Software Requirements Specification

E-Elections Software

"The Pericles Project"

Produced by: Team Dragon

Amir Reda Atalla
Mathew Evans
Mike Levy
Cory McKay
Sitai Sun

TABLE OF CONTENTS

I. INTRODUCTION

1.1 Purpose of this Document

This document is provided in order to ensure that the software that the *development team* produces will be consistent with the needs of all *customers*. It is a description and elaboration of the project requirements that the development team has been provided with. Stating these requirements explicitly helps ensure that any potential miscommunications are dealt with at an early stage, when the cost of implementing changes is still low.

Customers are encouraged to distribute this document among their potential *users* and management in order to provide us with feedback. This will help the development team ensure that the end product fully meets all needs. This document will also be a useful resource for those who will be upgrading or maintaining the software after it has been completed.

1.2 Scope of this Document

The development team arrived at the information contained in the original version of this document by examining the original project description in an individual and group setting, by conducting research on the web and in libraries and by discussing the system.

Many updates have already been made to this document in order to make it more readable. Some changes have also been incorporated into the requirements themselves, as a result of the response to the original requirements document. Further updates to this document could occur if this project is explored further in the future.

This document makes use of several terms in very narrowly defined ways. The reader is referred to the glossary at the end of this documents if he or she encounters a word that seems confusing. The first occurrences of all words in the glossary are italicized in the text, except in cases where the text itself defines them explicitly.

1.3 Overview

A software package entitled “Pericles” will be the ultimate product of the development team. This software will allow organizations to design and run elections securely and privately on a *server* connected to a *distributed network*. It will be possible to customize the *ballots* for each election as well as to set certain parameters for each election, such as whether or not *voters* may change their votes once they have been submitted. The Pericles package will also include the software needed by voters to register their votes using computers connected to a distributed network. The server will then track votes and calculate statistics on the results of the election. The

long term goal of this project is to greatly reduce the cost and complexity of running elections by removing the direct involvement of humans in the mechanics of gathering and counting votes.

All aspects of the software will use a graphical user interface. The system will be furnished with a full *on-line help system*, as well as installation software.

1.4 Scope of the Pericles Project

This Pericles software should only be regarded as a pilot project, meant to examine the feasibility of voting technology and to explore its potential. It is not intended for elections on a national scale, at least at this stage. However, the software will be designed to be scalable to full-scale elections given greater time, manpower and testing resources.

II. GENERAL DESCRIPTION

2.1 Product Functions

The Pericles software package will be made up of four basic components:

2.1.1 Pericles Elections Server

The Server will be responsible for storing the settings of each election, generating passwords for *authorized Voter*s, receiving and authenticating votes, storing votes on the Pericles Voting Database, generating statistics at the end of each election and maintaining and verifying security and voter privacy. The Server will also potentially contact all authorized voters by e-mail to give them their username information, passwords, Server address, election code, instructions to obtain the Voting Client and contact information of the Elections Officer.

2.1.2 Pericles Election Editor

The Election Editor will be a piece of software that enables Elections Officers to design custom ballots and define configuration settings for each election. It will also allow the Elections Officer to suspend an election. The Elections Officer will also use the Election editor to enter the *enumeration list*. The enumeration will be a list of all voters that are authorized to vote in an election. This list could include the usernames and e-mail addresses of each authorized voter. It will be possible to configure elections so as not to require voter e-mail addresses, if this is desired.

2.1.3 Pericles Voting Database

The Voting Database will hold the enumeration list for each election as well as all votes registered for each election. It will be encrypted and will not be directly accessible by anybody. The votes stored from each election will be deleted from the *database* at a preset time after the termination of each election.

2.1.4 Pericles Voting Client

The Pericles Voting Client will be a simple piece of software that voters with only a minimal background in computers will be able to install on their own computers. They will use it to vote by establishing a network connection to the Server and sending their encrypted votes. They will enter the Server address and their username, password and election code before establishing the connection. Once they have connected, the Server will send them the ballot, which they will then fill out using the Client and send back to the Server. Voters will also use the Client to change their votes if this option has been enabled in a particular election.

2.2 User Characteristics

2.2.1 Customers

The customers are the people or organizations who purchase the Pericles Elections Software. They will be authorized to host elections on the Server. The Voting Client will be available free of charge, and any purchasers of the server software will be authorized to distribute it to their voters. Each customer will be responsible for providing a System Administrator to overlook the installation and operation of the Server. The customers will also be responsible for providing a host for the Server.

2.2.2 System Administrator

A System Administrator will be required to oversee the installation and operation of each Pericles Elections Server. The System Administrator will not have control over or access to any particular elections once they are activated on the server, but he or she will be the only one able to authorize Elections Officers to start new elections. The System Administrator is assumed to have at least a college level background in computers and networking. This person is necessary to ensure that the overall system is working and that its security integrity is not breached.

2.2.3 Elections Officer

The Elections Officer is an impartial individual who is given responsibility for overseeing individual elections by the organization holding each election. Each election housed on a Server may have one or more Elections Officers and it is possible for an Elections Officer to be responsible for more than one election. This person is needed to monitor each election in a way that is independent of the general overall maintenance provided by the Systems Administrator.

Once an Elections Officer is authorized to set up an election by the System Administrator, he or she is the only one with the ability to design ballots, configure election options, enter an enumeration list and suspend the election. He or she is also responsible for answering questions over e-mail that voters or others may have during the election.

The Elections Officer is expected to be well versed in the elections protocol of the organization whose election he or she is supervising and is expected to be comfortable using *GUI*-based computer applications. It is assumed that the Election Officer speaks and writes English, since the implementation team will not have the time or the knowledge to write multiple versions of the Elections Editor software to accommodate other languages. This could be done in a future release of the software.

2.2.4 Voters

Voters are those people who are authorized by the Elections Officer to vote in each election using the Pericles Voting Client. They are expected to have access to a fully networked computer and to be comfortable using GUI-based computer applications. They must also have a secure access to a private e-mail address.

2.3 Operating Environment

2.3.1 Pericles Elections Server

The Elections Server will be written in Java, so the computer hosting it must be capable of running Java bytecode. The Server will run under *Linux*. The computer hosting the server must be accessible by other machines on a network. Testing will be done using the computers in Reynolds 008 as a basis for the minimum hardware requirements to run the Pericles Election Server. Professor Stacey stated in class that she did not expect the software would run the US election, but that it would be reasonable to assume that the class could go down to Reynolds 008 to vote. From this statement we infer that the Pericles Election Server must be able to run on one of the computers in the lab. It must be able to host an election where Voters can use the Voter Client software on all of the other computers in Reynolds 008 to participate in an election. There are approximately forty machines in the Reynolds lab. Thus, we will insure that at least forty voters can simultaneously vote on any one election in the Reynolds lab.

2.3.2 Pericles Election Editor

The Election editor will be written in Java, so the computer hosting it must be capable of running Java bytecode. Since we are using the lab in Reynolds as a basis for the minimum requirement to host the Pericles Election Server, the lab computers will also serve as the basic hardware requirement for the Pericles Election editor.

2.3.3 Pericles Voting Database

The Voting Database will be required to hold voter identities, as stated in the project handout. The computer hosting the database must have either Postgres or *MySQL* installed.

2.3.4 Pericles Voting Client

The Voting Client will be written in Java, so the computers that it is installed on must be capable of running Java bytecode. It will run under Linux or Windows 95 and above. It is expected that all computers it will be installed on will have high speed Internet access such as cable modem or DSL, as was stated in class. The computers that will run the client should be able to run GUI-based Java computer programs.

2.4 User Problem Statement

As it stands, almost no elections are held electronically. This means that there is a great cost associated with collecting and counting votes, since many people must be hired to perform and check these tasks. Manual elections take a long time to set up, and occurrences such as recounts can greatly delay the reporting of results. All of these time delays can be at least partially eliminated by having computers run elections.

There are also many problems relating to the accuracy of manual elections. The intentional inaccuracies introduced by the corruption of election officials can be eliminated by having the election handled by an entirely impartial computer. The

unintentional inaccuracies of manual elections, such as improperly printed or filled out ballots, can also be eliminated by electronic elections which use clear and consistent interfaces.

It should be noted that although electronic elections have a great number of advantages, they are largely untested and thus even the best systems may be prone to problems, at least initially. Although they can certainly be designed with full security and privacy features, corruption of the results can be very difficult to detect if someone does manage to break through the security of the software. The privacy feature of the software means that what the software is actually doing during an election cannot be transparent to election officials. One must also be very careful that security holes are not built into the elections software by the developers and that the officials who run and maintain the elections do not have the power to corrupt results.

2.5 User Objectives

Any organization running an electronic election will want software that is easy to install and run. Ballots must be easy to design and they must be flexible as to the number and types of questions. The Election Server must run efficiently and securely. It must be impossible for anyone to break into the system and corrupt the results, prematurely know the results of the election, vote when they are not authorized to do so or vote more than once. The results of the election must be clearly presented upon the completion of the election.

Voters need the ballots to be clear and easy to fill out and answer. It is important that it be impossible for anyone to associate a voter's name with his or her vote.

2.6 General Constraints

The development team must design, develop and test this software within the space of three months. They also have important limitations placed on their time due to many other projects that they must work on. They also suffer from severe lack of funding. Due to these constraints, as well as the limited number of people working on the project, it may be necessary to prioritize certain aspects of the project over others. Functionality and security will be the first priorities.

The developing and testing environment is limited to the University of Guelph computer labs. This means that the developers do not have access to the full commercial system that is necessary to fully test this system under realistic working conditions.

III. SYSTEM REQUIREMENTS

The primary priorities of this design are, in order of importance:

1) Functionality
2) Reliability
3) Maintainability
4) Security and Privacy
5) Scalability
6) Interfaces

3.1 Functional Requirements

The requirements that are essential are marked with a dash, while the ones which are less essential are marked with a bullet.

3.1.1 Election editor

- There will be the option of providing additional information about election issues on the ballot.
- The ballot will have an arbitrary number of questions.
- The text of each question must be specified.
- All questions will be multiple choice. The number of choices as well as the texts of the choices may be customized.
- The start date and end date of the voting must be specified.
- Allows the Elections Officer the option of letting voters change their votes before the election ends.
- Lets it be specified how long the information stored on the database will persist after the election terminates.
- Lets it be specified from a limited list of options what statistics are wanted when an election ends. This list will include the results for each option of each question (total number of votes and percentages) as well as voter patterns such as the time distribution of votes.
- An enumeration list must be entered which could include the usernames, passwords and e-mail addresses of each authorized voter.
- Will allow the Elections Officer to change the election settings, provided that the election has not started yet.
- Provides Elections Officer with an election code to uniquely identify each new election that is created.

- The look and feel of the ballot will be customizable.
- Allows the Elections Officer to specify a time window to accommodate differences in time settings on individual computers.
- Allows the Elections Officer to decide what criteria will be used by voters to identify themselves from a limited numbers of options.

- What qualifies as a win must be customized from a limited number of options (i.e. is there a quorum for the election and/or each question, is there a minimum percentage of votes in favor needed for a choice in each question to win and do the results of certain questions depend on the results of other questions).
- There will be the option of having *weighted votes* to accommodate situations such as shareholders' elections.
- Specifies whether voters can refuse or spoil individual questions or their entire ballots.
- Must only allow creations of elections when it is authorized by the System Administrator.
- Will indicate to the Elections officer if there are not enough systems resources to accommodate a new election.

3.1.2 Election Server

- Stores the settings for each election after they are generated by the Election editor.
- Stores and generates passwords for all authorized voters on enumeration lists.
- Checks all Voting Clients when they attempt to log on to see if they are authentic (valid election code and a username and password that correspond to an authorized voter in that election that has not voted yet).
- Sends a copy of the appropriate ballots to Voting Clients.
- Sends each voter who has successfully voted a verification number to prove that they have voted.
- Encrypts all communication with the Voting Client.
- Stores all votes on the Election Database.
- Generates voting statistics at the end of each election. Statistics must only be available after the election has ended, not while it is still being carried out.
- Generates a complete list of statistics and a partial list of statistics.
- E-mails these statistics to the Elections Officers and another source.

- Must be able to accommodate multiple elections simultaneously.
- Contacts all authorized voters on enumeration lists by e-mail to give them their username information, passwords, Server address, election code, instructions to obtain the Voting Client and contact information of the Elections Officer.
- Reports any attempted breaches of security to the Elections Officer.
- Allows the Elections Officer to suspend the election.

3.1.3 Election Database

- Stores the enumeration list for each election.
- Stores votes as they are authenticated by the Server.
- Encrypts all stored data.
- Only allows the Server to read stored data.
- Deletes the data stored from each election at a preset time after the completion of each election.

3.1.4 Voting Client

- Can access an Election Server using a distributed network.

- Allows voters to log on to a Server by entering its address, their username, password and election code.
- Receives ballots from the Server after successfully logging on.
- Allows voters to enter their votes and send them to the Server.
- Encrypts all communications with the Server.

- Allows voters to change their votes after having submitted them if this is allowed in a particular election.

3.2 System Attributes

3.2.1 Communications Security

All communications between the Voting Client and the Elections Server must be encrypted to ensure the privacy of votes and voter information. *Encryption* of communications will also ensure that anyone *packet sniffing* over the network will be unable to extract any usable information from the data that is sent between the Client and Server. The server will time out after three minutes if no message is received from the Client.

All data sent to the server will conform to a pre-defined format to enable the software to detect any tampering of data in transit. If detected, the data will be discarded and the voter prompted to resubmit his or her vote.

Voters must be sure that nobody else has access to their e-mail addresses, as anyone reading their e-mail would have access to their voter identification information. The Elections Server will limit the number of login attempts to prevent automated attacks to gain or prevent access, however the voters must ensure that their password is secure as the system would not be able to detect misrepresentation of the voter.

No encryption is foolproof, although many are highly reliable. It must be understood that developments of new algorithms could potentially break any encryption, as could sustained efforts with high-powered computers. Elections should thus be run over limited periods of time, such as a single day, in order to minimize the chances of *security breaches.*

3.2.2 Storage Security

All voters must be able to record their votes anonymously without anybody being able to determine how they voted or change their votes. If voters are going to be able to change their own votes, the system must store their identification information along with their votes. This means that the file storing their votes must be encrypted so that nobody can read it directly at any stage and should be deleted soon after the election ends, but not so soon as to make a recount impossible. It must therefore be as close to impossible as possible for anyone to break into the database for at least the length of the election plus the amount of time the election data is stored afterwards.

3.2.3 Maintainability
The software will be well documented and it will be designed to be modular. The use of object oriented programming will also help to increase maintainability. This will make it easier for future developers to make changes and updates to the software with a minimal amount of effort.

3.2.4 Scalability
Both the Pericles software and this document are meant to be easily scalable to increase the scope and size of elections. All efforts will thus be made to use a software design that does not have built in size limitations.

3.2.5 Reliability
All efforts will be made to write software that is entirely reliable. However, the viability of electronic voting rests, in part, on the ability of systems administrators and elections officials to incorporate redundancy into any deployed voting system and to develop contingency plans for possible failures.

3.2.6 Interface
All aspects of the Pericles system will have a simple point and click interface using menus, text fields, buttons and all of the other components of systems with graphical user interfaces. This interface will be designed to be consistent. The interface will be designed to help accommodate people with disabilities such as colour blindness. The system will also be have a full on-line help system. Voting results will be posted on web servers in *HTML* format.

IV. DESIGN CONSTRAINTS

4.1 Language Constraints

The software will only operate in English and will only allow ballots that use 7-bit *ASCII*. This is because the first release of this project is only expected to be an exploration of voting technology, so it is reasonable to assume that it will be used primarily in North America. Both ASCII and English are used as standards in international computing. Future versions of this software could be produced in other language, but the current implementation team will not have the time or the linguistic expertise to do so.

4.2 Software and Hardware Constraints

4.2.1 Voting Client

The Voting Client software will only be tested to run on Linux or Microsoft Windows 95 or higher. The system with the Voting Client must be capable of running Java bytecode and must have access to the Internet with high-speed access such as cable or DSL. This was stated as a valid assumption in the class lab. The Voting Client must be written in Java and the description language for the ballot must be written in *XML*. The minimum hardware requirements for the Voting Client (which are the same as those for the JDK 1.1.2) are:

- Pentium 166-MHz or faster processor
- At least 32 Mbytes of physical RAM
- 65 Megabytes of free disks pace

4.2.2 Election Server

The Election Server will be written in Java with any configuration files written in XML. The Election Server will only be tested in the Linux environment. It may or may not run on other *operating systems*. The number of concurrent voters using the Election Server at any one time will not be limited, in order to allow for scalability. However, for the purposes of this project it will only be guaranteed that the system will function properly with less than forty concurrent voters. This number is based on the number of computers in the Reynolds 008 lab, since it was stated in the class lab by Professor Stacey that it would not be unreasonable to expect the class to be able to go to the lab and all vote in the Reynolds 008 lab. For similar reasons, the minimum hardware requirements for the Election Server are the same as the computers in Reynolds 008.

4.2.3 Election Database

The computer hosting the Election Database must have either MySQL or Postgres installed.

4.3 Computer Language Constraints

All configuration files must be in XML. The statistical reports must be generated in HTML. The database used by the server must use MySQL. All software must be written in Java.

4.4 Encryption Constraints

The development team is limited in the type of encryption that can be used for building the system by what is either available in the Linux and the Windows Operating System, what can be found in Java libraries or by what they can write themselves. It is not in the budget to purchase third- party encryption software.

4.5 Illegal Voter Activity Constraints

There is a danger that outside of a public polling place, a voter could be coerced into voting for a particular candidate, or selling his or her vote. It will also be difficult to control vote solicitation at the time of voting. The Pericles software will have no provisions to prevent any of these problems.

4.6 Installation Constraints

The installation of the Election Server will already assume that MySQL has been installed and that the computer running the Election Server will be able to connect to the MySQL database. It is also required that there be network access to the computer running the Election Server, and that there is a capable Network Administrator and Database Administrator to carry through the installation.

The installation of the Voting Client will assume that the target computer will have network communication already set up.

V. VALIDATION CRITERIA

5.0 Validating the Software Requirement Specification

The development team would like to ask the customer to review this requirements document and verify it with all of the software stakeholders. This will ensure that all conceptions of the product are consistent. Requests for additions or changes should be submitted at this stage so that they can be incorporated into this document. It is more costly to implement changes at later stages of development.

5.1 Performance Bounds

The following sections indicate the performance parameters that the development team will test and ensure:

5.1.1 Voter Authentication

All Voters attempting to access the system will be acknowledged by the server within a certain time frame of attempting to gain access. Given that the network is operational, the system will acknowledge the Voters presence with a message. The average computer user is only willing to wait so long for most software to respond. Typically, a use case would indicate the longest length of time a user is willing to wait for a response for a particular function of the software. Since the development team does not have this option, they will assume that the maximum waiting time is thirty seconds.

5.1.2 Voter Ballot Submission

All voters submitting a ballot will have their vote acknowledged within a certain time of submitting the ballot. Typically, a use case would indicate how long users are willing to wait for a response. Since that is not an option, it will be assumed that the maximum waiting time is thirty seconds.

5.1.3 Number of Concurrent Voters Using the System

The maximum number of Voters able to concurrently log into the election system is undetermined at this stage. The maximum number of Voters able to vote concurrently with guaranteed success is forty. This is based on the number of computers in Reynolds 008.

5.1.4 Election Registration

When an Elections Officer creates a new election on the system, it will be expected to take priority over voters trying to vote, so the time it takes to register should be only a few seconds (the time needed to access the database).

5.1.5 Number of Ongoing Elections Allowed on One Elections Server

The number of *ongoing election*s allowed on one election server is not specifically defined at this stage. There are three factors that will limit it: hard disk limitations, network server traffic and the number of concurrently logged in voters that the Election Server can accommodate.

Hard Disk limitations will dictate how many elections an Election Server can hold at a time. The software must indicate to the Elections Officer when hard disk space is too low to register another election. The System Administrator should have knowledge of the Network Traffic that is on the network. If the expected increase in network traffic from a new election is going to cause a Network problem, then the network traffic load will have to be increased or the Election Server will have to be hosted on a different network.

The Elections Server will have limitations with respect to the number of Voting Clients that it can serve concurrently. The Election Server is intended to be scalable in design. However, for the initial project delivery, we will be imposing a limit of forty concurrent voters. It is up to the discretion of the System Administrator to determine if another election on an Election Server will be too limiting to the Voters. Factors influencing this decision are election duration and the number of *enumerated voters*.

5.1.6 Largest Enumeration Size of an Election

The largest number of Voters that can be registered in the Election Database as eligible voters is limited only by hard disk availability on the system hosting the Election Database. However, the network traffic and the number of Voting Clients that the Election Server can serve at one time should be considered when an election is created.

5.1.7 Election Registration Limitation of One Registration at a time

Only one elections officer can register an election at any particular time. This is meant to increase the security of the election registration process.

5.2 Testing

A test plan will be developed from the onset of design to ensure that testing is not an afterthought. The development team will automate testing by writing software to test all of the components of the system. Some testing will also be done manually. Tests and their results will be documented. The following test classes will be considered necessary:

5.2.1 Code Reviews

The development team will have scheduled *code reviews* to continually reexamine each other's work and detect problems as soon as possible, before new components are integrated into the system.

5.2.2 Functional Testing

To check that the software will what it should under normal conditions, the development team will run through as many user scenarios as possible.

5.2.3 System Stress Testing

Tests that cause the system to exceed the specified limits of RAM, CPU, and hard disk space will be run to ensure that the system fails safely.

5.2.4 System Load Testing

It will be verified that each computer in Reynolds 008 can have a voter submitting votes simultaneously. Tests will also he run that exceed this load, to ensure that the system still behaves safely with higher than expected loads.

5.2.5 Data Verification Testing

5.2.5.1 Calculation of Election Results

Tests will be run to ensure that all totals are calculated correctly and that all statistics are accurate.

5.2.5.2 Dates

Tests will be run to ensure that date limitations are adhered to by the system and that Voting Clients are able to have their votes submitted and properly counted near election date boundaries.

5.2.6 Input Testing

All components of the system will be tested using erroneous input. It will be assumed that the software will be misused. The expectations of the software under any misuse is that its security will not fail, that no data will be irretrievably lost and that it will give informative error messages.

5.2.6.1 Voter Client Input Tests

Tests will be run to ensure that voters can only vote in elections that they are authorized to vote in and that they cannot revote unless this is permitted in a particular election.

5.2.6.2 Election Editor Input Tests.

Tests will be run with invalid election options such as inverted start at end dates for elections.

5.2.7 Security Testing

5.2.7.1 Denial of Service Attacks

The development team will subject the Server to *denial of service attacks* to ensure that the election can be properly shut down and restarted in the exact state that it left off.

5.2.7.2 Encryption Breaking

Tests will be run to ensure that encrypted transmitted information cannot be decoded within a time that falls within the duration of an election. This must be further researched. The development team must know how long election data is typically saved and how long it takes to break various encryption methods.

5.2.7.3 Faulty Database Access

Tests will be run to ensure that the Database cannot be accessed in a manner which would breach security or the privacy of Voters.

5.2.8 Regression Testing

The development team will continually test the system at all stages of production in order to ensure that it is fully functional. All errors and malfunctions reported in the software testing process will be documented and tested with each new addition or release of the software.

5.2.9 Recovery Testing

Tests will be run to ensure that the state of an election can be completely restored if a power failure occurs or the Elections Server is shut down.

5.2.10 White Box Testing

During the design phase of the software, the development group will be creating white box test plans. These plans will involve writing extra components to test the internal structures of the software.

5.3 Prototyping

The development team will present the customers with working prototypes of limited functionality at various stages of the production process. This will enable the client to be fully aware of all progress and provide useful feedback.

VI. OPERATIONAL SCENARIOS

The development team will provide operational scenarios to highlight the major functionality to be delivered in the software. These scenarios can be used to validate the functionality of the system. It is expected that more scenarios will be added and that the details of existing scenarios will be filled in as the projects scope is better realized.

Scenario 1.0: Elections Officer creates a new election and registers it.
An Elections Officer goes to the computer that has the Election editor component on it and securely gains access to use it. He or she then enters the appropriate information needed to create a new election.

After entering all the appropriate information, the Elections Officer can preview the ballot. After previewing the ballot, the Election chooses to submit the election for registration. The system will register the election, and report to the Election Officer that either the election was registered successfully or that an error occurred with an explanation of why the election could not be registered.

Scenario 1.1 – Elections Officer creates a new election but cancels before registering it.
The Elections Officer performs all of the actions from Scenario 1.0 up to the point after previewing the ballot. The Elections Officer, after previewing the ballot, chooses to cancel. The ballot and election information is not registered. It is discarded by the system.

Scenario 1.2 – Elections Officer creates a new election and makes changes to the election before registering it.
The Elections Officer performs all of the actions from Scenario 1 up to the point after previewing the ballot. The Elections Officer, after previewing the ballot, chooses to make changes. He or she can change any of the information that was entered into the Election editor up to this point. After making changes, the Election Officer can submit the election or cancel all the information entered.

Scenario 2 –Elections Officer calls off an election registered to occur.
The Elections Officer goes to the computer housing the Election editor and securely gains access to it. The Elections Officer requests that the system remove an election. It is required to identify the election that he or she wishes to remove and confirm that the election is to be deleted.

Scenario 3 – Elections Officer changes a ballot prior to an election date.
The Elections Officer goes to the computer housing the Election editor component and securely gains access to it. He or she requests to change an election. He or she is required to identify the election that he or she wishes to change. As long as the current time is before the deadline specified to disallow changes, the Election editor will allow the Elections Officer to change any of the data that was input into the system when the ballot was generated. If the time is after the deadline that was specified to be able to cancel or

change the election, the system will display a message informing the Election Officer that it is too late to change the election.

Scenario 4 – Elections Officer tries to change or delete a ballot after the deadline for last election changes.
An Elections Officer attempts to change an election that is ongoing. He or she goes to the computer with the Election editor component and securely gains access to it. The Elections Officer requests to change an election. He or she is required to enter the election code of the election. The election that he or she specifies is an ongoing election. The system informs the Elections Officer that the election is already underway, and that it cannot be changed or deleted.

Scenario 5 – Enumerated Voter votes in an election with a valid vote submission.
An enumerated voter has the Voting Client software up and running. He or she has been enumerated to vote in an election. The computer that he or she is at has access to the server that is hosting the election. The date is during the period of the election. The Voter is able to securely connect to the Elections Server that is hosting the election. The Voting Client software displays the appropriate ballot for the election. The voter fills in his or her choices to the questions. The ballot is filled in correctly, so the voter is able to submit the ballot. Before the Ballot is submitted, the Voting Client software asks the Voter to review his or her vote before it is sent to the Server. The Voter reviews and submits the vote. Upon submission, the Voter gets a receipt that can be printed. The receipt contains the votes that were made and a verification number.

Scenario 6 – Enumerated Voter participates in an election but cancels his or her vote.
Everything proceeds as in scenario 5 up until the ballot is submitted. The Voter reviews and decides to cancel his or her vote when the Voting Client software asks the him or her to review her vote before it is sent to the server.

Scenario 7 – Enumerated Voter attempts to participate in an election after the election date.
An enumerated voter has the Voting Client software at his or her computer. He or she has been enumerated to vote in an election. The computer that he or she is at has access to the server that is hosting the election. The date is after the period of the election. The Voter attempts to securely connect to the election server that hosted the election. The Client informs the Voter that the election has already taken place and that it is too late to vote in the election.

VII. PRELIMINARY SCHEDULE

- The Software Requirements Specification Document will be completed and submitted for review by May 22, 2001.
- The next step will be to amend this document based on the feedback received.
- A design document will then be produced, which will also be submitted for review.
- Once the design document is completed, work will begin on implementing the system.
- Testing will be performed throughout all stages the development phase to ensure quality.
- Once the development phase begins, the software will be demonstrated every Monday.
- A first release version of the software will be completed for distribution by early August of 2001.

VIII. CONCLUSION

It is the development team's hope that this document will be the first part of a continuing series of interchanges between themselves and customers. This will ensure that customers' needs are met in a cheap and timely fashion. It will be important to involve potential Elections Officers and Voters in this feedback process, as end-users such as they often have many unique insights that might not occur to software developers or people involved in management. This interchange will involve both information such as this document and prototypes of the product. The end result will be a product that is functional, reliable, secure and easy to learn and use.

IX. APPENDICES

9.1 Glossary

ASCII: a standard 7-bit code for representing characters—letters, digits, punctuation marks and control instructions—with binary values, the code values ranging from 1 to 127.

Authorized Voter: a voter who is on the enumeration list and who has been given the means to use the Pericles Voting Client software.

Ballot: a means of registering a vote. In the case of the Pericles system, this is a set of information and questions that can be transmitted from a Pericles Elections Server to a Pericles Voting Client and back.

Client: a program that connects to a server. In the case of the Pericles system, the Voting Client is a piece of software that Voters can use to connect to a Pericles Election Server and receive ballots, send their votes and possibly change their votes if this is permitted in an election.

Code Reviews: meetings where software developers review each other's code as a quality control mechanism.

Customers: peoples or agencies who purchase the Pericles system.

Database: any aggregation data. Files consisting of records (or tables), each of which is constructed of fields (or columns) of a particular type, together with a collection of operations.

Denial of Service: explicit attempts by attackers to prevent legitimate users of a service from using that service. Examples include:

- attempts to "flood" a network, thereby preventing legitimate network traffic
- attempts to disrupt connections between two machines, thereby preventing access to a service
- attempts to disrupt service to a specific system or person

Development Team: a team of people developing software.

Distributed Network: a network in which processing, storage and other functions are handled by separate units rather than by a single main computer.

Elections Database: database system used for storing election questions, Elections Officer information, voter information and votes.

Election Editor: Graphical User Interface (GUI) application that allows the Election Officer register an election, change its settings and customize ballots.

Election Officer: an impartial individual authorized to run an election.

Elections Server: a server that manages an election and that the Pericles Voting Client can connect to. It is responsible for jobs such as database management, user identity checks and reporting of statistics.

Encryption: the process to making information indecipherable to unauthorized readers. This can be used for the storage and transmission of information.

Enumeration List: a list of identification information and other required information for all authorized voters.

Enumerated Voter: a voter that is registered for specific election.

GUI: a graphics–based user interface that incorporates icons, buttons, pull-sown menus and a mouse.

Hacker: a person who secretively invades other people's computers to inspect or tamper with the programs or data stored on them.

HTML: an acronym for HyperText Markup Language. The document format used on the World Wide Web. Web pages are built with HTML tags, or codes, embedded in the text.

Linux: an operating system. A version of Unix.

MySQL: a relational database management system (RDBMS) that uses Structured Query Language (SQL) for adding, accessing and processing data in a database.

Ongoing Election: an election that is currently in progress.

On-line Help System: a reference on how to use a piece of software that is integrated into the software.

Operating System: the software responsible for controlling the allocation and usage of hardware resources such as memory, central processing unit (CPU) time, disk space and peripheral devices.

Packet Sniffing: the act of intercepting information traveling across a network.

Security Breach: the act or a result of breaking the security of a software system.

Server: a program that serves clients and controls access to all or part of a computer's resources.

System Administrator: a person responsible for setting up and maintaining the operational performance of a server and database.

Users: Persons who will use the Pericles system to run elections or vote. This includes Election Officers, Systems Administrators and Voters.

Voter: a person with the right to vote for a given election.

Voting Client: client software that allows authorized voters to cast and possibly change a ballot.

Weighted Vote: votes of unequal votes that are weighted based on criteria such as stake in a company.

XML: acronym for eXtensible Markup Language. It is a subset of the SGML document language designed for use on the Web and sanctioned by the W3C. It allows for definable tags.

9.2 References

http://avirubin.com/evoting.security.html
http://internetdollar.com/elections/elections.html
http://notablesoftware.com/evote.html
http://www.cert.org/tech_tips/denial_of_service.html
http://www.netvoting.org/resources.htm
http://www.research.att.com/~lorrie/voting/hotlist.html