

VIT Bhopal University

School of Computer Science & Engineering

SRS Group Report

Course Code:CSE3005

Course Name:**SOFTWARE ENGINEERING**

Slot: A11+A12+A13

Semester: Winter 2020-2021 Submitted To:

**Dr. Subhash Chandra Patel**

**GROUP MEMBERS**

|  |  |
| --- | --- |
| 1. **ADITYA VISHWAKARMA** | **18BCE10331** |
| 2. **ANUJ MATHUR** | **18BCE10045** |
| 3. **PIYUSH VISHWAKARMA** | **18BCE10186** |
| 4. **PRANSHU KHARE** | **18BCE10193** |
| 5. **SIDDHANT SINGH** | **18BCE10266** |
| 6. **ANUGRAH DIXIT** | **18BCE10042** |
| 7. **SATYAM SHEKHAR** | **18BCE10316** |

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.N0** | **TOPIC** | **PAGE NO** |
| 1 | **INTRODUCTION** | 5 |
| 1.1 | PROBLEM DEFINITION | 5 |
| 1.2 | PURPOSE | 5 |
| 1.3 | SCOPE | 5 |
| 1.4 | OVERVIEW | 5 |
| **2** | **OVERALL DESCRIPTION** | 7 |
| 2.1 | PRODUCT PERSPECTIVE | 7 |
| 2.2 | PRODUCT FUNCTIONS | 8 |
| **2.2.1** | NORMAL INTERACTIVE MODE | 8 |
| 2.2.1.1 | APPROVE APPLICANT | 8 |
| 2.2.1.2 | UPDATE REGISTERED VOTERS | 8 |
| 2.2.1.3 | OPEN CANDIDATE ACCOUNT | 9 |
| 2.2.1.4 | LOGIN/LOGOUT | 9 |
| 2.2.1.5 | ACCOUNT UPDATE | 9 |
| 2.2.1.6 | VIEW EC INFORMATION | 10 |
| 2.2.1.7 | ASK TO CANDIDATE | 10 |
| 2.2.1.8 | CV EDIT | 10 |
| 2.2.1.9 | ADD/EDIT PROMISES | 11 |
| 2.2.1.10 | READ/ANSWER QUESTIONS | 11 |
| 2.2.1.11 | VIEW ELECTION RESULTS | 11 |
| **2.2.2** | ELECTION MODE | 12 |
| 2.2.2.1 | OPEN SYSTEM | 12 |
| 2.2.2.2 | MARK GENERATE PASSWORD | 12 |
| 2.2.2.3 | ONLINE VOTE | 12 |
| 2.2.2.4 | ENTER OFFLINE VOTES | 12 |
| 2.3 | CONSTRAINTS,ASSUMPTIONS AND DEPENDENCIES | 13 |
| **3** | **SPECIFIC REQUIREMENTS** | 13 |
| 3.1 | INTERFACE REQUIREMENTS | 13 |
| 3.1.1 | USER INTERFACE | 14 |
| 3.1.2 | HARDWARE INTERFACES | 14 |

|  |  |  |
| --- | --- | --- |
| **S.N0** | **TOPIC** | **PAGE NO** |
| 3.1.3 | SOFTWARE INTERFACES | 14 |
| 3.2 | FUNCTIONAL REQUIREMENTS | 15 |
| 3.2.1 | NORMAL INTERACTIVE MODE | 15 |
| 3.2.1.1 | VOTER REGISTRATION | 16 |
| 3.2.1.2 | APPROVE APPLICATION | 17 |
| 3.2.1.3 | UPDATE REGISTERED VOTERS | 19 |
| 3.2.1.4 | OPEN CANDIDATE ACCOUNT | 21 |
| 3.2.1.5 | LOG IN/LOG OUT | 23 |
| 3.2.1.6 | UPDATE ACCOUNT | 25 |
| 3.2.1.7 | VIEW ELECTION CANDIDATE INFORMATION | 27 |
| 3.2.1.8 | ASK QUESTION TO A CANDIDATE | 29 |
| 3.2.1.9 | CV EDIT(CANDIDATE) | 30 |
| 3.2.1.10 | ADD/EDIT PROMISES | 32 |
| 3.2.1.11 | READ/ANSWER QUESTIONS | 33 |
| 3.2.1.12 | VIEW ELECTION RESULTS | 35 |
| 3.2.2 | ELECTION MODE | 36 |
| 3.2.2.1 | OPEN SYSTEM | 36 |
| 3.2.2.2 | MARK VOTED AND GENERATE PASSWORD | 37 |
| 3.2.2.3 | VOTE ONLINE | 39 |
| 3.2.2.4 | ENTER OFFLINE VOTES | 41 |
| 3.3 | NON-FUNCTIONAL REQUIREMENTS | 43 |
| 3.3.1 | PERFORMANCE REQUIREMENTS | 43 |
| 3.3.2 | SECURITY REQUIREMENTS | 43 |
| 3.3.3 | SAFETY REQUIREMENTS | 43 |
| 3.3.4 | OTHER | 44 |
| **4** | **DATA MODEL AND DESCRIPTION** | 44 |
| 4.1 | DATA DESCRIPTION | 44 |
| 4.1.1 | RELATIONSHIPS | 44 |
| 4.1.2 | COMPLETE DATA MODEL | 45 |
| 4.1.3 | DATA DICTIONARY | 46 |
| **5** | **BEHAVIORAL MODEL AND DESCRIPTION** | 46 |

|  |  |  |
| --- | --- | --- |
| **S.N0** | **TOPIC** | **PAGE NO** |
| 5.1 | DESCRIPTION OF SOFTWARE BEHAVIOR | 46 |
| 5.2 | STATE TRANSITION DIAGRAMS | 47 |
| **6** | **PLANNING** | 47 |
| 6.1 | PROCESS MODEL | 47 |
| **7** | **CONCLUSION** | 48 |
| **8** | **REFRENCES** | 48 |

## Introduction

This document describes the structural properties and software requirements of the Online National Election Voting System project.

## Problem Definition

Manual voting system hasbeendeployed for many years in our country. However, in many parts of our country people cannot attend the voting because of several reasons. To illustrate, sometimes people may not be in their own registration region and due to this fact, they cannot fulJill their voting duties. In order to solve these problems, there is a need of online election voting system in addition to manual voting system.

After registering to system, the voters will use their votes at any Jield areas by using the system if they prefer online voting.

## Purpose

Thepurposeofthisdocument is to makethefunctional and non- functionalrequirementsof the Online National Election Voting System easy to comprehend. It alsoserves the purpose of making the functionality clear to end users.

## Scope

This SRS document applies to the initial version (release 1.0) of the “Online National Election System” software package. This document describes the modeling and the requirement analysisofthesystem. The mainaim ofthe system is to provideasetofprotocolsthatallow voters to cast ballots while a group of authorities collect votes and output Jinal results.

# Overview

The remainder of this document identiJies the actors, use-cases, use- case scenarios, activity diagrams, assumptions and dependencies

needed for the analysis and design of the Online National Election Voting software package. The rest of the document contains the overall description of the system, requirements, data model and behavioral description of the system and project planning.

|  |  |
| --- | --- |
| **ABREVETIONS** | **DEFINITIONS** |
| **ONEV** | Online National Election Voting |
| **EC** | Election Candidate |
| **ECA** | Election Commission Authority |
| **ESS** | Election StationSupervisor |
| **VIN** | Voter Identity Number |
| **DB** | Database |
| **VIC** | Voter Identity Card |

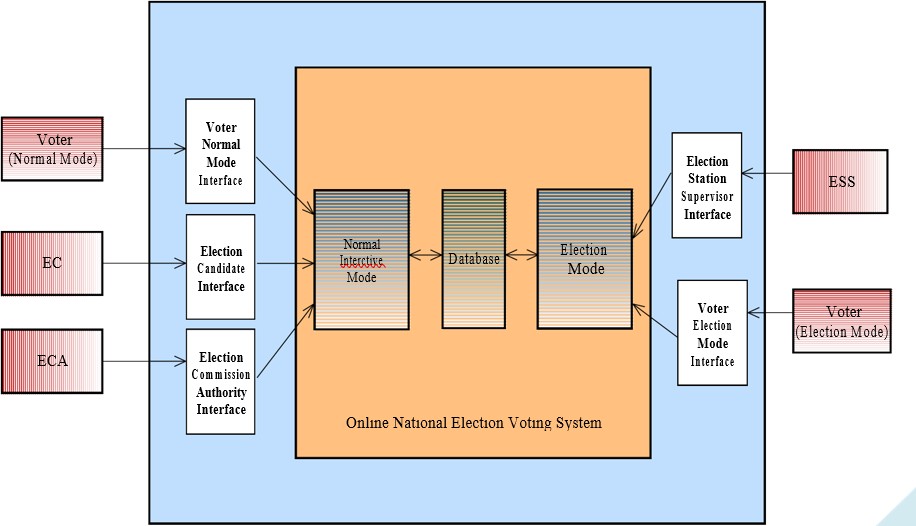
## Overall Description

The ONEV is a web-based system so fundamental features related with web- based technologies such as client-server and database properties determine the software requirements of that project.

# Product Perspective

The softwareproduct is a standalone system and not a part of a largersystem. The system will be made up of two parts. Before the election day the system will be used for general purposes such as viewing candidates’ proJiles and past years’ election results. The voters will reach the system through web pages by using web-browsers such as Mozilla, Internet Explorer and Google Chrome.

On the election day another independent system will be used for voting operations. This system will be adapted to thecomputers at the polling stations. The voters cast their votes using the interface that are provided at these machines. These votes are accepted by the system on theserver. The ECA conJigures the whole system according to its needs on the server where the system is running.



## Product Functions

The system can function in two modes, namely, Normal Interactive Mode and Election Mode. Thesystemwill be in Election Mode,forthepurpose ofvote polling onlyonthe Election Day. Normal Interactive Mode is for accepting registrations, discussions between voters and candidates,campaigns andthesystem is available in thismode allthetimeexceptElection Days.

## Normal Interactive Mode Voter Registration

That system will be used only by the people who havebeen registered to the system. Main actor of the registration operator is the voter. The registration operator is approved by the ECAs.



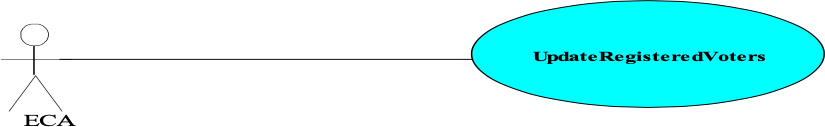
## Approve Applicant

By using this function, ECA approves the application sent by the voters in order to use the ONEV. The main actor is the ECA.



## Update Registered Voters

ECAdeletes voters from the system who cannot use their vote ofJicially. ECAalso updates voter’s information. The main actor is the ECA.



## Open Candidate Account

The EC’sproJile must be created by the ECA. This functionalityhelps to performthis action. The ECA is the main actor of this functionality.



**OpenCandidaeAccount**

## Login/Logout

All of the systemuser’s logintosystembytheiruseridsandpasswords.Allofthe usersare the main actor of this use case.



Voter

EC

**LoginLogout**

ESS



ECA

## Account Update

By using this function, the EC may change his passwordthat enters the system. The main actor of this use case is the EC.



## View EC Information

This function allows the voters to reach information about the EC such as their CVs, promises etc. Main actor is the voter for this use case.

**ViewECInformation**

## Ask To Candidate

By using this functionality, the voters can direct questions to the ECs about their election campaigns. The main actor for this use case is voter.

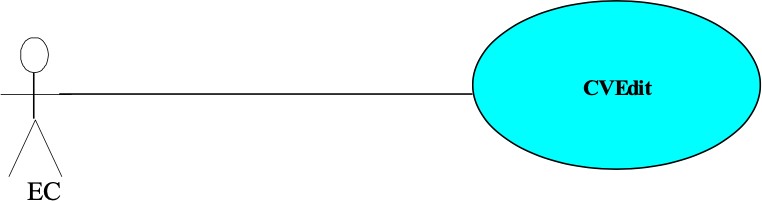


**AskToCandidate**

Voter

## CV Edit

ThisfunctionprovidestheECtoedithisCVinformationonhisownproJile. TheECis themain actor of thatfunctionality.





**ViewElectionResult**

## Add / Edit Promises

By using that function the EC’s may add or edit promises to their own proJile. The main actor of this use case is the EC.



**Add/EditPromises**

EC

## Read/Answer Questions

This function provides ECs to read or answerquestionsabouttheirelection campaigns. The main actor is the EC.



**Read/AnswerQuestions**

EC

## View Election Results

This functionality provides voters to seethe current orpast years’ election result in aproper way. The main actor is the voter.

## Election Mode {Open System}

This functionprovides ESS to startthesystemduringthe Election Day orbefore. The ESSis the main actor of this operation.



**OpenSystem**

ESS

## Mark Generate Password

By using that function the ESS will generate a password which will be used at voting operation by the voters. Main actor of this operation is the ESS.



**MarkGeneratePassword**

## Online Vote

This is themainfunction ofthesystemthatprovidesonlinevotingforthegeneral public. The main actor is the voter and votes are collected in the DB.



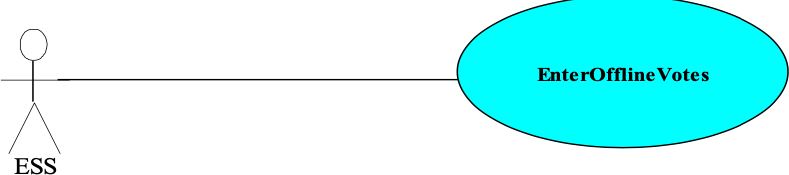
**OnlineVote**

Voter

## Enter Offline Votes

By using this function the ESSs entersthe ofJline votes to thesystem. The main actor

ofthis use case is the ESS.



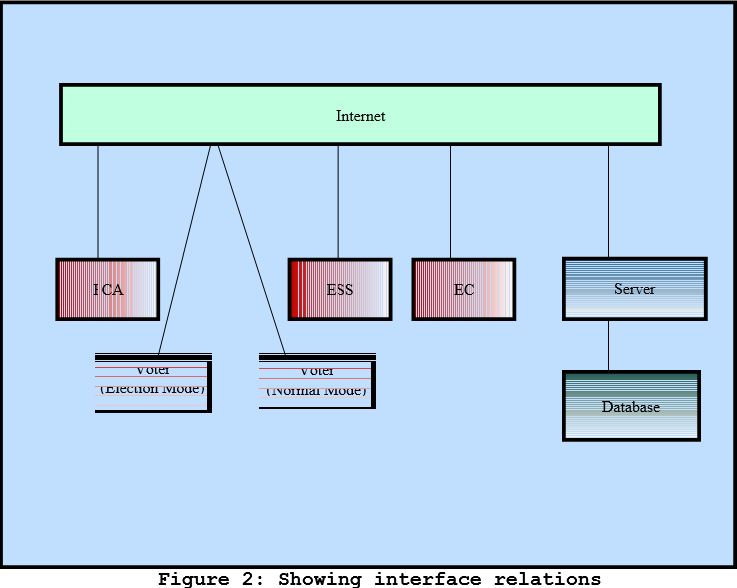
## Constraints, Assumptions and Dependencies

The system enables voters to poll their vote from any election centers that the system is installed in. In Turkey,the voting operation is executednearly in 150,000 ballotboxes. This means that the system will work on these boxes at the same time.

For the proper working of the system we can list our assumptions and dependencies as follows.

* Working internet connection
* A web 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.
* A web browser through which the voters access the server should have minimal support for cookies and encrypted transactions

**Specific Requirements {Interface Requirements}**



# User Interfaces

The system must provide a user interface for all types of users (ECA, ESS, EC, and Voter) thatis available through all Web browsers. The user interface for voter must be different for Election Mode and Normal Interactive Mode.

# Hardware Interfaces

Thereare no hardwareinterfaces to thissoftwaresystem. Theonlyinterfacesare througha computer system.

## Software Interfaces

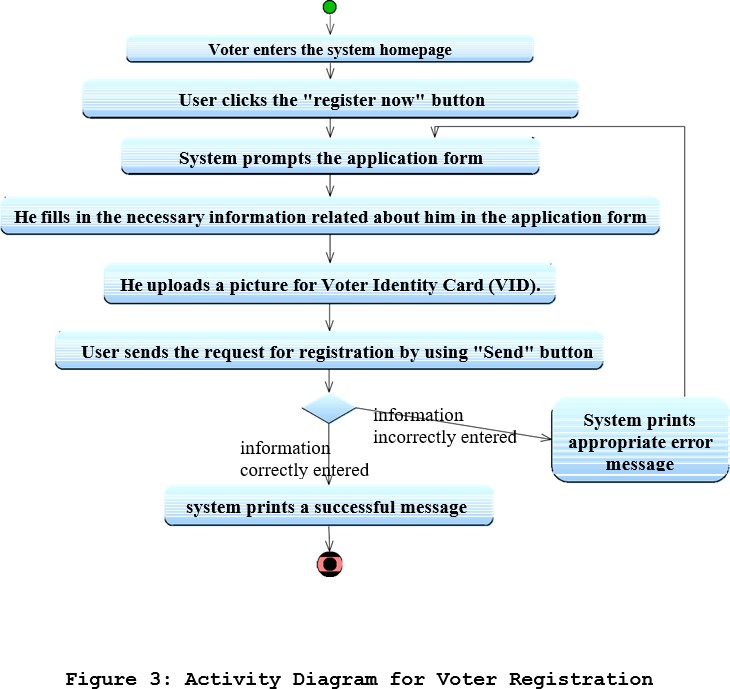
Thepollserverruns onhttpserverthat is enabled tohandleserverpages. It uses a relational database to keep track of the polls, which it connects through standard database connectivity interfaces. In order to runthesetupsoftware, the environment needs to have a Java Virtual Machine running onit.

## Functional Requirements {Normal Interactive Mode}

This is a normal mode –before and after Election Day - a user interacts with the system. It involves registration for voting, updating proJile, viewing election candidates (EC) as well as sending them questions. It also includes functions for the Election Commission Authority (ECA) to register EC and approve registered voters. The following use-cases describethe functional requirements.

**Voter Registration**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name**: VoterRegistration | **ID:** 1 | | **Priority:** High |
| **Primary actor:** Voter | | **Use case type:** Detail, essential | |
| **Stakeholders and interests:**  Voter – wants to register to system. | | | |
| **Brief description:** In order to use the system the voters must register to system. This explains the registration process. | | | |
| **Precondition:** None  **Trigger:** None | | | |
| **Relation ships: Associa tion: Include: Extend:** | | | |
| **Normal flow of events:**   1. Voter enters the system homepage. 2. He clicks the “register now” button. 3. The system prompts the application form. 4. He fills in the necessary information related with him in the application form. 5. He uploads a picture for Voter Identity Card (VID). 6. He sends the request for registration by using “send” button.    1. If the information is correctly entered the system prints a successful message.    2. Otherwise, it printsappropriateerrormessage,redisplays the application form. | | | |



**Approve Application**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name**: ApproveApplicant | **ID:** 2 | | **Priority:** High |
| **Primary actor:** ECA | | **Use case type:** Detail, essential | |
| Stakeholders and Interests :  Voters - Wants ECA to approve their application Form  ECA - Wants to approve the Voters by checking Applicaitons Form | | | |
| **Brief description:** This describe how ECA will approve the application form of voter and generate the new account to that voter | | | |
| **Precondition:** The voter should have Jilled his application form  **Trigger:** | | | |

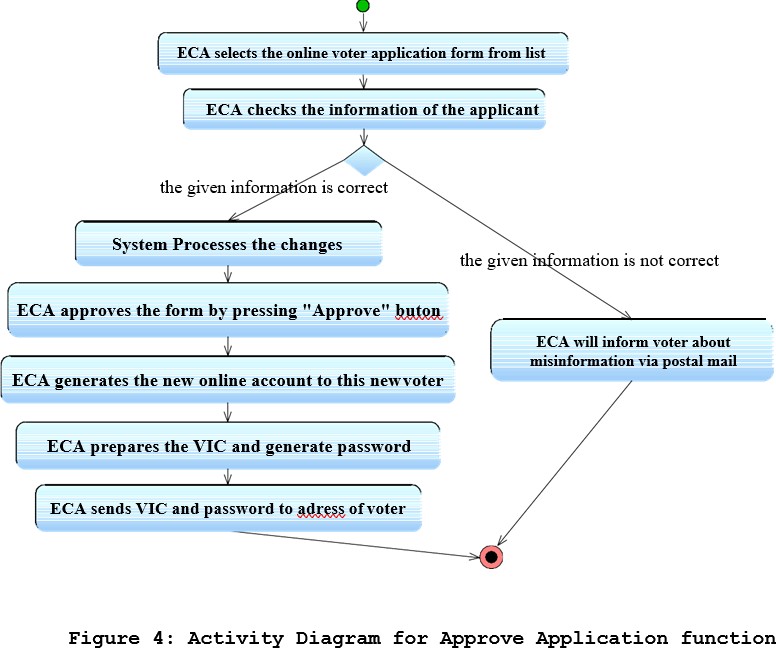
**Normal flow of events:**

1. ECA selects the online voter application form from list
2. ECA checks the information of the applicant
   1. If the the given information is correct
      1. ECA approves the form by pressing “Approve” buton
      2. ECA generates the new online account to this new voter
      3. ECA prepares the VIC and generates password
      4. ECA sends VIC and password to adress of voter
   2. if the given information is not correct
      1. ECA will inform voter about misinformation via postal mail

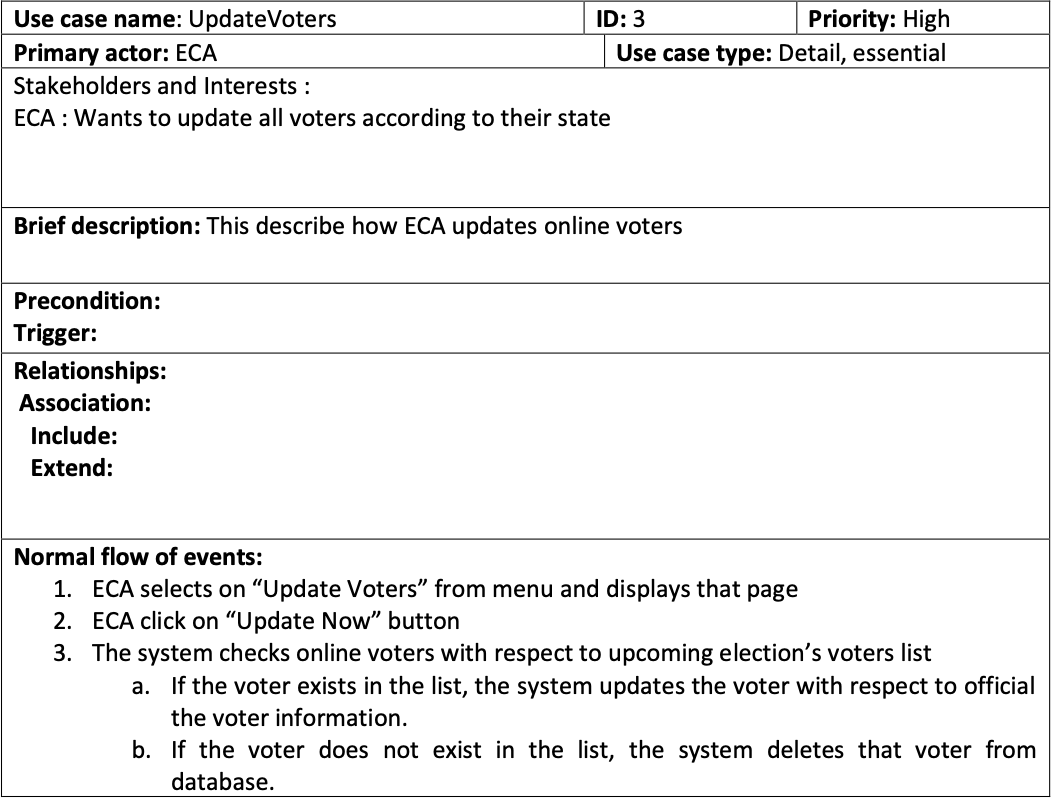
**Relation ships: Associa tion: Include**

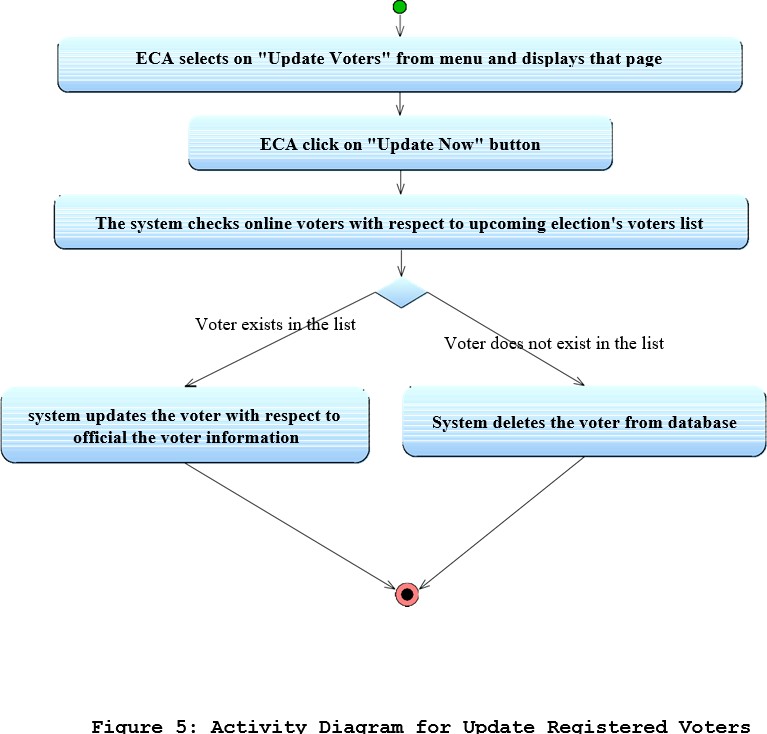
**:**

**Extend:**

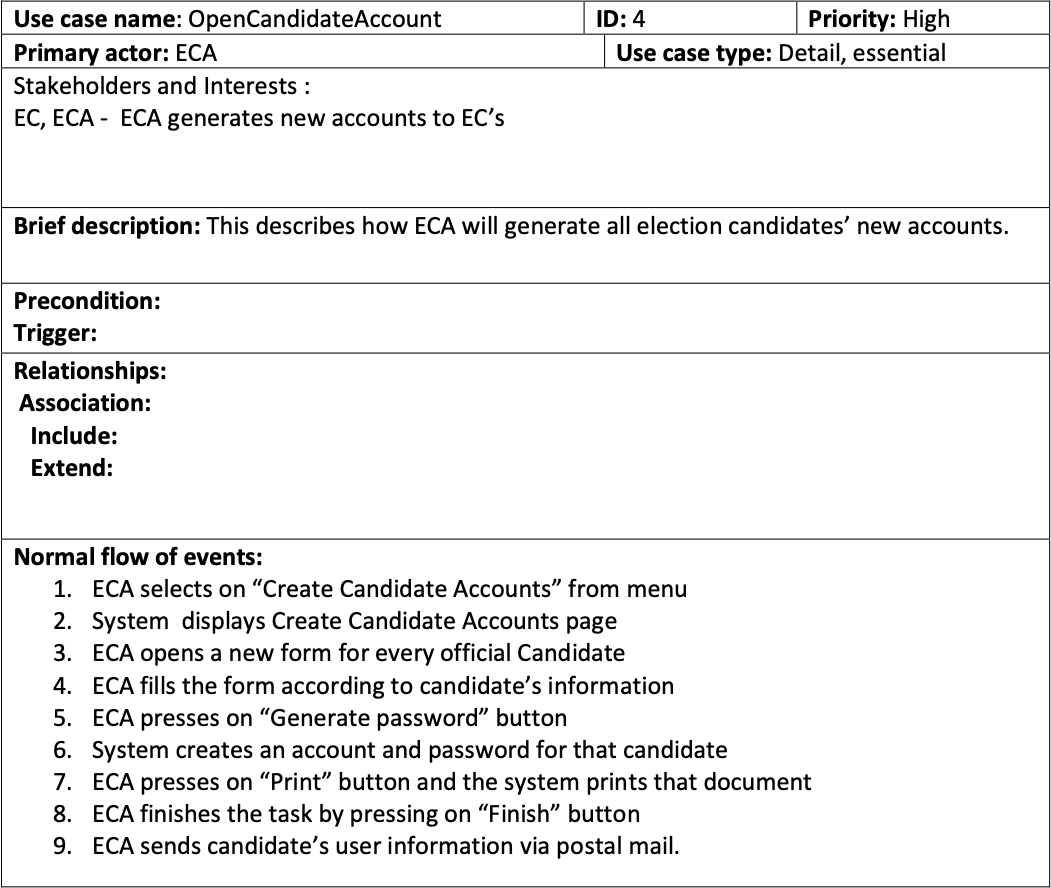


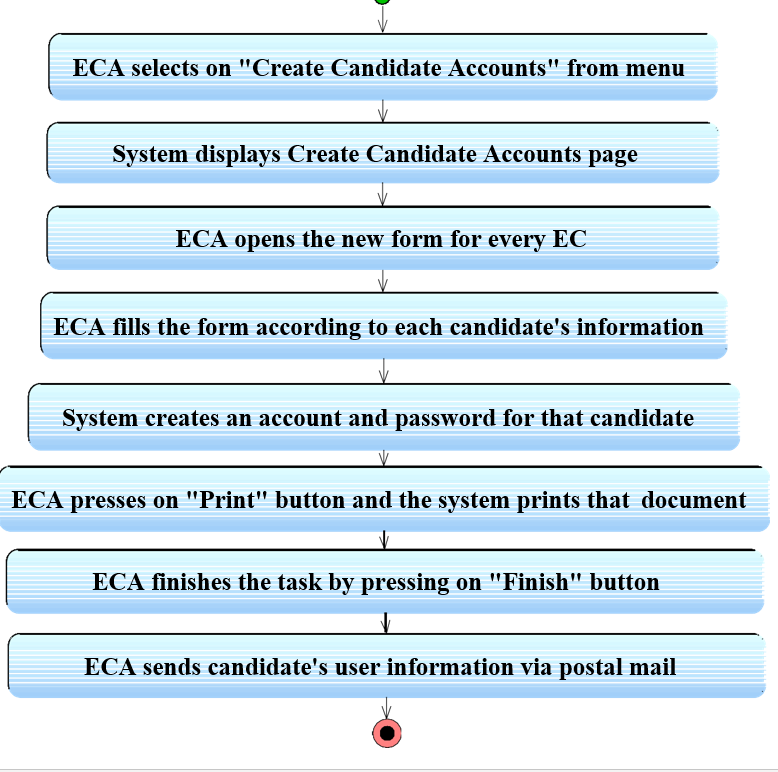
**Update Registered Voters**



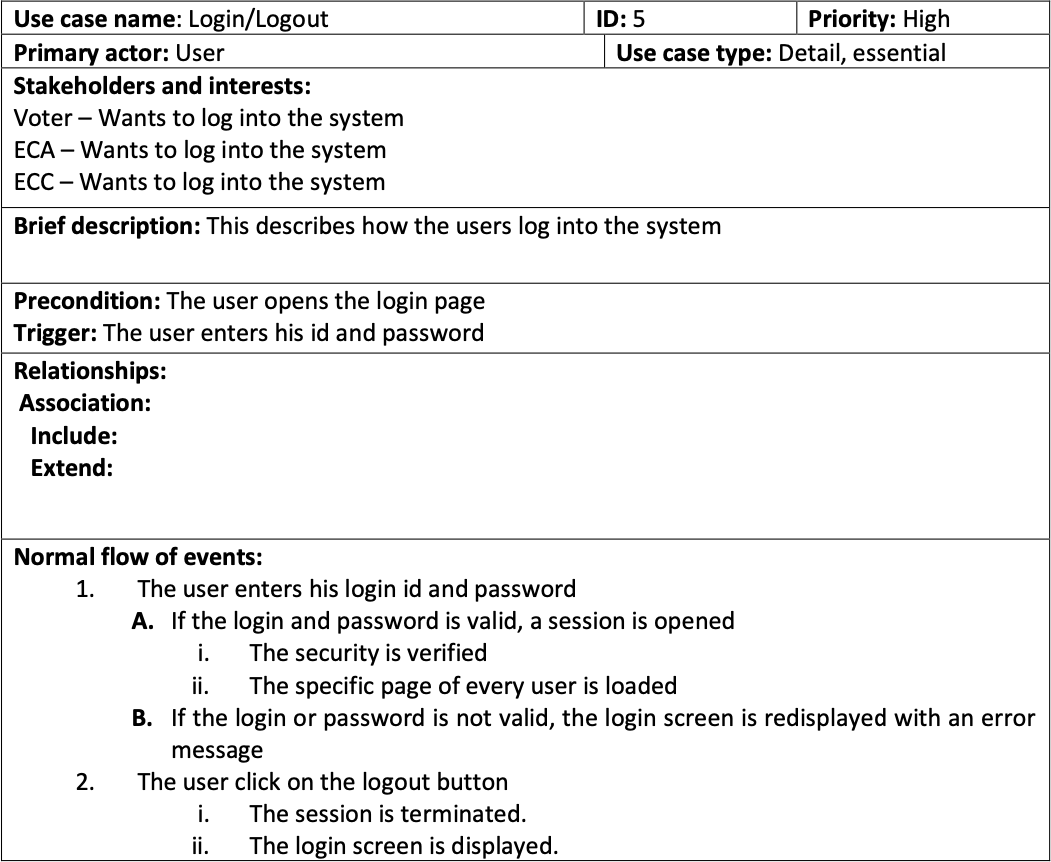


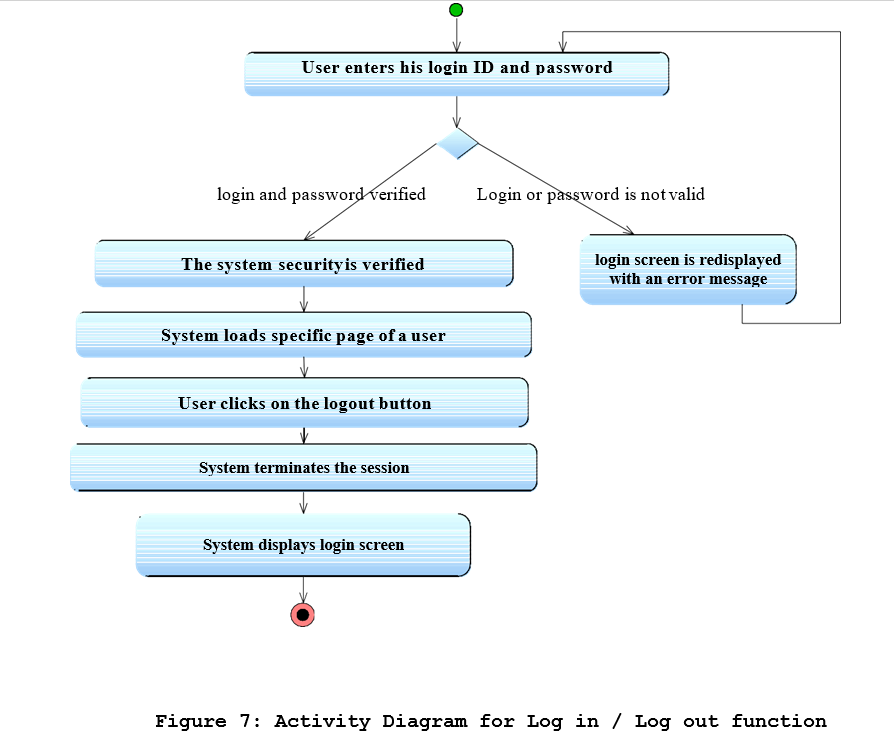
**Open Candidate Account**



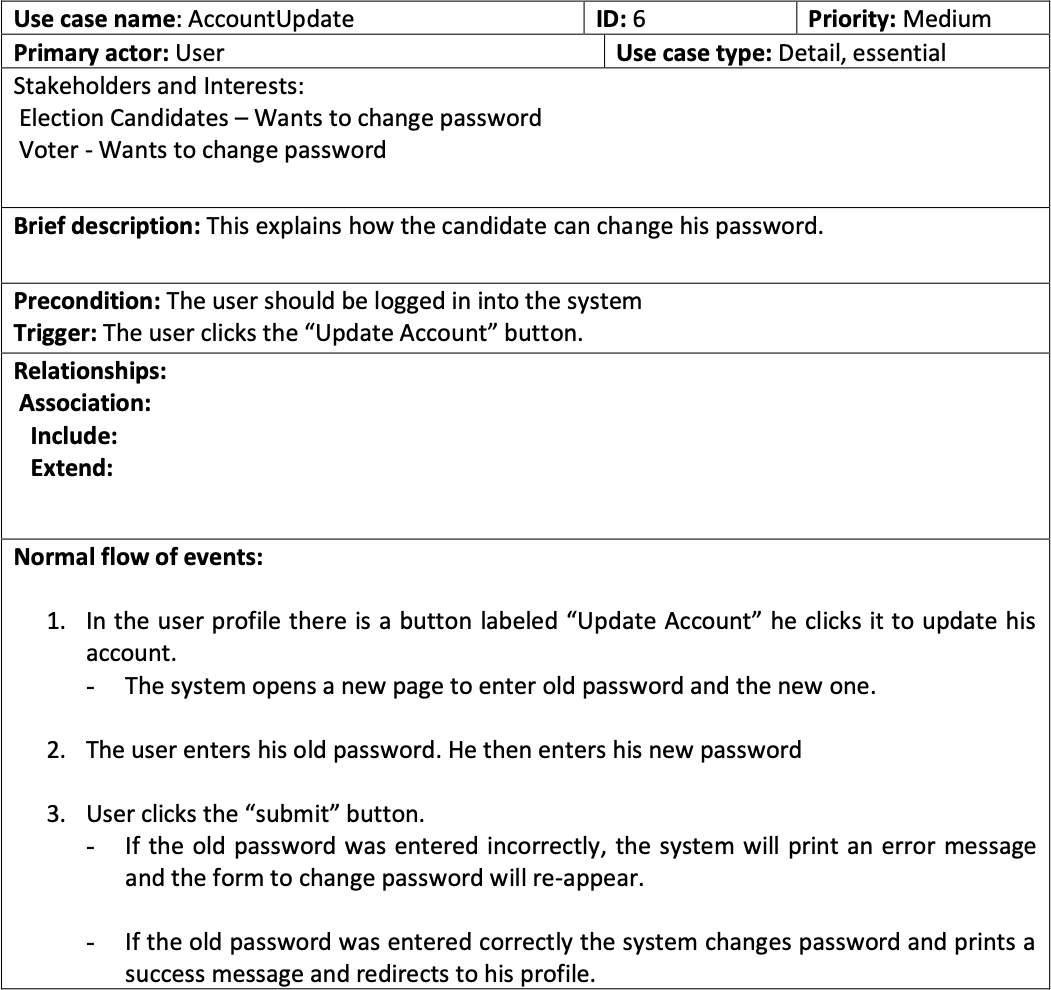


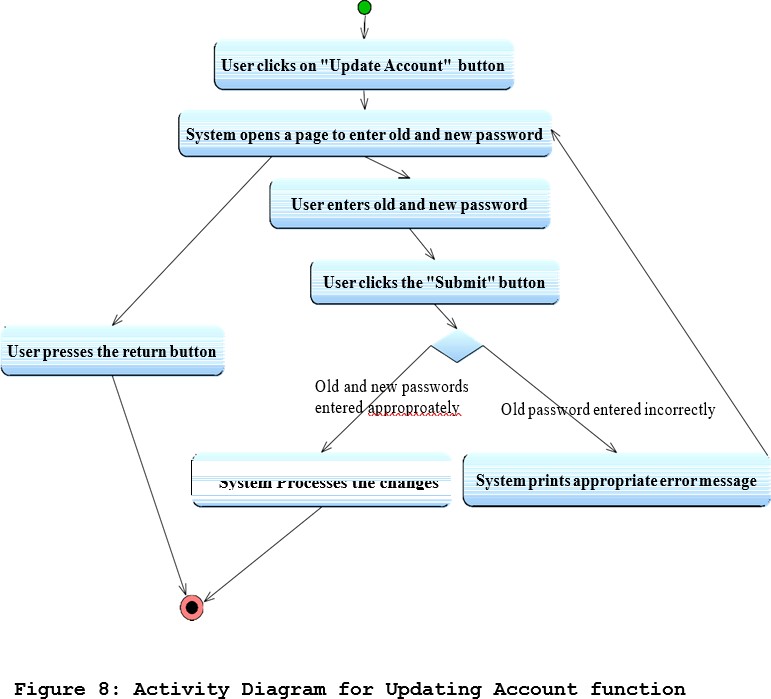
**Log In / Log out**





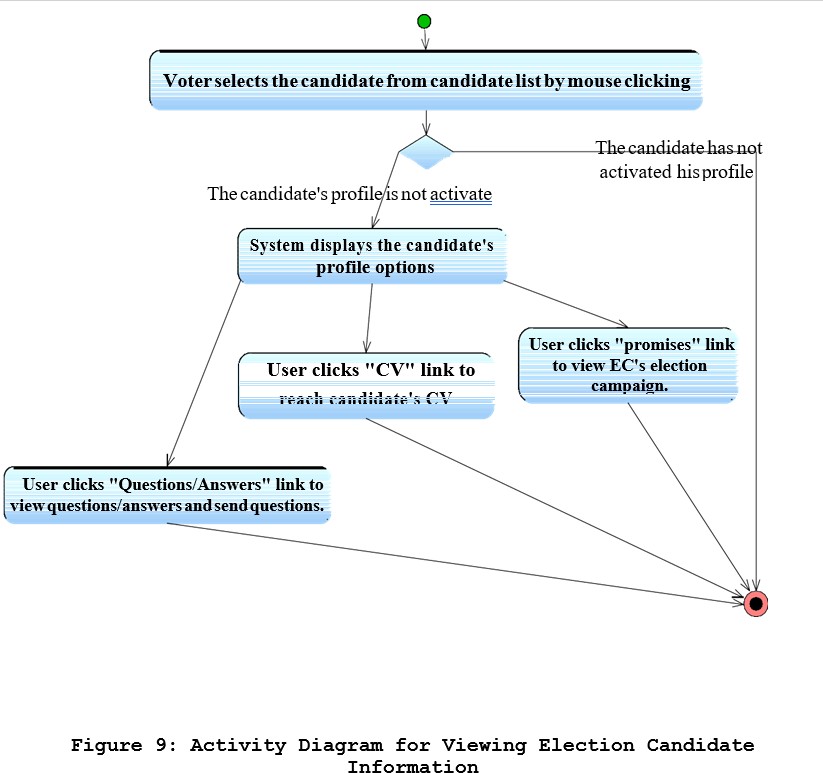
**Update Account**



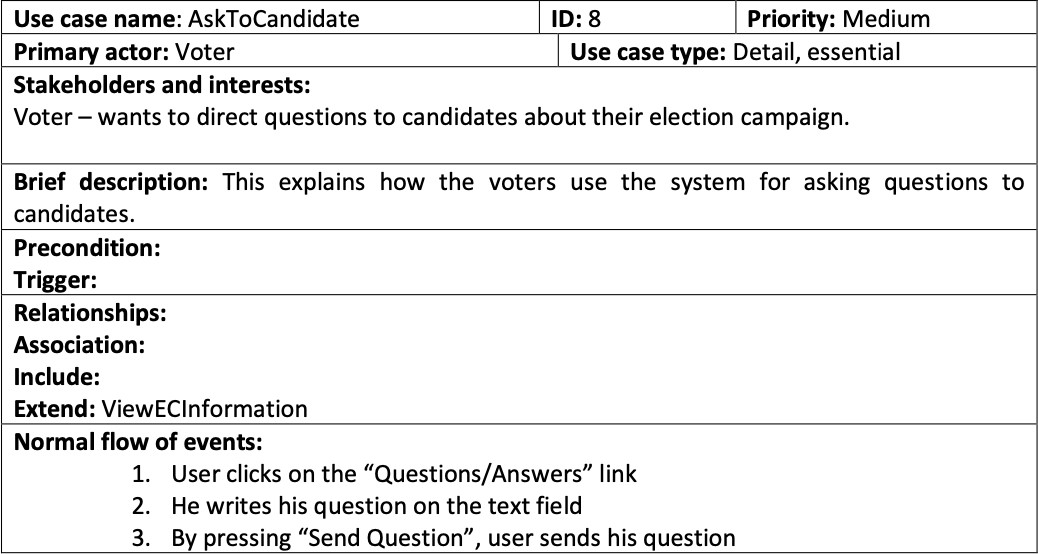


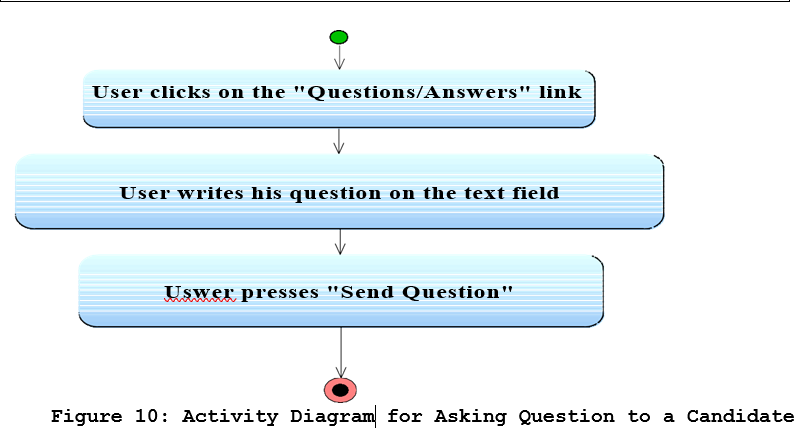
**View Election Candidate Information**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name**: ViewECInformation | **ID:** 7 | | **Priority:** Medium |
| **Primary actor:** Voter | | **Use case type:** Detail, essential | |
| **Stakeholders and interests:**  Voter – wants to see the candidates’ profiles in his own election region. | | | |
| **Brief description:** By using this function the voters can reach information about the candidates’ CVs, promises and answers to asked questions. | | | |
| **Precondition:** -The voter should be already registered to the system  -The voter should have logged in to the system  -Account of the EC should be activated by himself.  **Trigger:** | | | |
| **Relation ships: Associa tion: Include: Extend:** | | | |
| **Normal flow of events:**   1. Voter selects the candidate from candidate list by mouse clicking.    1. If the candidate has not activated his profile then there will not be any link to his profile    2. If the candidate’s page is activated then voter clicks on the candidate’s profile link and EC’s profile page is displayed       1. By clicking the “CV” link voter can reach the general information about the EC.       2. By clicking the “promises” link voter can view the EC’selection campaign.       3. By clicking the “Questions/Answers” link voter can view questions/answers and send questions. | | | |

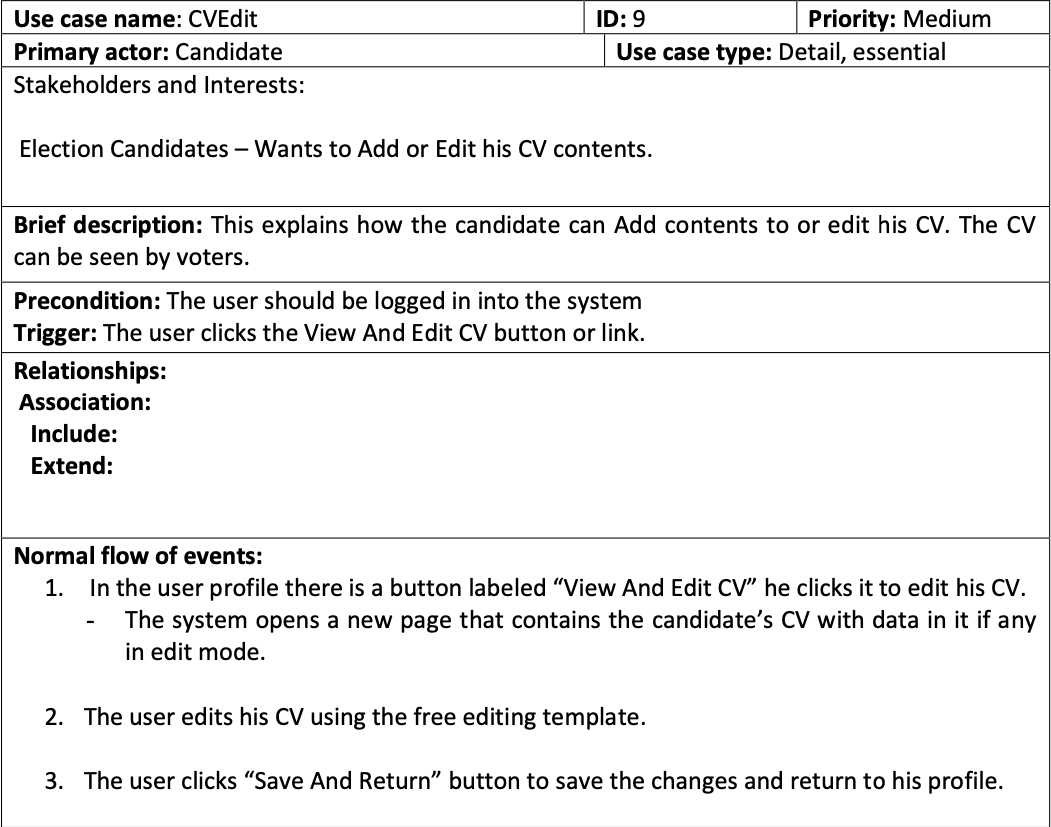


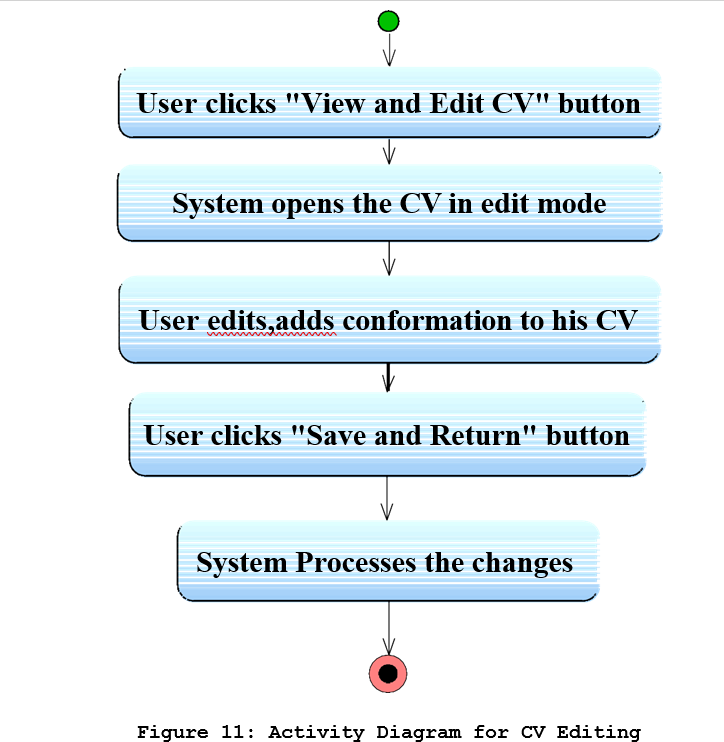
**Ask Question to A Candidate**





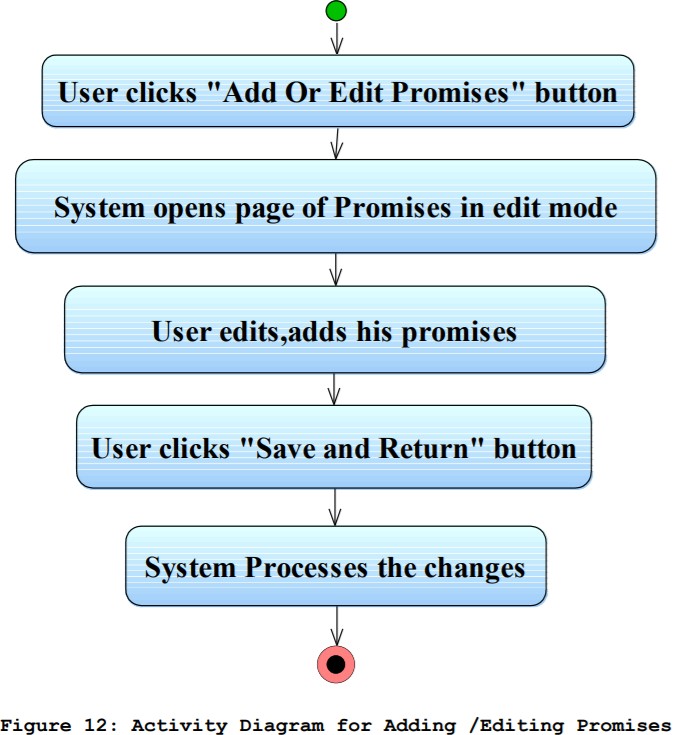
**CV Edit (Candidate)**





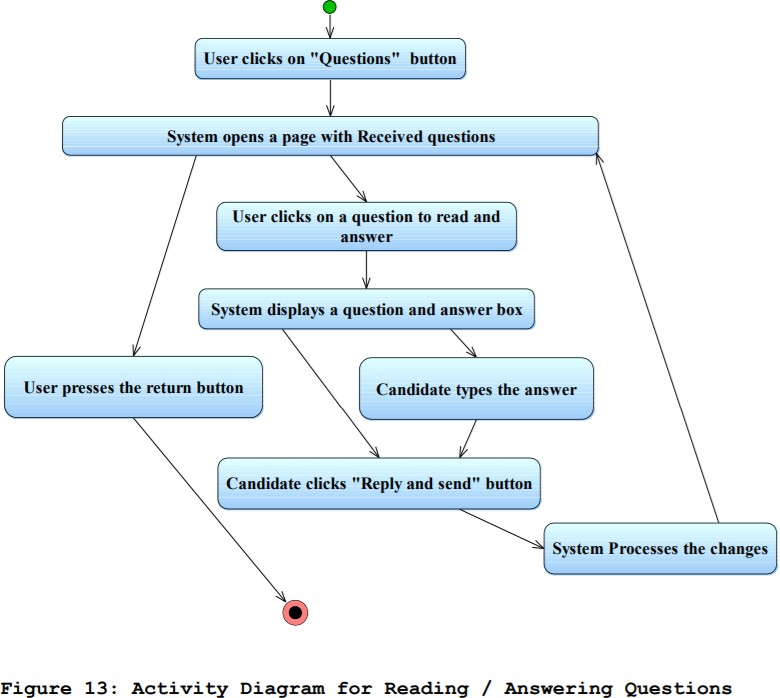
**Add /Edit Promises**

|  |  |  |
| --- | --- | --- |
| **Use case name:** Add/Edit Promises | **ID:** 10 | **Priority:** Medium |
| **Primary actor:** Candidate | | **Use case type:** |
| **Stakeholders and Interests:**  Election Candidates – Wants to Add or Edit his promises, that is, what he promises to do to his people he will lead if they select him. | | |
| **Brief description:** This explains how the candidate can Add or edit his promises. | | |
| **Precondition:** The user should be logged in into the system.  **Trigger:** The user clicks the Add or edit promises button or link. | | |
| **Relationships: Association: Include: Extend:** | | |
| **Normal flow of events:**  1. In the user profile there is a button labeled “Add Or Edit Promises” he clicks it to add or edit his promises.  - The system opens a new page that contains the candidate’s promises with data in it if any in edit mode.   1. The user adds or edits his promises using the free editing template. 2. The user clicks “Save And Return” button to save the changes and return to his profile. | | |



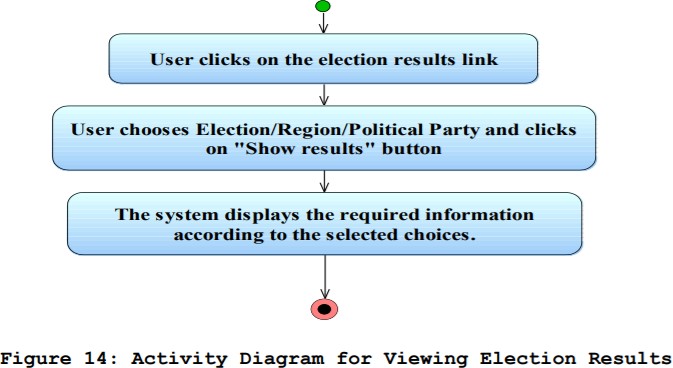
**Read / Answer Questions**

|  |  |  |
| --- | --- | --- |
| **Use case name:** Read/Answer Questions | **ID:** 11 | **Priority:** Medium |
| **Primary actor:** Candidate | | **Use case type:** |
| **Stakeholders and Interests:**  Election Candidates – Wants to Read and/or answer questions from the voters before the election. | | |
| **Brief description:** This explains how the candidate can read and/or write answers to the questions asked by the voter-to-be. | | |
| **Precondition:** The user should be logged in into the system.  **Trigger:** The user clicks the Read and/or Answer questions button or link. | | |
| **Relationships: Association: Include: Extend:** | | |
| **Normal flow of events:**  1. In the user profile there is a button labeled “Questions” he clicks it to read and/or answer the questions.   * The system opens a new page that contains the questions from the voters * If there are questions the candidate clicks on the question to read and answer it on   the provided answer text box.   * User can choose to return to his profile.  1. The user reads and/or answers questions if any. 2. The user clicks ”Reply and send” to save and send anwers of the questions. 3. The system takes the user to the questions page. 4. User clicks on “Return” button to return to his profile. | | |



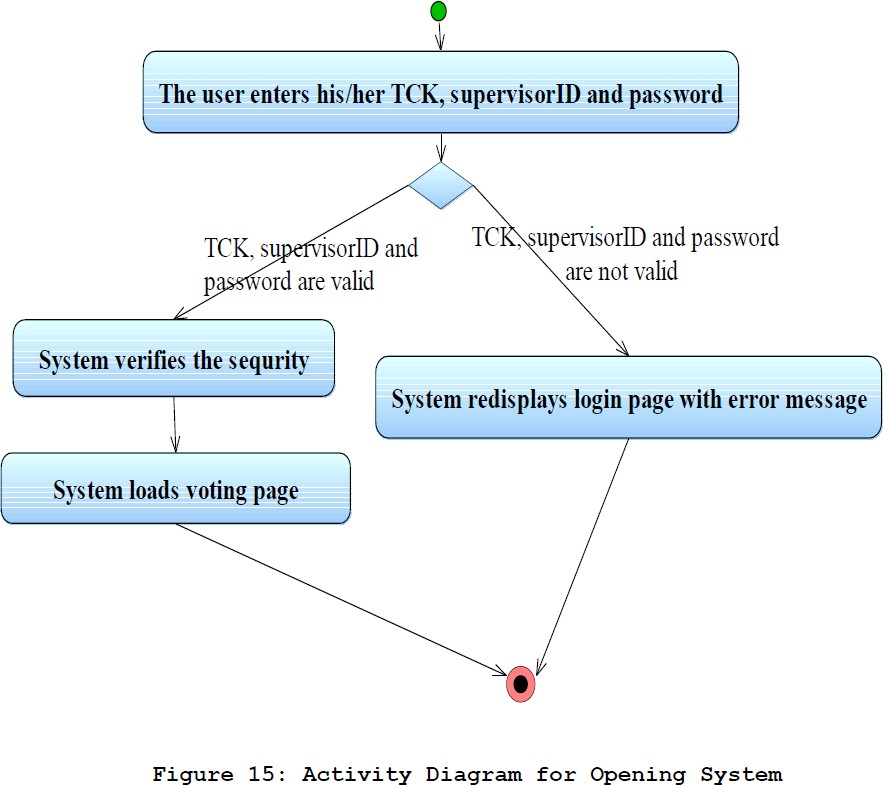
**View Election Results**

|  |  |  |
| --- | --- | --- |
| **Use case name:**  ViewElectionResults | **ID:** 12 | **Priority:** Medium |
| **Primary actor:** Voter | | **Use case type:** Detail, Essential |
| **Stakeholders and Interests:**  General public (Voters, ECs, ESS, etc.) – wants to see the election results. | | |
| **Brief description:** This describes the process of how the voters view the election results by using the system. | | |
| **Precondition: Trigger:** | | |
| **Relationships: Association: Include: Extend:** | | |
| **Normal flow of events:**   1. He clicks on the election results link. 2. He chooses Election/Region/Political Party and presses click on button “show results” 3. The system displays the required information according to the selected choices. | | |



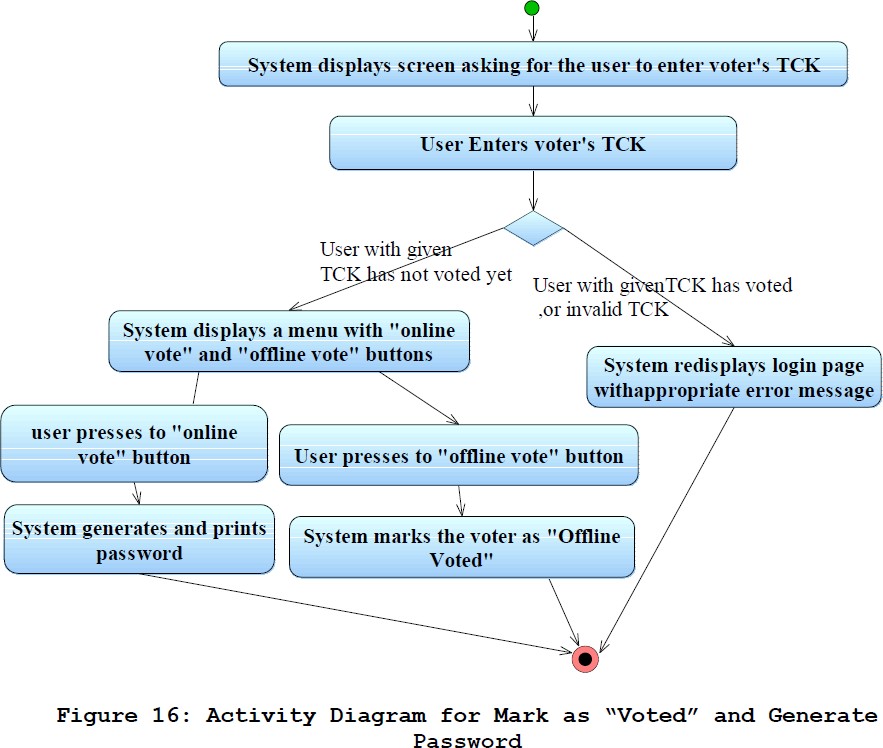
**Election Mode {Open System}**

|  |  |  |
| --- | --- | --- |
| **Use case name:** Open System | **ID:** 13 | **Priority:** High |
| **Primary actor:** User | | **Use case type:** Detail, essential |
| **Stakeholders and Interests:**  Election Station Supervisor – Wants to initiate the system. | | |
| **Brief description:** This use case describes how the supervisor starts the system(s) of the station of his responsibility. | | |
| **Precondition:** The user turns on the system/systems  **Trigger:** The user enters his/her TCK, supervisorID and password | | |
| **Relationships: Association: Include: Extend:** | | |
| **Normal flow of events:**  The user enters his/her TCK, supervisorID and password   1. If the TCK, supervisorID and password is valid, a session is opened.    1. The security is verified.    2. The voting page is loaded. 2. If the TCK, supervisorID and password is not valid, the login screen is redisplayed with an error message. | | |



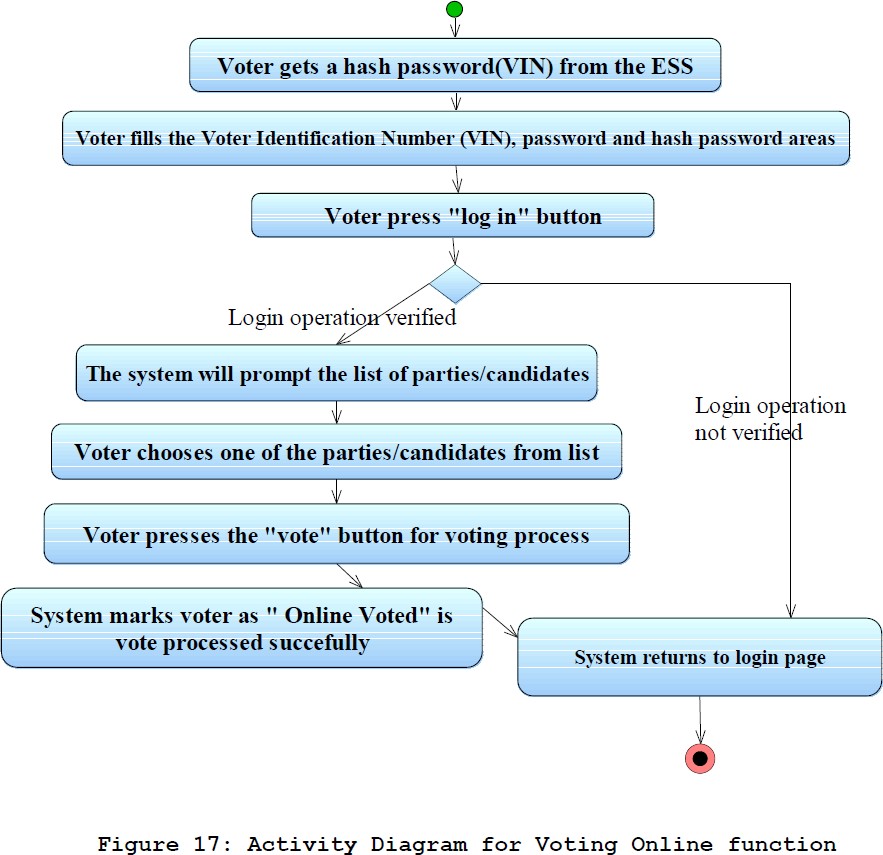
**Mark Voted and Generate Password**

|  |  |  |
| --- | --- | --- |
| **Use case name:**  MarkGeneratePassword | **ID:** 14 | **Priority:** High |
| **Primary actor:** User | | **Use case type:** Detail, essential |
| **Stakeholders and Interests:**  Election Station Supervisor – Wants to generate a password for a voter and check and mark him/her as “Has Voted”. | | |
| **Brief description:** This describes how ESS checks voter’s voting condition and mark him/her as “Has Voted” and if voter wants to use the online system generate a password for the voter to be used in voting. | | |
| **Precondition:** The ESS opened the system.  **Trigger:** The user enters voter’s TCK. | | |
| **Relationships: Association: Include: Extend:** | | |
| **Normal flow of events:**   1. A screen displays asking for the user to enter voter’s TCK. 2. User Enters voter’s TCK.    1. If the voter with specified TCK has not voted yet.       1. The menu is appears with “online vote” and “offline vote” buttons.          1. If user presses to “online vote”, the password is generated and printed.          2. If user presses to “offline vote”, the voter marked as “ Offline Voted” by the system.    2. If the voter with specified TCK has voted or wrong TCK entered.       1. The login screen is redisplayed with appropriate error message. | | |



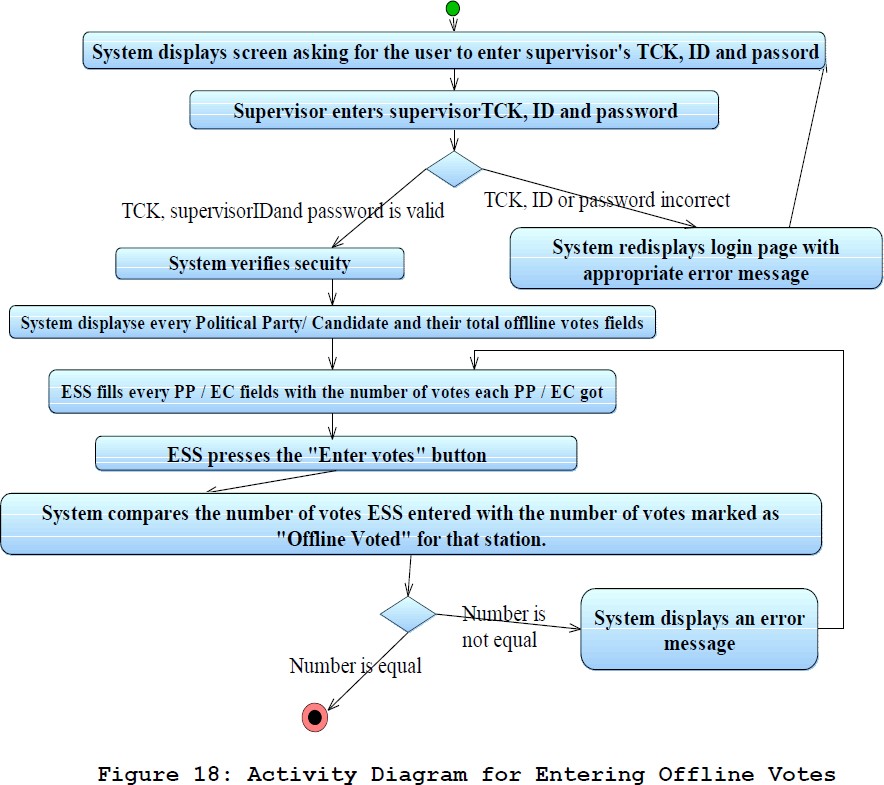
**Vote Online**

|  |  |  |
| --- | --- | --- |
| **Use case name:** Vote Online | **ID:** 15 | **Priority:** High |
| **Primary actor:** Voter | | **Use case type:** Detail, essential |
| **Stakeholders and Interests:**  Voter – wants to use his vote by using system. | | |
| **Brief description:** This explains voting process by using the system. | | |
| **Precondition: Trigger:** | | |
| **Relationships: Association: Include: Extend:** | | |
| **Normal flow of events:**   1. Voter gets a hash password from the ESS. 2. Voter fills the Voter Identification Number (VIN), password and hash password areas. 3. Voter press “log in” button.    1. If the login operation is not verified the system prompts an error message and returns to login page.    2. If login operation is verified       1. The system will prompt the list of parties/candidates.       2. Voter chooses one of the parties/candidates from list.       3. He presses the “vote” button for voting process.       4. If the operation is successful, voter marked as “Online Voted” by the system.       5. System automatically returns to the log in page. | | |



**Enter Offline Votes**

|  |  |  |
| --- | --- | --- |
| **Use case name:** EnterOfflineVotes | **ID:** 16 | **Priority:** High |
| **Primary actor:** Election Station Supervisor | | **Use case type:** Detail, essential |
| **Stakeholders and Interests:**  Election Station Supervisor – Wants to enter offline vote results to the system | | |
| **Brief description:** This describes how ESS enters offline vote results to the system immediately after the voting period. | | |
| **Precondition:** The ESS opened the system.  **Trigger:** | | |
| **Relationships: Association: Include: Extend:** | | |
| **Normal flow of events:**   1. A screen displays asking for the user to enter supervisorID and password. 2. Supervisor enters supervisorID and password.    1. If the TCK, supervisorID and password is not valid, the login screen is redisplayed with an error message.    2. If the TCK, supervisorID and password are valid:       1. The secuity is verified.       2. The screen displaying every Political Party/ Candidate and their input fields is opened.       3. The ESS fills every PP / EC fields with the number of votes each PP / EC got and presses button “Enter Votes”.       4. The system compares the number of votes ESS entered with the number of votes marked as “Offline Voted” for that station. 3. If equality holds the system stores given values to the system and the main page is displayed. 4. If equality does not match, the screen displaying every Political Party/ Candidate and their input fields is reopened with en error message. | | |



**Non-functional Requirements**

**{Performance Requirements}**

The system is expected to have reasonable short time response. The voter should be able to login and should be able to get response for his requests in 2-3 seconds.

The system’s performance is different according to its mode

**In Election Mode**: The system is expected to serve a maximum of up to 50000 voters instantly. This shows that the system should be able to handle about 2000 transactions each second. In addition, the system must be working at 100% peak efJiciency during the voting process.

In Normal Interactive Mode: The system in this mode is expected to serve maximum of up to 50000 voters, but each voter can be active for a long time.

## Security Requirements

* The data transaction between client and server must be encrypted using SSL technology.
* All the passwords that are generated or accepted must be stored in database in an encrypted form.
* To prevent attacks the system should generate random word and ask the user to enter it correctly for multiple tryings.
* In election mode, the different password should be generated for a TCK in every different election.

## 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 detects any security problem in the system, he should be able to shut down the system and prevent all connection to the server immediately to preserve already polled votes.
* The system should be able to recover itself from previous crashes and continue the voting process.
* The system should warn ECA users about the malfunction of the system.

## Other

* JAVA EE is used for development of the system
* Tools that are used for development and deployment of the system:
* ROSE / RSA / WebSphere Modeler for modeling and prototyping the system
* IDEs – Eclipse, RAD, Lotus Forms Designer, Portlet Factory
* Server applications: WebSphere Portal, WAS, WAS CE, WPS
* Linux is the system’s OS.

## Data Model and Description

* We can classify our data objects and their main attributes as follows.
* Login: UserId, Password.
* UserList: Voters [], ECs[], ECAs[], ESSs.
* User: Name, Address, VotingCenterId, Age, Sex, TCK, userID, password
* CandidateVotes: earntVotes, totalVotes, percentage, rank
* Candidate: PoliticalParty, RullingArea, position
* ESS: userId, password
* Stations: stationId, ESS

## Relationship

Associations

The following object relationships show association in ONEV system

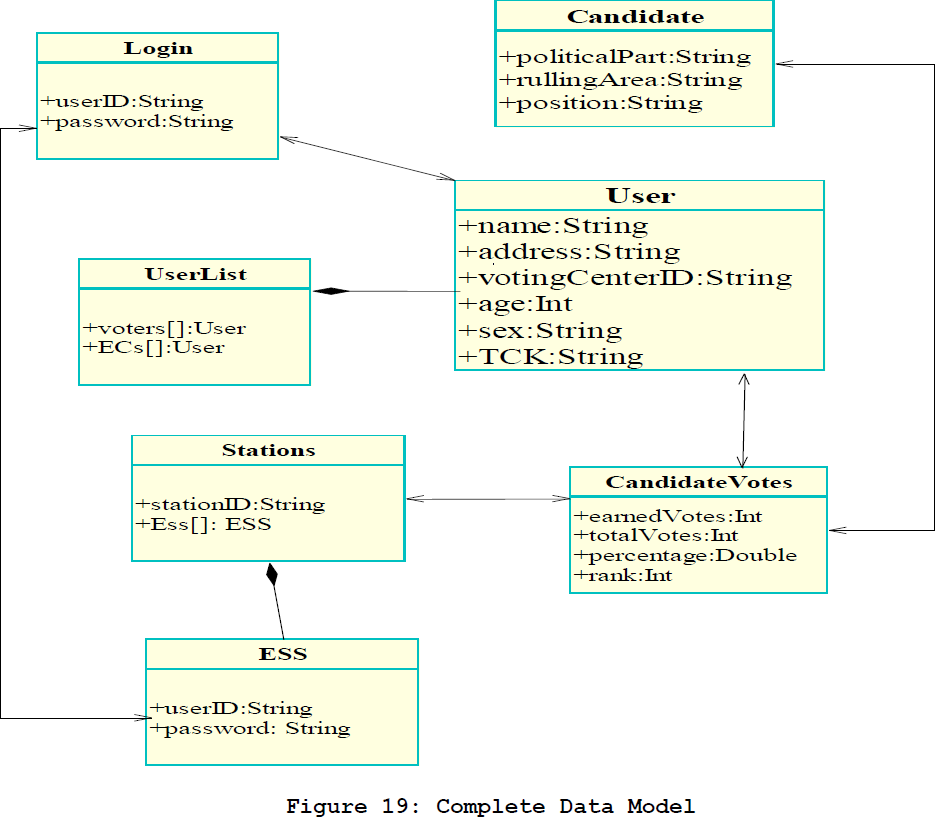
* + Login and User
  + Login and ESS
  + Candidate and CandidateVotes
  + User and CandidateVotes
  + Stations and CandidateVotes Composition

The following object relationships show composition in ONEV system

* + User and UserList
  + ESS and Stations

## Complete Data Model

This diagram shows the data objects with relationships among each other.



## Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Description** | **Field Type** | **Field Length/ Maximum Number** |
| **userID** | This is a unique user identiJication word which is unique to every registered user. | String (of characters) | 20 |
| **password** | A password for every user to log in into the system. | String (of characters) | 30 |
| **voters** | An array of voters registered for the coming election. | Numeric | 70,000,000 |
| **electionCandid ates** | An array of election candidates registered for the coming election. | Numeric | 100,000 |
| **earnedVotes** | Keeps the total votes a candidate has got from voters. | Numeric | 70,000,000 |
| **totalVotes** | Keeps the total votes in a given region of a candidate given by voters. | Numeric | 70,000,000 |

**Description of Software Behavior**

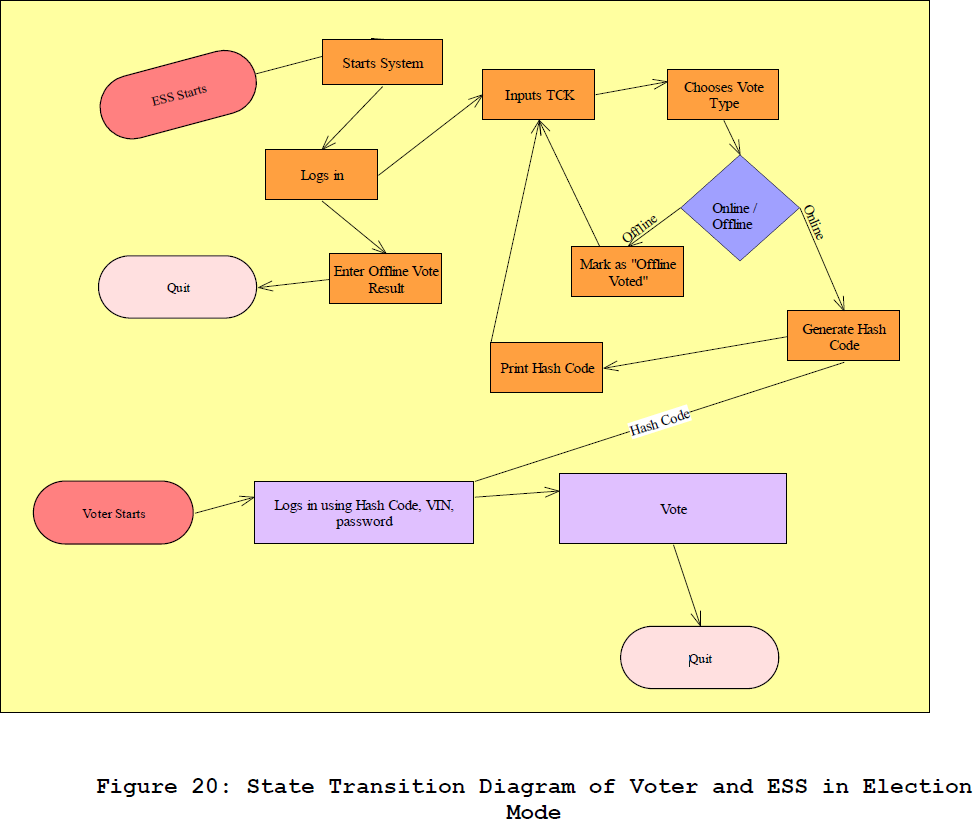
* Election Mode

System can be used only for aim of voting, voter cannot see any daily activity of the system as in ordinary day. After the voter login to system, system will give hash code to voter will use this code to vote. Giving hash code is valid only in the Election Day mode. System will show only political parties after login with hash code.

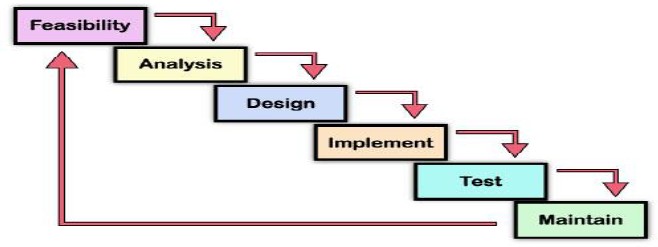
* Normal Interactive Mode

The system is available to everywhere. The voter who has already VIC can use system. System in ordinary day behavior has all functions which are explained in the part of functional requirements except the voting function.

## State Transition Diagrams



**Process Model**



**Figure 21: Waterfall Model showing the step by step activities in our software development processes**

## Conclusion

This SRS document is prepared for a better design of Online National Election Voting system. The functional and other requirements of the system are described and the needs of the users are stated thought the document.

## References

1. J. Mohen, and J. Glidden, “The Case for Internet Voting,” CACM, Vol. 44, No. 1, pp. 72- 85, Jan.2001.
2. Chevallier, M.: Internet voting: Status; perspectives and Issues, ITU E-Government Workshop Geneva, 6 June 2003
3. D. P. Gilliam, T. L. Wolfe, J. S. Sherif, and M. Bishop, “Software Security Checklist for the Software Life Cycle,” in Proc. WETICE’03, 2003, pp. 243-248.
4. A. D. Rubin, “Security Considerations for Remote Electronic Voting,” CACM, vol. 45, pp. 39-44, Dec. 2002.
5. IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements SpeciJication

J. Peters, and W. Pedrycz, Software Engineering – An Engineering Approach. New York, NY: Wiley, 2000.