



**UNIVERSITY OF NAIROBI**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING**

**PROJECT: FINGERPRINT-BASED EXAMINATION HALL AUTHENTICATION  
SYSTEM**

**PROJECT INDEX : PRJ-013**

**SUBMITTED BY : KIRAGU FRANCIS KIIRU**

**REG. NO. : F17/1685/2015**

**SUPERVISOR : PROF. VITALIS ODUOL**

**EXAMINER : DR. PETER AKUON**

**SUBMITTED ON: .....**

This project report is submitted in fulfillment of the requirement for the award of the degree of Bachelor of Science in Electrical and Electronic Engineering at the University of Nairobi.

## DECLARATION OF ORIGINALITY

**NAME OF STUDENT:** Kiragu Francis Kiiru

**REG. NO:** F17/1685/2015

**COLLEGE** : Architecture and Engineering

**FACULTY** : School of Engineering

**DEPARTMENT** : Electrical and Information Engineering

**COURSE NAME** : Bachelor of Science in Electrical and Information Engineering

**TITLE OF WORK** : Fingerprint-Based Examination Hall Authentication System

- 1) I understand what plagiarism is and I am aware of the university policy in this regard.
- 2) I declare that this final year project report is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people's work or my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi's requirements.
- 3) I have not sought or used the services of any professional agencies to produce this work.
- 4) I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work.
- 5) I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with University anti-plagiarism policy.

**SIGNATURE:** \_\_\_\_\_

**DATE:** \_\_\_\_\_

**APPROVED BY:**

**Supervisor: Prof. Vitalis Oduol**

**SIGNATURE:** \_\_\_\_\_

**DATE:** \_\_\_\_\_

## **DEDICATION**

I dedicate this to my God Yahweh who through my Father supported me in the entirety of this project both financially and with all the resources I needed to its completion.

## **ACKNOWLEDGMENT**

First, I would like to thank my God for giving me life and chance to take this course at this great University and for enabling me to successfully do the project.

Special thanks goes to my supervisor Prof. Vitalis Oduol for believing in my capability to do the project and also for his great encouragement and support.

I am also grateful to the department of Electrical and Information Engineering for entrusting me with this project and for providing with the resources and facilities I needed to do this project.

I am very grateful to my family for supporting me throughout my campus life until I completed this project.

## **TABLE OF CONTENTS**

<b>DECLARATION OF ORIGINALITY .....</b>	<b>2</b>
<b>DEDICATION .....</b>	<b>3</b>
<b>ACKNOWLEDGMENT .....</b>	<b>4</b>
<b>CHAPTER 1 .....</b>	<b>8</b>
<b>INTRODUCTION .....</b>	<b>8</b>
Background.....	8
Problem Statement.....	9
Project Justification.....	10
Main Objectives .....	10
Scope .....	11
<b>CHAPTER 2: .....</b>	<b>12</b>
<b>LITERATURE REVIEW .....</b>	<b>12</b>
Technology used .....	12
Fingerprint Sensor Technology .....	12
Capacitive fingerprint sensors .....	12
Pressure Fingerprint sensor .....	13
Optical Fingerprint sensor .....	13
Ultrasound Fingerprint sensors .....	15
Liquid Crystal Display (LCD) .....	16
ATMEGA 328P .....	17
Communication Protocols .....	20

Serial Peripheral Interface (SPI) protocol .....	21
Universal Asynchronous Receiver/ Transmitter Protocol .....	22
Inter-Integrated Circuit Protocol .....	23
<b>CHAPTER 3: .....</b>	<b>23</b>
<b>METHODOLOGY AND DESIGN .....</b>	<b>23</b>
Introduction .....	23
<b>HARDWARE .....</b>	<b>24</b>
Arduino UNO .....	24
LCD Display .....	25
The I2C Bus .....	27
Keypad .....	28
Piezoelectric buzzer .....	29
R307-Fingerprint scanner .....	29
Design of the hardware outline .....	31
Software algorithms .....	33
Fingerprint scanner software .....	33
Flowcharts .....	33
Web-based complimentary system software .....	40
Flowchart .....	40
<b>CHAPTER 4 .....</b>	<b>41</b>
<b>RESULTS AND DISCUSSION .....</b>	<b>41</b>
Hardware Device .....	41
Software Device .....	42
Process flow .....	47
<b>CHAPTER 5 .....</b>	<b>48</b>
<b>RECOMMENDATIONS .....</b>	<b>48</b>
<b>CHAPTER 6 .....</b>	<b>49</b>
<b>CONCLUSION .....</b>	<b>49</b>
<b>REFERENCES .....</b>	<b>50</b>
<b>APPENDICES .....</b>	<b>51</b>
Appendix 1: Bill of Quantities .....	51
Appendix 2.0: Device Code .....	51
Appendix 2.1: Importing Libraries and declaring variables .....	51

Appendix 2.2: Setup function .....	53
Appendix 2.3:Initialization/greetings .....	53
Appendix 2.4: The main function .....	55
Appendix 2.5: Authentication of Admin .....	60
Appendix 2.6: Informing modes of operation .....	61
Appendix 2.7 Reading keypad.....	62
Appendix 2.8 Emptying flash memory .....	63
Appendix 2.9: Deleting a template.....	64
Appendix 2.10: Enrollment .....	66
Appendix 2.11:Getting fingerprint id .....	76

## **CHAPTER 1**

### **INTRODUCTION**

#### **Background**

For decades examinations have been the mode of testing the knowledge acquired by learners in institutions. This method has always been found to rely on the grades acquired to prove a pass of qualification test. Examinations take minutes to hours and in this period, the candidates are expected to deliver what they have learnt for a given term such that examinations' times are the most critical times for students and learners in various levels of education. Due to the pressure associated with this critical time, there has been rise in the cases of examination malpractices such as cheating and many other irregularities. There has been an increase of cases of examination cheating in most tertiary, secondary and primary learning institutions. There are extreme cases when a learner is not ready to sit for an examination, therefore he/she hires another student to sit for his examination. In this case, the student who sits in the examination room presents identification materials to the invigilators and in most cases their identity is not well matched or established. The materials used are such as: identity cards with passport photo or national identity cards. These identity cards are easy to transfer from one person to the other especially if the passport photo in them are not clear to differentiate the owner from the impostor. These photos are usually incomplete, discolored and of poor quality due to wear and tear. This poses as a challenge since the examiner cannot tell whether the candidate who presented himself for an examination is accurately the valid candidate or an impostor. Most cases are when students sit for their colleagues' examinations unsuspectedly using the latter's identity cards. The other methods used are examination cards which contain the details of the courses and the examinations that an examination candidate is allowed to sit for, alongside the student details like registration number and name. The shortcomings of this method is worse than the use of the



identity cards with passport photos since it's really difficult to establish the identity of the student who presents the examination card.

The other methods used are biometric authentication systems. Cambridge Advanced Learner's dictionary defines biometrics as the use of someone's unique body characteristics and features to establish their identity. The most commonly known methods are, use of fingerprints, eye retina scanning, voice recognition, iris scanning, facial recognition, finger geometry among others. The use of biometrics is unique due to the fact that biometric data is difficult to transfer from one person to the other and also due to the fact that it's biologically unique in every individual. A fingerprint authentication system utilizes the unique nature of every finger prints pattern in every individual. Fingerprints have the advantage that they do not change with age, as compared to facial scanning whose efficiency relies on the facial features which can change with time.

## **Problem Statement**

As briefly stated above, there are problems that arise from using examination cards and identity cards. They are easily transferred from one person to the other, hence they are misused and impostors can easily acquire them to sit for an examination. These methods that rely on something that a candidate holds are also very unreliable in the case that these items of identification are lost or misplaced. In such a scenario it's difficult to establish his/her identity and it would lead to automatic disqualification from sitting for exams. This poses as a problem in case the misplacement happens by accident or if the items of identification are severely damaged. It would unfairly deny the student the right to sit for an exam. The same problems result in case the cards are stolen.

It is also worth to note that using of cards as authentication materials is neither accurate nor quick enough. The process of checking verifying the details by human personnel suffers human error due to reasons such as fatigue and poor concentration. For large numbers of examination candidates, the process would be very slow unless a large number of human personnel is assigned the verification process. This poses the problem of man-power cost. It is also a problem since the higher the number of people involved in authentication would lead to integrity problems. This leads us to the other challenge.

Corruption and favoritism is another challenge that comes along with the conventional methods of examination hall authentication system. There are possibilities of the invigilators allowing some students to irregularly sit for an examination so as to gain some personal favors or due to blackmail. In such a case it's difficult to fairly verify all the candidates. In this case, the authentication is not credible when such possibilities are considered. If there are bribes and preferential treatments by the personnel expected to actualize the authentication, the process becomes useless.

It's also of worth to note that reliance on human support and intervention poses to be the major challenge in the authentication of examination halls as seen in the problems stated above. Human is prone to error, fatigue, biasness, inaccuracy, inefficiency, time wastage and man-power costs. In addition to that, the materials used have got the risks of wear and tear and misplacements. They are also easily transferable to impostors who sit for examinations in place of the supposed students.

These challenges are can be solved using a fingerprint based hall authentication system.

## **Project Justification**

The project solves the above problems by using biometrics to establish the identity of examination candidates by an efficient authentication. The fingerprints cannot be transferred from a candidate to an impostor. Fingerprints are also reliable than identity cards since during verification and authentication, the process is achieved by microcontroller technology which does not suffer fatigue as compared to human verification processes. Finger print authentication system offers a high level of integrity when the data is securely stored. Corruption and other preferential treatments are impossible while using these machine interfaces. This project focuses on using fingerprint authentication that authenticates examination candidates while its complimented by a separate database that stores students' information corresponding to a given fingerprint template in the physical electronic system.

## **Main Objectives**

- The authentication system will enroll students who are allowed to sit for an examination.
- The system stores the details of these students into remote database alongside with the id of the fingerprint templates stored in the fingerprint scanner.

- The system will authenticate the students by scanning their prints at the examination hall entrance and it will indicate the candidature of the student is valid, if and only if their prints match with those already stored.
- The system displays the fingerprint id of the scanned candidate if there is need to confirm their details from the remote database on a browser interface.
- The system has got a web-browser interface to query the details of the student by fingerprint to view extra information about a candidate.

## **Scope**

The project goal is use fingerprint biometrics to establish the identity of examination candidates. These details are used to authenticate the examination candidates into an exam hall. The system uses an electronic fingerprint scanner to facilitate enrollment, storage and authentication of the fingerprints. The system utilizes a complimentary database system that remotely stores the students' details separately for a more detailed verification. It utilizes a web-based interface to query the database using the candidate id and returns all the information about that student including a facial passport photo. If a student or candidate is successfully authenticated, the system gives an alert by LED and also on a display screen. The fingerprint scanner used stores the fingerprint templates in its own flash memory which has got a limit of the templates stored up to 1000 pieces.

## **The process would be as follows:**

All students that are to sit for the examination must first register with their fingerprints as biometric details. The fingerprint templates are stored using electronic fingerprint scanner while the other information is stored into a separate remote database. During the examination, the fingerprints are scanned, they are checked whether they are in the scanner memory by matching the templates with existing ones. The absence of the fingerprints indicates that the student is not authorized to sit for the examination. If the prints match with any of those in the storage, then the student is allowed to sit for the exam by giving a green/blue light with an LED. The system would therefore need, a storage means for the student prints and other information, it would also require a fingerprint scanner operated by a microcontroller that enables it to perform all the operations.

## **CHAPTER 2:**

### **LITERATURE REVIEW**

#### **Technology used**

##### **Fingerprint Sensor Technology**

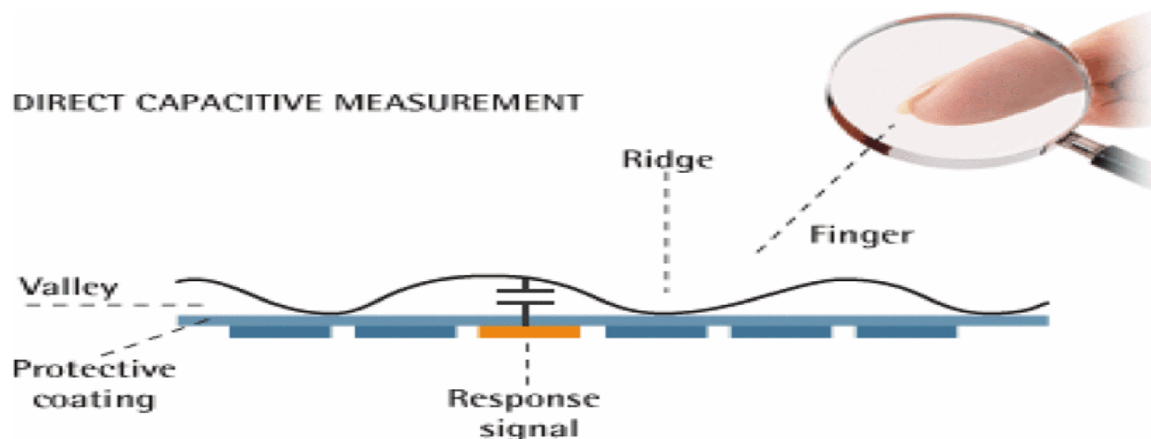
What is a fingerprint? A fingerprint is an impression formed by the epidermis of the skin at the tips of a human fingers due to the ridges and furrows. This impression is defined by three main characteristics, namely: patterns of ridges and furrows (level 1), minutiae points (level 2) which are the unique properties of a given ridge and shapes of ridges and furrows (level 3). Level 1 features can be whorl type, loop type, arch type or different type of loop. Level 2 features can be defined by the endings of ridges, the islands, bifurcations, scars, incipient ridges and flexion creases. Level 3 is defined by pores and the ridge units. According to the Scientific American publication, two like fingerprints would be found only once every  $10^{48}$  years, which makes them reliable biometric features for unique recognition.

A fingerprint sensor is an electronic device used to take the features of fingerprints. The sensing process is either live-scanning or offline scanning where in the former case, the fingerprints are directly scanned by the sensor while for the latter, the fingerprints are scanned from a recorded image. There are different types of scanners namely: Optical fingerprint sensors, capacitive fingerprint solid-state sensors, pressure fingerprint sensors and thermal fingerprint sensors including others.

##### **Capacitive fingerprint sensors**

These are sensors that utilize the electrical capacitance phenomenon where electric fields exists between two separated plates which are at reasonable proximity. The capacitance between two plates is directly proportional to the area between them and inversely proportional to the length of the distance between them. This is applied to the separation between the fingerprint furrows

and the capacitive sensor. Furrows are distanced from the sensors more than the ridges hence lesser capacitance as compared to the ridge points. These values of capacitances are captured by the sensors at different points and hence help to distinguish ridges and furrows. A capacitive sensor can either be active capacitive or direct capacitive (Qiu, 2020).



The capacitive fingerprint sensors are smaller in size as compared to optical scanners. This type of scanners have also have got low power consumption. Their drawback is that they can suffer from electrostatic discharge. They are also affected by stray electromagnetic fields. The sensors used are also more expensive.

### **Pressure Fingerprint sensor**

This is a type of fingerprint sensor that utilizes piezoelectric sensors to detect the pressure applied by the ridges as compared to the little or no pressure applied by the furrows. The sensors will distinguish the ridges and the furrows using the pressure difference. This will generate a pattern that forms the template generated by the microprocessor of the fingerprint scanner. The images generated by this method are not good due to the poor sensitivity of pressure sensors on the scanner surface (How fingerprint scanners work: optical, capacitive, and ultrasonic variants explained, 2020).

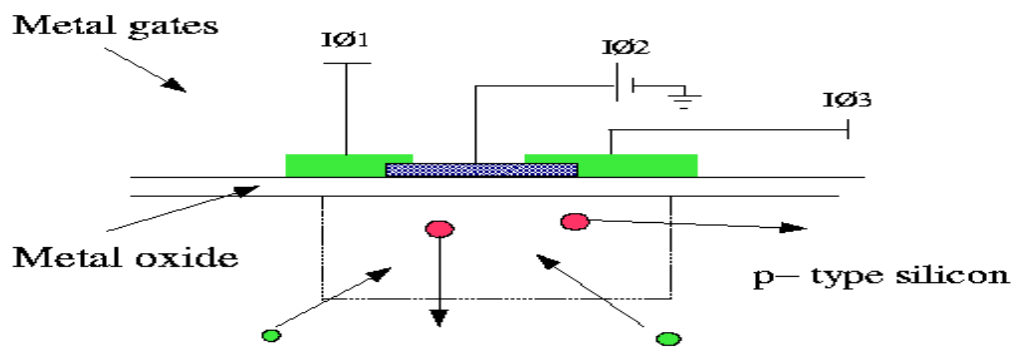
### **Optical Fingerprint sensor**

An optical fingerprint sensor works by capturing an optical. This method is the oldest one for taking fingerprints. It basically uses a camera. Fundamentally a camera is made of charge coupled devices which constitute a number of light sensitive picture elements. These picture elements are potential wells that can store electrons or holes when charge is applied on one of

their sides. When a photon of light falls on the surface of the camera screen, a given picture element (pixel) converts the light photon into a number of electrons depending on the intensity of light at each photon. Therefore given points on an image are represented by charge stored in the pixels which are potential wells.

The pixels' potential is then read by clocking the potential into a microprocessor which measures the potential in each pixel and reconstructs these charges into binary data that can be stored to represent an image. The camera for the scanner collects patterns of dark and light points due to the ridges and furrows of the fingerprint surface. The pixels also collect the image elements using the minutiae points of the shapes of these prints.

The camera processor utilizes algorithms to detect and distinguish patterns of dark points and light points of the taken image or photograph to construct a fingerprint template. These templates are formed by using the unique features of given samples of images taken. These templates are stored as the fingerprint to be used for matching. The figure below shows a basic charge-coupled device with a picture element (McFee, 2020).



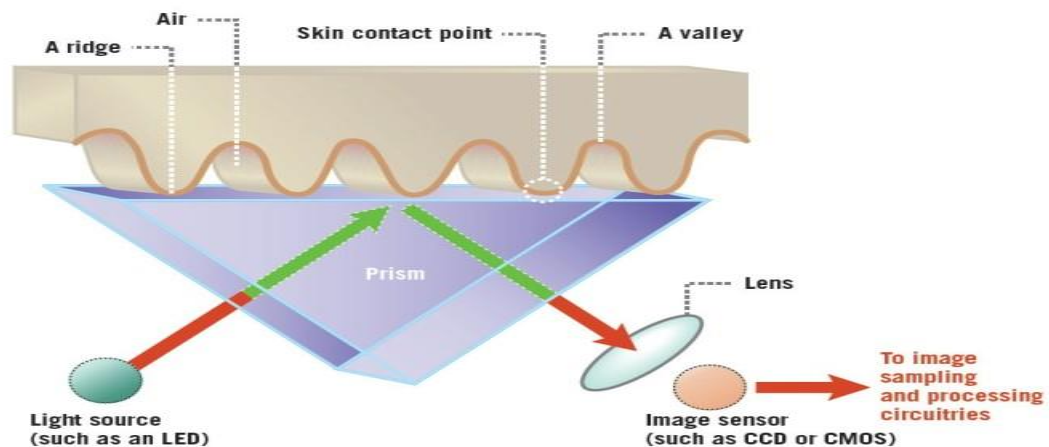
When a finger is placed on the scanner screen, the scanner illuminates the finger with LED light. The light falls on the ridges and furrows on the fingerprint surface. The camera takes a picture of the finger ridges and furrows. The image will be inverted with darker areas and lighter areas. The darker areas represent the ridges due to greater reflection at the ridges while the furrows are represented by the lighter areas due to less reflection of light. The processor checks the quality of

the image taken by the charge-coupled devices. It will reject the scan if the sampled pixels contains extremes of too light or too dark points.

The main drawback with optical scanners is that they can be fooled if plastic copy of a finger is used in authentications. The plastic material having identical patterns as the live fingerprints cannot be distinguished by the scanner. The other drawback is that they big in size as compared to capacitive scanners. They are also power consuming since the light emitting diodes must keep on illuminating the fingerprints for quality images. The optical scanners would also not work well in a strong light environment.

However the optical fingerprint scanners have the advantage that they do not suffer from electrostatic discharge as compared to capacitive scanners (How fingerprint scanners work: optical, capacitive, and ultrasonic variants explained, 2020).

### **An optical sensor.**



**Figure 2**

### **Ultrasound Fingerprint sensors**

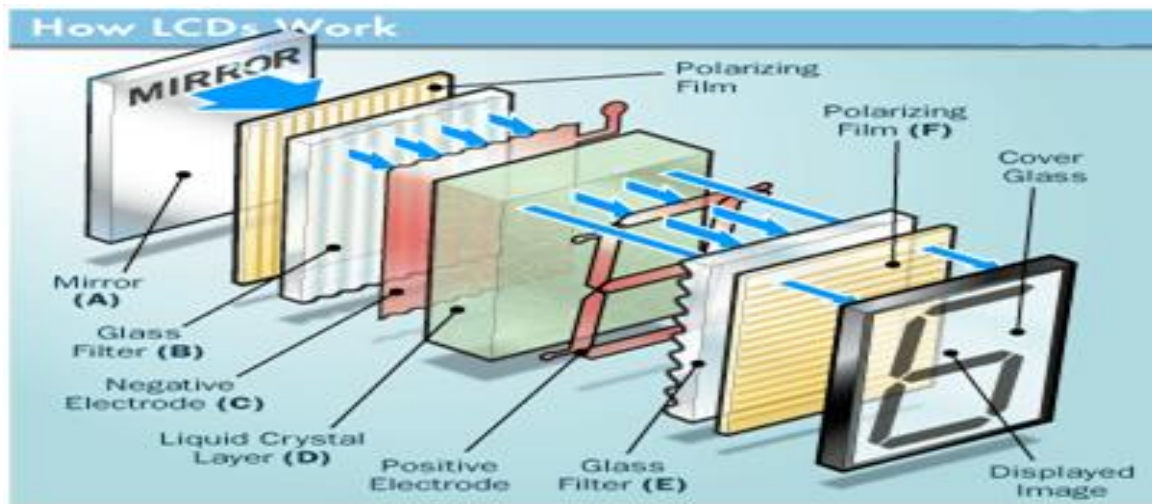
These are sensors that utilize medical ultrasonography in their working principle. High frequency sound waves are sent through the epidermis of the skin to be reflected back with the fingerprint detail. The reflected sound waves are sensed by piezoelectric sensors to obtain fingerprint details depending on the different intensities of the waves coming from the ridge and valley regions.

The epidermal characteristics in this case serve the image of the fingerprint. This method has the

advantage that the fingers do not have to be clean to get a quality image. These makes them more reliable than capacitive and optical sensors. They are also more expensive than the rest of the sensors, hence they are used in applications where reliability is critical.

## Liquid Crystal Display (LCD)

A liquid crystal is a middle phase between solid and liquid. This makes up one component of the liquid crystal displays picture elements. A pixel is therefore made of different layers of components. These are: electrodes, polarizing glasses, liquid crystal layer, mirror and a polarizing film. The following image shows the different parts (Agarwal, 2020).



The liquid crystal is the substance used to change light effects and be able to display an image on the pixel. The crystal operates in two phases, namely nematic-phase and smectic-phase. In the nematic phase, the crystals are more of a liquid than a solid. In this phase, the crystals can twist or even straighten up when electricity is applied unto them. The LCD uses polarized light in its working. What is polarized light? This is light whose transfer vector is confined to a single plane.

The LCD display has got a bright light source at the back of the screen.

To switch off the pixels of the screen, light travels from the back of the screen towards the screen. The light rays of horizontal plane vibrations is allowed to pass while light photons of any other orientation are blocked by the horizontal plane filter. The picture elements are straightened



by supplying electricity which causes them to allow light through them unchanged. The light photons with horizontal vibrations appear on the other side of the liquid crystal unchanged. The light photons then find a vertical polarizing filter which only allows vertically vibrating photons. The horizontally vibrating photons are hence blocked hence no light reaches the pixel. This turns the pixel dark.

To switch on the light, the backlight releases light photons. The horizontal plane filter blocks all light except the photons that vibrate in the horizontal plane. The power to the pixel crystal is switched off which makes them to twist. The twisted crystals, allow the light through them. This light was vibrating horizontally but now the twisted crystals makes the light to come out of the crystal vibrating vertically. This light finds the vertical plane filter which allows the light through towards the pixel which are now lit bright. The color of the filter gives the light its color, with three primary colors being blue, red and green.

This principle works in all the pixels of the screen to display different images on each pixel depending on application of electricity through switching transistors. In this project, an LCD display screen is used.

The LCDs have advantage over other types of displays since:

- They are relatively cheap.
- They are thinner and lighter than CRTs.
- They utilize microwatts.
- They provide good contrast ability.

They also have got the following disadvantages:

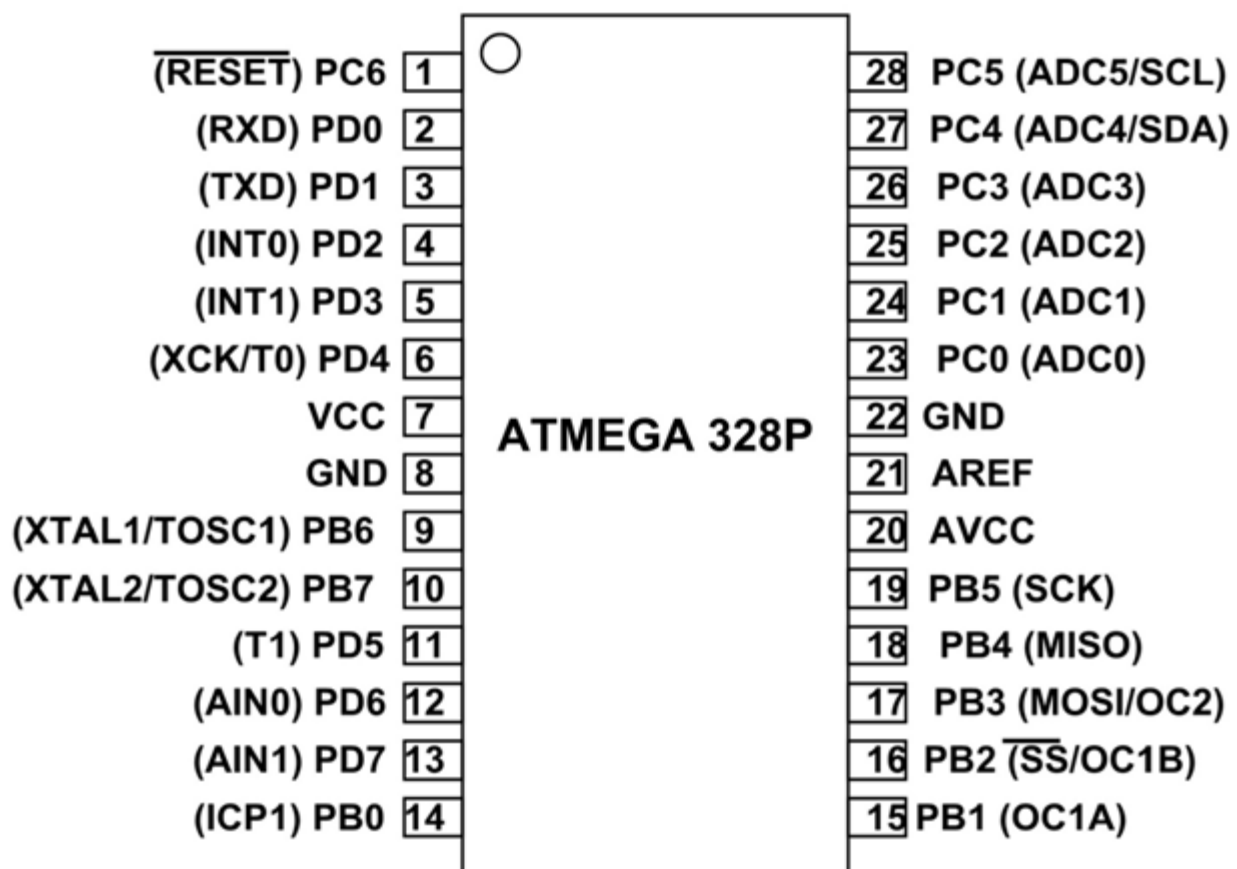
- They demand an extra light source.
- They have low speeds.
- They operate with temperature limitations.

## **ATMEGA 328P**

A microcontroller is an integrated circuit that offers control of certain operations in an embedded system. A microcontroller contains a microprocessor which provides the (CPU) central

processing unit functions to the chip. Atmega 328p is a host processor of Arduino UNO R3. The name given to the chip gives the detail about it: the 328 means it's a 28 pin chip as an 8bit microcontroller. The chip is based on RISP (Reduced Instruction Set Computer) architecture that can perform instructions at a frequency of 20MHz, at which frequency it can perform 20 million instructions per second (MISP). The microcontroller contains a memory system, a timer system, a port system, an analog to digital converter, an interrupt system and serial communications system.

The following is a pin out diagram for Atmega 328 P:



("ATMega328P Microcontroller Pinout, Pin Configuration, Features & Datasheet", 2020)

### Atmega 328 P Memory

The microcontroller uses the following types of memory:

- Flash Electrically Erasable Programmable Read Only Memory (Flash EEPROM)

This is used to store programs in the microcontroller. It is programmed as a block since during the uploading of programs, the preceding program data is overwritten. The flash memory is organized as 16K words with 16 bit bits per word. This memory is non-volatile, therefore it stores the data even after power is not supplied to the chip.

- Static Random Access Memory.

This is volatile memory which is used as the random access memory which the processor uses to store data during the program execution. Its capacity id 2K, some of it used by registers and other used by input/output.

- Byte-Addressable EEPROM.

This memory is used to store variables permanently then it uses them during the execution of the program. This memory stores the logs in case there are errors during the functioning of the chip. It is also non-volatile. Its capacity is 1024 bytes of EEPROM. Which is 1K of 8 bit words.

## **Atmega Ports**

The microcontroller 20 pins serve as general purpose **input/output pins**. They are PORT B (8 bits), PORT C (7 bits) and PORT D (8 bits). 14 of the pins are digital while 6 are analog input/output pins. 6 of the digital pins can be used for pulse width modulation. Each of these ports is designated three registers:

- Data register which writes output data to the port.
- Data Direction Register which sets a pin to either input or output.
- Input Pin Address PIN used to read input data from the port.

Out of the 28 pins, 2 are used for the **crystal oscillator pins**, which provides the clock for the synchronization of the chip.

In addition, 2 pins serve as **ground pin and power pin**.

The analog input/output pins must have their data converted from analog to digital form using an analogue to digital converter. The analog to digital converter has three pins for its functionality:

**AVCC, AREF and GND.** AVCC supplies positive voltage to the converter and AREF is its reference voltage that it uses to convert analog signals to digital signals, while GND provides the reference for power.

**RESET pin** is used to rerun the program and make it start all over.

### **Atmega 328 P Time Base**

The microcontroller performs instructions by executing the program uploaded by the user. The speed at which it performs the operations depends on the clocking system. This clock is applied also to the peripheral devices. A crystal oscillator is provided as an external clock for the sake of stability and accuracy. This external oscillator gives room for the user/designer to dictate the frequency of the clock.

### **Atmega 328 P Timing Sub-systems**

This is composed of timers and counters. These help obtain a precision output.

### **Pulse Width Modulation Channels**

Pulse width modulated signals are those signals which have a varying ON-time and OFF-time but with a constant frequency. The ON-time with relation to OFF-time is referred to as **Duty cycle**.

$$\text{Duty Cycle}[\%] = (\text{ON Time/Period}) \times (100\%)$$

The microcontroller offers analog pins that allow the setting for duty cycle and the frequency of the signals to control peripheral subsystems.

### **Communication Protocols**

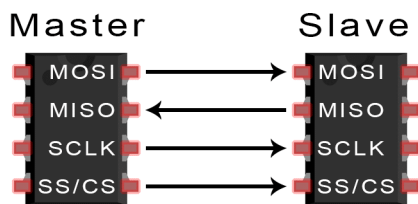
The microcontroller uses three types of protocols for communication. These are:

- Universal Synchronous and Asynchronous Serial Receiver and Transmitter (USART)
- Serial Peripheral Interface (SPI) protocol.
- Inter-integrated Circuit (I2C) protocol.

- Fast methods (Bluetooth, USB, Ethernet and WIFI).

### **Serial Peripheral Interface (SPI) protocol.**

This is a communication protocol that transfers data in continuously without interrupting the flow. In this mode, data is sent in packets of bits. This communication protocol works on the master-slave principle where the master controls the slave during sending and receiving of data. The master can control and serve many slaves (Basics, 2020).



The protocol utilizes the following pins:

- Master Input Slave Output (MISO)

This line is used by the slave to send data to the master. The data is sent with least significant bit first.

- Master Output Slave Input (MOSI)

This line is used by the master to send data to the slave. The most significant bit is sent first.

- Clock (SCLK)

This line is used to pass clock signals to the slave. In this mode of communication, a clock signal is used to synchronize the speed at which the master sends the data to the slave. It sends data in packets of 8 bits and the frequency of the clock signal controls the speed of communication between the master and slave. The clock polarity setting is done in the master and it determines where the clock signals are utilized. It can be the rising edge, falling edge or level trigger.

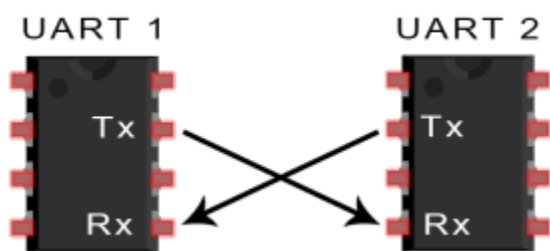
- Slave Select/ Chip Select

This is used by the master to select the chip or slave that it is to control. Different slaves are connected in parallel to the master and then each is selected using the chip select. To send data, the master sends out a clock signal, it then pulls to low a chip select pin to activate a given slave. Data is serially sent to the slave in bits.

This method is advantageous in that: it doesn't use start or end bits, has no complex addressing and data can be sent and received simultaneously. It is disadvantageous since it uses four wires, has got no error checking and allows only a single master.

### **Universal Asynchronous Receiver/ Transmitter Protocol**

In this communication protocol, parallel data is converted to a serial format. The sending UART sends data to the receiving UART. The sending UART converts parallel data into a serial form while the receiving UART converts the data back to a parallel format (Basics, 2020).



This method does not use a clock for synchronization. In place it utilizes start and stop bits to designate a packet of data sent through the serial line. The baud rate is the rate at which data is transmitted and received. The transmitting and receiving baud rates are made to be the same for proper communication. The data starts to be read when a start bit is detected while reading ends when a stop bit is detected. A parity bit is used to ensure that the integrity of the data is maintained.

This method is advantageous since: no clock signal is needed, allows error checking and uses only two wires or lines. It is also disadvantageous since: it allows packet to have a maximum of 9 bits, it does not support multiple slaves. The baud rates must be maintained at most 10 % difference between receiver and transmitter UART (Basics, 2020).

### **Inter-Integrated Circuit Protocol**

This protocol utilizes two wires only for communication. Multiple slaves can be connected to the master. It utilizes a serial data line (SDA) and also a serial clock line (SCL). The protocol is synchronous in that a single clock is used by both the master and the slave to synchronize the transmission of data bits between the slave and the master. Data is transmitted in frames with messages. Each frame contains an address for the slave to receive the data frame also contains a start bit and an end bit. This protocol also allows error checking by sending acknowledgement bits between each sent and received frame. The start condition is set by setting of SDA and SCL from high to low. The address frame is usually sent as a bit sequence to identify a given slave. A read/write bit is sent to inform the slave whether it's to read or write data to the master. The data frame is a sequence of 8 bits it's always followed by ACK/NACK bit to acknowledge the successful transmission of data. This method is advantageous since it allows acknowledgement of sent data, it supports multiple masters on a slave and only uses two wires. It's also disadvantageous since: the data frame is limited to 8 bits, it is also slower than SPI communication (Basics, 2020).

## **CHAPTER 3:**

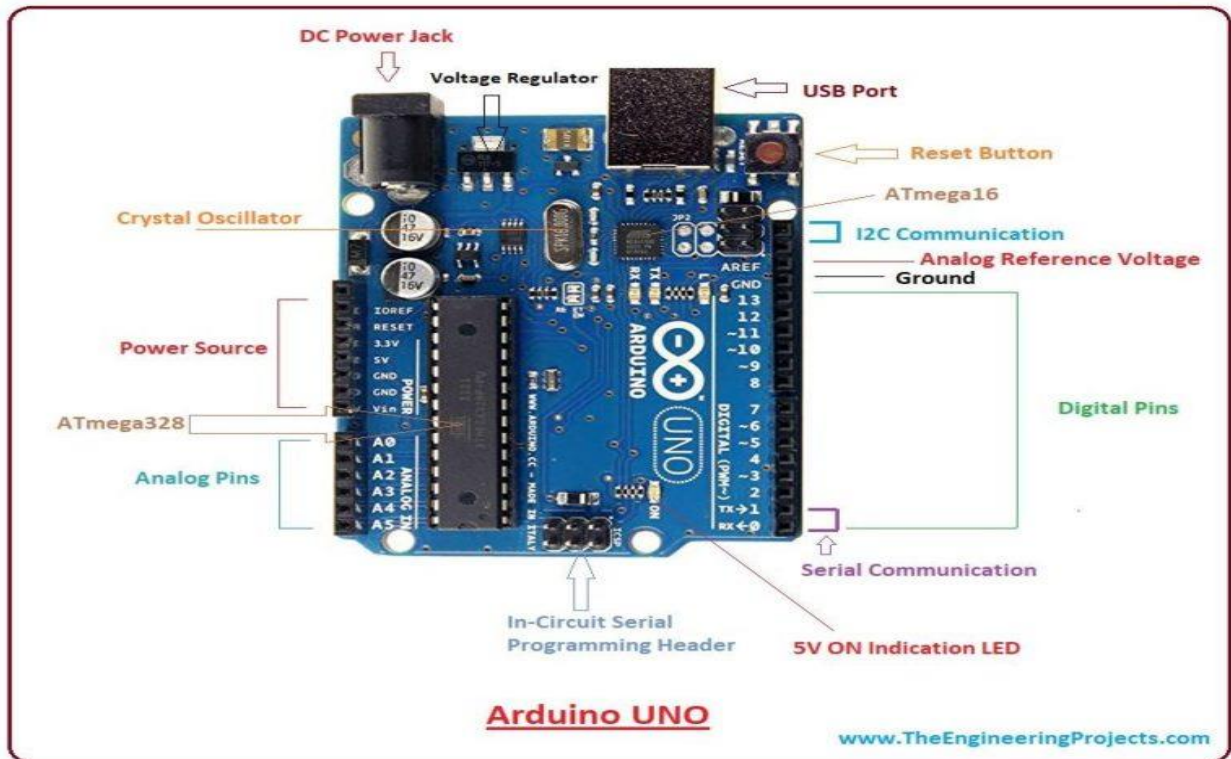
### **METHODOLOGY AND DESIGN**

#### **Introduction**

In this chapter, the methods used to achieve the final product design and implementation are outlined. These include, the software design and the hardware design. The hardware design includes the hardware components used and their characteristics and also the physical connections outlay. The software design includes database design and program algorithm flowcharts.

## HARDWARE

### Arduino UNO



("ATMega328P Microcontroller Pinout, Pin Configuration, Features & Datasheet", 2020)

The Arduino UNO was used for the design process since it makes the use of ATMEGA 328 P. The Arduino provides the supporting circuitry to the Atmega 328 P. The supporting circuitry is made of the following:

- Voltage regulator – This is used to maintain a constant and stable 5V to the AVR chip. It receives a voltage of 7 to 9 volts from the adapter port and converts it to 5v.
- Crystal oscillator - This provides a clocking means to the Atmega 328 chip. It is used for synchronization of the system using 16MHz frequency.
- Reset button – This resets the program by taking the program back to start. This button is connected to the microcontroller.
- There are 14 input/output pins on the Arduino which map to the input/output pins of the microcontroller.



- Analog Pins (A0 – A5) – These pins measure 0 to 5V. These pins have got a configuration of 10 bits.
- USB port – This is used to upload code to the microcontroller. It also supplies 5V to the board when there is no other supply.
- IOREF – This is used to provide voltage reference to the board.
- Pulse Width Modulation Pins (PMW) -3, 5, 6, 9, 10.
- Serial Peripheral Interface Pins (SPI) – pin 10 SS, pin 11 MOSI, pin 12 MISO, pin 13 SCK.
- AREF – This provides a reference voltage for the analog pins.
- Two Wire Interface which used by the I2C communication protocol. These are A4 and A5.
- Serial Communication pins – pin 0 TX and pin 1 RX.
- .LED – This is connected to pin 13. It lights if pin 13 is set HIGH.

The Arduino board is was programmed using the Arduino IDE via a USB cable.

## LCD Display



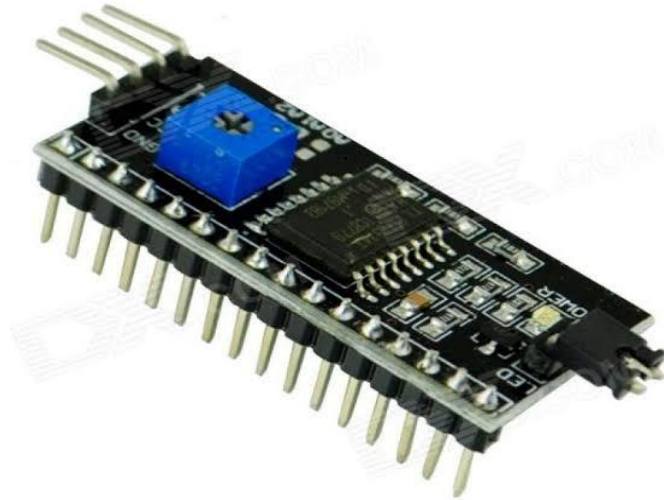
The LCD display was used as the user interface for giving instructions for the use of the system. The LCD has got two rows with 16 columns each. It uses a backlight to light the liquid crystals. It uses parallel data lines. The lines transmits ASCII characters and control characters. The

control characters are manipulate the various display functions while the ASCII characters are the one which are displayed by the module. The module therefore has got the following pins:

- Pin D0 – D7 – These are data input pins.
- GND pin – Ground pin.
- 5V pin – power pin.
- Brightness pin – This pin is supplied with the variable voltage control the brightness.
- RS pin – This is Register Select pin, it establishes whether the data sent is used as control or display data.
- RW – This pin sets the LCD module to either read or write mode. We set this pin to ground voltage to use it in the read mode for this project.
- EN – This pin is Enable pin. It is set to high to allow reading of the data pins. It is set to low to allow execution of the instructions sent to it.

Since the LCD module uses a total of 8 input/output pins, it's undesirable to use it on Arduino which has a limited number of pins. The Arduino has got 14 input/output pins and if we use the LCD module as it is, we would utilize more than half of the digital input/output pins of the Arduino leaving no pins for the other peripheral components. The I2C bus is used to reduce the number of pins used to only two.

## The I2C Bus



This is used to establish an Inter-Integrated Circuit communication protocol. This protocol utilizes only two serial lines: a serial data line and a serial clock line, SDA and SCL. This allows sending of data serial minimizing the large number of pins utilized by parallel communication of the LCD module. This clock line is used by both the master and the slave for synchronization of their communication and data transfer. The I2C has got the following:

- Power which is 5V or can also be 3.3V, but depends on the application of the bus.
- SDA line – This is used for data transfer: both transmission and reception of data serially.
- SCL – This is the clock line.
- GNG – This is for connecting to the ground.

The master is the Arduino and the slave is the I2C adapter. The master clocks the slave through SCL. The LCD is assigned by the I2C a unique identifier to be addressed for communication. This identifier is stated in the program code uploaded to the Arduino.

## Keypad



The keypad was used as a user input device. It helps the user to interact with the system. The keypad used in this project has got 4 rows and 3 columns. Each of the keys are connected to a switch beneath them. The pressing of a key is the closure of a switch which is detected by the Arduino. The keypad operates as follows:

The column lines are by default held high when no key is pressed while the row lines are held low. When a key is pressed, the column of the pressed key is pulled low by the row of that key. This is due to the switching action beneath each key on the membrane. Therefore the column is now known. The row is determined by sequentially setting each row high checking whether the column goes high again, on the row at which the known column turns high, that's the row of the pressed key. Knowing the row and the column gives the key pressed.

### **Piezoelectric buzzer**



The piezoelectric buzzer is a loudspeaker activated by supplying a voltage. They operate with a frequency of up to 100 KHz. They can be connected to TTL outputs. The specifications for the buzzer are:

- Buzzer type 0.5 W, 0.8 ohm
- Rated voltage 1.5V.DC.
- Output 85 decibels.
- Resonance frequency 2048 Hz.
- Operating temperature -20 to 45 degrees Celsius.
- Rated current less than or equal to 60mA.

### **R307-Fingerprint scanner**

This module has an optical fingerprint sensor with a high speed DSP(Digital Signal Processor) processor. It has got a FLASH memory for storing fingerprint templates using an algorithm that performs image processing and good image alignment. It offers functions such as fingerprint matching, searching, template storage, deletion and emptying of it's flash memory. It can be interfaced using USB2.0 to a computer or TTL UART to a microcontroller. The module has got the following features:

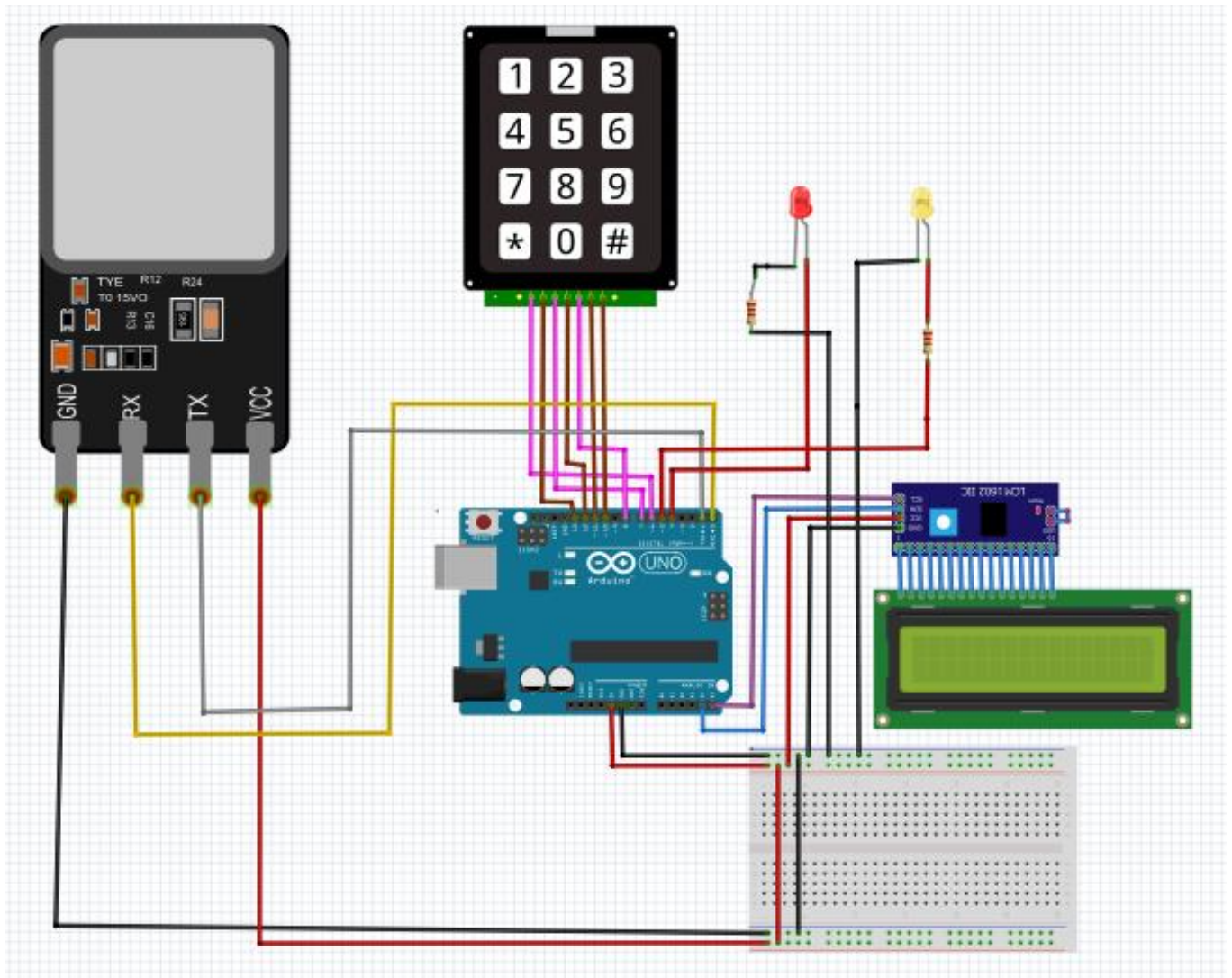
- Host interface USB2.0 or TTL UART.
- Storage capacity 1000 pieces.

- 1 template storage size 512 bytes.
- False Acceptance Rate (FAR) less than 0.001%.
- False Rejection Rate (FRR) less than 1.0%.
- Search method (1: N) - used during searching of prints to obtain the id of a given print or to check its existence in the system.
- Matching method: Comparison method (1:1) - used during enrolment to make a template using two matching prints.
- Search time less than 1 second.
- Working environment temperature: -20 to 40 degrees Celsius.
- Working environment humidity: 40% to 80%.
- Fingerprint image input time less than 0.3 seconds.
- Window area 14 by 18 mm.
- Supply voltage 4.2V to 6V DC.
- Working current 50mA.
- Peak Current 80mA.

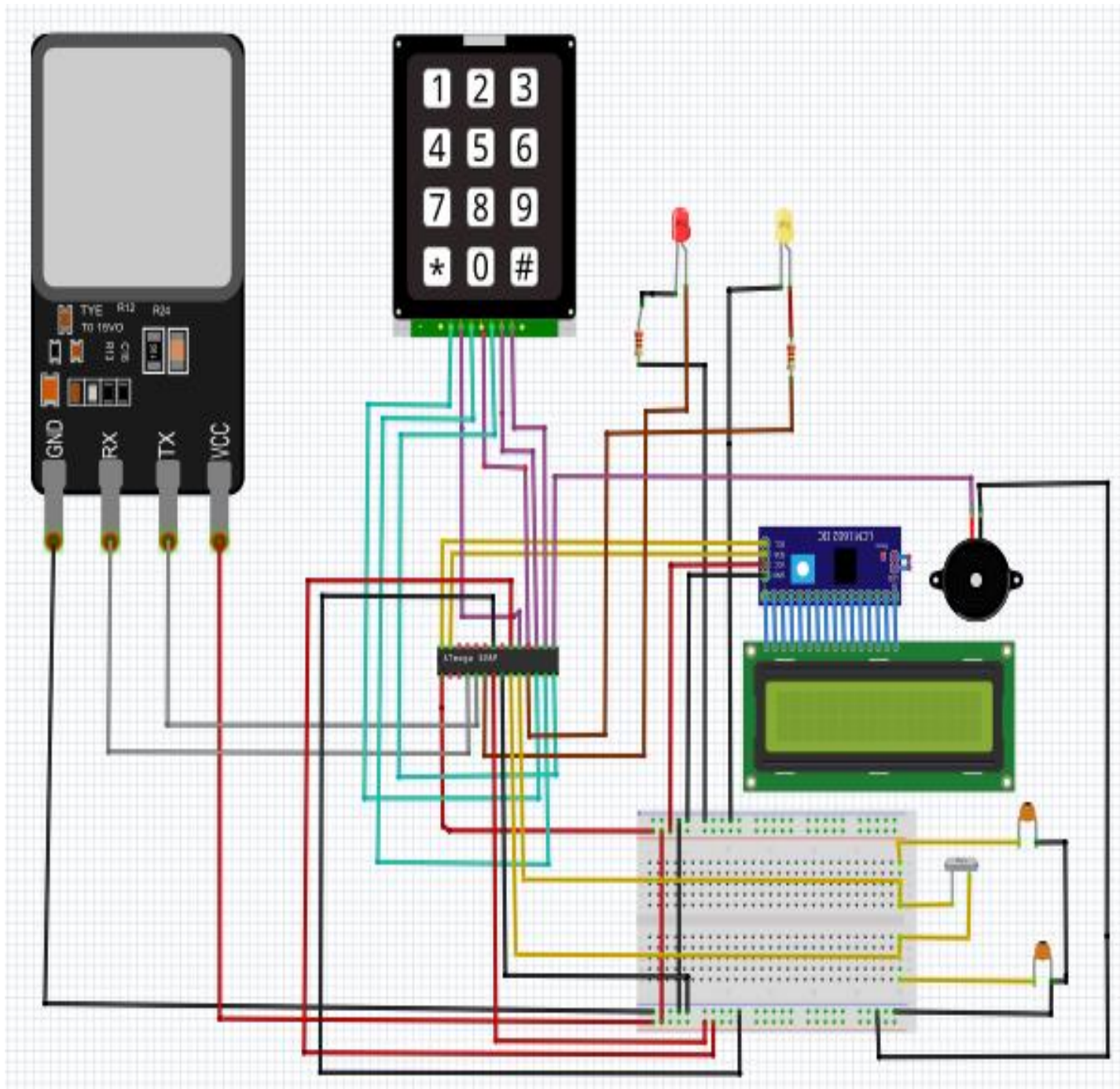


## Design of the hardware outline

The above components were connected as follows during the development phase. They were all tested individually and then connected as follows. This was done using the Arduino board as shown to ensure ease of making changes to the software (the uploaded code) until the final desired results would be obtained. Then the Arduino board would be replaced by the Atmega 328 P microcontroller. In addition, the circuit would be transferred onto a board permanently soldered. This setup was very useful since the success in this hardware configuration would indicate the success of the final hardware configuration since the changes to be made were few.









In this design, the Arduino board is replaced by the Atmega microcontroller. In this design, the final circuitry was obtained. All the components were included and tested successfully. The software design used in this final set up is as detailed in the next section.

## **Software algorithms**

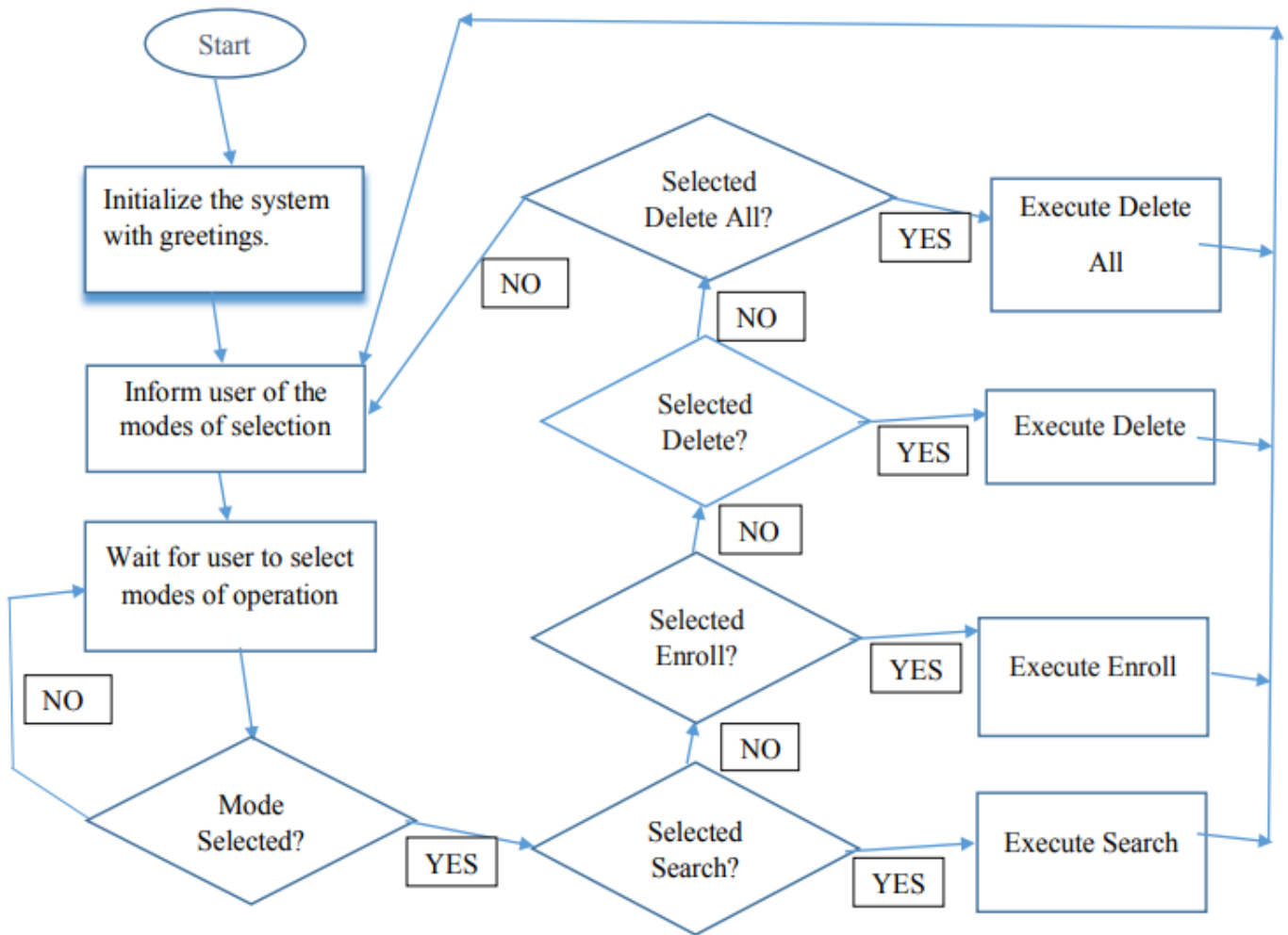
### **Fingerprint scanner software**

#### **Flowcharts**

The fingerprint processing device performs its operations in 4 different modes. These are: enrolment mode, search mode, delete mode and empty database mode. Enrolment mode facilitates adding of examination candidates with a valid candidature into the system. The validity of candidature is determined by the web-based information system which contains extra information about a candidate. Those candidates with a valid candidature are enrolled into the system. The device also operates in delete mode, where any candidate with an invalid candidature is removed from the system by feeding his fingerprint id and deleting his fingerprint from the system. It also operates under search mode where a print is searched whether it exists in the system. Any student whose candidature is valid has his/her fingerprint inside the system. Therefore during search mode, his fingerprints will match and the system alerts the invigilator of valid candidature. The system also operates in empty data mode where all the prints can be deleted from the system. Apart from the search mode, the other modes of operations cannot be operated unless the user is an admin. The following are flowcharts that give a detailed flow of each process and how they relate to one another.

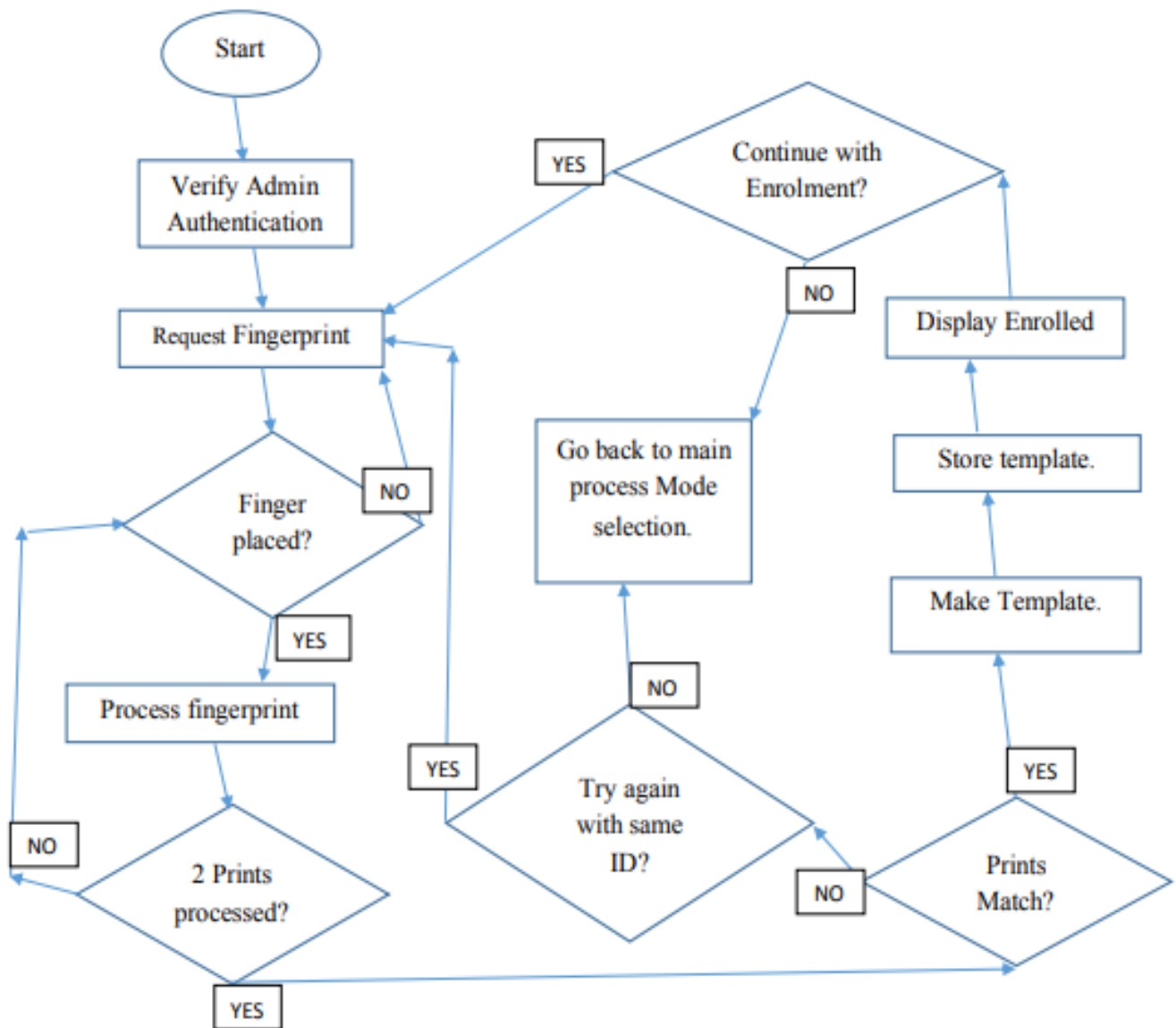
The flowcharts are modularized for clarity purposes. As mentioned earlier, all the three processes apart from search mode, require admin verification. Therefore the admin process is outlined with its own flowchart. Similarly, the four modes of operation have their own flowcharts which are called by the main system flowchart process. This makes the whole system easier to follow.

## MAIN PROCESS



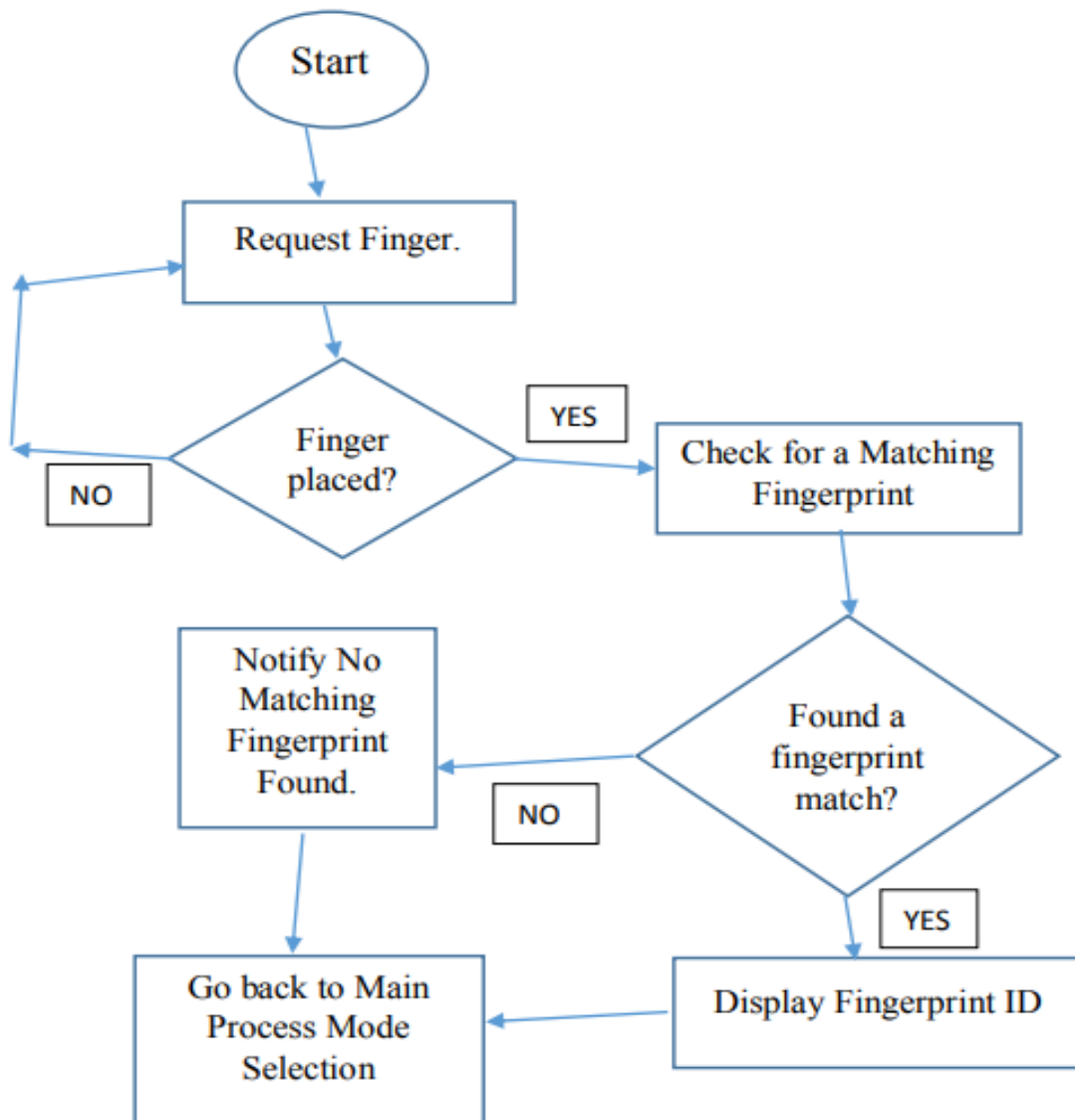
This is the process in which the system is initialized and the user is informed of all the modes of operation of the device. This main process calls the other processes through functions depending on what the user of the system selects. Upon completion of other processes or modes, the system returns to this default process.

## ENROLL PROCESS

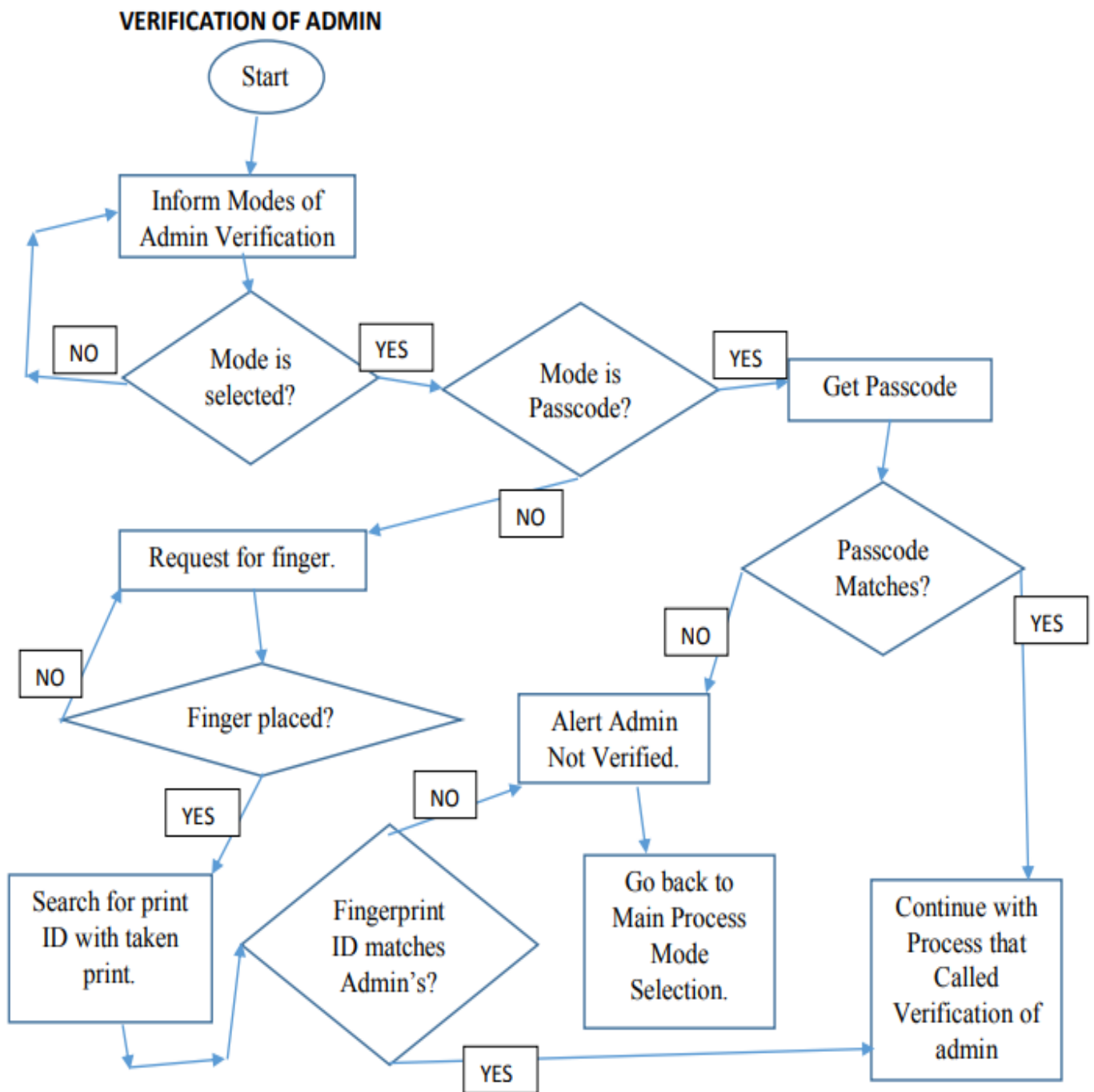


In this process, the user is allowed to enrol new examination candidates into the system. This process begins with admin verification. The user is then requested to key in the id of the candidate he/she wants to enrol. Upon submission of the id, the user's prints are taken and they are checked whether they already exist in the system or not. If they exist, the process goes to start and requests the user to try enrolling again with the same id. Once they are successfully accepted, their prints are enrolled and the system returns their ID.

## SEARCH PROCESS



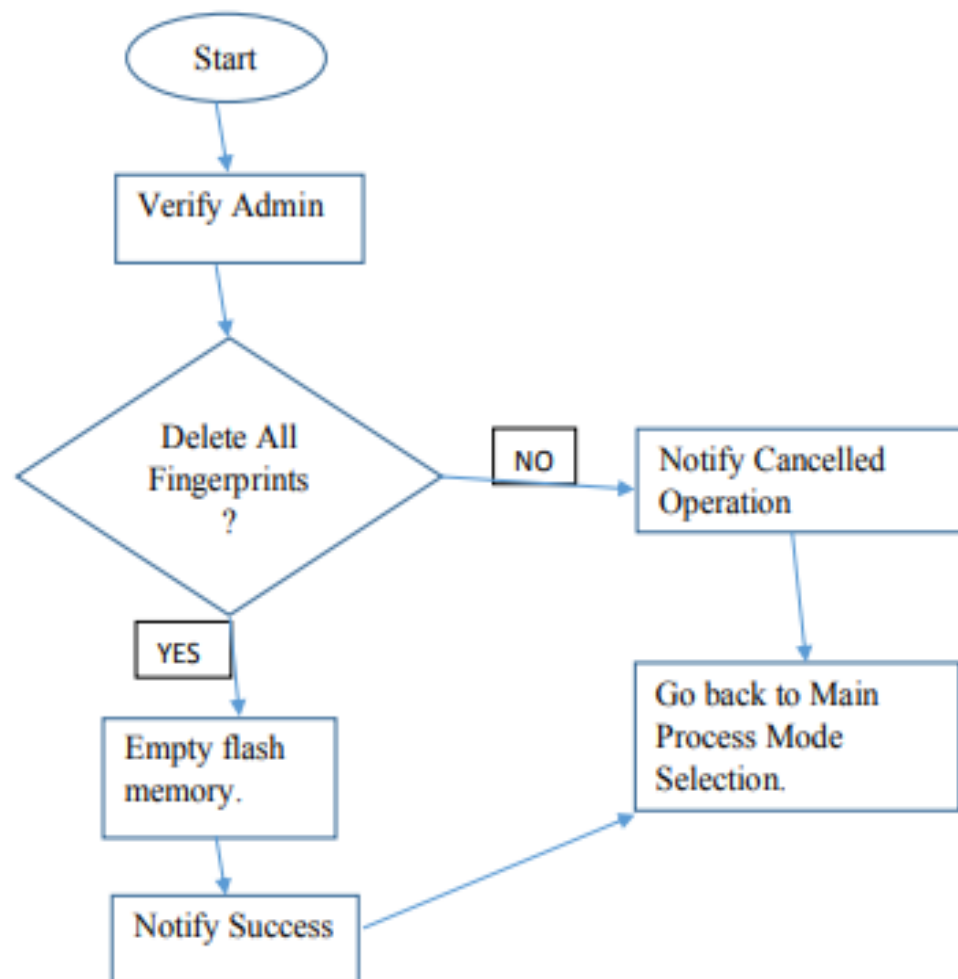
In the search process, the user is requested to place finger. Once it's scanned, the systems checks of a matching print. If the there is a match, the system returns the id of the match. If there is no match, the system goes to the main process where the user is presented with modes of operation of the device. The system waits for user to select mode.



In this verification of admin process, the user is verified to be an admin or not. The verification process is done using fingerprints or using a passcode. The passcode is known by the super admin only. The fingerprints for the admins have reserved ids. If the searched print returns an id

similar to the reserved ids, then the user is allowed to continue with the other processes. If the authentication fails, the buzzer alarms and the system goes back to the main process where the user is requested to choose the mode of operation.

### **DELETE ALL PRINTS**

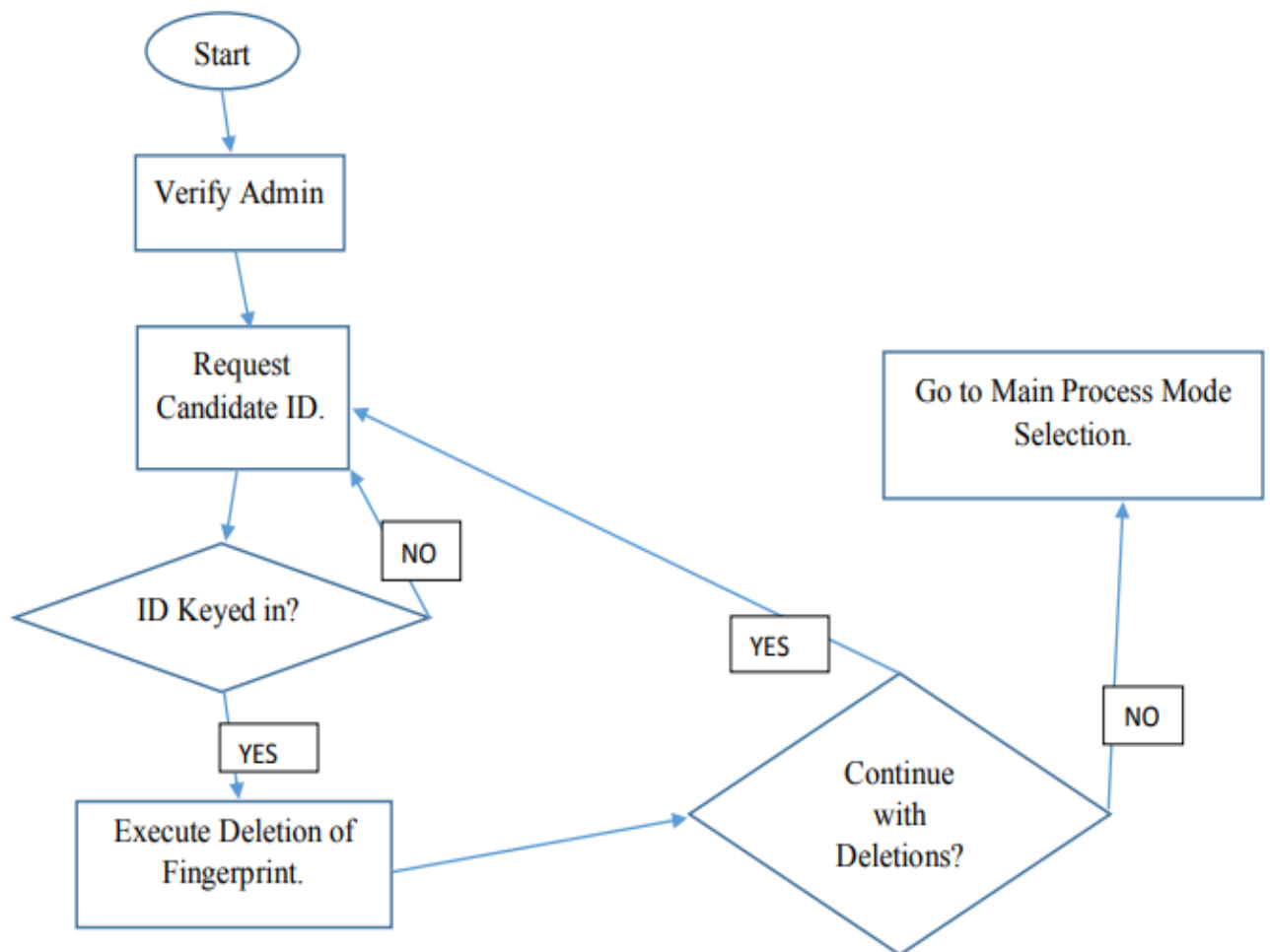


In the deletion-of-all-prints process, the user is verified to be the admin. This is a precaution and a security enhancement measure to prevent the disastrous deletion of all the fingerprints by an

unauthorized user. On top of that, once the admin is verified, there is a confirmation of the process. Once confirmed, the deletion of all prints is executed and the system goes back to the main process where the user is presented with modes of operation of the scanner. In case, the process was cancelled, the system still goes back to the fore-mentioned state.

---

### DELETION PROCESS



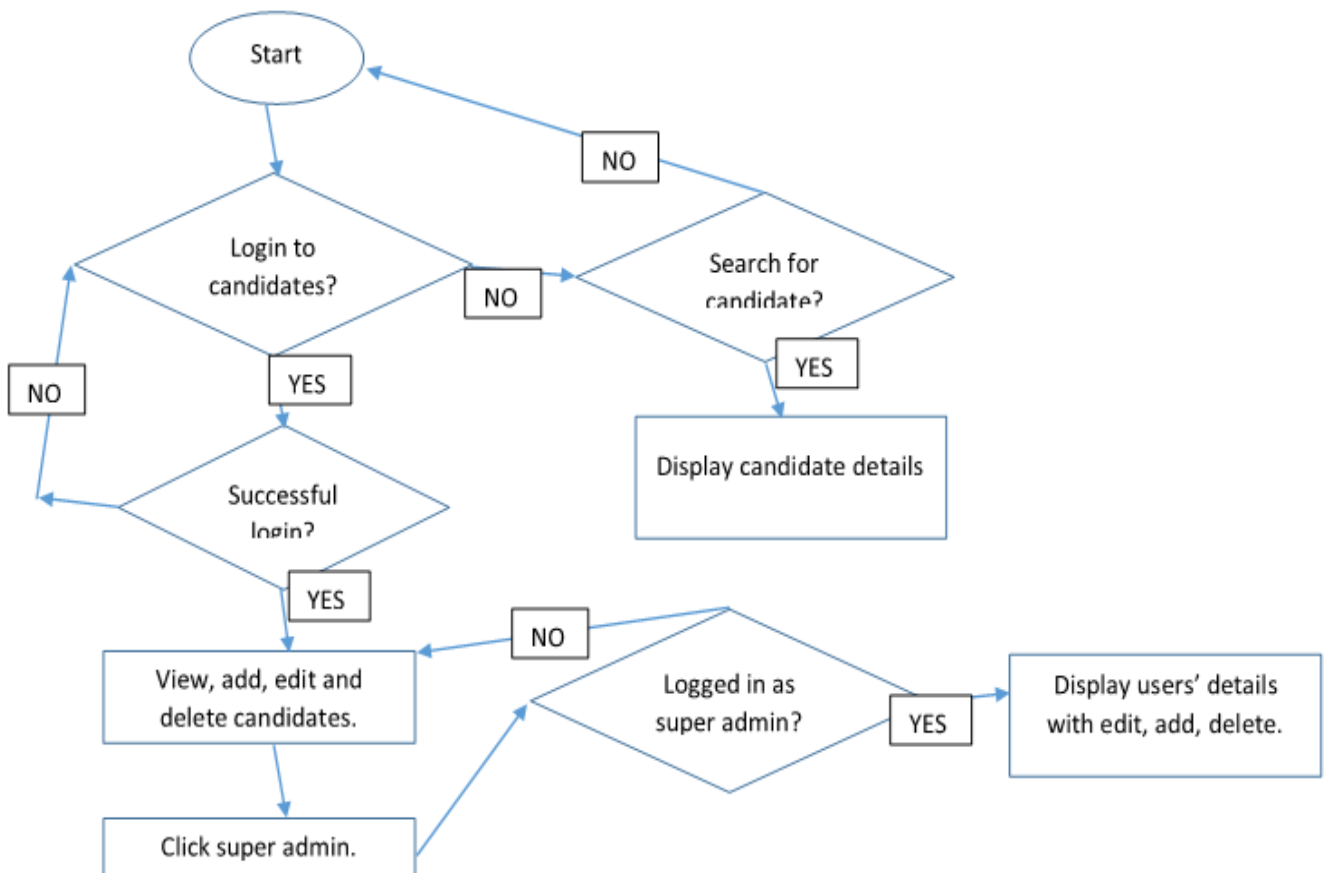
Deletion of a user must be done by an admin, therefore the system first verifies whether the user is an admin when this mode is selected. Once the admin is verified, the user is requested to input an id of the fingerprint he/she wants to delete from the

system. Once provided, the system deletes the print and notifies the user of successful deletion. The system then requests the user to continue with deletion mode or not. If he/she chooses to continue, the system requests for an id to delete as before. If chooses not to continue, the system goes to the main process where the user is requested to select the mode of operation.

## Web-based complimentary system software

### Flowchart

The fingerprint based system is complimented by a web-based information system to store the extra information for the examination candidates. The web-based software allows the searching of the details of a given fingerprint id obtained from the fingerprint processing device. The following are the flowcharts of the process flow in the web-based information system.



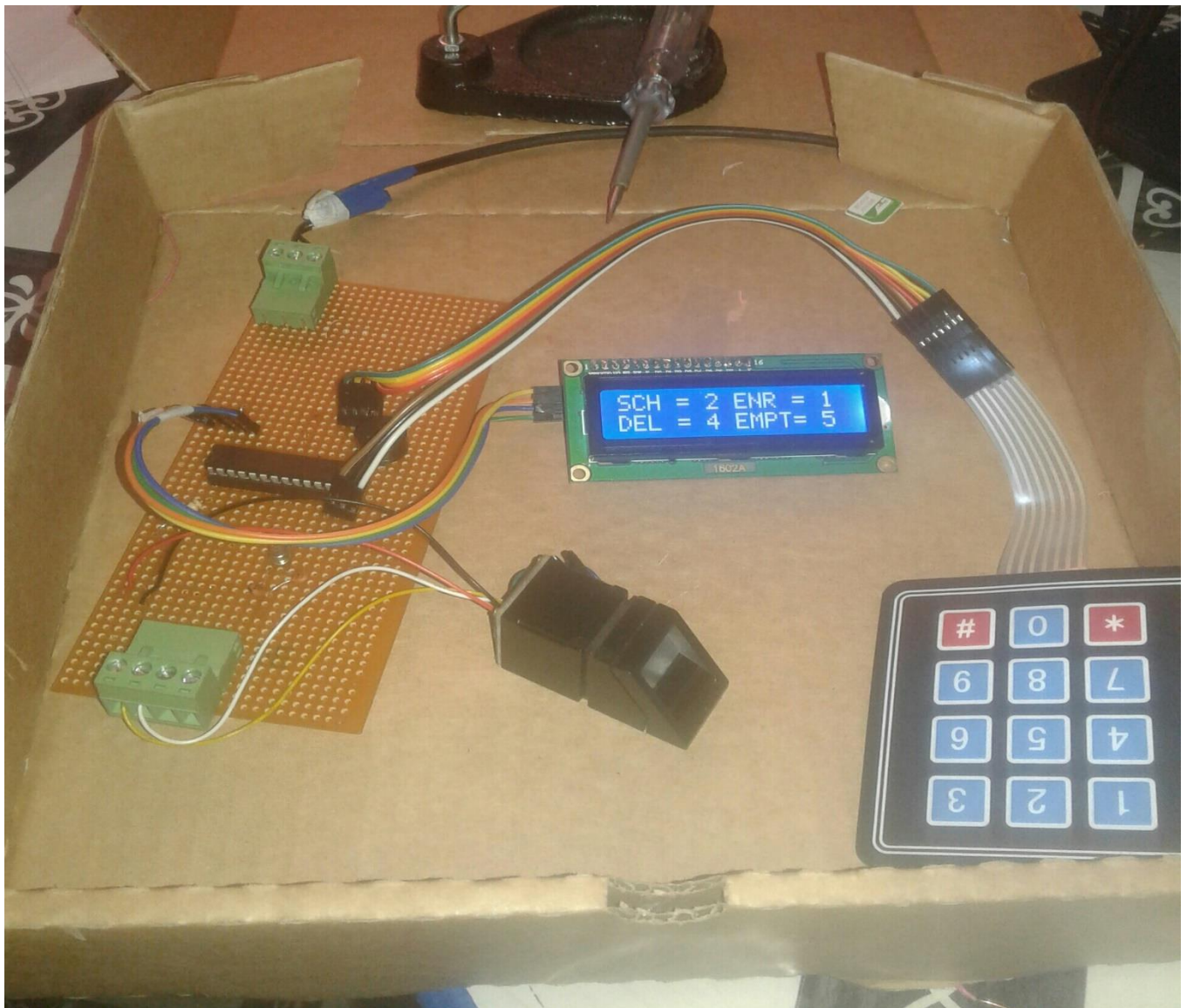


## CHAPTER 4

### RESULTS AND DISCUSSION

#### Hardware Device

This is the device with all the components in use and connected together. The components have been soldered together ready for a casing. The circuit was laid on a verro-board since there were more peripheral components more than the supporting circuitry. The supporting circuitry is basically two capacitors, a crystal oscillator and a resistor. It was therefore not economical to use a printed circuit board for prototyping.



This is the final device powered by USB2.0 cable which can be attached to a standard power bank, a standard 5V 1.2A phone charger adapter or a computer USB ports. This power source was chosen since the device is used indoors where these fore mentioned power sources are available.

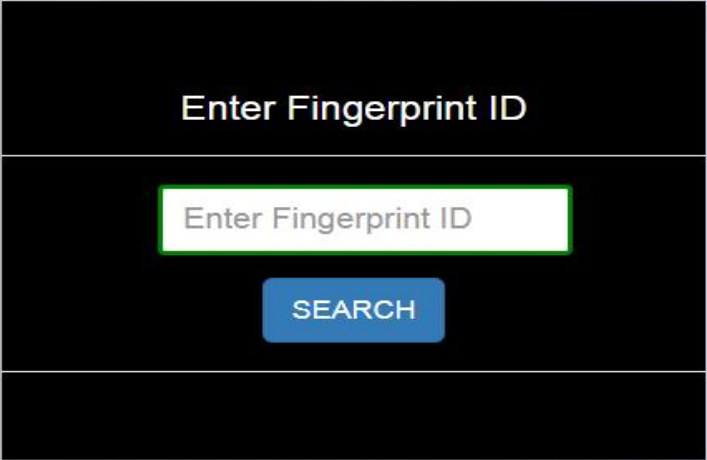


## Software Device

*The software is hosted on an online server, it can be accessed using the following link:*

<https://avowable-reserves.000webhostapp.com/>

This page for searching candidate details using a fingerprint id. This page also gives a button to open the login form for the admin to access the student-detail area and also the super admin.





Enter Fingerprint ID

Enter Fingerprint ID

SEARCH

This is the searched candidate. This modal displays the details of the candidate including the passport photo to confirm their identity.





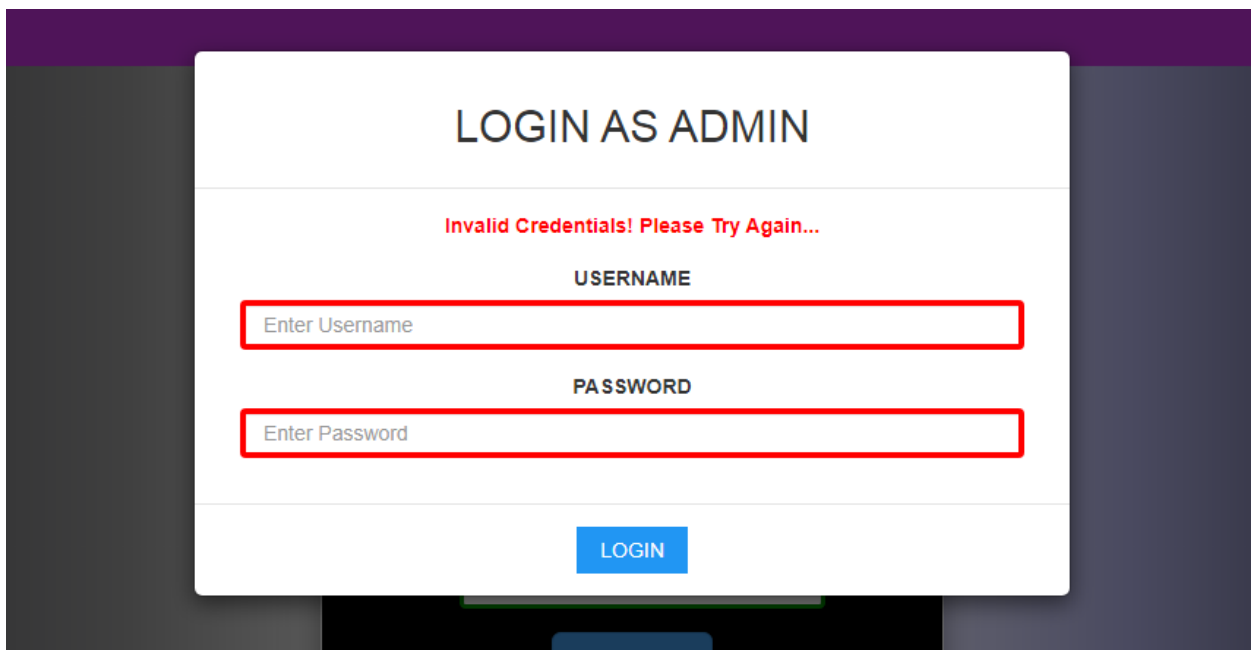
**Juma Zioka**  
F1/1635/2015

**FingerPrint : 1**

Electrical and Information Engineering, 5th Year

**Candidature : Valid**

This is the login form for admins to view all candidates and to edit, delete or add new candidates. This form allows only the admins with password and username. If the user does not have correct credentials, the system will invoke him/her to use the correct credentials. This login form appears also as modal to reduce loading of static html pages.



**LOGIN AS ADMIN**

Invalid Credentials! Please Try Again...

**USERNAME**

Enter Username

**PASSWORD**

Enter Password

**LOGIN**

The following interface allows viewing, editing, adding and deleting of examination candidates. The examination candidature validity is also edited in this interface. This interface is accessible by the admins only.

SUPER\_ADMIN LOGOUT

## STUDENTS

+ Add Person
Reload

Show 10 entries
Search:

FingerPrint ID	Registration No.	Department	Year of Study	First Name	Last Name	Gender	Address	Date of Birth	Exam Candidature	Photo	Action
1	F1/1635/2015	Electrical and Information Engineering	5th Year	Juma	Zioka	male	Nairobi Ngar...	2019-12-03	Valid		
2	F18/1205/201	Mechatronic Engineering	3rd Year	Joseph	Kamau	male	Nairobi Ngar...	2019-12-05	Invalid		
3	F17/1685/201	Arts and Design	4th Year	Francis	Kiiru	male	Nakuru, Lare...	2019-12-03	Valid		
FingerPrint ID	Registration No.	Department	Year of Study	First Name	Last Name	Gender	Address	Date of Birth	Exam Candidature	Photo	Action

The following modal allows editing of a student/candidate.

STUDENTS

+ Add Person
Reload

Show 10 entries

FingerPrint ID	Registration No.	Dep
1	F1/1635/2015	Eled Infot Engi
2	F18/1205/201	Mec Engi
3	F17/1685/201	Arts
FingerPrint ID	Registration No.	Dep

Edit Student

FingerPrint ID1

Registration No.F1/1635/2015

Department.Electrical and Information Engineering

Year of Study.5th Year

First NameJuma

Last NameZioka

GenderMale

AddressNairobi Ngara Plaza  
Maclavin House 3

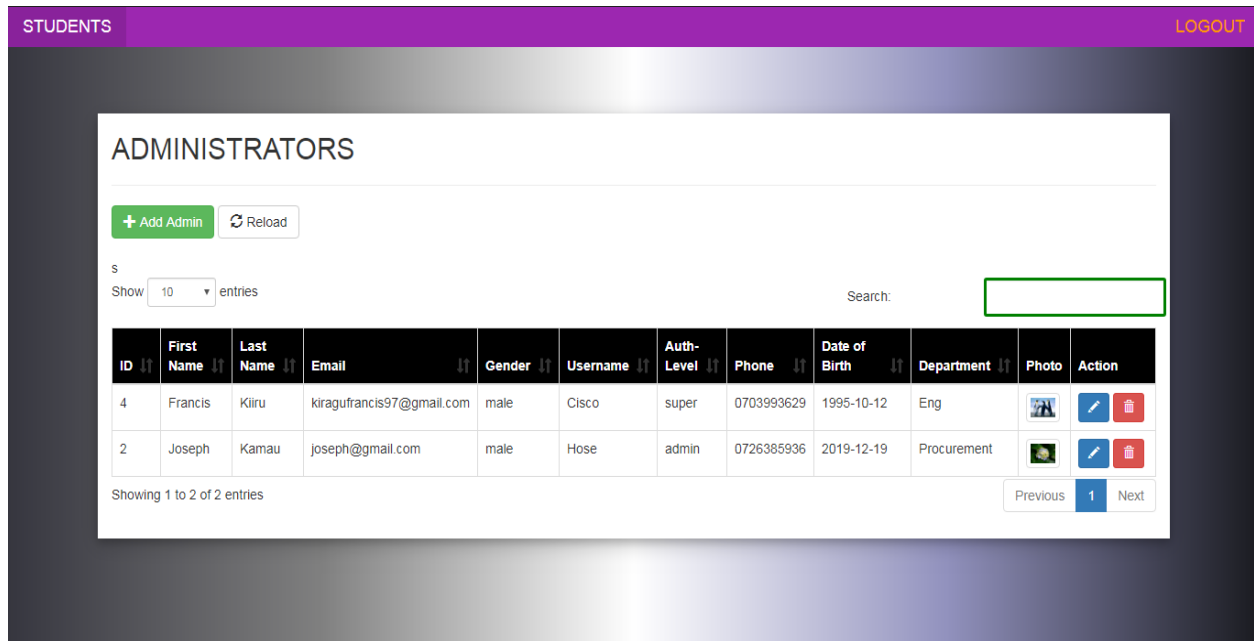
Date of Birth2019-12-03

am candidature

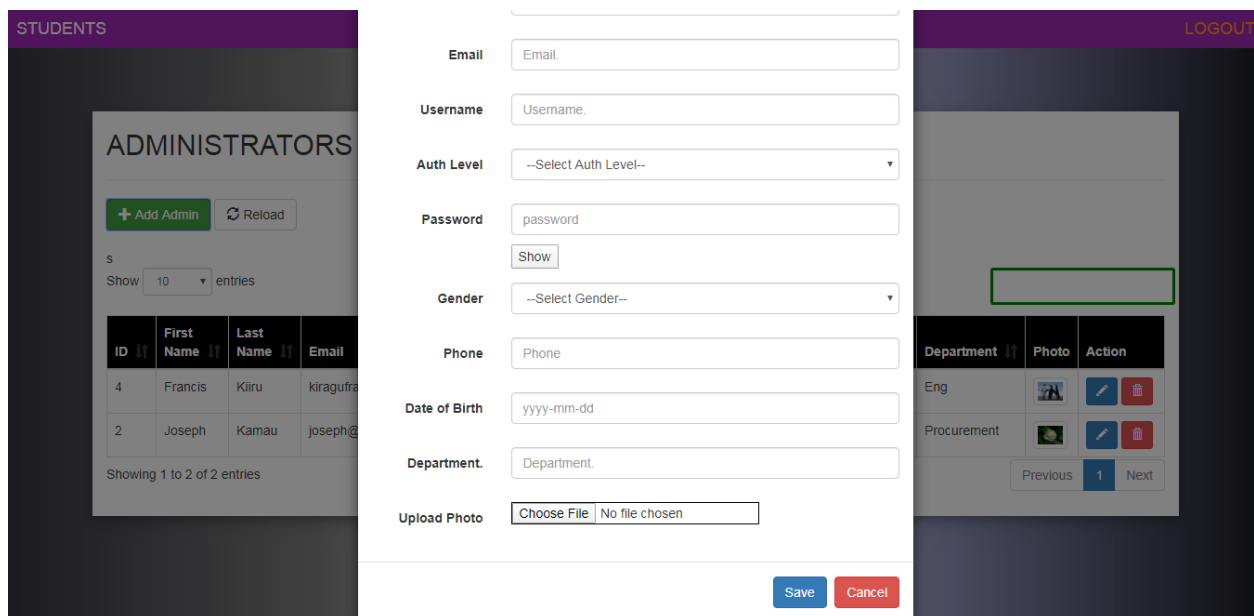
Photo

Action

This is the interface for adding administrators to the system. This interface is only accessible by the super-admin. The super admin can add, delete, edit and view the administrators in this interface. The super admin can also remove the print of another admin using the id obtained from this interface.



This interface allows you to add an admin.



## **Process flow**

The system follows the following steps:

The fingerprint device is initialized and prompt the user to choose the operation mode.

### **Enrollment**

If the user chooses enrolment mode, the system guides the user to the whole enrolment process until it's successfully completed. This process is only done by authorized persons only. The precaution is undertaken using a passcode or using fingerprint matching for admins. Upon completion, the system displays the fingerprint id of the enrolled candidate. The admin then notes this id. The admin registers the student into the online web application using the noted fingerprint id and uploads a passport photo of the candidate. This completes the enrolment of a candidate.

### **Searching**

If the user selects searching mode, the device directs the candidate to place the finger on the scanner. Once scanned, the system searches the print and if there is no match, the system alerts the user with a red LED and the LCD. If there is a match, the blue LED is turned on and the LCD displays the id of the candidate. The user notes the candidate fingerprint id. The user opens the web application through a browser link and uses the fingerprint id to search for the details of the candidate. The system returns the details including the passport photo of the student. This shows the candidature of the student if it is valid or not. This confirms the correctness of the fingerprint device.

### **Deleting**

In this process, an administrator removes all the students with invalid examination candidature from the system. The process is protected and it's only done by the authorized persons only. This precaution is undertaken using a passcode or admin fingerprint matching. The admin first access the candidate details from the online web application. He searches for all candidates with an



invalid examination candidature. Using their fingerprint ids, he/she deletes the students from the fingerprint device. The process repeats until all students with invalid candidature are removed from the system.

### **Empty Flash Memory**

In this process, the flash memory is emptied. The precaution is taken to prevent anyone from doing this through admin passcode or admin fingerprint matching. The system can be emptied when there are enrolled prints in the system without knowledge of their ids. This would prevent fraudsters to exist in the system. After this process, the enrolment process must be done to add the valid candidates into the system.

The administrators are enrolled into the system using reserved fingerprint ids. The super-admin accesses the system using the passcode and enrolls them. If a search returns the fingerprint id amongst the reserved ids, the user is allowed access to deletion, emptying flash memory and enrollment.

## **CHAPTER 5**

### **RECOMMENDATIONS**

The fingerprint scanner used in this project can store at most 1000 pieces of prints. This limits the scalability of the project if the examination candidates are more than that. This device also has a high false rejection rate (FRR) that goes up to 1%. This is noted during searching, sometimes enrolled prints do not match with searched prints of the same finger. This can be attributed to the technology used in the scanner, where it uses photo to take prints.

I recommend use of capacitive scanners which are more accurate than this scanner and can store millions of templates. These capacitive scanners are also designed to be used with computer database to store the templates as strings. This makes them to have no limitation of the number of prints they can store. This gives an added advantage of using the computer pc instead of microcontroller as the main control device. This recommendation however comes with a higher



cost. The other advantage is that the system would be more versatile since changes are done on the software alone, making the system to be more scalable.

However, if the above fore mentioned limitations of the scanner would be solved, then the system would be good as it is using an improved R307 fingerprint scanner.

This device can also be used for other applications where a party needs to have their identity established before they obtain a service. For example in library checkout systems.

## **CHAPTER 6**

### **CONCLUSION**

Examinations are used as the major method of testing learners in the academic institutions. The credibility of examinations is therefore very critical to every learning institution as it defines its reputation and quality of services they offer and the credibility of its certifications. This calls for examinations which are free of malpractices. This system makes that possible by authenticating the candidates in an exam hall. This helps to get rid of impostors who are hired to sit for other students' exams. It also helps to only allow only those students who have a valid candidature to sit for the examinations. Using fingerprints to establish the identity of students is an efficient way of establishing if the student is indeed who he/she says he/she is. In this way, all the information about a student is obtained upon establishment of their identity. If their identity is not established through the fingerprint matching device, then they are deemed to be illegally in an examination. In this way, examination malpractices will be hindered. The learning institutions will establish credibility in their examinations and establish a good reputation for themselves.

## REFERENCES

- Woojung Kim, Woojin Hong, Taekmoo Kim, Dongwoon Kim, Myunghee Lee, "RF Sensor-Based Liveness Detection Scheme With Loop Stability Compensation Circuit for a Capacitive Fingerprint System", Access IEEE, vol. 7, pp. 152545-152551, 2019.
- Takahiro Hashizume, Takuya Arizono, Koji Yatani, "Auth 'n' Scan", Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, pp. 1, 2018.
- Meng-Lieh Sheu, Lin-Jie Tsao, "A sub-fF capacitive fingerprint sensor with neighbor pixel difference sensing", Next-Generation Electronics (ISNE) 2016 5th International Symposium on, pp. 1-2, 2016.
- Qiu, L. (2020). Fingerprint sensor technology - IEEE Conference Publication. Retrieved 19 January 2020, from <https://ieeexplore.ieee.org/document/6931393/references#references>
- Agarwal, T. (2020). LCD - What is LCD: Construction and Working Principles of LCD Display. Retrieved 19 January 2020, from <https://www.elprocus.com/ever-wondered-lcd-works/>
- ATMega328P Microcontroller Pinout, Pin Configuration, Features & Datasheet. (2020). Retrieved 19 January 2020, from <https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet>
- Basics, C. (2020). Basics of the SPI Communication Protocol. Retrieved 19 January 2020, from <http://www.circuitbasics.com/basics-of-the-spi-communication-protocol/>
- How fingerprint scanners work: optical, capacitive, and ultrasonic variants explained. (2020). Retrieved 19 January 2020, from <https://www.androidauthority.com/how-fingerprint-scanners-work-670934/>
- Mainguet, J. (2020). Biometrics: fingerprint. Retrieved 19 January 2020, from <http://fingerchip.pagesperso-orange.fr/biometrics/types/fingerprint.htm>
- McFee, C. (2020). An introduction to CCD operation. Retrieved 19 January 2020, from [http://www.mssl.ucl.ac.uk/www\\_detector/ccdgroup/opttheory/ccdoperation.html](http://www.mssl.ucl.ac.uk/www_detector/ccdgroup/opttheory/ccdoperation.html)

百度百科——全球最大中文百科全书. (2020). Retrieved 19 January 2020, from  
<http://baike.baidu.com/link?url=JmZ74MO33ITGgC8yMk20hpm44Rm37mOZGVSNuQduXgHBLyIZSHkXcrsMqQ5D9tOjhz9ZFwJtyUrxzMI8Qspuu>

## APPENDICES

### Appendix 1: Bill of Quantities

Item	Cost in Ksh.
Atmel 328P	400
LCD	400
I2C Module	200
R307 fingerprint scanner	2,800
Keypad membrane	100
Buzzer	20
Verro Board	100
Connecting cables	100
2 LEDs	10
Power-USB Cable	100
Connectors	150
<b>Total</b>	<b>4,380</b>

### Appendix 2.0: Device Code

The online information system code is found in the following github public account:

Clone URL: <https://github.com/Cisco/FingerprintDeviceComplementor.git>

### Appendix 2.1: Importing Libraries and declaring variables

```
#include <Adafruit_Fingerprint.h>
```

```
#include <Wire.h> //import library for I2C
```

```

#include <String.h>//Import library for String types
#include <LiquidCrystal_I2C.h>//import library for lcd module
#include <Keypad.h>//import keypad membrane library
#define mySerial Serial1

SoftwareSerial mySerial(2, 3);//converts pin 2 and 3 to rx and tx for spi communication
protocol

using namespace std;//namespace for string functions and types

const int en = 2, rw = 1, rs = 0, d4 = 4, d5 = 5, d6 = 6, d7 = 7, bl = 3;//initializes the pins used
by the I2C on the LCD

const int i2c_addr = 0x27;//the I2C address initialization
LiquidCrystal_I2C lcd(i2c_addr, en, rw, rs, d4, d5, d6, d7, bl, POSITIVE);

//keypad initialization

const byte ROWS = 4; //four rows
const byte COLS = 3; //three columns
char keys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}
};

bool adminmode = true;//variable for setting the admin mode true or false

byte rowPins[ROWS] = {13, 12, 11, 10}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {6, 7, 8}; //connect to the column pinouts of the keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);//the variables from software
serial are passed to fingerprint library object

//global variables init

int password;//variable for holding admin passcode

int id;//the id of searched fingerprint

```

```

int admin;// THE ADMIN ID

char keyPad = '3';//holds default char for keyPad

int mode = 0;//

bool searched = false;

int buzz = 9;//buzzer makes noise when unauthorized personnel tries to access admin
operations

int alert = 4;//an led to alert as the buzzer

int lock = 5;//an alert to show access granted

bool lockmode = false;//only true if the program is at search mode

```

## Appendix 2.2: Setup function

```

void setup() //runs once

{
    Serial.begin(9600);
    delay(100);
    pinMode(buzz, OUTPUT);//setting buzzer as output
    pinMode(alert, OUTPUT);//red led for denied/not found fingerprint
    pinMode(lock, OUTPUT);//led to show searched print found
    finger.begin(57600);//setting baud rate for scanner
    delay(5);
    init_system();//initialisation informing modes of operation
}

void loop()

{
    processing();//the main function that calls other functions
}

```

## Appendix 2.3:Initialization/greetings

```

//*****INITIALIZATION*****

void init_system(){

```

```

    lcd.begin(16,2);//initialize lcd object

    lcd.setCursor(0,0);

    if (finger.verifyPassword()) { //checks for successful communication of scanner and
microcontroller

        lcd.print("SCANNER FOUND!");
    } else {

        lcd.print("SCANNER NOT FOUND!");
        while (1) { delay(1); }

    }

    delay(1000);

    lcd.clear();

    lcd.setCursor(1,0);

    lcd.print("-WELCOME TO THE-");

    lcd.setCursor(0,1);

    lcd.print("EXAMINATION HALL");

    delay(2000);

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("ENROLL = 1");

    lcd.setCursor(0, 1);

    lcd.print("DELETE = 4");

    delay(1000);

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("SEARCH = 2 ");

    lcd.setCursor(0, 1);

    lcd.print("EMPTY DB = 5");

    delay(1000);

}

```

## Appendix 2.4: The main function

```
//***** The main custom function*****

void processing(){

    enquireMode();//Enquires/informs what mode of operation the user wants: search, enroll,
delete or empty database

    keyPad = keypad.getKey();

    while(!keyPad || keyPad == '3')//loops until key is pressed

    {

        keyPad = keypad.getKey();

    }

    if(keyPad == '1'){//if 1 was keyyed

        admin = 0;//initializes the admin variable with id 0

        authenticate_admin();//calls function to verify admin before enrollment; it returns
admin fingerprint id

        if(admin == 20 || admin == 30 || admin == 40 || admin == 50 || admin == 60 || admin ==
70 || admin == 80 || password == 1995)

        {

            lcd.clear();

            lcd.setCursor(0, 0);

            lcd.print("-ENROLLMENT MODE-");

            delay(2000);

            searched = false;

            bool cont = true;//keeps the mode as enrollment as long as it's true

            while(cont)

            {

                lcd.clear();

                lcd.setCursor(0, 0);

                lcd.print("-TYPE THE ID-");

                lcd.setCursor(0, 1);
```

```

    lcd.print("PRESS # TO ENTER");
    id = GetNumber();//id to be enrolled
    getFingerprintEnroll();//calls enrolment function
    keyPad = keypad.getKey();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("To Continue");
    lcd.setCursor(0,1);
    lcd.print("Press #");
    keyPad = keypad.getKey();//resets the variable
    while(!keyPad)//waits for keypress to continue with enrol or end
    {
        keyPad = keypad.getKey();
    }
    if(keyPad == '#')
    {
        cont = true;
    }
    else
    {
        cont = false;
    }
}
}

else{//when admin is not verified
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("-NOT AUTHORIZED !-");

```



```

    tone(buzz, 1000); //buzzer makes noise
    digitalWrite(alert, HIGH); //red LED alerts
    delay(2000);
    noTone(buzz);
    digitalWrite(alert, LOW);
}
}

if(keyPad == '2'){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("-SEARCH MODE-");
    delay(1000);

    lockmode = true; //ensures that the blue LED is turned on with successful found id for
search mode only

    adminmode = false; //ensures the admin verification codes are not executed inside the
search function

    int candidate = getFingerprintID(); //calls search function
    lockmode = false;
    adminmode = true;
    keyPad = keypad.getKey(); //resets keyPad Variable
}

if(keyPad == '4')
{
    admin = 0;
    authenticate_admin(); //calls function to verify admin

    if(admin == 20 || admin == 30 || admin == 40 || admin == 50 || admin == 60 || admin ==
70 || admin == 80 || password == 1995)

    { //if the admin was verified

        bool cont = true;

```

```

while(cont)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("-DELETE ONE -");
    delay(1000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("-ENTER ID -");
    delay(1000);
    int a = GetNumber();//stores id of the id to delete
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Deleting ID ");
    lcd.setCursor(0, 1);
    lcd.print(a);
    deleteFingerprint(a);//calls function to delete a fingerprint with id 'a'
    delay(1000);
    keyPad = keypad.getKey();//resets the keypad variable to empty/unset value
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("To Continue");
    lcd.setCursor(0,1);
    lcd.print("Press #");
    keyPad = keypad.getKey();
    while(!keyPad)//waits for keypress until the user selects the mode
    {
        keyPad = keypad.getKey();
    }
}

```

```

    }
    if(keyPad == '#')//if true the mode remains delete
    {
        cont = true;
    }
    else
    {
        cont = false;
    }
}
}

else{// alerts user is not admin

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("-NOT AUTHORIZED !-");
    tone(buzz, 1000);
    digitalWrite(alert, HIGH);
    delay(2000);
    noTone(buzz);
    digitalWrite(alert, LOW);
}
}

if(keyPad == '5')
{
    admin = 0;

    authenticate_admin();//calls to verify admin

    if(admin == 20 || admin == 30 || admin == 40 || admin == 50 || admin == 60 || admin ==
70 || admin == 80 || password == 1995)
    {

```

```

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("-DELETE ALL -");
        delay(1000);
        emptydb();
        keyPad = keypad.getKey();//resetting the keyPad variable
    }
    else{
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("-NOT AUTHORIZED !-");
        tone(buzz, 1000);
        digitalWrite(alert, HIGH);
        delay(2000);
        noTone(buzz);
        digitalWrite(alert, LOW);
    }
}
else
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("...");
    keyPad = keypad.getKey();
}
}

```

## **Appendix 2.5: Authentication of Admin**

//\*\*\*\*\*Authentication of Admin\*\*\*\*\*

```

void authenticate_admin();//this function verifies admin
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("USE-CODE = *");
    lcd.setCursor(0, 1);
    lcd.print("USE-PRINT= 0");
    while(!keyPad || keyPad == '1' || keyPad == '4' || keyPad == '5')
    {
        //waits for user to select mode of admin verification
        keyPad = keypad.getKey();
    }
    if(keyPad == '*')
    {
        //if chose passcode
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("-TYPE THE CODE-");
        lcd.setCursor(0, 1);
        lcd.print("PRESS # TO ENTER");
        password = GetNumber();
    }
    else
    {
        admin = getFingerprintID();
    }
}

```

## Appendix 2.6: Informing modes of operation

```

//*****Choosing mode of operation*****

void enquireMode()

```

```

{ //informs of modes of operation

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("SCH = 2 ");

    lcd.setCursor(8, 0);

    lcd.print("ENR = 1 ");

    lcd.setCursor(0, 1);

    lcd.print("DEL = 4");

    lcd.setCursor(8, 1);

    lcd.print("EMPT= 5");

} //*****end of choosing mode*****

```

## Appendix 2.7 Reading keypad

```

//*****reading keypad data*****

int GetNumber()

{

    int numba = 0;

    char pressedkey = keypad.getKey();//resets key variable

    while(pressedkey != '#')//wait for keypad data until # is pressed to end

    {

        switch (pressedkey)

        {

            case NO_KEY:

                break;

            case '0': case '1': case '2': case '3': case '4':

            case '5': case '6': case '7': case '8': case '9':

                lcd.clear();

                lcd.setCursor(0, 1);

```

```

        lcd.print(pressedkey);//prints pressed key

        //ASCII for 0 is 48 and increments by 1 from 1 hence we must deduct 48 then add to
the following place value

        numba = numba * 10 + (pressedkey - '0');

        break;
    case '*':

        numba = 0;

        lcd.clear();

        break;

    }

    //h++;

    pressedkey = keypad.getKey();

}

return numba;

}

```

## Appendix 2.8 Emptying flash memory

//\*\*\*\*\*EMPTYING Flash memory\*\*\*\*\*

```

void emptydb()
{
    lcd.clear();

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Confirm Delete");

    lcd.setCursor(0, 1);

    lcd.print("All: YES=9, NO=8");

    while(keyPad == '0' || keyPad == '*' || !keyPad)//loops until keypad value changes
    {

        keyPad = keypad.getKey();
    }
}

```

```

    }

    if(keyPad == '9')//if confirmed delete all prints
    {
        finger.emptyDatabase();
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Emptied");
        lcd.setCursor(0, 1);
        lcd.print("Successfully!");
        keyPad = '3';
        delay(1000);
    }

    if(keyPad == '8')
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Cancelled!");
        delay(1000);
        keyPad = '3';
    }

}

```

## Appendix 2.9: Deleting a template

//\*\*\*\*\*DELETE A PRINT\*\*\*\*\*

```

uint8_t deleteFingerprint(uint8_t id) { //function to delete a given print
    uint8_t p = -1;
    p = finger.deleteModel(id);
    if (p == FINGERPRINT_OK) {
        lcd.clear();
    }
}

```



```

    lcd.setCursor(0,0);
    lcd.print("Deleted!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Communication error");
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Could not delete");
    lcd.setCursor(0,1);
    lcd.print("in that location");
    return p;
} else if (p == FINGERPRINT_FLASHERR) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Error writing");
    return p;
} else {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Unknown error: 0x");
    lcd.setCursor(0,1);
    lcd.print(p, HEX);
    return p;
}
}

```

## Appendix 2.10: Enrollment

```

//*****Enrollment of new prints*****

uint8_t getFingerprintEnroll() {
    bool mismatch = true;
    while(mismatch){
        //loops until there is no mismatch of prints of same finger; allows enrolling with same id
        until successful

        int p = -1;
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Place finger...");
        delay(1000);
        while(p != FINGERPRINT_OK) {
            p = finger.getImage();
            switch (p) {
                case FINGERPRINT_OK:
                    lcd.clear();
                    lcd.setCursor(0,0);
                    lcd.print("Image taken!");
                    break;
                case FINGERPRINT_NOFINGER:
                    break;
                case FINGERPRINT_PACKETRECEIVEERR:
                    lcd.clear();
                    lcd.setCursor(0,0);
                    lcd.print("Communication Error.");
                    break;
                case FINGERPRINT_IMAGEFAIL:
                    lcd.clear();

```

```

lcd.setCursor(0,0);

    lcd.print("Imaging error.");
    break;
default:
    lcd.clear();
lcd.setCursor(0,0);
    lcd.print("Unknown error!");
    break;
}
delay(1000);
}

p = finger.image2Tz(1);
switch (p) {
    case FINGERPRINT_OK:
        lcd.clear();
lcd.setCursor(0,0);
        lcd.print("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        lcd.clear();
lcd.setCursor(0,0);
        lcd.print("Image too messy");
        return p;
    case FINGERPRINT_PACKETRECEIVEERR:
        lcd.clear();
lcd.setCursor(0,0);
        lcd.print("Communication error");
        return p;
}

```

```

    case FINGERPRINT_FEATUREFAIL:
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("fingerprint");
        lcd.setCursor(0,1);
        lcd.print("features unfound");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("fingerprint");
        lcd.setCursor(0,1);
        lcd.print("features unfound");
        return p;
    default:
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Unknown error!");
        return p;
}

lcd.clear();
lcd.setCursor(0,0);
lcd.print("Remove finger");
delay(1000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
}

```

```

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("ID ");

lcd.print(id);

p = -1;

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Place same ");

lcd.setCursor(0, 1);

lcd.print("finger again");

while (p != FINGERPRINT_OK){
  p = finger.getImage();
  switch (p){
    case FINGERPRINT_OK:
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Image taken!");
      break;
    case FINGERPRINT_NOFINGER:
      break;
    case FINGERPRINT_PACKETRECEIVEERR:
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Communication Error.");
      break;
    case FINGERPRINT_IMAGEFAIL:
      lcd.clear();
      lcd.setCursor(0,0);

```

```

    lcd.print("Imaging error.");
    break;
default:
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Unknown error!");
    break;
}
delay(1000);
}
// OK success!
p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Image converted");
    break;
case FINGERPRINT_IMAGEMESS:
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Image too messy");
    break;
// return p;
case FINGERPRINT_PACKETRECEIVEERR:
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Communication error");

```

```

        break;

        //return p;

    case FINGERPRINT_FEATUREFAIL:

        lcd.clear();

        lcd.setCursor(0,0);

        lcd.print("fingerprint");

    lcd.setCursor(0,1);

    lcd.print("features unfound");

    break;

        //return p;

    case FINGERPRINT_INVALIDIMAGE:

        lcd.clear();

        lcd.setCursor(0,0);

        lcd.print("fingerprint");

    lcd.setCursor(0,1);

    lcd.print("features unfound");

    break;

        // return p;

    default:

        //Serial.println("Unknown error");

        lcd.clear();

        lcd.setCursor(0,0);

        lcd.print("Unknown error!");

        break;

        //return p;

        delay(1000);

    }

    searched = false;

```

```

//*****checks if the finger was already enrolled before.*****

// OK converted!

p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Found a Print");
    lcd.setCursor(0,1);
    lcd.print("Already Enrolled.");
    searched = true;
    delay(2000);
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Communication error!");
    delay(2000);
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    delay(2000);
} else {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Unknown error");
    delay(2000);
    return p;
}

//*****end of already present checked*****

if(!searched)

```



```

{
// OK converted!

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Processing model");

  lcd.setCursor(0,1);

  lcd.print("For # ");

  lcd.print(id);

  delay(1000);

p = finger.createModel();//this creates templates with two prints
if (p == FINGERPRINT_OK) {
  lcd.clear();

  lcd.setCursor(0,0);

  mismatch = false;//states that there was no mismatch hence loop will terminate

  lcd.print("Prints Matched!");

  delay(2000);
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Communication error");

  delay(2000);

  //return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Prints Did Not");

  lcd.setCursor(0,1);

  lcd.print("match, Try again..");
}

```

```

    delay(1500);

    //break;

    //return p;
} else {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Unknown error");
    delay(1000);
    //return p;
}
}

if(!mismatch)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("ID : ");
    lcd.print(id);
    delay(1000);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Template Stored!");
    delay(1000);
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Communication error");

```

```

        delay(1000);
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Could not store");
    lcd.setCursor(0,1);
    lcd.print("in that location");
    delay(1000);
    return p;
} else if (p == FINGERPRINT_FLASHERR) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Error writing to flash");
    delay(1000);
    return p;
} else {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Unknown error");
    delay(1000);
    return p;
}
delay(1000);
}

if(mismatch)
    {
        //this is true if there was a mismatch when scanning same finger; allows enrolment with
        same id

        lcd.clear();

```

```

    lcd.setCursor(0,0);
    lcd.print("Try Enrolling ID");
    lcd.setCursor(0,1);
    lcd.print(id);
    lcd.print(" again-Press #");
    keyPad = keypad.getKey();
    while(!keyPad)
    {
        keyPad = keypad.getKey();
    }
    if(keyPad == '#')
    {
        mismatch = true;
    }
    else
    {
        mismatch = false;
    }
}
}
}
}

```

## **Appendix 2.11: Getting fingerprint id**

//\*\*\*\*\*GETTING FINGER ID FROM FLASH\*\*\*\*\*

//Detailed response

```

uint8_t getFingerprintID(){
    int p = -1;
    lcd.clear();

```

```

if(adminmode)
{
    lcd.setCursor(0,0);
    lcd.print("Place Admin");
    lcd.setCursor(0,1);
    lcd.print("Finger");
}
else
{
    lcd.setCursor(0,0);
    lcd.print("Place Your");
    lcd.setCursor(0,1);
    lcd.print("Finger");
}

delay(1000);
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("Image taken!");
            break;
        case FINGERPRINT_NOFINGER:
            break;
        case FINGERPRINT_PACKETRECEIVEERR:
            lcd.clear();
            lcd.setCursor(0,0);

```

```

        lcd.print("Communication Error.");
        break;
    case FINGERPRINT_IMAGEFAIL:
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Imaging error.");
        break;
    default:
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Unknown error!");
        //Serial.println("Unknown error");
        break;
    }
    delay(1000);
}

// OK success!
delay(1000);
p = finger.image2Tz();
switch (p) {
    case FINGERPRINT_OK:
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        lcd.clear();
        lcd.setCursor(0,0);

```

```

    lcd.print("Image too messy");
    delay(1000);
    //Serial.println("Image too messy");
    return p;
case FINGERPRINT_PACKETRECEIVEERR:
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Communication error");
    delay(1000);
    return p;
case FINGERPRINT_FEATUREFAIL:
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Could not find");
    lcd.setCursor(0,1);
    lcd.print("fingerprint features");
    delay(1000);
    return p;
case FINGERPRINT_INVALIDIMAGE:
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Could not find");
    lcd.setCursor(0,1);
    lcd.print("fingerprint features");
    delay(1000);
    return p;
default:
    lcd.clear();

```

```

    lcd.setCursor(0,0);
    lcd.print("Unknown error");
    delay(1000);
    return p;
}
// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    if(!adminmode)
    {

    }

} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Communication error!");
    delay(2000);
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    if(lockmode)//only works if in search mode
    {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("No Match!");
        digitalWrite(alert, HIGH);
        delay(3000);
        digitalWrite(alert, LOW);
    }
}

```



```

    }
    return p;
} else {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Unknown error");
    delay(2000);
    return p;
}
if(!adminmode)//ensures it's not verification of admin process
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("ENROLLED AS");
    lcd.setCursor(0,1);
    lcd.print("ID => ");
    lcd.print(finger.fingerID);
    delay(2000);
    searched = true;
    if(lockmode)//this ensures led goes high only in search mode
    {
        digitalWrite(lock, HIGH);
        delay(2000);
        digitalWrite(lock, LOW);
    }
}
return finger.fingerID;
}

```

