Software Requirements Specification

for

Fingerprint Voting System

Version 1.0 approved

Prepared by Gamzenur Dorgut Gülten Şevval Erdal Barış Yaşar Fatih Sultan Mehmet Çelik

TEDU Department of Computer Engineering

12.12.2021

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 2

1.5 References 2

2. Overall Description 3

2.1 Product Perspective 3

2.2 Product Functions 3

2.3 User Classes and Characteristics 3

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

3. External Interface Requirements 5

3.1 User Interfaces 5

3.2 Hardware Interfaces 7

3.3 Software Interfaces 7

3.4 Communications Interfaces 7

4. System Features 7

4.1 System Feature 1 7

4.2 System Feature 2 (and so on) 8

5. Other Nonfunctional Requirements 12

5.1 Performance Requirements 12

5.2 Safety Requirements 13

5.3 Security Requirements 13

5.4 Software Quality Attributes 14

5.5 Business Rules 17

6. Other Requirements 17

Appendix A: Glossary 18

Appendix B: Analysis Models 18

Appendix C: To Be Determined List 19

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Gamzenur Dorgut  Gülten Şevval Erdal  Barış Yaşar  Fatih Sultan Mehmet Çelik | 12.12.2021 | First version of the Software Requirements Specification document and the Fingerprint Voting System. | Fingerprint Voting System 1.0 |

# Introduction

## Purpose

The Fingerprint Voting System's Software Requirements Specification (SRS) will offer a complete explanation of the system's requirements. This SRS will provide a decent comprehension of what to anticipate from the system that will be built. The right software will be built for the final client and will be utilized for the progress of the project's future phases due to a clear perception of the system and its performance. This SRS will serve as the project's basis. The Fingerprint Voting System may be planned, built, and eventually tested using this SRS.

The software development team building the Fingerprint Voting System will utilize this SRS, as will the client which is the government. The project team will utilize the SRS to thoroughly comprehend the Fingerprint Voting System's requirements so that the proper software can be built. End consumers of the voting system will be able to practice this SRS as a "test" to evaluate if the construction team is meeting their expectations. If it does not meet their expectations, the end users can explain why, and the team will modify the SRS to meet their requirements.

## Document Conventions

The document is was created with Microsoft Word 2013 with the font type 'Times New Roman'. The fixed font size that has used to type this document is 12pt with 1.5 line spacing. The document's headings have been set using the bold feature. Alistair Cockburn's template is used to write the use case scenarios. The UML diagrams were established in  UML 2.0 standards. The standard IEEE template is used to organize the presentation and circulation of the document.

## Intended Audience and Reading Suggestions

The government and citizens who are old enough to vote, as well as the development team, are the target audience for this document, with the goal of referring to and analyzing the material. The SRS document can be utilized in any situation regarding to the project's requirements and the strategies that have been executed. Finally, the paper will give a clear picture of the system that is being developed.

The following is a brief summary of the document:

1. Overall Description

2. System Features

3. External Interface Requirements

4. Non Functional Requirements

## Product Scope

The government will install the latest software, Fingerprint Voting System, which will automate the key processes of the voting process. Checking the legitimacy of the vote is to make sure the vote is valid. Seeing distribution of the votes according to the region is to observe which part of the country mostly voted for which party. There are three actors on the Fingerprint Voting System. They are, Admin, User and Polling clerk. None of the actors can access all of the system functionalities, each of them has their restrictions to provide a safe and fair voting environment.

Objectives of Fingerprint Voting System are in order to ensure a peaceful, easy, safe and equal voting process to citizens. The development team will show various benefits with this system, including reliability, simplicity, and, most crucially, the efficiency of data analysis. The system should be designed user-friendly, simple to work with, quick to recover from faults, and deliver high personal contentedness to the end user.

## References

1. <https://senior.ceng.metu.edu.tr/2011/iteam4/documents/srs-iTeam4.pdf>

We used the example above to check whether our progress was in the right direction.

2. CMPE 313 SRS TemplateFile

We used this file to follow the SRS structure.

3. <https://www.youtube.com/watch?v=pCK6prSq8aw>

We followed this video to create the sequence diagrams.

# Overall Description

## Product Perspective

Fingerprint Voting System is a new software application created to replace the existing traditional, pen and paper voting systems in the governmental elections, in order to get election results faster and safer while also reducing paper waste. This newly introduced system will provide an easy election process for all citizens while being reliable. It will have a user friendly interface where the voters can vote using only their fingerprints. Since the system will be using the government database for the fingerprint matching it will be more secure and reliable than the traditional voting systems. Final outcome of this project will shorten the governmental election processes and provide fairer result.

## Product Functions

* Register Possible Candidates
* Provide List of Voters
* Set Bounding Number of Votes
* View the Results
* View the List of Faulty Votes
* Voting
* Approve The Vote
* Check ID
* Check Tally
* Check The Legitimacy of The Vote

## User Classes and Characteristics

There are three main user levels of the Fingerprint Voting System:

1. System Administrator
2. Polling Clerk
3. Voter

Administrator:

System administrator is responsible from the management and set up of the system. They provide the data of voters from a governmental database. They also control the list of possible candidates in the system. As the main authorized actor, they have access to the results of the voting. In addition to that, they can also view the list of people who tried to give a faulty vote, whether they tried to vote more than once or tried to vote as someone else. These users should be highly knowledgeable about the system. They should know every detail about it in order to be able to use it sufficiently and not cause any mistakes. They should also be very well-informed about computer technologies, especially databases.

Polling Clerk:

Polling Clerk is firstly responsible with checking whether the Voters ID is legit. They are also responsible with checking if the voter’s vote is approved by the system and added to the tally. These users should be knowledgeable about the system and how it works. They should be able to detect if the vote has not been added to the tally.

Voter:

Voter is responsible from the voting. They should be able to vote using the user interface and approve their vote. They do not need to have any qualifications in order to use the system, being able to use your finger to vote is enough.

## Operating Environment

Hardware and Software Requirements

Hardware:

1. Operating System: Supports all known operating systems such as Linux, Windows, etc.
2. Computer: 1GB+ RAM, touch screen monitor with minimum resolution of 1024x768, keyboard, mouse.
3. Hard Drive: Should have minimum of 15 GB of free space.

Software:

1. Software is designed to work on any platform above Microsoft Windows 7 (32bit).
2. Microsoft .NET Frameworks 4.0 or above.
3. MySQL 8.0.

## Design and Implementation Constraints

Since Fingerprint Voting System is based on taking fingerprint readings as input, the system needs a touch screen monitor to be able to process the fingerprint. System can run on memory of 1GB, but it is recommended to have more than this minimum requirement. Also, since the system is dependent on the government database for fingerprint matching it is important to implement the relationship between the two systems securely.

## User Documentation

* Admin Manual: Provides an in depth knowledge about the software system. It contains all the information that an admin should know before interacting with the system.
* Clerk Manual: Provides a simple guideline for polling clerks to use, containing surface information about how the system should behave in which situations.
* User Tutorial Video: A very short, animated video for the voters, explaining the process of voting in a simple manner.

## Assumptions and Dependencies

With the hardware requirements and the software requirements already being on the expensive side, the end product will most likely be high cost which the client already agreed upon. Our team assumes that the client will provide devices with Windows 8 and higher, therefore not creating any problems. The product is dependent on government databases, therefore implemented according to the current database system that is being used by the client. If the client changes the database system that they are using the software and the SRS documentation should be altered accordingly.

# External Interface Requirements

## User Interfaces

### Administrator Interface

Administrator will open the system. The main page of the system appears. In this page there are main functions on the top side of the screen. These functions are:

* Add candidate: When clicked, a new page with a list of citizens will appear and admin can select the candidates from that list. There will be a search bar to make it easy to find candidates in that list. When a candidate selected, they will be added to already existing “Candidate List”.
* Provide List of Voters: This function communicates with the citizen database and brings the eligible voters list to the screen. Then there will be button to add these people into voting database.
* View the results: This button brings the information of the election in a new page. If the election is still in progress, a warning will appear on the page about ongoing election.
* View Faulty Vote List: This button shows the list of faulty votes. In this list, there will be the ID and the location of the voter will be displayed. This button will become active after the election is over.

### Stimulus / Response Sequences

In this interface, users will vote for their choice of candidate. A page where all the candidates with empty boxes underneath their names and pictures open on the system. Users will scan their fingerprint under the screen which they want to vote. An X shape appears on the box that the user pressed. User then presses the ‘Approve My Vote’ button if they are sure of their choice to approve their vote on the system. If the vote is legit, system will send the information of the vote to database.

### Functional Requirements

In this interface, polling clerk needs to log in into the system. In the main page there will be ID and password section which clerk uses to log in. If it is not valid, an error message in a box will appear. If the info is legit, polling clerk will enter the system and encounter with a page which contains the list of citizens who is assigned to that ballot box. Clerk checks the ID information of the citizens who came to vote and compares it with the information in the system. If they match, clerk clicks the box on the side of the citizen to mark them as present in the election. Another function in this page is after checking the of the citizen letting them into voting area, system waits for the vote input. If vote is legit, a message that says “Vote is legit” will appear on the screen. If citizen cannot vote after 5 tries, a message will appear on the screen to warn polling clerk.

## Hardware Interfaces

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

## Software Interfaces

The poll server runs on a local server and then it sends all the data to main server so mini servers can handle this workload. 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 Java Virtual Machine running on it.

## Communications Interfaces

Every device that will be used with the interface needs to have a stable Internet connection since it needs to interact with a database. The system should use HTTP communication standard.

# System Features

## Register Candidates

**4.1.1 Description and Priority**

This feature is for registering the names of the candidates into the Fingerprint Voting System. Admin will register the name and background information about the candidates into the system. High priority.

**4.1.2 Stimulus / Response Sequences**

* Admin opens the system.
* Admin clicks on the ‘Add Candidate’ button.
* Manually registers the name of the candidate and chooses the right candidate from the citizen list provided from the database.
* The chosen citizen is added to the list of the candidates in the system.

**4.1.3 Functional Requirements**

REQ1.1: There should be an ‘Add Candidate’ button.

REQ1.2: There should be a search bar where the admin can search for a candidate by their name.

REQ1.3: The chosen candidate should be added to an already existing ‘Candidate List’ on the system.

## Provide List of Voters

**4.2.1 Description and Priority**

This feature is for getting the list of voters from the database and entering them into the Fingerprint Voting System. Admin will register the ID information about the voters onto the system from the database. High Priority.

**4.2.2 Stimulus / Response Sequences**

* Admin opens the system.
* Connects into the citizen database.
* Clicks on the ‘List Eligible Voters’ button.
* Saves the given list into the system by clicking the ‘Save’ button.

**4.2.3 Functional Requirements**

REQ2.1: The system should be able to connect with the database.

REQ2.2: There should be a ‘List Eligible Voters’ button so that the system should be able to list the eligible voters, checking their age status.

REQ2.3: There should be a ‘Save‘ button for the admin to be ab le to save the list of the voters for the election into the system.

## View the Results

**4.3.1 Description and Priority**

This feature is for viewing the result of the election. The admin will be able to view the result of the election in numbers of fingerprints and percentages. It will give a warning if the election is still in the process. High priority.

**4.3.2 Stimulus / Response Sequences**

* Admin opens the system.
* Admin clicks on the ‘See Results’ button.
* A page opens, displaying the results of the election in different forms.

**4.3.3 Functional Requirements**

REQ3.1: There should be an ‘See Results’ button.

REQ3.2: If the election process is over See Results button should lead the user to the result page.

REQ3.3: If the election is still going on, the system should display a warning message on the screen.

REQ3.4: The result page should display the results in both numbers and percentages, distributed between the candidates.

**4.4 Viewing the List of Faulty Votes**

**4.4.1 Description and Priority**

This feature is for viewing the list of faulty votes that have been used in the election. The admin will be able to view list of the citizens who either tried to vote twice or tried to vote for someone else. Admin can only see this list after the election is over. Low priority.

**4.4.2 Stimulus / Response Sequences**

* Admin opens the system.
* Admin clicks on the ‘Faulty Votes’ button.
* A page opens, displaying the list of citizens who gave a faulty vote and their ID information.

**4.4.3 Functional Requirements**

REQ4.1: There should be an ‘Faulty Votes’ button.

REQ4.2: If the election process is over ‘Faulty Votes’ button should lead the user to the faulty votes list.

REQ4.3: If the election is still going on, the system should display a warning message on the screen.

REQ4.4: The faulty votes list should display the ID info of the voter and location that the tried to give a faulty vote.

**4.5 Voting**

**4.5.1 Description and Priority**

This feature is for users to vote. The user will give a vote to a candidate of their choice by pressing their fingers to a box underneath the said candidate. High Priority.

**4.5.2 Stimulus / Response Sequences**

* A page where all the candidates with empty boxes underneath their names and pictures opens up on the system.
* User presses their finger on one of these boxes to indicate their choice of a candidate.
* An X shape appears on the box that the user pressed.
* User then presses the ‘Approve My Vote’ button if they are sure of their choice to approve their vote on the system.
* If the vote is legit, tally is incremented.

**4.5.3 Functional Requirements**

REQ5.1: There should be a Candidates page where the user can see all the candidates clearly.

REQ5.2: There should be empty boxes underneath the candidates’ info for the user to press on.

REQ5.3: The box that the user pressed on should display an X, indicating the voted candidate.

REQ5.4: There should be an ‘Approve My Vote’ button for users to lock their vote on the system.

REQ5.5: When clicked ‘Approve My Vote’ button, the vote should be registered in the system if the fingerprint matches.

REQ5.6: The tally should be incremented.

**4.6 ID Checking**

**4.6.1 Description and Priority**

This feature is for polling clerks to view the ID information of the voters which was assigned to them. The polling clerk checks if the ID information of the citizen matches with the one on the system before accepting them into voting area. If the information matches the Clerk marks citizen as present on the system. Medium priority.

**4.6.2 Stimulus / Response Sequences**

* Polling Clerk opens the system.
* A list of citizens that are assigned to their ballot box is displayed on the system.
* Clerk checks the ID of the citizen manually to determine whether if it matches the one on the system.
* If the ID info matches, the clerk clicks the empty box beside the citizen’s name to indicate they were present in the election.
* When clicked, the empty box displays a check mark.

**4.6.3 Functional Requirements**

REQ6.1: There should be a unique list of voters for each Polling Clerk.

REQ6.2: There should be a clickable empty slot besides every citizen’s name.

REQ6.3: The empty slot should display a check mark when clicked.

**4.7 Checking the Legitimacy of The Vote**

**4.7.1 Description and Priority**

This feature is for polling clerks to check whether the users vote is legit (acceptable) or not. After the polling clerk checks for the ID of the voter and lets them in, the voter will vote. In this process if the vote matches with their fingerprint in the database and wasn’t used before, system will let the clerk know that the vote is legit. If not system will display a warning on the clerk’s monitor. High priority.

**4.7.2 Stimulus / Response Sequences**

* After the polling clerk marks the voter as present, system waits for the vote.
* When the user votes, system automatically checks whether the fingerprint matches the one on the database for this particular database.
* If the fingerprint matches, displays a message on the clerk’s monitor that the vote is legit or acceptable.
* If the fingerprint does not match after 5 tries, the system will display a warning message on the clerk’s monitor.

**4.7.3 Functional Requirements**

REQ7.1: The system should wait for a vote when the citizen is marked as present.

REQ7.2: System should be able to decide if the fingerprint is acceptable.

REQ7.3: System should display a message on the clerk’s monitor according to the state of the vote.

**4.8 Tally Checking**

**4.8.1 Description and Priority**

This feature is for polling clerks to check whether the vote of the citizen added to the tally, the total number of votes used in that particular device. When it is confirmed that the vote is legitimate, the system will increase the tally count. Low Priority.

**4.8.2 Stimulus / Response Sequences**

* After system confirms that the vote is acceptable and displays the message on the clerk’s monitor, it increments the tally value by one.
* Displays the updated tally count on clerk’s monitor.

**4.8.3 Functional Requirements**

REQ8.1: There should be a tally value that the system holds to record the number of accepted votes.

REQ8.2: The system should upgrade this tally value every time a new vote is registered in the system.

REQ8.3: The number should be displayed on the polling clerk’s monitor.

# Other Nonfunctional Requirements

## Performance Requirements

There are many users in the system. In order to speed up the voting process, the fingerprint scanner used in the voting device should work properly and the database matching the users should yield fast results.

**5.1.1 Fingerprint Scanner**

The scanner scans the entire finger and distinguishes similar fingerprints. The scanner is kept clean in multiple uses. This reduces the possibility of mismatching fingerprints.

**5.1.2 Database**

The database, which will choose among millions of people, should be designed considering the number of users. Results from fingerprint matches should return fast.

## Safety Requirements

Important safety elements in the system are confirmation of fingerprint, using the vote, confirming the vote and counting the vote in the right place. Confirmation and error pages to be applied at each step prevent these elements. Thus, the elements that will damage the system or cause loss in the system are prevented.

## Security Requirements

**5.3.1 Polling Clerk – ID Checking**

The identity checks performed provide the right number of protection with the right purpose.

**5.3.2 Encrypted Votes**

Who voted for whom is encrypted. The user only knows his own vote. Thus, confidentiality is maintained.

**5.3.3 Secure Connection**

The votes cast are end-to-end encrypted and protected against cyber-attacks. Thus, security is ensured.

**5.3.4 Prevention of SQL Injection**

In order to keep the votes safe, the database is designed in such a way that SQL is not used except for system requirements.

## Software Quality Attributes

**5.4.1 Adaptability**

*System is a government project. Therefore, it works in a variety of environments.*

**5.4.2 Availability**

The system makes an appropriate return when considering errors caused by user errors or caused by the system.

**5.4.3 Correctness**

Considering the problems that may occur in voting, errors are determined and the system proceeds correctly.

**5.4.4 Flexibility**

The system ensures that the users understand the system well with the directions. Directions made at each step make the system simple to use.

**5.4.5 Interoperability**

The system is designed to receive and export data from the government. Data is used implicitly where specified in the requirements. It does not transfer data to third parties.

**5.4.6 Maintainability**

The system supports change. It is implemented according to the project development method that is decided after the requirements are determined.

**5.4.7 Portability**

The system is prepared in portable sizes according to the size of the data to be used and the desired speed of the operations to be performed. The design of the database and system is designed to operate at maximum capacity in low-capacity environments.

**5.4.8 Reliability**

The system continues to operate under predetermined conditions without interruption.

**5.4.9 Reusability**

The system is designed for long-term use. Therefore, in the implementation part, the codes are written as non-repeatable and reusable.

**5.4.10 Robustness**

The system is ready for different actions based on best practices and as determined by the requirements.

**5.4.11 Testability**

The system is developed according to tests.

**5.4.12 Usability**

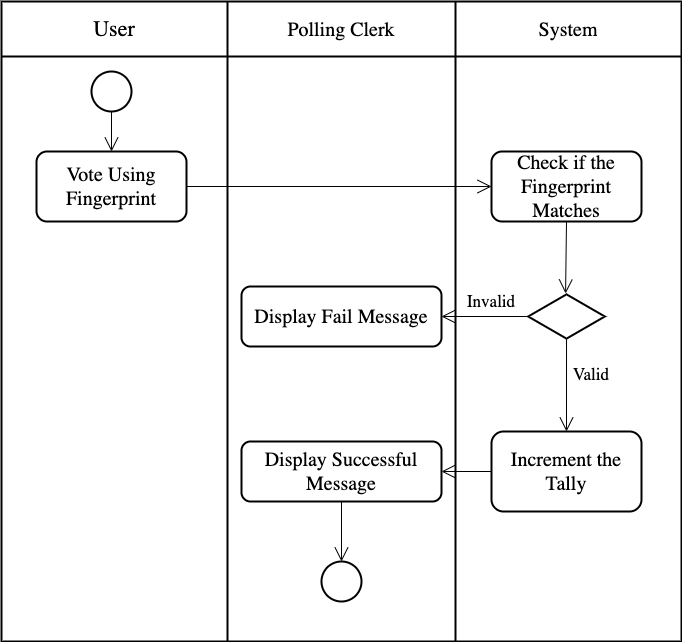
Since various users will use the system, the user interface is designed to be simple and understandable. Voters and parties are clearly indicated. Confirmation, finger press processes. Detailed and simple confirmation and error screens are used to increase interaction with the user.

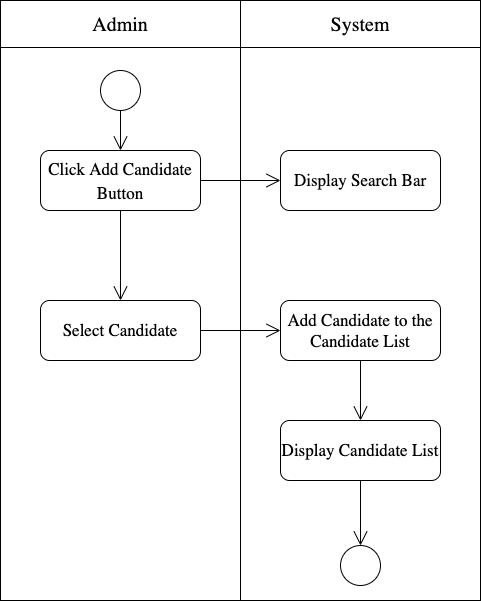
## Business Rules

The system depends on three people. These are system admin, polling clerk, and user. Admin is responsible for making the system usable. It identifies candidates and voters. Also admin creates database for Fingerprint match. Polling clerk provides security and provides system flow by solving problems outside the system. Identity checking is one of the polling clerk's responsibility. It also checks whether the votes are valid. User, on the other hand, uses the system to give the game and approves his/her vote.

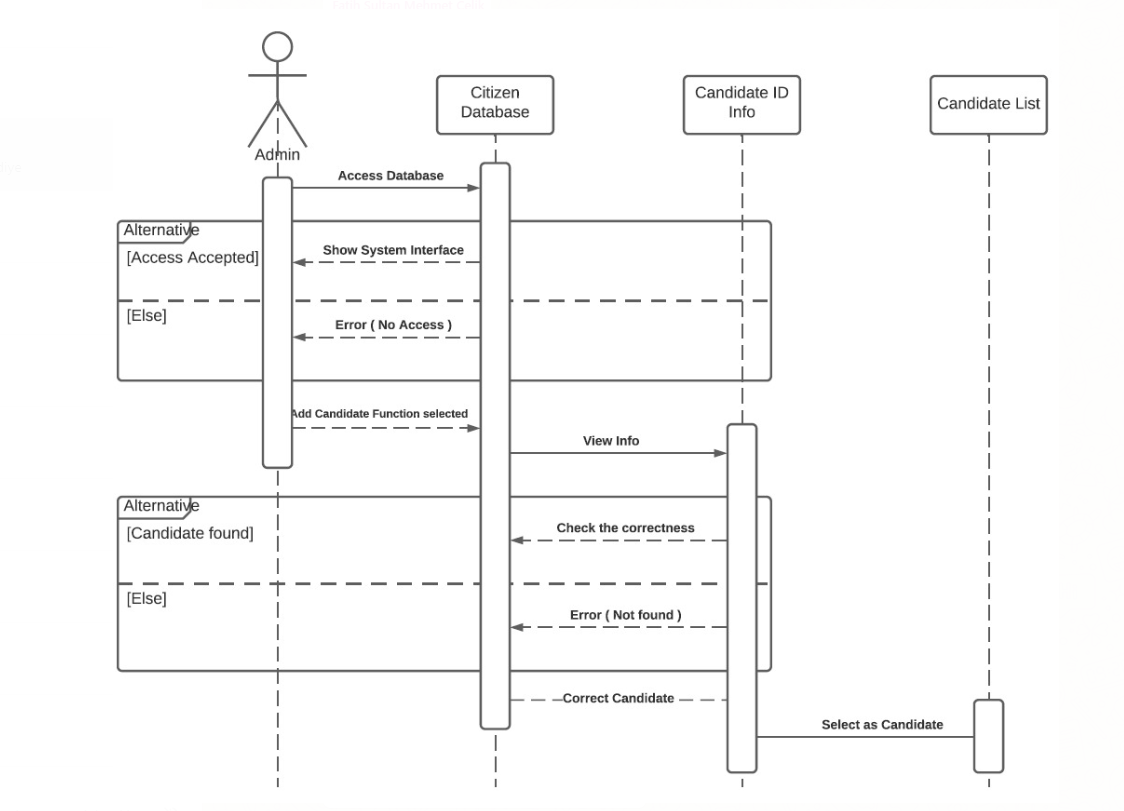
# Other Requirements

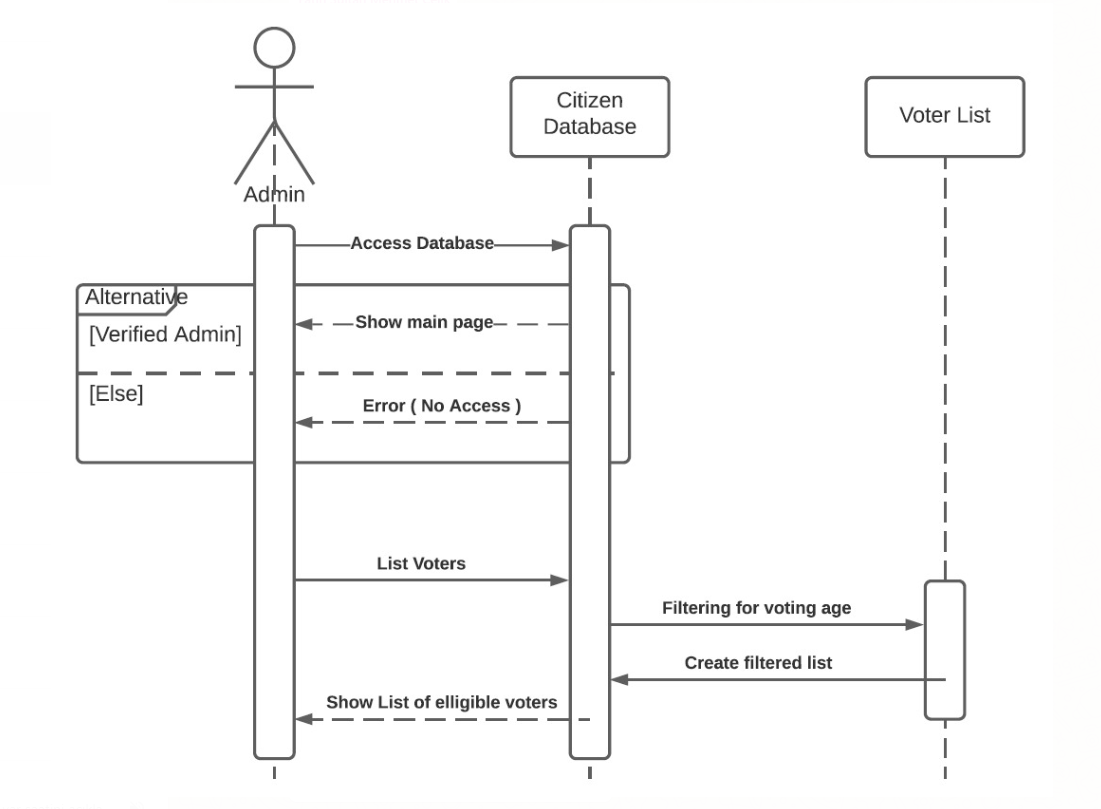
ACTIVITY DIAGRAMS:

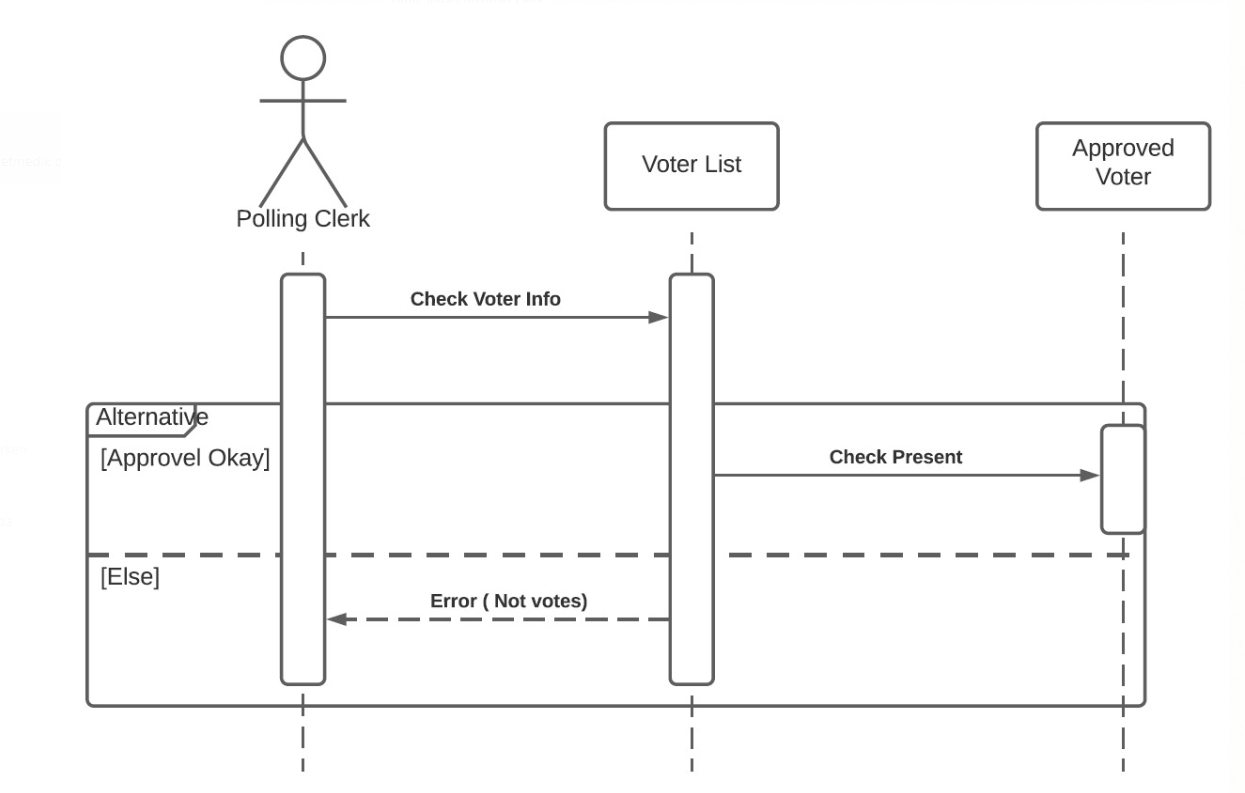


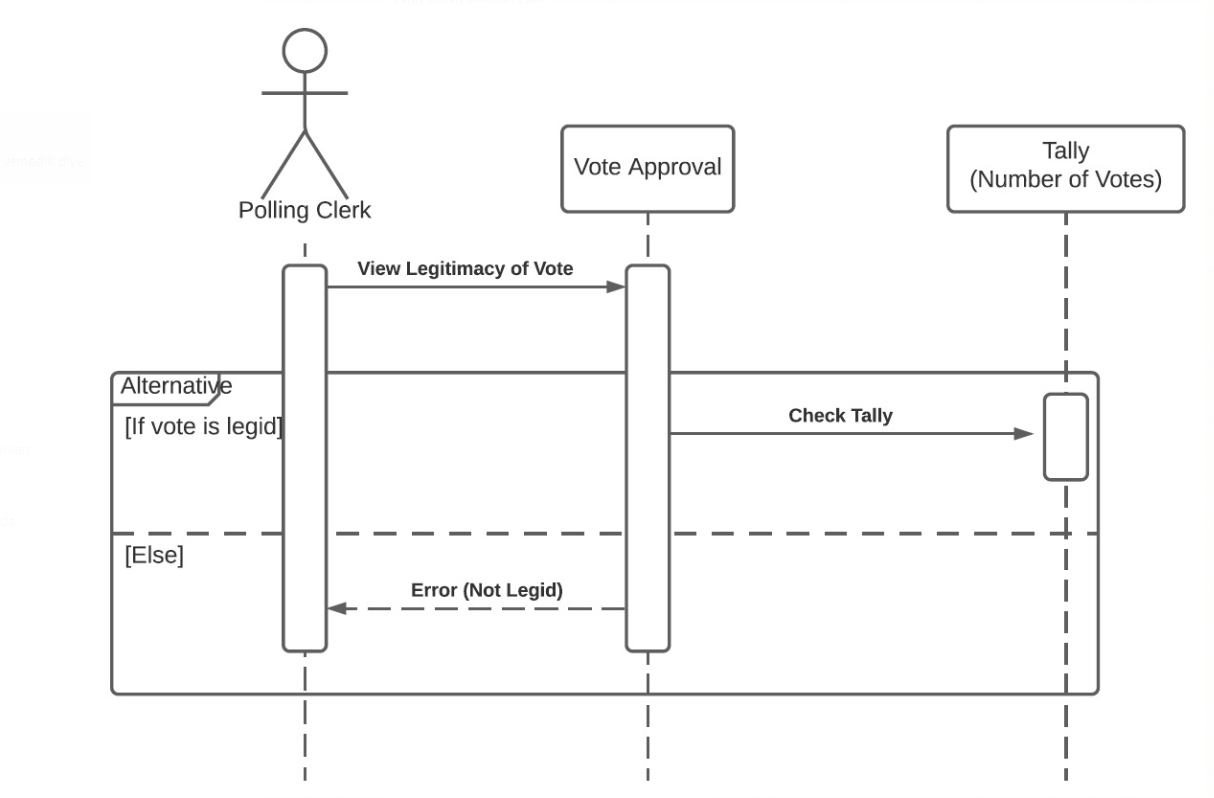


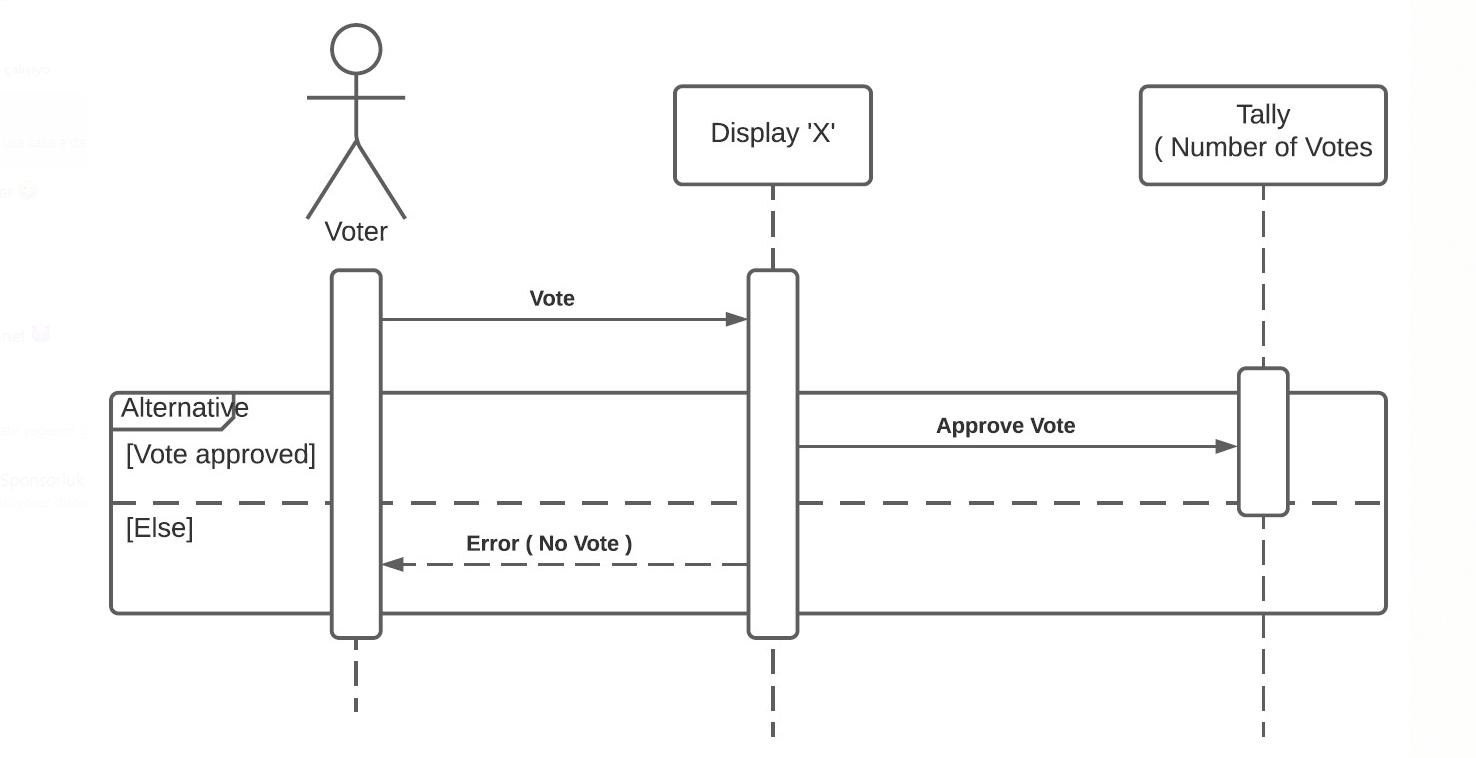
SEQUENCE DIAGRAMS:



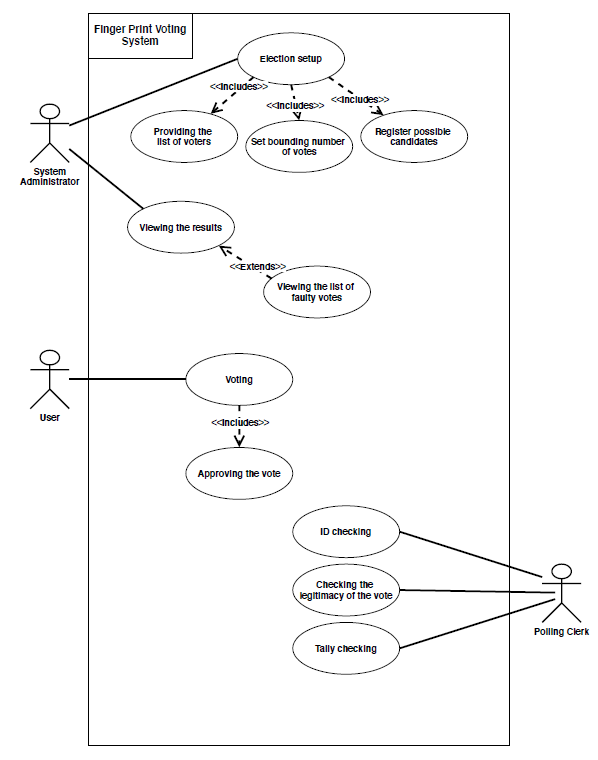








USE CASE DIAGRAM:



Appendix A: Glossary

**Tally:** Number of votes that was given using that particular device.

**Ballot Device:** It is like a traditional ballot box but it is a device that voters use to vote.

**Polling Clerk:** A person responsible from a polling device in the election process.