**Source Code**

**#include <LiquidCrystal\_I2C.h>**

**#include <Keypad.h>**

**#include <RTClib.h>**

**#include <Servo.h>**

**#include <HX711\_ADC.h>**

**#include <Adafruit\_NeoPixel.h>**

**#include <SoftwareSerial.h>**

**SoftwareSerial SIM800(1, 0);**

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

**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, 26, 28, 34 };**

**byte colPins[COLS] = { 38, 42, 44, 50 };**

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

**RTC\_DS3231 rtc;**

**Servo servos[5];**

**const int buzzerPin = 9;**

**bool doorsOpen[5] = { false, false, false, false, false };**

**// Unique open and close positions for each servo**

**int servoOpenPositions[5] = { 0, 180, 170, 0, 90 };**

**int servoClosePositions[5] = { 100, 80, 70, 90, 0 };**

**HX711\_ADC LoadCell2(43, 41);**

**HX711\_ADC LoadCell3(39, 37);**

**HX711\_ADC LoadCell4(35, 33);**

**float initialWeights[3] = { 0.0, 0.0, 0.0 };**

**int timesPerDay[5];**

**int hours[5][10];**

**int minutes[5][10];**

**int maxTemperature[5];**

**bool medicineReady[5] = { false, false, false, false, false };**

**Adafruit\_NeoPixel strip = Adafruit\_NeoPixel(4, 47, NEO\_GRB + NEO\_KHZ800);**

**enum State {**

**WELCOME,**

**REFILL,**

**SET\_CONTAINER\_DATA,**

**SET\_TIMES,**

**SET\_TIME\_VALUES,**

**SET\_MAX\_TEMP,**

**DISPLAY\_TEMPERATURE,**

**CHECK\_MEDICINE\_TIME,**

**MEDICINE\_READY**

**} state = WELCOME;**

**int currentContainer = 0;**

**int currentTimeIndex = 0;**

**void setup() {**

**Serial.begin(9600);**

**SIM800.begin(9600);**

**lcd.init();**

**lcd.backlight();**

**pinMode(buzzerPin, OUTPUT);**

**strip.begin();**

**strip.show();  // Initialize all pixels to 'off'**

**if (!rtc.begin()) {**

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

**lcd.print("Couldn't find RTC");**

**while (1)**

**;**

**}**

**// Attach servos to the specified pins**

**servos[0].attach(4);**

**servos[1].attach(8);**

**servos[2].attach(6);**

**servos[3].attach(3);**

**servos[4].attach(2);**

**for (int i = 0; i < 5; i++) {**

**servos[i].write(servoClosePositions[i]);**

**}**

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

**lcd.print("Welcome!");**

**delay(4000);**

**float temperature = rtc.getTemperature();**

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

**lcd.print("Room Temp:");**

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

**lcd.print(temperature);**

**lcd.print("C");**

**delay(4000);**

**LoadCell2.begin();**

**LoadCell2.start(2000);**

**LoadCell2.setCalFactor(1020);**

**LoadCell3.begin();**

**LoadCell3.start(2000);**

**LoadCell3.setCalFactor(1090);**

**LoadCell4.begin();**

**LoadCell4.start(2000);**

**LoadCell4.setCalFactor(-920);**

**if (rtc.lostPower()) {**

**lcd.clear();**

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

**lcd.print("Enter Date&Time:");**

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

**lcd.print("DDMMYYYYHHMMSS");**

**String dateTimeInput = "";**

**while (dateTimeInput.length() < 14) {**

**char key = keypad.getKey();**

**if (key) {**

**if (key == '\*') {**

**dateTimeInput = "";**

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

**lcd.print("DDMMYYYYHHMMSS");**

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

**} else {**

**dateTimeInput += key;**

**lcd.setCursor(dateTimeInput.length() - 1, 1);**

**lcd.print(key);**

**}**

**}**

**}**

**int day = dateTimeInput.substring(0, 2).toInt();**

**int month = dateTimeInput.substring(2, 4).toInt();**

**int year = dateTimeInput.substring(4, 8).toInt();**

**int hour = dateTimeInput.substring(8, 10).toInt();**

**int minute = dateTimeInput.substring(10, 12).toInt();**

**int second = dateTimeInput.substring(12, 14).toInt();**

**rtc.adjust(DateTime(year, month, day, hour, minute, second));**

**}**

**lcd.clear();**

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

**lcd.print("1-5 to open/close");**

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

**lcd.print("A to proceed");**

**}**

**void loop() {**

**switch (state) {**

**case WELCOME:**

**{**

**char key = keypad.getKey();**

**if (key >= '1' && key <= '5') {**

**int containerIndex = key - '1';**

**if (doorsOpen[containerIndex]) {**

**lcd.clear();**

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

**lcd.print("Container ");**

**lcd.print(containerIndex + 1);**

**lcd.print(" Closed");**

**moveServoSmoothly(servos[containerIndex], servoClosePositions[containerIndex]);**

**delay(1000);**

**lcd.clear();**

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

**lcd.print("1-5 to open/close");**

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

**lcd.print("A to proceed");**

**} else {**

**lcd.clear();**

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

**lcd.print("Container ");**

**lcd.print(containerIndex + 1);**

**lcd.print(" Opened");**

**moveServoSmoothly(servos[containerIndex], servoOpenPositions[containerIndex]);**

**delay(1000);**

**lcd.clear();**

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

**lcd.print("1-5 to open/close");**

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

**lcd.print("A to proceed");**

**}**

**doorsOpen[containerIndex] = !doorsOpen[containerIndex];**

**} else if (key == 'A') {**

**if (anyContainerOpen()) {**

**for (int i = 0; i < 5; i++) {**

**lcd.clear();**

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

**lcd.print("All containers");**

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

**lcd.print("closed");**

**if (doorsOpen[i]) {**

**moveServoSmoothly(servos[i], servoClosePositions[i]);**

**doorsOpen[i] = false;**

**}**

**}**

**delay(1000);**

**}**

**state = SET\_CONTAINER\_DATA;**

**}**

**checkMedicineQuantity();**

**break;**

**}**

**case SET\_CONTAINER\_DATA:**

**{**

**lcd.clear();**

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

**lcd.print("Container ");**

**lcd.print(currentContainer + 1);**

**delay(1000);**

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

**lcd.print("Times per day:");**

**timesPerDay[currentContainer] = getNumericInput(1);**

**currentTimeIndex = 0;**

**state = SET\_TIMES;**

**break;**

**}**

**case SET\_TIMES:**

**{**

**if (currentTimeIndex < timesPerDay[currentContainer]) {**

**lcd.clear();**

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

**lcd.print("Time ");**

**lcd.print(currentTimeIndex + 1);**

**delay(1000);**

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

**lcd.print("Hour:");**

**hours[currentContainer][currentTimeIndex] = getNumericInput(2);**

**lcd.clear();**

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

**lcd.print("Minute:");**

**minutes[currentContainer][currentTimeIndex] = getNumericInput(2);**

**currentTimeIndex++;**

**} else {**

**state = SET\_MAX\_TEMP;**

**}**

**break;**

**}**

**case SET\_MAX\_TEMP:**

**{**

**lcd.clear();**

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

**lcd.print("Container ");**

**lcd.print(currentContainer + 1);**

**lcd.print(" Temp:");**

**maxTemperature[currentContainer] = getNumericInput(2);**

**currentContainer++;**

**if (currentContainer < 5) {**

**state = SET\_CONTAINER\_DATA;**

**} else {**

**lcd.clear();**

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

**lcd.print("Setup Complete");**

**state = DISPLAY\_TEMPERATURE;**

**}**

**break;**

**}**

**case DISPLAY\_TEMPERATURE:**

**{**

**lcd.clear();**

**while (state == DISPLAY\_TEMPERATURE) {**

**DateTime now = rtc.now();**

**float temperature = rtc.getTemperature();**

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

**lcd.print("Room Temp:");**

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

**lcd.print(temperature);**

**lcd.print("C");**

**bool highTemperature = false;**

**for (int i = 0; i < 5; i++) {**

**if (temperature > maxTemperature[i]) {**

**highTemperature = true;**

**break;**

**}**

**}**

**if (highTemperature) {**

**lcd.clear();**

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

**lcd.print("High Temperature");**

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

**lcd.print(temperature);**

**lcd.print("C");**

**setAllLEDsColor(255, 0, 0);  // Red for high temperature**

**digitalWrite(buzzerPin, HIGH);**

**delay(500);**

**digitalWrite(buzzerPin, LOW);**

**delay(500);**

**} else {**

**setAllLEDsColor(0, 0, 0);  // Turn off the LED**

**}**

**checkMedicineTimes(now);**

**if (state == DISPLAY\_TEMPERATURE) {**

**delay(30000);  // Check every 30 seconds**

**}**

**}**

**break;**

**}**

**case MEDICINE\_READY:**

**{**

**lcd.clear();**

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

**lcd.print("Your medicine");**

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

**lcd.print("is ready!");**

**// Store initial weights when the doors open**

**if (doorsOpen[1] && LoadCell2.update()) {**

**initialWeights[0] = LoadCell2.getData();**

**}**

**if (doorsOpen[2] && LoadCell3.update()) {**

**initialWeights[1] = LoadCell3.getData();**

**}**

**if (doorsOpen[3] && LoadCell4.update()) {**

**initialWeights[2] = LoadCell4.getData();**

**}**

**setAllLEDsColor(0, 255, 0);  // Green for medicine ready**

**for (int i = 0; i < 10; i++) {**

**digitalWrite(buzzerPin, HIGH);**

**delay(500);**

**digitalWrite(buzzerPin, LOW);**

**delay(500);**

**}**

**long start = millis();**

**while (millis() - start < 1800000) {  // 30 minutes**

**char key = keypad.getKey();**

**if (key == 'A') {**

**if (anyContainerOpen()) {**

**lcd.clear();**

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

**lcd.print("Doors Closed");**

**for (int i = 0; i < 5; i++) {**

**if (doorsOpen[i] && medicineReady[i]) {**

**moveServoSmoothly(servos[i], servoClosePositions[i]);**

**doorsOpen[i] = false;**

**}**

**}**

**setAllLEDsColor(0, 0, 0);  // Turn off the LED**

**delay(1000);**

**state = DISPLAY\_TEMPERATURE;**

**break;**

**} else {**

**lcd.clear();**

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

**lcd.print("Opened");**

**for (int i = 0; i < 5; i++) {**

**if (medicineReady[i]) {**

**moveServoSmoothly(servos[i], servoOpenPositions[i]);**

**doorsOpen[i] = true;**

**}**

**}**

**}**

**}**

**checkMedicineQuantity();**

**}**

**if (state == MEDICINE\_READY) {**

**if (anyContainerOpen()) {**

**for (int i = 0; i < 5; i++) {**

**if (doorsOpen[i] && medicineReady[i]) {**

**moveServoSmoothly(servos[i], servoClosePositions[i]);**

**doorsOpen[i] = false;**

**}**

**}**

**lcd.clear();**

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

**lcd.print("Doors Closed");**

**delay(1000);**

**setAllLEDsColor(0, 0, 0);  // Turn off the LED**

**state = DISPLAY\_TEMPERATURE;**

**break;**

**}**

**}**

**break;**

**}**

**}**

**}**

**void moveServoSmoothly(Servo &servo, int targetPosition) {**

**int currentPosition = servo.read();**

**int step = currentPosition < targetPosition ? 1 : -1;**

**while (currentPosition != targetPosition) {**

**currentPosition += step;**

**servo.write(currentPosition);**

**delay(15);  // Adjust delay for smoother motion, increase for slower, smoother motion**

**}**

**}**

**int getNumericInput(int numDigits) {**

**String input = "";**

**while (input.length() < numDigits) {**

**char key = keypad.getKey();**

**if (key) {**

**if (key == '\*') {**

**input = "";**

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

**lcd.print(String(numDigits, ' '));**

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

**} else if (key >= '0' && key <= '9') {**

**input += key;**

**lcd.setCursor(input.length() - 1, 1);**

**lcd.print(key);**

**}**

**}**

**}**

**return input.toInt();**

**}**

**void checkMedicineTimes(DateTime now) {**

**for (int i = 0; i < 5; i++) {**

**medicineReady[i] = false;**

**}**

**for (int i = 0; i < 5; i++) {**

**for (int j = 0; j < timesPerDay[i]; j++) {**

**if (now.hour() == hours[i][j] && now.minute() == minutes[i][j]) {**

**medicineReady[i] = true;**

**}**

**}**

**}**

**for (int i = 0; i < 5; i++) {**

**if (medicineReady[i]) {**

**state = MEDICINE\_READY;**

**sendSMS("+94757507441", "Your medicine is ready now!");**

**return;**

**}**

**}**

**}**

**void checkMedicineQuantity() {**

**for (int i = 1; i < 4; i++) {**

**if (doorsOpen[i]) {**

**if (i == 1) {**

**if (LoadCell2.update()) {**

**float weight = LoadCell2.getData();**

**weightCheck(weight, i, initialWeights[0]);**

**}**

**} else if (i == 2) {**

**if (LoadCell3.update()) {**

**float weight = LoadCell3.getData();**

**weightCheck(weight, i, initialWeights[1]);**

**}**

**} else if (i == 3) {**

**if (LoadCell4.update()) {**

**float weight = LoadCell4.getData();**

**weightCheck(weight, i, initialWeights[2]);**

**}**

**}**

**}**

**}**

**}**

**void weightCheck(float weight, int i, float &initialWeight) {**

**if (weight < 5.0) {**

**lcd.clear();**

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

**lcd.print("Container ");**

**lcd.print(i + 1);**

**lcd.print(" has");**

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

**lcd.print("less medicine");**

**digitalWrite(buzzerPin, HIGH);**

**setAllLEDsColor(0, 0, 255);  // Blue for less medicine**

**delay(500);**

**digitalWrite(buzzerPin, LOW);**

**setAllLEDsColor(0, 0, 0);  // Turn off the LED**

**} else if (fabs(weight - initialWeight) > 2.0 && state == MEDICINE\_READY) {**

**lcd.clear();**

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

**lcd.print("Medicine ");**

**lcd.print(i + 1);**

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

**lcd.print("taken");**

**digitalWrite(buzzerPin, LOW);**

**setAllLEDsColor(255, 0, 255);  // Green for medicine taken**

**delay(2000);**

**initialWeight = weight;**

**} else {**

**digitalWrite(buzzerPin, LOW);**

**setAllLEDsColor(0, 0, 0);  // Turn off the LED**

**if (state == WELCOME) {**

**lcd.clear();**

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

**lcd.print("1-5 to open/close");**

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

**lcd.print("A to proceed");**

**} else if (state == MEDICINE\_READY) {**

**lcd.clear();**

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

**lcd.print("Opened");**

**}**

**delay(1000);**

**}**

**}**

**bool anyContainerOpen() {**

**for (int i = 0; i < 5; i++) {**

**if (doorsOpen[i]) {**

**return true;**

**}**

**}**

**return false;**

**}**

**void setAllLEDsColor(uint8\_t red, uint8\_t green, uint8\_t blue) {**

**for (int i = 0; i < strip.numPixels(); i++) {**

**strip.setPixelColor(i, strip.Color(red, green, blue));**

**}**

**strip.show();**

**}**

**void updateSerial() {**

**delay(500);**

**while (Serial.available()) {**

**SIM800.write(Serial.read());**

**}**

**while (SIM800.available()) {**

**Serial.write(SIM800.read());**

**}**

**}**

**void sendSMS(String phoneNumber, String message) {**

**SIM800.println("AT+CMGS=\"" + phoneNumber + "\"");**

**updateSerial();**

**delay(100);**

**SIM800.print(message);**

**updateSerial();**

**delay(100);**

**SIM800.write(26);**

**updateSerial();**

**delay(100);**

**}**