Online Election System

Submitted by : Aditi Singh

Roll no. : 21051874

Batch : Cse -22

Subject : Software Engineering

Project : SRS Document

**1.Introduction :**

The document classifies the basic properties and and programming necessities

the Online Election System .

* 1. **Problem Definition**

Manual voting framework has been utilized for a long time. However , in numerous parts of the nation , individuals cannot go to the voting site as a result of various reasons . Occasionally , individuals may not be in their own enrollment area . To take care of these issues , there is a need of online decision voting framework along with manual voting framework . Subsequent to enrolling to framework , the voters will utilize their votes at any field zones by deploying the framework .

* 1. **Purpose**

The reason for this document is to make the practical and non -practical prerequisites of the Online Election Voting System straightforward. It likewise effectively makes the usefulness clear to end clients .

1.3 **Scope**

This SRS document applies to the preliminary version of the Online Election System software package . This document portrays the demonstrating and necessity examination of the framework . the fundamental point of the framework is to give an arrangement of conventions that enable voters to cast votes while a gathering of specialists gather votes and yield last outcomes .

1.4 **Definitions and Abbreviations**

The subsequent is a rundown of terms , acronyms and abbreviations used by Online Election System software package and related documentation

|  |  |
| --- | --- |
| ONEV | Online National Election Voting |
| EC | Election Candidate |
| ECA | Election Commission Authority |
| ESS | Election Station Supervisor |
| VIN | Voter Identity Number |
| DB | Database |

Table 1 . Acronyms and Abbreviations used in system

1.5. **Overview**

The rest of this document categorizes the actors , use cases, use case scenarios , activity diagrams, assumptions and dependencies required for the investigation and plan of the Online Election voting software . The remaining document contains the general depiction of the framework , prerequisites , information and social portrayal of the framework .

1. **Functional requirements**

**2.1 Manage voters**

**2.1.1 Voter registration**

Description : A form filled where name , gender , D.O.B , VIN and other personal details are filled.

Input :Personal Details

Output :List of voters applied is generated

**2.1.2 Approve/Confirm applicant**

Description : The ECA has the job to approve and verify the applied applicant .

Input :Form verification

Output :Confirmation of registration

**2.1.3 Update registered voters**

Description : An official voter list is generated and voters who cannot vote are discarded

**2.2 Manage candidates**

**2.2.1 Open candidate account**

Description : A form filled where name , gender , D.O.B , VIN and other personal details are filled and candidates with no past criminal record is checked

Input :Personal Details

Output :List of candidates applied is generated

**2.2.2 Approve candidate**

Description : The ECA has the job to approve and verify the applied candidate.

Input :Form verification

Output :Confirmation of registration of candidate.

**2.2.3 Update candidate list**

Description : An official candidate list is generated and candidate who cannot stand for the election are discarded.

**2.2.4 Campaign for election**

Description : The candidates of the party campaign about their agendas , policies and working principle

Input :Personal agenda , posters , party policies are promoted ,

Output : Displayed on all applicants user screen

**2.2.5 Reading and answering public questions**

Description : The candidates must be able to answer to the public queries that arise throughout the campaign .

Input : Question from the public

Output : Answers to specific question in the dedicated panel .

**2.3. Mange Votes**

**2.3.1 Login and password generation**

Description :The ESS generates the password for each registered voters to login for casting their votes

Input : Election day framework set up by ESS

Output : Unique password generated for each registered voter

**2.3.2 Online voting**

Description : The voters can vote from anywhere to any candidate as per their choice using the password given to them .

Input :Login with password

Output :Voters vote to their decided candidate and voters\_count database updated.

**2.3.2 Offline voting**

Description : The ESS has the job to enter the offline vote count to the framework .

**2.4 Counting and displaying the result**

Description : Counting the votes stored in Database on the election result day .

Input :The ESS is assigned the job for counting votes from database

Output :The result is displayed.

**Asking**

**questions**

**Register**

**Voter**

**verify**

Voter include

ECA

**Login**

**Voting**

>

include

**Register**

**Candidate**

**Verify**

include

**Campaign**

Candidate ESS

**Answer questions**

Use case diagram for the ONEV

**2.5 Product Perspective**

The product item is an independent framework and not a piece of a bigger framework . The framework will be comprised of two sections . Prior to the election day the framework will be utilized for general purposes , for example , seeing applicants profiles and past years election comes about . The voters will achieve the framework through site pages by utilizing web -programs , for example ,Mozilla , Internet Explorer and Google Chrome . On the Election day another autonomous framework will be utilized for voting tasks . This framework will be adjusted to the PCs at the surveying stations . The voters cast their votes utilizing the interface that are given at these machines . These votes are acknowledged by the framework on the server . The ECA arranges the entire framework as per its needs on the server where the framework is running.

ESS

Election

Station

Supervisor

Interface

Voter

(Normal

Mode )

Voter

Normal

Mode

Interface

DB

Election

Mode

Normal

Interaction

Mode

EC

Election

Candidate

Interface

Voter

(Election

Mode)

Voter

Election

Mode

Interface

Election

Commission

Authority

Interface

ECA

ONLINE NATIONAL ELECTION VOTING SYSTEM

**Diagram depicting interaction between users and system**

**2.6 Constraints , Assumptions and Dependencies**

Security and safety are the most essential basics on the ONEV framework . The framework has zero-resistance with respect trading off. The framework ought not enable ESS to download votes to deduce how voters in their areas have voted . The framework ought to give intends to ensure security relates of polls cast in election. For the best possible working of the framework we can list our presumptions and conditions as takes after .

* Working web associations
* A web server ought to have Java introduced on the machine , alongside Java’s cryptography bundles.
* The decision server keeps running on a http server, that is “jsp” empowered.

**3.Specific Requirements**

**3.1 Interface Requirements**

INTERNET

ESS

EC

Server

ECA

Voter

(Election Mode)

DB

Voter

(Normal Mode)

Relationship diagram depicting interface relations

**3.1.1User Interfaces**

The framework must give a UI to a wide range of clients (ECA , ESS, EC and Voter) that is accessible through all web programs . The UI for voter be distinctive for Election Mode and Normal Interactive Mode.

**3.1.2Hardware Interfaces**

There are no equipment interfaces to this product framework . The main interfaces are through a PC framework .

**3.1.3 Software Interfaces**

The survey server keeps running on http server that is empowered to deal with server pages.It utilizes a social database to monitor the surveys , which it associates through standard database network interfaces. With a specific end goal to run the setup programming , the machine needs JVM running on it .

**3.2 Normal Interaction Mode**

This is a typical mode -when election day - a client collaborates with the framework . It includes enrollment for voting , refreshing profile , seeing decision competitors(EC) and in addition sending them questions. It additionally incorporates capacities for the Election Commission Authority(ECA) to enlist EC and favor enrolled voters. The use cases depict the useful prerequisites.

**3.2.1 Activity diagram for Voter Registration**

Voters enters the system homepage

User clicks the “register now” button

System prompts the application form

Fills in the necessary information related about him in application form

He uploads a picture for Voter Identity Card (VID)

User sends the request for registration by using “Send” button

System prints appropriate error message

Information

Incorrectly entered

Information

Correctly entered

System prints a successful message

Activity Diagram of Voter registration

3.2.2 Activity diagram for asking questions .

User clicks on the “Questions/Answers “ link

User writes his questions on the text field

User presses “Send Questions”

Activity diagram for asking /answering questions

**3.3Election Mode**

**3.3.1 Open System**

The user enters his/her name, supervisorID and password

supervisorID and supervisorID and password are

Password are valid not valid

System verifies the security

System redisplays login page with

error message

System loads voting page

Activity diagram for opening system

**3.4 Non-functional requirements**

3.3.1. Performance Requirements

The framework is relied upon to have brief time response . The voter ought to have the capacity to login and ought to have the capacity to get reaction for his solicitations in 2-3 seconds . The framework’s execution is diverse as indicated by its mode .

* In Election Mode : The framework is relied upon to serev a greatest up to 50000 voters in a split second , every voter being dynamic for at most 5 min asking for upto 5 pages .This demonstrates the framework ought to have the capacity to deal with 2000 transactions each second . Likewise, the framework must work at 100% peak proficiency amid the voting procedure .
* In Normal Interactive Mode: The framework in this mode is required to serve greatest of up to 5000 voters, however every voter can be dynamic for quite a while .

3.3.2 **Security Requirements**

* The information exchange amongst customer and server must be encoded
* All the passwords that are created or acknowledged must be put away in database in an encrypted shape
* To counteract assaults the framework ought to create arbitrary word and request that the client enter it accurately for various trying.

3.3.3 **Safety Requirements**

* To prevent data loss in case of system failure , the result of votes that are polled till then have to be saved in database .
* In case ECA identifies any security issue in the framework, he ought to have the capacity to closed down the framework and keep all association with the server quickly to protect polled votes .
* The framework ought to have the capacity to recoup itself from past accidents
* and proceed with the voting procedure .
* The framework ought to caution ECA clients about the glitch of the framework .

3.3.3.4 **Reliability**

* In election mode : The framework ought to be 99% dependable.
* In normal mode:Keeps the upkeep for election day so around 80% dependability.

**4.Data Model and Description**

**4.1 Data description**

**4.1.1 Data objects**

The data objects and their fundamental characteristics are classified as follows .

Login : User ID, Password .

User Lists: Voters , ECs , ECAs, ESSs

User: Name , Address, VotingCenterID, Age, Sex , UserID, password

Candidate Votes: earnt votes, total votes, percentage, rank

Candidate: Political party , Ruling area , position

Stations : station ID , ESS

**4.1.2. Relationships**

**A. Associations**

The subsequent relationships depict association in ONEV framework .

* Login and User
* Login and ESS
* Candidate and Candidate votes
* User and Candidate votes
* Stations and Candidate votes

1. Composition

The subsequent relationships depict composition in ONEV framework

* User and User List
* ESS and Stations

**Uname**

1. Behavioral Model and description

5.1 Entity Relationship Diagram

**ECA**

1

**Login-id**

**Manages**

**Pwd**

**Constitution**

**VIN**

1

**DOB**

M

**Manages**

**Voter**

1

**Votes**

**Pwd**

**Age**

**L-name**

**M-name**

**F-name**

**Constitution**

**Status**

**Name**

**Gender**

**C-id**

**Age**

**Candidate**

1 M

**Name**

**Gender**

ERD of Voter and Candidate in Election mode

5.2 **Data Flow Diagram**

**Voter**

Voter details

**ONEV**

Confirm registration winner list

candidate details

**Candidate**

Generate result

Confirm

registration

**ESS**

Context diagram of system

**Manage**

**candidate**

Voter details

Candidate details

**Manage**

**voters**

U

Confm\_reg

Confm\_reg

Candidate list

Voter list

**Manage**

**votes**

Vote\_update

**Display**

**Result**

Generate result

winner list

Level1 DFD

1. **Process model**

As our requirements are clear beforehand and wont have enormous changes , we will use the waterfall model



**7.Conclusion**

The SRS archive is set up for a superior plan of ONEV framework . The practical and different necessities of the framework are depicted and the requirements of the client are expressed all through the document .

Candidate

Login

+political party :String

+ruling area:String

+position :string

+earned votes :int

+total votes:int

+station id :string

+ESS : string

+name:string

+address: string

+age:int

+sex:string

+userid:string

+password:string

user List

Voter

+voter:user

+ESS:user

+ECA :user

votes

Stations

ESS

+ userid :string

+password :string