



# **ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY**

## **SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING**

### **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**COURSE TITLE: - FUDAMENTAL OF SOFTWARE ENGINEERING**

**COURSE CODE:-CSE 3205**

**PROJECT TITLE: ONLINE ELECTION SYSTEM**

#### **MEMBERS**

<b><u>NAME</u></b>	<b><u>ID No</u></b>
1. Lencha Fikiru	A/UR4355/09
2. Yadeta Abera	A/UR4387/09
3. Amir Idris	A/UR4168/09
4. Bosona Motuma	A/UR4292/09
5. Bamud Ashemo	A/UR4713/09

SUBMISSION DATE:-29/12/2018

SUPERVISED BY: Lect. Antenh Alemu

# Contents

CHAPTER ONE .....	1
1.1 INTRODUCTION .....	1
1.2 BACKGROUND .....	1
1.3 STATEMENT OF THE PROBLEM.....	2
1.4 PURPOSE OF THE PROJECT .....	3
1.5 TEAM COMPOSITION .....	3
1. 6 OBJECTIVES OF THE PROJECT .....	4
1.6.1 General objective .....	4
1.6.2 Specific objective .....	4
1.7 FEASIBILITY .....	4
1.8 scope and limitation .....	5
1.8.1 Scope .....	5
1.8.2 Limitation.....	5
1.9 Significance of the project.....	5
1.10 Methodology .....	5
1.11 Development tools.....	6
1.12 Testing .....	6
1.12.1 Unit Testing.....	7
1.12.2 Integration Testing .....	7
1.12.3 Systems Testing.....	7
1.12.4 Acceptance Testing .....	7
1.13 Project plan .....	8
Chapter Two.....	9
Description of the Existing System .....	9
2.1 Major Function of the Current System .....	9
2.2 Users of the current System .....	9
2.3 Drawback of the current System (Manual) .....	9
2.4 Software Process Model .....	9
2.4.1 Waterfall Model.....	9

2.5 Business Rule of the Current System .....	10
CHAPTER 3 .....	11
PROPOSED SYSTEM .....	11
3.1 OVERALL DESCRIPTION .....	11
3.2 FUNCTIONAL REQUIREMENTS.....	11
3.3 NONFUNCTIONAL REQUIREMENTS .....	12
3.4 SECURITY REQUIREMENTS .....	12
3.5 SYSTEM MODEL .....	13
3.5.1 Use Case Scenario .....	14
3.6 OBJECT MODEL .....	17
3.6.1 DATA DICTIONARY .....	17
3.6.2 CLASS DIAGRAM.....	18
3.6.3 Use Case Diagram .....	19
3.7 DYNAMIC MODEL .....	20
3.7 .1 Sequence Diagram .....	20
3.7.2 Activity Diagrams .....	30
3.7.3 State diagram.....	26
CHAPTER FOUR .....	42
SOFTWARE DESIGN .....	42
4.1 INTRODUCTION .....	42
4.1.1 PURPOSE OF THE SYSTEM .....	42
4.1.2 DESIGN GOALS .....	42
4.1.3 DEFINITION, ACRONYM, AND ABBREVIATION.....	43
4.1.4 REFERENCES.....	43
4.1.5 OVERVIEW .....	43
4.2 CURRENT SOFTWARE ARCHITECTURE .....	43
4.3 PROPOSED SOFTWARE ARCHITECTURE .....	44
4.3.1 OVERVIEW .....	45
4.3.2 SUBSYSTEM DECOMPOSTION.....	45
4.3.4 COMPONENT DIAGRAM.....	46
4.3.5 DATABASE DESIGN .....	47
4.3.6 PERSISTENT DATA MANAGEMENT.....	48
4.3.5 ACCESS CONTROL AND SECURITY .....	53

4.3.6 BOUNDARY CONDITIONS .....	55
4.4 SUBSYSTEM SERVICES .....	57

## Tables

Table 1 Team composition.....	4
Table 2 Development .....	6
Table 3 schedule.....	8
Table 4 Business rule.....	10
Table 5 Data dictionary .....	18
Table 6 Access matrix for class's login, create account and view candidate's information.....	54
Table 7 Access matrix for classes, reset password, post information, set election .....	54
Table 8 Access matrix for classes, port data, Approve candidates, forget password .....	55
Table 9 Boundary condition failures .....	57
Table 10 Subsystem services .....	57

## Figures

Figure 1 Class Diagram .....	19
Figure 2 use case diagram.....	20
Figure 3 Create account sequence diagram.....	21
Figure 4 Post info sequence diagram.....	22
Figure 5 Approve candidate sequence diagram .....	23
Figure 6 Forget password sequence diagram .....	24
Figure 7 Set election sequence diagram .....	25
Figure 8 Vote sequence diagram .....	26
Figure 9 Create account state diagram.....	27
Figure 10 Forgot password State diagram .....	27
Figure 11 view candidate info State Diagram.....	28
Figure 12 Vote State Diagram.....	28
Figure 13 Change password state diagram .....	29
Figure 14 Login state diagram .....	29
Figure 15 Approve candidate state diagram.....	30
Figure 16 Post info state diagram.....	30

Figure 17 Admin Login activity diagram .....	31
Figure 18 Approve candidate activity diagram .....	32
Figure 19 Port data Activity diagram .....	33
Figure 20 Forget password Activity diagram.....	34
Figure 21 Create account Activity diagram .....	35
Figure 22 Post info activity diagram .....	36
Figure 23 Candidate Login activity diagram .....	37
Figure 24 Forget password Activity diagram.....	38
Figure 25 Candidate Info Activity diagram .....	39
Figure 26 change password Activity diagram.....	40
Figure 27 Vote Activity Diagram.....	41
Figure 28 Current OES architecture .....	44
Figure 29 Subsystem decomposition .....	45
Figure 30 Hardware or software mapping .....	46
Figure 31 Component diagram.....	47
Figure 32 Database design .....	48
Figure 33 Object Account mapping .....	49
Figure 34 Objet admin mapping.....	49
Figure 35 Object Account mapping .....	50
Figure 36 Object Candidate mapping.....	50
Figure 37 Object Result mapping.....	51
Figure 38 Object Candidate_info mapping .....	51
Figure 39 Object Vote mapping .....	52
Figure 40 Object student_profile mapping .....	52
Figure 41 object Voter mapping.....	53

# CHAPTER ONE

## 1.1 INTRODUCTION

Most institutions of higher learning such as universities and colleges provide an electioneering process where students elect their union leaders in a democratic manner. This is of great importance as it inculcates the principle of democracy into the students who are at the peak of their learning stage and would need these skills for better statesmanship. Proper management therefore is called for to provide foolproof processes which can be termed as free and fair in the standards of universal democracy and employment of Information Technology is a sure way to realize this. Online election is the application of web based technologies to the automation of voting processes. In online voting a voter cast their ballot from a remote terminal that is connected to the central database where actual processing of the ballot is done. Adama Science and Technology University (ASTU) online election system is a system that student's use to vote their student union president easily and simply. Online election systems have the advantage of providing convenience to the voter and reduce the time wasted in the queuing process at election centers. The system described is in form of a portal that is embedded on the Universities website. The system was developed using the prototyping due to the adaptive nature of web based applications and the system proved that a computerized solution is possible with elimination of human related faults that are a commonplace in employment of human clerks to manage the election process. Application of the online voting has resulted in many advantages in the efficiency of the entire electioneering process and reduced costs the university used to incur using the human clerk mechanism.

## 1.2 BACKGROUND

Adama Science and Technology University (ASTU) was first established in 1993 as Nazareth Technical College (NTC), offering degree and diploma level education in technology fields. Later, the institution was renamed as Nazareth College of Technical Teacher Education (NCTTE), a self-explanatory label that describes what the institution used to train back then: candidates who would become technical teachers for TVET colleges/Schools across the country.

In 2003, a new addition to NCTTE came about—introduction of business education. Nonetheless, the new entries were solely meant for similar purposes: these graduates were also expected to help overcome the existing dearth of educators in vocational institutions.

Although it is an institution with a history of only two decades, ASTU is known for its dynamic past. It has always been responsive to the realization of national policies: training of technologists at its infant stage, and later shifting to training of technical trainers, as well as business educators, to fill the gap in TVETs. Following its inauguration in May 2006 as Adama University, the full-fledged university started opening other academic programs in other areas—an extension to its original mission.

However, it was not until it was nominated by the Ministry of Education as Center of Excellence in Technology in 2008 that it opened various programs in applied engineering and technology. Following its renaming by the Council of Ministers as Adama Science and Technology University

in May 2011, the university has started working towards the attainment of becoming a center of excellence in science and technology, thereby allowing for the realization of goals set in the Growth and Transformation Plan (GTP). Currently, ASTU is setting up a Research Park, in collaboration with stakeholders and other concerned bodies: one of a kind in the Ethiopian context. The university is also venturing out to the wider community and is currently engaged in various joint undertakings.

Election of student union president in ASTU starting from its establishment time, has been held year by year using manual election system in which all students registered to vote, physical go to the election poll and elect for their choice candidates. In this type of election since all process of election has to be done manually by human, there has been so many errors and conflicts on the election results. Recently on the election held on 2010 E.C there was a conflict between students on the election results and most of student voice were canceled due to the mark they putted on the election paper and the manner in which they have putted right mark for the candidate they voted for.

Hopefully, Online election system we are currently working on will put an end mark to all this human errors and it will open new door for ASTU students to give their voice to candidate they want to make president of ASTU student union easily.

### **1.3 STATEMENT OF THE PROBLEM**

Students and management in university head many kinds of problem to vote their student union representative (student president), which leads them to excessive costs, unnecessary participation of labor from staff and waste their very expensive time.

#### **Cost problem**

- ✓ leading the management for extra cost

#### **Information**

- ✓ Students may not found the location of poll easily.
- ✓ Staff get into unimportant exhaust to inform the date of voting.

#### **Time management**

- ✓ It takes excessive time to present at the location of poll to vote their student union representative (students president)
- ✓ Human counting may have an error, or biased towards one of the candidates.
- ✓ Hard to trust the election result.

Therefore, hopefully ASTU online election system is vital to overcome all the problem that are listed above.

## 1.4 PURPOSE OF THE PROJECT

The core purpose of this project is developing an election system which helps student vote for the candidates they want for the election position admin of the election system hosted election for. And this project is also aimed at decreasing cost wasted on election from session to session. Below, we have listed basic purpose of this project from students and staff perspective. They are:

❖ **FROM STUDENTS PERSPECTIVE**

- ✓ Keep students from wastage of time.
- ✓ Students vote their student union representative (student president) easily.

❖ **FROM STAFF PERSPECTIVE**

- ✓ minimize the cost
- ✓ They inform the date of vote for their students through online system.

## 1.5 TEAM COMPOSITION

Team Composition			
NAME	Responsibility	Main Activity	Other
Lencha Fikiru	Team Leader	Coordinate and lead the members, divide tasks to team members, Design, Architecture, implementation and Documentation	Participate all activities
Yadeta Abera	General secretary	Requirement analysis, Design architecture, implementation and participates in documentation	
Amhir Edris	Member	Requirement Analysis, Design, Architecture, implementation and Documentation	
Bosona Motuma	Member	Requirement Analysis, Design, Architecture, implementation and Documentation	



Bamud Ashemo	Member	Requirement Analysis, Design, Architecture, implementation and Documentation	
--------------	--------	--	--

*Table 1 Team composition*

## 1. 6 OBJECTIVES OF THE PROJECT

### 1.6.1 General objective

The general objective of this project is to design and develop a dynamic voting system using website which allow the students to vote their student union representative (Students president) and any positons an admin of the system hosts election for.

### 1.6.2 Specific objective

- ✓ Develop a system to vote student union representative (Students president) online for students.
- ✓ Design and develop the database to store the election results.
- ✓ Allow the staff to insert new notification through the backend database server.

## 1.7 FEASIBILITY

Feasibility is a measure of how beneficial or practical the development of a project can be performed successfully. It is an analysis and evaluation of a proposed project in its simplest term. It also determines if it is feasible within the estimated cost and will be profitable .It is also called as feasibility study.

### 1.7.1 Technical Feasibility

Technical feasibility in this project accesses the current resources such as hardware and software which are required to accomplish the end user requirements within the allocated time. It also determines whether the relevant technology is stable and established. The software used in this project is Notepad ++ and the backend used is My Structured Query Language.

### 1.7.2 Operational Feasibility

Operational feasibility in this project accesses the extent to which the required software performs a series of steps to solve the problems of the user and user requirements. It also determines whether the problems anticipated in the end user requirements are of high priority. It also involves visualizing whether the software will operate after it is developed and be operative once it is installed. Automation makes like easier. The proposed system is very user friendly and user is much easily able to interact with the system. The operational feasibility is thus satisfied.

### 1.7.3 Economic Feasibility

Economic feasibility must be justified by cost and benefit. The cost required for the development and implementation of the system. The suggested project will give best return at the earliest with the proposed hardware and software. The proposed system is economically feasible as the system will be economical to the people and the cost of the project is less. Therefore the economic feasibility is thus satisfied.

### **1.7.4 Resource Feasibility**

Resource feasibility includes the study of available resources required to complete the project successfully within the allocated time. As all the resources are available to develop and complete the project successfully, resource feasibility is thus satisfied.

### **1.7.5 Legal Feasibility**

Legal feasibility determines whether the proposed system conflicts with legal requirements. As the proposed system complies with the local server and all the data protection regulation. Thus it satisfies legal feasibility.

## **1.8 scope and limitation**

### **1.8.1 Scope**

Online election system (OES) scopes:

- ✓ Online working website that can be accessed easily and easy to use.
- ✓ Database to store election votes.

### **1.8.2 Limitation**

- ✓ All of the feature of this web depend on internet connection so in order to get this service user need to have an internet connection.

## **1.9 Significance of the project**

Significance of this online election system for Adama Science and Technology universities students and concerned faculties are:

- ✓ Saving time of student from being wasted by queuing to vote for the candidate of their choice they are concerned with.
- ✓ Removing human error while counting voices to know who the winner is.
- ✓ Minimizing workload of staff and require less number of staff during voting
- ✓ Posting election result just on time as soon as the election time is up (Automatic counting).
- ✓ Inform the date of voting for all students at the same time through the web
- ✓ Less capital, less effort and less lobar intensive, as the primary cost and effort will focus primary on creating, managing and running a secure online portal.

## **1.10 Methodology**

To develop this project (online election System) since all its requirements are known and we don't expect requirement changes after the processes are started, we decided to use waterfall method.

## 1.11 Development tools

<u>Activities</u>	<u>Tools/ Software's</u>	<u>Purpose</u>
<u>Platforms</u>	Window 7/8 or 10, Linux, Mac, android devices	OS(operating System to run our webpage)
<u>Database server</u>	<u>MySQL</u>	To store election voices for the candidates and election results
<u>Browsers</u>	Google chrome, Opera, Firefox, Internet explorer and etc.(all internet browsers)	Is used to display webpage Of online election system
<u>Editors</u>	Sublime text, Adobe Dreamweaver, Adobe Photoshop	To edit our webpage sources and photos and images used in our system.
<u>Client side coding</u>	HTML, CSS, PHP	A language used to develop the webpage
<u>Documentation</u>	MS word 2016, MS Excel 2016	To document our project files
<u>Group working platform</u>	Slack	It allows the group members to access the source code remotely that will help the development to be quick and allows our instructor to see our project progresses.
<u>Design</u>	Enterprise architecture	To design SRS of Online election system

*Table 2 Development*

## 1.12 Testing

Testing is generally associated with the execution of programs. The emphasis is on the outcome of the testing, rather than what is tested and how it's tested. Testing is not a

one-step activity; execute the test. It requires planning and design. The tests should be reviewed prior to execution to verify their accuracy and completeness. They must be documented and saved for reuse.

System testing is the most extensive testing of the system. It requires more manpower and machine processing time than any other testing level. It is therefore the most expensive testing level. It is critical process in the system development. It verifies that the system performs the business requirements accurately, completely, and within the required performance limits. It must be thorough, controlled and managed.

Software development has several levels of testing.

- ✓ Unit Testing
- ✓ Systems Testing
- ✓ Acceptance Testing

### **1.12.1 Unit Testing**

The first level of testing is called unit testing which is done during the development of the system. Unit testing is essential for verification of the code produced during the coding phase. Errors were been noted down and corrected immediately. It is performed by the programmer. It uses the program specifications and the program itself as its source. Thus, our modules are individually tested here. There is no formal documentation required for unit-testing program.

### **1.12.2 Integration Testing**

The second level of testing includes integration testing. Here different dependent modules are assembled and tested for any bugs that may surface due to the integration of modules. Thus, the administrator module and various modules are tested here.

### **1.12.3 Systems Testing**

The third level of testing includes systems testing. Systems testing verify that the system performs the all assigned functions while meeting the specified performance requirements. It is performed by a team consisting of software technicians and users. It uses the Systems Requirements document, the System Architectural Design and Detailed Design Documents, and the Information Systems Department standards as its sources. Documentation is recorded and saved for systems testing. Since our main aim is at securing the system we have worked on this step carefully.

### **1.12.4 Acceptance Testing**

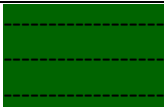
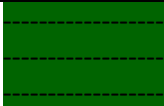
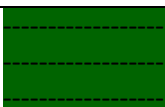
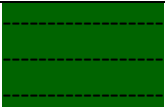
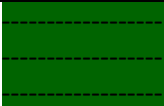
The final level of testing is the acceptance testing. Acceptance testing provides the users with assurance that the system is ready for production use; it is performed by the users. It uses the System Requirements document as its source. There is no formal documentation required for acceptance testing.

Systems testing are the major testing effort of the project. It is the functional testing of the application and is concerned with following,

1. Quality/standards compliance
2. Business requirements
3. Performance capabilities
4. Operational capabilities

## 1.13 Project plan

### SCHEDULE

PHASE	DURATION				
	Dec 10-dec 25 (2018)	Dec 20-Dec 30 (2018)	Jan 1- Jan 15 (2019)	Jan 15- Jan 20 (2019)	Jan 20-Jan 25 (2019)
REQUIREMENT GATHERING AND ANALYSIS					
DESIGN					
IMPLEMENTATION					
TESTING					
MAINTENANCE					

*Table 3 schedule*

# Chapter Two

## Description of the Existing System

### 2.1 Major Function of the Current System

Based on our conduct interview to ASTU Election board the board implied that the current system in our university is through manual interaction and also everything about the election will be informed to the student by printing the hard copy of the information and distributing over to the students.

Major Function of the Election board

- ✓ Providing information about the day election will conduct
- ✓ Informing the place for electing Student union president
- ✓ Familiarizing the candidates of election

### 2.2 Users of the current System

- ✓ Student like
  - Post graduate: Students who have completed their first degree and learning the second degree (graduated Student).
  - Under graduate: Student which are not yet graduated
- ✓ Service provider: Adama science and Technology University.
  - Providing there Service to the Student
  - Provide limitation time for the system to work
- ✓ Adama science and Technology university Election board
  - Informing the student about the election

### 2.3 Drawback of the current System (Manual)

In manual election system the student and office (Election Board) can't share information directly about the election. To communicate with each other, they use stamped papers and office signatures. This data sharing mechanism can cause many problems, for instance in terms of time we spend to get approval of paper and also the security of the information matters.

Students of science and technology needs almost all the services automated and easy to use. Being part of student's yearly routine, election is one of the services they like and engage if it's automated. Election system currently being practiced in ASTU doesn't meet the needs of students and is not trustful to count on.

### 2.4 Software Process Model

In the process model we will use waterfall model which we consider that it fit our projects criteria.

#### 2.4.1 Waterfall Model

Waterfall model is applicable to systems that are requirements fully understood for.

Waterfall model has the following approaches.

**Requirements definition:** this step involve understanding the very basic product requirements especially in terms of user interfaces. We will gather required data's and requirements for OES.

**System and software design:** At this stage we design entire OES software design.

**Implementation and unity testing:** After a partition is created and coded, at this stage we check it individually whether it works as specified in stage one (requirement definition)

**Integration and system Testing:** At this stage we combine all partition and check the entire system whether it fulfilled what is specified in requirement definition.

## 2.5 Business Rule of the Current System

The following table illustrates the business rules of our proposed system.

No.	Rule definition	Type	May change
Business rule-1	A person who wants to use this application must have a smart phone/computer	Static	Static
Business rule-2	There should be internet connection	Computation	Static
Business rule-3	The user should have reading and writing skill of English language to get exact information	Fact	Static

*Table 4 Business rule*

# CHAPTER 3

## PROPOSED SYSTEM

### 3.1 OVERALL DESCRIPTION

The system, Online election system we proposed to replace manual (traditional) election system found in ASTU helps ASTU students and concerned staff members in holding a peaceful and trustful election online in our University. This system we proposed have many advantages like time saving, avoiding wastage of costs and human labors when compared with the previous election system our University have been using since student Union president election began taking place up to now. In this proposed system student and concerned staffs login into the system using browser on any platform available to them and do what they are assigned to do.

In this chapter we discuss proposed online election systems use cases, functional and non-functional requirements, sequence and class diagrams, dataflow diagrams, data model and database diagrams.

### 3.2 FUNCTIONAL REQUIREMENTS

Enhancement to the online election system will primarily provide a more precise vote management tool that will establish accountability and improve data accuracy, and thus allowing voters to feel a greater level of confidence in the reported data. Appropriate behavior constitutes the functionality of a system and there is often a tight correspondence between particular requirements and particular functions of the solution system. The following represents a partial lists of functional requirements for the online election System:

- ✓ The system must provide voters with accurate data
- ✓ Metric reports of current/live votes must be provided
- ✓ The system should make use of tools available for users on the internet
- ✓ It must adhere to universities election rule( requirements)
- ✓ Ease of GUI use that can be accessed via web browser must be established
- ✓ The system must follow technical development standards supported on known operating systems such as Windows, Linux, and UNIX, in addition to future operating systems versions
- ✓ The system must supply a prototype or process to approve site customization
- ✓ Backup data restore capabilities should be granted
- ✓ The system must supply standard reports for decision making
- ✓ Audit trails of who made changes to the database must be maintained
- ✓ The system should allow voting administrators to make updates to the voter information database
- ✓ The system must provide standard error checking
- ✓ The system must provide data integrity checks to ensure data remains consistent and updated



### 3.3 NONFUNCTIONAL REQUIREMENTS

Nonfunctional requirements are requirements that are not specifically concerned with the functionality of a system. They normally place restrictions on the product being developed and the development process. Nonfunctional requirements may be regarded as parameters of functionality in that they determine how quickly, how accurately, how reliably, how securely, etc., functions must operate.

Some of the online election System's nonfunctional requirements are as follows:

#### **Reliability:**

- ✓ Defects in the election database file must be less than a very small positive value
- ✓ Defects contained in the collection server must be less than a very small positive value
- ✓ Defects in the master/server database must be less than a very small positive value
- ✓ When checking the database for errors, a 100% scan of the data is required, rather than selecting a sample set.

#### **Performance:**

- ✓ The system must be working at 100% peak efficiency during the voting process.
- ✓ Response and net processing time must be acceptable by user and by application
- ✓ The system should be allow adding more voters to allow a greater connectivity rate
- ✓ Due to the shortness of the voting timeframe, the system should support response time for addressing severe issues in less than 5 minutes

#### **Usability:**

- ✓ The system should provide friendly graphical Interface to ensure ease of use when end users utilize system functionality
- ✓ The system should provide documentation to inform users of system functionality and any change to the system

#### **Operation:**

- ✓ Assigned administrator to the system should manages the running system and respond to requests accordingly.

### 3.4 SECURITY REQUIREMENTS

Online election systems represent a great security challenge. Any successful attack would be highly visible, and thus, motivating much of the related hacking activity to date. Traditionally, security is incorporated in a software system after all the functional requirements have been addressed. Due to its criticality, security should be integrated in the software life cycle. Online election software security can be achieved if security is merged into election software functional requirements during the early stages of software requirements engineering. Although, security requirements are nonfunctional requirements, we deliberately avoided including them within the nonfunctional requirements due to the crucial role they play in the success of the any online voting system.

Below is a partial list of the Online Election system security requirements.

- ✓ The election system should include controls to prevent deliberate or accidental attempts to replace code such as unbounded arrays and strings
- ✓ The system should have zero-tolerant with regard to compromising
- ✓ Election process should not be subject to any manipulation including even a single vote manipulation
- ✓ The system should provide accurate time and date settings
- ✓ The system should not allow improper actions by voters and election officials
- ✓ The system should provide means for protecting and securing recounts of ballots cast in elections
- ✓ The system should not allow voter submissions to be observed or recorded in any way that is traceable to the individual voter
- ✓ The system should ensure that election results would be verifiable to independent observers. This implies that published election results correspond to the ballots cast by legitimate voters
- ✓ The system should not allow tampering with audit logs.

### 3.5 SYSTEM MODEL

To model a system the most important aspect is to capture the dynamic behavior. To clarify in details, dynamic behavior means the behavior of the system when it is running or operating. So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. So use-case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system or subsystem. A single use case diagram captures a particular functionality of a system. So to model the system we used the following use case diagrams.

The major concepts involved in use case model are:

- **Actor:** an actor represents anybody that interacts with the system.
- **Use case:** a use case is a sequence of action a system perform that yields a result to the actor.
- **Relation:** it defines the relationship between actor and a user.

In Online election system the actors that interact with the system are:-

Voter  
Administrator  
Candidate  
Registration System

#### Use case identification:

For online election system we are working on we have identified the following use cases:-

- Login
- Forgot password

- Create Account
- View Candidate info
- change password
- Post info
- Approve Candidate
- Set Election
- Port data
- Vote

### 3.5.1 Use Case Scenario

General entry conditions for both Admin and voters (actors)

- There must be a device like computer, smart phone or any device that have a browser from which they can access online election system web address and services.
- Internet connection is must.
- Searching for our website using our websites web address.

#### Scenario 1

##### Use case 1: Login

##### Preconditions:

→ If it is a user who is going to login, he/she must be registered by the system administrators prior to the Election Day.

**Actors:** Admin, Voters, Candidate

##### Flow of events:

1. Search's for ASTU online election system web address on their browsers
2. Provides username and password.
3. Clicks on the Login button placed under the Username and password forms.

##### Alternate Case:

If either wrong username or Password or both username and password are entered to the system, then the system detects it by comparing with data's inside database server and displays and error message and prompts user to enter correct data again. If user already have forgotten its password, it can clicks on "Forgot password?" button described under Scenario named "**reset password**".

**Exit Conditions:** Once the voter clicks on Logout it will leave the system.

#### Scenario 2

##### Use case 2: Forgot password

**Precondition:** User must open Login page of the system and remember it's username in order to request password reset.

**Actors:** Voters, Candidate, Admin

##### Flow of events:

1. Clicks on "Forgot password" button and open it.
2. Enter email or phone number found in database
3. Confirm weather that phone or email available to him/her.

4. Enter new password and click change

**Alternate case:** If a voter enters an email address that is not inside database(filled up information while registration), then the system checks and tell the voter that the email on which its requesting to reset is not valid and prompts to try again.

### **Scenario 3**

#### **Use case 3: Create Account**

Actor: Voter, Candidate

Flow of events:

1. Open online election system webpage
2. Clicks on create account
3. Insert ID and search to load all it's data
4. If it's data is found, choice to choose from phone and email address will be provided by the system to confirm ID.
5. Then user will confirm it ID
6. Fills in password in the fill form displayed
7. System will ask if voter wants to be registered as a candidate or not
8. Insert choice, if yes is choosed then pending request will be sent to admin for approval
9. Click create button at the end and account will be created successfully

Alternate Case: If the ID is not available in the registrar System the system will display error message and suggest the user to retry. If the sent request is not approved by the Admin the Student will not allowed to navigate Candidate page.

Exit Conditions: All the information filled by Student is stored inside the database.

### **Scenario 4**

#### **Use case 4: View Candidates info**

Actor: Voter

Flow of events:

1. Search's for ASTU online election system web address on their browsers
2. Then the system opens Home page containing the information posted by Candidate
3. Then voter views all the information of the candidates.

**Exit condition:** no change is made to the posted information

### **Scenario 5**

#### **Use case 5: Approve Candidate**

Actor: Administrator

Entry condition:

The administrator must have a valid username/password and login with its own username and

password which is recognize by the system.

Flow of events:

1. Admin inserts username and password
2. Admin login in to the system.
3. Selects Approve candidates and then the page having Candidate who sent the pending will be displayed.
4. The Admin approve the candidate that fit the requirement to be candidate.

Alternate case:

If an administrator try to Approve more than the predefined Number of candidate by the system then it will display the error message.

**Exit condition:** The system saves the updated data into database.

## **Scenario 6**

### **Use case 6: Post info**

**Actor:** Candidate

**Precondition:** Before going to post the information student should have to be approved by the admin so that the system Recognize him as Candidate and make Candidate page visible for the student

**Entry condition:**

The candidate must have a valid username/password and login with its own username and password which is recognize by the system.

Flow of events:

1. Candidate inserts username and password
2. Candidate login in to the system.
3. Clicks post info and fills in information to post
4. Clicks post.

Alternate case:

If must fill information left blank the system will notify the candidate to fill it.

**Exit condition:** The system saves posted info to database

## **Scenario 7**

### **Use case 7: Set election**

**Actor:** Administrator

**Flow of events:**

1. Admin inserts username and password
2. Admin login in to the system.
3. Click the Button 'Set election' found in the Navigation bar at the top of System home page interface
4. The system display the page having the dropdown list of the start date and time with

end date and time of the election.

5. Administrator fills in the election start time and date with end date and time.

6. Click on 'Set election' button.

**Exit condition:** The system saves start date and time with end date and time of election data into database.

## Scenario 8

### Use case 8: Port Data

**Actor:** Administrator

Entry condition:

The administrator must have a valid username/password and login with its own username and password which is recognize by the system.

Flow of events:

1. Admin inserts username and password
2. Admin login in to the system.
3. Open the menu 'Port data' found in the Navigation bar at the top of System and click on 'Port Data'.
4. The system access information it need from Registration system fetch to its own database.

Alternate case:

If an administrator try to port data repeatedly the system display that the data is already fetched message.

Exit condition: The system saves the updated data into database.

## Scenario 9

### Use case 9: Vote

**Actor:** Voter, Candidate

Entry condition:

Both the Voter and Candidate must have a valid username/password and login with its own username and password which is recognize by the system.

Flow of events:

1. user inserts username and password
2. User login in to the system.
3. Click the button called vote which is only visible when the election time started
4. View candidate info and select candidate

Exit condition: The system saves the vote data into database. And finally the vote will disabled again.

## 3.6 OBJECT MODEL

### 3.6.1 DATA DICTIONARY

Class	Attributes	Operations	Description
-------	------------	------------	-------------

Voter	Voter_ID, Fname, Mname, Lname, Gender, email, department, year	Vote() change_password(), create_Account() Forgot_password()	Used to vote
Candidate	Cnadidate_ID, CFname, CMname, CLname, Department, email, year	Vote(), Post_Info(), change_password()	Used to post information, vote and see result
Admin	Admin_ID, AFname, AMname, ALnaame, email, password,	Approve_Candidate() Change_password() Set_election()	Used to respond to candidates request, set election
Account	Account_ID, email, phone, password	Login()	Used for authentication purpose
Student Profile	Stud_ID, FName, Mname, Lname, Gender, email, phone, school, department, year	Portdata()	Used to identify and provide full student profile during registration/ account creation
Vote	Vote_ID, candidate_ID	Vote()	Used to store vote results
Result	Result_ID, Total_Voice	viewResult()	Used to count all vote data's and tell who the winner is.
Set election	Election_ID, startdate, starttime, endtime	Setelection()	Used to set election start date and time and election end time
Candidate Info	Info_ID, Background(), Department, year, short_videos, photo, canidate_ID	View() setInfo()	Used to store candidates information and show it to voters

*Table 5 Data dictionary*

### 3.6.2 CLASS DIAGRAM

The Class diagram captures the logical structure of the system; the classes and things that make up the model. It is a static model, describing what exists and what attributes and behavior it has, rather than how something is done. Class diagrams are most useful to illustrate relationships between classes and interfaces. It shows the classes of the system and their interaction which are typically used to:-

- ✓ Explore domain concept
- ✓ Analyze requirement in the form of conceptual analyses model

A class diagram is typically modeled rectangles with three-section:

- ✓ The top one indicates the name of the class

- ✓ The middle one lists the attributes of the class and
- ✓ The third one lists the methods

The class Diagram for ASTU online Election system is as follow:-

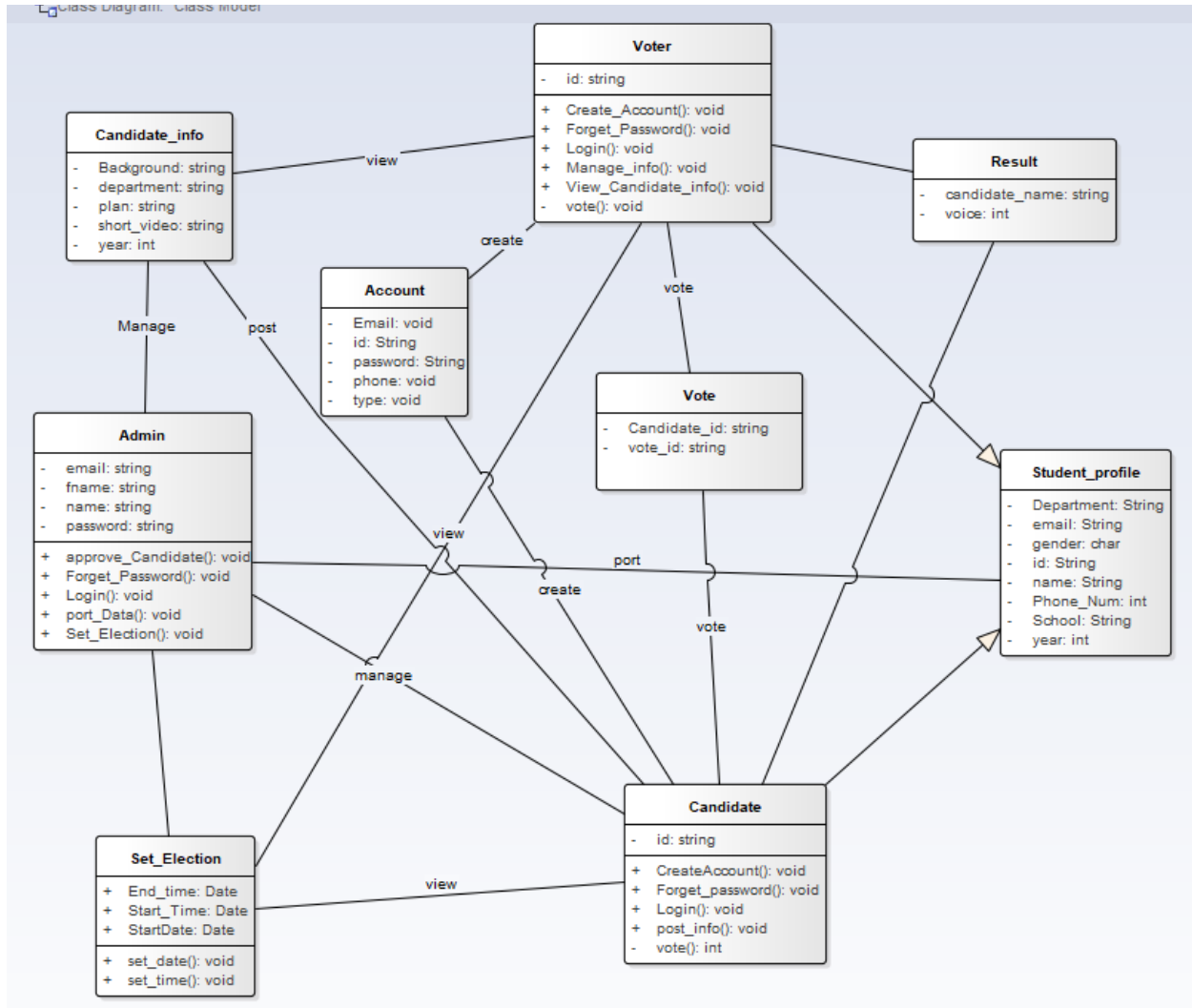
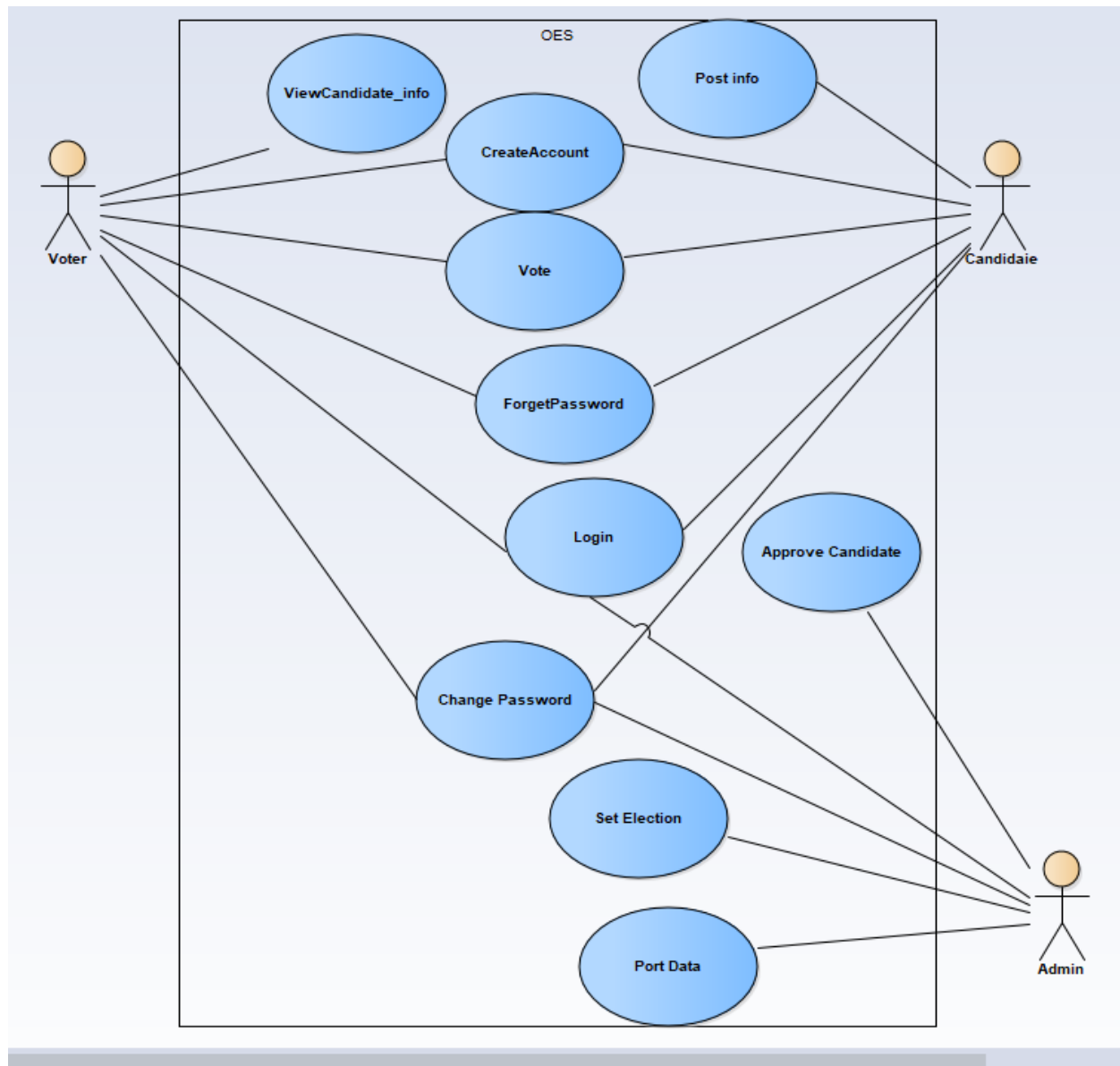


Figure 1 Class Diagram

### 3.6.3 Use Case Diagram

Use case diagram helps to identify and clearly define what content of the system the user/actor is allowed to access. Since use case is a boundary of the system, use case diagram shows actors with relationship they have with the content of the system. Use case diagram for ASTU online election system is presented in figure below:





*Figure 2 use case diagram*

## 3.7 DYNAMIC MODEL

### 3.7 .1 Sequence Diagram

A sequence diagram links use case with objects. It shows the interaction between participating objects in a given use case. It is helpful to identify the missing objects that are not identified in the analysis object model. From the use case and the class diagrams shown in the previous section the sequence diagrams of the system is shown as follows

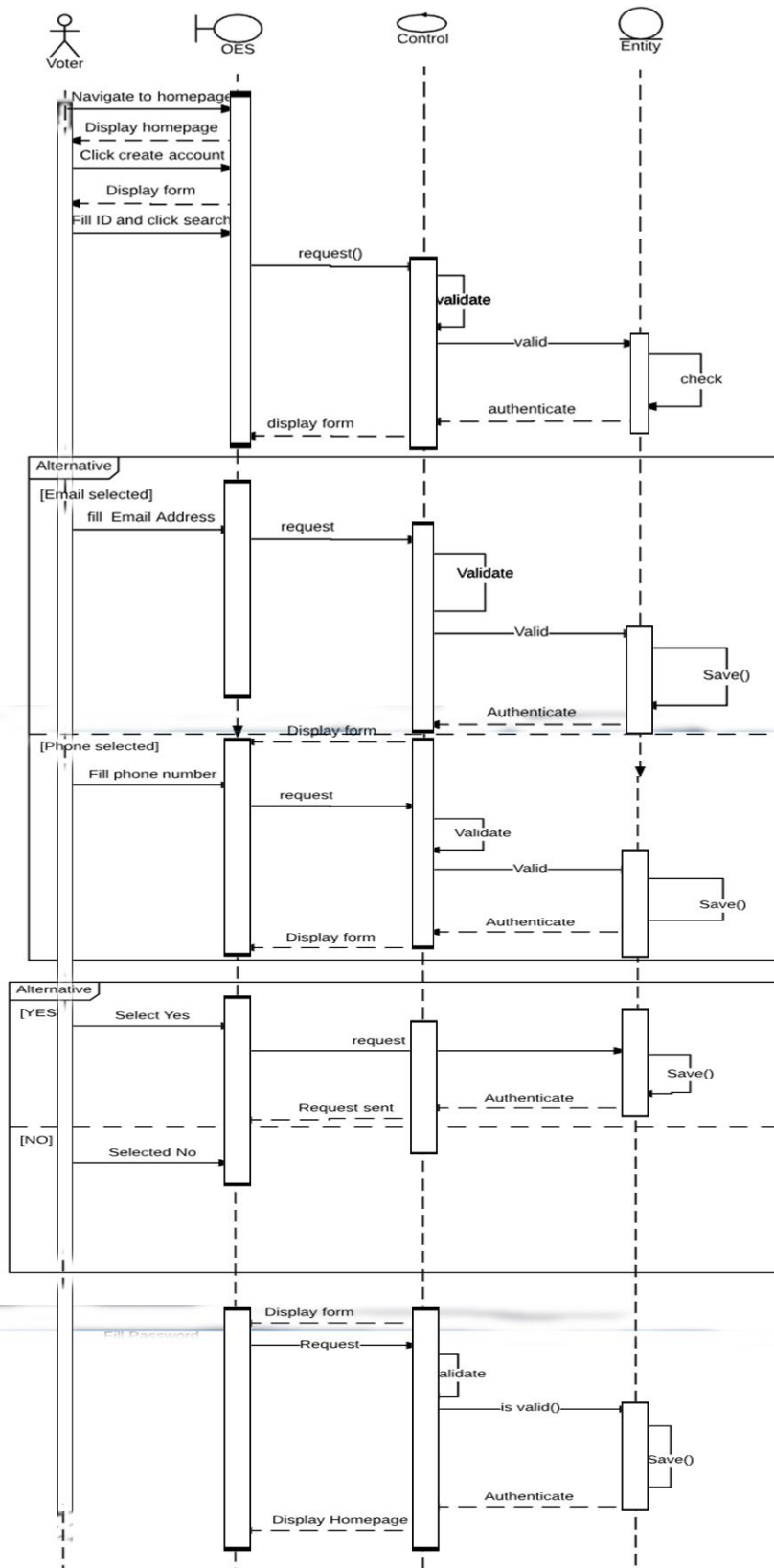


Figure 3 Create account sequence diagram

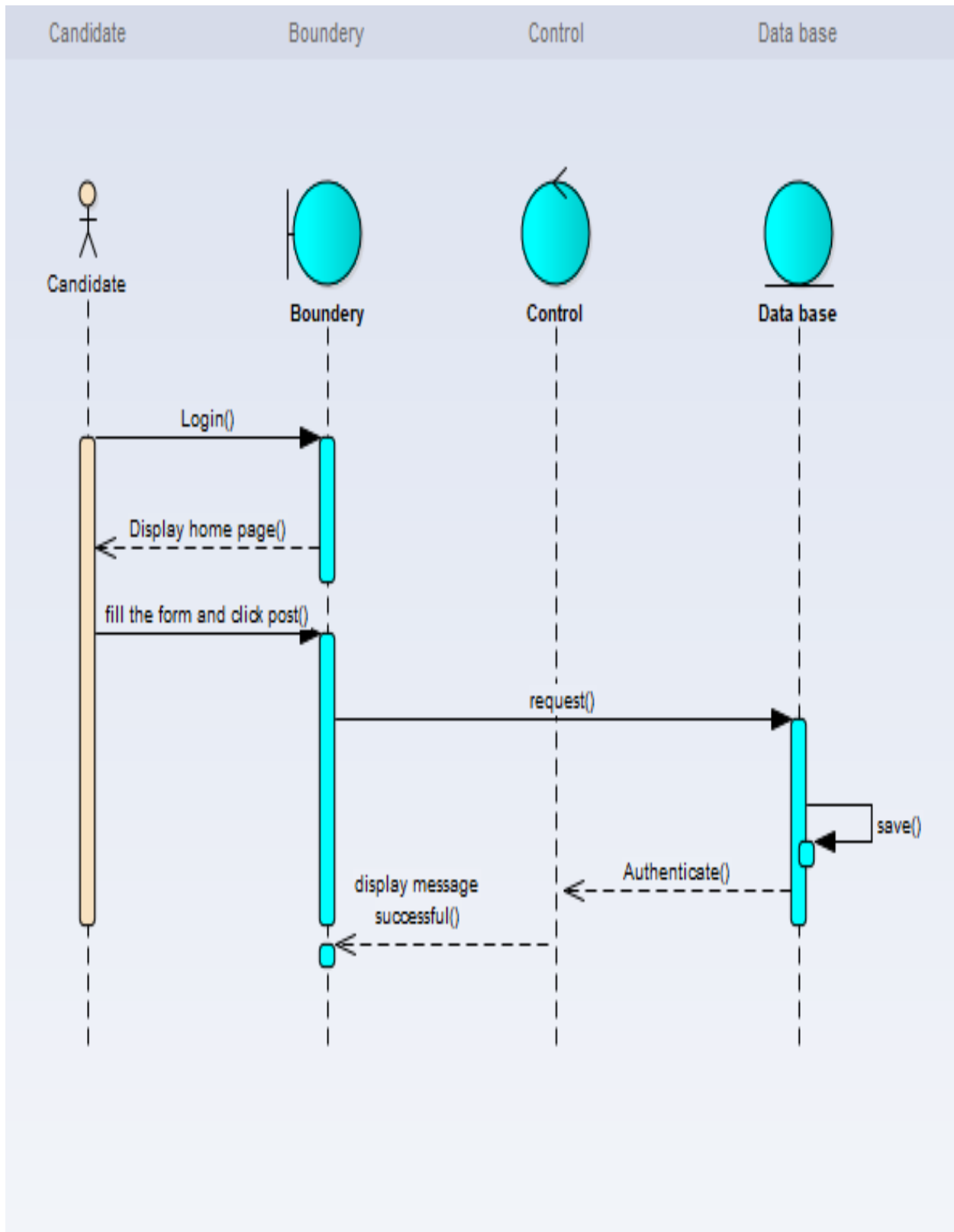


Figure 4 Post info sequence diagram

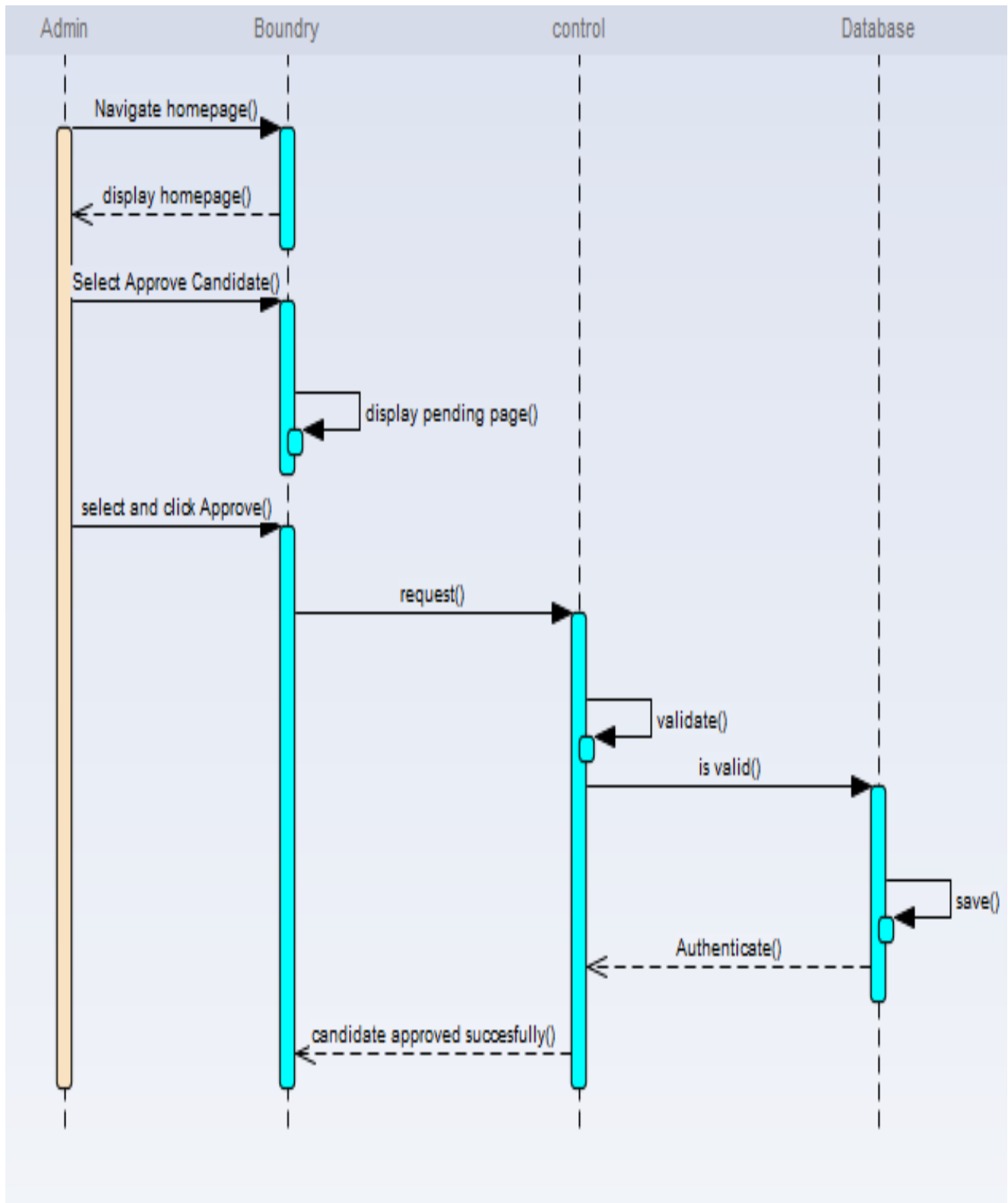


Figure 5 Approve candidate sequence diagram

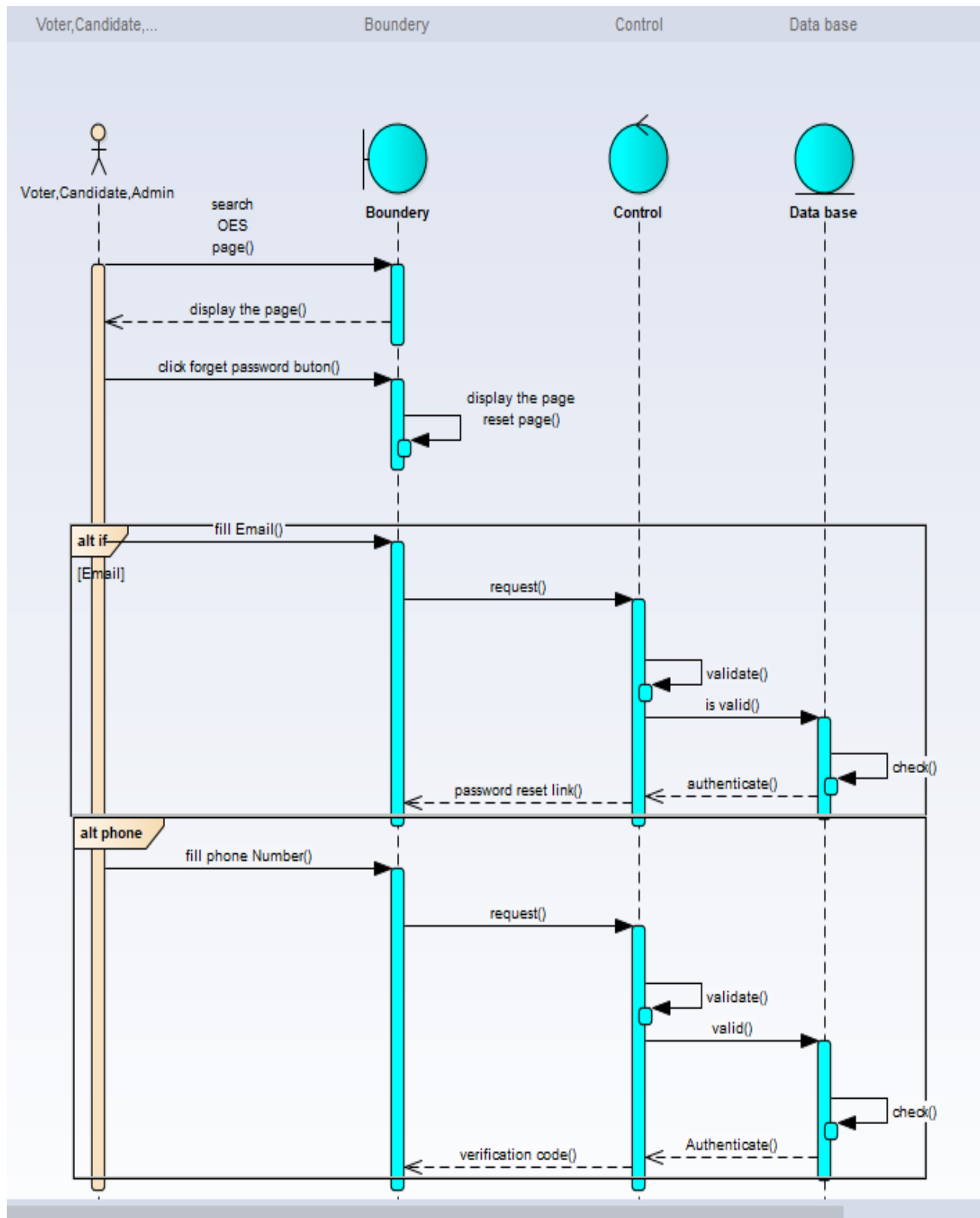


Figure 6 Forget password sequence diagram

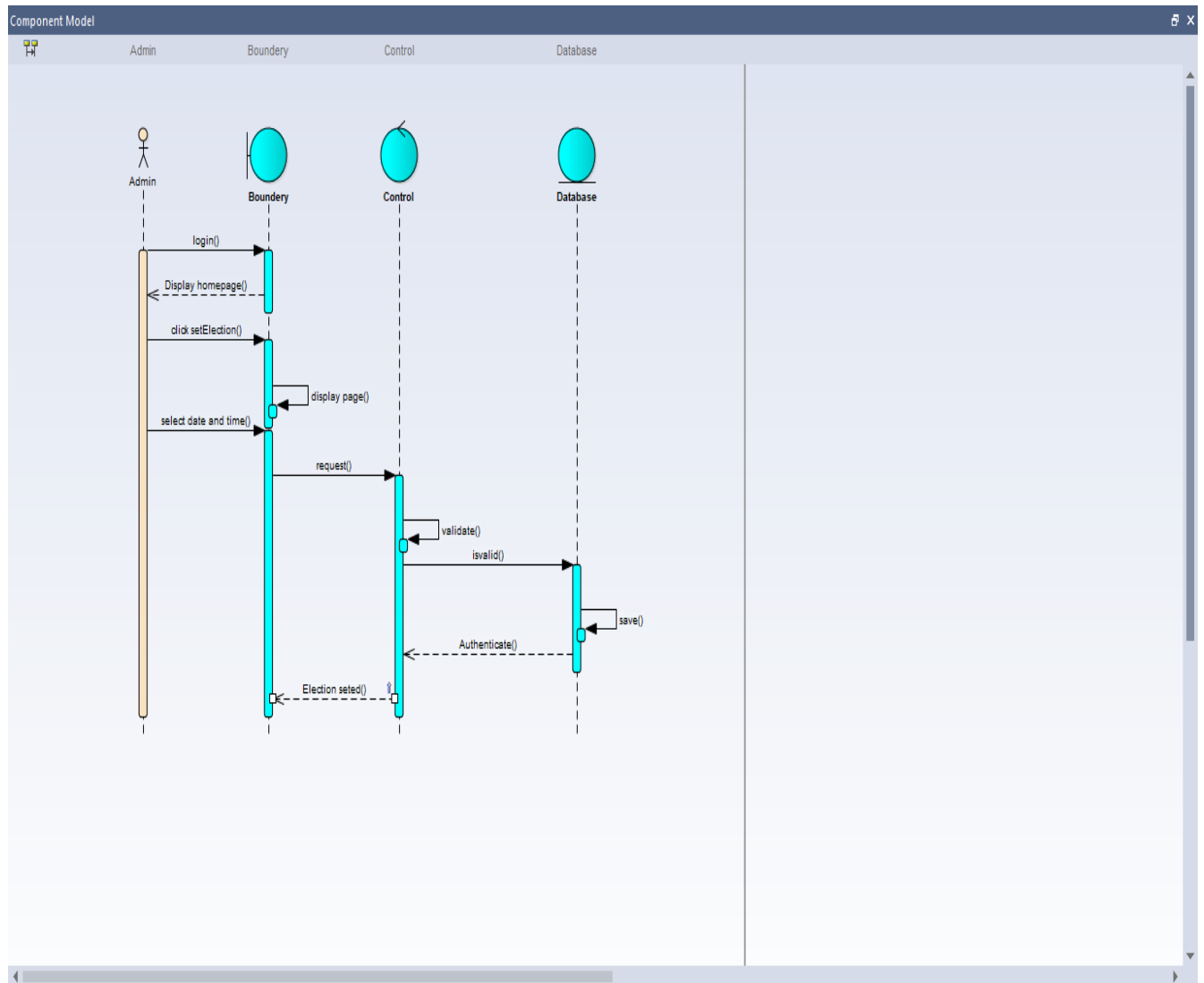
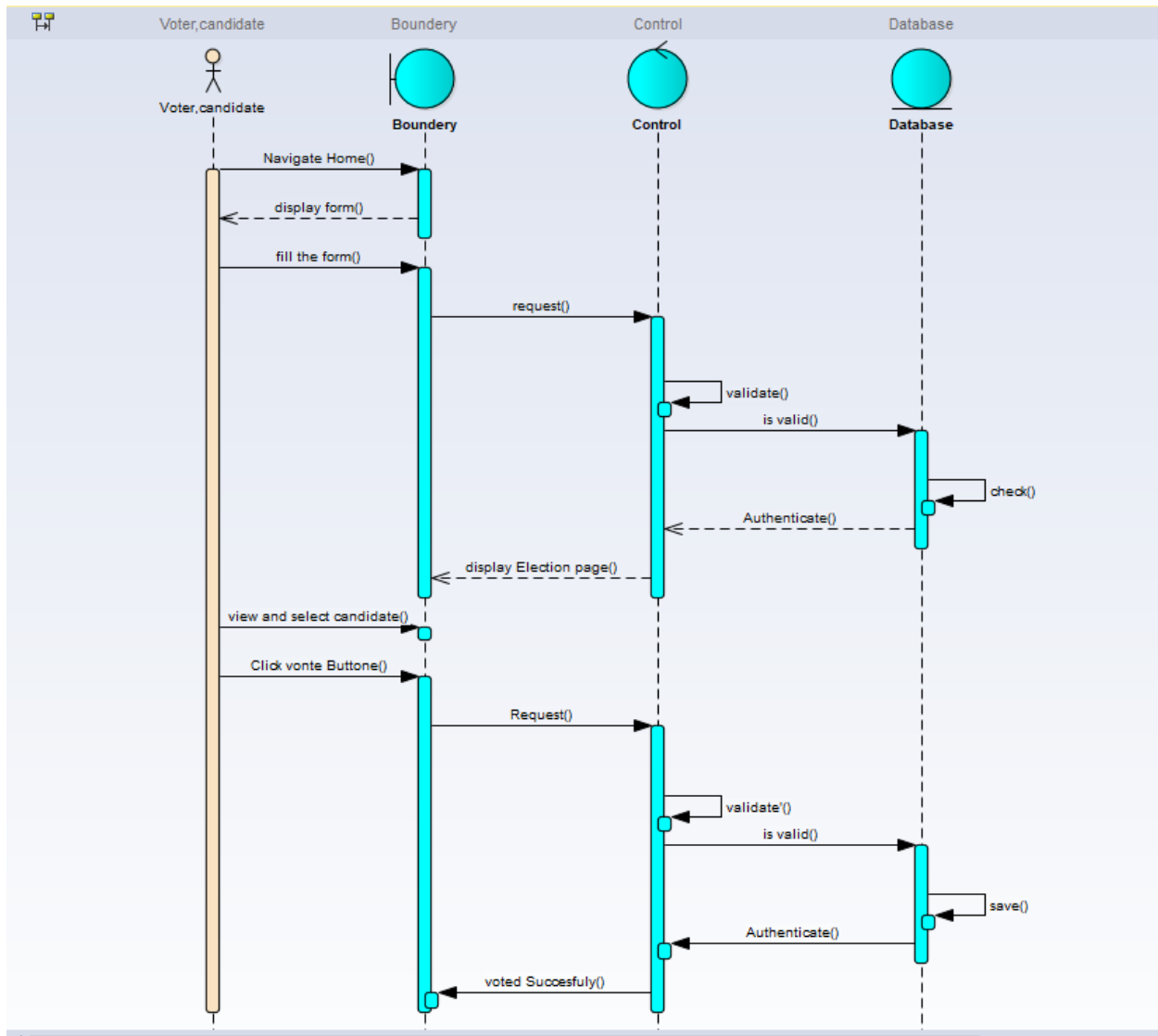


Figure 7 Set election sequence diagram



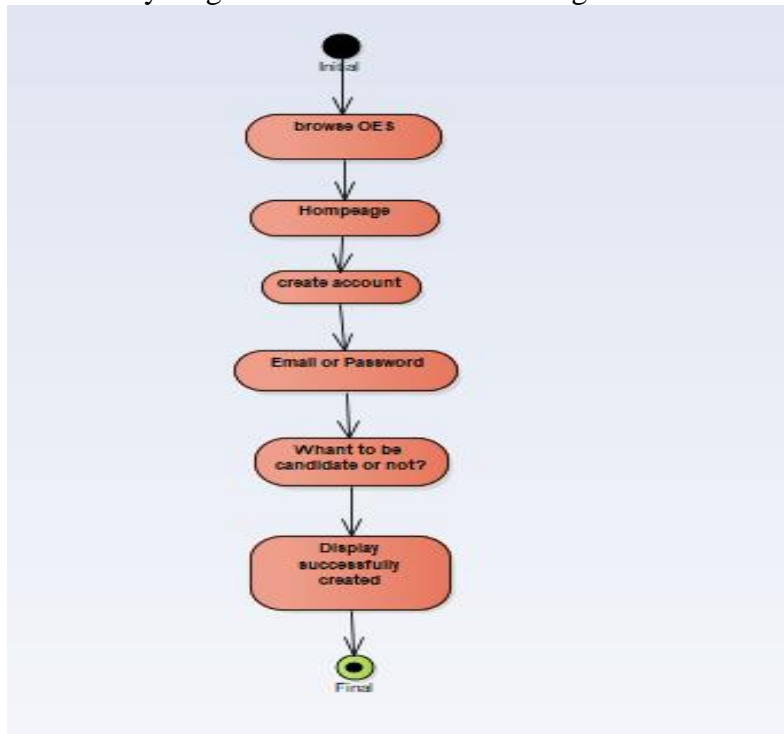
**Figure 8** Vote sequence diagram

### 3.7.3 State diagram

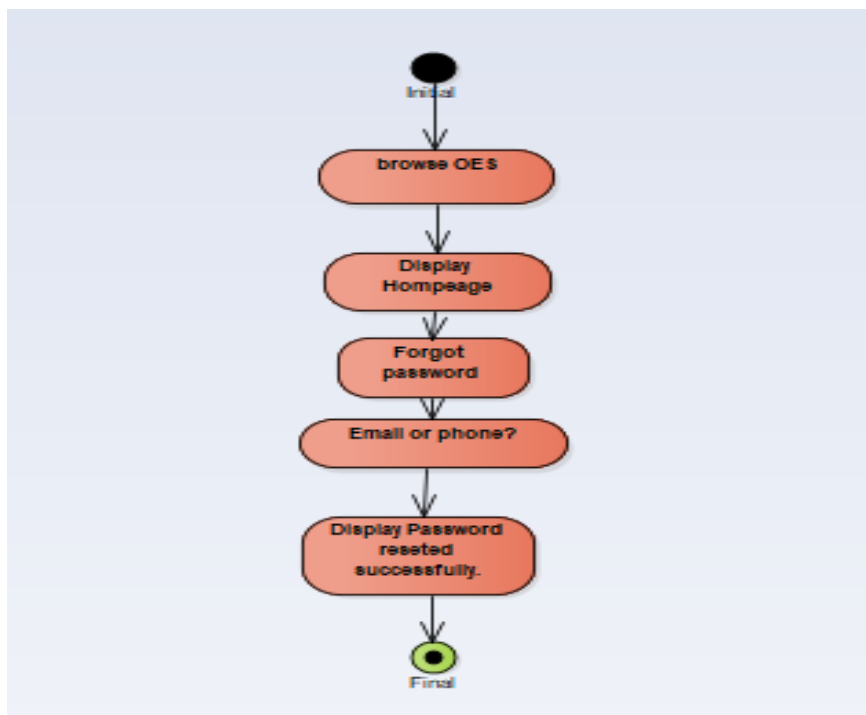
State chart diagram is one of the five UML diagrams used to model dynamic nature of a system. They define different states of an object during its lifetime. And these states are changed by events. So State chart diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events. State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination. State chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system. The main purposes of using State chart diagrams:

- ✓ To model the dynamic aspect of a system.
- ✓ To model the lifetime of a reactive system.
- ✓ To describe different states of an object during its lifetime.

Activity diagrams of ASTU online voting looks like the following pictures.

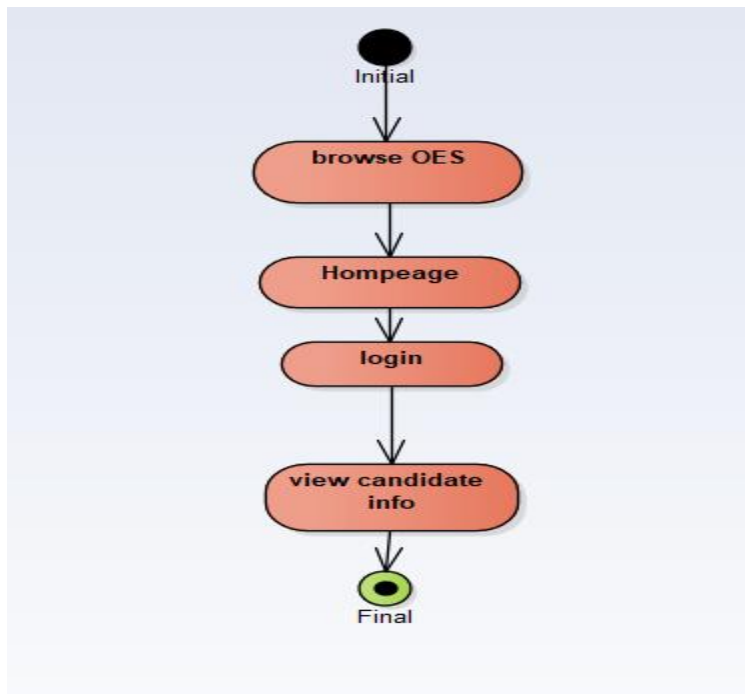


*Figure 9 Create account state diagram*

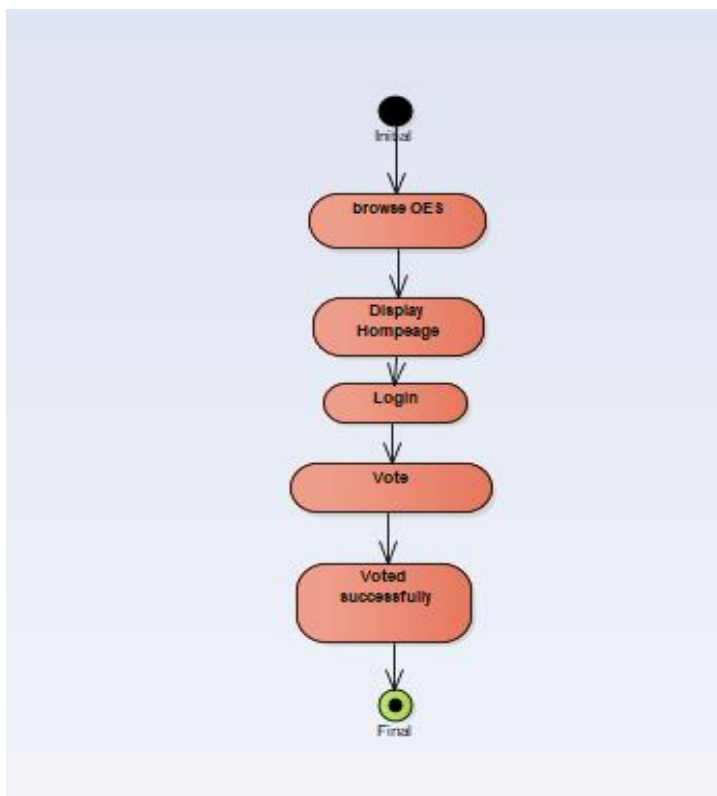


*Figure 10 Forgot password State diagram*

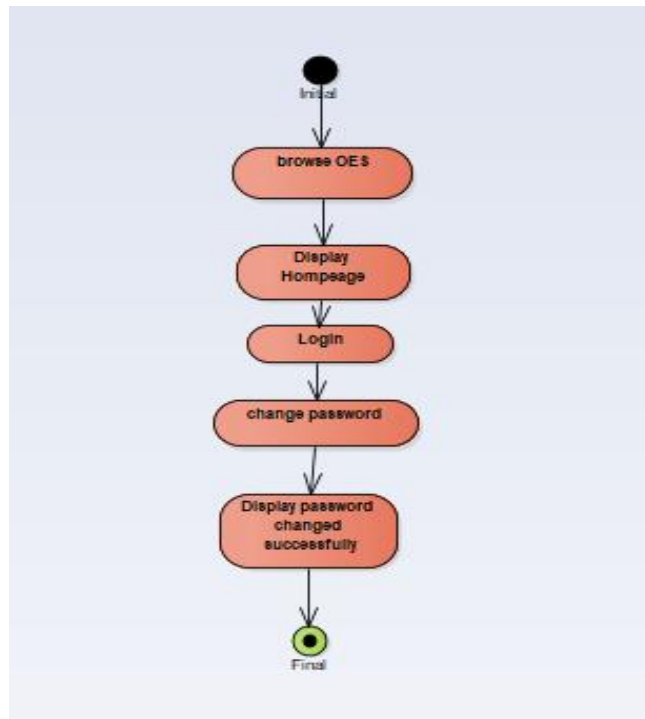




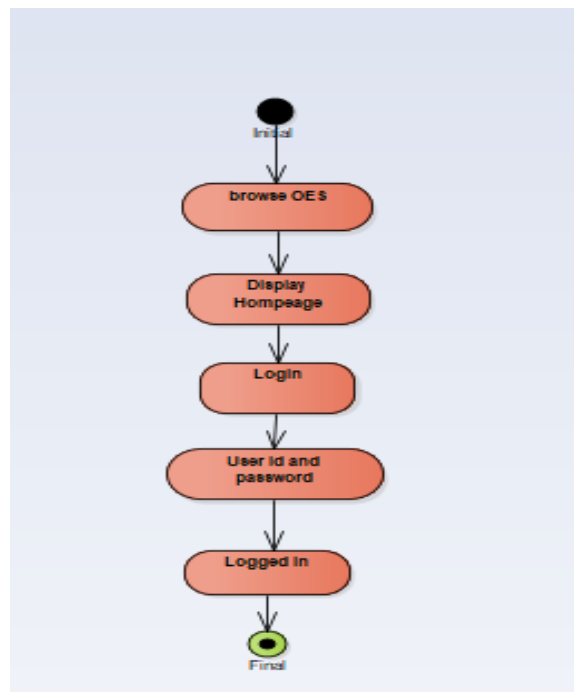
*Figure 11 view candidate info State Diagram*



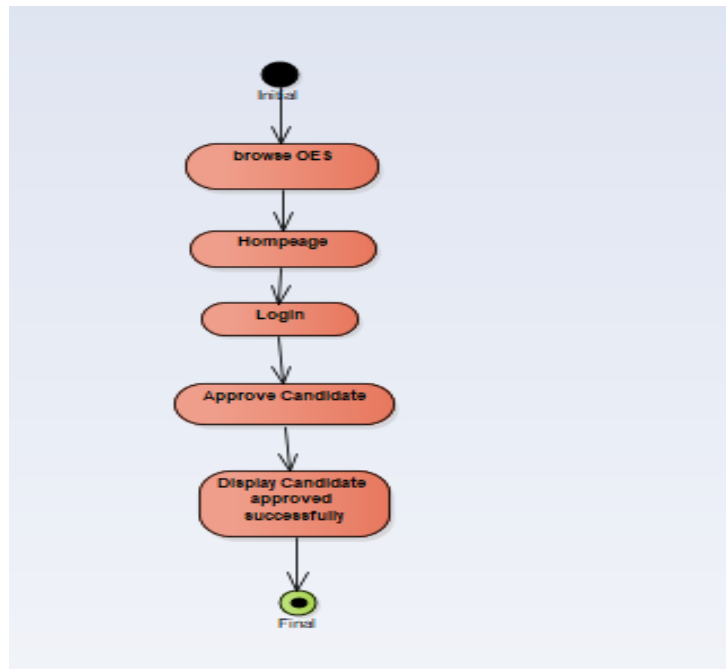
*Figure 12 Vote State Diagram*



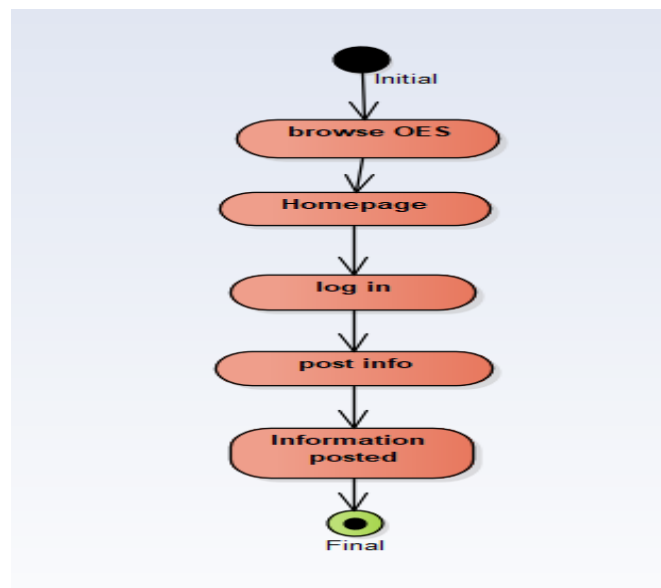
*Figure 13 Change password state diagram*



*Figure 14 Login state diagram*



*Figure 15 Approve candidate state diagram*



*Figure 16 Post info state diagram*

### 3.7.2 Activity Diagrams

Activity Diagrams consist of activities, states and transitions between activities and states.

Activity Diagrams describe

- ✓ how activities are coordinated to provide a service
- ✓ the events needed to achieve some operation
- ✓ how the events in a single use case relate to one another
- ✓ How a collection of use cases coordinate to create a workflow for an organization.
- ✓ the service can be at different levels of abstraction

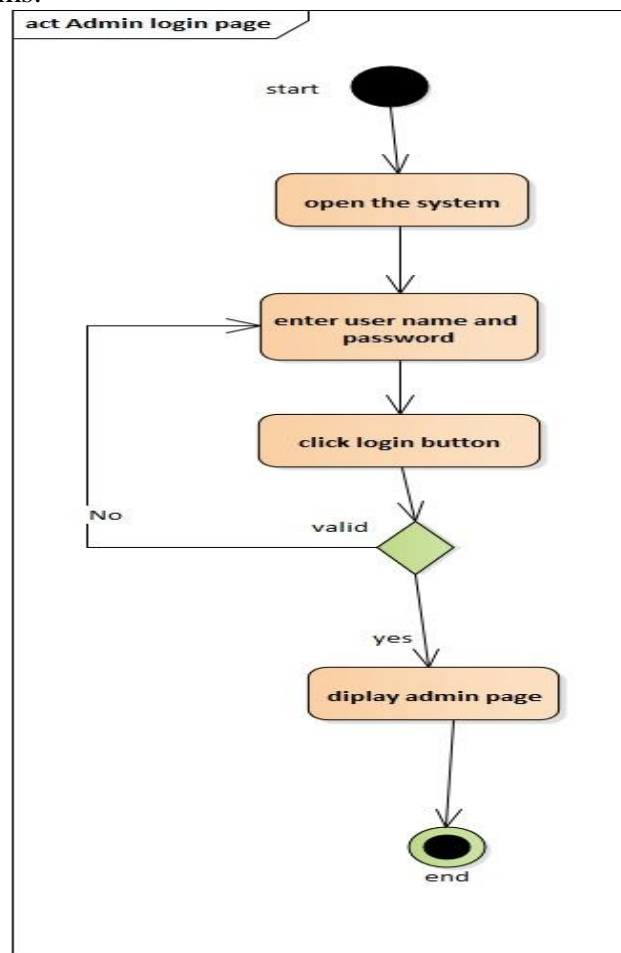
#### Activity Diagrams

- ✓ focus on the flow of activities involved in a single process
- ✓ Show how activities depend on one another
- ✓ Capture activities that are made up of smaller actions.

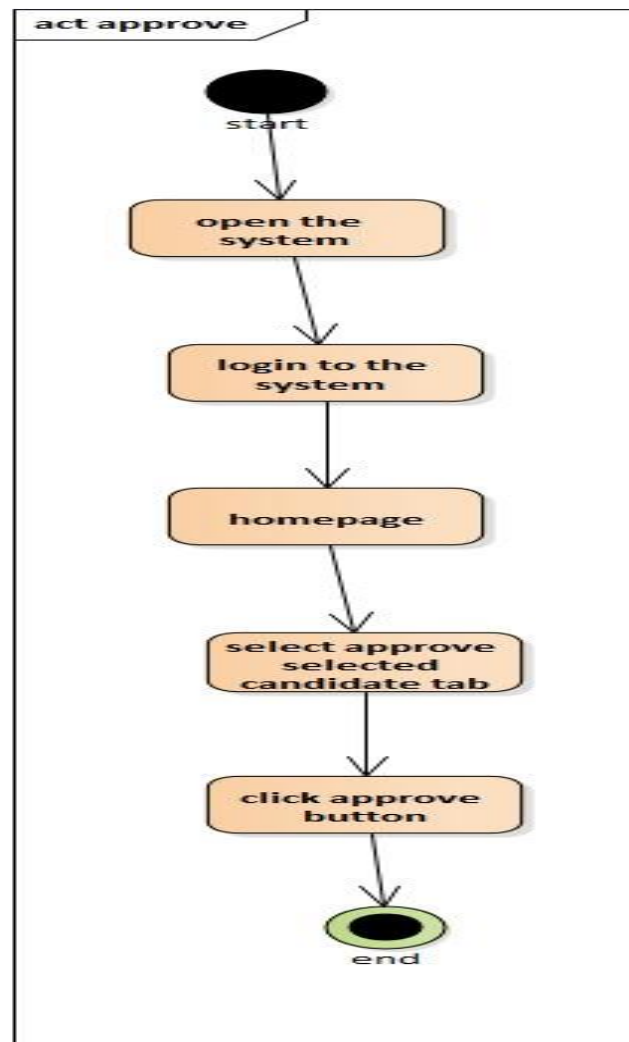
Activity diagram can be used for:

- ✓ Modeling workflow by using activities.
- ✓ Modeling business requirements
- ✓ High level understanding of the system's functionalities.
- ✓ Investigating business requirements at a later stage.

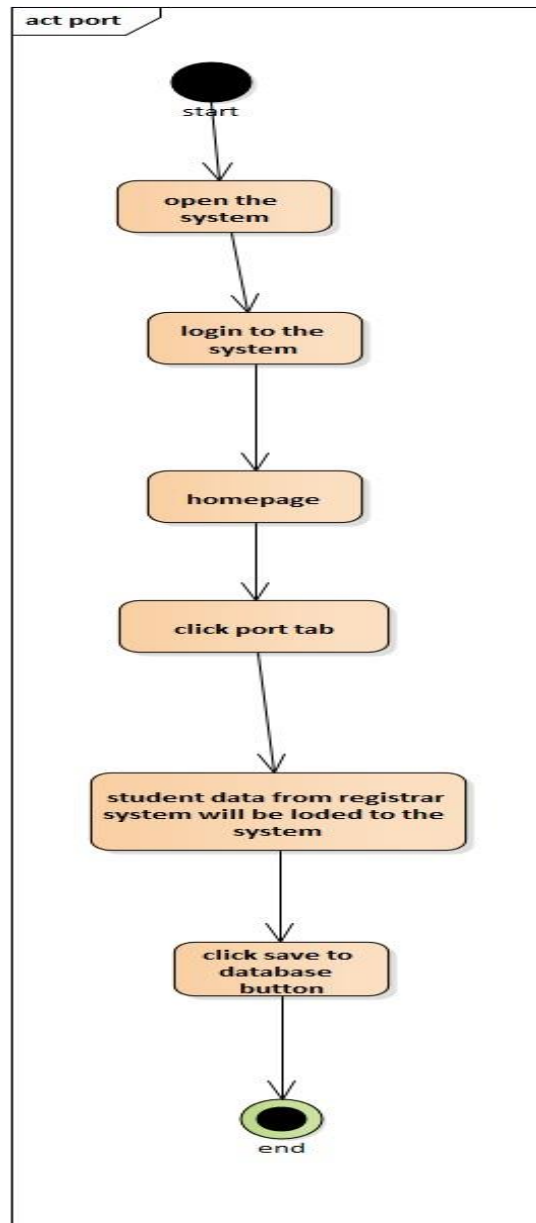
OES Activity diagrams:



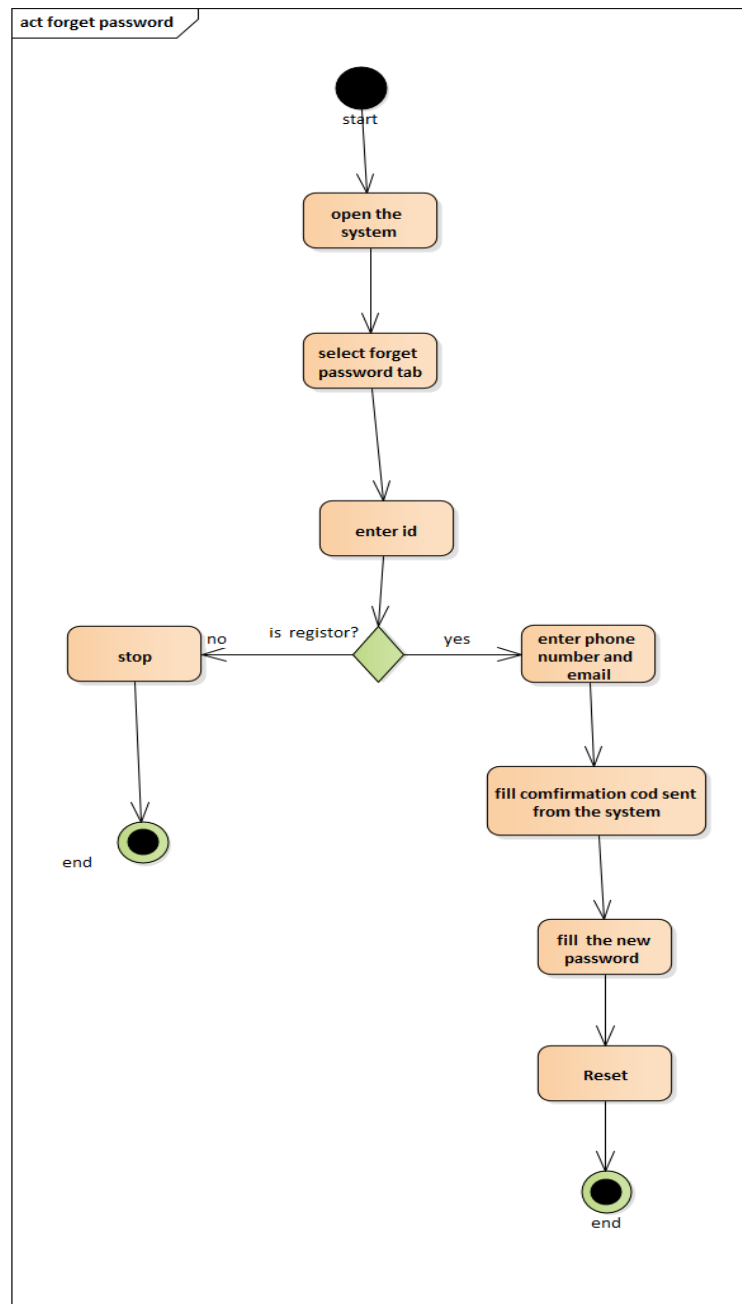
*Figure 17 Admin Login activity diagram*



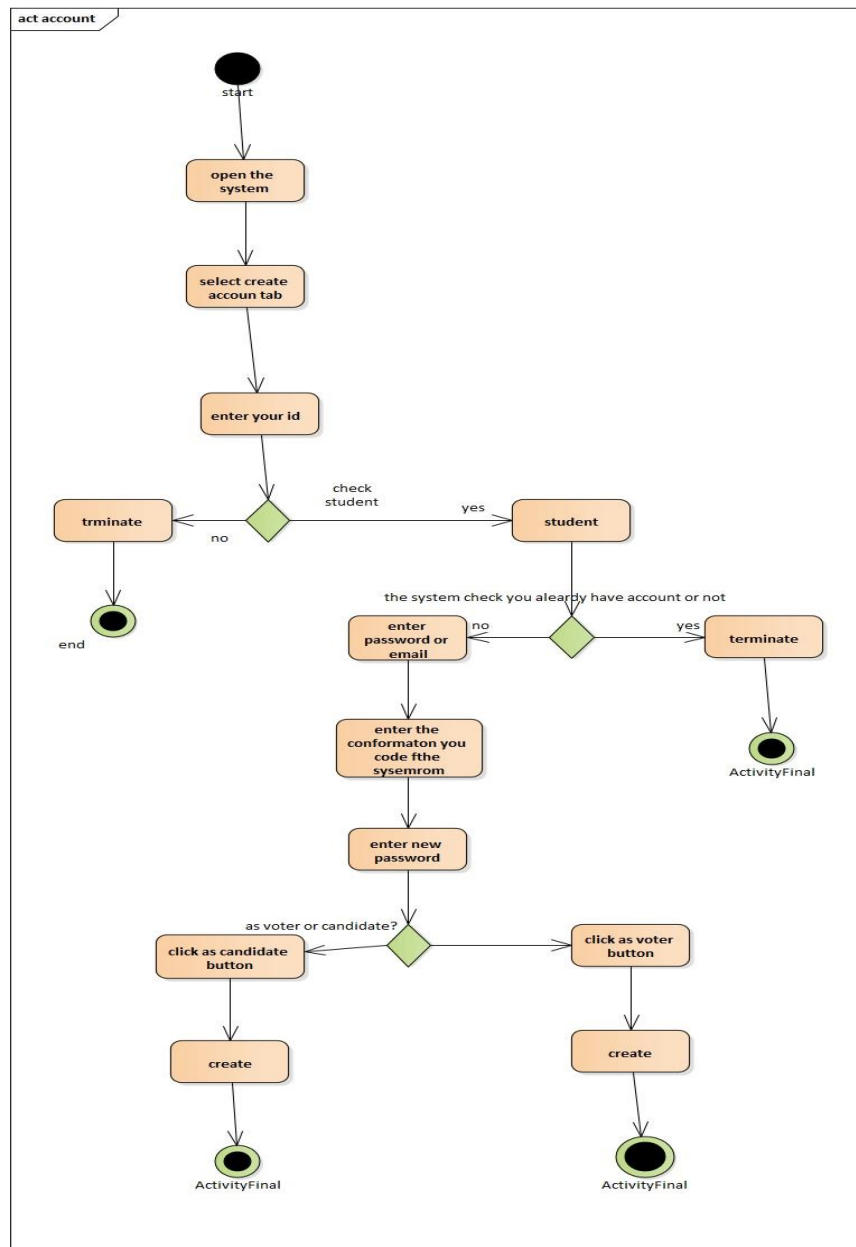
*Figure 18 Approve candidate activity diagram*



*Figure 19 Port data Activity diagram*

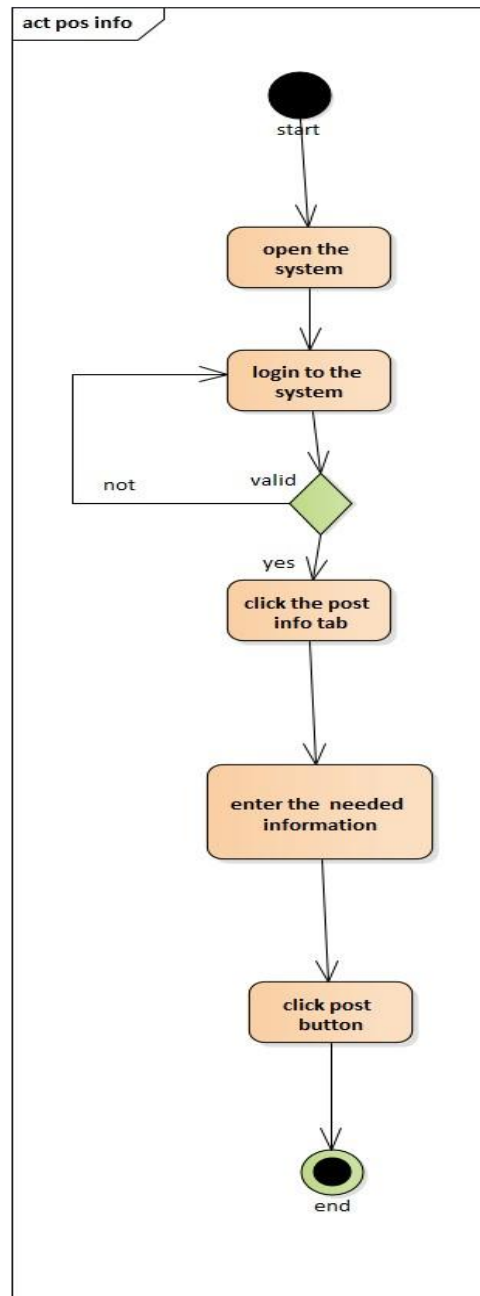


*Figure 20 Forget password Activity diagram*

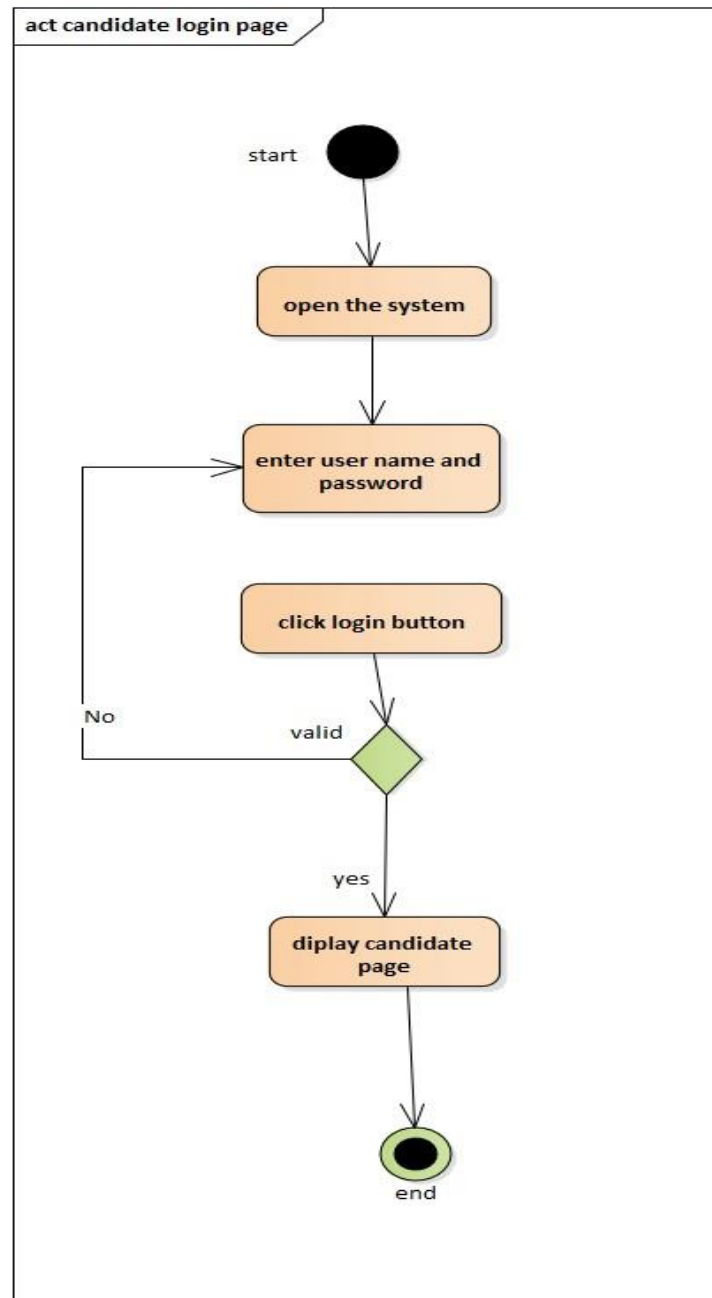


**Figure 21 Create account Activity diagram**





*Figure 22 Post info activity diagram*



*Figure 23 Candidate Login activity diagram*

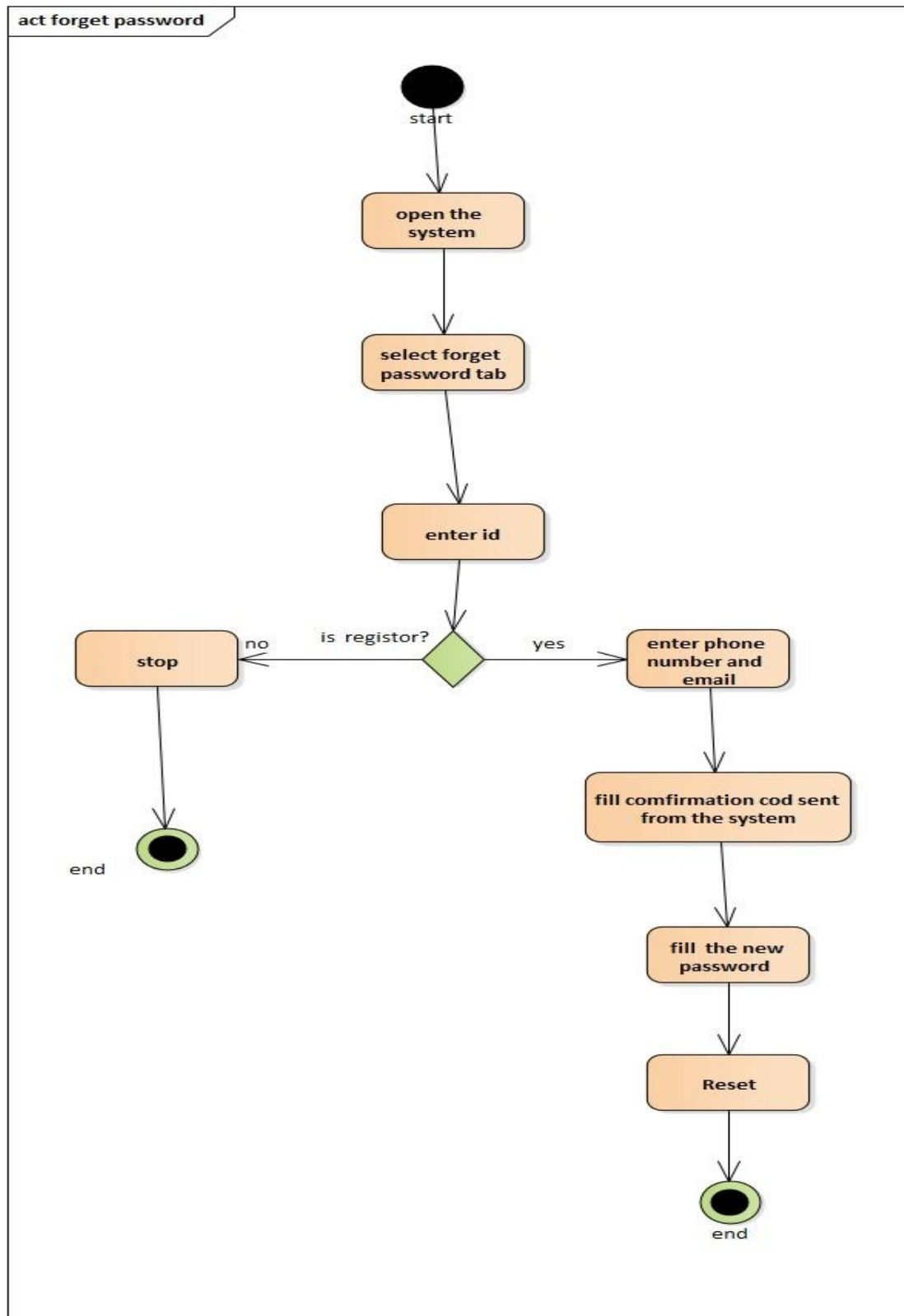
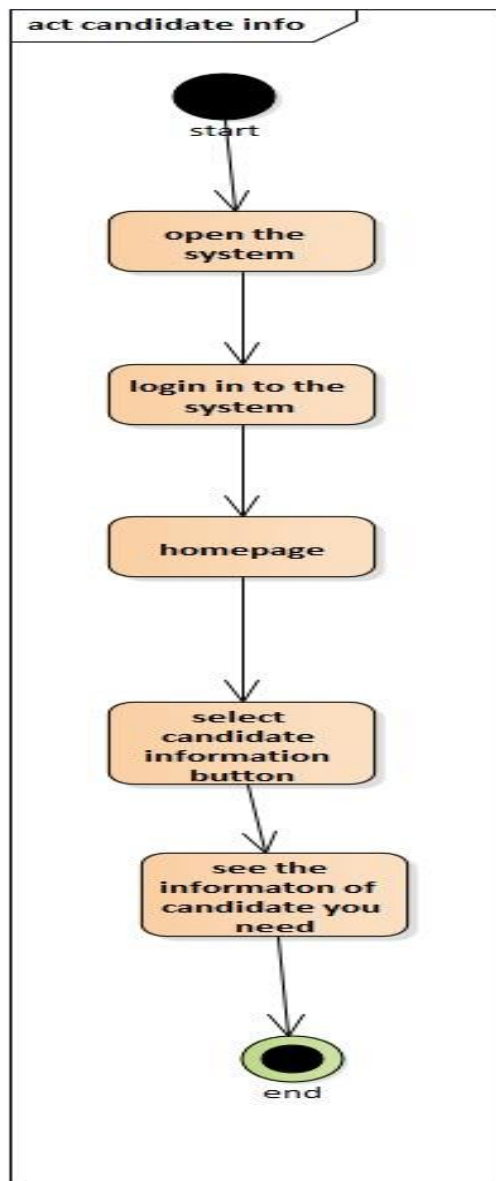
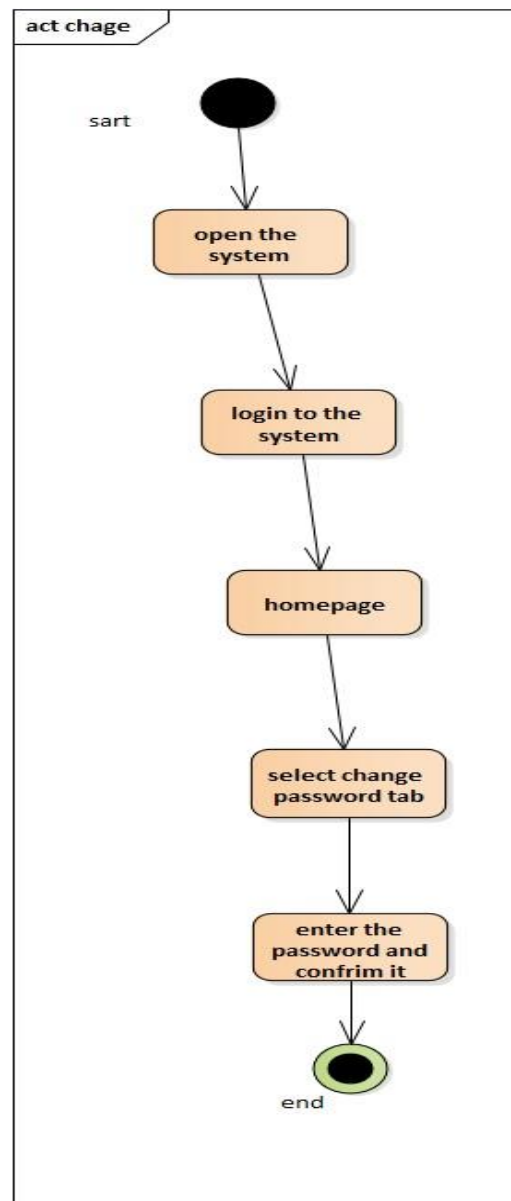


Figure 24 Forget password Activity diagram



*Figure 25 Candidate Info Activity diagram*



*Figure 26 change password Activity diagram*

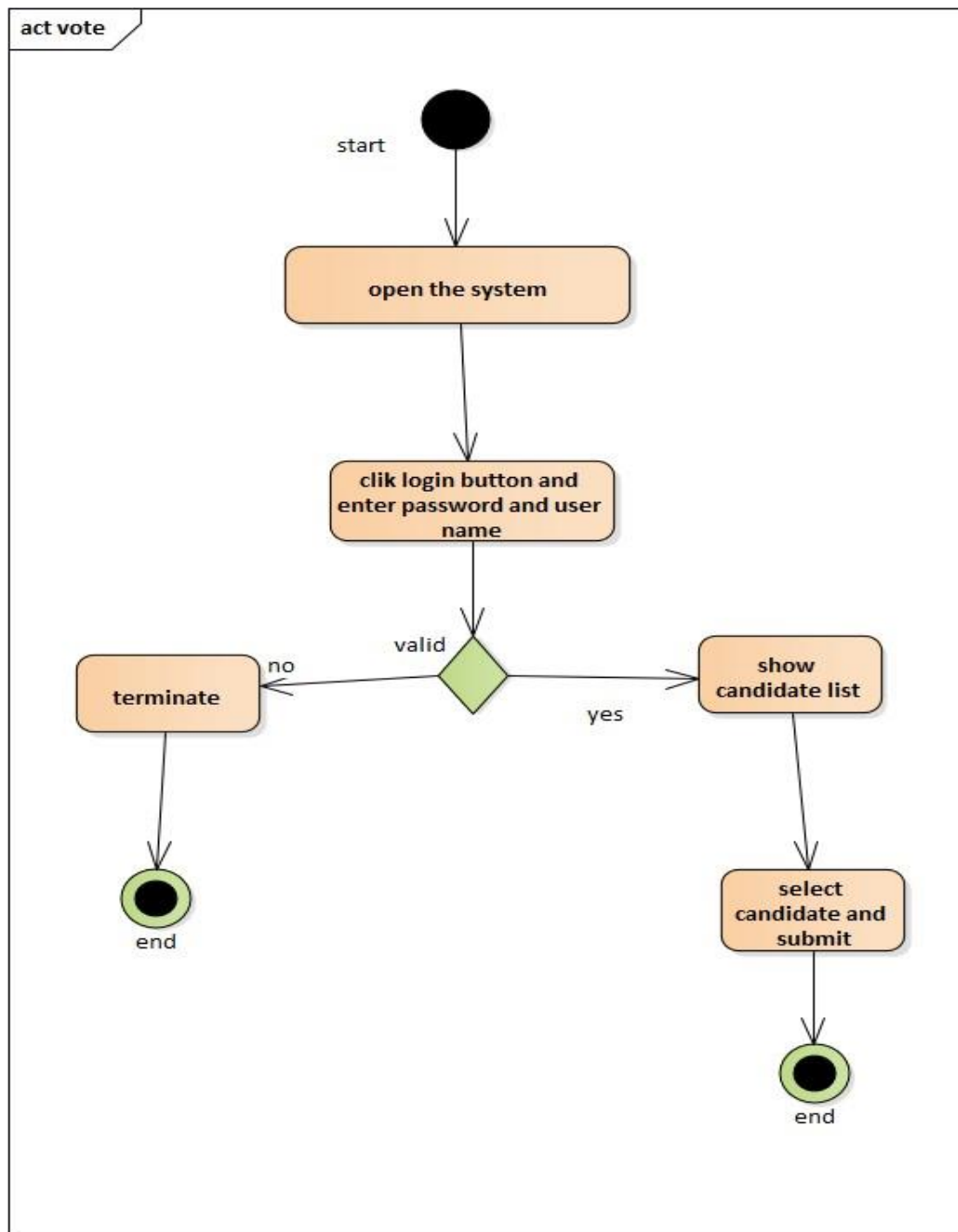


Figure 27 Vote Activity Diagram

# CHAPTER FOUR

## SOFTWARE DESIGN

### 4.1 INTRODUCTION

This document is about software design of online election system in ASTU. Online election system is a web based software that is used for casting vote.

#### 4.1.1 PURPOSE OF THE SYSTEM

The purpose of this project is creating a web based software that helps ASTU students to vote for their student union presidents and associated places like vice president, writer.

#### 4.1.2 DESIGN GOALS

Qualities that our system should focus on; which are mostly inferred from the nonfunctional requirements or from the application domain:-

##### **Performance**

OES must respond to request within a short period of time and must be capable to be accessed by many users at the same time.

##### **Reliability /dependability**

Online election system should be reliable. To ensure its reliability our system must be checked in use prior to the actual election.

##### **Security**

Security is critical requirement for OES. Online election system should be highly secured. i.e. It doesn't have to allow unauthorized access to the system. A vote casted by voter must be kept in a way that the voter identity is unknown (The privacy of the vote and voter has to be guaranteed). The system cannot be re-configured during operation. Additional ballots cannot be casted once the polling has closed (once the election session is over). A voter must not be able to vote more than once. The system is must be protected against accidental and malicious denial of service attacks.

##### **Maintainability**

The code for the system should be easily readable, understandable and should be easily mapped to specific requirements. In case of system failure, automatic system maintenance is must for our system. OES must undergo a basic testing procedure prior to an election event to take an assumption of where likely failure might happen. A team is sated up to do maintenance to OES.

**Cost:**

The system should be developed, deployed, administered and maintained with minimum cost possible.

**Usability:**

From the end users' perspective the system should be designed in such a way that it is easy to learn and use, efficient to do assigned tasks. The system should have simple and understandable graphical user interface such as forms and buttons which have descriptive names. It should give reliable response for each user request at least before the session expires.

### 4.1.3 DEFINITION, ACRONYM, AND ABBREVIATION

Acronym = OES (online election system)

Definition: A web based system that is used to cast vote.

### 4.1.4 REFERENCES

1. Prentice Object Oriented Software Engineering Using UML Patterns and Java 3rd 2012
2. Navathe, Elmasri. *Fundamentals of Database System 6th edition*.
3. [www.tutorialspoint.com](http://www.tutorialspoint.com) SQL Tutorial
4. [www.google.com](http://www.google.com)

### 4.1.5 OVERVIEW

This software design document will detail the design of online election system for ASTU to elect student union president and other associated places. It focuses on transforming the analysis model into the design model that takes into account the nonfunctional requirements and constraints described in the problem statement and requirement analysis sections discussed in earlier chapters.

This Software Design Document for OES is divided into four sections with various subsections. This sections of the OES Software Design Document are:

- 4.1 Introduction
- 4.2 Current software architecture
- 4.3 Proposed software architecture
- 4.4 Subsystem services

## 4.2 CURRENT SOFTWARE ARCHITECTURE

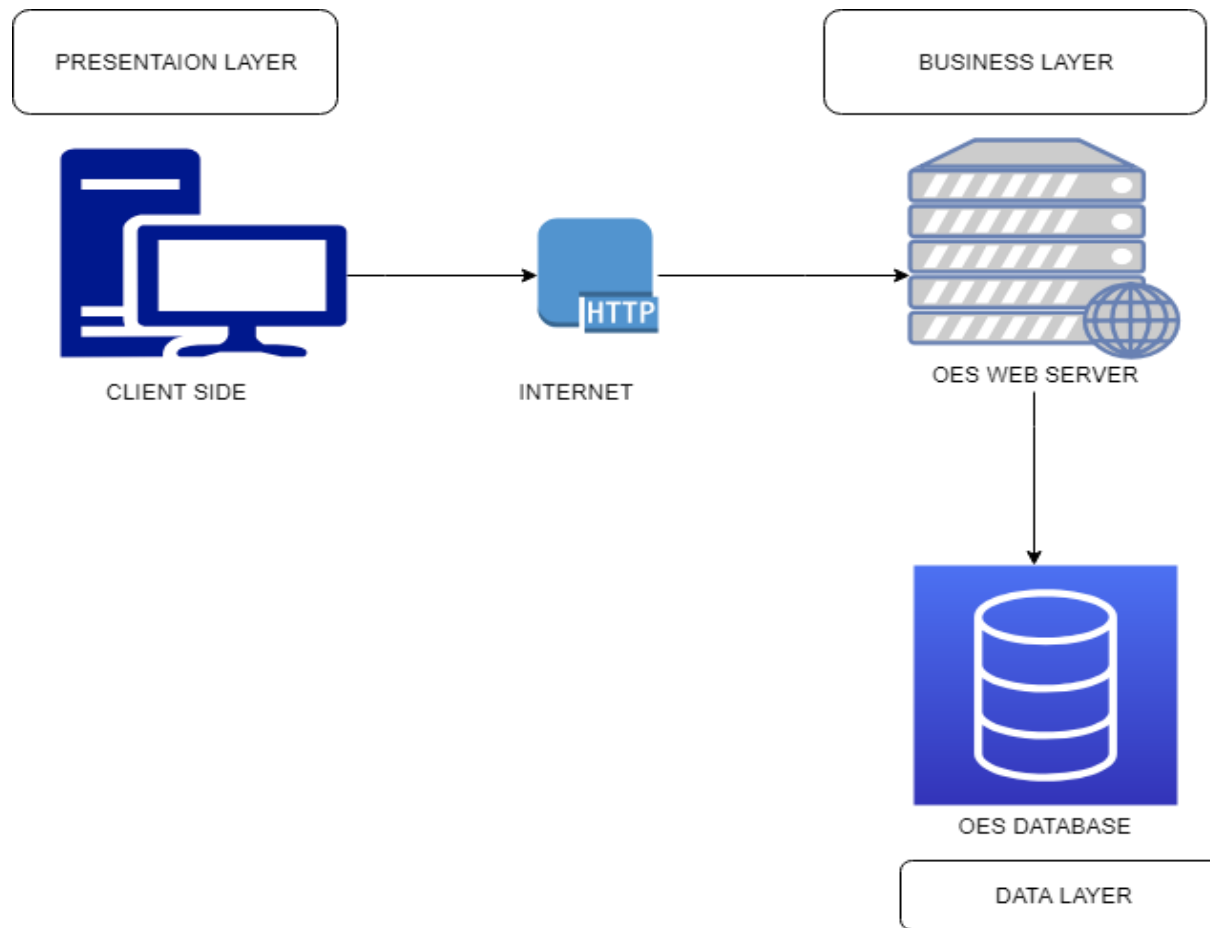
Our system is a three tires architectural it has client side, server and database.

**Client side:** here in the client side Employee, Administrator, and user interface will be existing.

**Server side:** here the web servers to connect the data base application are found; mean that the application logic to perform the application by the client is found.



**Data base:** here the data bases that store the information are found.



*Figure 28 Current OES architecture*

### 4.3 PROPOSED SOFTWARE ARCHITECTURE

In this project, the team uses a three-tier architecture, which has three layers. These three layers are the Application or Presentation layer, the business layer and the data access layer.

Application or presentation layer is the form, which provides the user interface to either programmer or end user. The business layer is the class, which the team uses to write the function, which works as a mediator to transfer data from application layer or presentation layer to data layer. This layer also has a property layer which is a class where variables are declared corresponding to the fields of the database which can be required for the application and make the properties so that the team can get or set the data using these properties into the variables.

The third tire is the data access layer which is also a class to get or set data to the database queries back and forth. This layer only interacts with the database. The database queries or stored procedures will be written here to access the data from the database or to perform any operation to the database.

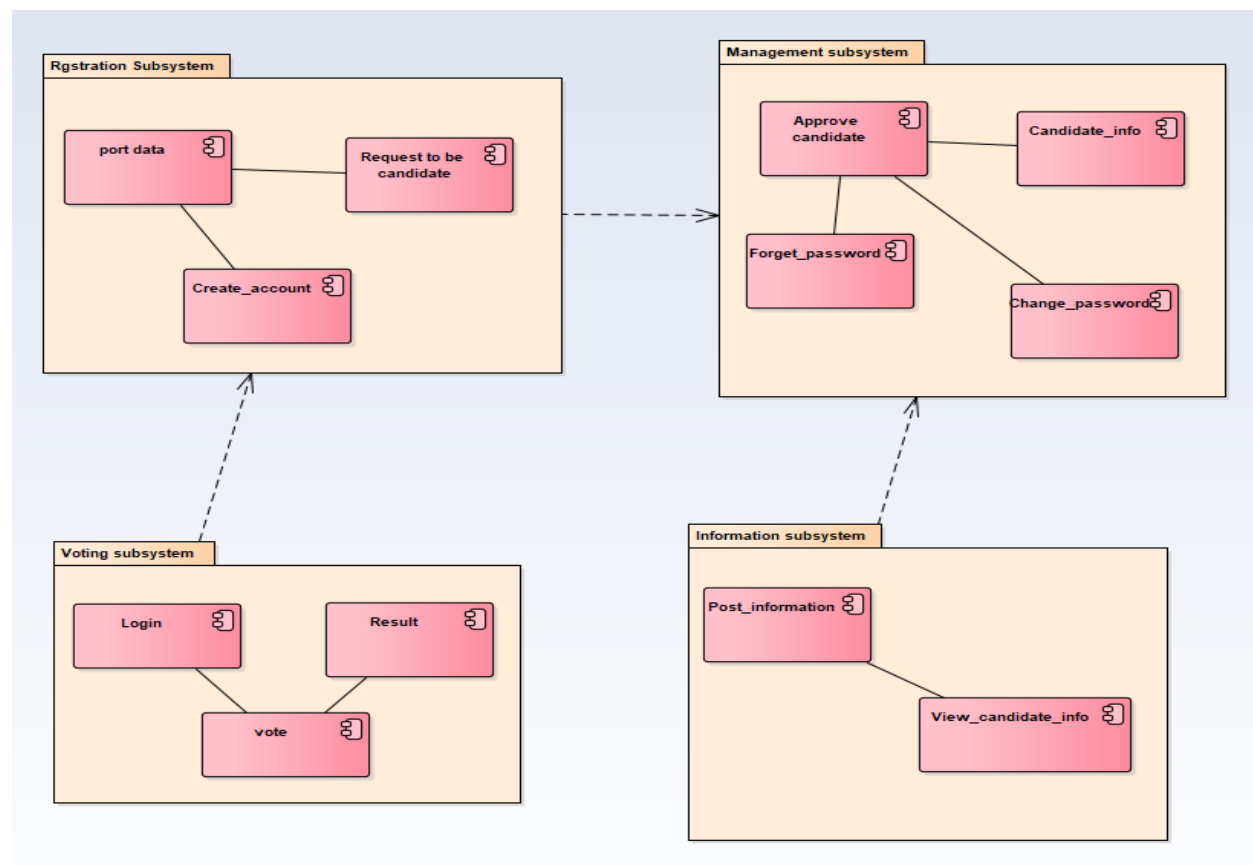
### 4.3.1 OVERVIEW

This is the system design document of OES for ASTU student to elect their representative.

System design part of the system is the transformation of the analysis model into a system design model. Up to now, we were in the problem domain. System design is the first part to get into the solution domain in a software development. This chapter focuses on transforming the analysis model into the design model that takes into account the non-functional requirements and constraints described in the problem statement and requirement analysis sections discussed earlier. This document describes the design issues of the overall system, such as design goal, subsystem decomposition, hardware/software mapping, and persistent data management. It provides the complete architectural overview of the proposed system. It is intended to capture and express the significant architectural decisions, which have been made, on the system.

### 4.3.2 SUBSYSTEM DECOMPOSITION

To reduce the complexity of the solution domain, we decompose a system into simpler parts, called subsystems. The main need of this portion is to design the external part of the system. In this project, there are four subsystem decompositions.

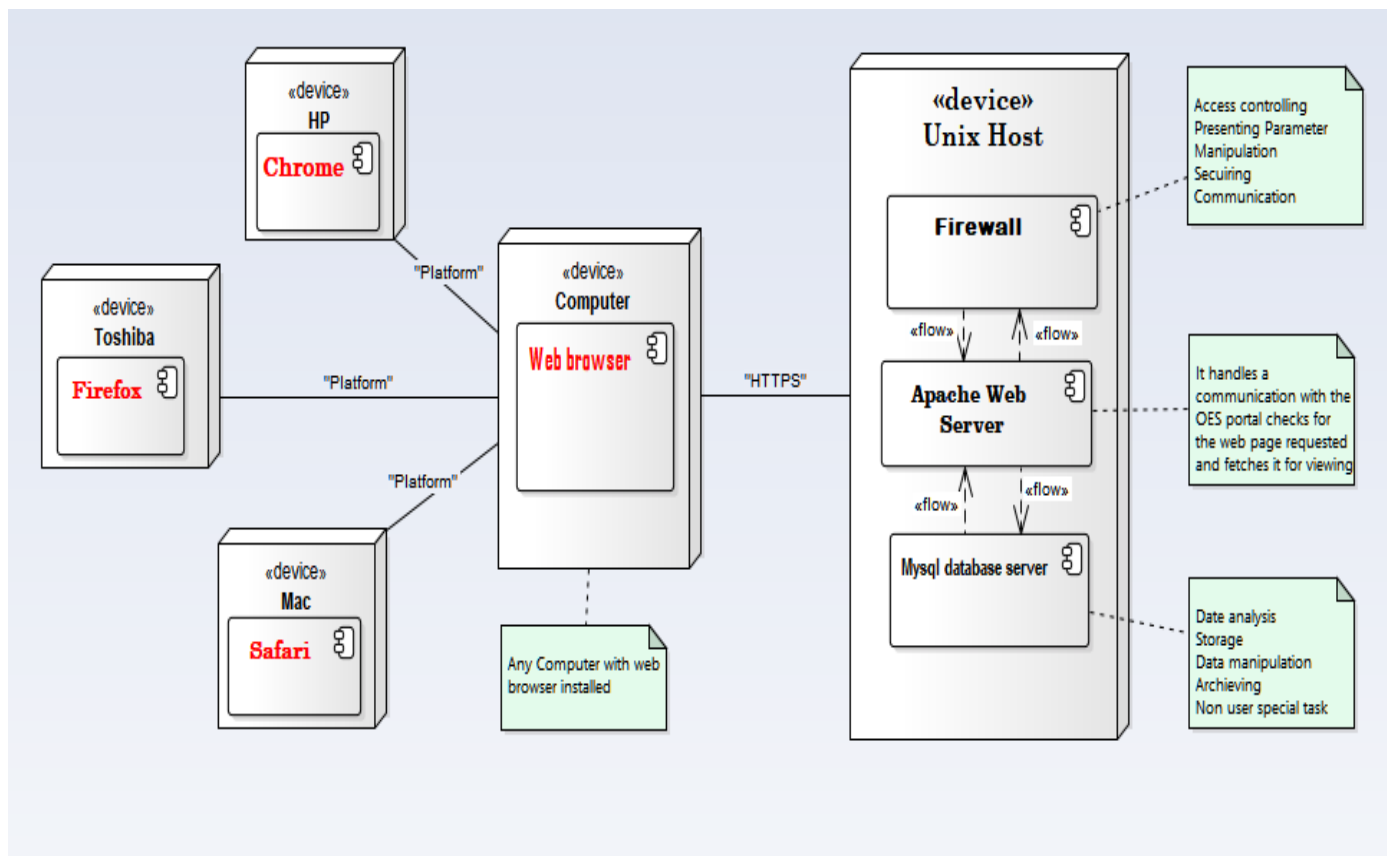


*Figure 29 Subsystem decomposition*

### 4.3.3 HARDWARE OR SOFTWARE MAPPING

The system will have two processes deployed in single or separate machine that run in parallel namely web server process and the database process. The database processors which runs on MYSQL/SQLserver database engine, is responsible for maintaining data manipulation operation.

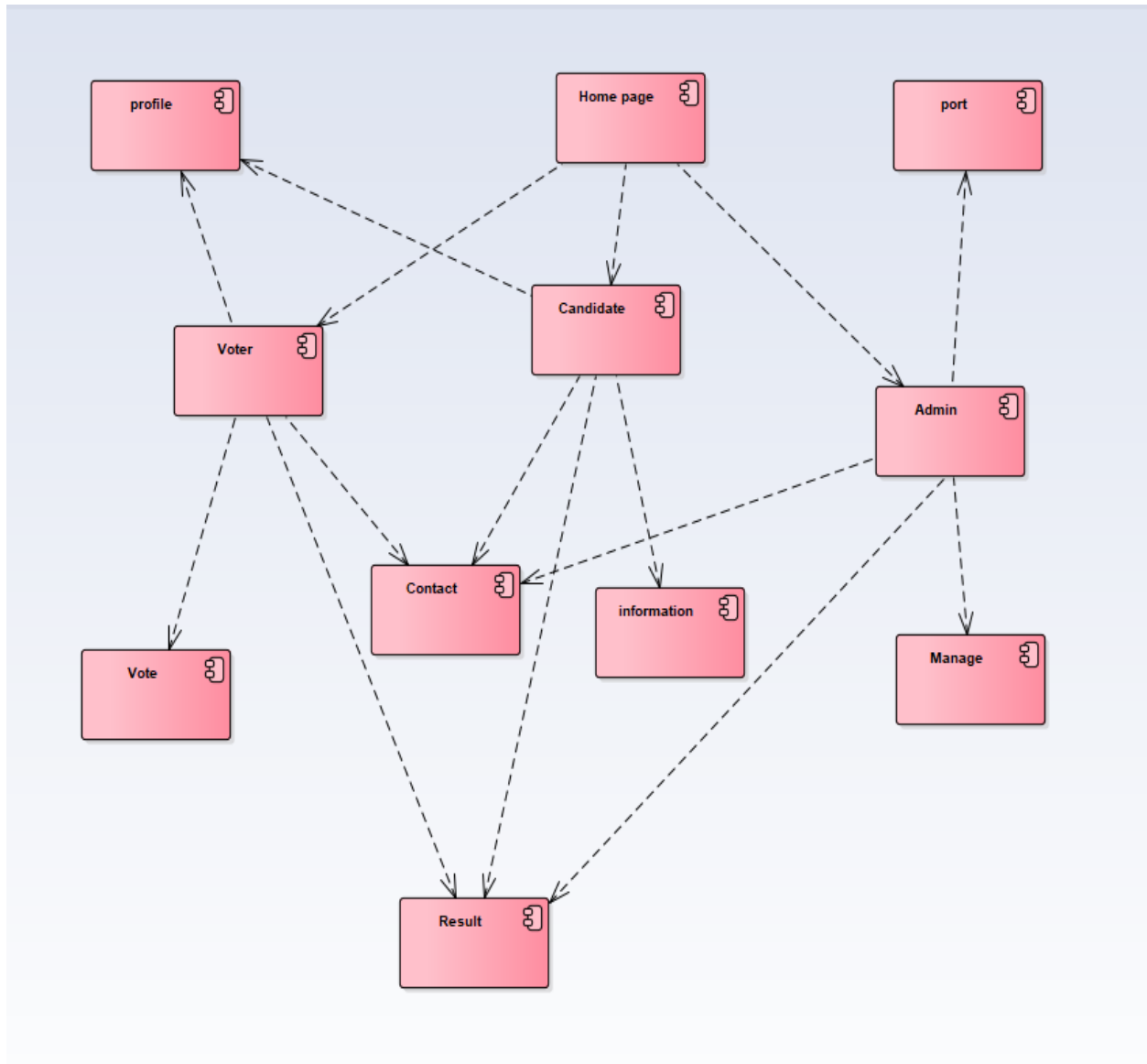
Where us the web server process is responsible to host the web pages of the system and process clients' request. In case of the client side, only a browser is required to access the objects.



*Figure 30 Hardware or software mapping*

### 4.3.4 COMPONENT DIAGRAM

It Describes all components in a system, their interrelationships, interactions, and the interface of the system. It is an outline of the composition structure of components or modules



*Figure 31 Component diagram*

### 4.3.5 DATABASE DESIGN

Relationship mapping of each table in a relational database:

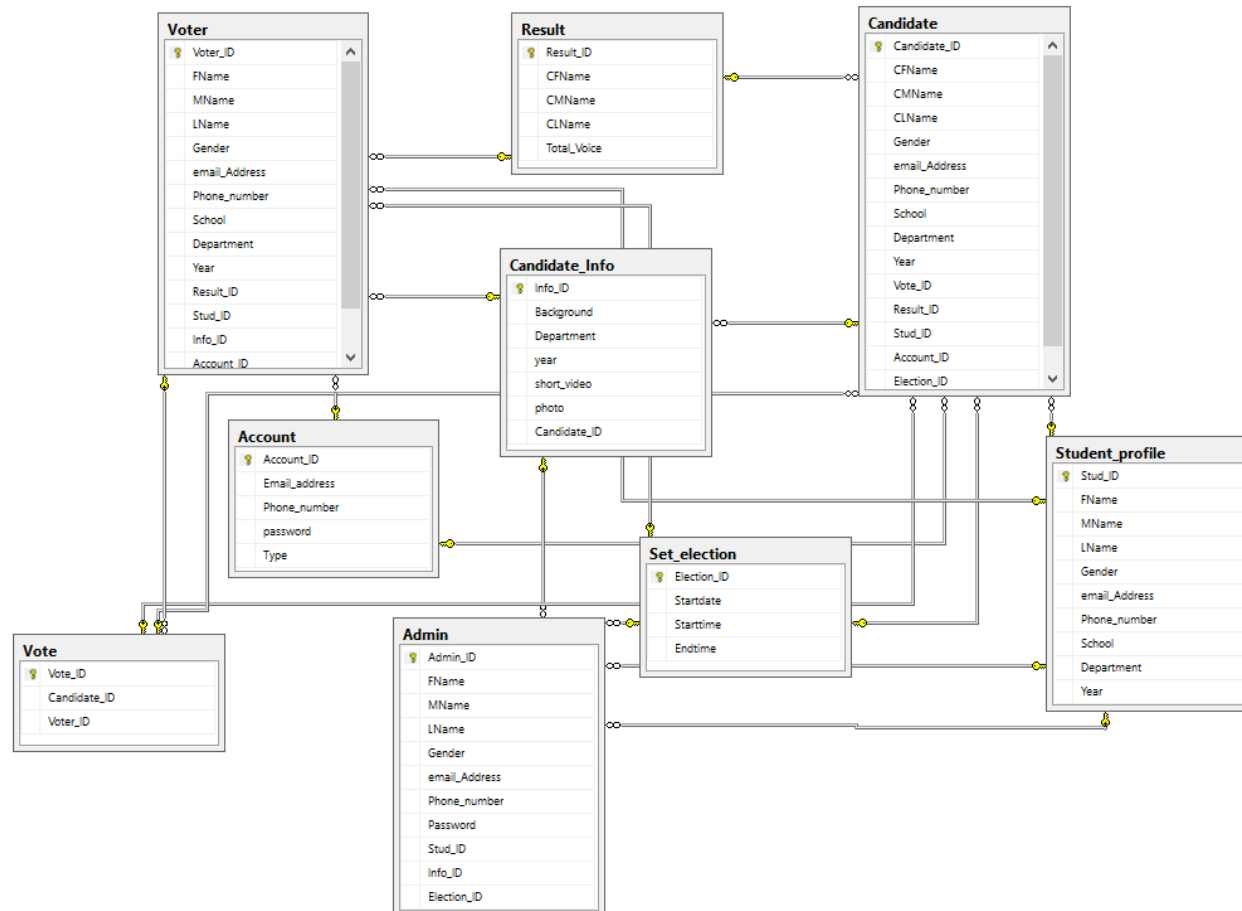
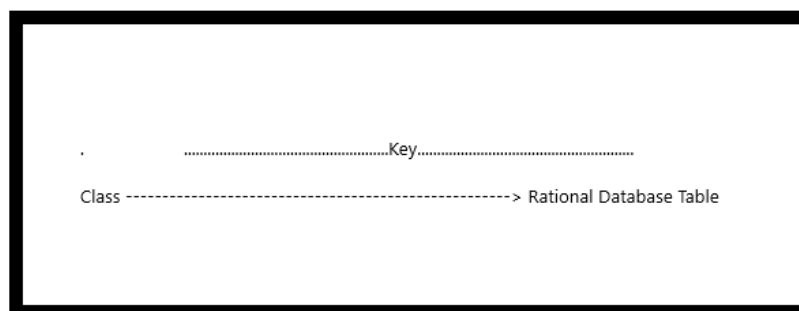


Figure 32 Database design

#### 4.3.6 PERSISTENT DATA MANAGEMENT

The purpose of this section is to show the mapping of the objects/classes of the system, identified during the analysis stage, in to the corresponding relational database.

Diagram bellows are mapped in (form):



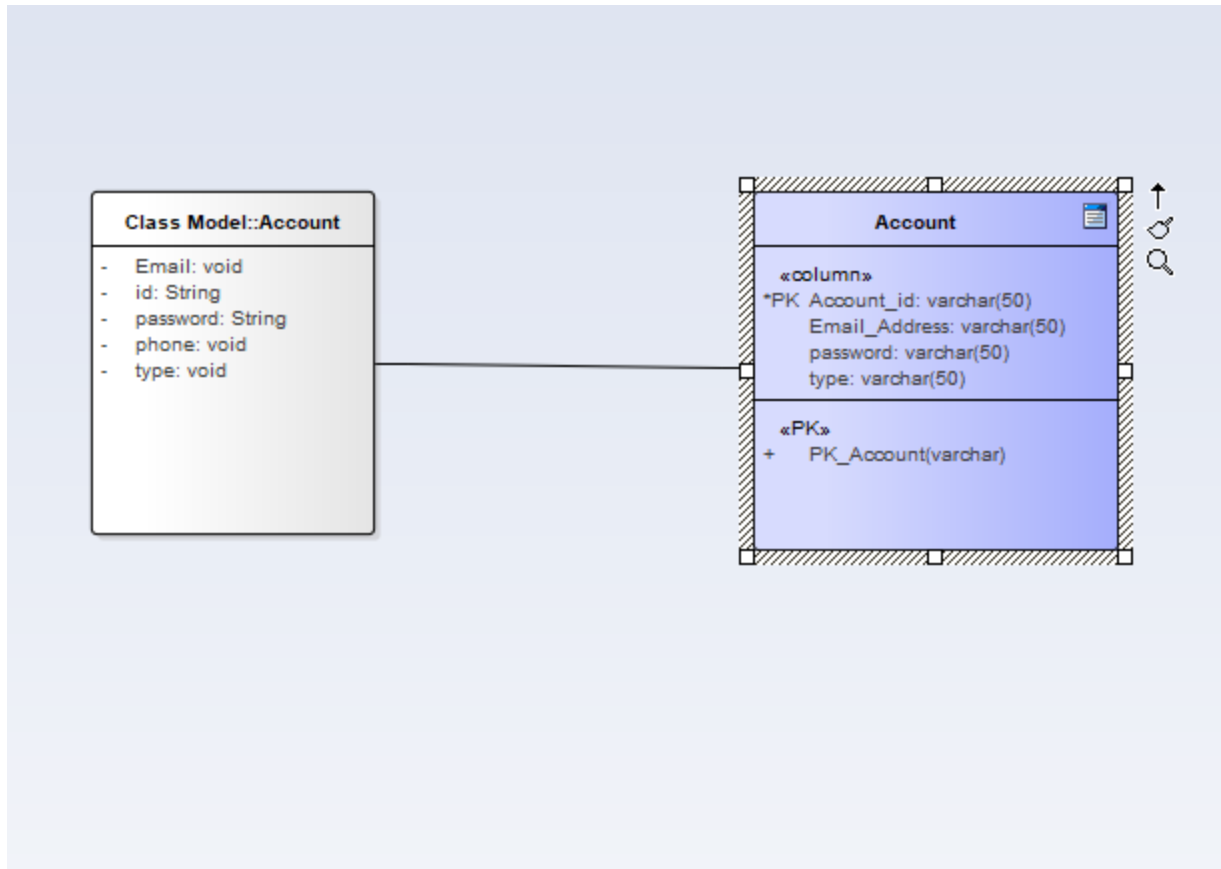


Figure 33 Object Account mapping

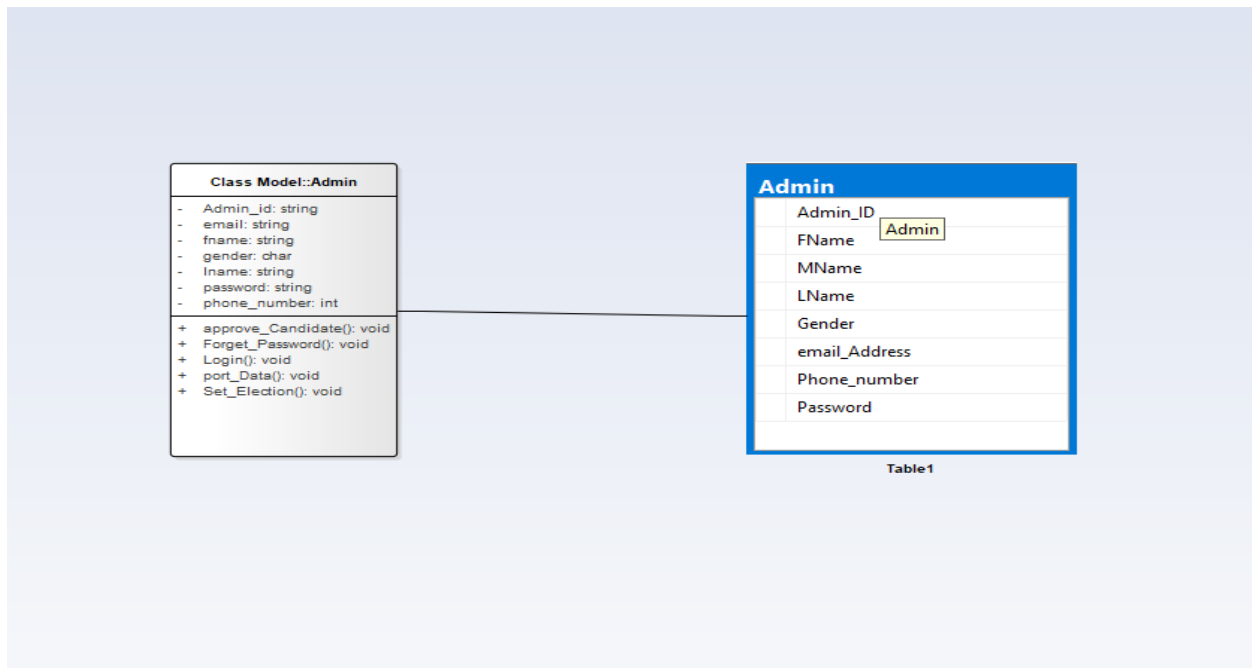


Figure 34 Objet admin mapping

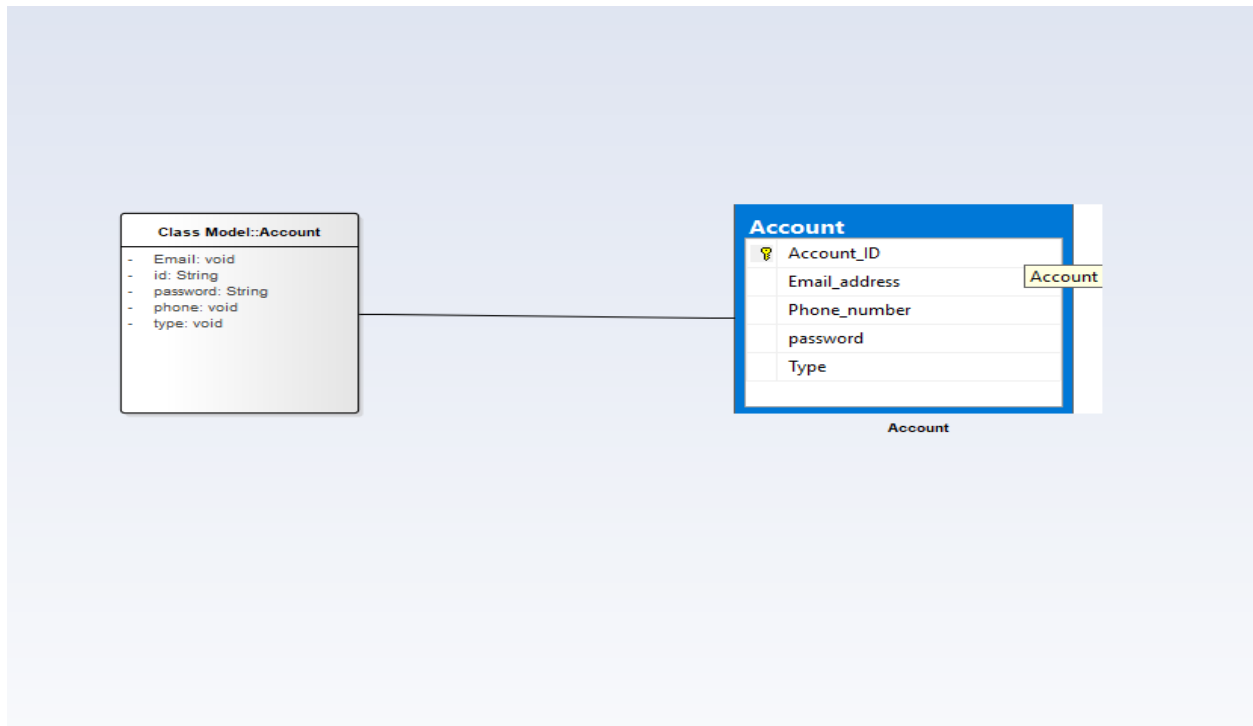


Figure 35 Object Account mapping

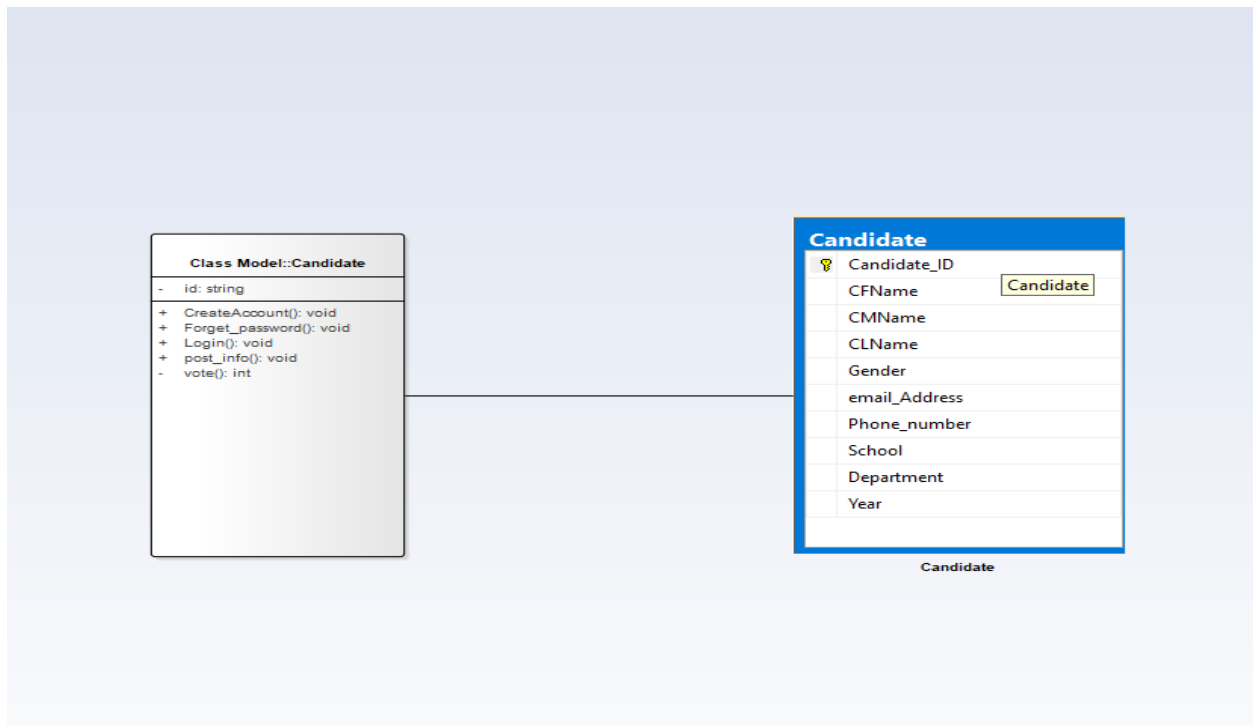


Figure 36 Object Candidate mapping

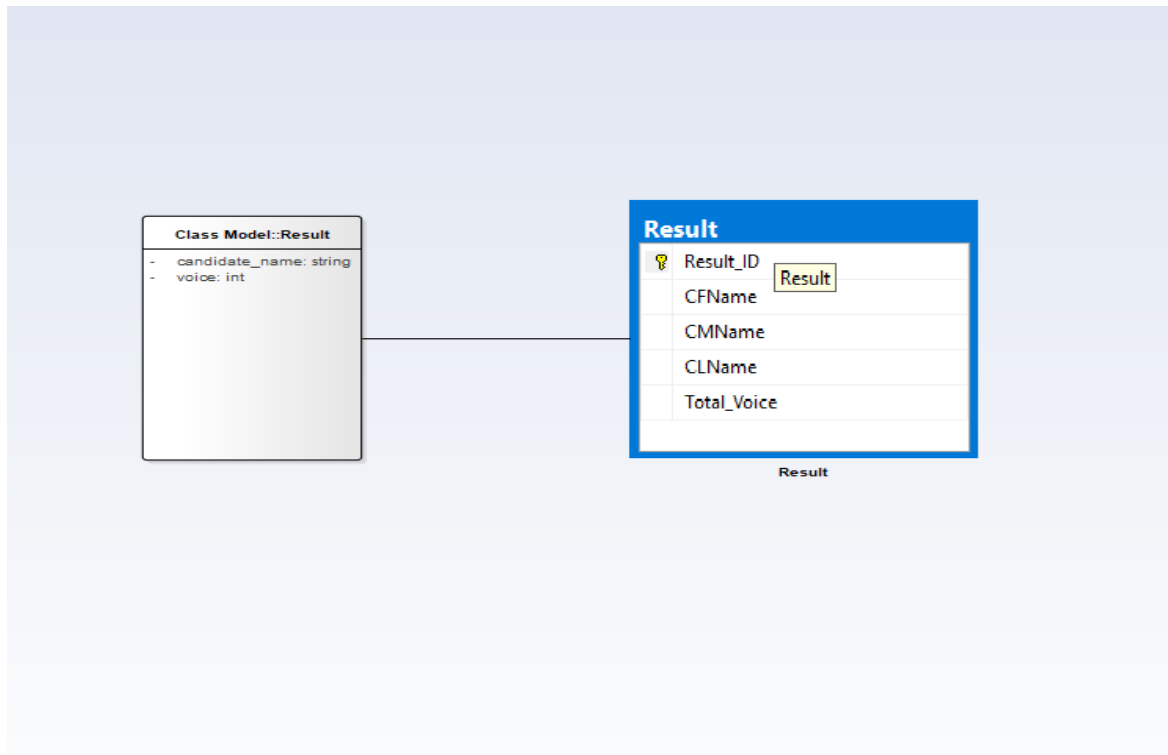


Figure 37 Object Result mapping

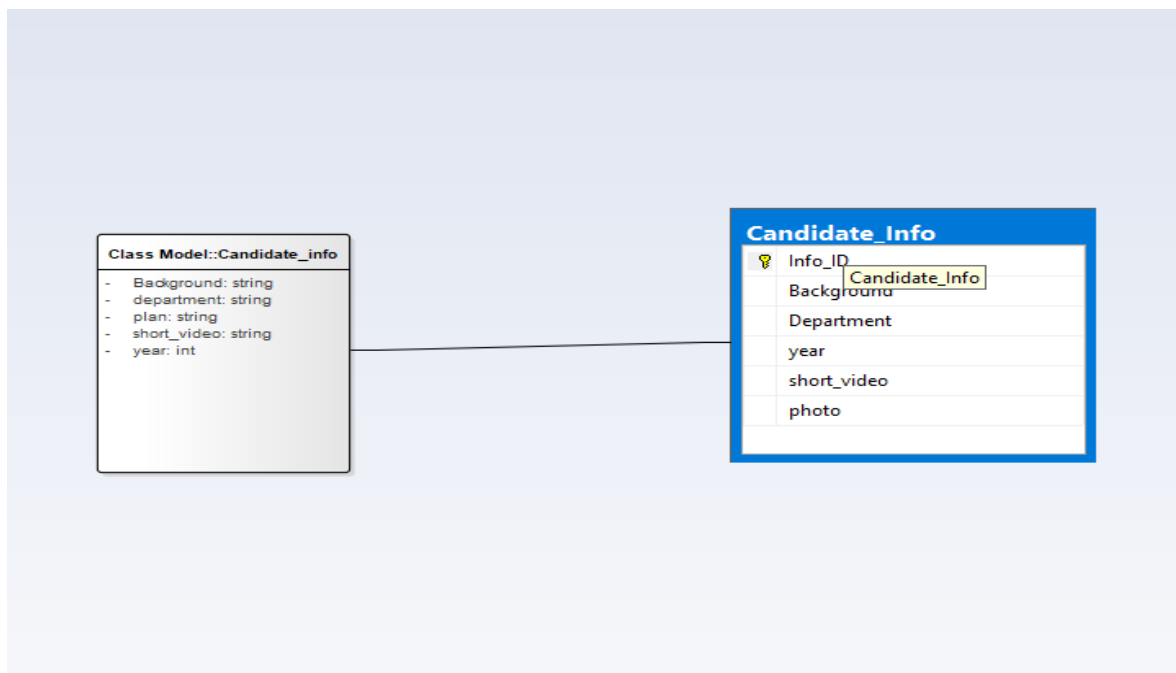


Figure 38 Object Candidate\_info mapping



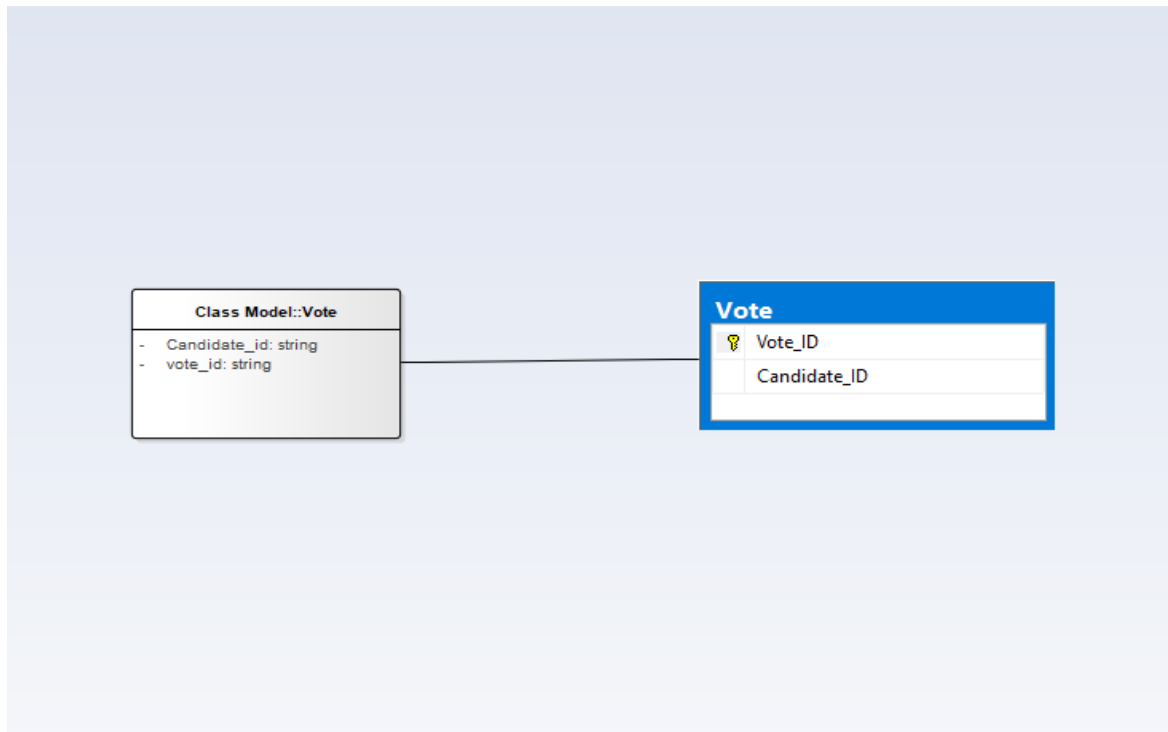


Figure 39 Object Vote mapping

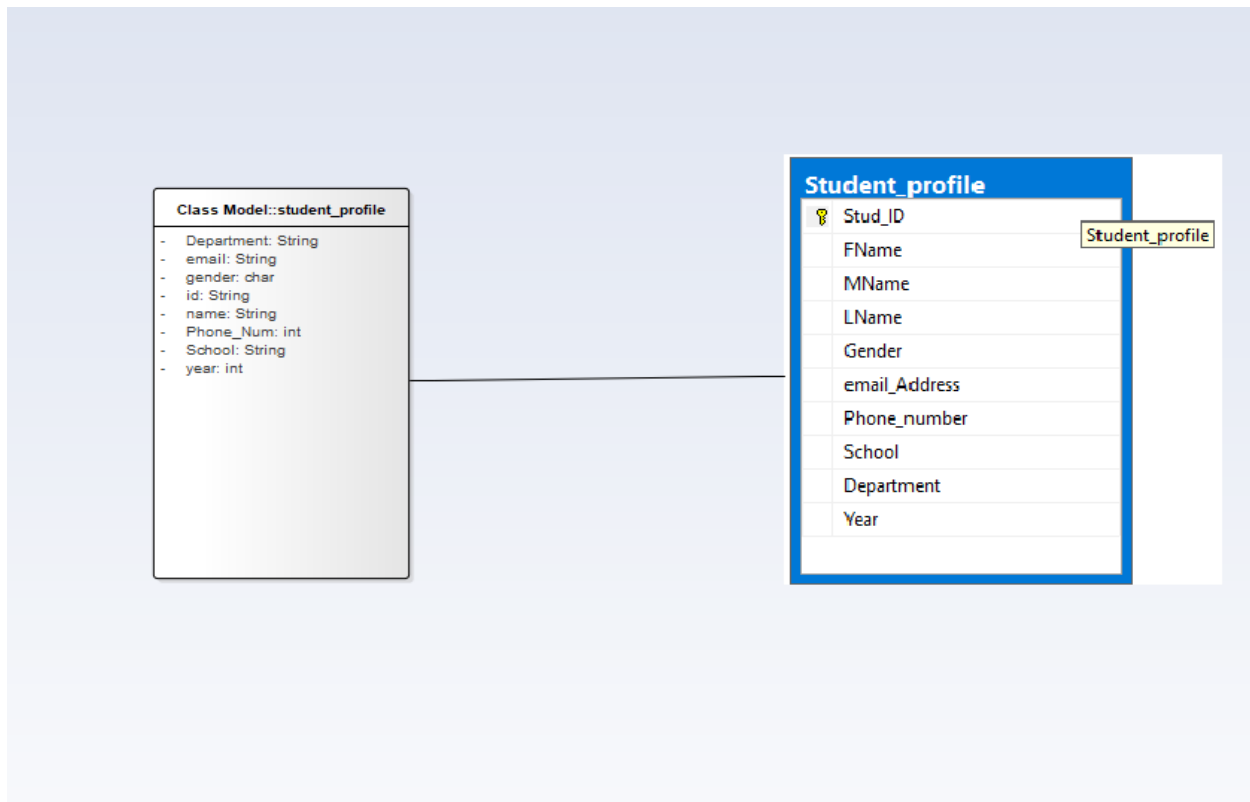
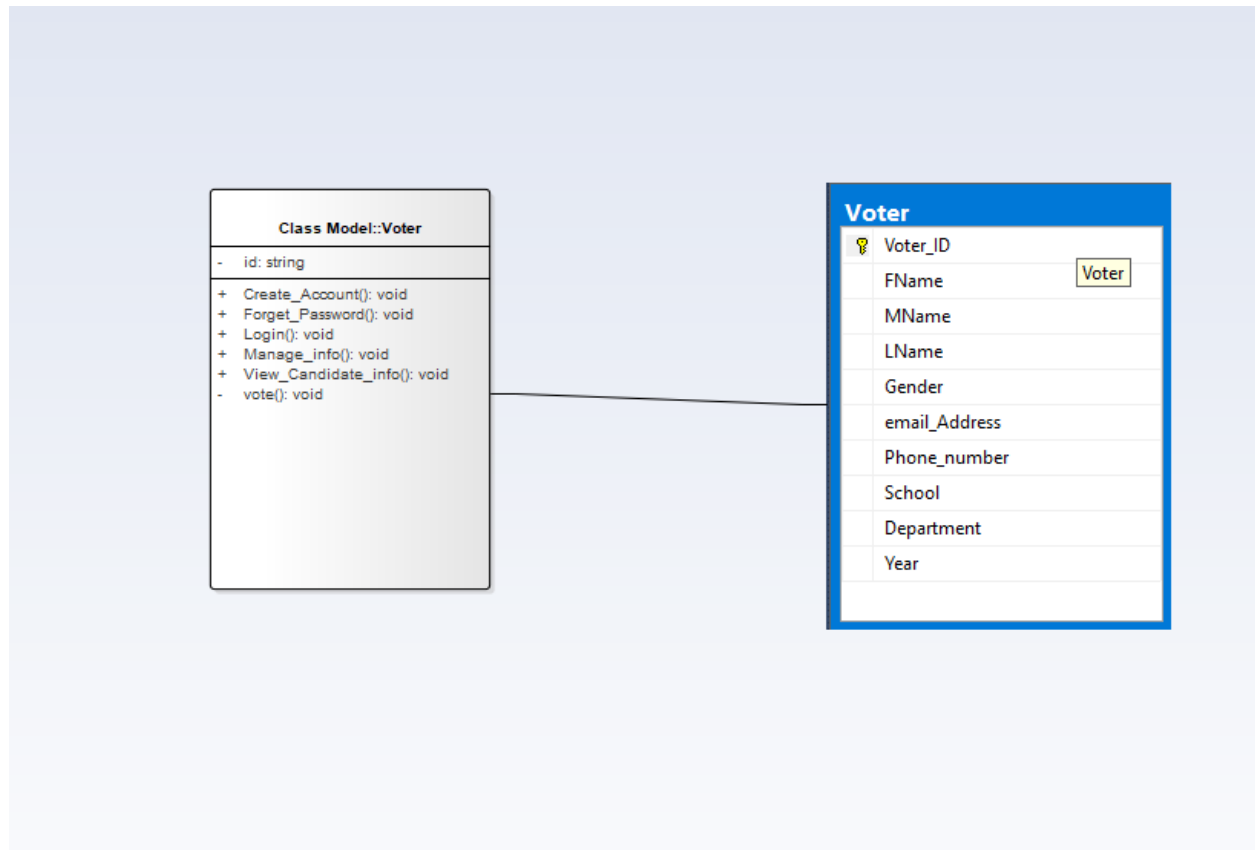


Figure 40 Object student\_profile mapping



**Figure 41** object Voter mapping

#### 4.3.5 ACCESS CONTROL AND SECURITY

The information which registered in the system we bulled have to be secure. In our system, different actors have access to different functionality and data. Therefore these privileges put off unauthorized users from accessing data's which they don't have privilege to access

Objects Actors	Login	Create account	View candidates info
Admin	Get login() Verify() Validate() Set log()		
Voter	Get login() Verify() Validate() Set log()	Get ID() Verify() Validate() create()	Get detail info()
Candidates	Get login() Verify()	Get ID() Verify()	

	Validate() Set log()	Validate() create()	
--	-------------------------	------------------------	--

**Table 6 Access matrix for class's login, create account and view candidate's information**

Objects Actors	reset password	Post information	Set election
Admin	Send password reset link() Validate email() Verify security question()		Activate the system and deactivate when time is over()
Voter	Send password reset link() Validate email() Verify security question()		
Candidates	Send password reset link() Validate email() Verify security question()	Add information() Get detail info()	

**Table 7 Access matrix for classes, reset password, post information, set election**

Objects Actors	Port data	Approve candidates	Forget password
Admin	Add data from registration system to the System	Approve The request of to be candidates()	Add email or phone no() Send reset link() Set new password() Sent confirmation code()

Voter			Add email or phone no() Send reset link() Set new password() Sent confirmation code()
Candidates			Add email or phone no() Send reset link() Set new password() Sent confirmation code()
Objects Actors	Port data	Approve candidates	Forget password
Admin	Add data from registration system to the System	Approve The request of to be candidates()	Add email or phone no() Send reset link() Set new password() Sent confirmation code()
Voter			Add email or phone no() Send reset link() Set new password() Sent confirmation code()
Candidates			Add email or phone no() Send reset link() Set new password() Sent confirmation code()

*Table 8 Access matrix for classes, port data, Approve candidates, forget password*

#### 4.3.6 BOUNDARY CONDITIONS

- **Start up the system:** Browsing for the URL.
- **Termination of the system:** closing the browser leads to session destroy.
- **Error condition:** connection down between data transfer.

Most of the system design effort is concerned with steady-state behavior. However, the system design must be address the initiation and termination of the system. This addressed by administration use case.

**a) Dynamic model of the system startup**

Online election system is a system that run properly when the user needs to use the system. Therefore, user of the system must start the system from their android smartphone when they to use the system. After the system is started up, it is necessary to connect the internet.

For the Administrator to be able to access the system it is also necessary to connect the internet through web browser, which must be installed on the machine to start the connect to the server.

**b) Termination**

Describes what resources are cleaned up and which systems are notified upon termination. If web server or web server is not started to run then web interface or web interface will not be visible by the users. After finishing the system interaction user can be logging out and closes the web.

**c) Error Condition**

Many possible causes: internet connection fails, Bugs, errors, external problems (power supply) and also web application faller may happen.

Error	Cause	solution
Server not found	No internet connection	Network connection
Communication link fails	Wrong IP address. Port number is missing or wrong. Net blocking connection. DB server is down etc.	Verify and test them with ping. Verify it based on my.cnf of SQL DB.
Web server fail	Computational/logic Errors Power outages Overheating High humidity Natural disasters, e.g., hurricanes, Floods	Using Server failover. To send a backup automatically on Backup server. That means An automatic server failover solution can prevent your website from going down in the event of a server failure.
Web Application fail	web application attacks (treat) cross-site scripting(XSS) and SQL	Firewalls and similar intrusion detection mechanisms provide little

	injection Device driver failures I/O errors, e.g., hard disk failures(see database media failures) Memory parity errors Network hardware failures	defense against full-scale web attacks. Using Server failover.
--	---	---

*Table 9 Boundary condition failures*

## 4.4 SUBSYSTEM SERVICES

	Subsystem name	Subsystem service
1	Registration System	<ul style="list-style-type: none"> <li>These subsystem provide the list of students those belongs to the University to the OES.</li> <li>Student can only register if there name found in the list given by these subsystem.</li> </ul>
2	Management subsystem	<ul style="list-style-type: none"> <li>Management subsystem helps Admin to prove or disprove the pending request sent by students depending on the predetermined rule and regulation of the university.</li> <li>Admin can also manage posted candidate information using these subsystem.</li> <li>He can also forget and change his own password.</li> </ul>
3	Voting subsystem	<ul style="list-style-type: none"> <li>Allows already registered student to vote there representative by login into these subsystem.</li> <li>These subsystem capable of displaying the result of election through the election processes.</li> </ul>
4	Information subsystem	<ul style="list-style-type: none"> <li>Candidate can post there information by the help of information subsystem</li> <li>The information posted by the candidate are visible to the voter.</li> </ul>

*Table 10 Subsystem services*

