

# ADVANCED HEALTH CARE MONITORING SYSTEM

M.M.M AMRY	- KAHDSE24.1F - 023
A. DHANUSHANANDAN	- KAHDSE24.1F - 028
M.A.M AMMAR	- KAHDSE24.1F – 026
M.Z.F.ZEENA	- KAHDSE24.1F - 022

## PROBLEM STATEMENT

Paralyzed patients need continuous monitoring of vital signs because of their inability communication problem. Current healthcare monitoring system does not have a real-time monitoring alert system which can help in a emergency situation.

## OBJECTIVES

We are developing a continuously monitoring patient vital signs such as heart rate, temperature, blood pressure, and ECG. Reports display in Display and check through a mobile app real-time. In case of emergency alert SMS sent to the hospital and guardian with live location of the patient Additionally a voice will play. If the patient gets fall, then get an alert also.

## PROPOSED SOLUTION

We proposed a solution for the problem, continuously monitor the patient using some health care sensors like heart rate sensor, blood pressure sensor, ECG sensor. The patient report can be view in a display and through a mobile application. Patient's abnormal health situation guardian and doctor get a SMS alert and live location of the patient for this we use a GPS module. voice alert in emergency. If patient get fall from bed, then the guardian gets a SMS alert to mobile.

## SCOPE & LIMITATIONS

<b>SCOPE</b>	: This project monitors the patient's vital signs continuously and show the data in display and mobile app for this we use multiple sensors. Advanced the project with emergency SMS alert and live locations sharing.
<b>LIMITATIONS</b>	: Sensor's accuracy can be affected by the environment changes. Sending SMS and Location need a strong WIFI connection.

## SYSTEM ARCHITECTURE

In our system we used Arduino Mega as a microcontroller to connect the sensors like heart rate sensor, temperature sensor, blood pressure sensor, fall detection sensor, ESP8266 for mobile app connectivity , GPS module for live location and GSM module for SMS alerts.

## HARDWARE & SOFTWARE REQUIREMENTS

### HARDWARE:

- |                                     |  |
|-------------------------------------|--|
| ➤ Arduino Mega.                     | => Micro controller                      |
| ➤ Esp8266.                          | => WIFI module for app connection        |
| ➤ GSM Module (SIM800L).             | => SMS send module                       |
| ➤ OLED Display.                     | => Display reports                       |
| ➤ Heart rate sensor (MAX30105).     | => Calculate heart rate & SO2 & pressure |
| ➤ Temperature sensor (DS18B20).     | => Calculate Temperature                 |
| ➤ Fall detection sensor. (MPU6050). | => Detect the falling                    |
| ➤ Flux sensor .                     | => Emergency activation                  |
| ➤ ECG sensor (AD8232).              | => Calculate ECG                         |
| ➤ DF player module and speaker.     | => Voice message                         |
| ➤ GPS module (NEO6MV2).             | => Live location share                   |

### SOFTWARE:

- Arduino IDE
- BLYNK App

## DATA PROCESSING

The system will continuously collect data from the different sensor using Arduino and process the define functions. In case of emergency like patients' abnormal health conditions or user ask for help send SMS alerts and locations.

## NETWORK & SECURITY

We used ESP8266 module for Blynk app connection, GMS module for send SMS and NEO6MV2 for send live locations of the patients. Patient's reports can only see through the connected mobile app only for the security feature.

## **EXPECTED BENEFITS**

This system helps the patients to monitoring the health conditions. Very helpful in a emergency cases. Doctors can easily view the patient's vital signs without real time visiting. Blynk app is a user-friendly interface.

## **TIMELINE**

- **WEEK 1**                   => Research and find the suitable components for the project and discuss about the project.
- **WEEK 2**                   => Finalize the circuit diagrams and hardware connections.
- **WEEK 3 &4**               => Develop the code for the project.
- **WEEK 5**                   => Test the code and components finalize the project.