VIT Bhopal University

School of Computer Science & Engineering

VIT®
BHOPAL
www.vitbhopal.ac.in

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

Introduction

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

Problem Definition

Manual voting system has been deployed 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 fulfill 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 field areas by using the system if they prefer online voting.

Purpose

The purpose of this document is to make the functional and non-functional requirements of the Online National Election Voting System easy to comprehend. It also serves 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 analysis of the system. The main aim of the system is to provide a set of protocols that allow voters to cast ballots while a group of authorities collect votes and output final results.

Overview

The remainder of this document identifies 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

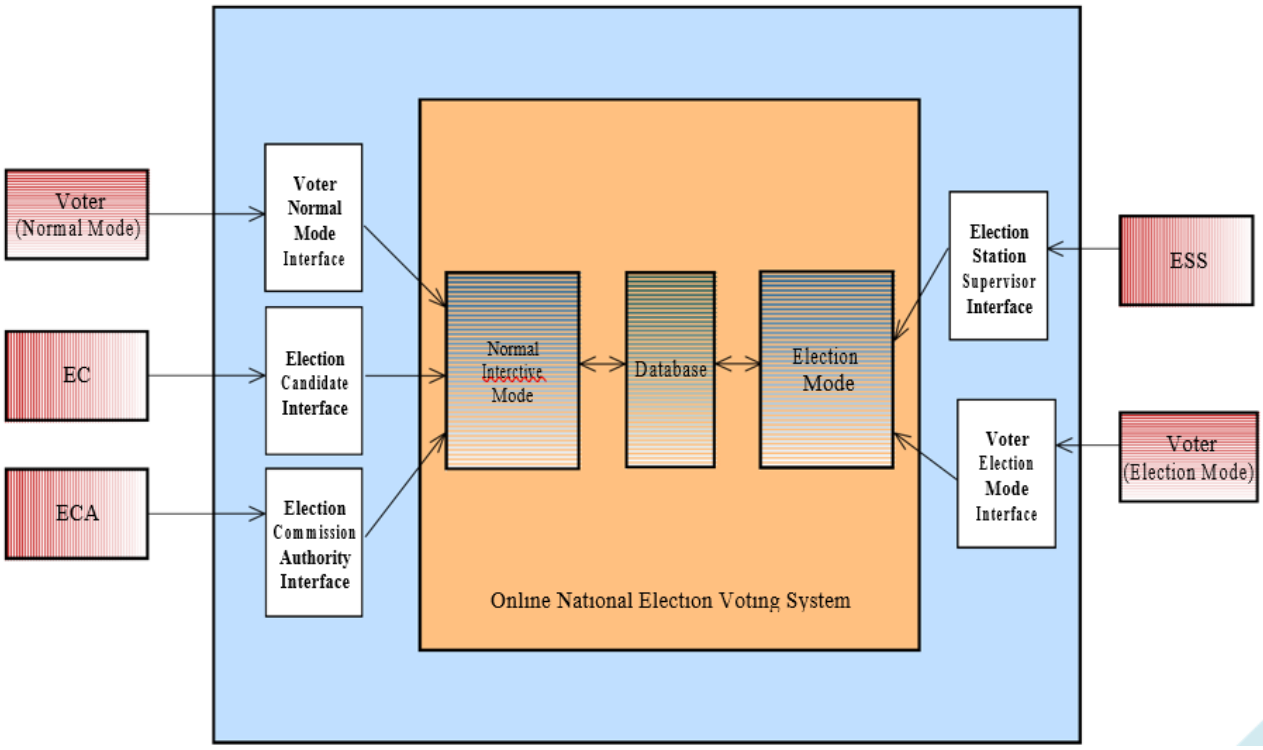
Overall Description

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 software product is a standalone system and not a part of a larger system. 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' profiles 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 the computers 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 the server. The ECA configures 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. The system will be in Election Mode, for the purpose of vote polling only on the Election Day. Normal Interactive Mode is for accepting registrations, discussions between voters and candidates, campaigns and the system is available in this mode all the time except Election Days.

Normal Interactive Mode Voter Registration

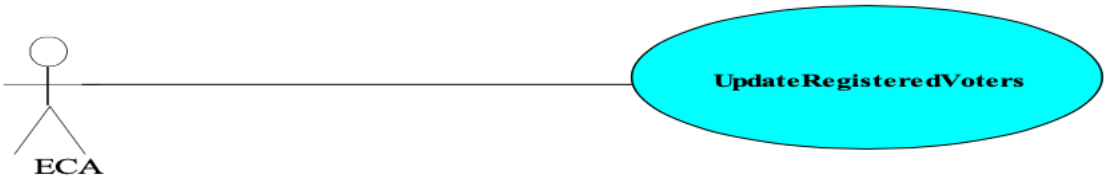
That system will be used only by the people who have been registered to the system. Main actor of the registration operator is the voter. The registration operator is approved by the ECAs.

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.

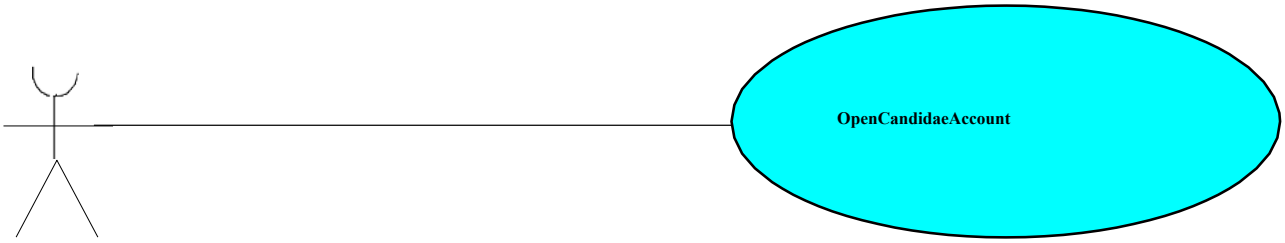
Update Registered Voters

ECA deletes voters from the system who cannot use their vote officially. ECA also updates voter's information. The main actor is the ECA.



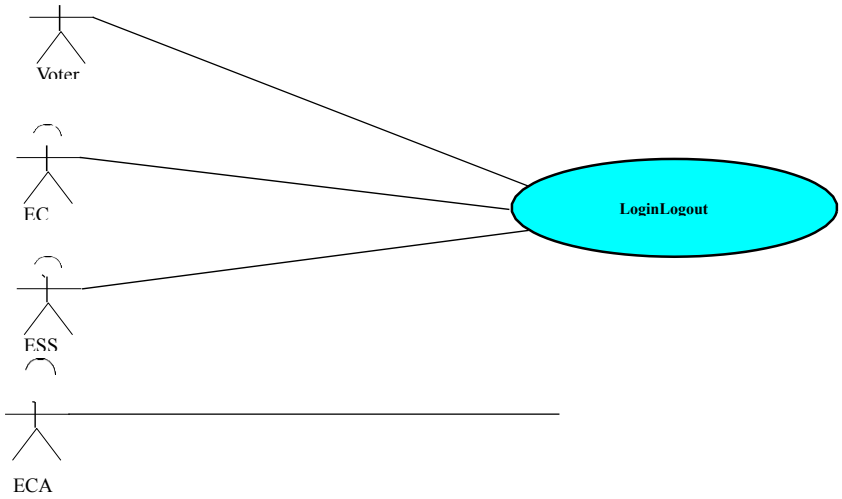
Open Candidate Account

The EC's profile must be created by the ECA. This functionality helps to perform this action. The ECA is the main actor of this functionality.



Login/Logout

All of the system user's login to system by their user ids and passwords. All of the users are the main actor of this use case.

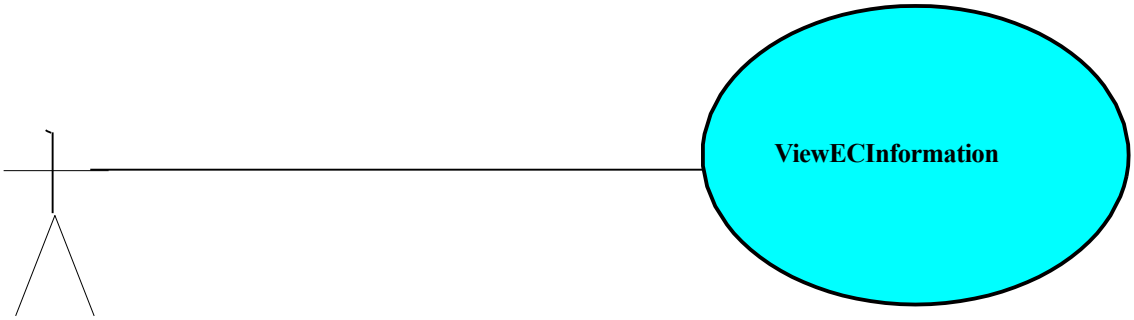


Account Update

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

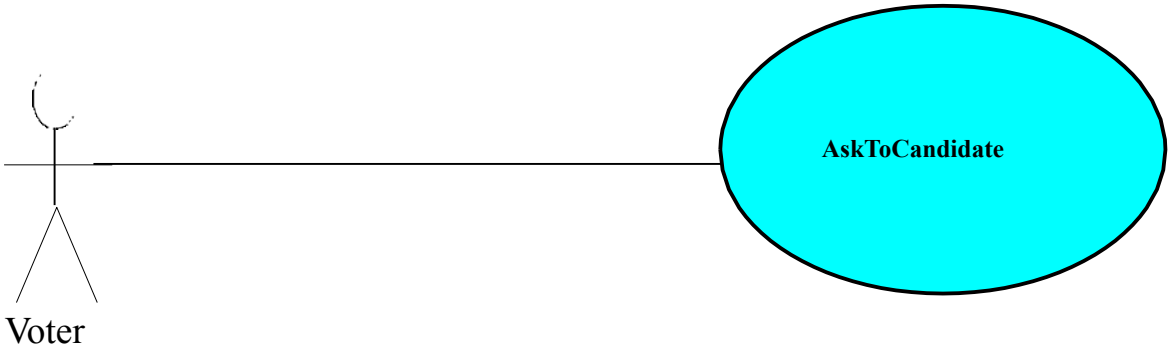
View EC Information

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.



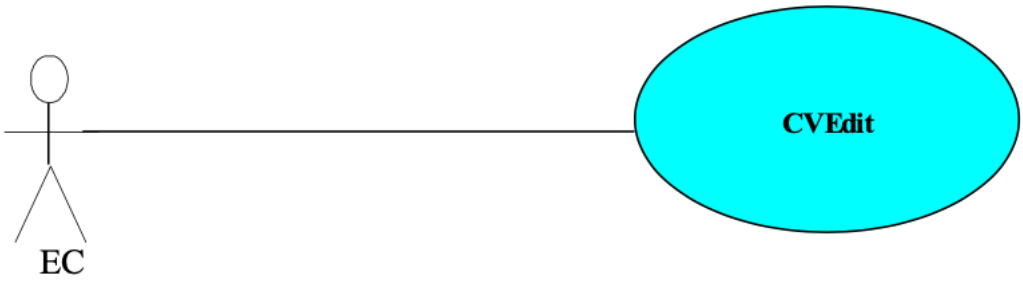
Ask To Candidate

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.



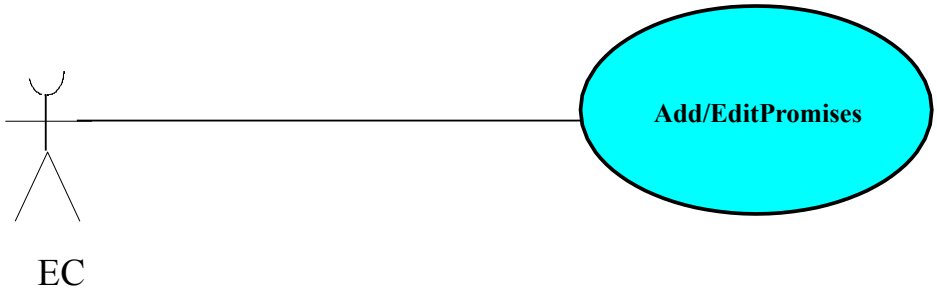
CV Edit

This function provides the EC to edit his CV information on his own profile. The EC is the main actor of that functionality.



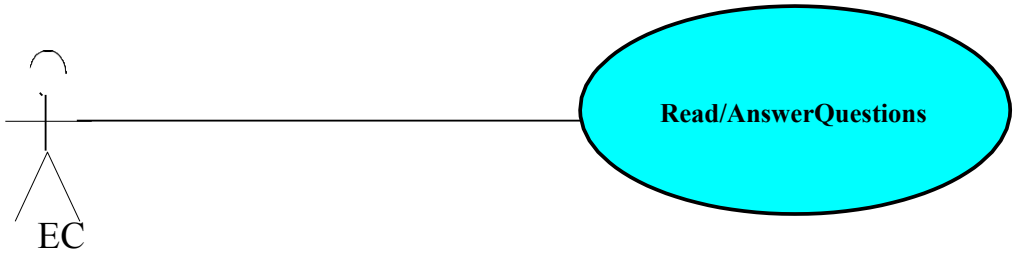
Add / Edit Promises

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



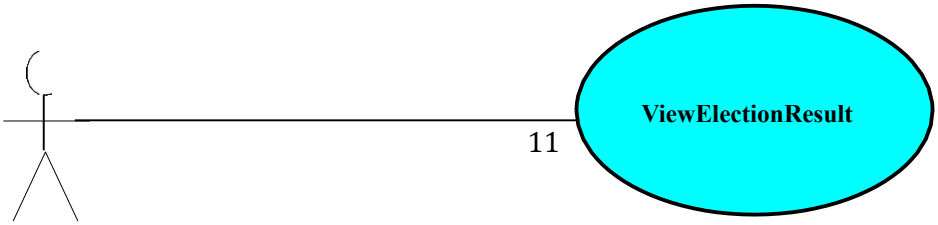
Read/Answer Questions

This function provides ECs to read or answer questions about their election campaigns. The main actor is the EC.



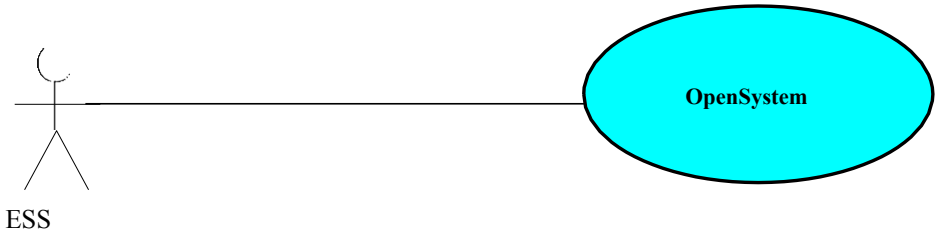
View Election Results

This functionality provides voters to see the current or past years' election result in a proper way. The main actor is the voter.



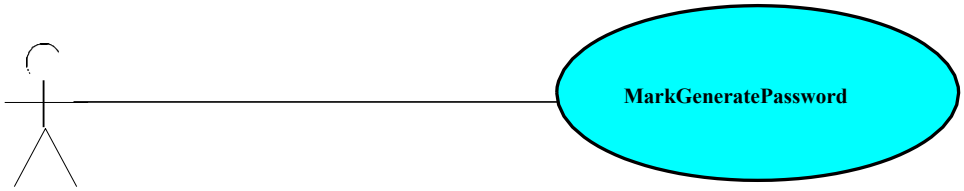
Election Mode {Open System}

This function provides ESS to start the system during the Election Day or before. The ESS is the main actor of this operation.



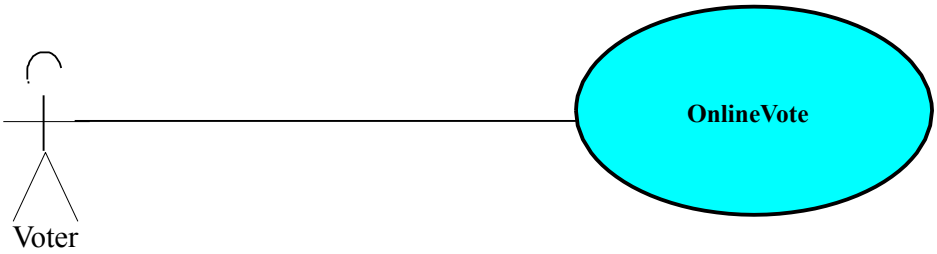
Mark Generate Password

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.



Online Vote

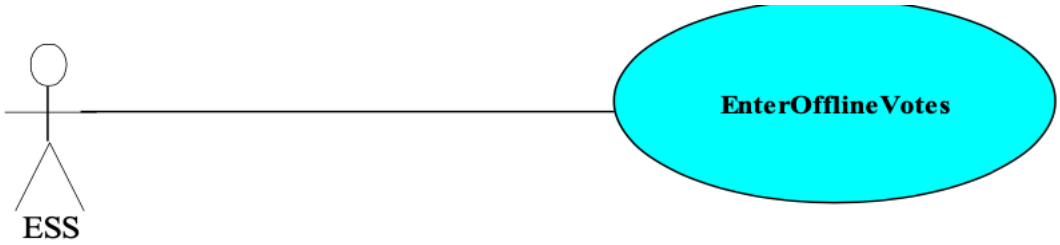
This is the main function of the system that provides online voting for the general public. The main actor is the voter and votes are collected in the DB.



Enter Offline Votes

By using this function the ESSs enters the offline votes to the system. The main actor

of this 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 executed nearly in 150,000 ballot boxes. 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}

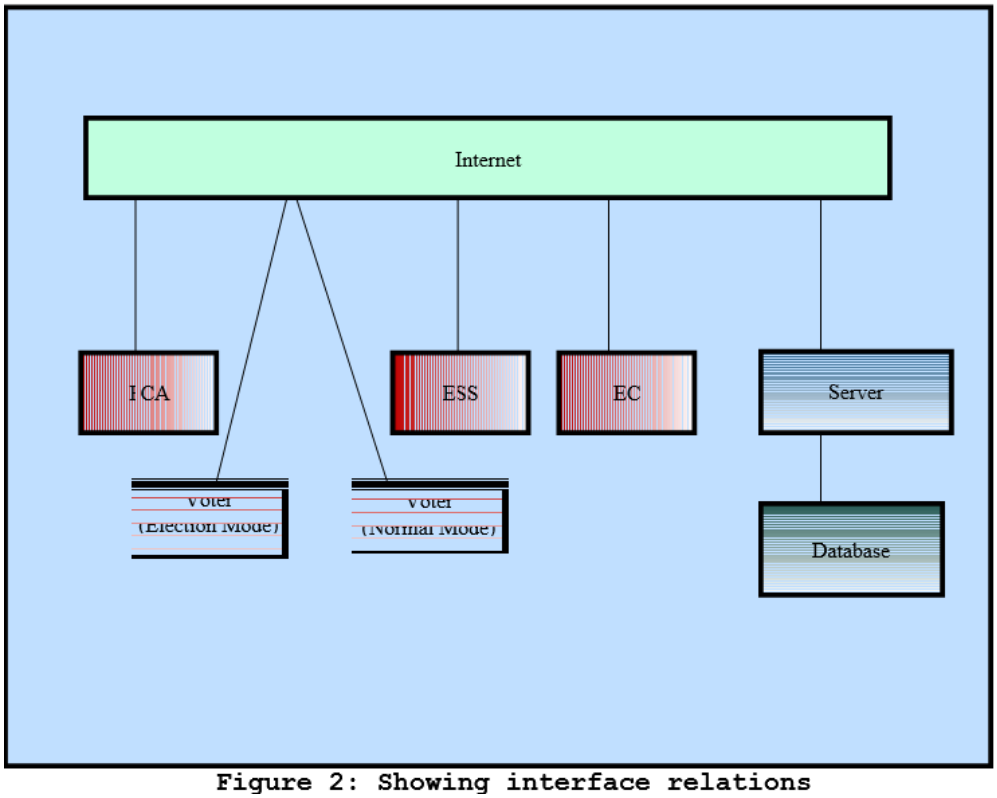


Figure 2: Showing interface relations

User Interfaces

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

Hardware Interfaces

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

Software Interfaces

The poll server runs on http server that is enabled to handle server pages. 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.

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 profile, 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 describe the 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		
Relationships: **Association:** **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. a. If the information is correctly entered the system prints a successful message. b. Otherwise, it prints appropriate error message, redisplays the application form.		

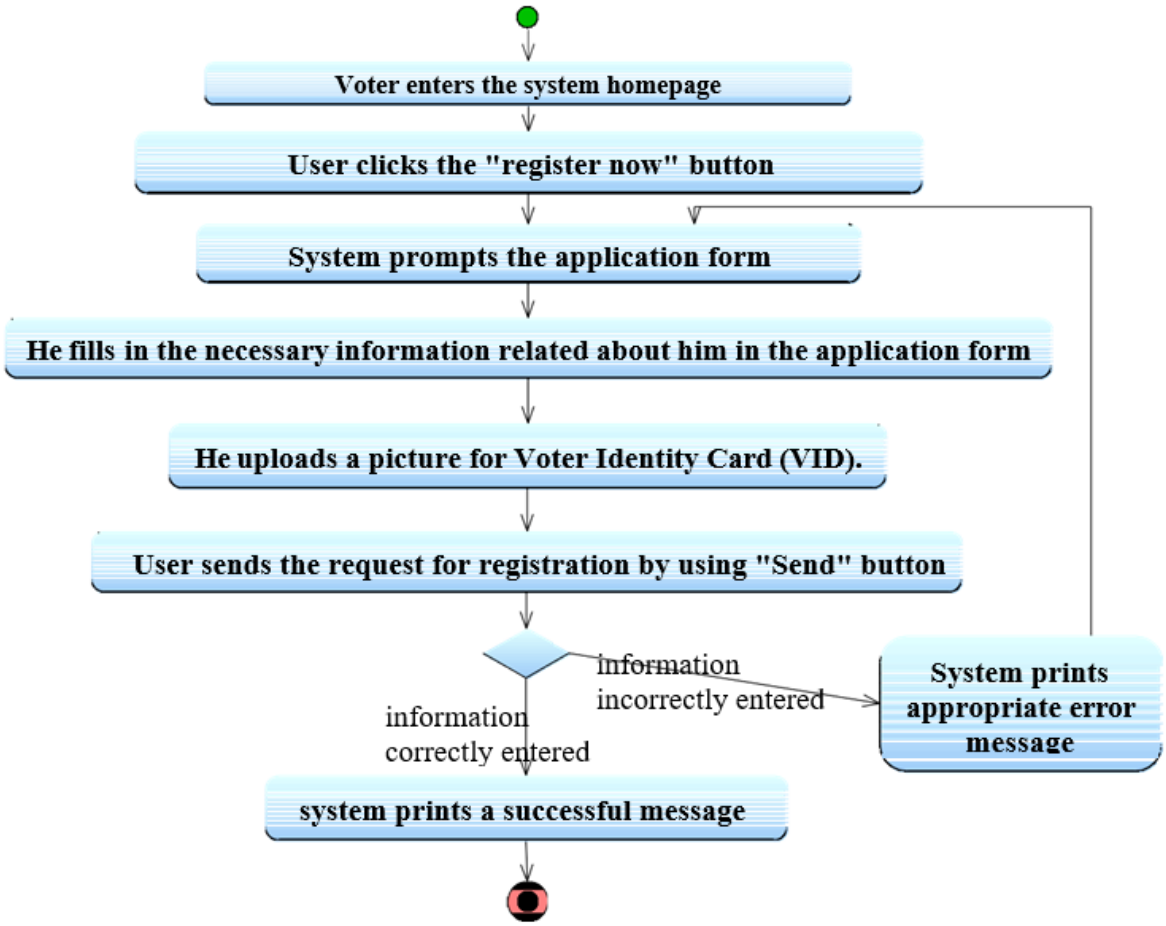


Figure 3: Activity Diagram for Voter Registration

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 filled his application form **Trigger:**		

Relationships: **Association:** **Include:** **Extend:**
Normal flow of events: 1. ECA selects the online voter application form from list 2. ECA checks the information of the applicant a. If the the given information is correct i. ECA approves the form by pressing "Approve" buton ii. ECA generates the new online account to this new voter iii. ECA prepares the VIC and generates password iv. ECA sends VIC and password to adress of voter b. if the given information is not correct i. ECA will inform voter about misinformation via postal mail

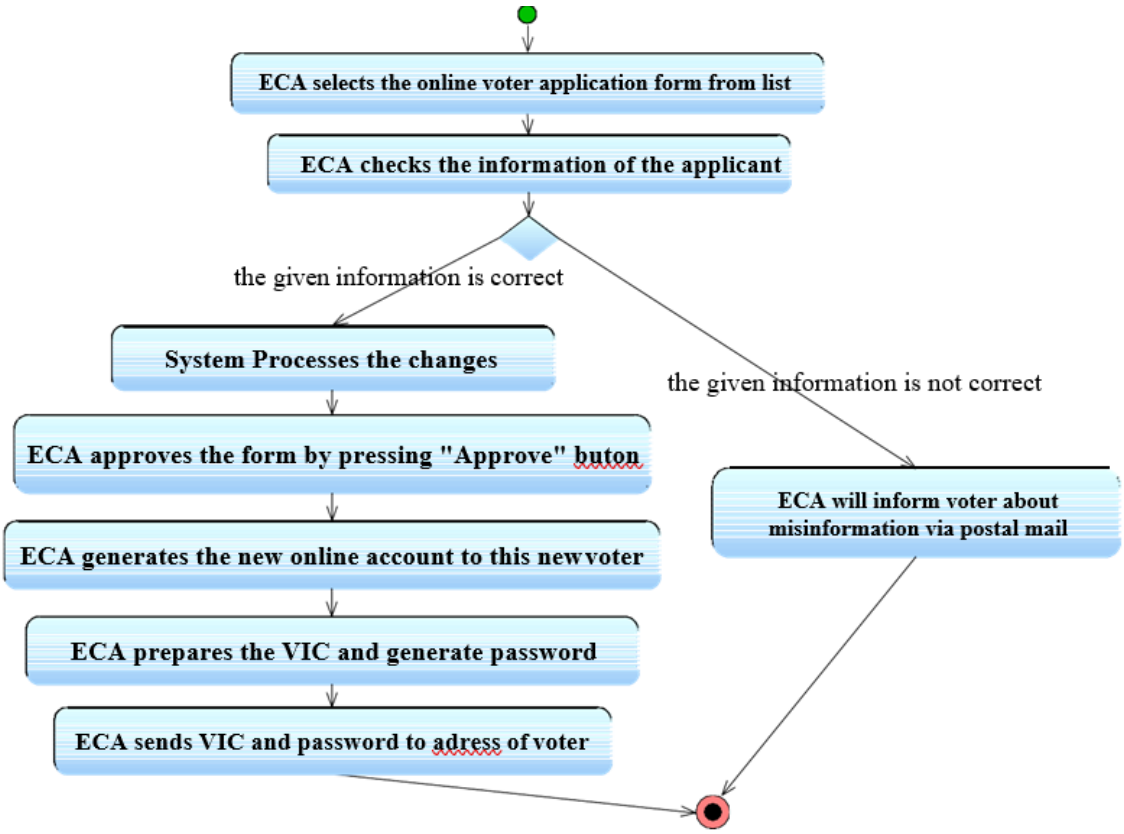


Figure 4: Activity Diagram for Approve Application function

Update Registered Voters

Use case name: UpdateVoters	**ID:** 3	**Priority:** High
Primary actor: ECA	**Use case type:** Detail, essential	
Stakeholders and Interests : ECA : Wants to update all voters according to their state		
Brief description: This describe how ECA updates online voters		
Precondition: **Trigger:**		
Relationships: **Association:** **Include:** **Extend:**		
Normal flow of events: 1. ECA selects on "Update Voters" from menu and displays that page 2. ECA click on "Update Now" button 3. The system checks online voters with respect to upcoming election's voters list a. If the voter exists in the list, the system updates the voter with respect to official the voter information. b. If the voter does not exist in the list, the system deletes that voter from database.		

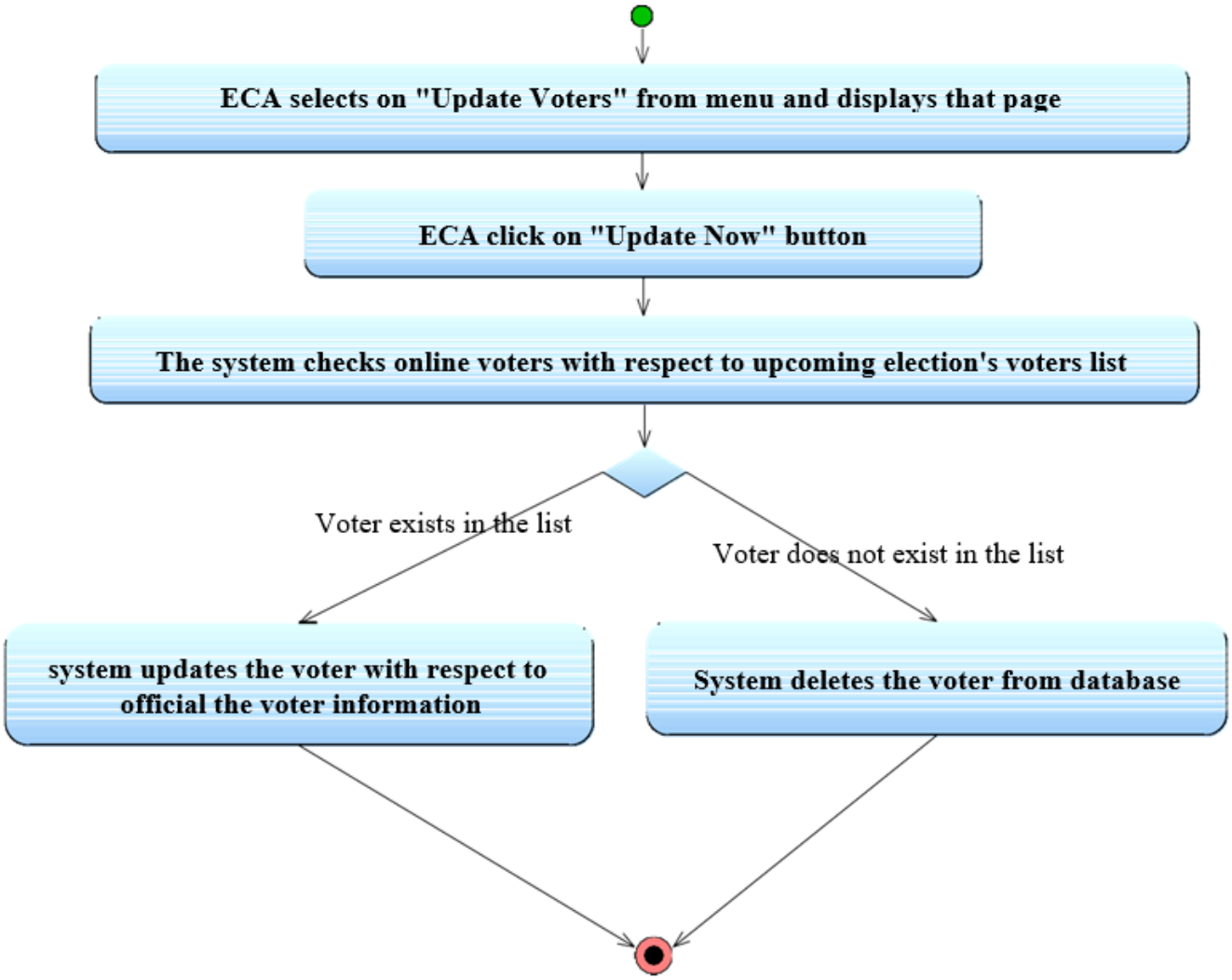
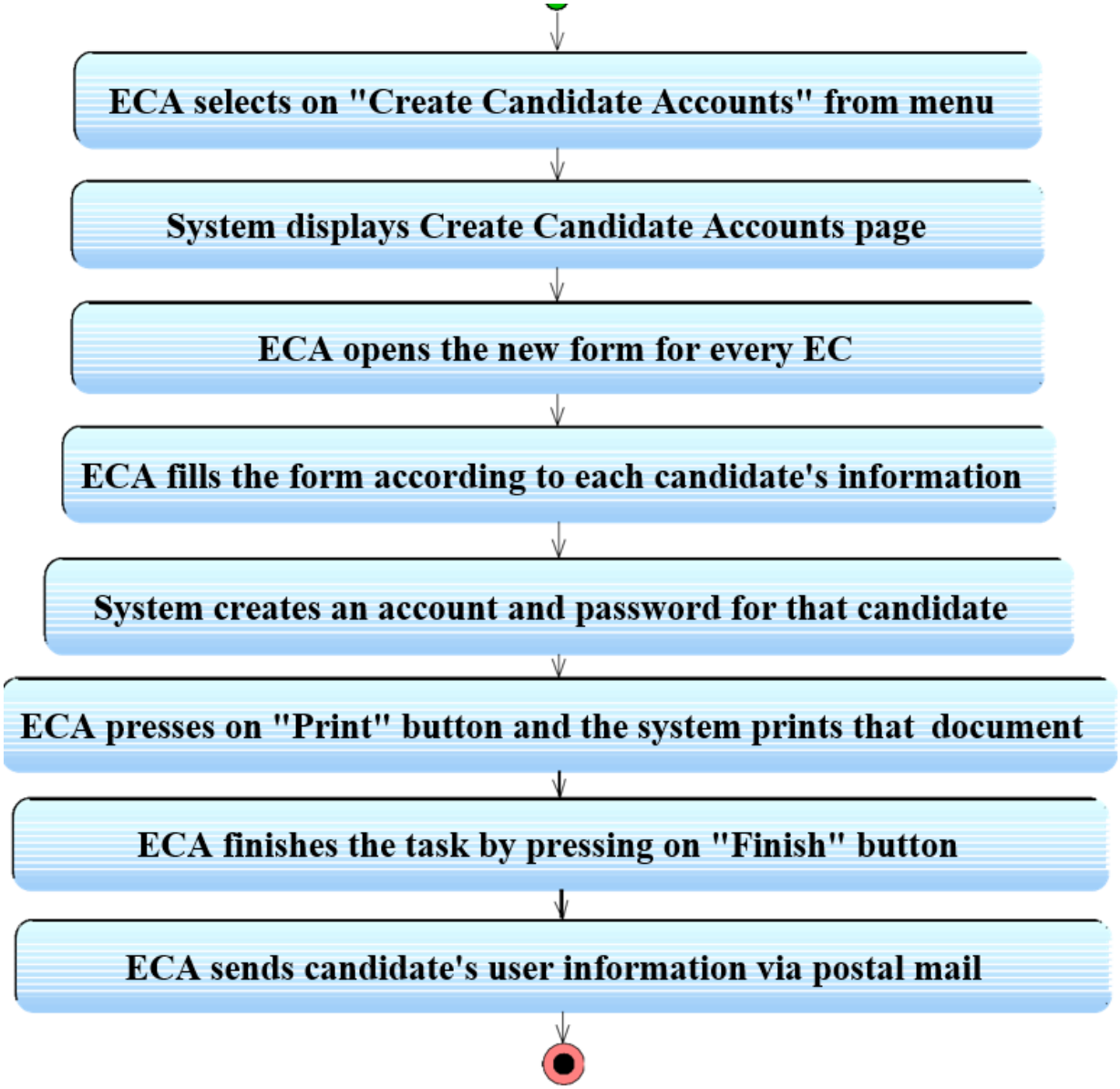


Figure 5: Activity Diagram for Update Registered Voters

Open Candidate Account

Use case name: OpenCandidateAccount	**ID:** 4	**Priority:** High
Primary actor: ECA	**Use case type:** Detail, essential	
Stakeholders and Interests : EC, ECA - ECA generates new accounts to EC's		
Brief description: This describes how ECA will generate all election candidates' new accounts.		
Precondition: **Trigger:**		
Relationships: **Association:** **Include:** **Extend:**		
Normal flow of events: 1. ECA selects on "Create Candidate Accounts" from menu 2. System displays Create Candidate Accounts page 3. ECA opens a new form for every official Candidate 4. ECA fills the form according to candidate's information 5. ECA presses on "Generate password" button 6. System creates an account and password for that candidate 7. ECA presses on "Print" button and the system prints that document 8. ECA finishes the task by pressing on "Finish" button 9. ECA sends candidate's user information via postal mail.		


ECA selects on "Create Candidate Accounts" from menu
System displays Create Candidate Accounts page
ECA opens the new form for every EC
ECA fills the form according to each candidate's information
System creates an account and password for that candidate
ECA presses on "Print" button and the system prints that document
ECA finishes the task by pressing on "Finish" button
ECA sends candidate's user information via postal mail

Log In / Log out

Use case name: Login/Logout	**ID:** 5	**Priority:** High
Primary actor: User	**Use case type:** Detail, essential	
Stakeholders and interests: Voter – Wants to log into the system ECA – Wants to log into the system ECC – Wants to log into the system		
Brief description: This describes how the users log into the system		
Precondition: The user opens the login page **Trigger:** The user enters his id and password		
Relationships: **Association:** **Include:** **Extend:**		
Normal flow of events: 1. The user enters his login id and password **A.** If the login and password is valid, a session is opened i. The security is verified ii. The specific page of every user is loaded **B.** If the login or password is not valid, the login screen is redisplayed with an error message 2. The user click on the logout button i. The session is terminated. ii. The login screen is displayed.		

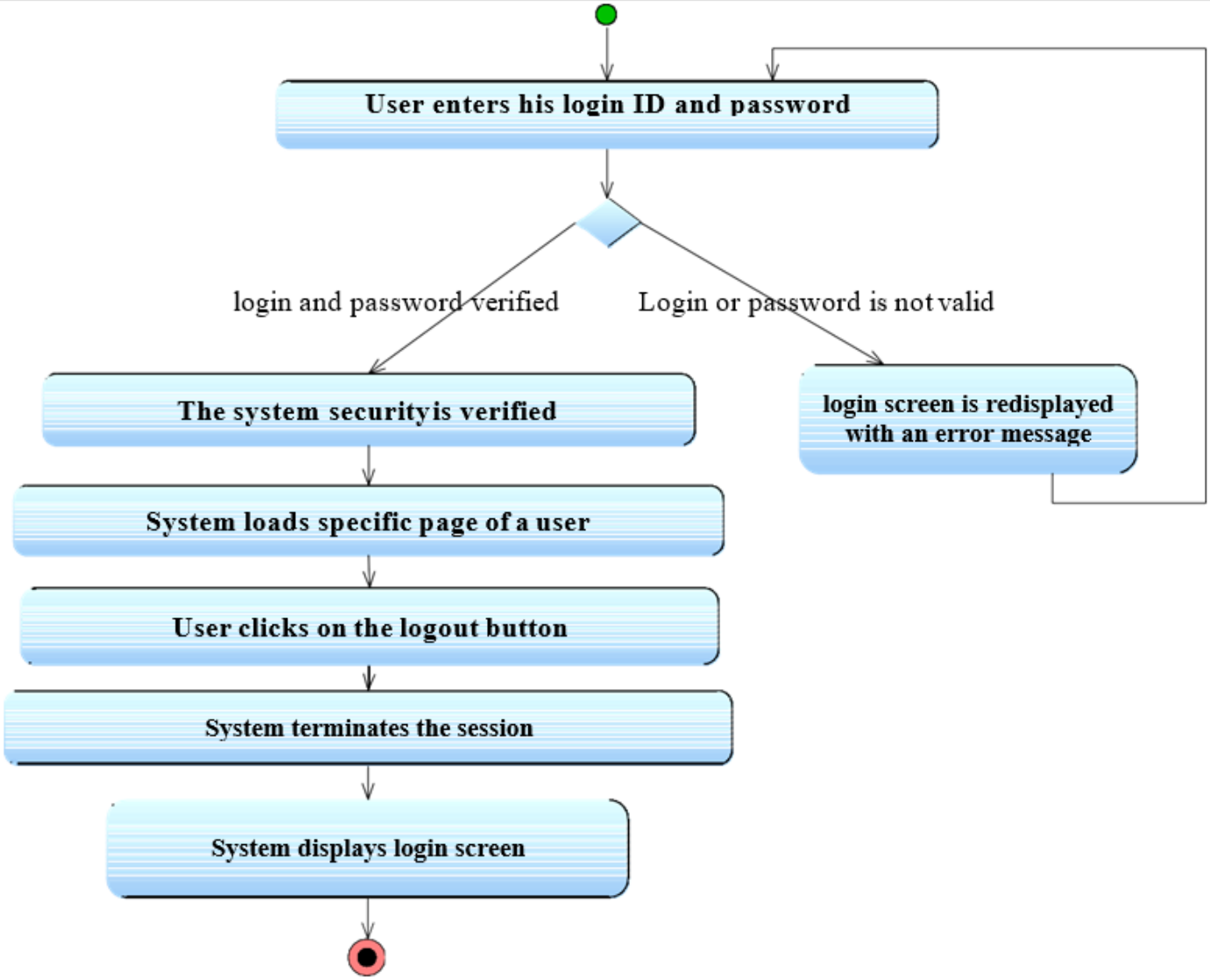


Figure 7: Activity Diagram for Log in / Log out function

Update Account

Use case name: AccountUpdate	**ID:** 6	**Priority:** Medium
Primary actor: User	**Use case type:** Detail, essential	
Stakeholders and Interests: Election Candidates – Wants to change password Voter - Wants to change password		
Brief description: This explains how the candidate can change his password.		
Precondition: The user should be logged in into the system **Trigger:** The user clicks the "Update Account" button.		
Relationships: **Association:** **Include:** **Extend:**		
Normal flow of events: 1. In the user profile there is a button labeled "Update Account" he clicks it to update his account. - The system opens a new page to enter old password and the new one. 2. The user enters his old password. He then enters his new password 3. User clicks the "submit" button. - If the old password was entered incorrectly, the system will print an error message and the form to change password will re-appear. - If the old password was entered correctly the system changes password and prints a success message and redirects to his profile.		

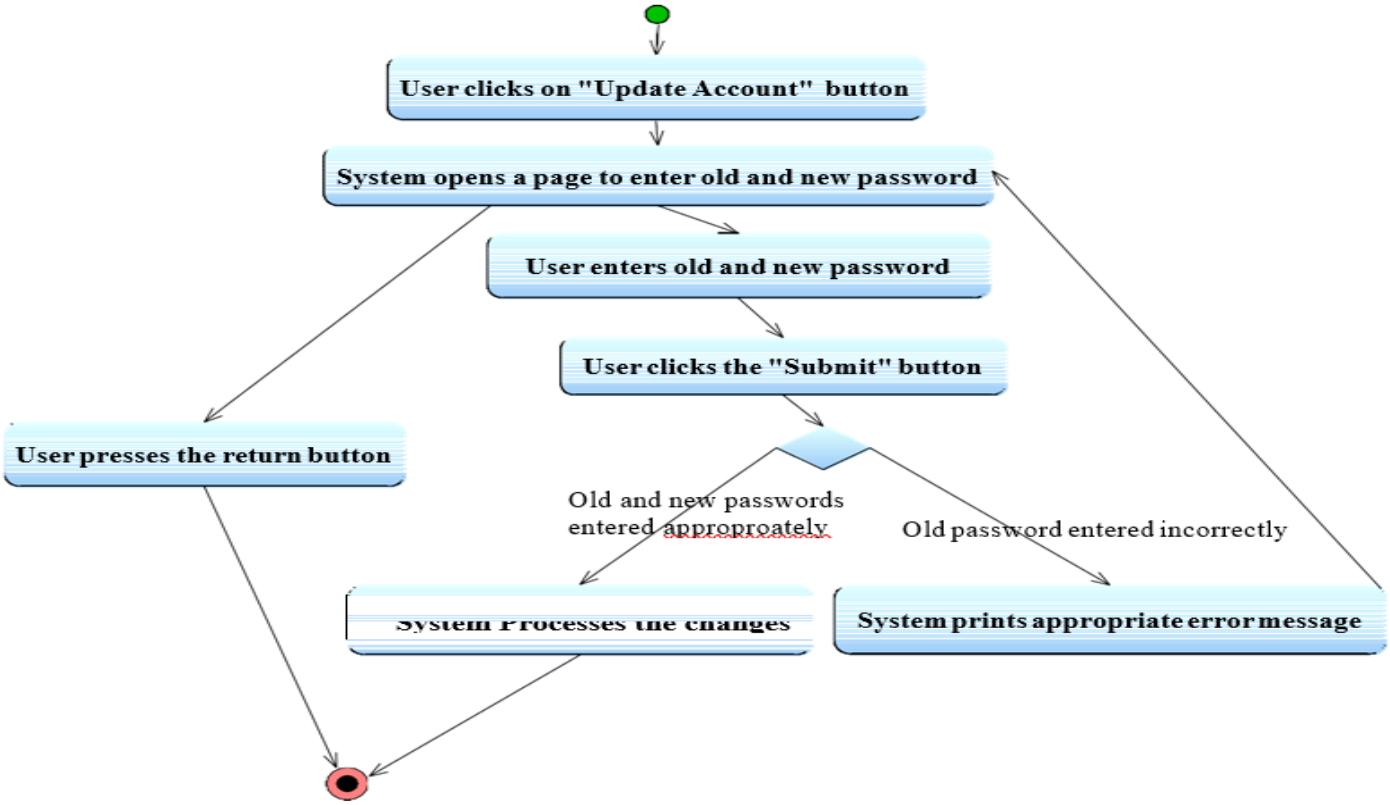


Figure 8: Activity Diagram for Updating Account function

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:

Relationships:
Association:
Include:
Extend:

Normal flow of events:

1. Voter selects the candidate from candidate list by mouse clicking.
   A. If the candidate has not activated his profile then there will not be any link to his profile
   B. If the candidate's page is activated then voter clicks on the candidate's profile link and EC's profile page is displayed
      i. By clicking the "CV" link voter can reach the general information about the EC.
      ii. By clicking the "promises" link voter can view the EC's election campaign.
      iii. By clicking the "Questions/Answers" link voter can view questions/answers and send questions.

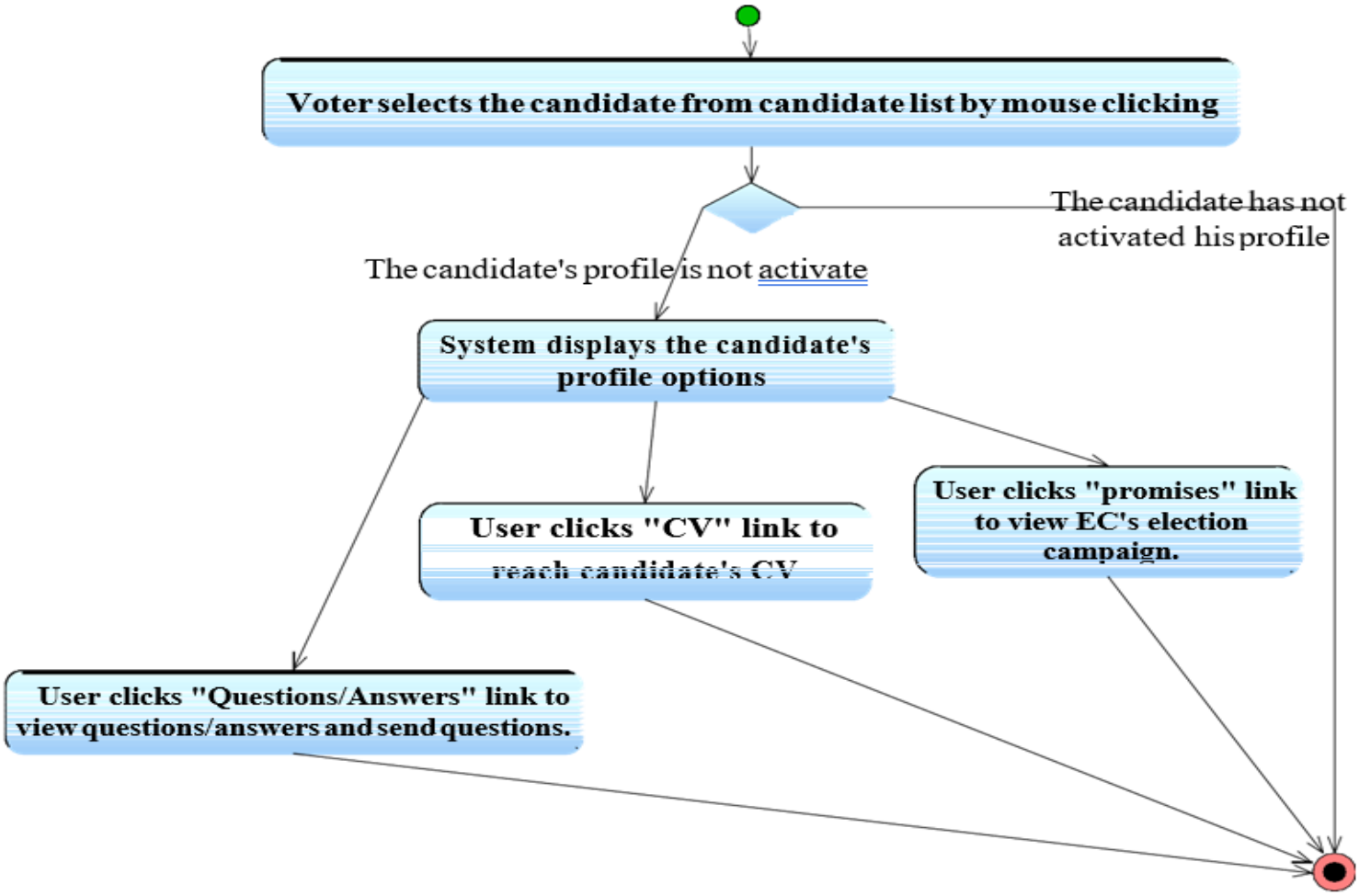


Figure 9: Activity Diagram for Viewing Election Candidate Information

Ask Question to A Candidate

Use case name: AskToCandidate	**ID:** 8	**Priority:** Medium
Primary actor: Voter	**Use case type:** Detail, essential	
Stakeholders and interests: Voter – wants to direct questions to candidates about their election campaign.		
Brief description: This explains how the voters use the system for asking questions to candidates.		
Precondition: **Trigger:**		
Relationships: **Association:** **Include:** **Extend:** ViewECInformation		
Normal flow of events: 1. User clicks on the "Questions/Answers" link 2. He writes his question on the text field 3. By pressing "Send Question", user sends his question		

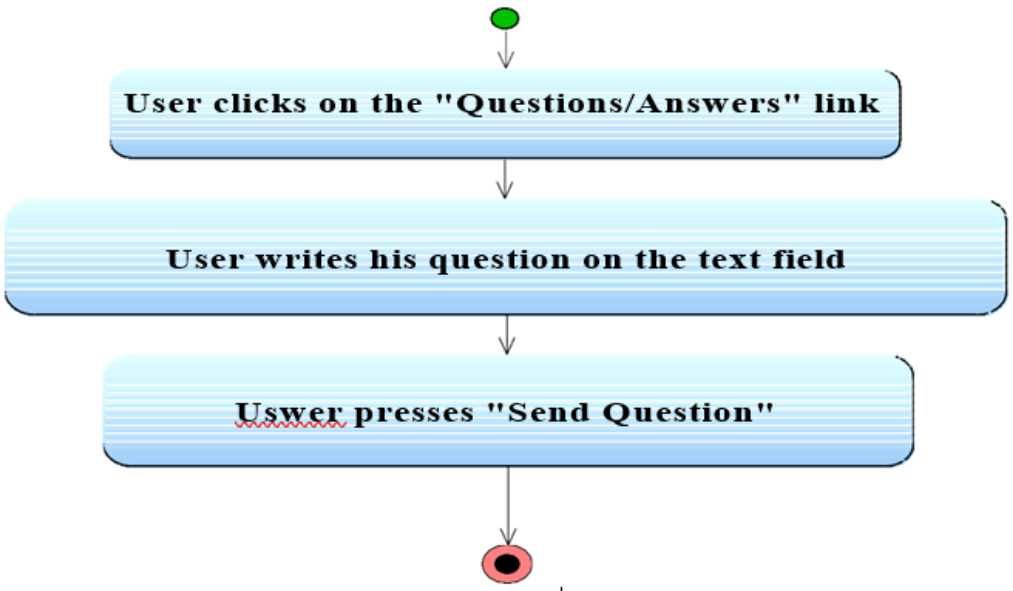


Figure 10: Activity Diagram for Asking Question to a Candidate

CV Edit (Candidate)

Use case name: CVEdit	**ID:** 9	**Priority:** Medium
Primary actor: Candidate	**Use case type:** Detail, essential	
Stakeholders and Interests: Election Candidates – Wants to Add or Edit his CV contents.		
Brief description: This explains how the candidate can Add contents to or edit his CV. The CV can be seen by voters.		
Precondition: The user should be logged in into the system **Trigger:** The user clicks the View And Edit CV button or link.		
Relationships: **Association:** **Include:** **Extend:**		
Normal flow of events: 1. In the user profile there is a button labeled "View And Edit CV" he clicks it to edit his CV. - The system opens a new page that contains the candidate's CV with data in it if any in edit mode. 2. The user edits his CV using the free editing template. 3. The user clicks "Save And Return" button to save the changes and return to his profile.		

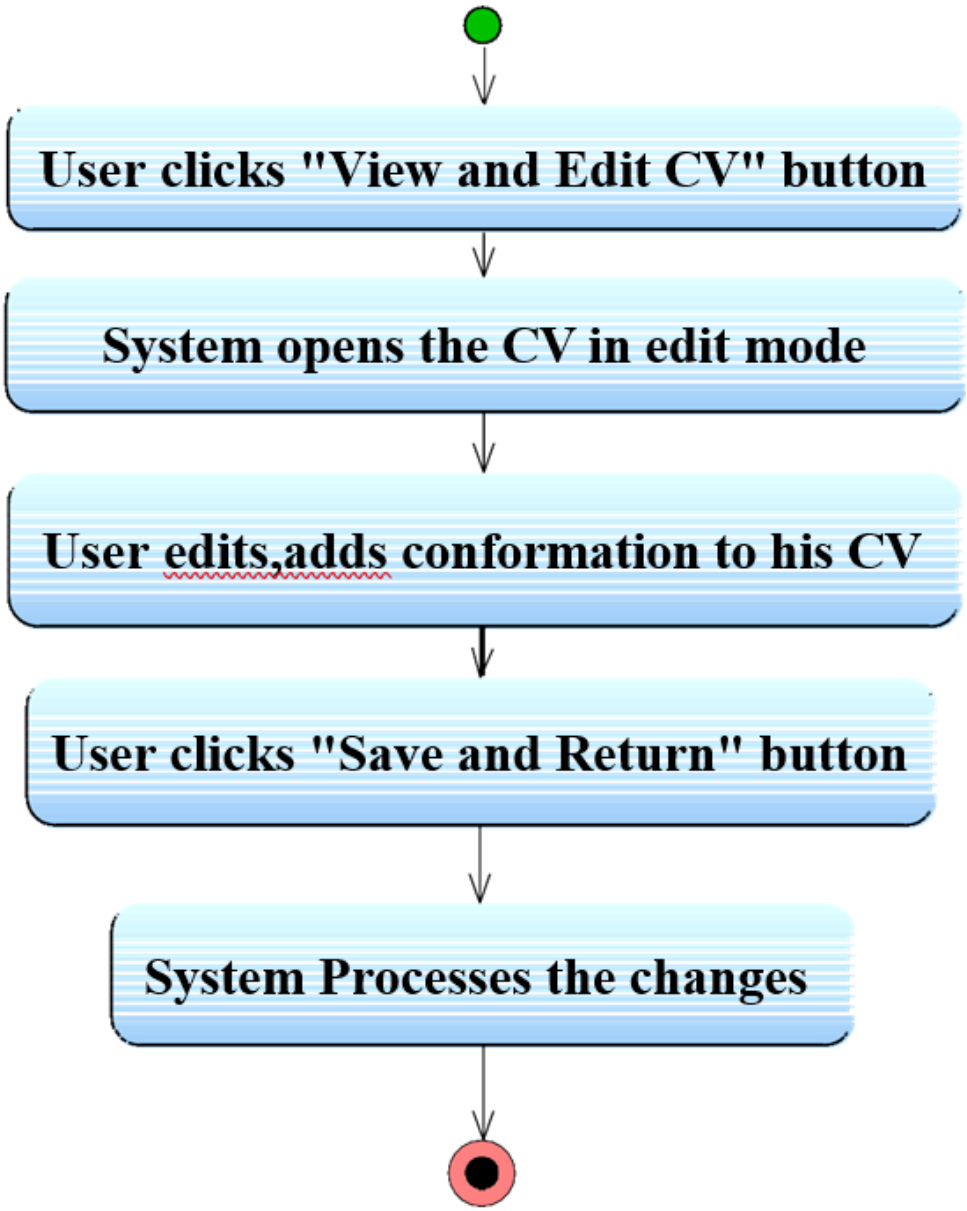


Figure 11: Activity Diagram for CV Editing

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. 2. The user adds or edits his promises using the free editing template. 3. The user clicks “Save And Return” button to save the changes and return to his profile.		

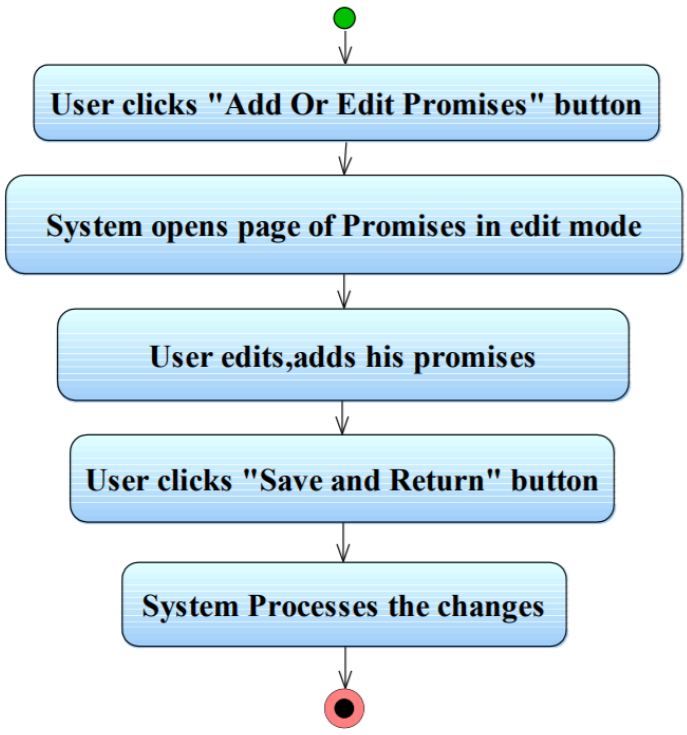


Figure 12: Activity Diagram for Adding /Editing Promises

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. 2. The user reads and/or answers questions if any. 3. The user clicks ”Reply and send” to save and send anwers of the questions. 4. The system takes the user to the questions page. 5. User clicks on “Return” button to return to his profile.		

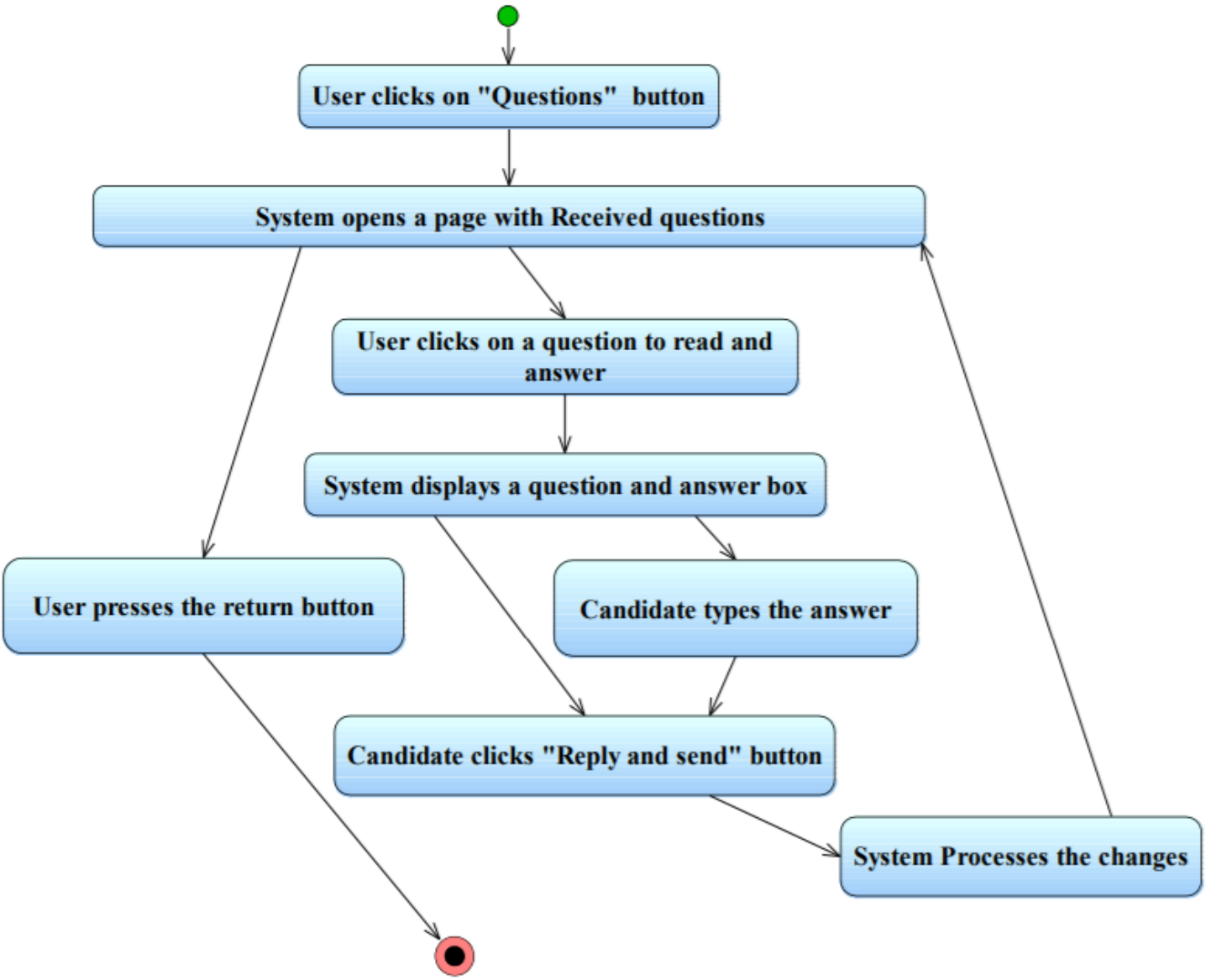


Figure 13: Activity Diagram for Reading / Answering Questions

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.		

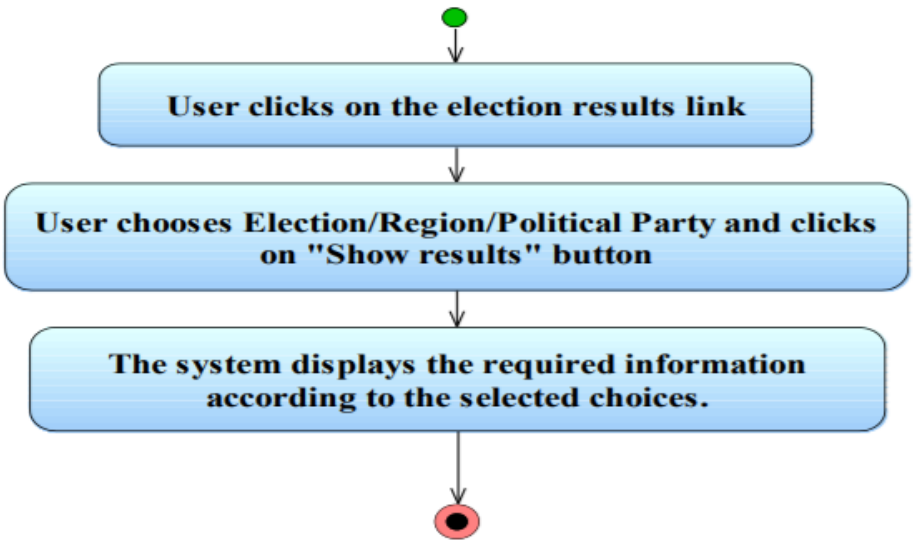


Figure 14: Activity Diagram for Viewing Election Results

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 A. If the TCK, supervisorID and password is valid, a session is opened. I. The security is verified. II. The voting page is loaded. B. If the TCK, supervisorID and password is not valid, the login screen is redisplayed with an error message.		

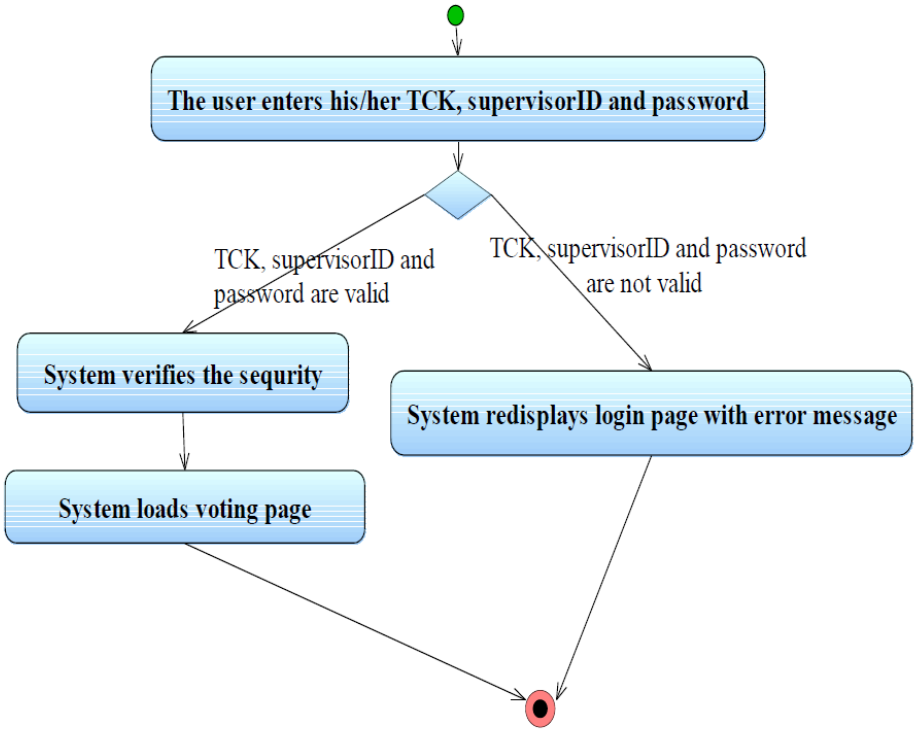


Figure 15: Activity Diagram for Opening System

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. A. If the voter with specified TCK has not voted yet. a. The menu is appears with “online vote” and “offline vote” buttons. (i) If user presses to “online vote”, the password is generated and printed. (ii) If user presses to “offline vote”, the voter marked as “ Offline Voted” by the system. B. If the voter with specified TCK has voted or wrong TCK entered. a. The login screen is redisplayed with appropriate error message.		

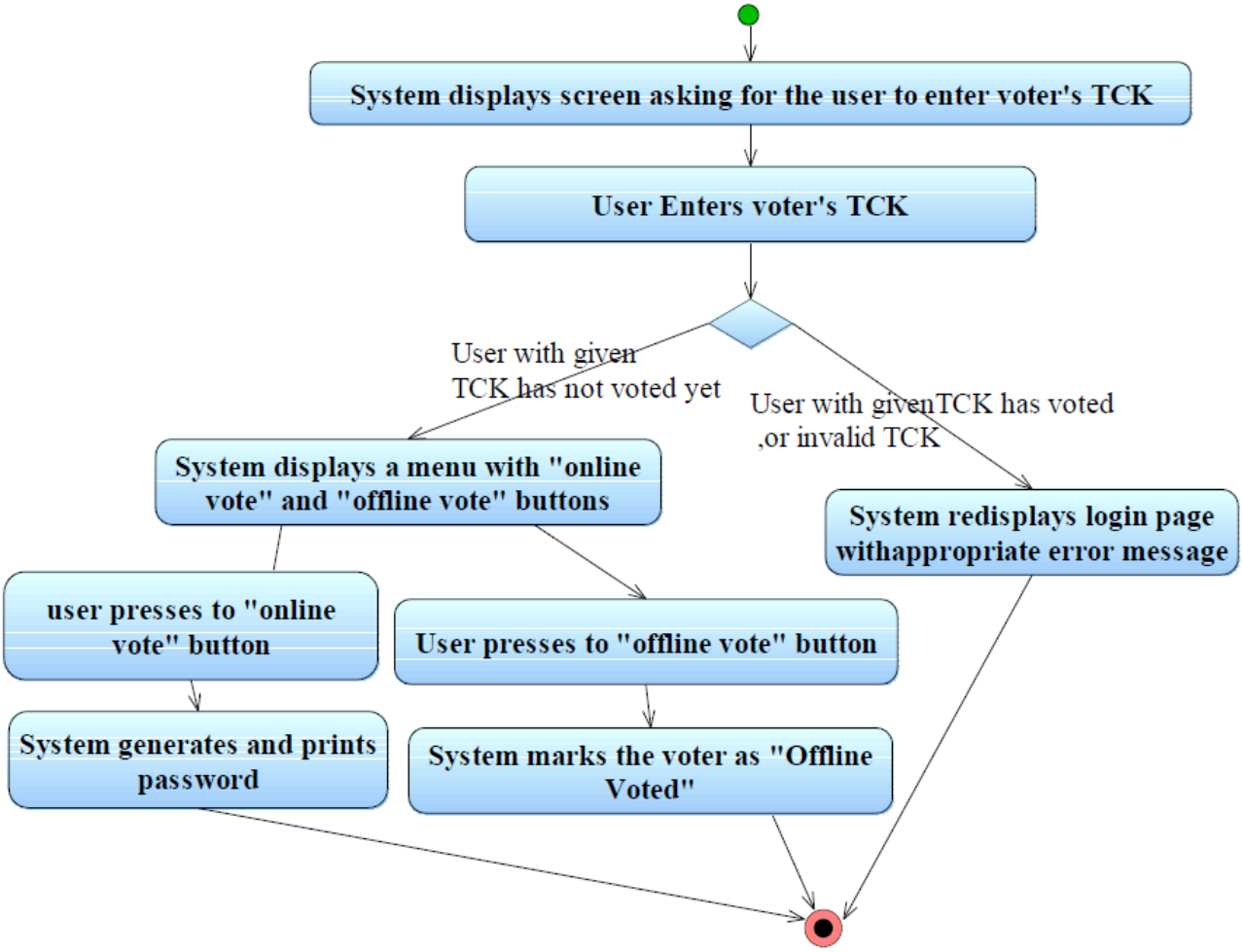


Figure 16: Activity Diagram for Mark as "Voted" and Generate Password

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. A. If the login operation is not verified the system prompts an error message and returns to login page. B. If login operation is verified i. The system will prompt the list of parties/candidates. ii. Voter chooses one of the parties/candidates from list. iii. He presses the “vote” button for voting process. iv. If the operation is successful, voter marked as “Online Voted” by the system. v. System automatically returns to the log in page.		

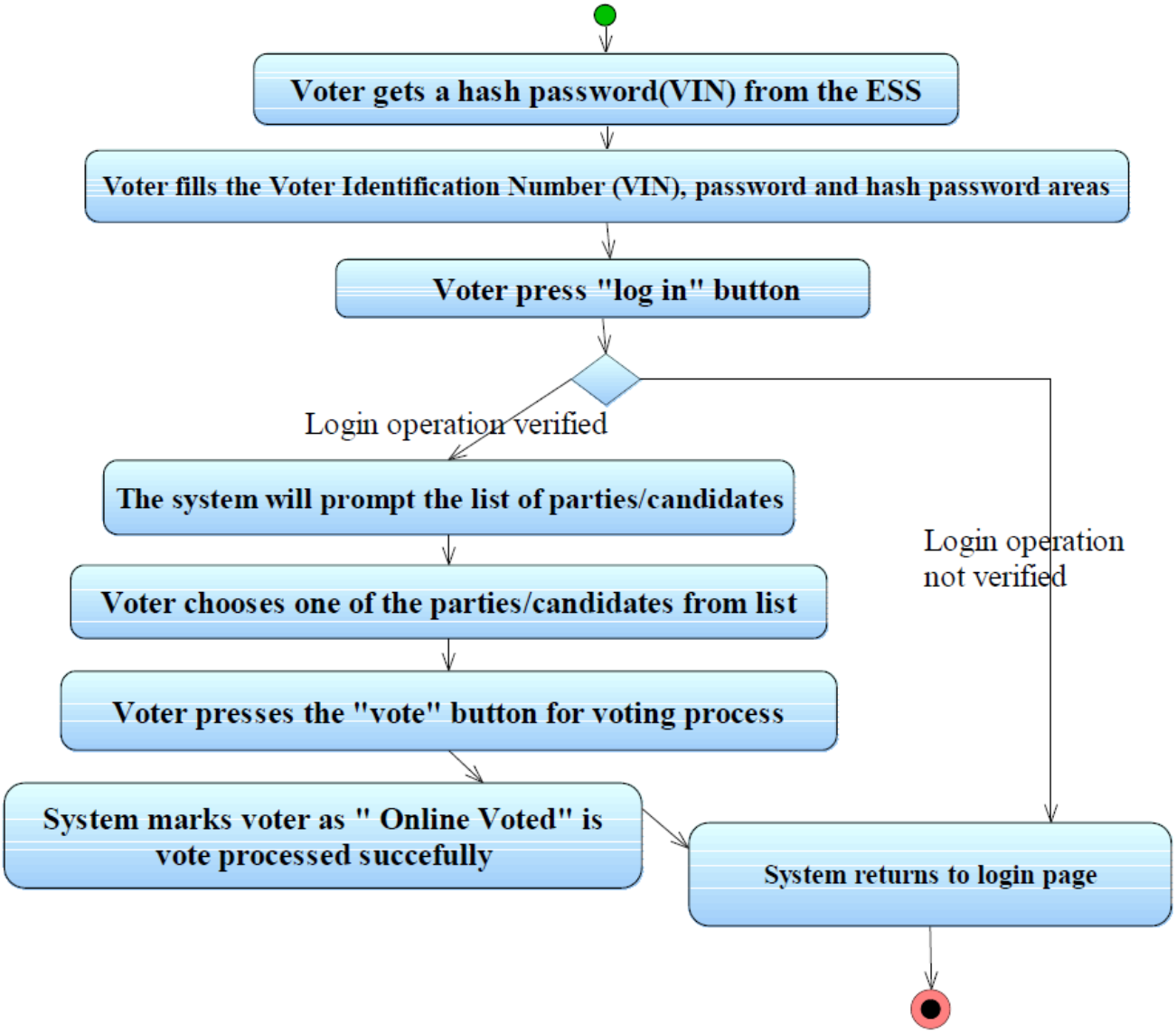


Figure 17: Activity Diagram for Voting Online function

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. A. If the TCK, supervisorID and password is not valid, the login screen is redisplayed with an error message. B. If the TCK, supervisorID and password are valid: a. The secuity is verified. b. The screen displaying every Political Party/ Candidate and their input fields is opened. c. The ESS fills every PP / EC fields with the number of votes each PP / EC got and presses button “Enter Votes”. d. The system compares the number of votes ESS entered with the number of votes marked as “Offline Voted” for that station. i. If equality holds the system stores given values to the system and the main page is displayed. ii. If equality does not match, the screen displaying every Political Party/ Candidate and their input fields is reopened with en error message.		

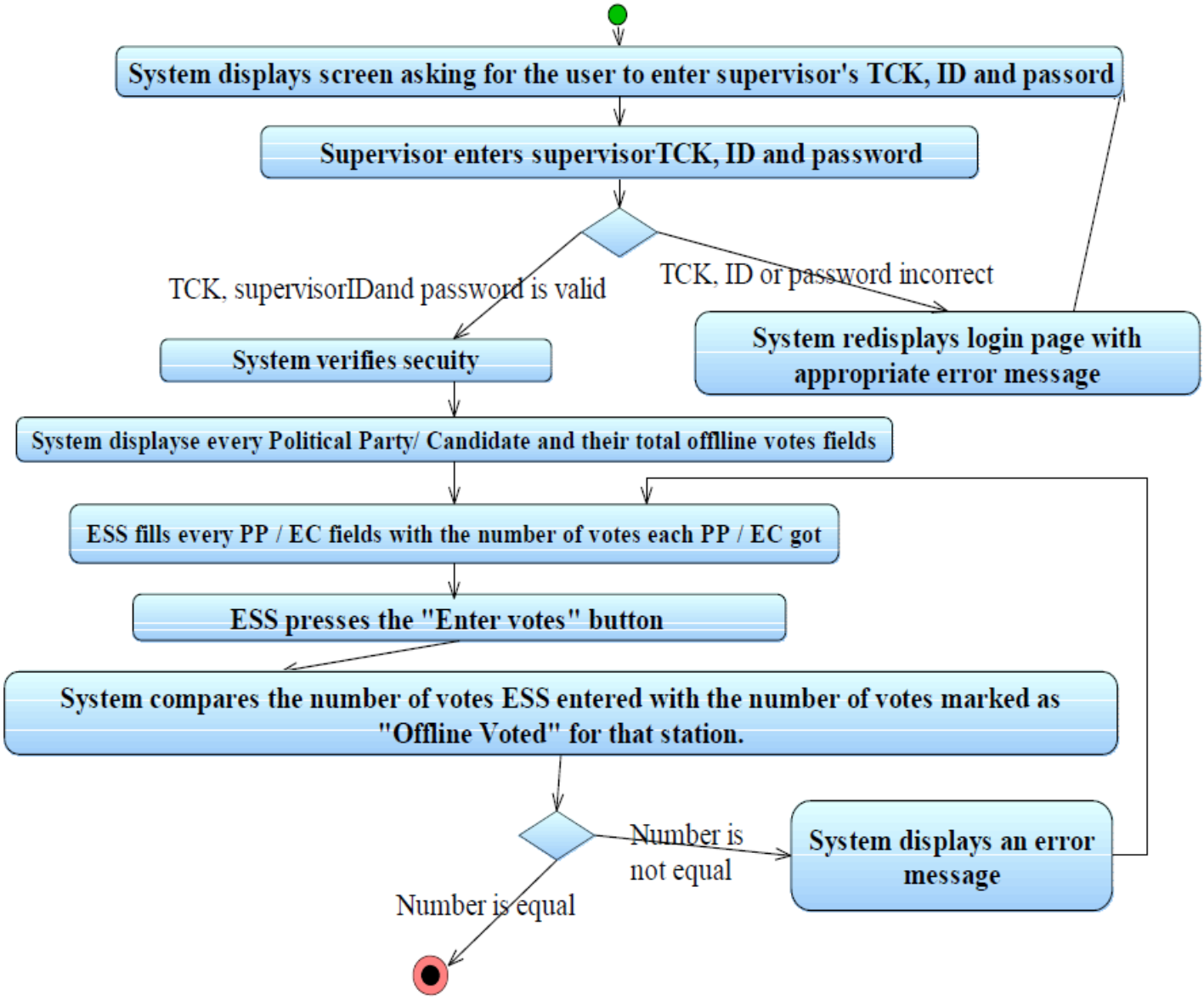


Figure 18: Activity Diagram for Entering Offline Votes

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 efficiency 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.

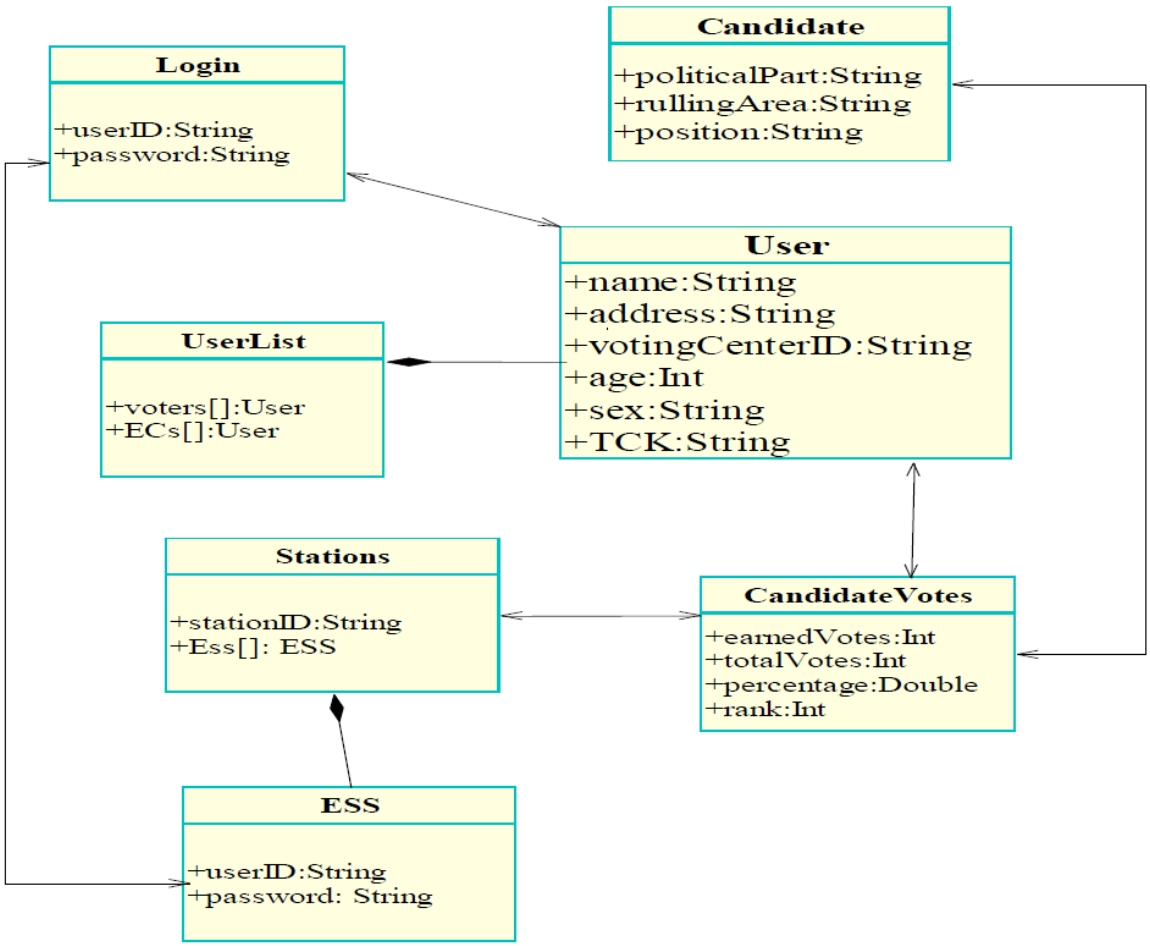


Figure 19: Complete Data Model

Data Dictionary

Header	Description	Field Type	Field Length/ Maximum Number
userID	This is a unique user identification 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
electionCandidates	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

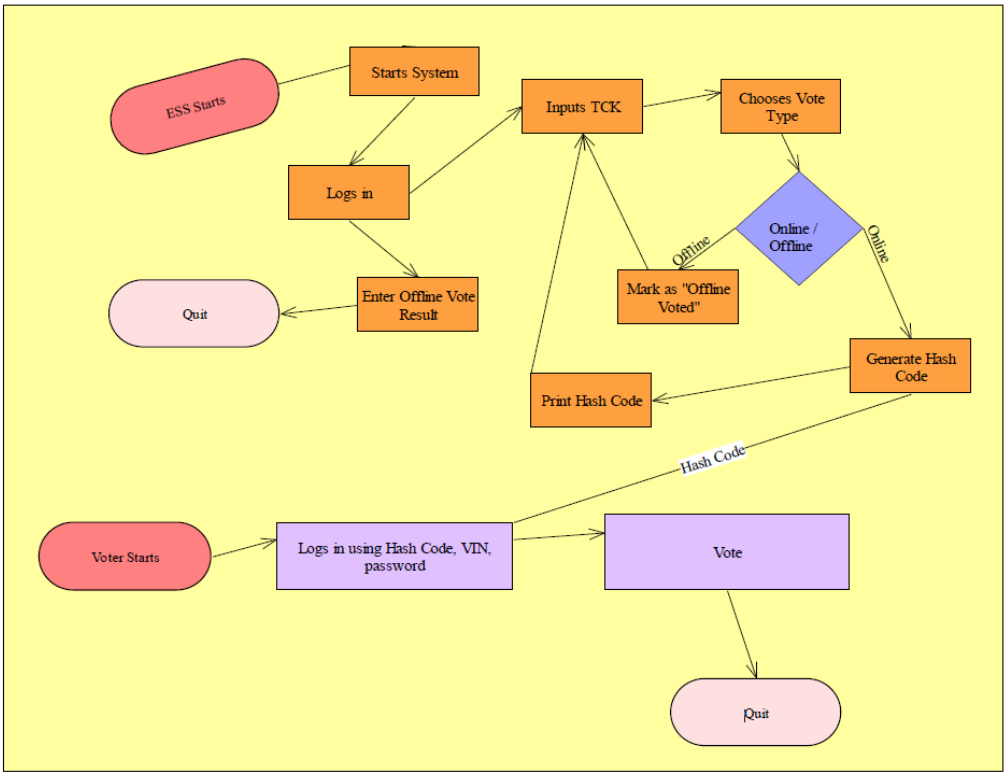


Figure 20: State Transition Diagram of Voter and ESS in Election Mode

Process Model

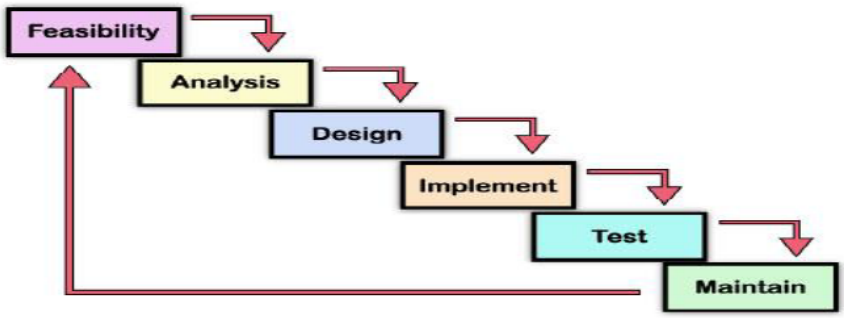


Figure 21: Waterfall Model showing the step by step activities in our software development processes

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 Specification

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