E-voting system for Ireland

17.10.2017

— **CS3500 Software Engineering**

Team L

Ács Dávid - 117106523

Gonzalo Carranza Pérez-Tinao - 117106527

Palcu Liana- Daniela - 117106643

Martinez de Rute Maria - 117106133

Overview

The purpose of the system will be to conduct all elections of Ireland. The system will allow citizens of the country to vote in a secure, anonymous and reliable way for the candidates of the ongoing election. The system will allow the introduction of the candidates and the rules of the election by authorized election officers.

The system will have several modules, notably the server side of the election system and the client side. Two or more instances of these modules will collaborate during the election. Please note that there can be more than one instance of the server part of the system, which must be able to communicate between them and to share the workload among them. Obviously there will be more than one client component in the system, all of which will connect to one of the servers available.

The server side will be operated by the system administrators and government officials of the system and they can set up the available choices during the election. The client side of the system will be operated by the voters of the country.

Correct functionality, availability, security and robustness are among the most important requirements of the system, since without these qualities the product is worthless in the eyes of the public, the government and practically all stakeholders of the system.

Server side of the system

This part of the system is responsible for authenticating the voter, validating and counting all the votes of the election. The server side of the system must be deployed to a physically secure location and on a private network.

This part of system is made of up multiple modules: a module which stores data, such as a database management software, the email subsystem, and another application which enables the authorized users to perform operations on the system and to view statistics, logs and perform audits. The requirements of the server side of the system are specifically tailored for stakeholders that interact with this part of the system: system administrators and government officials.

Client side of the system

The client side of the system will be a piece of software which will communicate with the server side of the system, in order to convey the choice of the user in a structured way. The client side can be any piece of software that respects the interface of the server side of the system. This will allow us to extend the system by creating new types of clients, which can easily be incorporated into the existing system. The requirements of the client side should reflect the needs of the voters regarding interaction with the client side of the system.

Functional requirements

Desktop application and mobile application

The client side of the system should be an application on desktop, like a program which you can download on the internet and then, being connected to the internet you can begin your session of voting. The application will be an executable file which you can use without an installation and you can delete afterwards. The application desktop must be available only during the voting session. The desktop application will support operating systems such as Windows, Linux and MacOS.

Also, the system can be implemented as a mobile application. By using a mobile application, the flexibility of voting increases, because 86% of Irish consumers are using smartphones according to www.businessworld.ie and it can lead to an increasing numbers of people who vote.The mobile application will have a similar interface as desktop application and will be built to be supported on platforms such as Android, IOS and Windows Phone.

The system should comply with the current constitution of the country.

The system should be based on the electoral statutes of the country. The electoral system used in Ireland is single transferable vote. This type of electoral system is designed to achieve proportional representation and preferential voting. A voter's vote is initially assigned to his or her favorite candidate, and if the candidate has already been elected or removes, all the other votes are transferred according to the voter's preferences.

Must provide a result

The system, after the voting period has expired, shall make a count of votes. The system will show the votes of each county separately and then the final votes showing the exact number of votes each political party has. All of this process would be represented with graphs and statistics where users can observe the results in detail and more visually.

Change your previous vote

The system, as the traditional voting, will only allow one vote per user. When the user's vote has been made, the server will not give the user the option to vote again. The system will allow the user to change their previous vote in case the user regrets the previous election or confused by selecting. Once the vote was taken, the system will redirect you to a page where the user can find the option to modify the vote.

Support for physical voting

The system can be installed both at home and in the voting station for those users with problems accessing the system from home either because they do not have a computer or because they prefer someone who advises them when using the new voting system. For the voting stations the user must access the system using the provided passwords and through facial recognition. The assumption about facial recognition is that each of the citizens' faces will be scanned when they create their ID, and that scan will be used in the authentication process to verify the identity of the citizen.

Authentication

Authentication is made always by id card. Citizens should have on their computer a slot to enter the id. Also, the system should has another way of authentication in complement to the id. The system should be able to do facial recognition if the person who is voting goes to an official place or to generate a personal password that is given by email before the election, in the case that the person votes from home. For that reason there is a database with the necessary information of citizens, their email or the facial recognition, which are made when their id is issued.

Email subsystem

An emailing system should be in place to provide means to quickly notify users. This email system could be used to remind users to vote, warn them about potential dangers (phishing attacks), and to send notification and confirmation to them. For example: after the user completes a voting session he/she should get an email announcing that the his/her vote has been recorded.

The need for the emailing system arises from the fact that fast, country wide, communication is needed for the complete system to work. In special cases, personal messages have to be sent to the email addresses of the voters, such as when we are confirming the registering of a vote of a user. The assumption is that every voter should supply a valid email address where they can be contacted.

Tutorials

The client side of the application must contain tutorials explaining the functionality of software. These tutorials should be simple videos or interactive demos clarifying the use of the voting system. By including these tutorials we can avoid a great deal of confusion on the voters' part. When a user starts the voting session, he/she will be presented with the possibility of taking the voting tutorial or not.

If the choice is yes, then the tutorial itself will load and appear on the screen guiding the user through the whole process by an example election. The interactive tutorial would require the user to select candidates and to submit a correct vote. The tutorial needs to be short and concise, a user must be able to complete it less than 5 minutes.

Finishing session

Network failures are inevitable, and since we cannot control the networks of the voters, we can mitigate the effect of a network failure by saving the progress of the voter in the voting session. This solution is only required in the following situation: the voter connects to the network and starts a voting session. After the user finished ordering the candidates, a network failure occurs and the vote cannot be registered in the central database. In order to not force the user to complete the voting process again, the client program will save the current status of the voting and send the result to the central database when there is an internet connection.

Database

Every county should provide its database for collecting information and centralise its data. Every database will hold the votes registered for every candidate in every election. It will be encrypted and will not be accessible directly by anybody. After the election was finished, the votes stored will be deleted and it will be kept only the results of each election to be able to make statistics after that. Every database from every county will be connected to a main database which will collect all the results. The main database will collect only the number of votes for every candidate.

Perform audits

All operations should be stored on a secure hard disk. Any vote or change should be stored in at most 1ms. You can't delete the information during election, it is only for append and read. There must be at least two hard disk in which the votes are stored to ensure the counting in case of failure.The information of the number of votes for every candidate should be refresh every second on the disk.

Non-functional requirements

Should be robust

The system should be enough robust to support all the activity in case that all citizens are connected to the application to vote. The whole country could vote at the same time. Each server must be able to serve 100 000 clients simultaneously.

The time to restart after failure should be at most 2 hours for hard problems, but for minor problems it should be at most 10 minutes.

The percentage of events causing failure should be less than 0.00001%. The probability of data corruption on failure should be less than 0.001%

Security

The security must assure that anybody couldn't break into database during the election and afterwards, until the data will be deleted. The system must store the recording data into encrypted files so that anybody could be able to read the identification information along with their votes. These files will be deleted after the election ends and all the data was centralized and sent to main centre. Security of the system will be tested through pentesting which means a professional will try to infiltrate the system and to document his/her findings.

Availability

The system must be available 99.99% of the voting time to the use, to ensure that everyone can vote when they want. The application should be available for the citizens at the moment that the period of vote starts until the moment that voting time finishes. In case that failure for unavailability occur it should be solve in less than 2 minutes to ensure that a person does not spend more than 3 minutes voting.

Reliable

The system should be reliable. The probability of unavailability should be 0.01% at most. Create fault tolerance is necessary to ensure that the application are constantly available. The failover should be an isolated process for the end user. The failover solution should guarantee that if the server or internet connection is offline, traffic is automatically re-routed to a secondary server or provider.

The mean time between failures should be at least 24 hours, the same time that the users can vote with the application.

Must handle exceptional situations

The system should handle some exceptional situations such as no current or others natural disasters. The system should store all the operations in that case.

The system has to be formed with at least two hard disks and a secondary server which will work in case that an exceptional situation occur. If one of the hard disk fail, the system has another one, in which all the votes are stored too.

Extensibility

The system must be capable to add new features. The idea is to have common interface and all the clients must implement this interface. For example, in the future all the technology will be based on virtual reality. You can vote in virtual reality, this means you can be at home with a pair of VR which will simulate an environment similar with a voting booth. To implement this, the interface should contain only what actions a voter should perform. The behaviour should be implemented separately by every client.

Easy to setup

The system would be easily deployable through a executable file, where the person in charge of the installation should only has to follow some steps without having to take much time. It would be a executable file with which the installation should be practically automatic. After installation there should be a configuration section where the person in charge of the installation can make the necessary settings as required.

Cheap to maintain and to buy

The buyers (Government) would like the system to be cheap. Should be cheaper to maintain than the traditional voting, meaning less man-hours. The total cost of the maintenance should be 60% of the maintenance of the traditional voting system currently in place. For example, it should be $100 000/month. It is a non-functional requirement since it does not describe what the system does, but describes the cost of the system maintenance.

Should be easy to use

The client side of the system should need minimal computer skills for being able to cast a vote. With the simple use of a mouse and the keyboard or the touchscreen, a user should be able to cast his/her vote in less than 10 mouse clicks or taps on the screen of the smartphone. The client side of the system should run without installation, simply by double clicking on the icon of the program in a GUI based operating system.

Fast to respond

Users want the program to react fast. The program in this context mean the client side of the program and the piece of software on the server side that the government officials use. The program should be able to receive new UI operations in 0.1 second after the user made an interaction with the program in 90% of the time. In the remaining 10% the program should react in 1 second. UI operations include, clicking or selecting something on the screen. Network operations should complete in less than 5 seconds on a stable network connection. If the program takes more than 2 seconds to complete an operation then it should announce the user with a notification or a progress bar to indicate that the system is busy.

Support for disabled people

Since the client side of the system will be used by all inhabitants of the country, we need to ensure that disabled people can use the system without special help. Persons with vision, hearing or mobility problems need to receive help from the client side application in order for them to be able to vote with ease. Disability support should include magnifier, high contrast mode, text to speech capabilities and voice recognition (only for home voting).

Contributions

We started out by meeting for the first time and discussing the requirements of this assignments, such what should be included in the document. After that we decided how to collaborate and to communicate efficiently. At the next meeting we started brainstorming ideas related to an election process and take notes of them. Everybody took equal part of this process until now. The next step was to divide and conquer, each one of us took some of the initial ideas and started expanding them, explaining them, making them SMART etc.

We incrementally reviewed each other, gave feedback to our teammates and based on those reviews and feedbacks we modified the requirements and this document. Once the contents of the document were close to being finished we started formatting the document. The final step was to write this part of the document, the contributions page.