DESIGN AND IMPLEMENTATION OF AN E-VOTING SYSTEM (CASE STUDY NACOSS, FEDERAL POLYTECHNIC, MUBI)

BY

SUNDAY PHILIP
(ST/CS/HND/21/054)

DEPARTMENT OF COMPUTER SCIENCE,
SCHOOL OF SCIENCE AND TECHNOLOGY,
FEDERAL POLYTECHNIC, MUBI, ADAMAWA STATE.

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF HIGHER NATIONAL DIPLOMA (HND) IN COMPUTER SCIENCE.

SEPTEMBER, 2023

DECLARATION

I hereby declare that the work in this project titled **"Design and Implementation of an E-Voting System (Case Study NACOSS, Federal Polytechnic, Mubi)"** was performed by me under the supervision of Mal. Yayirus Garba. The information derived from literatures has been duly acknowledged in the text and a list of references provided. The work embodied in this project is original and had not been submitted in part or in full for any other diploma or certificate of this or any other institution.

SUNDAY PHILIP	_____	_____
(ST/CS/HND/21/054)	Signature	Date

CERTIFICATION

This project titled **"Design and Implementation of an E-Voting System (Case Study NACOSS, Federal Polytechnic, Mubi)"** meets the regulations governing the award of Higher National Diploma (HND) in Computer Science, Federal Polytechnic Mubi, Adamawa State

Mal. Yayirus Garba _____

(Project Supervisor) Sign/Date

Mr. Kassim Mustapha _____

(Head of Department) Sign/Date

Mal. Abdulrahaman Saidu _____

(External Examiner) Sign/Date

DEDICATION

This project is dedicated to my beloved parents for their advice, encouragement and financial support towards my academic pursuit.

ACKNOWLEDGEMENTS

I want to acknowledge Almighty God for his infinite mercy and protection throughout my academic activities. And for the understanding in achieving our academic success.

I also recognize my Supervisor Mal. Yayirus Garba who took time, despite his busy schedule to direct and guide me throughout this research work.

I also acknowledge the Head of Department Computer Science Mal. Kassim Mustapha for his moral encouragement throughout my period of study. I also acknowledge all Staff of Computer Science Department for their support and encouragement and the knowledge they've impacted on me throughout my studies.

I also want to appreciate my lovely parents for their love and care and for giving me the opportunity to be trained and achieve my dreams.

Finally, I appreciate the efforts of my Uncles and aunties, for their encouragement and support throughout the course of my study and also my friends and relatives, course mates and all well-wishers. I love you all, may the Almighty God bless you abundantly, Amen.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In the era of advancing technology, the need for efficient, secure, and transparent electoral processes within student organizations has become increasingly imperative. This research project presents the design and successful implementation of an E-Voting System tailored for the National Association of Computer Science Students (NACOSS) at the Federal Polytechnic, Mubi. The primary objective was to modernize the electoral procedures, alleviate administrative burdens, and foster trust in the electoral process among members. The E-Voting System encompasses various components, including a user-friendly interface for voters and administrators, voter and candidate registration modules, secure ballot casting, and real-time results tabulation. The system was rigorously tested and validated to ensure its reliability and security. The findings of this project underscore the advantages of adopting electronic voting systems, including increased efficiency, enhanced security, transparency, and accessibility. Moreover, it serves as a case study and practical demonstration for other student organizations and institutions seeking to embrace technology-driven electoral processes. As part of the project's contribution to knowledge, it emphasizes the significance of robust security measures and transparency in electronic voting systems, paving the way for improved electoral processes in various organizational contexts. Additionally, recommendations are provided to guide the successful adoption and continued improvement of the E-Voting System. The successful implementation of this E-Voting System showcases the transformative potential of technology in shaping the future of student organization elections, setting a precedent for more efficient, secure, and inclusive democratic processes.

.

CHAPTER ONE
INTRODUCTION

1.1 Background to the Study

In the modern era, technology has permeated almost every aspect of human life, including governance and decision-making processes. Voting is a fundamental democratic process that allows students to participate in the selection of their representatives and the decision-making process. Traditionally, voting has been conducted using paper-based systems, which can be time-consuming, prone to errors, and vulnerable to manipulation. Voting is a cornerstone of democratic societies, providing citizens with the opportunity to participate in decision-making processes and elect their representatives. However, traditional paper-based voting systems have been plagued with numerous challenges, including inefficiency, errors, and vulnerabilities to fraud. As technology continues to advance, electronic voting (e-voting) systems have emerged as a potential solution to address these issues and improve the overall voting experience (Kreischer & Krimmer, 2021).

E-voting systems leverage technology to facilitate the casting and counting of votes electronically, offering advantages such as increased efficiency, accuracy, and transparency. These systems can streamline the voting process by automating various tasks, including vote casting, counting, and result announcement. Moreover, e-voting systems can enhance the accuracy of the voting process by minimizing human errors and eliminating the need for manual vote counting. Furthermore, they can provide transparency by enabling voters and stakeholders to verify and audit the voting results (Magkos & Mentzas, 2021).

Recent studies have emphasized the importance of implementing e-voting systems to enhance the integrity and efficiency of democratic processes. For instance, a research paper by Hanandeh *et al.* (2021) highlighted the benefits of e-voting systems, including reduced voting time, increased accessibility, and improved accuracy. The study also emphasized the need for robust security measures to protect against potential threats and ensure the credibility of the e-voting system.

In the context of educational institutions, student organizations often conduct elections to select their representatives and leaders. The National Association of Computer Science Students (NACOSS) at the Federal Polytechnic, Mubi, is one such organization that recognizes the limitations of the traditional paper-based voting system. The existing system poses challenges such as time-consuming vote counting, difficulties in verifying votes, and the potential for fraudulent activities.

To address these challenges and provide a more efficient and transparent voting process, the design and implementation of an e-voting system tailored to the specific requirements of NACOSS at the

Federal Polytechnic, Mubi, is essential. By adopting an e-voting system, NACOSS can streamline the voting process, improve accuracy, and enhance transparency.

1.2 Problem Statement

The NACOSS (National Association of Computer Science Students) at the Federal Polytechnic, Mubi, recognizes the need to modernize their voting system. The existing paper-based voting system used by NACOSS at the Federal Polytechnic, Mubi, suffers from several limitations and challenges. These include:

i. Time-consuming processes: The manual counting of votes and the compilation of results can be a time-consuming task, causing delays in announcing the final results.
ii. Error-prone: Manual data entry and vote counting processes are susceptible to human errors, which can compromise the accuracy and reliability of the election results.
iii. Verification challenges: With a paper-based system, it can be challenging to verify the authenticity and integrity of votes, leading to doubts and disputes regarding the election outcomes.
iv. Potential for fraud: Paper-based systems are vulnerable to various fraudulent activities, such as ballot stuffing or tampering with ballot boxes, which can undermine the credibility and fairness of the elections.
v. One person can vote more than one time.

1.3 Aim and Objectives

The aim of this project is to design and implementation of an e-voting system for NACOSS, Federal Polytechnic, Mubi. The specific objectives of this study are as follows:

i. To design an e-voting system tailored to the specific requirements of NACOSS, Federal Polytechnic, Mubi.
ii. To implement the e-voting system and ensure its usability, reliability, and security.
iii. To evaluate the effectiveness of the e-voting system in terms of efficiency, accuracy, and transparency.
iv. To control over voting.

1.4 Significance of the Study

The successful design and implementation of an e-voting system for NACOSS, Federal Polytechnic, Mubi, will bring several benefits. Firstly, it will streamline the voting process, saving time and effort for both the voters and the organizers. Secondly, it will enhance the accuracy of the voting system by minimizing errors and eliminating manual counting procedures. Thirdly, the system will provide transparency, allowing for easy verification and auditing of votes, thereby increasing trust in the

electoral process. Finally, the implementation of an e-voting system will serve as a model for other institutions and organizations seeking to modernize their voting systems.

1.5 Scope of the Study

This study focuses specifically on the design and implementation of an e-voting system for NACOSS at the Federal Polytechnic, Mubi. The system will be tailored to meet the unique requirements and constraints of the organization. However, it is important to acknowledge that the successful adoption of the system relies on various factors, including infrastructure availability, user acceptance, and cybersecurity measures. Therefore, this study does not address these external factors but primarily focuses on the technical aspects of the e-voting system.

1.6 Definition of Some Operational Terms

Database: In the context of an e-voting system, the database stores and manages various information, including voter details, candidate profiles, voting transactions, and other relevant data necessary for the voting process (Magkos & Mentzas, 2021).

Electronic Voting (E-Voting): Electronic voting, commonly referred to as e-voting, is a voting method that employs electronic systems and technologies to facilitate the casting and counting of votes (Magkos & Mentzas, 2021).

Result: The result refers to the outcome of an election, indicating the final tally of votes for each candidate or option. In the context of an e-voting system, the result is determined by aggregating and analyzing the recorded votes from the database (Kreischer & Krimmer, 2021).

Security: Security in the context of e-voting systems refers to the measures and protocols implemented to protect the integrity, confidentiality, and availability of the voting process and data (Jain & Jain, 2021).

System: In the context of this study, a system refers to the e-voting system designed and implemented for NACOSS at the Federal Polytechnic, Mubi. It encompasses the hardware, software, and associated components that enable the electronic casting, storage, and processing of votes (Kreischer & Krimmer, 2021).

Transparency: Transparency in e-voting systems refers to the openness and visibility of the voting process and the ability to verify and audit the election results (Kreischer & Krimmer, 2021).

Usability: Usability refers to the extent to which a system or software is user-friendly, intuitive, and easy to use (Schürmann & Krimmer, 2021).

Votes: In an e-voting system, votes are cast electronically by the voters through the designated interfaces or platforms provided by the system (Jain & Jain, 2021).

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

This chapter provides a comprehensive review of the existing literature related to e-voting systems. It examines studies, research articles, and relevant publications that contribute to the understanding of e-voting technologies, their implementation, challenges, and best practices. The literature review serves as a foundation for the design and implementation of an e-voting system for NACOSS at the Federal Polytechnic, Mubi.

2.2 Overview of E-Voting Systems

2.2.1 Definition and Components of E-Voting Systems

E-voting systems refer to the use of electronic technologies and systems to facilitate the casting, storage, and processing of votes in an election. These systems consist of various components, including hardware, software, databases, and user interfaces. The hardware components can include voting machines, ballot scanners, and biometric devices, while the software components encompass the applications and algorithms that manage the voting process (Kreischer & Krimmer, 2021).

E-voting systems consist of several essential components:

i. Hardware: E-voting systems require hardware components such as electronic voting machines, ballot scanners, biometric devices, and other peripherals. These devices are used by voters to interact with the system and cast their votes (Kreischer & Krimmer, 2021).

ii. Software: The software component of an e-voting system includes the applications, algorithms, and protocols that facilitate the entire voting process. This encompasses the user interfaces for casting votes, data encryption and decryption mechanisms, database management systems, and result tabulation algorithms (Klemenc & Volk, 2021).

iii. Database: A database is a crucial component of an e-voting system. It serves as a central repository for storing and managing various data related to the election, including voter registration details, candidate profiles, and voting transactions. The database ensures the integrity and security of the stored information (Magkos & Mentzas, 2021).

iv. User Interfaces: E-voting systems offer different user interfaces to facilitate vote casting. These interfaces can be in the form of touch screens on voting machines, web-based interfaces for internet voting, or mobile applications for mobile voting. User interfaces need to be intuitive, user-friendly, and accessible to voters with varying levels of technical proficiency (Schürmann & Krimmer, 2021).

2.2.2 Types of E-Voting Systems

E-voting systems can be categorized into different types based on their implementation methods:

i. Machine-based voting systems: This type of e-voting system utilizes dedicated electronic voting machines located at polling stations. Voters interact with these machines to cast their votes. Machine-based systems may utilize touch screens, buttons, or other input mechanisms for vote selection (Klemenc & Volk, 2021).

ii. Internet-based voting systems: Internet voting systems allow eligible voters to cast their votes remotely using secure online platforms. Voters can access the system through their personal devices, such as computers or smartphones, and cast their votes from any location with an internet connection (Kreischer & Krimmer, 2021).

iii. Mobile-based voting systems: Mobile voting systems leverage mobile applications to enable voters to cast their votes using their smartphones or other mobile devices. These systems often employ encryption and secure communication protocols to ensure the confidentiality and integrity of votes (Jain & Jain, 2021).

2.3 Information system

Information systems are starting to become indistinguishable from communication systems. Some of the many different ways that will change healthcare are the way communication occurs and to the change it will have upon the doctor-patient relationship. For optimum performance of telemedicine, three aspects of the management of the medical records for teleconsultations are particularly important: multimedia collection, standardization of patient record identification and classification, and information management (Lian, Chong, Zhai & Ning, 2013).

2.4 Web-portal

A web portal is a specially designed website that brings information from diverse sources, like emails, online forums and search engines, together in a uniform way. Usually, each information source gets its dedicated area on the page for displaying information (a portlet); often, the user can configure which ones to display. The extent to which content is displayed in a "uniform way" may depend on the intended user and the intended purpose, as well as the diversity of the content. Very often design emphasis is on a certain "metaphor" for configuring and customizing the presentation of the content (e.g., a dashboard or map) and the chosen implementation framework or code libraries (Richard, 2016).

In addition, the role of the user in an organization may determine which content can be added to the portal or deleted from the portal configuration. A portal may use a search engine's Application Programming Interface (API) to permit users to search intranet content as opposed to extranet content by restricting which domains may be searched. Apart from this common search engines

feature, web portals may offer other services such as e-mail, news, stock quotes, information from databases and even entertainment content. Portals provide a way for enterprises and organizations to provide a consistent "look and feel" with access control and procedures for multiple applications and databases, which otherwise would have been different web entities at various URLs. The features available may be restricted by whether access is by an authorized and authenticated user (employee, member) or an anonymous website visitor (Navjot & Supriti, 2019).

Raenu and Rafidah (2013), point out that portal is a Web site which acts as a starting point or 'gateway' and provides a wide variety of resources, services, tasks and links to other websites. Among those resources there are search engines, news, e-mail, discussion groups, online shopping, references and so on. This type of portals, sometimes called horizontal portals (Babie, 2014), is generally offered by Internet Service Providers or search engines. Yahoo! is an example, with an index to a lot of services, that is, the first screen that a user will see when going online, a place to go to find an organized view of the online information space.

More specialized portals, sometimes called vertical portals (Rao, 2013), are those addressed to a specific interest or field, for example portals with the aim at medical information. The, users can get information about clinical trials, professional directories, patient forums, support groups, health articles, health care associations, and so on. Even more specialized portals, enterprise portals deliver organization wide information in a user centric manner, based on user authentication they offer customized services to specific users, employees, customers, and the like. They offer support for tasks, workflow, groupware, and the creation and integration of knowledge. In this last category, we can find, for example, the employee portal of an institution. The, employees, in general, can access their salaries, information about their medical insurances, and the like, and, more specifically, research staff can access a service to complete their curriculum vitae, forms to request financial support for research, and so on. Personal portals are also distinguished. They are customized by the user and typically are associated with a search engine and display selected information such as news, weather, dictionaries and so on. Google and My Yahoo are examples of this type of portals.

2.5 Database Management System

2.5.1 Definition of Database Management System (DBMS)

A Database Management System (DBMS) is software that enables the creation, organization, management, and retrieval of data in a structured manner. It provides a systematic approach to store, update, and retrieve data from a database, ensuring data integrity, security, and efficient access. DBMS serves as an essential component of e-voting systems, facilitating the storage and management of various data related to the election process (Elmasri & Navathe, 2020).

2.5.2 Components of a Database Management System

A DBMS consists of several components that work together to manage the database effectively:

Data Definition Language (DDL): DDL is used to define and specify the database structure and schema. It allows the creation, modification, and deletion of database objects such as tables, views, and indexes.

Data Manipulation Language (DML): DML enables users to interact with the database by performing operations such as inserting, updating, and deleting data records. It provides a set of commands and operations to retrieve and manipulate data stored in the database.

Data Query Language (DQL): DQL is used to retrieve specific data from the database based on user-defined queries. It allows users to specify search criteria, sort data, and extract information from the database.

Data Control Language (DCL): DCL defines and manages access control and security mechanisms for the database. It includes commands to grant or revoke permissions, define user roles, and enforce data security policies.

Database Transaction Management: DBMS provides transaction management capabilities to ensure the integrity and consistency of data. It supports transactional operations, including transaction initiation, commit, rollback, and concurrency control to manage concurrent access to the database.

Database Recovery and Backup: DBMS includes mechanisms for data recovery and backup to protect against data loss or system failures. It allows for the restoration of the database to a previous consistent state in case of failures.

2.5.3 Importance of Database Management in E-Voting Systems

In the context of e-voting systems, the database management system plays a critical role in storing and managing various data related to the election process. It ensures the integrity and security of the stored information and provides efficient access to retrieve and process the data. Key aspects of the database management system in e-voting systems include:

Voter Registration: DBMS manages the voter registration process, storing voter details such as names, identification numbers, and eligibility status. It allows for efficient verification of voter credentials during the voting process.

Candidate Profiles: DBMS stores information about the candidates participating in the election, including their names, party affiliations, and other relevant details. This data is retrieved and presented to the voters during the voting process.

Vote Recording: DBMS records and stores the votes cast by each voter, ensuring accuracy and integrity. It allows for the retrieval and aggregation of votes for result tabulation.

Auditability: DBMS provides mechanisms for auditing and maintaining a trail of all activities related to the database. It enables the tracking of changes, ensuring transparency and accountability in the e-voting process.

Data Security: DBMS incorporates security measures to protect the database from unauthorized access, tampering, or data breaches. It includes encryption techniques, access control mechanisms, and data backup strategies to ensure data confidentiality and integrity.

2.6 Related Literatures

Al-Nashif and Milutinovic (2020), proposed an e-voting system based on blockchain technology, which aims to enhance the security, transparency, and integrity of the voting process. It discusses the design and implementation of the system and evaluates its effectiveness in ensuring secure and tamper-resistant voting.

Dhamija and Dhamija (2020) in their study presents an e-voting system that employs a homomorphic encryption scheme to ensure the privacy and integrity of votes. It discusses the implementation of the system and evaluates its performance and security properties, highlighting the advantages of using homomorphic encryption in e-voting systems.

Garcia *et al*. (2021), proposed an e-voting system that utilizes biometric authentication for secure and accurate voter identification. It discusses the design principles, implementation details, and evaluation of the system, emphasizing the advantages of biometric-based authentication in preventing voter fraud and ensuring the integrity of the voting process.

Osorio (2020), presents an e-voting system that combines blockchain technology and blind signature techniques to enhance the security and anonymity of votes. It discusses the design and implementation of the system and evaluates its effectiveness in ensuring transparent and tamper-resistant voting.

Trabelsi *et al*. (2020), proposes a secure e-voting system that leverages the Internet of Things (IoT) and blockchain technology. It discusses the integration of IoT devices for voter authentication and the use of blockchain for secure vote recording and verification. The study evaluates the system's security properties and highlights the potential of IoT and blockchain in e-voting systems.

These related studies offer insights into various approaches and technologies used in e-voting systems, including blockchain, encryption schemes, biometric authentication, and IoT. They provide valuable contributions to the design, implementation, and evaluation of secure, transparent, and efficient e-voting systems.

2.7 Summary

This chapter has provided a comprehensive literature review on e-voting systems, including their definition, components, types, benefits, and challenges. It has also highlighted best practices in system design and implementation and presented case studies of successful e-voting implementations in various countries. The findings from this literature review will inform the design and implementation of an e-voting system tailored to the specific requirements of NACOSS at the Federal Polytechnic, Mubi.

CHAPTER THREE

SYSTEM DESIGN AND ANALYSIS

3.1 Introduction

This chapter presents the design and implementation details of the Electronic Voting (E-Voting) system for the National Association of Computer Science Students (NACOSS) at Federal Polytechnic, Mubi. This chapter contains the system design and analysis of the proposed system, the disadvantages of the existing system, the advantages of the proposed system over the existing system, the requirements (Hardware and Software), the design and the system architecture.

3.2 Disadvantages of the Existing System

The existing system of manual voting used by NACOSS at Federal Polytechnic, Mubi has several disadvantages. Manual voting processes are often time-consuming, particularly during large-scale elections. Voters need to physically travel to the polling stations, wait in queues, and manually mark their ballots. This process can result in long waiting times and delays in vote counting. Manual voting is susceptible to human errors. Both voters and electoral staff may make mistakes while marking the ballots or recording the votes. These errors can lead to inaccurate vote counts and potentially impact the election results. The existing system of voting poses challenges for individuals with disabilities or those who are unable to travel to the polling stations. Manual voting systems are vulnerable to security risks such as ballot tampering, fraudulent activities, or misplacement of ballots. Without robust security measures, it becomes difficult to ensure the integrity and confidentiality of the voting process. Manual voting processes may lack transparency, making it challenging for stakeholders to verify the fairness and accuracy of the election results. Without a clear audit trail, it becomes challenging to track and resolve any discrepancies or disputes that may arise.

3.3 Advantages of the Proposed System

The proposed E-Voting system for NACOSS at Federal Polytechnic, Mubi offers several advantages over the existing manual voting system. Some of the key advantages of the proposed system are:

i. Increased Efficiency
ii. Enhanced Accuracy
iii. Improved Accessibility
iv. Enhanced Security
v. Improved Transparency
vi. Cost-effective
vii. Efficient Data Management

3.4 The Proposed Method

The waterfall model is a traditional sequential approach to software development that consists of distinct phases that follow a linear sequence. Here is a simplified version of the waterfall model for the development of an E-Voting system for NACOSS at Federal Polytechnic, Mubi:

Requirements Gathering and Analysis:

i. Identify the requirements and objectives of the E-Voting system.
ii. Conduct interviews and discussions with stakeholders to understand their needs.
iii. Define the system's functionalities, user roles, and security requirements.

System Design:

i. Design the system architecture, including the client-side and server-side components.
ii. Create the database schema and define the data model.
iii. Develop the user interface design, considering usability and accessibility.

Implementation:

i. Develop the client-side application using web technologies like HTML, CSS, and JavaScript.
ii. Implement the server-side application using a suitable programming language and framework.
iii. Integrate the user interface with the backend functionalities.
iv. Implement security measures such as encryption, authentication protocols, and access control.

Testing:

i. Conduct unit testing to verify the correctness of individual components.
ii. Perform integration testing to ensure the proper functioning of the system as a whole.
iii. Carry out system testing to validate the system against the defined requirements.
iv. Perform security testing to identify and address any vulnerabilities.

Deployment:

i. Prepare the system for deployment by configuring the necessary infrastructure and servers.
ii. Install and set up the required software and dependencies.
iii. Migrate the database and ensure data integrity.
iv. Conduct user acceptance testing to gain feedback and ensure readiness for production use.

Maintenance and Support:

i. Provide ongoing maintenance and support for the E-Voting system.
ii. Address any reported issues, bugs, or security vulnerabilities.
iii. Perform regular system updates and enhancements based on user feedback and changing requirements.
iv. Ensure the system remains secure, reliable, and up-to-date.

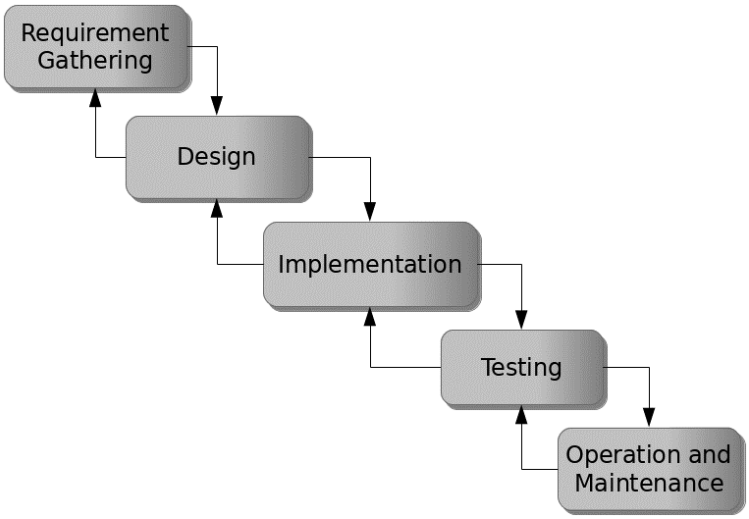


Figure 3.1: Waterfall model

3.5 Method of Data Collection

This study will adopt two methods of data collection:

Primary Source: Primary source refers to the sources of collecting original data in which the researcher makes use of empirical approach such as personal interview, questionnaires or observation.

Secondary Source: The need for the secondary sources of data for this kind of project cannot be over emphasized. The secondary data were obtained from magazines, Journal, newspapers, library source and most of the information from the library research has been covered in the literature review section.

3.6 System Design

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

3.6.1 Algorithm Diagram

Use case diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram shows the system and the various ways that they interact with the system.

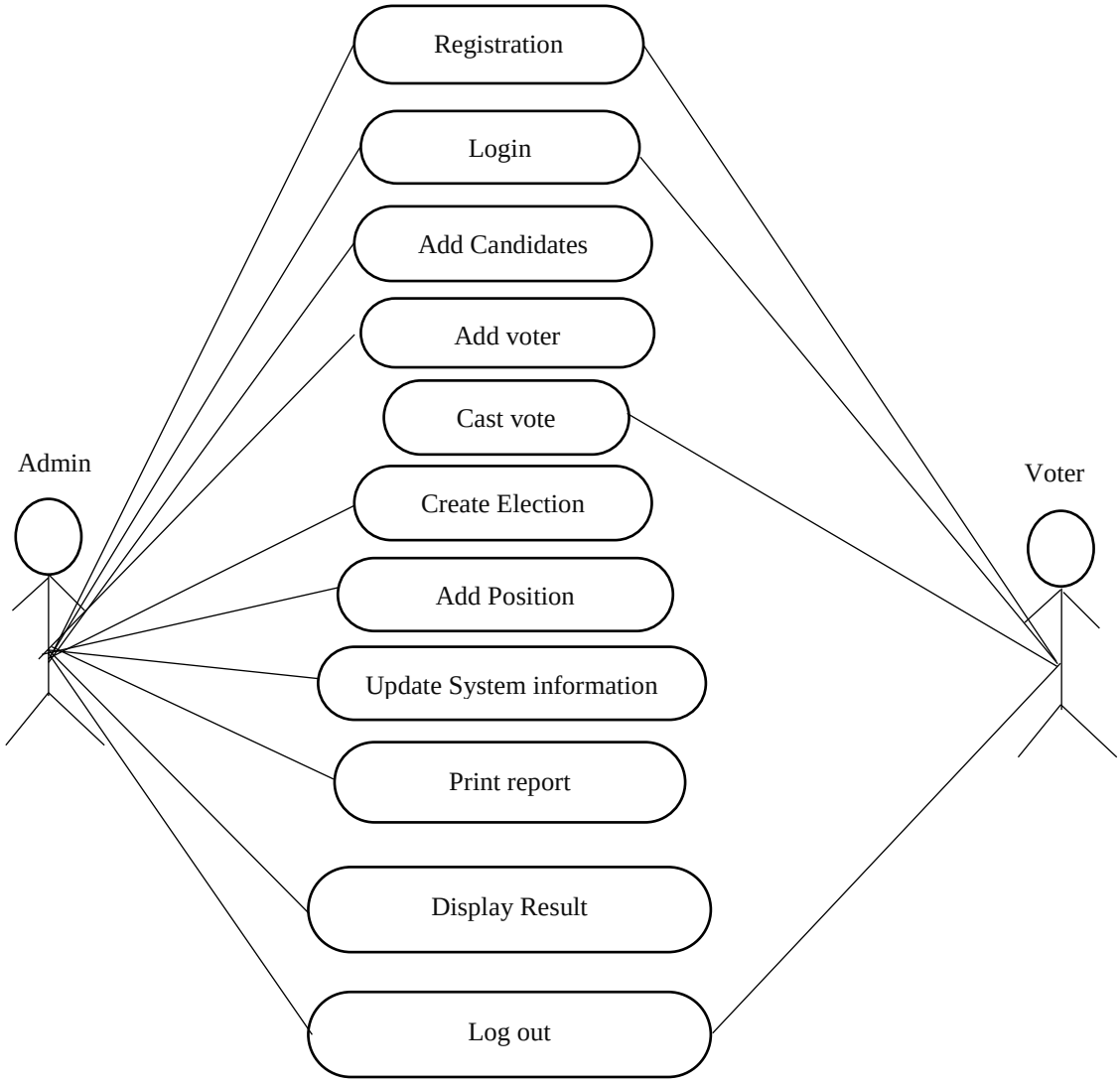


Figure 3.2: Use case diagram

Activity Diagram

An activity diagram shows a flow of control in a system similar to a flowchart or a data flow diagram.

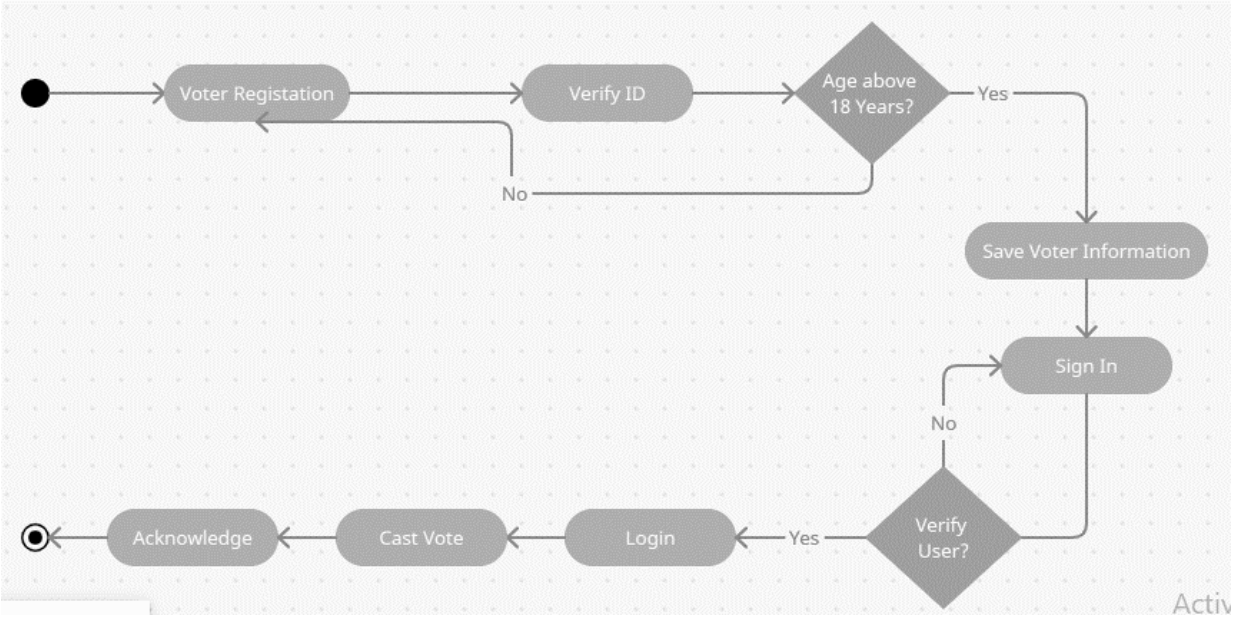


Figure 3.3: Activity Diagram for E-voting system

3.6.2 System Architecture

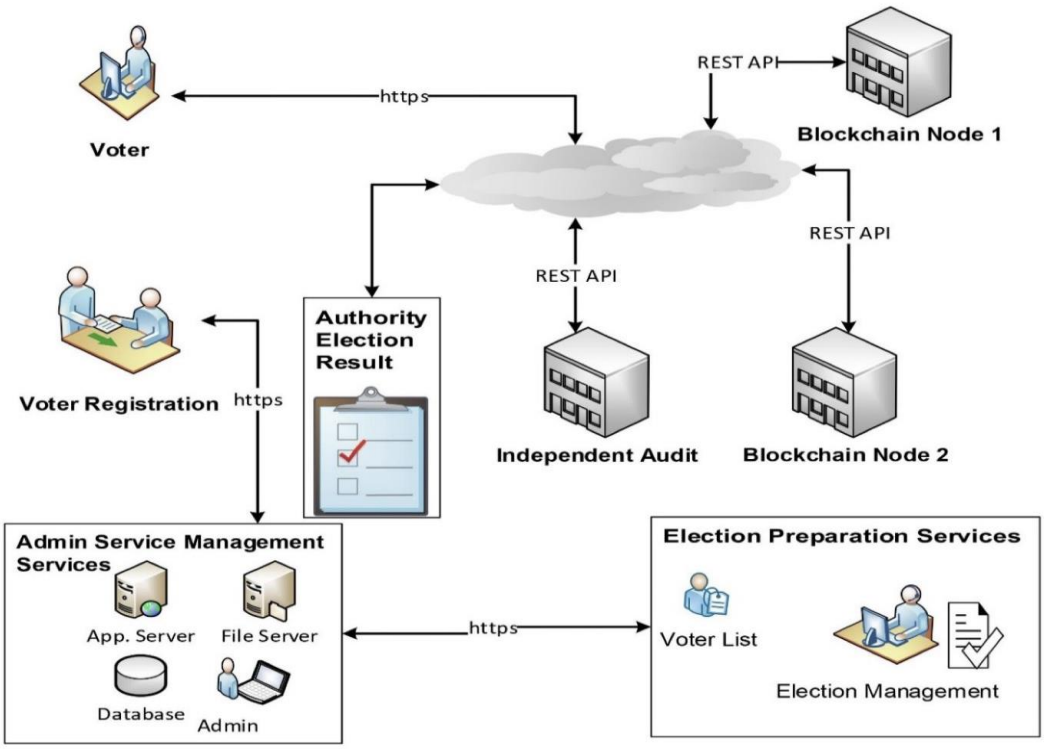


Figure 3.4: System architecture

3.6.3 Database Tables/Queries Structures

The database is used to store all information that pertain the food ordering records. Below are the database table for the new system.

Table 1: Admin Table

Name	Type	Extra
id	int(11)	AUTO_INCREMENT
username	varchar(250)	
firstname	varchar(250)	
lastname	varchar(250)	
photo	varchar(250)	
password	varchar(250)	
Created_on	Time_stamp	

Table 2: Candidates table

Name	Type	Extra
id	int(11)	AUTO_INCREMENT
Position_id	int(11)	
firstname	varchar(250)	
lastname	varchar(250)	
photo	varchar(250)	
association	varchar(250)	
Created_on	Time_stamp	

Table 3: Position Table

Name	Type	Extra
id	int(11)	AUTO_INCREMENT
description	varchar(250)	
Max_vote	varchar(250)	
priority	varchar(250)	

Table 4: Votes table

Name	Type	Extra
id	int(11)	AUTO_INCREMENT
Position_id	int(11)	
Candidate_id	int(11)	
Voters_id	int(11)	

Table 5: Voters Table

Name	Type	Extra
id	int(11)	AUTO_INCREMENT
voters_id	int(11)	
firstname	varchar(250)	
lastname	varchar(250)	
photo	varchar(250)	
level	varchar(250)	
Created_on	Time_stamp	

3.6.4 Entity Relationship Modelling

An Entity Relationship (ER) Diagram is a sort of flowchart that shows how "entities" in a system, such as people, things, or concepts, interact with one another. ER Diagrams are most commonly used in the disciplines of software engineering, corporate information systems, education, and research to build or troubleshoot relational databases.

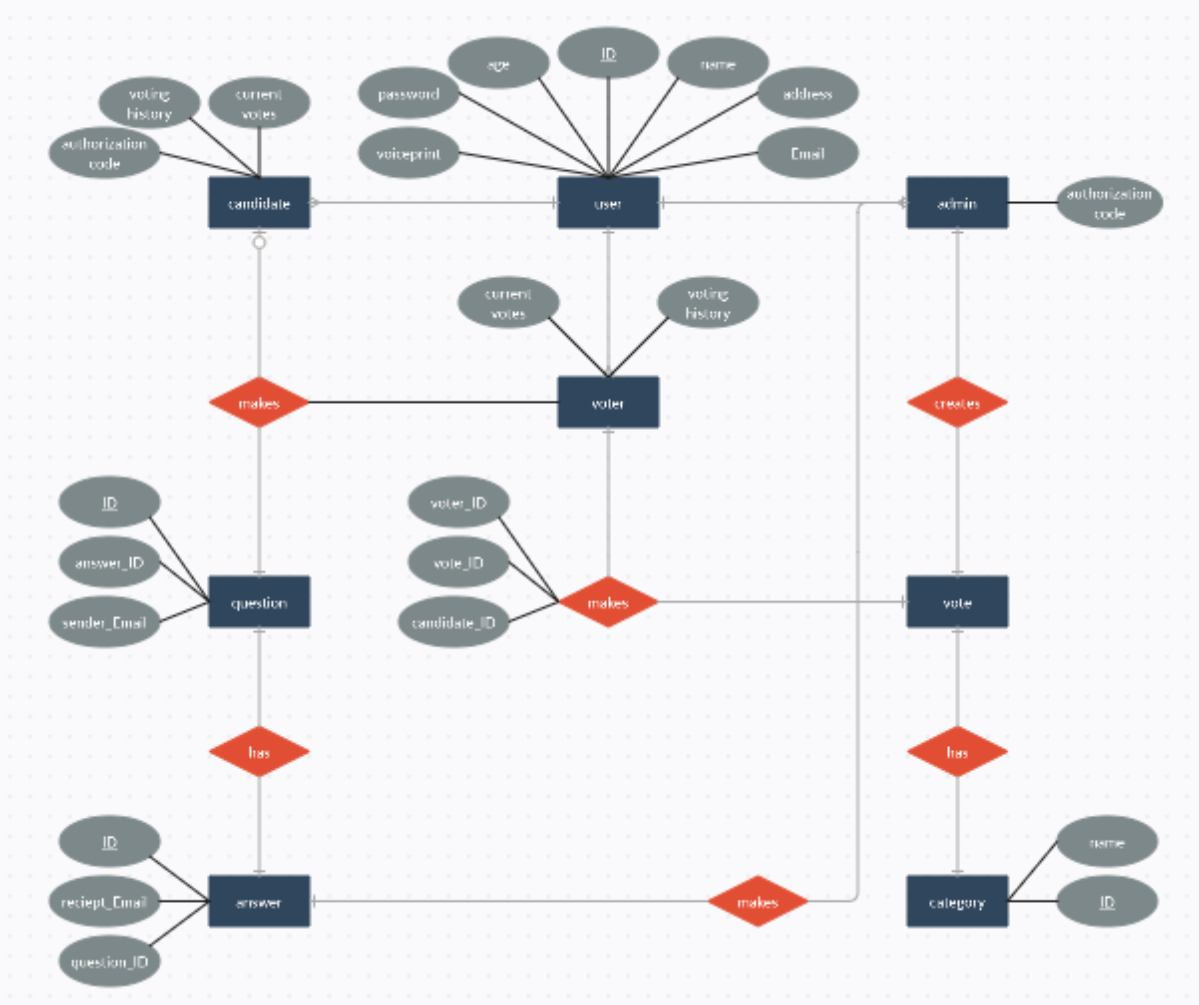


Figure 3.5: Entity Relationship Model

3.6.5 Database Entity Relationship Diagram

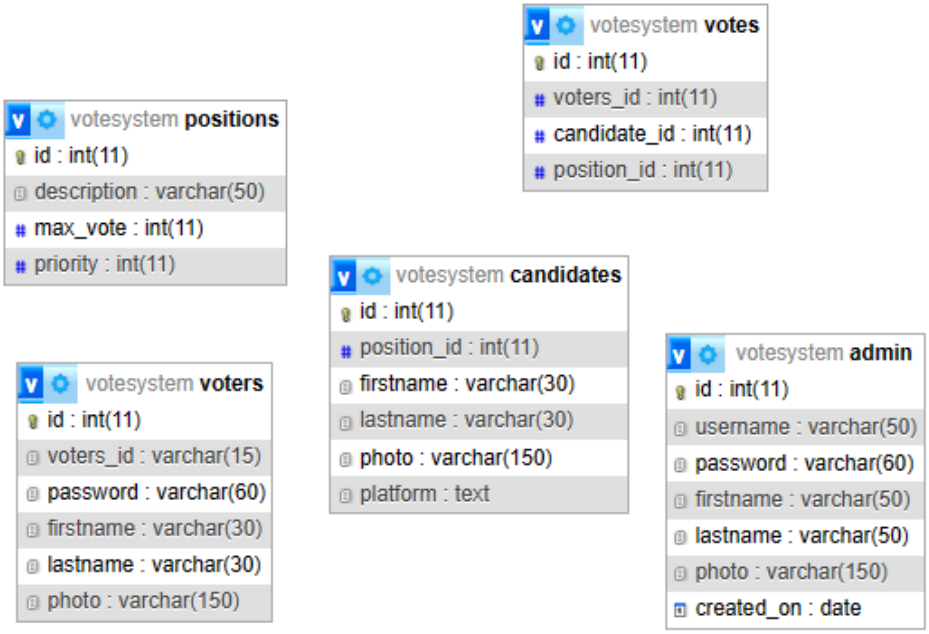


Figure 3.4: Entity Relationship Diagram

3.6.6 Input and Output Design

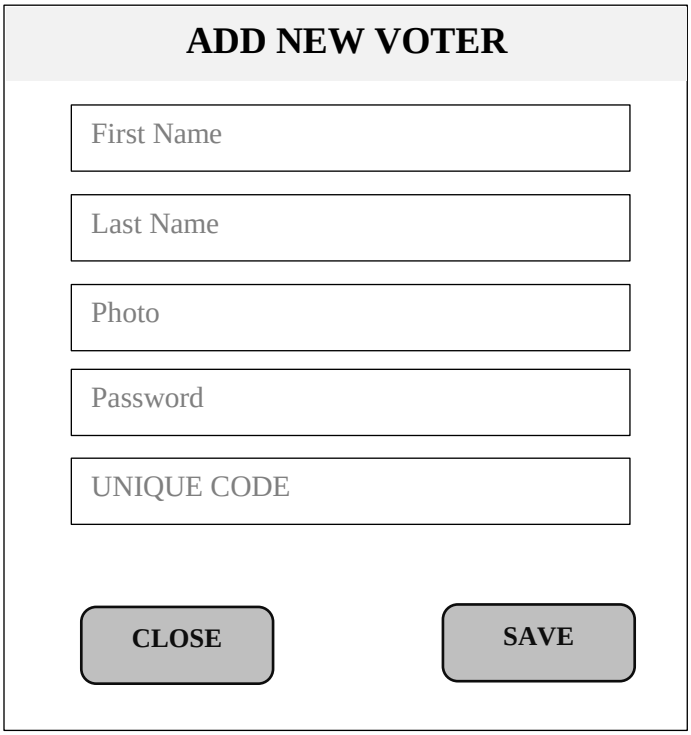


Figure 3.5: Add New Voter Form

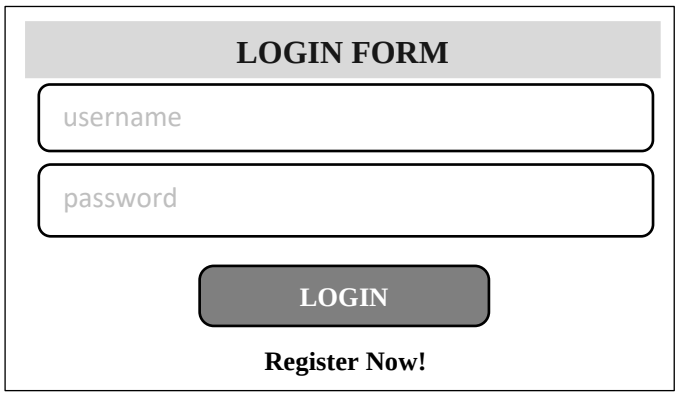


Figure 3.6: Login form

Figure 3.7: Add New Candidate Form

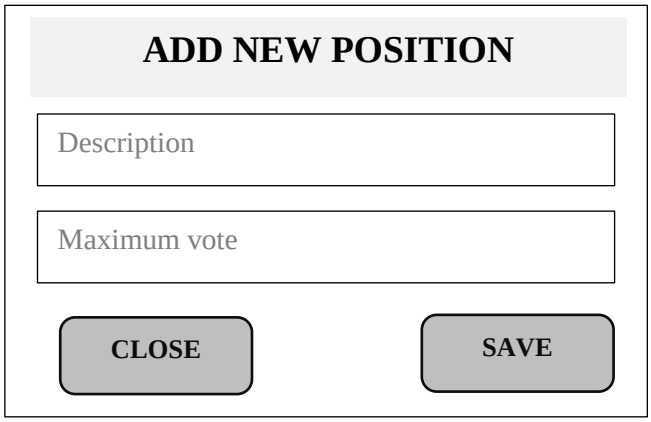


Figure 3.8: Add New Position

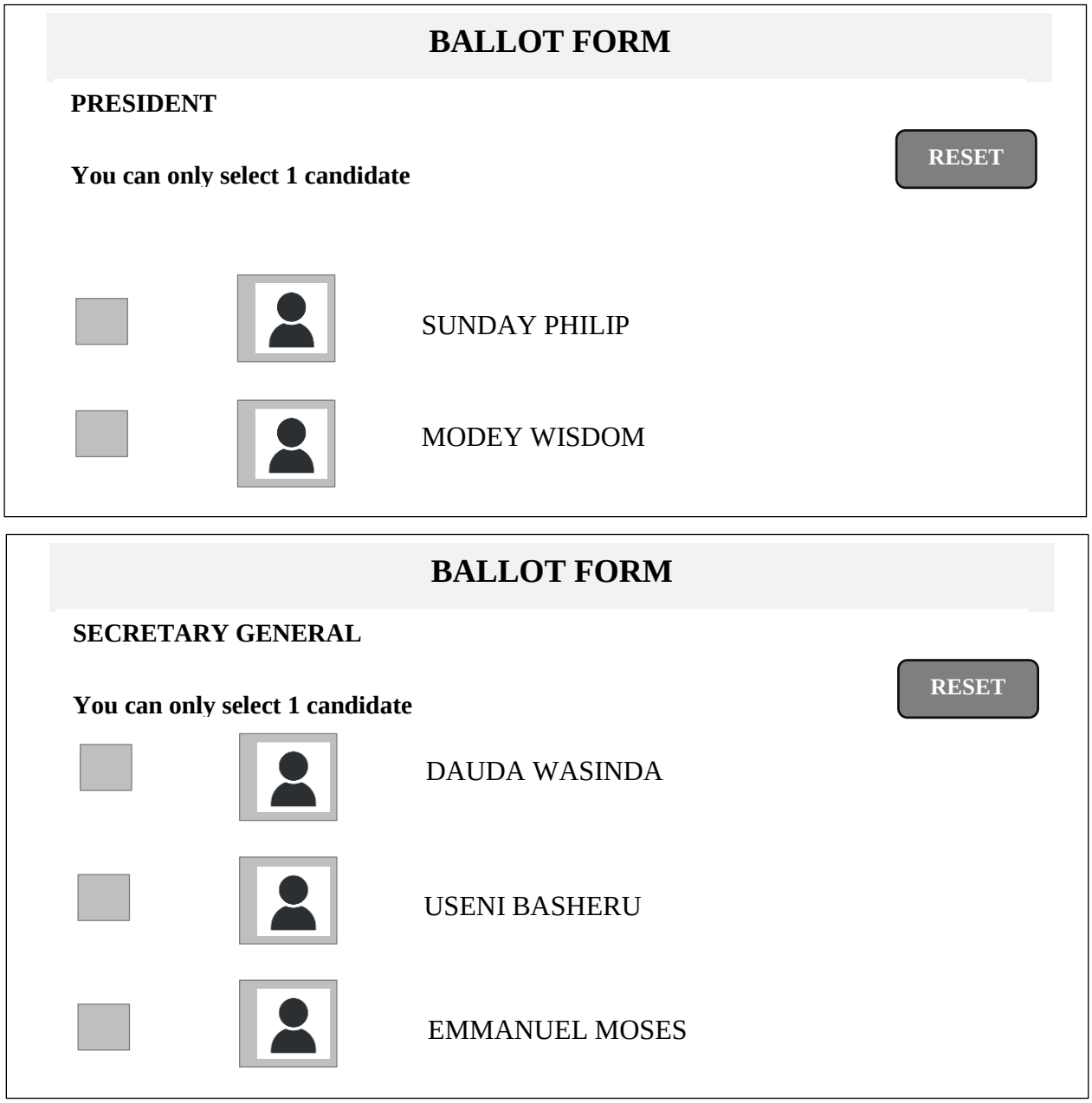


Figure 3.9: Election ballot form

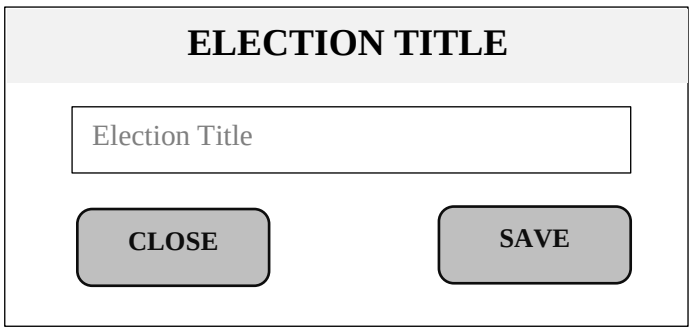


Figure 3.10: Election Title

3.7 System Requirement Specification

3.7.1 Hardware Requirements

The software to be design needs the following hardware for an effective operation of the newly designed system.

- **i.** A system running on intel, P(R) duo core with higher processor
- **ii.** The-Random Access Memory (RAM) should be at least 512MB.
- **iii.** At least 20-GB hard disk.
- **iv.** A monitor.

3.7.2 Software Requirements

The software requirements include:

- **i.** A window 7 or higher version of operating system.
- **ii.** XAMP or WAMP for Database
- **iii.** PHP
- **iv.** MySQL
- **v.** Browser

3.7.3 Personnel Requirements

Any computer literate who has a technical knowhow of internet surfing can use the system because it is user friendly.

CHAPTER FOUR
RESULTS AND DISCUSSION

4.1 Introduction

The new system is designed using PHP and MySQL programming language for easy records inserting and updating. This system will help in managing and easily retrieving of information from the system for management purposes. The new system an Electronic Voting System for NACOSS.

4.2 Results

4.2.1 Login Interface

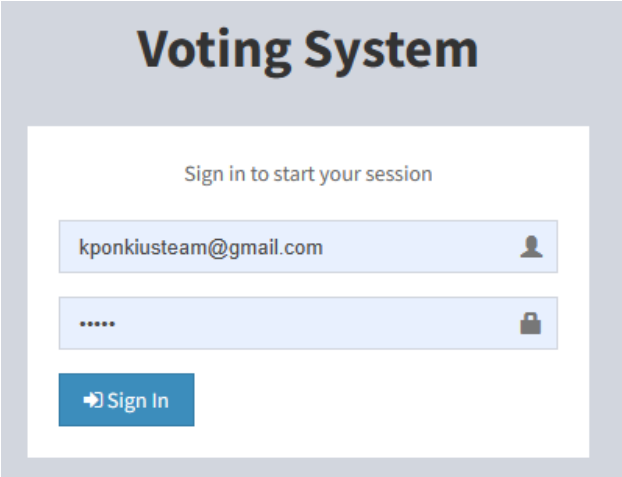


Figure 4.1: Login interface

Figure 4.1 above shows the system login page interface. The login interface allows the voters and Administrator to enter his username and password to get access to the system.

4.2.2 Admin Dashboard

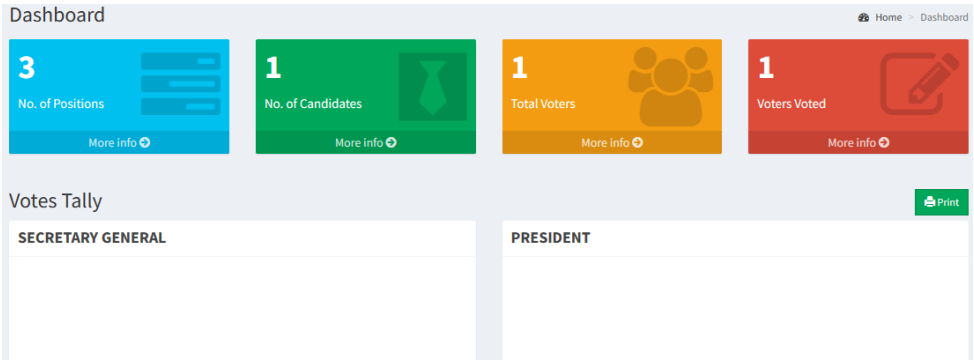


Figure 4.2: Admin Dashboard

Figure 4.2 above shows the system admin dashboard interface. The dashboard interface shows all the tasks that can be performed by the Administrator such as register an applicant, update records, add academic records etc.

4.2.3 Add Voter Interface

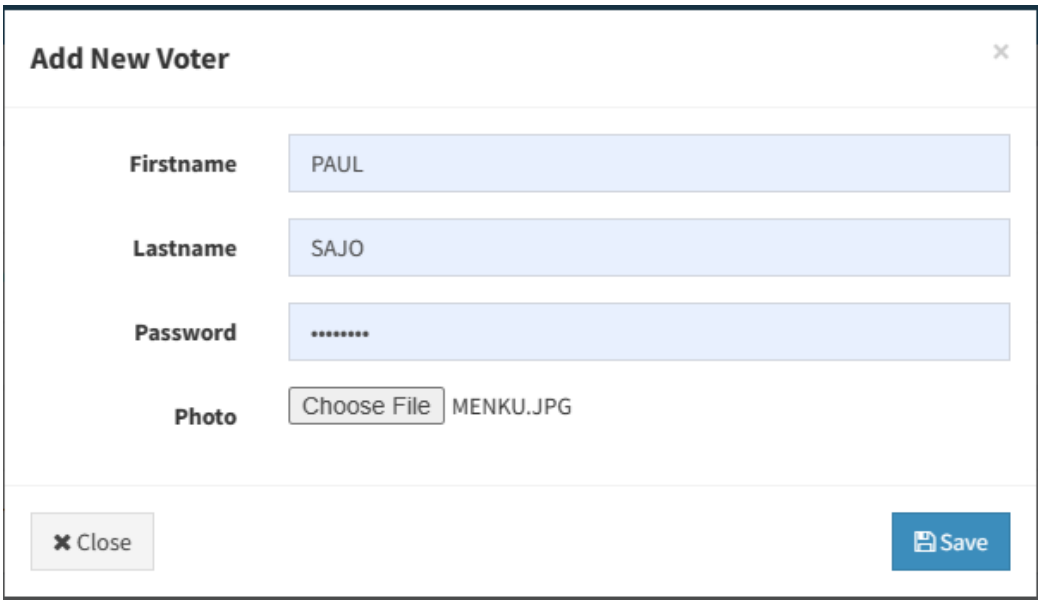


Figure 4.3: Add Voter Interface

Figure 4.3 above shows where administrators can register or add voters into the system which will enable them to cast their votes for a particular candidate and position.

4.2.4 Add Candidate Interface

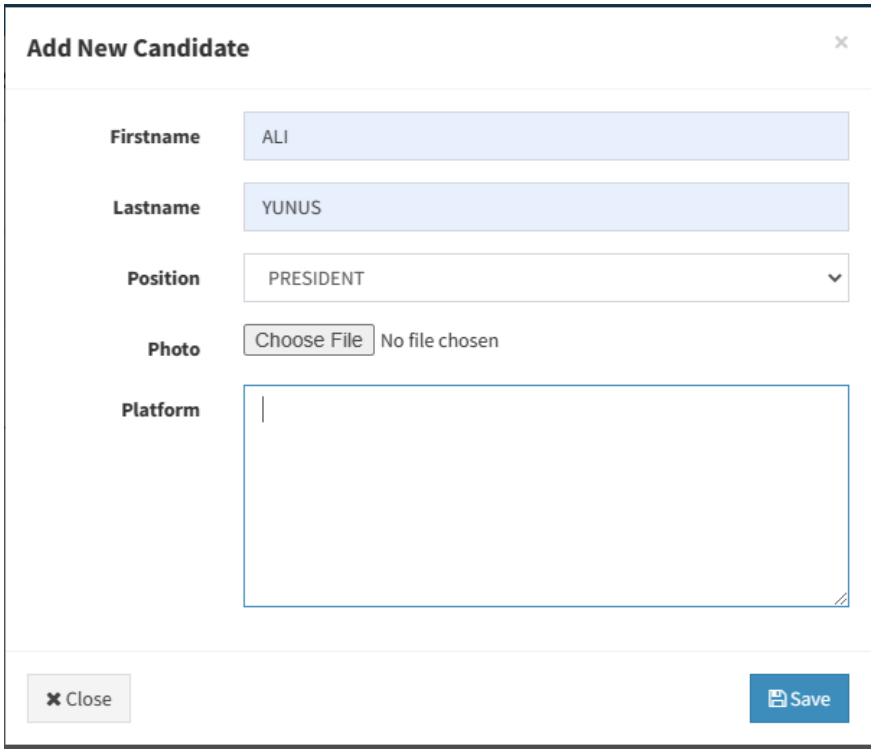


Figure 4.4: Add Candidate Interface

Figure 4.4 is used to add a candidate contesting for the election whom the voter will vote for in a particular position the candidate is contesting for.

4.2.5 Candidate List Interface

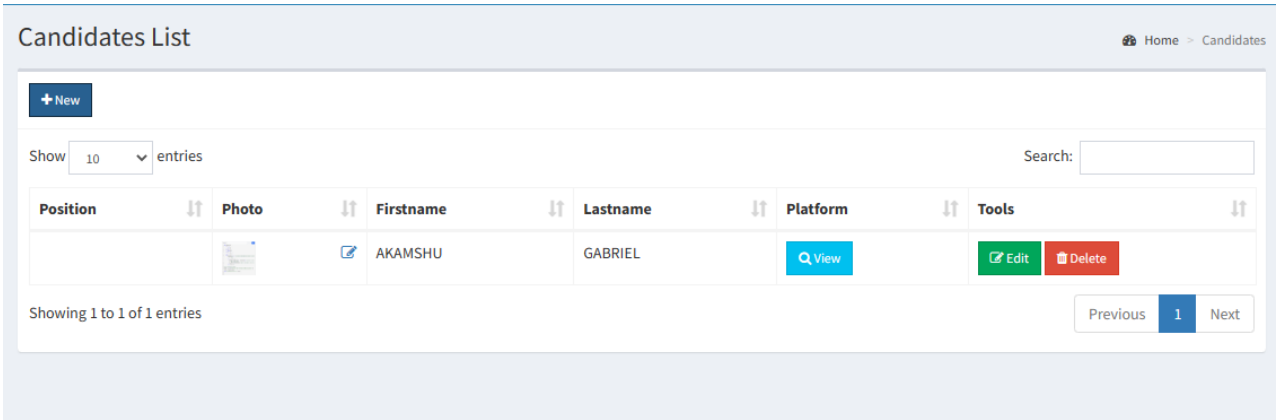


Figure 4.5: Candidate List Interface

Figure 4.5 above displays all the candidates that have been registered into the Voting system, with their respective position.

4.2.6 Add Position Interface

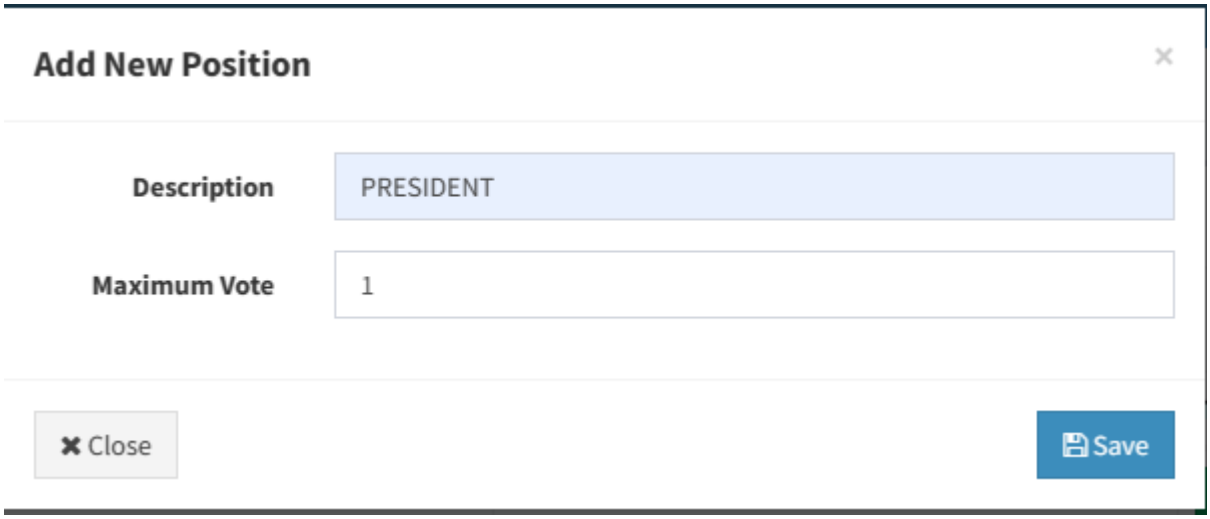


Figure 4.6: Add Position Interface

Figure 4.6 above is used to create or add a position for an election into the system which will be used for the voting process.

4.2.7 Ballot Position Interface

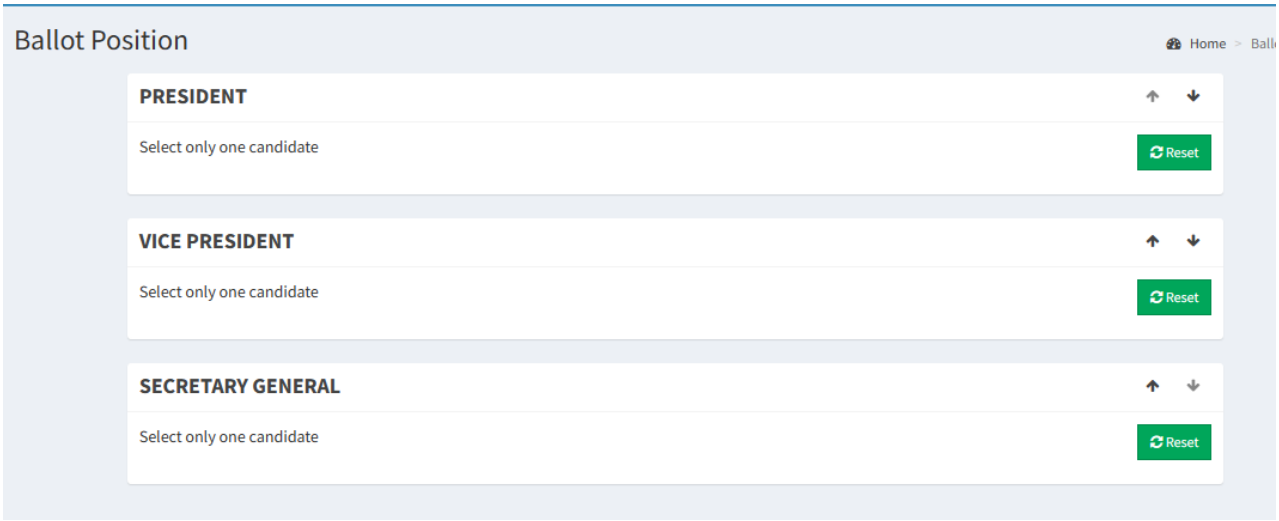


Figure 4.7: Ballot Position Interface

Figure 4.7 above displays the ballot for all the position in a particular election.

4.3 Discussion

Figure 4.1 Login Interface, is the entry point for users, where they input their credentials (e.g., username and password) to access the Evoting System for NACOSS. This interface ensures secure access, authentication, and authorization for users, allowing them to participate in the electronic voting process.

Figure 4.2 shows the Admin Dashboard, which is the central control panel for administrators overseeing the Evoting System. It provides a comprehensive overview of the system's status and functions. Administrators can monitor and manage the voting process, view real-time data, configure settings, and address any issues through this interface.

Figure 4.3 shows the Add Voter Interface enables administrators to add eligible voters to the system. It typically involves inputting voter information and verifying their eligibility. This interface ensures that only authorized individuals are granted the privilege to participate in the electronic voting process.

Figure 4.4 shows the Add Candidate Interface allows administrators to register candidates for NACOSS elections. Admins input candidate details and validate their eligibility. It facilitates the creation of a list of eligible candidates for different positions in the organization, ensuring a structured election process.

Figure 4.5 shows the Candidate List Interface provides a view of the registered candidates and their respective positions. It is often accessible to voters for reviewing candidates before voting. Voters

can make informed decisions by accessing this interface, understanding the available choices for each position.

Figure 4.6 is the Add Position Interface permits administrators to define and create positions for the NACOSS elections. They specify the roles or offices to be contested. This interface organizes the election process by establishing the positions for which candidates can run.

Figure 4.7 shows the Ballot Position Interface is where voters cast their votes for their preferred candidates or choices. It typically presents a digital ballot with the list of positions and candidates. This is the core interface for voters, enabling them to participate in the election by selecting their preferred candidates for each position.

These sections collectively form the user interfaces and functionalities of the Evoting System for NACOSS. They are designed to ensure a secure, efficient, and transparent electronic voting process, benefiting both administrators and voters in the organization's democratic processes.

4.4 User Manual

The following are the necessary steps to take in order to use the system efficiently and effectively.

i. Load the url of the system https://localhost/votingsystem/ the welcome page will be displayed.
ii. Click on the **Proceed** button to proceed to the main system.
iii. If you created an account, provide your login details by entering your username and password.
iv. Depending on the login details provided you will be automatically directed to the dashboard.
v. The various task that you can perform on the portal will be displayed on the sidebar of the dashboard.

CHAPTER FIVE
SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary

In this research project, we have successfully designed and implemented an E-Voting System tailored for NACOSS (the National Association of Computer Science Students) at the Federal Polytechnic, Mubi. The system aimed to modernize and streamline the electoral process within the organization, making it more efficient, secure, and accessible to members. Key components of the system include a user-friendly interface for voters and administrators, voter registration, candidate registration, ballot casting, and real-time results tabulation.

Throughout the project, we conducted a thorough analysis of the existing electoral processes, identified the challenges, and leveraged modern technology to address these issues. The system was rigorously tested to ensure reliability, security, and ease of use.

5.2 Conclusion

The successful design and implementation of the E-Voting System for NACOSS at the Federal Polytechnic, Mubi, mark a significant milestone in the organization's electoral processes. This system provides numerous advantages, including:

Efficiency: The electronic voting system simplifies the entire voting process, from registration to result tabulation, reducing the time and effort required to conduct elections.

Security: Robust security measures, including user authentication and data encryption, protect the integrity of the voting process and ensure that only eligible voters can participate.

Transparency: The system promotes transparency by providing real-time updates on voter turnout and election results, enhancing trust in the electoral process.

Accessibility: Members can participate in NACOSS elections from anywhere, removing geographical barriers and increasing voter engagement.

The successful implementation demonstrates the feasibility and benefits of adopting E-Voting systems in similar organizations, contributing to the advancement of technology in the context of student associations.

5.3 Recommendations

Based on our findings and experience with the E-Voting System, we offer the following recommendations:

i. Training and Awareness: Conduct training sessions and awareness campaigns to ensure that members of NACOSS understand and can effectively use the E-Voting System in future elections.
ii. Regular System Updates: Continuously update and maintain the system to address emerging security threats and improve functionality.
iii. Feedback Mechanism: Implement a feedback mechanism to collect input from users and administrators for ongoing system improvement.
iv. Expand Usage: Consider expanding the use of the E-Voting System to other student organizations and institutions, sharing the benefits of efficient and transparent elections.
v. Legal Framework: Develop and adopt a legal framework or constitution that officially recognizes the use of electronic voting systems in NACOSS elections.

5.4 Contribution to Knowledge

This research project contributes to knowledge in several ways:

It demonstrates the practical implementation of an E-Voting System in a student organization, showcasing the benefits and feasibility of adopting modern technology in electoral processes.

It highlights the importance of security measures and transparency in electronic voting systems, offering insights into best practices for ensuring the integrity of the voting process.

It serves as a case study that can inform similar organizations and institutions seeking to modernize their election procedures.

5.5 Area for Further Work

While the E-Voting System for NACOSS at the Federal Polytechnic, Mubi, represents a significant achievement, there are opportunities for further work in the following areas:

i. Usability Study: Conduct a usability study to identify any user experience challenges and make improvements to enhance user satisfaction.
ii. Integration: Explore the integration of biometric authentication for an even more secure and user-friendly voting process.
iii. Scalability: Assess the system's scalability to accommodate larger memberships and elections.
iv. Accessibility: Ensure that the system is accessible to members with disabilities, complying with accessibility standards.
v. Security Audits: Regularly conduct security audits and vulnerability assessments to maintain the system's robustness.

REFERENCES

Al-Nashif, Y., & Milutinovic, V. (2020). E-Voting System Based on Blockchain Technology. *Journal of Information Management,* 11(1), 38.

Dhamija, R., & Dhamija, G. (2020). An E-Voting System Using Homomorphic Encryption Scheme. *In 2020 International Conference on Power, Energy, Communication, Smart Grid and Computing (PECSGC),* 361-365.

García, R. R., Mateos, C. A., & Grijalva, S. (2021). Design and Implementation of an E-Voting System Using Biometric Authentication. *Journal of Information Systems Engineering & Management,* 6(1), 3.

Hanandeh, E., Jbara, S., Shatnawi, M. A., & Alkhaldi, K. (2021). The Adoption of E-Voting Systems in Developing Countries. *International Journal of Interactive Mobile Technologies*, 15(2), 4-18.

Jain, P., & Jain, S. (2021). Security and Usability Analysis of E-Voting Systems: A Review. *Journal of Ambient Intelligence and Humanized Computing,* 12(10), 13379-13396.

Klemenc, B., & Volk, M. (2021). E-Voting Systems: A Review. *Journal of Ambient Intelligence and Humanized Computing*, 12(11), 14189-14203.

Kreischer, C., & Krimmer, R. (2021). Electronic Voting in Europe: Innovations and Challenges. *In Proceedings of the 12th International Conference on Electronic Voting (EVOTE 2021),* 53-61.

Magkos, E., & Mentzas, G. (2021). Blockchain for Secure and Transparent E-Voting: A Systematic Literature Review. *International Journal of Electronic Government Research,* 17(1), 1-23.

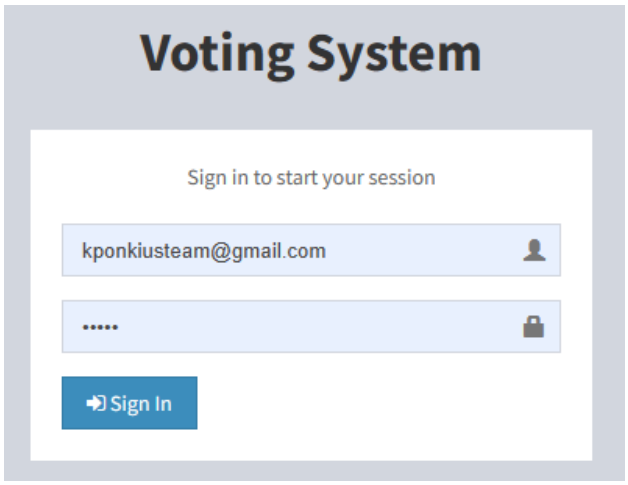
Osório, A., Santos, J., & Morais, M. (2020). E-Voting System with Blockchain and Blind Signature. *In Proceedings of the 15th Iberian Conference on Information Systems and Technologies (CISTI),* 1-6.

Schürmann, C., & Krimmer, R. (2021). Improving Usability in E-Voting Systems: A Systematic Review of International Research. *Government Information Quarterly*, 38(1), 101513.

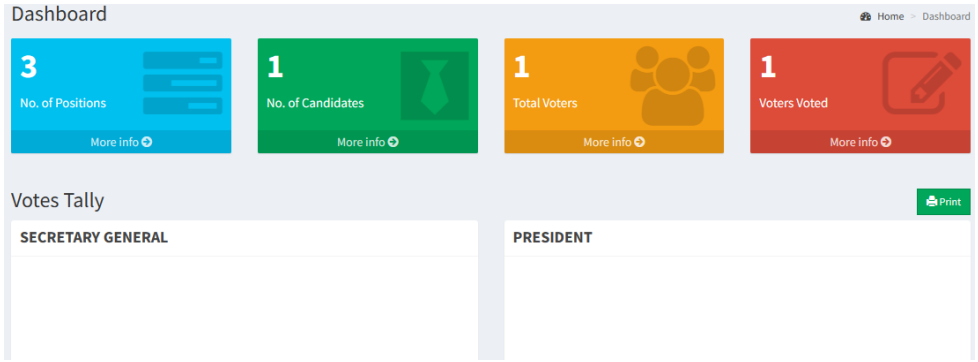
Trabelsi, S., Abid, M., & Mrabet, H. (2020). Towards a Secure E-Voting System Based on the Internet of Things and Blockchain. *In 2020 International Conference on Wireless Networks and Mobile Communications (WINCOM),* 1-6.

APPENDIX A

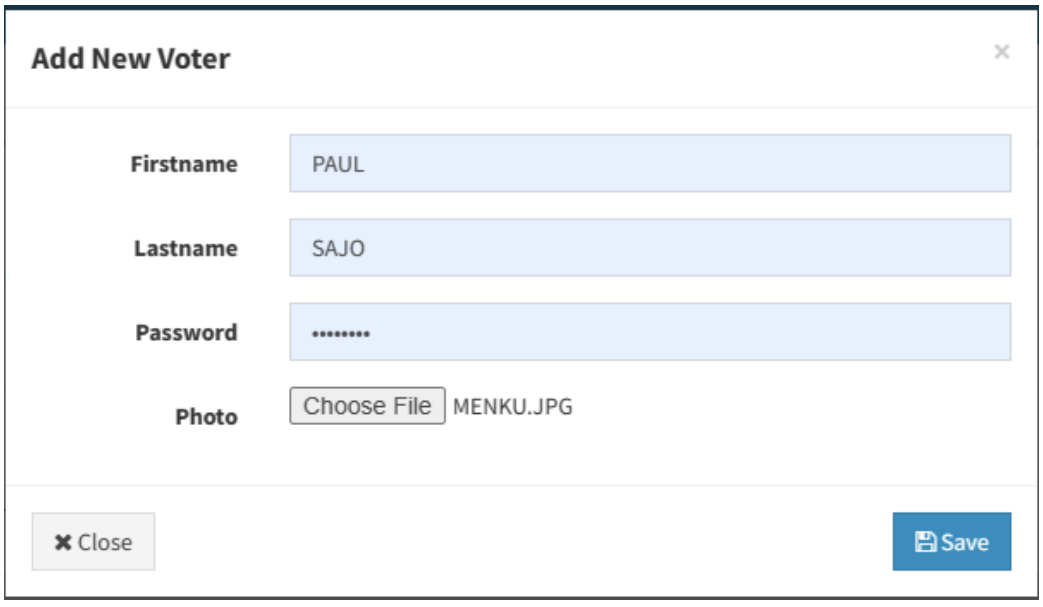
Login Interface



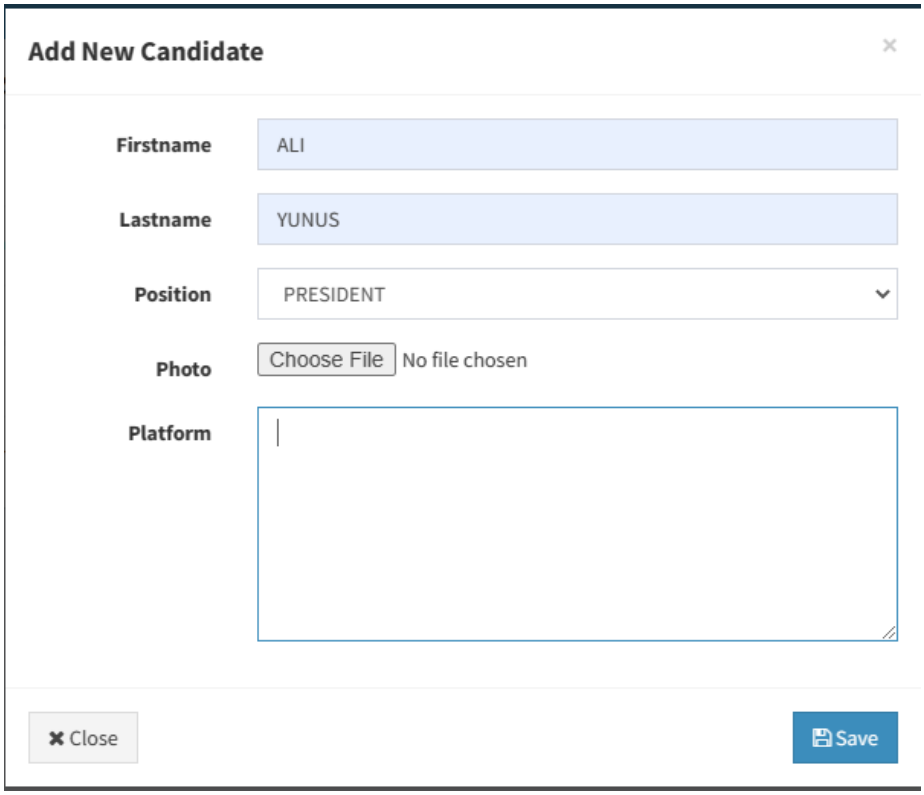
Admin Dashboard



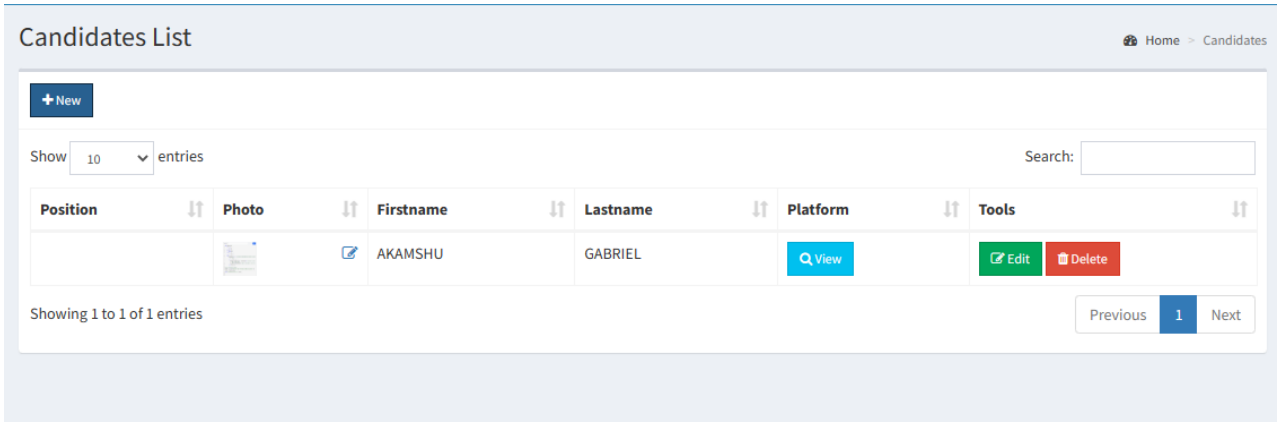
4.2.3 Add Voter Interface



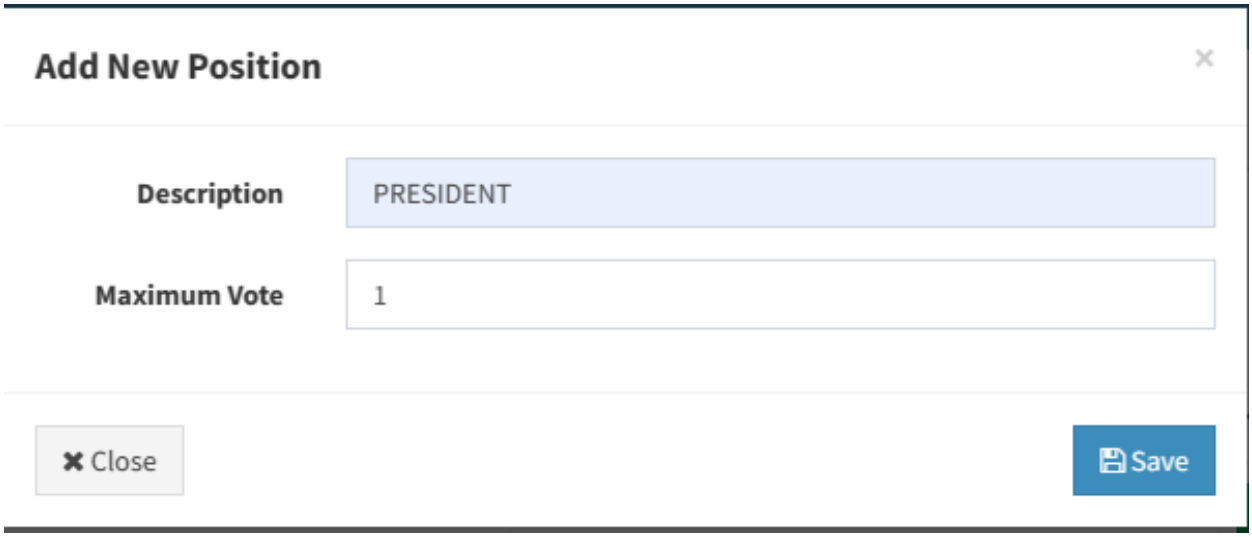
Add Candidate Interface



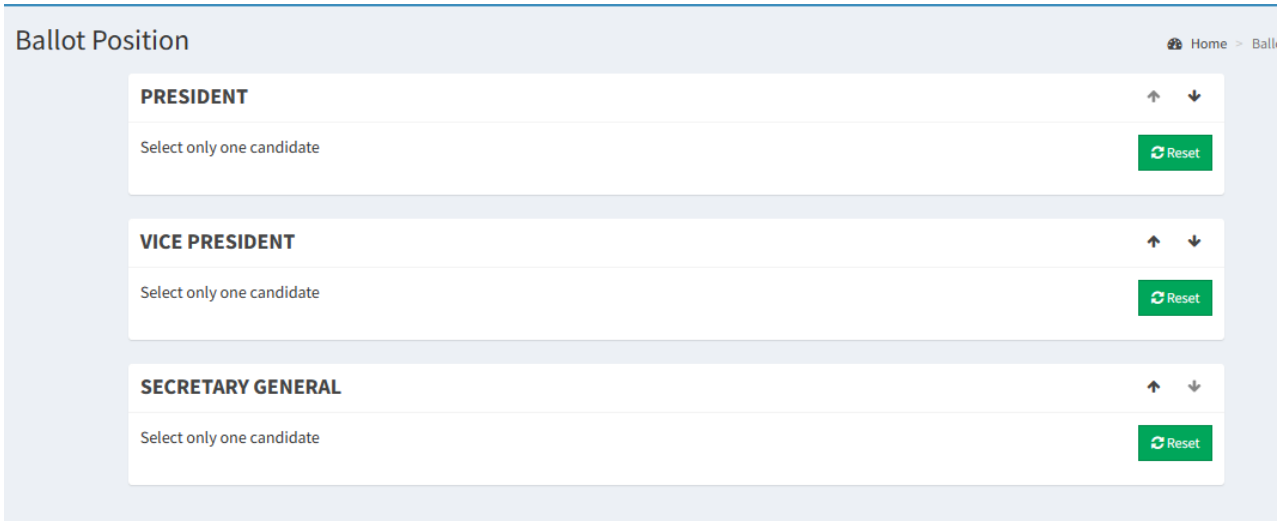
Candidate List Interface



4.2.6 Add Position Interface



Ballot Position Interface



APPENDIX B

PROGRAM CODE

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Voting System using PHP</title>
    <!-- Tell the browser to be responsive to screen width -->
    <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no" name="viewport">
    <!-- Bootstrap 3.3.7 -->
    <link rel="stylesheet"
href="../bower_components/bootstrap/dist/css/bootstrap.min.css">
    <!-- iCheck for checkboxes and radio inputs -->
    <link rel="stylesheet" href="../plugins/iCheck/all.css">
    <!-- Font Awesome -->
    <link rel="stylesheet" href="../bower_components/font-awesome/css/font-
awesome.min.css">
    <!-- Theme style -->
    <link rel="stylesheet" href="../dist/css/AdminLTE.min.css">
    <!-- DataTables -->
    <link rel="stylesheet" href="../bower_components/datatables.net-
bs/css/dataTables.bootstrap.min.css">
    <!-- daterange picker -->
    <link rel="stylesheet" href="../bower_components/bootstrap-
daterangepicker/daterangepicker.css">
    <!-- Bootstrap time Picker -->
    <link rel="stylesheet" href="../plugins/timepicker/bootstrap-
timepicker.min.css">
    <!-- bootstrap datepicker -->
    <link rel="stylesheet" href="../bower_components/bootstrap-
datepicker/dist/css/bootstrap-datepicker.min.css">
    <!-- AdminLTE Skins. Choose a skin from the css/skins
         folder instead of downloading all of them to reduce the load. -->
    <link rel="stylesheet" href="../dist/css/skins/_all-skins.min.css">
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries
-->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->

    <!-- Google Font -->
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,700,300i
talic,400italic,600italic">
```

```
      #candidate_list ul li{
        margin:0 30px 30px 0;
        vertical-align:top
      }

      .clist{
        margin-left: 20px;
      }

      .cname{
        font-size: 25px;
      }
    </style>
</head><body class="hold-transition skin-blue sidebar-mini">
<div class="wrapper">

  <header class="main-header">
  <!-- Logo -->
  <a href="#" class="logo">
    <!-- mini logo for sidebar mini 50x50 pixels -->
    <span class="logo-mini"><b>V</b>TS</span>
    <!-- logo for regular state and mobile devices -->
    <span class="logo-lg"><b>Voting</b>System</span>
  </a>
  <!-- Header Navbar: style can be found in header.less -->
  <nav class="navbar navbar-static-top">
    <!-- Sidebar toggle button-->
    <a href="#" class="sidebar-toggle" data-toggle="push-menu" role="button">
      <span class="sr-only">Toggle navigation</span>
    </a>

    <div class="navbar-custom-menu">
      <ul class="nav navbar-nav">
        <!-- User Account: style can be found in dropdown.less -->
        <li class="dropdown user user-menu">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown">
            <img src="../images/facebook-profile-image.jpeg" class="user-image"
alt="User Image">
            <span class="hidden-xs">Neovic Devierte</span>
          </a>
          <ul class="dropdown-menu">
            <!-- User image -->
            <li class="user-header">
              <img src="../images/facebook-profile-image.jpeg" class="img-circle"
alt="User Image">

              <p>
                Neovic Devierte                <small>Member since Apr.
2018</small>
              </p>
```

```
            </li>
            <li class="user-footer">
              <div class="pull-left">
                <a href="#profile" data-toggle="modal" class="btn btn-default btn-
flat" id="admin_profile">Update</a>
              </div>
              <div class="pull-right">
                <a href="logout.php" class="btn btn-default btn-flat">Sign out</a>
              </div>
            </li>
          </ul>
        </li>
      </ul>
    </div>
  </nav>
</header>
<!-- Add -->
<div class="modal fade" id="profile">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
              <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
                  <span aria-hidden="true">&times;</span></button>
              <h4 class="modal-title"><b>Admin Profile</b></h4>
            </div>
            <div class="modal-body">
              <form class="form-horizontal" method="POST"
action="profile_update.php?return=votes.php" enctype="multipart/form-data">
                <div class="form-group">
                    <label for="username" class="col-sm-3 control-
label">Username</label>

                    <div class="col-sm-9">
                      <input type="text" class="form-control" id="username"
name="username" value="nurhodelta">
                    </div>
  <footer class="main-footer">
    <div class="pull-right hidden-xs">
      <b>All rights reserved</b>
    </div>
    <strong>Copyright &copy; 2021 <a
href="https://www.campcodes.com">CampCodes</a></strong>
</footer>  <!-- Reset -->
<div class="modal fade" id="reset">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
              <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
                  <span aria-hidden="true">&times;</span></button>
```

```
              <h4 class="modal-title"><b>Reseting...</b></h4>
            </div>
            <div class="modal-body">
              <div class="text-center">
                  <p>RESET VOTES</p>
                  <h4>This will delete all votes and counting back to 0.</h4>
              </div>
            </div>
            <div class="modal-footer">
              <button type="button" class="btn btn-default btn-flat pull-left"
data-dismiss="modal"><i class="fa fa-close"></i> Close</button>
              <a href="votes_reset.php" class="btn btn-danger btn-flat"><i
class="fa fa-refresh"></i> Reset</a>
            </div>
        </div>
    </div>
</div></div>
<!-- jQuery 3 -->
<script src="../bower_components/jquery/dist/jquery.min.js"></script>
<!-- jQuery UI 1.11.4 -->
<script src="../bower_components/jquery-ui/jquery-ui.min.js"></script>
<!-- Bootstrap 3.3.7 -->
<script src="../bower_components/bootstrap/dist/js/bootstrap.min.js"></script>
<!-- iCheck 1.0.1 -->
<script src="../plugins/iCheck/icheck.min.js"></script>
<!-- Moment JS -->
<script src="../bower_components/moment/moment.js"></script>
<!-- DataTables -->
<script
src="../bower_components/datatables.net/js/jquery.dataTables.min.js"></script>
<script src="../bower_components/datatables.net-
bs/js/dataTables.bootstrap.min.js"></script>
<!-- ChartJS -->
<script src="../bower_components/chart.js/Chart.js"></script>
<!-- ChartJS Horizontal Bar -->
<script src="../bower_components/chart.js/Chart.HorizontalBar.js"></script>
<!-- daterangepicker -->
<script src="../bower_components/moment/min/moment.min.js"></script>
<script src="../bower_components/bootstrap-
daterangepicker/daterangepicker.js"></script>
<!-- datepicker -->
<script src="../bower_components/bootstrap-datepicker/dist/js/bootstrap-
datepicker.min.js"></script>
<!-- bootstrap time picker -->
<script src="../plugins/timepicker/bootstrap-timepicker.min.js"></script>
<!-- Slimscroll -->
<script src="../bower_components/jquery-
slimscroll/jquery.slimscroll.min.js"></script>
<!-- FastClick -->
<script src="../bower_components/fastclick/lib/fastclick.js"></script>
<!-- AdminLTE App -->
```

```
<script src="../dist/js/adminlte.min.js"></script>
<!-- Active Script -->
<script>
$(function(){
  /** add active class and stay opened when selected */
  var url = window.location;

  // for sidebar menu entirely but not cover treeview
  $('ul.sidebar-menu a').filter(function() {
      return this.href == url;
  }).parent().addClass('active');

  // for treeview
  $('ul.treeview-menu a').filter(function() {
      return this.href == url;
  }).parentsUntil(".sidebar-menu > .treeview-menu").addClass('active');

});
</script>
<!-- Data Table Initialize -->
<script>
  $(function () {
    $('#example1').DataTable()
    $('#example2').DataTable({
      'paging'      : true,
      'lengthChange': false,
      'searching'   : false,
      'ordering'    : true,
      'info'        : true,
      'autoWidth'   : false
    })
  })
</script>
<!-- Date and Timepicker -->
<script>
$(function(){
  //Date picker
  $('#datepicker_add').datepicker({
    autoclose: true,
    format: 'yyyy-mm-dd'
  })
  $('#datepicker_edit').datepicker({
    autoclose: true,
    format: 'yyyy-mm-dd'
  })
});
</script>


</body>
</html>
```