PERSONAL ASSISTANCE FOR SENIOR CITIZENS WHO ARE SELF-RELIANT

IBM NAALAIYA THIRAN (TEAM ID - PNT2022TMID35860)

PROJECT REPORT SUBMITTED BY

GURUBARN K (2019504524)

MEJALIN ARNO B (2019504037)

VINOTHAGAN J (2019504056)

ARUNKUMAR R (2019504008)

BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING

MADRAS INSTITUTE OF TECHNOLOGY CAMPUS

ANNA UNIVERSITY

CONTENTS

CHAPTER 1

INTRODUCTION

1.1) Project Overview

Sometimes elderly people forget to take their medicine at the correct time. They also forget which medicine He / She should take at that particular time. And it is difficult for doctors/caretakers to monitor the patients around the clock. To avoid this problem, this medicine reminder system is developed. A web app is built for the user (caretaker) which enables him to set the desired time and medicine using Node-red platform of IBM. These details will be stored in the IBM Cloudant DB. When the medicine time arrives the web application will send the medicine name to the IoT Device through the IBM IoT platform. This can be achieved by developing a python code. The device will receive the medicine name and notify the user with voice commands using TTS mechanism. This mechanism could be implemented using IBM Text to Speech service. An email is also sent to the care-taker, thus intimating the care-taker of the elderly person that the medicine intake time has arrived.

1.2) Purpose

Health is one of the biggest assets in our life. When elderly people falls sick, the only treatment which could be preferred is medicine therapy. Hence it becomes a job of paramount importance to take the doctor's prescribed medicines at the right time. But many elderly people might forget to take the medicines at the appropriate time. It is difficult for the care takers and medical professionals to look after them round the clock. Failing to take the medicines at the appropriate time can lead to deterioration of health of the senior citizens. The advent of technology has opened lot of ways to solve this problem. Hence using the technology of Internet of Things, it is possible to remind the elderly patients to take the medicine at the right time. The main purpose of the project is to enhance personal assistance to the senior citizens, improve the quality of their life and making them independent.

CHAPTER 2

LITERATURE SURVEY

2.1) Existing Problem

1)Health Alert and Medicine Remainder using Internet of Things

Methodology:

This paper proposes a model of automatic medicine reminder and apothecary system. It also continuously checks the people's health condition like Blood pressure, ECG through the tensors kept at home and inform them to take necessary action

Requirements:

1. LCD screen
2. buzzer
3. Arduino

2) Mobilizing your medications:

Methodology:

Use of mobile app for the medications were proposed where the mobile app is developed an automatic medication reminder is given to the patient's mobile phone.

Requirements:

Medication reminder software

3) Remote wireless health monitoring systems

Methodology:

The SMS framework is used for wellbeing checking because of its colossal application in many organizations and it is additionally remote. The information gathered through sensors are changed over to advanced shape and sent to the microcontroller for further preparing from it after handling the information is sent to a visual fundamental programming for graphical UI

Requirements:

1. Short Message Service (SMS)
2. Visual Basic data processing software
3. LM35 temperature sensor
4. Analog to Digital Converter (ADC)

4) Heart Attack Detection by Heartbeat Sensing using Internet of Things: *Methodology:*

The heart pulsates are seen using the sensors and alarms the client about the heart beat rate using the IoT gadgets. The principle preferred standpoint of this framework is it spares the life of people. The sensor is associated with the microcontroller where the microcontroller is changed to check the heart beat rate and send over internet.

Requirements:

1. Heart Beat sensor
2. Temperature sensor,
3. Pressure Sensor
4. Wifi Module
5. Atmega 328

5) A Health-IoT Platform :

Methodology:

The proposed system has various parts and supplies different services, e.g., Medication Reminder, Remote Prescriptions, Medication Noncompliance Control and Intelligent Analysis, and First Aid Alarm using RFID, WLAN, GPRS and 3G Technology.

Requirements:

1. Bio-medical sensor device
2. Inkjet-Printed Electrodes and Interconnections

2.2) References:

1) Health Alert and Medicine Remainder using Internet of Things, P. Ranjana, Elizabeth Alexander,2018

2) an automated medication reminder application for mobile phones and hypertension medication adherence in a high-risk urban population, Samir Patel 1, Laura Jacobus-Kantor, Lorraine Marshall, Clark Ritchie, Michelle Kaplinski, Parvinder S Khurana,Richard J Katz,2013.

3) Remote wireless health monitoring systems B. Priya, S. Rajendran, R. Bala and Gobbi. R, July 2009.

4) Heart Attack Detection by Heartbeat Sensing using Internet of Things: IoT, Dr.A.A.Gurjar, Neha A. Sarnaik, 2018.

5) A Health-IoT Platform Based on the Integration of Intelligent Packaging, Unobtrusive Bio- Sensor and Intelligent Medicine Box, Geng Yang,Li Xie,Matti Mäntysalo,2014.

2.3) Problem Statement:

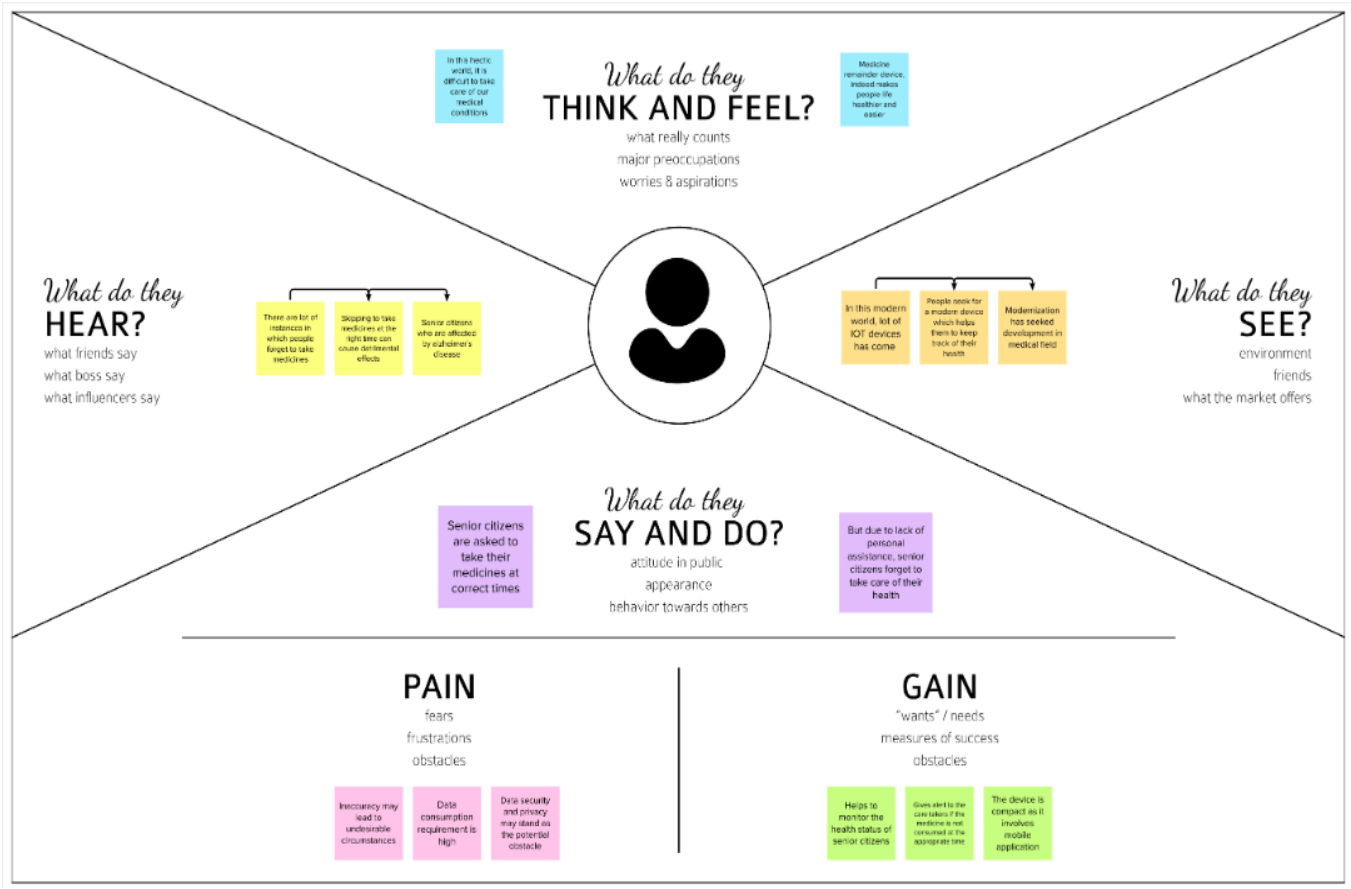
Good health has been a major concern since the start of humankind whilst for some people reaching good health requires taking prescribed medicines or pills routinely. However, many patients find it difficult to keep track of taking their medication in the right time and proportion. This happens especially if it involves taking pills or medication on daily basis due to several reasons such as heavy work load, forgetfulness, old age and alterations in day-today behavior can have a significant result on whether patients recall to take their prescribed medications which can be termed as medicine adherence, which is a serious problem because it may affect the total well-being of the patient ,medical cost of the patient and can be a matter of life and death .The medicine reminder application could impact positively on the life of the patient as it will help the patient by remembering the intake of these prescribed medications could be a matter of life and death.

Who does the problem affect?	It will affect the senior citizen.
What are the boundaries of the problem?	Senior citizen who cannot take care of themselves.
What is the issue?	Senior citizen with the assistant of the family members can take care by their family, but who are self-reliant should make themselves of their own with every problem.
When does the issue Occur?	Older people who are self-reliant are suffered by this problem.
Where does the issue Occur?	When the senior citizen's live self-dependently.
Why is it important that we fix the problem?	Taking health care as the primary aim, senior citizen who are not in ease should take the medicine at the right time.
What solution to solve this issue?	By creating a personal assistant that reminds the elder people to consume their tablet on time.
What method used to solve the issue?	Creating IOT based system to remind the elder people with their medicines.

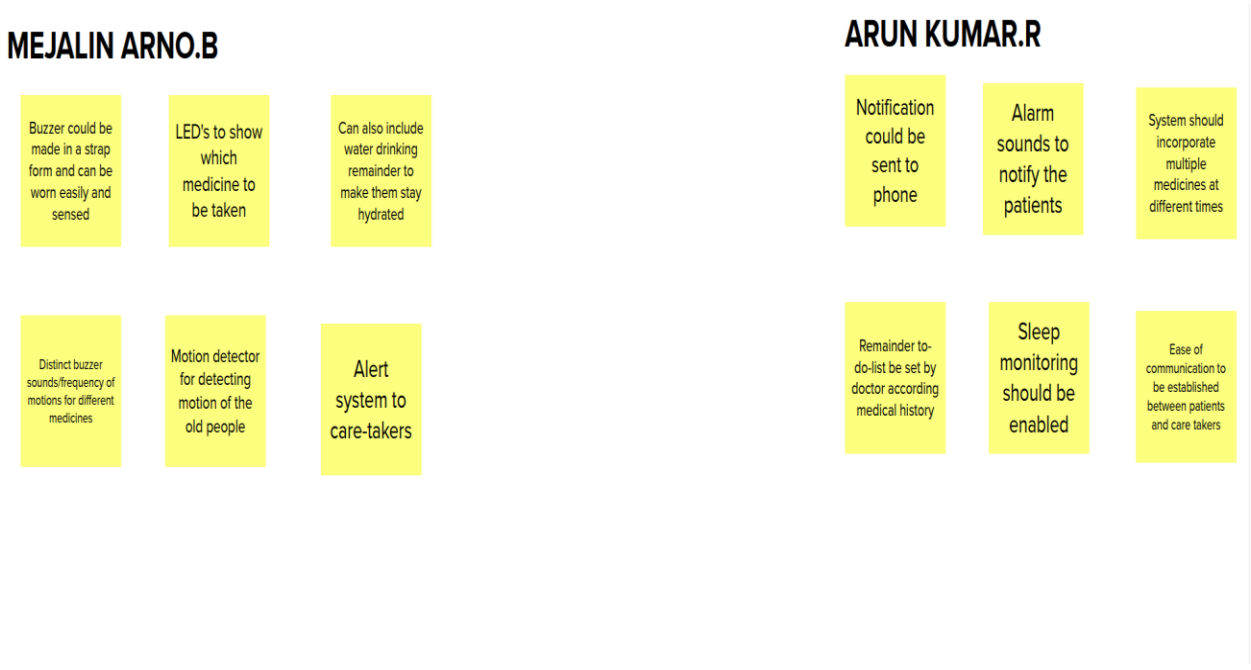
CHAPTER 3

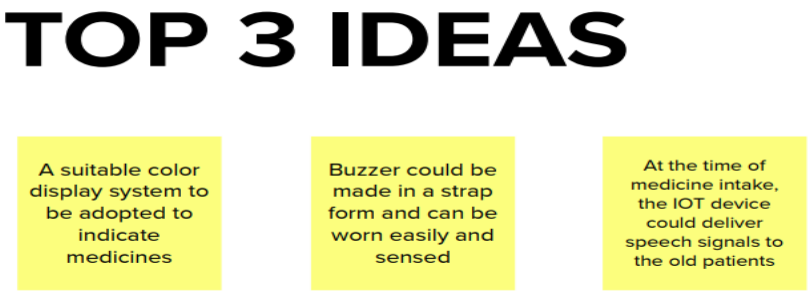
IDEATION AND PROPOSED SOLUTION

3.1) Empathy Map Canvas:



3.2) Ideation and Brainstorming:



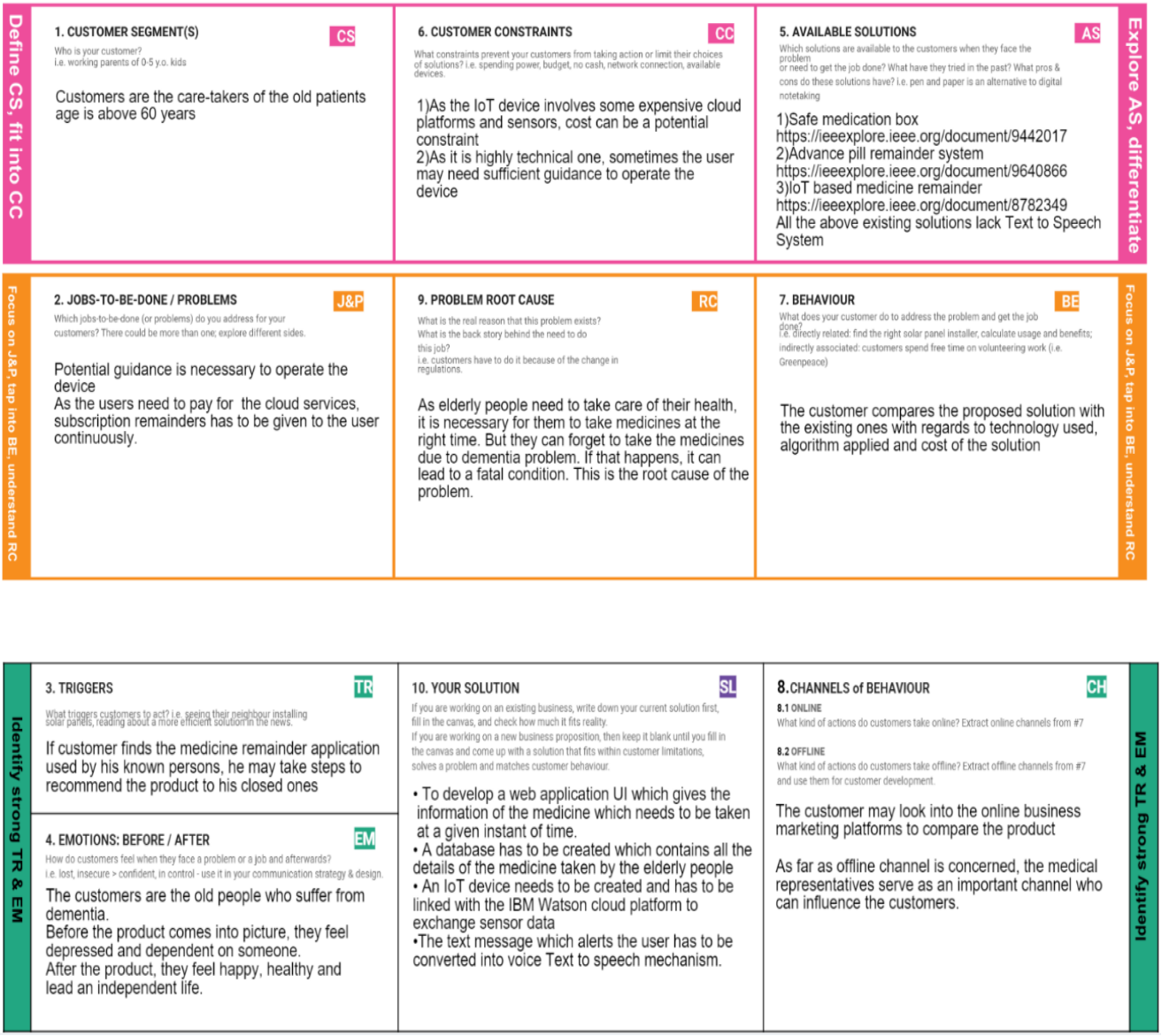


3.3) Proposed Solution:

S.NO	PARAMETER	DESCRIPTION
1)	Problem Statement (Problem to be solved)	Human life span is approximately 70 years. This is because of the increase in medical technology. Hence it is necessary to look after the elderly people as they need to take necessary medicines at the right time. But it is difficult for the doctors and care takers to look after them round the clock. So, it is extremely important to develop an IoT device which gives rest of the medicine to be taken and at the right time.
2)	Idea / Solution description	• To develop a web application UI which gives the information of the medicine which needs to be taken at a given instant of time. • A database must be created which holds all the details of the medicine taken by the elderly people. • An IoT device must be created and linked with the IBM Watson cloud platform to exchange sensor data. • The text message alerting the user must be converted into voice Text to speech mechanism.
3)	Novelty / Uniqueness	The unique aspect of our project is the TTS mechanism (which is Text to Speech mechanism) which serves as a distinct feature when compared with the existing solutions
4)	Social Impact / Customer Satisfaction	The elderly people can lead a healthy and independent life as they do not need to seek the help of a care taker/medical personnel to remind their medical specifications.
5)	Business Model (Revenue Model)	The developed IoT device could be marketed based on subscription service. As the model involves cloud and database service which is

		not free of cost, it can be only be sold based on subscription
6)	Scalability of the Solution	The proposed solution has high success rate and good in terms of practical implementation as we are going to inculcate standard technical services like IBM Watson cloud services, Node Red service for web development. Moreover, the model is highly user-friendly.

3.4) Problem Solution Fit



CHAPTER 4

REQUIREMENT ANALYSIS

4.1) Functional Requirement

FR.NO	Functional Requirement (Epic)	Sub-Requirement(story/subtask)
FR.1)	User Registration	Registration through our web application.
FR.2)	User Confirmation	Confirmation within the web application.
FR.3)	User Input Medication Data	Data should be fed to the dashboard text fields in the application.
FR.4)	Acknowledgement	Data will be saved in the application and acknowledgement will be given to the user
FR.5)	Internet Connectivity	User should have a stable internet connection to access the functionality of our project via web application.
FR.6)	Actuators	Speakers must notify the users.

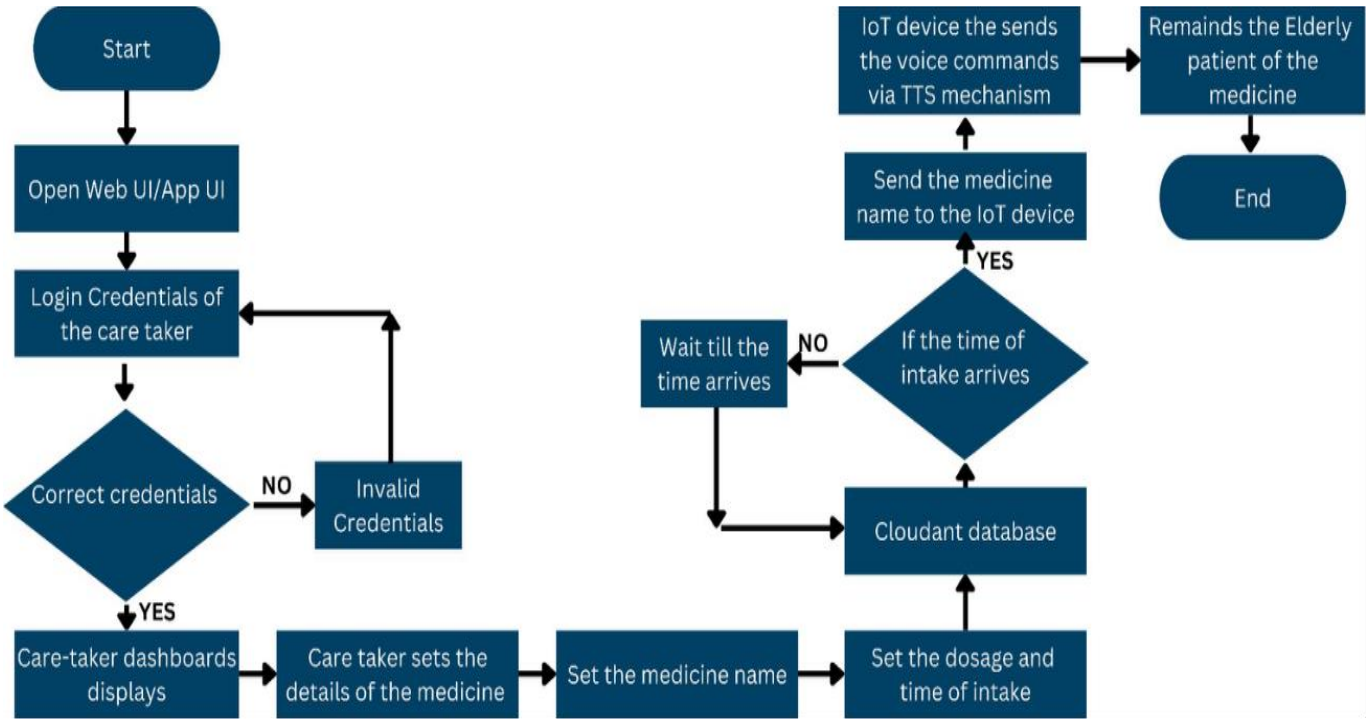
4.2) Non – Functional Requirement:

NFR.NO	Non-Functional Requirement	Description
NFR.1)	Usability	The whole system can be accessed through web application. Hence it is very easy to use.
NFR.2)	Security	The data will be stored in the cloud so the user's data is secured.
NFR.3)	Reliability	As the data is stored in cloud, the data cannot be manipulated externally so it is highly reliable.
NFR.4)	Performance	As virtual sensors are used for sensing operation sits values are quite accurate. Hence performance would be good.
NFR.5)	Availability	The Cloud server is active all the time the user can avail it anytime.
NFR.6)	Scalability	The application can be used in any kind of operating system either in small or large OS so the scalability is high.

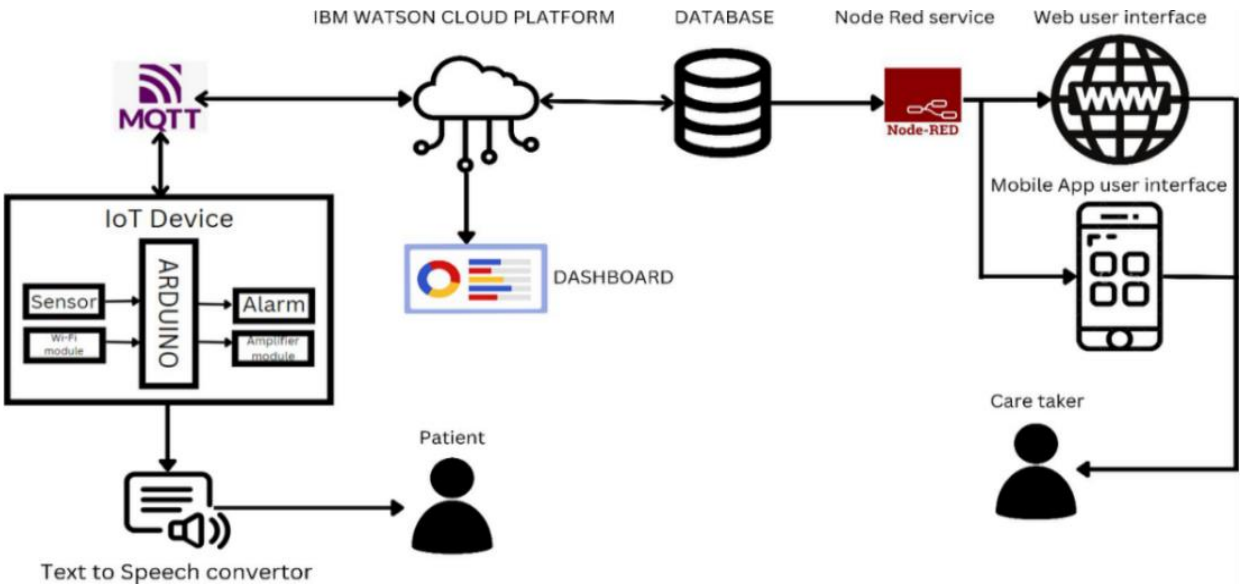
CHAPTER 5

PROJECT DESIGN

5.1) Data Flow Diagrams:



5.2) Solution and Technical Architecture



Components and technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	MIT app inventor
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson TTS service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Cloud , Node Red
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM Cloudant DB.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	DailyMed API
9.	External API-2	Purpose of External API used in the application	GoodRx API
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration:	IBM HTTP web server IaaS (Infrastructure as a Service)

Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	MIT app inventor
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Obfuscation block implementation level in MIT app inventor
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	IBM Watson, TTS service
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	As the product needs to remind the medicine via IoT, we need Cloud services which is available
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Microcontroller (arduino) controls the entire process and repeats the procedure on a daily basis

5.3) User Stories:

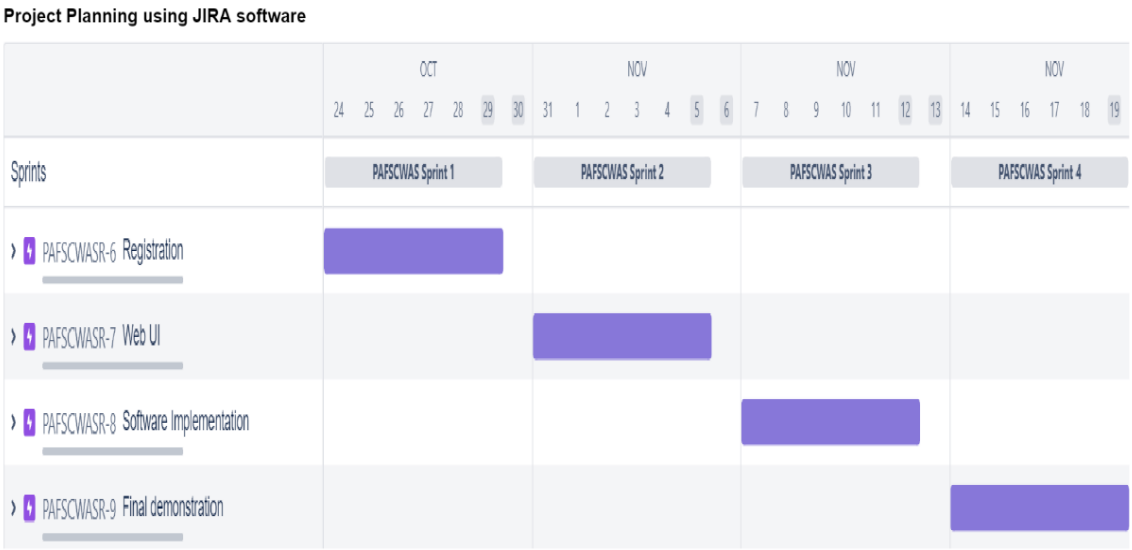
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can register & access the dashboard	Low	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can login to the application	Medium	Sprint-1
Customer (Web user)	Dashboard	USN-5	As a caretaker, I can update the medicine details	I can update the details of the medicine	High	Sprint-2

		USN-6	As a caretaker, I can set the name of the medicine, its dosage and its time of intake	I can set the name of the medicine, its dosage and its time of intake	High	Sprint-3
Customer Care Executive	The one who provides cloud access	USN-7	As a CCE, I should provide the best services to my customer upon subscription	I will get the subscription charges from the customer to provide some technical requirements	Medium	Sprint-3
Administrator	The one who moderates/ governs the entire system	USN-8	As an administrator, I shall look for any new updates in the system and regulate the work flow	I will ensure the product working condition and its efficiency	Low	Sprint-4

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1) Sprint planning and estimation



Jira is a proprietary issue tracking product developed by Atlassian that enables issue tracking and agile project management. Jira comes in his four packages. Jira Work Management is intended for general project management. Jira Software includes base software that includes agile project management functionality (previously a separate product: Jira Agile). Jira Service Management is intended for use by IT operations or business service desks. Jira Align is intended for strategic product and portfolio management. Jira is written in Java and uses the Control Container Picoine version, the Apache OFBiz Entity Engine and the WebWork 1 technology stack. For remote procedure calls (RPC), Jira has REST, SOAP and XML-RPC interfaces. Jira integrates with source control programs such as Clearcase, Concurrent Versions System (CVS), Git, Mercurial, Perforce, Subversion and Team Foundation Server. Jira implements a Networked Help Desk API for sharing customer support tickets with other issue trackers.

6.2) Sprint delivery schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration. Creation of IBM services like NodeRED, Cloudant DB, TTS Service and design of IoT system	USN-1	As a user, I must be able to login to the IBM platform	2	High	Gurubaran, Arunkumar
Sprint-2	Web UI. Creation of Web UI using NodeRED service	USN-2	As a user, I must be able to update the medicine details in the web UI	2	High	Vinothagan, Mejalin Arno
Sprint-3	Software implementation. Developing Python code to retrieve data from cloudant db to send that data to IoT device	USN-3	As a user, I must be push the details to the IoT device	2	High	Gurubaran, Mejalin Arno
Sprint-4	Final demonstration and user testing. Generating voice commands using IBM Text to Speech service	USN-4	As a user, I must be able hear the medicine name which is to be taken at the appropriate time and check its accuracy	2	High	Vinothagan, Arunkumar

The project is planned in four phases. In phase 1, the user must be able to login into the IBM platform. In the following phase the user must be able to update the medicine details in the web user interface. In the next phase we enable the user to push the details into the IoT device. Finally, the user must be able to hear the medicine name at appropriate time.

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022		
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

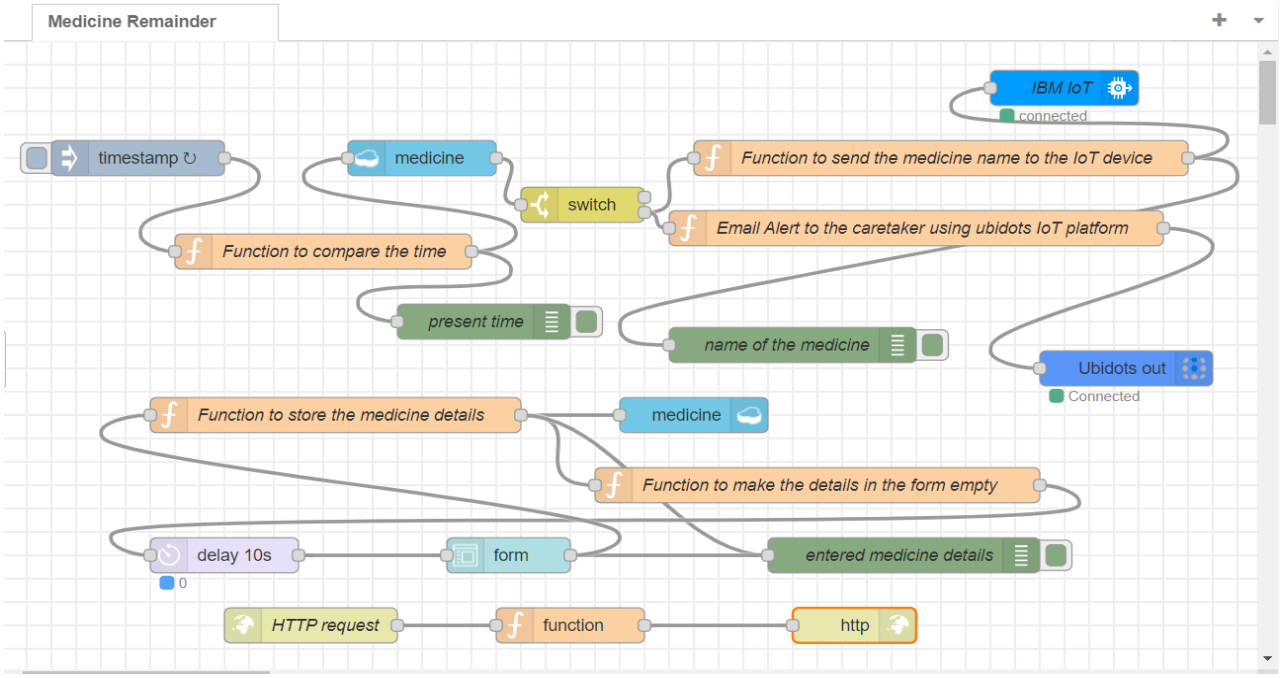
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

CHAPTER 7

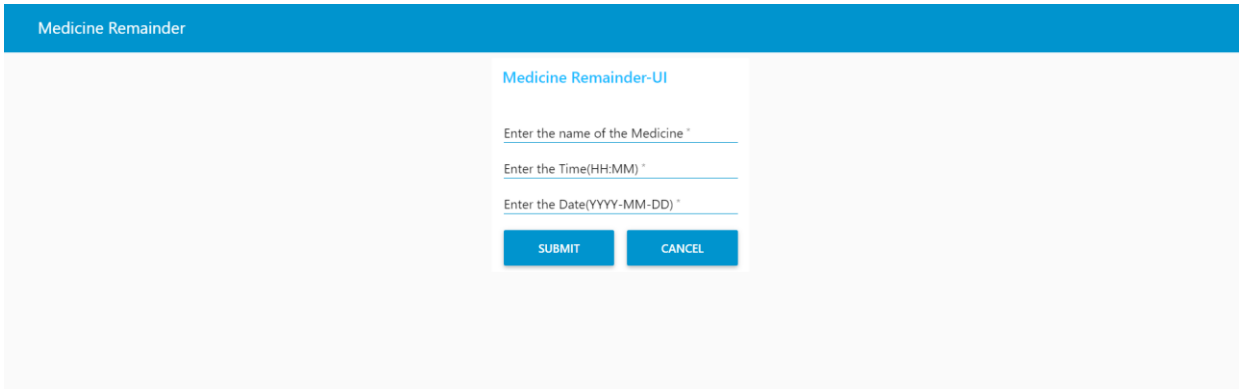
CODING & SOLUTIONING

7.1) Feature 1

The Web UI is created using Node-Red.



The form is created



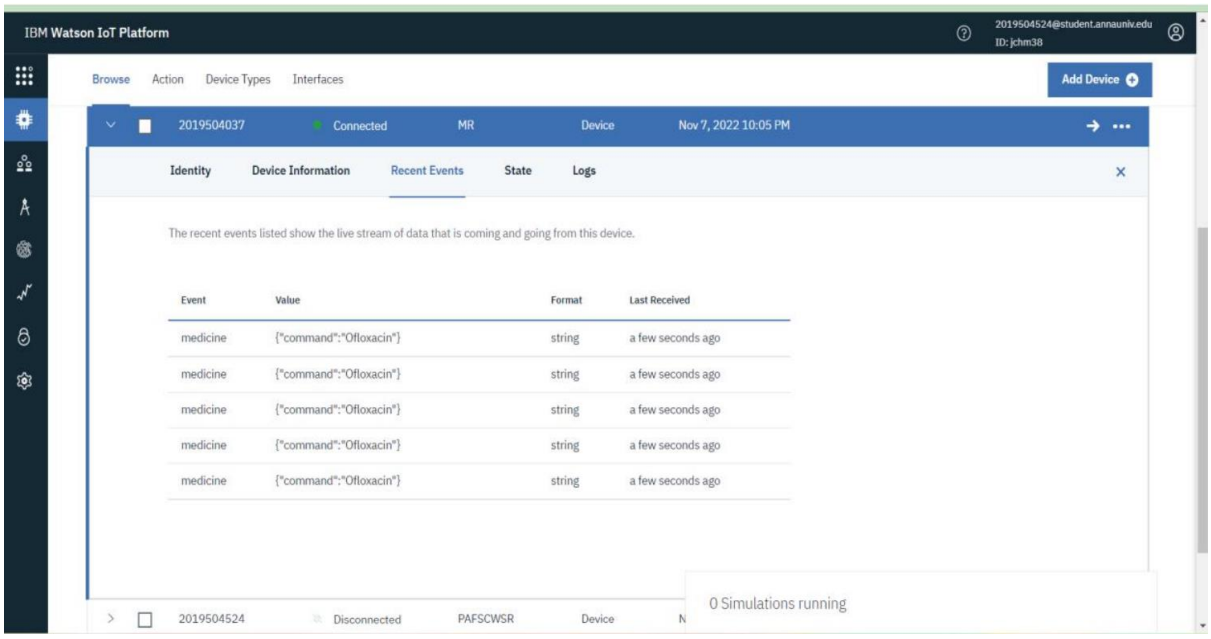
7.2) Feature 2

```
import requests
import paho.mqtt.client as mqtt
import json
prevMedicine = ''
currMedicine = ''
while True:
    req=requests.get("http://169.51.206.114:32641/remainder")
    value=req.json()
    print(value['command'])

    try:
        prevMedicine = value['command']
        ORG= "jchm38"
        DEVICE_TYPE ="MR"
        DEVICE_ID ="2019504037"
        TOKEN ="()!xRUci*BCpeso-rk"
        server = ORG + ".messaging.internetofthings.ibmcloud.com";
        pubTopic1 = "iot-2/evt/medicine/fmt/string"
        pubTopic2 = "iot-2/evt/pH/fmt/json"
        pubTopic3 = "iot-2/evt/turb/fmt/json"
        #pubTopic3 = "iot-2/evt/wf/fmt/json"
        authMethod = "use-token-auth";
        token = TOKEN;
        clientId = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID
          if currMedicine != prevMedicine:
              mqttc = mqtt.Client(client_id=clientId)
              mqttc.username_pw_set(authMethod, token)
              mqttc.connect(server, 1883, 60)
              mqttc.publish(pubTopic1,json.dumps(value))
              print("Published Successfully!")
              currMedicine = prevMedicine
    except Exception as error:
        print(error.args[0])
        print("Error!")
mqttc.loop_forever()
```

The purpose of feature is to send data from nodered software to the IBM Watson platform. Necessary libraries are imported. The Requests module allows you to send HTTP requests using Python. The IoT protocol deployed is MQTT which is imported from the library paho.mqtt.client

An HTTP request returns a Response object containing all response data (content, encoding, status, etc.). Because of the dynamic nature of the program, the current medicine should be updated as the specified time arrives. JSON is a syntax for storing and exchanging data. JSON is text, written with JavaScript object notation. The name of the medicine can be obtained from the website. If in any case, exception arrives it is tackled my using try except statements in python.

The medicine details are sent to the IBM IoT Watson platform



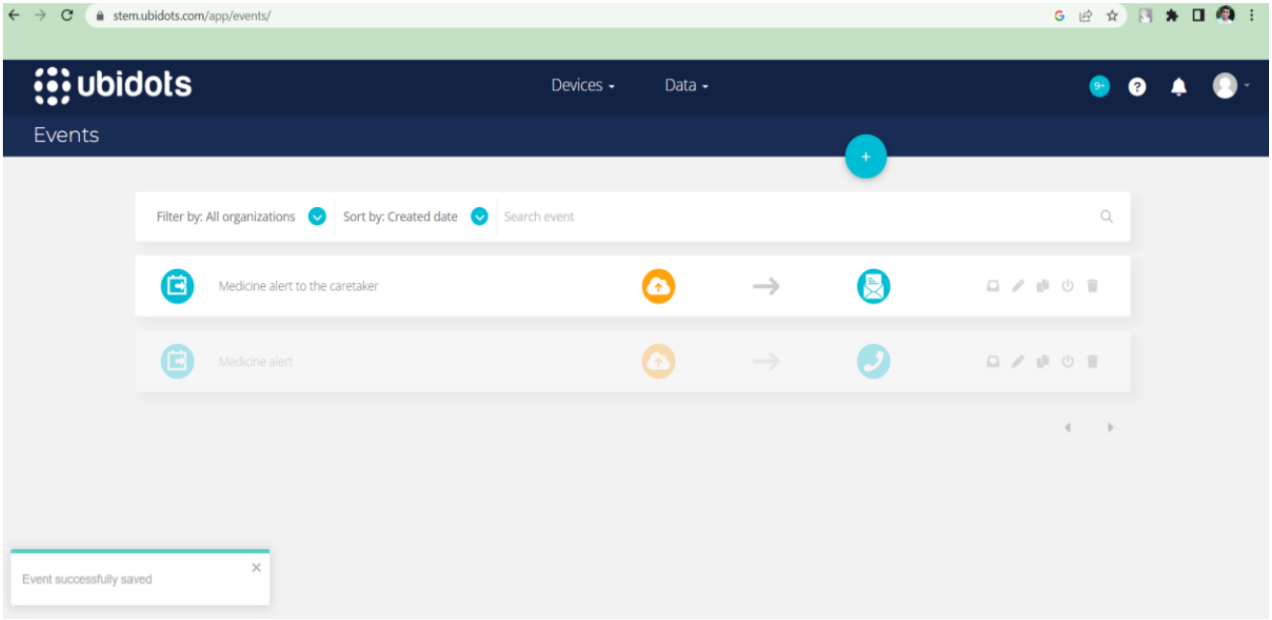
7.3) Feature 3

```
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from playsound import playsound

authenticator =
IAMAuthenticator('jHG72RxBZzpJDs4vkDt6DySXoaJu9hylmn0hjE_p-F0g')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)

text_to_speech.set_service_url('https://api.au-syd.text-to-
speech.watson.cloud.ibm.com/instances/74dc1eed-1e64-4f57-ba4a-
2031a8f39d85')
with open('med.mp3', 'wb') as audio_file:
    audio_file.write(text_to_speech.synthesize('Take Crocin 50 mg Now',
voice='en-US_AllisonV3Voice', accept='audio/wav').get_result().content)

print("playing")
playsound('med.mp3')
```

The second part is the implementation of Text to Speech Mechanism. Necessary libraries are imported and the text to speech conversion is made using IBM text to speech service. Relevant sound files are used to achieve this task.

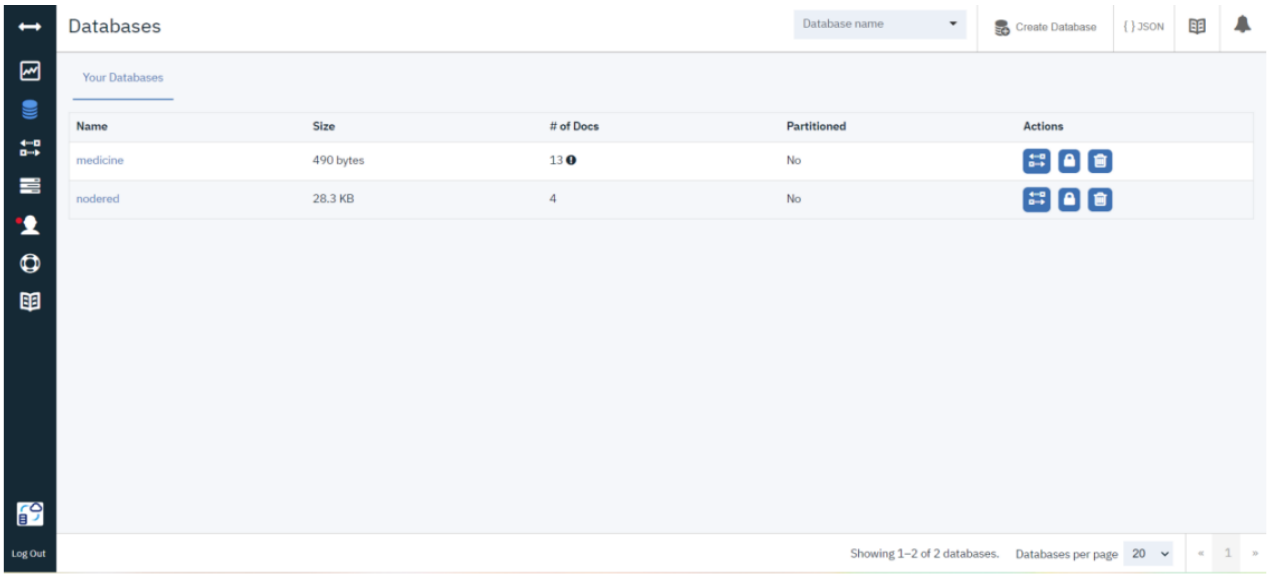
7.4) Feature 4

Ubidots platform is used to send e-mail



7.5) Database Schema

The entered medicine details are stored in the cloudant db database



CHAPTER 8

TESTING

8.1) Test cases

Test scenarios

1)Whether Web UI dashboard link is functional?

2)Whether the details of the medicine reach the database?

3)Whether the name of the medicine is displayed in the IBM IoT Watson platform?

4)Is the python code publishing the medicine name to the IoT platform?

5)Are the voice commands generated correctly?

6)Is an email sent to the care taker at the time of the medicine intake?

7)Whether ubidots platform functions properly?

8.2) User Acceptance Testing

a) Defect analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	2	4	2	4	12
Duplicate	0	0	0	1	1
External	0	2	1	2	5
Fixed	4	8	0	6	17
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	0	0
Won't Fix	0	2	1	1	4
Totals	8	17	5	12	42

b) Test case analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Web UI dashboard	3	0	0	3
Database	3	0	0	3
IoT platform	3	0	0	3
Node red	3	0	0	3
E-mail service	3	0	0	3
Text to Speech mechanism	3	0	0	3
Final Report Output	3	0	0	3

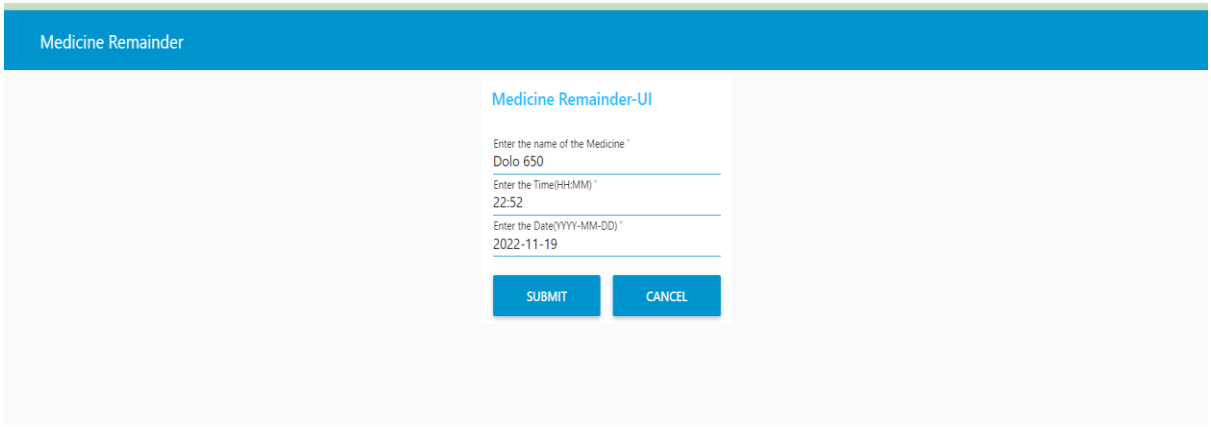
CHAPTER 9

RESULTS

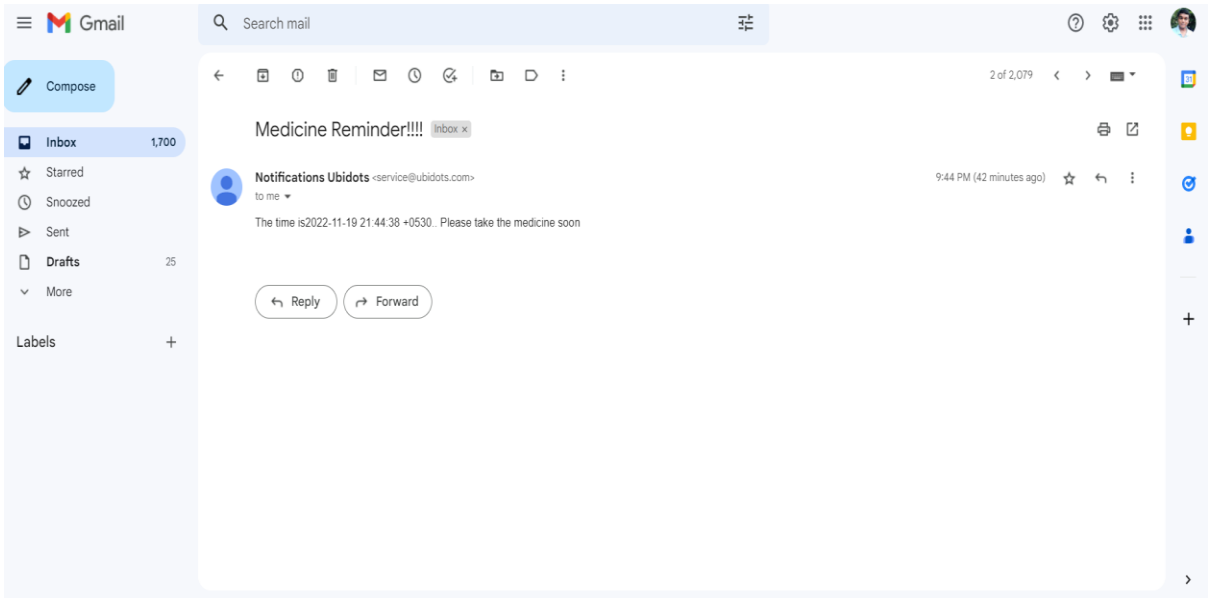
9.1) Performance metrics

The web UI has been created successfully with the help of node-red service. A user-friendly dashboard has been successfully deployed to facilitate the care taker to enter the medicine details. When the medicine time arrives, the name of the medicine spelled out to the user and an e-mail is sent to the care-taker successfully, thereby meeting the functional and non-functional requirements of the project successfully.

9.1.1) Web UI-Dashboard



9.1.2) E-Mail sent to care taker at the time of medicine intake



9.1.3) Voice commands are generated and name of the medicine is speeled out to the end user. (A google drive link is attached to hear the voice commands)

https://drive.google.com/file/d/190BT9642_KT801JEqu5nEtIwwkkUql1Y/view?usp=sharing

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

10.1) Advantages

- The elderly patient is reminded to take the medicine at the right time thus ensuring a good health.
- Care-taker is also intimated of the medicine time at which the elderly patient has to take the medicine via e-mail
- The voice commands direct the user to take the right medicine at the appropriate time, thus ensuring high degree of accuracy

10.2) Disadvantages

- This system is ineffective for the senior citizens who have hearing impairments.
- It requires continuous usage of internet and data. Any fluctuations in the data rate many cause inaccuracies
- Sometimes medicine names are difficult to spell. Thus, reception of incorrectly spelled medicine can cause confusion.

CHAPTER 11

CONCLUSION

This medicine remainder system gives a confidence for the senior citizens who are self-reliant to lead an independent life without the help of other individuals. The voice commands help the senior citizens to pick up the correct medicine at the right time. This system also helps the care-takers of the senior citizens to keep a track of their medicine status and make them diligent. It improves the quality of life and in a way, it also helps to increase the human life-span. This stands as an epitome of how the IoT technology is making human lives easier and healthier.

CHAPTER 12

FUTURE SCOPE

The medicine reminder developed in this project facilitates the process of only reminding the senior citizens to take the medicine at the appropriate time. In future, this reminder system can be updated to monitoring system which facilitates continuous monitoring of patient's vital parameters like blood pressure, pulse rate and blood oxygen level. As medical technology is heading towards telemedicine now a days, the care-taker access could be extended to health care professionals. On continuous monitoring, if any fluctuations in vital parameters of the patient prevails, a notification of emergency could be delivered to the care-taker, health care professionals, ambulance driver and nearby hospital of the patient's residence. As medicine field is seeing lot of advancements, the advent of Internet of Things opens new regime to improve the quality of life in near future.

CHAPTER 13

APPENDIX

13.1) Node-Red: Functions

Function to store medicine details to database

```
var d=msg.payload.date
var t=msg.payload.time
msg.payload ={
   "_id":d+" "+t,
   "name":msg.payload.name,
}
return msg;
```

Function to make the details in the form empty

```
msg.payload={
   "date":"",
   "name":"",
   "time":""
}
return msg;
```

Function to compare time

```
var d=new Date();
var utc= d.getTime()+ (d.getTimezoneOffset()*60000);
var offset= 5.5;
newDate= new Date(utc + (3600000*offset));
var n=newDate.toISOString()
var date = n.slice(0, 10)
var time = n.slice(11,16)
global.set('time', time)
msg.payload=date+" "+time
return msg
```

Function to send the name of the medicine name to the IoT platform

```
msg.payload={"command":msg.payload.name}
global.set("mr",msg.payload)
return msg;
```

Function HTTP request-response

```
msg.payload=global.get("mr")
return msg;
```

13.2) Python Source code:

```
import requests
import paho.mqtt.client as mqtt
import json
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from playsound import playsound
prevMedicine = ''
currMedicine = ''
while True:
    req=requests.get("http://169.51.206.114:32641/remainder")
    value=req.json()
    try:
        prevMedicine = value['command']
        ORG= "jchm38"
        DEVICE_TYPE ="MR"
        DEVICE_ID ="2019504037"
        TOKEN ="()!xRUci*BCpeso-rk"
        server = ORG + ".messaging.internetofthings.ibmcloud.com";
        pubTopic1 = "iot-2/evt/medicine/fmt/string"
        pubTopic2 = "iot-2/evt/pH/fmt/json"
        pubTopic3 = "iot-2/evt/turb/fmt/json"
        #pubTopic3 = "iot-2/evt/wf/fmt/json"

        authMethod = "use-token-auth";
        token = TOKEN;
        clientId = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;
        if currMedicine != prevMedicine:
                mqttc = mqtt.Client(client_id=clientId)
                mqttc.username_pw_set(authMethod, token)
                mqttc.connect(server, 1883, 60)
                mqttc.publish(pubTopic1,json.dumps(value))
                print("Published Successfully!")
                authenticator =
IAMAuthenticator('jHG72RxBZzpJDs4vkDt6DySXoaJu9hylmn0hjE_p-F0g')
                text_to_speech = TextToSpeechV1(
                    authenticator=authenticator
                )

                text_to_speech.set_service_url('https://api.au-syd.text-to-
speech.watson.cloud.ibm.com/instances/74dc1eed-1e64-4f57-ba4a-
2031a8f39d85')
                with open('try.mp3', 'wb') as audio_file:
```

```
                audio_file.write(text_to_speech.synthesize('Please Take'+'
'+value['command']+'   '+'tablet now', voice='en-US_MichaelExpressive',
accept='audio/wav').get_result().content)

            print("playing")
            #playsound('try.mp3')

            currMedicine = prevMedicine
    except Exception as error:
        print(error.args[0])
        print("Error!")
mqttc.loop_forever()
```

13.3) Node flows:


```
[{"id":"75978a6437b1bbee","type":"tab","label":"Medicine
Remainder","disabled":false,"info":"","env":[]},{"id":"355d6edfb21a37ac","typ
e":"ibmiot","name":"IBM  IOT API
KEY","keepalive":"60","serverName":"jchm38.messaging.internetofthings.ibm
cloud.com","cleansession":true,"appId":"","shared":false},{"id":"ea854c1fd001
2e7c","type":"cloudant","host":"https://apikey-v2-
g2mgoaa13ktty45ajlxsbsu036aet74dzzcaijky3xd:6046fe3b196e02c18463596e2f
6d101e@6b668acd-11ae-4491-9d1f-31de9ab54fa1-
bluemix.cloudantnosqldb.appdomain.cloud","name":""},{"id":"f3161ec9f28526
21","type":"ui_group","name":"Medicine Remainder-
UI","tab":"ccb424f206929e3f","order":2,"disp":true,"width":"6","collapse":fals
e,"className":""},{"id":"0140cf3398d7f6bb","type":"ui_group","name":"Medi
cine
Remainder","tab":"88ea0e5bc209b315","order":1,"disp":true,"width":"6","colla
pse":false,"className":""},{"id":"ccb424f206929e3f","type":"ui_tab","name":"
Medicine
Remainder","icon":"dashboard","disabled":false,"hidden":false},{"id":"37155f3
f1ee0a80e","type":"ui_base","theme":{"name":"theme-
light","lightTheme":{"default":"#0094CE","baseColor":"#0094CE","baseFont":
"-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
Sans,Ubuntu,Cantarell,Helvetica Neue,sans-
serif","edited":true,"reset":false},"darkTheme":{"default":"#097479","baseColo
r":"#097479","baseFont":"-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-
serif","edited":false},"customTheme":{"name":"Untitled Theme
1","default":"#4B7930","baseColor":"#4B7930","baseFont":"-apple-
system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"},"themeState":{"base-
```

color":{"default":"#0094CE","value":"#0094CE","edited":false},"page-titlebar-backgroundColor":{"value":"#0094CE","edited":false},"page-backgroundColor":{"value":"#fafafa","edited":false},"page-sidebar-backgroundColor":{"value":"#ffffff","edited":false},"group-textColor":{"value":"#1bbfff","edited":false},"group-borderColor":{"value":"#ffffff","edited":false},"group-backgroundColor":{"value":"#ffffff","edited":false},"widget-textColor":{"value":"#111111","edited":false},"widget-backgroundColor":{"value":"#0094ce","edited":false},"widget-borderColor":{"value":"#ffffff","edited":false},"base-font":{"value":"-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"}},"angularTheme":{"primary":"indigo","accents":"blue","warn":"red","background":"grey","palette":"light"}},"site":{"name":"Node-RED Dashboard","hideToolbar":"false","allowSwipe":"false","lockMenu":"false","allowTempTheme":"true","dateFormat":"DD/MM/YYYY","sizes":{"sx":48,"sy":48,"gx":6,"gy":6,"cx":6,"cy":6,"px":0,"py":0}}},{"id":"88ea0e5bc209b315","type":"ui_tab","name":"Speech","icon":"dashboard","disabled":false,"hidden":false},{"id":"b2402918def42420","type":"ibmiot out","z":"75978a6437b1bbee","authentication":"apiKey","apiKey":"355d6edfb21a37ac","outputType":"cmd","deviceId":"2019504037","deviceType":"MR","eventCommandType":"name","format":"json","data":"medicine","qos":0,"name":"IBM IoT","service":"registered","x":860,"y":40,"wires":[]},{"id":"010bbce5a3cd2909","type":"inject","z":"75978a6437b1bbee","name":"","props":[{"p":"payload"},{"p":"topic","vt":"str"}],"repeat":"60","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":110,"y":100,"wires":[["6564307ed1adbb56"]]},{"id":"7b271549d44feb08","type":"cloudant in","z":"75978a6437b1bbee","name":"","cloudant":"ea854c1fd0012e7c","database":"medicine","service":"_ext_","search":"_id_","design":"medicine","index":"id","x":340,"y":100,"wires":[["4eedb76048312a39"]]},{"id":"5a8fbd45311c12bb","type":"function","z":"75978a6437b1bbee","name":"Function to send the medicine name to the IoT device","func":"msg.payload={\"command\":msg.payload.name}\nglobal.set(\"mr\",msg.payload)\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":760,"y":100,"wires":[["e22a5ccd54ac0b2f","b2402918def42420"]]},{"id":"6564307ed1adbb56","type":"function","z":"75978a6437b1bbee","name":"Function to compare the time","func":"var d=new Date();\nvar utc= d.getTime()+(d.getTimezoneOffset()*60000);\nvar offset= 5.5;\nnewDate= new Date(utc + (3600000*offset));\nvar n=newDate.toISOString()\nvar date = n.slice(0, 10)\nvar time = n.slice(11,16)\nglobal.set('time', time) \nmsg.payload=date+\" \"+time\nreturn

msg","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":260,"y":180,"wires":[["7b271549d44feb08","bbce3d9843e96361"]]},{"id":"9b460f01bcf1402e","type":"function","z":"75978a6437b1bbee","name":"Function to store the medicine details","func":"var d=msg.payload.date \nvar t=msg.payload.time\nmsg.payload ={\n   \"_id\":d+\" \"+t,\n \"name\":msg.payload.name,\n}\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":270,"y":320,"wires":[["340461ca087b47fd","04e99526fcbd3453","44659e3d6b4ce218"]]},{"id":"04e99526fcbd3453","type":"function","z":"75978a6437b1bbee","name":"Function to make the details in the form empty","func":"msg.payload={\n \"date\":\"\",\n   \"name\":\"\",\n   \"time\":\"\"\n}\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":660,"y":380,"wires":[["e798fd955a7f2182"]]},{"id":"e22a5ccd54ac0b2f","type":"debug","z":"75978a6437b1bbee","name":"name of the medicine","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","statusVal":"","statusType":"auto","x":640,"y":260,"wires":[]},{"id":"44659e3d6b4ce218","type":"debug","z":"75978a6437b1bbee","name":"entered medicine details","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","statusVal":"","statusType":"auto","x":730,"y":440,"wires":[]},{"id":"340461ca087b47fd","type":"cloudant out","z":"75978a6437b1bbee","name":"","cloudant":"ea854c1fd0012e7c","database":"medicine","service":"_ext_","payonly":true,"operation":"insert","x":560,"y":320,"wires":[]},{"id":"96078aa6c00ddf0d","type":"ui_form","z":"75978a6437b1bbee","name":"","label":"","group":"f3161ec9f2852621","order":1,"width":0,"height":0,"options":[{"label":"Enter the name of the Medicine","value":"name","type":"text","required":true,"rows":null},{"label":"Enter the Time(HH:MM)","value":"time","type":"text","required":true,"rows":null},{"label":"Enter the Date(YYYY-MM-DD)","value":"date","type":"text","required":true,"rows":null}],"formValue":{"name":"","time":"","date":""},"payload":"","submit":"submit","cancel":"cancel","topic":"topic","topicType":"msg","splitLayout":"","className":"","x":410,"y":440,"wires":[["44659e3d6b4ce218","9b460f01bcf1402e"]]},{"id":"e798fd955a7f2182","type":"delay","z":"75978a6437b1bbee","name":"","pauseType":"delay","timeout":"10","timeoutUnits":"seconds","rate":"1","nbRateUnits":"1","rateUnits":"second","randomFirst":"1","randomLast":"5","randomUnits":"seconds","drop":false,"allowrate":false,"outputs":1,"x":180,"y":440,"wires":[["96078aa6c00ddf0d"]]},{"id":"4eedb76048312a39","type":"switch","z":"75978a6437b1bbee","name":"","property":"payload","propertyType":"msg","rules":[{"t":"null"},{"t":"else"}],"checkall":"true","repair":false,"outputs":2,"x":470,"y":140,"wires":[[],["5a8fbd45311c12bb","f44de8519caea91e"]]},{"id":"bbce3d9843e96361","type":"debug","z":"75978a6437b1bbee","name":"present

time","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","statusVal":"","statusType":"auto","x":390,"y":240,"wires":[]},{"id":"84a35a31f044a4bb","type":"http in","z":"75978a6437b1bbee","name":"HTTP request","url":"/remainder","method":"get","upload":false,"swaggerDoc":"","x":250,"y":500,"wires":[["976b50d5a54c8088"]]},{"id":"1c0f8856268e30b0","type":"http response","z":"75978a6437b1bbee","name":"","statusCode":"","headers":{},"x":690,"y":500,"wires":[]},{"id":"976b50d5a54c8088","type":"function","z":"75978a6437b1bbee","name":"","func":"msg.payload=global.get(\"mr\")\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":460,"y":500,"wires":[["1c0f8856268e30b0"]]},{"id":"efbfc84d2dd97a08","type":"ubidots_out","z":"75978a6437b1bbee","name":"","token":"BBFF-6rQfgTeR6xojkByAkMpS5Azz9zFJtI","label_device":"","device_label":"Medicine_reminder","tier":"educational","tls_checkbox":true,"x":910,"y":280,"wires":[]},{"id":"f44de8519caea91e","type":"function","z":"75978a6437b1bbee","name":"Email Alert to the caretaker using ubidots IoT platform","func":"msg.payload={\"gurubaran\":1}\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":740,"y":160,"wires":[["efbfc84d2dd97a08"]]}]

13.4) Github link:

https://github.com/IBM-EPBL/IBM-Project-501-1658304225

13.5) Project-Demo link:

https://youtu.be/YM8yFS6ChkMh