IoT Project Report for Medicine and Health tracker

Student 1: Aparna Chinya **Student id: 10506242**

Student 2: Darshan Shah **Student id: 71761462**

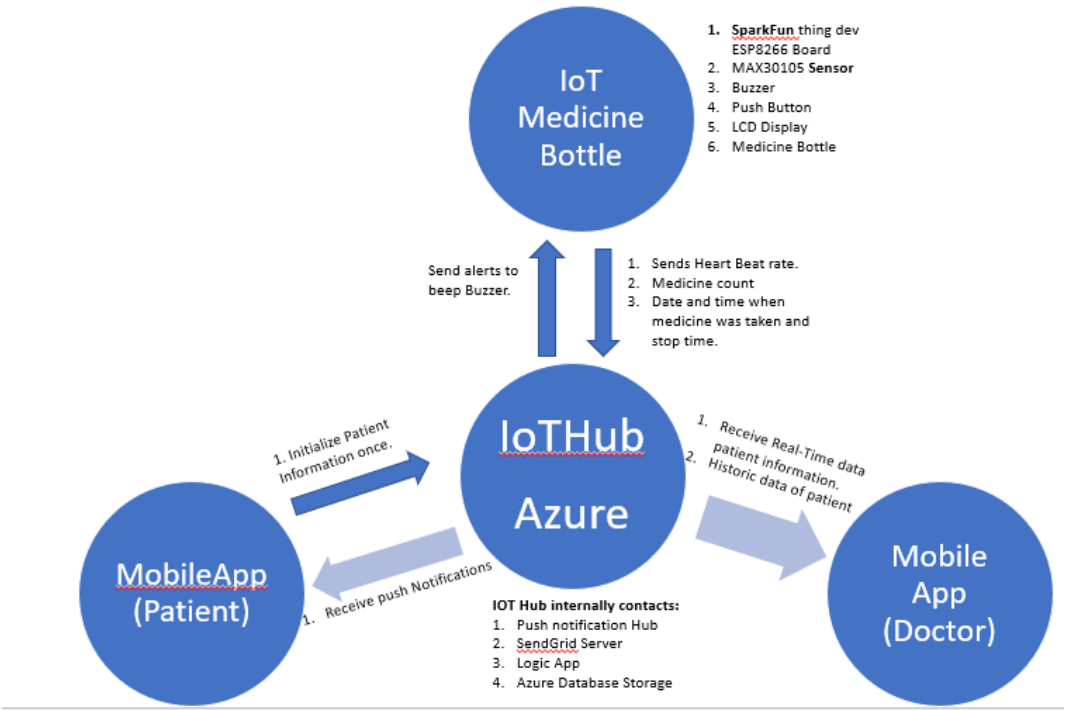
Introduction:
There arises situations where people have to keep track of their schedule when taking medicines. Either they tend to forget taking it or he might forget where the medicine box is or might not know if medicines are over or not.

On the other hand, doctors might not get real-time data of a patient. Remote information from the patient to the doctor is not often sent and only by meeting in person does the information flow happen.

Purpose:
Our system solves the above **two** problems.
1. The main purpose of this system is to make sure that user can take his/her medicines properly and on **timely** basis.
2. The doctor gets to know the exact time when a patient has consumed the medicine. There are various analysis options like delay in taking medicines, total count of the medicines remaining. The start date and stop date when the patient took the medicines. The doctor also gets to know the real-time information of patients **heartbeat.**

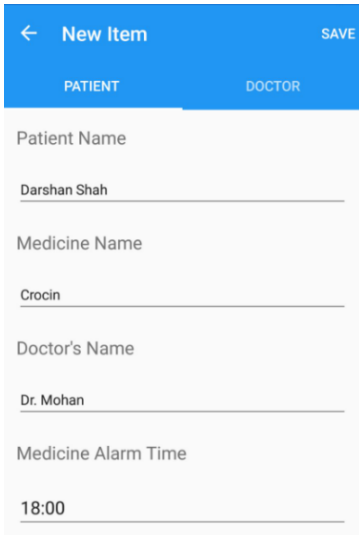
Intended Audience: This system can be used by anyone who is consulting a **doctor** and any doctor who is treating a **patient**. It builds an ecosystem for patients-doctors.

Architecture and Working:
We have built a cross platform mobile app (runs on Android, Ios and Windows) which can be used by both the patients and doctors. We will prove a smart IoT Medicine bottle.



Patient View:
On the client side, we are taking the user related details like Username, contact details, medicine name, health issues etc. via our mobile app. We are creating login facility for the user where he can add these items. The user will set the alarm for taking medicines through his app. On the server side, we are able to decrypt the message and based on the header message we are figuring out the time which the user has passed as a message. Once the alarm is set on the client side, the buzzer at our IOT Medicine Bootle will start beeping at the allocated time so that the patient will be aware about where exactly the medicine bottle is present in the house and also acts as a reminder to take medicines.

Here is how the form looks like on the application:



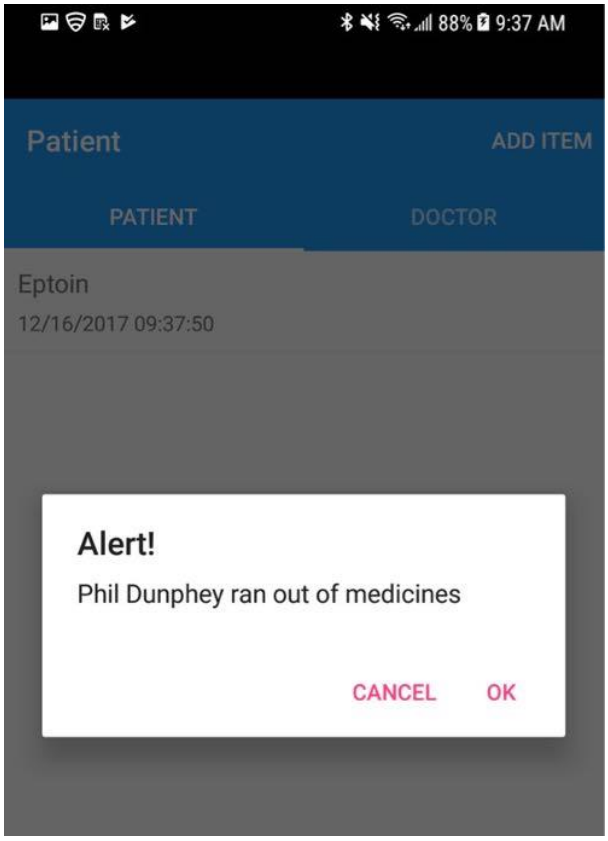
Once the buzzer is ON, the user has to press the button to indicate the system that he/she has taken the medicine. The buzzer will beep continuously which gives an advantage that the user can hardly forget to take the medicine. Once the button is pressed, we will ask the user to place his index finger on the **MAX 30105 sensor** for measuring the heart beat values. We have set the calibration time as 80 seconds so that we can read the values accurately. All we need is a rubber band that will pass through the mounting holes of the sensor. The user will have to hold the figure against the sensor. Based on the IR data, several instantaneous heart rates are calculated and the beats per minute will give you the average heart rate.

The **average heart** beats are continuously sent for 80 seconds to be stored on the cloud for the doctor to analyse. The **starting date** when the patient started to take medicine along with the **stop date** is passed to the cloud. Each time the use takes a medicine, the **medicine count** is decremented and even this is sent to the cloud for further analysis.

We are also using LCD for displaying the total number of medicines left inside the box for easier visibility to the patient. It can also be seen from his mobile app. Also a message which asks user to place **index finger** and when to **remove** finger from the sensor.

The user can now take medicine. If the medicines are getting empty, we will send a notification / alert message through the app to BOTH the patient and the doctor asking the user to refill the bottle or for the doctor to do further analysis.

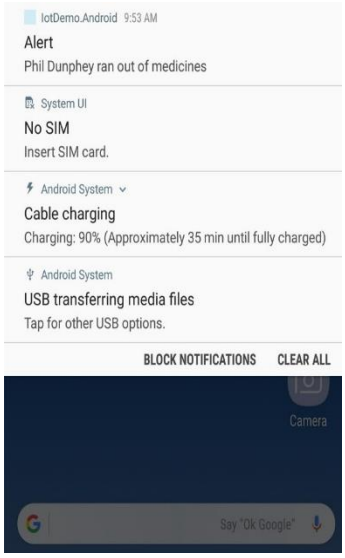
This is how alert for the patient is shown:



Doctor View:

On the other side, if the heart rate of the user falls below a particular value we send an alert message to the Doctor so that patient can be treated properly (**Implemented**). When the heartbeat of a patient drops continuously, the corresponding doctor of the patient can be alerted.

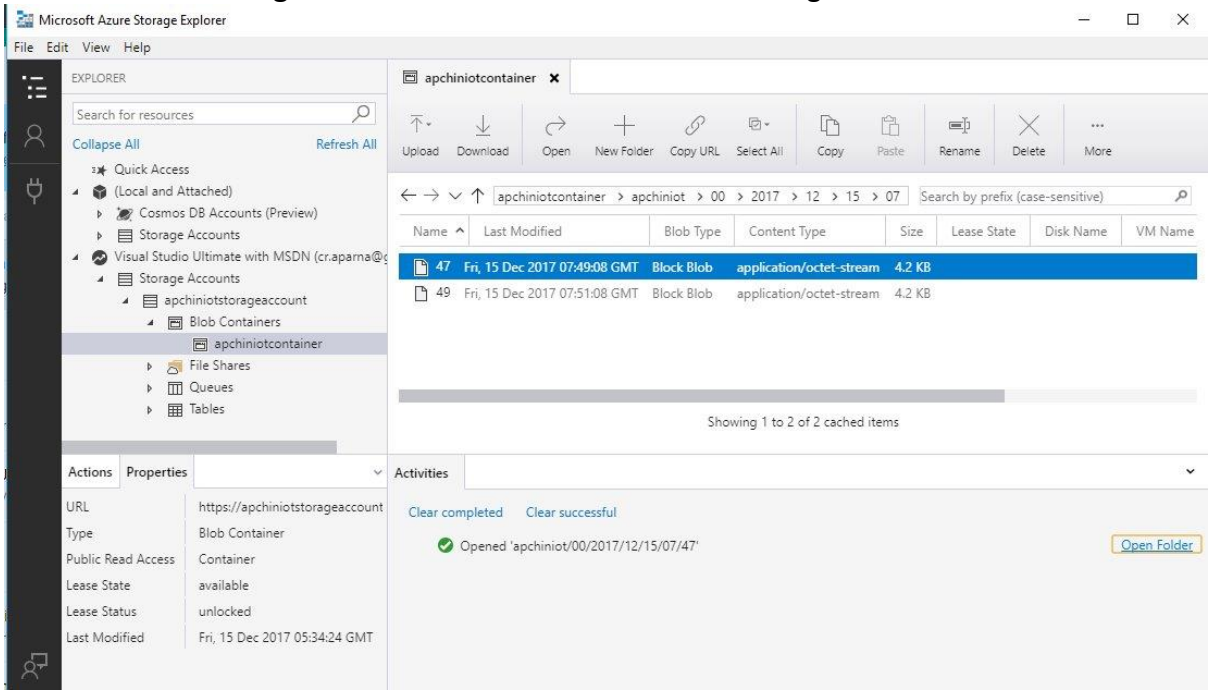
Alert message for the Doctor side:



The details related to the user are sent from the mobile application side and are received by the simulator app which serves as an intermediate between the client side and the server side.

We will send the IR values and related data like time delay now to the server. On the server side, we will plot a graph and do data analysis. We are storing the data like total medicine count, medicines taken till date etc. on the Azure Database so as to maintain historical data which can be used in future. Thus, we are having a mobile App, IoT Hub and Database on the Azure.

This is how the **storage** at Azure looks like. We use blob storage.



The App will reflect all the details which the doctor must know. Like the delays in taking medicine and the heart rate. To do analysis , we are storing all the **historical** data into the **Azure Database.** The graph we will show will plot data about the heart rate of user whenever he/she takes the medicine. Thus, the user can visualize the data as well.

Hardware components

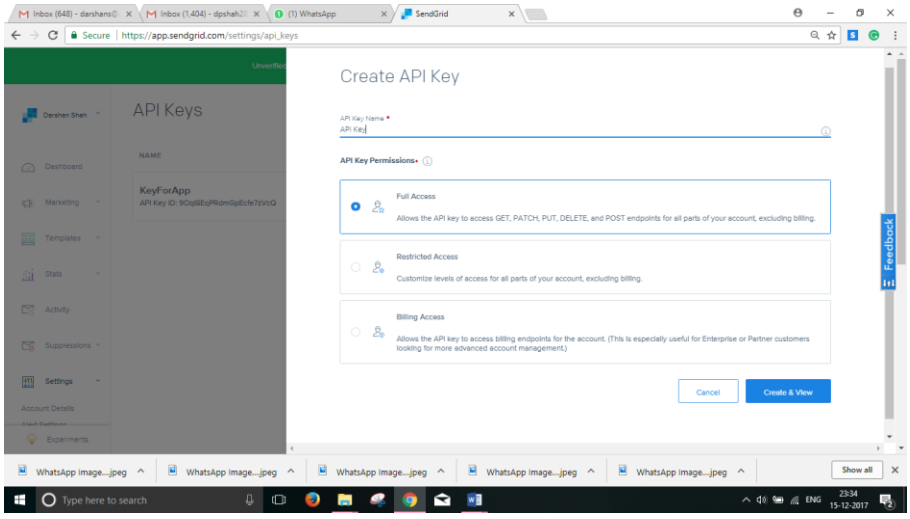
1. SparkFun ESP8266 module for WiFi connectivity
2. MAX 30105 particle sensor breakout for heart rate monitoring
3. Buzzer for our IOT Medicine bottle to start beeping when medicines shuld be taken.
4. Push button for the user to stop beeping of the alarm.

Software components:

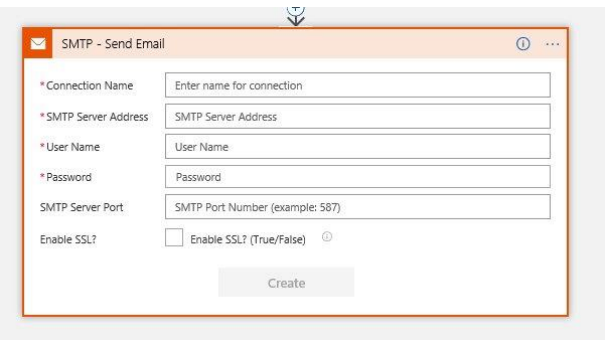
1. Cross platform **Mobile App** (runs on IOS, Android and Windows)
2. **IoT Hub**
3. **NotificationHub** on azure to send push notifications to the doctor.
4. **Blob Database** on Azure.
5. **SendGrid** server to send Emails to the doctor through **LogicApp** on Azure.

Now, we have provided additional facility of sending an email to the Doctor if the patient is in critical condition. For that we are integrating SMTP mail server of sendgrid with Azure.

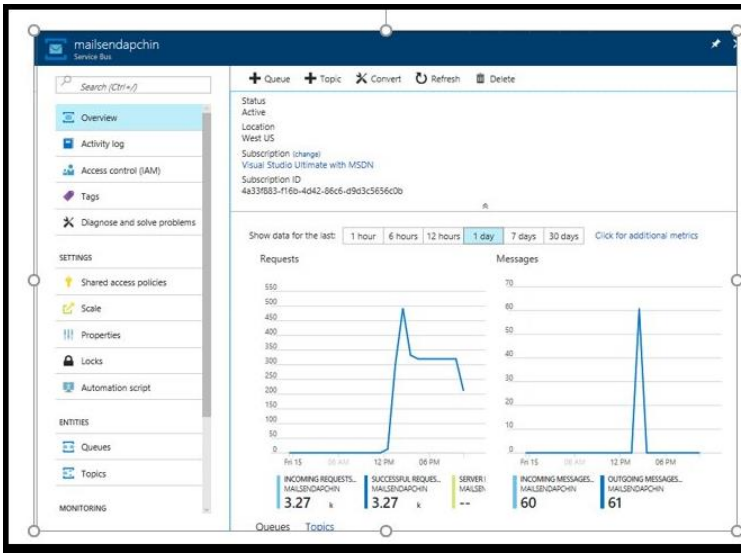
For creating secured connection, we have to generate API Key on the sendgrid site:

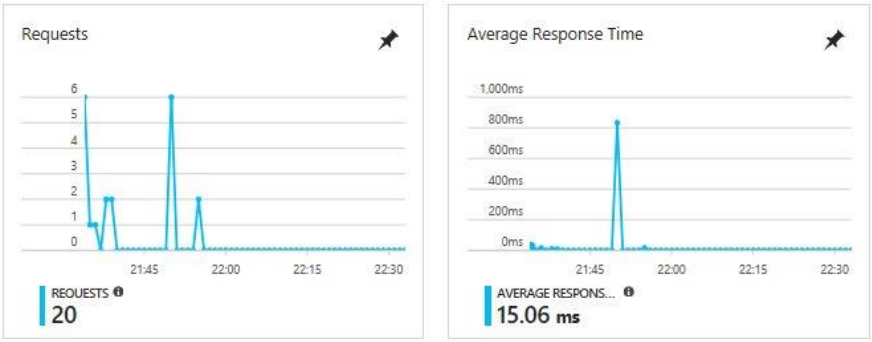


Once the API key is generated, we have to add the corresponding details on the Azure portal:



By entering the details related to SMTP server, we can set the email preferences and the message that has to be sent via the email. We can see the analysis about the emails sent on the Azure portal. After we have received enough data, this is how the web analytics is seen at Azure.





Analytics on the client side (on the mobile app for Doctor). Shows medicine count and heart beat.

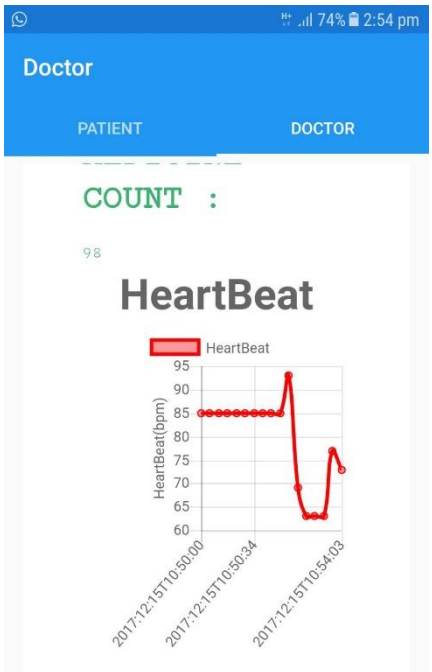
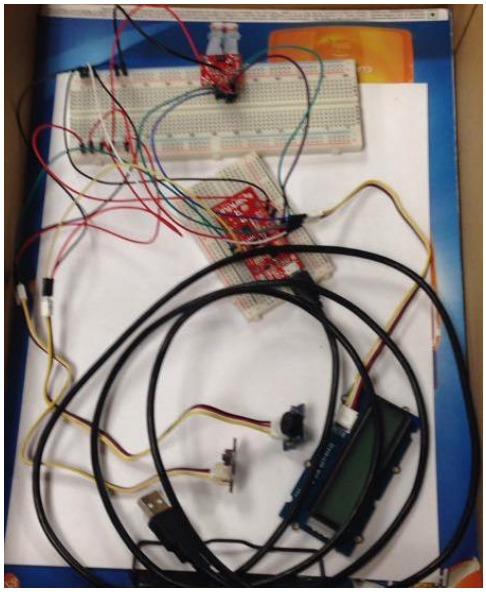


Figure showing the actual circuit and connections:



Design and Implementation Constraints: There were many constraints while building the system.

- The system won't work for a longer time on small cells. We will require batteries and we will have to incorporate the battery holder for that which can make the pill box bulky.
- On the other side, we were not able to figure out what will be the best way to attach this sparkfun sensor to the bottle as again this might increase the size of the bottle.
- The sparkfun ESP8266 requires the support of RTC- Real Time Clock module for the timer and alarm related functions.
- If different types of sensors related to project were readily available, it would have been easier for us to add more features.

Future scope: There can be many features that can be added to this medicine tracker system over time.

- We can use battery holders and provide batteries to make sure we have enough power to run the sensors.
- Our system considers one pill box per user. However, in the real world the same bottle can be reused for different users.
- One can even add some security features on the Web App and provide sessions as well.

In summary, this system makes human life easier by giving a timely reminder when a particular medicine has to be taken. We can analyse different factors like time delay in taking medicine and can even monitor heart rate. The software components make it easy for the user to deal with system and visualize data. This system can be widely used in the healthcare domain.