**Software Requirements Specification**

**for**

Electronic Voting System

**Prepared by**

| **Mohammad Khashashneh** |  | **2018902125** |
| --- | --- | --- |
| **Aseel Haddad** |  | **2018902126** |
|  |  |  |
|  |  |  |
|  |  |  |

| **Instructor:** | **Dr.Anas Al Sabeh** |
| --- | --- |
| **Course:** | **CIS 499** |

**Contents**

[**Introduction**](#_2et92p0) **1**

[Document Purpose](#_xio2kdkqeqhh) 1

[Product Scope](#_jynjwsml05jt) 1

[Intended Audience and Document Overview](#_8341gwlroof9) 1

[Definitions, Acronyms and Abbreviations](#_2lyi1fo89pc1) 1

[References and Acknowledgments](#_kabqdlvhrjgh) 2

[**Overall Description**](#_wvrda3lr0oi6) **3**

[Product Overview](#_26in1rg) 3

[Product Functionality](#_vdgf0453wx52) 3

[Design and Implementation Constraints](#_u07urdkyxm9j) 3

[Assumptions and Dependencies](#_xpatwau6rc4i) 3

[**Specific Requirements**](#_6j6vs3ngkkhv) **4**

[Functional Requirements](#_8hq7z02t5iql) 4

[Use Case Model](#_mj45pouvgwl1) 5

[Use Case #1 Voting](#_yr3z6pfz5424) 5

[Use Case #2 Voter Verification](#_buynjve65ql9) 6

[Use Case #3 Cast Vote](#_4t63n8jiutfm) 6

[Use Case #4 Modifying/Reviewing Vote](#_ymiodyrhx0s0) 6

[Use Case #5 Confirm Votes](#_hwsuas9i9x1w) 7

[Use Case #6 Build voting](#_maozb7vdfzaw) 7

[Use Case #7 View Results](#_w8mm49ivtsf9) 7

[Use Case #8 Create a Ballot Item](#_8texus9mxi8b) 8

[Use Case #9 Enter Voters Information](#_dpibcmppy8uz) 8

[Use Case #10 Verify Result Integrity](#_bo87qd7j97qf) 8

[Use Case #11 Send Reports](#_wmc2sdsu6txi) 9

[**Logical Design (Relational Schema)**](#_8i7w9538szxw) **10**

[**Conceptual Design (ER Diagram)**](#_guifapoqdd3g) **10**

# **Introduction**

## **Document Purpose**

*Online Voting System is a system which enables all citizens to cast their vote online. The purpose is to increase the voting percentage across the country; In traditional voting people must visit the booth to cast their vote and This costs a lot of time, effort and money and contributes to the spread of diseases and epidemics. So due to this, the voting percentage across the country is very low. Through this software, citizens will be able to vote in their homes through their phones, thus avoiding gatherings and mixing, especially during the epidemic.*

## **Product Scope**

*The software produced will be an online voting system. The main objective of this software is to increase the overall voting percent. It will maintain the database of all the eligible citizens and candidates. It will manage all the account details of the voters such as citizen name, date of birth, their constituency area, governorates, login id and password of the voter from one central location.*

## **Intended Audience and Document Overview**

*The intended audience of this document is the potential end user (Voters and candidates). The document may also serve as a reference guide to the developers of the system.*

## **Definitions, Acronyms and Abbreviations**

*Admin: An administrator.*

*Ballot: A round of voting, or the record of a voter’s choices.*

*Candidate: A person who seeks or is nominated for an office.*

*Cast: The process by which one votes.*

*Database: A collection of data arranged for ease and speed of retrieval or search.*

*Election: The selection of a person or persons for office by vote, or a public vote on a proposed submittal.*

*Electorate(voters) : The body of persons enlisted to vote in an election.*

*Poll: The place where votes are taken.*

*Voter: One who votes, a member of the electorate, or a citizen with the right to vote.*

## **References and Acknowledgments**

*None*

# **Overall Description**

## **Product Overview**

*The software product is a standalone system and not a part 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 webpages. The two users of the system, namely the voters and the admin interact with the system in different ways. The admin configures the whole system according to its 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.*

## **Product Functionality**

*On the voters side, 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. From the Admin side, the system can be used to create/update/delete the election details (posts, candidates, electoral rolls etc. ). The admin should be able to specify the different attributes it wants for posts/candidates of a particular election instance and voters.*

## **Design and Implementation Constraints**

*•GUI should be in Arabic and English.*

*•Limited to HTTP/HTTPS.*

*•Users should have basic knowledge of computers.*

## **Assumptions and Dependencies**

*•The end user should have a basic knowledge of Arabic and computer usage.*

*•The voting results will be managed and calculated by the admin.*

*•Roles and tasks are predefined.*

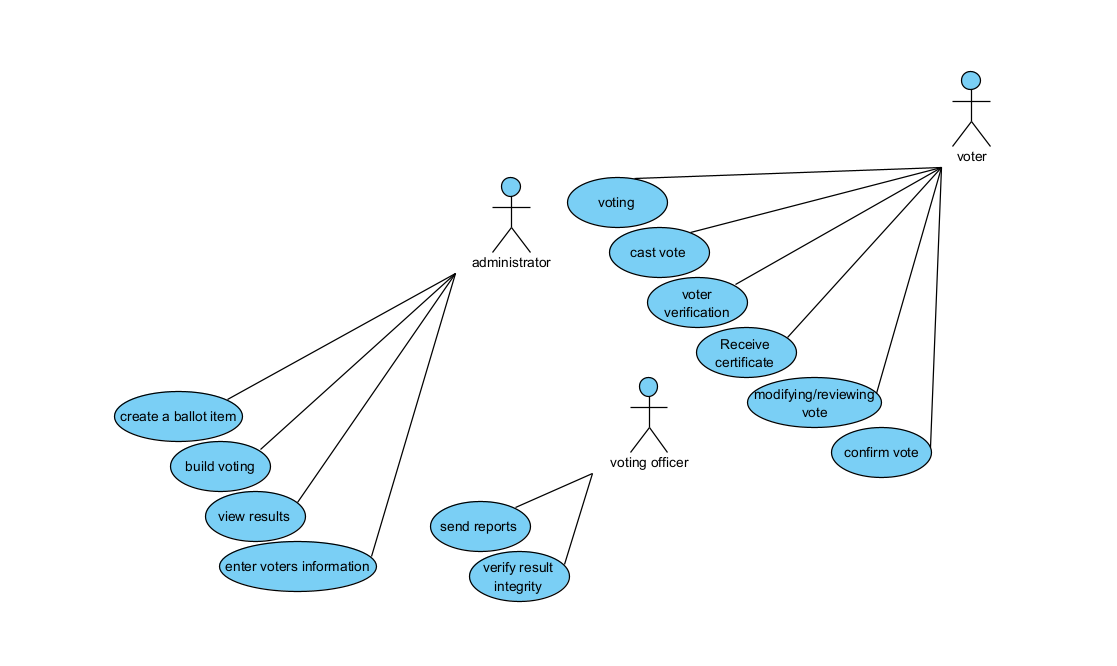
*•The Database system is fully functional and has enough space for at least one vote.*

# **Specific Requirements**

## **Functional Requirements**

* *The system shall provide an easy-to-use user-interface. Also, it shall not disadvantage any candidate while displaying the choices.*
* *The system shall record and count all the votes and shall do so correctly.*
* *The system shall allow the voters to cast their votes quickly and shall be given a chance to modify his vote before he commits it., from the first login, and should not require special skills.*
* *The voter shall be able to confirm clearly how his vote is being cast.*
* *Voters should not be able to vote more than once.*
* *Voters should be able to possess a general knowledge and understanding of the electronic voting process.*
* *Admin shall not change the voter's decision in choosing the person who elected him.*
* *Only allowed voters should be able to vote.*
* *Online election systems should be affordable and efficient.*
* *The design, implementation, and testing steps must be well documented so that the voter-confidence in the election process is ensured.*
* *The voter shall not choose more than one candidate.*
* *The system shall issue a certificate to the vote After voting process, so that he may verify his voting.*
* *The system should provide restricted ID no as a password for every user.*

## **Use Case Model**

**

### **Use Case #1** Voting

**Actor**: Voter

**Description**: this use case allows voters doing the voting process.

**Precondition**:

1. The Voter has been registered to cast his vote.

2. The voter has been logged in

**Postcondition**:none

**Basic Path**:

1. Voter Logs in.

2. Voter selects the existing ballot and makes a selection on the ballot.

3. Voter submits a vote.

3. Voter logs out of the system.

**Alternate Path**:

1a. System ascertains that it was given incorrect login information.

1a.1. System informs voters.

1b. Voter entered the wrong password for the third time.

1b.1. Voters are not allowed to login( use case ends, failure).

2a. Voter can't see the vote he is looking for because he is not a member of the voting group.

2b.1. System informs voters. ( Use case ends in failure).

3a. The Voting period is already over.

3b.1. System informs voters. ( Use case ends in failure).

### **Use Case #2** Voter Verification

**Actor**: Voter

**Description**: This use case allows voters to verify the voters' registration.

**Preconditions**: None

**Postconditions**: The voter is a registered voter and is authorized to vote.

**Basic Path**:

1. The voter submits data for the system to verify.

2. The system validates the data to check if the voter is registered to vote.

3. The System then checks if the voter has already cast his vote in this election.

**Alternate Path**:

2. The voter has entered incorrect data. The System prompts the user to enter correct data.

3. The voter has already voted in this election, so the System informs them they have already voted in the election.

### Use Case #3 Cast Vote

**Actor**: Voter

**Description**: This use case collects the voter's ballot and sends them to the Independent Electoral Commission.

**Preconditions**: The Voter is a registered voter.

**Postconditions**: The Voter casts his vote and the records are updated.

**Basic Path**:

1. The Voter is presented with an appropriate ballot.

2. The Voter casts his vote and submits it to the System.

3. The System asks the voter to verify the vote.

4. The Voter confirms the vote.

5. The System updates the appropriate records.

6. A receipt, confirming the Voter has voted ,is provided.

**Alternate Path**:

4. The Voter does not confirm the vote. The System presents him the ballot to cast his vote.

### Use Case #4 Modifying/Reviewing Vote

**Actor**: Voter

**Description**: This use case allows voters to modify and review their votes before confirming it.

**Precondition**: The voter has been authorized to cast his vote.

**postcondition**: none

**Basic Path**:

1. After a vote has been cast by the voter, the voter can review his vote and can modify his vote until he presses the confirm button.

2. Voter submits a vote.

3. Voter logs out of the system.

**Alternate Path**:

1a. Vote has already pressed the confirm button.

1a.1. System informs voters. (Use case ends in failure).

### Use Case #5 Confirm Votes

**Actor**: Voter

**Description**: This use case displays the voter's ballot to them for confirmation prior to them being officially cast.

**Precondition**: The voter has completed their election selections.

**postcondition**: The voter indicates that these votes are the ones they wish to submit.

**Basic Path**:

Once a voter selects his candidate he clicks on the confirm button.

**Alternate Path**:

The voter lost his internet connection. (Use case ends in failure).

### Use Case #6 Build voting

**Actor**: Administrator

**Description**: this use case Build voting and upon completion all registered voters are notified of the voting and ballots are sent.

**Precondition**: Administrator is authenticated to enter details.

**Postcondition**: voting is created.

**Basic Path**:

1. Administrator chooses a date, duration for voting and fills all relevant information of the voting.

2. System creates the new voting.

**Alternate Path**:

1a. Administrator decides to quit System does not store anything.(Use case ends in failure).

### Use Case #7 View Results

**Actor**: Administrator

**Description**: This use case shows the results of the votes.

**Preconditions**: The election time has expired.

**Postcondition**: Get the result of votes

**Basic Path**:

1. The Administrator displays a list of candidates with the highest votes.

**Alternate Path**:

The voter lost his internet connection and can't see the results.(Use case ends in failure).

### Use Case #8 Create a Ballot Item

**Actor**: Administrator

**Description**: Create Ballot Item and add it to the election.

**Preconditions**: The Administrator created the election.

**Postcondition**: Ballot items are created and added to the election.

**Basic Path**:

1. The Administrator enters the information about the ballot. Like the candidate name, party name.etc.

2. The System asks for confirmation.

3. The voting coordinator gives confirmation.

4. The System adds the ballot to the voting.

**Alternate Path**:

1. The Administrator does not give his confirmation. The System displays a menu: delete ballot item, modify ballot item.etc

### Use Case #9 Enter Voters Information

**Actor**: Administrator

**Description**: This use case allows an Administrator to enter the list of registered voters for an election.

**Preconditions**: The voting system has been properly initialized(Database created).

**Postconditions**: The voter has been confirmed to be eligible to vote.

**Basic Path:**

1 . The Administrator enters the voters' data into a database designated for this.

2 . The system stores voter data.

Alternate Path:

1a.Wrong data entry.(Use case ends in failure).

### Use Case #10 Verify Result Integrity

**Actor**: Voting Officer

**Description**: This use case tally the votes casted in the system with the paper output printed and collected after the vote.

**Preconditions**: Voting Officer should login as election observer.

**Postconditions**: Tally result is right or there is a discrepancy in the tally of votes.

**Basic Path**:

1. The Voting Officer logs into the system.

2. The system gives a tallied count of the votes casted to the Voting Officer.

**Alternate Path**:

1a. The Voting Officer quits the election System without doing anything.(Use case ends in failure).

### Use Case #11 Send Reports

**Actor**: Voting Officer

**Description**: This use case allows a Voting Officer to send election reports to the study centers in order to study (the number of voters, the plans of the candidates, etc.)

**Preconditions**: Voting Officer should login as election observer.

**Postconditions**: Reports are sent to the study centers properly.

**Basic Path**:

The voting officer sends the reports by pressing the “Send Reports” button.

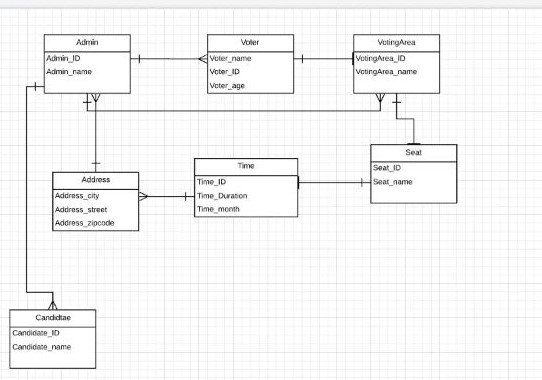
**Alternate Path**:

Export the voting report as an excel / PDF file, and send them by email.

# Logical Design (Relational Schema)



Relational schema:



# Conceptual Design (ER Diagram)

open link:

<https://drive.google.com/file/d/18mLufVSW1C3KO6QKrmfeOPu7_87LY9HG/view?usp=sharing>

