IOT BASED BANK SECURITY SYSTEM

CODING

```
#include <call.h>

#include <gps.h> #include <HWSerial.h> #include <inetGSM.h> #include <LOG.h> #include
<SIM900.h> #include <sms.h> #include <Streaming.h>

#include <WideTextFinder.h> #include <Wire.h>

#include <RTClib.h> #include <Keypad.h>

#include <LiquidCrystal_I2C.h> #include "FPS_GT511C3.h" #include "SoftwareSerial.h" #include
<String.h>

#include <sms.h>

#include <SoftwareSerial.h> #include "Adafruit_FONA.h" #define PIN_RELAY 32

#define PIN_BUZZER 13

#define PIN_POWER  35

#define PIN_PIR    33

#define PIN_VIB    41

#define PIN_GSM_ON 9

#define VIB_HR 9
```

```
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns


char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
 {'S','0','E','D'}

};

byte rowPins[ROWS] = {30, 28, 26, 24}; //connect to the row pinouts of the keypad byte
colPins[COLS] = {25, 27, 29, 31}; //connect to the column pinouts of the keypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);


SMSGSM sms;

SoftwareSerial mySerial(4, 5);


RTC_DS1307 RTC;

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);


FPS_GT511C3 fps(11, 12);


int hr_24, hr_12; String
hr_Status="";


String str_Password="";

String str_AdminPW="5647";

String str_acno="";
```

```
int  menu_status=0;

int admin_status=0;

int acc_no=0; int





pir_val=0; int

vib_val=0;


int buttonState = 0;



int randNumber; String

str_tmprndpw="";

String str_rndpw="";

unsigned long start, finished, elapsed;


//char smsbuffer[160];

//char n[20];



#define FONA_RX 2

#define FONA_TX 10

#define FONA_RST 4



SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);

Adafruit_FONA fona = Adafruit_FONA(FONA_RST); char

sendto[21]="+94777211092", message[141]="Anis";


void setup()
```

```
{

  pinMode(PIN_RELAY, OUTPUT);
pinMode(PIN_BUZZER, OUTPUT); pinMode(PIN_POWER,
INPUT); pinMode(PIN_GSM_ON, OUTPUT);
pinMode(PIN_PIR, INPUT);

  Serial.begin(9600);
lcd.begin(20,4);       Wire.begin();
mySerial.begin(9600);



  RTC.begin();

  RTC.adjust(DateTime(_DATE_,_TIME_));

  fps.UseSerialDebug = false; // messages in the serial debug screen fps.Open();
 //fps.SetLED(true); // turn on the LED inside the fps

lcd.setCursor(0,0); lcd.print("
Bank Locker ");
lcd.setCursor(0,1); lcd.print("
System     "); delay(2000);
lcd.setCursor(0,0); lcd.print("
Developed By ");
lcd.setCursor(0,1); lcd.print("
Sutharshan S"); delay(2000);
lcd.clear();

  digitalWrite(PIN_GSM_ON, HIGH);
}

   beep(200); menu_status=0;
  admin_status=0;

   digitalWrite(PIN_RELAY,HIGH);

  fonaSS.begin(4800); // if you're using software serial //Serial1.begin(4800); // if
  you're using hardware serial

   if (! fona.begin(fonaSS)) // can also try fona.begin(Serial1)

   {

     Serial.println(F("Couldn't find FONA"));     while
  (1);

   }

    char sendto[21]="+94777211092", message[141]="Welcome to Bank Locker
System";

     if (!fona.sendSMS(sendto, message)) {

       Serial.println(F("Failed"));

     } else {

       Serial.println(F("Sent!"));

     }

  delay(3000);

}
```

```
void loop()

{

 buttonState = digitalRead(PIN_POWER);



 if(buttonState==HIGH)

 {

  fps.SetLED(true);

 }

 else

 {

  fps.SetLED(false);

 }



 //Date and Time
// Date time now = RTC.now();
  if(menu_status==0)

  {

   hr_24=now.hour();
 if (hr_24==0)

   {

    hr_12=12;

   }
```

```
else
   {

     hr_12=hr_24%12;

   }


   if (hr_24<12)

   {

     hr_Status="AM";

   }
else
   {

     hr_Status="PM";

   }


   String v_date= String(now.year()) + "-" + String(now.month()) + "-" +
String(now.day());

   String v_time= String(hr_12) + ":" + String(now.minute()) + ":" + String(now.second()) + " "
+ hr_Status;


   //Serial.println(v_date);
   //VIB     vib_val = digitalRead(PIN_VIB); // read input
value


   if(vib_val==0)

   {
     if(now.hour()>=VIB_HR)

     {

       Serial.print("Vibration Detected");

                      lcd.clear();
lcd.setCursor(0,0);
lcd.print("Vibration Detected");
lcd.setCursor(0,1);        lcd.print("Be
Alert");                       beep(300);
delay(2000);        menu_status=4;

     }


    }

    lcd.setCursor(0,0);
lcd.print("Date: " + v_date);
lcd.setCursor(0,1);
lcd.print("Time: " + v_time);

//lcd.setCursor(14,1);
//lcd.print(hr_Status);


    if (fps.IsPressFinger())

    {

      fps.CaptureFinger(false);
int id = fps.Identify1_N();      if
(id <200)

 {
      Serial.print("Verified ID:");

      Serial.println(id);

       lcd.clear();
lcd.setCursor(0,0);
lcd.print("Valid Id");
lcd.setCursor(0,1);
lcd.print(id);


      str_tmprndpw="";
generatePassword();


      str_tmprndpw="Your PIN Code is:" + String(str_tmprndpw);


      str_tmprndpw.toCharArray(message,160);


    char sendto[21]="+94777211092", message[141]="";
str_tmprndpw.toCharArray(message,141);


    fonaSS.begin(4800); // if you're using software serial

    //Serial1.begin(4800); // if you're using hardware serial


    if (! fona.begin(fonaSS)) // can also try fona.begin(Serial1)

    {

      Serial.println(F("Couldn't find FONA"));
while (1);

    }
```

```
        Serial.println(F("FONA is OK"));


      if (!fona.sendSMS(sendto, message)) {
      serial.println(F(“failed”));
                    } else {
        Serial.println(F("Sent!"));

         start=millis();
Serial.print("Start:");
        Serial.println(start);

      }


lcd.clear();         lcd.setCursor(0,0);
lcd.print("SMS Sent");
        //lcd.setCursor(0,1);

        //lcd.print("Pin Code:" + str_tmprndpw);



        delay(5000);

      }
else
      {

        Serial.println("Finger not found");
                 lcd.clear();
lcd.setCursor(0,0);
lcd.print("Finger not found");
delay(1000);


      }
```

```
  }
else
  {

   Serial.println("Please press finger");

  }


  if(digitalRead(PIN_PIR) == HIGH && now.hour()>22)    {

   beep(200);

  }

 }


 if(menu_status==4)

 {

  for(int x=1;x<=5;x++)

  {

   beep(500);

  }

  menu_status=0;


 }
 char customKey = customKeypad.getKey();


 if (customKey)

 {
      Serial.println(customKey);


      if(customKey=='A')//Admin

      {

        menu_status=1;
beep(200);
Serial.println("Admin");

         lcd.clear();
lcd.setCursor(0,0);
lcd.print("Admin Password:");
lcd.setCursor(0,1);        lcd.blink();

}
      if(customKey=='B')//Enroll

      {

        menu_status=2;
beep(200);


        if(admin_status==1)

        {

          Serial.println("Enrollment");

           lcd.clear();
lcd.setCursor(0,0);
lcd.print("E-Enter Ac/No:");
lcd.setCursor(0,1);

          lcd.blink();
        }       else        {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Log in");
    lcd.setCursor(0,1);
    lcd.print("Required");
    delay(2000);

        }

      }



      if(customKey=='C')//Delete

      {

        menu_status=3;
    beep(200);


        if(admin_status==1)
{

          Serial.println("Delete");

           lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("D-Enter Ac/No:");

    lcd.setCursor(0,1);          lcd.blink();
          Serial.println(fps.DeleteID(str_acno.toInt()));

        }       else        {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Log in");
    lcd.setCursor(0,1);
    lcd.print("Required");
    delay(2000);
        }
      }

     if(customKey=='D')//Process
     {
      beep(200);
    lcd.noBlink();
    if(menu_status==1)
      {
        if(str_Password==str_AdminPW)
        {
          beep(200);
    admin_status=1;
    menu_status=0;
        }     else     {
    lcd.clear();
    lcd.setCursor(0,0);
          lcd.print("Invalid Password");
    beep(500);     delay(2000);
    lcd.clear();

          str_Password="";
    admin_status=0;
    menu_status=0;
        }
      }


      if(menu_status==2)

      {

        acc_no=str_acno.toInt();
    Enroll(acc_no);        acc_no=0;
    menu_status=0;

      }


      if(menu_status==3)

      {

        acc_no=str_acno.toInt();              fps.DeleteID(acc_no);
              lcd.clear();

    lcd.setCursor(0,0);
    lcd.print("Deleted");
    beep(500);
    delay(2000);
    lcd.clear();


        acc_no=0;
    menu_status=0;

      }

      if(menu_status==5)

      {
```

```
    finished=millis();

    Serial.print("Finished:");

    Serial.println(finished);


    elapsed=finished-start;

    Serial.print("Milliseconds Elapsed:");

    Serial.println(elapsed);


    str_tmprndpw=str_tmprndpw.substring(str_tmprndpw.length()-
6,str_tmprndpw.length());


    if(elapsed<30000)

    {

     if(str_rndpw==str_tmprndpw)

     {

      //Relay
digitalWrite(PIN_RELAY,LOW);

      //beep(200);

      //delay(2000);


      //digitalWrite(PIN_RELAY,HIGH);
beep(200);

      Serial.println("Pin Success");

             lcd.clear();
lcd.setCursor(0,0);

lcd.print("PIN Success");
```

```
          delay(2000);

        }                                          else
        {

          Serial.println("Invalid Pin Code");
lcd.clear();               lcd.setCursor(0,0);
lcd.print("Invalid PIN");          delay(2000);

        }

       menu_status=0;

     }
else
     {        lcd.clear();
lcd.setCursor(0,0);
lcd.print("Time Out");

     }

     str_rndpw="";
str_tmprndpw="";

    }

  }
    if(customKey=='S')//Enroll

    {

       menu_status=5;
beep(200);       Serial.println("Pin
Enter");

         lcd.clear();
lcd.setCursor(0,0);
lcd.print("Enter Pin:");
lcd.setCursor(0,1);
           lcd.blink();

   }



   if(customKey=='E')//Enroll

   {      lcd.clear();      menu_status=0;
admin_status=0;      acc_no=0;
digitalWrite(PIN_RELAY,HIGH);

   }


   if (customKey=='1' || customKey=='2' || customKey=='3' || customKey=='4' ||
customKey=='5' || customKey=='6' || customKey=='7' || customKey=='8' ||
customKey=='9' || customKey=='0' || customKey=='.')
   {

     if(menu_status==1)

     {
lcd.print("*");
       str_Password += customKey;
       Serial.println(str_Password);

     }



     if(menu_status==2 || menu_status==3)

     {

       lcd.print(customKey);
str_acno += customKey;
```

```
         Serial.println(str_acno);

        }


        if(menu_status==5)

        {
        lcd.print(customKey);
str_rndpw += customKey;

        Serial.println(str_rndpw);

      }

    }

    delay(100);

  }

}

 void beep(unsigned char delayms){ analogWrite(PIN_BUZZER, 20);


  delay(delayms);
analogWrite(PIN_BUZZER,0);
delay(delayms);

}


//void SendSMSMessage()

//{

// if (!fona.sendSMS("+94777211092", smsbuffer)) {

//    Serial.println(F("Failed"));
// } else {

//    Serial.println(F("Sent!"));

// }

//}


void generatePassword()

{

  //String temp_pw=""; randomSeed(analogRead(0)); // read from an analog port with
nothing connected

   for(int i=0; i < 6; i++)

   {

     randNumber = random(0,9);        str_tmprndpw=str_tmprndpw
 + String(randNumber);
   }

   Serial.print(str_tmprndpw);

 }


 void Enroll(int en)

 {

             // Enroll test


             // find open enroll id
```

```
            //int enrollid = 0;
            bool usedid = true;
            while (usedid == true)

            {

                         usedid = fps.CheckEnrolled(en);          //enrollid++;

                       if (usedid==true)

              {

                lcd.clear();
lcd.setCursor(0,0);
lcd.print("Id Already Exist");

              }


            }

             fps.EnrollStart(en);



// enroll
              Serial.print("Press finger to Enroll #");

             Serial.println(en);

         lcd.clear();
lcd.setCursor(0,0);
lcd.print("Place finger to");
lcd.setCursor(0,1);
lcd.print("Enroll -1");


            while(fps.IsPressFinger() == false) delay(100);
            bool bret = fps.CaptureFinger(true);        int iret =

0;        if (bret != false)
            {
                    Serial.println("Remove finger");
            lcd.clear();
lcd.setCursor(0,0);
lcd.print("Remove Finger");


                   fps.Enroll1();

                     while(fps.IsPressFinger() == true) delay(100);
                    Serial.println("Press same finger again");
                    lcd.clear();
lcd.setCursor(0,0);
lcd.print("Place Finger to");
lcd.setCursor(0,1);
lcd.print("Enroll -2");
         while(fps.IsPressFinger() ==
false) delay(100);                 bret =
fps.CaptureFinger(true);
         if (bret != false)
                  {

                             Serial.println("Remove finger");
                                 lcd.clear();
lcd.setCursor(0,0);
lcd.print("Remove Finger");


                             fps.Enroll2();

                               while(fps.IsPressFinger() == true) delay(100);
                              Serial.println("Press same finger yet again");

                 lcd.clear();
lcd.setCursor(0,0);
lcd.print("Place Finger to");
lcd.setCursor(0,1);
lcd.print("Enroll -2");


          while(fps.IsPressFinger() == false) delay(100); bret =
          fps.CaptureFinger(true);                                          if (bret !=
false)
                                 {
                                              Serial.println("Remove finger");
```

```
                lcd.clear();
lcd.setCursor(0,0);                lcd.print("Remove
Finger");


                            iret = fps.Enroll3();
                            if (iret == 0)
                            {
                                    Serial.println("Enrolling Successfull");
```

```
                         lcd.clear();
lcd.setCursor(0,0);                          lcd.print("Enroll.
Success");                                 lcd.setCursor(0,1);
lcd.print("Id:" + String(en));
delay(2000);                          menu_status=0;
                                     }
                                     else
                                     {
                                                Serial.print("Enrolling Failed with error code:");
                                               Serial.println(iret);

                         lcd.clear();
lcd.setCursor(0,0);
lcd.print("Enrollment");
lcd.setCursor(0,1);
                                                lcd.print("
Failed");                                       delay(2000);
menu_status=0;
                                              }
                                      }
                                         else Serial.println("Failed to capture third finger");
                          }
                             else Serial.println("Failed to capture second finger");
             }
                else Serial.println("Failed to capture first finger");
}
```