Personal Health CareTaker Bot Using IOT

Anshul Nagar
Department of electronics and telecommunications engineering
Shri Govindram Seksaria Institute of Technology and Science
Indore, M.P. ,India
Nagaranshul9@gmail.com

Himanshi Agrawal
Department of electronics and telecommunications engineering
Shri Govindram Seksaria Institute of Technology and Science
Indore, M.P. ,India
ahimanshi631@gmail.com

Sazal Jain
Department of electronics and telecommunications engineering
Shri Govindram Seksaria Institute of Technology and Science
Indore, M.P. ,India
sazaljain16@gmail.com

Somya Jain
Department of electronics and telecommunications engineering
Shri Govindram Seksaria Institute of Technology and Science
Indore, M.P. ,India
jainsomya.618@gmail.com

Soumya Patel
Department of electronics and telecommunications engineering
Shri Govindram Seksaria Institute of Technology and Science
Indore, M.P. ,India
soumyaapatel25@gmail.com

***Abstract*— This paper presents an Automatic Medicine Dispenser and Health monitoring system prototype enabled with IOT capability. There are many challenges that adults face, including taking their medication on time. Older people often forget to take their medication on time and find it difficult to remember if they have had their medication, which can sometimes lead to overdose and serious health problems. There are a few expensive pharmaceutical providers available in the market right now. However, the majority of older people around the world do not even know about such products and still turn to prescription drugs. A few types of pharmaceutical suppliers are commercially available worldwide. However, they have a few problems that need to be resolved. These issues can be resolved using a reliable, affordable and affordable Automatic Medicine Dispenser and health monitoring system in such a way that adults do not need to depend on anyone else. The product is designed to ensure that the amount and time of the pills to be released can be controlled and monitored using the app, making things easier for everyone, including children working abroad. Also, it offers a clear bet between customer and parent statistics as it will promptly notify the caregiver in case a patient misses a pill.**

I. INTRODUCTION

As a person enters old age, it becomes increasingly vital for them to undergo standard medical health checkups. Since it may be time-consuming and difficult for most people to get regular health checkup appointments, IoT-based arrangements can be beneficial to individuals for routine health checkups. IoT technology has developed into an imperative innovation with applications in numerous areas. Pulse rate and body temperature are the most basic markers of human health. The pulse rate, also known as the beat rate, is the number of pulses per minute. An IOT based pulse oximetry, temperature monitoring and medicine time reminding device was demonstrated. The patient's pulse rate and temperature was measured using a pulse rate sensor and temperature sensor respectively in this system, and the data were sent to the cloud . The measured information was delivered to the web application.The main objective of this research is to develop and implement a novel IoT-based smart health monitoring system for patients based on human body temperature, pulse, and SpO2. The device can display measured human body temperature, oxygen saturation level, and pulse rate on it, also the data can be accessed through a web application, which has been developed so that the patient can seek medical attention even if the specialist is physically unavailable. Patients can analyze the measured oxygen saturation level, pulse rate, and body temperature to avoid critical health conditions. In a Personal health caretaker bot, when it is time to take the medicine, the device can be designed into a small room that can be easily opened.. The patient is usually notified by a loud alarm signal. When a patient presses the medicine dispenser button the required medicine gets dispensed and the alarm stops. Also , IoT has been widely used to coordinate advanced medical services and to provide intelligent and effective health care services to individuals. Improved nerves can be worn or embedded in patients' bodies, to keep their health in check. Information collected in this way can be analyzed, compiled and excavated to identify diseases early. Diagnostic algorithms help physicians make treatment more personal and help to make health care safer, at the same time, more effective.

II. OBJECTIVES

The main objectives that needs to be fulfilled for the project are as follows -

- It must be user friendly easy to operate and affordable

- It will be able to remind to take the medicines on time and dispense the required medicines properly and on time
- It is also capable of taking and displaying pulse rate and blood oxygen readings.
- It also contains a Body temperature sensor for temperature readings.
- Since it must have IOT capability we can set the medicine reminder and dispensing time and also monitor and track data and display it in a web app.
- The web app must be able to handle and manage various devices and their data and access permissions. it must be easy and secure to use.
- The LED display in the project also displays other info. Like time and prescription messages that will be sent using a web app.

III. TECHNOLOGICAL TOOLS IN GENERAL

A. Arduino Embedded C

The term "Arduino-compatible coding" refers to all Arduino and Arduino-compatible microcontroller boards that can be programmed and uploaded using Arduino IDE. Arduino boards are programmed in "C." C is a popular system programming language that has minimal execution time on hardware in comparison to other high-level programming languages. Much like other microcontrollers, the AVR microcontrollers housed in Arduino boards are programmed in a subset of C. A general term for such subsets is "Embedded C" because they apply to programming embedded controllers. The language in which Arduino is programmed is a subset of C and it includes only those features of standard C that are supported by the Arduino IDE.

B. Firebase

It is nothing but a set of tools used to create, improve and transform your app. It covers most of those processes and services that programmers often create themselves, but do not want to, because they often like to focus on the application. The cloud handles processes and services such as websites, authentication, etc. The client SDKs provided by Firebase interact directly with these background services, without the need to establish any middleware between the service and your application.

C. Python

Python is a high level programming language which is easy to code and has extensive library support. In context for this project we will use python for coding the backend server for telegram bot and for hosting web apps using flask framework. Flask is a small web framework written in Python. They are classified as microframework because they do not require special tools or libraries. It does not have a base for website summaries, form verification, or any other sections where existing third-party libraries provide similar services.

IV. PROJECT WORKFLOW

A. Hardware Level

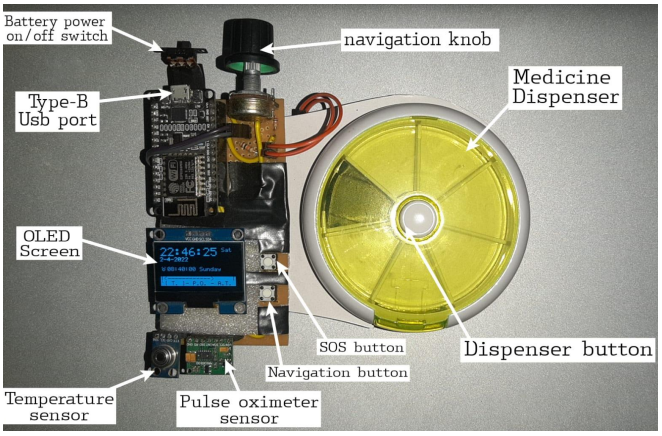


Figure 1 Personal health caretaker bot hardware

The personal-health caretaker bot consist of a switch to turn it on or off a usb-port for power supply and internal battery charging , an O-led display and two buttons and one knob for interaction with bot, from two given buttons one navigation button is used to select options and skipping medicine time and the other button is reserve only for SOS messages only. The knob is used to navigate between various options. It also contains a medicine dispensing assembly with an inbuilt button to dispense medicine, when this button is pressed the signal is sent to bot and it updates it on cloud during the medicine time. Besides this it also contains temperature and pulse oximetry sensors for monitoring health.

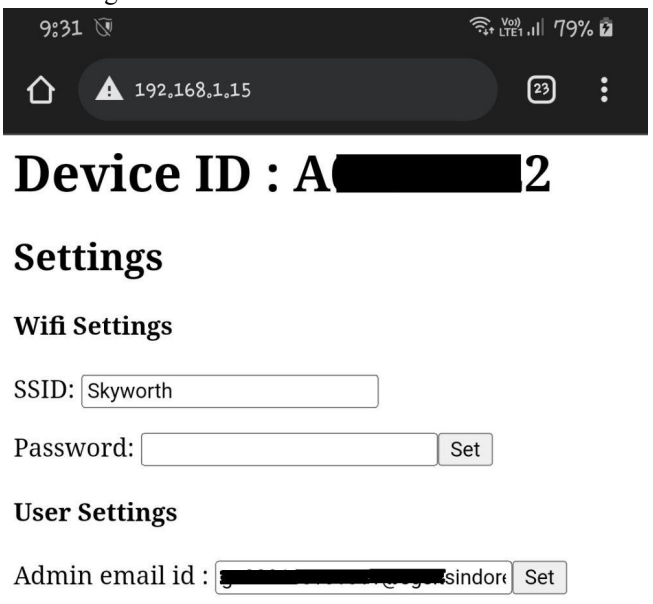


Figure 2 Accessing device settings from web browser by entering device local IP over LAN

When the bot is turned on first it shows a greeting message just after that it tries to connect to a wifi network which is saved in its eeprom memory or to a network whose SSID is "reset" and password is "reset@123" . Once it gets connected to any one of the networks it displays a local ip address of the device where you can access its settings and

can set the wifi connection to which you want to connect and the admin account email for the device to get access to the device from the web app. It also displays the next medicine alarm time after that and then the admin user for the device. After that we can see the home screen of the device in oled it shows the current time in 24 hours format day and date then in next to alarm clock icon it shows the next medicine alarm time with day and in below is the navigation menu.

Figure 3 Home Screen at three different instances of time with different options on select

As shown in figure the home screen menu contains 5 different options they are :-

- T. – for taking temperature readings
- P.O. – for taking SPO2 and heart rate readings
- A.T. – for seeing all medicine alarm times
- M. – for seeing last note or message
- R. – for seeing last temperature and pulse oximetry readings.

The option inside the square brackets is the current option selected and on pressing the navigation button will enter into that option.

When T. option is opened it immediately starts taking the reading and once it gets stable readings or after few seconds the final reading has been taken then it asks to upload the readings to cloud you can simply use knob to set it to yes or no the selected option font size is bigger after few seconds the action has been taken according to the option selected.

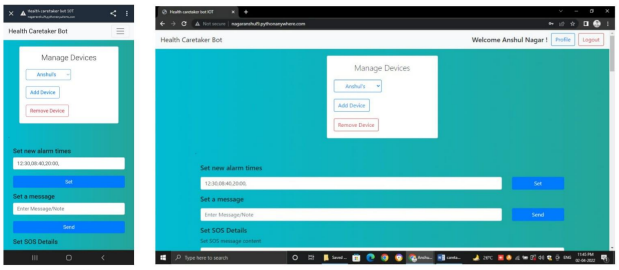
When P.O. option is opened the pulse oximetry sensor gets activated u have to place your finger on the sensor for 10 seconds to get a stable reading once the reading is recorded you can send it to cloud if you want just like with T. option.

On opening A.T. M. or R. option the result will display on screen for a few seconds then returned to the home screen.

B. *Web Application*

Link to Web app
https://personal-healthcaretaker.web.app/
https://personal-healthcaretaker.firebaseapp.com/
http://nagaranshul9.pythonanywhere.com/

Responsive web app suited for all devices



Mobile version

Desktop or tablet version

Figure 4 Personal Health Caretaker bot web app

To handle access, manage and operate the personal health caretaker bots effectively the personal health-caretaker project has its own web based application which can be easily accessible on all devices. It is easy to manage bots in a secure way. In order to get started with a web app you need to first sign up to the web app and once your account has been created you have to verify your email id in order to get access to the application. Once you have verified your email you can manage your multiple devices linked with your id, in order to access a new device first you need to add your device by filling the details in add devices option, it asks for a device id which is a unique Alphanumeric ID for identifying device (you can find your device id by accessing settings as shown in figure 2), once you filled all the details click on add device to send a request to the user who has the admin access for the device. To get the admin access set your registered and verified email id in the admin email id settings of your device. If the admin accepts the request you can access the device also only the admin of the device can see who has the access to the device and can revoke the access at any time. Once you can access a device you can set alarm times, send messages to the device, set SOS message and SOS message receivers , and see the medicine taken status, temperature,SPO2 and heart-rate readings . you can also link your telegram id to the web app in order to access device using the telegram bot service.

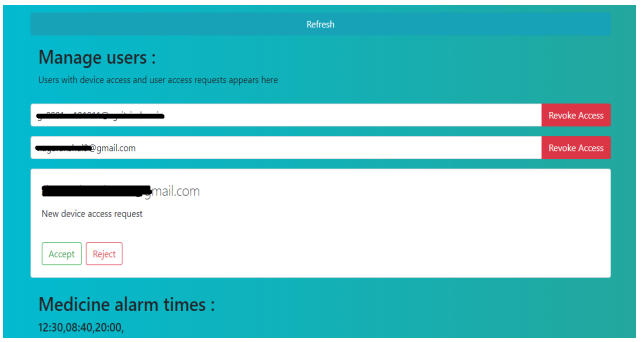


Figure 5 Access to Manage Users tab only to device admin id to manage device access to its all users

In order to access your devices using telegram you need to add your telegram chat id with telegram health caretaker bot in the profile settings Link telegram id

Scan the below QR Code to add a Health caretaker bot to your telegram. Once you add a health care taker, both get

the chat id by selecting Get telegram chat id in the menu option then fill this chat id in the web app link telegram id option and now you can access your devices using telegram only. You can also receive SOS messages from devices here.

Figure 6 Scan to add bot to your telegram

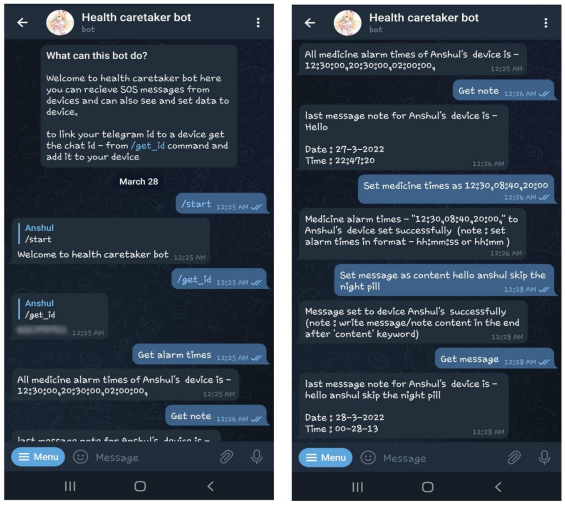


Figure 7 Using telegram bot to access devices

V. PROJECT STRUCTURE

A. Overview

The development of a personal health caretaker bot project requires integration of various technologies and cloud based services. The hardware itself is integrated with firebase for storing data on cloud realtime-database and for retrieving data . All the data gathered by device is available on firebase database which can be accessed or modified using web application and telegram bot api hosted using python on python anywhere free cloud service.

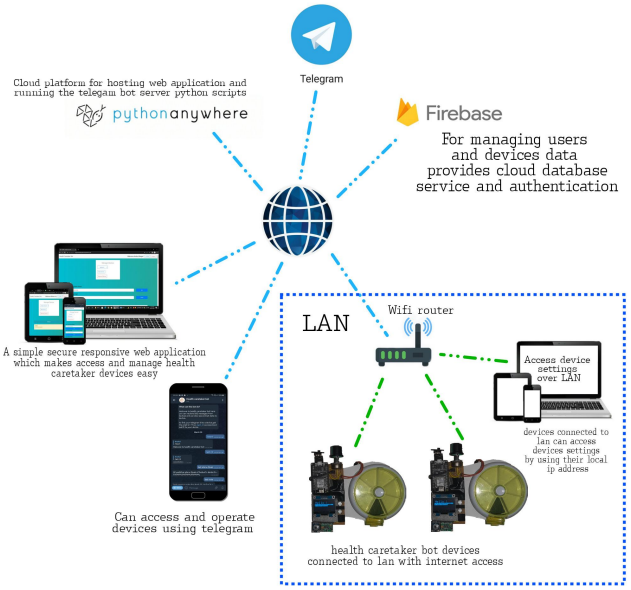


Figure 8 Overview architecture of personal health caretaker bot project

B. Hardware

1. Components Required :

I) NodeMCU

The term Node MCU is a combination of the two words “node” and “MCU” which is a short term Microcontroller unit. Firmware and prototyping design suites are both open source platforms. The Node MCU is the most expensive IoT platform. The hardware is based on the ESP 12 module. It is a single board microcontroller with an active XTOS system. The CPU microcontroller is ESP8266. Lua Typing is used by firmware. Node MCU supports the 32-bit ESP module. The Node MCU has an integrated WiFi board, ESP8266, so the Node MCU gets its widespread use in IoT applications. Fig.9 shows the NodeMCU image with the integrated ESP8266.

Features of ESP8266 :

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

Figure 9.1 NodeMCU with integrated ESP8266

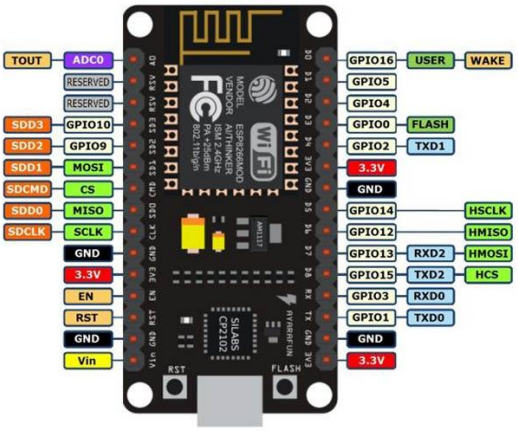


Figure 9.2 NodeMCU Pinout Diagram

II) MAX30100

The MAX30100 is an integrated pulse oximetry and heart rate monitor sensor solution. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals. The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software with negligible standby current, permitting the power supply to remain connected at all times.

Features and Benefits of MAX300 :

- Complete Pulse Oximeter and Heart-Rate Sensor Solution Simplifies Design
- Integrated LEDs, Photo Sensor, and High-Performance Analog Front -End
- Tiny 5.6mm x 2.8mm x 1.2mm 14-Pin Optically Enhanced System-in-Package
- Ultra-Low-Power Operation Increases Battery Life for Wearable Devices
- Programmable Sample Rate and LED Current for Power Savings
- Ultra-Low Shutdown Current (0.7µA, typ)
- Advanced Functionality Improves Measurement Performance
- High SNR Provides Robust Motion Artifact Resilience
- Integrated Ambient Light Cancellation
- High Sample Rate Capability
- Fast Data Output Capability

Figure 10 MAX30100 Pulse Oximeter sensor

III) OLED-Display (SH1106 driver) :

SH1106 is a single-chip CMOS OLED/PLED driver with controller for organic/polymer light emitting diode dot matrix graphic display system.SH1106 consist of 132 segments,64 commons that can support a maximum display resolution of 132x64, it is designed for common cathode type oled panel. It is embedded with a contrast control display RAM oscillator and efficient DC-DC converter, which reduces the number of external components and power consumption. below is some of its features listed

- 132x64 dot matrix panel
- Embedded 132x64 bits SRAM
- Operating voltage

  -Logic voltage supply Vdd1 = 1.65V - 3.5V

  -DC-DC voltage supply Vdd2 = 3.0V - 4.2V

  -OLED Operating voltage supply:

  External Vpp supply = 6.4V - 14V

  Internal Vpp generator = 6.4V - 9V

- Maximum segment output current: 200µA
- Maximum common sink current: 27mA
- 8-bit 8080-series parallel interface, 3-wire & 4-wire serial peripheral interface, 400KHz fast I2C bus interfaceI
- Wide range of operating temperatures: -40 to +85°C
- Programmable frame frequency and multiplexing ratio
- 256-step contrast control on monochrome passive OLED panel
- Programmable Internal charge pump circuit output
- Vertical Scrolling
- On-chip Oscillator
- Low power consumption , Sleep mode: <5µAI , VDD1=0V VDD2=3.0V – 4.2V: <5µAI ,VDD1,2=0V, VPP=6.4V–14.0V: <5µA1
- Available in COG form, thickness: 300umI

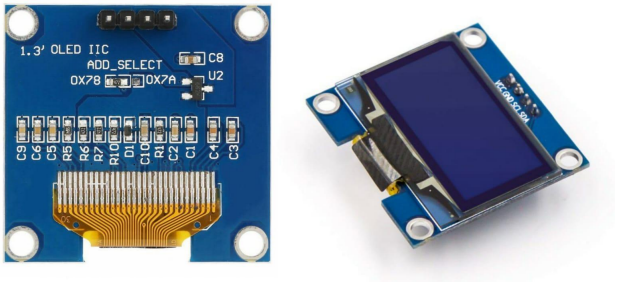


Figure 11 SH1106 I2C Dotmatrix OLED Display

IV) MLX90614

The MLX90614 is an InfraRed thermometer for non contact temperature measurements. Both the IR sensitive thermopile detector chip and the signal conditioning ASSP are integrated in the same TO-39 can.

Features and Benefits of MLX90614

- Small size, low cost
- Easy to integrate
- Factory calibrated in wide temperature range:
- -40 to 125 ˚C for sensor temperature and
- -70 to 380 ˚C for object temperature.
- High accuracy of 0.5°C over wide temperature
- range (0..+50°C for both Ta and To)
- High (medical) accuracy calibration optional
- Measurement resolution of 0.02°C
- Single and dual zone versions
- SMBus compatible digital interface
- Customizable PWM output for continuous
- reading
- Available in 3V and 5V versions
- Simple adaptation for 8 to 16V applications
- Power saving mode
- Different package options for applications and
- measurements versatility
- Automotive grade

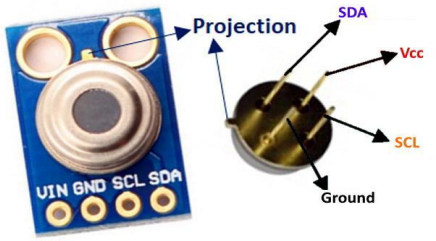


Figure 12 MLX90614 Infrared Temperature sensor

V) All Components list

TABLE 1 Components required

No	Component	Description
1	NodeMCU-ESP 8266	Microcontroller unit with inbuilt wifi (IOT development board)
2	MAX30100	Pulse-Oximetry sensor for SPO2 and heart rate readings
3	OLED-Display (SH1106 driver)	For display
4	MLX90614	Contactless IR temperature sensor
5	Knob Potentiometer	10k value , used for navigation purpose
6	buzzer	For alarm purpose
7	Push buttons (x2)	For navigation and operation purpose
8	Safety press button	In medicine dispensing button
9	Medicine dispenser assembly	For dispensing medicine
10	3.7v rechargeable battery	For powering the device
11	Resistors	1k,2k2,100E,100E
12	PCB (Custom made or Zero) and Soldering cost and other miscellaneous small items	Variable

2. *Circuit Diagram*

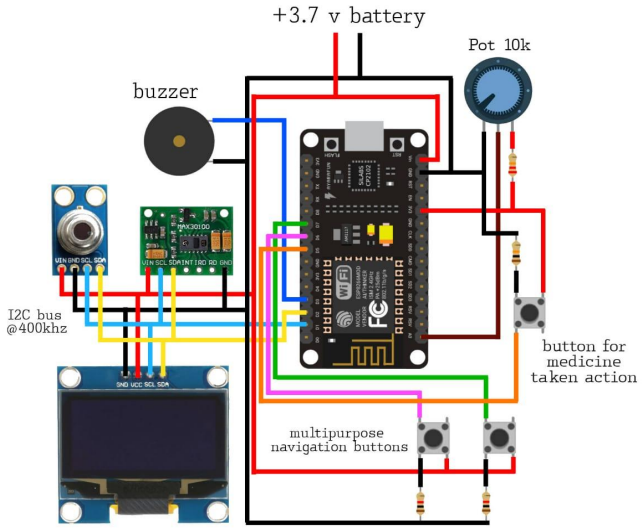


Figure 13 : Circuit diagram of personal health caretaker bot hardware

3. *Connections*

- GPIO 5 (D1 nodeMCU) -> SCL (I2C bus)
- GPIO 4 (D2 nodeMCU) -> SDA (I2C bus)
- GPIO 0 (D3 nodeMCU) -> Buzzer(+)
- GPIO 14 (D5 nodeMCU) -> press button signal medicine dispenser
- GPIO 12 (D6 nodeMCU) -> SOS button
- GPIO 13 (D7 nodeMCU) -> Menu select button
- ADC 0 (A0 nodeMCU) -> Potentiometer key
- VIN or VCC of I2C devices -> VIN of NodeMCU

4. *Embedded C coding using Arduino IDE*

I) Main operations/functionalities to be performed by the code :

- Connecting to wifi
- Get current time using NTP client
- Getting data from sensors using I2C comm.
- Interfacing i2c oled display and buzzer for output
- Interfacing buttons and potentiometer for navigation and operation of device
- Designing menu for various options
- Using EEPROM to store data like Wifi configurations and user email ID permanently
- Setting up local esp8266 web server for setting configurations for wifi and firebase
- Accessing firebase functionality like Authentication and RTDB(Real Time Database)
- Sending Device and sensors data to RTDB
- Getting real time updates from database using streams
- Processing the data and show output according to data received
- Managing and Updating the Alarm times and output through buzzer and oled

II) Flowchart

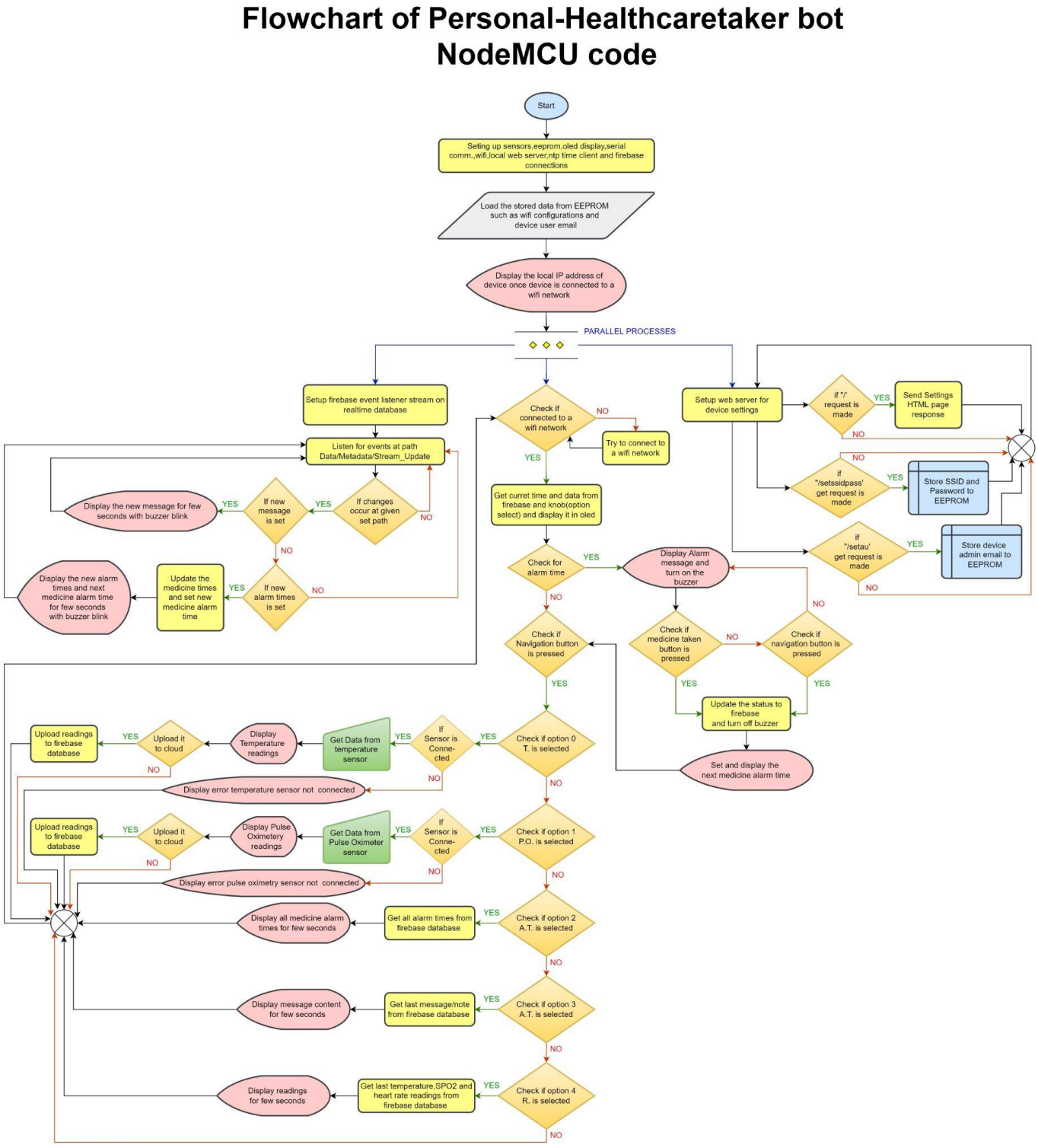


Figure 14 : Flowchart of ESP8266 code of personal health caretaker bot hardware

III) Libraries Required and Modifications :

TABLE 2 Libraries required

Library	**Version**
SPI	-
Wire	-
GyverOLED	1.3.2
ESP8266WiFiMulti	-
ESP8266mDNS	-
NTPClient (by fabrice Weinberg)	3.2.0
WiFiUdp	-
Firebase_ESP_Client (by Mobizt)	2.5.5 (may cause some issues in newer versions)
ESP8266HTTPClient	-
DFRobot_MLX90614	1.0.0
MAX30100_PulseOximeter (by OXullo Intersecans)	1.2.1
addons/TokenHelper	-
addons/RTDBHelper	-
ESP8266WiFi	-
ESP8266WebServer	-
EEPROM	-

Modifications : Make sure that the I2C BUS SPEED for all the I2C devices is same for MAX30100 bus speed is set to 400000UL by default change I2C_BUS_SPEED from 400000UL to 100000UL in MAX30100.h file present in sensor's library folder

IV) Code

Link to documented esp8266 nodemcu code : https://github.com/Anshulnagar00/Personal-Healthcaretaker/blob/master/Code/ESP8266%20code/personal_healthcaretaker_bot.ino

C. Firebase Setup

1. Creating a new firebase project

2. Setting up the firebase Authentication

- Set up sign-in method add email/password as sign-in provider

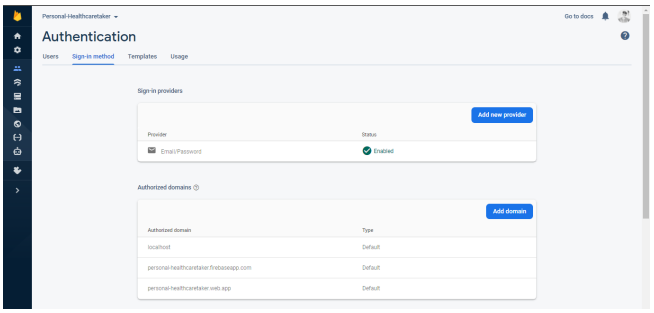


Figure 15 : Sign-in method tab after email/password provider is set

- Create a admin access account by clicking add user then enter email and password you want
- Once the user is added you can see the User UID which will be needed to give access to database in security rules later

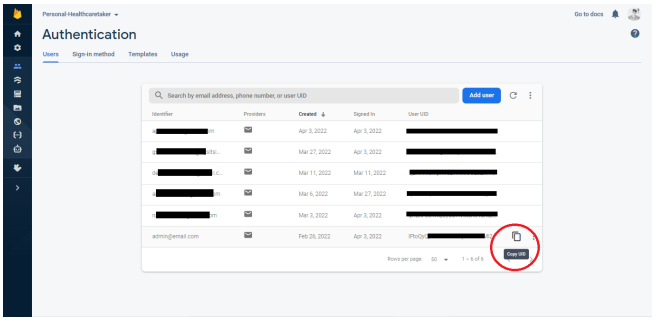
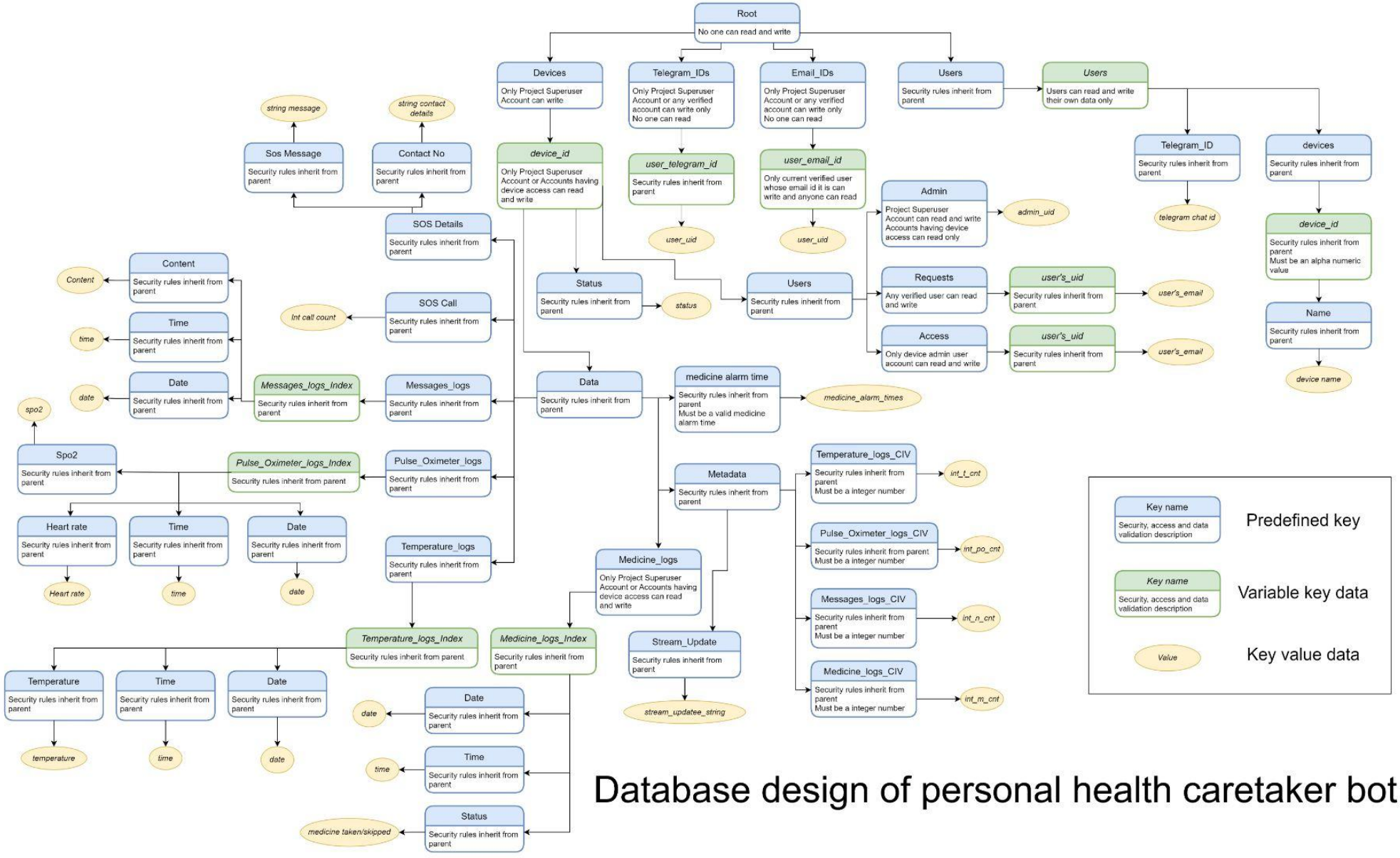


Figure 16 Users tab where all users details can be seen copy the user uid from icon shown in red circle

3. Setting up the firebase real time database

Firebase realtime database is a no-sql non-relational database which does not store data in form of tables; it uses a key value pair type structure or similar to JSON structure to store data.
It is free to use to some limit on the amount of data downloaded and the number of queries processed in a particular time interval so we have to keep this in mind while using firebase realtime database in our project. It also provides security rules functionality to secure the data and give only limited and necessary access to our database to the users.

Figure 17 Database Design of personal health caretaker bot



Database design of personal health caretaker bot

D. Web App development

For the development of Responsive web app the following technologies is been used :

1. HTML
2. CSS
3. Javascript
4. Bootstrap
5. Firebase

Below is the html web app code link

E. Telegram chat bot development

Development of telegram chat bot is done using python programming language in backend.but first we need to create a new bot in telegram using BotFather bot Search for BotFather in telegram and send '/newbot' command to create a new bot once bot is created it provides a HTTP API token for the bot this API token is used for automating queries from bot using programming languages such as python. below is the main tasks done in order to prepare the functioning bot

- Automate reply messages and queries from telegram users through python script
- Accessing the firebase real time database from python and perform operations through it
- Implementing the SOS functionality by using event listeners functionality of firebase RTDB
- Running the Bot script on cloud console using pythonanywhere.com for free of charge

Code link for backend telegram bot : https://github.com/Anshulnagar00/Personal-Healthcaretaker/tree/master/Code

F. Hosting

1. Setting up a python flask app on local machine:

For local testing put the html web app file and py file in same directory then run the below python code

Code : -

```
from flask import Flask,render_template
app = Flask(__name__)
@app.route("/")
def home():
    return render_template("app.html")
# delete the below lines of code for
deployment in pythonanywhere
if __name__ == "__main__" :
    app.run(host="0.0.0.0",port=5500)
```

2. Deploying the flask app on pythonanywhere.com

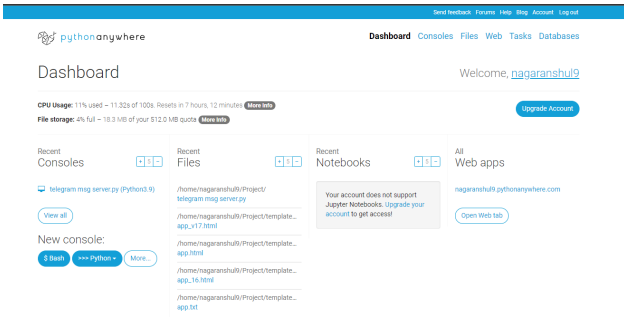


Figure 18 Pythonanywhere.com dashboard

First Create a Project folder inside which is python flask code file named 'flask_app.py' is present and also a directory named templates in which the html file must be there then go to web tab and update the working,source code and static directory and then reload the webapp Below is the default link to web app hosted using pythonanywhere http://nagaranshul9.pythonanywhere.com/

3. Hosting web app on firebase

For hosting static web pages on firebase hosting we need to first install the firebase cli in the command line using npm. Once the installation is done login to firebase and then run `firebase init hosting` command to setup the hosting once done with hosting the some files will be created and a folder `public` inside that folder replace the `index.html` file with your web application file with the same name as `index.html` and then deploy the app using `firebase deploy` command.once it is complete go to hosting tab in firebase console inside your project to see status of web app deployed

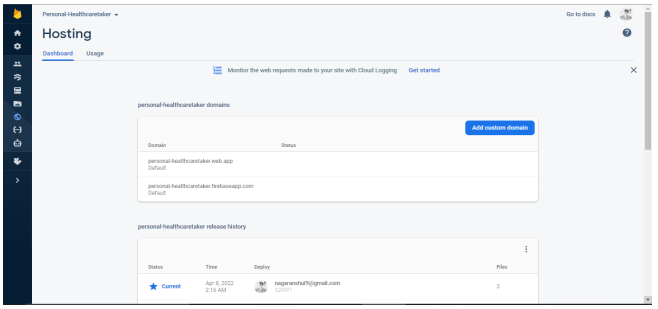


Figure 19 Firebase hosting tab in firebase project console

Below is the default hosted urls for the personal healthcare taker web app hosted using firebase hosting https://personal-healthcaretaker.web.app/ https://personal-healthcaretaker.firebaseapp.com/

VI. APPLICATIONS

Carebots are an emerging breed of robotics catered to the specialized care of the elderly and individuals who face cognitive or social decline.

The device can be portable and it can be used as a portable bot which can easily be used to take medicines on time or to take temperature ,pulse rate or SPO2 readings anywhere at any time. it easily fits inside a carry hand bag

All the data regarding sensor readings and medicine taken status can be accessed from any device using web based applications with necessary security functions. Only users registered and verified to web app and having necessary access to device can operate that device using web application this feature is useful for the guardians or doctor of the patient to monitor the health and activity regarding medicines taken can be monitored by them easily

During such COVID times where the staff is limited in hospitals and there are too many patients then in order to operate them properly and track their medicine taken activity or to track their health or to handle an emergency call made by patient these bots can do their job with minimal effort and less human to human contact chances. which drastically helps and assists the nurses and doctors to manage patients effectively

These bots can also be operated and used where other smart devices are restricted. For example in a prison environment the prisoners with health issues can have these bots so that their health can be monitored and they can also make sos calls and can get a quick response in time.you can also disable this feature if they misuses it and spams SOS calls.

VII. RESULTS

After evaluating the system separately, it was determined that the system was efficient enough. This means that the system construction and project implementation process was correct; therefore, user data is rated satisfactorily. The complete system has two main components combined. With this system, users can get a limited amount of their key features through the web app and the OLED screen of the hardware device..

The system was tested on a real human test subject. The user experience and measured values of vital signs for the user. The device and web app show the measured heart rate, SpO2, and temperature, in which the user can see the displayed results from the system. This system passes the data from the personal health caretaker bot device to the cloud server (Firebase RTDB) and then the data can be accessed using a web app or telegram, which is one of its crucial parts.
The device also reminds users to take the medicines according to the medicine alarm times set by the user using a web app or telegram accurately and on time. Also the updation of alarm times or setting a note or message to the device from a web app also reaches and gets updated real time in the device.
The personal health caretaker bot also incorporates the SOS functionality which sends an SOS message to add users telegram instantly when the SOS button is pressed on the device.
Through this, users who have access to the device can obtain the required results through a web app; thus, this method is user-friendly and convenient.Also the system meets all the data security requirements and the users can only access their own data and the data of devices whose access it has. The user's email verification is necessary to get access to web applications. This device was used by five people aged between 19 and 47 years. It provided precise values for all features that have been included in this system.

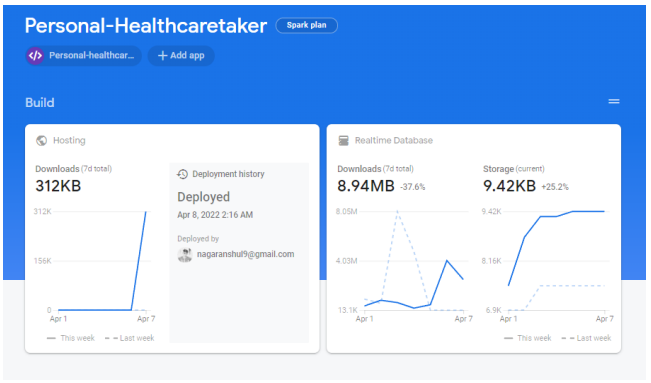


Figure 20 Firebase firebase project console

Figure 21.1 Readings taken by temperature sensor.

Figure 21.2 Readings taken by pulse oximeter sensor.

Figure 21.3 Latest record of messages or notes set to device

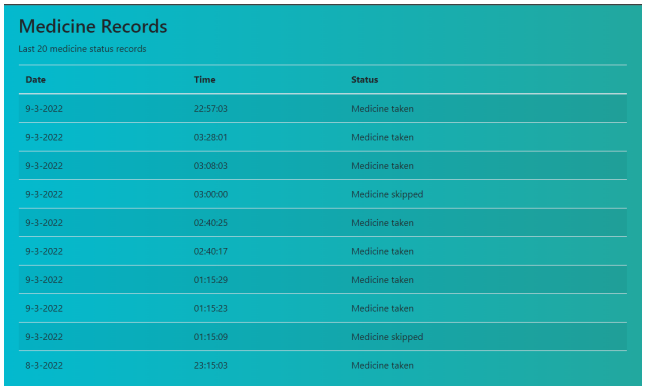


Figure 21.4 Medicine taken/skipped records.

VIII. CONCLUSIONS

With the use of IoT based personal health care bot, adherence to medication will improve, especially in elderly patients and patients with chronic illnesses and periods, which will ensure effective treatment. Insurance companies will definitely benefit from this as it will be helpful to their clients living a healthier and better life without the catastrophic risks of missing medication and early determination of health related risks from analysing medical records. Lastly, the same user interaction across all apps and devices is clear, easy to use, accurate and easy to use, even for older patients. The design is flexible and gives the user the freedom to add more containers and be open to other enhancements in the future.

IX. ACKNOWLEDGMENT

First and foremost we would like to thank God for his blessings, showered on us during the project. We are heartily thankful to our mentors Dr. LD. Malviya and Mrs. Vaishali Naik Ma'am , whose encouragement, guidance and support from the initial level enabled us to develop an understanding of the project. We would like to thank our teammates who were always ready to do a lot of research .Secondly, we would also like to thank our parents and friends who helped us a lot in this project research.

Lastly, we offer our regards and blessings to all of those who supported us in any respect during the project.

X. REFERENCES

[1] Wissam Antoun, Ali Abdo, Suleiman Al-Yaman, Abdallah Kassem, Mustapha Hamad and Chady El-Moucary, "Smart Medicine Dispenser (SMD)", 2018 IEEE 4th Middle East Conference on Biomedical Engineering (MECBME), pp. 20-23, 2018.

[2] A. V. Dhukaram and C. Baber, "Elderly Cardiac Patients' Medication Management: Patient Day-to-Day Needs and Review of Medication Management System," 2013 IEEE International Conference on Healthcare Informatics, pp. 107-114, 2013.

[3] World Health Organization, http://www.who.int/medicines/en/ , accessed on July 21, 2017.

[4] S. L. Gray, J. E.Mahoney, and D. K.Blough, "Medication Adherence in Elderly Patients Receiving Home Health Services following Hospital Discharge." Annals of Pharmacotherapy, 35(5), pp. 539-545, 2001.

[5] https://iotdunia.com/iot-architecture/

[6] https://www.researchgate.net/figure/nternal-architecture-of-SQLite_fig1_2 71891972 [

[7] C. Fărcaú, I. Ciocan, N. PalaghiĠă and R. Fizeúan, "Weekly electronic pills dispenser with circular containers," 2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME), pp. 125-129, 2015.

[8] N. B. Othman and O. P. Ek, "Pill dispenser with alarm via smart phone notification," 2016 IEEE 5th

[9] https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf

[10] https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet

[11] https://www.velleman.eu/downloads/29/infosheets/sh1106_datasheet.pdf

[12] https://datasheetspdf.com/pdf-file/774392/Melexis/MLX90614/1