#include<SoftwareSerial.h>

SoftwareSerial SCAN(11,1);

SoftwareSerial SMS(12,13);

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,16,2);

char CARD\_1[]="$0000959393";

char CARD\_2[]="$0000963967";

String message="";

#include <Key.h>

#include <Keypad.h>

const byte ROWS = 4;

const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {

{'1', '2', '3', 'A'},

{'4', '5', '6', 'B'},

{'7', '8', '9', 'C'},

{'\*', '0', '#', 'D'}

};

char customKey=0;

byte rowPins[ROWS] = {3,4,5,6};

byte colPins[COLS] = {7,8,9,10};

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

#define Password\_Length 5

#define solar\_read A3

#define mains\_read A2

byte data\_count = 0, master\_count = 0;

long randomNumber;

String mystr;

char buff[10],buf[40];

int limt\_flag1=0;

int limt\_flag2=0;

int limt\_flag3=0;

int charging1\_flag =0;

int charging\_flag=0;

char Data[Password\_Length];

int card1\_balance = 500;

int card2\_balance = 500;

int flag1 = 0;

int card1\_flag = 0;

int card2\_flag = 0;

int otp\_flag = 0;

int input\_read1 = 0;

int input\_read2 = 0;

//int input\_read = 0;

int relay\_pin2 = A1;

int relay\_pin1 = A0;

int relay\_pin3 = 2;

void setup()

{

Serial.begin(9600);

SCAN.begin(9600);

SMS.begin(9600);

pinMode(mains\_read,INPUT);

pinMode(solar\_read,INPUT);

pinMode(relay\_pin1,OUTPUT);

pinMode(relay\_pin2,OUTPUT);

pinMode(relay\_pin3,OUTPUT);

Wire.begin();

lcd.init();

lcd.backlight();

lcd.setCursor(3,0);

lcd.print("EV CHARGING");

lcd.setCursor(4,1);

lcd.print("STATION");

intGsm("6300134294","EV - CHARGING STATION");

delay(2000);

}

void loop()

{

while(1)

{

int i = 1;

int recharge\_amount = 0;

char buff2[15];

input\_read1 = analogRead(solar\_read);

input\_read2 = analogRead(mains\_read);

// Serial.println(input\_read);

SCAN.begin(9600);

lcd.clear();

lcd.setCursor(2,1);

lcd.print("SCAN YOUR CARD");

delay(1000);

SCAN.begin(9600);

// Serial.println("ENTERED");

while(SCAN.available()>0)

{

message=SCAN.readString();

Serial.println("ENTERED");

int str\_len = message.length() + 1;

char textmessage[11];

message.toCharArray(textmessage,str\_len);

Serial.println(textmessage);

textmessage[11]='\0';

if(!strcmp(textmessage,CARD\_1))

{

while(i)

{

sprintf(buff2,"amt in card:%d",card1\_balance);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("NAME: Shaik Moin");

lcd.setCursor(0,1);

lcd.print(buff2);

delay(3000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("PRESS 1 FOR CONFM");

lcd.setCursor(0,1);

lcd.print("PRESS D FOR rchrge");

limt\_flag1 =1;

customKey=0;

while(customKey==0)

{

customKey = customKeypad.getKey();

}

if(customKey == '1')

{

customKey=0;

IntToASCI();

intGsm("6300134294",buff);

delay(2000);

OTP\_checking();

Serial.println("out from charging");

i = 0;

}

if(customKey == 'D')

{

customKey=0;

recharge\_amount = recharge();

card1\_balance += recharge\_amount;

}

}

}

if(!strcmp(textmessage,CARD\_2))

{

while(i)

{

sprintf(buff2,"amt in card:%d",card2\_balance);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("NAME: Ahmed");

lcd.setCursor(0,1);

lcd.print(buff2);

delay(3000);

customKey=0;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("PRESS 1 FOR CONFM");;

lcd.setCursor(0,1);

lcd.print("PRESS D FOR rchrge");

limt\_flag2 =1;

while(customKey==0)

{

customKey = customKeypad.getKey();

}

if(customKey == '1')

{

customKey=0;

IntToASCI();

intGsm("9550964921",buff);

delay(2000);

OTP\_checking();

Serial.println("out from charging");

customKey=0;

i = 0;

}

if(customKey == 'D')

{

customKey=0;

recharge\_amount = recharge();

card2\_balance += recharge\_amount;

}

}

Serial.println("out");

}

}

}

}

void PASSWORD()

{

customKey=0;

while(customKey==0)

{

customKey = customKeypad.getKey();

}

Serial.println(customKey);

}

void clearData()

{

while(data\_count !=0)

{

Data[data\_count--] = 0;

}

return;

}

void IntToASCI()

{

clearData();

int i=0;

randomNumber = random(1000,9999);

mystr= String(randomNumber);

String one = mystr;

String message = one;

String str = "otp is ";

String all = str + message;

// Convert String to char array

int str\_len = message.length() + 1;

char textmessage[str\_len];

message.toCharArray(textmessage,str\_len);

strcpy(buff,textmessage);

Serial.println(buff);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("sending otp");

lcd.setCursor(0,1);

lcd.print("........ ");

}

void OTP\_checking()

{

clearData();

int k=1;

int p=0;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("ENTER OTP");

lcd.setCursor(0,1);

lcd.print(".......");

while(k<=4)

{

customKey = customKeypad.getKey();

if (customKey)

{

Data[data\_count] = customKey;

lcd.setCursor(p,2);

lcd.print(Data[data\_count]);

p++;

data\_count++;

}

if(data\_count == Password\_Length-1)

{

if(!strcmp(Data, buff))

{

lcd.clear();

lcd.print("Correct");

delay(2000);

charging();

clearData();

break;

}

else

{

if(k==3)

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("ATTEMPTED MORE TIMES");

// intGsm("7338546982","ATTEMPTED MORE TIMES");

customKey=0;

clearData();

loop();

delay(2000);

exit;

}

else

{

clearData();

k++;

p=0;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("INCORRECT PASSWORD");

delay(1000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("RE-ENTER PASSWORD");

delay(1000);

}

}

}

}

}

void buff\_clear()

{

for(int i=0;i<4;i++)

buff[i]=0;

}

void intGsm( char \*num1, char \* str1 )

{

char buff[10],i=0;

SMS.write('A');

delay(100);

SMS.write('T');

delay(100);

SMS.write('E');

delay(100);

SMS.write('0');

delay(100);

SMS.write('\r');

SMS.write("AT+CMGF=1\r"); //Initialize GSM For mobile

delay(500);

SMS.write("AT+CMGS=\"");

delay(500);

SMS.write(num1);

delay(500);

SMS.write("\"\r");

SMS.write(str1);

delay(500);

SMS.write(26);

delay(500);

Serial.println("meassage sent");

}

void charging()

{

Serial.print("Entered");

input\_read1 = analogRead(solar\_read);

input\_read2 = analogRead(mains\_read);

Serial.println(input\_read1);

if(input\_read1 > 500)

{

flag1 = 1;

charging\_flag =1;

billing();

}

else

{

flag1 = 1;

charging\_flag = 2;

billing();

}

}

void charging\_1()

{

input\_read1 = analogRead(solar\_read);

input\_read2 = analogRead(mains\_read);

if(input\_read1 > 500)

{

flag1 = 1;

charging1\_flag = 1;

limitted();

}

else

{

flag1 = 1;

charging1\_flag = 2;

limitted();

}

}

void limitted()

{

if(flag1 == 1)

{

char ch;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("2.limitted");

lcd.setCursor(0,1);

lcd.print("selected");

delay(2000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("1.30s");

lcd.setCursor(8,0);

lcd.print("2.1 Min");

lcd.setCursor(0,1);

lcd.print("3.5 Min");

lcd.setCursor(8,1);

lcd.print("4.30 Min");

delay(2000);

while(1)

{

ch = customKeypad.getKey();

switch(ch)

{

case '1':limit\_billing(30);

break;

case '2':limit\_billing(60);

break;

case '3':limit\_billing(300);

break;

case '4':limit\_billing(1800);

break;

}

if(ch=='1'||ch=='2'||ch=='3'||ch=='4')

{

delay(2000);

break;

}

}

}

}

void billing()

{

Serial.println("entered billing");

char ch;

int amount = 0;

if(flag1 == 1)

{

int count = 0;

int i = 1;

char buff1[15];

lcd.clear();

lcd.setCursor(0,0);

lcd.print("1.unlimitted");

lcd.setCursor(0,1);

lcd.print("2.limitted");

delay(2000);

while(1)

{

ch = customKeypad.getKey();

if(ch == '1')

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("1).unlimitted");

lcd.setCursor(0,1);

lcd.print("selected");

delay(2000);

lcd.clear();

if(charging\_flag == 1)

{

digitalWrite(relay\_pin2,HIGH);

digitalWrite(relay\_pin3,HIGH);

charging\_flag = 0;

}

if(charging\_flag == 2)

{

digitalWrite(relay\_pin1,HIGH);

digitalWrite(relay\_pin3,HIGH);

charging\_flag = 0;

}

lcd.print("billing.....");

lcd.setCursor(0,1);

lcd.print("TIME:");

while(i)

{

count++;

lcd.setCursor(6,1);

lcd.print(count);

lcd.print("s");

delay(1000);

if(SCAN.available()>0)

{

message=SCAN.readString();

int str\_len = message.length() + 1;

char textmessage[11];

message.toCharArray(textmessage,str\_len);

Serial.println(textmessage);

textmessage[11]='\0';

if(!strcmp(textmessage,CARD\_1))

{

amount = count \*1;

sprintf(buff1,"total amt =%d",amount);

lcd.clear();

lcd.setCursor(0,0);

lcd.print(buff1);

delay(2000);

lcd.clear();

card1\_balance -= amount;

i = 0;

flag1 = 0;

digitalWrite(relay\_pin1,LOW);

digitalWrite(relay\_pin3,LOW);

digitalWrite(relay\_pin2,LOW);

}

if(!strcmp(textmessage,CARD\_2) )

{

amount = count \*1;

sprintf(buff1,"total amt =%d",amount);

lcd.clear();

lcd.setCursor(0,0);

lcd.print(buff1);

delay(2000);

lcd.clear();

card2\_balance -=amount;

i = 0;

flag1 = 0;

digitalWrite(relay\_pin1,LOW);

digitalWrite(relay\_pin3,LOW);

digitalWrite(relay\_pin2,LOW);

}

}

}

break;

}

else if(ch == '2')

{

charging\_1();

break;

}

}

}

}

int recharge()

{

char Data1[10];

int data\_count1 = 0;

clearData();

int k=1,num;

int p=0;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("enter recharge amount:");

while(k)

{

customKey = customKeypad.getKey();

if (customKey && customKey != 'D')

{

Data1[data\_count1] = customKey;

lcd.setCursor(p,1);

lcd.print(Data1[data\_count1]);

p++;

data\_count1++;

}

if (customKey == 'D')

{

Serial.println(Data1);

num = atoi(Data1);

k = 0;

}

}

return num;

}

void limit\_billing(int t)

{

int amount = 0;

int count = 0;

char buff1[15];

if(charging1\_flag == 1)

{

digitalWrite(relay\_pin2,HIGH);

digitalWrite(relay\_pin3,HIGH);

charging1\_flag = 0 ;

}

if(charging1\_flag == 2)

{

digitalWrite(relay\_pin1,HIGH);

digitalWrite(relay\_pin3,HIGH);

charging1\_flag = 0;

}

lcd.clear();

lcd.print("billing.....");

lcd.setCursor(0,1);

lcd.print("TIME:");

while(count < t)

{

count++;

lcd.setCursor(6,1);

lcd.print(count);

lcd.print("s");

delay(1000);

}

amount = count \*1;

sprintf(buff1,"total amt =%d",amount);

lcd.clear();

lcd.setCursor(0,0);

lcd.print(buff1);

delay(2000);

lcd.clear();

if(limt\_flag1 == 1)

{

card1\_balance -= amount;

Serial.print(card1\_balance);

limt\_flag1 =0;

}

else if(limt\_flag2 == 1)

{

card2\_balance -= amount;

limt\_flag2 =0;

}

flag1 = 0;

digitalWrite(relay\_pin1,LOW);

digitalWrite(relay\_pin3,LOW);

digitalWrite(relay\_pin2,LOW);

}