



دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

پروژه‌ی درس آزمایشگاه سخت‌افزار

عنوان:

مجموعه مستندات سیستم اندازه‌گیری علائم حیاتی بیمار

نگارندهان:

علیرضا تاج‌میرریاحی، امیرمهدی نامجو، صبا هاشمی

استاد گرامی:

جناب آقای دکتر اجلالی - جناب آقای دکتر فصحتی

زمستان ۱۴۰۰، بهار و تابستان ۱۴۰۱

اللهُ أَكْبَرُ

فهرست مطالب

۸	۱ مقدمه
۹	۲ دیتاشیت محصول
۱۰	۳ معماری سیستم
۱۲	۱-۳ طراحی و پیاده‌سازی سخت‌افزار
۱۲	۱-۱-۳ سنسورهای محیطی
۱۵	۱-۲ سنسورهای بدن
۲۳	۱-۳ کدهای سخت‌افزار
۲۴	۱-۴ بسته بندی
۲۶	۲-۳ طراحی و پیاده‌سازی سرور
۲۸	۳-۳ طراحی و پیاده‌سازی اپلیکیشن
۲۸	۱-۳-۳ مقدمه
۲۸	۲-۳-۳ تصاویری از محیط نرم‌افزار
۲۸	۳-۳-۳ توضیحاتی در ارتباط با کد برنامه
۳۰	۴ قیمت

فهرست شکل‌ها

۱۱	۱-۳	معماری سطح بالای سیستم
۱۳	۲-۳	اتصال سنسور DHT11
۱۳	۳-۳	پین‌های سنسور DHT11
۱۴	۴-۳	اتصال سنسور MQ135
۱۵	۵-۳	پین‌های مژول سنسور MQ135
۱۶	۶-۳	اتصال سنسور Max30205
۱۷	۷-۳	پین‌های سنسور MAX30205
۱۸	۸-۳	اتصال سنسور Max30102
۱۹	۹-۳	پین‌های سنسور MAX30102
۲۰	۱۰-۳	نحوه اتصال Lead های سنسور نوار قلب به بدن
۲۰	۱۱-۳	اتصال سنسورهای بدن به رزبری‌پای و آردوینو در کنار هم
۲۱	۱۲-۳	پین‌های سنسور AD8232
۲۲	۱۳-۳	فشارسنج
۲۵	۱۴-۳	بسته‌بندی طراحی شده در نرم‌افزار Tinkercad
۲۵	۱۵-۳	محصول به همراه تمامی سنسورها
۲۷	۱۶-۳	مستندات Swagger

۲۹ ۱۷-۳ تصاویری از محیط اپ موبایل

فهرست جداول

۱-۴ جدول قیمت محصول (قیمت‌ها به واحد هزارتومان) ۳۱

فصل ۱

مقدمه

هدف از این پژوهه، ارائه محصولی جامع برای اندازهگیری خودکار و نظارت بر علائم حیاتی بیمار به همراه شرایط محیطی است. این پژوهه به شکل کلی‌تر، قابلیت استقرار در محیط‌های بیمارستانی و همچنین محیط‌های خانگی برای افرادی که نیازمند مراقبت ویژه هستند را دارد.

محصول توانایی اندازهگیری دما، رطوبت و آلودگی هوا را به عنوان عوامل محیطی اثرگذار در شرایط بیمار دارد. در کنار آن، دمای بدن، ضربان قلب، اکسیژن خون و نوار قلب به طور خودکار ثبت شده و امکان وارد کردن فشار خون که باید به صورت دستی و توسط متخصص اندازه گیری بشود، نیز وجود دارد.

همه این داده‌ها، به تفکیک بیمار در دیتاییسی تجمعی شده و از طریق نرم‌افزار موبایلی طراحی شده که قابلیت استفاده به صورت وب‌اپلیکیشن را هم دارد، امکان مشاهده آنی این اطلاعات برای پزشک یا سایر کادر درمانی میسر خواهد بود.

مزیت رقابتی اصلی این محصول، تجمعی سیستم‌های جمع‌آوری داده‌ها در یک محصول به همراه ارائه قابلیت نظارت همزمان و گزارش‌گیری از طریق اپلیکیشن طراحی شده است. نمونه‌های موجود در بازار، هیچ‌کدام به صورت یک محصول جامع شامل همه سنسورها نیستند و به علاوه اکثر آنان قابلیت ارتباط برقرار کردن با نرم‌افزارهای موبایلی که به راحتی قابل استفاده باشند را ندارند. تجمعی سیستم‌های جمع‌آوری داده به همراه نرم‌افزار کاربرپسند که به راحتی قابل استفاده باشد، مزایای رقابتی اصلی این محصول هستند.

فصل ۲

دیتاشیت محصول

-10°C - 45°C	محدوده‌ی دمای قابل استفاده
95%	حداکثر سطح رطوبت محیط قرارگیری
5 ولت	ولتاژ ورودی
2 آمپر	جريان ورودی
30 cm * 20 cm * 20 cm	ابعاد
-	وزن
±2°C	دقت اندازه‌گیری دمای محیط
±5 %. RH	دقت اندازه‌گیری رطوبت محیط
±0/1°C	دقت اندازه‌گیری دمای بدن انسان
91/62%	دقت ثبت ECG
صفحه نمایش تاچ اسکرین 7 اینچ	سایر امکانات

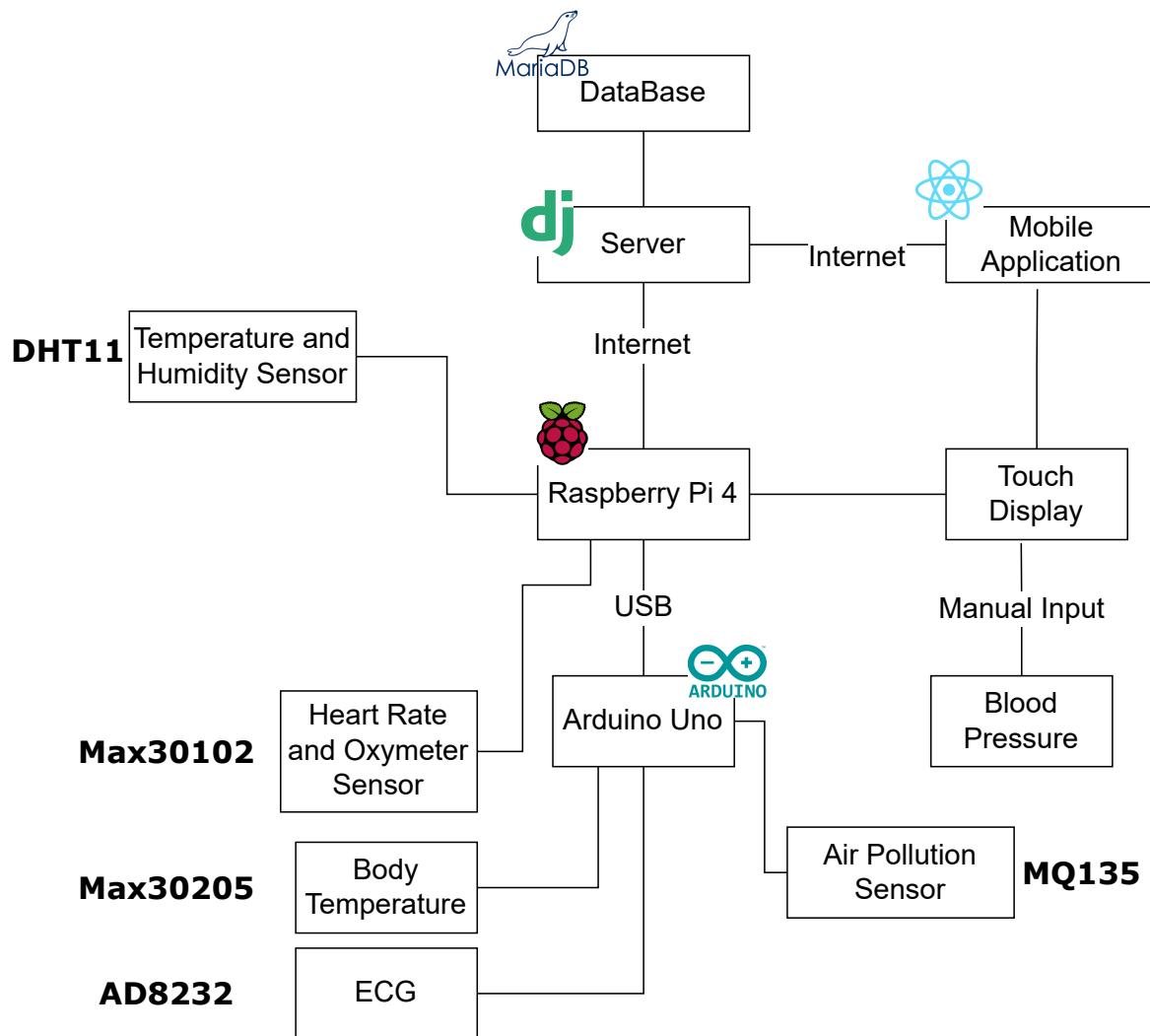
- وزن محصول بسته به شرایط و مواد مورد استفاده در تولید جعبه‌ی آن متغیر است.

فصل ۳

معماری سیستم

سیستم طراحی شده توسط ما از سه قسمت اساسی تشکیل شده است. قسمت سخت‌افزاری متشکل از رزبری‌پای، آردوینو و سنسورهای مختلف که برای اندازه‌گیری علائم حیاتی و شرایط محیطی استفاده می‌شوند. سرور که برای دریافت داده‌ها و ساماندهی آنان استفاده شده است و همچنین اپلیکیشن موبایلی که برای نمایش داده‌های جمع‌آوری شده و نظارت همزمان بر آن‌ها استفاده شده است.

معماری سطح بالای سیستم در شکل ۱-۳ قابل مشاهده است.



شکل ۳-۱: معماری سطح بالای سیستم

۳-۱ طراحی و پیاده‌سازی سخت‌افزار

اصلی‌ترین قسمت این پروژه، طراحی و پیاده‌سازی قسمت‌های سخت‌افزاری آن است. در زیر لیستی از قطعات سخت‌افزاری مورد استفاده آمده است و پس از آن توضیحاتی در مورد هر یک از سنسورها و نحوه کارکرد و راهاندازی آن ذکر شده است.

- برد Raspberry Pi 4

- برد Arduino UNO

- صفحه نمایش لمسی ۷ اینچ مخصوص Raspberry Pi

- سنسور آلودگی هوا MQ135

- سنسور دما و رطوبت هوا DHT11

- سنسور ضربان قلب و اکسیژن خون Max30102

- سنسور دمای بدن Max30205

- سنسور نوار قلب AD8232

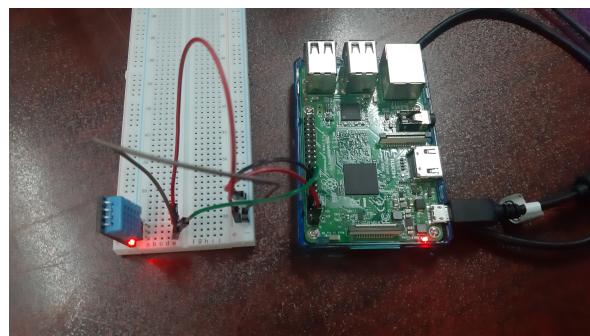
- فشار سنج و گوشی پزشکی

۳-۱-۱ سنسورهای محیطی

د، سنسور محیطی اصلی در این پروژه وجود دارند. سنسور MQ135 که وظیفه اندازه‌گیری آلودگی هوا را داشته و سنسور DHT11 که وظیفه اندازه‌گیری دما و رطوبت را دارد. سنسور آلودگی هوا به آردوینو متصل شده و سنسور اندازه‌گیری دما و رطوبت هوا مستقیماً به رزبری‌پای متصل می‌شود.

سنسور دما و رطوبت هوا

سنسور مورد استفاده برای این بخش، DHT11 است که از قابلیت انتقال داده به صورت دیجیتال پشتیبانی کرده و برای همین به راحتی مطابق شکل ۳-۲ به رزبری‌پای متصل می‌شود.

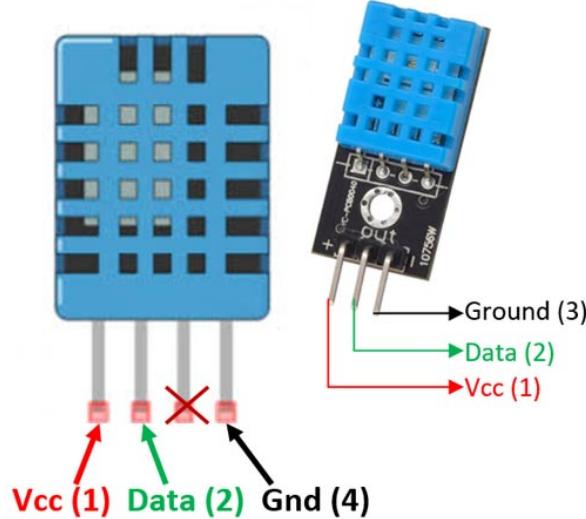


شکل ۳-۲: اتصال سنسور DHT11

برای خواندن مقادیر از کتابخانه Adafruit-Blinka^۱ استفاده شده است. این کتابخانه با مشخص کردن پین متصل به سنسور، به راحتی امکان خواندن دما و رطوبت هوا را به ما می‌دهد. کد اصلی مربوط به این قسمت در زیر آورده شده است:

```
dht11_sensor = adafruit_dht.DHT11(board.D23)
temp = dht11_sensor.temperature
humidity = dht11_sensor.humidity
```

همچنین در شکل ۳-۳ پین‌های مازول DHT11 مشخص شده‌اند.



شکل ۳-۳: پین‌های سنسور DHT11

همان طور که مشخص است، این سنسور دو پین GND و VCC برای تغذیه داشته و یک پین DATA برای انتقال داده‌ها به صورت دیجیتالی دارد.

سنسور آلودگی هوا

با توجه به این که سنسور MQ135 خروجی اصلی خود را به صورت آنالوگ تحویل داده و حتی رابط I2C هم ندارد، آن را به برد آردوینو متصل کرده و از طریق اتصال رزبریپای به آردوینو با پورت USB، کد مربوط به آن را از طریق رزبری به برد آردوینو انتقال داده و داده‌های لازم را دریافت می‌کنیم.

نحوه اتصال این سنسور در کنار سنسور قبلی در شکل ۳-۴ قرار دارد.



شکل ۳-۴: اتصال سنسور MQ135

در کد آردوینو، مقادیر مربوط به این سنسور هر ۵ ثانیه خوانده می‌شود:

```
#include <Wire.h>

void setup() {
    Serial.begin(9600);
    Wire.begin();
}

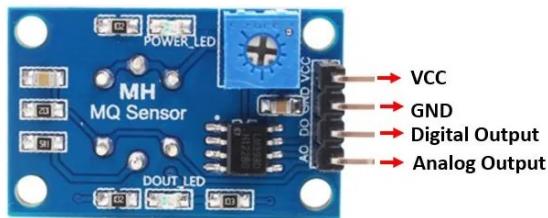
int counter = 0; // 1 milisecond

void loop() {
    if (counter % 5000 == 0) // 5 second
    {
        int pollution = analogRead(A0);
        Serial.print("pollution ,");
        Serial.println(pollution);
    }
    counter += 10;
    delay(10);
}
```

{

سپس در کد پایتون روی رزبری، این مقادیر روی یک فایل ریخته می‌شود و پس از آن مقادیر باقی سنسورها به سرور ارسال می‌گردد.

همچنین در شکل ۳-۵ پین‌های ماژول MQ135، مشخص شده‌اند.



شکل ۳-۵: پین‌های ماژول سنسور MQ135

همان طور که مشخص است، این سنسور دو پین GND و VCC برای تغذیه داشته و دو پین دیگر برای انتقال آنالوگ و دیجیتال دارد. نکته قابل توجه این است که پین دیجیتالی فقط به صورت صفر و یکی و برای مشخص کردن گذر مقدار گاز موجود از حد مجاز کاربرد دارد و نمی‌تواند مقدار دقیق آن را انتقال بدهد و مقدار دقیق آن تنها از طریق پین آنالوگ قابل دریافت است. برای تنظیم حد مجاز، از پتانسیومتری که در قسمت بالا سمت راست ۳-۵ قرار دارد استفاده می‌شود.

۲-۱-۳ سنسورهای بدن

سه سنسور اصلی برای علائم مربوط به بدن انسان در این پروژه وجود دارند. سنسور دمای بدن Max30205، سنسور ضربان قلب و اکسیژن خون Max30102 و سنسور نوار قلب یعنی AD8232. سنسور Max30102 مستقیماً به رزبری‌پای متصل می‌شود ولی دو سنسور دیگر از طریق آردوبینو با رزبری‌پای ارتباط برقرار می‌کنند.

سنسور دمای بدن

برای سنجش دمای بدن از سنسور Max30205 استفاده شده است. با تماس انگشت به آن، بعد از مدتی دمای سنسور با دمای انگشت همدما شده و دمای بدن را نشان خواهد داد. در صورت عدم تماس هم می‌توان از آن برای مشاهده دمای محیط استفاده کرد.

این سنسور به آردوینو متصل شده و از طریق اتصال آردوینو به رزبریپای، اطلاعات آن را مشاهده می‌کنیم. در شکل ۳-۶ تصویر اتصال این سنسور قرار دارد.



شکل ۳-۶: اتصال سنسور Max30205

برای خواندن مقادیر این سنسور از کتابخانه Protocentral_MAX30205^۲ استفاده شده که امكان خواندن دمای بدن را از طریق آردوینو می‌دهد.

```
#include <Wire.h>
#include "Protocentral_MAX30205.h"
MAX30205 tempSensor;

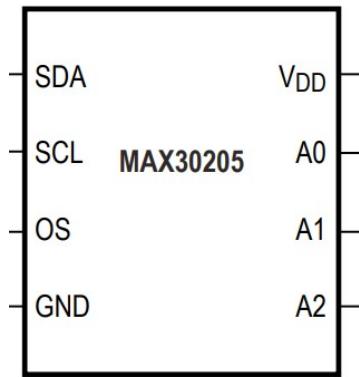
void setup() {
    Serial.begin(9600);
    Wire.begin();
    while (!tempSensor.scanAvailableSensors()) {
        Serial.println("Couldn't find the sensor.");
        delay(3000);
    }
    int counter = 0; // 1 milisecond
```

```

void loop() {
    if (counter % 5000 == 0) // 5 second
    {
        float temp = tempSensor.getTemperature();
        Serial.print("temp, ");
        Serial.println(temp, 2);
    }
    counter += 10;
    delay(10);
}

```

همچنین در شکل ۷-۳ پین‌های ماژول MAX30205 مشخص شده‌اند.



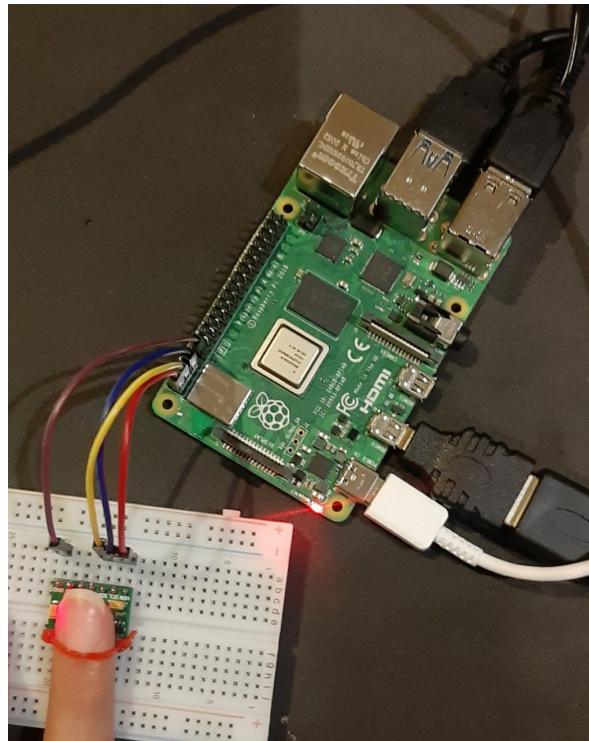
شکل ۷-۳: پین‌های سنسور MAX30205

دو پین GND و VDD برای تغذیه مدار استفاده می‌شوند. پین‌های SCL و SDA پین‌هایی هستند که در تمامی سنسورهایی که به صورت I2C کار می‌کنند، مشاهده می‌کنیم. SCL برای تامین کلاک استفاده شده و انتقال داده از طریق SDA صورت می‌گیرد. پین‌های A0,A1,A2 برای تعیین شماره آدرس I2C مدار استفاده می‌شوند و بسته به مقادیری که به آنان داده بشود، آدرس مشخصی برای I2C در مداری که سنسور به آن متصل می‌شود مشخص می‌شود. پین OS مخفف Overtemperature Shutdown است. این پین به نوعی مانند یک ترمومترات عمل می‌کند و وقتی دمای اندازه‌گیری شده، از مقداری که درون یکی از رجیسترها درونی مدار مشخص شده است بالاتر برود، مقدار آن یک می‌شود. برای کاربردهای رایج توصیه سازنده این است که این پین یا اصلاً متصل نشده و یا به مدار PullUp با مقاومت ۴.۷kΩ

متصل شود.

سنسور ضربان قلب و اکسیژن خون

برای این بخش از سنسور Max30102 استفاده شده است. این سنسور با کمک دو چراغ کوچک قرمز و مادون قرمز، ضربان قلب و درصد اشباع اکسیژن در خون (SpO_2) را اندازه‌گیری می‌کند. این سنسور بدون مشکل از طریق I2C به رزبری‌پای متصل می‌شود. نحوه اتصال این سنسور را در شکل ۳-۸ مشاهده می‌کنید.



شکل ۳-۸: اتصال سنسور Max30102

برای راهاندازی این سنسور از کدهای مخزن متن باز [doug-burrell/max30102](#)^۳ با اندکی تغییرات استفاده شده است. این مخزن با خواندن مقادیر سنسورهای قرمز/مادون قرمز و پردازش آنها، مقادیر ضربان قلب و اکسیژن خون را به طور دقیق محاسبه می‌کند. کدهای مربوط به این مخزن در فایل‌های `hrcalc.py` و `max30102.py` قرار دارند. از امکانات این کتابخانه به صورت زیر در کد رزبری استفاده شده است:

```
from heartrate_monitor import HeartRateMonitor
```

[https://github.com/doug-burrell/max30102^۴](https://github.com/doug-burrell/max30102)

```
hrm = HeartRateMonitor( print_raw=False , print_result=False )
hrm.start_sensor()
```

```
bpm = hrm.bpm
```

```
spo2 = hrm.spo2
```

همچنین در شکل ۹-۳ پین‌های مازول MAX30102 مشخص شده‌اند.



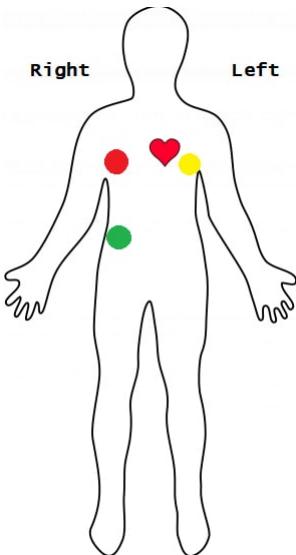
شکل ۹-۳: پین‌های سنسور MAX30102

دو پین GND و VIN برای تغذیه مدار استفاده می‌شوند. پین‌های SCL و SDA پین‌هایی هستند که در تمامی سنسورهایی که به صورت I2C کار می‌کنند، مشاهده می‌کنیم. SCL برای تامین کلاک استفاده شده و انتقال داده از طریق SDA صورت می‌گیرد. پین‌های RD و IRD برای این هستند که به صورت دستی، LED های قرمز و مادون قرمز مازول را روشن کرد. در حالت کلی نیازی به وصل کردن سیمی به این دو سنسور نیست. پین INT در زمان‌هایی که در سنسورهای درونی مدار Interrupt رخ بدهد به حالت LOW وارد می‌شود و در بقیه زمان‌ها HIGH است. Interrupt ها می‌توانند شامل مواردی نظیر پرشدن صفحه FIFO درونی سنسور، تاثیر گذاشتن نور زمینه محیط بر نتیجه در اثر Overflow واحد Ambient Light Cancellation و موارد دیگر رخ بدهند. در کاربرد عمومی نیازی به وصل کردن این پین به خروجی خاصی نیست.

سنسور نوار قلب

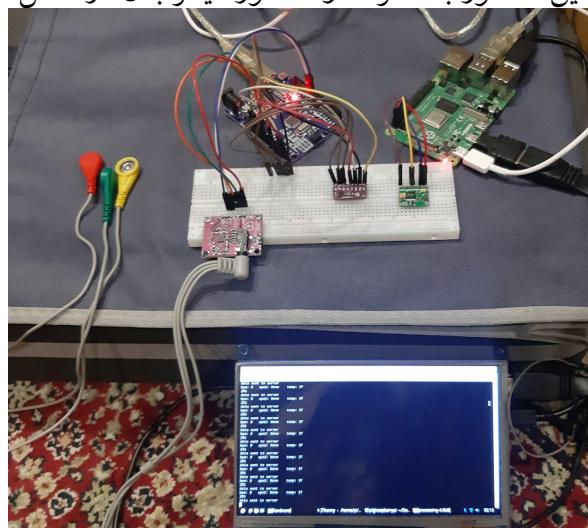
از سنسور AD8232 برای بدست آوردن ECG (نوار قلب) استفاده می‌شود. همراه این سنسور بسته سه تایی اتصال به بدن وجود دارد. به این Lead ها باید پدهای مخصوص متصل شده و به بدن متصل شوند. برای استفاده از سنسور، آن را به آردوینو متصل می‌کنیم. Lead های آن هم باید مطابق

شکل ۳-۱۰ به بدن بیمار متصل بشوند.



شکل ۳-۱۰: نحوه اتصال Lead های سنسور نوار قلب به بدن

همچنین نحوه اتصال این سنسور به همراه دو سنسور دیگر بدن در شکل ۳-۱۱ قرار دارد.



شکل ۳-۱۱: اتصال سنسورهای بدن به رزبری پای و آردوینو در کنار هم

کدهای آردوینو مربوط به این سنسور در زیر آمده است. هر ۱۰ میلی ثانیه مقادیر مربوط به این سنسور از طریق آردوینو خوانده می شود.

```
#include <Wire.h>
void setup() {
    Serial.begin(9600);
    Wire.begin();
```

```

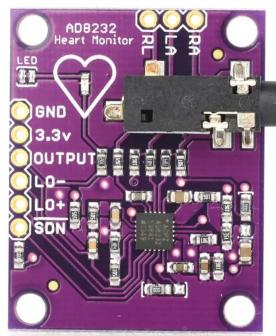
pinMode(10, INPUT);
pinMode(11, OUTPUT);
}

int counter = 0; // 1 milisecond

void loop() {
    if (counter % 10 == 0) // 10 milisecond
    {
        Serial.print("ecg,");
        if(digitalRead(10) == 1 || digitalRead(11) == 1) {
            Serial.println('!');
        }
        else {
            int ecg = analogRead(A1);
            Serial.println(ecg);
        }
    }
    counter += 10;
    delay(10);
}

```

همچنین در شکل ۱۲-۳ پین‌های مازول AD8232 مشخص شده‌اند.



شکل ۱۲-۳: پین‌های سنسور AD8232

دو پین GND و 3.3V برای تغذیه مدار استفاده می‌شوند. سه پین RA, LA, RL برای گرفتن خروجی Lead های متصل به بدن به صورت جداگانه استفاده می‌شوند که در این پروژه از آن‌ها استفاده‌ای نشده است. در عوض از جک 3.5 میلی‌متری برای اتصال Lead‌ها استفاده شده است. پین OUT خروجی آنالوگ مدار است که با استفاده از آپ‌امپ، تقویت و کنترل شده است. پین‌های LOD- و LOD+ به معنی Leads of Detection یعنی زمانی که سنسور نتواند اتصال درست الکترود منفی به بدن را تشخیص بدهد به حالت HIGH می‌رود. پین LOD- زمانی که سنسور نتواند حضور الکترود مثبت را تشخیص بدهد به حالت HIGH می‌رود. در صورتی که هر دو LOW باشند، یعنی ها به درستی متصل شده‌اند. پین SDN، به معنی Shutdown Control Input است و به صورت Lead Active Low کار می‌کند. این پین در مدارهای سیستم‌های نهفته که تنها قرار است گاهی اوقات قابلیت اندازه‌گیری ضربان قلبشان فعال شده و در سایر زمان‌ها غیرفعال است استفاده می‌شود. در صورتی که ورودی LOW به این پین داده بشود، عملاً بخش عمدۀ ماژول از کار می‌افتد و مدار جریان خیلی کمی از منبع تغذیه کشیده و مصرف کمی پیدا می‌کند. در نتیجه در سیستم‌هایی که مصرف توان اهمیت بالایی دارد و از طرفی قصد داشته باشیم امکان کنترل الکترونیکی فعال یا غیرفعال بودن سنسور را داشته باشیم، می‌توان از طریق این پین کل سنسور را به حالت خاموش وارد کرد.

اندازه‌گیری فشار خون

با توجه به این که اندازه‌گیری فشار خون به شکل دقیق معمولاً توسط فرد آموزش دیده انجام می‌گیرد و ساخت فشار سنج اتوماتیک از حوزه کاری این محصول خارج بوده و خود یک محصول جداگانه است، فشار در این محصول به شکل انسانی با استفاده از فشارسنج اندازه‌گیری شده و در منویی که برای ثبت فشار در نرمافزار قرار گرفته است، وارد می‌شود. تصویر فشارسنج در شکل ۱۳-۳ قرار گرفته است.



شکل ۱۳-۳ : فشارسنج

۳-۱-۳ کدهای سخت افزار

در قسمت های بالا، برای هر سنسور کدهای مهم آن توضیح داده شده است. در این قسمت به طور خلاصه به ساختار فایل هایی که برای کدهای سخت افزار وجود دارند پرداخته می شود.

```

Hardware
└── arduino
    └── main
        └── main.ino

└── raspberry
    ├── heartrate_monitor.py
    ├── hrcalc.py
    └── max30102.py

└── transmission
    ├── send_values.py
    └── store_arduino_values.py

└── utils
    └── utils.py

└── main.py
└── patient_id.txt
└── requirements.txt
└── README.md

```

کد موجود در پوشه `arduino`، باید از طریق نرم افزار آردوینو به روی آردوینو منتقل شود. این کد برای سنسور هایی که به آردوینو متصل هستند استفاده شده و داده های آنان را دریافت می کنند.

کدهای موجود در پوشه `heartrate+` کدهای پایتونی هستند که مربوط به اندازه گیری ضربان قلب و اکسیژن خون می شوند. در فایل `max30102` اطلاعات لازم برای ارتباط برقرار کردن با ماژول قرار گرفته و مستقیماً از کدهای مخزن متن باز `doug-burrell/max30102` با اندکی تغییرات گرفته شده است. فایل `hrcalc` برای انجام یکسری محاسبات براساس داده های دریافتی و تبدیل آنها به ضربان قلب است. فایل `heartrate_monitor+` فایل نهایی است که داده ها را با استفاده از دو فایل دیگر جمع آوری می کند.

کدهای موجود در پوشه `transmission` برای دریافت و ارسال فایل‌ها هستند. کدهای موجود در `store_arduino_values` از طریق پورت USB با آردینو ارتباط برقرار کرده و اطلاعات دریافتی از سمت آن را روی فایل می‌نویسد. فایل `send_values` اطلاعاتی که مستقیماً از طریق رزبری قابل دریافت هستند را دریافت کرده و همچنین از اطلاعات ذخیره شده توسط `store_arduino_values` استفاده می‌کند تا تمامی اطلاعات جمع‌آوری شده و به سرور ارسال بشوند.

کد موجود در پوشه `utils` شامل یکسری داده‌های سراسری مورد استفاده برنامه نظیر آدرس سرور، محل ذخیره‌سازی فایل‌ها، فعال بودن حالت `DEBUG` و... است.

فایل `patient_id.txt` تنها شامل شماره آی‌دی بیمار خواهد بود. از آن برای ارسال اطلاعات به سرور استفاده می‌شود.

فایل `main.py` کد اصلی برای اجرای برنامه است که کارهای لازم برای اجرای کل برنامه را در خود دارد و از بقیه فایل‌ها استفاده می‌کند و در نهایت این فایل باید اجرا بشود.

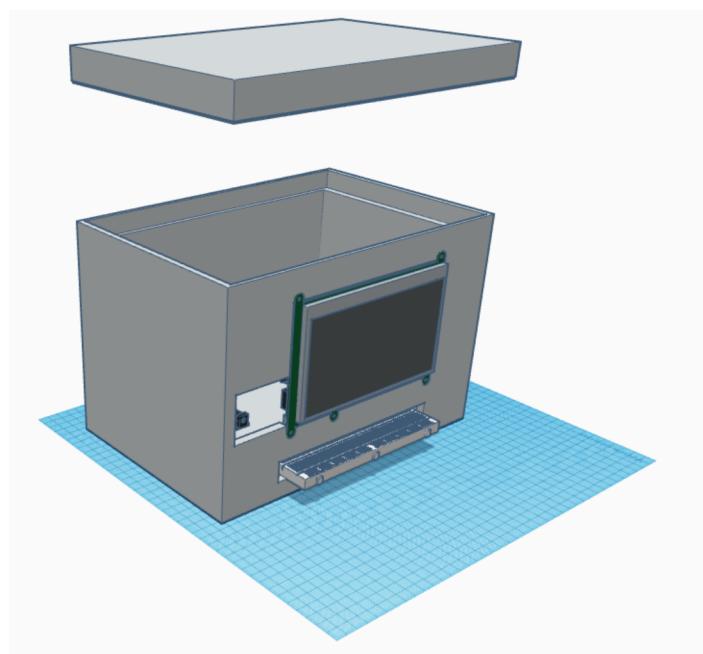
فایل `requirements.txt` شامل لیست پکیج‌های مورد نیاز پایتونی برای اجرای برنامه است.

فایل `README.md` شامل اطلاعاتی در مورد نحوه اجرا کردن برنامه است.

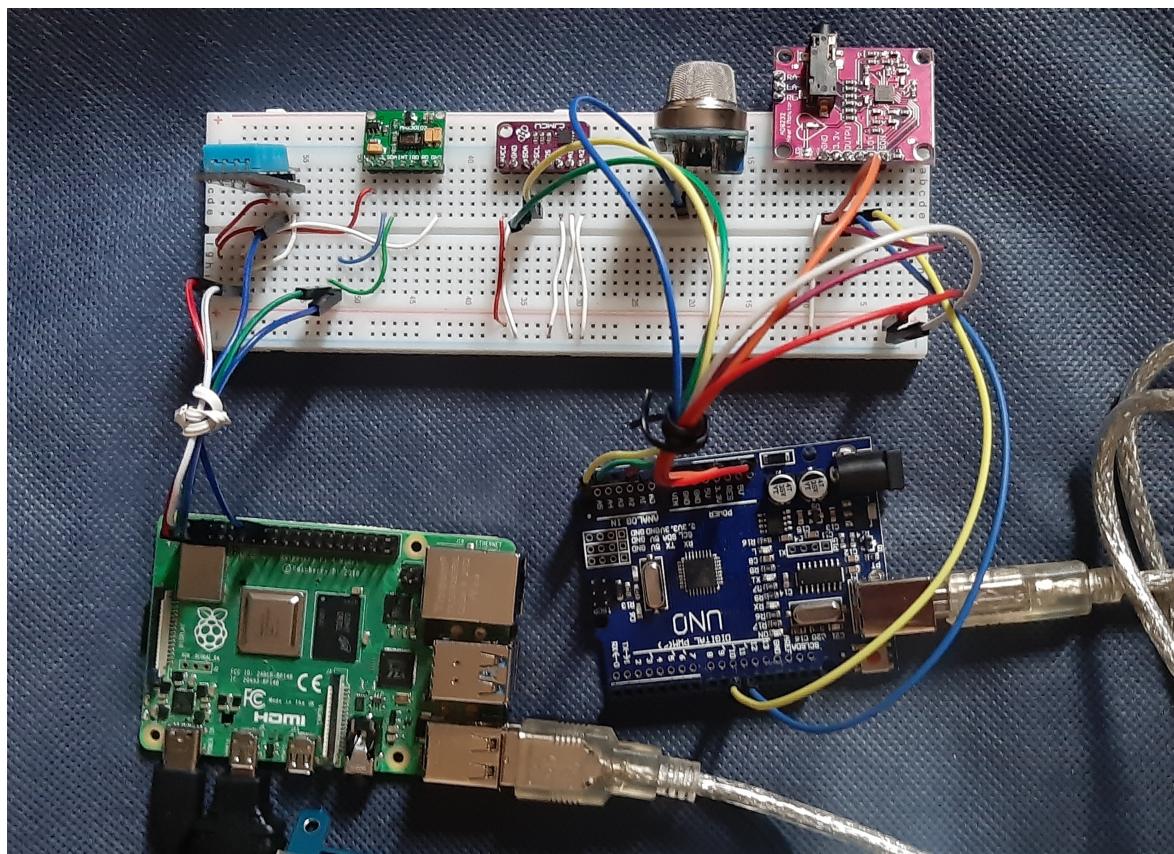
۴-۱-۳ بسته‌بندی

برای بسته‌بندی، با توجه به هماهنگی‌های صورت گرفته قرار است که از پرینتر سه‌بعدی استفاده شود. از این رو قسمت بسته‌بندی محصول هنوز کامل نیست. در شکل ۱۴-۳ طرح آماده شده برای بسته‌بندی را مشاهده می‌کنید.

وضعیت فعلی محصول به همراه تمامی سنسورها در شکل ۱۵-۳ آمده است.



شکل ۳-۱۴: بسته‌بندی طراحی شده در نرم‌افزار Tinkercad



شکل ۳-۱۵: محصول به همراه تمامی سنسورها

۲-۳ طراحی و پیاده‌سازی سرور

سرور در این پروژه وظیفه دریافت اطلاعات از رزبری‌پای، ذخیره آن‌ها در پایگاه داده و تحويل دادن آن‌ها با فیلترها و به شکل مناسب به نرم‌افزار طراحی شده را دارد.

برای طراحی سرور از فریم‌ورک Django Rest Framework و Django استفاده شده است که به ما امکان طراحی سریع و در عین حال اصولی سروری که برای این پروژه مناسب باشد را می‌داد. برای پایگاه‌داده هم از MariaDB استفاده شده است که براساس MySQL توسعه یافته است و با قواعد MySQL هماهنگی کامل دارد. برای راحتی کار توسعه سرور و نصب نیازمندی‌های آن، از Docker Compose استفاده شده است تا به راحتی همه اجزای مختلف سرور مستقل از سیستمی که توسعه روی آن صورت می‌گیرد به شکل مناسب و سریع استقرار یابد.

همچنین برای این که نیاز به استفاده از دستورات Docker کمینه شود، یک فایل Makefile هم نوشته شده است که با دو دستور اصلی make build و make up، راهاندازی اولیه سرور و بالا آوردن آن بعد از راهاندازی اولیه قابل انجام است. همچنین با دستور make down می‌توان سرور را خاموش کرد.

سرور پروژه از دو قسمت اصلی تشکیل شده است. بخش Patients که در آن مدل مربوط به بیماران (شامل نام و نام خانوادگی، شماره تلفن، قد، وزن و جنسیت) در فایل models.py در فایل تعريف شده است. همچنین با استفاده از کلاس‌های Django Rest Framework تمامی کارهای مربوط به ایجاد، آپدیت، حذف و دریافت اطلاعات هویتی به بیماران در این قسمت و فایل views.py انجام می‌شود.

بخش دیگر Records است. در این جا مدل اصلی برای ذخیره داده‌های بیماران و همچنین داده‌های محیطی تعريف شده است. همچنین در قسمت views.py علاوه بر تعريف دو کلاس براساس Django Framework Rest که امکان عملیات‌های یجاد، آپدیت، حذف و دریافت را برای داده‌های بیماران فراهم کنند، سه تابع برای دریافت آخرین داده، دریافت آخرین داده‌ها از زمان مشخص شده و دریافت داده‌ها با تعیین فیلترهای مختلف (زمان، بیمار و سنسورهای خاص) تعريف شده است. همچنین تابعی که آخرین داده را برای ما فراهم می‌کند، برای هر فیلد مستقل به دنبال آخرین داده می‌گردد تا اگر در داده‌هایی که برای آخرین زمان ثبت شده‌اند، برای برخی از سنسورها داده‌ای وجود نداشت، آخرین داده موجود گزارش شود. در صورتی که هیچ داده‌ای وجود نداشته باشد، مقادیر پیش‌فرض ثبت شده برای آن فیلد گزارش می‌شوند.

علاوه بر این، با استفاده از ابزار Swagger مستندات خودکار برای استفاده از API‌ها هم ایجاد شده

است. در صورتی که سرور به صورت محلی اجرا شود، این مستندات از آدرس `localhost:8000/swagger` در دسترس خواهد بود. تصویر آن را در شکل ۱۶-۳ مشاهده می‌کنید.

شکل ۳-۱۶: مستندات Swagger

۳-۳ طراحی و پیاده‌سازی اپلیکیشن

۱-۳-۳ مقدمه

اپلیکیشن موبایل با کمک ابزار React Native ایجاد شده است و دارای قابلیت‌های مشاهده‌ی لیست بیماران (تصویر ۱۷-۳)، مشاهده آخرین وضعیت علائم حیاتی بیمار و محیط اطراف، مشاهده رکوردهای قدیمی و نهایتاً مصورسازی دادگان ثبت شده در نمودارهای متعدد با اتصال به سرور جهت کشف روابط و تحلیل داده‌ها فراهم شده است.

۲-۳-۳ تصاویری از محیط نرم‌افزار

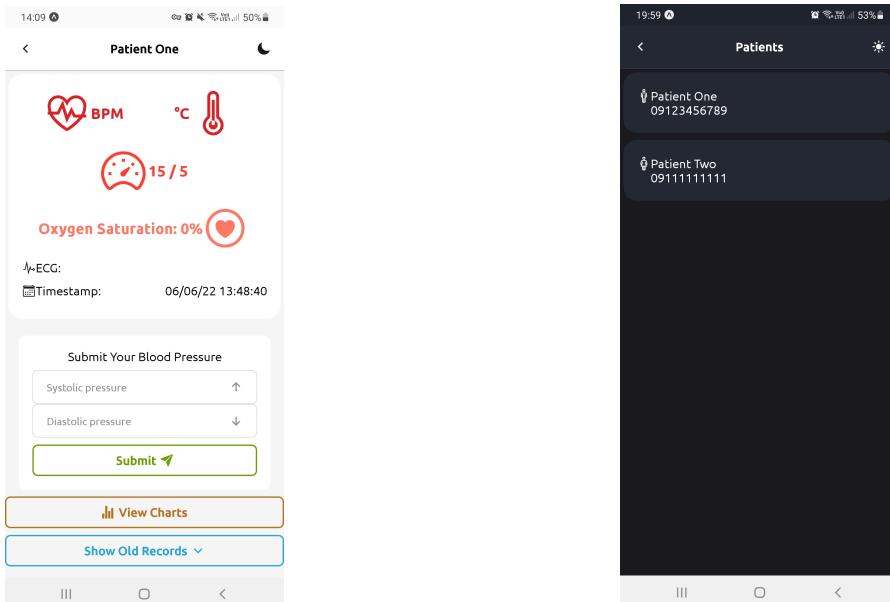
در شکل ۱۷-۳، تصاویری از محیط برنامه که به کمک Expo اجرا شده قابل مشاهده است.

در راستای مصورسازی داده‌ها با استفاده از کتابخانه‌ی Recharts صفحه‌ای برای هر بیمار به نرم‌افزار اضافه شد که با در کنار هم قرار دادن اطلاعات سنسورها در طول زمان، امکان استخراج اطلاعات مفیدی به دست آید. این صفحه از طریق صفحه‌ی اطلاعات بیمار (تصویر ۱۷-۳ ب) قابل دسترسی می‌باشد. نمودارهای نمایش داده شده در تصویر ۱۷-۳ ج و ۱۷-۳ د به ترتیب، اطلاعات مربوط به میزان اکسیژن خون، و بررسی توأم دمای بدن و محیط، ارتباط آلودگی هوا و رطوبت نسبی (محور افقی) پیا میزان اکسیژن خون (محور عمودی) را با فراهم کردن امکان انتخاب بازه‌ی زمانی برای نمایش داده‌های نمودارها از طریق یک منو مصور می‌کند.

۳-۳-۳ توضیحاتی در ارتباط با کد برنامه

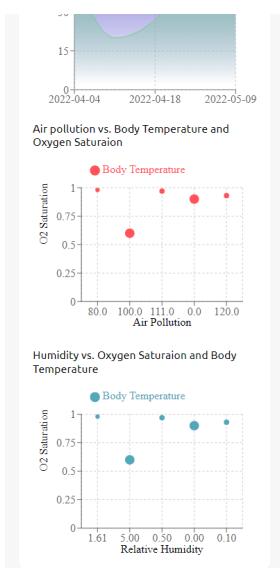
همانطور که بالاتر ذکر شد، نرم‌افزار با کمک React Native ایجاد شده که امکان نمایش واسط کاربری طراحی شده روی تمامی پلتفرم‌ها را محقق می‌سازد. برای طراحی صفحات اپلیکیشن از قالب Rapi و برای رسم نمودارها از کتابخانه‌ی Recharts بهره گرفته شد. کد برنامه و نحوه‌ی نصب و راهاندازی آن در این مسیر قرار گرفته است. این کد با کمک کامنت‌ها و جاوا‌دک^۴ مستند شده است.

⁴Javadoc

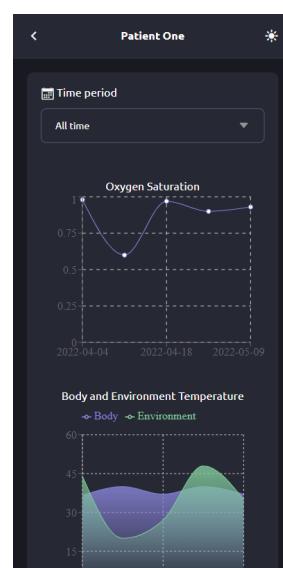


(ب) صفحه‌ی بیمار

(آ) لیست بیماران ثبت شده



(د) برخی دیگر از نمودارها



(ج) برخی از نمودارها

شکل ۳-۱۷: تصاویری از محیط اپ موبایل

فصل ۴

قیمت

یکی از مسائل مهم در طراحی محصول قیمت آن است. البته با توجه به این که این محصول به صورت نمونه اولیه طراحی شده است، طبیعتاً قیمت تمام شده آن از محصولی که بخواهد تولید عمده بشود بالاتر خواهد بود. در جدول ۱-۴ قیمتی تخمین زده شده و هزینه نهایی پروژه آورده شده است.

عمده تفاوت قیمت بین اعداد تخمینی و عدد نهایی، به دلیل تغییر رزبری‌پای از ۳B به ۴، اضافه شدن آردوبینو و همچنین اضافه شدن فشارسنج دستی صورت گرفته است.

ردیف	قطعه	قیمت تخمینی	قیمت نهایی
۱	MQ-135	۳۸.۳	۳۸.۳
۲	MPS20N004D	۲۶.۵	۰
۳	DHT11 or KY-015	۴۵	۴۵
۴	MAX30205	۲۹۰.۹	۲۹۰.۹
۵	AD8232	۱۲۸	۱۲۸
۶	ECG Electrode	۹۶	۹۶
۷	MAX30102	۱۶۳.۳	۱۶۳.۳
۸	Raspberry Pi LCD	۱۴۸۰	۲۰۰۰
۹	Raspberry Pi 3B	۳۰۰۰	۰
۱۰	Raspberry Pi 4	۰	۴۰۰۰
۱۱	Arduino Uno	۰	۴۰۰
۱۲	Breadboard	۴۷.۵	۴۷.۵
۱۳	Wires	۱۰۰	۱۰۰
۱۴	Resistors	۷	۱۰
۱۵	ADS1115	۱۵۴	۰
۱۶	Sphygmomanometer	۰	۷۰۰
۱۷	ECG Pads	۱۲	۵۰
۱۸	Total	۵۵۸۸.۵	۸۰۶۹

جدول ۴-۱: جدول قیمت محصول (قیمت‌ها به واحد هزار تومان)

فصل ۵

جمع‌بندی

در این پروژه به پیاده‌سازی سیستم نمایش‌گر علائم حیاتی بیمار پرداختیم. در این سیستم با استفاده از سنسورهای مختلف محیطی و همچنین سنسورهای بدن انسان و به کمک رزبری‌پای و آردوبینو، اطلاعات مختلفی شامل دمای هوا، رطوبت هوا، آلودگی هوا، دمای بدن، ضربان قلب، اکسیژن خون، نوار قلب و فشار خون اندازه‌گیری می‌شوند.

در کنار این ابزار، نرمافزاری ارائه شده است که قابلیت استفاده بر روی موبایل و وب و دستگاه‌هایی با ابعاد مختلف را دارد. این نرمافزار به پزشکان و پرستاران و سایر افرادی که مسئول مراقبت از بیمار هستند، کمک می‌کند که به طور لحظه‌ای شرایط حیاتی بیمار را کنترل کرده و همچنین در نمودارهایی وضعیت بیمار در زمان‌های مختلف و همچنین ارتباط وضعیت بیمار با شرایط محیطی را مشاهده نمایند.

آنچه که این محصول را از نمونه‌های مشابه متمایز می‌کند، ترکیب سنسورهای گوناگون در یک محصول به همراه ارائه نرمافزاری جامع برای مدیریت و نظارت بر وضعیت بیماران مختلف و مشاهده لحظه‌ای تمامی علائم حیاتی آنان و همچنین محیطی که در آن حضور دارند است.