Modern University for Technology and
Information Faculty of Computers and
Information Department of Information Systems

Electronic Voting Using Fingerprint

Computer Science Graduation project 2022

Submitted by

Youssef Mostafa Mohamed

Mostafa Mahmoud Ahmed

Mohamed AbdElmoaty Mohamed

Mohamed Mohsen Mohamed

Kerolos Naem Habib

Mohamed Khaled Mahmoud

Supervised by

Prof. Dr. Alaa Abd El-Raheem

T.A Mohamed Amin

T.A Aya Khaled

Acknowledgment

Our greatest in deep and gratitude to ***Modern University for Technology and Information (M.T.I)*** for the great role it played in forming our character and personalities, providing us with all means necessary for our educational development and providing us as well with a wonderful environment to learn in throughout our educational years.

This would not have been possible unless ***Prof. Dr. Olfat Kamel,*** President of the M.T.I University has established such a learning environment, providing the most effective facilities for our job to be done.

It is a pleasure to thank those who maneuvered this Possible,***Prof. Dr. Nabil El Deeb*** Vice President of M.T.I University.***Prof. Dr.Mohamed Taher ElMayah*** Dean of the faculty of Computer Science for his unlimited help, support, encouragement and guidance.

A Special thanks to our head of the Computer Science Department ***Prof. Dr. Hanafy Ismail***, we are Honored and got the pleasure of working with as our supervisor and our teaching Assistants ***Mohamed Amin and Aya khaled.***

As well as the Department staff, ***Prof. Dr. Alaa Abd El-Raheem and Prof. Dr. Hesham El Deeb*** for their effort along our university studies.

Special thanks to ***Prof. Dr. Hafez Abdel-Wahab***, head of the Information System Department, as well as ***Prof. Dr. Mahmoud El-Shishtawy*** for their effort along our university studies.

Special thanks to ***Prof. Dr. Emain Taha,*** head of Basic Science Department. As well as the department staff, ***Prof. Dr. Elsayed Bakkar*** for their effort teaching us the basic for our computer science degree, ***Prof. Dr.Rasha Mohammad*** and ***Prof.Dr.Rania Ahmed*** Lecture in Basic science Department.

Abstract

Voting is a fundamental process in a democratic system. It is a chance for the citizens of a country to have a fair say in the people who represent them or an argument that impacts them. Voting and participating in elections is one of the main responsibilities of citizens. In the present day, the voting process is fairly straightforward. First, citizen registers his identity to vote, studies the candidates and issues, look up their polling location, and then casts their ballot during the election.

Problem solution This project discuss and propose a simple solution to help fix the issues with the traditional voting techniques used nowadays by most countries and governments , the design and development of a Biometric Electronic Voting System. The suggested fingerprint voting system allows users to scan their fingerprints, in order to check his identity by comparing his current fingerprint with the one already stored in the system's database, using optical fingerprint scanner ZKT-Eco 4500, Matching algorithms match the fingerprint as a method for authentication after the profile of citizen is found to secure the voting operation.

A friendly graphical user interface developed by using Angular frame work, java script programming language and linked with Mongo DB server database for data manipulations. Once the voters complete the identification process, they will be allowed to cast their vote using friendly user interface developed using Angular framework. The counting of the votes will be automatically send to the server once the process is complete and that makes the voting process efficient, fast, and secure.

Table of Contents

Table of Figure:

Table of Tables

Chapter One
Introduction

1. INTRODUCTION:

Through history voting process has been used in every debate and decision making, using the recent day technology in the voting process is a big step which solve many problems with the old voting techniques.

Egypt uses paper ballot system for all its elections through the history. This system involves printing ballot paper on which voter will choose his candidate and then the votes are collected in a sealed box then transfer to the counting phase where the votes are counted to publish the result later after the election is ended.

This entire system, as with any other electoral system has many problems. This problems could be transparency, cheating, security, and cost. Therefore, this leads to a loss of authenticity and trust in the electoral process.

This chapter introduces the background of our project and describes the scope of the document and gives a brief explanation of the document. Section 1.1 shows the main project problem definition, and objective and gives the solution approach of this project. Section 1.2 describes the motivations that this project is going to give. Section 1.3 describes the basic concepts of the project. Section 1.4 describe the ream management and the tasks of the project .Section 1.5 discusses the documentation organization of the project.

1.1 Problem definition:

The traditional voting system have many problems, examples of this problems is "Paper waste" every voter use a ballot paper which is thrown away after election is finished, "Cost of money" ballot papers, members of election process, and the transportation of the boxes cost a lot of money**,** "Human error" human make mistakes, during counting phase the votes are counted manually which lead to error percentage in the result, "Time cost**"** voting and counting results process take a lot of time due to human factor. Also "Transparency" members responsible for counting could cheat duo to weak security by changing the votes to their own profit. With this in view, it is necessary to adopt a better method of the electoral process.

1.2 Project Objective:

This project aims to solve the problems like paper waste, human error, security, and time cost by using the latest technologies of computers and biometrics authentication which will also use pattern recognition and image processing techniques to identify and recognize the voters from their fingerprint, so they could cast their votes safely.

The result of this work is an electronic voting machine that is power efficient, secure and easy to use for the electorate. It characterizes elections with credibility, integrity and vote authenticity.

1.3 Solution Approach:

1. Literature review for similar systems.
2. Defining all image processing techniques and pattern recognition matching techniques for fingerprint.
3. System design which include the flow charts, prototypes, and database scheme diagram needed for the system.
4. System implementation which includes to implement the matching algorithm and the GUI for the system by using Python, and C#.net framework.
5. System testing.

1.4 Project Management:

1.4.1 Team Members:

1. Youssef Mostafa Mohamed Ahmed
2. Mostafa Mahmoud Ahmed
3. Mohamed AbdElmoaty Mohamed
4. Mohamed Mohsen Mohamed
5. Kerolos Naem Habib
6. Mohamed Khaled Mahmoud

1.4.2 Task Description:

- **Voting System (GUI):** with simple (GUI) to achieve a whole voting process.
- **Pre-processing:** after receiving the fingerprint image it must run through some operation first like histogram, binarization, and thinning before feature extraction phase.
- **Matching:** extracting features from fingerprint image like termination, bifurcation, and dots and save them as key points. Key points are used in 1:1 matching with each other with different algorithms to give best accuracy and identify the person fingerprint.

1.4.3 Project Resources:

The project is divided into three main parts:

Part 1: Image processing: pre-processing and feature extraction and matching algorithms implementation, this part is assigned to all students.

Part 2: implementation of database sample and GUI subsystem to combine all the sub-systems together in one web application, this part is assigned to all students.

Part 3: Evaluation and testing: assigned to all students.

1.4.4 Task Definition:

1. Literature Review (All Students)
   a. Literature Survey : (Mostafa, Khaled, Kerolos)
      i. Brute force matching
      ii. KD-Tree
      iii. LSH (Locality Sensitive Hashing)
      iv. HNSW (Hierarchical Navigable Small Worlds)
   b. Related Works (All students)
      i. E-voting in India
      ii. E-voting in Nigeria
      iii. E-voting in Estonia
2. Research for Image processing and Matching Algorithms:
   a. Image Processing Techniques for Fingerprint (Youssef, Abdel moaty, and Kerolos).
   b. Matching Algorithms for fingerprint (Mostafa, Youssef).
3. System Implementation:
   a. Database module (Mohsen, Youssef).
   b. Image processing module (Khaled, Kerolos, Mostafa).
   c. Matching algorithm (Mostafa).
   d. GUI Module (front-end) (Youssef).
   e. Back-end implementation (Abd-El Moaty).

4. Testing:
    a. Unit testing (Mohsen, Youssef).
    b. Module testing (Khaled, Mostafa).
    c. System testing (Abd-El Moaty, Kerolos).
    d. Experiments (matching algorithm)
5. Documentation (All Students):
6. Demo (All Students).
7. Presentation (All Students).

TASK SHEET

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		**E-Voting Using Fingerprint**	**4 wks**	**Tue 10/5/21**	**Mon 11/1/21**		**All Students**
2		Litrature Review	2 wks	Tue 10/5/21	Mon 10/18/21		Youssef Mostafa,Mohamed Abdel Moaty,Mohamed Mohsen
3		Research Image Processing Techniques for fingerprint	2 wks	Tue 10/19/21	Mon 11/1/21	2	Mostafa Sallem,Mohamed Khaled,Kerolos Naem
4		Research Matching algorithms for fingerprint	2.2 wks	Tue 11/2/21	Tue 11/16/21	3	Youssef Mostafa,Mostafa Sallem
5		Implementing Matching Algorithm using Python	4 wks	Wed 11/17/21	Tue 12/14/21	4	Mostafa Sallem,Mohamed Khaled,Kerolos Naem
6		Implementing Desktop System Prototype	4 wks	Wed 12/15/21	Tue 1/11/22	5	Youssef Mostafa,Mostafa Sallem,Mohamed Mohsen
7		Implementing Web application System prototype	4 wks	Wed 1/12/22	Tue 2/8/22	6	Mohamed Abdel Moaty,Mohamed Khaled,Kerolos Naem
8		Matching Algorithm Expermients	2 wks	Wed 2/9/22	Tue 2/22/22	7	Mostafa Sallem
9		System Testing	2 wks	Wed 2/23/22	Tue 3/8/22	8	All Students
10		System Demo	2 wks	Wed 3/9/22	Tue 3/22/22	9	All Students
11		Project Documentation	4 wks	Wed 3/23/22	Tue 4/19/22	10	All Students
12		Project Presentation	2 wks	Wed 4/20/22	Tue 5/3/22	11	All Students

Figure 1:1 Project Tasks Sheet

Gantt chart:

A Gantt chart commonly used in project management is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflect the start date, duration, and end date of the activity, Figure 2.3 shows the Gantt chart:

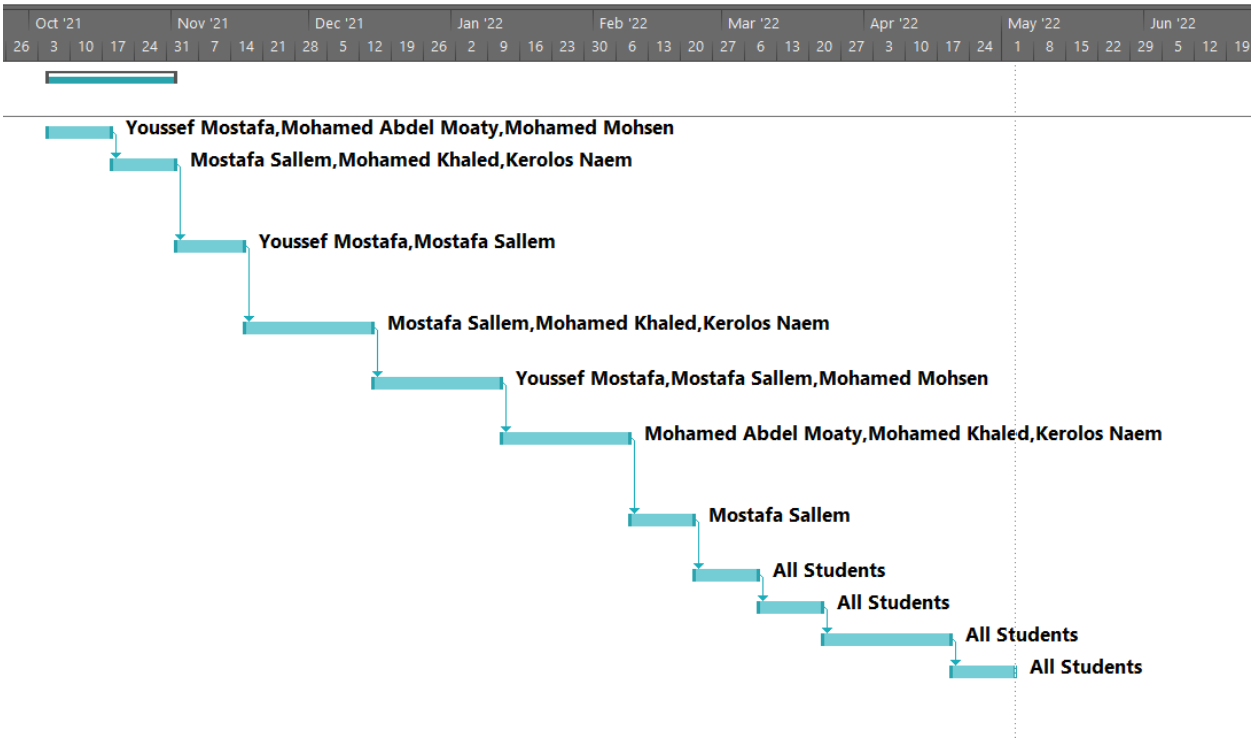

Figure 1:2 Gantt chart tasks Overview

1.5 Documentation Organization:

The rest of the documentation is organized as follows:

- **Chapter Two:** literature review including Matching, Image processing, histogram & binarization techniques.
- **Chapter Three:** introduces the related work & similar applications will be pointed out.
- **Chapter Four:** introduces the implementation of the most well-known algorithms with some modifications. The platforms used and the algorithms for each implementation. And then the Graphical User Interface (GUI) will be presented.
- **Chapter Five:** introduces the used data set as well as our dataset and testing and evaluation of the system, where some experiments are carried out and the results will be stated.
- **Chapter Six:** introduces the conclusion established based on the results of the experiments done, and the future work.

Chapter Two
Literature Review

2. Literature Survey:

In this chapter is an overview about image processing techniques, and the most popular methods & algorithms used for feature detection, minutiae recognition, and fingerprint authentication in real-time.

2.1 Image processing techniques:

In the preprocessing phase we apply a number of transformations and filters on the fingerprint image in order to be able to extract features in the following phase. Image processing approaches may vary, the following ones are the ones that are of concern to getting the best fingerprint enhancement results.

2.1.1 Histogram equalization

An image histogram is a type of histogram that acts as a graphical way to represent the tonal distribution in an image. It plots the number of pixels for each value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution immediately.

Histograms are made up of bins, each bin representing intensity. The histogram is computed by checking all pixels in the image and assigning each to a bin depending on the pixel intensity. The final value of a bin is the number of pixels assigned with it.

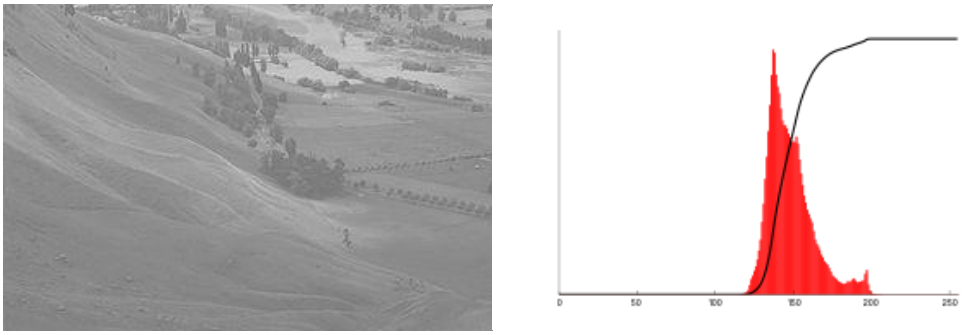

Figure 2:1 Image before histogram Equalization

Image and the corresponding histogram before equalization

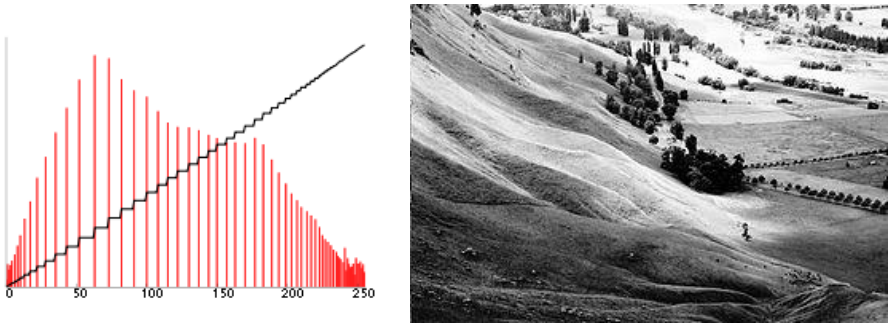

Figure 2:2 Image after histogram equalization

This method usually increases the contrast of images, particularly when the image is represented by a small range of intensity values. Through this adjustment, the intensities can be better distributed along the histogram graph utilizing the full range of intensities equally. This allows for areas of less contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the highly crowded intensity values. The method is useful in images with backgrounds and foregrounds that are both bright or both dark (scanned fingerprint images)

The first histogram equalization saw in the previous figures, considers the global contrast of the image. In many cases, it is not the best idea. For example, below image shows an image and its result after global histogram equalization.

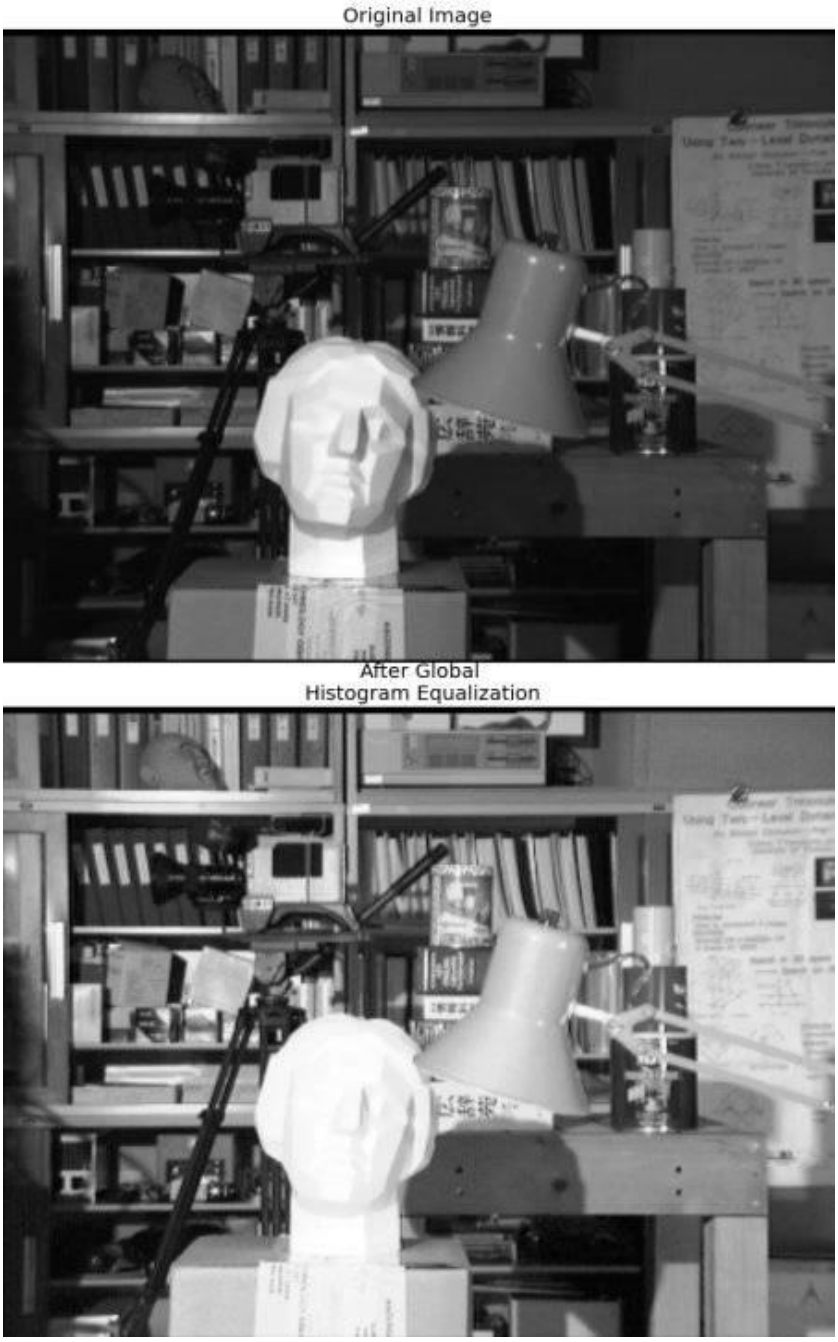

Figure 2:3 Image global Equalization of image

It is true that the contrast in the image has improved. But compare the face of statue in both images. We lost a lot of the information there because of over-brightness. It is because its histogram is not confined to a particular region as we came across in previous cases.

To solve this problem, adaptive histogram equalization is used. In this case, the image is divided into small blocks called "tiles" (tile Size is 8x8 by default in OpenCV). Then each of these tiles are histogram equalized normal. So in a small area, histogram would gathered in a small region. If there are noise, then it will be amplified. To avoid that, we apply contrast limiting. If any histogram bin is above the specified contrast limit (by default 40 in OpenCV), those pixels are clipped and distributed uniformly to different bins before applying histogram equalization. After equalization, we apply bilinear interpolation.

The result below compared with results above, especially the statue region:

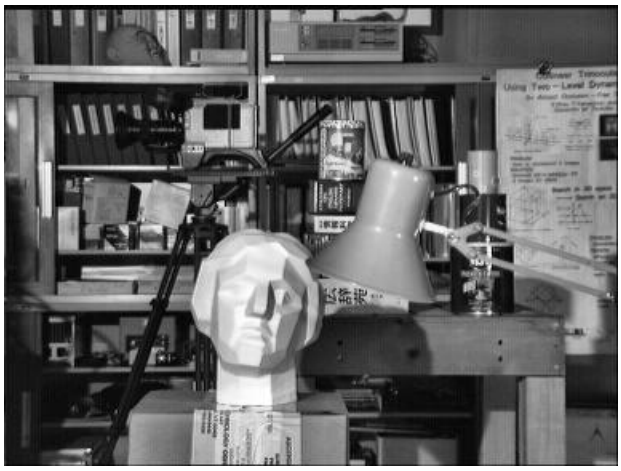

Figure 2:4 Global Equalization effect on image details

2.1.2 Binarization:

Image binarization is the process of acquiring a grayscale image and converting it to black and white only, basically reducing the information contained within the image from 256 shades of gray to binary values of 2: black and white. This is also known as image thresholding, although thresholding may produce images with more than 2 levels of gray. It is a form of *segmentation*, whereby an image is divided into objects. This is a task usually performed when trying to extract an object from an image. However, it is not trivial, and is completely dependent on the content within the image. The trick is images that may look easy to convert to binary are many times not.

Figure 2:5 comparison between binary, gray-scale, colored image

A greyscale pixel can have a color intensity between 0 - 255 representing 256 shades of gray from black to white.

2.1.3 Image enhancement:

To enhance the ridges in fingerprint images we use the oriented gabor filter.

Gabor filters consist of two components, sinusoidal and Gaussian. The Gabor filter method connects the optimal representation of orientation direction and the frequency domain.

The Gabor method was discovered by Gabor in 1946, where the function was defined in one dimension with t stating time and then developed into two dimension in the spatial domain formulated the following equation (1):

$$G(x, y;\ \theta, f) = exp\left\{-\frac{1}{2}\left[\frac{x_\theta'^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right\} \cos(2\pi f x_\theta)$$

where

$$x_\theta = x\cos\theta - y\,sin\theta$$

$$y_\theta = x\sin\theta + y\,cos\theta$$

θ is the orientation direction, f is the cosine wave frequency □□ and □□ is a fixed distance from the

Gaussian properties respectively along the x and y-axes.

2.1.4 Thinning:

Thinning is a morphology operation that is used to remove selected foreground pixels from a binary image, it can be used for several applications, but is particularly useful for skeletonization. In this mode it is usually used to tidy up the output of edge detectors by reducing all lines to single pixel thickness. Thinning is standardly only applied to binary images, and produces another binary image as the output.

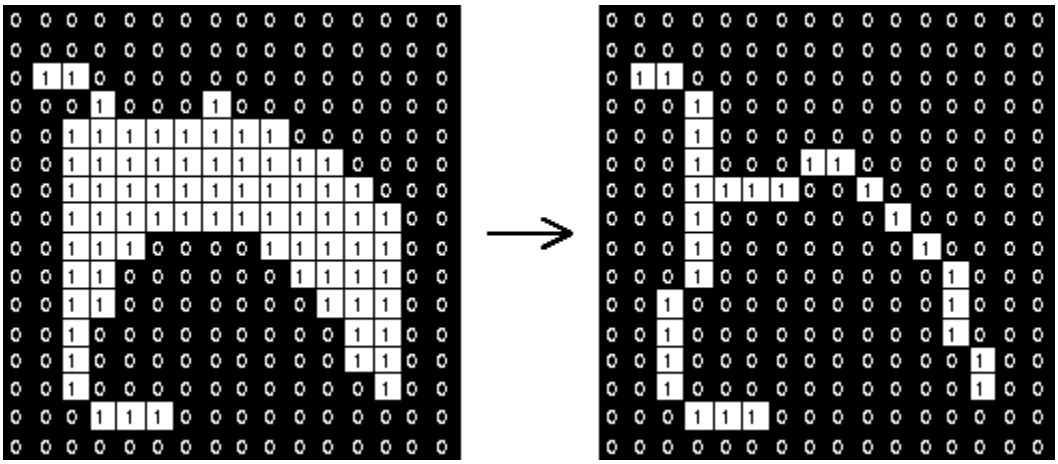

Figure 2:6 Thinning binary image

In fingerprint images it's visible how making a skeleton of the image helps at defining the lines of the fingerprint, crucial step before moving to feature extraction

2.1.5 Feature Extraction Techniques:

A feature extractor finds the ridge termination and ridge bifurcations from the input fingerprint images. If ridges can be perfectly spotted in an input fingerprint image, then minutiae extraction is just a daunting task of extracting singular points in a thinned ridge map. However, it is impossible to always obtain a perfect ridge map. The performance of current minutiae extraction algorithms mainly depends on the quality of input fingerprint images.

Major feature extraction methods:

1. Crossing number minutiae detection
2. Corner detection

2.1.6 Crossing number minutiae extraction:

The minutiae position and angles are derived after minutiae extraction. The terminations which lie at the outer boundaries are not considered as minutiae points, and Crossing Number method is used to locate the minutiae points in fingerprint image. The Crossing Number is defined as half of the sum of differences between intensity values in two neighbor pixels. If the crossing number is 1, 2 and 3 or greater than 3 then minutiae points are classified as Termination, Normal ridge and Bifurcation respectively, is shown in the figure

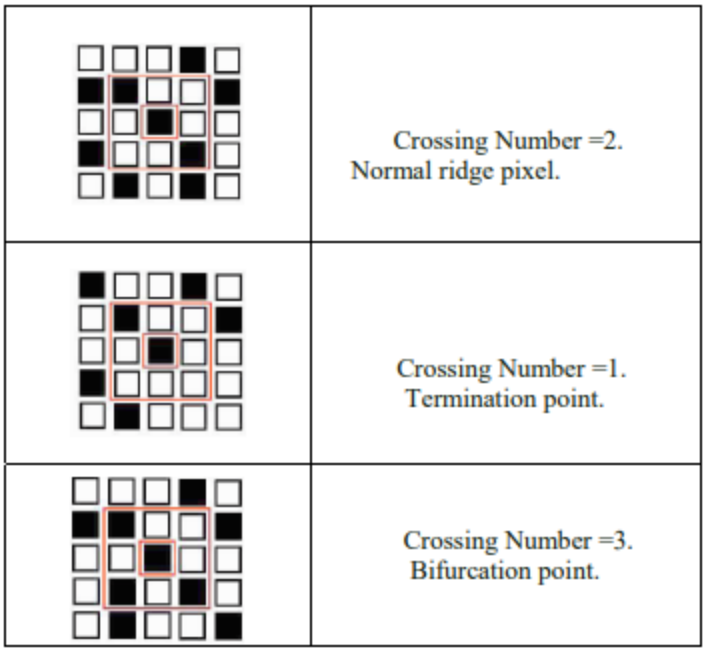

Figure 2:7 Cross Numbering ridge calculation

To calculate the bifurcation angle/rotation, we can use the advantage of the fact that termination and bifurcation are opposite in nature. The termination in an image corresponds to the bifurcation in its negative image thus by applying the same rules on the negative image, we will be able to get the bifurcation angles.

The following figure shows the original fingerprint (a) and the extracted minutiae points. (b) Shows the position of termination and diamond shape shows the position of bifurcation as in the figure.

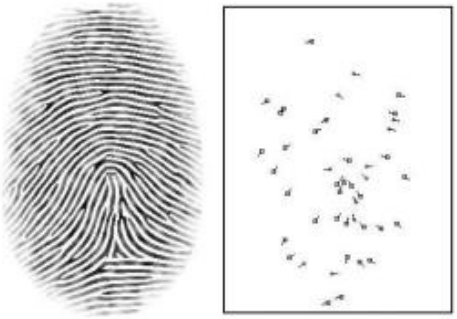

Figure 2:8 Minutiae extraction points

2.1.7 Corner detection:

Corners are regions in the image with large variation in intensity in all the directions. One early attempt to find these corners was done by Chris Harris & Mike Stephens in their paper A Combined Corner and Edge Sensor in 1988, so now it's called the Harris Corner Sensor. He took this simple idea to a fine form. It principally finds the difference in intensity for a relegation of (u, v) in all directions. This is expressed as below:

$$E(u,v)=\sum x,yw(x,y)[I(x+u,y+v)-I(x,y)]^2$$

The window function is either a rectangular window or a Gaussian window which gives weights to pixels underneath.

The window function is either a blockish window or a Gaussian window which gives weights to pixels underneath. We've to maximize this function E (u, v) for corner discovery. That means we've to maximize the alternate term. Applying Taylor Expansion to the below equation and using some fine way (please relate to any standard textbook books you like for full derivate), we get the final equation as:

$$E(u,v)\approx[uv]M[uv]$$

Where

$$M=\sum x,yw(x,y)[IxIxIxIyIxIyIyIy]$$

Then, Ix and Iy are image derivations in x and y directions independently. (These can be fluently set up using cv.Sobel()). Also comes the main part. After this, they

created a score, principally an equation, which determines if a window can contain a corner or not.

$R=det(M)-k(trace(M))2$

Where

- $det(M)=\lambda1\lambda2$
- $trace(M)=\lambda1+\lambda2$
- $\lambda1$ and $\lambda2$ are the eigenvalues of M

So the bulks of these eigenvalues decide whether a region is a corner, an edge, or flat.

- When |R| is small, which means $\lambda1$ and $\lambda2$ are small, the region is flat.
- When R<0, which means $\lambda1>>\lambda2$ or vice versa, the region is edge.
- When R is large, which means $\lambda1$ and $\lambda2$ are large and $\lambda1\sim\lambda2$, the region is a corner

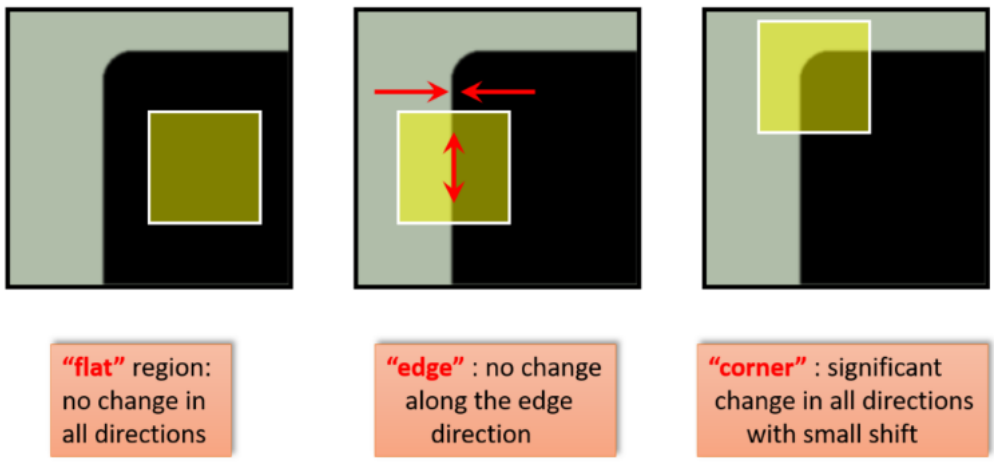

Figure 2:9 Harris corner edge detection

The algorithm extract terminations and bifurcations in a fingerprint image.

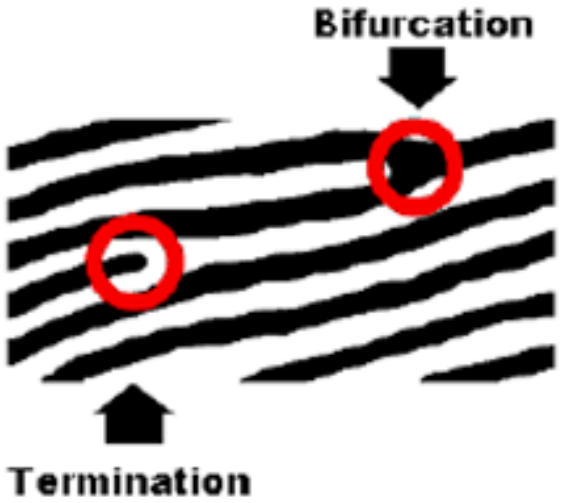

Figure 2:10 Bifurcation, Termination features in fingerprint

2.2 Matching techniques:

Given two (test and reference) representations, the matching module determines whether the fingerprints are of the same finger. The matching stage typically defines a metric of the similarity between two fingerprint representations. The matching phase also defines a certain threshold to decide whether a given pair of representations are of the same finger or not.

We will discuss how we matched the features we extracted in the Feature detection part above. We selected four algorithms that had the highest potential to fill the needs of accurate and fast biometric fingerprint feature matching.

2.2.1 Brute force matching

Brute-Force matching takes the extracted features (descriptors) of one image, matches it with all extracted features of another images in the database, and returns the similar one. This matching algorithm is so slow and has linear time complexity. This results in higher computational costs. But definitively much higher accuracy and precision than other methods.

2.2.2 KD-Tree

FLANN (Fast Library for Approximate Nearest Neighbors) is an image matching algorithm for fast approximated nearest neighbor searches in high dimensional spaces. These functions project the high dimensional features to a low dimensional space and generate compact binary codes. Making use of the produced binary codes, it can carry out fast image search via binary pattern matching or Hamming distance measurement, which greatly reduces the computational cost and further optimizes the efficiency of the search.

To avoid the brute force approach of comparing all points we can think about building an index, called spatial indexes, KD-Trees are one of the many forms in the family of spatial indexes.

The drawback of KD-Trees and all spatial indexes overall is that when the number of dimensions of the space gets larger the index tends to perform similarly to the brute force search.

2.2.3 LSH (Locality Sensitive Hashing)

LSH takes a completely different approach, the most important thing is that LSH will not give you exactly your absolute k nearest neighbors but will give you an approximate, maybe there's a very close point that is ignored by LSH.

LSH works by hashing the query point in an attempt to make points that are close to each other collide. Then if you have a query point, you just hash it and that gives you a group number, you go to the group and the points inside that bucket are the candidates to be the nearest neighbors, a brute force search of only those candidate points and will find an answer. If false-negatives are a big issue you can do this "n" times having "n" hash tables so you go to "n" buckets to pick-up candidates instead of just one group, this makes you check more data points which is slower but reduces the chances of missing a near neighbor.

2.2.4 HNSW (Hierarchical Navigable Small Worlds)

HNSW method works by building a hierarchical graph incrementally, an approximate K-nearest neighbor descriptor based on small-world graphs with hierarchy. This algorithm is fully graph based, without any more search structures that have high recall. Each node in the graph is linked to the other node in the closest space. The hierarchical structure in HNSW is beneficial to skip far away from neighbors on the low dimension cases. The speed efficiency over the approaches without hierarchy support is around the doublet. The drawback of HNSW is the problem of dimensionality that cannot address this issue.

Chapter Three
Related Works

3. Similar Systems:

In this chapter we discuss different e-voting systems applied in different countries in section 3.1, 3.2, and 3.3.

3.1 (EVM) Fingerprint Based Voting System:

This system describes e-voting system used in Belgaum, India in 2015.Mainly intended to develop a fingerprint based advanced Electronic Voting Machine (EVM) by using:

1) Software
- Keil TOOLS by arm version 4
- Proteus

2) Hardware
- Fingerprint module
- ARM processor
- LCD display

Which helps in free and fair way of conducting elections which are basis for democratic country like India.

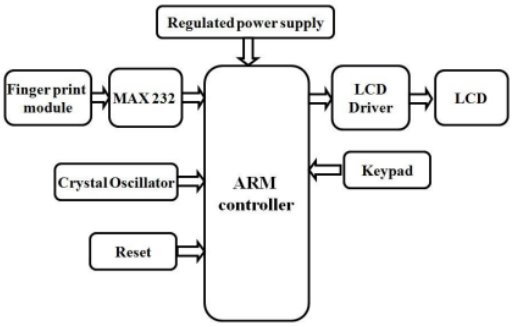

Figure 3:2 ARM system block diagram

Figure 3:1 photo of the system in real life

Advantages:

- Cost effective
- Low power consumption

Disadvantages:

- Before voting the user has to enroll first.
- Sensitivity of finger print module causes sometimes Combine character error.

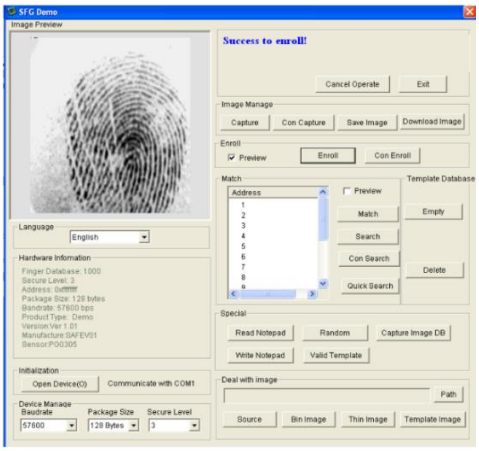

Figure 3:3 System software design

3.2 “i-Voting” online voting system:

i-Voting is an online voting system developed by the Estonian government in 2009 and since then they updated it make it more secure and used in many elections from 2009 till 2019.

How does it work?

The voter can vote from anywhere across the country but he must have a gadget connected through USB port and internet connection to cast his vote. Voter confirm his identity by using his ID card which uses NFS technology to send the data to the gadget then identify the voter. The figure below describe the id card of the voter used with the gadget connected to a pc device

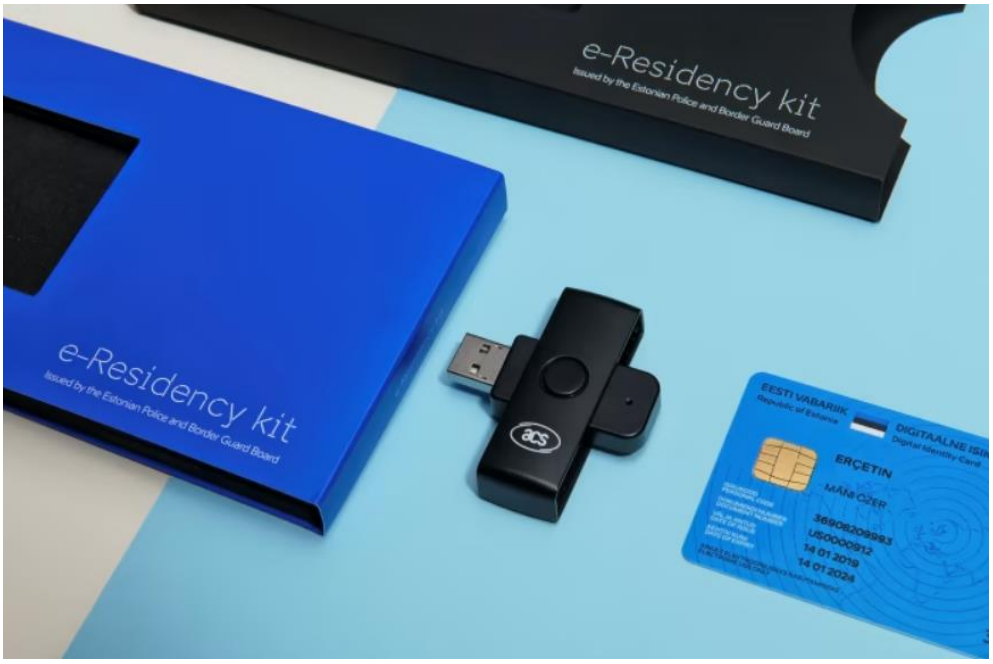

Figure 3:4 USB device for ID cards

.

Figure 3:5 USB device Connected to PC for voting system

Advantages:

1- Providing the preventive measures system for voting.

2- It results in polling time, provide easy and accurate counting without any mischief at the counting center.

3- Lower risk of human and mechanical errors.

Disadvantages:

Security problem as anyone can steal the ID card and vote instead of another person without which cause loss of authenticity in the result.

3.3 Permanent Voters Card (PVC) voting system:

This system applied in Nigeria since 2020, and it is developed by The Independent National Electoral Commission (INEC) which us established by the 1999 Constitution of the Federal Republic of Nigeria to organize elections in various political offices in the country.

This Voting system depends on (PVC) which is ID cards for citizens specially made for the voting process also the system uses (EVM) Electronic Voting Machines connected to a server directly to count the result and publish them automatically after the election is complete.

Advantages:

- The false voters can be easily identified.
- The voters can cast their voting from anywhere

Disadvantages:

- Too much hardware used
- Cost is high duo to hardware use and server maintenance

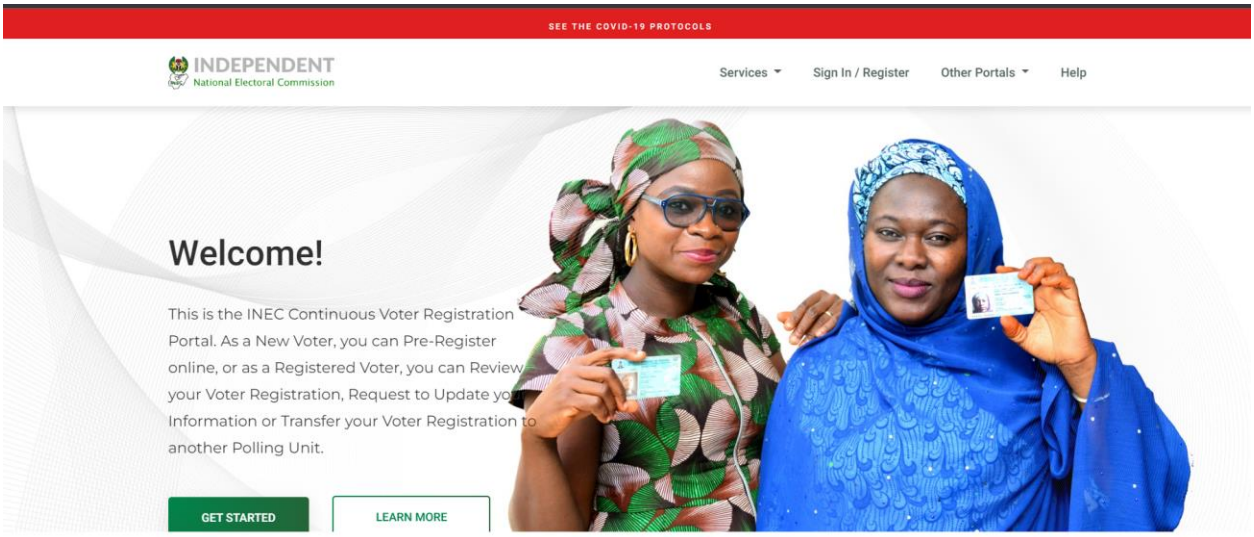

Figure 3:6 INEC Nigeria voting website system

Chapter 4
Proposed System

4. System Implementation:

In this chapter, the implementation of the proposed project system is carried out, and the preparation is explained. The implemented algorithms will be denoted and explained. After that the GUI of the system is introduced and explained with all its controls.

4.1 System Description:

The system described in this report is a Windows desktop application that allow a person in a room with a computer and a finger print scanner to cast a vote in real-time and authenticate his identity using his pre-registered fingerprint. The software also consists of a graphical user interface (GUI), where ID and fingerprint can be actuated in real time. Machine learning, Image processing and feature extraction are used for the Detection. Firstly, Dataset is inserted to the system. Then, preprocessing is applied on the fingerprint image stream Edge Detection, Resizing, Converting to Grey Scale, etc. Next, extracting features from the images such as extract minutiae of fingerprint (bifurcation, termination). Depending on what the algorithms use such as HNSW, Brute Force, LSF, and KD-Tree, etc. After that, setting the parameters threshold rate, etc. Then, saving the weights of features extracted so it can be used by the model again. Last but not least, validating the model to make sure that no over fitting or high loss function detected in the training session. Finally, after the validation is approved by the results, the model will be ready to run some test on images and detect if person is authorized or not, there is a valid matching with the fingerprint in the dataset or not, shown in the figure below.

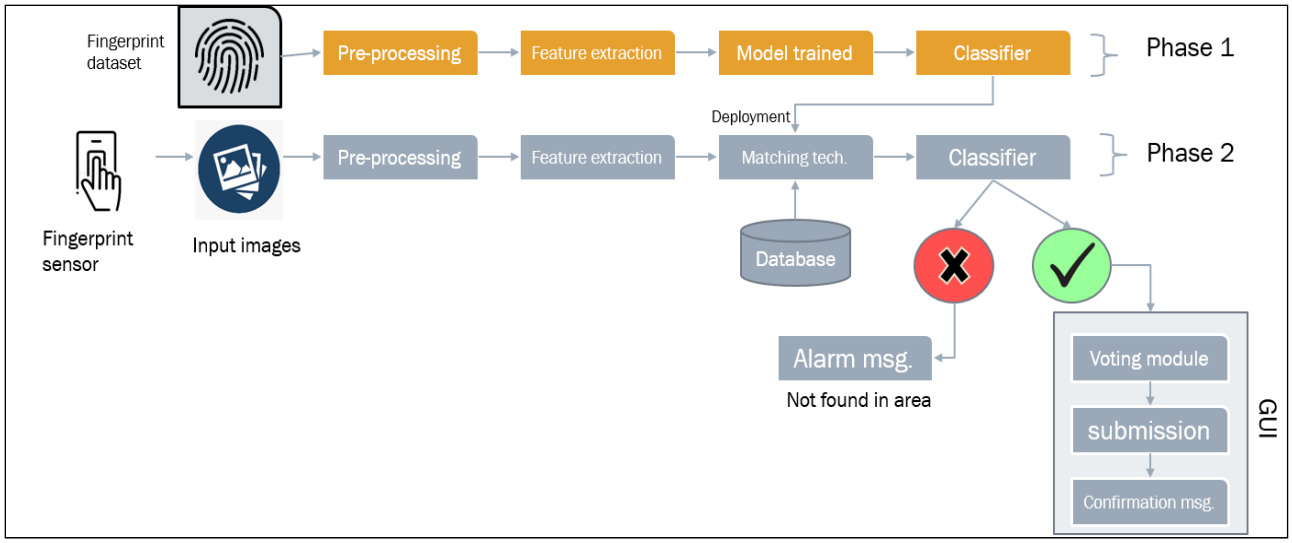

Figure 4:1 Proposed System Block Diagram

4.2 Sequence Diagram:

Sequence diagram is a depiction of the interaction among objects during certain periods it shows the interactions of a specific use case because the pattern of interactions varies from one use case to another. It shows the participating objects by their lifelines, and the interactions among those objects (arranged in time sequence) by messages they exchange with one another. The sequence diagram will be discussed according to the use cases illustrated in the previous section.

4.2.1 Sequence Diagram for Admin Login:

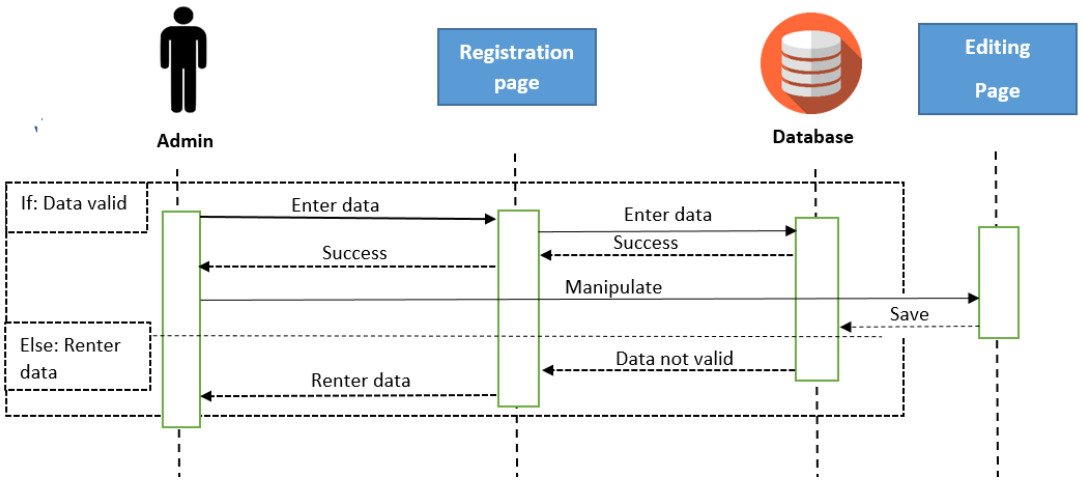

Figure 4:2 Admin Login Sequence Diagram

4.2.2 Sequence Diagram for Voting Process:

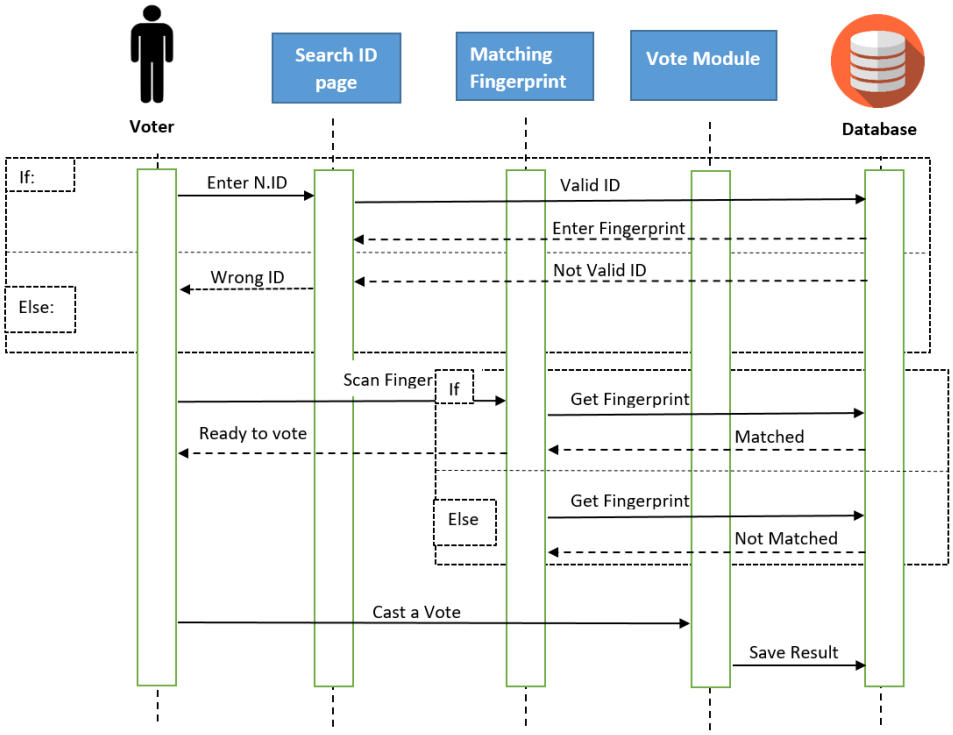

Figure 4:3 Sequence Diagram of Voting process

4.3 Platforms used:

The platforms we used during the implementation is PYTHON 3.7 (ANACONDA platform), Visual Studio IDE, Visual Studio Code, MSSQL server management studio, MongoDB , and Angular now we will introduce each of them with their most important features and uses.

4.3.1 Python:

Python is an interpreter, high level & general purpose programming language it is somehow a brand new platform as it was first released in 1991. Python has gained in popularity because of its clear syntax and readability also it is said to be relatively easy to learn and portable. Python considered is speed, efficient and reliable language, and it can work across many domains such as web development, mobile applications, hardware, so it will help if this application 110 converted to cross platform app in future work. It is easy to import, export and read data from wav files, also you can play it inside the platform itself. Python has plotting libraries which produces plots, histograms, power spectra, error charts, and more charts to trace changes in the data. These plots can be easily saved to hard disk. The powerfulness of python also comes from the powerful APIs and frame works that is linked to it, in our implementations we used some of those framework for the machine learning tasks.

4.3.2 Visual Studio:

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.

The integrated debugger works both as a source-level debugger and a machine-level debugger.

It accepts plug-ins that expand the functionality at almost every level—including adding support for source control systems(like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Azure DevOps client: Team Explorer).Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C,[9] C++, C++/CLI, Visual Basic .

NET, C#, F#,[10] JavaScript, Typescript, XML, XSLT, HTML, and CSS. Support for other languages such as Python,[11] Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

4.3.3 SQL Server Management Studio (SSMS):

SQL Server Management Studio (SSMS) is a software application first launched with Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server. It is the successor to the Enterprise Manager in SQL 2000 or before. The tool includes both script editors and graphical tools which work with objects and features of the server. SSMS is one of the SQL Server management tools, regardless of your location, used for designing queries and managing databases and data warehouses via personal computer or Cloud.

4.4 Image Processing techniques used in the system:

Fingerprint images must pass through preprocessing phases to extract the features which will be our descriptors that identify the person from another.

4.4.1 Histogram equalization

This method usually increases the global contrast of many images, especially when the image is represented by a narrow range of intensity values. Through this adjustment, the intensities can be better distributed on the histogram utilizing the full range of intensities evenly.

Example of Histogram Equalization on a scanned fingerprint

Figure 4:4 Histogram equalization on fingerprint image sample

4.4.2 Binarization:

Image binarization can be implemented by normalizing the values above and under 127 to either 0 or 1 for each pixel in the image.(the threshold of 127 may vary depending on the image at hand)

Example of image binarization on a greyscale scanned fingerprint:

Figure 4:5 Fingerprint Image Binarization

4.4.3 Oriented Gabor filter

Example of enhanced fingerprint image using oriented Gabor filter:

Figure 4:6 fingerprint after using Gabor filter and preprocessing

4.4.4 Thinning

Thinning is a morphological operation that is used to remove selected foreground pixels from binary images, it can be used for several applications, but is particularly useful for skeletonization.

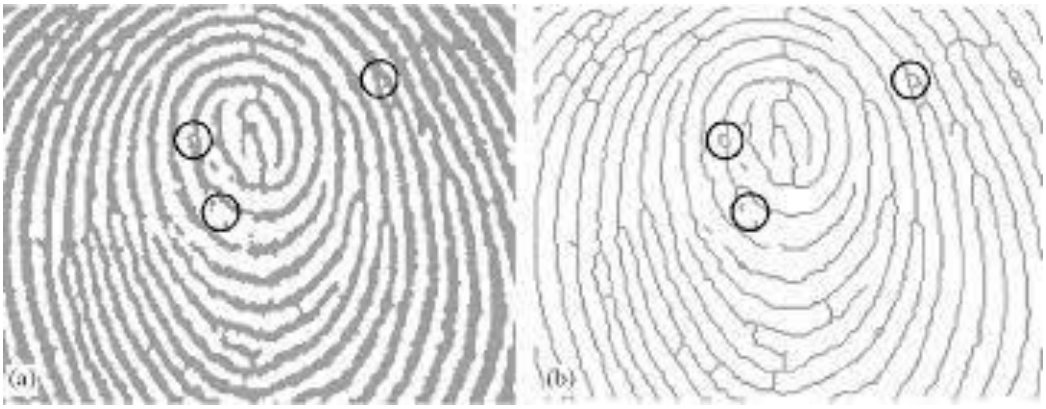

Figure 4:7 Fingerprint Thinning sample

4.4.5 Feature extraction:

Is a corner detection operator that is commonly used in computer vision algorithms to extract corners and features of an image.

Example of two fingerprints after detecting features:

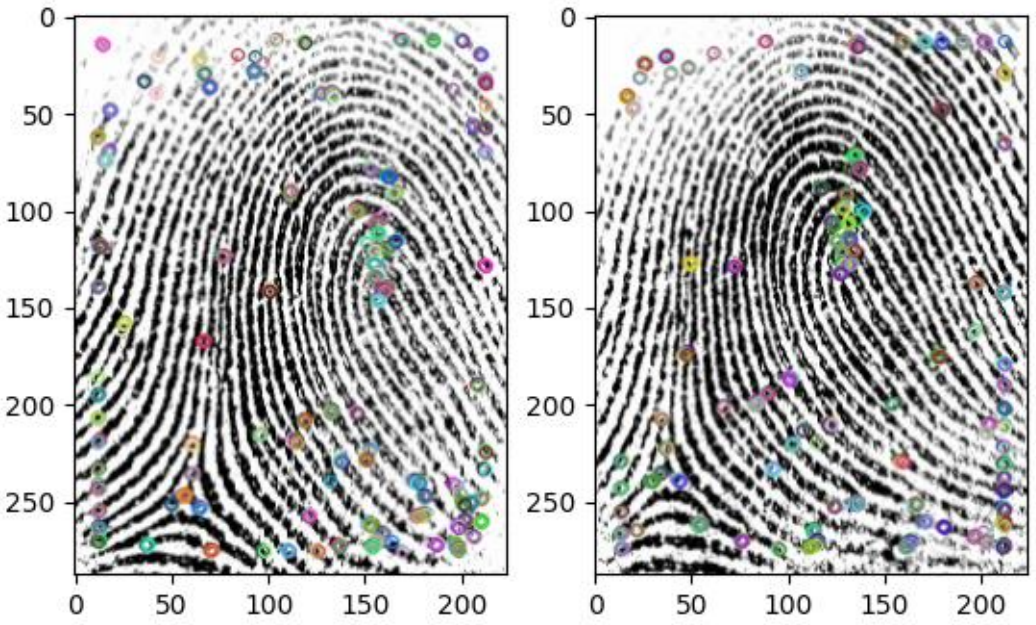

Figure 4:8 Fingerprint Feature Extraction and detection

4.4.6 Fingerprint matching:

Nearest Neighbor search is the problem of finding the point in a given set that is closest (or most similar) to a given point.

Example solution: KNN (K-Nearest Neighbor)

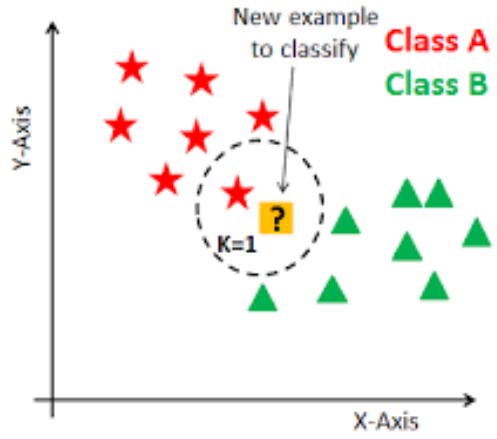

Figure 4:9 Example of 2 classes KNN algorithm

KNN steps:-

- Step 1: Obtain the new data point
- Step 2: Calculate the hamming distance between the given point and the rest of the data points
- Step 3: Take the nearest neighbor as per calculated distance
- Step 4: Assign the new data point to that category for which the nearest neighbor belongs to

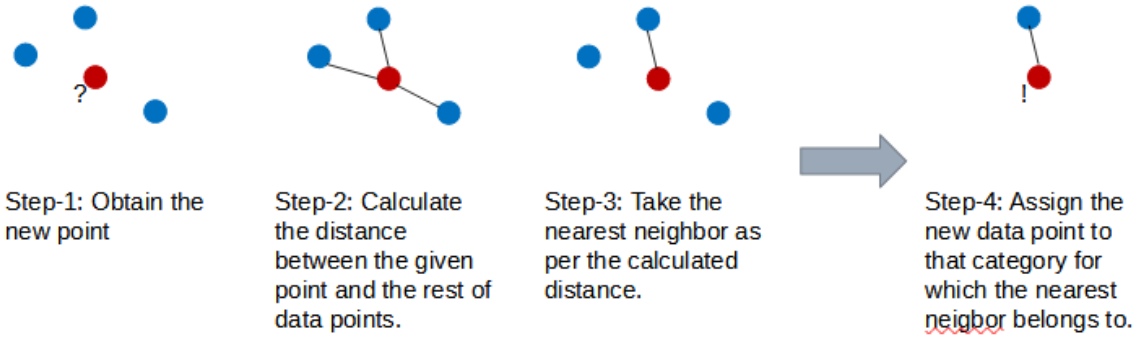

Figure 4:10 Explanation of KNN algorithm

Example of two fingerprints after matching the features using brute force KNN:

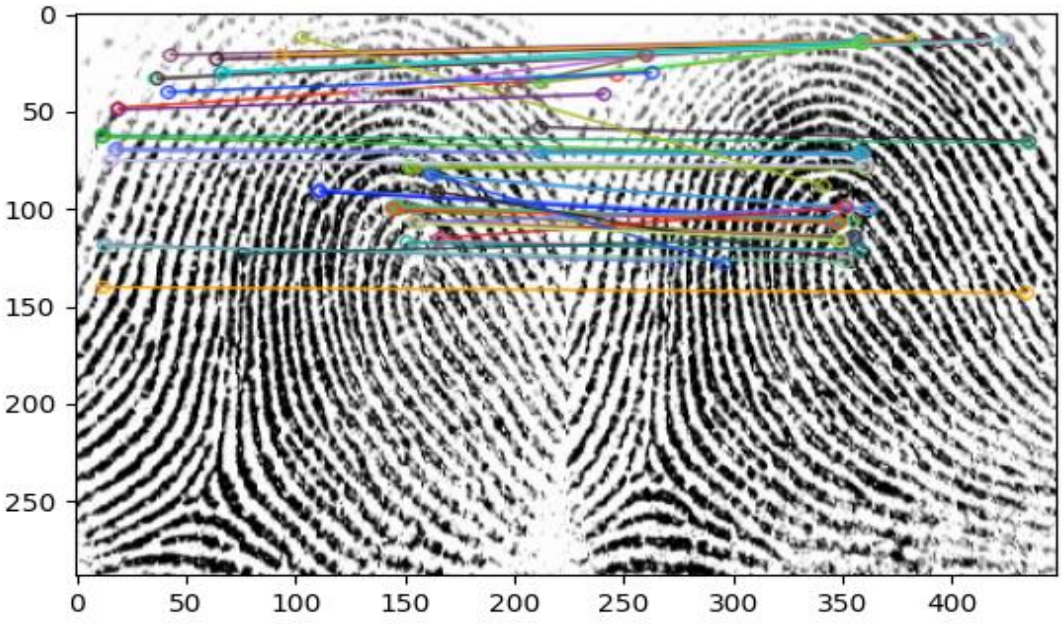

Figure 4:11 Matching Features on Fingerprint Image

4.5 Graphical User Interface (GUI):

The graphical user interface is presented (displayed) on the computer screen. It is the result of processed user input and usually the main interface for human machine interaction. The screen user interfaces popular on computer devices are an overlay of the visual output to the visual input. The 3 core activities in this process:

- Understand what the users will do with the system.
- Develop a series of prototypes for experiment, And Interface evaluation.
- Experiment these prototypes with users.

4.5.1 System GUI:

The system GUI was implemented with C#, SQL, and Python libraries based on the phases in the next sections.

4.5.2 Scenarios:

Scenario 1: Admin need to login to create and edit elections for the system.
Scenario 2: user enter his national ID to search for his profile.
Scenario 3: user authenticate by using his fingerprint if the profile is found.
Scenario 4: if the fingerprint matches then the voter go to voting module to choose his candidate.
Scenario 5: voter submit his vote and the system get ready for next voter.

4.5.3 Storyboard of GUI system:

Is used in software development as part of identification to the specifications for the software developed. Throughout the specification phase, for illustrating the important steps of the user experience. The reason why storyboarding is useful is that it helps the user understand exactly how the system will work, much better than description of an abstract. Its cost is low so it is easier to make changes to a storyboard than an implemented piece of software which will be expensive to redo with all the changes that need to be done in the system.

4.6 System Use Case:

Use case diagrams model behavior within a system and helps the developers understand of what the user requires. The stick man represents what's called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can't do. Use case diagram consists of use cases and actors and shows the interaction between the use case and actors. The purpose is to show the interactions between the use case and actor. To represent the system requirements from user's perspective. An actor could be the end-user of the system or an external as shown in the diagram below.

UML Use case diagram parts	Description
Actor	An actor is a role that a user plays in the system. It is important to distinguish between a user and an actor (better thought of as a role)
Associations	associations are drawn between actors and use cases to show that an actor carries out a use case
Includes	The includes link is to avoid repetition of scenarios in multiple use cases.
Extends	In some instances, you want to describe a variety of behavior in a more controlled form. In such instances, you can define extension points; in the extended use case

Table 1 Use case description

Use case of the whole E-voting system:

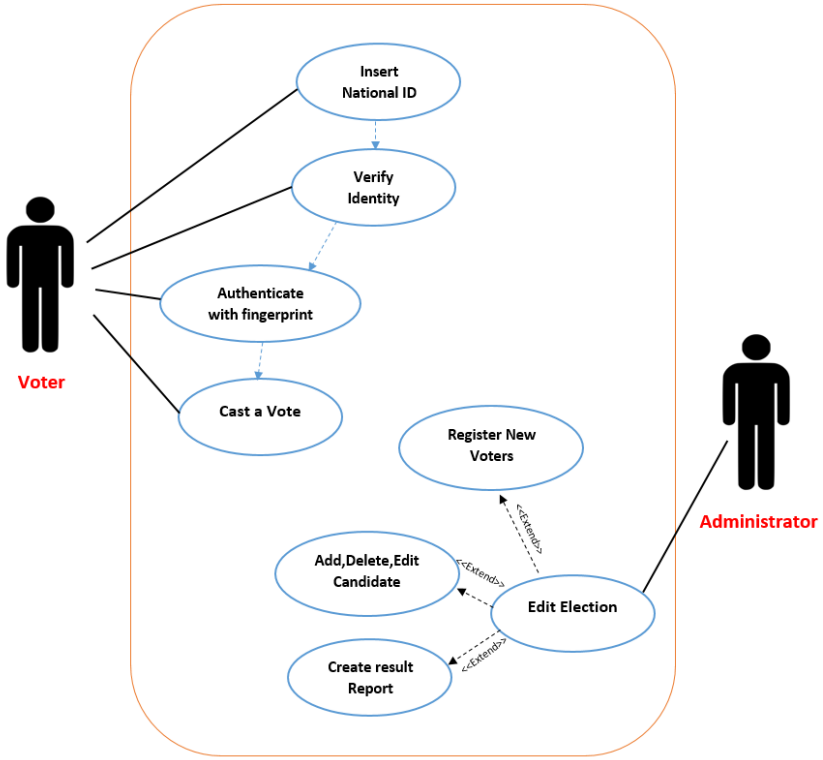

Figure 4:12 Use Case of Proposed System

4.7 System Prototype (Windows Application Version 1.0):

Version 1.0 of the proposed system implemented by windows form application and Database Management System (DBMS) describe the first version of the proposed system.

4.7.1 Home window:

The sketch of the home window of the GUI of the system as shown Figure 4.9. containing a logo with the application name (E-Voting System) and push button "Vote Now" that move the user to the window to search for his profile and push button "Administrator" that move you to login panel to enter admin mode using verified account, The system is divided into two section administrator side and voters side.

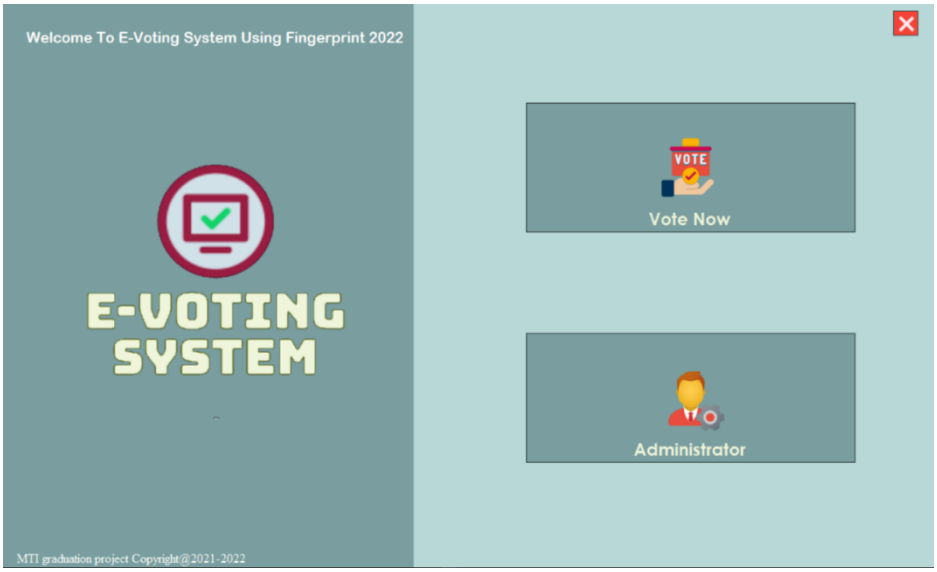

Figure 4:13 Home Window desktop app v1.0

4.7.2 Voters Side:

Search Screen : A sketch of the Vote Now screen which contain a tutorial image describe the number which voter should enter and a text box waiting for user to enter his national ID then a push button to start searching for his profile in the database.

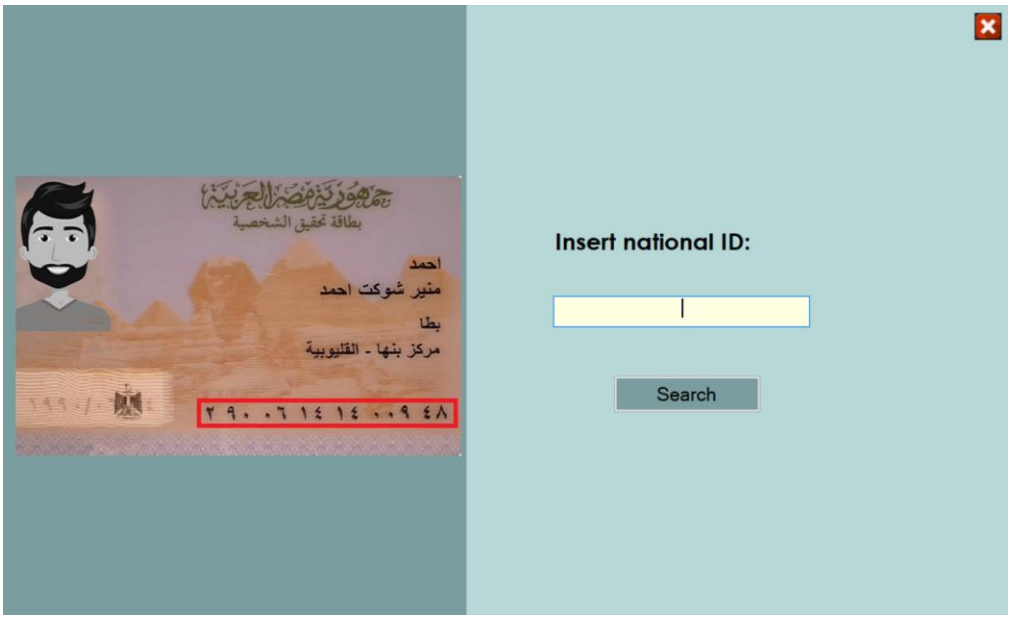

Figure 4:14 Search window desktop app v1.0

Profile Screen:

A sketch of found profile in case the person is registered and his national id matches with one in the database.

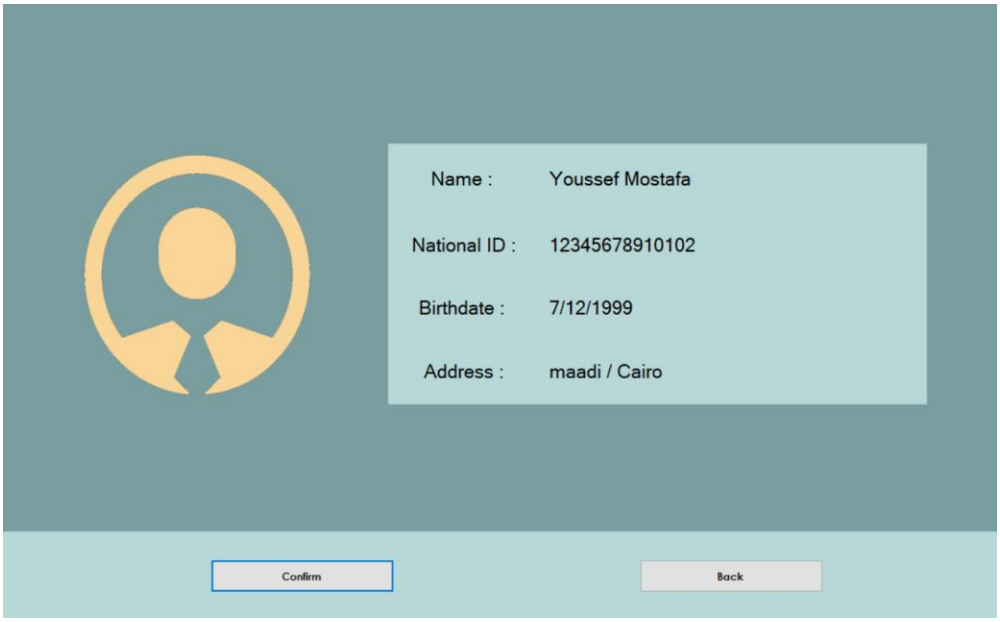

Figure 4:15 Found profile scree desktop app v1.0

Authentication Screen:

After the profile is found then the user must put his finger on the scanner to authenticate his identity using biometric, the matching algorithm (described above in details) start to match the live fingerprint input by the one registered in the database before the election.

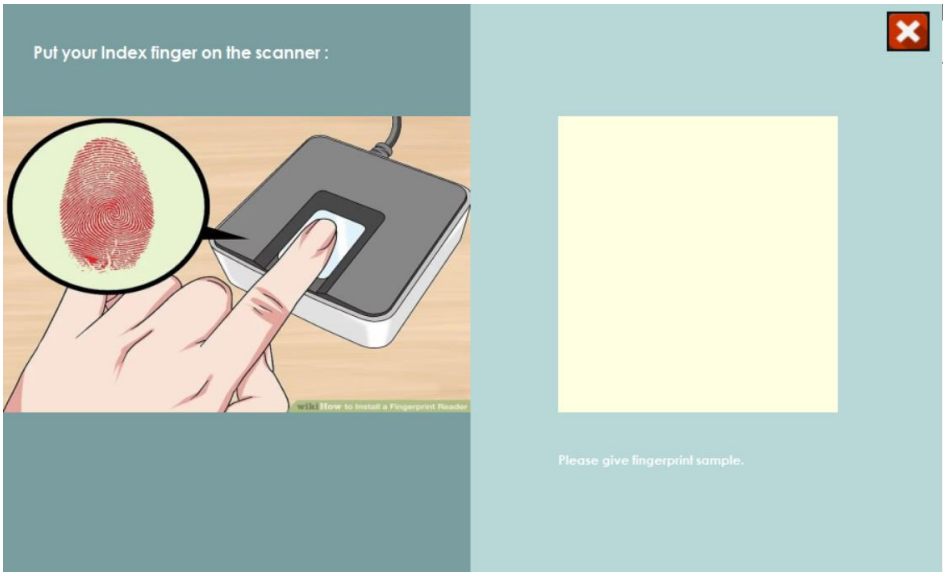

Figure 4:16 Authentication screen desktop app v1.0

Voting screen:

If the fingerprint matches then voter start voting process in the voting screen which include panel for every candidate having his name, nickname, photo, and symbol, Then press the button to choose the candidate.

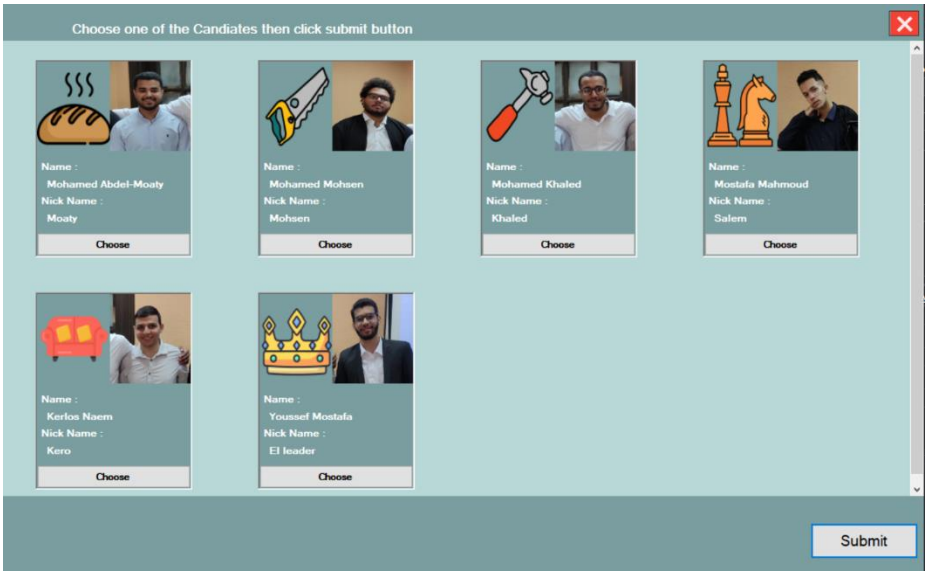

Figure 4:17 voting screen desktop app v1.0

After submitting the vote a message box appear to confirm a successful operation and increase the candidate vote counts in the database.

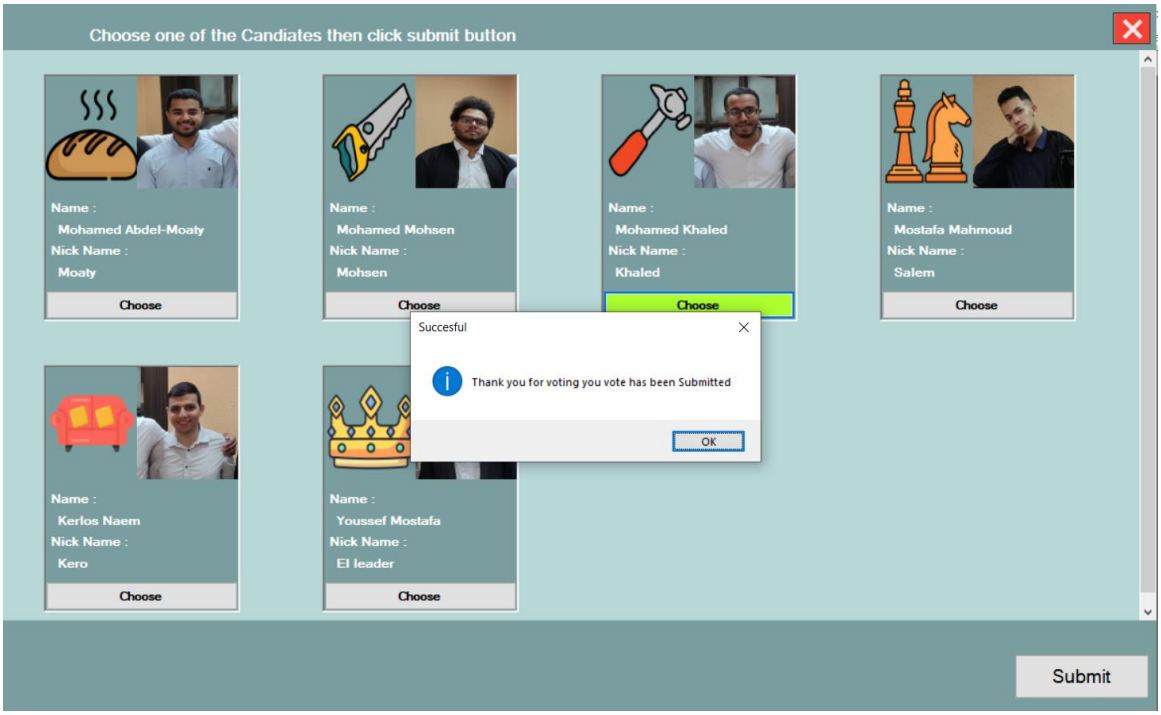

Figure 4:18 Vote Submitted desktop app v1.0

Then voting screen close and open a new search by id screen and get ready for next voter.

4.7.3 Login Admin Side:

A sketch screen include email text box and password textbox to enter admin dash board only using verified accounts.

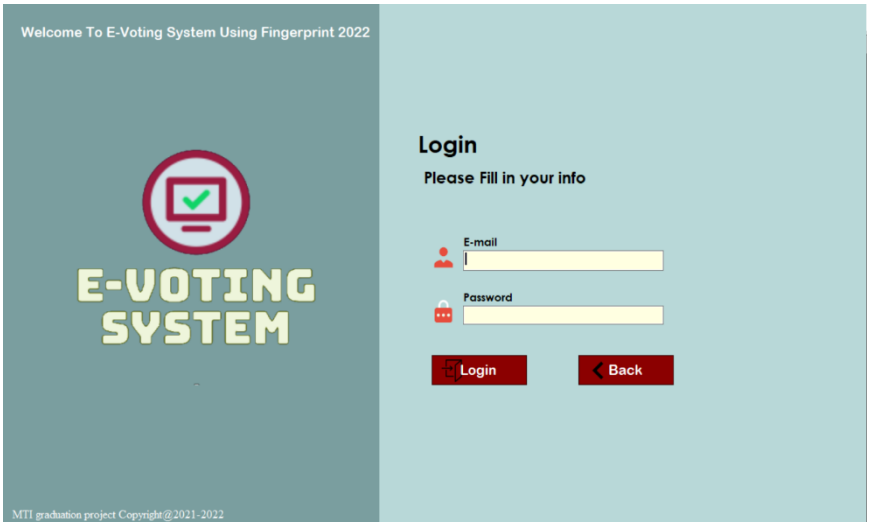

Figure 4:19 Admin Login desktop app v1.0

Dashboard Screen:

The screen include full control for authorized people to control the election process starting from editing data for voters and candidates also reports and data about number of votes and number of candidates and people who voted.

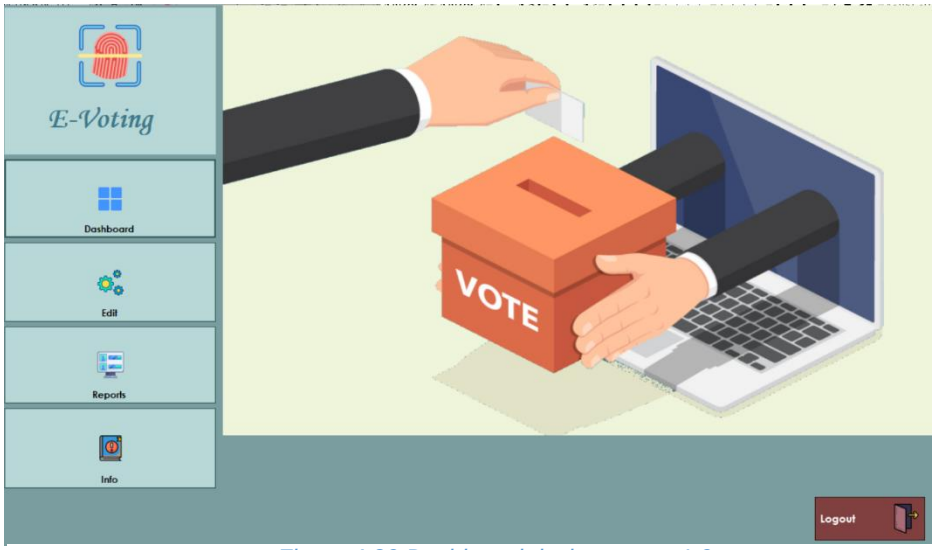

Figure 4:20 Dashboard desktop app v1.0

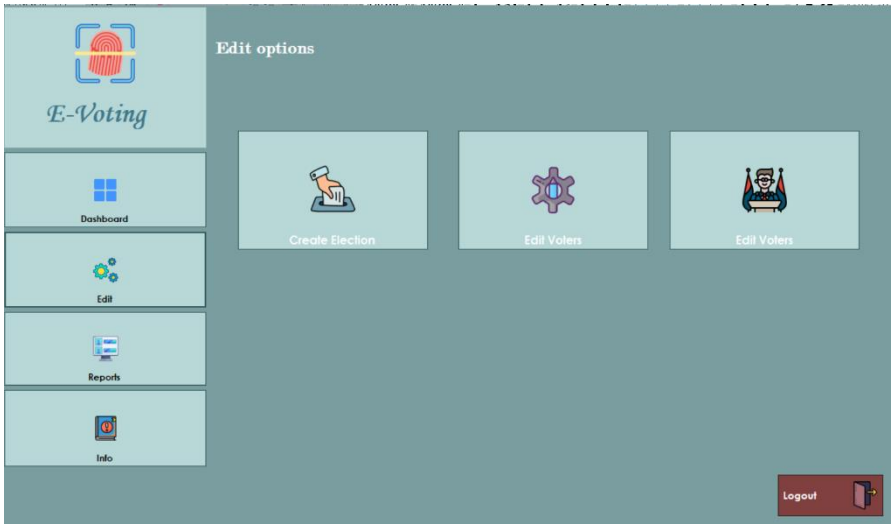

Figure 4:21 Edit screen desktop app v1.0

Edit Voter Form:

The screen include full control for authorized people to control the Editing data for voters and candidates also candidates and people who voted.

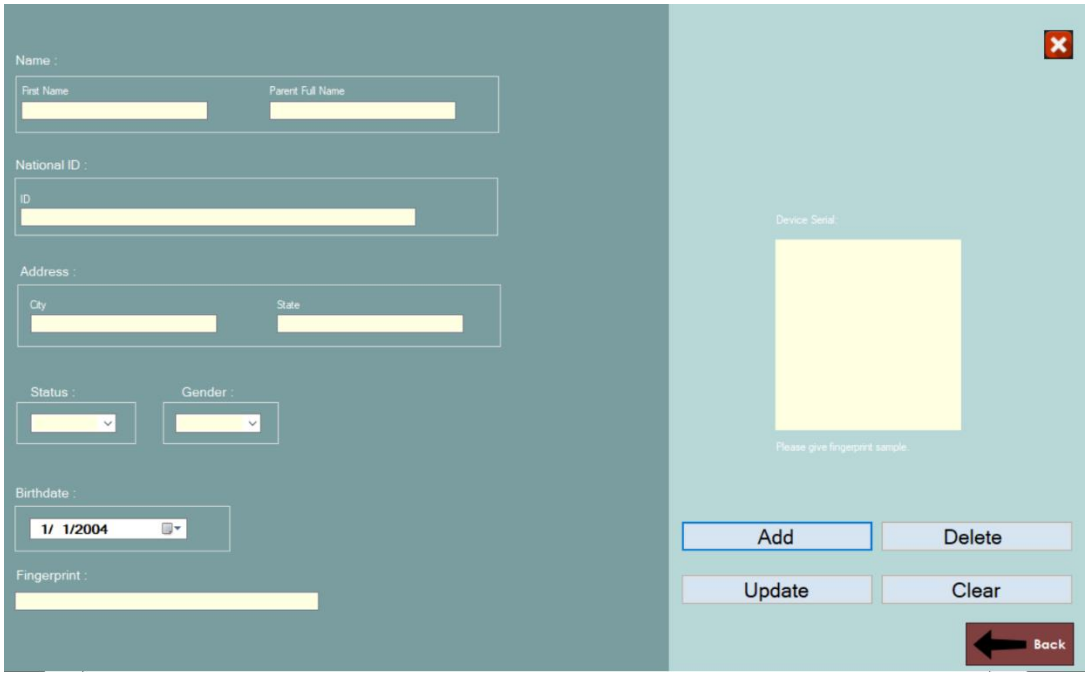

Figure 4:22 Edit Voter Form desktop app v1.0

4.8 System Web Application (Version 2.0):

Web application is a version two of the system which fix sum of the disadvantages in the desktop version.

4.8.1 System Home Screen

The screen describe what the voter will see at the first look, a simple design with Vote now button to redirect the voter to cast his vote.

Figure 4:23 Home screen web app v2.0

4.8.2 Voter Side Search screen:

The voter start to enter his national Id which is located on his ID card to search for his profile.

Figure 4:24 Search by ID web screen

4.8.3 Profile Found Screen:

If the voter is registered and his ID matches with the one in the database then the voter confirm and move to the authentication phase.

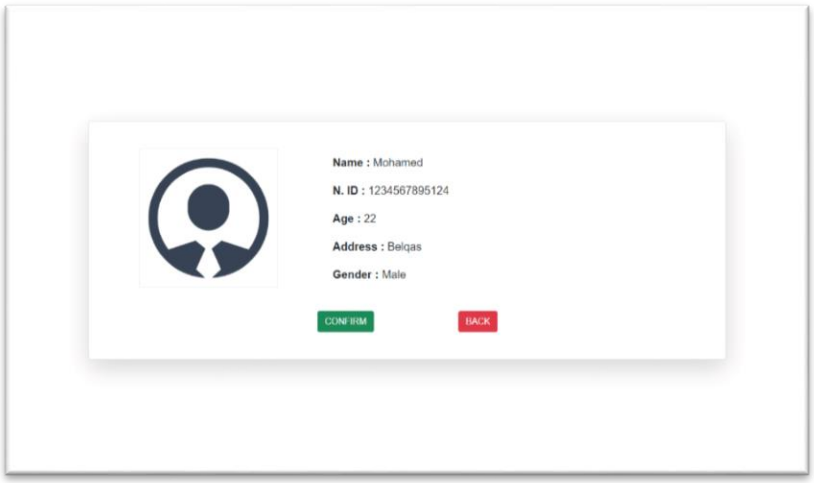

Figure 4:25 Found profile web screen

4.8.4 Authentication Using Fingerprint:

The user authenticate his identity by scanning his fingerprint to ensure that the same person who is using the ID card is the same person who is voting. The system take the fingerprint image and apply the matching module which include the image preprocessing and then match it with the one saved in the database before during the registration phase.

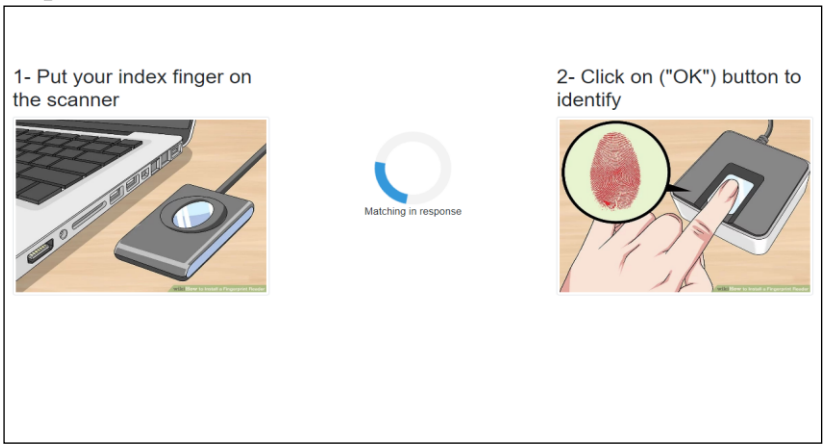

Figure 4:26 matching module for authentication screen

4.8.5 Voting Screen:

After the system recognize the person now it's time to choose his candidate and submit his vote.

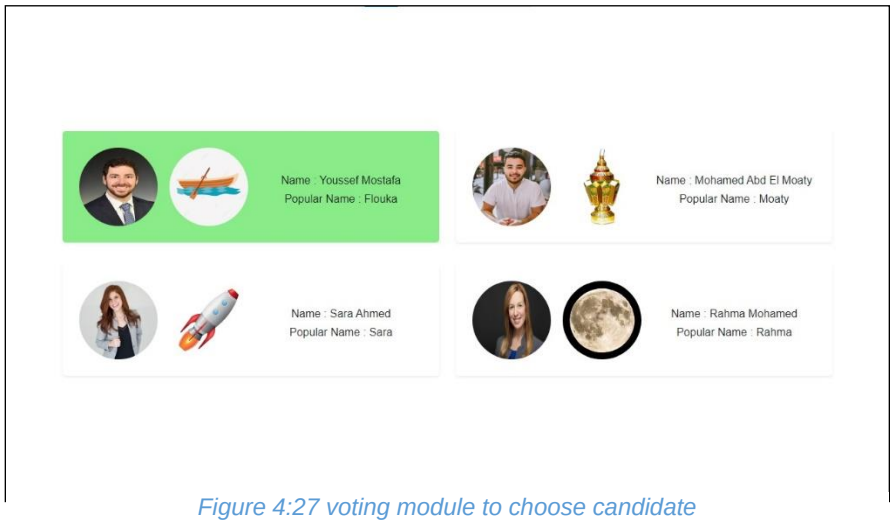

Figure 4:27 voting module to choose candidate

4.8.6 Vote Submitted Successfully:

The vote is submitted successfully saved and added to the candidate votes number in the database which will affect the result in the end.

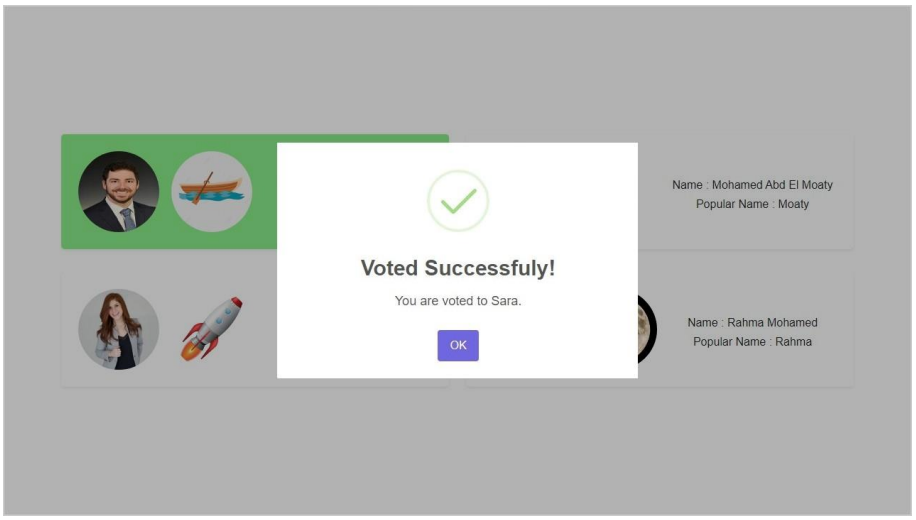

Figure 4:28 Successful vote

4.8.7 Administrator Login Screen:

The admin side of the system gives the administrator the control of adding, deleting, and editing the data of the voters, also the admin can control an election and see the results of the election process with no access in editing anything in the results.

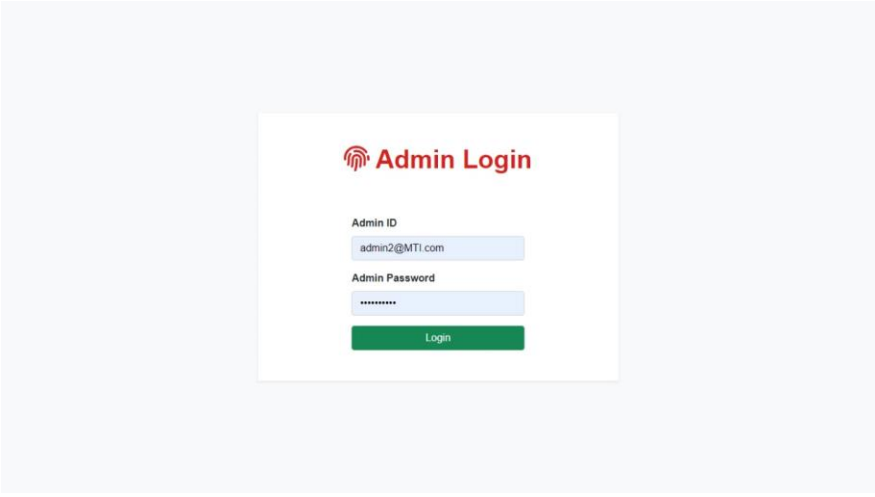

Figure 4:29 Login Admin Screen

4.8.8 Admin Dashboard screen:

A Dashboard is the control panel for the admin which can access only using verified email and password so he can control and edit the data.

4.8.9 Overview Section:

A screen include number of voters and number of candidates and election date and time.

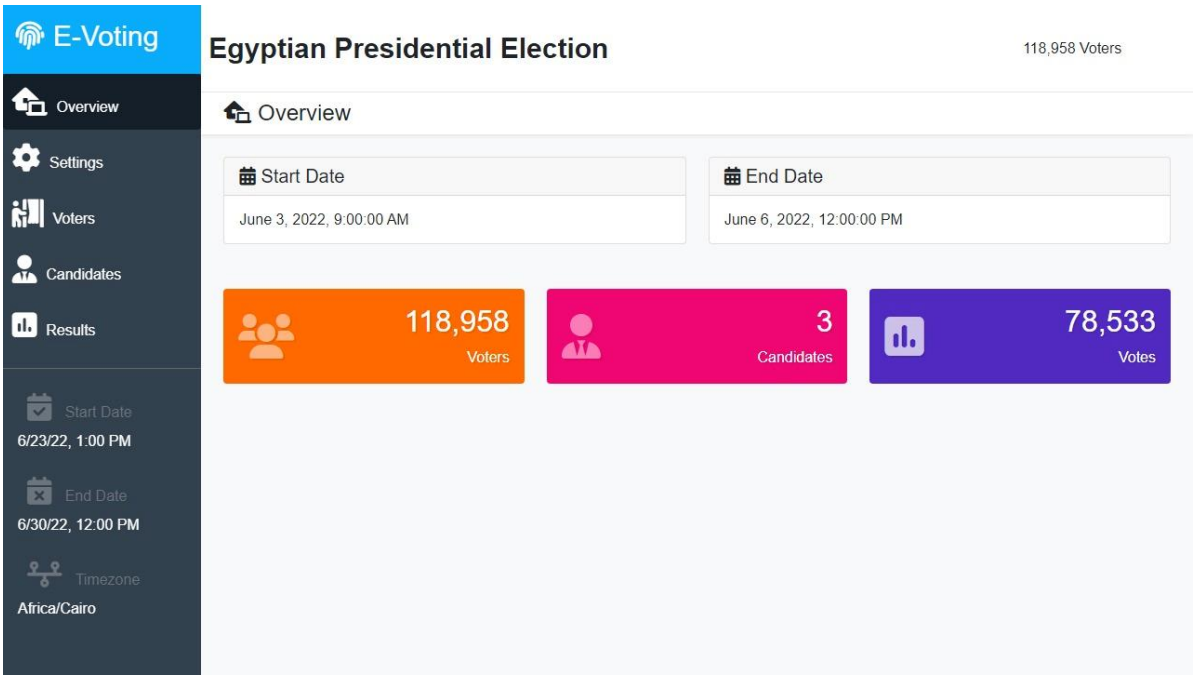

Figure 4:30 Overview Dashboard

4.8.10 Settings Section:

This section which the admin can control election name, date, and delete election.

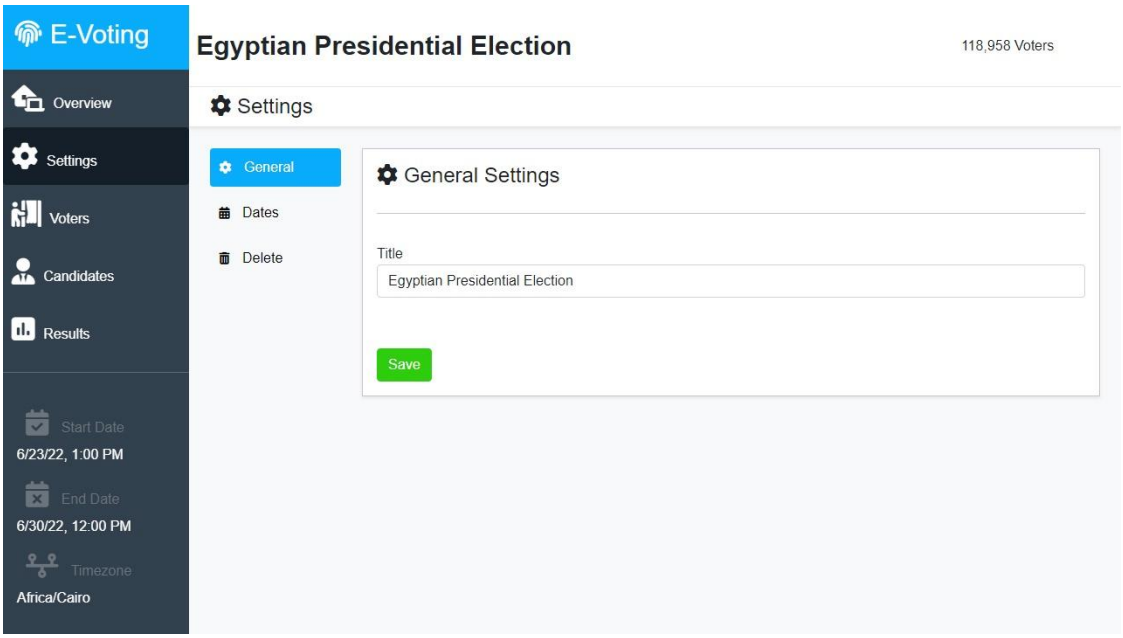

Figure 4:31 General setting in the dashboard

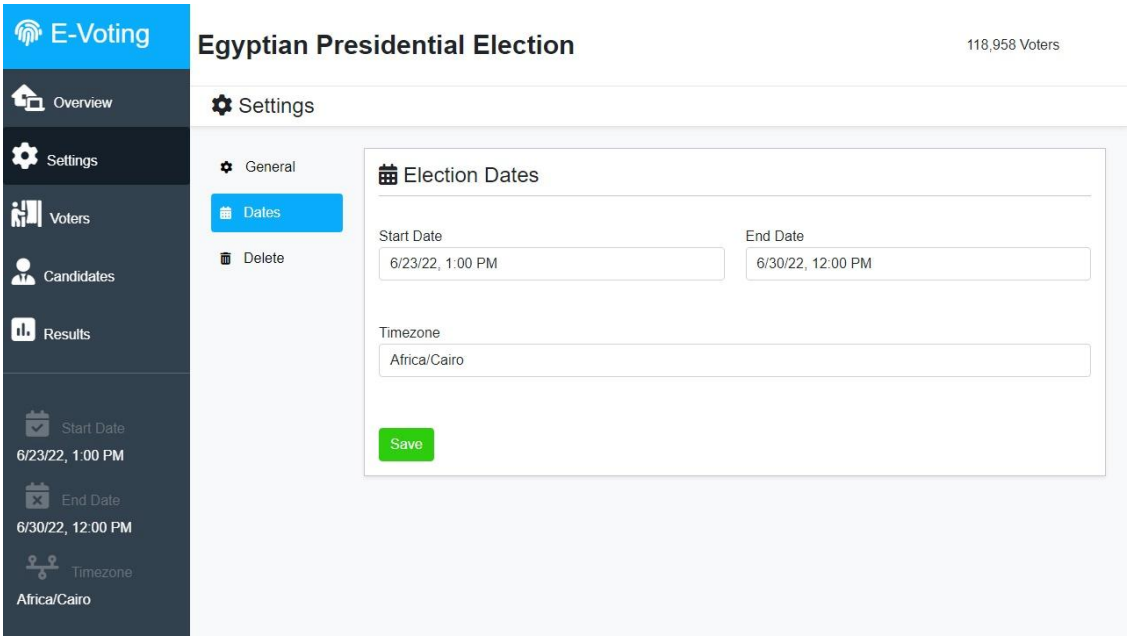

Figure 4:32 Election start, end date setting

The admin can delete the election by using the delete election button in the dash board.

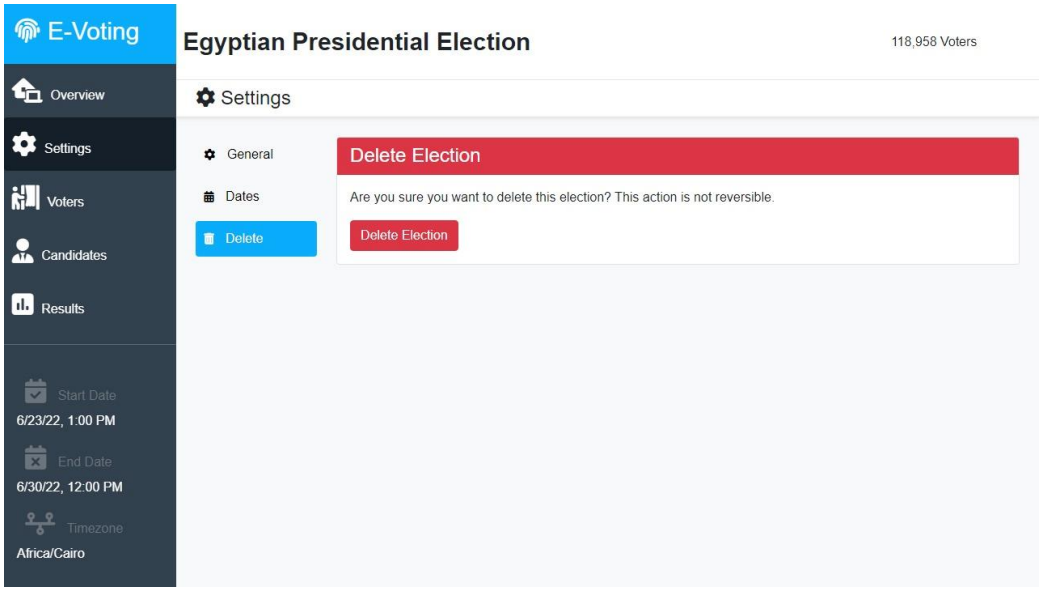

Figure 4:33 Delete election setting

4.8.11 Voter Section:

The admin can add or edit voters data from this section also voters data can be uploaded as a excel sheet with certain constraints on the file will be uploaded.

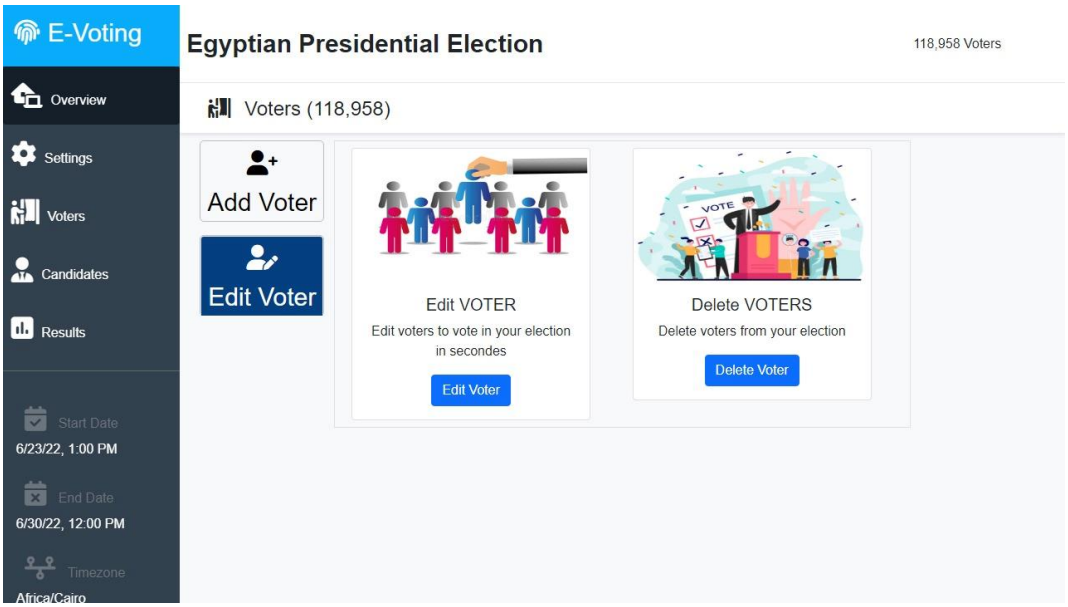

Figure 4:34 Voter data section

4.8.12 Adding, Editing Voters Data:

Editing the data which is the voter name, national ID, status, gender, address, and Fingerprint.

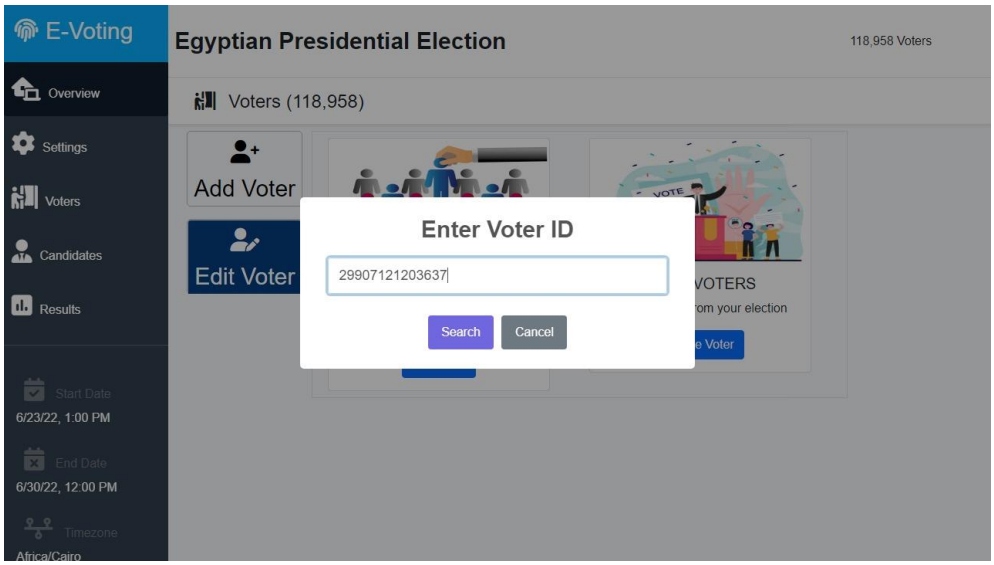

Figure 4:35 Search voter by ID to edit his data

Admin change the data of the voter then click on the Edit Voter to update this voter.

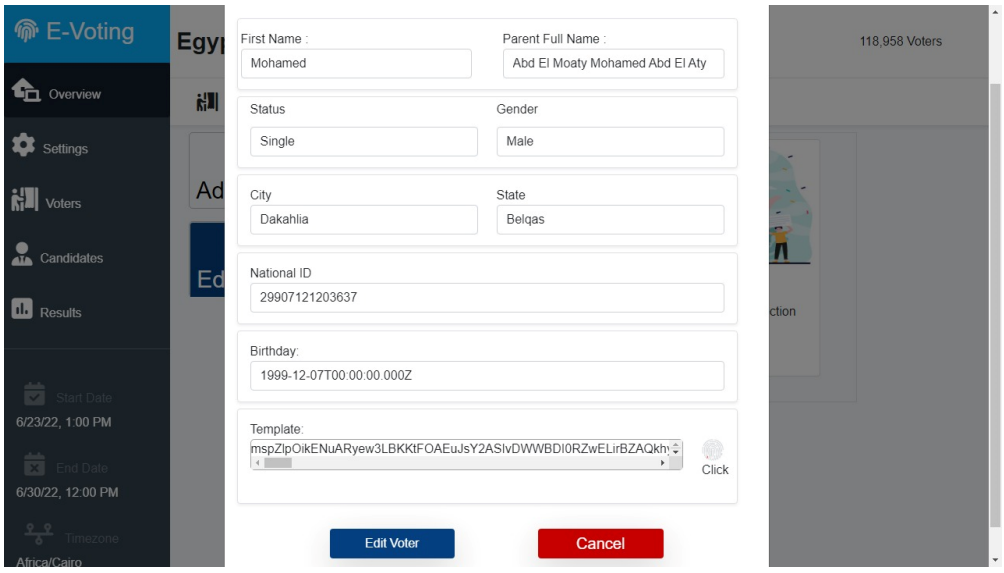

Figure 4:36 Editing Voter data

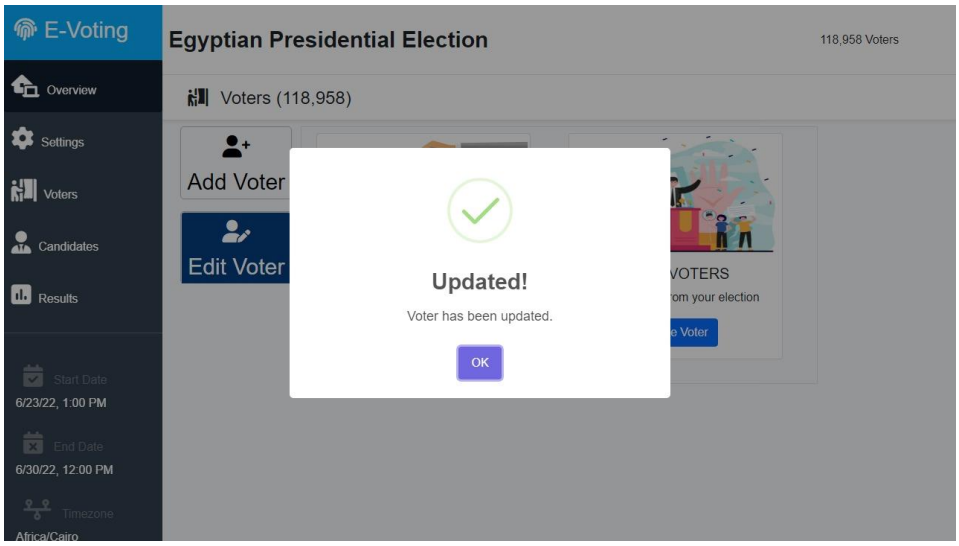

Figure 4:37 Voter data updated successfully

4.8.13 Candidates Section:

The admin can add or edit Candidates data from this section also Candidates data can be uploaded as a excel sheet with certain constraints on the file will be uploaded.

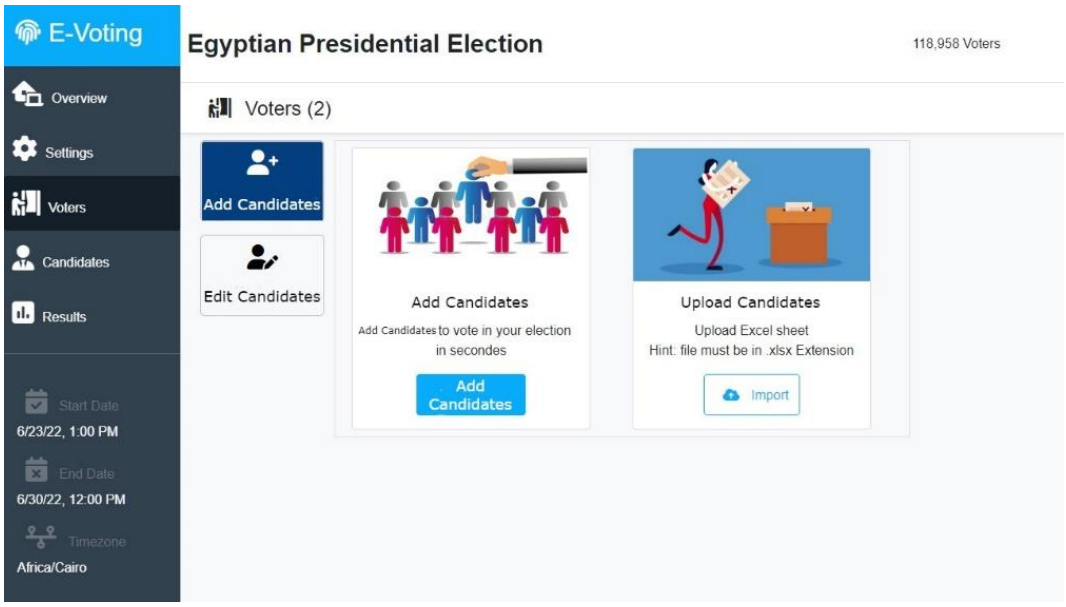

Figure 4:38 Candidate Section overview

4.8.14 Adding, Editing Candidates Data:

Editing the data which is the Candidate name, Nickname, National ID, Personal photo, and Symbol image.

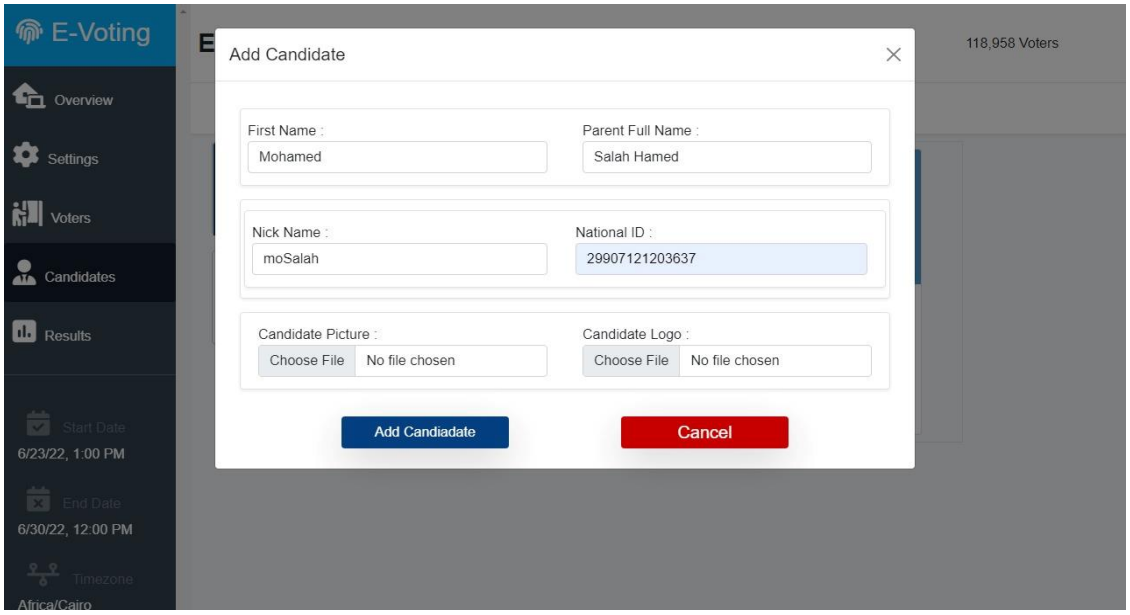

Figure 4:39 adding new candidate

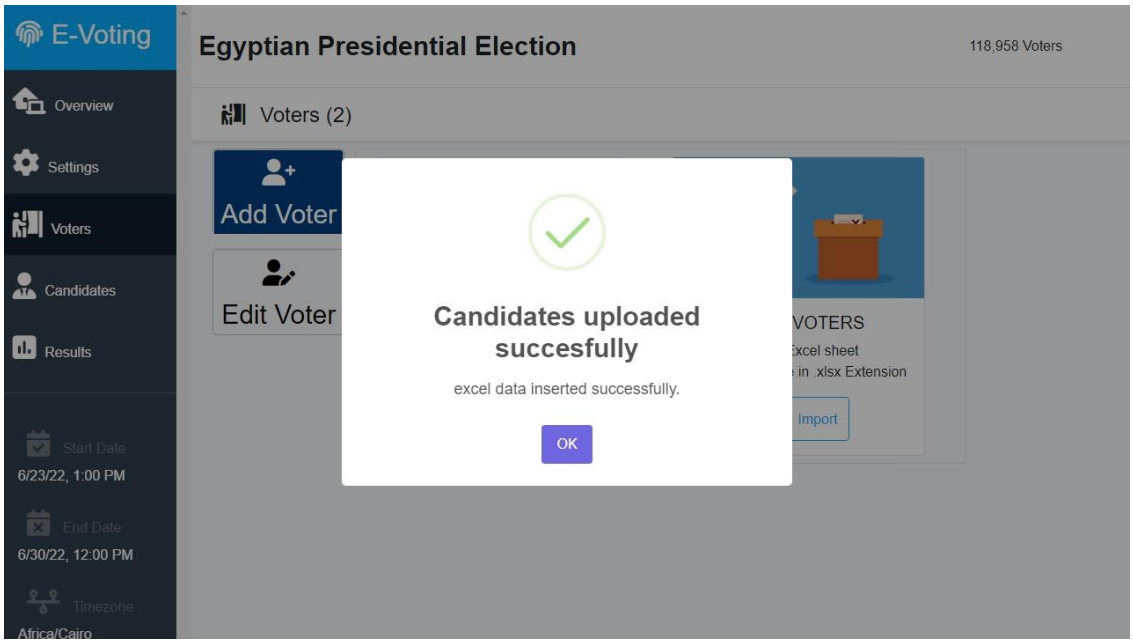

Figure 4:40 Candidate added successfully

- In case the admin need to delete a specific candidate he will search by his national ID the press delete button to delete his profile from the election.

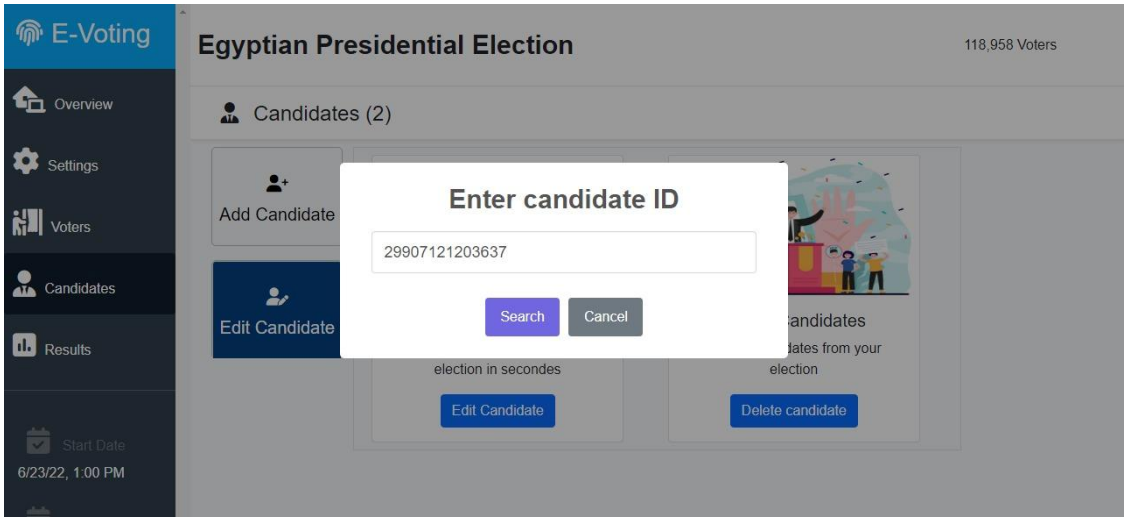

Figure 4:42 search for candidate by ID

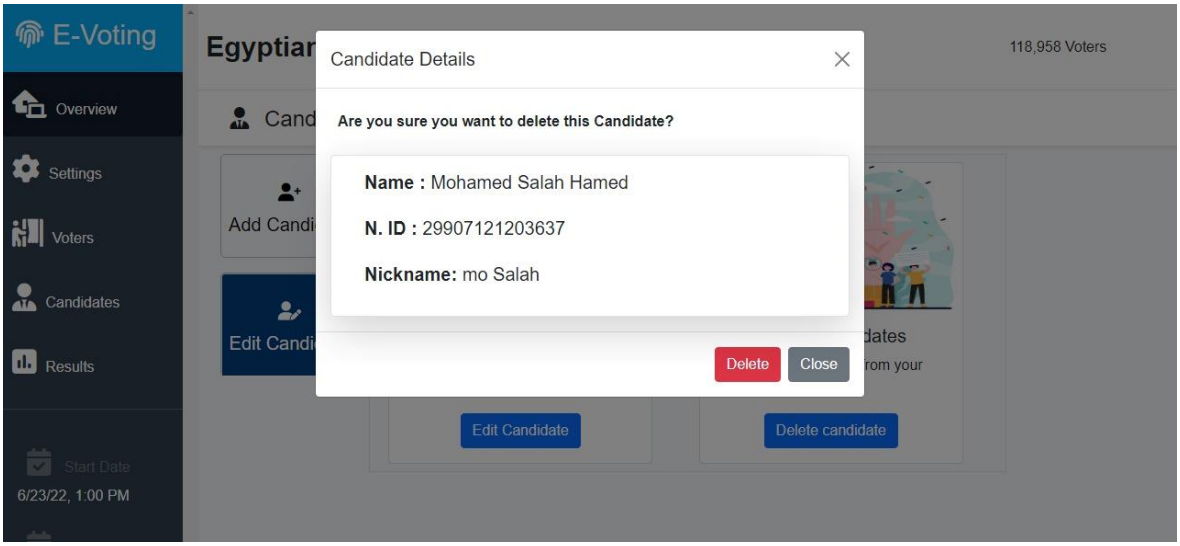

Figure 4:41 Candidate profile found

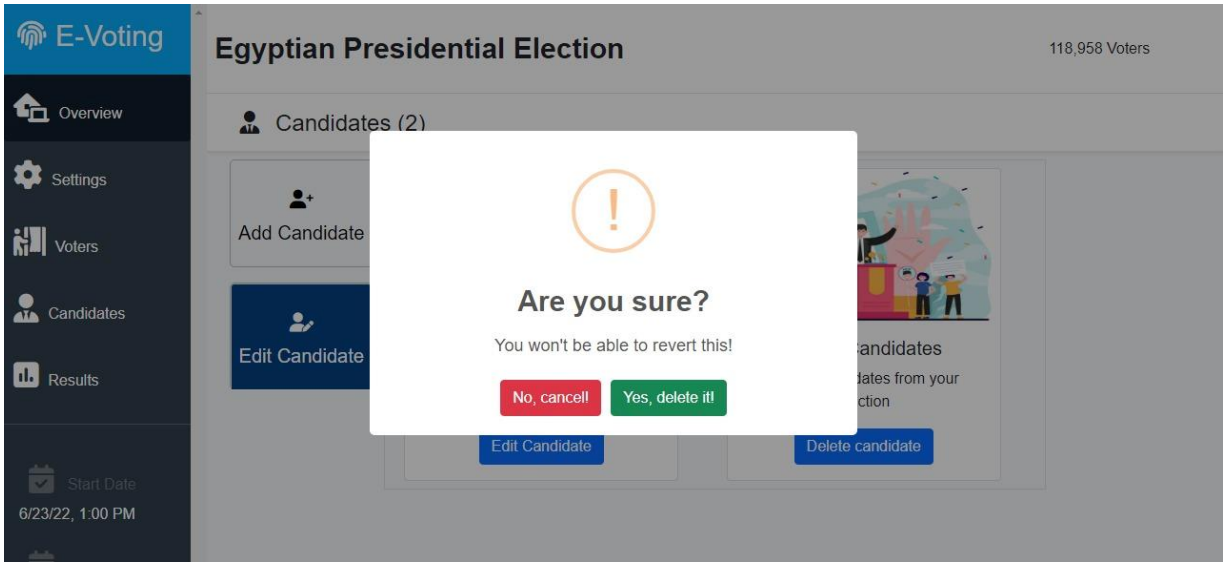

Figure 4:43 Candidate delete profile

4.8.15 Results Section:

This section contain the result of the elections process with number of votes each candidate had described is a simple easy readable bar chart and pie chart.

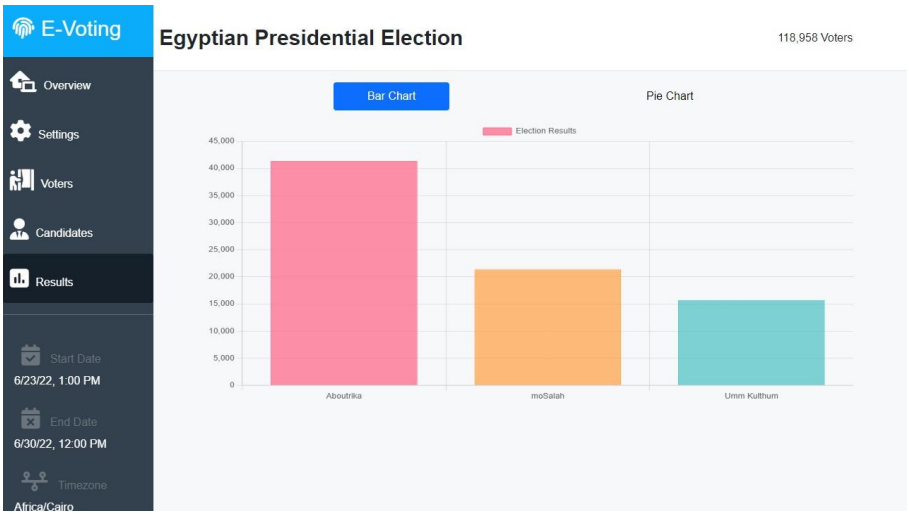

Figure 4:44 Bar chart result represent candidates and votes

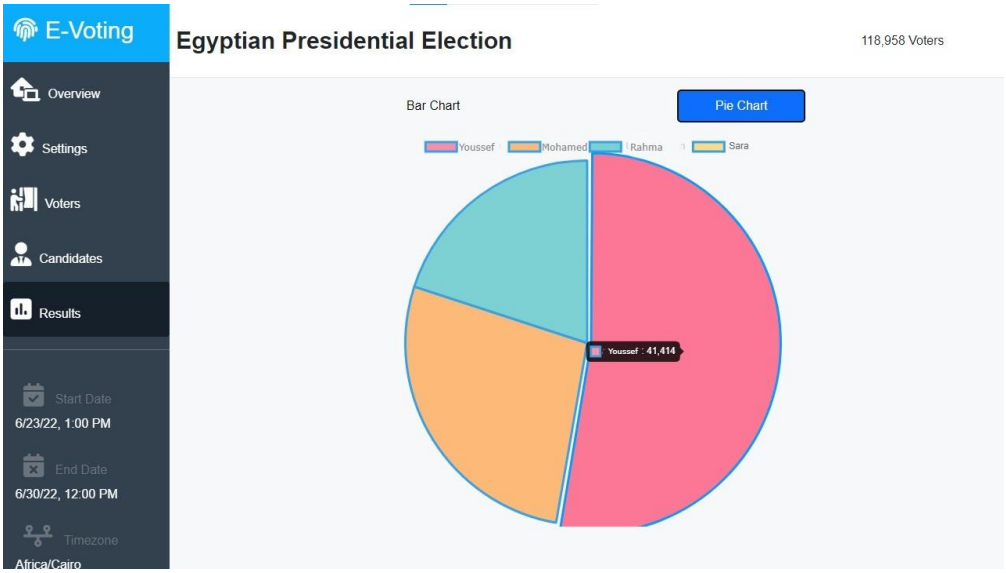

Figure 4:45 Pie chart result represent candidates and votes

Chapter 5
System Testing and Experiments

5. System Testing and Experiments:

The main purpose of system testing is to assess and value the standard of work performed at every step of the system development process. The goal of testing is to make sure that the system performs as meant and to improve the system quality, irresponsibleness, and maintainability.

5.1 Datasets:

This section describe the dataset used in the following experiments for the proposed matching algorithm which we be used in the authentication phase.

Sokoto Coventry Fingerprint Dataset (SOCOFing) is a biometric fingerprint dataset designed for academic research purposes. SOCOFing contain 6,000 fingerprint images from 600 African people and contains unique features such as labels for gender, hand and finger name as well as synthetically alternative versions with three different levels of alteration for obliteration, central rotation, and z-cut.

SOCOFing include unique features such as labels for gender, hand and finger name. Moreover, synthetically alternative versions of these fingerprints are provided with three levels of alteration for obliteration, central rotation, and z-cut using the STRANGE toolbox. STRANGE is a novel framework for the realistic synthetic alterations on fingerprint images. Alterations were done using easy, medium and hard parameter settings in the STRANGE toolbox over 500dbi resolution images. Therefore we got a total of 17,934 altered images with easy parameter settings, 17,067 with medium parameter settings, and 14,272 with hard parameter settings. Note that in some cases some

Images did not meet the criteria for alteration with specific settings using the STRANGE toolbox, hence the unequal number of altered images across all three alternation categories.

All original images were acquired based on impressions collected with Hamster plus (HSDU03PTM) and SecuGen SDU03PTM sensor scanners. SOCOFing include total of 55,273 fingerprint images combined together. All file images have the same resolution of 1 × 96 × 103 (gray × width × height).

The dataset contain two subfolders one is the real, i.e. original images, and altered images. The altered folder is further divided into three levels of alteration difficulty: easy, medium and hard.

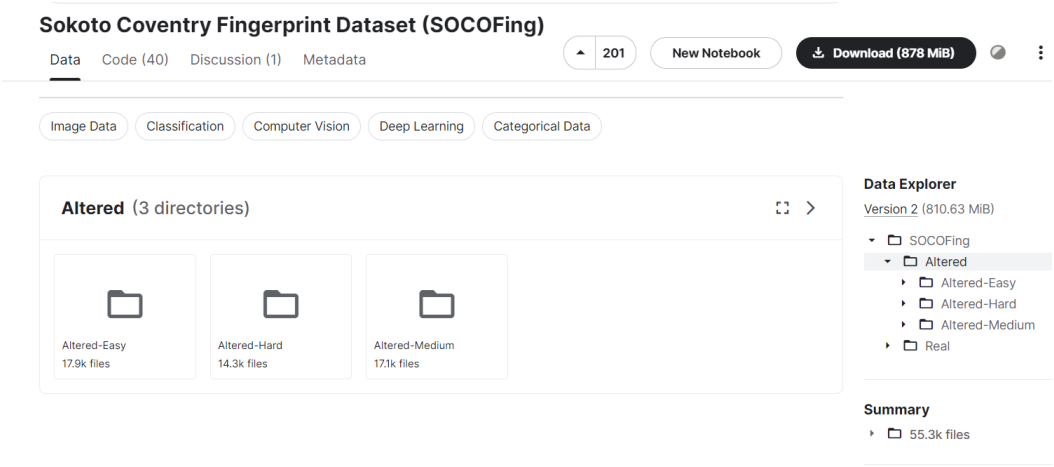

Figure 5:1 Data set altered subfolders Sokoto

5.2 Matching Algorithm Experiments:

In order to accurately evaluate the performance of the method that was picked, which is the use of Brute Force KNN matching on corner points detected by Harris Corner Detection algorithm, we used the Sokoto Coventry Fingerprint Dataset (SOCOfing).

5.2.1 Experiment 1 (Easy-altered):

Alternations on fingerprints in this experiment resulted in images with a small loss of data, the altered results still share a lot of similarities with the original image.

Figure 5:2 real Image of fingerprint dataset inside a folder

We picked 500 matching pairs and 500 non matching pairs from the easy altered data to use them for testing the fingerprint recognition system.

Confusion matrix:

		Prediction	
		Match	**No match**
Truth	**Match**	432	68
	No match	12	488

Table 2 Confusion matrix of easy altered fingerprints

Accuracy = 92%

Loss = 8%

	Precision	**Recall**	**F1 score**
Match	97.297%	86.4%	91.849%
No match	87.77%	97.6%	92.685%

Table 3 Confusion matrix of easy altered fingerprints results

5.2.2 Experiment 2 (Medium-altered)

Alternations on fingerprints in this experiment resulted in images with a medium loss of data, some fingerprint features are lost and became unable to extract.

For this experiment we picked 500 matching pairs and 500 non matching pairs from the medium altered data to use them for testing the fingerprint recognition system.

Confusion matrix:

		Prediction	
		Match	No match
Truth	Match	411	89
	No match	39	461

Table 4 Confusion matrix of medium altered fingerprints

Accuracy = 87.2%

Loss = 12.8%

	Precision	Recall	F1 score
Match	91.334%	82.2%	87.067%
No match	83.818%	92.2%	88.009%

Table 5 Confusion matrix of medium altered fingerprint results

5.2.3 Experiment 3 (Hard-altered):

Alternations on fingerprints in this experiment resulted in images with a high loss of data, a lot of the fingerprint features are lost and became unable to extract. The altered results share only few similarities with the original image.

For this experiment we picked 500 matching pairs and 500 non matching pairs from the hard altered data to use them for testing the fingerprint recognition system.

Confusion matrix:

		Prediction	
		Match	No match
Truth	Match	339	161
	No match	124	376

Table 6 Confusion matrix of hard altered fingerprints

Accuracy = 71.5%

Loss = 28.5%

	Precision	**Recall**	**F1 score**
Match	73.218%	67.8%	70.509%
No match	70.018%	75.2%	27.609%

Table 7 Confusion matrix of hard altered fingerprints results

5.3 Performance evaluation:

Evaluation of the experiments done in the above section in details experimented on different devices to calculate the average time on different devices.

5.3.1 Experiment 1:

Computer specifications:

CPU	Intel core i7 7th generation
GPU	Nvidia GTX 1050ti
RAM	20gb DDR4

Table 8 Device 1 used in experiment no.1

Processing time averages:

Average image preprocessing time	2.61s
Average feature extraction time	0.74s
Average matching time	0.47s
Total	3.72s

Table 9 Experiment 1 evaluation

5.3.2 Experiment 2:

Computer specifications:

CPU	Intel core i7 4th generation
GPU	Amd R5
RAM	8gb DDR4

Table 10 Device 2 used in experiment no.2

Processing time averages:

Average image preprocessing time	3.13s
Average feature extraction time	0.89s
Average matching time	0.56s
Total	4.58s

Table 11 Experiment 2 evaluation

5.3.3 Experiment 3:

Computer specifications:

CPU	Intel core i5 7th generation
GPU	Intel HD
RAM	6gb DDR4

Table 12 Device 3 used in experiment no.3

Processing time averages:

Average image preprocessing time	3.39s
Average feature extraction time	0.96s
Average matching time	0.61s
Total	4.96s

Table 13 Experiment 3 evaluation

5.4 Types of System Testing:

There are numbers of types of software testing, categorized by what is being tested and the purpose, or objective, of the test. The objectives range from usability to disaster recovery. Generally, the most common testing types are unit testing, interface testing, functionality was testing, compatibility testing, performance testing, scalability testing, and security testing. A simple definition of the previous testing **types is as follows:**

- **Unit testing:** Functional and reliability testing in an Engineering environment. Producing tests for the behavior of components of a product to ensure their correct behavior before system integration.
- **Interface testing:** done by the user.
- **Functionality testing:** Validating an application or website conforms to its specifications and correctly accomplishes all its required functions. It may include testing the mathematical and algorithm correctness of scientific and financial software, as well as testing GUI functionality.
- **Compatibility testing:** To ensure compatibility of an application or website with different browsers, operating systems, and hardware platforms. Compatibility testing can be performed manually or can be driven by an automated functional or regression test suite.

- **Performance testing:** Performance testing determines how well the software performs in terms of the speed of computations or responsiveness to the user.
- **Scalability testing:** Scalability testing is performed to ensure that the software will function well as the number of users, size of data sets other factors change from small to large values.
- **Security testing:** Application security testing determines how well the software can defend against attacks, such as firewall software securing a computer against Internet viruses and worms. In the case of the web application, it refers to the testing of the site and web server configuration to eliminate any security or access loopholes.

5.5 Testing Template:

Score	Bad	Average	Good	Very Good	Excellent
User Interface					
Design					
Functionality					
Usefulness					

Table 14 Testing Template explanation

- **User Interface:** how easy does the user navigate and use the interface?
- **Design:** the user's opinion on the GUI layout
- **Functionality:** how well the system operates its functions?
- **Usefulness:** how well do the users find our system useful?

5.6 System Testing Evaluation:

- Evaluation is made to determine how our system is qualified and how the user interacts with it.
- User testing provides what users think of the system. If the scores of the user interface didn't rate highly, then the program didn't meet the user's expectations.
- If the functionality and the design were complicated and not easy to use; therefore, the user will not be satisfied using our system.

Score	Bad	Average	Good	Very Good	Excellent
User Interface				**4.1**	
Design				**3.5**	
Functionality			**4.3**		
Usefulness				**4.5**	

Table 15 Proposed system testing evaluation

- **User Interface (4.1):** System User Interface will be easily used by many users, and it can be improved in the next version.
- **Design (3.5):** System Design colors need to be changed better and more special effects to be added.
- **Functionality (4.3):** System specifications are great but need more improvements when it comes to the speed of response of the system.
- **Usefulness (4.5):** System is useful but needs to be improved on the services page to make it more useful to the user to use the application.

Chapter 6
Conclusion and future work

6. Conclusion and Future Work:

This chapter is about the conclusion and the future work of the project. Section 6.1 includes the conclusion of the project. Section 6.2 presents the future work of the project.

6.1 Conclusion:

In this project, we have learned a lot of skills that improved our ability to work in teamwork and gained a lot of knowledge in the following areas:

- Voting history and how the operation of voting is created.
- The importance of biometrics in our daily life and how to use it with the present technology.
- Surveying the different types of machine learning models and image processing and how to implement them.
- Featuring the system requirements using UML diagrams and database.
- The importance of system analysis UML diagrams.
- How teamwork can be powerful in developing large and effective applications.
- How system testing makes sure that the system runs without errors or bugs.

Our project implementation consists of:

- Machine learning models for biometrics recognition.
- GUI for user and admin to cast and create a voting process.
- Database contain the data of the users and admins which will use the system.

6.2 Future Work:

In addition to our work done it can be improved by adding some functions to improve our system such as:

- Using NFC technology included inside our national ID so that the user doesn't need to wright his id number during searching phase.
- Running different election processes at the same time.
- Add IRIS print as advanced method beside fingerprint for more security as the optical scanner can be easily hacked.
- Cross platform so that it can be used with different types of fingerprint sensors.
- Provide the capability to replace the national id and search for citizen with his biometrics feature directly.

References:

1. Hum, Yan Chai; Lai, Khin Wee; Mohamad Salim, Maheza Irna (October 11, 2018). "Multiobjectives bihistogram equalization for image contrast
2. Laughlin, S.B (1981). "A simple coding procedure enhances a neuron's information capacity".
3. *Intel Corporation (2001).* "Open Source Computer Vision Library Reference Manual" (PDF). Retrieved January 11, 2017.
4. https://www.ijert.org/research/a-finger-print-based-voting-system-IJERTV4IS050948.pdf
5. https://www.ft.com/content/b4425338-6207-49a0-bbfb-6ae5460fc1c1
6. https://inecnigeria.org/
7. Mistry, Darshana, and Asim Banerjee. "Comparison of feature detection and matching approaches: SIFT and SURF." GRD Journals-Global Research and Development Journal for Engineering 2.4 (2017): 7-13.
8. Tareen, Shaharyar Ahmed Khan, and Zahra Saleem. "A comparative analysis of sift, surf, kaze, akaze, orb, and brisk." 2018 International conference on computing, mathematics and engineering technologies (iCoMET). IEEE, 2018.
9. Malkov, Yu A., and Dmitry A. Yashunin. "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs." IEEE transactions on pattern analysis and machine intelligence 42.4 (2018): 824-836.

10. Roosab, Daniel R., Elcio H. Shiguemori, and Ana Carolina Lorenac. "Comparing ORB and AKAZE for visual odometry of unmanned aerial vehicles." (2016)

11. Chris Harris & Mike Stephens in their paper A Combined Corner and Edge Detector in 1988

Appendix:

Image processing Implemented Code:

```
# Histogram equalization

def histeq(img):

clahe = cv2.createCLAHE(clipLimit=2.0,tileGridSize=(8,8))

        img = clahe.apply(img)

Return(img)

# Binarization

def binarize(img):

ret, img = cv2.threshold(img ,127,255,cv2.THRESH_BINARY)

return(img)

def enhance(self, img, resize=True):

    # main function to enhance the image.

    # calls all other subroutines

    if(resize):

        rows, cols = np.shape(img)

        aspect_ratio = np.double(rows) / np.double(cols)
```

```
        new_rows = 350                  # randomly selected number
        new_cols = new_rows / aspect_ratio
        img = cv2.resize(img, (np.int(new_cols), np.int(new_rows)))
    self.__ridge_segment(img)   # normalise the image and find a ROI
    self.__ridge_orient()       # compute orientation image
    self.__ridge_freq()         # compute major frequency of ridges
    self.__ridge_filter()       # filter the image using oriented gabor filter
    return(self._binim)
#Thinning
def thin(img):
        skeleton = skeletonize(img)
        skeleton = numpy.array(skeleton, dtype=numpy.uint8)
return(skeleton)
```

Feature Extraction Appendix:

```
#Feature Extraction

def get_des(img):

    # Normalize to 0 and 1 range

    img[img == 255] = 1

    # Harris corners

    harris_corners = cv2.cornerHarris(img, 3, 3, 0.04)

    harris_normalized = cv2.normalize(harris_corners, 0, 255,
norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32FC1)

    threshold_harris = 75

    # Extract keypoints

    keypoints = []

    for x in range(0, harris_normalized.shape[0]):

        for y in range(0, harris_normalized.shape[1]):

            if harris_normalized[x][y] > threshold_harris:

                keypoints.append(cv2.KeyPoint(y, x, 1))

    # Define descriptor
```

```
    orb = cv2.ORB_create()
    # Compute descriptors
    _, des = orb.compute(img, keypoints)
    return (keypoints, des);
```

#KNN and score Calculation:

```
def BFKNN(des1,des2):
    bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
    matches = sorted(bf.match(des1, des2), key= lambda match:match.distance)
    # Calculate score
    score = 0;
    for match in matches:
        score += match.distance
    score_threshold = 34.5
    l = len(matches)
    x = score/l
    print(l,x)
    if x < score_threshold and l>20:return True   else:   return False
```