InnoVote ReliaVote Central Server

Functional Design

By
Erin Thead
Software Engineer
erin@erinthead.com

Table of Contents – ReliaVote Central Server Functional Design

1. Introduction

1.1. Purpose.

The purpose of this document is to communicate the software requirements and functional design for the InnoVote ReliaVote Central Server software. The document provides a detailed description of functional, performance, and security requirements, design constraints, and classes of persons who will be using the software.

The intended audience of this document is the developer and any other persons interested in the project, including election reform activists, computer security professionals, political figures with an interest in election reform, and potential buyers of the design.

1.2. Scope.

InnoVote ReliaVote Central Server (CS) is one component of an interoperable line of products. It is a software product designed to execute on a standard commercially available Intel™ or Macintosh™-compatible computer.

ReliaVote CS will send and receive input from computers executing InnoVote ReliaVote Precinct Edition [ref. 6]. ReliaVote CS will store election data received from these products in a secure electronic database.

ReliaVote CS will also have the ability to initiate certain operations on the ReliaVote PE software by sending data transmissions to this product in a form that it will be able to recognize. The details of the operations that ReliaVote CS can initiate on external machines can be found in reference [6].

Current vote tabulation software is insecure and unreliable. Existing tabulators, which are used in voting sites across the United States, have had numerous and severe documented errors and security problems, including the following:

- Little to no protection of vote tallies from tampering
- The suspicion of pre-election and pre-recount rigging of tabulators
- Unverified updates being made to election software
- Public online access to sensitive operations of election software

ReliaVote CS is a highly secure and accountable system with strong anti-fraud protection. Election data are heavily protected both from errors and from tampering. In the event that data stored on a ReliaVote CS machine *are* compromised, the system will detect this and inform the users (presumably election officials) of it, at which point they can act in a manner prescribed by law with respect to ensuring the integrity of the data. As is described in references [2], [6], and [7], other InnoVote products have built-in security measures to ensure that at least one uncompromised copy of an election’s data will remain.

1.3. References.

[1] Thead, E. *InnoVote CardReader Hardware Requirements Overview*, 2005.

[2] Thead, E. *InnoVote CardReader Functional Design*, 2005.

[3] Thead, E. *InnoVote Database Detailed Design*, 2005.

[4] Thead, E. *InnoVote MyVotronic Hardware and Operating System Overview*, 2005.

[5] Thead, E. *InnoVote Network Detailed Design*, 2005.

[6] Thead, E. *InnoVote ReliaVote Precinct Edition Functional Design*, 2005.

[7] Thead, E. *InnoVote SecureDRE Functional Design*, 2005.

[8] Thead, E. *InnoVote Database Access Matrix*, 2005.

[9] Thead, E. *Security Analysis of InnoVote Products*, 2005.

1.4. Overview.

The remainder of this document is organized in the following fashion:

Section 2: Provides overall description of the system, including product functions, user classes, assumptions, and generalized dependencies.

Section 3: Provides specific requirements including functional requirements, performance requirements, and security requirements.

Section 4: Provides supporting figures and tables for information in other sections of the document.

2. Overall Description

2.1. Product Functions.

ReliaVote Central Server will need to perform the following basic functions:

1. Provide user authentication measures to ensure that no unauthorized persons may use the software
2. Provide user management features to add, modify, and remove software users
3. Detect all precinct computers in the county in which it will have to communicate
4. Allow a user to “program” the software to read and store election results. This involves adding and/or modifying entries in Database tables.
5. Send data transfers to a ReliaVote PE-compatible computer to “program” it
6. Receive final vote information and tallies from precinct computers
7. Check the finalized votes and tallies against each other and inform the user if there are anomalies
8. Display election results
9. Accept, verify, and display recount results
10. Maintain error logs for itself and receive error logs from voting equipment in the county

2.2. User Classes.

2.2.1. Administrator user class

The Administrator user class represents any person who has the legal authority to use a computer in the possession of a Board of Elections and view the results of an election. This class of user will need to initiate operations that modify sensitive tables in the InnoVote Database but should *not* have direct access to the Database.

Administrator privileges are granted to any user who has knowledge of a correct user name and password for ReliaVote CS.

An Administrator does not have direct access to the Database. Any requests for modifications to the database will be performed at the System privilege level.

2.2.2. System user

The System shall function as a user class. It must be able to perform any operation necessary on the hardware and data stored in the system, within the permissions of the computer's operating system. Operations in software functions initiated by Administrators may be executed at System level. These functions are described in §3 of this document.

The System will in theory have full access to the Database at all times, but the software must be coded so that it will in *practice* not perform certain damaging operations to the Database, and that no human user or external machine will be able to interact with ReliaVote CS with System privileges. Additionally, the Database shall be configured to disallow certain actions [reference 2].

2.3. Assumptions and Dependencies.

2.3.1. Hardware platform assumption

The document assumes that InnoVote ReliaVote CS will execute on an Intel- or Macintosh-compatible processor. It should exist in versions for at least the following operating system families: Microsoft® Windows®, Macintosh® OS X®, and Linux.

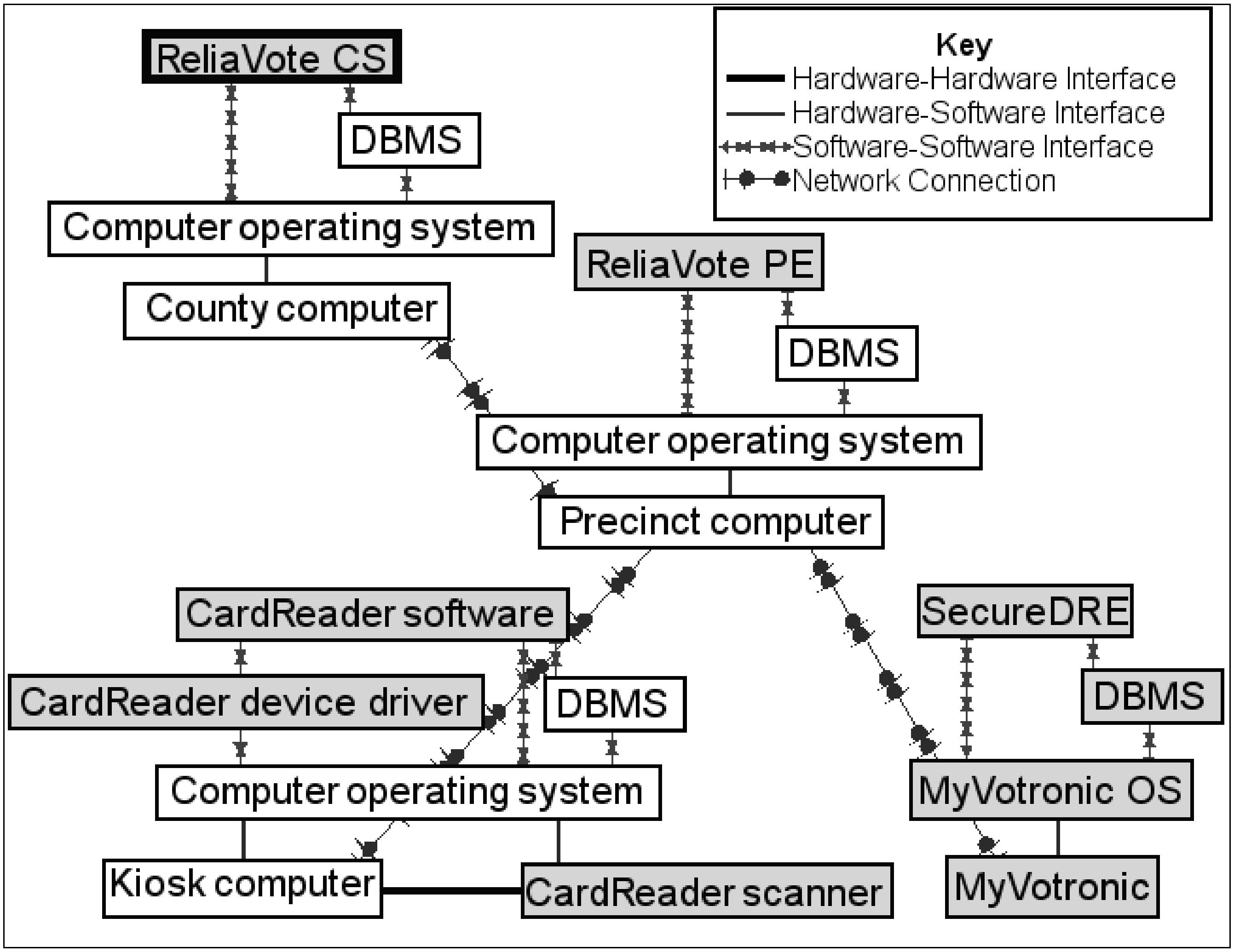
2.3.2. Internet connectivity assumption

The document assumes that hardware running InnoVote ReliaVote CS will be connected to a TCP/IP-compatible network during operation. IP version 6 is not necessary for correct operation of ReliaVote CS, but it does provide extra security benefits.

2.4. Deployment of the Software.

Figure 1 shows the deployment diagram for all InnoVote products and necessary third-party components. Items that this document describes are surrounded with thick boxes.

Figure 1: Deployment Diagram.



3. Specific Requirements

3.1. Functional Requirements.

The features described in this section are operations that are necessary for correct and useful operation of the ReliaVote CS software product.

3.1.1. System Feature 1: Detect precinct computers

3.1.1.1. Purpose of Feature

This feature allows the system to establish a means of communication with the precinct computers in the county, which is running InnoVote ReliaVote PE or compatible software.

3.1.1.2. Stimulus-Response Sequence

1. The system obtains the IP addresses and communication port numbers of the ReliaVote PE-compatible systems within the county. (The system can obtain this information either from a stored list or from user input.)
2. The system accesses the county's Kerberos server and receives a ticket for each precinct computer. *Note:* The tickets are not obtained all at once; rather, the system executes the remaining steps for each precinct computer.
3. The system sends a "beacon" packet to each of the machines at the IP addresses and port numbers that it has read. The packet contains instructions to send back a particular response.
4. The behavior of the system depends on the success of Step 3.
    a. If an IP address does not exist or the port number is not open, the system does not receive any packet in response. It displays an error message after a given period of time and record the event in the Events table of the Database. It then exits this series of steps.
    b. If a packet was received but could not be decrypted, the recipient sends back a generic acknowledgment packet. The system executes Steps 5 – 7.
    c. If a packet was received and decrypted, the recipient sends back the correct data. The system stores the IP address and port number as the correct address and port number for communication with the precinct computer. It records an entry in the Events table and exits this series of steps.
5. The system processes the packet and determine that the machine it sent its packet to was not the correct precinct computer.
6. The system outputs an error message and record the event in the Events table of the Database.
7. The system reverts to Step 1.

3.1.1.3. Dependencies

This feature requires the successful completion of System Feature 2, “Login a user,” before it can begin execution.

This feature requires Security Features 1, “Verify identity of data transmitters,” 4, “Encrypt outbound data,” 5, “Database login,” and 7, “Block all ports except two.”

3.1.2. System Feature 2: Login a user

3.1.2.1. Purpose of Feature

This feature allows the system to authenticate a stored username and grant the user privileges to perform the remainder of the system features.

3.1.2.2. Stimulus-Response Sequence

1. A user indicates to the system that he or she wishes to login.
2. The system prompts the user for a username.
3. The user inputs a username to the system.
4. The system validates the username against a list of usernames.
5. The operation of the system depends on the result of Step 4.
    a. If the user’s username is present on the system’s list, the system executes Steps 6 – 9.
    b. If the user’s username is not present, the system displays an error message indicating that the username is not valid and reverts to Step 2.
6. The system prompts the user for a password.
7. The user inputs a password to the system.
8. The system validates the password against the password stored for the username that the user is attempting to use.
9. The operation of the system depends on the result of Step 8.
    a. If the user’s password matches the password that the system has stored, then the system executes Steps 10 – 11.
    b. If the user’s password is not correct, the system displays an error message indicating that the password is not valid and reverts to Step 6.
    c. If the user inputs an incorrect password 5 consecutive times for a particular username, the system reverts to Step 2 and creates an entry in its Events table indicating that a user input an incorrect password 5 times. The entry also contains the username that the user was attempting to use and the date and time of each login attempt.
10. The system creates an entry in its Events table indicating that the username has logged in. This entry also contains the date and time of the login and the number of attempts that the user required to input the correct password.
11. The system displays the ReliaVote CS main screen to the user and grants the user Administrator privileges, and with this, the ability to perform the remainder of the System Features.

3.1.2.3. Dependencies

This operation requires the completion of no other System Features before it can begin execution. Conditional: If a user is already logged in, then that user must logout before a new user can login. ReliaVote CS is not a multi-user software product.

3.1.3. System Feature 3: Configure the Database

3.1.3.1. Purpose of Feature

This feature allows an authenticated user to store in the Database lists of election information, plus all contests, candidates/ballot options, and political parties that are participating in the election.

3.1.3.2. Stimulus-Response Sequence

1. A user indicates to the system that he or she wishes to configure the Database with election information.
2. The system displays an interface for the user to input the contests, precincts, candidates/ballot options, political parties, and any pertinent information about the data items. This interface allows the user to add, modify, and delete entries.
3. The system reads the item input by the user and the requested operation to perform on it.
4. The system attempts to perform the operation.
5. The system's operation depends on the result of Step 4.
    a. If the operation is permitted by the database management system, the system executes it.
    b. If the operation is not permitted, the system rejects it and displays a message to the user indicating the reason for its rejection.
6. The system reverts to Step 2 and continues the sequence of steps until the user indicates that he or she is finished.
7. The system reads the user's input indicating completion of the configuring.
8. The system records an entry in its Events table indicating the success or failure of the operation.
9. The system displays the ReliaVote CS main screen to the user.

3.1.3.3. Dependencies

This feature requires the successful completion of System Feature 2, "Login a user," before it can begin execution.

This feature is restricted by Security Feature 3, "Limit changes to Database on Election Day." The feature requires Security Feature 5, "Database login."

3.1.4. System Feature 4: Design a ballot

3.1.4.1. Purpose of Feature

This feature allows the system to provide a human user with a means of designing a paper ballot that can be read by the CardReader ballot scanner.

3.1.4.2. Stimulus-Response Sequence

1. The user indicates to the system that he or she wishes to design a ballot to be used in a county.
2. The system reads the list of candidates/ballot options, political parties, party affiliations, contests, and precincts for a particular election.
3. The system prompts the user for the size of paper on which the ballot will be printed.
4. The system calculates the number of ballot options to appear on the ballot and rescales the text of each one so that they all fit on the user-selected page size.
5. The system displays a screen to the user containing a ballot layout editor. The editor contains options for every candidate or option that will appear on the ballot except for the option "no candidate" (abstention from voting), and the user is not permitted to modify, add, or delete information from the ballot editor.
6. The user arranges the ballot options in a particular order on the ballot using "drag and drop" functionality of the mouse.
7. The system detects when the user has moved a particular option to a horizontal or vertical coordinate very close to that of another option, and it arranges the options to align perfectly.
8. The user indicates to the system that he or she is finished arranging options on the ballot.
9. The system prompts the user for the type of shape to appear next to each ballot option.
10. The user selects a shape.
11. The system generates a hollow shape next to each option on the ballot. This shape will be filled in by a voter during an election.
12. The user indicates to the system that he or she wishes to save the ballot.
13. The system saves the ballot configuration to the hard disk.

3.1.4.3. Dependencies

This feature requires the successful completion of System Feature 3, "Configure the Database," before it can begin execution.

This feature requires Security Features 2, "Encrypt Database tables," and 5, "Database login."

3.1.5. System Feature 5: Print ballots

3.1.5.1. Purpose of Feature

This feature allows a user to print a given number of uniquely coded ballots that conform to a design saved in the ReliaVote CS system.

3.1.5.2. Stimulus-Response Sequence

1. The user indicates to the System that he or she wishes to print a ballot design that is currently opened by the Software for editing.
2. The System allows the user to specify the number of ballots to be printed. Depending on the options that the printer driver allows, the System may allow one or more of the following options: printing on one or both sides of the ballot paper, specifying paper size, specifying paper type, and specifying color features of the printouts.
3. The System chooses a random number and uses it to generate sequentially-increasing numeric representations of bar codes. The System generates the same number of bar codes as the number of ballots specified by the user to print.
4. The System allows the user to associate the random number with an election stored in the Database. (This value is stored in the “random_seed” attribute of the Elections table.)
5. The System converts the numeric representations to binary format.
6. The System converts the binary representations to black-and-white graphical representations of bar codes in a format that the printer driver can use.
7. The System analyzes the ballot design and determines where it can place the bar code so that it does not interfere with anything else on the ballot.
8. The System instructs the printer to print one and only one bar code on the same location on each ballot. Once a bar code has been used by the printer, the System discards it and does not allow it to be used again.
9. The System waits for the printer to complete the print job.
10. The System detects when the printer has finished its task and displays a message to the user indicating that the operation has finished.

3.1.5.3. Dependencies

This feature requires the successful completion of System Feature 4, “Design a ballot,” before it can begin execution.

This feature is restricted by Security Features 3, “Restrict changes to Database on Election Day,” and 5, “Database login.”

3.1.6. System Feature 6: Program the ReliaVote PE machines

3.1.6.1. Purpose of Feature

This feature allows the system to configure the Database of any ReliaVote PE-compatible computer over a network.

3.1.6.2. Stimulus-Response Sequence

1. The user indicates to the system that he/she wishes to program the database of one or more ReliaVote PE-compatible computers in the county.
2. The system displays a list of all ReliaVote PE-compatible computers in the county.
3. The user selects either a single machine or multiple machines to program.
4. The system creates a copy in RAM of the contests, candidates, and political parties for a given precinct that are stored in the Database tables.
5. The system sends the data to the selected computer or computers, transmitting to each precinct computer only the information for that precinct.
6. The system receives a notification from the computer(s) of the computer's success or failure to add the data.
7. The system's operation depends on the results of Step 6.
    a. If the notification indicates a success, the system displays a message to the user that the data were successfully added to the machine.
    b. If the notification indicates a failure, the system displays a message to the user that the programming failed. This message also contains the error code and error message sent by the voting machine.
8. The system records an entry in its Events table indicating the success or failure of the operation.
9. The system displays the ReliaVote CS main screen.

3.1.6.3. Dependencies

This feature requires the successful completion of System Features 1, "Detect precinct computers," and 3, "Configure the Database," before it can begin execution.

This feature is restricted by Security Feature 3, "Limit changes to Database on Election Day." It requires Security Features 1, "Verify identity of data transmitters," 4, "Encrypt outbound data," 5, "Database login," and 7, "Block all ports except two."

3.1.7. System Feature 7: Accept final data from ReliaVote PE

3.1.7.1. Purpose of Feature

This feature allows ReliaVote CS to accept and store the final or recounted election results from each ReliaVote PE-compatible computer in the county.

3.1.7.2. Stimulus-Response Sequence

1. The system receives a request from a ReliaVote PE-compatible computer to send its finalized vote tallies, its “Votes” (or “Recount_Votes”) Database table, and its “Realtime_Votes” Database table.
2. The system sends a response to the machine requesting its final tallies and votes.
3. The system receives the data from the machine.
4. The system attempts to add the data to the Database.
5. The system’s operation depends on the result of Step 4.
   a. If the system successfully added the data to the Database, it records an entry in its Events table, transmits a response to the machine that the data were successfully added, and executes Step 9.
   b. If the system failed to add the data to the Database, it transmits a response to the machine that the data were rejected and executes Steps 6 – 9.
6. The system creates an entry in the Events table containing the error code, the machine on which the error occurred, the date and time of the error, and any other error information.
7. The system displays a message to the user that the data were rejected from the Database and allows the user to dismiss the message. This message also contains any error codes and error messages generated by the database management system.
8. The system processes the user’s input from Step 7.
9. The system displays the ReliaVote CS main screen.

3.1.7.3. Dependencies

This feature requires the successful completion of System Feature 6, “Program the ReliaVote PE machines.”

This feature requires Security Features 1, “Verify identity of data transmitters,” 2, “Encrypt Database tables,” 4, “Encrypt outbound data,” and 5, “Database login.” It is restricted by Security Feature 6, “Restrict data flow to Database tables.”

3.1.8. System Feature 8: Check voting results for anomalies

3.1.8.1. Purpose of Feature

This feature allows the ReliaVote CS system to check the three sources of election data – real-time voting results, finalized votes, and final candidate tallies – for inconsistencies and anomalies. (There are only two sources of data in a ballot recount, no real-time results.) It informs the user of inconsistent data.

3.1.8.2. Stimulus-Response Sequence

1. The system creates copies in RAM of the Database tables containing ballots, real-time individual votes (if applicable), final individual votes, and final candidate tallies.
2. The system parses the real-time vote table and calculates candidate tallies from the entries in the table. It excludes any votes from the tallies if they are associated with an archived ballot. <u>Conditional</u>: If the system is checking recounted results, it skips this step.
3. The system parses the final vote table and calculates candidate tallies from the entries in the table. It excludes any votes from the tallies if they are associated with an archived ballot.
4. The system compares the two calculated candidate tallies to the separately stored candidate tallies and to each other. <u>Conditional</u>: If the system is checking recounted results, it does not use real-time data in its comparison.
5. The system's behavior depends on the result of Step 4.
   a. If all three tallies are the same for every candidate on the ballot, then the system records an entry in its Events table and exits this sequence of steps.
   b. If any one of the three tallies for any candidate is different from the other two, then the system performs Steps 6 – 8.
6. The system displays a prominent message to the user indicating that the vote totals do not all match.
7. The system recommends that the user conduct a recount of the paper ballots.
8. The system generates entries in the Events table in the Database detailing what has occurred.

3.1.8.3. Dependencies

This feature requires the successful completion of System Feature 7, "Accept final data from ReliaVote PE," before it can begin execution.

This feature requires Security Features 1, "Verify identity of data transmitters," 2, "Encrypt Database tables," 4, "Encrypt outbound data," and 5, "Database login."

3.1.9. System Feature 9: Accept error records from ReliaVote PE

3.1.9.1. Purpose of Feature

This feature allows the ReliaVote CS system to accept and store error records from all ReliaVote PE-compatible computers in the county.

3.1.9.2. Stimulus-Response Sequence

1. The system receives a request from a ReliaVote PE-compatible computer to send an error record from its "Events" Database table.
2. The system sends a response to the machine requesting its error record.
3. The system receives the data from the machine.
4. The system attempts to add the record to its Events table in the Database.
5. The system's operation depends on the result of Step 4.
   a. If the system successfully added the data to the Database, it records a success message in its Events table, transmits a response to the machine that the data were successfully added, and executes Step 9.
   b. If the system failed to add the data to the Database, it transmits a response to the machine that the data were rejected and executes Steps 6 – 9.
6. The system creates an entry in the Events table containing the error code, the machine on which the error occurred, the date and time of the error, and any other error information.
7. The system displays a message to the user that the data were rejected from the Database and allows the user to dismiss the message. This message also contains any error codes and error messages generated by the database management system.
8. The system processes the user's input from Step 7.
9. The system displays the ReliaVote CS main screen.

3.1.9.3. Dependencies

This feature requires the successful completion of System Feature 6, "Program the ReliaVote PE machines."

This feature requires Security Features 1, "Verify identity of data transmitters," 4, "Encrypt outbound data," and 5, "Database login."

3.1.10. System Feature 10: Detect and log errors

3.1.10.1. Purpose of Feature

This feature allows the system to detect exceptions and errors and store auditable information about them in the Database for later retrieval.

3.1.10.2. Stimulus-Response Sequence

1. While performing an operation, ReliaVote CS encounters an error or exception.
2. ReliaVote CS attempts to write the contents of the system's temporary memory (hereafter the "memory dump") to the hard disk.
3. The system's behavior depends on the type of error that is encountered.
   a. If it is a non-fatal error, ReliaVote CS records the system error code, identification of the machine where the error occurred, date of the error, and time of the error in the "Events" Database table. ReliaVote CS also records the type of error and an error message, if provided. It then executes Steps 5 – 6.
   b. If it is a fatal error that requires that ReliaVote CS be restarted, ReliaVote CS attempts to write the error code, date, time, error type, and error message to the hard disk. It then reboots and performs Steps 4 – 6.
4. The software restarts. ReliaVote CS reads the hard disk to determine whether the software experienced a fatal error condition the last time it shut down. Since this was the case, ReliaVote CS reads the memory dump from the hard disk.
5. Using the system state information in the memory dump, ReliaVote CS attempts to continue or restart the operation that was being performed when the error occurred.
6. ReliaVote CS resumes normal operation.

3.1.10.3. Dependencies

This feature requires the completion of no other System Features before it can begin execution. It does require that an error condition occur.

This feature requires Security Features 1, "Verify identity of data transmitters," 2, "Encrypt Database tables," and 4, "Encrypt outbound data."

3.1.11. System Feature 11: Display election results

3.1.11.1. Purpose of Feature

This feature allows the system to display tallies for candidates for the county.

3.1.11.2. Stimulus-Response Sequence

1. The user indicates to the system that he/she wishes to view the tallies for each candidate for the county.
2. The system retrieves the entries from the Tallies or Recount_Tallies table of the Database and orders them by precinct.
3. The system displays the results in a readable format.

3.1.11.3. Dependencies

This feature requires the successful completion of System Feature 8, "Check voting results for anomalies," before it can begin execution.

This feature requires Security Features 2, "Encrypt Database tables." It is restricted by Security Feature 5, "Database login."

3.1.12. System Feature 12: Clear election results

3.1.12.1. Purpose of Feature

This feature allows the system to remove all vote data and tallies from the Database after an election is over and the data are no longer needed. This operation can be performed *only* after a predetermined period of time after the day set for Election Day. It should be noted here that results cannot be deleted in part; this operation clears the entire Database table. This is an anti-fraud mechanism.

3.1.12.2. Stimulus-Response Sequence

1. The user indicates to the system that he/she wishes to clear election results.
2. The system checks to ensure that enough time has elapsed. Conditional: If the enough time has not elapsed, the system displays a message indicating that the operation cannot be performed yet and skips to Step 4.
3. The system deletes all entries from the Votes, Realtime_Votes, Recount_Votes, Tallies, Recount_Tallies, and Ballots tables in the Database.
4. The system records an entry in its Events table indicating the success or failure of the operation.
5. The system displays the ReliaVote CS main screen.

3.1.12.3. Dependencies

This feature requires the successful completion of System Feature 2, "Login a user," before it can begin execution.

This feature is restricted by Security Feature 3, "Limit changes to Database on Election Day." It requires Security Features 1, "Verify identity of data transmitters," 4, "Encrypt outbound data," and 5, "Database login."

3.1.13. System Feature 13: Create a new user

3.1.13.1. Purpose of Feature

This feature allows the system to create new usernames with Administrator privileges.

3.1.13.2. Stimulus-Response Sequence

1. The user indicates to the system that he/she wishes to create a new username.
2. The system prompts the user for the username to be created.
3. The user inputs a username to the system.
4. The system determines whether the requested username contains invalid characters. Conditional: If the system finds that the username contains invalid characters, it displays a message to the user and exits this sequence of steps.
5. The system compares the requested username against all existing usernames to determine if it is a duplicate. Conditional: If the system finds that the username already exists, it displays a message to the user and reverts to Step 2.
6. The system prompts the user for a password for this user.
7. The user inputs a password to the system.
8. The system determines whether the password is sufficiently long. Conditional: If the password is shorter than a predetermined length, the system displays a message to the user and reverts to Step 6.
9. The system assigns the new username Administrator privileges.
10. The system converts the new user's password to a hash value using a hash encryption function.
11. The system stores the username, hashed password, and privilege level in its list of users.
12. The system records an entry in its Events table indicating the success or failure of the operation.

3.1.13.3. Dependencies

This feature requires the successful completion of System Feature 2, "Login a user," before it can begin execution.

3.1.14. System Feature 14: Change a user's password

3.1.14.1. Purpose of Feature

This feature allows a user who is logged in to change his/her password.

3.1.14.2. Stimulus-Response Sequence

1. The user indicates to the system that he/she wishes to change his/her password.
2. The system prompts the user for a new password.
3. The user inputs a password to the system.
4. The system determines whether the password is sufficiently long. <u>Conditional</u>: If the password is shorter than a predetermined length, the system displays a message to the user and reverts to Step 2.
5. The system prompts the user to re-enter the password to confirm it.
6. The user inputs the password to the system.
7. The system determines whether the password is the same as the first password that the user input. <u>Conditional</u>: If the passwords are different, then the system displays a message to the user and exits this sequence of steps.
8. The system converts the new password to a hash value using a hash encryption function.
9. The system replaces the old hashed password with the new one in the list of users.
10. The system records an entry in its Events table indicating the success or failure of the operation.

3.1.14.3. Dependencies

This feature requires the successful completion of System Feature 2, "Login a user," before it can begin execution.

3.1.15. System Feature 15: Delete a user

3.1.15.1. Purpose of Feature

This feature allows a user to delete another username from the list of users. The system does not allow the user to delete the username that is currently logged in; this is to ensure that there is always at least one username that can login to the system.

3.1.15.2. Stimulus-Response Sequence

1. The user indicates to the system that he/she wishes to delete a username.
2. The system displays the list of all usernames to the user. The system also provides the user with the opportunity to cancel the operation.
3. The user makes a single selection on the list.
4. The system prompts the user to confirm the deletion of the selected username. The system also provides the user with the opportunity to cancel the operation.
5. The user indicates to the system that he/she wishes for the username to be deleted.
6. The system determines if the username to be deleted is currently logged in. Conditional: If the username is currently logged in, the system does not allow the deletion. It displays a message to the user and exits this sequence of steps.
7. The system removes the username, password, and privilege level from the list of users.
8. The system records an entry in its Events table indicating the success or failure of the operation.

3.1.15.3. Dependencies

This feature requires the successful completion of System Feature 2, “Login a user,” before it can begin execution.

3.1.16. System Feature 16: Manually logout a user

3.1.16.1. Purpose of Feature

This feature allows the system to logout a user. ReliaVote Central Server allows only one user at a time to be logged in; if a user wishes to use the software when another one is logged on, the current one must logout first.

3.1.16.2. Stimulus-Response Sequence

2. The user indicates to the system that he/she wishes to logout.
3. The system allows any currently executing functions to finish and terminates any that are not responding.
4. The system terminates the user's session.
5. The system creates an entry in its Events table that the username has logged out. This entry also contains the date and time of the logout.

3.1.16.3. Dependencies

This feature requires the successful completion of System Feature 2, "Login a user," before it can begin execution.

3.1.17. System Feature 17: Automatically logout a user

3.1.17.1. Purpose of Feature

This feature allows the system to automatically terminate a user session when the system has been idle for a certain period of time. This allows for security in case the computer is left unattended for extended periods of time.

3.1.17.2. Stimulus-Response Sequence

2. The system begins a timer after a given period in which none of the system events detailed in this document have executed.
3. If a system event executes, the system stops the timer and exits this sequence of steps.
4. If the timer reaches a predetermined number, the system automatically logs out the user that is logged in.
5. The system allows any currently executing functions to finish and terminates any that are not responding.
6. The system terminates the user's session.
7. The system creates an entry in its Events table that the username has logged out. This entry also contains the date and time of the logout.

3.1.17.3. Dependencies

This feature requires the successful completion of System Feature 2, "Login a user," before it can begin execution.

3.1.18. System Feature 18: View events

3.1.18.1. Purpose of Feature

This feature allows the user to view the events that have occurred during the course of operation or that have been received from a precinct computer.

3.1.18.2. Stimulus-Response Sequence

1. The user indicates to the system that he or she wishes to view the event log for the software.
2. The software reads the entries in the Events table of the Database and displays this information in read-only format to the user. The software allows the user to dismiss the information.
3. The software processes the user's input from Step 2.
4. The software displays the ReliaVote CS main screen.

3.1.18.3. Dependencies

This feature requires the successful completion of System Feature 2, "Login a user," before it can begin execution.

This feature requires Security Feature 5, "Database login."

3.2. Performance Requirements.

The features described in this section are requirements that are necessary for ReliaVote Central Server to perform in a reasonable amount of time.

3.2.1. Performance Requirement 1: Modify Database quickly

Numerous features of ReliaVote Central Server require changes to be made to the Database. Any operations involving the Database must take place in an amount of time that would be unnoticeable to a typical user of the software.

As is described in §3.3, subsections, numerous operations require that parts of the Database employ cryptography for data protection and machine identity verification. The cryptographic algorithms used must be executed in a reasonable amount of time and be unnoticeable to a typical user.

3.2.2. Performance Requirement 2: Transmit and receive data quickly

ReliaVote Central Server has to receive, transmit, and process large amounts of data. This requires that the computer have access to a broadband Internet connection and a fast link between it and all computers with which it exchanges data. ReliaVote Central Server itself must be able to accept the large amounts of data without slowing its processing excessively or losing any of the data.

3.3. Security Requirements.

The features described in this section are requirements that are necessary to ensure the integrity of election data generated and stored by the ReliaVote Precinct Edition software product.

3.3.1. Security Feature 1: Verify identity of data transmitters

3.3.1.1. Purpose of Feature

This feature restricts the sources of data that will be allowed to perform privileged operations on ReliaVote CS. This security feature protects the Database from malicious operations. It requires that any data transfers processed by ReliaVote CS must have originated at a precinct computer within the county.

3.3.1.2. Characteristics of Feature

When any data transfers are received, the system will check to ensure that they have been encrypted. The details of this system are given in reference [5], *Network Detailed Design*.

The keys will be managed by a Kerberos-compatible key management system. Systems whose keys the ReliaVote CS system will need are the system itself and the precinct computers for each precinct in the county.

If ReliaVote CS determines that a packet was sent by the machine it appears to be from and intended for the machine that received it, then the packet will be processed. Otherwise, the packet is discarded.

This security feature acts as a virtual private network connection between the central computer and the precinct computer(s).

3.3.2. Security Feature 2: Encrypt Database tables

3.3.2.1. Purpose of Feature

This feature provides for encryption of sensitive Database tables in the ReliaVote CS internal Database.

3.3.2.2. Characteristics of Feature

The Votes and Realtime_Votes tables in the Database are highly sensitive and must be protected with encryption of at least 128-bit strength when not being modified. This encryption scheme can be symmetric or asymmetric; the two options are detailed in reference [3]. The keys to this encryption system must not be known or recoverable by human users.

The entire table is encrypted rather than individual entries. This means that it is not possible to add, modify, or delete entries while the table is encrypted.

When the System initiates an operation that involves one of these tables, it decrypts the affected table, performs the operation, generates a new key, and re-encrypts the table with the new key. This means that the key is changed every time an operation is performed on one of the tables. The software generates different keys for each table.

3.3.3. Security Feature 3: Limit changes to Database on Election Day

3.3.3.1. Purpose of Feature

This feature restricts user access to particular tables in the Database for a time period on and immediately after Election Day.

3.3.3.2. Characteristics of Feature

The Database contains numerous tables containing information about candidates, contests, and political parties. These tables can be modified by Administrators until 12:00 A.M. on Election Day. At this time all of the Database will be locked from modification (except for the Ballots, Votes, Realtime_Votes, Tallies, and Events tables) for a given period of time after Election Day ends. Only the System user can modify these tables during this lockdown.

3.3.4. Security Feature 4: Encrypt outbound data

3.3.4.1. Purpose of Feature

This feature requires that all data transfers destined for a machine external to the precinct computer must be encrypted.

3.3.4.2. Characteristics of Feature

All outbound data transmissions from ReliaVote CS will be protected with encryption of at least 192-bit strength.

The feature utilizes the same cryptosystem described in Security Feature 1.

3.3.5. Security Feature 5: Database login

3.3.5.1. Purpose of Feature

This feature requires that all accesses of the Database be made by a verified "user" that the database management system recognizes. This is to prevent unauthorized SQL querying.

3.3.5.2. Characteristics of Feature

The database management system will recognize the ReliaVote CS software as a "user." The software must authenticate itself when it makes any modification to the Database. The database management system will not permit anonymous SQL querying.

3.3.6. Security Feature 6: Restrict data flow to Database tables

3.3.6.1.Purpose of Feature

This feature restricts traffic to sensitive tables in the Database.

3.3.6.2.Characteristics of Feature

The Votes and Realtime_Votes, and Recount_Votes tables in the Database are highly sensitive. In addition to being protected by strong encryption and secret keys, they are protected from access by operations that did not originate on the local computer. ReliaVote CS analyzes all incoming data packets to determine if they contain instructions to modify or view either of these tables. If the packets contain modification instructions, then the instructions contained therein are processed only if they originated on a precinct PC in the county. Otherwise, and for *any* packets containing view instructions, the packets are discarded. No other machine should be initiating such operations. All packets that involve Database operations are "repackaged" by the ReliaVote CS software as a ReliaVote CS operation, so that the Database will recognize and execute them.

3.3.7. Security Feature 7: Block all ports except two

3.3.7.1.Purpose of Feature

This feature requires all data ports on the county computer will be blocked from sending and receiving data transmissions except for one over which the System will exchange data with the precinct computer and another which it will use for data transfer with the Kerberos server.

3.3.7.2.Characteristics of Feature

ReliaVote CS will initiate connections on one data port, and this port will be used only by InnoVote software. Additionally, the client installation of Kerberos software will use a data port to communicate with the key server for the county network. However, a hardware firewall will be configured to disallow traffic originating from or destined for any port other than the chosen ones. The firewall must not permit any configurations that would permit incoming or outgoing traffic from ports other than these on the computer running ReliaVote CS. More information about this is present in *Network Detailed Design* [4].

3.4. System Attributes

The attributes described in this section are, unless otherwise stated, general to the ReliaVote CS software product rather than specific to a particular system feature.

3.4.1. Reliability

The ReliaVote CS software must experience normal exception-free behavior at least 99.999 percent of the time. This would correspond to no more than one exception within a 24-hour period.

3.4.2. Availability

The ReliaVote CS software will execute on a computer, and all users must be physically present to use it. Remote logins are not permitted. Availability is not an issue with this software product.

3.4.3. Security

The security requirements of ReliaVote Central Server are detailed in §3.3, “Security Features.”

3.4.4. Maintainability

The system must be upgradable if necessary. Any upgrades must require no changes to the existing relational schema for the Database. They must not compromise any Security Features of the software.

3.4.5. Portability

The software must execute on any Intel- or Macintosh-compatible uniprocessor computer system.