Software Requirements Specification

For

Elect and Verify (EnV): A system for online verifiable elections

Version 1.0 approved

Prepared by:
Ahmed Abd El Fattah Ewais
Hussam Ashraf El-Araby
Islam Faisal Ibrahim
Mohamed Amr Gadalla

Supervised by:
Dr. Sherif El-Kassas
Mr. Hossam Medhat

The American University in Cairo
November 14th, 2015

Table of Contents

Revision History

Name	Date	Notes	Version
Requirements Engineering Team	11/10/2015	First Partial Draft Version Created	0.1
Requirements Engineering Team	11/15/2015	First Full Draft Version Created	0.5
Requirements Engineering Team	11/14/2015	Second Draft Version	0.6

Requirements Engineering Team	11/14/2015	First Version	1.0

1. Introduction

1.1 Purpose

This document specifies the requirements for version 1.0 of Elect and Verify (EnV). EnV is a digital democracy solution for the whole elections process. EnV shall make the whole campaigning, voting and result publishing online and at the same time maintaining the privacy of ballots and integrity of the elections. The elections laws and rules are flexible and can be determined via configuration files.

1.2 Document Conventions

This document is a simple IEEE requirements specification document. The font for content is Calibri and of size 12 pts. The headings are in bold. Citation of references is in IEEE citation form. All appendices diagrams are in UML.

1.3 Definitions, acronyms and abbreviations

Table 1: Important Definitions

Term	Definition
Administrators, (System)	System users who are in charge of technical configurations, installations and maintenance.
Ballot	A secured verifiable cast vote.
Candidate	A person who is running for elections.
Elect and Verify (EnV)	The whole system (see system) and all its documentation, and accompanying source code and design documents.
Election	The process of asking a group of people for choices in a private aggregated way.
Officials, Elections	Users of the system who are authorized to configure all the election configurations and laws.
Subsystem	Any partial component or communication interface of the system.
System, (The)	The whole electoral system including the registration, pre-elections, voting, counting, verification, campaigning and debating subsystems with all its offline and online components.
Trustees, Elections	Users of the system who are capable of collabroatively issue tallying requests.
Type, Election	The type of elections including the type of candidates and voting type.
Voters, (Eligible)	All eligible voters who can participate in an election.

Table 2: Important acronyms

Acronym	Meaning
API	Application Programming Interface
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CSRF	Cross-Site Request Forgery

DoS	Denial-of-Service
XSS	Cross-site scripting

1.4 Intended Audience and Reading Suggestions

This document has been designed to be read by all the personnel involved in the development of the system including managers, marketing staff, documentation writers and testers as well as the client government agencies willing to implement this system. Appendices, however, may require UML knowledge to be interpreted.

1.5 Product Scope

1.5.1 Objectives

i. Provide an online system for candidates to publish their campaigns.
ii. Provide an online voting system for voters to cast their ballots.
iii. The elections process be transparent by allowing each voter to verify that his vote was counted.
iv. Provide a flexible electoral system whose laws and configurations can be modified easily by the authority.
v. Build a secure system that does not allow unauthorized access, electoral forgery or unlawful change of configurations.
vi. Provide an API for third party developers who are willing to publish the results or statistics.

1.5.2 Benefits

i. Increase the elections turnout by providing an easy-to-use online system.
ii. Reduce the error rates of manual vote counting and statistics calculations.
iii. Achieve equality among campaigns of candidates by providing a standardized campaigning system.
iv. Allow disabled eligible voters to cast their ballots easily.
v. Transparency in the whole elections process since the source code is available for scrutiny and the votes are published for voters to verify their votes were counted.

1.6 References

[1] B. Schneier, *Applied cryptography*. New York: Wiley, 1996.

[2] B. Adida, 'Helios: Web-based Open-Audit Voting', in *USENIX Security Symposium*, San Jose, CA, 2015, pp. 336-348.

2. Overall Description

2.1 Product Perspective

EnV is intended to replace traditional paper-based voting systems as well as online voting systems whose results cannot be publicly audited. EnV is an online verifiable open-audit voting system. EnV also includes solutions for electoral campaigns and debates. EnV is not designed however to solve the problem of coercion.

2.2 Product Functions

EnV will provide functions for elections officials to organize elections, configure the voting system and candidates' eligibility, receive nominations, invite voters and organize debates between candidates. Details of these functionalities are listed and discussed in section 3 as well as the appendices.

2.2.1 Pre-Elections and Voters Registration

The aim of pre voting is to give candidates or parties, who wish to participate in the elections, the opportunity to register for the vacant places and compete in the elections. The registration process includes registration and eligibility check to make sure that the candidates are fit to fulfill the position's mission and goals.

2.2.2 Debating and campaigning

This platform's main goal is to allow candidates to publish content of their campaign to let voters know about them. This platform will also facilitate debates between different candidates that can be viewed by the voters.

2.2.3 Voting

This is the core functionality of the system. This module should provide a secure verifiable way of casting ballots. Each voter should be able to cast their ballots for the candidate they want. After the conclusion of the elections, the voters should be able to verify their vote. No votes can be duplicated without being detected.

2.3 User Classes and Characteristics

EnV is an online system that makes it possible to handle the campaigning, debating and actual elections online, and so the targeted actors are: public viewers, voters, candidates, system administrators, trustees and debate moderators.

All the users are expected to know how to read either Arabic or English as the system interface will offer both, and also they are expected to know how to deal with web browsers and electronic devices such as mobiles or personal computers since the system is an online system. Trustees, however require further training on how to keep confidential data secure. The following subsection provides more details about each user role.

2.3.1 Elections Trustees

Elections officials are responsible for setting the elections type, inserting the elections laws and protocols in the system and updating the eligible voters' database before the elections. Trustees will feed eligibility information of both candidates and voters to the system and will deal with all legal lists and databases. In case of building a Helios [2] similar system, elections trustees are responsible for initiating the elections by providing the public keys and share the keys that decrypt the tallies.

2.3.2 System Administrators

A system administrator is a logged in user that creates, sets up, and updates the system platform. Every subsystem will have a different group of admins. System administrators are responsible for technical details of the system, backups, logs, maintenance and providing help to other user roles. The authority must be distributed between all system administrators. Sensitive functionalities require multi-user authentication.

2.3.3 Voters

A voter must be an eligible voter and his name must be in the list of eligible citizens to vote that will be provided by the government, otherwise he will not be able to sign up for the online voting.

2.3.4 Candidates

Candidates are the political parties or persons that will be running for the elections depending on whether the elections is list or individuals based. Candidates must be eligible to run for this elections, he must be able to handle his personal space and publish content online. He is expected to behave in ethical manners.

2.3.5 Public Viewers

A public viewer is a person or search engine that has internet access and is able to navigate to the system's web address. He may not be logged in. Public viewers including search engines are viewers to the voting portal, campaigning and debating system who are not authenticated or do not have credentials.

2.3.6 Debate Moderators

A debate moderator is a logged in user that represents a well-known public figure that is assigned by the system administrator prior to each debate. He is responsible for debating with the candidate(s) moderating the debate.

Remark 2.3.1. We sometimes refer by *Logged-in user* or *authenticated user* to a user that has passed a successful secure authentication process on our system and has logged in under a valid account with a corresponding user role on the system. This includes all users of the system except public viewers.

Remark 2.3.2. A candidate is typically a voter. However, the candidate's role in the system is effective when authenticating to the campaigning/debating platform. This account cannot vote but instead represents the candidate (single person or party) during the campaigning and debates. When authenticating to the voting subsystem, a candidate or member of a party should use their personal credentials.

2.4 Operating Environment

2.4.1 Physical Environment

The elections shall be distributed on many servers located in different geographical locations. Backups should also be located in several geographical locations. Datacenters need to be physically secured and safe against fire and other hazards.

2.4.2 Server Side Environment

Webservers need to be deployed as well as database servers to host the data. Communication protocols such as HTTPs will also need to be well configured. The operating system is CentOS for servers.

2.4.3 Client Side Environment

The client voting system is web-based, so it requires a web-browser to open that can handle SSL and TLS protocols. No special machine requirements are needed. The user needs to possess a phone for the two-factor authentication, but with no special requirements.

2.5 Design and Implementation Constraints

To allow for scrutiny, all the systems dependencies and components on which the voting process depends must be [publicly?] open-sourced. No sub-components or dependencies to voting can be in the form of closed source (or proprietary). This does not apply to integrated services such as the payment system.

2.6 User Documentation

After the completion of the project, the following documents should be available for system use and maintenance.

1. Design document explaining the architecture of the system and subsystems.
2. Source code documentation for all the code and executables of the system.
3. Database structure document
4. A technical document explaining and describing the algorithms, encryption and verification techniques and protocols used in the system with (or with reference to) mathematical proofs whenever possible.
5. A document for public release explaining to the public why EnV is secure and verifiable. This document should also draw attention to coercion and the importance of users verifying their votes.
6. A document explaining how to configure the system before elections and how the trustees should provide their keys.
7. A document for the trustees explaining the sensitivity of the keys they hold and how they can keep them secure.
8. A document for security, maintenance and backup checklists for system administrators

2.7 Assumptions and Dependencies

9. The citizens' database operates independently of the elections system and it is always up-to-date.
10. The election trustees are not all dishonest and corrupt. If all of them happen to be so and agree to tamper the results, the elections forgery may not be detected and the verification process may become invalid.
11. This verifiable electoral system does not prevent coercion. All of the system users should be aware that this may affect the results.
12. A certain percentage of users will verify their votes and audit the results. If this does not happen, the verifiability condition is useless.

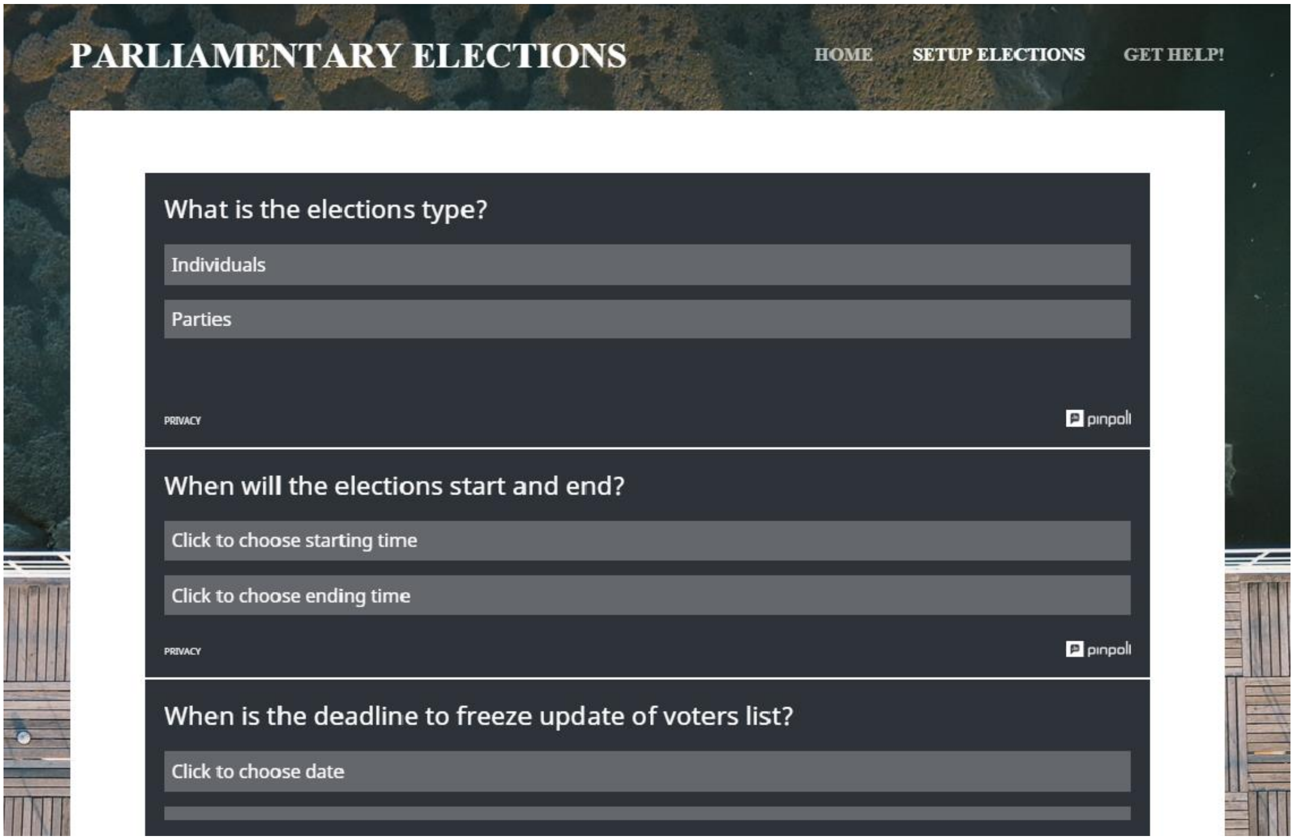
3. External Interface Requirements

3.1 User Interfaces

Each of the user classes described in section 2 will interact with the system. However, in this section, we talk about the user interfaces in terms of the system function.

3.1.1 Elections Configuration UI

This user interface will enable elections officials to configure the elections and for trustees to install their public keys and initiate the elections process. Figure 3.1.1 1 shows how this user interface should look like.



What is the elections type?

Individuals

Parties

When will the elections start and end?

Click to choose starting time

Click to choose ending time

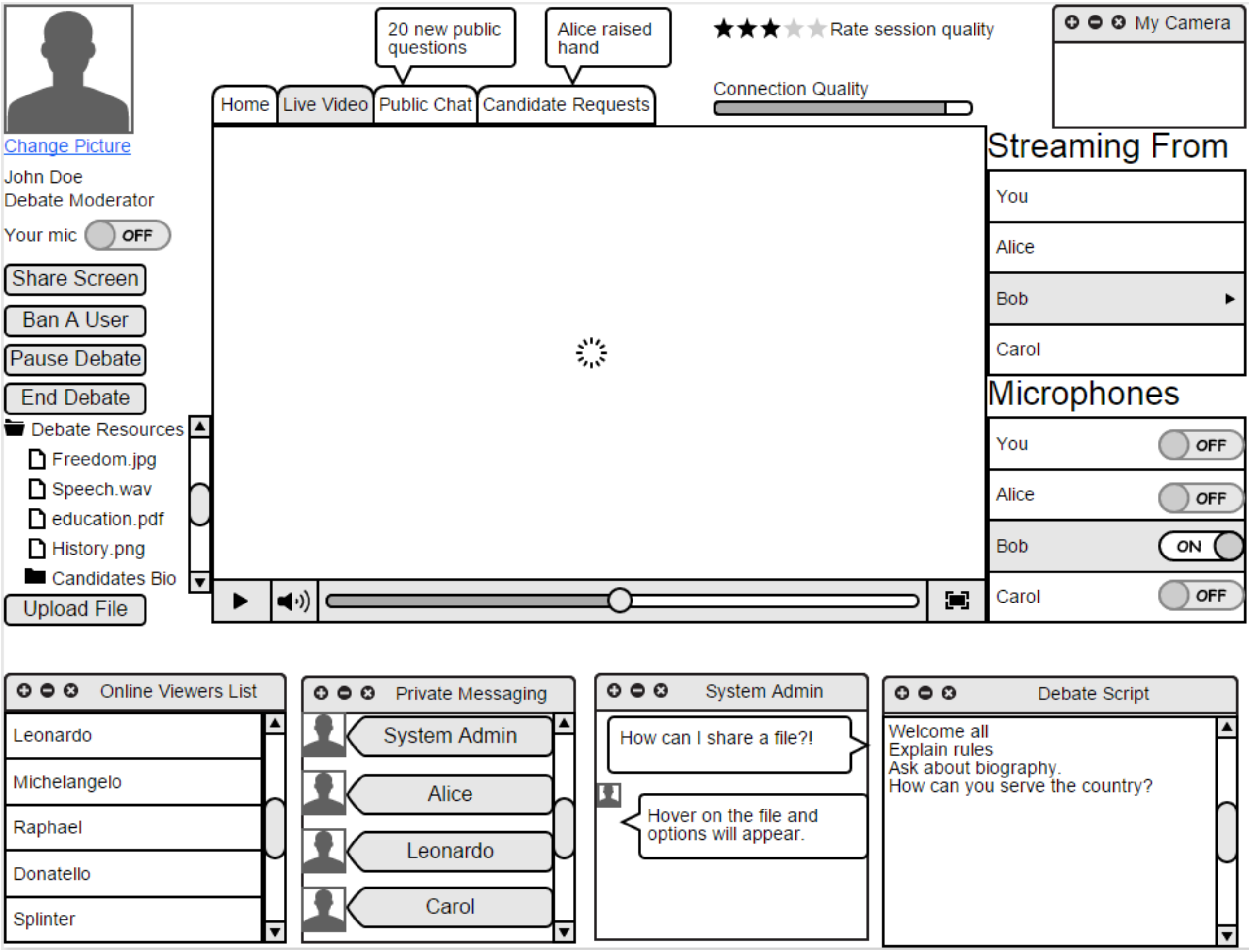
When is the deadline to freeze update of voters list?

Click to choose date

Figure 1: Example of the expected elections setup screen for elections trustees

3.1.2 Campaign and Debate Platform UI

This user interface will allow candidates to publish campaigning content for the candidates to reach out to voters. Also, it will facilitate the debating process for moderators, clients and public (voters) viewers. Figure 3.1.2-1 shows an example of the expected moderator screen in the debating platform.



How can I share a file?!

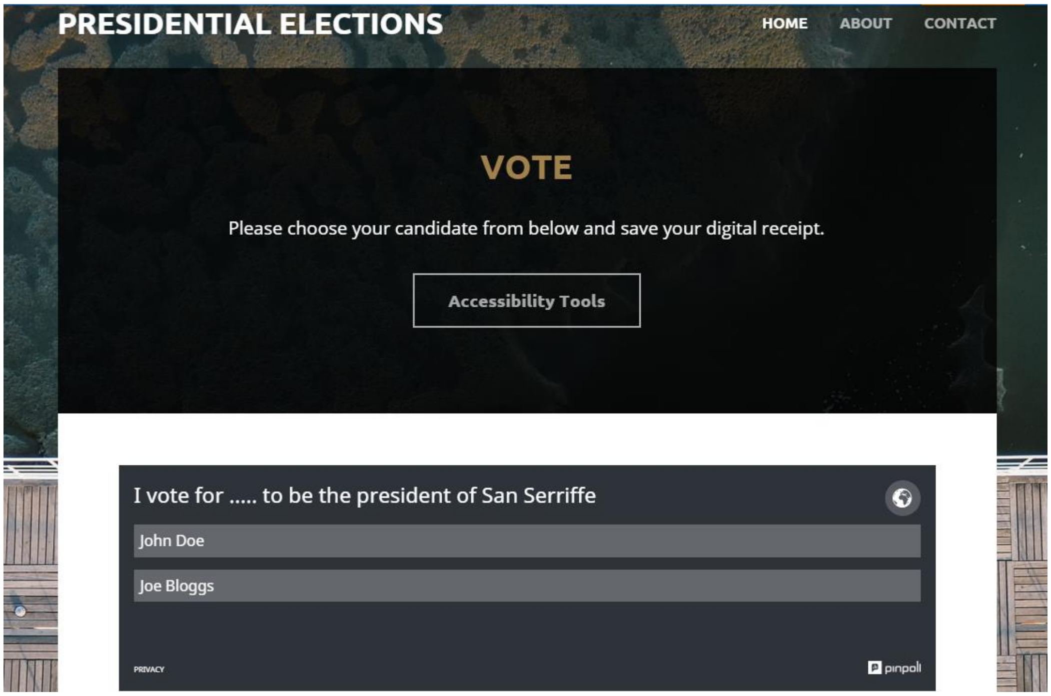
Hover on the file and options will appear.

Welcome all
Explain rules
Ask about biography.
How can you serve the country?

Figure 2: Example of the moderator screen in the debating platform

3.1.3 Voting UI

This user interface will help the voters view the candidate, cast their ballot and verify the result using their digital receipts. To satisfy the non-functional requirement of accessibility and disability tools, the system will have a button to show accessibility tools such as magnifying glass. Also, once the page loads, there will be voice interaction to narrate the instructions and even capture the choice via voice and confirm it. Figure 3 shows how this user interface should look like.



VOTE

Please choose your candidate from below and save your digital receipt.

I vote for ..... to be the president of San Serriffe

Figure 4: The voting user interface including option for accessibility tools.
Figure 5

3.2 Software Interfaces

3.2.1 Payment System

The candidates who use the campaigning system are required to pay some fees. These fees will be collected via an online banking applications or other payment systems. These payment systems need to be securely integrated with the system.

3.2.2 Cell Phone Authentication System

A system for calling/sending an SMS to users is used for two-factor authentication. The system and the generation of authentication codes shall be integrated in the authentication system securely. This system shall be implemented independently of the voting system (i.e. it is used in authentication, however it can never affect the elections results).

3.3 Communications Interfaces

1. The communications between the voter and the voting system shall be designed in a way to disallow any man-in-the-middle from changing the content being transmitted.
2. Anything being transmitted among subsystems or users shall be done so in a way that any unauthorized man-in-the-middle cannot extract the actual messages from.

3.3.1 Citizens Database

The voters' database shall be updated with the citizens' database during a specified duration before the voter registration phase and frozen during the whole elections, once the voter registration phase starts.

3.3.2 Payment System

The candidates who use the campaigning system are required to pay some fees. These fees will be collected via an online banking applications or other payment systems. These payment systems need to be securely integrated with the system.

4. System Features

In this section, the system is divided into main modules according to functional similarities and also as per the elections timeline. The functionalities in each module is describing according to the users' use cases. For more details on the functional features, refer to the appendices where there are use cases, use case tables and activity diagrams.

4.1 Pre-Voting module

During this process, Elections trustees feed the system with rules, laws and dates of the elections. The voters' database is also updated from the citizens database. Also, candidates or parties, who wish to participate in the elections, have the opportunity to register for the vacant places and compete in the elections. The registration process includes registration and eligibility check to make sure that the candidates are fit to fulfill the position's mission and goals. Now, we give description of the most important use cases. Details can be found in the appendices.

4.1.1 Candidate

The political group or person that will be running for the elections depending on whether the elections is list based or individuals. The candidate will be able to:

4.1.1.1 Register

The function allowing the candidate to nominate himself to run for the elections. Anyone who will fit in with the requirements will have the capability of registering for candidacy. This check requires the availability of another function called check eligibility. Registration also includes registering for a space on the website to allow for debating and campaigning, which in turns requires paying a fee.

4.1.1.2 Register for website space

The registration on the website function will give every candidate the luxury to post every detail about him and all the plans he wishes to implement.

4.1.1.3 Start to campaign and debate as officially

The candidate/party is now considered as official nominees for the elections so they can start campaigning and debating using the campaigning and debating platform (See 4.3).

4.1.1.4 Check candidate/party eligibility

This function is shared by both the candidate/party and the government representative. The government representative will check all the requirements and compare them to the candidates'/parties' information in order to make sure they are all capable of registration and participation.

4.1.1.5 Pay funds for campaign

The candidate/party will pay the government funds after his successful registration.

4.1.2 Elections Trustees

As described in section 2.3., elections trustees are responsible for the following functionalities:

4.1.2.1 Update list of eligible voters

The aim of this function is to have an updated list of all the citizens who have the right to vote. For example, it will exclude all the dead people since the last elections, and it will include all the new people who have passed the legal age and are now capable of voting.

4.1.2.2 Publish results

This function will be used after the end of the elections to post all the results and announce the ranking of all candidates (if applicable) and identify the winner. If the design document designs something similar to Helios [2], trustees will also give their keys to do the tallying before publishing the results.

4.1.2.3 Set elections time and type

This function allows the government representative to choose specific time for the elections, and to identify its type; whether it is presidential, parliament, individuals or lists based.

4.1.2.4 Check candidate eligibility

This function is shared by both the candidate and the government representative. The government representative will check all the requirements and compare them to the candidates' information in order to make sure they are all eligible to register and participate.

4.1.2.5 Determine voting & candidacy eligibility conditions and other elections rules

4.1.3 Voter

As described in section 2.3., a voter the right to vote and choose to which candidate or party he will be casting his vote. His functionalities are as shown in the following points:

4.1.3.1 Register for online voting

A voter will be able to register for online voting which will include an eligibility check and will collect the contact information of the voter to validate it at the time of voting. The voter will also have to choose a username, a password and a mobile number for two-factor authentication.

4.1.3.2 Check his eligibility

When the user tries to register for online voting, it will first check if he is in the database of eligible voters and then allows him to register.

4.1.3.3 Provide/change information

The user will provide his contact information and credentials for the first time (preferably, he visits a government office for doing so). After that, he can update his information from the system using his credentials.

4.2 Voting module

This module is for the voting process. It describes how voters cast and verify their votes. For more details, please refer to the appendices. Now, we list the major functional requirements of this module sub-divided by users.

4.2.1 Voter

4.2.1.1 Login

This function will use two-factor authentication protocol to assure that the voter actually is the same person who registered before. The voter will be asked to enter the username and password and then a verification message will be sent to his personal mobile with a code that he needs to submit to the website within a certain timeout.

4.2.1.2 Vote

At the time of the election, the voter will be able to choose one of the candidates or one of the lists if the election is using the lists system. The function requires login, and will include sending the vote to the election system which will verify that the user hasn’t voted before, also voting will include generating a number which will be used to track the vote after the results are published.

4.2.1.3 Get Digital Receipt

The voter will be able to get a unique number that he can use to track his vote after the election results are published to make sure that his vote was counted. This function can be invoked only after the voter had voted.

4.2.1.4 View results

After the results have been approved and published on the website, a voter can view the final results, the winner candidates and the statistics associated with the elections.

4.2.1.5 Verify Vote

After the result has been approved and published on the website, the voter can use the receipt or the unique number he received after voting to track his vote and ensure that it was counted for the right candidate or list. If applying a system similar to Helios [2], the user will verify also other sub-processes of the elections.

4.2.2 Elections Trustee

4.2.2.1 Publish result

After the elections, the judge published the result and if the online system is used alongside with conventional elections he uploads its result on the website and publishes all the result on the website.
Extend/stop elections: The judge can have the option to extend the time of the elections for some extra hours if a lot of voters are still voting or have not yet voted.

4.3 Campaigning and Debating Platform

This module is dedicated for candidates to publish content related to their campaigns and join debates with other candidates. Voters can view the campaigns, debates and interact and ask questions. Moderators are public figures who are responsible for managing live debates. Now, we present the use cases for this module by listing them for each user role. More details are in the use case tables, use case diagrams and activity diagrams.

Remark 4.3.1. The roles “User” and “Authenticated User” are explained previously in remark 2.3.1.

4.3.1 User

Any user *including* public viewers (and all the others) can:

4.3.1.1 View Candidate List:

A user is able to view the lists of candidates and clicking on any candidate’s link opens his respective website where information about him and his campaign can be found.

1. View the whole list in a random order, or a specific order if elections law specifies one.
2. Filter the list according to one or more filters.
   For example, if elections law specifies electoral districts, user must be able to query his district and know the candidates that he can vote for.

4.3.1.2 View Live and Past Debates

A user should be able to watch the live debates and review the past debates. He is not able to participate in any debates if he is not logged in.

4.3.2 Authenticated User

Any authenticated user *(See remark 2.3.1)* can:

4.3.2.1 All User’s functions stated in section 4.3.1

4.3.2.2 Change account information

A Logged in user can reset his password or update his other information such as electoral district (according to the procedure that elections law states, the feature included in the system is to update his information not to validate it which is the work of other entities.)

4.3.3 Voter

A voter can perform the following functions:

4.3.3.1 All logged in user's functions stated above

4.3.3.2 Voter sends questions to debate moderator

The voter can ask the candidate(s) in the debate questions by forwarding his questions to the debate moderator who decides when and what to include from the questions he receives. The question can take the form of a text, video, image, or other media such as sound recordings.

4.3.4 Candidate Account

4.3.4.1 All logged in user's functions stated above.

4.3.4.2 Post content on website

After registration, each candidate is able to post content to his website that is provided to him (a platform similar to WordPress that we provide). The content that can be posted is in the form of text, images, videos, links, or other media, such as sound recordings.

4.3.4.3 Delete content from website

Candidate can delete content he previously posted on his website

4.3.4.4 Participate in a debate

The candidate is able to participate in timed debates. He can reply through live video feed, posting text and video or other media in chat, or by microphone. His access to these tools is controlled by the debate moderator.

4.3.5 System Administrator

A system administrator does the following functionalities:

4.3.5.1 All logged in user's functions stated above

4.3.5.2 Maintains Platform

1. **Configure platform:** Prior to voting registration, the system administrator creates and configures the system.
2. **Update platform:** Throughout the elections process, he updates the platform as needed.

4.3.5.3 Setups debates:

1. Assigns time of a debate and its participants
2. Assigns at least one moderator to a debate

4.3.6 Debate Moderator

4.3.6.1 All logged in user's functions stated above

4.3.6.2 Receive voter questions that is sent to him during debates

4.3.6.3 Participate in debate

In a debate, a moderator participates by asking the candidate(s) questions through sending questions in form of text, video, or other media through chat, through microphone, or through live video feed.

4.3.6.4 Moderate debate

The debate moderator needs to keep the debate in an orderly manner by forcing time constraints and dealing with unruly participants. Moderation includes:

1. **Grant Mic:** broadcasts the feed from the microphone of one or more debate participants to all the other participants and viewers.
2. **Grant Chat:** broadcasts the uploaded content from one or more debate participants to all the other participants and viewers.
3. **Grant Feed:** broadcasts the feed from the webcam of one or more debate participants to all the other participants and viewers.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. The system shall be able tested for both load and response time testing to ensure availability and fast performance. The load testing shall be inspired from the expected load on the system.
2. The encryption, decryption, shuffling and tallying processes should be verifiable as a whole and for each sub-process and vote.

5.2 Security Requirements

3. All credentials used to authenticate as any role in the system shall be composed of strict passphrases that are not easy to guess. These regulations may vary in strictness by the user's role and shall be well-documented in the design document.
4. The system shall prove to be invulnerable to common DoS, CSRF and XSS attacks.
5. After two unsuccessful voter authentication attempts, a well-established CAPTCHA shall be used to make sure credentials are being entered by a human.
6. A timeout shall occur after a specific number of minutes of inactivity after a voter's successful authentication.
7. **Distribution of authority:** In order for the government to change any sensitive configuration in the system, multiple authorized users must provide their credentials. The list of these sensitive functionalities that require distributed authority shall be appended in the design document.

5.3 Privacy

1. The users' private data shall not be shared with any third party that is not stated explicitly in the election laws.
2. Any sensitive data including passwords, biometrics and other credentials shall be hashed in an irreversible and secure way. The hashing method used shall be acceptable by the scientific community.
3. Neither any automated subsystem nor actors shall be able to associate voters with votes.

5.4 Data Integrity

1. There should be a verification function that verifies the integrity of the system database and voting tables at any point during the system to ensure that nothing is tampered with.

5.5 Availability

1. During the voting process, the system should be available to all online users on the internet. Any unavailability in the system exceeding a specific time limit should be reported in logs. Then, cancelling the elections is at the discretion of the elections officials.
2. The campaigning system shall be available for public users to view and search engines to access.

5.6 Software Quality Attributes

5.6.1 Dependability and Reliability

3. An eligible voter shall not be allowed to send more than one vote.
4. Only eligible voters are allowed to vote.
5. The system must be unable of spoofing a voter's identity, duplicate a vote or send in an unauthorized vote without being detected.
6. The system cannot tamper with verified ballots without being detected and [optionally] traced.
7. During the voting process, it must be taken into consideration that a user cannot authorize himself in the voting system by two different simultaneous (with very small time delay; that the first authentication does not reflect in the system) requests.

5.6.2 Accessibility and Ease of Use

1. The system shall provide accessibility tools for disabled users. Any disabled users of any kind of disability including visual impairment, hearing disability, paralysis and mental disability that are eligible to vote, shall be able to do so in an easy feasible way that abide by all functional and nonfunctional requirements.

2. The entire voting process shall be done by a moderate internet user in an average time of 5 minutes.
3. The user shall be able to save the digital voting receipt in a file that can be easily saved on a non-volatile memory, and its integrity must be verified upon delivery.

5.6.3 Portability

1. The front-user interface including the voting, campaign viewing and result viewing shall all be designed to be accessed from any online device.

5.6.4 Logging and Tracing

1. Any modification that happens to the system by either system administrators or elections officials shall be logged with timestamps and user identification.
2. The system logs should be both human readable and easy to parse and mine by automated tools.

5.6.5 Maintainability and Reusability

1. Source code shall be well-commented, modular and maintainable.
2. All the C++ source code shall abide by Google C++ Style Guide allowing the code to be maintained and reused.
3. Source code shall be completely separated from configuration files.
4. No credentials, entity names and any sensitive data shall ever appear in the source code.
5. The entire source code with all of its versions shall be published online as open-source via the git version control system.

6. Other Requirements

1. The resources for candidates such as campaigning websites shall be distributed equally between all candidates to allow fair use.
2. The system shall respect all human rights, all state legislations and codes of ethics.

Appendix A: Use Case Diagrams

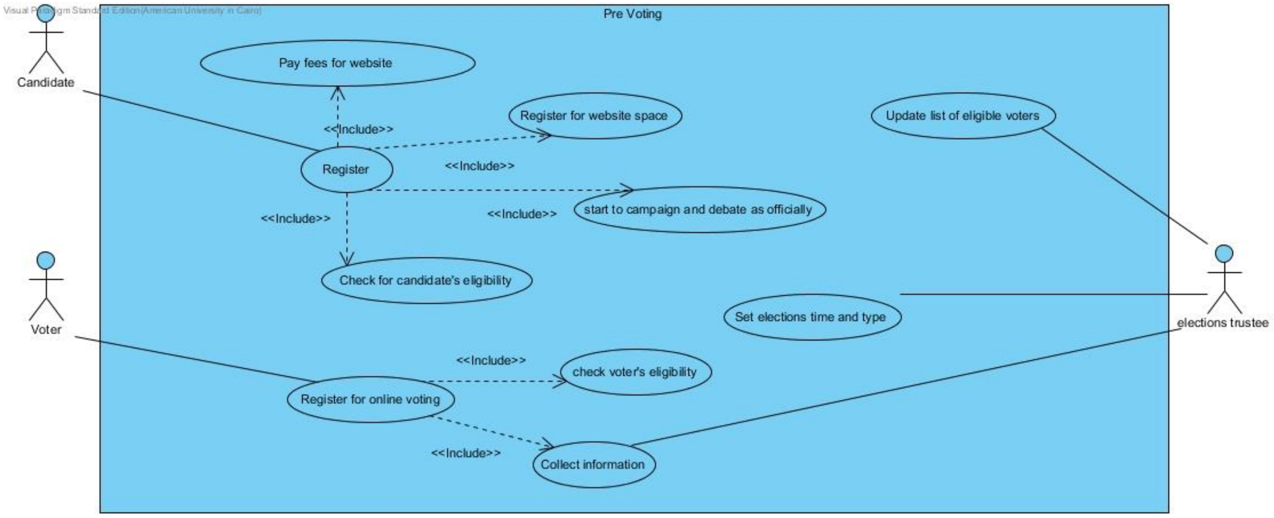

Visual Paradigm Standard Edition(American University in Cairo)
Pre Voting
Candidate
Pay fees for website
<<Include>>
Register for website space
<<Include>>
Register
start to campaign and debate as officially
<<Include>>
<<Include>>
Check for candidate's eligibility
Update list of eligible voters
Voter
Set elections time and type
elections trustee
<<Include>>
check voter's eligibility
Register for online voting
<<Include>>
Collect information

Figure 6: Pre-voting Module Use Case Diagram

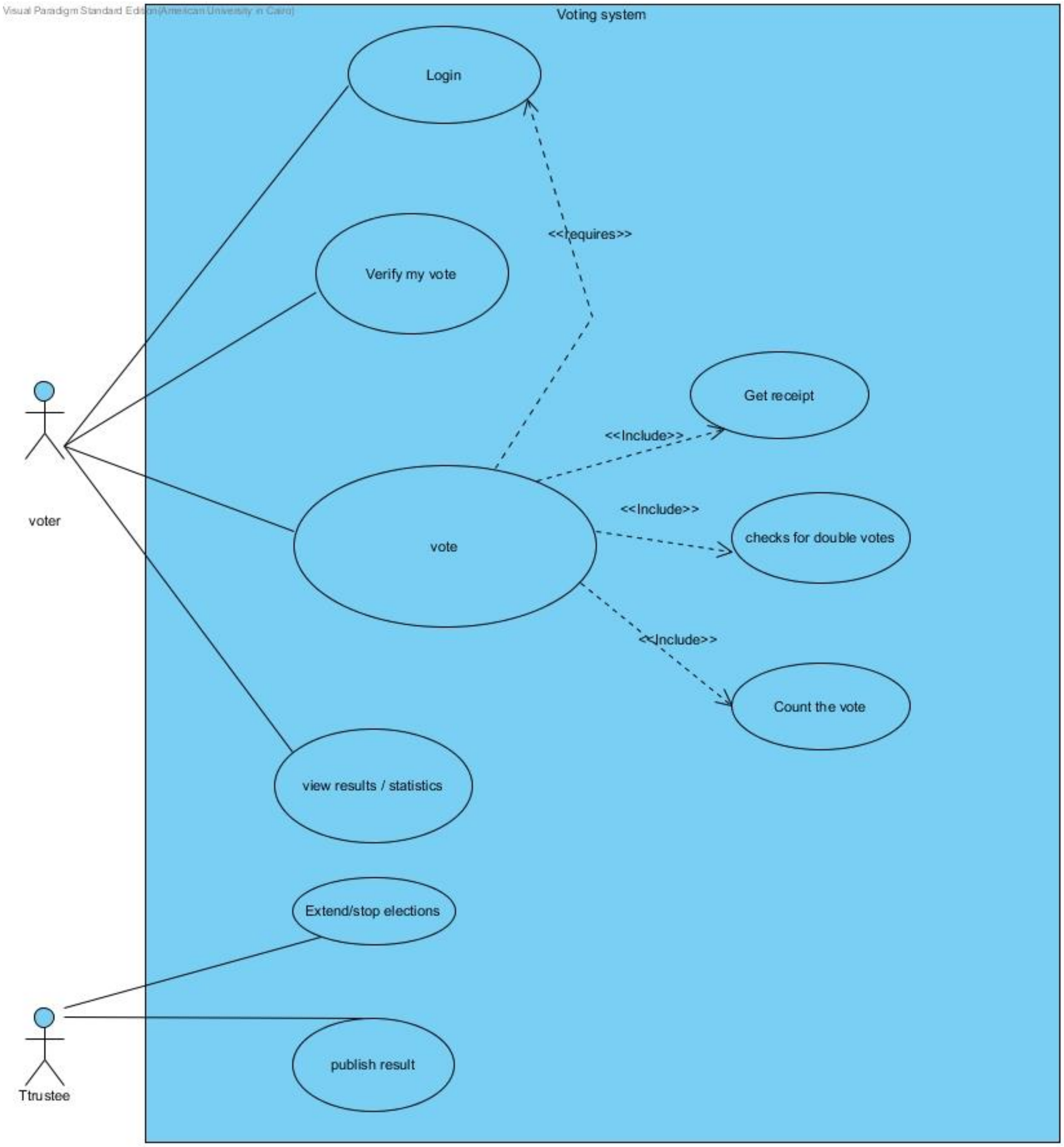


Figure 7: Voting Module Use Case Diagram

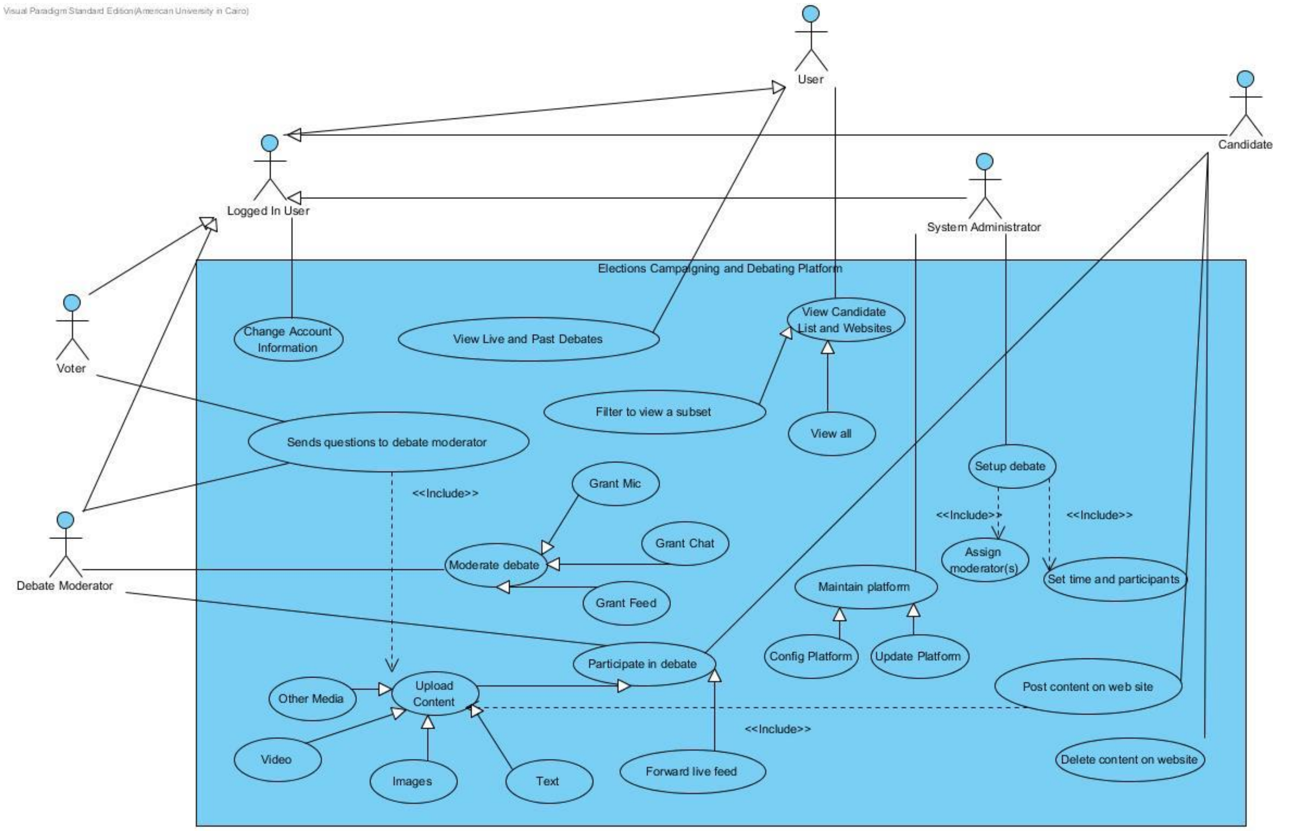


Figure 8: Campaigning and Debating Platform Use Case Diagram

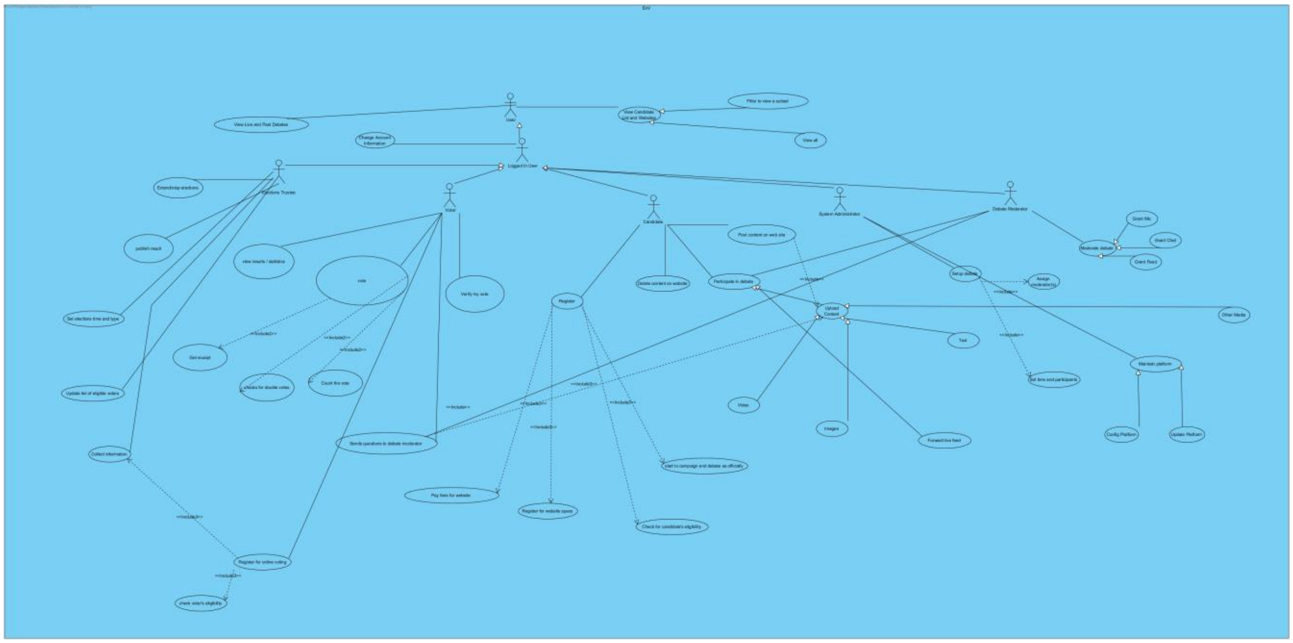


Figure 9: EnV Use Case Diagram

Appendix B: Use Case Tables

Table 3

Use Case ID:	UC01		
Use Case Name:	Vote		
Created By:	A. Ewais	**Last Updated By:**	I. Faisal
Date Created:	13-11-2015	**Last Revision Date:**	14-11-2015
Actors:	Voter		
Description:	The Voter chooses one of the candidates to give him his vote.		
Trigger:	The voter specifies one voter and clicks on vote.		
Preconditions:	The Voter must be logged into the system.		
Postconditions:	• The system must check first if this user has already voted. • The voter received a receipt number which is generated randomly by the system. • The vote will be tallied automatically by the system.		
Normal Flow:	1. The voter logs in to the elections website. 2. He chooses one of the candidates and clicks on vote. 3. The system checks if he has voted already or not. 4. The system generates a random receipt number for his vote. 5. The vote will be counted by the system.		
Includes:	1. Get a receipt. 2. Count the vote 3. Checks if voted before.		

Table 4

Use Case ID:	UC02		
Use Case Name:	Track Vote		
Created By:	A.Ewais	**Last Updated By:**	I. Faisal
Date	13-11 - 2015	**Last Revision Date:**	14-11 - 2015

Created:			
Actors:	Voter		
Description:	If the Voter wants to make sure his vote was counted and given to the right candidate, he uses the digital receipt he received after voting and will be able to view his vote.		
Trigger:	The voter views the elections results and proofs published online.		
Preconditions:	• The voter must have voted earlier and has the digital receipt		
Postconditions:	• The voter verified his vote. Any forgery must have been detected.		
Normal Flow:	1. The voter logs in to the elections website after the result is published. 2. He views the result. 3. He verifies using his digital receipt his vote was tallied correctly		
Includes:	None.		

Table 5

Use Case ID:	UC03		
Use Case Name:	Start to campaign and debate as officially		
Created By:	M.Gadalla	**Last Updated By:**	I. Faisal
Date Created:	13-11 - 2015	**Last Revision Date:**	14-11 – 2015
Actors:	Candidate		
Description:	The candidate is now considered as official nominees for the elections so they can start campaigning and debating.		
Trigger:	A candidate application and payment has successfully been completed		
Preconditions:	1. Candidate's registration completed successfully 2. Trustees have approved his candidacy 3. Candidate has paid the fees		
Post-conditions:	1. Web space created for candidate/party. 2. Candidate starts to access the website and post all of the desired material. 3. Viewers can access the website to check the candidate's webpage and see all of his debates and electronic campaigns.		

Normal Flow:	1. Candidate files application 2. Application approved 3. Candidate pays a fee 4. The webpage is created for the candidate. 5. All the new campaigns and debates are posted on the page. 6. Viewers see the page. 7. Votes are made after the electronic campaign.
Includes:	

Table 6

Use Case ID:	UC04		
Use Case Name:	Register		
Created By:	M.Gadalla	**Last Updated By:**	I. Faisal
Date Created:	13-11 - 2015	**Last Revision Date:**	14-11 – 2015
Actors:	Candidate		
Description:	The function allowing the candidate to nominate himself to run for the elections. Anyone who will fit in with the requirements will have the capability of registering for the elections. This check requires the availability of another function called check eligibility. Registration also includes registering for a space on the website to allow for debating and campaigning.		
Trigger:	Candidate's decision to run for the elections.		
Preconditions:	1. Candidate is in the citizens database 2. Candidate has filed a complete application		
Postconditions:	1. The candidate is either accepted or rejected		
Normal Flow:	1. The candidate/party decides to run for the elections 2. He goes through the registration process. 3. The elections trustees checks if the candidate is eligible or not.		

	4. He either gets accepted or rejected. 5. He then pays the fees and starts the campaign.
Includes:	• Check for candidate’s eligbility • Register for website space • Pay fees for website • Start to campaign officially

Table 7

Use Case ID:	UC05		
Use Case Name:	View Candidate List and Websites		
Created By:	Hussam El-Araby	Last Updated By:	Hussam El-Araby
Date Created:	14-11 - 2015	Last Revision Date:	14-11 - 2015
Actors:	User		
Description:	A user is able to view the lists of candidates and clicking on any open his website, where the post their campaign. There are two specialization of this use case, where the user can view the whole list of candidates or just a filtered subset.		
Trigger:			
Preconditions:	User visited the homepage of the elections system.		
Postconditions:	The list appears on homepage, and the website is opened when any candidate is clicked on.		
Normal Flow:	User opens homepage User views the list of candidates User may use a filter, to view a subset of the list User may click on one of the candidates in the list, where he will be redirected to the candidate’s website.		
Includes:	None.		

Table 8

Use Case ID:	UC06

Use Case Name:	View Live and Past Debates		
Created By:	Hussam El-Araby	Last Updated By:	Hussam El-Araby
Date Created:	14-11 - 2015	Last Revision Date:	14-11 - 2015
Actors:	User		
Description:	A user is able to view live debates. However, he is not able to send questions unless he is logged in. A user also is able to view the content of past debates.		
Trigger:			
Preconditions:	User visited the “Debates” page on the system		
Postconditions:	The list of debates and a short description (time and participant(s), and moderator(s)) is displayed. An indicator tells the user if debate is live. User is redirected to any of the debates is he clicked on any.		
Normal Flow:	User opens homepage User clicks on “Debates” in navigation bar of the website. User views list of live and past debates. User clicks on any of them to be redirected to it.		
Includes:	None.		

Table 9

Use Case ID:	UC07		
Use Case Name:	Voter sends questions to debate moderator		
Created By:	Hussam El-Araby	Last Updated By:	Hussam El-Araby
Date Created:	14-11 - 2015	Last Revision Date:	14-11 - 2015
Actors:	Voter, Debate Moderator		
Description:	During a live debate, a voter can send questions that he wants to ask the candidates, to the debate moderator. The content of the question can be any of the options the system provides for uploaded content: text, images, video, or other media.		
Trigger:			

Preconditions:	User opened a live debate.
Postconditions:	From voter perspective: Message indicates success or failure to send. From moderator perspective: One new message is received in the questions inbox.
Normal Flow:	From voter perspective: Voter opens homepage. Voter clicks on "Debates" in navigation bar of the website. Voter views list of live and past debates. Voter clicks on any of the live debates to be redirected to it. Voter send a question to the moderator through the send to moderator button, which redirects to a form. Voter uploads the question as content then send it to moderator. From moderator perspective: Moderator logs on system. Moderator starts participating in live debate Moderator receives a new message in questions inbox
Includes:	Upload content

Table 10

Use Case ID:	UC08		
Use Case Name:	Post content on website		
Created By:	Hussam El-Araby	Last Updated By:	Hussam El-Araby
Date Created:	14-11 - 2015	Last Revision Date:	14-11 - 2015
Actors:	Candidate		
Description:	After registration, each candidate is able to post content to his website that is provided to him (a platform similar to WordPress that we provide). The content that can be posted is in the form of text, images, videos, links, or other media, such as sound recordings.		
Trigger:			
Preconditions:	Candidate is successfully registered on system and his account added by system administrator		
Postconditions:	New content is posted on candidate website		
Normal Flow:	Candidate logs in Candidate opens his website Candidate uses the upload content tool		

	Candidate verifies that new content now appears on his website
Includes:	Upload content

Table 11

Use Case ID:	UC09		
Use Case Name:	Participate in Debate		
Created By:	Hussam El-Araby	Last Updated By:	Hussam El-Araby
Date Created:	14-11 - 2015	Last Revision Date:	14-11 - 2015
Actors:	Debate Moderator , Candidate		
Description:	The candidate is able to participate in timed debates. He can reply through live video feed, posting text and video or other media in chat, or by microphone. The debate moderator participates by asking the candidate(s) questions through sending questions in form of text, video, or other media through chat, through microphone, or through live video feed.		
Trigger:			
Preconditions:	A live debate is ongoing.		
Postconditions:	The participation of a candidate or a moderators is viewed by other candidates, moderators, and all viewers		
Normal Flow:	From moderator perspective: Moderator logs in Moderator joins the debate that is assigned to him Moderator asks a candidate or candidates a question Moderator uploads content of question, or uses live feed to communicate it through. From candidate perspective: Candidate logs in Candidate joins the debate he is part of Candidate replies to a question asked by the moderator through live feed or uploading content of the reply to the debate chat.		
Includes:	Upload content		

Table 12

Use Case ID:	UC10		
Use Case Name:	Moderate debate		
Created By:	Hussam El-Araby	Last Updated By:	Hussam El-Araby

Date Created:	14-11 - 2015	Last Revision Date:	14-11 - 2015
Actors:	Debate Moderator		
Description:	The debate moderator needs to keep the debate in an orderly manner by forcing time constraints and dealing with unruly participants. The system provide several ways to achieve this: **I) Grant Mic:** broadcasts the feed from the microphone of one or more debate participants to all the other participants and viewers. **II) Grant Chat:** broadcasts the uploaded content from one or more debate participants to all the other participants and viewers. **III) Grant Feed:** broadcasts the feed from the webcam of one or more debate participants to all the other participants and viewers.		
Trigger:			
Preconditions:	A live debate is ongoing.		
Postconditions:	Action of moderator affect the debate progression.		
Normal Flow:	Moderator logs in. Moderator participates in the debate he is assigned. Moderator decides to take a specific moderating action. Moderator uses the specified buttons/menus to enforce the action.		
Includes:			

Appendix C: Activity Diagrams

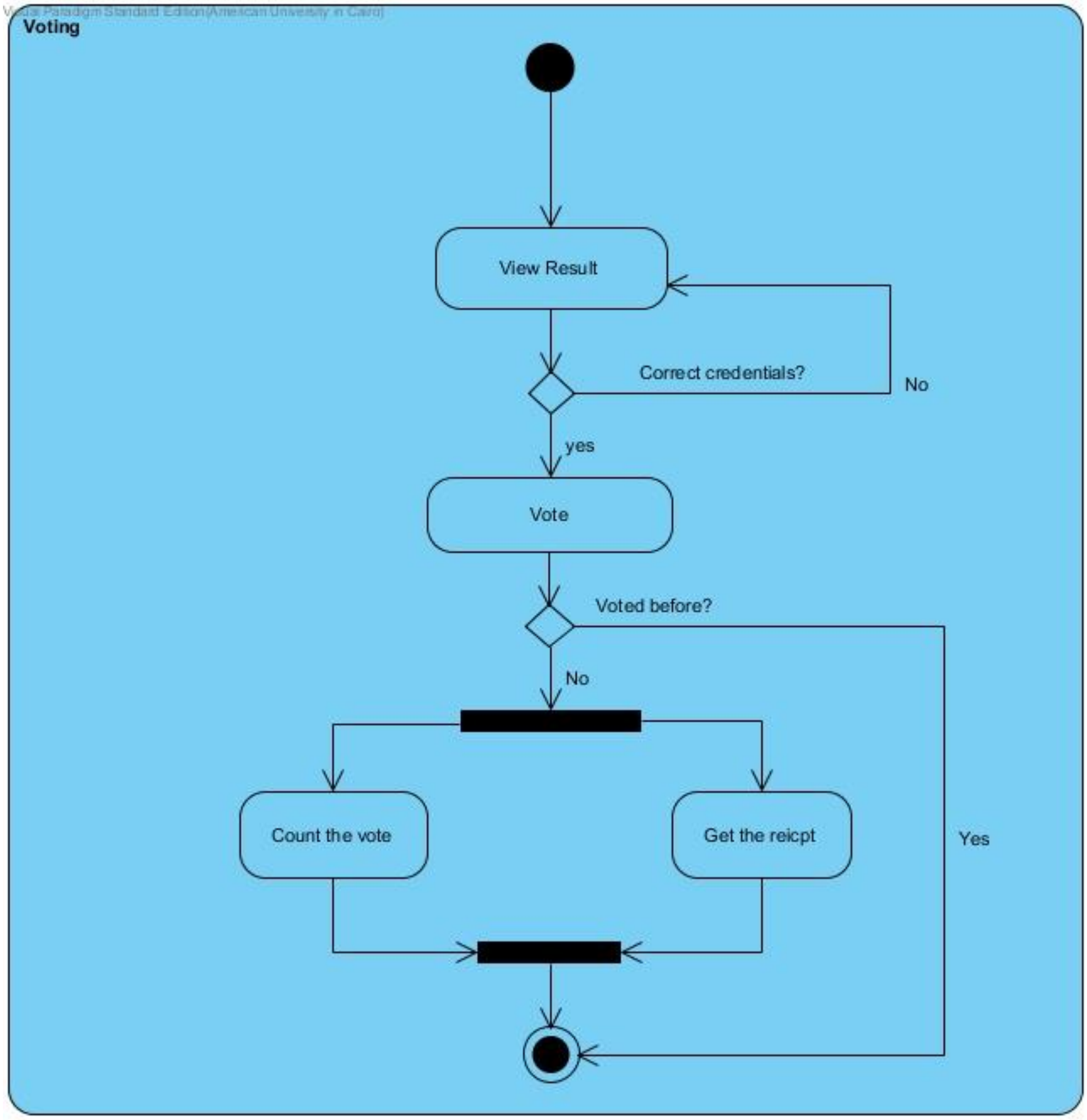


Figure 10: Voting Process Activity Diagram

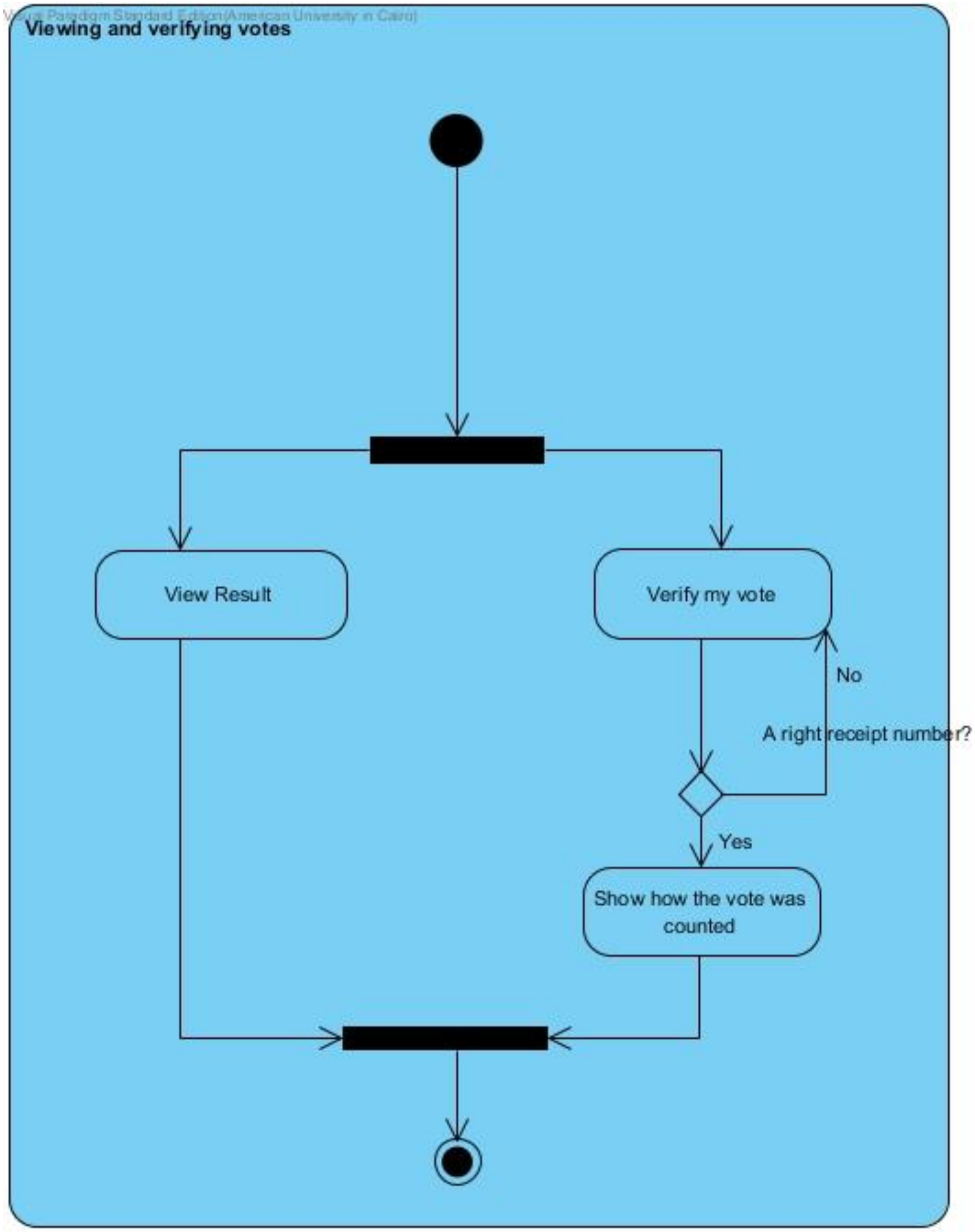

Visual Paradigm Standard Edition(American University in Cairo)
Viewing and verifying votes
View Result
Verify my vote
No
A right receipt number?
Yes
Show how the vote was counted

Figure 11: Viewing and verifying votes

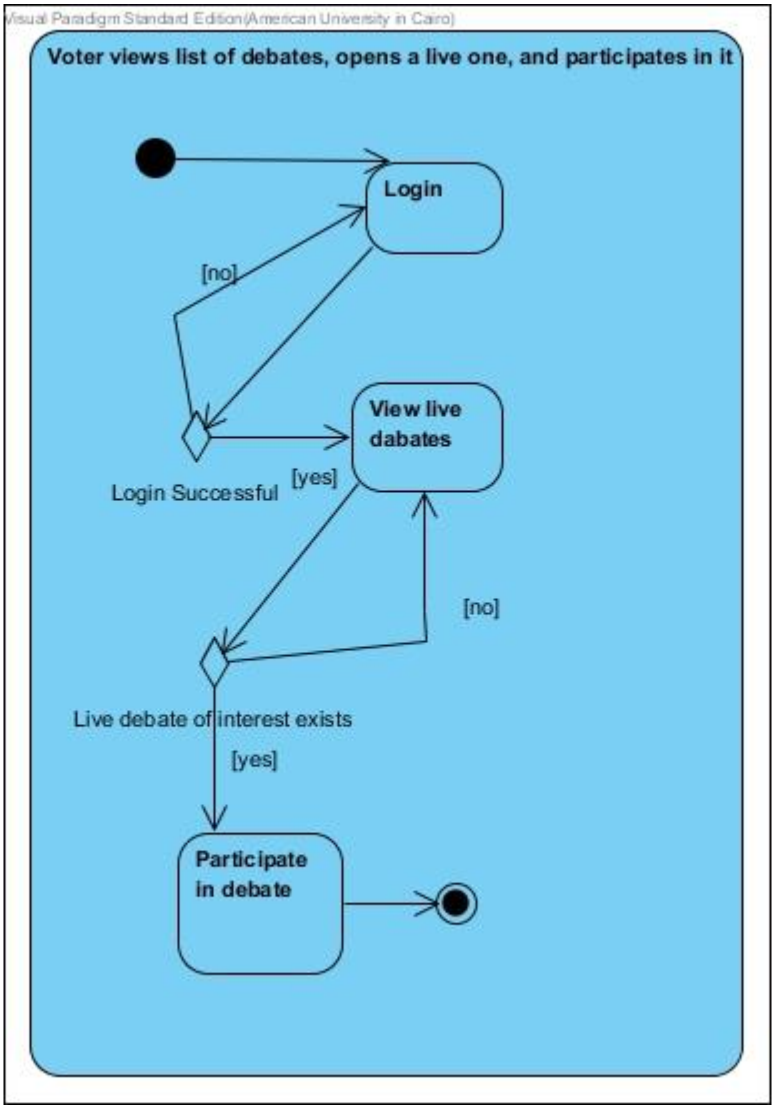

Visual Paradigm Standard Edition(American University in Cairo)
Voter views list of debates, opens a live one, and participates in it
Login
[no]
View live dabates
Login Successful
[yes]
[no]
Live debate of interest exists
[yes]
Participate in debate

Figure 12: Voter views list of debates, opens a live one, and participates in it

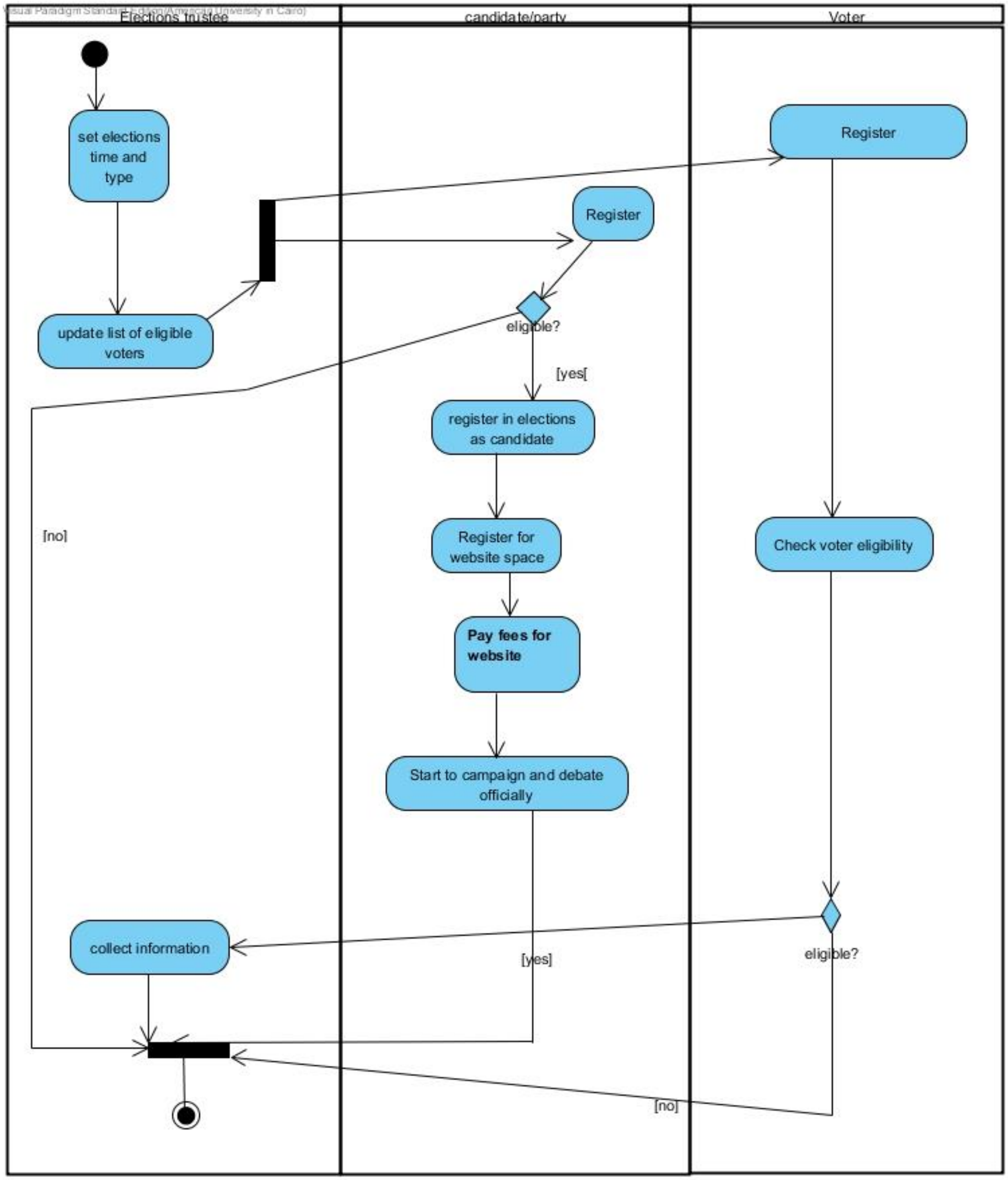

Visual Paradigm Standard Edition(American University in Cairo)
Elections trustee
candidate/party
Voter
set elections time and type
update list of eligible voters
Register
Register
eligible?
[yes[
[no]
register in elections as candidate
Register for website space
Pay fees for website
Start to campaign and debate officially
Check voter eligibility
eligible?
collect information
[yes]
[no]

Figure 13: Voting Registration Process Activity Diagram