# Software Requirements Specification

Online Voting System

Produced by: RHYTHM DHINGRA

MCA 2ND YEAR

3RD SEMESTER

2200290140123

# TABLE OF CONTENTS

1. [INTRODUCTION 3](#_TOC_250008)
   1. Purpose of this Document 3
   2. Scope of this Document 3
   3. Overview 3
   4. Scope of the Pericles Project 4
2. [GENERAL DESCRIPTION 5](#_TOC_250007)
   1. Product Functions 5
   2. User Characteristics 5
   3. Operating Environment 6
   4. User Problem Statement 7
   5. User Objectives 8
   6. General Constraints 8
3. [SYSTEM REQUIREMENTS 9](#_TOC_250006)

Non-Functional Requirements………………………………………………….. 9

# INTRODUCTION

* 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. 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. 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. 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.

# GENERAL DESCRIPTION

* 1. Product Functions

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

* + 1. Pericles Elections Server

The Server will be responsible for storing the settings of each election, generating passwords for authorized Voters, 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.

* + 1. 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.

* + 1. 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.

* + 1. 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.

* 1. User Characteristics
     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.

* + 1. 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.

* + 1. 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.

* + 1. 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.

* 1. Operating Environment
     1. Elections Server

The Elections Server will be written in PHP, The Server will run under windows. 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 PElection 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.

* + 1. Election Editor

The Election editor will be written in PHP,. 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.

* + 1. 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.

* 1. 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.

* 1. 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.

* 1. 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.

# 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
   1. System Attributes
      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.

* + 1. 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.

* + 1. 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.

* + 1. 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.

* + 1. 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.

* + 1. 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.

# DESIGN CONSTRAINTS

* 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.

* 1. Software and Hardware Constraints
     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
  + 1. 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.

* + 1. Election Database

The computer hosting the Election Database must have MySQL installed.

* 1. 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.

* 1. 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.

* 1. 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.

* 1. 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.