#include <Password.h>

Password pass = Password("1234");

Password pass\_m = Password("0000");

int c = 0;

int alarm = 0;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <Keypad.h>

const byte ROWS = 4;

const byte COLS = 4;

char keys[ROWS][COLS] = {

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

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

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

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

};

byte rowPins[ROWS] = { 22, 24, 26, 28 };

byte colPins[COLS] = { 30, 32, 34, 36 };

Keypad koko = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

char clik;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <LiquidCrystal\_I2C.h>

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

unsigned long lcd\_delay = 0;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <Servo.h>

Servo servo1;

Servo servo2;

Servo servo3;

#define servo1\_pin 3

#define servo2\_pin 4

#define servo3\_pin 5

#define led1\_pin 6

#define led2\_pin 7

#define led3\_pin 8

#define btn1\_pin 9

#define btn2\_pin 10

#define btn3\_pin 11

#define buzz 48

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#define sim Serial1

#include <Adafruit\_Fingerprint.h>

Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&Serial2);

uint8\_t id;

unsigned long page\_delay = 0;

int page = 1;

int del = 100;

String number = "+96893261458";

#include <Wire.h>

#include "RTClib.h"

RTC\_DS1307 rtc;

unsigned long sec\_del = 0;

int main\_h = 0;

int main\_m = 0;

int main\_s = 0;

int main\_D = 0;

int main\_M = 0;

int main\_Y = 0;

////////////////////////////////

// when get pass or finger right

bool medicine\_1\_allowed\_to\_open = false;

bool medicine\_2\_allowed\_to\_open = false;

bool medicine\_3\_allowed\_to\_open = false;

// when time hit

bool medicine\_1\_time = false;

bool medicine\_2\_time = false;

bool medicine\_3\_time = false;

// medicine time

int main\_h1 = 19;

int main\_m1 = 26;

int main\_s1 = 0;

int main\_D1 = 26;

int main\_M1 = 5;

int main\_Y1 = 2024;

int main\_h2 = 19;

int main\_m2 = 13;

int main\_s2 = 0;

int main\_D2 = 1;

int main\_M2 = 5;

int main\_Y2 = 2024;

int main\_h3 = 21;

int main\_m3 = 43;

int main\_s3 = 0;

int main\_D3 = 14;

int main\_M3 = 4;

int main\_Y3 = 2024;

void setup() {

Serial.begin(9600);

finger.begin(57600);

sim.begin(9600);

rtc.begin();

//rtc.adjust(DateTime(2024, 5, 26, 19, 1, 0));

pinMode(btn1\_pin, INPUT\_PULLUP);

pinMode(btn2\_pin, INPUT\_PULLUP);

pinMode(btn3\_pin, INPUT\_PULLUP);

pinMode(led1\_pin, OUTPUT);

pinMode(led2\_pin, OUTPUT);

pinMode(led3\_pin, OUTPUT);

servo1.attach(servo1\_pin);

servo2.attach(servo2\_pin);

servo3.attach(servo3\_pin);

servo1.write(0);

servo2.write(7);

servo3.write(0);

pinMode(buzz, OUTPUT);

lcd.init();

lcd.backlight();

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(" SMART ");

lcd.setCursor(0, 1);

lcd.print(" PHARMACY ");

delay(2000);

lcd.clear();

//send\_msg("hii", 1);

page\_delay = millis();

}

void loop() {

get\_time\_24();

check\_medicine();

buttons\_clicked();

//change page every 3 sec

if (millis() - page\_delay > 3000) {

page++;

if (page > 3) page = 1;

page\_delay = millis();

}

//update page every 1 sec

if (millis() - lcd\_delay > 1000) {

lcd.clear();

if (page == 1) {

lcd.setCursor(0, 0);

lcd.print("Time : ");

lcd.print(main\_h);

lcd.print(":");

lcd.print(main\_m);

lcd.print(":");

lcd.print(main\_s);

lcd.setCursor(0, 1);

lcd.print("Date :");

lcd.print(main\_D);

lcd.print("/");

lcd.print(main\_M);

lcd.print("/");

lcd.print(main\_Y);

} else if (page == 2) {

lcd.setCursor(0, 0);

lcd.print("A-Enter Sensor");

lcd.setCursor(0, 1);

lcd.print("B-Enter Password");

} else if (page == 3) {

lcd.setCursor(0, 0);

lcd.print("C-Clear Database");

lcd.setCursor(0, 1);

lcd.print("D-Add Print");

}

lcd\_delay = millis();

}

// wait for any keypad clicked

clik = koko.getKey();

if (clik) {

Serial.println(clik);

digitalWrite(buzz, 1);

delay(del);

digitalWrite(buzz, 0);

lcd.clear();

if (clik == 'A') {

enter\_finger();

} else if (clik == 'B') {

pass\_();

} else if (clik == 'C') {

if (pass\_f()) clear\_finger();

} else if (clik == 'D') {

if (pass\_f()) add\_finger();

} else if (clik == '\*') {

//servo.write(90);

}

delay(100);

lcd.clear();

}

///

digitalWrite(led1\_pin, medicine\_1\_time);

digitalWrite(led2\_pin, medicine\_2\_time);

digitalWrite(led3\_pin, medicine\_3\_time);

}

void enter\_finger() {

//lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Your Finger PLS!");

while (getFingerprintIDez() < 0) {

clik = koko.getKey();

if (clik) {

digitalWrite(buzz, 1);

delay(del);

digitalWrite(buzz, 0);

delay(del);

if (clik == 'B') return;

}

}

lcd.setCursor(0, 1);

lcd.print("Print accepted!");

delay(3000);

allow\_to\_open();

}

void add\_finger() {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Enter ID : ");

lcd.setCursor(0, 1);

int k = 0;

while (1) {

clik = koko.getKey();

if (clik) {

digitalWrite(buzz, 1);

delay(del);

digitalWrite(buzz, 0);

delay(del);

if ((clik == '1' || clik == '2' || clik == '3' || clik == '4' || clik == '5' || clik == '6' || clik == '7' || clik == '8' || clik == '9') & k == 0) {

k++;

lcd.setCursor(7, 1);

lcd.print(clik);

id = clik - '0';

}

if (clik == 'A' & k > 0) break;

if (clik == 'B') return;

if (clik == 'C') add\_finger();

}

}

getFingerprintEnroll();

delay(2000);

}

void clear\_finger() {

zz:

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Select mode : ");

lcd.setCursor(0, 1);

lcd.print("A-All\_Data C-ID");

while (1) {

clik = koko.getKey();

if (clik) {

digitalWrite(buzz, 1);

delay(del);

digitalWrite(buzz, 0);

delay(del);

if (clik == 'A') {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Confirm : ");

lcd.setCursor(0, 1);

lcd.print("A-Yes B-No");

while (1) {

clik = koko.getKey();

if (clik) {

digitalWrite(buzz, 1);

delay(del);

digitalWrite(buzz, 0);

delay(del);

if (clik == 'A') {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Deleting all...");

delay(2000);

finger.emptyDatabase();

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Done!");

delay(1000);

return;

}

if (clik == 'B') goto zz;

}

}

}

if (clik == 'B') return;

if (clik == 'C') {

z:

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Enter ID :");

lcd.setCursor(0, 1);

int k = 0;

while (1) {

clik = koko.getKey();

if (clik) {

digitalWrite(buzz, 1);

delay(del);

digitalWrite(buzz, 0);

delay(del);

if ((clik == '1' || clik == '2' || clik == '3' || clik == '4' || clik == '5' || clik == '6' || clik == '7' || clik == '8' || clik == '9') & k == 0) {

k++;

lcd.setCursor(7, 1);

lcd.print(clik);

id = clik - '0';

}

if (clik == 'A' & k > 0) break;

if (clik == 'B') goto zz;

if (clik == 'C') goto z;

}

}

delay(1000);

deleteFingerprint(id);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Select mode : ");

lcd.setCursor(0, 1);

lcd.print("A-All\_Data C-ID");

}

}

}

}

void pass\_() {

while (c < 4) {

//lcd.clear();

lcd.setCursor(0, 0);

lcd.print(" Enter Password ");

clik = koko.getKey();

if (clik) {

digitalWrite(buzz, 1);

delay(del);

digitalWrite(buzz, 0);

delay(del);

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

lcd.setCursor(c + 6, 1);

lcd.print("\*");

pass.append(clik);

c++;

delay(100);

}

if (clik == 'B') return;

}

}

delay(1000);

if (pass.evaluate()) {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Password Correct");

delay(2000);

pass.reset();

c = 0;

lcd.clear();

alarm = 0;

allow\_to\_open();

return;

} else {

lcd.clear();

delay(200);

lcd.setCursor(0, 0);

lcd.print(" Wrong Password ");

alarm++;

delay(2000);

pass.reset();

c = 0;

lcd.clear();

if (alarm > 2) {

unsigned long m = millis();

delay(50);

while (millis() - m < 10000) {

lcd.setCursor(0, 0);

lcd.print("System Is Locked");

lcd.setCursor(7, 1);

lcd.print(10 - ((millis() - m) / 1000));

lcd.print(" ");

}

return;

}

pass\_();

}

}

bool pass\_f() {

while (c < 4) {

//lcd.clear();

lcd.setCursor(0, 0);

lcd.print(" Enter Password ");

clik = koko.getKey();

if (clik) {

digitalWrite(buzz, 1);

delay(del);

digitalWrite(buzz, 0);

delay(del);

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

lcd.setCursor(c + 6, 1);

lcd.print("\*");

pass\_m.append(clik);

c++;

delay(100);

}

if (clik == 'B') return 0;

}

}

delay(1000);

if (pass\_m.evaluate()) {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Password Correct");

delay(2000);

pass\_m.reset();

c = 0;

lcd.clear();

return 1;

} else {

lcd.clear();

delay(200);

lcd.setCursor(0, 0);

lcd.print(" Wrong Password ");

delay(2000);

pass\_m.reset();

c = 0;

lcd.clear();

pass\_f();

}

}

uint8\_t getFingerprintEnroll() {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Put your finger:");

lcd.setCursor(0, 1);

//delay(200);

int p = -1; // -1 to make the while condition always true

while (p != FINGERPRINT\_OK) { //FINGERPRINT\_OK==0x00 :: the condition break when p=FINGERPRINT\_OK

p = finger.getImage(); // get image from sensor right now and it's return the result

switch (p) {

case FINGERPRINT\_OK: //0x00

//Serial.println("Image taken");

//lcd.clear();

lcd.print("Image taken");

break;

case FINGERPRINT\_NOFINGER: //0x02

//Serial.println(".");

break;

default:

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

lcd.print("Error");

break;

}

}

delay(500);

//lcd.clear();

//lcd.print("Remove finger");

//delay(2000);

lcd.clear();

// OK success!

p = finger.image2Tz(1); // 1 to to convert and store :: 2 to convert and verify (double cheak)

// if ok continuo else exit from the hole function and restart the void loop

switch (p) {

case FINGERPRINT\_OK:

//Serial.println("Image converted");

lcd.print("Image converted");

break;

default:

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

lcd.print("Error");

return p;

}

delay(1000);

//Serial.println("Remove finger");

lcd.clear();

lcd.print("Remove finger");

delay(2000);

p = 0;

// the while is wait the user to unput his finger from sensor

delay(500);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Verify Finger:");

lcd.setCursor(0, 1);

// Serial.print("ID "); Serial.println(id);

p = -1;

// double cheak the finger read and verfiy

//Serial.println("Place same finger again");

while (p != FINGERPRINT\_OK) {

p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

//Serial.println("Image taken");

lcd.setCursor(0, 1);

lcd.print("Image taken");

break;

case FINGERPRINT\_NOFINGER:

// Serial.print(".");

break;

default:

lcd.setCursor(0, 1);

lcd.print("Error");

break;

}

}

delay(500);

lcd.clear();

//lcd.print("Remove finger");

//delay(2000);

// OK success!

lcd.clear();

p = finger.image2Tz(2); // converted as verefy not to store

switch (p) {

case FINGERPRINT\_OK:

//Serial.println("Image converted");

lcd.print("Image converted");

break;

default:

lcd.setCursor(0, 0);

lcd.print("Error");

return p;

}

delay(1000);

lcd.clear();

// OK converted!

// Serial.print("Creating model for #"); Serial.println(id);

// to create model it must match the first fingerprint and the double cheak

p = finger.createModel();

if (p == FINGERPRINT\_OK) {

//Serial.println("Prints matched!");

// lcd.clear();

lcd.print("Prints matched!");

//myservo.write(0);

} else if (p == FINGERPRINT\_ENROLLMISMATCH) {

// Serial.println("Fingerprints did not match");

lcd.print("Not matched!");

return p;

} else {

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

lcd.print("Error");

return p;

}

delay(1500);

lcd.clear();

// Serial.print("ID "); Serial.println(id);

//if the model was created now we want to stor it in the choosen id

p = finger.storeModel(id);

if (p == FINGERPRINT\_OK) {

//Serial.println("Stored!");

lcd.print("Stored!");

} else {

lcd.print("error");

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

return p;

}

delay(1000);

}

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;

return finger.fingerID;

}

uint8\_t deleteFingerprint(uint8\_t id) {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Deleting ID: ");

lcd.print(id);

delay(500);

lcd.setCursor(0, 1);

uint8\_t p = -1; //for if condition

p = finger.deleteModel(id);

if (p == FINGERPRINT\_OK) {

lcd.print("Deleted");

} else {

lcd.print("error");

return p;

}

delay(2000);

}

void get\_time\_24() {

if (millis() - sec\_del > 500) {

DateTime now = rtc.now();

main\_h = now.hour();

main\_m = now.minute();

main\_s = now.second();

main\_Y = now.year();

main\_M = now.month();

main\_D = now.day();

sec\_del = millis();

}

}

void check\_medicine() {

if (main\_Y == main\_Y1 && main\_M == main\_M1 && main\_D == main\_D1 && main\_h == main\_h1 && main\_m == main\_m1 && medicine\_1\_time == false) {

medicine\_1\_time = true;

Serial.println("medicine\_1\_time = true");

send\_msg("it's time for medicine 1 , expire date : 30/9/2024", 1);

}

if (medicine\_1\_time == true && main\_m > (main\_m1 + 5)) {

medicine\_1\_time = false;

Serial.println("medicine\_1\_time = false");

}

//////

if (main\_Y == main\_Y2 && main\_M == main\_M2 && main\_D == main\_D2 && main\_h == main\_h2 && main\_m == main\_m2 && medicine\_2\_time == false) {

medicine\_2\_time = true;

Serial.println("medicine\_2\_time = true");

send\_msg("it's time for medicine 2 , expire date : 25/10/2026", 2);

}

if (medicine\_2\_time == true && main\_m > (main\_m2 + 5)) {

medicine\_2\_time = false;

Serial.println("medicine\_2\_time = false");

}

///////

if (main\_Y == main\_Y3 && main\_M == main\_M3 && main\_D == main\_D3 && main\_h == main\_h3 && main\_m == main\_m3 && medicine\_3\_time == false) {

medicine\_3\_time = true;

Serial.println("medicine\_3\_time = true");

send\_msg("it's time for medicine 3 , expire date : 1/1/2025", 3);

}

if (medicine\_3\_time == true && main\_m > (main\_m3 + 5)) {

medicine\_3\_time = false;

Serial.println("medicine\_3\_time = false");

}

}

void allow\_to\_open() {

if (medicine\_1\_time == true && medicine\_1\_allowed\_to\_open == false) {

medicine\_1\_allowed\_to\_open = true;

Serial.println("medicine\_1\_allowed\_to\_open = true");

}

if (medicine\_2\_time == true && medicine\_2\_allowed\_to\_open == false) {

medicine\_2\_allowed\_to\_open = true;

Serial.println("medicine\_2\_allowed\_to\_open = true");

}

if (medicine\_3\_time == true && medicine\_3\_allowed\_to\_open == false) {

medicine\_3\_allowed\_to\_open = true;

Serial.println("medicine\_3\_allowed\_to\_open = true");

}

}

void buttons\_clicked() {

if (digitalRead(btn1\_pin) == LOW) {

Serial.println("btn1 clicked");

if (medicine\_1\_allowed\_to\_open) {

servo1.write(90);

medicine\_1\_allowed\_to\_open = false;

medicine\_1\_time = false;

Serial.println("door1 open");

} else {

servo1.write(0);

Serial.println("door1 close");

}

delay(500);

}

if (digitalRead(btn2\_pin) == LOW) {

Serial.println("btn2 clicked");

if (medicine\_2\_allowed\_to\_open) {

servo2.write(90);

medicine\_2\_allowed\_to\_open = false;

medicine\_2\_time = false;

Serial.println("door2 open");

} else {

servo2.write(7);

Serial.println("door2 close");

}

delay(500);

}

if (digitalRead(btn3\_pin) == LOW) {

Serial.println("btn3 clicked");

if (medicine\_3\_allowed\_to\_open) {

servo3.write(90);

medicine\_3\_allowed\_to\_open = false;

medicine\_3\_time = false;

Serial.println("door3 open");

} else {

servo3.write(0);

Serial.println("door3 close");

}

delay(500);

}

}

void send\_msg(String msg, int mid\_num) {

digitalWrite(buzz, 1);

sim.println("AT+CMGF=1");

delay(1000);

sim.println("AT+CMGS=\"" + number + "\"\r");

delay(1000);

sim.println(msg);

delay(1000);

sim.println((char)26);

lcd.init();

lcd.backlight();

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(" It is Time For ");

lcd.setCursor(0, 1);

lcd.print(" medicine ");

lcd.print(mid\_num);

digitalWrite(buzz, 0);

delay(1000);

}