****

**EGERTON UNIVERSITY** **NJORO CAMPUS**

**NAME: JOSEPH WERU GITHUI**

**REG: S13/02314/19**

**SOFTWARE DESIGN DOCUMENT**

**YEAR: 4 (2023)**

**INSTRUCTOR: Dr. PHOEBE**

**Overview**

This System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

Table of Contents

[1 INTRODUCTION 4](#_Toc130761115)

[1.1 Purpose and Scope 4](#_Toc130761116)

[1.2 Project Executive Summary 5](#_Toc130761117)

[1.2.1 System Overview 5](#_Toc130761118)

[1.2.2 Design Constraints 6](#_Toc130761119)

[1.2.3 Future Contingencies 6](#_Toc130761120)

[1.3 Glossary 7](#_Toc130761121)

[2.0 SYSTEM ARCHITECTURE 8](#_Toc130761122)

[2.1 System Hardware Architecture 8](#_Toc130761123)

[2.2 System Software Architecture 8](#_Toc130761124)

[2.3 Internal communication architecture 9](#_Toc130761125)

[3.0 FILE AND DATABASE DESIGN 10](#_Toc130761126)

[3.1 File and Database Design 10](#_Toc130761127)

[3.1.1 Entity relationship diagram 10](#_Toc130761128)

[3.1.2 Physical database design 11](#_Toc130761129)

[3.1.3 Data dictionary 11](#_Toc130761130)

[4.0 HUMAN MACHINE INTERFACE. 13](#_Toc130761131)

[4.1 INPUTS 13](#_Toc130761132)

[4.1.1 Admin page 14](#_Toc130761133)

[4.1.2 ADD ELECTION PAGE 14](#_Toc130761134)

[4.1.3 ADD ELECTION PAGE 15](#_Toc130761135)

[4.1.4 ADD VOTER PAGE 16](#_Toc130761136)

[4.1.5 ADD NOMINEE PAGE 16](#_Toc130761137)

[4.1.6 STUDENT LOGIN PAGE 17](#_Toc130761138)

[4.1.7 VOTER SELECT ELECTION PAGE 17](#_Toc130761139)

[4.2 OUTPUTS 18](#_Toc130761140)

[4.2.1 VOTE RESULT PAGE 18](#_Toc130761141)

[4.2.2 VOTER SELECTS NOMINEE PAGE 19](#_Toc130761142)

[4.2.3 DELETE AN ELECTION 19](#_Toc130761143)

[5.1 Hardware Detailed Design 20](#_Toc130761144)

[5.2 Software Detailed Design 20](#_Toc130761145)

[5.2.1 Activity Diagrams 21](#_Toc130761146)

[5.3 Internal Communications Detailed Designs 24](#_Toc130761147)

[6.1 Internal Architecture 25](#_Toc130761148)

[6.2 Interface Detailed Design 25](#_Toc130761149)

# INTRODUCTION

The System Design Document (SDD) describes how the functional and nonfunctional requirements recorded in the Software Requirement Specifications (SRS), the preliminary user-oriented functional design and the preliminary data design transform into more technical system design specifications from which the system built. It documents the System Design Document, highlights the high-level system design and low-level detailed design specifications. It also describes design goals and considerations, provides a high level of the system architecture, and describes the data design associated with the system as well as the human-machine interface and operational scenarios. The high-level system design is further decomposed into low-level detailed design specifications for each system component, including hardware, internal communications, software, system integrity controls, and external interfaces.

## Purpose and Scope

The System Design Document for an online voting system documents and tracks the necessary information required to effectively define architecture and system design in order to give the development team guidance on the architecture of the system to be developed. The intent of this system is to automate voting activities that are currently carried on manually. Its intended audience is the project manager, project team, and development team. Some portions of this document, such as the user interface (UI), may be shared with the client/user, and other stakeholders whose input/approval into the UI is needed.

The scope of this document covers the various operations that were specified in the software requirement specification, the procedures that must be followed during the design of the system in order to ensure that they are properly developed. They include the following.

* Voter verification.
* Voter registration.
* Candidate registration.
* Adding of elections.
* Voting.
* Results report.

## Project Executive Summary

### System Overview

Traditional voting involves use of election administrators to control the election preceding from the start of the process, that is, voter registration, voter verification, actual voting, tallying and finally keeping records.

This process is tiresome when a large population is involved. It is difficult to conduct elections this way due to the bulkiness of the processes involved. A lot of paperwork is involved and managing such a process is hectic. Due to requiring a lot of paperwork and personnel, the process becomes more expensive with every new entry required.

After elections, there are very many documents to be managed according to the number of elective positions, contestants and voters as well as the organization’s requirements on election conducts. Managing all these records is expensive, time consuming, tiresome and also space consuming.

With this proposed system, these problems will be solved since the number of personnel involved will be reduced, the paperwork also reduced and record keeping made easier and in the end the process becomes cheaper and less time consuming.

The voting system basically automates the functionalities of every user in the process without denying them of any capabilities that were previously possible using the traditional method.

Below is a simplified use case that outlines how different users will be interacting with the system.

**Voting system**

Select Election

Vote

Logout

Login

**Voter**

**Voting system**

Add Election

Add Position

Add Nominee

Login

**ADMIN**

Add Voters

View Results

Logout

### Design Constraints

The system being web based, will be built using PHP. This ensures that connection to the database will be secure from any SQL injections that might be attempted. JavaScript will be used to make sure that the system is fully interactive and validates all the data entered by the clients. HTML, CSS and bootstrap will be used in the making of the user interface.

The system makes use of internet and during development, it is assumed that the end users have or can acquire internet for the use of the system.

The system does not conflict with any other system that exists in the market.

### Future Contingencies

During the development of online voting system, there might be changes in the development process which might be caused by the following factors:

User interface might be fine-tuned for usability purposes.

Use of web services to facilitate communication with external interfaces such as other systems to ensure election process monitoring.

## Glossary

SQL – Structured Query Language

PDF – Portable Document Format.

PHP – Hypertext Preprocessor

HTTP – Hypertext Transfer protocol

# 2.0 SYSTEM ARCHITECTURE

The general system architecture of an online voting system typically includes the following components:

1. User Interface: This is the part of the system that the voter interacts with, through a web browser. It provides the user with an interface to cast their vote and also allows them to view the status of their vote.
2. Application Server: This is the main processing unit of the system. It receives requests from the user interface and processes them accordingly. It also stores all the data related to the voting process, such as the list of candidates, voter registration data, and voting results.
3. Database Server: This is the storage unit of the system. It stores all the data related to the voting process, such as voter registration data, the list of candidates, and the voting results.
4. Security System: This component ensures the security of the system and the data stored in it. It includes various security mechanisms such as encryption, firewalls, and intrusion detection systems.

When a user casts their vote, a request is sent from the user interface to the application server. The request typically contains information such as the voter's identification and the candidate they are voting for. The application server processes the request, verifies the user's identification, and stores the vote in the database.

After the vote has been stored, the application server sends a response to the user interface indicating that the vote has been successfully cast. The response may also contain additional information, such as the status of the vote or the total number of votes cast so far.

Throughout the voting process, the security system constantly monitors the system for any potential security breaches and takes appropriate measures to prevent them.

## System Hardware Architecture

The online voting system being a web application requires a few hardware devices. The system needs a computerized device that has access to a stable internet connection, a server to store the data and process requests.

The computerized device can either be a desktop computer, a laptop or any internet abled hand gadget that supports a browser.

## System Software Architecture

The system will have two tiers;

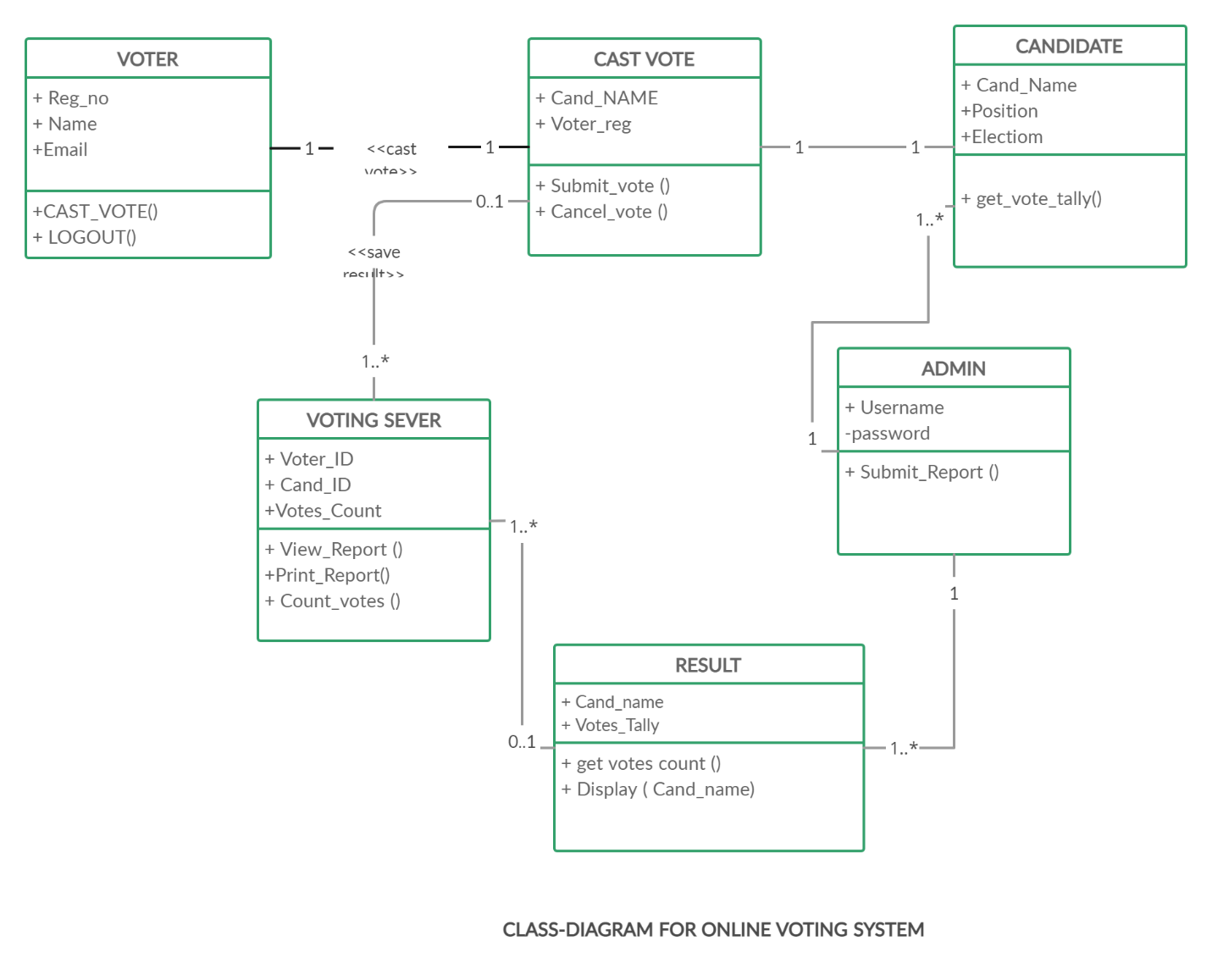
The admin tier that will include functionalities only meant for the administrators and the voter tier that will include the voting functionality for the voter.

In the first tier, the voter can; Login using their ID registered by Admin in the system, Select Election, Vote and Logout.

In the second tier, the admin can; Login using a username and a password, add election, add position, add nominees, add voters, view election results and Logout to end session.

Below is a class diagram for the online voting system.

Fig. Class Diagram for online voting system.



## Internal communication architecture

The communication process of the online voting system will be achieved by internet which will enable communication between the system and the database during user data storage and retrieval of the required data by the user and communication between the different modules of the system. A web browser will be used as an interface in order to make requests to the web server and also renders the response back the user for the requested resource. HTTP (Hyper Text Transfer Protocol) is used to ensure communication between the web server and the web browser.

# 3.0 FILE AND DATABASE DESIGN

## 3.1 File and Database Design

The database used in the online voting system is MySQL. Different methods such as index will be used to access the data stored in the database.

The data dictionary describes the content, format, and structure of the database and the relationship between its elements that is used to control access to and manipulation of the database. Is contains the metadata, that is, data about the database.

Different access methods such as indexed and set will be used to access data in the database.

### 3.1.1 Entity relationship diagram

ORGANIZATION

Adds

NOMINEE

**VOTER**

ADMIN

Added by

elects

M 1

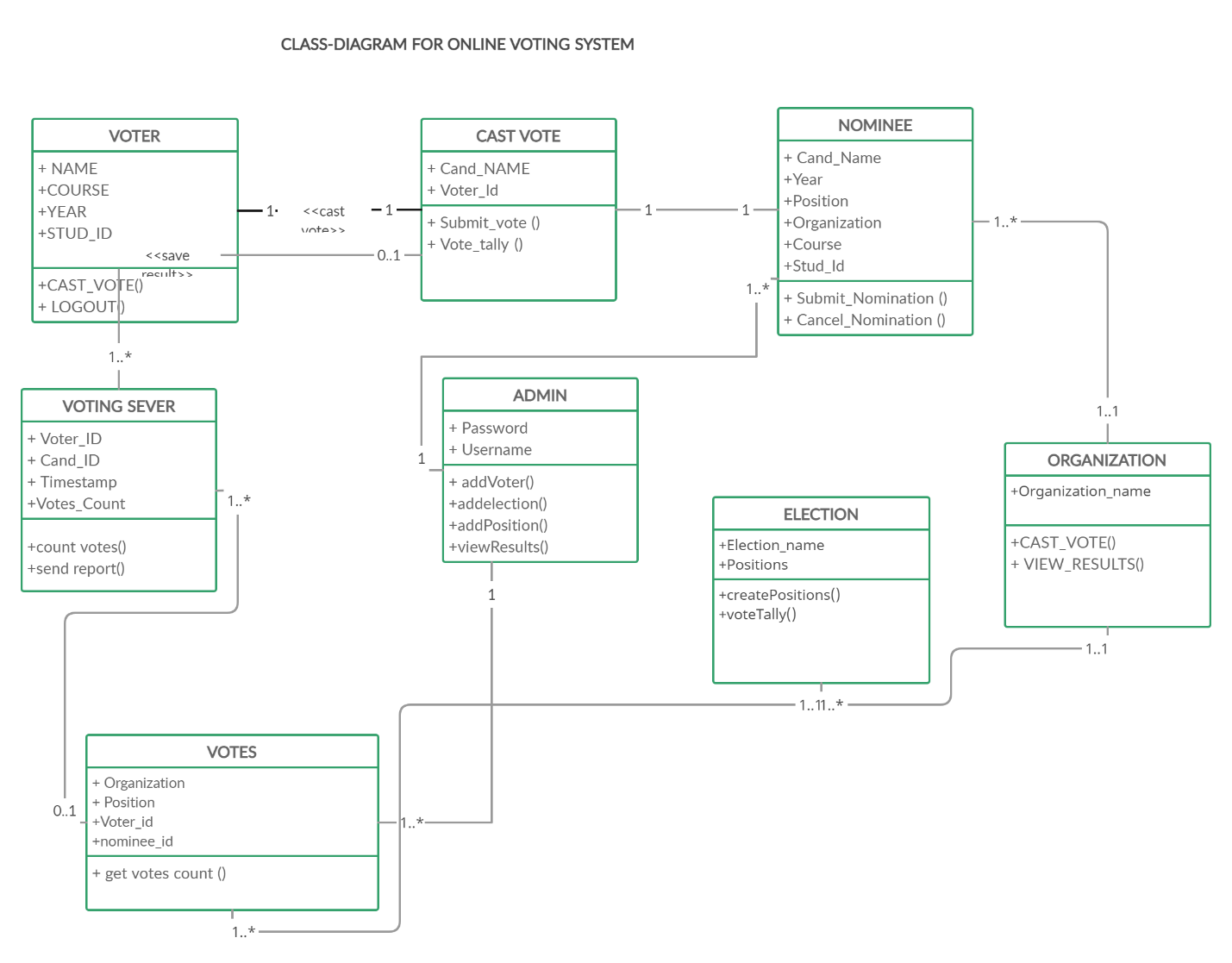
1

1 M

Belong to

M 1

### 3.1.2 Physical database design



### 3.1.3 Data dictionary

The following data dictionary contains the database’s metadata (data about the database)

**NOMINEES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Attribute | Data type | Nullable | Constraint | Reference | Description |
| Nominees | Organization | varchar(30) | Not null |  | Organization | organization |
| Nominees | Position | varchar(30) | Not null |  | election | Elective position |
| Nominees | Name | varchar(30) | Not null |  |  | Nominees name |
| Nominees | Course | varchar(30) | Not null |  |  | Nominee’s course |
| Nominees | Year | Date | Not null |  |  | Academic year |
| Nominees | Stud\_id | int | Not null | Primary key |  | Student’s id |

**ORGANIZATION**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Attribute | Data type | Nullable | Constraint | Reference | Description |
| Organization | Name | varchar(30) | Not null |  |  | Organization’s name |

**POSITIONS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Attribute | Data type | Nullable | Constraint | Reference | Description |
| Positions | Organization | varchar(30) | Not null |  |  | organization |
| Positions | Position | varchar(30) | Not null |  |  | Elective position |

**VOTERS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Attribute | Data type | Nullable | Constraint | Reference | Description |
| Voters | Name | varchar(30) | Not null |  |  | Voter’s name |
| Voters | Course | varchar(30) | Not null |  |  | course |
| Voters | Year | Date | Not null |  |  | Academic year |
| Voters | Stud\_id | int | Not null | Primary key |  | Student’s id |

**VOTES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Attribute | Data type | Nullable | Constraint | Reference | Description |
| Votes | Organization | varchar(30) | Not null |  | Organization | organization |
| Votes | Position | varchar(30) | Not null |  | election | Elective position |
| Votes | Candidate\_id | int | Not null | Secondary key |  | Nominees name |
| Votes | Voter\_id | int | Not null | Primary key |  | Nominee’s course |

# 4.0 HUMAN MACHINE INTERFACE.

## 4.1 INPUTS

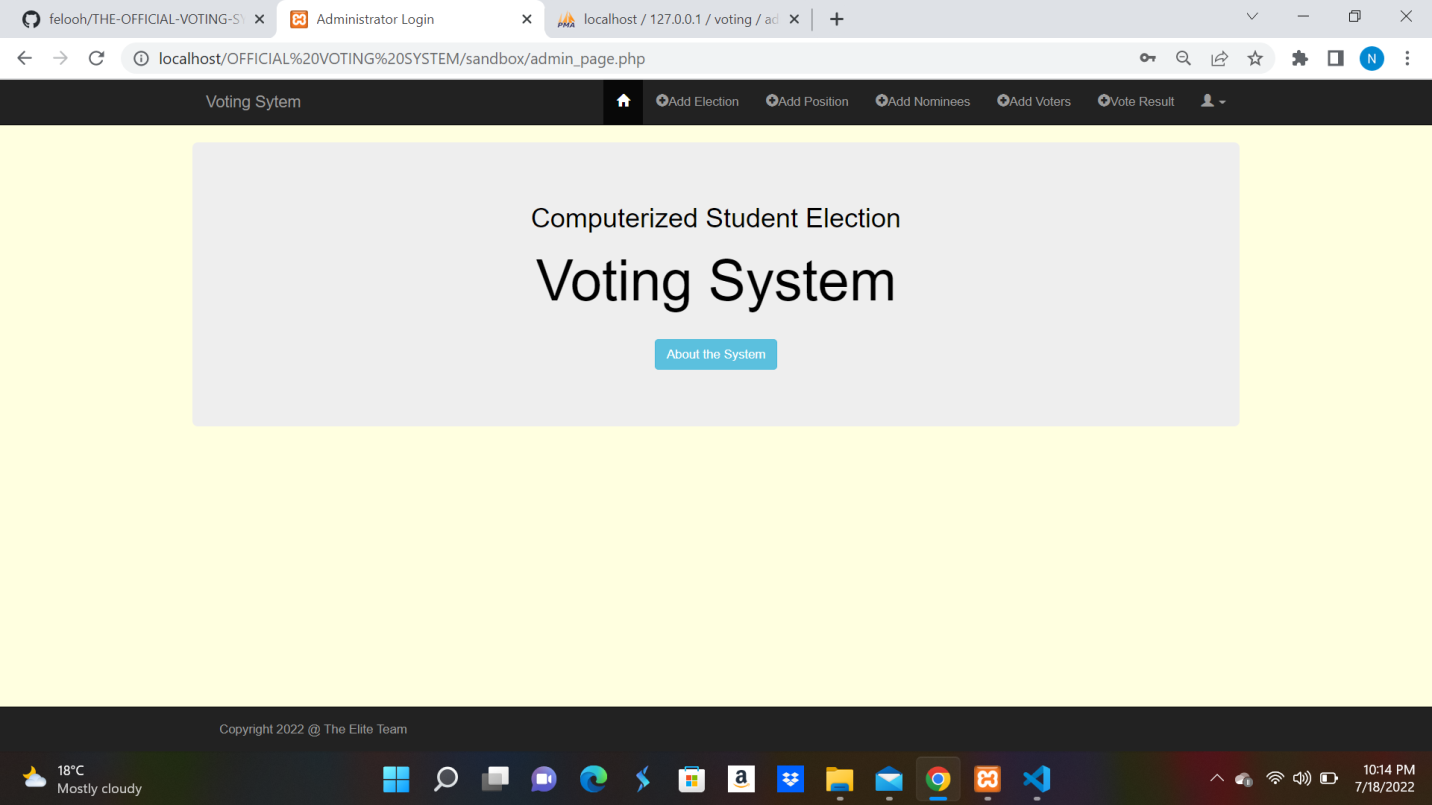
Data input into the system is done by the administrator or sourced from an existing organization’s database. This way, consistency is maintained as there is no need to manage duplicate databases within the same organization. Before data storage, all the data is validated to ensure that administrators do not submit some empty fields or data with errors. Error messages will be displayed so that users will be able to identify where they make mistakes so that they can correct.

The data types for different field are defined such as text, date, number or email so that the user inputs the correct data type. Some fields are unique since they only belong to a particular user and the system ensures that this is catered for.

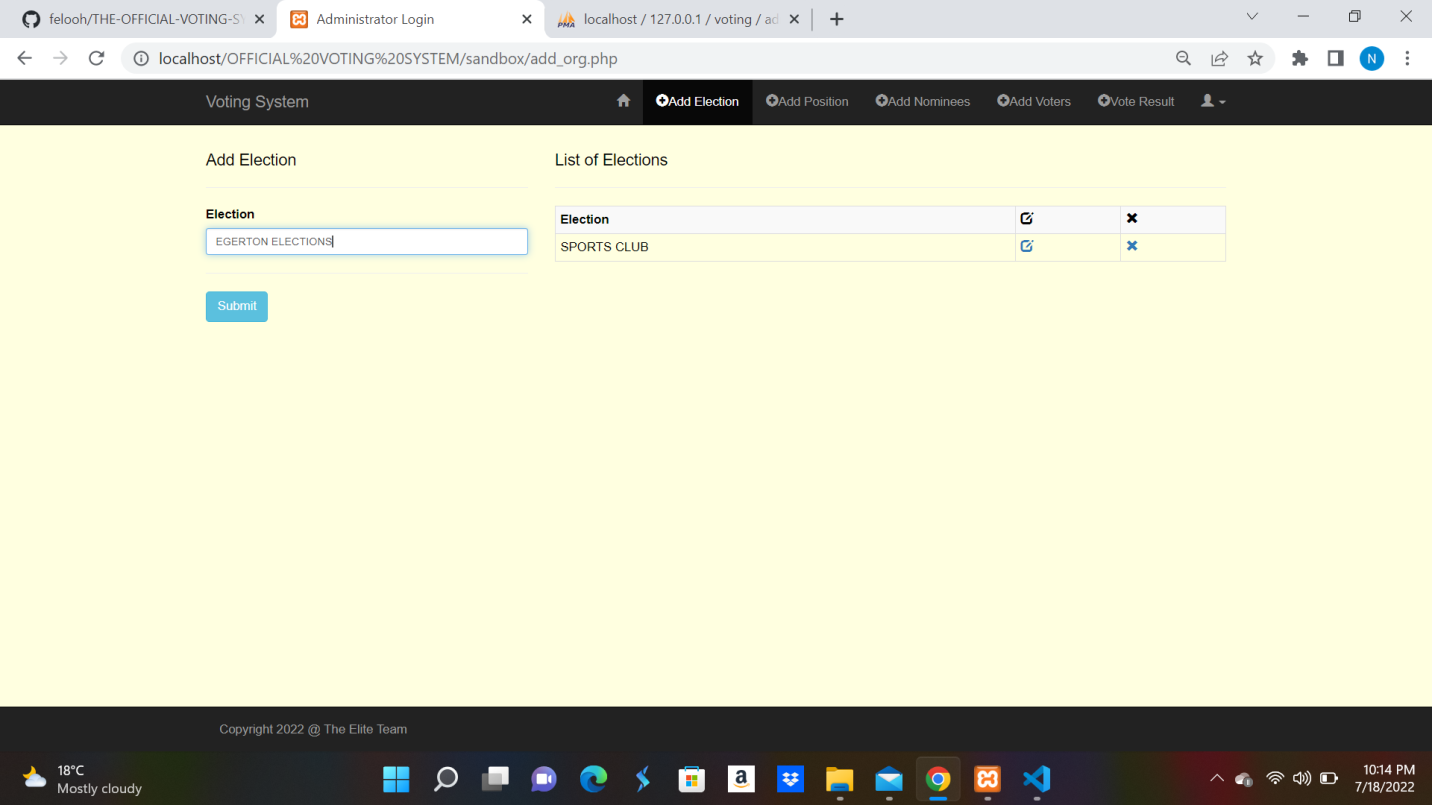
Users will only be registered once to the system, in which an attempt to register another user with the same credentials, there will be an error message indicating that the user already exists.

The following are sample pages in the system to show methods available for data input.

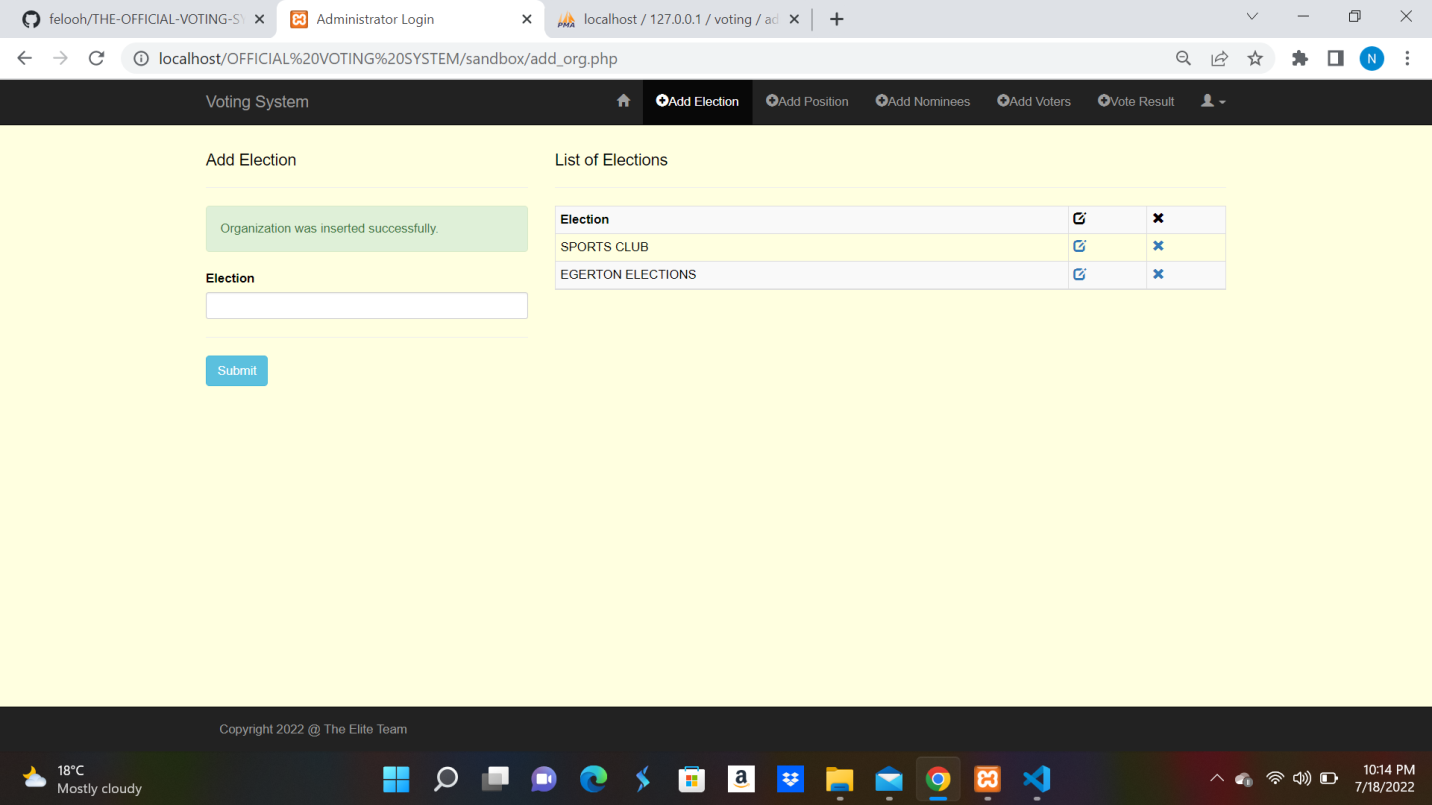
### 4.1.1 Admin page



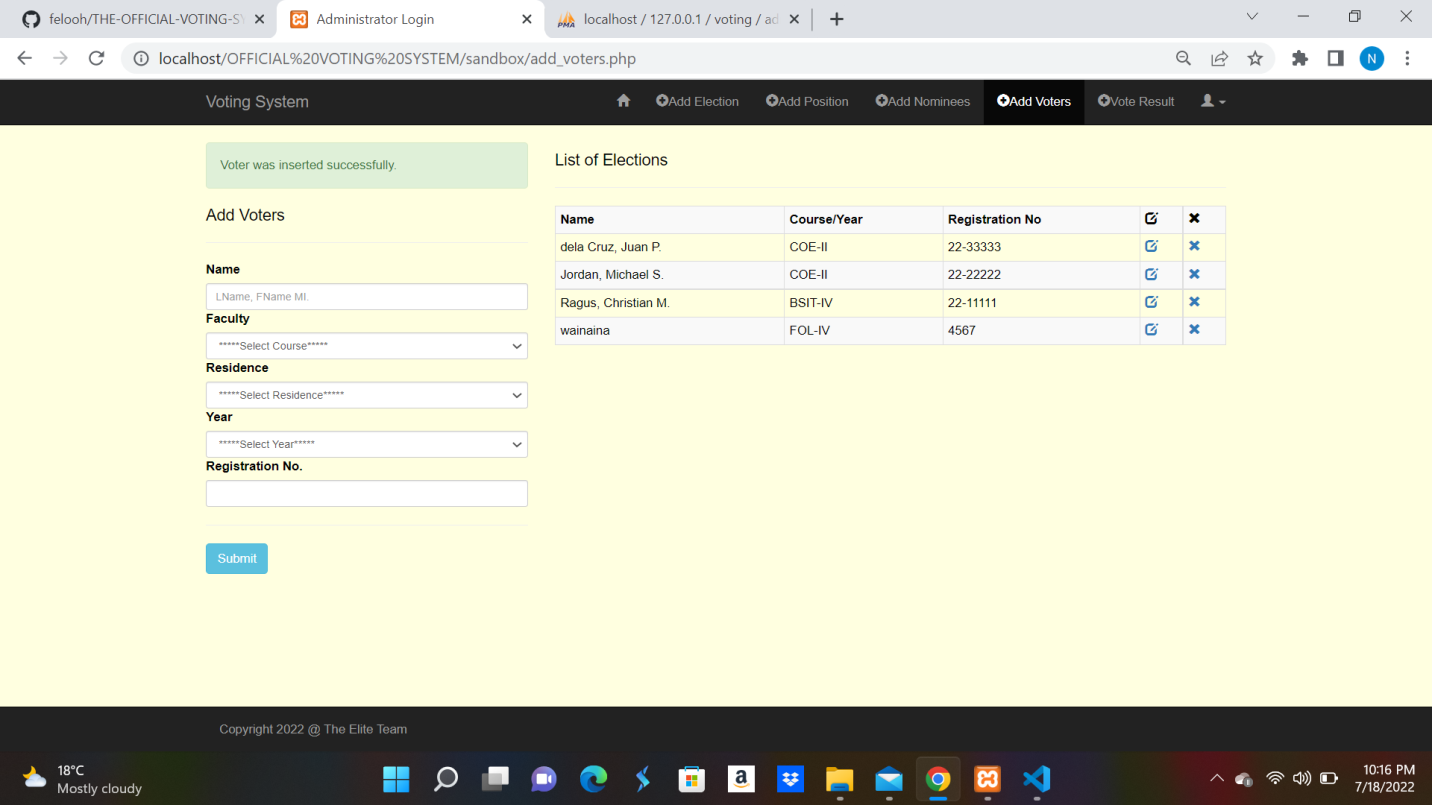
### 4.1.2 ADD ELECTION PAGE



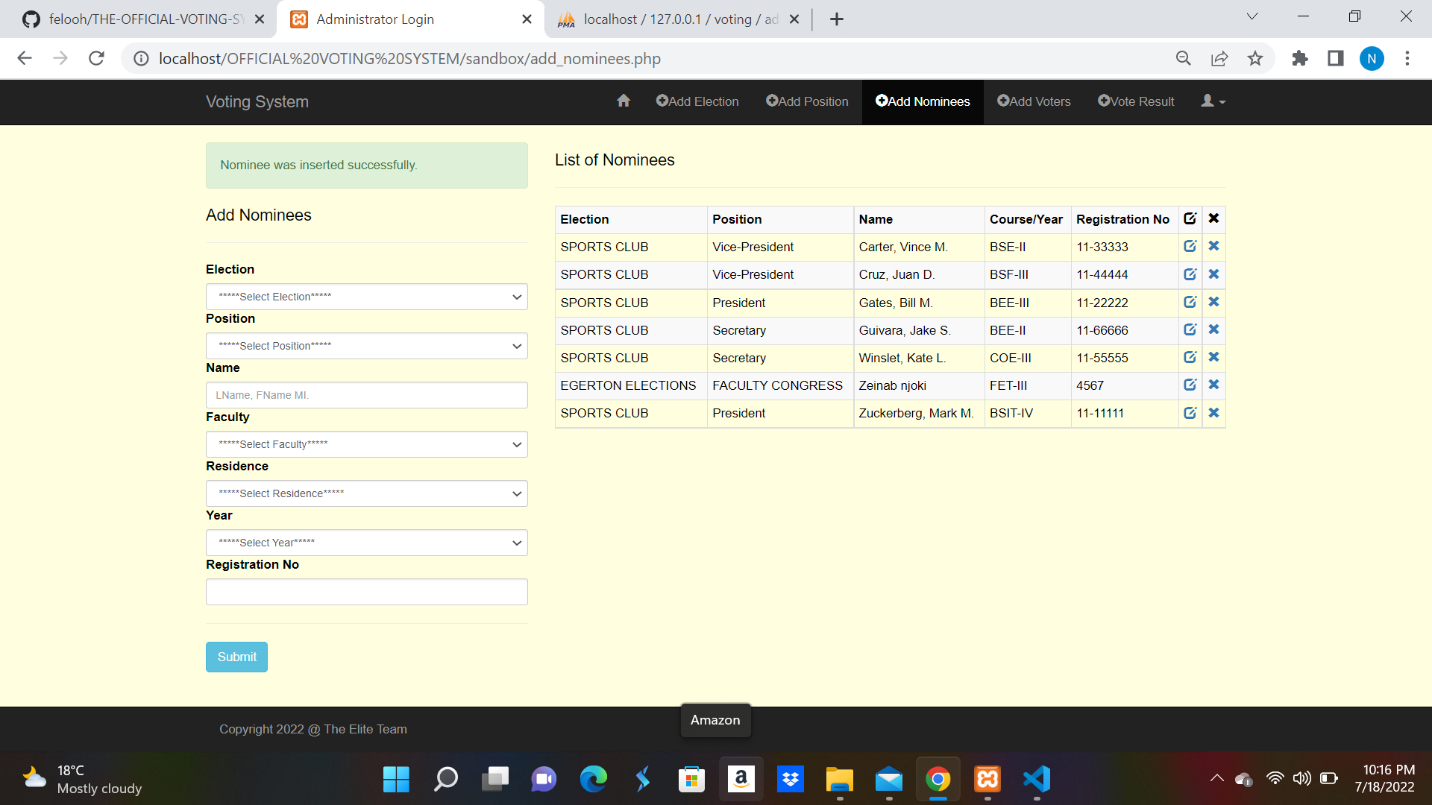
### 4.1.3 ADD ELECTION PAGE



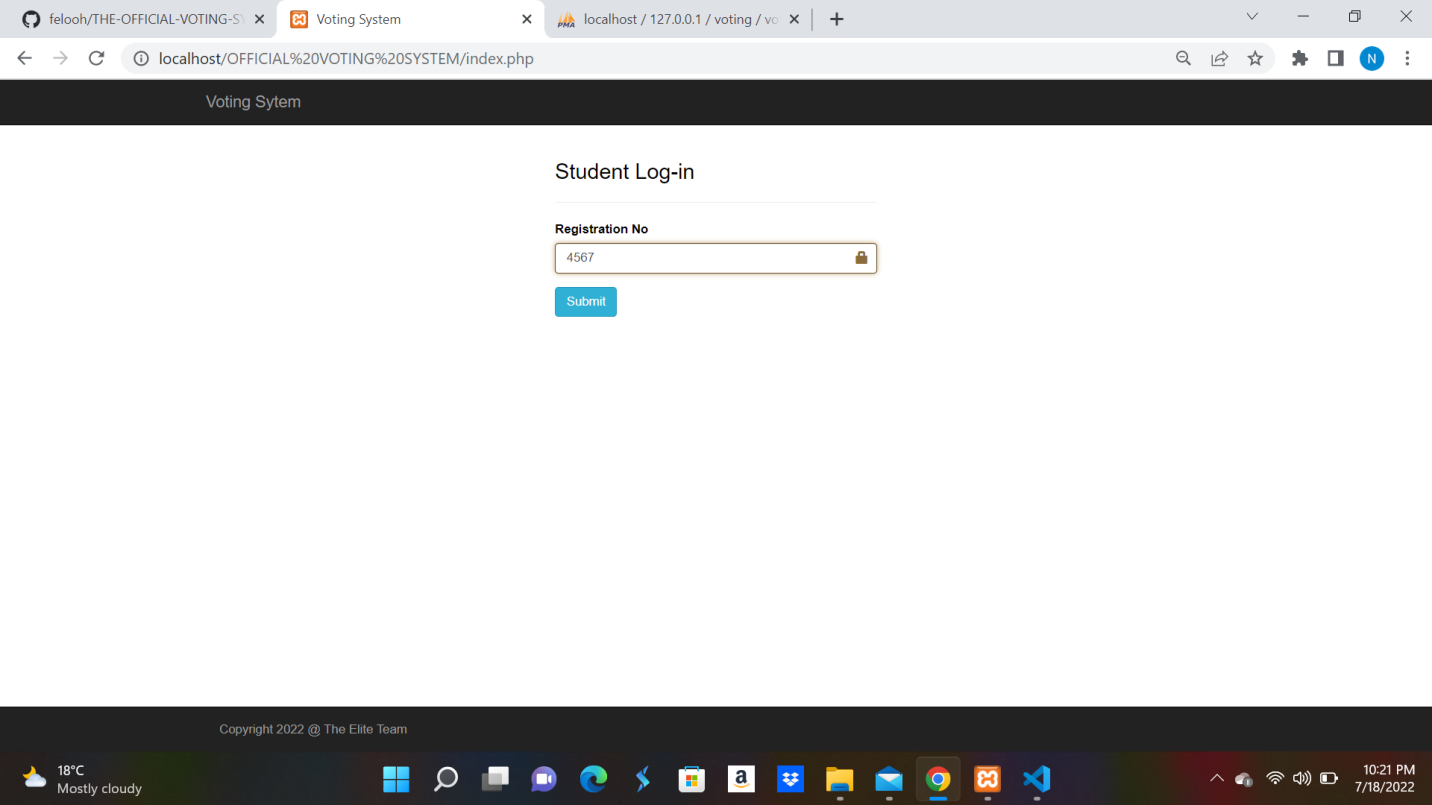
4.1.4 ADD VOTER PAGE

****

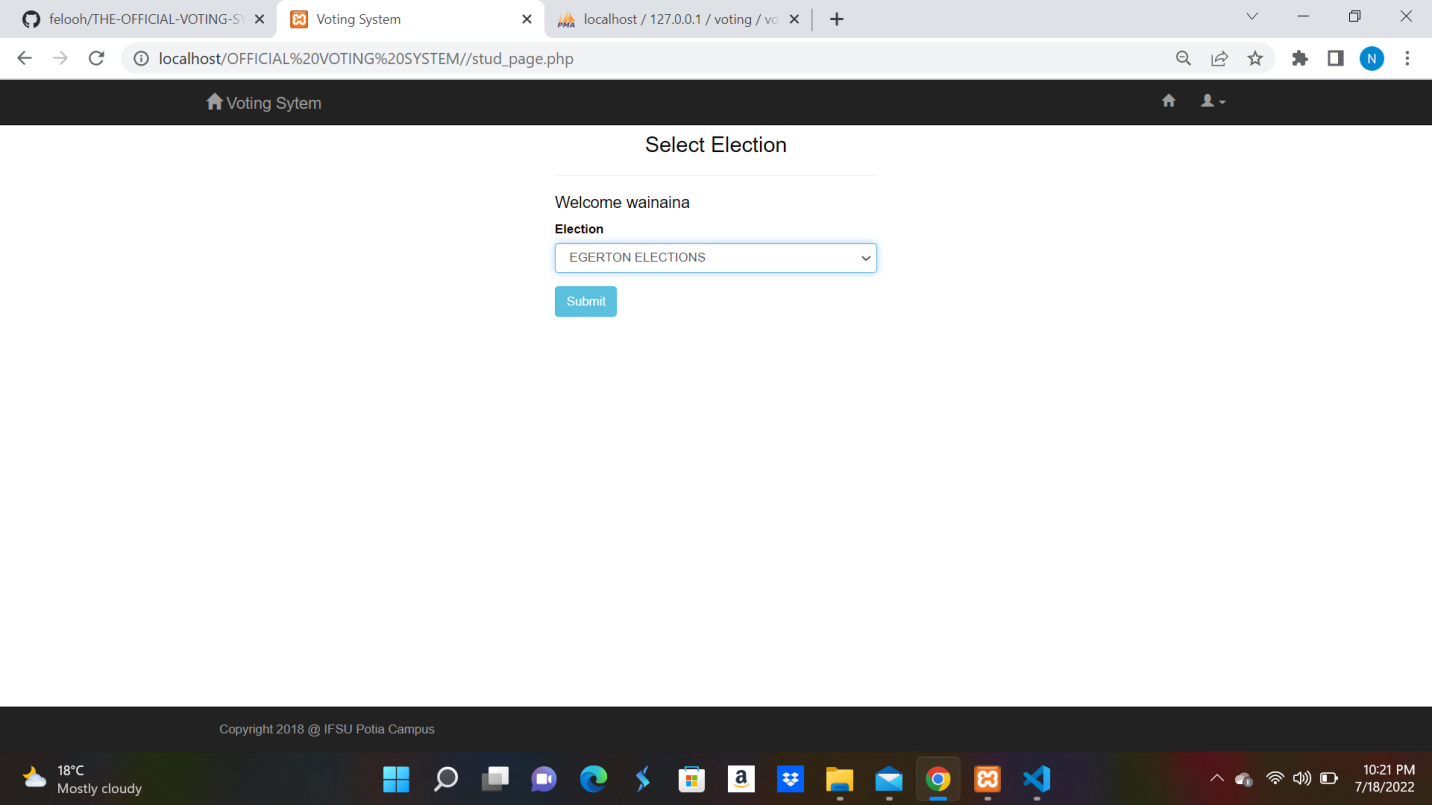
### 4.1.5 ADD NOMINEE PAGE

****

### 4.1.6 STUDENT LOGIN PAGE

****

### 4.1.7 VOTER SELECT ELECTION PAGE



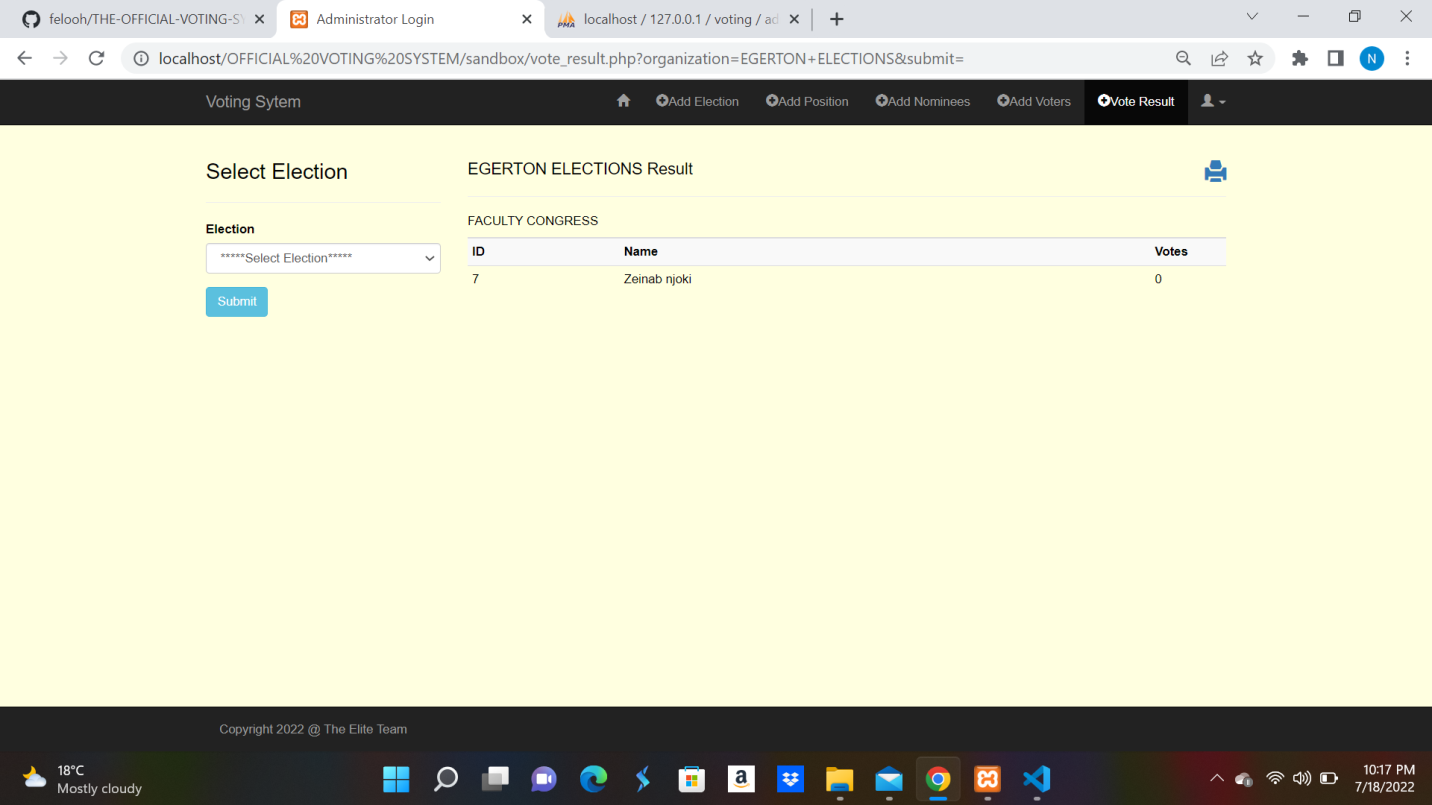
## 4.2 OUTPUTS

The online voting system will have several outputs in different forms. Some of those forms are;

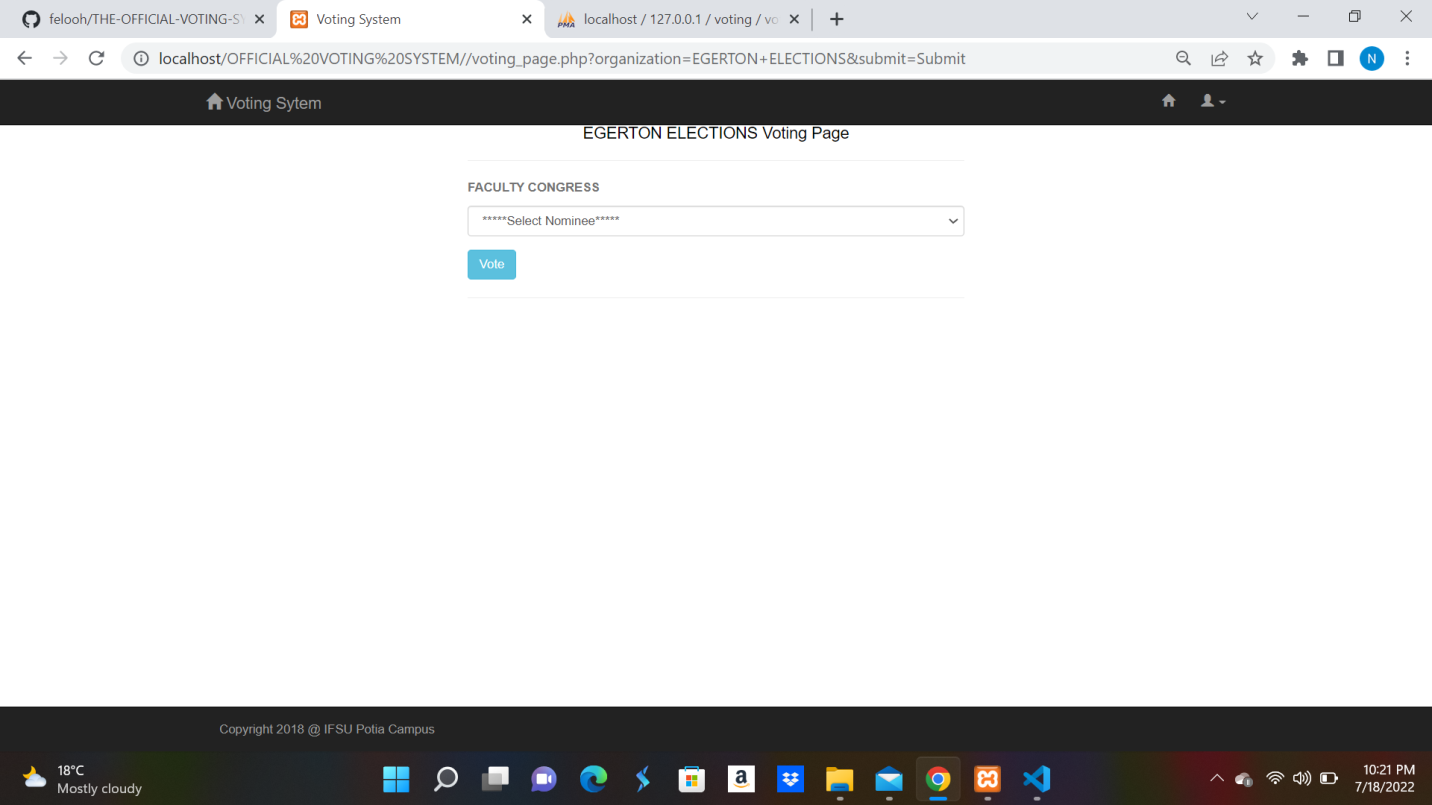
* Graphical User Interfaces – Are messages displayed directly on the screen during the use of the system.
* Print outs – The election results of a specific election will be printed out by the system.

The following are representations of output pages in the system.

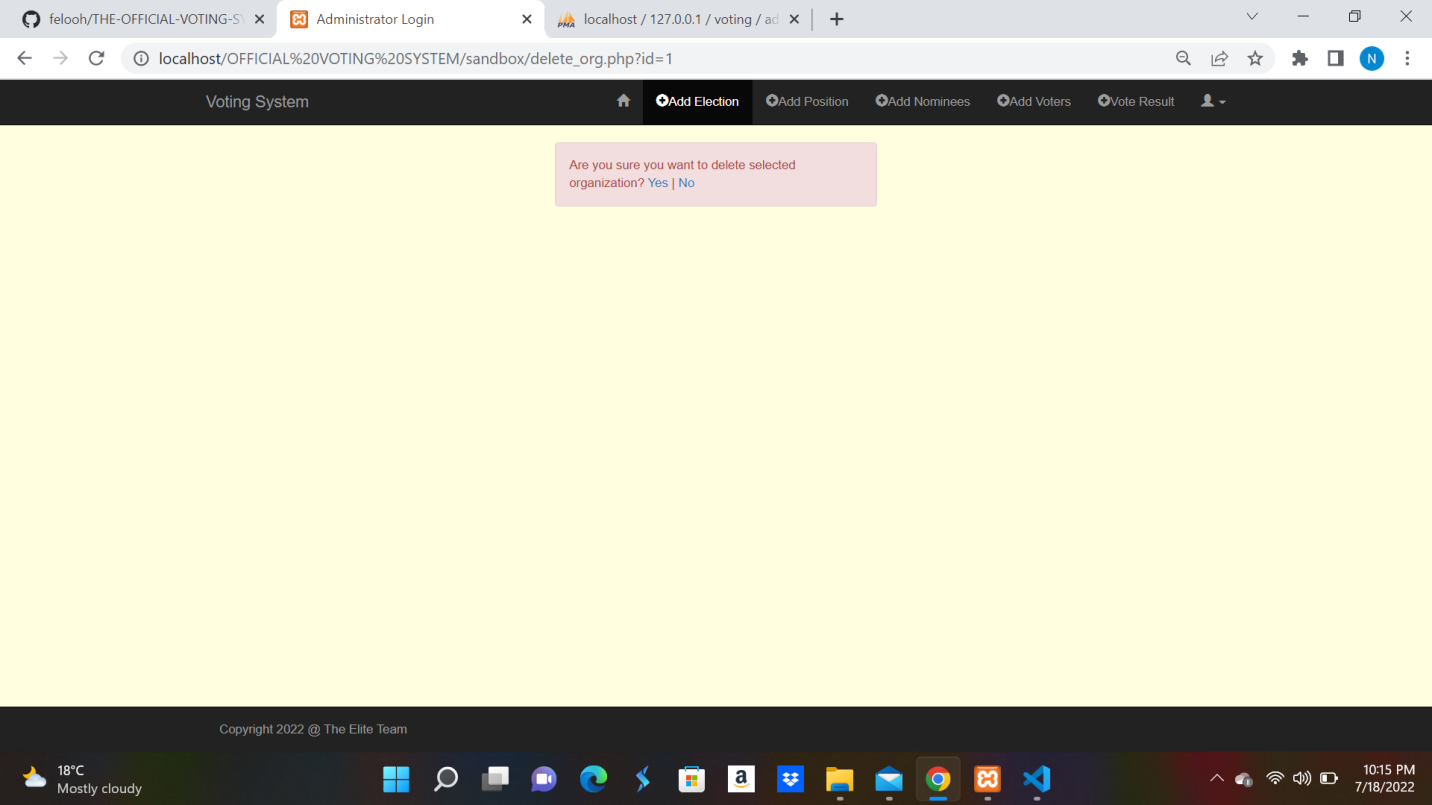
### 4.2.1 VOTE RESULT PAGE



### 4.2.2 VOTER SELECTS NOMINEE PAGE



### 4.2.3 DELETE AN ELECTION

****

1. DETAILED DESIGNS

This section provides the information needed for a system development team to actually build and integrate the hardware components, code and integrate the software modules, and interconnect the hardware and software segments into a functional product.

## 5.1 Hardware Detailed Design

The online voting system is web based therefore, only requires a few hardware components to ensure that it is useful to its clients. It requires a computer, either personal, desktop or hand devices such as mobile phones so that it can be used.

The personal computer used to access the system requires that each have a browser, which can either be Mozilla, Google Chrome, internet explorer and safari as explained in the system software requirement specification.

It also requires a server where the user data can be stored for easy accessibility over the internet. This is a high-speed server that can handle multiple requests from different users at the same time. The server should have a minimum of the following details:

1. 16 GB RAM (Random Access Memory)
2. 2TB hard disk to cater for the user information.
3. Continuous power supply to ensure that services are available all through.

## 5.2 Software Detailed Design

The system will contain several modules to ensure its full functionality as expected. These modules and their functionality are as follows:

1. User Registration

This module handles user registration to the system. The system admin registers all the users into the system so that they can be able to access the system using the accounts created for them. This is to ensure that the admins assign each user the

tasks they should do and what they are allowed to access.

1. User Login

This module facilitates authentication of users before they can access the system. Only the users who log in successfully are able to use the system. The system supports two kinds of users, that is, the voter and the Administrator. The user logs in to the system using their registration number whereas the administrator requires the amin username and password to log in.

1. Add Election

The module is an Admin only accessible module that facilitates the addition of an election in to a system and also removal of the elections. This module has been designed like that to enable the reuse of the system to conduct different elections in the organization starting from big to small elections.

1. Add Positions

Adding positions is a module under the add election module that allows administrators to add elective positions of an election. The position added can be removed still under the same module.

1. Add Candidates

Candidates participating in an election are added to the system by the admin through this module. The list of those who meet the qualifications of a certain position are manually validated by the administrator and then added in to the system manually. They can be revoked through the same module.

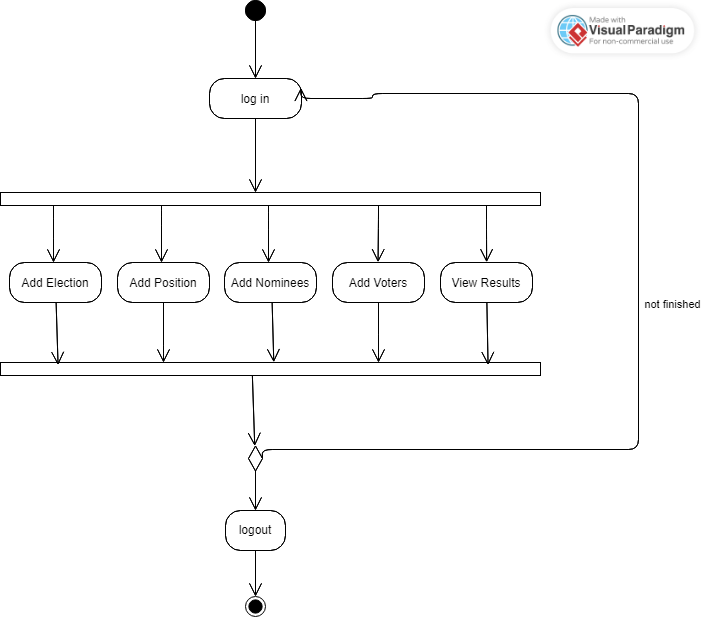
1. View Results

This module is designed to be view only even by the ADMIN to ensure the results integrity. The results can be viewed via a window on the screen, printed or saved as a PDF file.

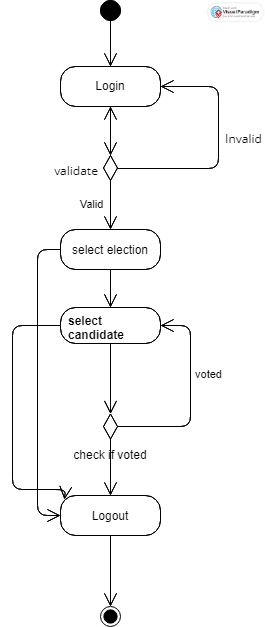
### 5.2.1 Activity Diagrams

The activity diagrams below represent how different actors will be interacting with the system, indicating the flow of the information within the online voting system as they perform their activities.

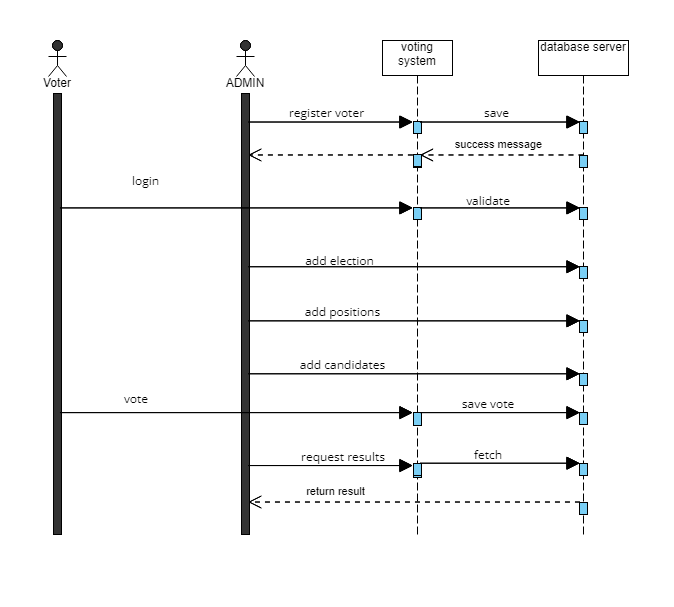
#### 5.2.1.1 ADMIN ACTIVITY DIAGRAM



#### 5.2.1.1 VOTER ACTIVITY DIAGRAM



#### 5.2.1.2 Sequence Diagram

****

## 5.3 Internal Communications Detailed Designs

The communication process of the online voting system will be achieved by internet which will enable communication between the system and the database during user data storage and retrieval of the required data by the user and communication between the different modules of the system. A web browser will be used as an interface in order to make requests to the web server and also renders the response back the user for the requested resource. HTTP (Hyper Text Transfer Protocol) is used to ensure communication between the web server and the web browser.

1. EXTERNAL INTERFACES

External systems are any systems that are not within the scope of the system under development, regardless whether the other systems are managed by the State or another agency. This section describes the electronic interface(s) between this system and each of the other systems and/or subsystem(s), emphasizing the point of view of the system being developed.

## 6.1 Internal Architecture

The online voting system interfaces with google mailing API, which will enable voters to communicate with ADMIN and also facilitate communication from admin to the voters. It will also interface with the database servers in which they will be connected through the of internet. The database server will be used for storing the data that will be used within the online voting system. The communications occur through the use of HTTP protocol.

## 6.2 Interface Detailed Design

The data formats to be saved in to the database server include, text, numbers, date and characters. The data need not to be reformatted before storage.

If there are errors in accessing the database server, the user will be notified in a user-friendly message and informed that the system is down.

1. SYSTEM INTEGRITY CONTROLS

In order to ensure a strong system base security, the proposed system will implement data encryption before any piece of critical and confidential information or data is accessed from the database. Successful protection of data will ensure system integrity which will aid in the usability of the system by the end users.

The following security controls will be implemented as part of the system in order to ensure its integrity.

1. **Authentication** – Each user will be authenticated by use of log in details which includes the registration number before being allowed to use the system.
2. **Authorization of users** – The system will categorize users into various categories of access a sign of privileges so as to be able to obtain data, processes and operations they have been allowed.
3. **Validation** – This ensures that the data entered in the system is of the right type so that it does not interfere with the integrity of the data being accessed.