

e-voting software

CS 312 Project

Software Requirements Specification (SRS) Document

{Anurag, Ashish, Harshavardhan, Ramkrishan, Sumesh}

1 Introduction

1.1 Purpose

The purpose of this document is to make the functional requirements of the Software Engineering project on an Electronic-voting system easy to comprehend. It also serves the purpose of making the functionality clear to the end users. The reader is expected to have prerequisite knowledge of online voting systems to be able to understand the document.

1.2 Document Conventions

- Throughout the document we refer to the authority which is organizing the elections as the EA(acronym for Election Authority).
- The term 'system' refers to the software system that is setup for the purpose of holding the elections.
- The use of the term 'election instance' refers to a particular election with its own timings, candidates, posts. The voters' database remains common across these election instances.

1.3 Intended Audience

The intended audience of this document is the potential end user. The document may also serve as a reference guide to the developers of the system.

1.4 Contact Information

1. Anurag Verma
email : anuverma@cse.iitm.ernet.in
phone : +91-9884590642
2. Ashish Gandhe
email : ashishg@cse.iitm.ernet.in
phone : +91-9840064027
3. Ramkrishan Kumar
email : ramki@cse.iitm.ernet.in
phone : +91-9884121902
4. Simhadri Harshavardhan
email : harshas@cse.iitm.ernet.in
phone : +91-9884212789
5. Sumesh Dutt Sharma
email : sumesh@cse.iitm.ernet.in
phone : +91-9840542339

1.5 References

1. IEEE Standards Description : 830-1998
2. Writing Effective Use Cases - Alistair Cockburn (1998)

2 Overall Description

2.1 Product perspective

The software product is a standalone system and not apart of a larger system. The system will be made up of two parts, one running visible directly to the administrator on the server machine and the other visible to the end users, in this case the voters, through web pages. The two users of the system, namely the voters and the election authority(EA) interact with the system in different ways. The election authority configures the whole system according to it's needs on the server where the system is running.

The voters cast their votes using the web interface provided. These votes are accepted by the system on the server.

2.2 Product Functions

On the EA side, the system can be used to create/update/delete the election details (posts, candidates, electoral rolls etc). The EA should be able to specify the different attributes it wants for posts/candidates of a particular election instance and voters. For example, one EA may want the candidate's photograph as an attribute, where as another EA may not find it necessary. Similarly, they may want one set of attributes for voters in one setting and a different one in another. For Example, in a university, the EA might be happy with just the roll numbers of each voter while an election in an association may require voters' name, phone number, address etc. After the election is set up, passwords must be generated and mailed to voters on request.

The system should also be able to run separate election instances at the same time.

From the voters perspective, the system is used to help them cast their votes and after the elections are over, allow them to view the results, which are automatically posted on the same site after the election duration is over.

2.3 User Classes and Characteristics

The users can be divided into two main classes:

- The EA :It's primary objective is to conduct fair and hassle-free elections. The EA has to be a neutral party and should not have any gain/loss from the election results. The EA invites potential candidates to file their nomination for certain posts depending on certain constraints. As explained earlier, the EA decides the classes of voters eligible to vote for a certain post. They should have adequate experience of using a computer to be able to configure the election properly.
- The Voters : The voters should have a basic knowledge of how to use a web browser and navigate through web pages. The voters should be aware that they have to keep their user-id and password confidential.

2.4 Operating Environment

The server should have Java installed on the machine, along with Java's cryptographic packages. The election server runs on a http server, that is "jsp" enabled. The browsers through which the voters access the server should have minimal support for cookies and encrypted transactions.

2.5 Design/Implementation Constraints

Even though the system enables voters to poll their vote from any terminal connected to the Internet, the voters should initially contact the election administrator's office to authenticate themselves and establish their user-ids. This constraint is imposed to ensure that only the genuine person is allowed to vote in the elections.

Also, it is assumed that only the EA has access to the server that hosts the election.

2.6 Assumptions and Dependencies

User side assumptions and dependencies

- PC (Personal Computer) or workstation with GUI.
- A web browser with support for cookies.
- Working Internet connection.

Server side assumptions and dependencies

- A web server with GUI, Java and an http server installed .

3 External Interface Requirements

3.1 User Interfaces

We have given all the use cases that are there for the system to specify the user interface. They are given below:

3.1.1 Use case 0 : Welcome screen for the Administrator

Main success scenario:

1. The EA is asked to login using his administrator password.
2. The EA is provided the option of working with voters database or the election instances.
3. The EA chooses to work with election instances. Refer to use case 4.

Extensions

- 1a. The EA enters wrong user-id or password. He is asked to re-enter the password.
- 2a. The EA chooses to work with voters' database. Refer to Use case 1.

3.1.2 Use case 1 : Creating the Voters Database

Main success scenario:

1. On start-up, the EA is presented with a screen (called the voter start screen henceforth) where he can choose to either create, modify or delete the voters database.
2. The EA chooses 'create'.
3. The system takes EA to a new screen and asks the EA to give a name for the new voter class and asks the EA to provide some identification information for the class.
4. The system provides a field where the EA can add a roll number and his attributes by clicking add, in addition to a facility to load entries from a file.
5. The EA confirms the entries it has made. The system registers the entry and presents the EA with welcome screens of Use case 0.

Extensions

- 2a. The EA chooses to modify a voter instance. Refer to use-case 2.
- 2b. The EA chooses to delete the voters' database. Refer to use-case 4.
- 2c. The EA chooses to create a new voter class for the first time ever, the EA is redirected to a new screen where he is required to specify the attributes of the voters that are needed to be stored in the database along with the primary key among them. The system then takes the EA to step 3 of the main success scenario.
- 3a. The class name specified already exists. The system asks the EA to enter a new name, go back to step 5.
- 4a. Improper attribute format. Signal an error and go back to the same screen without any action.
- 4b. The entered entity shares the same primary key with an existing entity, but the other attributes differ in their values. The system throws up an error displaying the clashing attributes and asks the EA to reenter the entity.
- 4c. Improper file format or file read error. Signal an error and go back to the same screen without any action.
- 4d. Clash in the attributes of an entity in the file with that in the database. The system throws an error indicating the entity and its attributes that went wrong. EA is shown the same screen as above.

5a. The EA chooses to Exit. The system takes it back to the start screen.

3.1.3 Use case 2 : Modify the voters databases

Main success scenario:

1. The system presents the EA with the search screen. The EA selects the attributes he wants to search for and specifies their value.
2. The system presents the EA with the results. The EA chooses the voter(s) to modify.
3. The EA either modifies the details or deletes the voter(s).
4. The system asks the EA to confirm the changes. The EA confirms.
5. The system presents the EA with the search screen of step 1.

Extensions:

4a. The EA cancels without confirming. The system does nothing and takes the EA back to the search screen of step 1.

3.1.4 Use case 3 : Delete the Voters' database

Main success scenario:

1. The system asks the EA to confirm his wish to delete the voters' database for ever.
2. The EA confirms it.
3. The database is deleted.

Extensions

2a. The EA cancels without confirming the deletion. The system does not perform any action.

3.1.5 Use case 4 : Creating the Election Instance

Main success scenario:

1. The EA is presented with a screen (called the start screen henceforth) where he can create, modify or delete an election instance.
2. The EA chooses to create an election.
3. The system asks the EA whether a candidate can stand for more than one posts.
4. The system asks the EA the number of posts. For each of the posts, steps 6-8 are repeated.

5. For a post, the system asks the EA the designation and other required attributes of the candidates competing for that post.
6. For the post, the system asks the EA for the class of voters that are allowed to vote for the post.
7. For the post, the EA specifies the eligible nominated candidates. The system asks the EA to fill in the required attributes.
8. The system asks the EA for the start and finish time of the Election. The EA provides the necessary details.
9. The system asks the EA to confirm his election instance.
10. The system creates an election instance.

Extensions

- 1a. The EA enters wrong user-id or password. He is asked to re-enter the password.
- 3a. The EA chooses to modify an election instance. Refer to use-case 5.
- 3b. The EA chooses to delete an election instance. Refer to use-case 8.
- 3c. The election instance already exists. The system notifies the EA about this.
- 5a. The EA enters the number of posts as zero. The system asks him to re-enter the number of posts.
- 6a. The designation of two posts is the same. The System asks the EA to delete/modify one of them.
- 7a. The EA does not specify any of the classes of voters who can vote for a post. The EA is directed to specify at least one class.
- 8a. The key attributes of the candidate are repeated. The system does not allow a candidate to contest for the same post twice.
- 8b. A candidate is nominated for more than one post. The system acts according to the response to 4. If required, the system notifies the EA and asks it to modify/delete one of them.
- 9a. The values given by EA are improper. The system asks the EA to re-enter the values.
- 10a. The EA quits without confirming. The system does not perform any action.

3.1.6 Use case 5 : Modify the Posts

Main success scenario:

1. The system presents the user with the update options, namely Posts,Candidates,Voters and time of election.
2. The EA chooses to update 'posts'.
3. The system presents the EA with the list of all posts and an option to add a post.
4. The EA chooses the post it wants to modify.
5. The EA either modifies the details or deletes the post.
6. The system presents the EA with the screen of step 4.

Extensions

- 2a. The EA chooses to update the candidates. refer to use case 6.
- 2b. The EA chooses to update the time of elections. Refer to use case 7.
- 4a. The EA chooses to add another post. Refer to Use case 4, step 6 onwards.

3.1.7 Use case 6 : Modify the Candidates

Main success scenario:

1. The system presents the user with the update options, namely Posts,Candidates,Voters and time of election.
2. The EA chooses to update 'candidates'.
3. The system presents the EA with the list of all posts.
4. The EA chooses the post it wants to modify the candidates of.
5. The system presents the EA with the list of all candidates in that post and an option to add a candidate.
6. The EA chooses the candidate to modify.
7. The EA either modifies the details or deletes the candidate.
8. The system presents the EA with the screen of step 4.

Extensions

- 2a. The EA chooses to update the posts. refer to use case 5.
- 2b. The EA chooses to update the time of elections. Refer to use case 7.
- 5a. The EA chooses to add another candidate. Refer to Use case 4, step 7 onwards.

3.1.8 Use case 7 : Modify the Election Time

Main success scenario:

1. The system presents the current election time and asks the EA to reset and confirm.
2. The system presents the EA with the screen of step 4.

Extensions

- 1a. The EA exits without confirming. The system does not do anything.

3.1.9 Use case 8 : Deletion of the election instance

Main success scenario:

1. The EA chooses to delete the election instance.
2. The system asks the EA to confirm the request.
3. The EA confirms the request.
4. The system deletes all the informations about the election instance and is now ready for configuration of a new election instance.

Extensions:

- 1a. The election instance does not exist. The system notifies the EA about this.
- 3a. EA does not confirm the request. The system takes the EA back to the start screen.

3.1.10 Use case 9 : The Voting on the Voter's end

Main success scenario:

1. The voter is asked to login using the user-id and password provided to him earlier. Steps 2 and 3 are repeated for all the posts that this voter is available to vote for.
2. The system presents the voter with successive screens for voting for each of the posts.
3. The voter selects one of the candidates and submits his choice to the system.
4. The system presents the voter with the final choices of the candidates for each post made by the voter for confirmation.
5. The voter confirms the choices.
6. The system registers the choices made by the voter.
7. The voter logs out.

Extensions:

- 1a. The voter enters wrong user-id or password. He is asked to re-enter the password.
- 1b. The voter has already voted once. The system does not allow him to log in again.
- 3a. The voter does not select any candidate for a post and attempts to submit. The system asks him to select one of the candidates.
- 3b. The voter wishes to reconsider his choices. The system allows to re-vote for the posts he is not content with.

System crashes or power failure at any point:

- a. The voter has not confirmed. He has to re-login and vote again.
- b. The voter has not confirmed his vote. He need not login again to vote.

3.2 Hardware Interfaces

There are no hardware interfaces to this software system. The only interfaces are through a computer system.

3.3 Software Interfaces

The poll server runs on http server that is enabled to handle server pages (eg. Apache Tomcat for support jsp). It uses a relational database to keep track of the polls, which it connects through standard database connectivity interfaces. In order to run the setup software, the environment needs to have a JVM running on it.

3.4 Communications Protocols and Interfaces

- For the purpose of giving the EA an option of sending the voters their passwords through mail, the system should use the SMTP protocol.
- The system should also use standard protocols for secure transactions between the Voter and the system through the internet.

4 Non Functional Requirements

4.1 Performance Requirements

The software is expected to have reasonably short response time. It should be able to log-in and feed the voter with new pages on request with a response time of the order

of a few seconds. The system is expected to serve a maximum of up to 100 voters at any point of time, each voter being active for about a couple of minutes and requiring about 10 pages (depending on the number of posts to be voted for) in all. This would imply that the system should have the capability for processing about 20 transactions each second.

4.2 Safety Requirements

- * In order to prevent data loss in case of system failure, the result of votes that were polled till then have to be saved in the database, for the system to resume the counting process on reboot.
- * The EA should set up his system time appropriately for the election process to start at the correct time.
- * In case the EA detects any security lapse in the system, he should be able to shut down the server and close all connections immediately while preserving the already polled votes.
- * The system should be capable of gracefully recovering from earlier crashes and continuing the voting process.

4.3 Security Requirements

- * The system should provide basic security features like password authentication and encrypted transactions.
- * All the passwords generated and communicated to the users should be stored in the server only in an encrypted form for login management to prevent misuse.
- * Serial attacks should be avoided by maintaining a minimum time gap between successive invalid log-in attempts.
- * Additional security features like voter anonymity and threshold schemes for multiple EAs might be provided later on as an add-on feature to the software.

4.4 Project Documentation

A detailed design document should be presented which should describe in detail, the classes, interfaces and control flow involved in detail. The final code would contain a complete description of classes, their methods, their inputs and outputs.

4.5 User Documentation

Documentation for EA

A step by step cross-referenced tutorial like manual should be provided for the EA in order to help him set-up a server and the e-voting system on it. All features and GUI interface details should also be clearly provided in the document

Documentation for voters

The user interface (through browser) is easy enough to use even for a lay user. But minimal instructions may be provided at the bottom of each web-page as an aid for the un-introduced.