DESIGN AND IMPLEMENTATION OF AN E-VOTING SYSTEM

UMEH MARYBLESSING C.
2014364386

DEPARTMENT OF ELECTRONIC AND COMPUTER ENGINEERING
NNAMDI AZIKIWE UNIVERSITY, AWKA

NOVEMBER, 2019.

DESIGN AND IMPLEMENTATION OF AN E-VOTING SYSTEM

UMEH MARYBLESSING C.

2014364386

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF ENGINEERING (B.ENG) IN THE DEPARTMENT OF ELECTRONIC AND COMPUTER ENGINEERING
NNAMDI AZIKIWE UNIVERSITY, AWKA

NOVEMBER, 2019.

CERTIFICATION

This project work “Design and Implementation of an E-voting System” was carried out by me under the supervision of Engr. Dr. Ogwugwuam Ezeagwu and has not been submitted in part or full to this university or other institutions for the award of a degree.

Umeh MaryBlessing C.

13th November, 2019

Date

APPROVAL

This is to certify that this project work written by Umeh MaryBlessing C. with registration number 2014364386 has been supervised and approved by the Department of Electronic and Computer Engineering, Nnamdi Azikiwe University Awka.

_____ _____

Engr. Dr. Ogwugwuam Ezeagwu Date
Project Supervisor

_____ _____

Engr. Dr. C. O. Ohaneme Date
Head of Department

_____ _____

External Supervisor Date

DEDICATION

I am dedicating this project to almighty God, from whom I draw my inspirations from, for his grace and mercy to start and finish this work.

ACKNOWLEDGEMENT

I want to specially thank my amiable supervisors, Engr. Dr. Ogwugwuam Ezeagwu - a man of wisdom and deep thinking, for all his efforts and patience in making this project work a reality, Engr. Dr. K. A. Akpado for his thoughtfulness and support and Engr. Dr Azubuike Aniedu, who enthusiastically aided with all the technicalities of this of the project. I really appreciate you all.

Worthy of special recognition and appreciation are my lecturers who imparted in me both scholarly and otherwise. Eng. Dr. C. O. Ohaneme, head of this great department, whose efficient administration and fatherly council has been of great help to me as an individual.

Engr. Prof Idigo Victor Eze, Prof.(Mrs) C. C. Okezie, Prof H. C. Inyiama, Prof E. C. Okafor, Engr. Dr A. C. O. Azubogu, Engr. Dr Steve Ufoaroh, Prof G. N. Onoh, Engr. Dr. T. L. Alumona, Engr. Dr. Tony Isizoh, Engr Dr A. I. Udenze, Engr. Peace Obioma, Engr. Ekene Alagu have all helped me in making this project work a success.

I also thank my parents Mr. and Mrs. Basil Umeh, who has supported me financially and in many ways during my stay in school, may God bless you.

I also acknowledge all my project partners especially Agbafune Anthony O. for his helpful skills and patience.

I also specially appreciate my friends who endured and supported me during the period of this project, Maduabuchi Chinaechetam, Nnamdi-Nwaeze Tochukwu, Ndubuisi Uzondu, the stars would not be our limit, God bless you all.

And finally, to everyone that takes out time to update the Internet with reliable information, from Youtube videos to Meduim articles and even tweets, I really appreciate.

ABSTRACT

Fundamental right to vote or simply voting in elections forms the basis of democracy. The conduct of periodic, competitive, participatory, credible and non-violent elections is one of the main yardsticks used to determine the democratic condition of a state. In Nigeria, elections have been conducted using the manual system of voting ever since we started practicing democracy in 1999, but these elections using the manual means have been marred with a lot of electoral malpractices and hitches. These includes violent attack on the voters, result manipulations, vote buying, remoteness of polling centers etc. These are enough reasons that necessitates the design and construction of an electronic voting system, that goes a long way in addressing most of these problems.

The e-voting system aims to eliminate the bottlenecks evident in the manual voting system such as the lengthy registration process, unnecessary transportation, election violence and ultimately the incredibility of the votes.

This was achieved by developing a time effective registration platform which registers a voter and assigns a voter their voters card immediately. The voter also gets to vote from their nearest safe and convenient polling unit and their votes is counted where it belongs.

The results obtained from subsequent tests were very impressive in terms of time, security and accuracy as compared to the manual system.

Such system with all these capabilities will go a long in ameliorating the aforementioned problems of the existing manual system of voting in the Nigerian electoral process.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

EVS	-	Electronic Voting System
INEC	-	Independent National Electoral Commission
ICT	-	Information Communication Technology
BEME	-	Bill of Engineering Measurement and Evaluation
2FA	-	Two-Factor Authentication
ATM	-	Automated Teller Machine
PIN	-	Personal Identification Number
OTP	-	One Time Password
GPS	-	Global Positioning System
USB	-	Universal Serial Bus
PDA	-	Personal Digital Assistant
ISO	-	International Standard Organization
IEC	-	International Electrical Community
RFID	-	Radio Frequency Identification
EMV	-	Europay, MasterCard, Visa
APDU	-	Application Protocol Data Unit
IAFIS	-	Integrated Automated Fingerprint Identification System
CASE	-	Computer-Aided Software Engineering
DBMS	-	Database Management System
SQL	-	Structured Query Language
PVC	-	Permanent Voters Card
ATM	-	Automated Teller Machine

CHAPTER ONE
INTRODUCTION

1.1 Background of Study

In every democratic setting with persons of differing and inconsistent opinions, decisions must be made between several options. This happens in business environment, educational environment, social organizations, and mostly in governance. One of the ways of making such a decision is through voting. Voting is a formal process of expressing individual opinions for or against some motion. In the governance sector of many organisation this process is always used as a means of selecting or electing a leader. One of the key areas where voting is applied is in election. Election is the formal process of selecting a person for public office or of accepting or rejecting a political proposition by voting.[1]

1.1.1 E-voting System Overview

E-voting (Electronic Voting) as a term encompasses a broad range of voting systems that apply electronic elements in one or more steps of the electoral cycle [3]. There are many levels to e-voting in a broad sense which could be e-collation, e-verification, internet voting, remote online voting e.t.c. Following the definition of a system as anything that takes an input and gives an output, an e-voting system is any system that can offers both electronic and online voting. It could also incorporate e-registration, e-verification, e-collation, remote online voting and real-time result display. An E-voting system (EVS) generally comprises the following for it to work efficiently:

- An interactive voting user interface on an electronic device which provides a friendly environment for voters to authenticate and cast their votes, it also serves as a means of collection the individual votes and storing them in the local and central database.
- An administrative dashboard for voters registration, details update and elections coordination and monitoring.
- A database management system for the storage of election, voting and voters data.
- A result display interface.

E-voting system serves to reduce the cumulative costs of running elections and increase voters participation in election system as it offers voters an easy and convenient way to vote and most importantly, it is a panacea to the issue of long distances covered by voters to a specific destination for their votes to be counted, and also it combat the issues of ballot box snatching which is rampart in the conventional election process in Nigeria.

A great technological improvement is observed in election process mostly in the areas of result collation and transmission. Though the Independent National Electoral Commission INEC has not fully implemented the use of technologies for collation due to lack of legal framework [2]. But, most elections around the world use ICT in elections to some degree, at least to summarize and aggregate the votes. This electronic adaptation is the result of a long period of evolution during which not only the procedures but also the technological means for casting votes changed considerably.

1.2 Problem Statement

The present voting system applicable in the Nigerian electoral system has proved inefficient as the voters' registration process is slow, the manual collation of results takes time and gives room for result manipulation, also the inaccessible nature of election venues which includes the long distances to be covered by voters' to their registered location increases voters' apathy towards the election processes, and finally the issues of ballot box snatching and damage and other election violence and issues associated with the traditional ballot paper voting all defiles the purpose of voting in election process as a formal process of expressing individual opinions for or against some motion.

1.3 Aim and Objectives

In the quest to design a successful system to tackle the issues stated in the problem statement, the aim and objectives of the project are outlined below.

1.3.1 Aim

The aim of this project is to design and implement a low cost automated real-time e-voting system.

1.3.2 Objectives

Project Objectives includes

1. A detailed study of the election processes as it pertains to voting.
2. Design and develop a software platforms for voter registration, election voting, real-time election results collation and monitoring and mostly for voters remote access to elections.
3. Design and develop an electronic device that incorporates smart card reader and fingerprints technology for voters accreditation, authentication and verification.
4. Design and develop an administration dashboard for the election administrators.
5. Run simulations and compare the results of the designed e-voting system and other voting systems.

1.4 Significance of the Project

In view of the rapid development of computer technology in virtually all fields of operation and its use in relation to information management, the projects' benefits are itemized as follows:

1.4.1 To the University

An e-voting system is beneficial to the university as:

1. It will provides a means conduct a more less stressful and fair elections at different levels (faculty, departments, school wide e.t.c) in the university.
2. It will offer an in-depth knowledge of the practical approach to ICT education.
3. It will serve as a hands-on application of theories taught in class as it relates to database, software and hardware development.
4. As its' database is based on a flexible database management system, student and staff details can easily be collected for easy access and monitoring.
5. Its' smart card system can also be applied to other fields (e.g. networking) for easy access of each individuals' data.
6. It will serve as a base for other works in the field of ICT in governance.

1.4.2 To the Society

The significance of an e-voting system to the society and mostly to Nigeria are itemized as follows:

1. It will provide INEC (Independent National Electoral Commission) with a means to conduct less costly and fair elections.
2. The secure and flexible database management system safeguards data and information to account for credible elections.
3. It will serve to reduce the workload in the process of conducting election.
4. As it incorporates remote voting individuals can vote from their convenience.
5. It will enable INEC reduce the time wasted in collating and announcing election result.
6. It will greatly reduce and eliminate disenfranchising electorates.
7. It will serve to eliminate invalid votes, curb election violence as votes are counted immediately as they are cast.

1.5 Scope/Limitations of the Work

This project work is mainly designed to enable the Independent National Electoral Commission to use electronic device to capture voter's information, and to allow voters to their cast votes easily and comfortably to promote a more credible election which is efficient and less costly. The dynamic nature of the elections application interface and database structure allows for different organizations set up and conduct basic elections too. It's online interface enables real-time election monitoring and result collation. Some of its major limitations are:

1. It requires network access: Since the collection and sending of votes to the database requires an internet access which may not be readily available in some urban area would seem a limiting factor, though the local database and the printed vote can be used for counting until network is restored.
2. The cost of setting up an e-voting system is high: Due to the delicate nature of such a system and the fact that its' major components are presently not locally source, it would be quite costly to setup, but its usage and maintenance cost is far better than the present ballot paper system.
3. It depends on electricity to a point: In as much as it has an in built battery that can last for the required election duration on daily basis, a case of low battery would require it to recharge, which may not be possible if there is no electric power at the moment.

1.6 Project Outline

This project work on e-voting system is made up of five chapters: introduction, literature review, methodology and system design, systems implementation and result analysis, conclusion and recommendation.

In the chapter one of this project, the introduction which briefly explains voting and elections in general, is seen. It goes further to explain the background of an e-voting system, the aim and objectives of the e-voting system, its significance, scope, and constraints. It summaries by giving the project outline.

In the chapter two, a review of previous literature and technologies used for e-voting system was treated. We also see the different approaches to e-voting systems, their implementation, criticism with their literature reviews and noted the various gaps in the existing literatures.

In chapter three, we see the block diagram of the project work, different methodologies used in development stages, the different phases of the project work which include its research, design, microcomputer programming, display programming, testing and fabrication. We extensively cover the

requirements of the project, the mathematical models used, computer algorithms, designs and software incorporated in the work.

In chapter four, we talk about the steps taken and techniques used for the actual implementation of the project. We see tests carried out to ensure that the project is efficient and also display the result gotten and their significance. We also see the problems encountered and the techniques and solutions taken to overcome them or not. This chapter also sees the detailed analysis of the Bill of Engineering Measurement and Evaluation (BEME) for the project.

And finally, in chapter five, we conclude the work and give notable recommendations for optimal operation of the product. Also we provide suggestions for improvement, enhancement and optimization of our existing work. We also outline the major contribution to the body of knowledge in which our work has achieved.

CHAPTER TWO

LITERATURE REVIEW

2.1 Theories of the Technologies Involved

2.1.1 Overview of Two-factor Authentication

Two-factor authentication (also known as 2FA) is a type, or subset, of multi-factor authentication. Multi-factor authentication is an authentication method in which a computer user is granted access only after successfully presenting two or more pieces of evidence (or factors) to an authentication mechanism, knowledge (something the user and only the user knows), possession (something the user and only the user has), and inherence (something the user and only the user is). [5][6]

Hence, 2FA is a method of confirming users' claimed identities by using a combination of two different factors: 1) something they know, 2) something they have, or 3) something they are. A good example of two-factor authentication is the withdrawing of money from an ATM; only the correct combination of a bank smart card (something the user possesses) and a PIN (something the user knows) allows the transaction to be carried out. Two other examples are to supplement a user-controlled password with a one-time password (OTP) or code generated and received by user (e.g a security token) on smartphone that only the user possesses.[7]

Another subset is Two-step verification or two-step authentication which is a method of confirming a user's claimed identity by utilizing something they know (password) and a second factor other than something they have or something they are. An example of a second step is the user repeating back something that was sent to them through an out-of-band mechanism. Also, the second step might be a six digit number generated by another system that is common to the user and the authentication system. [8]

2.1.1.1 Authentication Factors

The use of multiple authentication factors to prove one's identity is based on the premise that an unauthorized actor is unlikely to be able to supply the factors required for access. If, in an authentication attempt, at least one of the components is missing or supplied incorrectly, the user's identity is not established with sufficient certainty and access to the system (e.g a building, or data,)

being protected by multi-factor authentication then remains blocked. The authentication factors of a multi-factor authentication scheme may include:

- some physical object in the possession of the user, such as a USB stick with a secret token, a bank smart card, a key, etc.
- some secret known to the user, such as a password, PIN, TAN etc.
- some physical characteristic of the user (biometric), such as a fingerprint, eye iris, voice, typing speed, pattern in key press intervals, etc.[9]
- somewhere you are, such as connection to a specific computing network or utilizing a GPS signal to identify the location.[10]

The above authentication factors are further discussed under the following sub headings:

1. Knowledge Factors
2. Possession Factors
3. Inherent Factors
4. Location Based Factors

Knowledge Factors: are the most commonly used form of authentication. In this form, the user is required to prove knowledge of a secret in order to authenticate. A password is a secret word or string of characters that is used for user authentication. This is the most commonly used mechanism of authentication. Many multi-factor authentication techniques rely on password as one factor of authentication.[11] Variations include both longer ones formed from multiple words (a passphrase) and the shorter, purely numeric, personal identification number (PIN) commonly used for ATM access. Traditionally, passwords are expected to be memorized. Many secret questions such as "Where were you born?" are poor examples of a knowledge factor because they may be known to a wide group of people, or be able to be researched.

Possession Factors: ("something the user and only the user has") have been used for authentication for centuries, in the form of a key to a lock. The basic principle is that the key embodies a secret which is shared between the lock and the key, and the same principle underlies possession factor authentication

in computer systems. A security token is an example of a possession factor. Possession factors could be grouped as follows:

i. Disconnected tokens.
ii. Connected tokens.
iii. Software tokens.

Disconnected tokens have no connections to the client computer. They typically use a built-in screen to display the generated authentication data, which is manually typed in by the user.[12]

Figure 2.1: RSA SecurID token, an example of a disconnected token generator

Connected tokens are devices that are physically connected to the computer to be used. Those devices transmit data automatically.[13] There are a number of different types, including card readers, wireless tags and USB tokens.[13]

Software token (a.k.a. soft token) is a type of two-factor authentication security device that may be used to authorize the use of computer services. Software tokens are stored on a general-purpose electronic device such as a desktop computer, laptop, PDA, or mobile phone and can be duplicated. (Contrast hardware tokens, where the credentials are stored on a dedicated hardware device and therefore cannot be duplicated, absent physical invasion of the device.) A soft token may not be a device the user interacts with. Typically an X.509v3 certificate is loaded onto the device and stored securely to serve this purpose.

Inherent Factors: are factors associated with the user, and are usually biometric methods, including fingerprints, face, voice, or iris recognition. Behavioral biometric such as keystroke dynamics can also be used.

Location Based Factors: Increasingly, a fourth factor is coming into play involving the physical location of the user. While hard wired to the corporate network, a user could be allowed to authenticate utilizing only a pin code while off the network entering a code from a soft token as well could be required. This could be seen as an acceptable standard where access into the office is controlled. Systems for network admission control work in similar ways where your level of network access can be contingent on the specific network your device is connected to, such as WIFi vs wired connectivity. This also allows a user to move between offices and dynamically receive the same level of network access in each.

Authentication is an important security feature in systems and should be further enhanced with two-layer authentication which creates added trust to the overall integrity of the system.

2.1.2 Smart Card Technologies

A smart card, typically a type of chip card, is a plastic card that contains an embedded computer chip–either a memory or microprocessor type–that stores and transacts data. This data is usually associated with either value, information, or both and is stored and processed within the card's chip. The card's data is transacted via a reader that is part of a computing system.

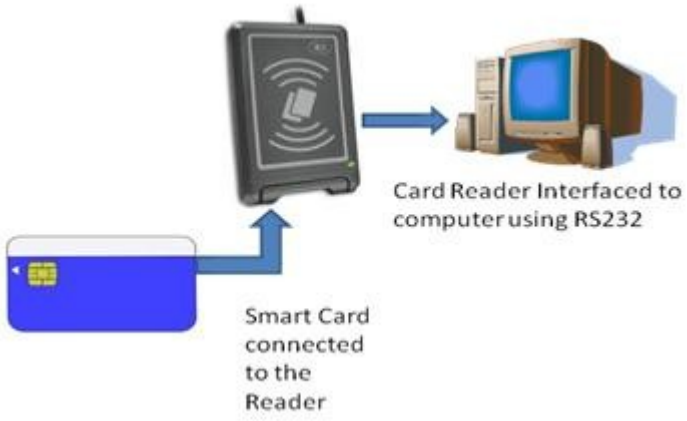

Figure 2.2: A Pictorial representation of a Smart Card System

Systems that are enhanced with smart cards are in use today throughout several key applications, including healthcare, banking, governance, entertainment, and transportation. All these applications can benefit from the added features and security that smart cards provide. Smart cards belong to the family of machine readable cards used for authentication, though markets that have been traditionally served

by other machine readable card technologies, such as bar-code and magnetic stripe, and conventional authentication means, such as password and forms, are converting to smart cards as the calculated return on investment is revisited by each card issuer year after year [14].

ISO/IEC is one of the worldwide standard-setting bodies for technology, including plastic chip cards. The primary standards for smart cards are ISO/IEC 7816, ISO/IEC 14443, ISO/IEC 15693 and ISO/IEC 7501. These standards define the physical dimensions, the electrical interface, the communication protocols and the database structure approach. A smart card system is consist mainly of a smart card and a card reader.

2.1.2.1 The Smart Card Chip

The smart card uses the same standard plastic media as magnetic stripe based cards to carry the electronic chip: eight gold platted contacts are used to connect the silicon to the external world. These contacts are arranged according to the ISO7816-1 specification as shown in figure 2.3 below.

Figure 2.3: ISO-7816 standard pin-out of a basic Smart card chip [17]

2.1.2.1.1 Description of Smart Card Pin-out

1. C1 (VCC +5V DC) : For power supply input (optional use by the card)
2. C2 (RESET) : Reset signal, used to reset the card's communications. Either used itself (reset signal supplied from the interface device) or in combination with an internal reset control circuit (optional use by the card). If internal reset is implemented, the voltage supply on Vcc is mandatory
3. C3 CLOCK : Provides the card with a clock signal, from which data communications timing is derived
4. C4 (RESERVED AUX1): Optionally used for USB interfaces and other uses.

5. C5 (GND): Ground (reference voltage)
6. C6 (Vpp): Programming voltage input (optional). This contact may be used to supply the voltage required to program or to erase the internal non-volatile memory. ISO/IEC 7816-3:1997 designated this as a programming voltage: an input for a higher voltage to program persistent memory (e.g., EEPROM). ISO/IEC 7816-3:2006 designates it SPU, for either standard or proprietary use, as input and/or output [17].
7. C7 (I/O) : Input or Output for serial data (half-duplex) to the integrated circuit inside the card.
8. C8 (RESERVED AUX2): Optionally used for USB interfaces and other uses.

2.1.2.2 Advantages of Smart Cards

1. Smart cards can provide a higher level of security than magnetic stripe cards as they can contain microprocessors capable of processing data directly without remote connections.
2. Smart cards are typically made of plastic, generally polyvinyl chloride and is of dimension 85.60 by 53.98 millimeters, which makes them portable and very easy to carry about.
3. Another advantage of smart cards is that once information is stored on a smart card, it can't easily be deleted, erased or altered. As such, smart cards are good for storing valuable data that can't be – or shouldn't be -- easily reproduced.
4. Smart card technology is generally safe against electronic interference and magnetic fields, unlike magnetic stripe cards. In addition, applications and data on a card can be updated through secure channels so issuers do not necessarily have to issue new cards when an update is necessary.
5. Multi-service smart card systems can enable users to access more than one different service with just one smart card.
6. The cost a smart card is very affordable and it is not costly to implement and manage.

2.1.2.3 Disadvantages of Smart Cards

1. While smart cards have many advantages, the cards themselves -- as well as the smart card readers -- can be expensive as it is not locally sourced in Nigeria at the moment.
2. Another disadvantage of smart cards is that not all smart card readers are compatible with all types of smart cards. With multiple types of smart cards available, some use nonstandard

protocols for data storage and card interface; some smart cards and readers also use proprietary software that is incompatible with other readers.

3. While smart cards can be more secure for many applications, they are still vulnerable to certain types of attack. Attacks that can recover information from the chip are possible against smart card technology. Differential power analysis can be used to deduce the on-chip private key used by public key algorithms. Some implementations of symmetric ciphers can be vulnerable to timing attacks or differential power analysis as well.
4. Smart cards may also be physically disassembled in order to gain access to the on-board microchip.

2.1.2.4 Types of Smart Cards

Smart cards can be categorized on different criteria including by how the card reads and writes data, by the type of chip implanted in the card and by the capabilities of that chip [4]. Some of the different of types of smart cards include:

1. **Contact smart cards** are the most common type of smart card. Contact smart cards are inserted into a smart card reader that has a direct connection to a conductive contact plate on the surface of the card. Commands, data and card status are transmitted over these physical contact points.

Figure 2.4: A Contact Smart Card [14].

2. **Contactless smart cards** require only close proximity to a card reader to be read; no direct contact is necessary for the card to function. The card and the reader are both equipped with antennae and communicate using radio frequencies over the contactless link. A contactless smart card functions by being put near the reader to be read.

Figure 2.5: A Contactless Smart Card [15]

3. **Dual-interface cards** are equipped with both contactless and contact interfaces. This type of card enables secure access to the smart card's chip with either the contactless or contact smart card interfaces.

4. **Hybrid smart cards** contain more than one smart card technology. For example, a hybrid smart card might have one embedded processor chip that is accessed through a contact reader as well as an RFID enabled chip used for proximity connection. The two different chips may be used for different applications linked to a single smart card, as when the proximity chip is used for physical access to restricted areas while the contact smart card chip is used for single sign-on authentication.

5. **Memory smart cards** contain memory chips only and can only store, read and write data to the chip; the data on memory smart cards can be over-written or modified, but the card itself is not programmable so data can't be processed or modified programmatically. Memory smart cards can be read-only and used to store data such as a PIN, password or public key; they can also be read-write and used to write or update user data. Memory smart cards can be configured to be rechargeable or disposable, in which case they contain data that can only be used once or for a limited time before being updated or discarded.

6. **Microprocessor smart cards** have a microprocessor embedded onto the chip in addition to memory blocks. A microprocessor card may also incorporate specific sections of files where

each file is associated with a specific function. The data in the files and the memory allocation are managed with a smart card operating system. This type of card can be used for more than one function and is usually designed to enable adding, deleting and otherwise manipulating data in memory.

Smart cards can also be categorized by their application, such as credit card, debit card, entitlement or other payment card, authentication token and so on [14].

2.1.2.5 The Smart Card Reader

Smart card reader provides a path for an application to send and receive commands from the smart card. Smart card readers obtain or "read" data from smart card. They serve as an interface between the smart card and the micro-controller system. These easy-to-install devices read the data that is stored on contact or contactless 13.56 MHz smart cards. The main purpose of the smart card reader interface is two fold:

i. To provide the right power supply voltage to the card, whatever be the external power source value [16].

ii. To translate the voltage levels necessaries to connect the card (pins C1 to C8) to the external controller.

On the other hand, most of the smart card readers serve as writers too and the interface shall support the ISO7816-3 and EMV specifications.

Figure 2.6: A Smart Card Reader and Writer

2.1.3 Fingerprint Authentication System

Biometric is a rapidly advancing field that is concerned with identifying a person based on his or her physiological or behavioral characteristics. Examples of automated biometric include fingerprint, face, iris, and speech recognition. Because a biometric property is an intrinsic property of an individual, it is difficult to surreptitiously duplicate and nearly impossible to share [18]. The greatest strength of biometric, the fact that the biometric does not change over time, is at the same time its greatest liability. Once a set of biometric data has been compromised, it is compromised forever.

A fingerprint is the composition of many ridges and furrows. Fingerprints can be distinguished by Minutia, which are some abnormal points on the ridges [18]. Minutia is divided in to two parts such as: termination and bifurcation. Termination is also called ending and bifurcation is also called branch.

Figure 2.7: Diagram of Minutia [18].

Fingerprint recognition or fingerprint authentication refers to the automated method of verifying a match between two human fingerprints. A fingerprint impression is acquired, typically using an ink-less scanner. The digital image of the fingerprint includes several unique features in terms of ridge bifurcations and ridge endings, collectively referred to as minutia [19].

2.1.3.1 Fingerprint Scanner

A fingerprint scanner is an electronic device used to capture a digital image of the fingerprint pattern. The analysis of fingerprints for matching purposes generally requires the comparison of several features of the print pattern. These include patterns, which are aggregate characteristics of ridges, and minutia points, which are unique features found within the patterns.

Figure 2.8: A Fingerprint Scanner

2.1.3.2 Parts of a Fingerprint Scanner

The fingerprint scanner consists of fingerprint sensor, ADC (Analog to Digital Converter), flash ROM and DSP (Digital Signal Processing) chip.

1. Fingerprint Sensor: The fingerprint sensor is used for scanning the finger impression. The scanning data is in the form of analog. Further, this process is converted by the A/D converter.
2. A/D Converter: Here the analog data from the sensor is converted to the digital data and it is transferred to the processor.
3. Flash ROM: The flash ROM is used to store the data temporarily in the DSP processor and this will work until the data is transferred to the main memory of the host.
4. DSP Chip: The DSP chip is used for processing and receiving the data. For further transfer of data the DSP port is used.
5. DSP Port: It is used for the communication between DSP processor and memory (database).

2.1.3.3 Components of a Fingerprint System

A typical fingerprint system consist of four major components, which consist of:

1. Image capture
2. Feature extraction
3. Pattern matching
4. Database

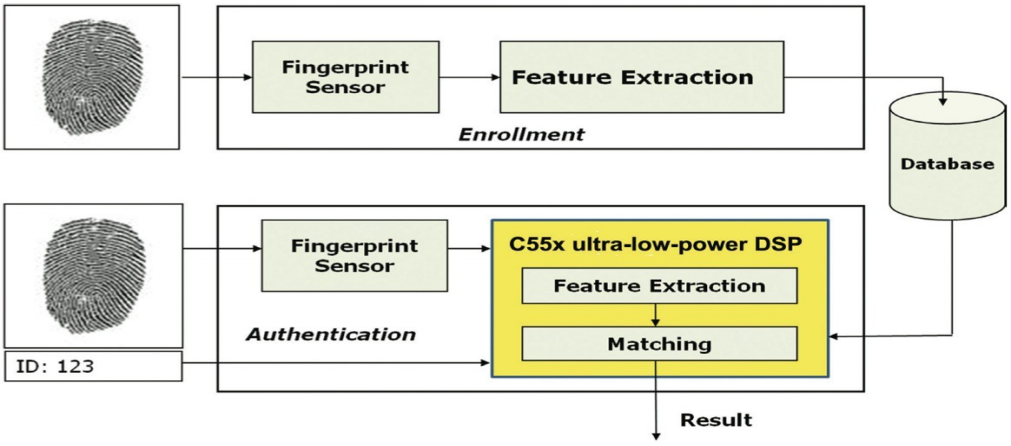

Figure 2.9: Block Diagram of a Fingerprint System [20].

Fingerprint systems translate illuminated images of fingerprints into digital code for further software such as enrollment (fingerprint registration) and verification (authentication or verification of registered users). The scanner uses an advanced CMOS image sensor to capture high contrast, high resolution fingerprint images that are virtually distortion-free. A series of powerful algorithms extract data from the image, mapping the distinguishing characteristics of the fingerprint.

This data is then converted into an encoded binary string known as a **digital template**, and stored in a database. The actual fingerprint image is never stored. To identify or verify a fingerprint, a proprietary matching algorithm compares the new template made from the extracted characteristics from the input fingerprint on the optical module to a previously stored sample. The entire matching process takes roughly one second. Authentication takes place locally at the device or on a server, depending on system configuration [20].

2.1.3.4 Mode of Operation of a Fingerprint System

A fingerprint biometric system can operate in two modes:

1. Verification mode: The system performs a one-to-one (1:1) comparison of a captured fingerprint with a specific template stored in the database in order to verify the individual is the person they claim to be.
2. Identification mode: The system performs a one-to-many (1:N) comparison against fingerprints in the database in an attempt to establish the identity of an unknown individual.

2.1.3.5 Applications of Fingerprint Biometric Technologies

There are numerous applications for the use of Biometric Technology, but the most common ones are as follows:

1. Logical Access Control: This refers to gaining access to a computer network either at the place of the business or corporation or via a secured remote connection from a distant location. Fingerprint systems are deployed to allow for easy access for authenticated users.
2. Physical Access Control: refers to giving a person or an employee of a business or a corporation access to a secure building, or even a secure office from within it. Fingerprint systems are installed at entrance points to grant entrance access to only authentic employers or person with such access or clearance.
3. Time and Attendance: Here, fingerprint biometric is used to take records of attendance of the members of an organisation. The time of arrival and departure can also be recorded and stored in a database for reference purposes.
4. Law Enforcement: This is the most widely known application of fingerprint biometric technologies. Here law enforcement agencies implement fingerprint biometric system as a means of collecting identities of criminals. IAFIS (Integrated Automated Fingerprint Identification System), a worldwide database of fingerprints of criminals is an example of such. IAFIS administrated and maintained by the FBI (Federal Bureau of Investigation) in the United States [21].
5. Surveillance: This is simply keeping tabs of a large group of people, and from there, determining any abnormal behavior from an established baseline. Fingerprint systems are deployed also with face recognition to track for example people with criminal records any erratic behavior.

2.1.3.6 Advantages of Fingerprints Technologies

1. Fairly small storage space is required for the biometric template, reducing the size of the database required.
2. It is one of the most developed biometrics.
3. Each and every fingerprint including all fingers are unique, even identical twins have different fingerprints and as such it is a safe way of identifying individuals.
4. Sound potential for forensic use as most of the countries have existing fingerprint databases.
5. Relatively inexpensive and offers high levels of accuracy.

2.1.3.7 Disadvantages of Fingerprints Technologies

1. Even with its many benefits, fingerprint systems are also associated with some disadvantages that makes their implementation controversial, and they includes:
2. The fingerprint scanner does not take into consideration when a person physically changes. Changes such as growth tends to change fingerprints and accidents such as bruises or cuts or even dirt on the finger can make an already existing user's verification invalid as the fingerprint is now altered.
3. For some people it is very intrusive, because is still related to criminal identification.
4. If anyone can access to an authorized user's prints, he can trick the scanner. The criminal can cut off somebody's finger to get a scanner security system but some scanners have additional pulse and heat sensors to verify that the finger is alive, but these systems can still be fooled by a gelatin print mold over a real finger [22].
5. Having a high security system may require expensive computer hardware and software, certain fingerprint scanners can be quite expensive.

Fingerprint authentication is the cheapest, fastest, most convenient and most reliable way to identify a particular person. It has many functional advantages over traditional systems such as passwords. The greatest strength of the fingerprint authentication technology, is the fact that the fingerprint does not change over time.

Today, fingerprint recognition technology is used for mostly security and identification purpose. As fingerprint recognition technology develops, it is expected that more affordable and more portable

fingerprint recognition devices will become available, and fingerprint recognition will be considered a safe and convenient authentication system.

2.1.4 Database Technologies

At the heart of every fully designed system are the collection, storage, aggregation, manipulation, dissemination, and management of data [23]. **Data** are raw facts. The word raw indicates that the facts have not yet been processed to reveal their meaning. **Information** are data that have been processed in such a way that the knowledge of the person who uses the data is increased. These facts are made available for processing because they are stored at a place for future reference. The two main techniques for data storage in computers are: file system and database.

A file system is a method for storing and organizing computer files and the data they contain to make it easy to find and access them. File systems may use a storage device such as a hard disk or CD-ROM and involve maintaining the physical location of the files. The file system can be used to store less complex data, but in a case of an organizations' data which includes employee details, financial records and so on, a well structured system is required for such task and such a system is referred to as a database.

Database is as an organized collection of logically related data. It is a collection of data, typically describing the activities of one or more related organizations. A database can also be seen as a shared, integrated computer structure that stores a collection of:

- End-user data, that is, raw facts of interest to the end user.
- Metadata, or data about data [23], through which the end-user data are integrated and managed.

The file systems became obsolete as their integration and use becomes difficult when the volume of data stored increases. Its numerous disadvantages led to the development of database as an easier means for data storage, but as the need for a good data manipulation system increased, there was need to develop a management system for databases for quick access and control and that gave rise to the Database Management System (DBMS).

2.1.4.1 Components of the Database Environment

The database operational environment shown in Figure 2.10 is an integrated system of hardware, software, and people, designed to facilitate the storage, retrieval, and control of the information resource and to improve the productivity of the organization. They are includes:

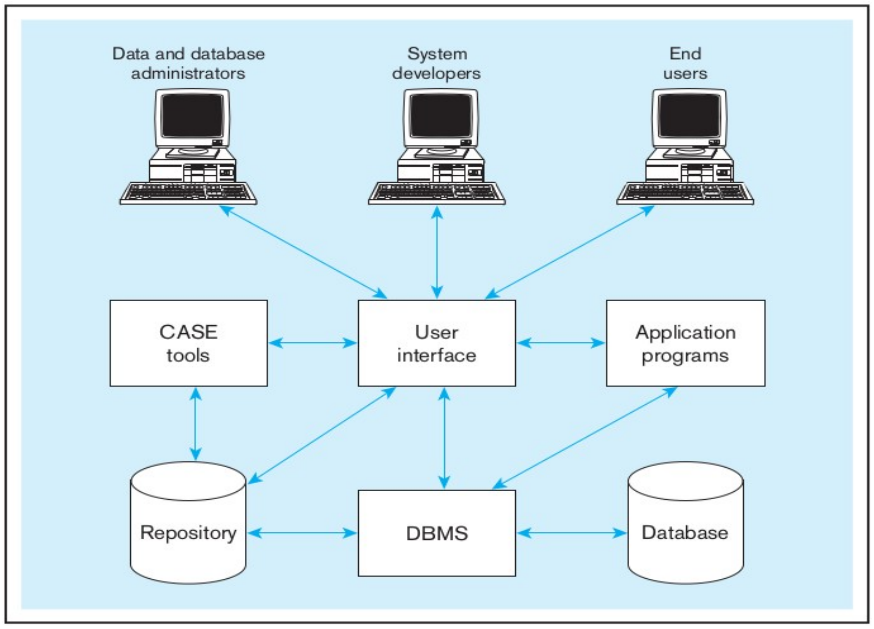

Figure 2.10: Components of the Database Environment [24].

Computer-aided software engineering (CASE) tools: CASE tools are automated tools used to design databases and application programs.

Repository: A repository contains an extended set of metadata important for managing databases as well as other components of an information system.

DBMS: It is a software system that is used to create, maintain, and provide controlled access to user databases.

Database: It is an organized collection of logically related data, usually designed to meet the information needs of multiple users in an organization. It is important to distinguish between the database and the repository. The repository contains definitions of data, whereas the database contains occurrences of data.

Application programs: Computer-based application programs are used to create and maintain the database and provide information to users.

User interface: This includes languages, menus, and other facilities by which users interact with various system components, such as CASE tools, application programs, the DBMS, and the repository.

Data and database administrators: Data administrators are persons who are responsible for the overall management of data resources in an organization. Database administrators are responsible for physical database design and for managing technical issues in the database environment.

System developers: They are persons such as systems analysts and programmers who design new application programs. System developers often use CASE tools for system requirements analysis and program design.

End users: These are persons throughout the organization who add, delete, and modify data in the database and who request or receive information from it. All user interactions with the database must be routed through the DBMS.

2.1.4.2 Database Management System

A database management system (DBMS) is a software system that enables the use of a database. The primary purpose of a DBMS is to provide a systematic method of creating, updating, storing, and retrieving the data stored in a database. It enables end users and application programmers to share data, and it enables data to be shared among multiple applications rather than propagated and stored in new files for every new application [24]. A DBMS also provides facilities for controlling data access, enforcing data integrity, managing concurrency control, and restoring a database.

There are many terms associated with the DBMS used to explain the different operations in a database environment, they includes terms like query -- a specific request issued to the DBMS for data manipulation — for example, to read or update the data. Simply put, a query is a question.

The terms database and DBMS are most times used interchangeably to refer to the database technology and as such we adopt such use here to us focus on the technological aspects of the database approach.

2.1.4.3 Advantages of Database

As a database is only as useful as its DBMS, the advantages of a DBMS are as follows:

1. Improved data sharing: The DBMS helps create an environment in which end users have better access to more and better-managed data. Such access makes it possible for end users to respond quickly to changes in their environment.
2. Improved data security: The more users access the data, the greater the risks of data security breaches. Corporations invest considerable amounts of time, effort, and money to ensure that corporate data are used properly. A DBMS provides a framework for better enforcement of data privacy and security policies.

3. Better data integration: Wider access to well-managed data promotes an integrated view of the organization's operations and a clearer view of the big picture. It becomes much easier to see how actions in one segment of the organisation affect other segments.
4. Minimized data inconsistency: Data inconsistency exists when different versions of the same data appear in different places. For example, data inconsistency exists when a company's sales department stores a sales representative's name as "MaryBlessing" and the company's personnel department stores that same person's name as "Mary-Blessing C.," or when the company's regional sales office shows the price of a product as #4,500 and its national sales office shows the same product's price as #4,490. The probability of data inconsistency is greatly reduced in a properly designed database.
5. Improved data access: The DBMS makes it possible to produce quick answers to ad hoc queries - a spur-of-the-moment question [23]. The DBMS sends back an answer (called the query result set) to the application. For example, end users, when dealing with elections data, might want quick answers to questions (ad hoc queries) such as:
   - What was the total number of registered voters during the past six months?
   - What is the total number of students who can vote?
   - How many candidates are contesting for a particular election?
6. Improved decision making. Better-managed data and improved data access make it possible to generate better-quality information, on which better decisions are based. The quality of the information generated depends on the quality of the underlying data. Data quality is a comprehensive approach to promoting the accuracy, validity, and timeliness of the data. While the DBMS does not guarantee data quality, it provides a framework to facilitate data quality initiatives.
7. Increased end-user productivity. The availability of data, combined with the tools that transform data into usable information, empowers end users to make quick, informed decisions that can make the difference between success and failure in the global economy.
8. Improved data independence: Application programs should be as independent as possible from details of data representation and storage. The DBMS can provide an abstract view of the data to insulate application code from such details.

2.1.4.4 Disadvantages of Database

The database approach entails some additional costs and risks that must be recognized and managed when it is implemented.

1. Need for new, specialized personnel: Frequently, organizations that adopt the database approach need to hire or train individuals to design and implement databases, provide database administration services, and manage a staff of new people.
2. Installation and management cost and complexity: Installing such a system may also require upgrades to the hardware and data communications systems in the organization. Substantial training is normally required on an ongoing basis to keep up with new releases and upgrades. Additional or more sophisticated and costly database software may be needed to provide security and to ensure proper concurrent updating of shared data.
3. Conversion costs: The cost of converting these older systems to modern database technology measured in terms of money, time, and organizational commitment may often seem prohibitive to an organization.
4. Need for explicit backup and recovery: This requires that comprehensive procedures be developed and used for providing backup copies of data and for restoring a database when damage occurs.
5. Organizational conflict: Experience has shown that conflicts on data definitions, data formats and coding, rights to update shared data, and associated issues are frequent and often difficult to resolve.

2.1.4.5 Types of Database

Depending upon the usage requirements, there are following types of databases available:

1. Centralised database: The information(data) is stored at a centralized location and the users from different locations can access this data. This type of database contains application procedures that help the users to access the data even from a remote location. Various kinds of authentication procedures are applied for the verification and validation of end users, likewise, a registration number is provided by the application procedures which keeps a track and record of data usage.
2. Distributed database: The data is not at one place and is distributed at various sites of an organization. These sites are connected to each other with the help of communication links which helps them to access the distributed data easily. There are two kinds of distributed

database, viz. homogeneous and heterogeneous. The databases which have same underlying hardware and run over same operating systems and application procedures are known as homogeneous DDB. Whereas, the operating systems, underlying hardware as well as application procedures can be different at various sites of a DDB which is known as heterogeneous DDB.

3. Personal database: Data is collected and stored on personal computers which is small and easily manageable. The data is generally used by the same department of an organization and is accessed by a small group of people.
4. End-user database: The end user is usually not concerned about the transaction or operations done at various levels and is only aware of the product which may be a software or an application. Therefore, this is a shared database which is specifically designed for the end user, just like different levels' managers. Summary of whole information is collected in this database.
5. Commercial database: These are the paid versions of the huge databases designed uniquely for the users who want to access the information for help. These databases are subject specific, and one cannot afford to maintain such a huge information. Access to such databases is provided through commercial links.
6. NoSQL database: These are used for large sets of distributed data. There are some big data performance issues which are effectively handled by relational databases, such kind of issues are easily managed by NoSQL databases. There are very efficient in analyzing large size unstructured data that may be stored at multiple virtual servers of the cloud. An example of a NoSQL database is MongoDB [25].
7. Operational database: Information related to operations of an enterprise is stored inside this database. Functional lines like marketing, employee relations, customer service etc. require such kind of databases.
8. Relational database: These databases are categorized by a set of tables where data gets fit into a predefined category. The table consists of rows and columns where the column has an entry for data for a specific category and rows contains instance for that data defined according to the category. The Structured Query Language (SQL) is the standard user and application program interface for a relational database.

There are various simple operations that can be applied over the table which makes these databases easier to extend, join two databases with a common relation and modify all existing applications.

9. Cloud database: Now a day, data has been specifically getting stored over clouds also known as a virtual environment, either in a hybrid cloud, public or private cloud. A cloud database is a database that has been optimized or built for such a virtualized environment. There are various benefits of a cloud database, some of which are the ability to pay for storage capacity and bandwidth on a per-user basis, and they provide scalability on demand, along with high availability.

   A cloud database also gives enterprises the opportunity to support business applications in a software-as-a-service deployment.

10. Object-oriented database: An object-oriented database is a collection of object-oriented programming and relational database. There are various items which are created using object-oriented programming languages like C++, Java which can be stored in relational databases, but object-oriented databases are well-suited for those items.

    An object-oriented database is organized around objects rather than actions, and data rather than logic. For example, a multimedia record in a relational database can be a definable data object, as opposed to an alphanumeric value.

11. Graph database: The graph is a collection of nodes and edges where each node is used to represent an entity and each edge describes the relationship between entities. A graph-oriented database, or graph database, is a type of NoSQL database that uses graph theory to store, map and query relationships. Graph databases are basically used for analyzing interconnections. For example, companies might use a graph database to mine data about customers from social media.

2.1.4.6 MongoDB DBMS

MongoDB is not only a general-purpose database which can perform only insert, update and delete data within it. Besides these, there are several important features which make the MongoDB one of the most popular and enriched databases in the world of NoSQL databases. Some of the features are as below,

1. MongoDB supports JSON data models with dynamic schema.
2. In MongoDB, we can perform a search on any field or any range query and also can use a regular expression for searching the data
3. MongoDB supports secondary indexes which allow us to search a variety of data in a very small time-span. It also provides us with different types of indexes like unique index, compound index, geospatial index etc.
4. MongoDB supports aggregation pipeline which helps us to build complex aggregations to optimize the database.
5. MongoDB supports Master-Slave replication [26].
6. MongoDB support automatic load balancing features.
7. MongoDB supports auto-sharding for horizontal scaling.
8. MongoDB can store any type of file which can be any size without affecting our stack.
9. MongoDB basically use JavaScript objects in place of the procedure.
10. MongoDB supports special collection type like TTL (Time-To-Live) for data storage which expires at a certain time.

MongoDB supports all types of operating systems. MongoDB is available in two versions – Community Server Edition (Perfect of Self Use or Developer Mode) and Enterprise Server Edition (For Business Purpose Use with Proper Licensing). The MongoDB installer is available for all types of operating systems like Windows, Linux or Mac OS. Installer for MongoDB can be downloaded from the MongoDB sites.

2.2 Review of Related Literatures

Polling place e-voting and remote i-voting (Internet voting) systems of election have been used in different democratic societies. The United States, Australia, Estonia, Japan, Brazil and India are at various stages of e-voting adoption. In Africa, Namibia was the first country to transit to e-voting in its 2014 general elections. In Nigeria, Kaduna State is the first state to adopt an e-voting system in her Local Government elections in 2018.

The advantage of e-voting over the conventional voting system is obvious. Convenience is an attribute of e-voting that enhances participation and remedies apathy associated with traditional voting methods. E-voting makes it easier for people to make their views known and cast their votes, an important

requisite for a constructive democratic process. Furthermore, poorly designed paper ballots, which might have been filled in or counted incorrectly becomes a thing of the past if e-elections are adopted.

2.2.1 Review of Related Works

In the past years, a lot of work has been carried out by people all over the world with respect to developing an efficient e-voting system for election purpose. Some of these works will now be reviewed.

In [27], the Brazilian e-voting machines are used for voter identification, vote casting and tallying. Political parties have access to the voting machines programs for auditing. The voting system has been widely accepted, since it speeds up the vote count tremendously and helps preventing fraud. Initially a paper trail was included in the e-voting systems. However, this was abandoned later due to technical problems associated with the printers. The missing of a paper trail is sometimes criticized since vote auditing is deemed impossible. Critics argue that this makes the whole process highly dependent on trusting the software [27].

The Brazilian Supreme Electoral Court regularly funds research aimed at improving security. E.g. in 2009, a hacking competition was organized to create additional confidence in the technology. In 2011, new biometrics based voting machines were being developed. The Electoral Court started implementing biometric identification in the electoral process in 2012.

In Estonia [28], the National Electoral Committee started the actual e-voting project. A public procurement procedure was carried out and the Estonian company Cybernetica Ltd. was mandated with the development of the e-voting system. The system includes the use of smart cards and electronic signatures. In late 2004 the first test of the whole e-voting system took take place during a consultative referendum in the capital city of Tallinn.

The Estonian Internet voting system offers various ways of voter identification:

1. ID card with PIN codes. The system requires PIN codes, PC with Internet access and a smart card reader as well as ID card software
2. Digital ID (document which allows identification of a person in the electronic environment and signing with digital signature)
3. Mobile ID. Requirements: mobile ID SIM card with PIN codes and certificates, PC with Internet connection, mobile phone (no card reader or special software is needed).

Voters can test the e-voting system at www.valimised.ee in order to check whether they have the appropriate software and identification device.

In 2005, Internet voting was used in municipal elections (more than nine thousand voters cast their vote via the Internet) and since then, legally binding remote Internet voting is offered as an additional voting channel for all elections. Remote Internet voting was thus applied in the 2005 municipal elections and after that in further municipal elections (2009 and 2013), in national parliamentary elections (2007 and 2011), European Parliament elections (2009 and 2014).
The basic protocol has remained essentially unchanged. Voters can cast their vote via the Internet from the 10th to the 4th day prior to Election Day. This is necessary in order to ensure there is time to eliminate double votes by the end of the Election Day. A voter may change his/her electronic vote during the advance-voting period by casting another vote electronically or by voting at a polling station by paper. The paper vote takes precedence over the electronic vote. On election day the electronic vote cannot be changed anymore.

In USA [29], DRE (direct recording electronic) systems were developed by Frank Thornber Company in Chicago. It uses one of three basic interfaces (buttons, touchscreens, or dials) to record votes into the computer's memory. Some DREs are VVPAT (voter verified paper audit trail) compatible, meaning the DRE is connected to a printer to allow the voter to verify his or her votes before the votes are saved in the computer's memory. The paper records are also kept and may be presented for audit or recount depending upon state election codes.

Figure 2.11: A DRE Voting Machine [30].

In [31], An EVM which consists of two units, a control unit, and the balloting unit was designed for Indians elections. The two units are joined by a five-meter cable. Balloting unit facilitates voting by a voter via labeled buttons while the control unit controls the ballot units, stores voting counts and displays the results on 7 segment LED displays. The controller used in EVMs has its operating program etched permanently in silicon at the time of manufacturing by the manufacturer. No one (including the manufacturer) can change the program once the controller is manufactured. The control unit is operated

by one of the polling booth officers, while the balloting unit is operated by the voter in privacy. The officer confirms the voter's identification then electronically activates the ballot unit to accept a new vote. Once the voter enters the vote, the balloting unit displays the vote to the voter, records it in its memory. A "close" command issued from the control unit by the polling booth officer registers the vote, re-locks the unit to prevent multiple votes. The process is repeated when the next voter with a new voter ID arrives before the polling booth officer.

The EVMs are powered by an ordinary 6 volt alkaline battery [31] manufactured by Bharat Electronics Limited, Bangalore and Electronics Corporation of India Limited, Hyderabad. This design enables the use of EVMs throughout the country without interruptions because several parts of India do not have the power supply and/or erratic power supply. The two units cannot work without the other. After a poll closes on a particular election day, the units are separated and the control units moved and stored separately in locked and guarded premises.

In [4], we see the Kaduna State Local Government Area election of 2018. The Voter Verifiable Paper Audit Trail (VVPAT) Electronic Voting Machine (EVM) Model number EMP2710 was built specifically for KAD-SIECOM (Kaduna State Independent Electoral Commission) by Chinese based EMPTECH; the same company that built handheld PVC scanners for the 2015 Nigerian presidential elections. Weighing in at 12kg, the EVMs are boxlike devices shaped like medium-sized printers.

They feature 1.8GHz quad-core processors, 2GB RAM, 8GB ROM, 12.2-inch LED backlit touch display, USB 2.0, fingerprint scanner, SIM and PSAM (Secure Access Module) slots, 13,000mAH batteries and run on Android 5.1.

On the election day, voters go to respective polling units and get accredited with their permanent voters cards (PVCs). Afterwards, they electronically vote their chosen party and accompanying candidate by selecting and pressing the appropriate icon on the EVM screen.

When the voting exercise ends, an electoral officer brings out printed ballot papers from the machine for manual counting among party agents and officials [4].

2.3 Summary of the Reviewed Literature

The e-voting systems we reviewed above were all developed out of quest to enhance the voting systems to meet the recent technological frame and as well provide a means to uphold a credible election. The different e-voting systems has different level of adoption of technology in them as seen fit by the authors and the users.

Some e-voting systems use a single device with different authentication technology, while some implement two for more devices for authentication and vote casting purposes. Based on the technology available and the interest of the users, an e-voting device that works well is developed and there is always room for constant upgrade and development as technology advances.

The common criticism about the e-voting system is the issue of software security [32], and this has poised a serious threat to the adoption of an e-voting system. In the quest to add a physical assurance to the e-voting systems, the VVPAT (Voter Verifiable Paper Audit Trail) was advocated to be adopted to allow for manual counting of votes also at the end of the election.

2.4 Literature Gaps

The concept of e-voting systems has its focus on eliminating the issues of multiple voting and other election malpractice associated with the conventional ballot paper voting. Though many works have been done on the area of e-voting and many countries has adopted it for different levels of election, none of them was designed to completely captured the election process in Nigeria. The solution to the issues of voters traveling from one location to another where their vote would count on the Federal level elections is one missing puzzle to the existing e-voting systems.

CHAPTER THREE

METHODOLOGY AND SYSTEM DESIGN

3.1 Methodology

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. The aim of this chapter is to give an introduction about the general research methodology and waterfall methodology for development used in this project.

3.1.1 Research purpose

In the information age, it seems that the application of information technology is an in-dispensable tendency for the evolution of organizations in 21st century, regardless of public or private organizations. The application of information technology into public affairs briefly includes the electronic democracy, which is governance-oriented, and e-government, which is service-oriented. E-Voting being a vital part of the services being offered by e-Government would lead the application of information technology to improve the efficiency of public sector obviously and the participation of the citizen through the electronic forum.

The purpose of this research is to identify the factors affecting the election process in Nigeria and ways they can be eliminated.

3.1.2 Research approach

There are two main research approaches used in scientific work, quantitative and qualitative. The main difference between these two is that the aim of quantitative research is to find explanation to a phenomenon or a situation that can be generalized to other people and places while in qualitative research the aim is to gain deeper understanding of a phenomena or a situation.

Quantitative approach will be used to discover the issues that threatens the election process in Nigeria especially as it relates to voters.

We are making use of existing data already collected by previous literature on Nigerian elections to analyze the election process and derive a conclusion on how to eliminate the issues.

3.1.3 Research conclusion

Based on the reviewed data collected on previous conducted elections in Nigeria, the main issue with the Nigerian election was the issue of voters apathy towards the electoral system which is as result of

many factor such as inaccessible registration and voting venue, election violence that could lead to loss of lives, result manipulation and so on.

A system that serves to increase voters participation in the electoral process is the remedy to these issues at hand.

E-voting system serves to provide a remedy for the inaccessible registration and voting venue as eligible citizens can be registered and vote at their place of residence for their votes to count for their particular place of origin. It also provides a means to eliminate ballot box snatching as votes are counted as they are cast. There is also less room for result manipulation because the result get updated and displayed to all as votes are being counted.

3.1.4 Waterfall Development Method

Several system development methodologies are suitable for this project, they are briefly discussed below.

1. Evolutionary Process Model: This methodology is for projects using new technology that is not well understood. It best used when its is not necessary to produce a minimal version of the system quickly.
2. Rapid Application Development (RAD) model: Here, user involvement is essential throughout the process. It emphasizes working system and user feedback over strict planning and requirements recording.
3. Prototyping Model: In this methodology, all functionality must be delivered at one time even though the project's requirements are unstable or not well understood at the beginning. It is suited for medium and complex projects.
4. Waterfall Model: Here, the requirements and their implementations are well understood, the stages do not overlap and must be followed sequentially. It reinforces the notion of 'define before design' and 'design before implement'.

Based on the guideline below a system design methodology was chosen.

Table 3.1: Selection based on project requirement and type of project with associated risk

	Waterfall	Prototype	Evolutionary development	RAD
Project Requirements				
Are requirements easily	✔ Yes	No	No	✔ Yes

understandable				
Do we change requirements quite often?	✔ No	Yes	✔ No	✔ No
Can we define requirements early in the cycle	✔ Yes	No	✔ Yes	✔ Yes
Requirements are indicating a complex system to be built	✔ No	Yes	Yes	✔ No
Project type and risk				
Project is the enhancement of the existing system	✔ No	✔ No	Yes	Yes
Funding is stable for the project	Yes	Yes	✔ No	Yes
High reliability requirements	✔ No	No	Yes	✔ No
Tight project schedule	✔ No	Yes	Yes	Yes
Use of reusable components	No	✔ Yes	No	✔ Yes
Are resources (time, money, people, etc) scarce?	No	✔ Yes	No	No

It is noted that the waterfall model satisfies majority of the project requirement and is thus chosen.

3.1.5 Steps for Waterfall Model

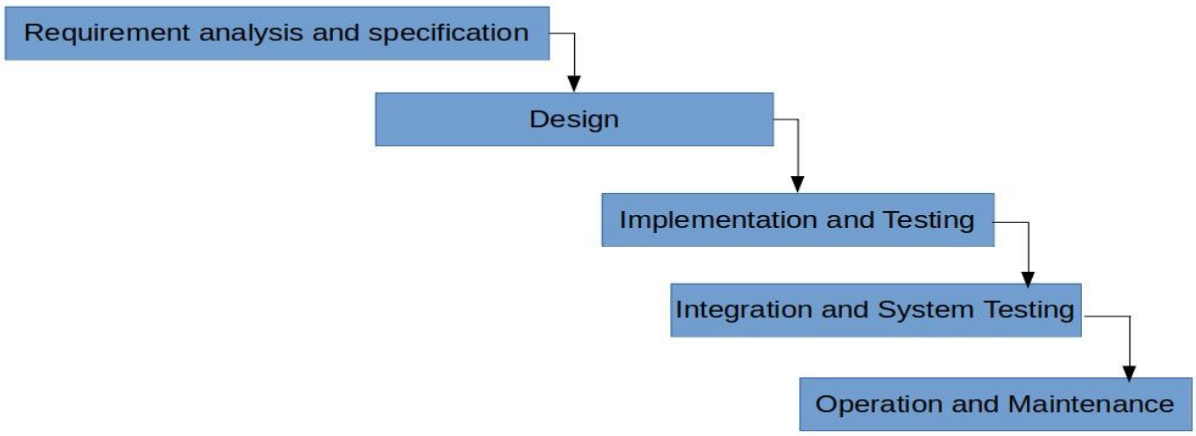

Figure 3.1: Steps of a Waterfall model.

From figure 3.1 above, the steps for a waterfall model can be briefly explained as:

Requirements analysis and specification: The first phase involves understanding what needs to design and what is its function, purpose, etc. Here, the specifications of the input and output or the final product are studied and marked.

System Design: The requirement specifications from the first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

Implementation and Testing: With inputs from system design, the system is developed and constructed and tested.

Integration and System Testing: The system is deployed and tested in the environment of operation.

Operation and Maintenance: This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance.

3.2 Requirements Analysis and Specification

The requirements for an e-voting system to be developed to model the Nigerian election at the Federal level and provide a remedy to voters' apathy is itemized below as:

1. The system should contain four subsystem as listed below:
   i. An electronic device with fingerprint and smart card reader for authentication and software for vote casting and internet access for vote sending.
   ii. A software platform for administrative management purpose.
   iii. A software platform for result display.
   iv. A database application for storage of election data.
2. The system should allow voters to be able to register against their place of origin from any place of residence or convenience.
3. The system should permit voters to be able to vote for their place of origin from any place of residence or convenience.
4. The registration and collection of voters card should be done at once.
5. The system should collect voters fingerprints for authentication purpose.
6. The system software platform should be accessible to all.
7. The system administrative dashboard for administrators should have access control.
8. The system electronic device should have a local database for verification purpose.
9. The system electronic device should have a built-in battery storage capability.

10. The votes should be transmitted over a secured wireless protocol.

A use case diagram is used to capture the e-voting system requirements as shown in figure 3.2 below.

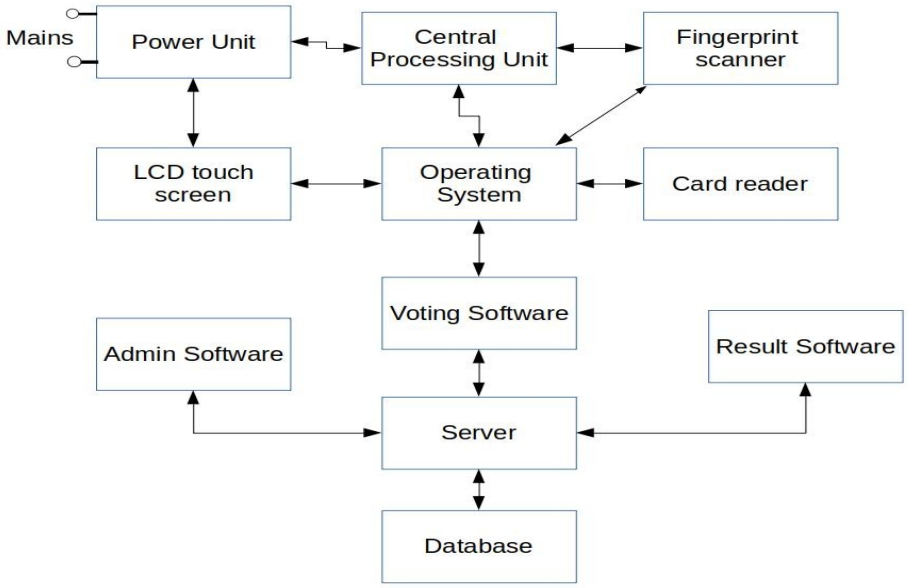

Figure 3.2: Use case Diagram for the E-voting System

3.3 System Design

The design of the system to meet the requirements above is depicted in the system block diagram as shown in figure 3.3 below and the system flowchart is shown below too in figure 3.3.

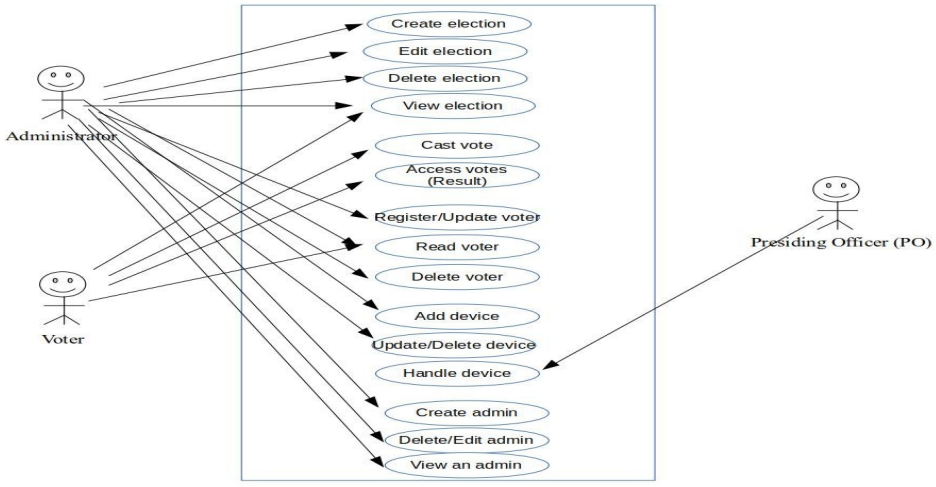

Figure 3.3: System Block Diagram

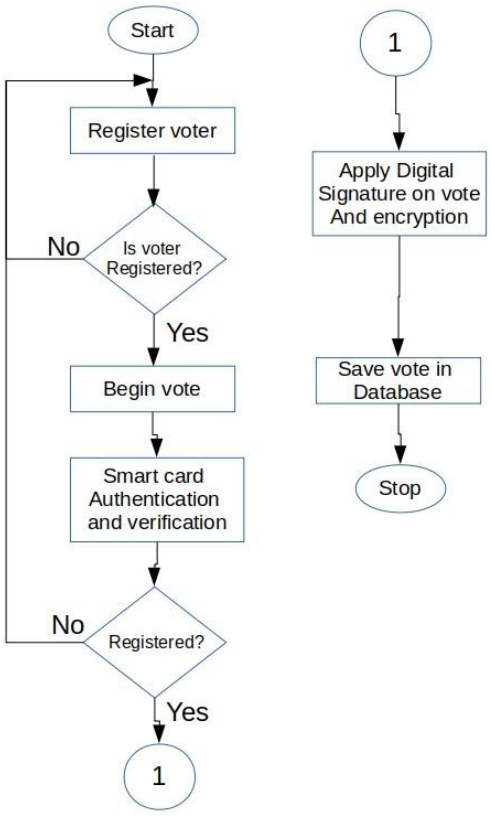

Figure 3.4: System Functional Flowchart

As can be seen above in figure 3.3, the system block diagram can be divided into;

1. Software design
2. Hardware design

3.3.1 Software Design

The software consists of all the software platforms needed for the system functioning and their interactions. The figure 3.4 below shows a block diagram of the software part of the system.

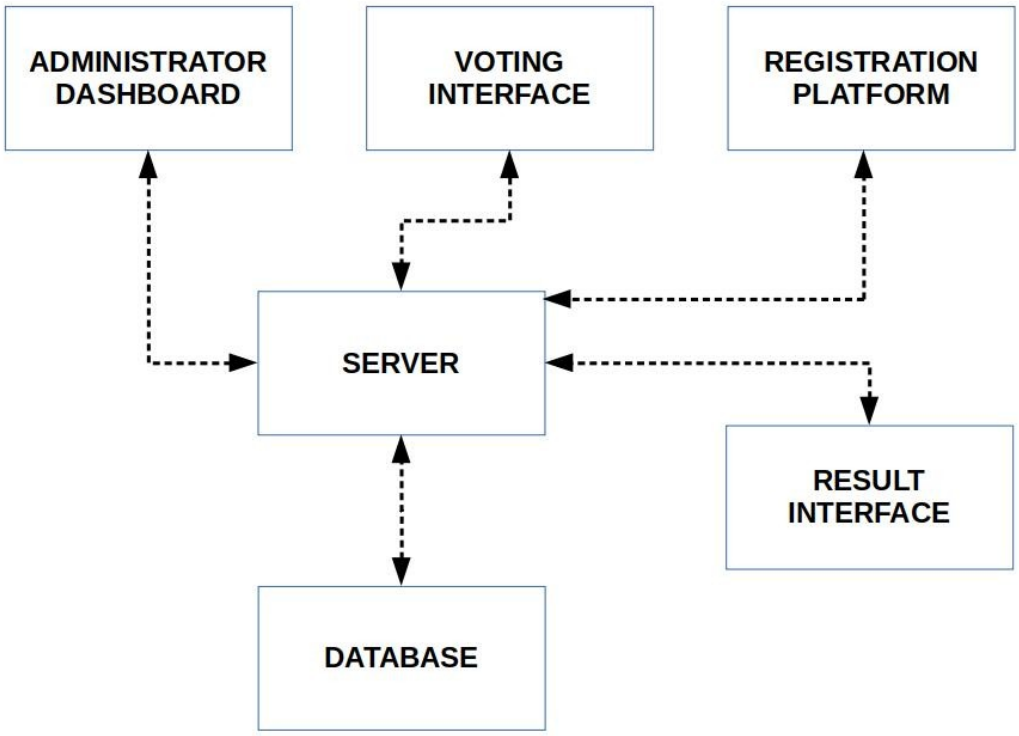

Figure 3.4: Software Design Block Diagram

3.3.1.1 Administrator Dashboard

The administrator dashboard is a desktop software application managing the elections. It incorporates the basic CRUD (Create, Read, Update and Delete) features for monitoring and managing voters, administrators, elections and the voting devices. It accesses the server which for its data that are contained in the database. The administrator dashboard has a login page as shown below to restrict unauthorized access to the dashboard. The login will require administrator email and password for authentication and validates it against the values in the database. If validation succeeds, access to the dashboard is granted, else he/she is restricted. The flowchart for the admin dashboard operation is shown below.

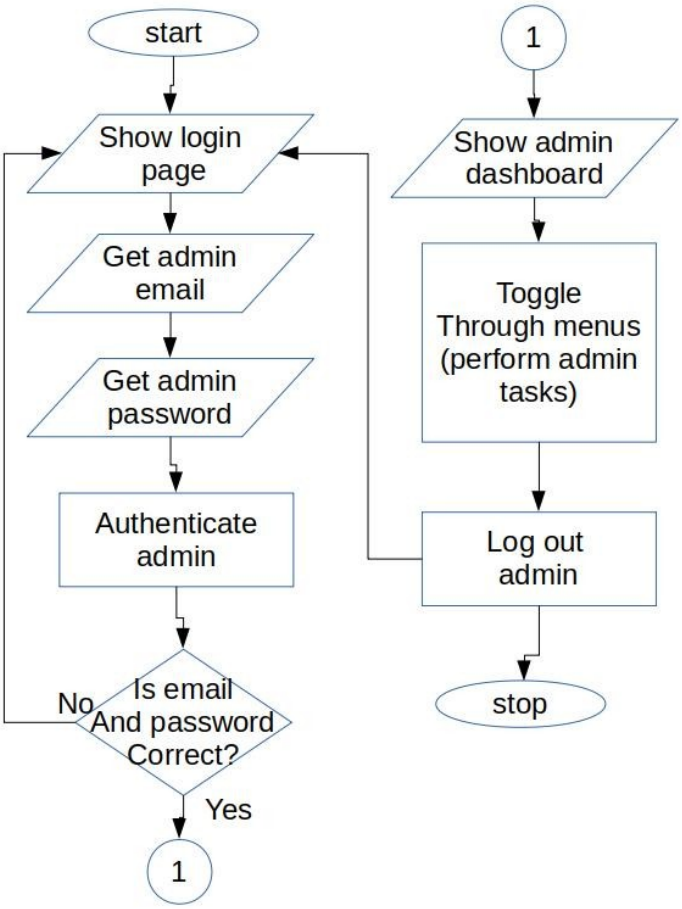

Figure 3.5: Flowchart of Administrator Dashboard

The Administrator dashboard has four sections of which basic CRUD operations can be performed on their corresponding data. These sections are:

- Voters
- Admins
- Election
- Device

3.3.1.2 Registration Platform

This is where the voters are registered prior to an election. The setup of the registration platform consists of a computer running Windows operating system and must have the following peripherals:

- A card reader

- A fingerprint scanner
- A camera

The registration software application has features the enables voters registration, which involves collecting voters data and storing the data in the database through the server. It interfaces with other built-in programs to be able to interact with the fingerprint scanner, card reader and camera hardware used for capturing different user data. The program flowchart is shown below.

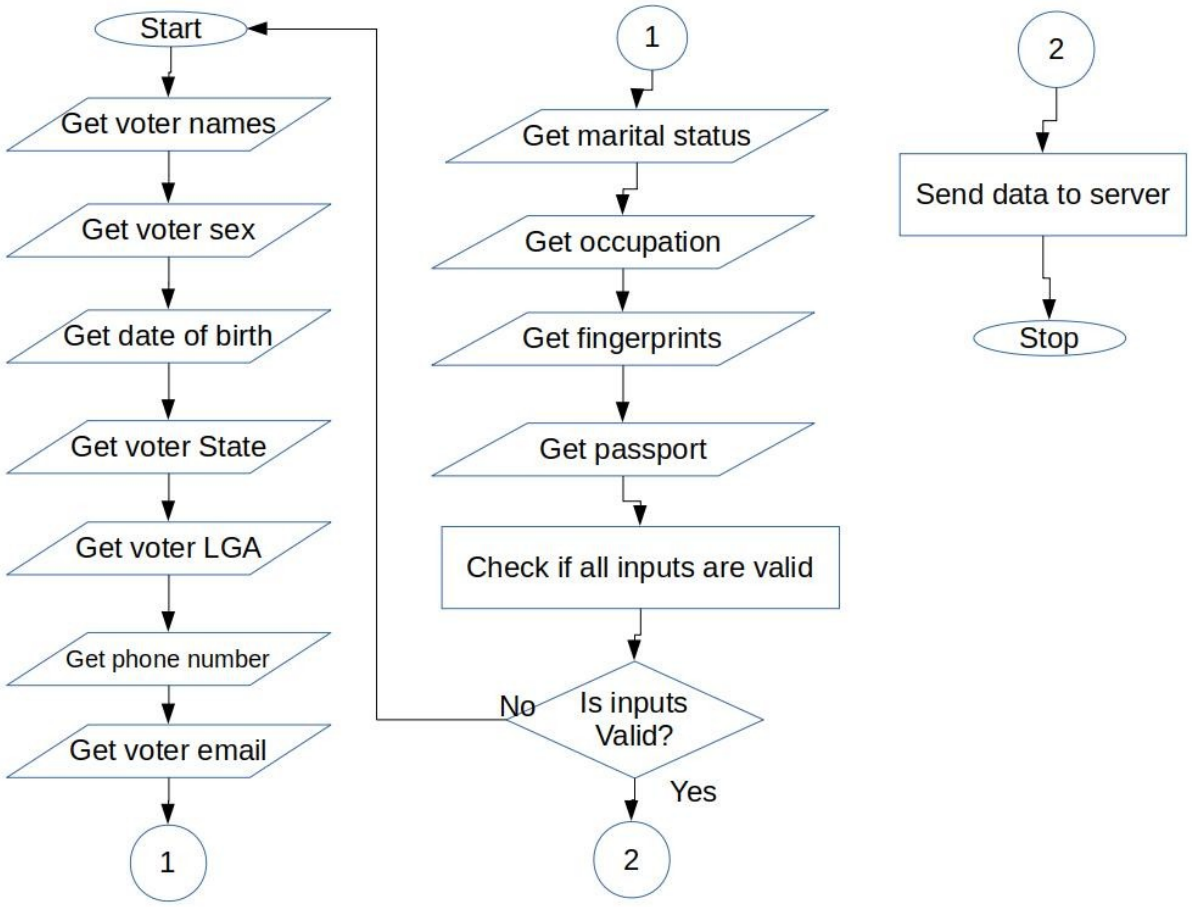

Figure 3.6: Flowchart for Voter Registration

The Registration platform requires the following voter details listed below

1. Voter names
2. Gender
3. Date of Birth
4. State

5. Local Government Area (LGA)
6. Phone number
7. Email
8. Marital status
9. Occupation
10. Fingerprints
11. Passport photograph

3.3.1.2 Voting Interface

The voting interface is the software that will run on the electronic device. It works with other programs to interact with the fingerprint scanner and the smart card reader. The flowchart for the voting interface is shown below.

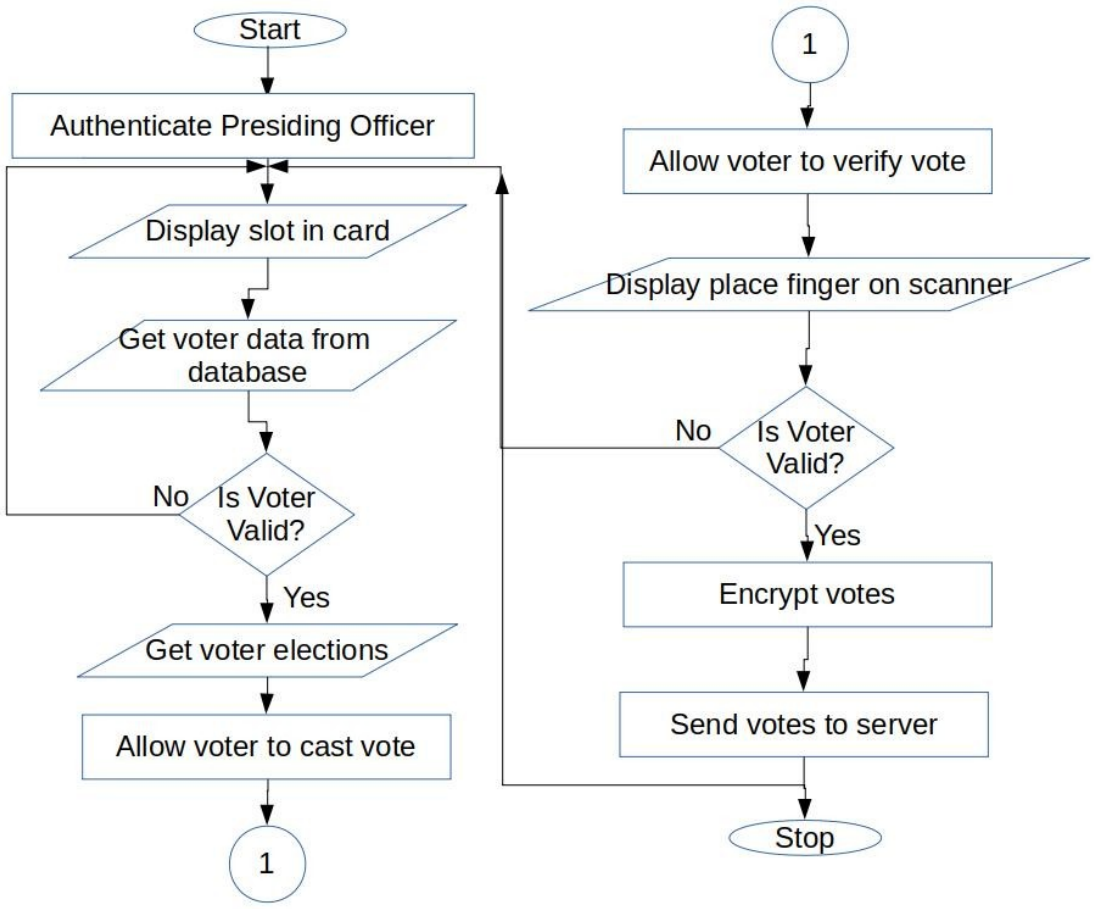

Figure 3.7: Flowchart of Voting Interface

3.3.1.4 Result Interface

The result interface design is such that anyone can have access to election results, hence no authentication is required to access this service. Nevertheless, data is transmitted over secure protocols to insure integrity of the results being shown. The interface makes use of bar graphs and chats to show live election results. The result interface program flowchart is shown below.

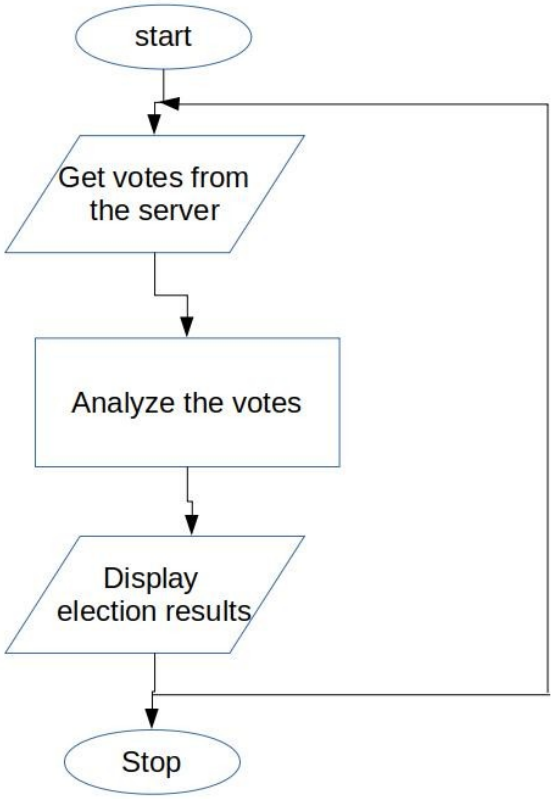

Figure 3.8: The Result Interface Flowchart

3.3.1.5 Server

The server is designed with a REST API architecture. REST in the sense that the server is stateless as it does not store data, it only defines certain protocols for which data in the database can be stored, accessed, modified or removed. The server will use access tokens for security while sensitive user data

like passwords will be hashed before being stored in the database. A flow block diagram of the server is shown below in figure 3.9.

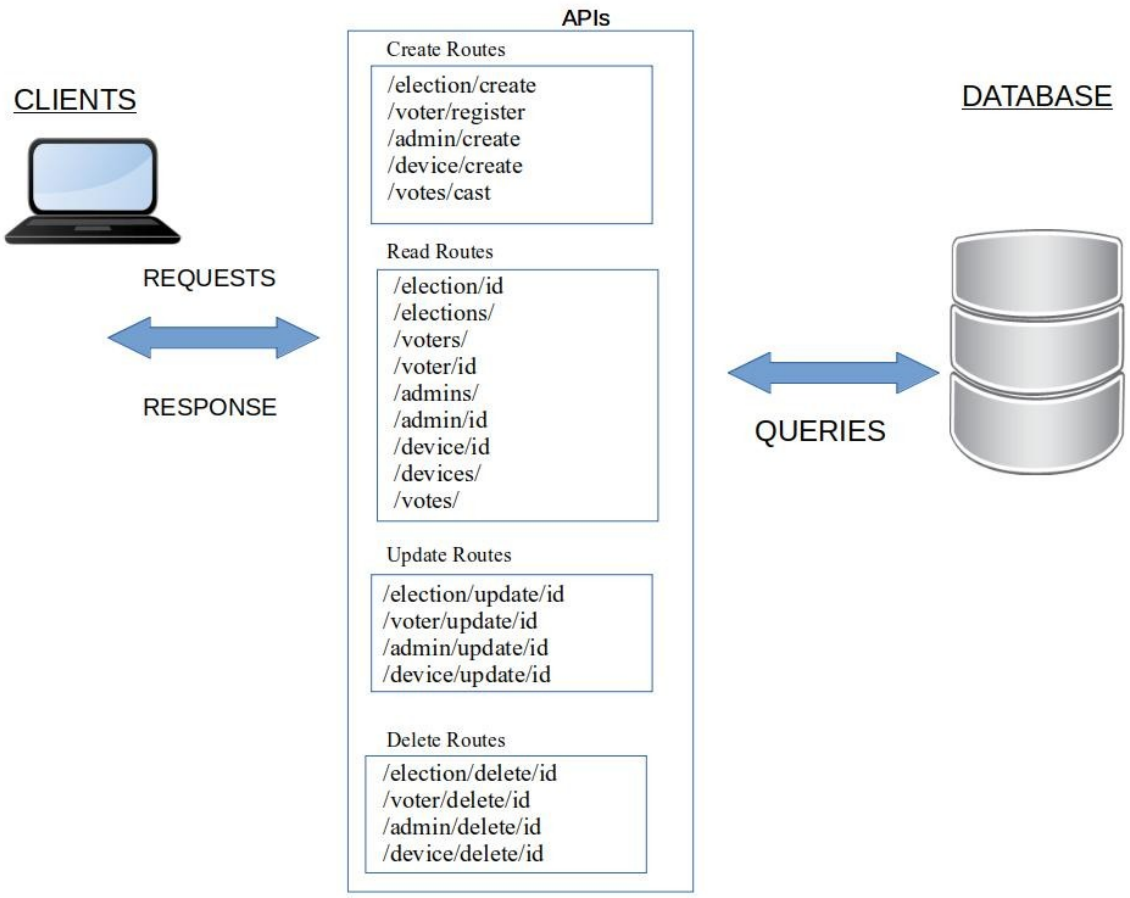

Figure 3.9: Flow Block Diagram of the Server

3.3.1.6 Database

A NoSql database which is an object-relational database is designed. This approach is used as election data is better represented as objects and it still maintains the advantage relational mapping of data brings. The Entity relationship diagram (ERD) shown below in figure 3.10 presents a schematic view of the e-voting database.

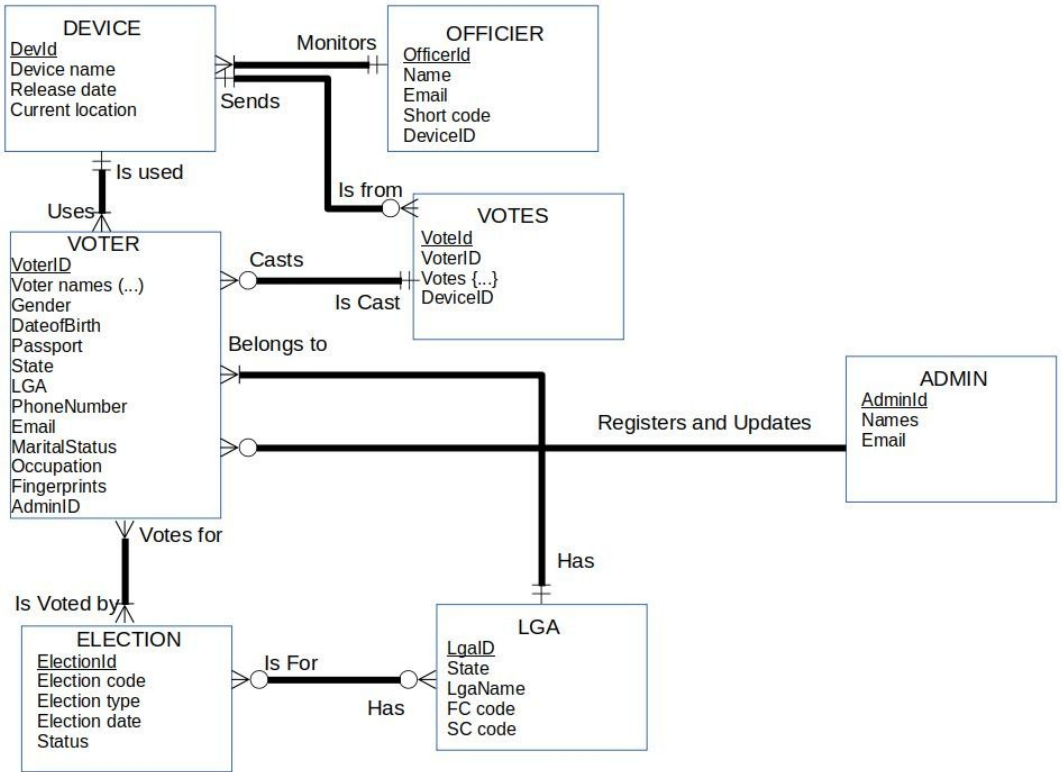

Figure 3.10: E-voting System ER Diagram

3.3.2 Hardware Design

This focuses on the hardware aspect required by the e-voting System Requirements Specification (SRS) the proper functioning of the system. The block diagram of the hardware design is shown below in Figure 3.11. All hardware sections as shown in the block diagram interacts with the System-on-chip (Raspberry Pi), which monitors and controls them.

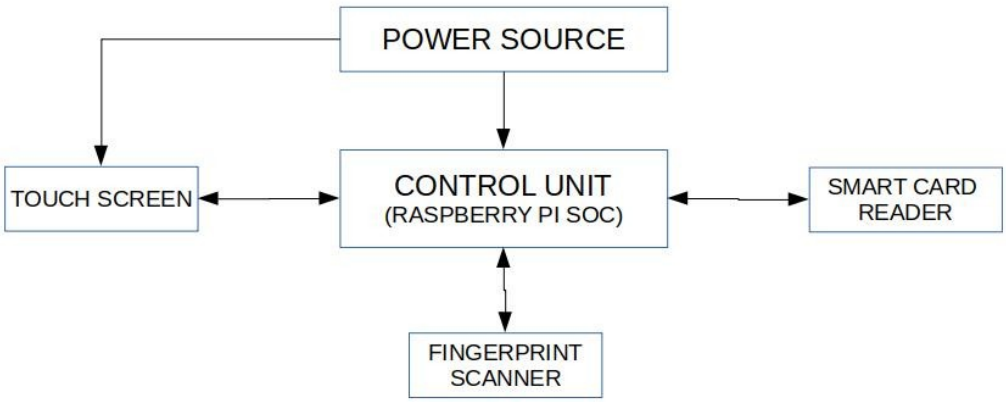

Figure 3.11: Block Diagram of the Hardware Design

3.3.2.1 Power Source

It is a 12V, 5A power supply that supplies power to the rest of the devices. It contains a battery for supplying power in the absence of external power. This unit is responsible for providing electrical power required by the other hardware units. It includes a battery for power storage as the other hardware unit receives power from it. The designed circuit diagram of the power source unit is shown below in figure 3.12.

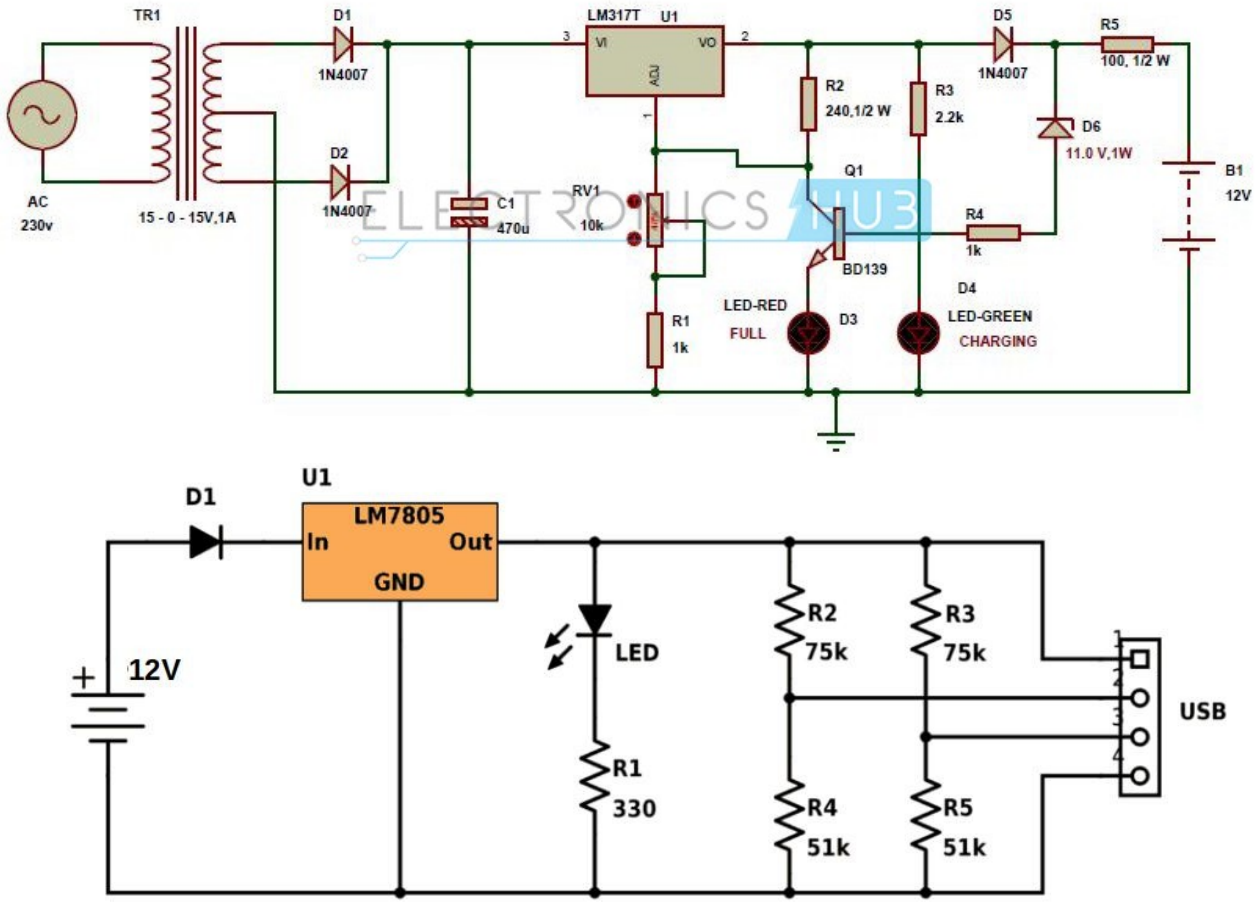

Figure 3.12: Circuit Diagram of the Power Source.

Calculation To Determine The Battery Capacity

Batteries are typically rated in mAh meaning milli-ampere hour and is a unit that measures (electric) power over time. This metric can be used to provide an idea of how long a device will last, given a constant (or average) power draw rate. For instance, a 3000mAh battery could power a device taking 100mA (milli-ampere) for 30 hours. A device using 200mA would last only 15 hours. In general, the more mAh and the longer the battery capacity or battery life.

Given the current rating and expected number of hours of usage of the battery, the expected mAh, of a battery can be calculated as thus:

$$X = H \times C \times 1000$$

where X = battery capacity in mAh

H = expected number of hours

C = current rating of the device in ampere

The current rating of the E-voting device is 3A. A typical Nigerian election lasts for 6 hours, therefore we would need a battery that can power the device for up to 7 hours. The mAh of this battery would be:

$$X = 7h \times 3A \times 1000$$

$$X = 21000mAh$$

This means that the battery to be used by the voting device should have a rating of 21000mAh.

3.3.2.2 Control Unit

The control unit is the central processing unit. It consists of raspberry pi model 3 which runs the voting software while interacting with the other peripherals.

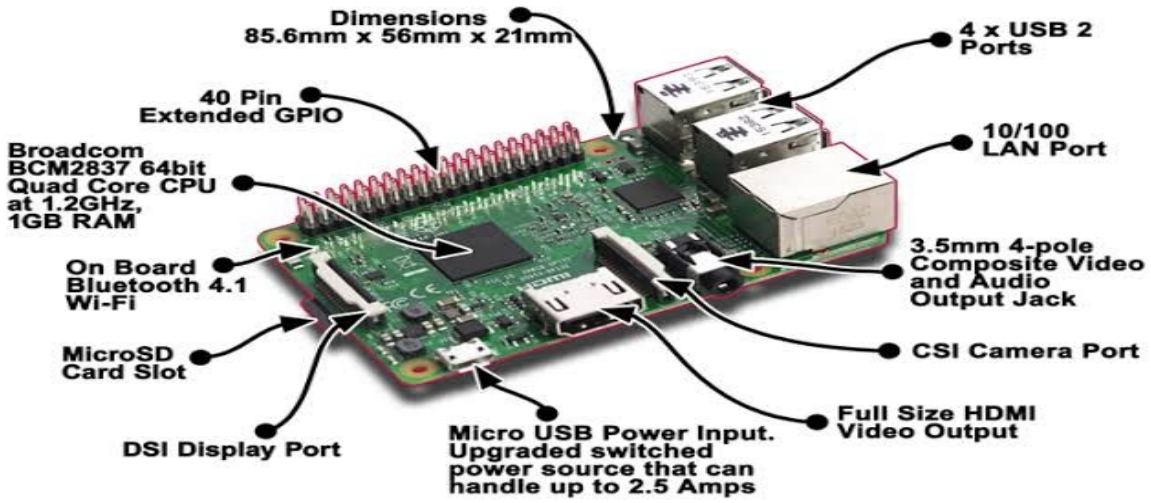

Figure 3.13: Raspberry Pi (Control Unit).

3.3.2.3 Touch Screen

The touch screen is a resistive LCD screen with a touch controller. The controller renders the graphics output of the raspberry pi to the touch screen while sending touch responses from the screen to the raspberry pi. The touch screen has an impressive response time and the specifications are shown below in table 3.2.

Table 3.2: Recommended LCD Touch Screen Specification

Property	Value
LCD size	9” inch
Power supply	5V-12V DC
Display size	7~10 inch
Screen type	Resistance screen
LCD resolution	800*480

3.3.2.4 Fingerprint Scanner

This is one out of the two factors of authentication for the voting device, providing strong security and confidence on a voter’s vote. The fingerprint scanner enables the fingerprint of the voter to be read for verification or identification of the voter.

Table 3.3: Recommended Fingerprint Scanner Specification

Property	Value
Interface	UART (TTL)
Voltage	4.2-6.0V DC
Resolution	508 DPI
Sensing area	160*160 pixel
Fingerprint capacity	200
Module Size	33.4*20.4 mm

3.3.2.5 Smart Card Reader

This is the second factor of authentication provided by the system. The card reader retrieves the voters’ details stored on the voters’ card fro identification and security.

Table 3.4: Smart Card Reader Specification

Property	Value
Card type	IC/ID smart card
Card slot	single
Interface	USB

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND RESULT ANALYSIS

4.1 System Implementation and Unit Testing

This chapter explains the implementation of the e-voting system to with respect to the system requirement specification (SRS) document. The system implementation is sub divided into:

1. System software implementation
2. System hardware implementation

4.1.1 Software Implementation and Unit Testing

The software section was implemented based on the design in the section 3.3.1. The software implementation is divided based on the block diagram of figure 3.4.

4.1.1.1 Administrator Dashboard

The admin dashboard was implemented with Electronjs, a framework for building cross platform desktop application with HTML CSS and JavaScript. The software was built and packaged for windows operating system. Its login page for restricted access is shown below in figure 4.1, while the main admin dashboard is shown in figure 4.2 below.

Table 3.5: Unit Test for Administrator Dashboard

Test	Steps	Expected Result	Test result
Install software	double click the installer follow the installation prompt	Software correctly installed	Software installed successfully
Login pass	input admin email input admin password click the login button	Show admin dashboard with welcome message with admin name	Admin dashboard shown
Login failed	input admin email input admin password click the login button	Display email or password invalid	Display email or password invalid
Display data	Automatically query the data from the server	Show data on the dashboard	Data shown on the dashboard
Log out	Click log out button	Send admin to log in screen	Admin sent to login screen

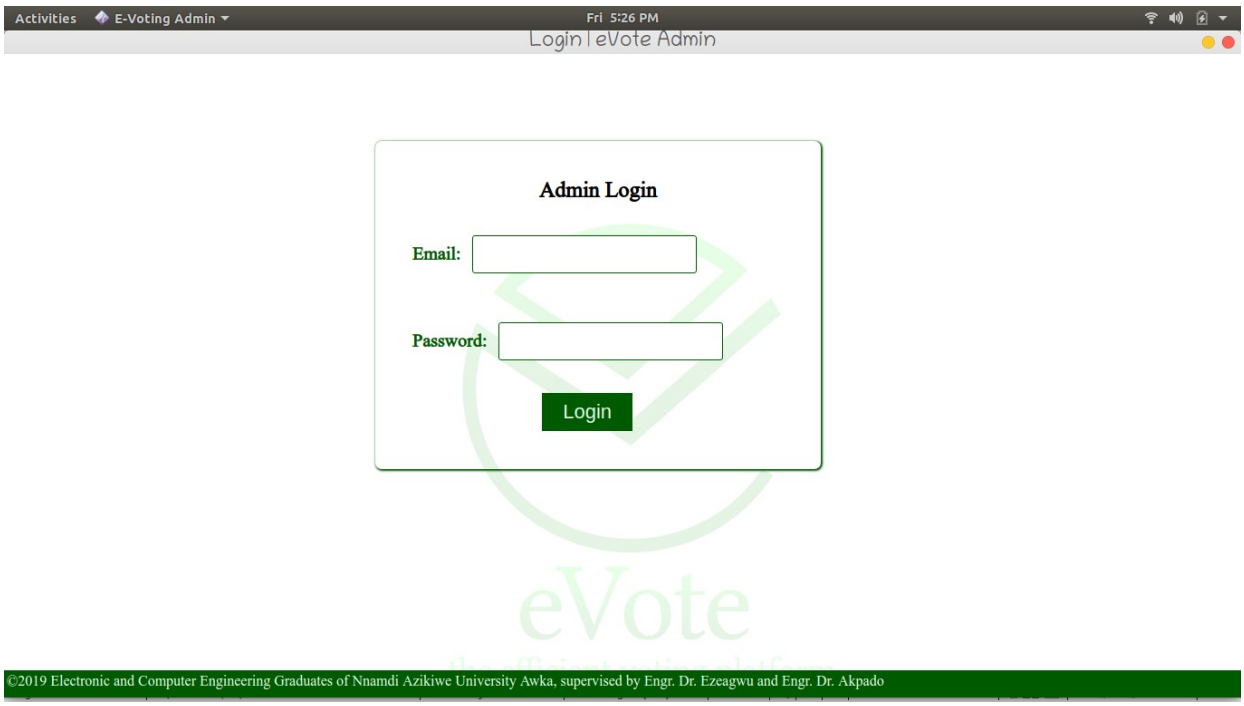

Figure 4.1: Administrator Login Screen

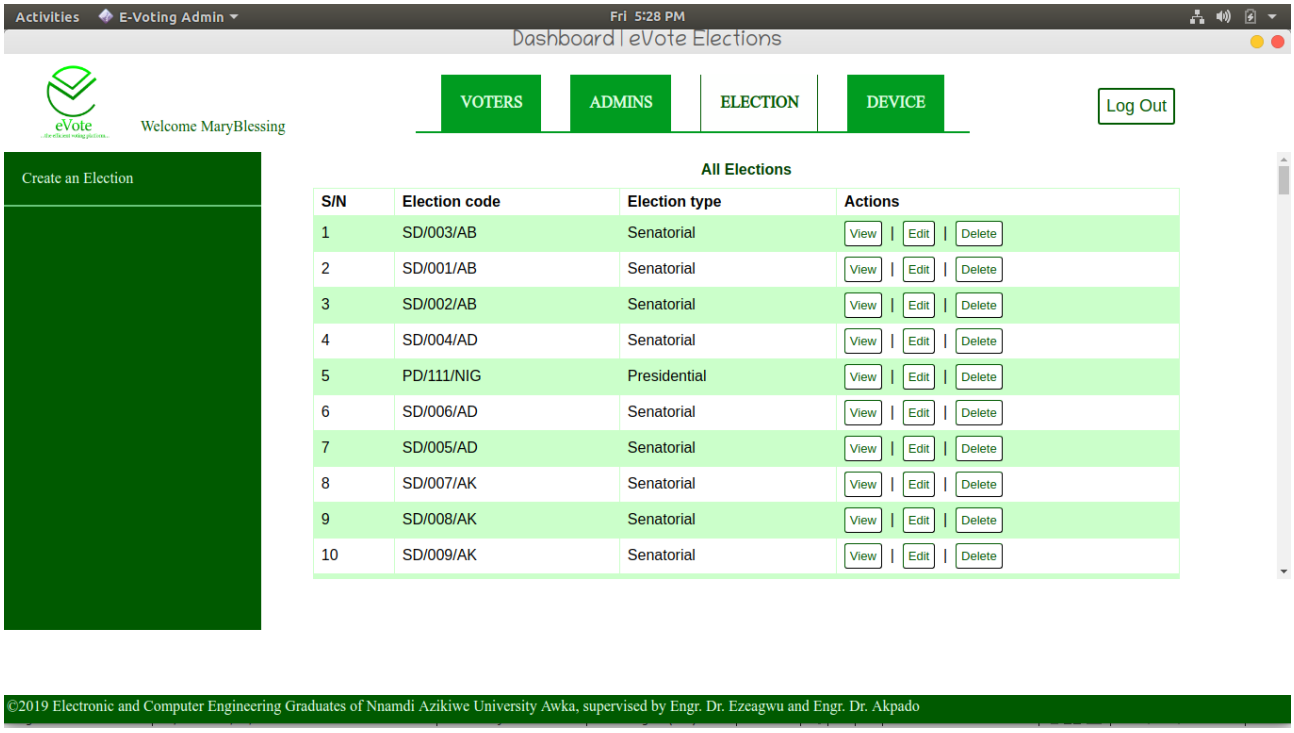

Figure 4.2: Administrator Dashboard Elections View

4.1.1.2 Voting Interface

The voting interface is written in Java which makes use of C++ and Python libraries to interface with the peripherals (card reader device and fingerprint scanner). Voting data is first encrypted before it is then transmitted over HTTPS to ensure security. The software code is contained in appendix A. The voting software implementation is shown below in figure 4.3.

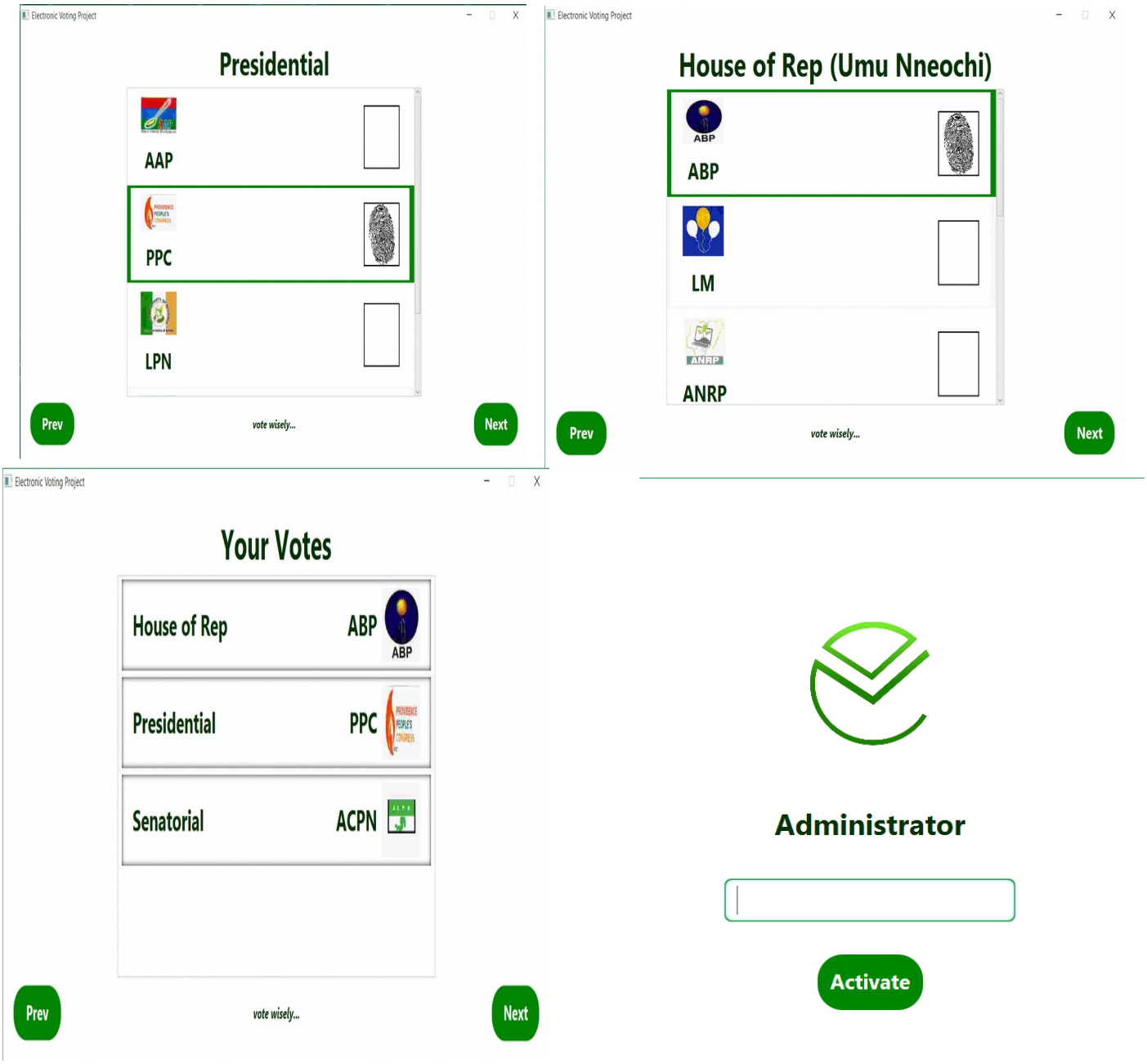

Figure 4.3: Voting Interfaces Implementation

Table 4.1: Unit Test for Voting interface

Test	Steps	Expected Result	Test result

Detect and read smart card reader	Plugin card reader to the raspberry pi. Run card software slot in the card	Card details gotten	Details gotten from laptop
Detect and read fingerprint scanner	Plugin fingerprint to the raspberry pi. Run fingerprint scanner software place finger	Fingerprints read	Fingerprint binaries read
PO Authentication	Put authentication code click authenticate	Show insert card screen	Insert card screen shown
Get voter detail	Get cardID from card Voter detail from server	Show voter detail	Voters' detail shown
Get voter election	Send request to the server Get voter election from server	Show voter elections	Voter elections shown
Cast vote	Click the selected party Click next button	Show Voted	Voted shown
Send votes to server	Get votes Encrypt votes Send encrypted votes to server	Votes sent	Votes sent

4.1.1.3 Result Interface

The result has two implementations: A desktop app built with Electronjs and a web portal built with HTML, CSS and JavaScript and hosted with Github Pages at (https://marybngozi.github.io/E-voting-result/index.html). The desktop app is built and packaged for windows operating system . These interfaces receive result data over through HTTPS from the server this is to ensure security.

Table 4.2: Unit test for Result Interface

Test	Steps	Expected Result	Test result
Load result page	Type url in browser click enter	Show result interface	Result interface shown
Page speed	Open network tab	Load time less than or equal to 1.7s	Load time 1.65s

The result interface is shown below.

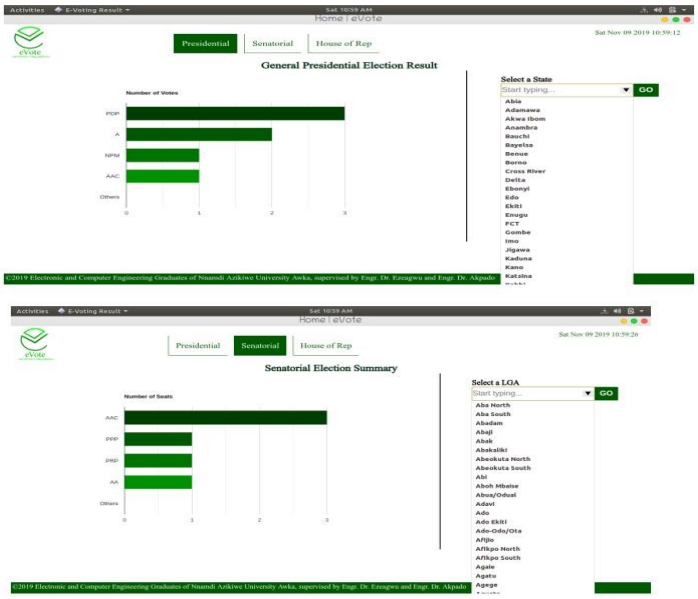

Figure 4.4: The Result Interface Implementation

4.1.1.4 Registration Interface

The registration software is implemented in Java and makes use of C++ and Python libraries to interface the system peripherals. All data is encrypted and transmitted over HTTPS to the server.

It begins with a login page to authentic the registrar.

Table 4.3: Unit test for Registration Interface

Test	Steps	Expected Result	Test result
Card reader test	Plug in card reader to the system Slot in card	Display smart card reader present	Displayed smart cart reader present
Camera module test	Click capture Position camera Take picture	Passport image will display	Passport image displayed
Read Fingerprint	Plug in fingerprint scanner select finger view place finger on scanner click capture	Fingerprint view updated(darkened)	Fingerprint view was updated
Register voter	Get all voter details Check for input validity Capture passport image Capture fingerprint	Display voter registered	Voter Registered Displayed

	Slot in a card Click register Send details to server Write detail to card		
Install registration software	Double click on installer Follow installation prompt	Software installed correctly	Software was installed correctly
Login registrar	Input registrar email Input registrar password click login	Display Registration Interface	Registration screen shown

The registration interface is shown below in figure 4.5.

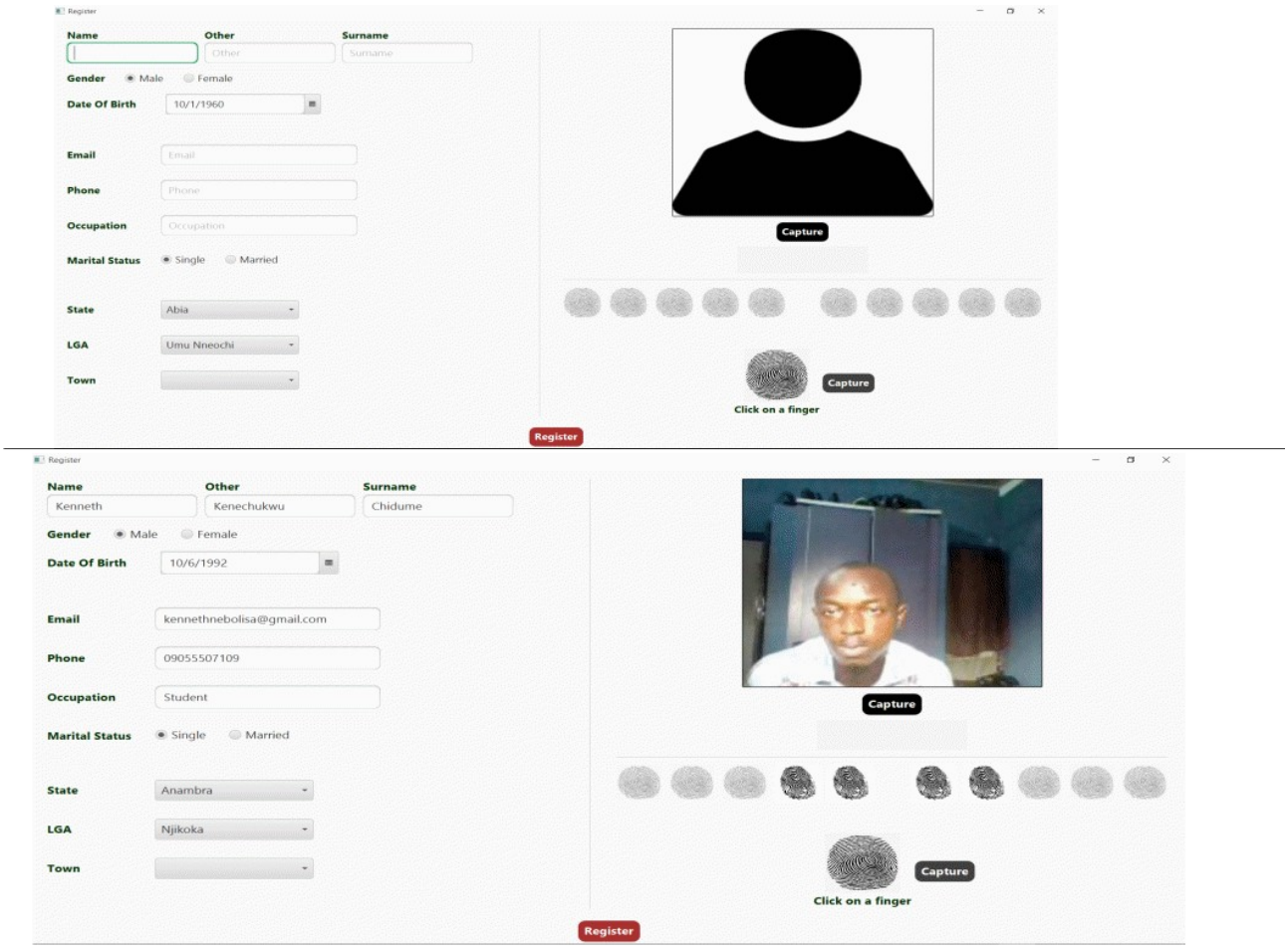

Figure 4.5: Registration Interface Screen showing empty field above and filled fields below.

4.1.1.5 Server

The server API is built with Nodejs, a JavaScript run-time outside the browser. The API routes are protected and can only be accessed via an authentication token. All sensitive data like passwords are

hashed before they are sent to the database for storage. The server is hosted on Heroku, a cloud based hosting service. The server was tested with Postman software application and the routes were documented using Postman docs.

Table 4.4: Unit test for Server

Test	Steps	Expected Result	Test result
Security	Make a request using Postman. Leave authentication token empty	Display unverified token	Displayed unverified token
Test for route consistency	Make a request to a route using Postman On received response Make another request to the same route	Response to a request from particular route is same for all request made	Same response no matter the number of times request was made
Latency test	Make a request using Postman Record response time and data size of data sent for each request made	Latency should be less than 48 for a good server	Latency was 43
Scalability	Add new routes Check if the former routes have issues	All routes both the new and old ones behave well.	All routes still functional
Deploy server online	Open the terminal type git push heroku master press enter	Display Build successful Deployed successful	Displayed Build successful Deployed successful

4.1.1.6 Database

A NoSql Object-Relational database was implemented using MongoDb and saved on Heroku server using mLab a MongoDB hosting service. Only the API has direct access to the database which it accessed with a database key on a secure production environment variable file hosted on Heroku.

Table 4.5: Unit test for the Database

Test	Steps	Expected Result	Test result
Security	Try to access the database without going through the server to verify	Access denial	Was denied access
Data integrity	Save data through the server Request for that data	The data is same as the one saved	Received same data, data was not damaged

Scalability	Increase the capacity of the database to accommodate more tables	The new tables were added and the old one wouldn't break	New tables added and other tables still intact
Latency	Fetch data from the database through the server, monitor the database response time		

4.1.2 Hardware Implementation and Unit Testing

4.1.2.1 Power Unit

This generates the power required for the devices to operate in a good working condition. It also powers some LEDs for indication purposes. It contains a 12V battery to supply power to the devices in the absence of external power.

The input to the power unit is a 220V AC which is stepped down and converted to DC. This output is used to power the touch screen module and the raspberry pi.

Table 4.6: Unit test for the Power Unit

Test	Steps	Expected Result	Test Result
Output voltage	Plug the power unit to an AC power source (220V-240V). Connect the probes of a multi-meter to the output end of the power source	The multi-meter should read 5 volts	4.9 volts of was read from the multi-meter

4.1.2.2 LCD Touch Screen

The LCD touch screen provides a means of interacting with the device. Its powered directly by a 5V power supply from the power unit.

The touch screen is connected to the raspberry pi via an HMDI cable for receiving video streams from the raspberry pi for display and a USB cable for transmitting input received from user touch to the raspberry pi. The controller directly communicates with the raspbian operating system running on the raspberry pi enabling input to the screen sensor to be interpreted properly.

Table 4.7: Unit test for Touch Screen

Test	Steps	Expected Result	Test Result
Screen Touch controller power	Connect the screen controller to the power source	The red LED on the controller should come on.	The red LED on the controller turned on.
Touch screen sensor power	Connect the touch screen to the touch controller while connected to power	The touch screen should come on.	The touch screen came on.
Touch screen sensor sensitivity/responsiveness	Connect the touch screen sensor cable to the raspberry pi USB port	A touch on the screen at +/-2mm should trigger a corresponding action.	A touch on the screen gave the corresponding mouse.

4.1.2.3 Control Unit

The control unit is the heart of the system. It is a raspberry pi model 3b running raspbian os (linux). The operating system provides the resources necessary to generate a graphical user interface for the application. It also provides low level libraries to enable easy integration with other peripheral (hardware) devices.

It communicates with the card reader, the fingerprint sensor and the touch screen via its USB ports which serves as a source of power to some of the peripherals like the fingerprint and the card reader.

The voting application or software written in java and python, runs on this operating system and communicates with the peripheral devices by using the low level libraries provided by the operation system.

Table 4.8: Unit test for Control Unit

Test	Steps	Expected Result	Test Result
Raspberry pi power	Plug the raspberry pi power cord to a power source	The red LED on the raspberry pi should come on	The red LED came on
Raspberry pi OS boot	Put the SD card in the raspberry pi and power on the raspberry pi	The green LED on the raspberry pi should come on	The green LED came on
Start Application	Attempt to run the vote application	The application should run without errors	The application ran without errors

4.1.2.4 Smart Card Reader

One of the means by which the device authenticates its users is through the smart card reader. The card reader is connected to the raspberry pi through a USB cable which also serves as its source of power. It interacts with the raspberry pi through the "T=0" protocol. Other protocols which may be used to interact with a smart card reader includes "T=1", "T=CL" or "DIRECT". The program for implementing this communication protocol was written in python and can be found in appendix B.

Table 4.9: Unit test for the Smart Card Reader

Test	Steps	Expected Result	Test Result
Power on	Connect the smart card reader to the raspberry pi USB port	The red LED of the smart card reader should turn on	The red LED of the smart card reader turn on
Card detection	Insert a card into the card reader	The green LED of the card reader should blink thrice	The green Led of the smart card reader blinked thrice

4.1.2.5 Fingerprint Scanner

This is the second means by which the system authenticates a voter. It exposes four pins of which two (the RX and TX) are for serial TTL communication while the other two provides the power supply. The fingerprint sensor is connected to the raspberry pi USB port through a TTL-USB converter which also provides enough voltage (5V) to power it. The fingerprint sensor is controlled by a python program which provides a wrapper for the low level libraries that communicates with it. The codes used for this communication is found in the appendix C.

Table 4.10: Unit test for the Fingerprint Scanner

Test	Steps	Expected Result	Test Result
Correct connection	Connect the pins of the finger print scanner to the appropriate pins of the raspberry pi	The yellow LED fingerprint scanner should blink and stay on steadily.	The yellow LED of the fingerprint blinked and stayed on.

4.2 System Integration and Testing

All the different units explained above where put together such that the fingerprint scanner, camera and smart cart reader writer for the registration was added to the registration platform running on a Windows system. The registration platform was also connected to the online Server.

The result website was hosted online at E-voting Result (https://bit.ly/32Y5z6q) and linked to the online Server.

The administrator dashboard was installed on a Windows system and linked to the online Server as well.

At the voting device end, the fingerprint module is coupled to the Raspberry pi, also the smart card reader and LCD touch screen is connected to the Raspberry pi and coupled into the voting system. The battery unit is added to the voting device too and the voting software is burnt to a memory card and inserted into the Raspberry pi memory card slot. The system is started up and the voting device is working.

Table 4.11: Overall System Testing

Test	Steps	Expected Result	Test Result
On/Off	Power the system on and off	On power ON, The system should correctly boot-up within 20 seconds. On power OFF, the system should shut down within 10 seconds	Expected result gotten, as system started within 15 seconds and was shut down withing 7 seconds.
Register a voter	Double click the registration software Login as a Registrar Fill all input fields Capture passport image and fingerprints Write details to card	Display Registration Successful	Registration Successful displayed
Cast vote	Authenticate as PO slot in card choose parties confirm votes with fingerprint	Screen should display ✔ “Voted ”	✔ “Voted ” Displayed
View election result	Visit https://bit.ly/32Y5z6q on the browser window.	Result interface should show	Result interface shown
Monitor changes on administrator dashboard	Double click administrator software icon Input admin email and password to login Click login	Administrator Dashboard should show with a welcome message for the admin	Administrator dashboard shown with “Welcome MaryBlessing”

Touch screen accuracy	Touch various parts of the voting device touch screen	The system should respond to all inputs within +/- 1.5mm of the center of the physical input	The system has 89% accuracy as it responses to almost all inputs within +/- 1.5mm of the physical touch point
Battery duration on system use	Use the voting device for some time without charging	The battery should last for____	The battery lasts as expected

4.3 Packaging

The voting device was coupled and packed in a plastic (PVC) square container of length 22 cm and breadth 12 cm. The plastic container was used to avoid any kind of shock and the container was also padded inside. The LCD touch screen is mounted in slant position on the top of the container and the fingerprint scanner is also mounted on the top of the container as well. The voting device weighs less 4kg. A diagram of the completed work is as seen in Appendix D.

4.4 Bill of Engineering Measurement and Evaluation (BEME)

The details of the costs incurred during the development of this project are shown in table 4.11 below.

Table 4.12: Bill of Engineering Measurement and Evaluation (BEME)

S/No	Item Description	Quantity	Unit Price(#)	Total Price(#)
1	Smart cards	30	310	9,300
2	Smart card reader writer	1	10,000	10,000
3	Smart card reader	1	7,000	7,000
4	Raspberry pi	1	25,000	25,000
5	LCD Screen	1	25,200	25,200
6	Touch screen	1	9,000	9,000
7	Fingerprint scanner	1	13,500	13,500
8	Fingerprint module	1	9,800	9,800
9	Way bill and shipping		20,000	20,000
10	8gb Memory card	1	1,700	1,700

11	HDMI cable	1	1000	1000
12	P-channel MOSFET IRF4905	3	300	900
13	6A10 diode	1	100	100
14	Led	6	10	60
15	USB port	1	200	200
16	Operational Amplifier	2	190	380
17	Resistor 220k	3	10	30
18	Resistor 100k	3	10	30
19	Resistor 1k	1	10	10
20	Resistor 560	1	10	10
21	Resistor 680	1	10	10
22	Variable resistor 10k	4	50	200
23	3.7V Li-ion battery	1	2000	2000
24	Voltage regulator LM7809	1	200	200
25	Voltage regulator LM7805	2	200	400
26	Power transistor D718	1	300	300
27	Plastic Casing	1	3000	3000
28	Tape- Abro tape	3	100	300
29	Adhesive (EverKing)	1	600	600
30	Switch- on/off switch	2	50	100
31	Bread board	1	200	200
32	Vero board	1	300	300
33	Jumper wires- ordinary	1	400	400
34	Jumper wires- female-female	1	800	800
35	Heat sink	3	830	2490
36	Screw driver set	1	1500	1500
37	Multi-meter	1	1300	1300
38	Soldering iron	1	1700	1700
39	Soldering led	1	300	300
40	Miscellaneous		50000	50000
41	Logistics		20000	20000
42	Server	1	7000	7000
43	Data subscription		5000	5000
44	Internet router	1	17,200	17,200
45	Total			248,520

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The manual system of voting in Nigeria has failed to tackle the basic issues necessary for a clean and trusted voting environment which has evidently driven some of its citizens to apathy.

The E-voting system was implemented to solve the proximity bottlenecks, unnecessary time delays, with very secure and accurate recording of votes. The system has been thoroughly tested in voting accuracy, ruggedness, responsiveness, battery life expectancy, and security by means of simulation and mini voting sessions to be a successful one.

It is seen that the system is fault tolerant at all end points (registration, voting platform and the server). The voting device can last for more than 6 hours which is very sufficient for a quick system like ours.

This system will provide boundless voter participation in remote areas with very little or no cost on the voter greatly reducing apathy. Further improvements can be done on the system to increase the credibility of the votes and further reduce proximity issues.

5.2 Recommendation

The following recommendations are made for optimal performance of the system:

1. The voting device should be operated in a dry environment with a fairly stable internet connection.

The following functionalities could be added to improve on the project:

1. Internet Voting (I-voting): the use of smart phones or any internet connected device to cast votes from any location.
2. The registered cards could be integrated into other areas of citizenship authentication and identification such as drivers' license and e-governance.

5.3 Contribution to Knowledge

Many works have been done with respect to making the electoral process better by increasing voters' interest to participate in the election especially in Nigeria, and based on these existing solutions, this project model introduces the concept of voting at the closest polling unit while vote is counted where it belongs.

REFERENCES

[1] Paul David Webb, Roger Gibbins, Heinz Eulau, “Election”, Encyclopaedia Britannica. [Online]. Available: https://www.britannica.com/topic/election-political-science. [Accessed: Aug. 05, 2019].

[2] Toba Paul Ayeni, Adebimpe Omolayo Esan, “The Impact of ICT in the Conduct of Elections in Nigeria”, American Journal of Computer Science and Information Technology, February 09, 2018 . [Online]. Available: http://www.imedpub.com/articles/the-impact-of-ict-in-the-conduct-of-elections-in-nigeria.php?aid=22211. [Accessed: Aug. 05, 2019].

[3] ACE, E-voting, The Electoral Knowledge Network, n.d., [Online]. Available: http://aceproject.org/ace-en/focus/e-voting/default. [Accessed: Aug. 07, 2019].

[4] Victor Ekwealor, Inside Nigeria’s first ever electronic voting exercise in Kaduna State, Techpoint Africa, May 14, 2018, [Online]. Available: https://techpoint.africa/2018/05/14/kaduna-electronic-voting/. [Access: Aug. 10, 2019].

[5] Seth Rosenblatt, Jason Cipriani, Two-factor authentication: What you need to know (FAQ), CNET, June 15, 2015, [Online]. Available: https://www.cnet.com/news/two-factor-authentication-what-you-need-to-know-faq. [Accessed: Aug. 10, 2019].

[6] Alexandra Petruș, What is two-factor authentication (2FA)?, iPhone Backup Extractor, Oct. 08, 2017, [Online]. Available: http://www.iphonebackupextractor.com/blog/2016/jun/3/extract-data-two-factor-authentication/. [Accessed: Aug. 11, 2019].

[7] van Tilborg, Henk C.A.; Jajodia, Sushil, eds. (2011). *Encyclopedia of Cryptography and Security, Volume 1*. Springer Science & Business Media. p.1305.

[8] Jason Cipriani, Seth Rosenblatt, Two-factor verification: What you need to know (FAQ), CNET, June 15, 2015, [Online]. Available: https://www.cnet.com/news/two-factor-authentication-what-you-need-to-know-faq. [Accessed: Aug. 10, 2019].

[9] Alexandra Dmitrienko, Christopher Liebchen, Christian Rossow, and Ahmad-Reza Sadeghi "On the (In) Security of Mobile Two-Factor Authentication" Lecture Notes in Computer Science, pp. 365-383, Nov 2014.

[10] Alireza Pirayesh Sabzevar, Angelos Stavrou "Universal Multi-Factor Authentication Using Graphical Passwords", Proceedings of the 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems. pp. 625-632, 2008.

[11] Olufemi Sunday Adeoye "Evaluating the Performance of two-factor authentication solution in the Banking Sector" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 2, July 2012.

[12] Rahul Kale, Neha Gore, Kavita, Nilesh Jadhav, Swapnil Shinde " Review Paper on Two Factor Authentication Using Mobile Phone" International Journal of Innovative research and Studies, Vol. 2, Issue 5, pp. 164 - 170, May 2013.

[13] van Tilborg, Henk C.A.; Jajodia, Sushil, eds. (2011). *Encyclopedia of Cryptography and Security, Volume 1*. Springer Science & Business Media. p.1305.

[14] CardLogix Corporation, Smart Card Basics, CardLogix Corporation, 2010. [Online] Available from: http://www.smartcardbasics.com [Accessed 1/05/19].

[15] Tarun Agarwal, How does the Smart Card Works?, ElProCus, n.d. [Online] Available from: https://www.elprocus.com/working-of-smart-card/ [Accessed 1/05/19].

[16] MichaL Bairanzade, Smart card integration and specifications, ASPENCORE, 2002. [Online] Available from: https://www.eetimes.com/document.asp? doc_id=1200923 [Accessed 1/05/19].

[17] Wikipedia, Smart card, Wikipedia, Wikipedia, n.d. [Online] Available from: https://en.wikipedia.org/wiki/Smart_card [Accessed 29/04/19].

[18] D. Maio, and D. Maltoni, "Direct gray-scale minutiae detection in fingerprints", IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 19(1), pp. 27-40, 1997.

[19] S.L.Nita, M.I.Mihailiscu, V.C. Pau," Security and CryptographicChallenges for Authentication Based on Biometrics Data", www.mdpi.com/journal/cryptography, pp.1-22, 2018,

[20] K. Nallaperumall, A. L. Fred and S. Padmapriya, "A Novel for Fingerprint Feature Extraction Using Fixed Size Templates", IEEE 2005 Conference, pp. 371-374, 2005.

[21] M. Egele, D. Brumley, Y. Fratantonio, and C. Kruegel, "An empirical study of cryptographic misuse in Android applications," in Proceedings of the ACM SIGSAC Conference on Computer & Communications Security (CCS 2013), pp. 73–83, ACM, Berlin, Germany, November 2013.

[22] Kumar,R. Chandra,P. Hanmandlu, M.,"Local directional pattern (LDP) based fingerprint matching using SLFNN", IEEE Second International Conference on Image Information Processing (ICIIP), Pages: 493 - 498, DOI: 10.1109/ICIIP.2013.6707640, 2013.

[23] N. Vurukonda and B.T. Rao, "A Study on Data Storage Security Issuesin Cloud," Computing. Procedia Computer Science, 92, pp. 128-135, 2016.

[24] Ruxandra Burtica, Eleonora Maria Mocanu, Mugurel Ionuț Andreica, Nicolae Țăpuş, "Practical application and evaluation of no-SQL databases in Cloud Computing", IEEE 2012

[25] Hoxmeier, J. A., &DiCesare, C. (2000). System response time and user satisfaction: An experimental study of browser-based applications. AMCIS 2000 Proceedings, 347.

[26] Agrawal, S., Chaudhuri, S., Kollar, L., Marathe, A., Narasayya, V., &Syamala, M. (2005, June). Database tuning advisor for microsoft SQL server 2005: demo. In Proceedings of the 2005 ACM SIGMOD international conference on Management of data (pp. 930-932). ACM.

[27] R.Jeberson Retna Raj, T.Sasipraba,(2014) "Privacy Preserving of Sensitive Data in Cloud based on Fully Homomorphic Encryption (FHE) Technique", Global Journal of Pure and Applied Mathematics. ISSN 0973-1768 Volume 10, Number 3 (2014), pp. 431-441.

[28] Sarah P. Everett, Kristen K. Greene, Michael D. Byrne, Dan S. Wallach, Kyle Derr, Daniel Sandler and Ted Torous, “Electronic Voting Machines versus Traditional Methods: Improved Preference, Similar Performance”, in the Proceedings of Measuring, Business, and Voting, Florence, Italy, April 5-10, 2008.

[29] T. Kohno, A. Stubblefield, A. D. Rubin and D. S. Wallach, “Analysis of an Electronic Voting System”, IEEE Symposium on Security and Privacy 2004. IEEE Computer Society Press, May 2004.

[30] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. In PODC, pages 274–283, 2001.

[31] Thomos M.Buchsbaum, “E-Voting: International developments and lesson learnt”, Technical Report by Australian Federal Ministry for Foreign Affairs, 2004.

[32] “Electronic Voting - Evaluating the Threat,” Michael Ian Shamos, CFP '93. [Online] Available: http://www.cpsr.org/conferences/cfp93/shamos.html\. [Accessed: Sept]

[33] N. Munassar and A. Govardhan, “A Comparison Between Five Models Of Software Engineering”, IJCSI International Journal of Computer Science Issues, vol. 7, no. 5, 2010

Appendix A

Java code for Voting Interface

```
//////////////////////////java/////////////////////////
package com.prime.ev;

import java.io.*;
import java.lang.reflect.Array;
import java.net.URL;
import java.nio.ByteBuffer;
import java.util.*;

import javafx.application.Platform;
import javafx.collections.FXCollections;
import javafx.fxml.FXMLLoader;
import javafx.geometry.Rectangle2D;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.ListView;
import javafx.scene.control.ScrollPane;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.StackPane;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Paint;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;

import javax.imageio.ImageIO;


/**
 * Created by Prime on 8/4/2019.
 * DisplayManager is responsible for handling graphical interfaces,
 * and update the interface on request.
 */
public class DisplayManager {
    private Stage primaryStage;
    private static ArrayList<Scene> sceneList;
    private final String SCENE_NAME_FORMAT = "scene/scene";

    final int DELAY_MILLIS = 2000;
    private final SceneFunction sceneFunction;
    public static int addedScenes = -1;

    boolean inFinalScenes = false;

    private Map<String, String> currentElectionCodeMap;


    DisplayManager(Stage primaryStage) {
        this.primaryStage = primaryStage;
        sceneList = new ArrayList<>();
        currentElectionCodeMap = new HashMap<>();
        sceneFunction = new SceneFunction();
```

```
        new Thread(this::initializeAndStartFirstScenes, "Initialize First Scenes").start();
    }



    private void initializeAndStartFirstScenes() {
        setSceneFromIndex(1, DisplayAccessor.NEW_VOTER_SCENE);
        setScene(sceneList.get(0)); //start first scene
    }

    private void setSceneFromIndex(int fromIndex, int toIndex){
        for(int i=fromIndex; i<=toIndex; i++){
            try {
                URL fxml_url = getClass().getResource(SCENE_NAME_FORMAT + i + ".fxml");
                if (fxml_url == null) break;
                Scene scene = new Scene(FXMLLoader.load(fxml_url));
                scene.getStylesheets().add(getClass().getResource("scene/scene_style.css").toExternalForm());
                sceneList.add(scene);
            }catch(IOException e){e.printStackTrace();}
        }
    }

    protected Map<String, String> getCurrentElectionCodeMap(){
        return currentElectionCodeMap;
    }


    private int initializeVoterScenes(ArrayList<ElectionData> electionBundle, Map<String, String> userDetails) throws
IOException{
        if (electionBundle==null) throw new NullPointerException("electionBundle is null");

        URL fxml_url = getClass().getResource("scene/scene5.fxml");
        int numberOfVoterScenes = 0;

        //backup and restore true scene3 as first scene either as sceneIndex 3(before final scenes)
        // or sceneIndex 1 after final scenes
        if(inFinalScenes){
            Scene scene3 = getScene(DisplayAccessor.ANOTHER_NEW_VOTER_SCENE); //as sceneIndex 1
            sceneList = new ArrayList<>();
            sceneList.add(scene3);
        } else{
            Scene scene3 = getScene(DisplayAccessor.NEW_VOTER_SCENE); //as sceneIndex 3
            sceneList = new ArrayList<>();
            sceneList.add(scene3);
        }

        //set scene 4 user details
        Parent parent4 = FXMLLoader.load(getClass().getResource("scene/scene4.fxml"));
        Scene scene4 = new Scene(parent4, DisplayAccessor.SCREEN_WIDTH, DisplayAccessor.SCREEN_HEIGHT);
        scene4.getStylesheets().add(getClass().getResource("scene/scene_style.css").toExternalForm());

        userDetails.forEach((data, value)->{
            try{
                ((Label) scene4.lookup("#"+data)).setText(":    "+value);
            }catch(NullPointerException npe){
                System.out.println("no "+data+" field found on scene");
            }
        });
```

```
Image userImage;

{
   UserData usd = sceneFunction.getUserData();
   ArrayList<Byte> imageByteList = (ArrayList<Byte>) usd.image.get("data");
   byte[] imageBytes = new byte[imageByteList.size()];
   Object[] bytes = imageByteList.toArray();
   for (int i = 0; i < imageByteList.size(); i++) {
      imageBytes[i] = (byte) (double) (Double) bytes[i];
   }

   InputStream i = new ByteArrayInputStream(imageBytes);


   //compress
   /*
    * This should be done on the registration end instead
    */
   try {
      ImageCompressor.compress(ImageIO.read(i), new File("imtemp"), "jpg", 0.4f);
      userImage = new Image(new FileInputStream("imtemp"));
      //if successful, clear waste data for raspi
      imageByteList = null;
      imageBytes = null;
      bytes = null;
   } catch (IOException ioe) {
      ioe.printStackTrace();
      userImage = new Image(i);
   }
}


ImageView imageView = (ImageView)scene4.lookup("#userImage");

imageView.setFitWidth(512);
imageView.setFitHeight(512);
imageView.setPreserveRatio(false);
imageView.setClip(new Circle(imageView.getFitWidth()/2, imageView.getFitHeight()/2,imageView.getFitWidth()/2));
imageView.setImage(userImage);

sceneList.add(scene4);


//set the screened vote scenes for the voter
for(ElectionData electionData: electionBundle){
   if(isVoterEligible(electionData, userDetails)) {
      Parent parent = FXMLLoader.load(fxml_url);
      Scene scene = new Scene(parent, DisplayAccessor.SCREEN_WIDTH, DisplayAccessor.SCREEN_HEIGHT);
      scene.getStylesheets().add(getClass().getResource("scene/scene_style.css").toExternalForm());
      String lgaInfoFormat = " ("+userDetails.get("lga")+")";
      String appendLga = "";
      try{
         appendLga = !electionData.getTitle().contains("President") ? lgaInfoFormat:"";
      }catch(NullPointerException npe){npe.printStackTrace();}
      ((Label) scene.lookup("#electionTitle")).setText(electionData.getTitle()+
            appendLga);
```

```
            ListView listView = ((ListView) scene.lookup("#partyList"));
            listView.setItems(FXCollections.observableArrayList(wrapInView(electionData.getPartyList())));

            currentElectionCodeMap.put(electionData.getTitle(), electionData.getCode());
            sceneList.add(scene);
            ++numberOfVoterScenes;
        }
    }

    //set the final last 3 scenes
    setSceneFromIndex(6, 8);

    return numberOfVoterScenes;
}


private class isEligible{
    private boolean value;
    isEligible(boolean b){value = b;}
    boolean getValue(){return value;}
    void setValue(boolean b){value = b;}
}

private boolean isVoterEligible(ElectionData electionData, Map<String, String> userDetails){
    final isEligible eligible = new isEligible(true);
    electionData.getCriteria().forEach((criteria, value)->{
        if(criteria.equals("age")){
            //do some calc in check
            /*@debug*/System.out.println("checking age restriction");
        }
        else if(!userDetails.get(criteria).equalsIgnoreCase(value)) eligible.setValue(false);
    });
    return eligible.getValue();
}


private ArrayList<StackPane> wrapInView(ArrayList<String> partyNameList) throws IOException{
    ArrayList<StackPane> sPanes = new ArrayList<>();
    for(String partyName: partyNameList){
        StackPane sPane = FXMLLoader.load(getClass().getResource("customfx/party_box.fxml"));
        ((Label) sPane.lookup("#party_name")).setText(partyName);

        Image partyLogo;
        try{ partyLogo = new Image(DisplayAccessor.RESOURCES+"/logo/"+partyName+".jpg"); }
        catch(IllegalArgumentException i){
            partyLogo = new Image(DisplayAccessor.RESOURCES+"/logo/default.jpg");
        }
        ((ImageView) sPane.lookup("#party_logo")).setImage(partyLogo);
        sPanes.add(sPane);
    }
    return sPanes;
}

/*
 * Runs the inner scenes
 * Note: i corresponds to scenex.i.fxml
 */
private void playScene(int scene_no) {
```

```
    /*@debug*/System.out.println("\nplayScene invoked with scene number: "+scene_no);

    if(inFinalScenes) return; ///for now

    new Thread(()->{
        try{
            for(int i=1; ; i++){
                Thread.sleep(DELAY_MILLIS);
                URL fxml_url = getClass().getResource(SCENE_NAME_FORMAT+scene_no+"."+i+".fxml");
                /*@debug*/System.out.println("searched resource: "+SCENE_NAME_FORMAT+scene_no+"."+i+".fxml");
                if(fxml_url == null) {
                    /*@debug*/System.out.println("resource not found"); break;
                }
                /*@debug*/System.out.println("found "+fxml_url.toExternalForm());
                setRoot(fxml_url, i);
            }
        } catch(Exception e){ e.printStackTrace(); }
    }, "Play Scenes").start();
}


private void setScene(Scene scene) {
    if(Thread.interrupted()) return;

    Platform.runLater(()->{
        primaryStage.setScene(scene);
        try{
            playScene(indexOfScene(getCurrentScene()));
            invokeSceneFunction(indexOfScene(getCurrentScene()));
        } // +1 to get actual file index
        catch(Exception e){e.printStackTrace();}
    });
}


void setScene(int sceneConstant){
    switch (sceneConstant){
        case DisplayAccessor.ANOTHER_NEW_VOTER_SCENE:
            setScene(getScene(DisplayAccessor.ANOTHER_NEW_VOTER_SCENE)); break;
        case DisplayAccessor.NEW_VOTER_SCENE:
            setScene(getScene(DisplayAccessor.NEW_VOTER_SCENE)); break;
        case DisplayAccessor.USER_DETAILS_ERROR_SCENE:
            break;
    }
}


Scene getCurrentScene(){return primaryStage.getScene();}

private Scene getScene(int sceneIndex){
    return sceneList.get(sceneIndex-1);
}

int indexOfScene(Scene scene) {
    int index = sceneList.indexOf(scene);
    index = index<0 ? index : index+1;
    return index;
}
```

```
private void setRoot(URL url, int rootNumber) {
   Platform.runLater(()->{
      try{
         getCurrentScene().setRoot(FXMLLoader.load(url));
         invokeRootFunction(rootNumber);
      } catch(Exception e){e.printStackTrace();}
   });
}


void nextScene() {
   int oldSceneIndex = indexOfScene(getCurrentScene());
   int newSceneIndex = oldSceneIndex + 1;
   /*@debug*/System.out.println("oldSceneIndex in nextScene: "+oldSceneIndex+"; new: "+newSceneIndex);
   if(newSceneIndex >= 1 && newSceneIndex <= sceneList.size()) //range(1 - sceneCount)
      setScene(sceneList.get(newSceneIndex-1)); //actual
}


void prevScene() {
   int currentSceneIndex = sceneList.indexOf(getCurrentScene());
   if(currentSceneIndex > 1)
      setScene(sceneList.get(--currentSceneIndex));
   else /*@debug*/System.out.println("no prev scene");
}


private void summarizeVoteData() throws IOException{
   ArrayList<StackPane> sPanes = new ArrayList<>();
   for(Map<String, String> voteMap: sceneFunction.getVotes(trimScenesToElect(sceneList))){
      StackPane sPane = FXMLLoader.load(getClass().getResource("customfx/voteItemBox.fxml"));
      ((Label) sPane.lookup("#electionTitle")).setText(voteMap.get("election"));
      ((Label) sPane.lookup("#partyName")).setText(voteMap.get("party"));

      Image partyLogo;
      try{ partyLogo = new Image(DisplayAccessor.RESOURCES+"/logo/"+voteMap.get("party")+".jpg"); }
      catch(IllegalArgumentException i){
         partyLogo = new Image(DisplayAccessor.RESOURCES+"/logo/default.jpg");
      }
      ((ImageView) sPane.lookup("#partyLogo")).setImage(partyLogo);
      sPanes.add(sPane);
   }

   ListView listView = ((ListView) getCurrentScene().lookup("#partyList"));
   listView.setItems(FXCollections.observableArrayList(sPanes));
}

int getSceneCount(){return sceneList.size();}

void invokeSceneFunction(int sceneIndex){

   if(inFinalScenes){
      if(sceneIndex == sceneList.size()-2)
         try {summarizeVoteData();} catch(IOException ioe){ioe.printStackTrace();}
      if(sceneIndex == sceneList.size()-1) //fingerprint reading/voting scene
         sceneFunction.castVote(trimScenesToElect(sceneList));
```

```
        if(sceneIndex == sceneList.size()) return;//////////////////do nothing
            //sceneFunction.newVote(); ////////////////////////////////////remove this when card is implemented
        }

        switch(sceneIndex) {
            case DisplayAccessor.ANOTHER_NEW_VOTER_SCENE:
                if(!inFinalScenes) break;
                Thread scene1Thread = new Thread(()->{ try {
                    /*
                     * note that when the sceneFunction.fetchUserDetails returns false,
                     * the program pauses and waits for the user to retract his/her card.
                     * This retraction reloads the voter scene, serving as a loop in any
                     * occurrence of error while fetchingUserDetails
                     */
                    if(!sceneFunction.fetchUserDetails()) return; //loop till it returns true
                    Map<String, String> userDetails = sceneFunction.getUserDetailsMap();
                    initializeVoterScenes(sceneFunction.getElectionBundle(), userDetails);
                    DisplayAccessor.nextScene();
                }
                catch(Exception e){
                    fatalError();
                    e.printStackTrace();
                }
                }, "Scene1 - Fetch Voter Details");
                scene1Thread.start();
                DisplayAccessor.addSceneThread(scene1Thread);
                break;

            case DisplayAccessor.NEW_VOTER_SCENE:
                if(inFinalScenes) break;
                //inFinalScenes = true;
                Thread scene3Thread = new Thread(()->{ try {
                    if(!sceneFunction.fetchUserDetails()) return; //loop till it returns true
                    Map<String, String> userDetails = sceneFunction.getUserDetailsMap();
                    initializeVoterScenes(sceneFunction.getElectionBundle(), userDetails);
                    inFinalScenes = true;
                    DisplayAccessor.nextScene();
                }
                catch(ArrayIndexOutOfBoundsException arrayException) {
                    arrayException.printStackTrace();
                    fatalError();
                }
                catch(Exception e){
                    System.out.println("Unknown error");
                    e.printStackTrace();
                }
                }, "Scene3 - Fetch Voter Details");
                scene3Thread.start();
                DisplayAccessor.addSceneThread(scene3Thread);
                break;
            //case DisplayAccessor.USER_DETAILS_ERROR_SCENE:
            //    sceneFunction.userDetailError(); break;
        }
    }

    //Note: sceneX.rootIndex.fxml
    void invokeRootFunction(int rootIndex) {
        switch(rootIndex){
```

```
        //scene number for particular root number

        case DisplayAccessor.FETCH_RESOURCES_ROOT:
            new Thread(()->{
                try{
                    //sceneFunction.fetchElectionBundle();
                    sceneFunction.showStartStatus(sceneFunction.createSocketConnection());
                }
                catch(Exception e){e.printStackTrace();}
            }, "Fetch Election Resource").start();
            break;
    }
}

void setResultScene(){
    try {
        Scene resultScene = new Scene(FXMLLoader.load(getClass().getResource("scene/results.fxml")));
        resultScene.getStylesheets().add(getClass().getResource("scene/scene_style.css").toExternalForm());
        StringBuilder presVoteCount = new StringBuilder();
        ArrayList<StackPane> sPanes = new ArrayList<>();

        if(Factory.presidentialVoteCount!=null){
            Factory.presidentialVoteCount.stream().limit(3).forEach(entry->{
                presVoteCount.append(String.format("%s, %d\n", entry.getKey(), entry.getValue()));

                try{
                    StackPane sPane = FXMLLoader.load(getClass().getResource("customfx/voteItemBox.fxml"));
                    //sPane.setMaxHeight(80);
                    ((Label) sPane.lookup("#electionTitle")).setText(entry.getKey());
                    ((Label) sPane.lookup("#partyName")).setText(entry.getValue().toString());

                    Image partyLogo;
                    try{ partyLogo = new Image(DisplayAccessor.RESOURCES+"/logo/"+entry.getKey()+".jpg"); }
                    catch(IllegalArgumentException i){
                        partyLogo = new Image(DisplayAccessor.RESOURCES+"/logo/default.jpg");
                    }
                    ImageView imView = (ImageView) sPane.lookup("#partyLogo");
                    imView.setFitWidth(80);
                    imView.setFitHeight(80);
                    imView.setPreserveRatio(false);
                    //imView.setViewport(new Rectangle2D(50,50,50,50));
                    imView.setImage(partyLogo);

                    sPanes.add(sPane);
                }catch (Exception e){e.printStackTrace();}
            });
            ListView listView = ((ListView) resultScene.lookup("#rankedVoteList"));
            listView.setItems(FXCollections.observableArrayList(sPanes));
        }

        StringBuilder summary = new StringBuilder();
        Factory.voteSummary.forEach((election, count)->{
            char[] _election = election.toCharArray();
            _election[0] = String.valueOf(_election[0]).toUpperCase().toCharArray()[0];
            summary.append(String.format("%-12s  %7d votes\n", String.valueOf(_election), count));
        });
```

```
            //((Label)(resultScene.lookup("#presidential"))).setText(presVoteCount.toString());
            ((Label)(resultScene.lookup("#summary"))).setText(summary.toString());
            primaryStage.setScene(resultScene);
        }catch (Exception e){e.printStackTrace();}
    }
    private void fatalError(){
        Platform.runLater(()->{
            ((Label) getCurrentScene().lookup("#prompt")).setText("fatal error, reboot device");
            getCurrentScene().lookup("#prompt").setStyle("-fx-font-size: 25px");

            //irrecoverable error by calling next scene with out of bound array index,
            // so no need for retry button
            //getCurrentScene().lookup("#retryButton").setVisible(true);
        });
    }
    private List<Scene> trimScenesToElect(ArrayList<Scene> scenes){
        return scenes.subList(DisplayAccessor.FINAL_VOTE_BEGIN_SCENE-1, sceneList.size()-3);
    }
}
```

Appendix B

Python code for Interfacing the Smart Card Reader

```
//////////////////read card from raspberrypi//////////////
from smartcard.System import readers

'''
returns -1 on error
'''

#get connected card readers
reader = readers()

connection = reader[0].createConnection()

#connect to card
try:
    connection.connect()

    #read data (20 bytes) from address 0x01 0x04
    data, sw1, sw2 = connection.transmit([0xFF, 0xB0, 0x01, 0x04, 0x14])

    #disconnect card


    connection.disconnect()

    ##print the data read from the card
    print "".join(map(chr, data))
except:
    print -1
    exit()
```

Appendix C

Python code for Interfacing the Fingerprint Scanner

```
//////////////////collect finger print//////////////////////
from pyfingerprint.pyfingerprint import PyFingerprint
import sys

## Search for a finger
## Tries to initialize the sensor
try:
    f = PyFingerprint('/dev/ttyUSB0', 57600, 0xFFFFFFFF, 0x00000000)

    if ( f.verifyPassword() == False ):
        raise ValueError('The given fingerprint sensor password is wrong!')

except Exception as e:
    print(-1)

## Tries to match the finger
try:
    buffer1 = 0x01
    buffer2 = 0x02


    if(len(sys.argv)<2):
        raise ValueError('no arguments!')

    #the fingerprint to be matched is sent as an argument
    arg = sys.argv[1]

    characteristicsData = list(map(int, arg))

    print('Waiting for finger...')

    ## Wait that finger is read
    while ( f.readImage() == False ):
        pass

    ## Converts read image to characteristics and stores it in charbuffer 1
    f.convertImage(buffer1)

    #uploads the characteristics from arguments to charbuffer 2
    f.uploadCharacteristics(buffer2, characteristicsData)

    #compare buffer 1 with buffer 2
    score = f.compareCharacteristics()

    #displays the match score
    print('score: ' + str(score))

except Exception as e:
    print(-1)
```

Appendix D

Pictures of the Complete Work