ISSN No:-2456-2165

Remote Suggestion Through IOT Enabled Personalized Healthcare

Lakshana R
Department of Electronics and Communication Engineering
K.C.G. College of Technology
Chennai, India

Aida Jones
Department of Electronics and Communication Engineering
K.C.G. College of Technology
Chennai, India

Abstract:- Remote Patient Monitoring is one of the major divisions of healthcare which has always gained attention with the increasing technological growth. This has been implemented in many ways like telemedicine, health band like smart watches. Remote patient monitoring uses digital technologies to collect medical and other forms of health data from individuals and transmit those data securely to health care providers for assessment and recommendations. There are various methodologies used in this field but most of which are not effective like health bands which generate inaccurate readings. With this method a more accurate way can be implemented because a more precision equipment can be used here. The main objective of this method is remote medical suggestion where your logged healthcare data is viewed by many medical practitioners and they can assist the patient in need. In the latter method, a preferred doctor may not be available at all times which can be compensated in this method by having another doctor to give medication as this method involves multiple doctors monitoring your health data. This method involves monitoring of the patient data by sensors like pulse sensor, temperature sensor to determine pulse rate and temperature. These data are stored in a MySQL database and also can be viewed in a website using a webserver like Flask, both by the doctor and the patient. In the future the whole process can be made automated by using AI.

Keywords:- *Flask, MySQL, AI.*

I. INTRODUCTION

This modern era has attracted many personal health assistants and remote monitoring system but the idea is pretty unique in its approach. Studying the previous works in the field of Remote Healthcare, it is seen that consultation with the doctor regarding the monitored health is the most important. Previous works does not suggest about the remote consultation mechanism with the doctors.

In this paper, the proposed method is to send the remotely monitored health data to the web dashboard which can be viewed and medication can be suggested by any doctor online. This method is greatly helpful for the patient when the patient's choice of doctor is not online and the patient is having queries about his health which needs to be clarified. The website consists of a login page, signup page and the dashboard which is separate for the doctor and the user. The doctor's dashboard shows all the patients online and is able to view their monitored health data and suggest some medication.

In the patient's dashboard, the patient is able to view his own health data and also by navigating to the message page, the patient is able to view a list of doctor's names who have messaged the patient. The proposed system not only does the health care data taken from the patients is logged into the database, but also provides a way to give immediate medication for the patient's health issue. The user health data is stored securely in a database.

II. PROPOSED METHODOLOGY

Studying the previous works in the field of Remote Healthcare, it is seen that consultation with the doctor regarding the monitored health is the most important. Previous works does not suggest about the remote consultation mechanism with the doctors. In this paper, the proposed method is to send the remotely monitored health data to the web dashboard which can be viewed and medication can be suggested by any doctor online. This method is greatly helpful for the patient when the patient's choice of doctor is not online and the patient is having queries about his health which needs to be clarified. The website consists of a login page, signup page and the dashboard which is separate for the doctor and the user. The doctor's dashboard shows all the patients online and is able to view their monitored health data and suggest some medication. In the patient's dashboard, the patient is able to view his own health data and also by navigating to the message page, the patient is able to view a list of doctor's names who have messaged the patient.

III. BLOCK DIAGRAM

The below block diagram explains the various blocks that are involved in implementing this project.

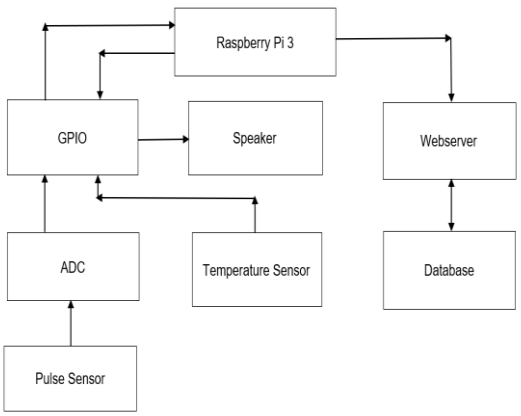

Fig 1:- Block Diagram of The Proposed Method

ISSN No:-2456-2165

The various blocks present in the block diagram are explained below.

A. Raspberry Pi

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. Raspberry Pi 3 Model B was released in February 2016 with a 64-bit quad core processor, and has on-board WiFi, Bluetooth and USB boot capabilities.

The Raspberry Pi 3 uses a Broadcom BCM2835B0 SoC with a 1.4 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache and 40 pin General Purpose Input Output pins (GPIO). The Raspberry Pi was designed for the Linux operating system, and many Linux distributions now have a version optimized for the Raspberry Pi. RPi runs Raspbian, which is based on the Debian operating system.

The Raspberry Pi is the main component which is used to interface the sensors and connect it to the webserver using IoT. This is done by connecting the sensors with the GPIO pins of the Pi.

B. Webserver

The Webserver used here is Flask which is installed in the Raspberry Pi for creating the website in which the doctors and patients interact. Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. Flask is called a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common frameworks related tools. Extensions are updated far more regularly than the core Flask program. The website is created by defining a flask app which is coded in python. The flask app also consists of html templates and CSS files.

C. Database

The Database used here is MySQL which is also installed in the Raspberry Pi. MySQL is an open-source relational database management system (RDBMS). MySQL is written in C and C++.MySQL can be built and installed manually from source code, but it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions, the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings. Using MySQL, a database is created having individual tables for doctors and patients containing their respective data.

D. Pulse Sensor

Pulse sensor is an analog sensor which is used to measure the pulse rate using a technique called as PPG (Photo plethysmography). Photo plethysmography (PPG) is a simple and low-cost optical technique that can be used to detect blood volume changes in the microvascular bed of tissue. It is often used non-invasively to make measurements at the skin surface. It is an analog sensor and because the Raspberry Pi does not take analog inputs, the pulse sensor data is given to an ADC and then to the Pi. The ADC does analog to digital conversion and sends the digital data to Pi. The ADC used here is MCP3008 which is an 8-channel 10-bit IC chip. Since it is a chip having 10-bit resolution, the readings are better.

E. Temperature Sensor

The temperature sensor used here is DS18B20. DS18B20 is a programmable one wire digital thermometer. This thermometer is directly connected with the RPi as the RPi accepts digital inputs. The Occidentalis Linux distribution for RPi includes support for the DS18B20 1-wire temperature sensor. A 4.7KΩ resistor is used to provide pull-up for the data line but connecting the resistor in between the data and the Vcc line. The chip includes the special 1-wire serial interface as well as control logic and the temperature sensor itself. Its output pin sends digital messages and Raspbian/Occidentalis includes an interface to read those messages.
The temperature data can be read from the w1_slave file present in the /sys/bus/w1/devices directory.

F. Speakers

The speakers are needed to remind the patient regarding the time to monitor their health and also to remind the patient regarding their timely intake of medicines in case they are following any prescriptions made by any medical practitioner. A module called espeak is downloaded in the Raspberry Pi for performing text-to-speech conversions. These are then outputted as audio from the speakers.

IV. HARDWARE IMPLEMENTATION

For the hardware implementation, Raspberry Pi 3 is used. The Raspberry Pi booted with Raspbian Stretch. In this project, there are two sensors used namely, temperature sensor and the pulse sensor. The pulse sensor is an analog sensor so an ADC must be used along with it as the Raspberry Pi does not take analog inputs. The temperature sensor is also connected with the GPIO pins of the Raspberry Pi. The schematic diagram for connecting both the sensors with the Raspberry Pi is shown.

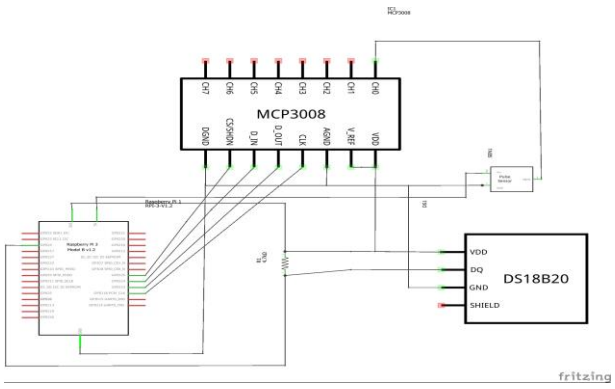

Fig 2:- Overall Circuit Schematic Diagram

V. SOFTWARE IMPLEMENTATION

For The software implementation, three software needs to be installed in the Raspberry Pi namely MySQL, Flask and ngrok. MySQL is needed for database management and table creation, Flask is the webserver in which webpages are created and ngrok is needed for tunneling the website which will be running on 'localhost'. A python program is written to create a Flask app containing HTML templates to provide structure for the webpages. Register and login pages are created for the patients and medical practitioners to sign up themselves to store their medical data and to get remote medical suggestion. Every time the user needs to view their data they need to login first. The dashboards for the patient and the medical practitioner are different. For the medical practitioner, the dashboard shows the list of patient names which he can click to view their health data and suggest some medication. The patient's dashboard shows the patient's record of health data and there is a separate messages page which will show the medical practitioners messages and suggestions to the patients. The Flask app is run on reboot using crontab and is hosted using ngrok.

VI. EXPERIMENTAL RESULTS

The results are presented in this section. The table showing a sample of patient's monitored health data is as follows:

Date Time	Temperature (F)	Pulse Rate (BPM)
2018-03-25 15:46:05	88.7	46
2018-03-25 15:46:07	88.7	50
2018-03-25 15:46:09	88.7	50
2018-03-25 15:46:11	88.7	51
2018-03-25 15:46:13	89.15	54
2018-03-25 15:46:15	89.7116	60
2018-03-25 15:46:17	90.5	68
2018-03-25 15:46:19	91.4	77
2018-03-25 15:46:21	92.1866	78
2018-03-25 15:46:23	92.75	78
2018-03-25 15:46:25	93.0866	78
2018-03-25 15:46:27	93.425	78
2018-03-25 15:46:28	93.65	78
2018-03-25 15:46:30	93.875	79
2018-03-25 15:46:34	94.2116	77
2018-03-25 15:46:37	94.4366	77
2018-03-25 15:46:40	94.55	78
2018-03-25 15:46:42	94.6616	78
2018-03-25 15:46:44	94.775	78
2018-03-25 15:46:46	94.8866	78
2018-03-25 15:46:48	94.8866	78
2018-03-25 15:46:50	95	78

Table 1. Table Showing the Monitored Health data

The screenshots of the webpages are shown:

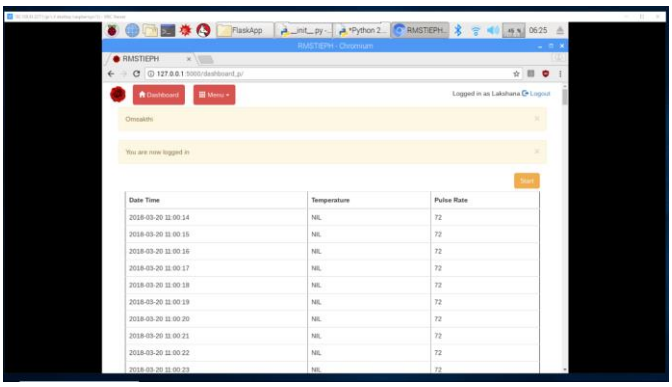

Fig 3:- Patient's Dashboard showing Patient's Health Data

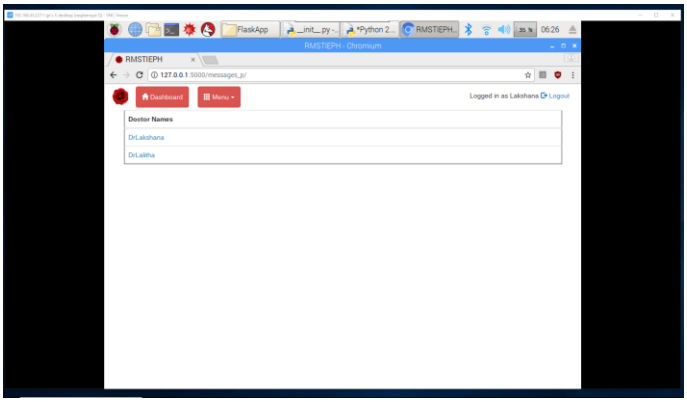

Fig 4:- Patient's Message Page which shows a list of Doctors

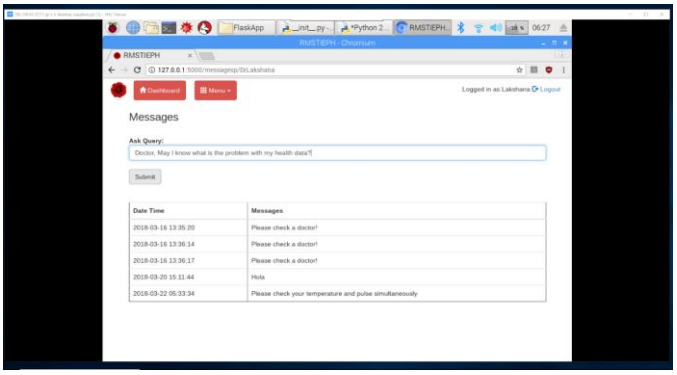

Fig 5:- Patient's Message Page which shows Messages from the Specified Doctor

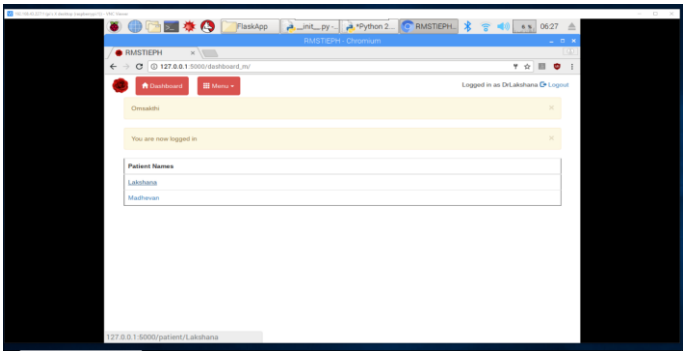

Fig 6:- Doctor's dashboard

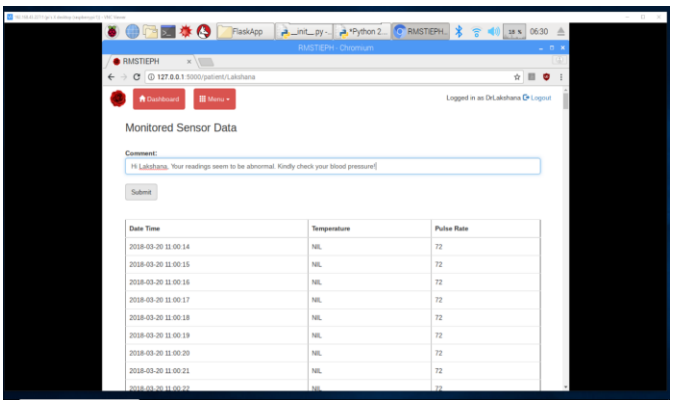

Fig 7:- Page Displaying Health Data of the Chosen Patient

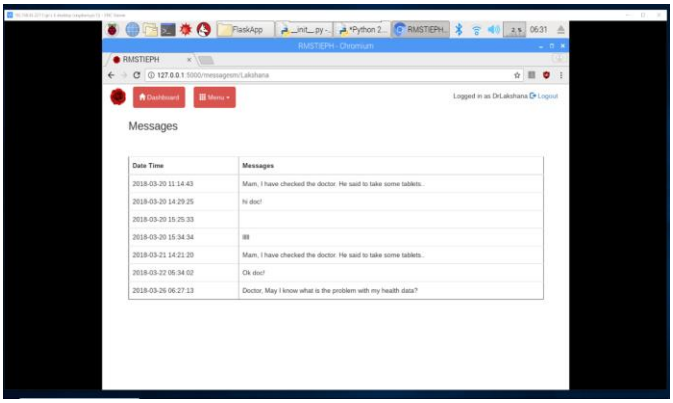

Fig 8:- Doctor Viewing Queries of The Chosen Patient

VII. CONCLUSION

This research investigates a novel approach for personal health monitoring and remote medical suggestion using a website for interaction. After sending the health data to the database, both the patient and the doctor can view the patient's health data in their web dashboard. Any medical practitioner or doctor can suggest medication by viewing the patient's health data. Since multiple doctors can suggest a medication, it will be timely for the patient to take the doctor's medication if his/her personal medical practitioner is not available at the moment. This method is more efficient compared to the other methods.

REFERENCES

[1]. Lymberis, "Smart Wearables For Remote Health Monitoring, From Prevention To Rehabilitation: Current R&D, Future Challenges", 4th Annual IEEE Conf on Information Technology Applications in Biomedicine.

[2]. Bingchuan Yuan, John Herbert, "Web-based Real-time Remote Monitoring for Pervasive Healthcare", 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops).

[3]. Kuo-Hui Yeh, "A Secure IoT-based Healthcare System with Body Sensor Networks", IEEE Access.

[4]. Pierre Lepage, Dominic L´etourneau, Mathieu Hamel, Simon Bri`ere, H´ el`ene Corriveau, Michel Tousignant and Franc¸ois Michaud, "Telehomecare Telecommunication Framework – From Remote Patient Monitoring to Video Visits and Robot Telepresence", 2016 IEEE 38th Annual International Conference of the Engineering in Medicine and Biology Society (EMBC).

[5]. Udit Satija, Barathram.Ramkumar, and M. Sabarimalai Manikandan, "Real-Time Signal Quality-Aware ECG Telemetry System for IoT-Based Health Care Monitoring", IEEE Internet of Things Journal.

[6]. Warish Patel, Sharnil Pandya, Viral Mistry, "i-MsRTRM:Developing an IoT based iNTELLIGENT Medicare system for Real-Time Remote Health monitoring", International Conference on Computational Intelligence and Communication Networks