

USG Elections  
Proposal of Electronic System Implementation  
June 8, 2009  
Michael Yaroshefsky

The following document outlines a rough sketch of the features and implementation I would recommend for an ideal elections system.

## USG Elections System

### • **Administrative Component**

All administrators can login to an administrative component at a simple URL such as [elections.princeton.edu/administrator](http://elections.princeton.edu/administrator). Inside the secure section, the top of the system will show the date and time reported by the server and the percentage load reported by the server. Directly below, the administrator will see a list of "Election" units separated into three clearly-separated categories, each of which will be referred to as a primary state of the election unit. These three primary states determine the options available for editing that election and also determine the behavior of that election (namely, whether it is active). The three primary states are: Live, Pending, Archived.

#### ○ **Live Elections**

Live elections represent those elections that are currently accepting votes and those which have finished collecting votes but are pending tabulation, verification, and publication.

- Live elections will make immediately visible the following information in the administration home panel
  - The substate of the election (explained below)
  - The total number of votes cast since election opening
  - A link to the public front page
  - The date and time of the scheduled election closing
- The following actions are defined for live elections
  - Get Details: Although we explicitly want to prohibit any person from gaining advanced knowledge of the live election tabulations, it would be beneficial to allow the administrators to view critical data regarding the progress election without revealing individual vote tallies. The data that should be displayed include:
    - The substate of the election
    - Actual time of election opening (may not be the scheduled time if "Force Live" option is used)
    - Number of ballots cast since election opening
    - Number of ballots cast in the last 60 minutes
    - Date and time of last ballot
    - Server load
    - URL of elections site
    - Date and time of prescheduled elections close.

- Suspend: Suspending an election simply prohibits any additional votes from being cast in case of an emergency during the live voting process. Suspended elections must be able to be reactivated. When an election is suspended, the voting page will display a public error message indicating that the election has been temporarily suspended from accepting ballots. Suspended elections cannot have results tabulated. This option is available only when the election is *accepting votes*. This changes the substate to *suspended*.
- Reactivate: This is the opposite of suspending an election. Reactivating an election allows voting to continue. This option is only available when the election is suspended. This changes the substate to *accepting votes*.
- Manage Candidates: Although it should be policy that no text changes can be made during a live election, the technology must not prohibit manual control over errors that must be changed and can be changed without requiring that the election be canceled and restarted. Functionality should generally include the same functionality as will be described below for Pending Elections, with a few yet-to-be-determined exceptions (i.e. removing candidates, removing referenda, etc.). This option is available only when the election is *accepting votes*, but not after the election has been *closed*.
- Manage Referenda: For the same reason as above, the administrator should be able to manage referenda during a live election. This option is only available while the election is accepting votes, but not after the election has closed.
- Force Close: In the event that the election must be closed before the scheduled closing, this action will close the election. Closed election cannot be reopened. Only after the election has been closed can the votes be tabulated. This changes the substate to *closed*. This action might require an email confirmation.
- Process: Processing is the action of performing any algorithms needed to determine the winners. This action iterates through all “Decisions” and applies the specified algorithm. This option is only available to *closed* elections, and can only be processed once. Once this action is complete, the substate changes to *processed*. Once the results are processed, all administrators receive a notification email. Once processed, all attributes of the election are frozen and can no longer be modified.
- View Results: Once the results of a closed election have been processed, the administrator must be able to view the results before publishing them. The page for viewing will be nearly identical to the public viewing page, but will be private only to the administrator. This option is only available to closed

elections that have been *processed*. The Registrar may have a unique viewing page that provides additional information required for the review and certification, along with the ability to click a button to signal certification. It might be policy that the elections manager cannot actually see results until the Registrar has verified the results.

- View Log: The system should log any irregularities encountered by the code. This log should be visible during any substate of the election. This is separate from the administrative access log.
  - Publish Results: Once the results of a closed election have been *processed*, this action will publish the results to the public results page (to be discussed later). This action changes the election from “Live” and moves it to “Archived” with the substate of *published*.
  - The following substates are defined for a live election:
    - *Accepting votes*: This is the substate that an election first takes when it is live. This indicates that the election is accepting votes. This is the only substate in which the election accepts votes.
    - *Suspended*: This is the substate in which the election has been temporarily suspended from accepting votes. Administrators should still be able to access the ballot while an election is suspended (in order, for example, to verify that it appears correctly to the public when live), but the public must not be able to do so.
    - *Closed*: This is the substate to which the election will go once it has been closed either by the automatically scheduled date or by the manual “Force Close” option. No votes are accepted. A closed election cannot be reactivated.
    - *Processed*: The election is considered processed only once the election results have been processed according to the individual algorithms.
    - (*Published*): This substate actually belongs to the “Archived” election category. It is included only to show that it is the final substate to which an election moves once it has been published.
- **Pending Elections**  
Pending elections are those elections which are currently being created by the administrators, are accepting candidate statements and referenda, or are pending the “go live” time.
- The following details will be made immediately visible in the administrative panel:
    - Title of election
    - Current substate

- The following actions are defined for live elections:
  - Modify Details: These details that can be modified include the following:
    - Title of election: This is the title of the election for the administrative panel and the live election site.
    - Election message: This is the text message that appears below the title but above the decisions on the ballot. For example, it could include instructions or information. Basic html tags should be allowed such as [b][u][i][link="URL"], and it would be even better to have a simple WYSIWYG interface.
    - Candidate Statement Start and End time: The time window when candidates can submit statements and advocates can submit referenda. Each time entered as MM-DD-YY HH:MM, (HH based on 24 hr clock)
    - Private Ballot Review Period Start and End time: The time during which candidates and advocates can review how their statement/referenda will appear once public. This does not enable them to see all of the statements, only those that they have submitted. Someone could potentially be a candidate and submit a referenda, so the system must be able to accommodate both. Viewing statements requires login using the netID that submitted the statement.
    - Public Ballot Review Period Start and End time: The time during which the public (does not require login) can view the ballot as it will appear.
    - Live Election Start and End time: The time during which the election accepts votes.
    - Unique URL key: The key for linking to the public side of this election. For example, if the key is "spring2009", the election would be published at <http://elections.princeton.edu/spring2009> (This URL may not be exact, but the concept of the URL key holds) At times when the election is accepting candidates and referenda, this address allows students to submit or modify their statements. At times when the election is between this "public modification" substate and the live election, this page displays the time of the scheduled election opening. At times during the live election when the election is accepting votes, it displays the ballot. During the period between the election close and the released results, the page just displays a "closed" message. At times when the election results have been published, this page displays the published election results.

- Allowed upload file types: Allow only certain extensions to be uploaded as an attachment to candidate statements and referenda, such as .pdf, .doc. The extensions should be entered in a textfield, separated by commas (such as: pdf,doc,docx,ppt,pptx)
- Max upload size: the maximum size (in KB) to allow for attachments
- Manage Positions: Adding, editing, and removing a list of positions that represent human candidates. If considering the project from an object-oriented approach, each position is a child implementation of the “decision” object (just like a referenda is also a child of the decision object). The positions must be able to be reordered easily. The order in which the positions appear in this panel will be the order in which they appear on the public ballot.
  - The position object includes the following attributes:
    - Title: Such as “2012 President”
    - Description: A paragraph as taken from the election packet describing the position. (Could be used in a pop-up window when clicking on a position on the ballot.)
    - Voting Mechanism: Whether to use the single-choice option, the STV option with ordinal preferences, the STV option with ranked candidates, or another option. Voting mechanisms should be like encapsulated packages that interact with the election system in clearly defined ways, much like some object implements an interface in the Java programming language. This will enable additional mechanisms to be added in the future just like plugins. In particular, each “mechanism” must do the following:
      - Be able to generate the “decision” view as it will appear on the ballot. In other words, each “mechanism” package will specify whether the decision should display radio buttons, select buttons, textfields, etc. When called upon, it provides the html code to display a specific decision. It will also be able to store the necessary information from the ballot in the database. So when the ballot is being generated, it calls upon this “plugin” to generate the code for the ballot.

- Be able to calculate the votes and specify a winner based on the mechanism's algorithm
- Be able to generate the "result" view as it will appear after the results have been published. Each "mechanism" will simply deliver the appropriate HTML/PHP code to display the results (for example, if using the single-choice option, it would just show the number of votes each candidate obtained. But if it were the STV mechanism, it would show how many votes each choice got as "First Preference," "Second Preference," etc.) It should be able to deliver two different versions, one that indicates the winner without showing individual vote counts, and one that indicates the winner with vote counts.
- Eligible Candidates: Specification of the four currently-enrolled classes. The program should automatically calculate based on the current date which four classes are currently enrolled, and display them as options.
  - Whitelist: Explicitly allowed candidates listed by netID, to override class eligibility
  - Blacklist: Explicitly prohibited candidates listed by netID, to override class eligibility
    - The order of precedence should be the following:
      - 1) Whitelist
      - 2) Blacklist
      - 3) Class affiliation
- Eligible Voters: Specification for the four currently enrolled classes using the same way as indicated above
  - Whitelist: Explicitly allowed candidates listed by netID, to override class eligibility
  - Blacklist: Explicitly prohibited candidates listed by netID, to override class eligibility
    - Same order of preference as above
- Publish Results: Whether to publish the results for this position when the election has been processed. This will allow the results for certain positions to be withheld from the public, as may

be the pleasure of the Senate or the rules of the constitution.

- Display Vote Counts: If “Publish Results” is true, whether to display numerical results, or just the winner. For example, in some cases the winner, but not the individual vote counts should be revealed.
  - Can Upload: Whether candidates for this position can upload an attachment to the candidate statement <Boolean true/false>
  - List order: Whether to display the candidates in <Random> order or <Alphabetical> order for public viewing. If the order is random, it must be dynamically random, so that the order is randomly determined each time the ballot is loaded.
- Manage Candidates: All candidates for the election are displayed in the administrative panel, according to position, and then by the order in which they submitted their statements within each position.
    - The candidates will be presented in a table with the following data fields:
      - NetID
      - Name <As submitted by the candidate, not from LDAP>
      - Class Year <From Registrar PeopleSoft, NOT LDAP>
      - Date Submitted <Date when candidate statement first submitted>
      - Petition <Yes/No Active Checkbox> (this must be an active checkbox on this page, so changes can be made to the candidates all at once and then saved using a “Save” button at the bottom) Default is false.
      - Penalty Points <not an active textbox> Default is 0. This value is calculated as the summation of all violations (described below).
      - Expense Report <Yes/No Active Checkbox> Default is false.
      - Approved <Yes/No Active Checkbox>: Determines whether the candidate will be shown in the election. Default this is false. If this checkbox is not checked, the candidate is omitted from any candidate lists.
    - The following can be manually edited by the administrator after clicking the candidate’s name and

opening a new panel for managing the individual candidate.

- Name (as it appears on the ballot)
  - Position: By constitutional rules, candidates can change their position up until the election, so this must be able to be changed during any time of the pending election category
  - Statement: The text area. Full details will be provided later in the description of the candidate statement area
  - Photo: Allow the administrator to preview the photo, to reupload another photo to overwrite the old one, or to delete the photograph altogether
  - Attachment: Allow the administrator to download the attachment, remove the attachment, or upload a new attachment
  - List of infractions/penalty points: The elections manager must be able to add, edit, and remove infractions. Each infraction consists of a short text statement such as “Candidate spent \$5 over budget” and an integer point value. It should be easy to add additional infractions, or simply provide room to display up to 10 infractions in a table. These infractions are totaled to obtain the full count of elections violations. This list must be able to be reproduced on the ballot near the candidate’s name or in a pop-up window. When the value of a candidate’s violation points increases from a previous value, an email might be dispatched to the candidate with a list of all violations and point values.
- Manage Referenda: Referenda will appear sequentially in the administrative panel. The order in which they appear in the administrative panel should be the same order in which they appear on the ballot, so it should be possible to rearrange the referenda.
    - The referenda should be listed in a simple table with the following information:
      - Title of Referenda
      - NetID of Advocate
      - Name of Advocate
      - Class Years Voting < active checkboxes>
    - Clicking on the title of a referenda will allow the administrator to edit the following details of the referenda:

- Title of Referenda
  - Name of Advocate
  - Class Years Voting
  - Referenda Text
  - Supporting Argument Text
- Force Live: This action brings the election immediately live and accepting votes.
- Duplicate: Doing so creates a copy of the entire election, including all positions. However, it does not duplicate the candidates or referenda. This action would make it easy to turn the election into a template for future elections.
- Archive: Archiving the elections changes the primary state to “Archived.” Since this is archiving an election that has never run, no results are displayed.
- Delete: This action deletes permanently the entire election. This action requires an email confirmation.
- The following substates are defined for a pending election
  - *Newly Created*: An election assumes this substate after being created from scratch or duplicated from an old election
  - *Accepting Submissions*: During this substate, the public front end is accepting candidates and referenda.
  - *First Administrative Review*: This is the substate after candidates have submitted referenda during which the elections manager can review candidate statements and make corrections/changes if necessary.
  - *Private Ballot Review*: During this substate, candidates and advocates can login to review their own candidate statement/referenda as it will appear on the ballot. If there are any changes they would like, they can use a simple form on the website to request the changes.
  - *Final Administrative Review*: During this substate, the elections manager makes the final review of the ballot.
  - *Public Ballot Review*: This is the substate after the elections manager has approved the ballot for public review. During this substate, there is a public page (no login required) where the ballot appears as it will during the voting, except that voting is disabled.
- Archived Elections
 

Archived elections are those which have either been archived after an election has taken place (votes have been cast) or after a pending election was prematurely archived. In the case of archived elections after the election has past, the election page should remain live with the final output (typically vote counts). The following actions are defined for archived elections:

  - Get Details: This action opens a new page with more details from the election. It displays all of the positions, corresponding candidates,

and referenda. It further allows the administrator to click on any one position to obtain vote details, any candidate to view candidate properties, and any referenda to view details an vote counts. Below the positions and referenda, there are the basic details about the election, including end time of the election, a link to the public result page,

- Duplicate
- Delete

- **Super Administrative Permissions Manager**

In addition to the primary administrative center, there is an additional panel, accessed through the main administrative panel, that allows super administrators (super administrator is a non-student position, such as Dean Dunne) to manage permissions for primary administrators. In other words, Dean Dunne can login to give access to new elections managers and remove access from old elections managers. This panel would also give the super administrator the ability to assign other super administrators. There are currently only two levels of administrator: *super administrator* is reserved for non-USG administrators such as Dean Dunne or the Registrar, and *administrator* is the access level of the elections manager and any other authorized USG personnel.

- The page would display two tables, one for each access level. The tables would display each person as a row in the table with the following columns:
  - Name (as retrieved from LDAP)
  - netID
  - Added (when this administrator was added)
  - Last login (date and time of last login)
  - Expiration (when this person's access expires – only non-super administrators can have expiring access)
  - <Remove> button to instantly remove access privileges
- The following restrictions apply to managing privileges
  - Super administrators can add other super administrators
  - Super administrators can remove other super administrators
  - Super administrators cannot have their access privileges expire on a timer
  - Super administrators cannot remove their own access privileges

- **Public Candidate and Advocate Submission Form**

Interface to be outlined at a later date

- **Private Candidate Statement and Referenda Review Form**

Interface to be outlined at a later date

- **Public Ballot**

Interface to be outlined at a later date

- **Public Results Page**

Interface to be outlined at a later date