QR BASED 24x7 SMART ATM SYSTEM

A PROJECT REPORT

Submitted by

MONISHA.P [REGISTER NO: 211417104152]
NAGIYA BHANU.G [REGISTER NO: 211417104157]
NIRANJANA.B [REGISTER NO: 211417104165]

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI - 600123.

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2021

BONAFIDE CERTIFICATE

Certified that this project report **“QR BASED 24x7 SMART ATM SYSTEM”** is the bonafide work of **NAGIYA BHANU.G [REG NO: 211417104157] NIRANJANA.B [REGNO: 211417104165]** and **MONISHA.P [REG NO: 211417104152]** who carried out the project work under my supervision.

SIGNATURE

Dr.S.MURUGAVALLI, M.E., Ph.D.,
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

Mrs.P.VIJAYALAKSHMI,
M.C.A., M.Phil, M.Tech,
SUPERVISOR
ASSOCIATE PROFESSOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidates were examined in the Anna University Project Viva-Voce Examination held on...........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHIKUMAR, M.E., Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E., M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.,** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI M.E., Ph.D.,**for the support extended throughout the project.

We would like to thank my **Project Guide Mrs.P.VIJAYALAKSHMI, M.C.A., M.Phil, M.Tech.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

NAGIYA BHANU. G
NIRANJANA. B
MONISHA. P

ABSTRACT

The ATM System is used to access their bank accounts in order to make cash withdrawals. Whenever the user needs to make cash withdraws, they can enter their PIN number and the amount to be withdrawn in the form of 100’s 500’s and 2000’s. Once their withdrawn was successful, the amount will be debited in their account. It requires more and more time .So, in order to reduce that inconvenience, we introduce this proposed concept for the ATM simply by showing Quick Response Code and Biometric to withdraw their money within 30 sec.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVATIONS

S.NO	ABBREVATION	EXPANSION
1.	ATM	Automated Teller Machine
2.	QR	Quick Response
3.	IDE	Integrated Development Environment

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW

With the development of computer network technology and e-commerce, the self-service banking system has got extensive popularization with the characteristic offering high-quality 24 hours service for customer. Nowadays, using the ATM (Automatic Teller Machine) which provides customers with the convenient bank note trading is very common. However, the financial crime case rises repeatedly in recent years, a lot of criminals tamper with the ATM terminal and steal user's credit card and password by illegal means. Once user's bank card is lost and the password is stolen, the criminal will draw all cash in the shortest time, which will bring enormous financial losses to customer. How to carry on the valid identity to the customer becomes the focus in current financial circle. Traditional ATM systems authenticate generally by using the credit card and the password, the method has some defects. Using credit card and password cannot verify the client's identity exactly. In recent years, the fingerprint recognition continuously updated, which has offered new verification means for us combined with the QR code technology verify the clients' identity better and achieve the purpose that uses of ATM machines improve the safety effectively. A QR Code is a two-dimensional barcode that is readable by smart phones. It allows to encode over 4000 characters in a two dimensional barcode. QR Codes may be used to display text to the user, to open a URL, save a contact to the address book or to compose text messages. "QR Code" is a registered trademark of DENSO WAVE INCORPORATED. To read QR Codes with your smart phone, you need appropriate software installed on your phone.

For Android-based devices, you can use Barcode Scanner by ZXing. On iOS-Devices like iPhones there are also QR Code readers available on the App Store, for Example i-nigma. On Firefox OS try QR Code Scanner. On Symbian devices, you can use Mobile tag barcodes reader for example. Fingerprint scanners are security systems of biometrics. They are used in police stations, security industries, smart phones, and other mobile devices. Everyone has patterns of friction ridges on their fingers, and it is this pattern that is called the fingerprint. Fingerprints are uniquely detailed, durable over an individual's lifetime, and difficult to alter. Because there are countless combinations, fingerprints have become an ideal means of identification.

1.2 Problem Definition

One big fear during COVID-19 has been associated with the use of ATMs. To ensure that the entire process of withdrawing cash from ATMs is completed without touching the machine or the screen is looking possible now. ATM cardholders can now withdraw cash from an ATM by scanning a QR code on the mobile phone by ATM machine’s screen without having to touch the surface. And along with that ATM Card has complications with validation and transmission process. Therefore, by using peeping attack, the secret code was stolen while entering PIN at the ATM machine. So the data of a user is to be ensured that secure and isolated from the data leakage and other attacks.

CHAPTER 2

LITERATURE SURVEY

1. **TITLE** **:** **A**TM Security System Using Arduino.

   AUTHOR **:** B. Saranraj, N. Sri Priya Dharshini, R. Suvetha, K. Uma Bharathi

   YEAR **:** 2020

   DESCRIPTION **:**

   B.Saranraj has proposed that ATM Security System Using Arduino can offer protected and secure support to the clients and to do exchanges without going to bank. Each record holder has an exceptional ATM card, each having a unique account number. To abstain from compromising in ATM machines, this paper gives safe arrangements, for example, biometric authentication. ATM cards have the data about unique mark. The primary target of the venture work is to guarantee better security in ATM exchanges. Right now, use RFID tag rather than ATM card. If there should been occurrence of three wrong endeavors in a day, the client can't play out the exchange.

   MERITS

   - It has high security authentication.
   - The proposed framework gives high proficiency and maintains a strategic distance from illicit exchanges.

   DEMERITS

   - Sometimes fingerprint does not co-ordinate.

2. TITLE : Biometric Based Smart ATM using RFID

AUTHOR : Gokul.S, Kukan.S, Meenakshi.K, Vishnu Priyan S S, Rolant Gini J, E.Harikumar

YEAR : 2020

DESCRIPTION :

Gokul.S have proposed a system,in the present world, the usage of ATM to withdraw cash has increased. At the same time, theft and robbery cases have also been increased that calls for the need for much-secured ATM that provides additional features for security. In this work, the aim is at security-based smart ATM which functions based on RFID and fingerprint authorization for its access. The RFID number and fingerprint details are obtained from the user after which the recognized card number, authorization status, and location of access are passed on for checking its authenticity with the database details. Once the information is validated with the retrieved database details then the corresponding account holder gets the message if the authorization is valid or not. The location, time, and date of the access are also informed to the account holder. Additionally, this enhances the security by placing vibration and flame sensors which immediately notify in case of fire and breakage. To achieve complete security, the face of the person accessing the ATM card is also recorded – using a camera – in the machine with time and date of access that could be used in case of suspicion.

MERITS

- Provides complete security as the face of the person accessing ATM card is recorded using camera with date and time.

DEMERITS

- Security can further be increased by using face detection,iris scanner,OTP generation.

3. TITLE : An Efficient Bar/QR Code Recognition System for Consumer Service Applications.

AUTHOR : Arju Aman, Aryan Singh, Ayush Raj Sandeep Raj

YEAR : 2020

DESCRIPTION :

Arju Aman has proposed that, in recent years, consumer electronics for providing better customer service has sought a significant growth. For production, storage and supply of goods to consumers, it is essential to have correct information, recognize and store the information efficiently in computers. Therefore, it is essential to have an efficient and handy bar code recognition system.This paper presents an efficient method for bar code and QRcode recognition together. The method automatically detects the Bar code QR code and displays the complete information of the product. The method is developed in python environment using Open CV library. However, Open CV does not have any dedicated modules that can be used to read and decode Bar codes and QR codes. The database is developed at the authors' Institute where bar codes and QR codes are separately assigned to more than 100 items such as books, sofa, tables and chairs. The image of the bar code or QR code is captured in real-time and further processed using the proposed method. The code is being decoded, compared with the Data frame of the stored product and finally, displays the result i.e, the complete information about the product. The execution time of the proposed method is 0.25 seconds.

The proposed method can be further prototyped on micro-controllers to develop an efficient bar code recognition system.

MERITS

- It takes less than 0-30 seconds to distinguish barcode
- It takes less time to compute.

DEMERITS

- In captured image we have many noises it can't be removed simultaneously.
- Select the pixels with high gradient.

4. TITLE : An IOT Based ATM Surveillance System

AUTHOR : Jacintha .VJ. Nagarajan, K .Thanga Yogesh, Tamilarasu. S, S .Yuvaraj

YEAR : 2017

DESCRIPTION :

Jacintha have proposed that, in the present scenario, majority of the population uses the ATM machine to withdraw cash. At the same time, there are many ATM robberies that have occurred in many areas, even if the CCTV cameras are placed in the ATM center. Hence the security system needs to be changed. In order to reduce these kinds of robberies, we present a security system for ATM theft by using a smart and effective technology. This system also analyses various physical attacks on ATM’s. In our proposed system we use Face Recognizing Camera to capture the face of the person, who is entering. Tilt and vibration sensors are used to detect the irregular activities that are done on the ATM machine. The purpose of the Temperature sensor is to determine the degree of temperature present in the ATM booth.

The main aim of our proposed system is to send an alert through social media's like Face book, twitter, and Gmail using IOT and GSM network. Liquidator chloroform is used to spread the chloroform to make the thief unconscious. This system caters realistic monitoring and control.

MERITS

- The sensors used in this system are stable, tenable and easy to use.

DEMERITS

- Face is not captured during door opening.
- No video recording system under suspicious facility.

5. **TITLE** : **ATM Terminal design is based on fingerprint recognition**

AUTHOR : Yun Yung, Jia Mi

YEAR : 2010

DESCRIPTION :

Yun Yung have proposed a system, for the traditional ATM terminal customer recognition systems only rely on bank cards, passwords, and such identity verification methods which measures are not perfect and functions are too single. For solving the bugs of traditional ones, the author designs new ATM terminal customer recognition systems. The chip of S3C2440 is used for the core of microprocessor in ARM9, furthermore, an improved enhancement algorithm of fingerprint image increase the security that customer use the ATM machine.

MERITS

- This system contains original verifying methods.
- The security features were enhanced for stability and reliability of recognition and more safe to use.

DEMERITS

- High quality or fingerprint images required.
- This system has only limited number of time for recognition of finger print.

CHAPTER 3
SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 Existing System

The existing ATM system authenticates transactions via the card and PIN-based system. Thereafter, it grants access to bank customers to several services such as cash withdrawal and deposits, account to account transfers, balance enquiry, top-up purchases and utility bills payment.

3.2 Proposed System

The proposed system is an enhancement of the existing system, and, it is built upon the existing card and PIN-based system. The proposed system we design a system based on quick response code with bio metric system it can help them to process cash withdrawal quickly.

3.3 Requirement Analysis

3.3.1 Hardware Requirements

Display	:	LCD Display
Device	:	Fingerprint Device, Relay

3.3.2 Software Requirements

Language	:	Python, Embedded C, Arduino IDLE
Application	:	Android Studio

3.4 Technology Stack

Android Studio

Android Studio is the official integrated development environment.It is used in to create the Application.

Python

Python is for generating QR code, read the QR code from the App and to send the serial data to the microcontroller.

Arduino IDE

The Aurdino Integrated Development Environment (IDE) is a cross-platform application. It is used to write and upload programs.

CHAPTER 4

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 Mobile App workflow

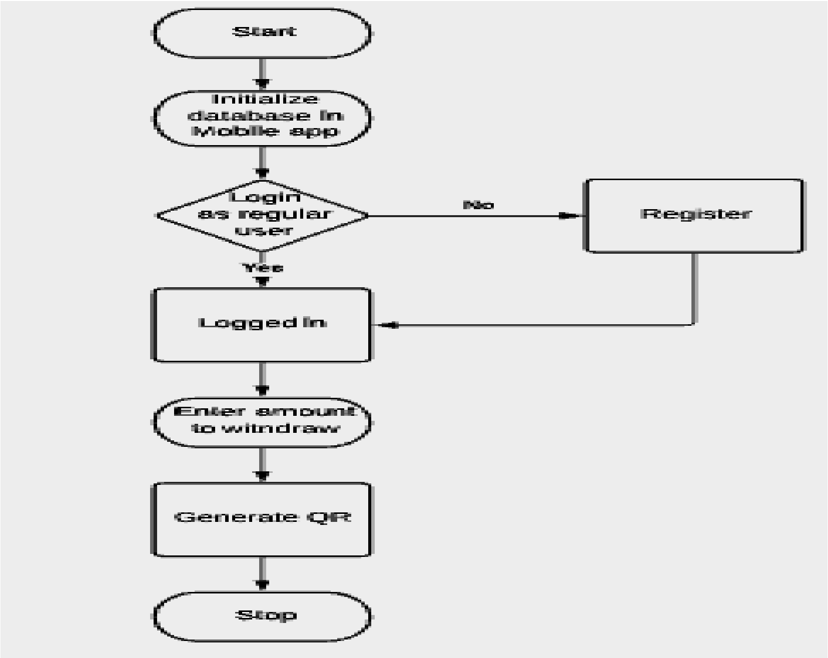

Fig 4.1 Mobile App workflow

The overall mobile workflow of the system is shown in Figure 4.1. In the Mobile Application work flow starts with initialisation of database. It then redirects to Login page where if the user is new then they should register with Name, email id and phone number before logging in and if they are already a registered user then they can login in directly with email id and password. After logged in the user should enter the amount to be withdrawn and press the generate QR code button so that a QR code will be generated.

4.2 System workflow

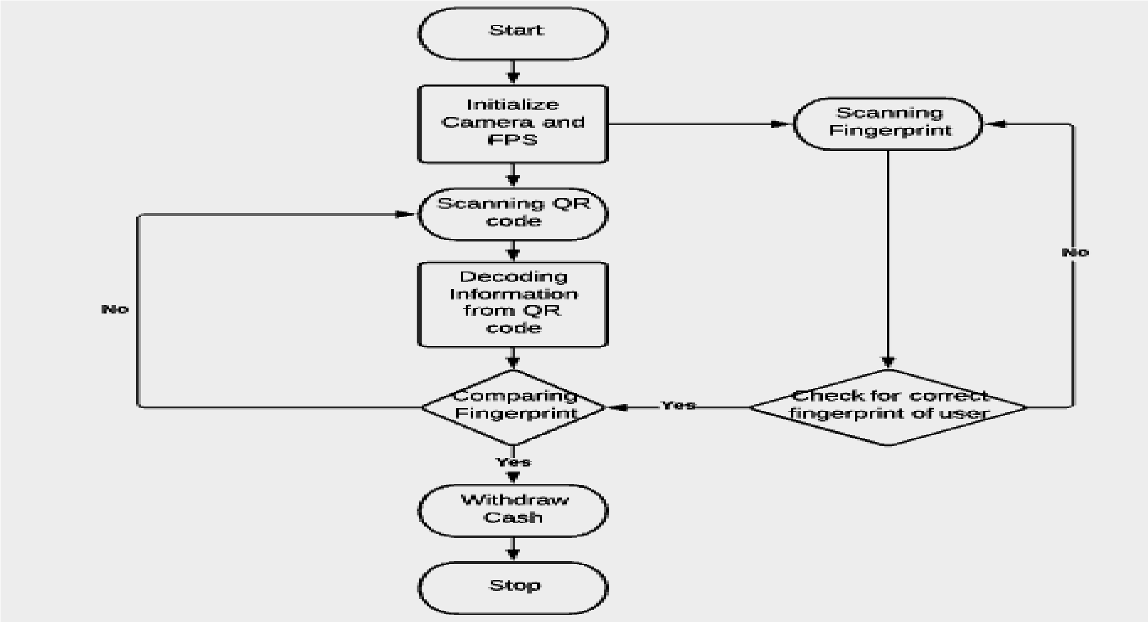
Start
Initialize Camera and FPS
Scanning Fingerprint
Scanning QR code
No
Decoding Information from QR code
No
Comparing Fingerprint
Yes
Check for correct fingerprint of user
Yes
Withdraw Cash
Stop

Fig 4.2 System workflow

The overall System workflow of this system is shown in the Figure 4.2.The system work flow starts with initialisation of Camera and Fingerprint Sensor. As soon as the QR code generated in mobile it should be scanned by the camera in ATM then it will be decoded. At the same time of scanning QR code the fingerprint should also be scanned. If the fingerprint is not registered before or if it is not scanned properly then the user should rescan the fingerprint.

If the fingerprint is inputted correctly then the system compares the QR code which stores information like email id, phone number and amount to withdraw with the fingerprint. If it matches then user can withdraw the money else the user should restart from scanning QR.

4.3 Communication Diagram

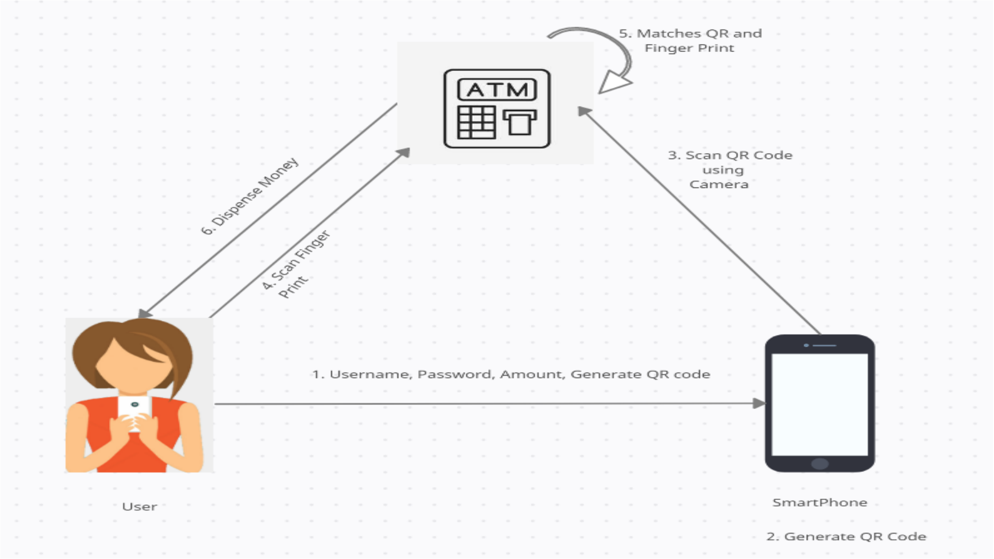
5. Matches QR and Finger Print
ATM
3. Scan QR Code using Camera
6. Dispense Money
4. Scan Finger Print
1. Username, Password, Amount, Generate QR code
User
SmartPhone
2. Generate QR Code

Fig 4.3 Communication Diagram

The overall Communication diagram of this system is shown in Figure 4.3. The complete work flow starts with user who register/login and generate QR code. The QR code generated should be scanned with the camera in ATM and at the same time the user should scan their fingerprint in Fingerprint sensor. The system checks for matching of QR and Fingerprint. If everything get success then the user can withdraw the money.

CHAPTER 5
SYSTEM ARCHITECTURE

5.1 Architecture Overview

The overall system architecture of this QR Based 24X7 Smart ATM system is shown in Figure 5.1.The QR based smart ATM system is based on fingerprint recognition and QR code generation which is designed after analyzed existed ATM system. Integration of hardware have camera to scan qr code and python IDLE for performing transaction and fingerprint sensor for reading finger print for authentication and micro controller for matching inputs and we have LCD to display the result. DC motor and motor driver for ATM operation.

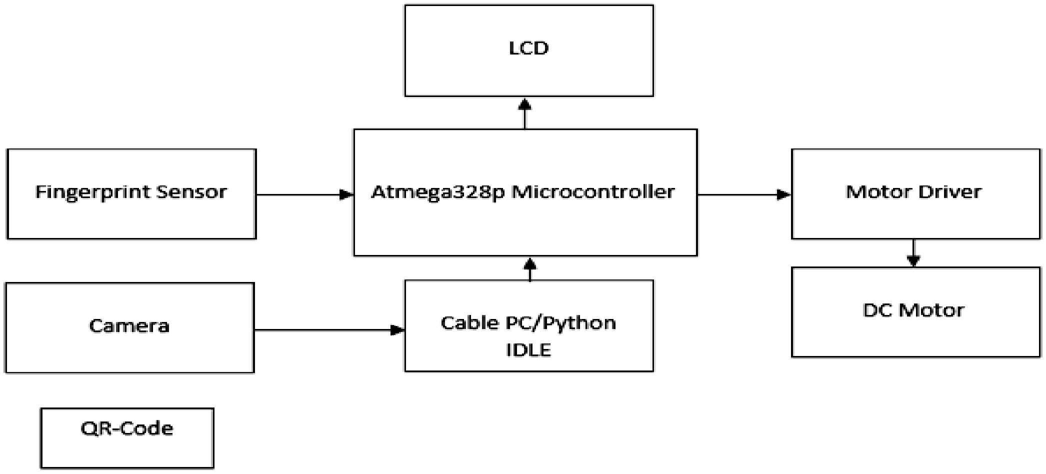

Fig 5.1 System Architecture

5.2 Module Design Specification

Module 1: Integration of Hardware

- Integration of Arduino Microcontroller to LCD Display.
- Integration of Fingerprint Sensor for Biometric Solution which stores fingerprint in the sensor model.
- Connecting Motor with Driver board to preview the Dispensing Method.
- Connecting Power Supply to the hardware kit.

Module 2: Implementation of Software

→ Design & Development of Android Application

- Android Studio is used in to create the Application.
- Mobile Application will have a Login Page where user can Register/Login.
- If the user is new, the user can Register by using their Mobile number, email id and password for first time.
- And if the user is regular, then they can Login using their Email id and Password. Then they can enter the amount to be withdrawn.
- By clicking Generate QR code a QR code will be generated.

→ Programming

- Programming language used are Python and Embedded C
- Python is for generating QR code, read the QR code from the App and to send the serial data to the microcontroller.
- QR code is generated using Google ZXing Package.
- Embedded C is used to writing and uploading programs to Arduino compatible boards.

- Comparing the QR code and Fingerprint is done using Verification module. If matches then motor rotates which means the 2-step authentication is completed successfully and Money will be dispensed.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Client side coding

```
from imutils.video import VideoStream
from pyzbar import pyzbar
import argparse
import datetime
import imutils
import time
import cv2
import serial
arduinoData = serial.Serial('COM5', 115200)
vs = VideoStream(src=0).start()
time.sleep(0.2)
found = set()
readdat=""
def send_cash():
arduinoData.write('Verify'.encode())
print("Place your Enrolled Finger on the sensor ")
DaTa_r = ""
def getSeriAl():
global DaTa_r
if(arduinoData.in_waiting >0):
        line = arduinoData.readline()
readdat=line.decode()
```

```
        print(readdat)
        if(readdat == "Ready\r\n"):
            print("Device online")
            found.clear()
            print("|-----------------------------------------------------------------------|")
            print("|-------------------------Welcome to QR-ATM----------------------------|")
            print("|-----------Please Show your QR code in front of Camera----------------|")
            print("|-----------------------------------------------------------------------|")
            print("")
            print("")
            print("")
        if(readdat == "<<Transaction Successful>>\r\n"):
           print("|--------------------Thank YoU for Using QR ATM------------------------|")
           print("|-----------------------------------------------------------------------|")
           print("")
        if(readdat == "Ready to enroll a fingerprint!\r\n"):
                print('Please Enter Id To enroll in end of #  :')
                x = input()
                arduinoData.write(str(x).encode())
while True:
    getSeriAl()
    frame = vs.read()
    frame = imutils.resize(frame, width=400)
    # find the barcodes in the frame and decode each of the barcodes
    barcodes = pyzbar.decode(frame)
    # loop over the detected barcodes
    for barcode in barcodes:
```

```
 # extract the bounding box location of the barcode and draw
 # the bounding box surrounding the barcode on the image
            (x, y, w, h) = barcode.rect
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
 # the barcode data is a bytes object so if we want to draw it
 # on our output image we need to convert it to a string first
            barcodeData = barcode.data.decode("utf-8")
            barcodeType = barcode.type
# draw the barcode data and barcode type on the image
            text = "{} ({})".format(barcodeData, barcodeType)
            # if the barcode text is currently not in our CSV file, write
            # the timestamp + barcode to disk and update the set
            if barcodeData not in found:
                    out=str(barcodeData).split(',')
                    print("Requested Amount to Withdraw : ",out[1])
                    send_cash();
                    found.add(barcodeData)
    # show the output frame
    cv2.imshow("Smart QR ATM", frame)
    key = cv2.waitKey(1) & 0xFF
    # if the `q` key was pressed, break from the loop
    if key == ord("q"):
            break
# close the output CSV file do a bit of cleanup
print("[INFO] cleaning up...")
cv2.destroyAllWindows()
vs.stop()
```

Hardware Coding

```
#include "SoftwareSerial.h"
#include<LiquidCrystal.h>
#include <Adafruit_Fingerprint.h>
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
SoftwareSerial fps_tx_rx(2, 3);
#define python Serial
#define splash splash1
#define enr A4
#define buz 5
#define rel 4
int enr_r,  buzz_out ;
String IncomingData = "";
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&fps_tx_rx);
uint8_t id;
void(* resetFunc) (void) = 0;
void setup() {
  // put your setup code here, to run once:
  LcDSet();
  python.begin(115200);
  finger.begin(57600);
  splash(0, "Initializing");
  splash(1, "Python");
python.println("Ready");
  delay(1000);
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) {
      delay(1);
    }
  }
  splash(0, "<<Initializing>>");
  splash(1, "<Finger Print>");
  delay(1000);
  pinMode(enr, INPUT_PULLUP);
  pinMode(buz, OUTPUT);
  pinMode(rel, OUTPUT);
  enr_r = digitalRead(enr);
  if (!enr_r) {
    splash(0, "Enrolling");
    splash(1, "FingerPrint");
    delay(1500);
    Enroll();
  }
  splash(0, "<Finger Print>");
  splash(1, "<<Initialized>>");
}

void LcDSet() {
 lcd.begin(16, 2);
  splash(0, "QR-ATM");
  splash(1, "SysteM");
  delay(2000);
  lcd.clear();
}
void loop() {
  // put your main code here, to run repeatedly:
  getSensor();
  getPython();
}
void getSensor() {
  if (buzz_out == 1) {
    digitalWrite(buz, HIGH);
    delay(1000);
    digitalWrite(buz, LOW);
    delay(250);
  }
}
void getPython() {
  while (python.available())
  {
    IncomingData = python.readString();
    delayMicroseconds(5);
  }
  if (IncomingData.length() > 0) {
    if (IncomingData == "Verify") {
      getFinger();
    }
    IncomingData = "";
  }
}
void getFinger() {
  while (1) {
    int id = getFingerprintIDez();
    Serial.println(id);
    if (id > 0 and id < 100)
    {
      splash(0, "Transaction");
      splash(1, "Successful");
      python.println("<<Transaction Successful>>");
      cash_open(1500);
   }
    else if(id== -1 )
    {
      splash(0, "Please Place");
      splash(1, "Your Finger");
      delay(1000);
    }
    else
    {
      splash(0, "FingerPrint");
      splash(1, "Missmatch");
      delay(1000);
    }
    delay(100); }}
```

```
void cash_open( int d) {
 digitalWrite(rel, HIGH);
 delay(d);
 digitalWrite(rel, LOW);
 digitalWrite(buz, HIGH);
 delay(50);
 digitalWrite(buz, LOW);
 delay(150);
 digitalWrite(buz, HIGH);
 delay(50);
 digitalWrite(buz, LOW);
 delay(150);
 digitalWrite(buz, HIGH);
 delay(50);
 digitalWrite(buz, LOW);
 delay(150);
 digitalWrite(buz, HIGH);
 delay(50);
 digitalWrite(buz, LOW);
 delay(50);
 splash(0, "Please Take");
 splash(1, "Your Cash");
 delay(1500);
 splash(0, "Thank You");
 splash(1, "For Using");
 delay(1500);
 resetFunc(); }
```

```
uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }
  // OK success!
  p = finger.image2Tz();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image converted");
      break;
    case FINGERPRINT_IMAGEMESS:
      Serial.println("Image too messy");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_FEATUREFAIL:
      Serial.println("Could not find fingerprint features");
      return p;
    case FINGERPRINT_INVALIDIMAGE:
      Serial.println("Could not find fingerprint features");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }

  // OK converted!
  p = finger.fingerSearch();
  if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
  }
else
```

```
{
    Serial.println("Unknown error");
    return p;
  }

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez()
{
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;
  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;
  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  int idd = finger.fingerID;
  return idd;
}
```

Application Development Code

Sign up Activity

```
package com.kathir.qpay;
import android.content.ContentValues;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class SignUpActivity extends AppCompatActivity {
    EditText
activity_signup_name,activity_sign_up_email,activity_sign_up_password,activity_sig
nup_phone;
    Button activity_sign_up_reg;
    TextView activity_sign_up_member;
SQLiteDatabase db ;
    private int flag = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_up);
        activity_signup_name=(EditText)findViewById(R.id.activity_signup_name);
        activity_sign_up_email=(EditText)findViewById(R.id.activity_sign_up_email);
activity_sign_up_password=(EditText)findViewById(R.id.activity_sign_up_password
);
        activity_signup_phone=(EditText)findViewById(R.id.activity_signup_phone);
activity_sign_up_reg=(Button)findViewById(R.id.activity_sign_up_reg);
        activity_sign_up_reg.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
           checkUsers(activity_sign_up_email.getText().toString().trim());
               insertUsers(activity_signup_name.getText().toString().trim(),
                       activity_sign_up_email.getText().toString().trim(),
                       activity_sign_up_password.getText().toString().trim(),
                       activity_signup_phone.getText().toString().trim());
            }
        }
);
activity_sign_up_member=(TextView)findViewById(R.id.activity_sign_up_member)
;
        activity_sign_up_member.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i =new Intent(getApplicationContext(),LoginActivity.class);
            startActivity(i);
         }
  });
   }
 public void insertUsers(String name1, String email1, String password1, String
phone1){
      if (flag == 0) {
         try {
 db = getApplicationContext().openOrCreateDatabase("QPaY",
SQLiteDatabase.CREATE_IF_NECESSARY, null);
 db.execSQL("CREATE TABLE users (id integer PRIMARY KEY
AUTOINCREMENT, name text, email text, password text, phone text)");
         }
catch (Exception e)
{
         e.printStackTrace(); }
         ContentValues values = new ContentValues();
         values.put("name", name1);
         values.put("email", email1);
         values.put("password", password1);
         values.put("phone", phone1);

if ((db.insert("users", null, values)) != -1)
{
Toast.makeText(getApplicationContext(), "Successfully Registered",
Toast.LENGTH_LONG).show();
    }
```

```
            else
            {
             Toast.makeText(getApplicationContext(), "Error...",
            Toast.LENGTH_LONG).show();
           }
            }
        else if(flag == 1)
        {
           Toast.makeText(getApplicationContext(), "This user is already registered",
          Toast.LENGTH_LONG).show();
        }
    }
    public void checkUsers(String emailId)
    {
        try
        {
            db =getApplicationContext().openOrCreateDatabase("Demo",
SQLiteDatabase.CREATE_IF_NECESSARY, null);
            Cursor m = db.rawQuery("SELECT * FROM users", null);
            m.moveToFirst();
            while (!m.isAfterLast())
            {
                String s0 = m.getString(0);
                String s1 = m.getString(1);
                String s2 = m.getString(2);
```

```
   // Toast.makeText(getApplicationContext(),s0+" "+s1+"
"+s2,Toast.LENGTH_SHORT).show();
                if (s2.equals(emailId))
              {
                    flag = 1;
                    // addToCart.setText(getResources().getString(R.string.added));
                }
                m.moveToNext();
            }
        }
        catch (SQLiteException e)
        {
          e.printStackTrace();
            System.out.print("ERROR.............");
        }}}
```

Login Activity

```
package com.kathir.qpay;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class LoginActivity extends AppCompatActivity {
    TextView activity_main_reg;
    EditText activity_main_email,activity_main_password;
    Button activity_main_btnlgn;
    SQLiteDatabase db ;
    private int flag = 0;
    @Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        activity_main_reg=(TextView)findViewById(R.id.activity_main_reg);
        activity_main_reg.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(),SignUpActivity.class);
                startActivity(i);
            }
        });

 activity_main_email=(EditText)findViewById(R.id.activity_main_email);
activity_main_password=(EditText)findViewById(R.id.activity_main_password);
        activity_main_btnlgn=(Button)findViewById(R.id.activity_main_btnlgn);
        activity_main_btnlgn.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v)
{

loginUsers(activity_main_email.getText().toString().trim(),activity_main_password.
getText().toString().trim());
      }
      });
}
 public void loginUsers(String email, String password){
  try {db = getApplicationContext().openOrCreateDatabase("QPaY",
SQLiteDatabase.CREATE_IF_NECESSARY, null);
Cursor m = db.rawQuery("SELECT * FROM users WHERE email='" + email + "'
AND password='" + password + "' ", null);
                if(m.getCount()==0){
                    Toast.makeText(getApplicationContext(), "Not Registered or check
username and password", Toast.LENGTH_SHORT).show();
 }
                m.moveToFirst();
                while (!m.isAfterLast()) {
                    String s0 = m.getString(0);
                    String s1 = m.getString(1);
                    String s2 = m.getString(2);
                    String s3 = m.getString(3);
                    //  Toast.makeText(getApplicationContext(),s0+" "+s1+"
"+s2,Toast.LENGTH_SHORT).show();
```

```
                if (s2.equals(email) && s3.equals(password)) {
                    Intent i = new Intent(LoginActivity.this, Qr_Activity.class);
                    //Change the activity.
                  //  i.putExtra(EXTRA_ADDRESS, address);
                    startActivity(i);
              overridePendingTransition(R.anim.right_slide_in, R.anim.right_slide_out);
            Toast.makeText(getApplicationContext(), "Login Success",
        Toast.LENGTH_SHORT).show();
                }
                m.moveToNext();}
          } catch (SQLiteException e) {
            e.printStackTrace();
            Toast.makeText(getApplicationContext(), "Not Registered",
Toast.LENGTH_SHORT).show();
          }}}
```

QR Activity

```
package com.kathir.qpay;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.graphics.Bitmap;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import com.google.zxing.BarcodeFormat;
import com.google.zxing.MultiFormatWriter;
import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;
import com.journeyapps.barcodescanner.BarcodeEncoder;

public class Qr_Activity extends AppCompatActivity {
    EditText amt,pin;
 Button gnt;
    ImageView qrimg;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_qr_);
        gnt =(Button)findViewById(R.id.gnt_btn);
        amt =(EditText)findViewById(R.id.amt_txt);
        pin= (EditText)findViewById(R.id.pin_txt);
        qrimg = (ImageView)findViewById(R.id.qr_img_v);
        gnt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Gnt_Qr();
            }
        });}
```

```
  private void Gnt_Qr() {
String fnQr = gnt.getText().toString();
    MultiFormatWriter multiFormatWriter = new MultiFormatWriter();
    try {
      BitMatrix bitMatrix = multiFormatWriter.encode(fnQr,
BarcodeFormat.QR_CODE, 500,500);
      BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
      Bitmap bitmap = barcodeEncoder.createBitmap(bitMatrix);
      qrimg.setImageBitmap(bitmap);
//           btn_save.setVisibility(View.VISIBLE);
    }catch(WriterException e){
      e.printStackTrace();
    }
  }}
```

CHAPTER 7

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that theSoftware system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : Identified classes of invalid input must be rejected.

Functions : Identified functions must be exercised.

Output : Identified classes of application outputs must be exercised.

Systems/Procedures : Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.3 Test cases and Reports

TEST REPORT: 1

- **Product: QR Based 24x7 Smart ATM System**
- **LOGIN DETAILS**

Test Case ID	Test Case / Action to perform	Expected Result	Actual Result	Pass / Fail
1.	Enter the correct username and Password	Displays “Username and Password is correct”	Username and Password is Correct	Pass
2.	Click the “Submit” button	Displays “Login is successful”	Login is successful.	Pass

Tab. No 7.1 Login Details

TEST REPORT: 2

- **Product: QR Based 24x7 Smart ATM System**
- **NEW USER REGISTRATION**

Test Case ID	Test Case / Action to perform	Expected Result	Actual Result	Pass / Fail
1.	New user Registration	Displays “Correct Username and password”	User name and password is correct	Pass
2.	Login to user Id	Displays “User”	User login correct	Pass

Tab. No 7.2 New User Registration

TEST REPORT: 3

- **PRODUCT : QR Based 24x7 Smart ATM System**
- **FINGERPRINT VERIFICATION**

Test Case ID	**Test Case / Action to perform**	**Expected Result**	**Actual Result**	**Pass / Fail**
1.	To check the fingerprint is valid or not	Fingerprint matched correctly	Valid Fingerprint	Pass

Tab. No 7.3 Fingerprint Verification

CHAPTER 8
CONCLUSION

8. CONCLUSION

8.1 Conclusion and Future Enhancements

The design of this ATM system based on QR code and fingerprint recognition took advantages of the stability and reliability of fingerprint characteristics, a new biological technology. Additionally, the system also Reduces Contact based use (very useful in the times like covid-19). And it is more Secured and Efficient. This project Saves Human Life as we have to spend only few seconds in ATM which becomes difficult for an attacker to attack us. Only Authorized person can use the card as we have multilevel Authentication. This system will also be suitable for people who have very less knowledge to use ATM(s).

In future even the fingerprint authentication can be done in Smart phone using sensors in it which makes it easier or instead of fingerprint even face recognition can also be made.

APPENDICES

A.1 Sample Screens

USER REGISTRATION

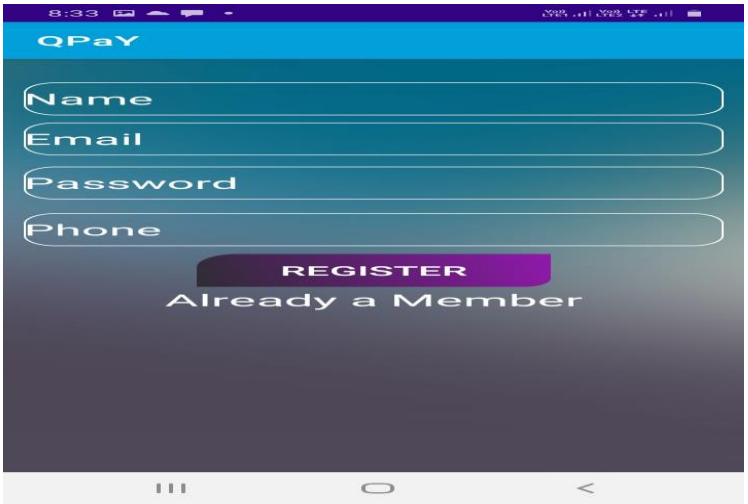

Fig A.1 User Registration

LOGIN PAGE

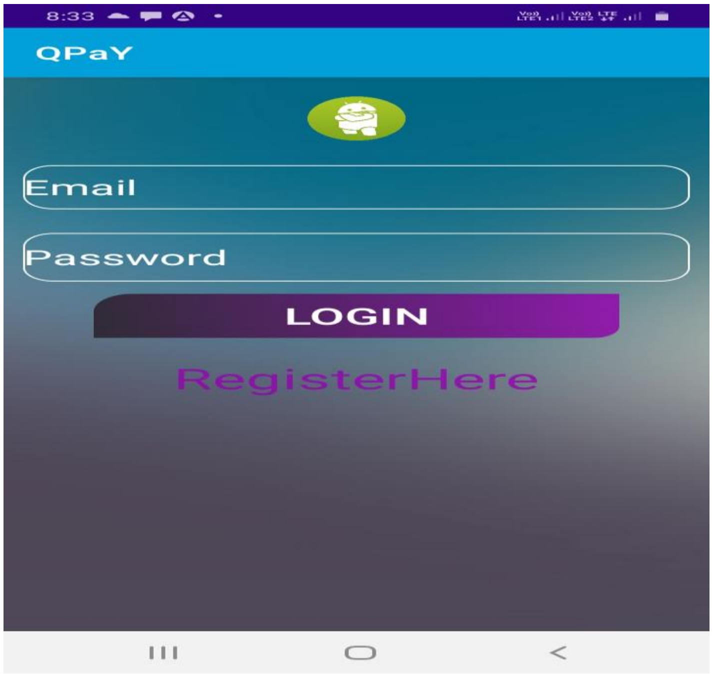

Fig A.2 Login Page

AMOUNT DETAILS

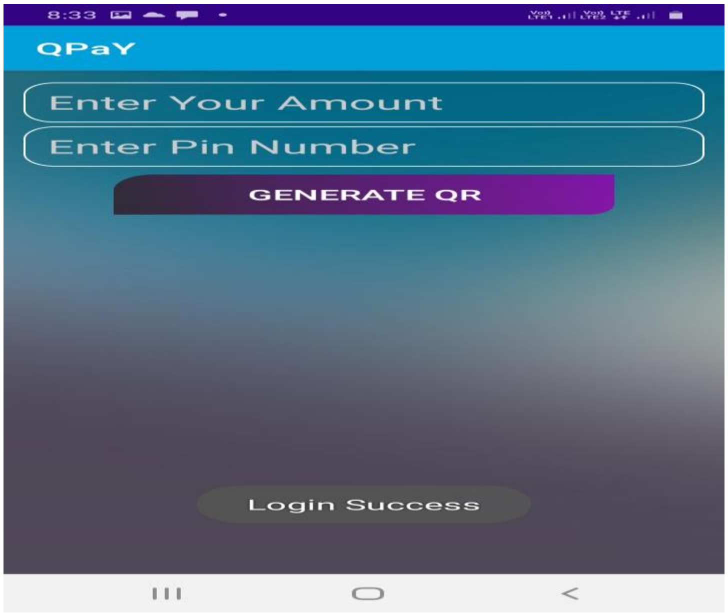

Fig A.3 Amount Details

QR GENERATION

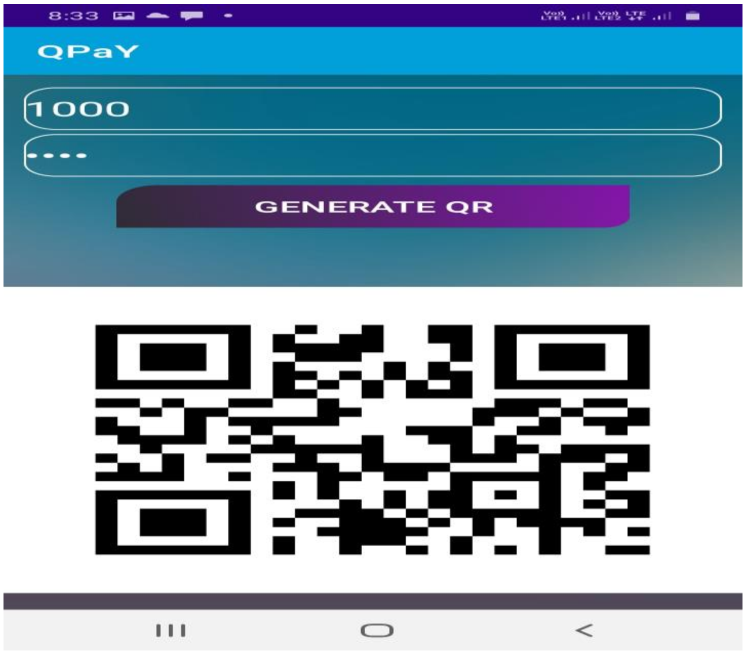

Fig A.4 QR Generation

HARDWARE KIT

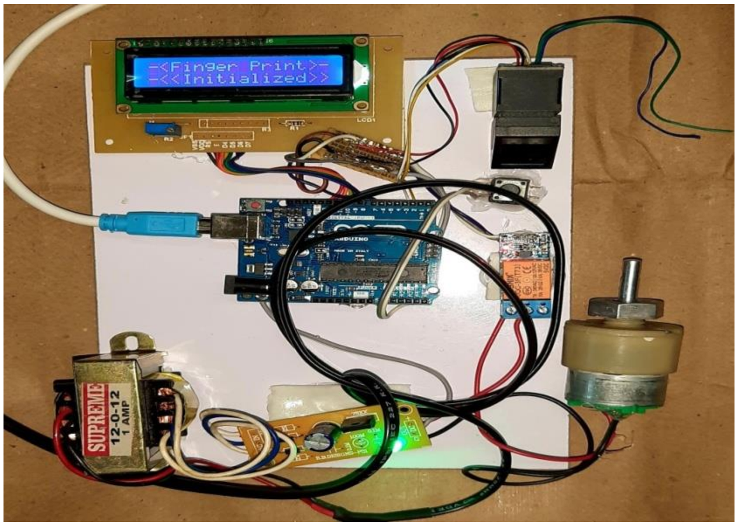

Fig A.5 Hardware Kit

A.2 Publications

https://ijesc.org/upload/a31d1b97f30c9afaeee19924b30b5296.QR%20Based%2024X7%20Smart%20ATM%20System-converted.pdf

REFERENCES

[1]. onyesolu. m. o and ezeani. i. m, “ATM security using fingerprint biometric identifier: an investigative study”, international journal of advanced computer science and applications 2012.

[2]. renee jebaline. g, gomathi. s, “a novel method to enhance the security of ATM using biometrics”, international conference on circuit, power and computing technologies 2015.

[3]. sweta singh, akhilesh singh, rakesh kumar, “a constraint based biometric scheme on ATM and swiping”, international conference on computational techniques in information and communication technologies (icctict) 2016.

[4]. T. R. Tuinstra, “Reading barcodes from digital imagery,” Ph.D. dissertation, Cedarville University, 2006.

[5] E. Tekin and J. M. Coughlan, “An algorithm enabling blind users to find and read barcodes,” in Applications of Computer Vision (WACV), 2009 Workshop on, 2009, pp. 1–8.

[6]. X. Q. James Juett, “Barcode localization using bottom-hat filter,” NSF Research Experience for Undergraduates, 2005.

[7]. Jun Zhou, Guangda Sua, Chun hongJiang. “A face and fingerprint identity authentication system based on multi-route detection”. Neurocomputing 70 (2007) pp. 922-931.

[8]. Yuliang He, Jie Tian, Xiping Luo, Tanghui Zhang. “Image enhancement and minutiae matching in fingerprint verification”. Pattern Recognition Letters 24 (2003) pp.1349-1360.

[9]. Wei Wang, Jianwei Li, Feifei Huang, Hailiang Feng. “Design and implementation of Log-Gabor filter in fingerprint image enhancement”. Pattern Recognition Letters 29 (2008) pp. 301-308.

[10]. Arju Aman,Aryan Singh,Ayush Raj and Sandeep Raj. “An Efficient Bar/QR code recognition system for consumer service applications”. Pattern recognition letters 2020.