



A Credible, Automated E-Voting System in the Context of Bangladesh

A Thesis submitted to the department of Computer Science
and Engineering of **BRAC University**

By

Abdulla Al Faiyaz (13101168)

Fairouz Sharif (13101116)

Supervised By

Mr. Hossain Arif

Assistant Professor

Department of Computer Science and Engineering

BRAC University, Dhaka, Bangladesh.

Declaration

This is to certify that this final thesis report is submitted by the authors for the purpose of obtaining the degree of Bachelor of Science in Computer Science, and the degree of Bachelor of Engineering in Computer Science and Engineering. We hereby declare that all the instances of work presented in this thesis are original and inspirations for the work that we have made use of have been duly accredited with proper referencing.

Signature of Supervisor

Signature of Authors

Mr. Hossain Arif

Abdulla Al Faiyaz

Fairouz Sharif

Acknowledgement

We would like to express our special thanks of gratitude to our supervisor Mr. Hossain Arif sir, Department of Computer Science And Engineering for his generous supervision and nonstop support throughout our work. Without his assistance and guidance, we wouldn't be able to finish our research successfully.

We express our deepest gratitude to our dearest friend Sebastain Romy Gomez. He guided us with an easy approach to start our work for which the project has now come into being. We also thank our respected faculty M. Abdur Rahman. He helped us resolve major difficulties in the development process, overcoming which we continued our work to reach the end goal.

We are extremely thankful to Almighty Allah and His many blessings which is why we are healthy and alive. We are grateful to our parents from the heart who always support us day and night and help us in every cross every barrier. And we appreciate our friends who supported and encouraged us along the way.

We would also like to take this opportunity to thank Mr. Touhid Hossain, Mr. Kazi Rezaul Karim – our Lab Technical Officers and Mr. Abdul Karim, Office Assistant who kept an eye on the environment of the lab and made sure that we got all the technical support we needed.

And finally we extend our gratitude to BRAC University for giving us the opportunity for developing this project and for helping us with necessary resources to conduct this thesis.

Abstract

Elections are believed to be the best possible way to live in democratic era and voting is one of the electoral processes that ensures the alignment of democracy in our society. In this thesis, we propose an electronic voting system which can minimize the flaws of traditional voting system. The manual system of voting includes major chances of vote rigging and manipulation, which can be brought to its barest minimum by our proposed system. Also, in paper-based voting system, great amount of people and money is invested and yet the results are not satisfying. Our proposed voting system promises to reduce these waste and serve as a cost-effective, easy-to-use and secure one. As our country have never fully adopted any electronic voting system before, we have tried to build it in accordance with the steps of procedural voting system. Along with biometric authentication, the system also includes candidate pictures along with their voting symbols, using which voters can easily pick their desired candidate with little amount of time. This system will be effective for both sides, one who are voting and another who are monitoring the whole process.

Table of Contents

Declaration.....	i
Acknowledgement.....	ii
Abstract.....	iii
List of Figures.....	vii
List of Tables.....	viii
Chapter 1 Introduction.....	1
1.1 Motivation.....	1
1.2 Election in Bangladesh.....	2
1.2 Thesis Outline.....	3
Chapter 2 Background Study.....	4
2.1 Literature Review.....	4
2.2 Fingerprint Recognition.....	7
2.3 Encryption.....	10
Chapter 3 System Architecture.....	11
3.1 System Overview.....	11
3.2 System Flow.....	12

3.2.1	Authentication.....	12
3.2.2	Flowchart of Voting Process.....	13
Chapter 4	System Setup.....	15
4.1	Hardware Requirements.....	15
4.2	Software Requirements.....	16
4.3	Development Tools.....	16
Chapter 5	System Implementation.....	16
5.1	Database Design.....	17
5.2	Authentication.....	20
5.3	Session Generation.....	22
5.4	Interaction View.....	23
5.5	Encryption Process.....	24
Chapter 6	Experiments & Results.....	25
6.1	Experiments	26
6.1.1	Testing the SDK.....	26
6.1.2	Experiments Step by Step.....	26
6.2	Results and Discussion	32
6.2.1	Findings.....	32
6.2.2	Problems Faced During Implementation.....	34
Chapter 7	Conclusion.....	35

7.1 Future Work	35
References.....	37

List of Figures

Figure 2.1. Different Minutia points.....	7
Figure 2.2 Working of typical feature extraction algorithm.....	8
Figure 3.1 Block Diagram of Proposed System.....	11
Figure 3.2 Fingerprint Enrollment and Authentication.....	12
Figure 3.3 System for Casting Votes.....	13
Figure 5.1 ER Diagram of the system.....	18
Figure 5.2 Fingerprint Sensor ZK4500.....	20
Figure 5.3 Use Case Diagram of Proposed System.....	23
Figure 5.4 Sequence Diagram of Voting Process.....	24
Figure 5.5 Block Diagram of TDES Algorithm.....	25
Figure 6.1 Front page of the application.....	27
Figure 6.2 HTML application with admin functionalities.....	27
Figure 6.3 Voter Registration.....	28
Figure 6.4 Edit Voter.....	29
Figure 6.5 Edit Candidate.....	29
Figure 6.6 Logging in with Fingerprint.....	30
Figure 6.7 Profile and Voting Panel.....	31
Figure 6.8 Final Countdown.....	32

List of Tables

Table 4.1 CPU Specification	15
Table 4.2 RAM and Disk.....	15
Table 4.3 Software Specification	16
Table 4.4 Tools used	16
Table 5.1 Voter Table	19
Table 5.2 Candidate Table	19
Table 5.3 Admin Table	20
Table 5.4 Votes Table	20

Chapter 1 Introduction

Elections are assumed to be the main pillar of democracy all over the globe and voting is a vital democratic right of every citizen of a democratic nation. Voting is the procedure that lets the common people choose their leaders and thus play some role in the governance of a nation. The reliability of a democratic system is key to the reliability of election itself. Therefore the process of election needs to be adequately secured. It must survive a range of deceitful behaviors and must be transparent and logical so that voters and candidates can agree on the result of an election. The past is littered with samples of elections being manipulated in order to change their outcome. Elections in most developed nations of the world with desirable democratic platforms have over the years been conducted electronically. In these countries, little or no setbacks follow the results of such elections as they are seen to be free, fair and credible. So it's really high time to introduce such systems in order to build a future we have been dreaming for so long.

1.1 Motivation

Violation of rules, vote rigging and fraudulent behavior have been major topics of discussion regarding elections all around the world. Both paper-based and electronic voting systems can trigger chaos, confusion and debate in the minds of participants as well as the observers during the election season. According to Schedler [1], Coercion refers to voter intimidation and 'Corruption' refers to vote buying. Intimidation is the way of threatening someone if he/she does not agree to perform a specific task, in this case, it is casting vote for a specified candidate or party. Schaffer et al. [2] refer to 'Vote buying' as exchange of money, goods or services for votes. This is a corrupted practice of bribing someone which imposes great threat to conduct free and fair elections. Anyone involved in the process of vote counting, can act as an agent of influential political people and look inside casted votes to check whether vote for specified candidate was cast or not by that voter. This loopholes can only be eliminated by a secure e-voting system. So we came forward to build one that reduces human involvement and also ensures confidentiality by protecting voter and his/her voting information.

Along with this there comes another term called 'integrity', this stands [1] for - One person, One vote. Traditional voting systems can violate this principle by allowing corrupt minds to participate

freely in the voting process. In the proposed system, we overcome this gap by using an authentication process that allows legitimate voters to vote only once.

Last major election was held in 2014 which also had these major allegation against the Election Commission of Bangladesh. Almost 92,007,113 voters had been registered for votes and only 47,262,168 votes had been cast which less than half of the registered voters. What went wrong? People don't feel secure when comes to voting because of the processes they had been offered. They are familiar with the holes in the system.

Another issue is growth rate of population, the next election will be more challenging and cost-effective and also time -consuming with a large group of new voters. In short, our procedural system is kind of old school, lack of security and waste of enormous money where we can offer a much better electronic voting system. It will gain the common people's trust, making the voting process much smoother and secure.

1.2 Election in Bangladesh

Mollah & Jahan [3] states that, in the last parliamentary election of 2014, voter turnout was around 39% - a steep decline from the last general elections, when more than 87 percent voted. Around 19 people were reported to have been killed in political violence, and 440 polls were closed early for security issues. Bangladeshi television stations broadcast images of rural polling places charred by arson attacks, and of bodies wrapped in red blankets. This chaos originated mainly from political unrest where the opposition and several other parties refused to compete for ideological clash with the party in power before that election. Such unrest is common in our country and from this rises the confusion of transparent and fair election, which demotivates voters to have some enthusiasm in the process of electing future candidates.

The purpose of using EVM is mainly to automate the voting process, mainly casting and counting. Thus ultimately, it would speed up the process. EVMs were first used in Bangladesh in the 2012 Chittagong Municipality Corporation Election and later in the Comilla City Council Election. The use of EVMs in those elections was quite satisfactory. However, there remained are always controversies. Recently [4] Electronic Voting machines were used in Rangpur City Corporation polls. Some voters said they were nervous at the beginning after seeing the machine while another group of voters appreciated the initiative of two-day campaign taken by the Election Commission.

1.3 Thesis Outline

The outline for our thesis is as follows:

- Chapter 1; the introduction of the thesis, the motivation behind making this system as well as the organization of our thesis work.
- Chapter 2; this contains the background study done for the thesis, which includes the literature review and fingerprint recognition and in the fingerprint recognition part we have shared some algorithm and data we have researched.
- Chapter 3; presents an overview of the system, a bird's eye view on how the system works. It also includes details of the system flow as well as flowchart of the whole system.
- Chapter 4; provides a detailed description of the system setup. In this section, we have provided hardware specs, software specs and development tools.
- Chapter 5; asserts the analysis and findings database design, authentication and interaction view.
- Chapter 6; highlights testing the SDK, testing the framework, voting panel functions and also discussion of final experiments.
- Chapter 7; In this concluding section we have discussed the benefits and future scopes
- References are provided at the end of the report.

Chapter 2 Background Study

2.1 Literature Review

To understand the existing voting system of Bangladesh, we looked into a [4] report prepared by Commonwealth Observer Group. They observed Parliamentary Elections of Bangladesh in 2008. They depicted the entire scenario with a description of step-by-step process. In the report, it is mentioned that the Election Commission is responsible for the administration of electoral process and is represented across Bangladesh. Each Constituency has a Returning Officer (RO appointed by the EC. There are also Assistant Returning Officers (ARO) to support the process. . For the purpose of polling, in 2008 election, 35,263 Polling Centers were established, containing 177,107 Polling Booths. Each Centre was headed by a Presiding Officer. All these data, undoubtedly shows the involvement of huge amount of people in the process. Any of them can be influenced and thereby corrupt the process of vote casting by a voter.

The Voting Technology Project of Caltech/MIT provides scientific analysis regarding voting technology and election administration. Alvarez et al. [4] state several problems in the 2000 presidential election of the United States. In that election, 4 to 6 million votes were lost due to the problems in ballot papers, voter registration and polling places. The project also suggests that every voting machine challenge should have four components. First, equipment must be reliable. Second, voting machines need to be secure. Third, standards must be followed in order to assist governments in making appropriate decisions. Fourth, and perhaps the most important, there needs to be a sustainable business model for the voting machine industry. This gives us some idea about how an electronic voting machine should be.

Djanali et al. [5] suggest a multilayer architecture for level-by-level communication. An intermediary server connects Central Committee (central server) with Voting Booths (Local server). For each communication within the levels, they introduced HELLO Packet. They prevented middle-attacks by securing the connections with https. The researchers strengthened the security of database storage and message exchange using a combination of SHA 256, digital signature and RSA asymmetric encryption.

Kumar & Begum [6] suggests that any electronic voting system needs to have several core properties-

1. Accuracy:
 1. A vote cannot be altered
 2. A validated vote cannot be eliminated from the final tally
 3. An invalid vote must not be counted in the final tally
2. Democracy:
 1. Only eligible voters can vote
 2. And that can be done only once
3. Privacy:
 1. Neither authorities nor anyone else can link any ballot to the voter who cast it
 2. And no voter can prove that he voted in a particular way
4. Verifiability:
 1. Anyone can independently verify that all votes have been counted correctly.
5. Collusion Resistance:

No electoral entity (any server participating in the election) or group of entities, running the election can work in a conspiracy to cast votes or prevent or influence voters from voting. If all entities conspire this property isn't achieved. So, this characteristic should be measured in terms of the total number of entities that must conspire to guarantee a successful interference in the election.
6. Availability:
 1. The system works properly as long as the poll stands
 2. And any voter can have access to it from the beginning to the end of the pol.
7. Resume Ability:
 1. The system allows any voter who had interrupted his/her voting process to resume it or restart it while the poll stands

Abdulhamid et al [7] proposed a system divided into several modules – registration, ballot design, database administration, voting module, real-time live results module. There are separate functionalities for voter and admin to enable online election. This system provides an user-

interface for login. After successful login, they are taken to the voting page to cast vote for local and central candidate. If that is also a success, then the vote is saved in database and the process ends thereby. Later on, the stored votes are counted for result calculation.

Kang & Lee [8] stated that the voting can be done only at the places where the voting places are installed. Though voting can be done using mobile terminals at any places if the wireless network develops further in the forthcoming days, the additional requirements for security will be required depending on the wireless circumstances. And the way of authentication must be provided more strongly and there should not be coercive voting or exposure of data in the wireless network. Voting is a key way of democracy reflecting the nation's intention. Therefore, a study on security technology applied to the electronic voting system should be progressed continuously in the future.

Al-Sammari & Tessaris [9] discussed about currently deployed vote verification methods. By discussing their weaknesses with the aim of proposing a more reliable and robust vote verification method. Authors in the paper sought to propose a vote verification technique which would be able to verify votes against major possible threats and enable all election participants to verify votes. For this purpose, they need to investigate a combination of both technological and procedural solutions.

Rahim et al. [10] proposed an e-voting prototype called improved DynaVote e-Vote protocol was planned to protect the counter from irregularities associated with counting impersonated (multiple votes) in the same election. This was accomplished by introducing biometric fingerprint and pseudo voter identities (PVID) encryption for each voter during voter registration via online or data mining of population data containing fingerprint biometrics. Furthermore, fingerprint reader and RSA public key cryptography are used in PVID to remove counting impersonated votes. The performance results showed that improved DynaVote e-Vote protocol is more reliable, eligible, and accurate and protects voter privacy against other e-Vote protocols.

Mahmood et al. [11] suggest an android based voting system with two layers – hardware and software. This project introduces android phone, Arduino mega, Fingerprint Scanner and the thermal printer all the same floor. The system is designed for different kinds of users like voter and the administrators. Then again the users can be divided into subclasses. It is stated that, integration of microcontroller with scanner can take significant amount of time making the system slower.

Anis et al. [12] suggest an electronic voting machine with the inclusion of Near Field ID Card Communication ID Cards and Fingerprint Sensor. These two provide an extra layer of verification ensuring reliability of the system. However, successful login takes the voter to ballot unit which is hardware based. For choosing candidates, buttons beside candidates need to be pressed. Apart from this, there is inclusion of Arduino Mega and Raspberry Pi and POS printer which makes the EVM a complex one. However, the system shows a trend towards better performance of an electronic voting machine. Smooth incorporation [12] of numerous devices it can alleviate the satisfaction of all parties involved in an election.

Okediran et al. [13] depicts various security issues regarding remote internet voting. In particular, the research examines the feasibility of running national elections over the internet. The focus of this research was on the limitations of the currently deployed infrastructure in terms of the security of the hosts and the Internet itself. It concludes that without appropriate security measures, internet-based elections can be a major challenge.

2.2 Fingerprint Recognition

Human fingerprints are made of unique patterns which makes it a valuable source to identify anyone. That's why we chose fingerprint matching for biometric authentication since it is largely used in various systems of the state.

- **Minutiae points**

Our fingerprints are made of ridges and valleys. Differences in these make up minutia points which can distinguish between users. Jain et al [14] state that, minutia points represent locations where friction ridges abruptly end (ridge endings) or where a ridge branches into two or more ridges (ridge bifurcations). Apart from these, minutia can be determined by several other patterns. Minutia extractor identifies these details (minutia) from a fingerprint image.

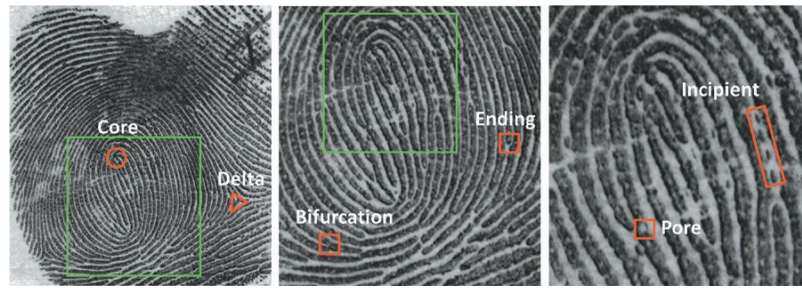


Fig 2.1: Different minutia points (green box in the first one is enlarged in the preceding images)

- **Minutiae extraction**

A good quality image is absolutely essential for extracting minutiae features. Sometimes captured image might be of poor quality due to cuts or bruises, skin dryness or damage. Due to this, it is necessary to enhance the fingerprint images before minutiae matching. The extraction methods can be classified into two broad categories:-

- Methods that work directly on gray-scale fingerprint images
- Methods that work directly on binarized fingerprint templates

A typical good-quality fingerprint image should have about 20-70 minutiae points, which depends on the surface area of the sensors used and also on the techniques of image enhancement and feature extraction. These features follow major pattern types like arch, loop or whorl.

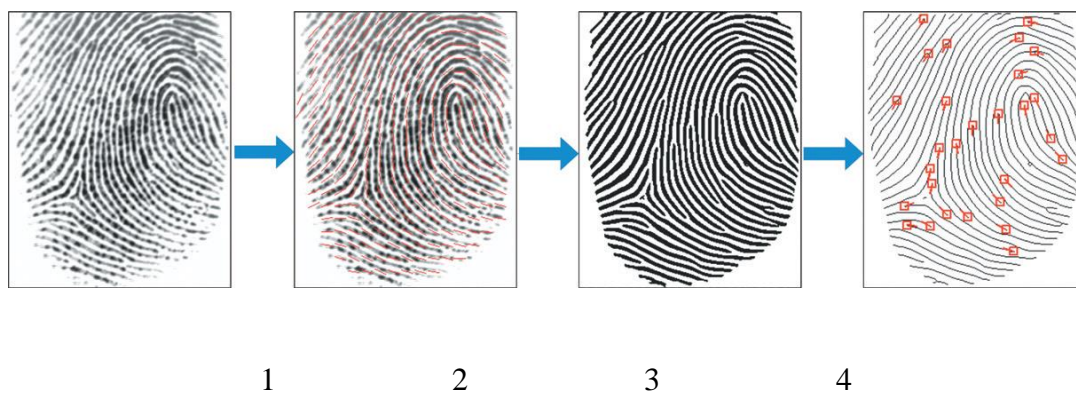


Fig 2.2: Working of typical feature extraction algorithm

Steps of such an algorithm are as follows –

1. A plain image is obtained from the sensor
2. Algorithm estimates ridge orientation and frequency from image
3. Contextual filtering improves image quality and ridge extraction
4. Obtains skeleton image from enhanced image by tracing ridge lines. From there minutiae points can be determined.

- **Verification**

The system takes one input fingerprint and compares it with one pre-enrolled fingerprint to see if they are from the same finger. This is also known as one-to-one (1:1) matching. The notion to understand this is it answers – are you who you say, you are?

- **Identification**

The system takes a template and matches with all the stored templates in database. A search query is run to find out whether there is any match in the database. The notion to understand this is that it answers who a person by taking their biometric as input. It is also known as one-to-many (1: N) matching.

- **Effectiveness**

In verification systems effectiveness is measured by two commonly used parameters:

1. **False Rejection Rate (FRR)**

It is also known as False Non-Match Rate (FNMR). FRR is a value that measures the percentage of times when a match for a biometric sample found against a single or multiple biometric templates, but the similarity score is below the decision threshold setting so no match occurs. In other words, it's the number of times people do not get identified when they should be identified.

2. False Accept Rate (FAR)

It is also known as False Match Rate (FMR). FAR is a value that measures the percentage of times a biometric sample is matched against a single or multiple biometric templates where a similar biometric template is not stored but the likeness between the sample and template is above the decision threshold. Thereby, the systems yield an incorrect match for that template.

2.3 Encryption

Basically, encryption means creating jumbled unreadable text so that information cannot be hacked even if the system falls into wrong hands. According to Stine & Dang [15], data we encrypt can be [15] of two types – data-at-rest and data in transit -

- **In transit**, meaning it's moving via email, in apps, or through browsers and other web connections
- **At rest**, when data is stored in databases, the cloud, computer hard drives, or mobile devices

Encryption for data-at-rest comes with the purpose of preventing the attacker to do anything with the unencrypted data. It is required in case of someone in maintenance attempts to launch an attack – for example, removing the hard drive and installing it in a computer under control or installing malicious software in the device. To protect sensitive data in such scenarios, encryption must be applied. Among many kinds, there are secret and public key encryption. Lozupone [16] suggests secret or symmetric encryption where one secret key is used to encrypt and decrypt data. It can be used for preserving 'confidentiality' and 'integrity'. The key that is used is nothing but a set of random bits. Most popular block cipher for this kind of encryption is DES (Data Encryption Standard). It has a key of 56 bits. Common symmetric encryption algorithms include DES, 3DES, AES, and RC4. Such algorithms can be extremely fast, and their relatively low complexity allows for easy implementation in hardware. On the other hand, asymmetric or public key cryptography consists of two keys, public and private, one for encryption and another for decryption. This one requires more processing power, hence a bit slower.

Chapter 3 System Architecture

3.1 System Overview

The proposed system consists of three modules. One is the Database Creation Module. Candidate and Voter Information is inserted into the database by admin. The Voter Side includes Fingerprints for registration which can be achieved using the sensor and later can be stored in local database. Then we have the Login Module which also consists of two parts – Admin Login and Voter Login. If admin logs in with proper credentials he/she is taken to the Admin Panel. Voters have to Login using their fingerprints. Captured Template is matched against all registered templates in the local database and if a match is found Voting Panel appears. There the voter can cast votes, vote counting takes place in the background for each candidate and finally, the stored votes remain encrypted for enhancing the security of the system.

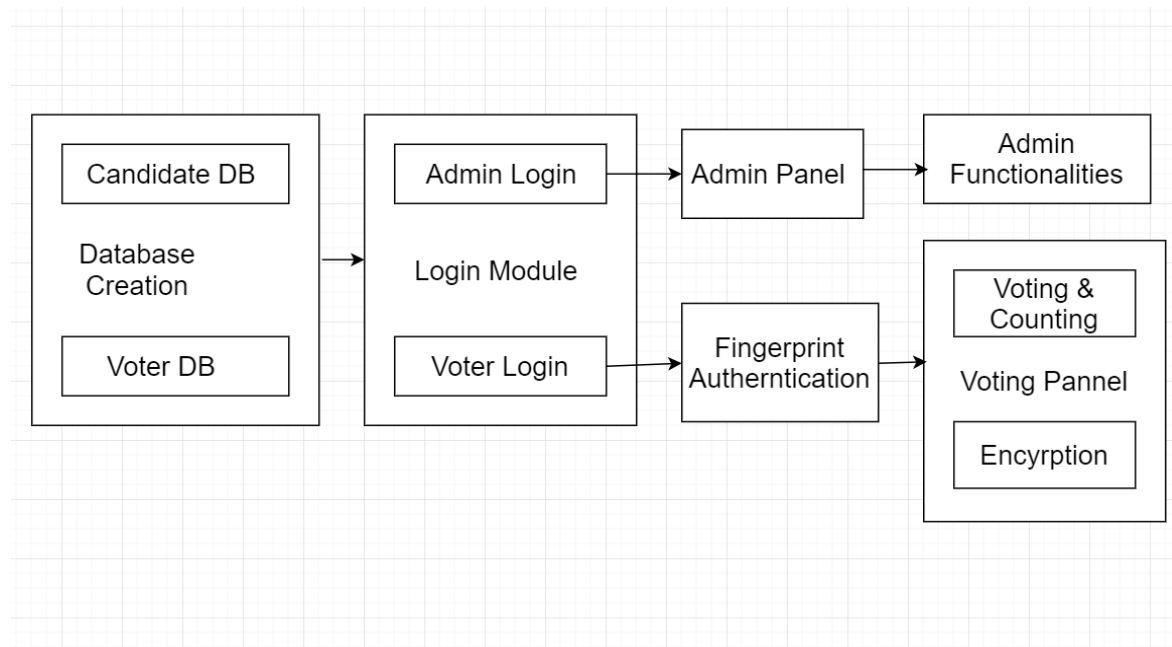


Fig 3.1: Block Diagram of Proposed System

3.2 System Flow

3.2.1 Authentication –

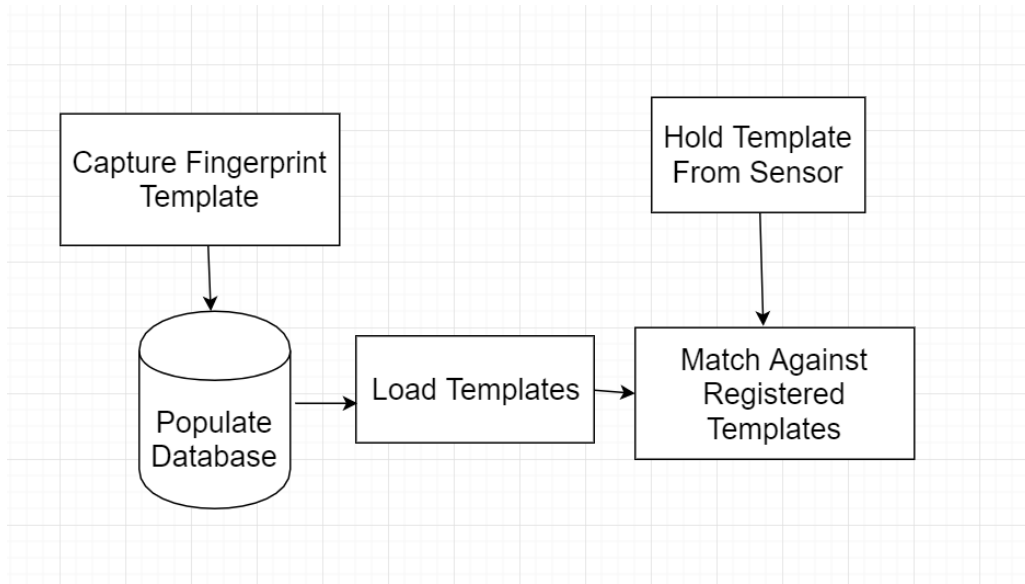


Fig 3.2: Fingerprint Enrollment and Authentication

3.2.2 Flowchart of Voting Process

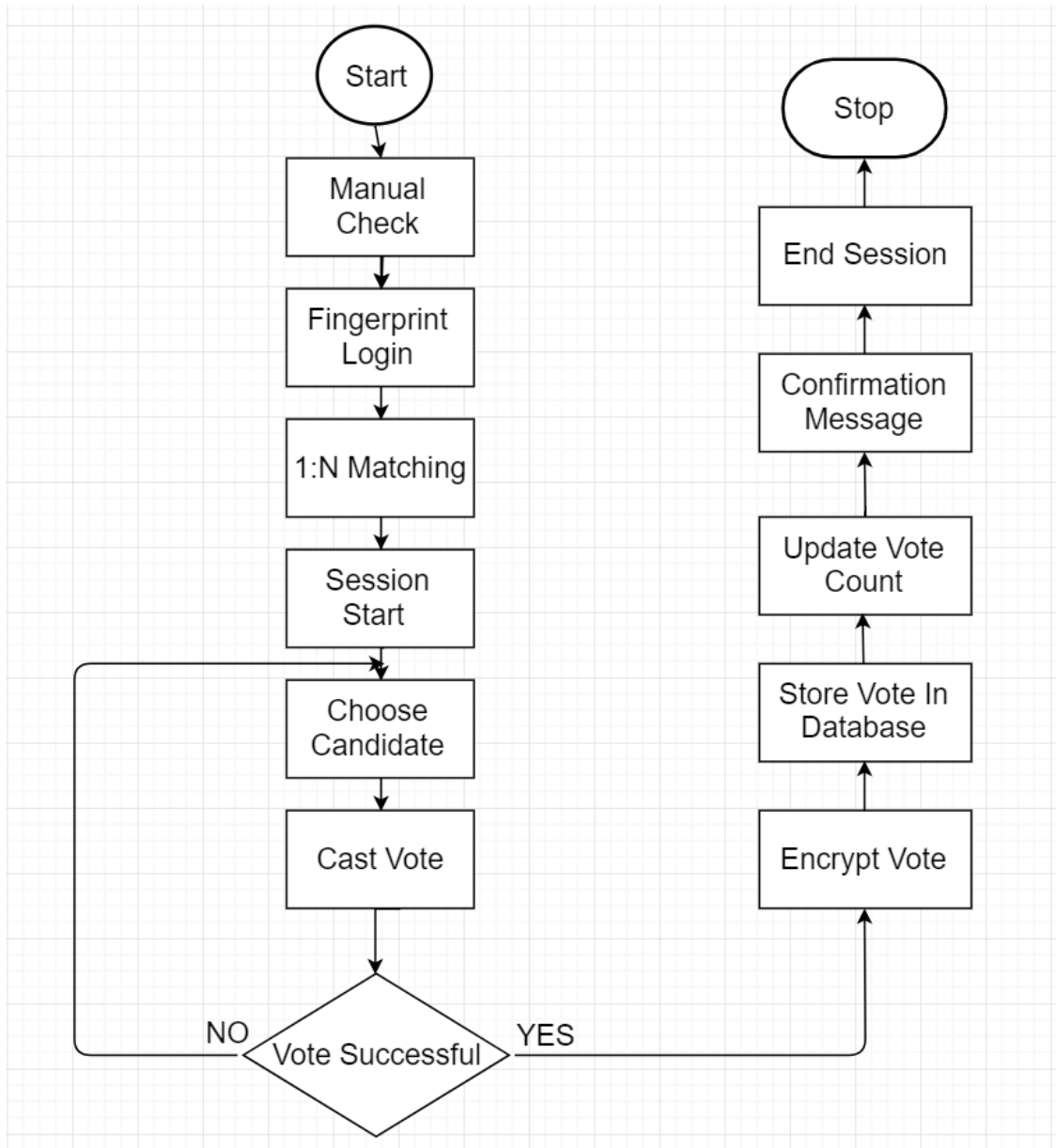


Fig 3.3: System for Casting Votes

Our proposed e-voting system can be categorized into three parts. First one in manual checking, second part is authentication by fingerprint scanner and final part is vote casting. There are several sub category into the system.

The process goes as follows –

1. First one field officer will check the voter identification of voter manually, by checking their name and NID card with the list they have.
2. Next step is fingerprint matching. They need to match their fingerprint by a scanner to login the profile to cast the vote. We assume the fingerprints are stored in a local database. Received template is matched against the stored templates in mentioned local database.
3. After logging on the profile successfully, a SESSION will start and it will run till the end of the user's vote cast.
4. The PROFILE of the user becomes visible on the first page.
5. By scrolling down all the candidate's name, their team name and symbol for the vote will be appeared.
6. Voter can also click on PARTIES to VIEW candidates participating from the same party.
7. The voter will be able to CHOOSE the candidate by simply clicking on the vote button which will be available for every candidate beside their information.
8. After selection, a confirmation windows will be pop up and by clicking on the confirmation voters vote will be CAST the vote. Then, the system will stores the vote, marks the voting status of the candidate and encrypts it with a symmetric encryption algorithm.
9. Files and info remains encrypted in database and immediately vote_count will increment itself.
10. A confirmation pop up rises by saying "Vote stored successfully". The session ends for that particular voter and refreshes for the next voter.

Chapter 4 System Setup

4.1 Hardware Requirements

Name	Intel Core i3-540
Cores	2
Clock Speed	3300 MHz
Typical TDP	75W
Socket	FCLGA1156
Threads	4
Processor Base Frequency	3.06 GHz
Cache	4 MB
Bus Speed	2.5 GT/s DMI
VID Voltage Range	0.6500V – 1.400V

Table 4.1: CPU Specification

Memory:

Physical Memory	4 GB
Disk Space	180 GB

Table 4.2: RAM and Disk

4.2 Software Requirements

OS	Windows 7/8/8.1/10 32/64-bit
Web Server	Xaamp
Languages	HTML,CSS,PHP and C#
Framework	.NET 4.6 or higher
RDBMS	Mysql

Table 4.3: Software Specification

4.3 Development Tools

Tools	Visual Studio 2017 and Sublime Text 3
Fingerprint Driver	Zk4500 Fingerprint SDK
Browser	Chrome, Firefox

Table 4.4: Tools Used

Chapter 5 System Implementation

In previous chapter, we became familiar with the segments the system comprises of. In this chapter, we shall go through the system development process in further detail. Here, we demonstrate the open source codes that we used and also the modified versions of those, which we came up with to get intended result.

The design and implementation stage of this e-voting application was developed with the design of planned system using unified modeling language (UML) and the conversion of the design into the desired design specifications into source code. The main goal of the implementation is to

modify and re-write the source code so that it follows desired outcome. Fundamentally in this project C#, SQL, BOOTSTRAP, CSS and HTML were used to design a user friendly interface (UFI) because these are the more suitable and most preferable programming language used to designing applications in current days.

The proposed system is made up of two parts. One is authentication and the other handles the voting process.

5.1 Database Design

The database design shows the numerous relationships via entity-relationship diagram designed for the project. There are four tables in our database for maintaining the whole process. In 'voter' table, voter's personal data is stored along with his/her fingerprint data. We can collect this data from government because all this data is already stored in government database for NID cards. Then there is 'candidate' table where candidate's information is stored along with party name, and symbol with unique id. We have two more tables in the database, one table is for admin called 'admin' table and another table for storing cast votes called 'votes' table. This table is designed using the required attributes from two other tables: voter and candidate table. We have performed right join in order to grab the information from those tables.

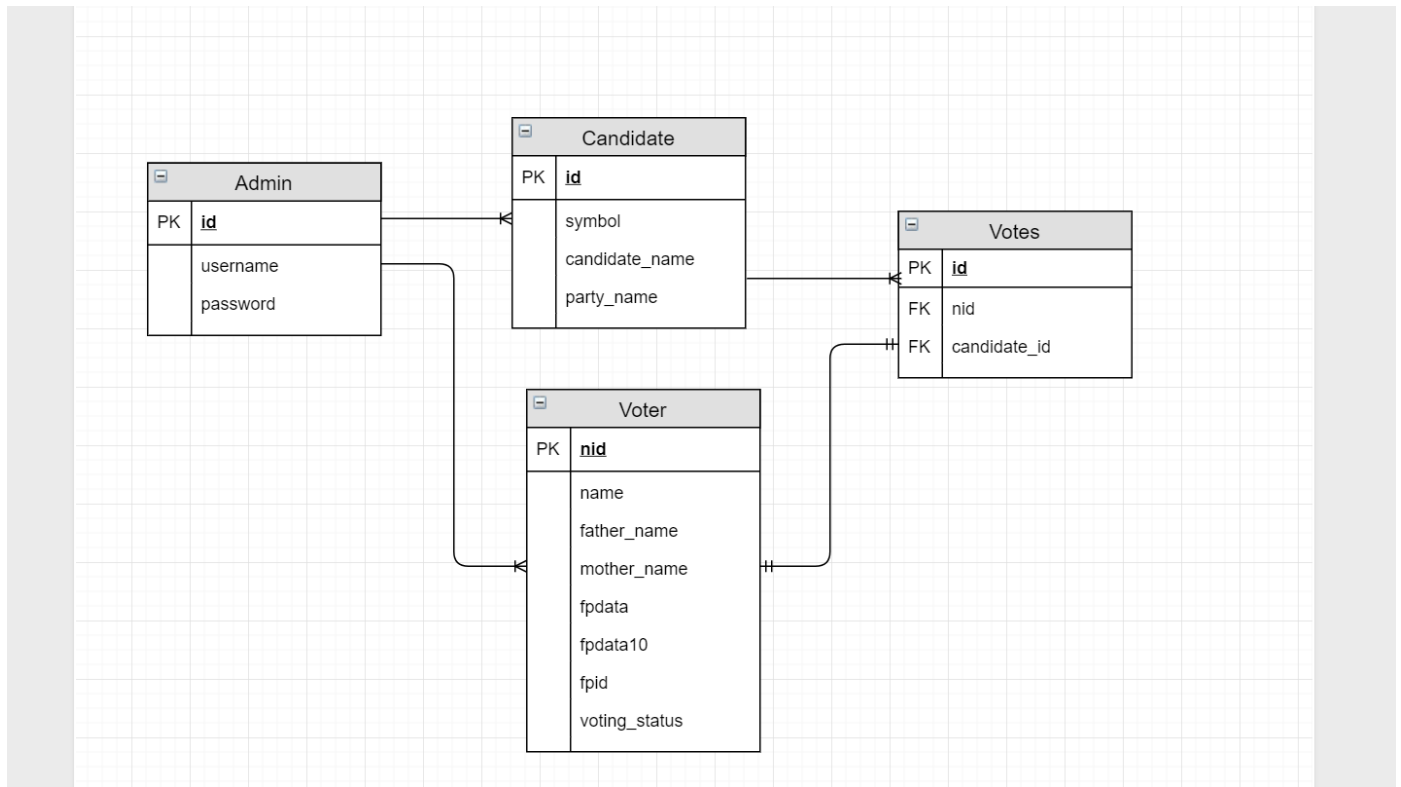


Fig 5.1: ER Diagram of the System

There are separate admin panel where admin can insert, edit and delete a voter as well as a candidate. In the above diagram we have showed cardinality between tables. We have designed our database in a way so that one voter can vote only one candidate. Since nid is unique for each voter, we are distinguishing our voters by their unique nid numbers.

Column Name	Data Type
nid	varchar(13)
name	varchar(20)
father_name	varchar(20)
mother_name	varchar(20)
fpdata	varchar(2000)
fpdata10	varchar(2000)
fpid	int(11)
voting status	tinyint(1)

Table 5.1 Voter Table

Column Name	Data Type
id	int(1000)
symbol	Varchar(12)
party_name	varchar(20)
Candidate_name	varchar(20)

Table 5.2 Candidate Table

Column Name	Data Type
id	int(1000)
username	varchar(20)
password	varchar(20)

Table 5.3 Admin Table

Column Name	Data Type
id	int(1000)
nid	varchar(13)
candidate_name	varchar(20)

Table 5.4 Votes Table

5.2 Authentication

We used ZK4500 fingerprint sensor developed by ZKTeco. This device [17] can capture fingerprint image and upload to the PC by USB interface. It includes scratchproof texture surface with LED indicating status of the sensor. It has a sensing area of 15X18 mm. It takes images of 500 DPI/ 256 gray which is sufficient for rendering a good quality fingerprint image. This high-performance, maintenance-free optical sensor can be connected with desktop/ laptop computer via USB 1.1/2.0 or higher. This USB connection feature makes the fingerprint image processing faster, also the available SDK led us to the path towards better understanding of the inside working process of the sensor.



Fig 5.2: Fingerprint Sensor ZK4500

The ZKFinger SDK-v5.3.0.21-Lite is basically a Windows Form Application developed in several languages. We chose C# and .net version in Visual Studio for studying and testing. It works on ZKFinger Algorithm in version 9.0 and 10.0. We used the updated 10.0 version for our tests.

Along our way, we changed the form and created new forms to proceed with our application.

In the beginning, the sensor overwrote each template with the last stored template. To overcome this we tried Guid. The following code generated random strings which enabled us to store multiple fingerprints for identification:-

```
string randoms = Guid.NewGuid().ToString().Replace("-", string.Empty).Replace("+", string.Empty).Substring(0, 4);
```

Initially, the btnVerify_Click(object sender, EventArgs e) worked for 1:1 fingerprint verification but there were no functionalities for identification (1: N). We had to delete FPCacheDBEX, this is a buffer for storing files from the external interface which has no utility in our system. Then we created a new buffer (FPC handle to store registered images) and used IdentificationInFPCacheDB(fpcHandle, e.aTemplate, ref score, ref processedNum) for matching query image against stored templates. This is where we moved toward identification from verification.

Storing the templates to the database was a crucial task. The SDK was able to populate cache memory of the sensor with just registered templates, thereby it didn't work if the sensor was disconnected. In that case, we had to register each time we wanted to authenticate. For this, we planned to store fingerprint templates to the local database and load those in the SDK whenever needed. For this we used the open source mysql.dll library, using this we established connection to the database and stored the fingerprints along with other voter information. This is how we completed the 'Registration' process.

Our next objective was to load the registered fingerprints and make the 'identification' process work using those, The fpcBuffer was being populated just fine but it was not really working. To resolve this, we began to look deep into the fingerprint templates. These are variant length templates, which means length can vary from user to user or finger to finger. It is stated that in ZKfinger 9.0 algorithm the string length is in between 600 to 900 and in 10.0 version it is 900 to 2000. For verification, template length [18] can be 310 bytes but for verification 1152 byte is required. As we checked we noticed that in the process of storing and loading, data was getting trimmed somewhere. To make it work, instead of handling templates in bytes or blobs, as directed in most resources, we directly stored the BASE64 encoded sRegTemplates produced by the sensor. This solved the problem and our 'Authentication' process began to work correctly.

5.3 Session Generation

We have used session in our program in order to maintain login and logout for both admin and voters. Session is about storing data across page requests. A session starts when someone first logs into a system and ends when the job is finished. In our case when a voter logs into the system, his/her session starts. Then the voting panel appears, where voters can cast vote. After casting vote, a pop-up button appears asking for confirmation. Just as the voter clicks 'Confirm', all cache data of that user is wiped out and he/she is automatically logged out from the system. Then the page gets refreshed for next user. Even if the last user tries to log in again, the system denies it. Thus we ensure 'integrity' by including one-time login in our proposed system.

5.4 Interaction View

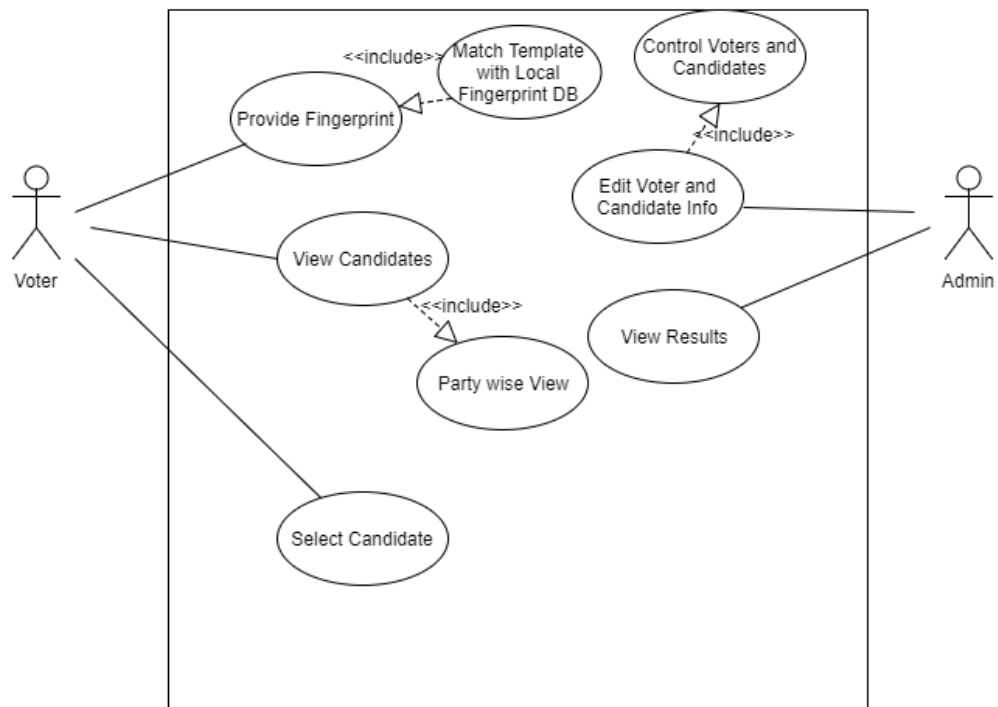


Fig 5.3: Use Case Diagram of Proposed System

The above diagram shows the basic functionalities of the project. The prime actors, who interact with the system directly are – voters and admin.

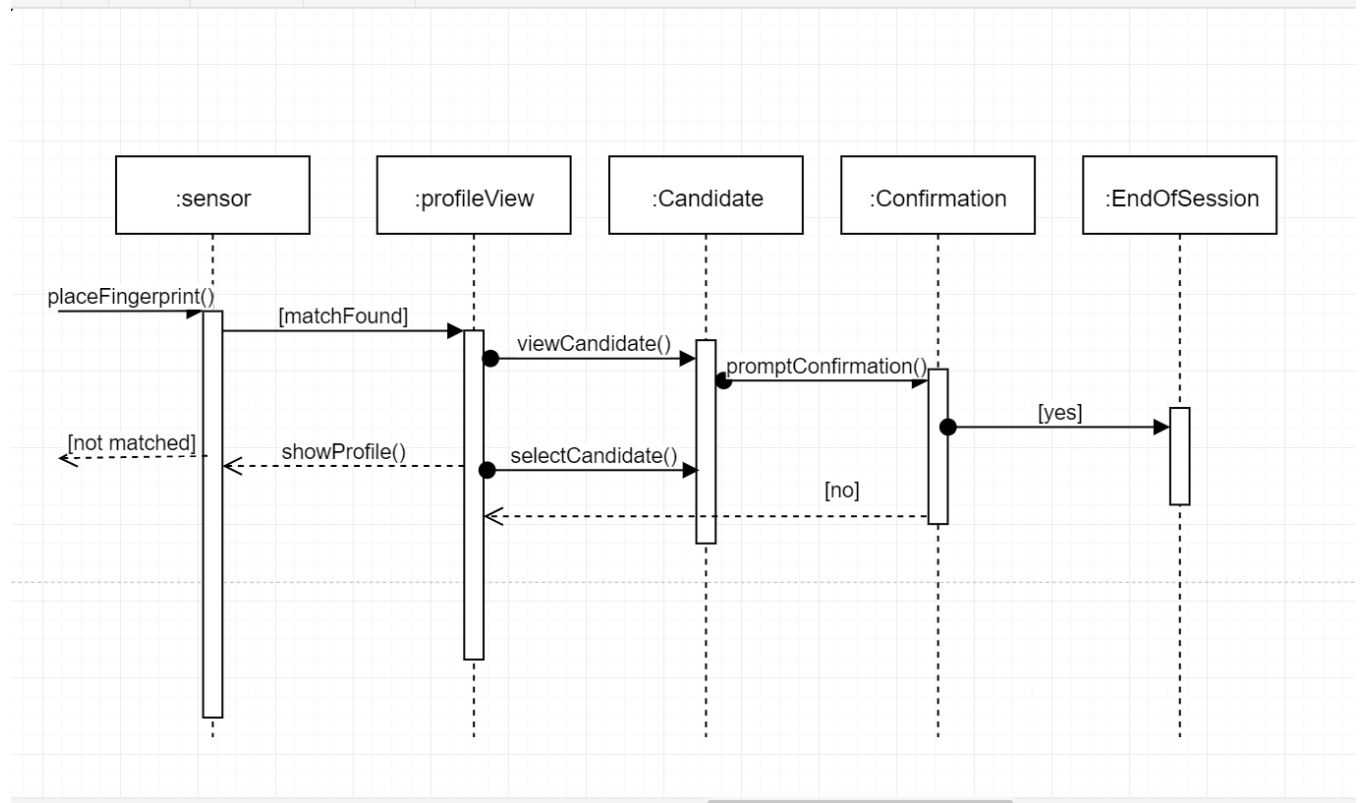


Fig 5.4: Sequence Diagram of Voting Process

This diagram shows interaction with time. Only after successful login, a sensor's lifeline (user's session) starts. Right after that, the user is taken to his/her profile and from there to candidate list. After casting votes, voters are asked for confirmation. From there with a positive answer, the session ends. Else the voter is taken back to the candidate page to where he/she can re-cast vote.

5.5 Encryption Process

We have encrypted our vote so that no one can see or change any of the votes. We have encrypted votes in a way that votes will be stored in votes table in the database and in the result panel, the votes will be decrypted and show the results. There are no way, an intruder can see or edit the votes. We have used TDES algorithm for the encrypted process which is a symmetric encrypted algorithm. It stands for Triple DES where DES is applied in three stages.

In our case we have used this algorithm because we are not using any centralized network, data is at-rest in our system and we just need one for encryption and decryption.

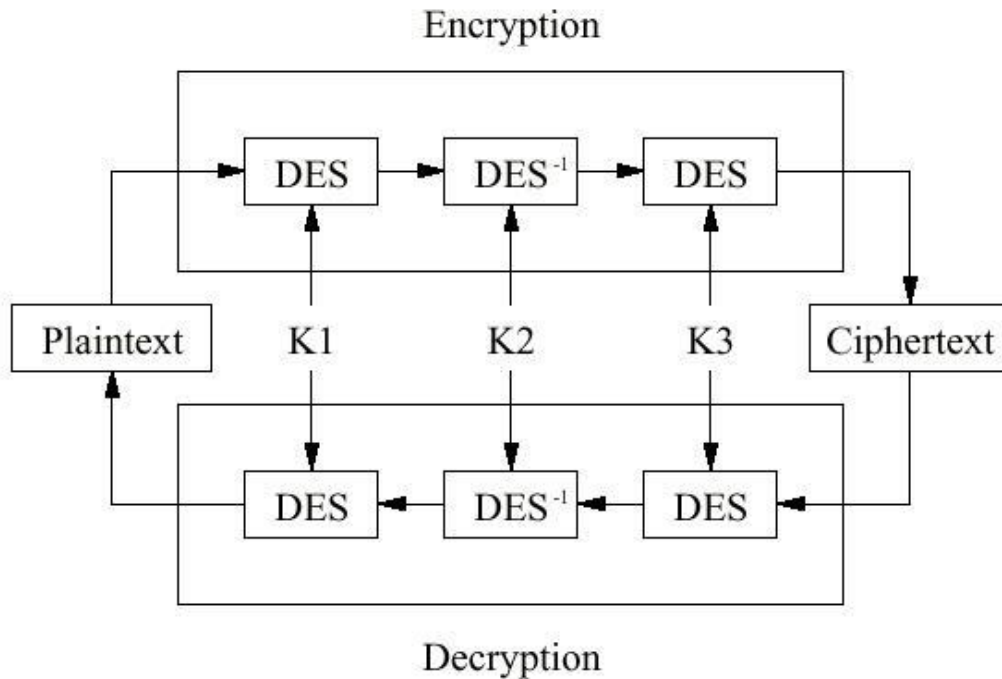


Fig 5.5: Block Diagram of TDES Algorithm

In the above block diagram, we can see the whole TDES process. Normally it takes 64 bit block of plain text and encrypts it with key k1. Then decrypts the cipher with k2 and encrypts it again using k3. We end up with 64 bit cipher text. In the decryption process the same keys are used but in reverse order. The whole process works with $56 \times 3 = 168$ bits length key size which is one of the strong feature of this algorithm and that is the key difference between actual DES algorithm and TDES algorithm.

Chapter 6 Experiment & Results

System implementation and testing are two important stage without which the system cannot be released for use. Testing is vital to the success of any system. Testing confirms that all courses are according to specification. The logical and physical design is continuously examined to ensure that test data are verified for exactness and accuracy. Different modules are tested individually and are made error free. In our case, another kind of problem can occur which is hardware problem. For example, we are using fingerprint reader to detect the voter and for some reason, the reader might not be able to read the fingerprint properly. And there can also be other issues, like the fingerprint sensor detecting a false match when actually no match should be found. We looked into these by testing the authentication module. Later on, we checked the voting panel and the admin panel separately. Finally, we integrated all the modules together and ran different test cases to check whether it works smoothly or not.

6.1 Experiment

6.1.1 Testing the SDK

First of all, we tested the provided SDK with one fingerprint for verification (1:1). When it was successful, we moved on to 1: N matching, also known as, identification. First, we were testing with very few fingerprints. In the beginning, some registrations were failing. Also, we were only getting a match for the image that was stored last. Later on, after clearing and handling the sensor more carefully, we achieved a better result. After several trials, the tweaked SDK yielded matches for all the registered templates.

6.1.2 Experiments step by step

First of all the portal appears with options for both admin and voters. These pages are made using .hta extension, so they work as HTML application, enabling both user interface design and scripting languages like JavaScript or VB.net. Then JavaScript Function RunExe() is used for

connecting .hta file with sensor SDK. Based on chosen option, the user can be taken either to the sensor SDK or to another HTML application.

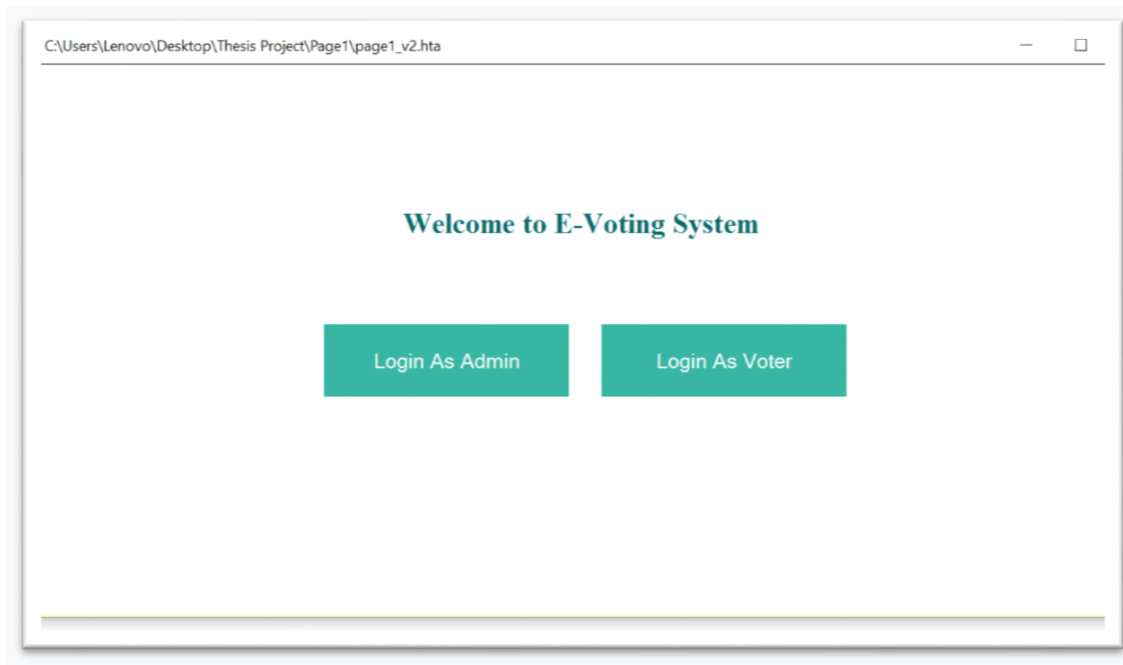


Fig 6.1: Front page of the application

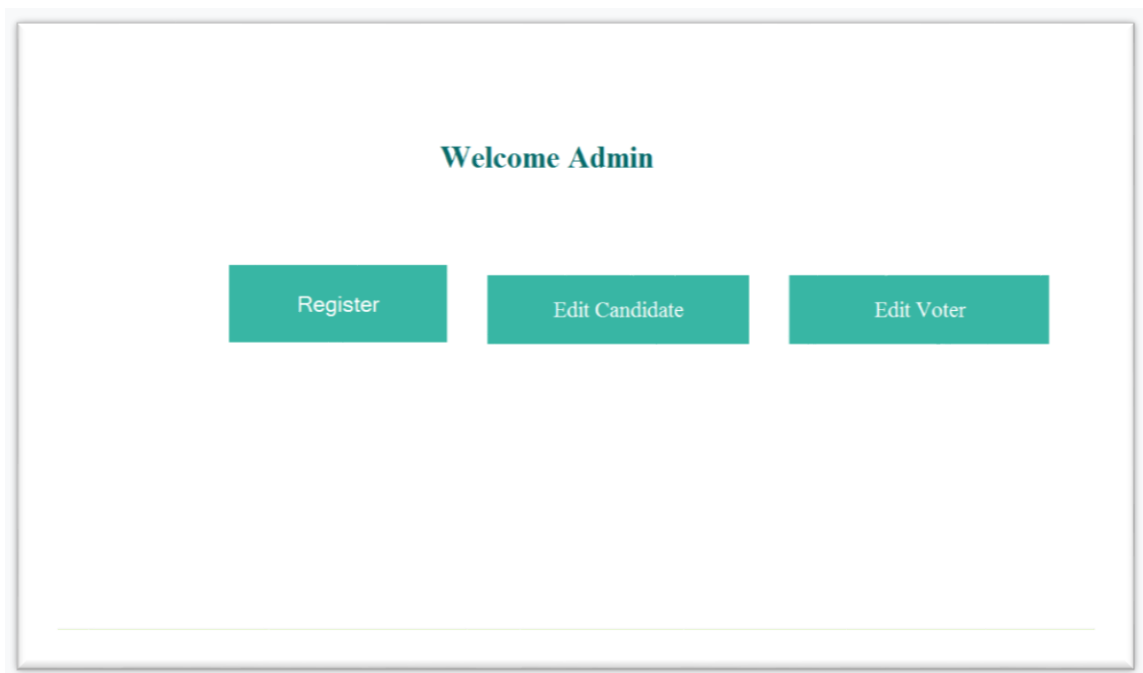


Fig 6.2: HTML application with admin functionalities

It takes the admin to Windows form Application called 'Demo' for enrolling voters or to candidate and voter editing pages –

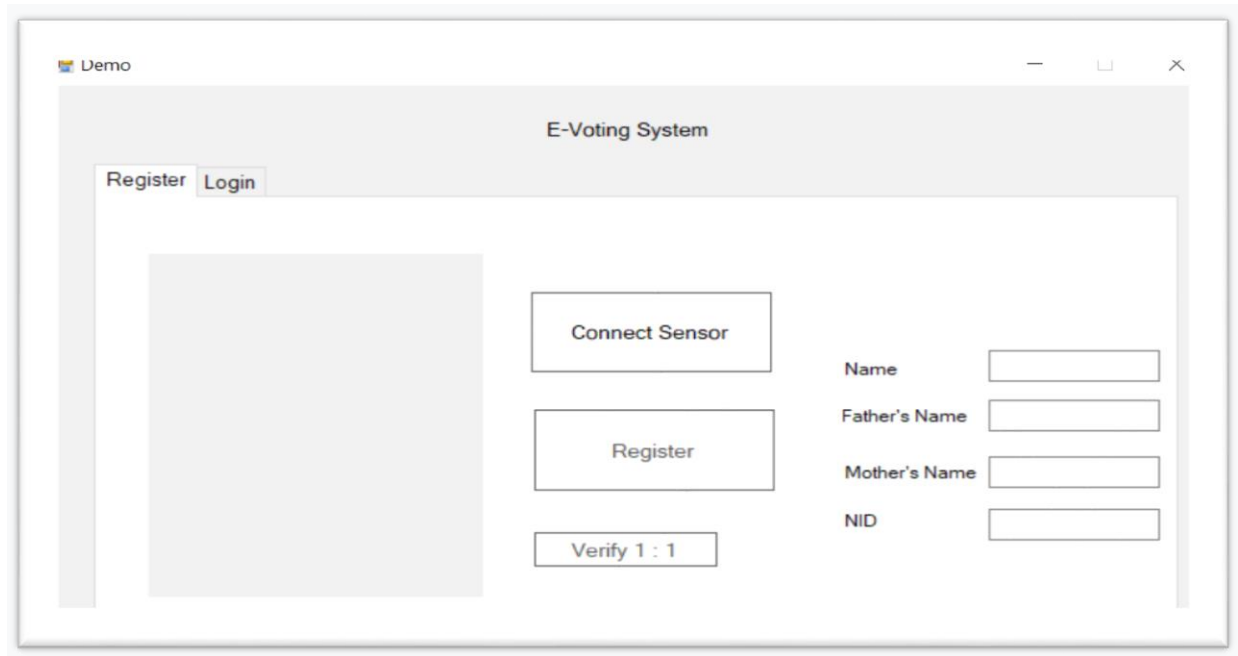


Fig 6.3: Voter Registration

Here we can register each and every voter smoothly with their information and fingerprint. Soon as the Register button is clicked, the information is passed and stored in the 'Voter' table of the local database.

ID	Voter	Father	Mother	NID			
2	Babul Mitra	Mujibor Rahman	Kamrun Nahar	1234567891	Add	Edit	Delete
3	Afroz Rahman	Hamidur Rahman	Rokeya Begum	2147483647	Add	Edit	Delete
4	Fairouz Sharif	MD Shahjahan	Ummul Khair	2147483647	Add	Edit	Delete

Figure 6.4 Edit Voter Page

Add New Candidate




Symbol	Candidate Name	Team Name		
	Donald Trump	Congress	Edit	Delete
	Barack Obama	Republic	Edit	Delete
	Hillary Clinton	Socialist	Edit	Delete

Fig 6.5: Edit Candidate Page

Next comes the modules of the voter. If the voter logs in then he/she will pass through the following –

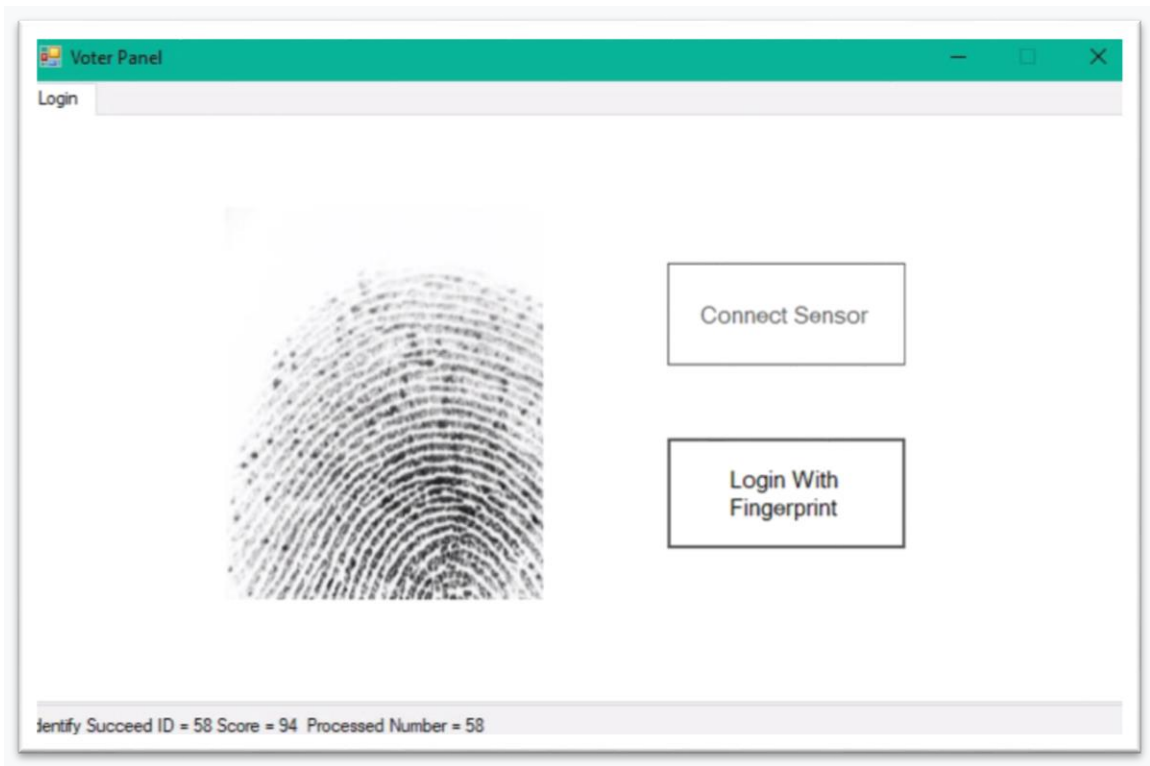


Fig 6.6: Logging in with Fingerprint

This part performs the authentication of voters. As we can see, in the form the identified ID is being shown along with a matching score. A score over 60 is considered a good fingerprint and is accepted for continuing the voting process. Then the system retrieves the information of the voter who just logged in and it shows up along with voting panel like the following –

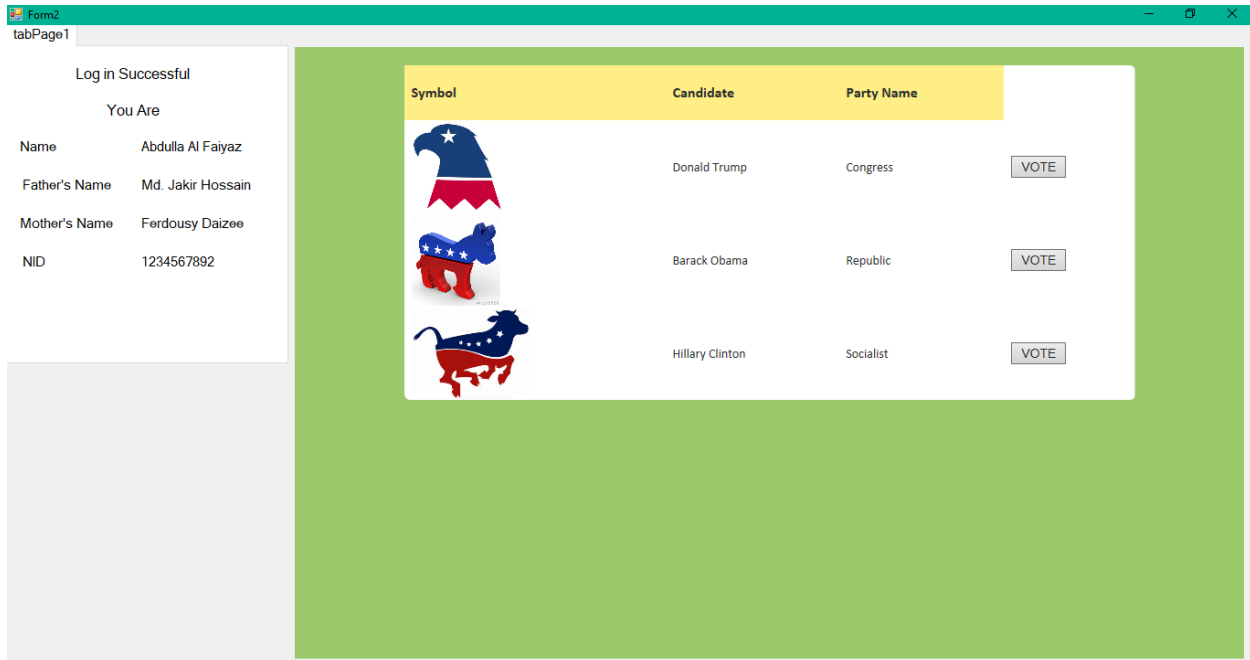


Fig 6.7: Profile and Voting Panel

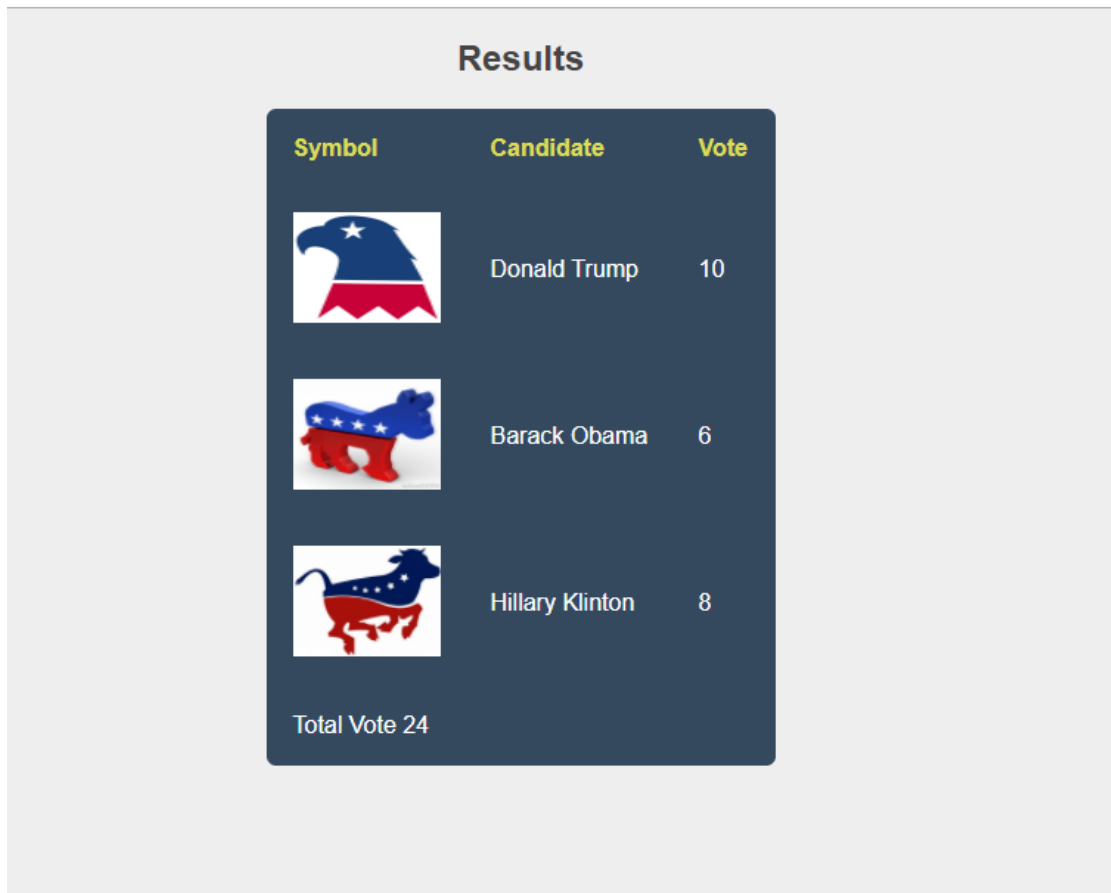


Fig 6.8: Final Countdown

6.2 Results & Discussion

6.2.1 Findings

We populated the local database with 100 voters to get approximate calculation in aspect of several parameters per booth. Here we are going to discuss those calculations one by one –

- **Time:** If the entire process goes smoothly, it takes around 10 to 12 seconds for a user to cast to vote. From this, it can be said it would take maximum 1 minute in case of any technical difficulty. So if we consider 500 voters present at a booth, using our system, on an average it will take 6000 seconds which is approximately 1.667 hours. In worst case,

where technical difficulty arises for each voter, which is highly unlikely, it can take 22 seconds. So for 500 voters it can take approximately 3.056 hours. On an average, it can be 20 seconds per user and 2.778 hours for 500 voters. All the cases, present lot less time than traditional paper-based systems. The inclusion of USB enabled sensor and its integration with the local database has made this possible. If microcontrollers were used, the process of authentication could have taken much more time resulting in a system slower than the proposed one.

- **Cost:** For measuring cost, we refer to the cost of ZKTeco sensor which is 4500 TK and as we already know this is a desktop application so we need a minimum requirement ready desktop for the operation and our database is locally stored in the desktop, so we don't have any extra server cost. For a minimum requirement ready desktop which is suitable to run our application may cost 20000 to 25000 taka.
- **Memory:** As we developed and tested the application with relatively small dataset compared to large-scale scenarios, the total size of it turned out to be 18 MB.
- **Security:** Since the proposed system, is a standalone application it does not require any kind of network connectivity. Thus, it is completely safe from network attacks and hacking. However, files inside the system need to remain encrypted for protecting data in case of loss or mal-handling. Here, we encrypted the stored votes using symmetric key encryption algorithm TDES (Triple Data Encryption Standard) algorithm. Thereby, even if anyone in the booth tries to hijack the data inside, he/she will be unable to do so since it will be locked and unreadable.
- **Number of People involved:** In the proposed system, only one person is needed for technical assistance per booth. Besides, there could be a number of field officers for manual checking according to the number of booths. So, if there are 10 booths, we can have 10 technical assistants and 5 field officers, in total 15 people need to be present at the poll during election hours. Thus we are ultimately reducing the total number of people involved. However, certain amount of trainers might be required for conducting campaigns but that is a different issue.

6.2.2. Problems Faced During Implementation

As we went along with these experiments, we faced several issues that should be brought into consideration:

- **Handling built-in SDK Functions:** First of all, we wanted to study the SDK for determining threshold score to find the best similarity score and threshold range for matching fingerprints. But we discovered that the actual code has not been released by the company, only functionalities can be edited as needed. Then, as we focused on those, it turned out that even all of those are not properly functional in real. So, we had to tweak various functions of the SDK to make it work as we needed.
- **Fingerprint template length:** Enrolling fingerprints to the database was a crucial task, but later on we were able to achieve it by establishing connection with local database from C#. Another job was populating sensor cache. First, we tried to do this by storing images or bytes. But the sensor actually needed template strings and that with exact length. By comparing the length of fingerprint data used in the built-in identification process with the one we are loading from the local database, we learned about the length and used it in the right way.
- **Integration of back-end and front-end:** Integrating the C# developed authentication module with the PHP written back-end was a challenge. First of all, we planned to develop back-end in PHP and convert it entirely to C#. But it was really time-consuming and hazardous. After research of few days, we came up with few third-party resources and compilers like Peachpie that supports this integration.

Chapter 7 Conclusion

7.1 Future Work

- Our system serves as a standalone device for election at that particular polling center. If we consider upper levels like from upazilla to districts, from districts to divisions, further networks might be needed. Different database systems can be introduced in such systems for integrating local databases with a centralized one.
- In a broader perspective, multiple server management will become a crucial issue. Also, designers must take care of potential network attacks and use proper communication protocols as necessary.
- In the proposed system data is at rest. Thereby, we chose secret key encryption. But when data will be in motion, for example in different levels of networks, applying public key encryption will be a better choice for data protection.

For the development of proposed e-voting system, we studied basics of fingerprint characteristics and matching techniques from various papers. It is an ideal solution for optimizing cost and risks of vote tampering. Also, the system includes fast biometric authentication which is a strong feature to identify any legitimate voter.

The system provides controlled access to admin and voters. One user remains alive only in his/her session and only if he/she has not voted before. Admin is privileged with the functionalities of editing registered voters and candidates. Record of detailed voting info does not exist alongside the voters. Thus no one can influence or look into a particular person's vote. Chances of occurrence of 'faked votes' are also almost zero. Thus all possible risks are taken into consideration and eliminated. We believe, if availed in large scale, our system can serve as a credible one and win back the lost trust of people when it comes to sensitive issues like election in a democratic nation.

References

1. Schedler, A. The Menu of Manipulation. *Journal of Democracy*, Vol. 13, No. 2, pp: 36-50. 2002.
2. Schaffer, F.C. (2007). Why study vote buying?. In F.C. Schaffer. (Ed.), *Elections for Sale: The Causes And Consequences of Vote Buying* (pp. 1–16). Boulder, Colorado: Lynne Rienner Publishers.
3. Mollah, M. A. H., Jahan. R. Parliamentary election and electoral violence in Bangladesh: the way forward. *International Journal of Law and Management*, Vol.60 No.02. February 2018.
4. Alvarez, R.M., Katz, J.N., Stewart, C., Rivest, R.L., Ansolabehere, S., Hall, T.E. (2012). *Voting: What Has Changed, What Hasn't, & What Needs Improvement*(Report No. 6) Retrieved from <http://vote.caltech.edu/reports/6>
5. Djanali, S., Pratomo, B.A., Cipto, K.P.N., Koesriputranto, A., & Studiawan, H. (2016). Design and development of voting data security for electronic voting (E-Voting). 4th International Conference on Information and Communication Technology (ICoICT). doi: 10.1109/ICoICT.2016.7571928
6. Kumar D. A., Begum T. U. S. A Novel design of Electronic Voting System Using Fingerprint, *International Journal Of Innovative Technology & Creative Engineering* Vol.1 No.1, pp:12-19, January 2011.
7. Abdulhamid, S. M., Adebayo, O.S., Ugiomoh, D.O., Malik, A. M. D. (2013). The Design and development of Real-Time E-Voting System in Nigeria with Emphasis on Security and Result Veracity. 4th International Conference on Information and Communication Technology (ICoICT). doi:10.5815/ijcnis.2013.05.02
8. Kang, S., Lee, I.Y. (2006). A Study on the Electronic Voting System using blind signature for Anonymity. *International Conference on Hybrid Information Technology*. doi: 10.1109/ICHIT.2006.253678
9. Al-Sammari, A. F. N., Tessaris, S. (2011). Vote Verification through Open Standard: A Roadmap. *International Workshop on Requirements Engineering for Electronic Voting Systems*. doi: 10.1109/REVOTE.2011.6045912
10. Rahim, A. K. A., Folorunso, O., Sharma, S. K. An Improved Dynavote E-Voting Protocol

- Implementation. International Journal of E-Adoptation, Vol.3 No.3, pp: 44-61. January 2011.
11. Mahmood. T., Zoha. S.M., Das A.K. (2016). Android-based Smart Voting System. Retrieved from http://dspace.bracu.ac.bd:8080/xmlui/bitstream/handle/10361/6376/10301006%2c%2011301007%20%26%2011301002_CSE.pdf?sequence=1&isAllowed=y
 12. Anis, M. A., Rahman, H., Alam, J. S., Nabil S. I. and Hasan, S. M. 2014 Development of Electronic Voting Machine with the Inclusion of Near Field Communication ID Cards, Biometric Fingerprint Sensor and POS Printer. Retrieved from <http://dspace.bracu.ac.bd/bitstream/handle/10361/3967/ThesisReportFinalv1.pdf?sequence=1>
 13. Okediran, O. O., Olabiyisi, S. O., Omidiora, E. O., Ganiyu, R. A. A Survey of Remote Internet Voting Vulnerabilities. World of Computer Science and Information Technology Journal (WCSIT), Vol.1 No.7, pp: 297-301. (July 2011)
 14. Jain, A.K., Feng, J., Nandakumar, K. Fingerprint matching. Computer 2010, 43, 36–44.
 15. Stine, K. Dang, Q. Encryption Basics. Journal of AHIMA, Vol.82 No.5, pp: 44-46. May 2011.
 16. Lozupone. V. Analyze encryption and public key infrastructure (PKI). International Journal of Information Management, Vol. 38 No.1, pp: 42-44. February 2018
 17. ZKTeco. ZK4500 Fingerprint Sensor. (n.d.). Retrieved from <http://www.zktechnology.com/zk/qeuryProductDetailed.do?id=392>
 18. ZKTeco. ZKFinger10.0 User Guide. (n.d.). Retrieved from <https://www.scribd.com/document/269240968/ZKFinger10-0-User-Guide>.

