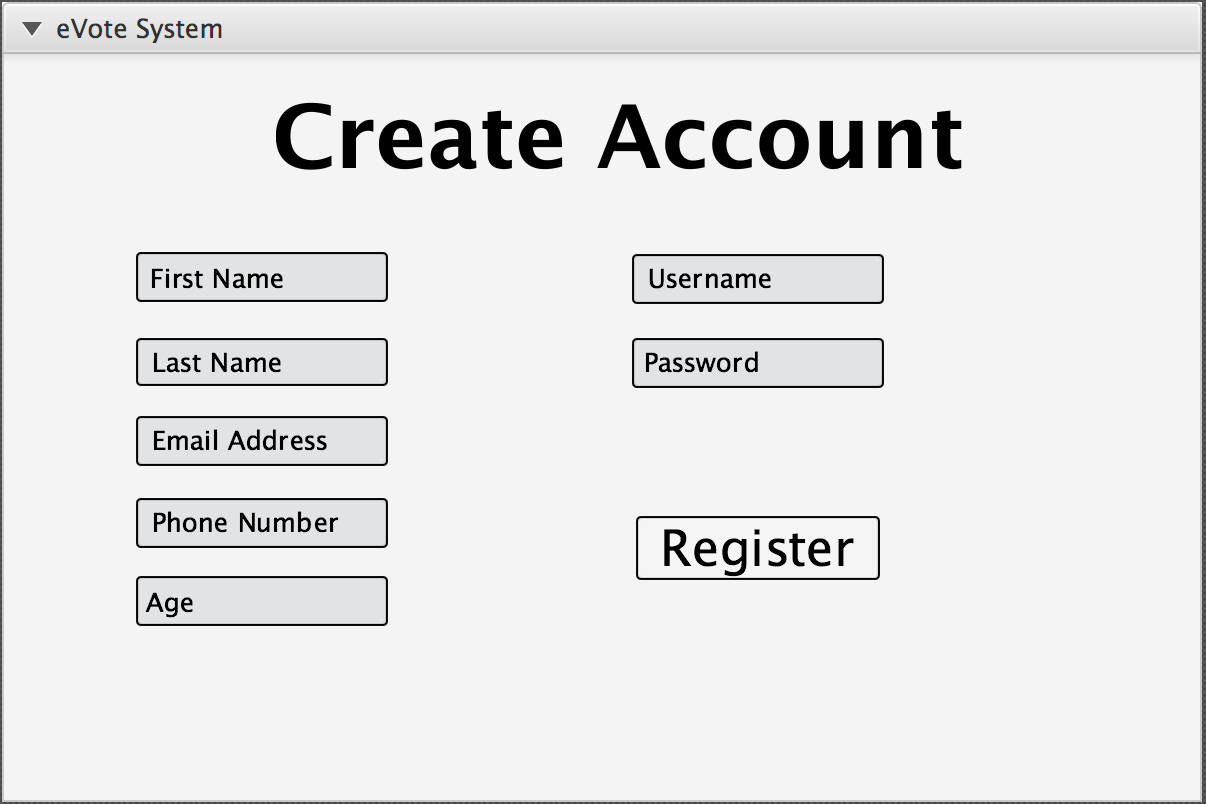
Deleting political party

2.4.5 User Interface

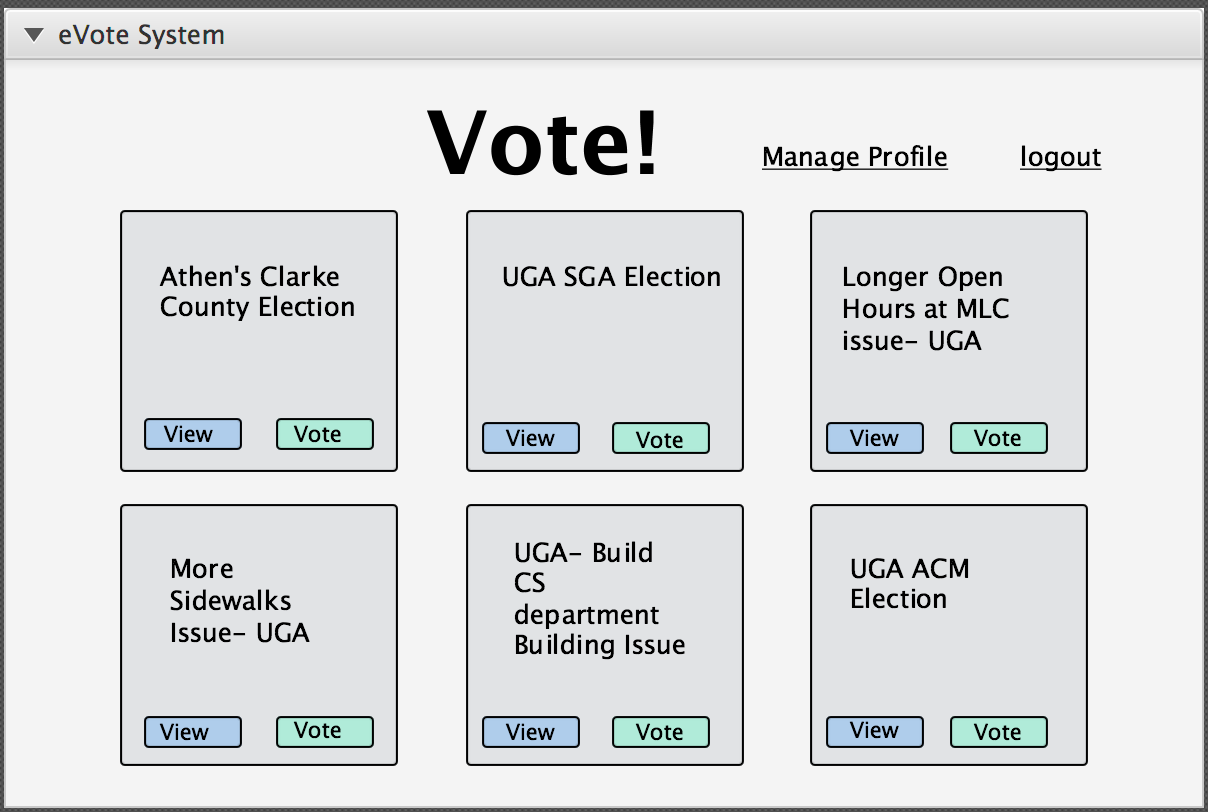
loginFrame



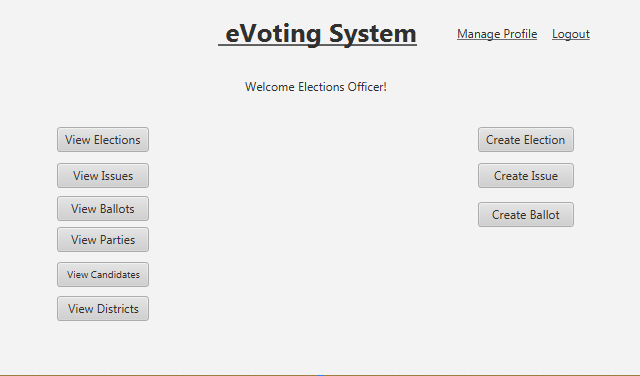
signUpFrame



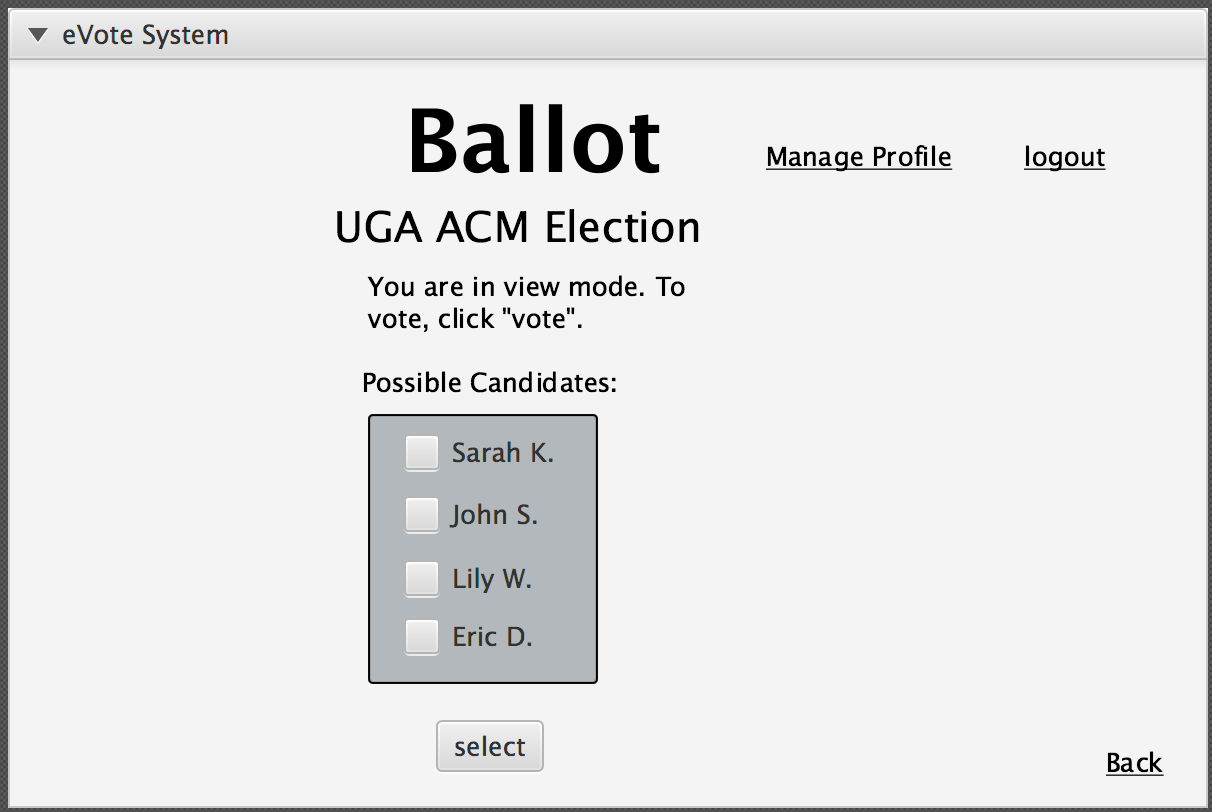
voterFrame



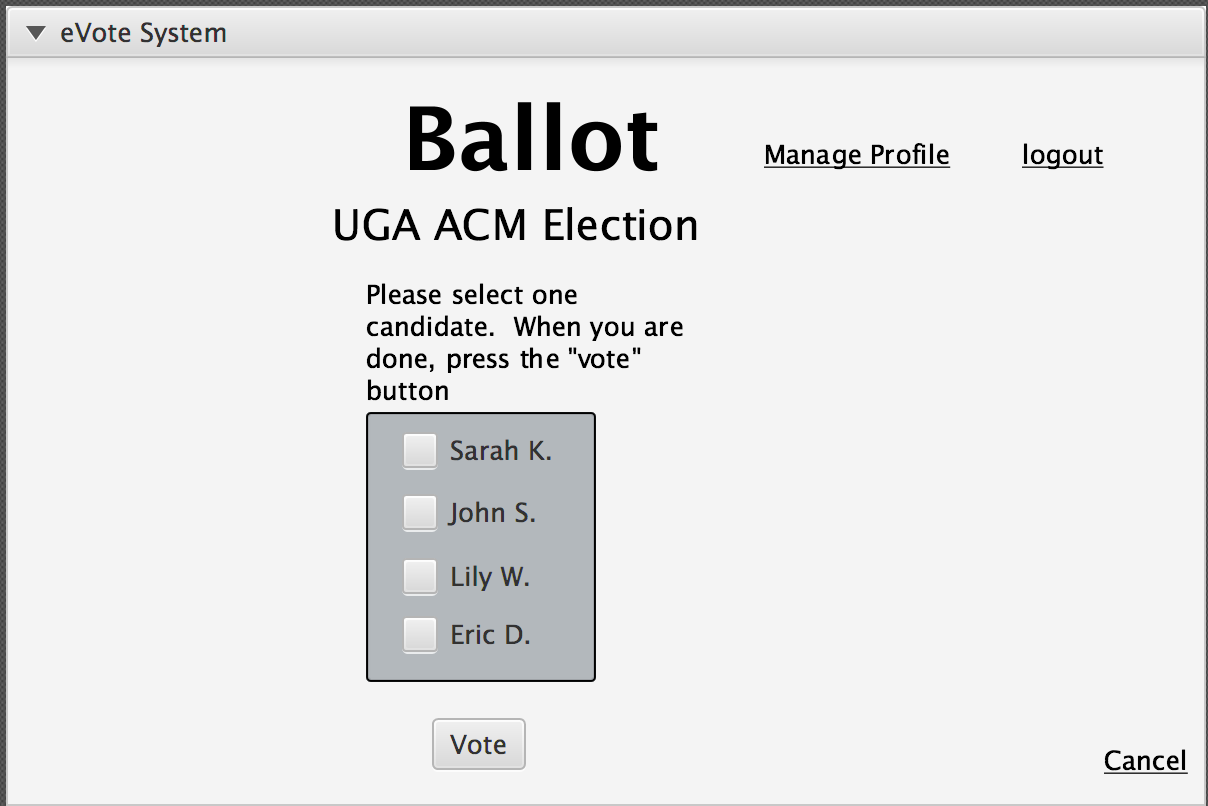
officerFrame



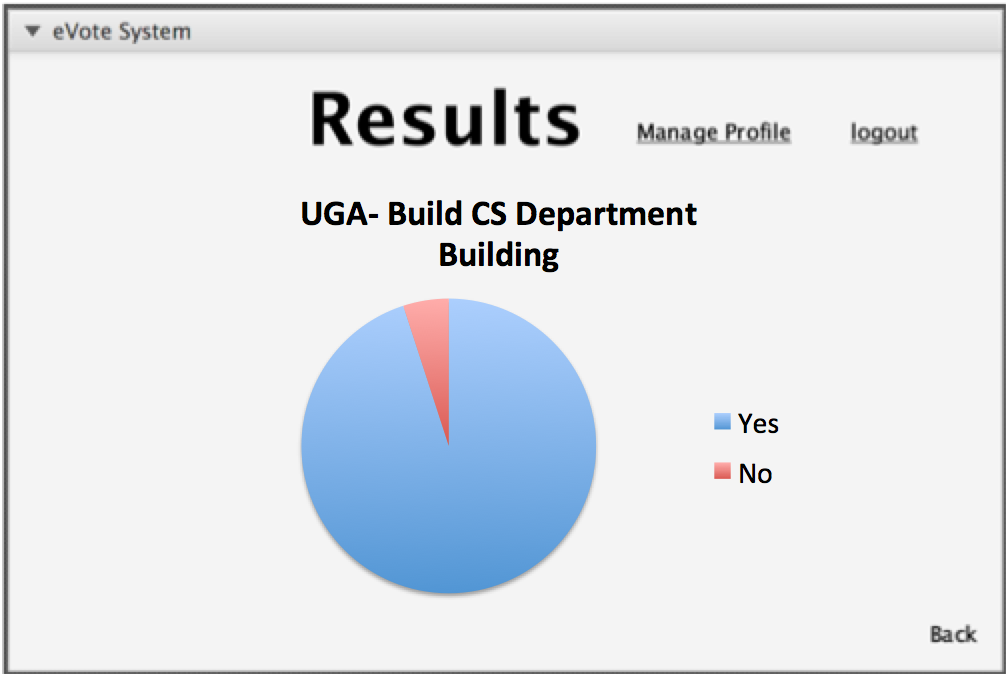
viewBallot



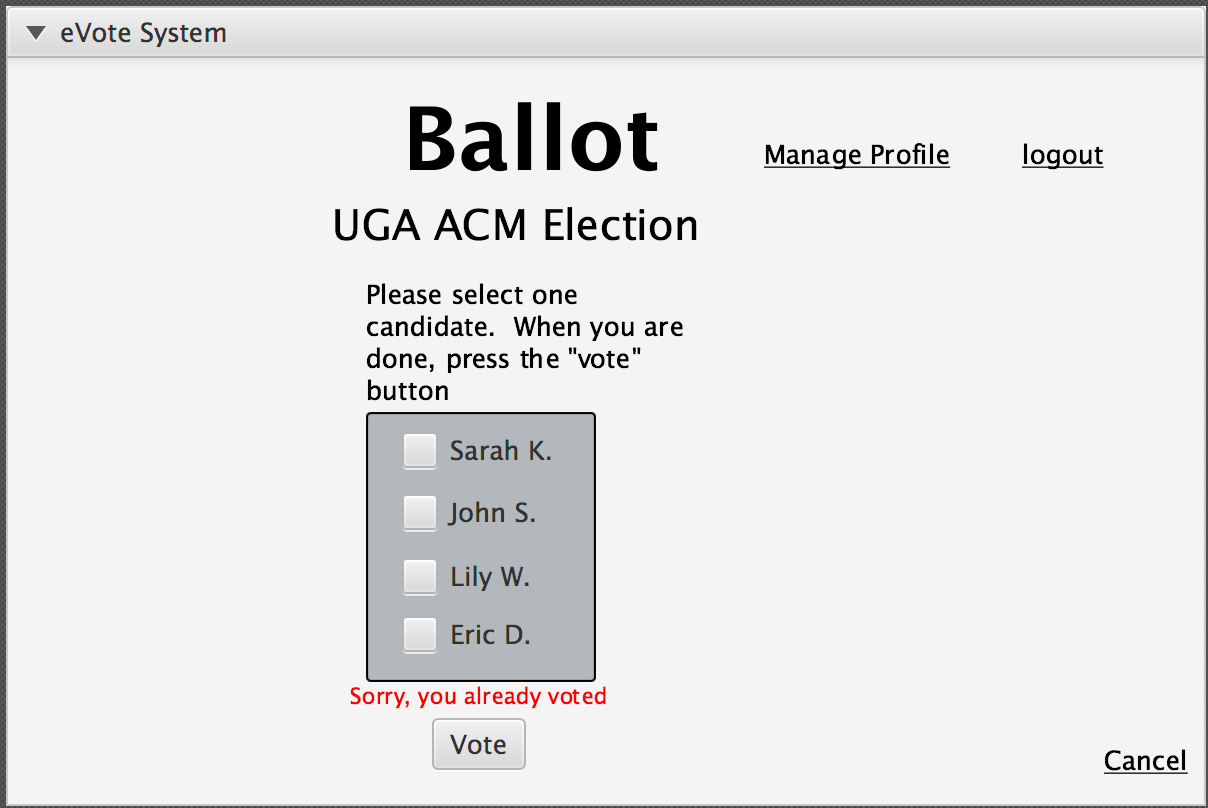
selectedBallot

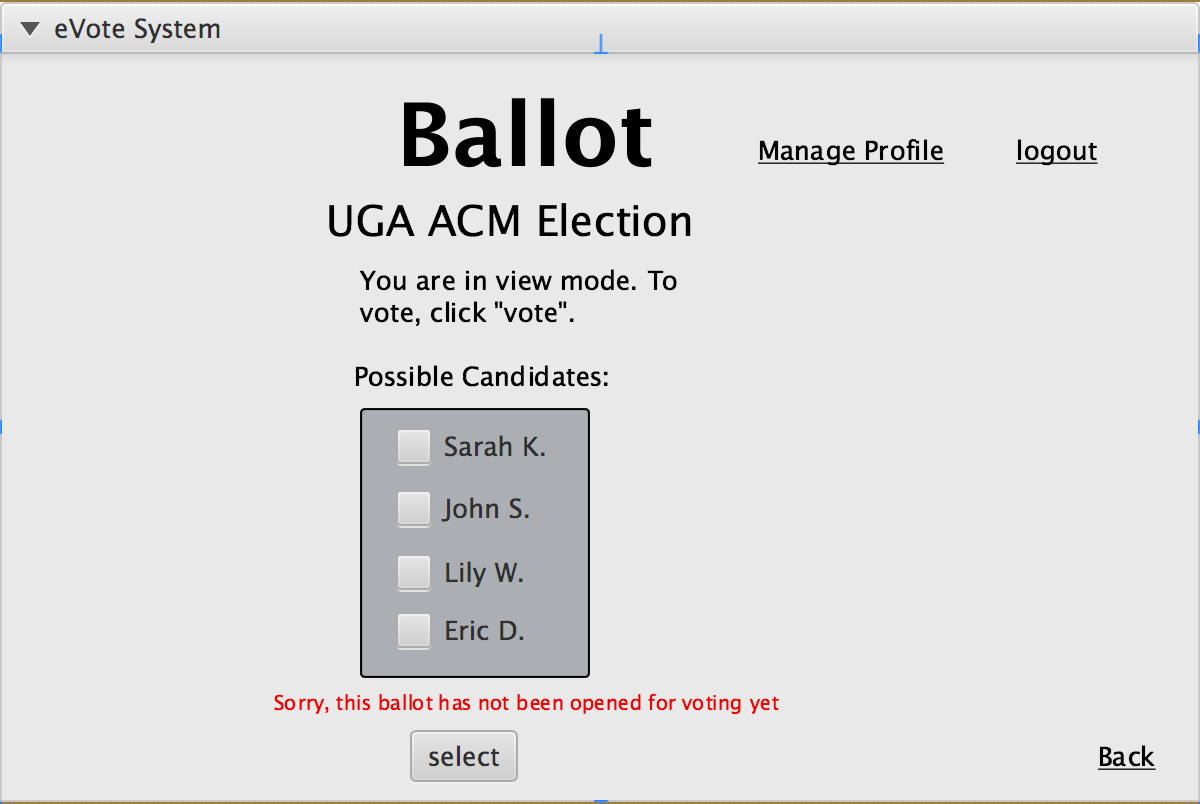


results



alreadyVoted



ballotNotOpen  


incorrect