# 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 ABREVATION

Acronym = OES (online election system)

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

### 4.1.4 REFERENCIES

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 SQL Tutorial

4. www.google.com

### 4.1.5 OVERRVIEW

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.

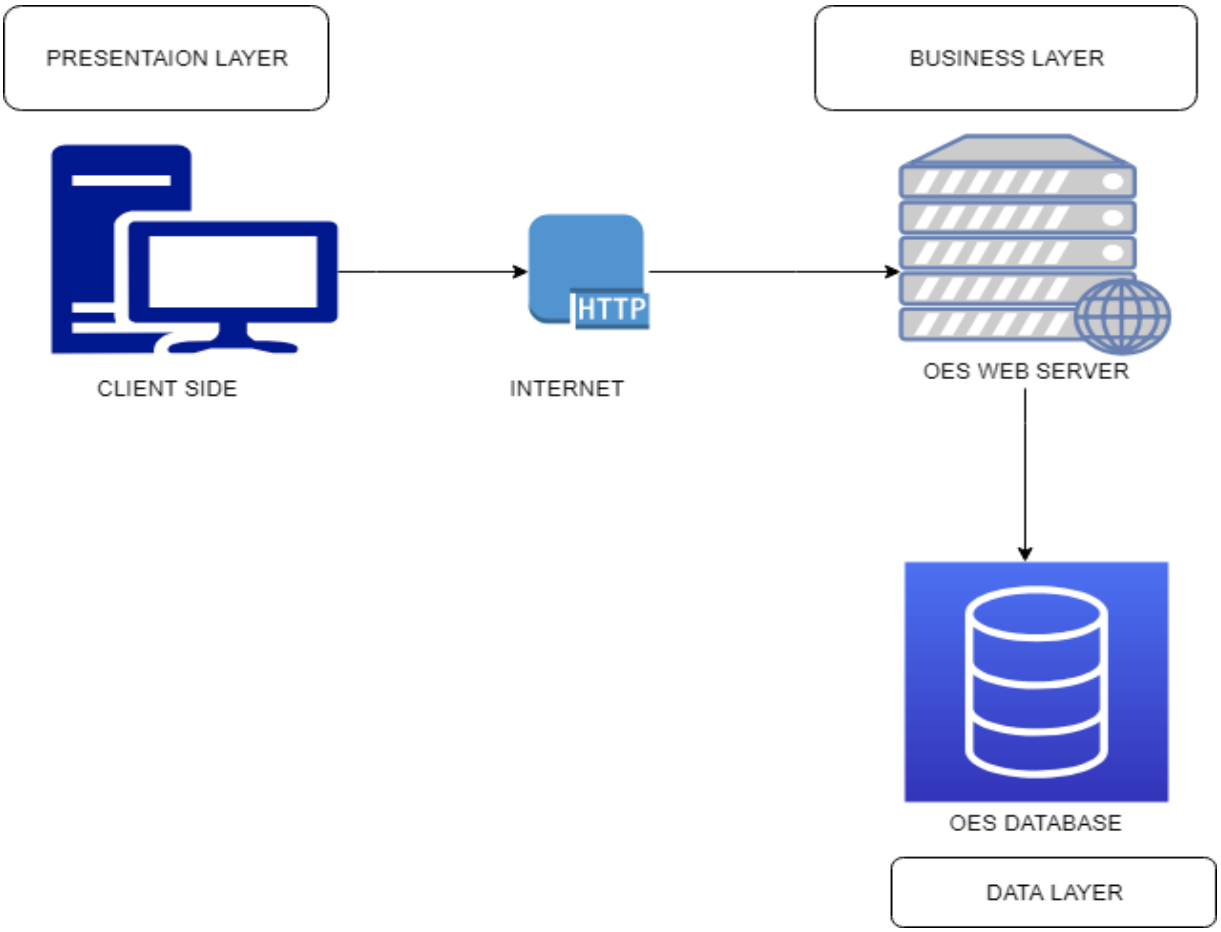



*Figure 1 Current OES architecture*

# 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 DECOMPOSTION

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.

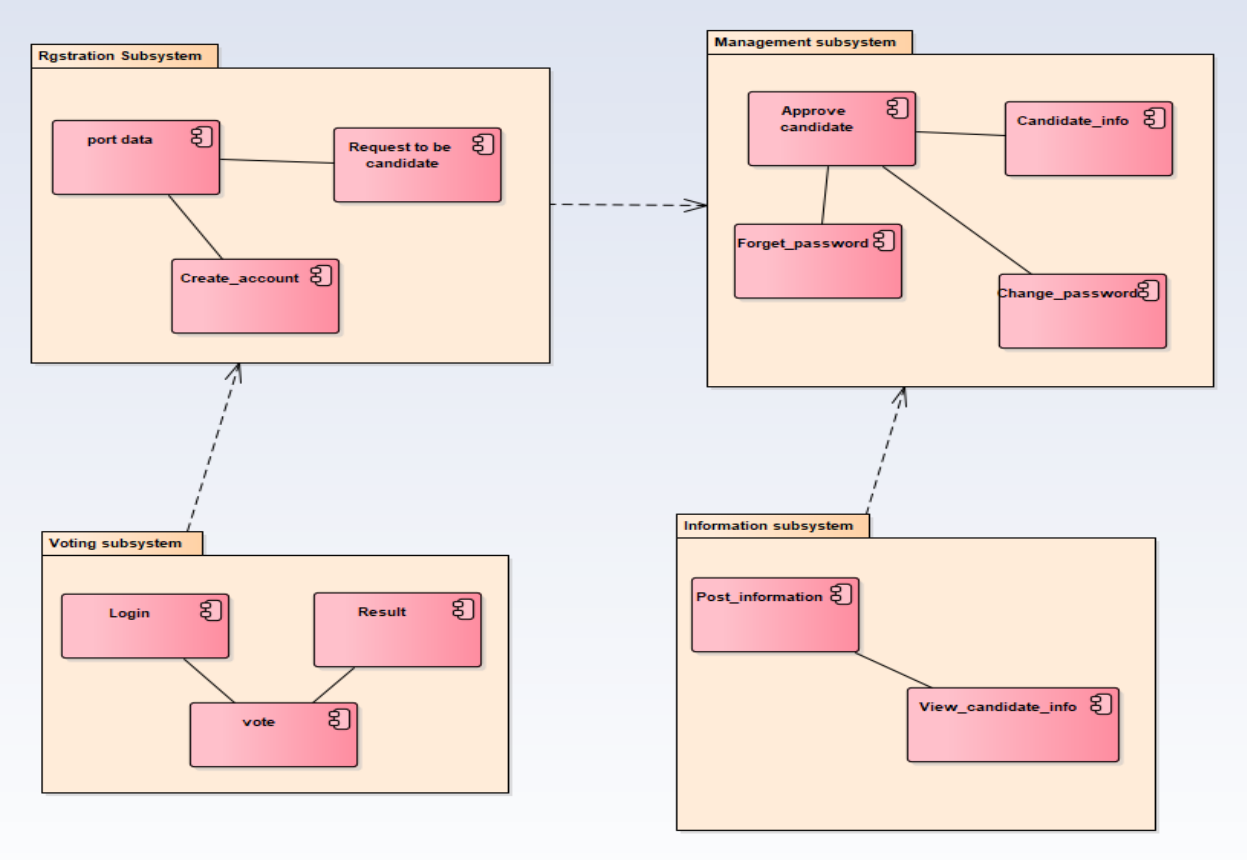


*Figure 2 Subsystem decomposition*

### 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.

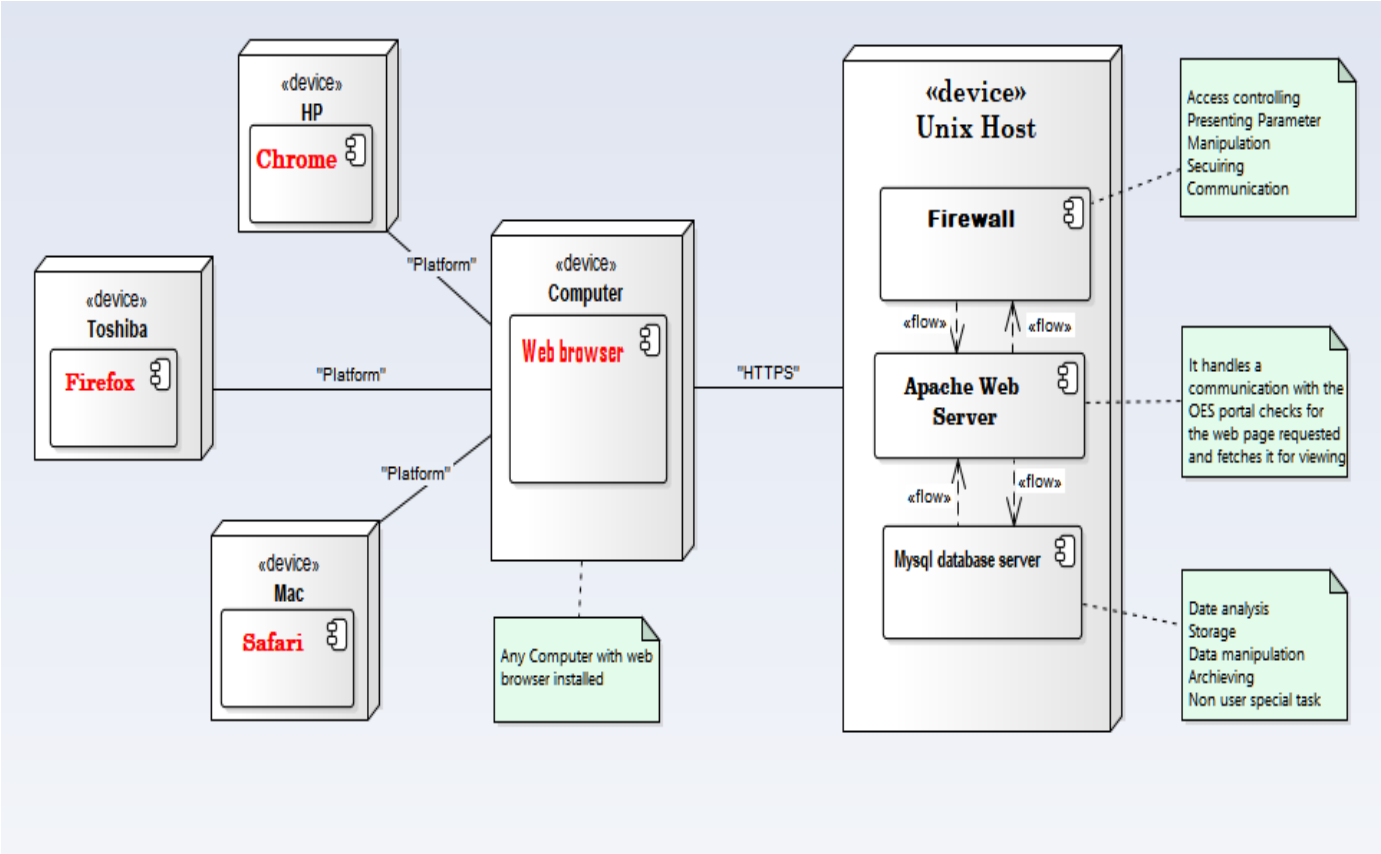



*Figure 3 Hardware or software mapping*

## 4.3.4 COMPONENT DIAGRAM

It is 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

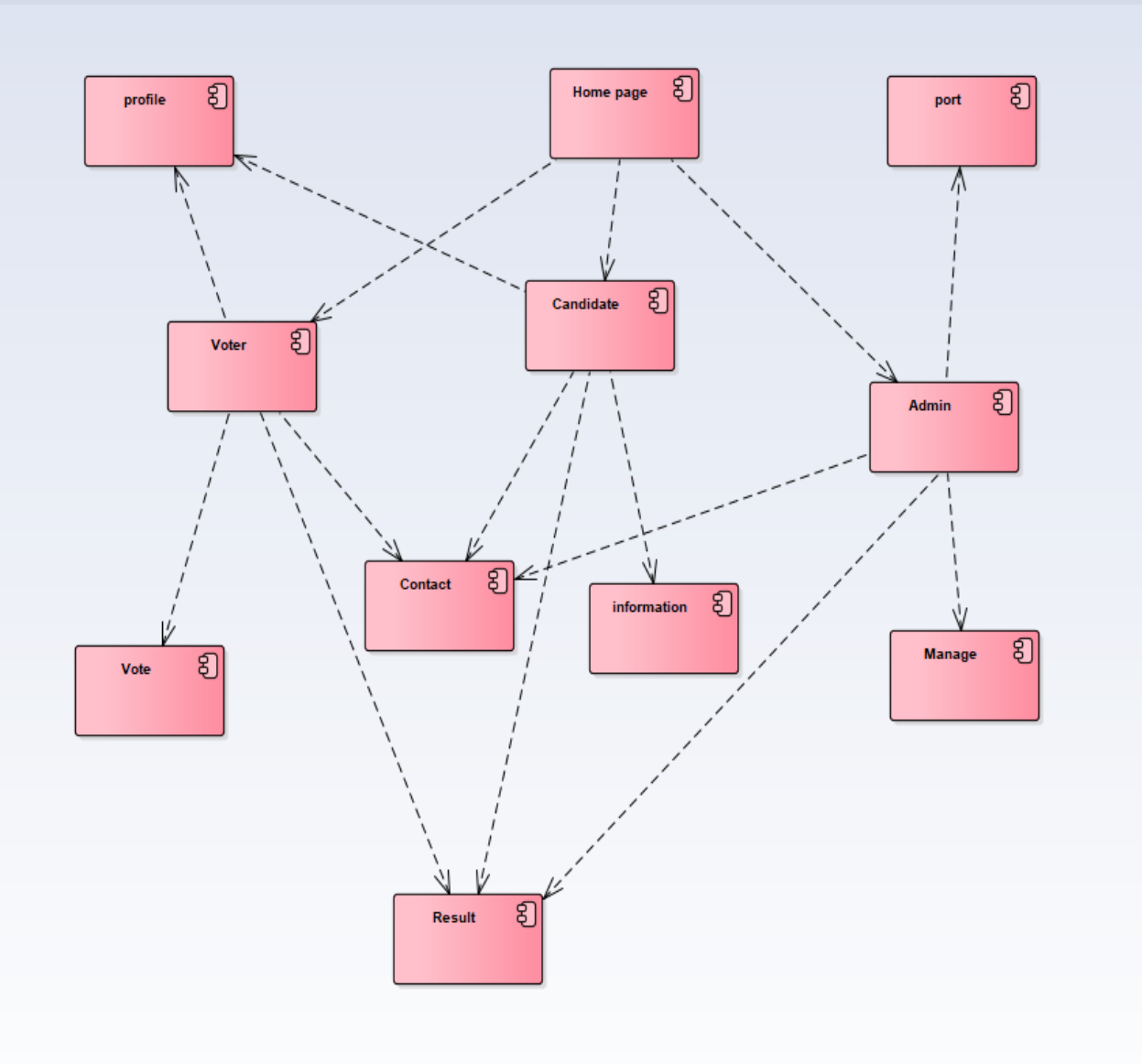



*Figure 4 Component diagram*

## 4.3.5 DATABASE DESIGN

Relationship mapping of each table in a relational database:

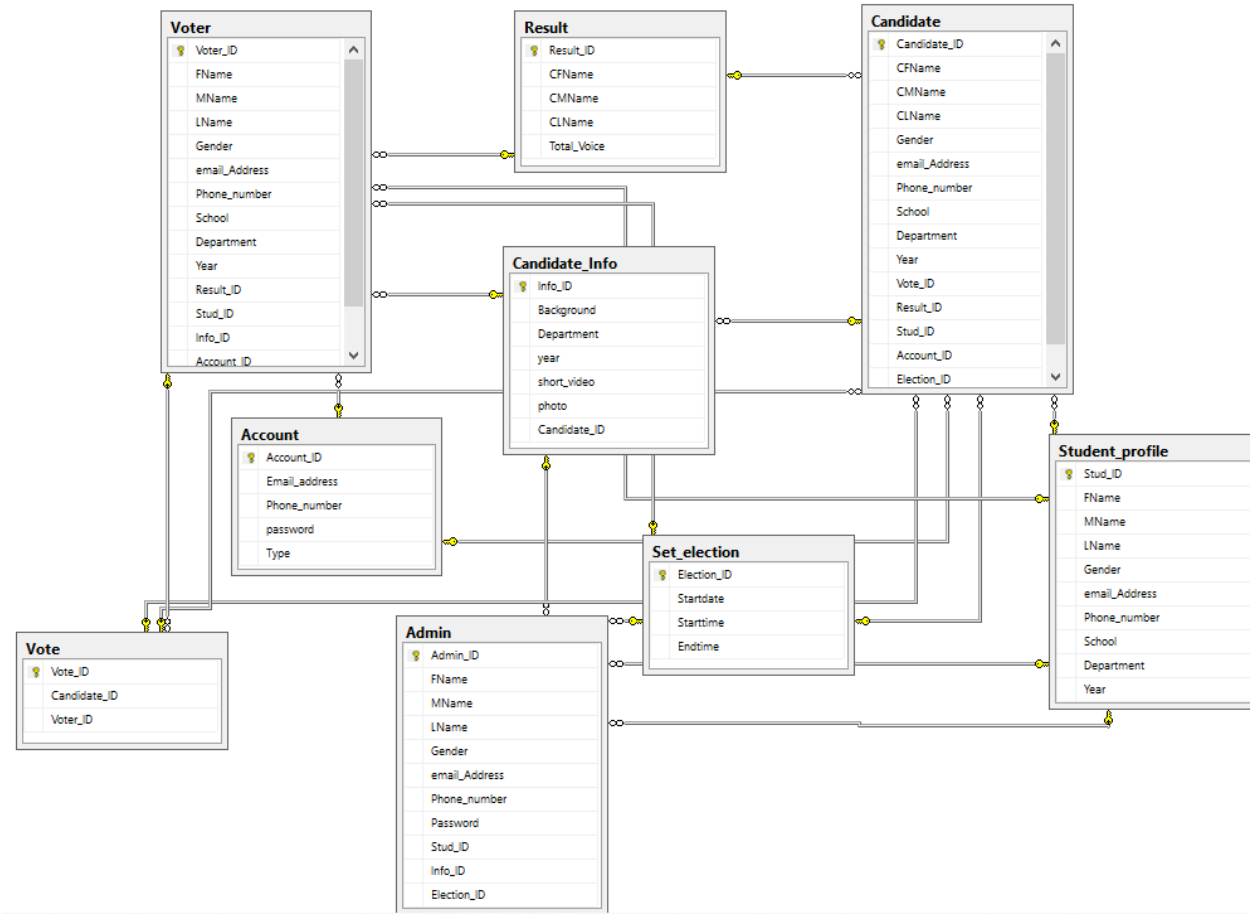
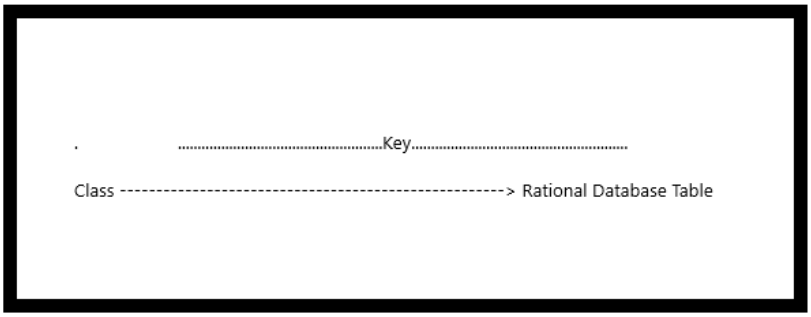



*Figure 5 Database design*

## 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):



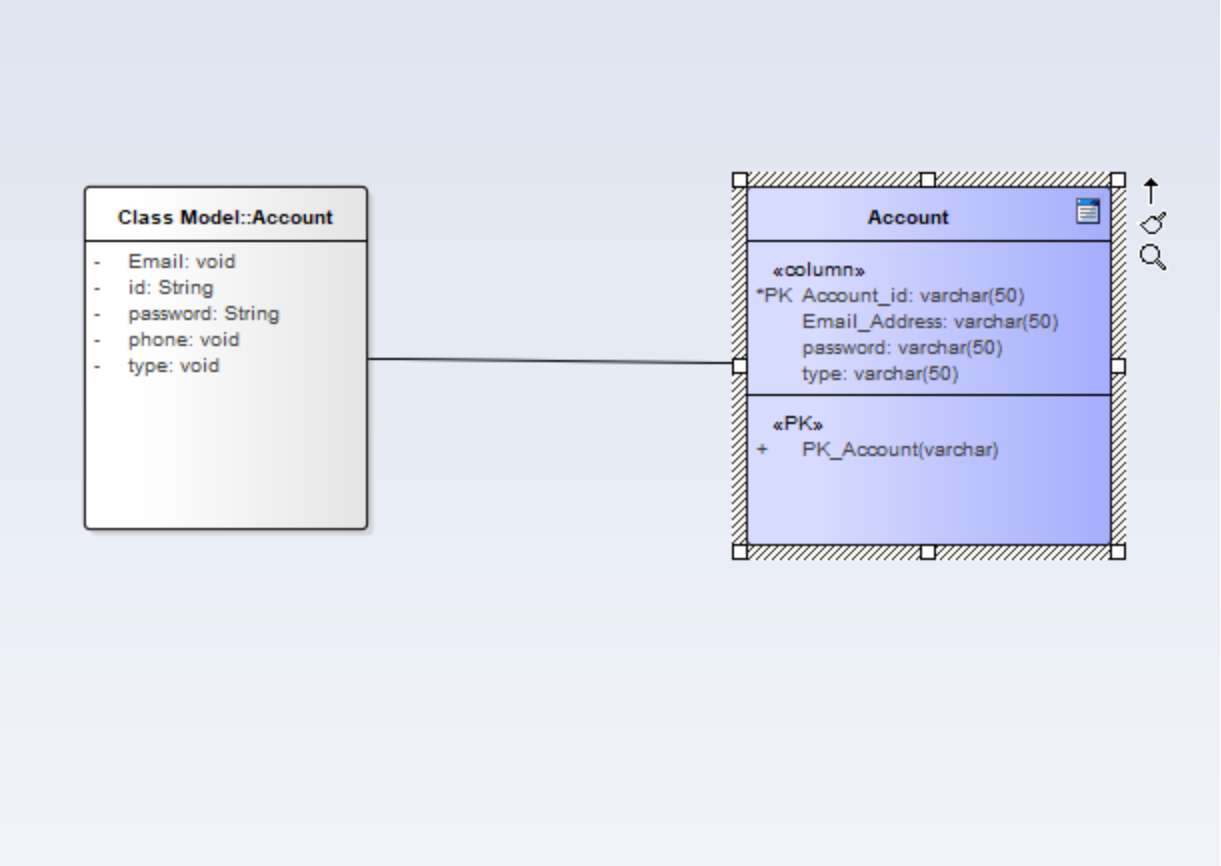



*Figure 6 Object Account mapping*

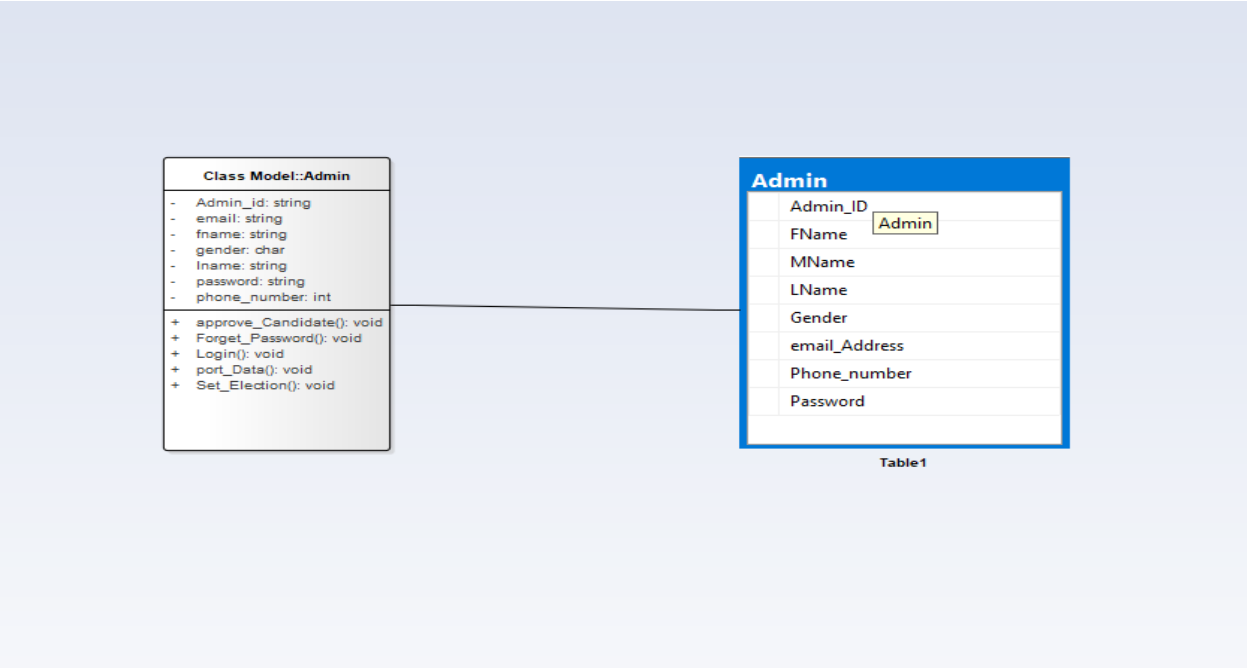



*Figure 7 Objet admin mapping*

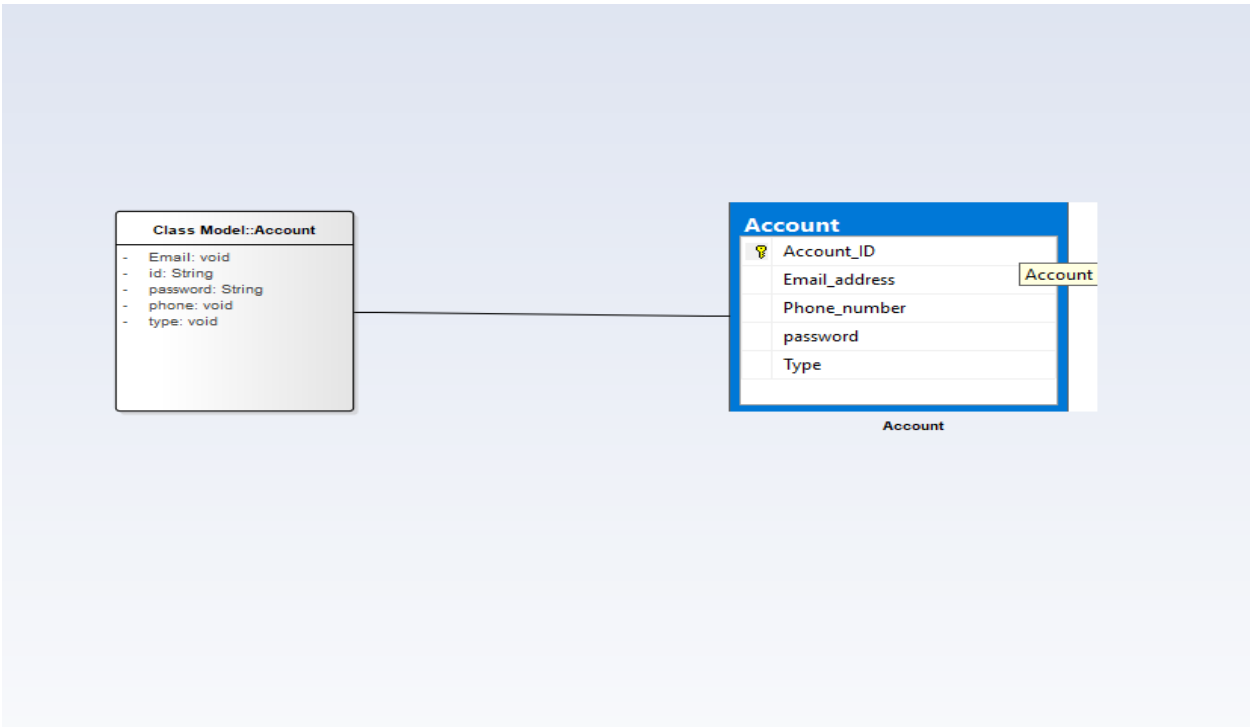



*Figure 8 Object Account mapping*

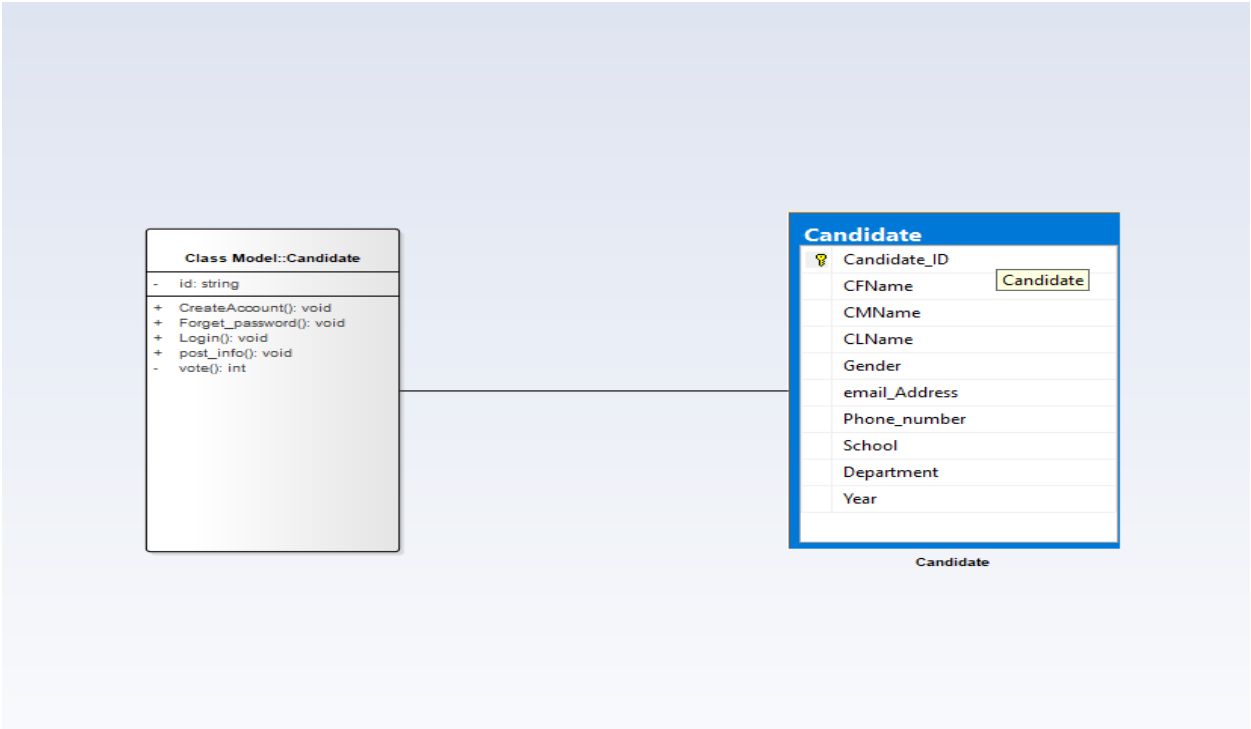



*Figure 9 Object Candidate mapping*

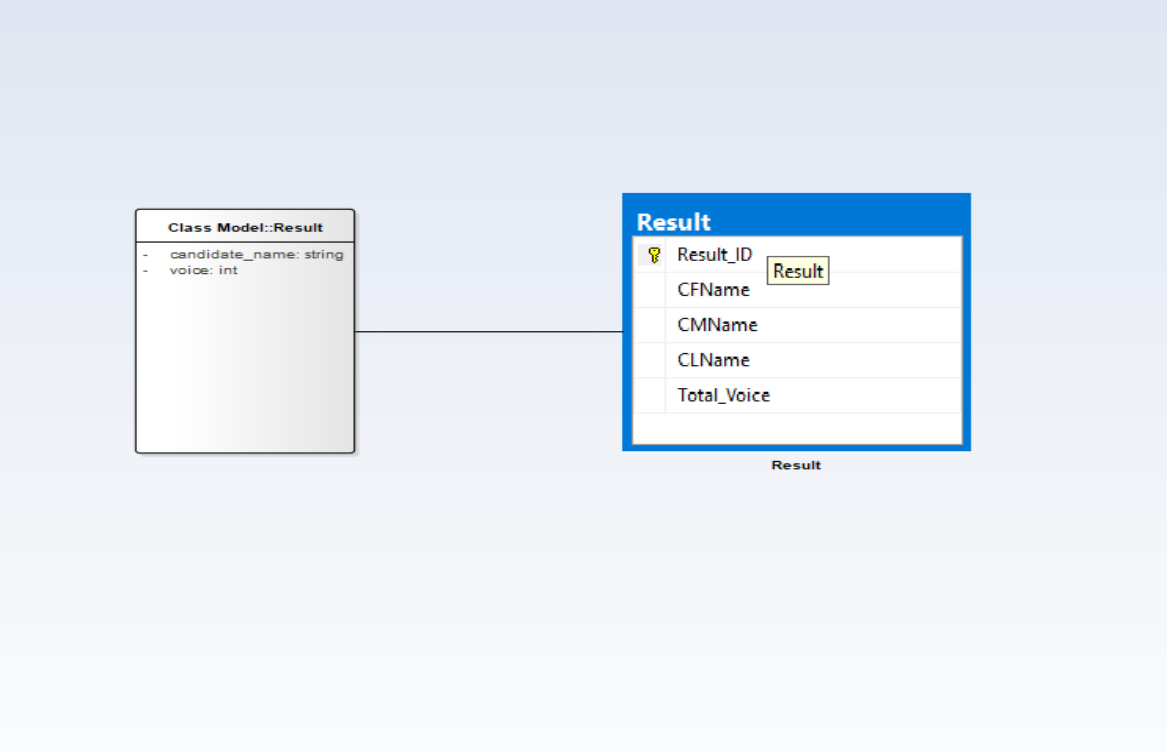



*Figure 10 Object Result mapping*

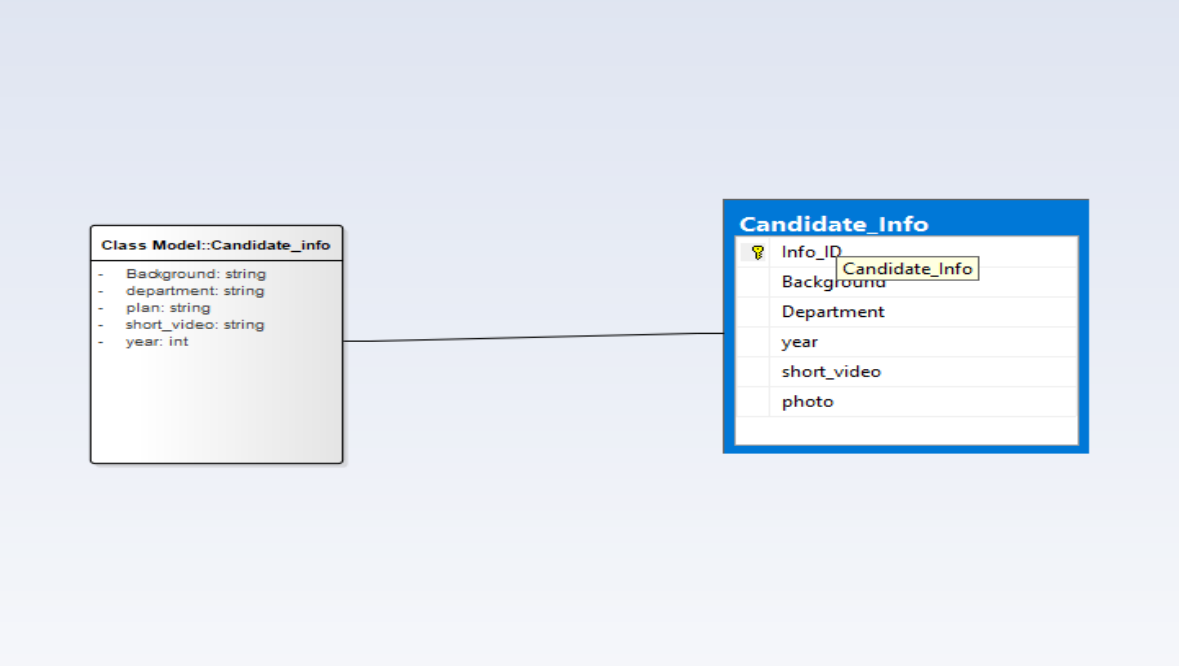



*Figure 11 Object Candidate_info mapping*

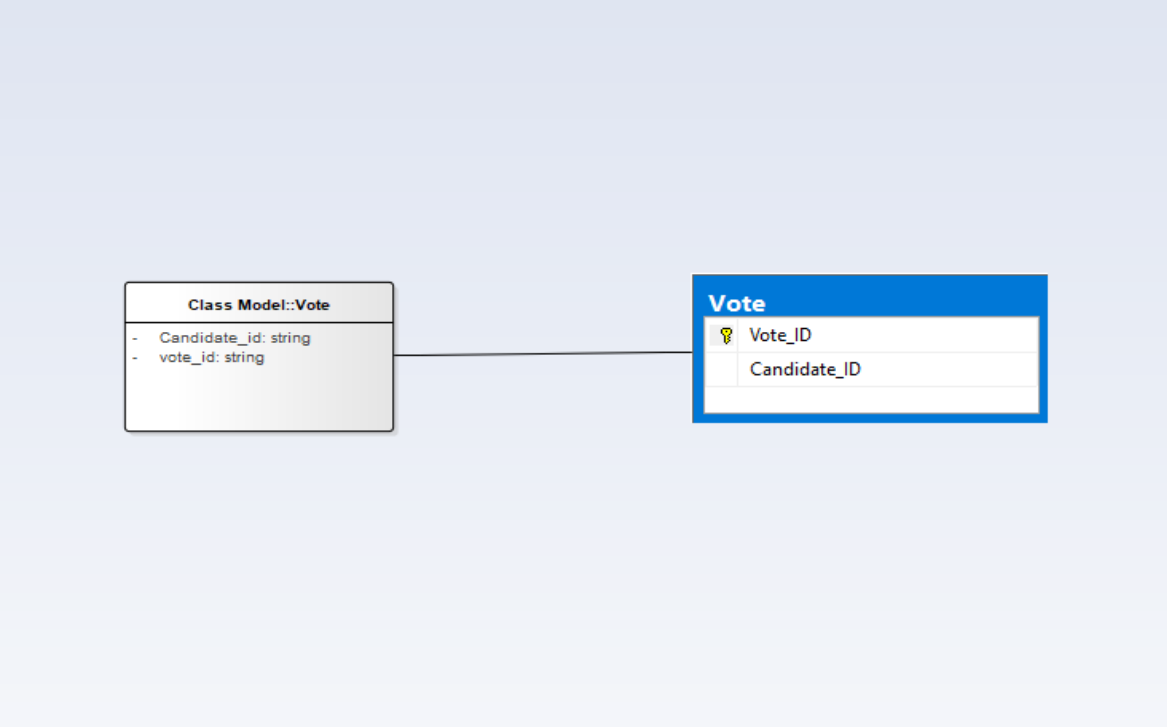



*Figure 12 Object Vote mapping*

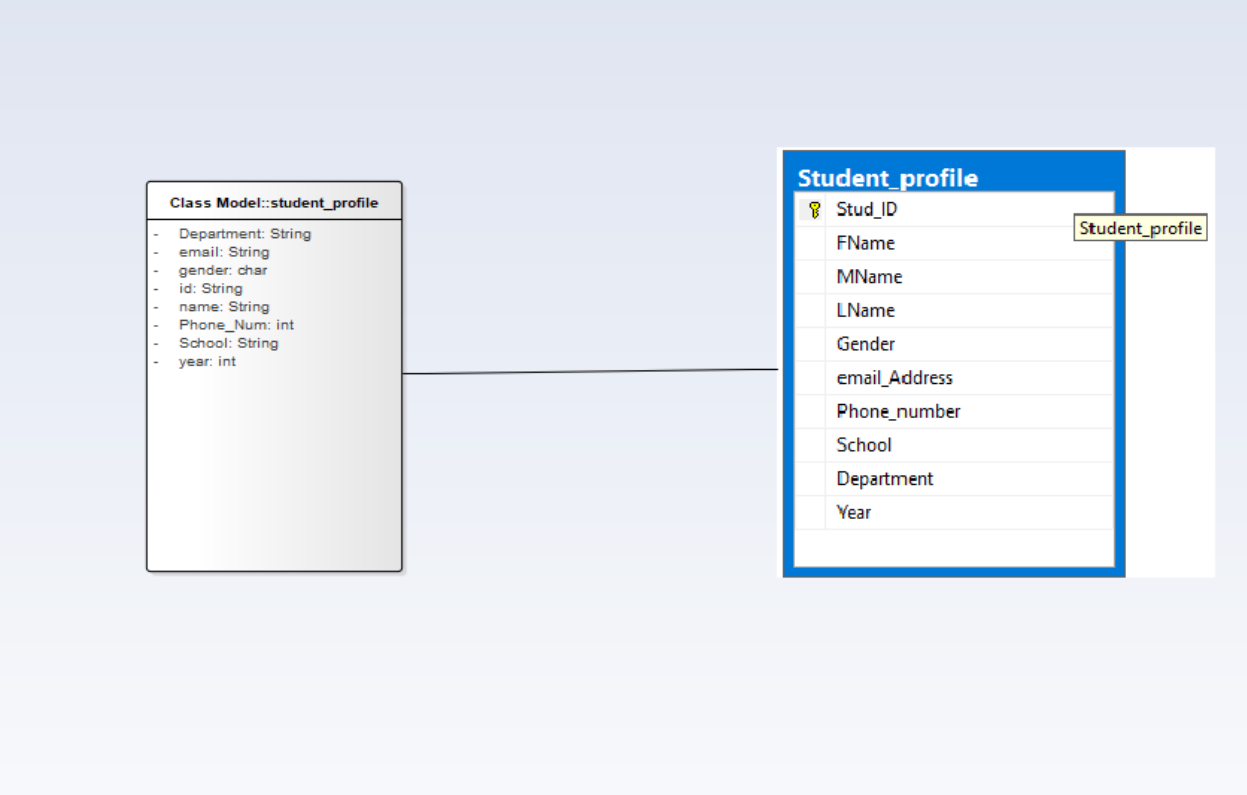



*Figure 13 Object student_profile mapping*

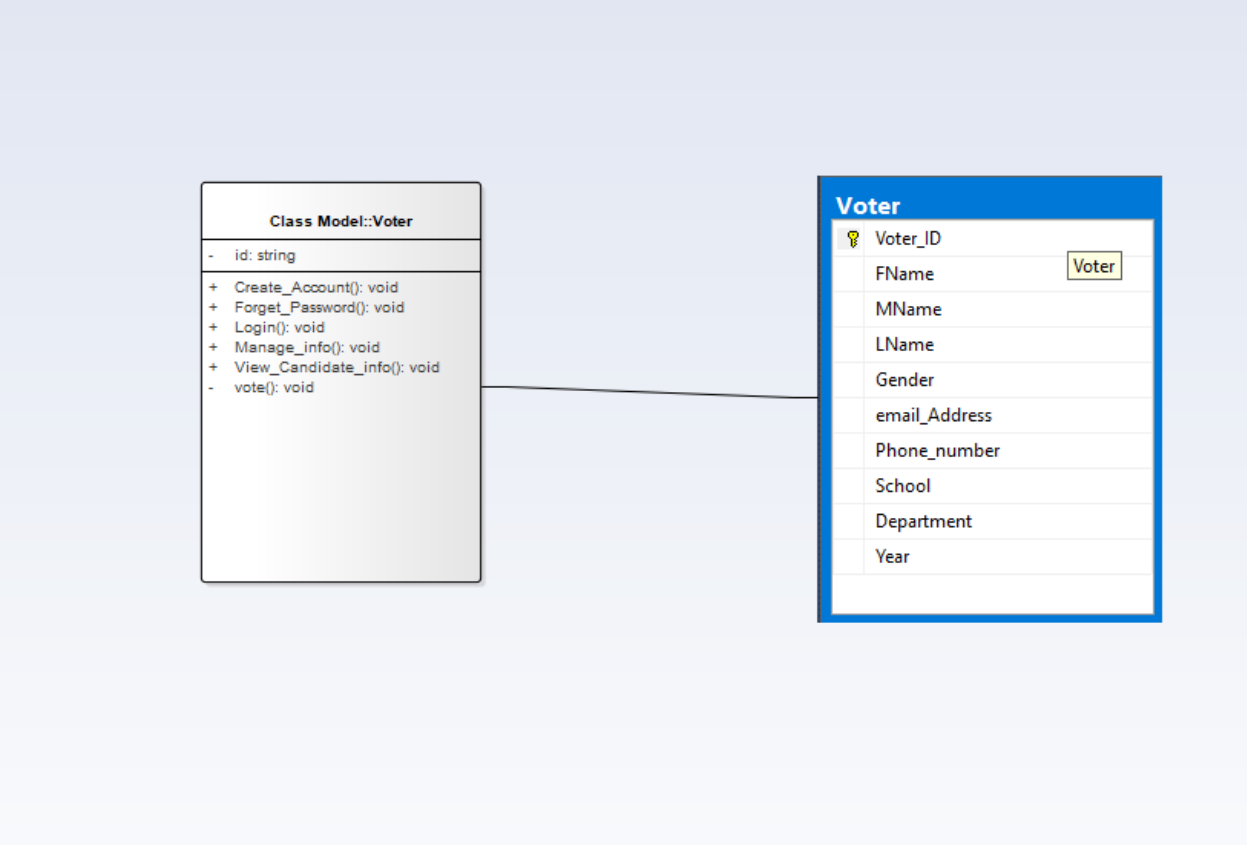



*Figure 14 object Voter mapping*

## 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<br>Actors | Login | Create account | View candidates info |
|---|---|---|---|
| Admin | Get login()<br>Verify()<br>Validate()<br>Set log() | | |
| Voter | Get login()<br>Verify()<br>Validate()<br>Set log() | Get ID()<br>Verify()<br>Validate()<br>create() | Get detail info() |
| Candidates | Get login()<br>Verify() | Get ID()<br>Verify() | |

| | | | |
|---|---|---|---|
| | Validate()<br>Set log() | Validate()<br>create() | |

*Table 1 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()<br>Validate email()<br>Verify security question() | | Activate the system and deactivate when time is over() |
| Voter | Send password reset link()<br>Validate email()<br>Verify security question() | | |
| Candidates | Send password reset link()<br>Validate email()<br>Verify security question() | Add information()<br>Get detail info() | |

*Table 2 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()<br>Send reset link()<br>Set new password()<br>Sent confirmation code() |

| | | | |
|---|---|---|---|
| Voter | | | Add email or phone no()<br>Send reset link()<br>Set new password()<br>Sent confirmation code() |
| Candidates | | | Add email or phone no()<br>Send reset link()<br>Set new password()<br>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()<br>Send reset link()<br>Set new password()<br>Sent confirmation code() |
| Voter | | | Add email or phone no()<br>Send reset link()<br>Set new password()<br>Sent confirmation code() |
| Candidates | | | Add email or phone no()<br>Send reset link()<br>Set new password()<br>Sent confirmation code() |

*Table 3 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.<br>Port number is missing or wrong.<br>Net blocking connection.<br>DB server is down etc. | Verify and test them with ping.<br>Verify it based on my.cnf of SQL DB. |
| Web server fail | Computational/logic Errors<br>Power outages<br>Overheating<br>High humidity<br>Natural disasters, e.g., hurricanes, Floods | Using Server failover.<br>To send a backup automatically on Backup server.<br>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<br>Device driver failures<br>I/O errors, e.g., hard disk failures(see database media failures)<br>Memory parity errors<br>Network hardware failures | defense against full-scale web attacks.<br>Using Server failover. |

*Table 4 Boundary condition failures*

## 4.4 SUBSYSETM SERVICES

| | **Subsystem name** | **Subsystem service** |
|---|---|---|
| **1** | Registration System | • These subsystem provide the list of students those belongs to the University to the OES.<br>• Student can only register if there name found in the list given by these subsystem. |
| **2** | Management subsystem | • Management subsystem helps Admin to prove or disprove the pending request sent by students depending on the predetermined rule and regulation of the university.<br>• Admin can also manage posted candidate information using these subsystem.<br>• He can also forget and change his own password. |
| **3** | Voting subsystem | • Allows already registered student to vote there representative by login into these subsystem.<br>• These subsystem capable of displaying the result of election through the election processes. |
| **4** | Information subsystem | • Candidate can post there information by the help of information subsystem<br>• The information posted by the candidate are visible to the voter. |

*Table 5 Subsystem services*