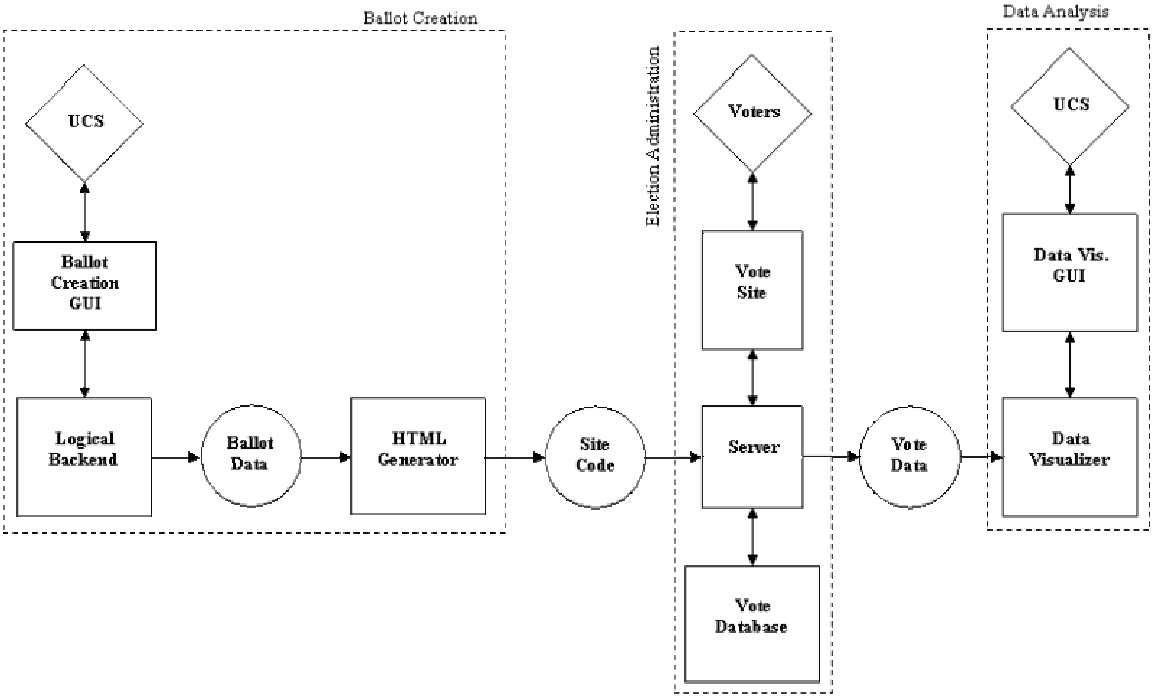
The Election Connection: Specifications

1.0 Description

The goal of this project is to design a system which allows the UCS Election Committee to design a ballot through the use of a straight-forward graphical user interface, similar to a standard Microsoft program. The software will then generate the necessary HTML, php and web scripts to administer the election online. The software would run a web server which students would connect to and submit their votes. At the conclusion of the election the system will end voting and will prepare a summary of the votes to be reviewed and analyzed by UCS.

2.0 System Model

2.1 System Model Diagram



2.2 System Model Description

The system is divided in to three main modules. Each of the modules is responsible for a different main function of the application. Here each module is discussed in turn.

2.2.1 Ballot Creation

This module provides a user interface similar to Microsoft PowerPoint through which the user designing the election can specify the format of the election as well as write the individual ballot questions. There is a logical backend which is keeping track of the questions that the user specifies and also checks to ensure that they are valid. Once the user has finished designing their ballot they can, at the click of a button, generate all the HTML, web scripts and php necessary to administer their election via the web. A data structure of ballot data is passed from the logical backend of the ballot creator to the HTML Generator. Here the ballot data is interpreted and turned in to appropriate web code.

2.2.2 Election Administration

The code generated in the preceding step is then stored on the local machine and a web server is established to host the election site. Again, at the click of a button the user can activate the server and the election will go live. As the election progresses and users visit the site and cast their votes this data is stored in a database which will serve to collect the full election results when voting is complete.

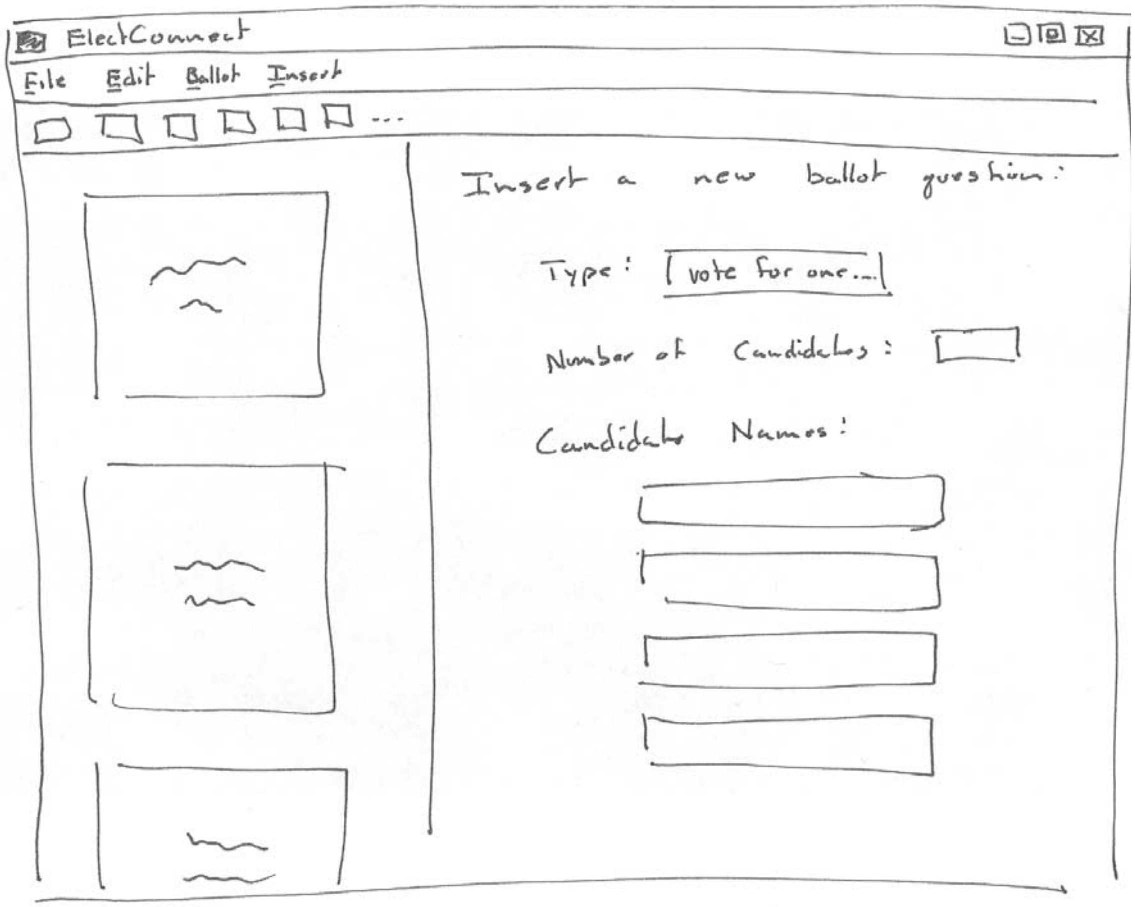
2.2.3 Data Analysis

At the conclusion of voting the server is shut down and the database of vote information is closed. This data is then passed off to the data analysis module. This module provides the most room for scalibility in this project. In the most robust version of the specifications the data analysis module will provide a graphical front end which allows the election administrators to view the election data both simply to determine who the winner was, or in more depth to seek other election statistics. In addition, this module will handle the different types of elections the software must support. For example, in a traditional election a candidate must receive a majority of the votes cast in order to win. If this majority is not achieved a re-vote is necessary after eliminating the least popular candidate. The software must also support instant run off voting, in which voters rank their candidates and if no majority is achieved votes for the last place candidate are re-distributed. The data analysis and visualization package would allow for these different schemes by providing the necessary flexibility in interpreting vote data.

3.0 User Interface

Each of the modules described above has a unique user interface. The three are discussed below.

3.1 Ballot Creation User Interface



Insert a new ballot question:

Type: vote for one...

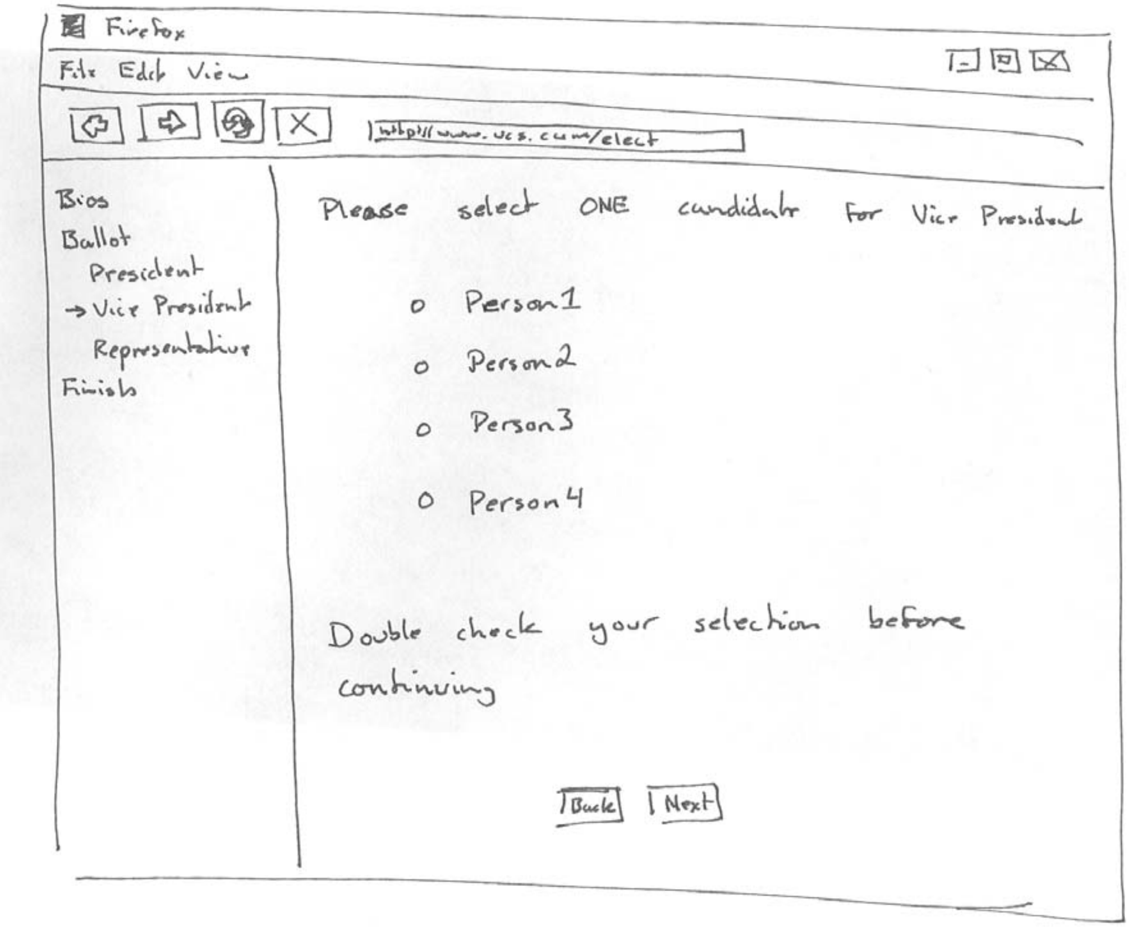
Number of Candidates:

Candidate Names:

The ballot creation interface as discussed earlier is similar in style and concept to the PowerPoint interface. There is a standard system of Windows menus along the top of the screen. The File menu provides the user with functions to save and load files as well as to print and other standard functions. Similarly the Edit menu allows the user to make alterations to previously created slides. The Ballot menu will have options tailored toward the type of ballot they are designing, allowing them to add questions, add candidates and other election specific functions. Below the textual File menu will be a row of icons providing quick, easy access to common operations which are also available through the drop down menu systems.

The main section of the screen is divided in to two parts. On the left is a chain of thumbnails of the existing ballot questions. In the right hand frame are the specific parameters of the ballot question the user is currently designing. The user can select questions in the left hand frame and they will be brought in to focus in the right hand frame for editing.

3.2 Election Administration User Interface



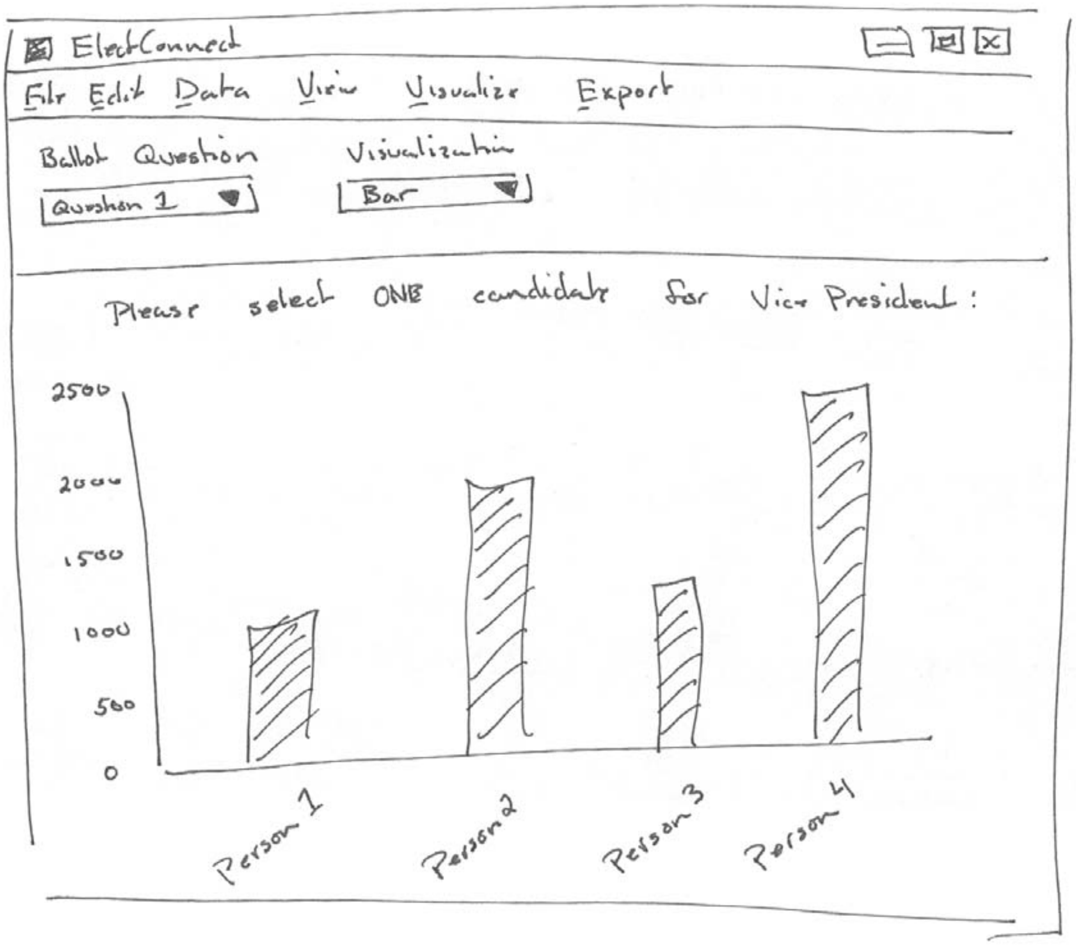
Please select ONE candidate for Vice President

Double check your selection before continuing

The election administration user interface is the web based interface which voters will visit to cast their ballots. It is critical that the web design be simple and straightforward. There will not be extensive graphics, animation etc. Features like these would only distract the user from their task and would risk their voting improperly.

Thus the website would be divided in to two sections. The left hand column will provide a constant view of where the voter is within the ballot and on the right the current ballot question will be displayed. The first section of the election website is the main page which welcomes the user to the site and thanks them for taking the time to vote. They are prompted to log in to begin the election process. The next page they see is a complete list of the positions being decided as well as the candidates running for each position. The candidates names are hyperlinks which open a new window containing their picture, short biography and personal statement. At the bottom of this page there is a link to continue. At this point the user has started toe voting process and will see a progression of pages like the one pictured above. Each page appearing in the right hand frame contains one ballot question and has both a back and a next button at the bottom. There will be limited text on the page. When the user has continued through all the ballot questions they are given a preview of their complete ballot and encouraged to review it before continuing. The page has a submit button at the bottom. When the user clicks this, their ballot is submitted to the database and they are shown a page thanking them for their votes.

3.3 Data Analysis User Interface



Ballot Question: Question 1

Visualization: Bar

Please select ONE candidate for Vice President:

The data visualization interface is a standard windows interface. It is intended for the group administering the election. It provides a framework for them to view and interpret the results of the election. At the very least it will have a drop down menu through which the user can select which ballot question they wish to analyze and a second drop down menu though which they can specify a visualization method (i.e. bar graph, pie chart, etc.). This interface will also offer options to view ballot questions which rank candidates as if they are instant run off questions. In addition the interface will allow the user the option of exporting the data to a spreadsheet.

4.0 Non-Functional Requirements

4.1 Long-Term Support

UCS requires at least one student who is familiar with the product to be on call to troubleshoot any issues the night of the election. Thus is will be important to have at least one enthusiastic and willing junior working on this project.

4.2 Performance

There are no particularly time consuming elements or complex computer science problems involved in this project. The HTML generation would have to be done within a reasonable amount of time, but there is no reason to believe that this would take anything more than minutes at most.

The website itself would have to run sufficiently fast that students would not be discouraged and quit the voting process half way through.

4.3 Reliability and Testing

Because of the infrequence of use and the importance of the function the application provides it will need to go through extensive testing and meet significant reliability metrics. The server must be able to run for 48 hours without crashing. It must be able to handle 3000 total voters and 200 simultaneous users. These standards are based on historical data, however and with an eye to the future it may be wise to revise the user numbers upwards.

The program must be subject to at least 3 full election simulations.

4.4 Portability

The system will run on windows based PCs. The possible exception would be if the stability gained from running a Linux server is enough to offset the increased work to the user of having to move the HTML to another machine.

4.5 Documentation

A full suite of documentation must be provided in the form of written instruction manuals. In application Help menus would be useful, but are not required. It is most important that the documentation allow a new user to design and administer a several question election within several hours.

5.0 Updated Functionality Requirements

5.1 Highest

- provide an interface similar to standard Windows applications for designing the ballot on a question by question basis (PowerPoint as a model)
- generate HTML, scripts, etc. necessary to run a web-based election
- election website is clean and uncluttered with a well defined flow for the user to follow
- output election results data in excel ready format
- the system must be able to support between 2,500 and 5,500 votes cast
- secure login system to ensure privacy and that each user votes only once
- support for instant run-off voting

5.2 High

- support multiple election (referendum, survey) formats
- provide flexibility for different types of ballot questions (eg. vote for one of the following, rank the following candidates, etc.)
- use a database of users to provide different ballots to different logins based on class year
- web interface viewable through IE, Netscape and Firefox

5.3 Medium

- host a site dedicated to administering the election
- the server must support approximately 200 simultaneous users
- allow for a section of the election website containing candidate biographies and personal statements

5.4 Low

- provide a feature rich interface for analyzing and interpreting election results
- support multiple election styles beyond majority wins and instant run-off

5.5 Lowest

- provide an interface through which the group designing the election can modify the appearance of the web ballot itself

6.0 Risks

6.1 Timing

Unfortunately spring elections are held before this project could reasonably be expected to be complete and fully tested. This means the application would not be used until the fall of next year, after many in this class have graduated.

6.2 Security

Security poses a risk for two reasons. First, there may be some concern that the application is vulnerable to attack because of poor design or that it has been designed specifically to allow tampering. Second, requiring students to login means we either need an independent database of student names and passwords or we must rely on the CIS NetID login system. This introduces an external dependency and may pose a technical challenge.