INDEX

CHAPTER 1
INTRODUCTION

1.1. PROJECT OVERVIEW

Personal assistance App is meant to aid the forgetful and busy with remembering to take their daily medications. It is designed for users who need a little help keeping track of their medication schedule and who are dedicated to keeping the schedule.

It is difficult for doctors/caretakers to monitor the patients around the clock. To avoid this problem, this medicine reminder system is developed. A personal assistance app will help you to take medications in a particular time.

Personal assistance applications will ask users to add the medicine names and time based upon their prescription, reminder will be updated which will be visible to the user. Also, users canget a voice commands to take their medications. They have an option to set an alarm which will notify them to take medications, they can either stop or snooze the alarm as per their need.

1.2. PURPOSE

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.

An app is built for the user (caretaker) which enables him to set the desired time and medicine. These details will be stored in the IBM Cloudant DB.If the medicine time arrives the web application will send the medicine name to the IoT Device through the IBM IoT platform.The device will receive the medicine name and notify the user with voice commands.

CHAPTER 2

LITERATURE SURVEY

2.1. EXISTING PROBLEM

1. Older adults often prefer to stay in their homes rather than enter a healthcare Institution. Although in own-home assistance might appear as the better solution to elderly people, it may require efforts from informal caregivers such as family members, friends, neighbours, and volunteers hardly compatible with the family and social lifestyle.

2.There is a possibility that the screening process, the analysis and interpretation of the them was influenced by the author's own perceptions or understanding of the topic.

3. This system can be improved, providing a more flexible way to schedule medicines consumption alarms such as twice a week, three times a week, every other day, among others.

4. To improve the proposed system, closing the dispenser compartments so that they only open when the camera detects the face of the caregiver who must place the medicine boxes in them .This would make it safer. It would also be good for the system to automatically detect which medicines and how many of them the caregiver has put in the different compartments; currently ,he/she is who must provide these data through the mobile app.

2.2. REFERENCES

1. https://pubmed.ncbi.nlm.nih.gov/31331279/

2. https://www.researchgate.net/publication/336879808_Using_IoT_technologies_to_develop_a_low-cost_smart_medicine_box

3. https://www.researchgate.net/publication/345347015_IoT-Based_Smart_Medicine_Dispenser_to_Control_and_Supervise_Medication_Intake

2.3. PROBLEM STATEMENT DEFINITION

Personal assistance App is meant to aid the forgetful and busy with remembering to take their daily medications. It is designed for users who need a little help keeping track of their medication schedule and who are dedicated to keeping the schedule.

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.

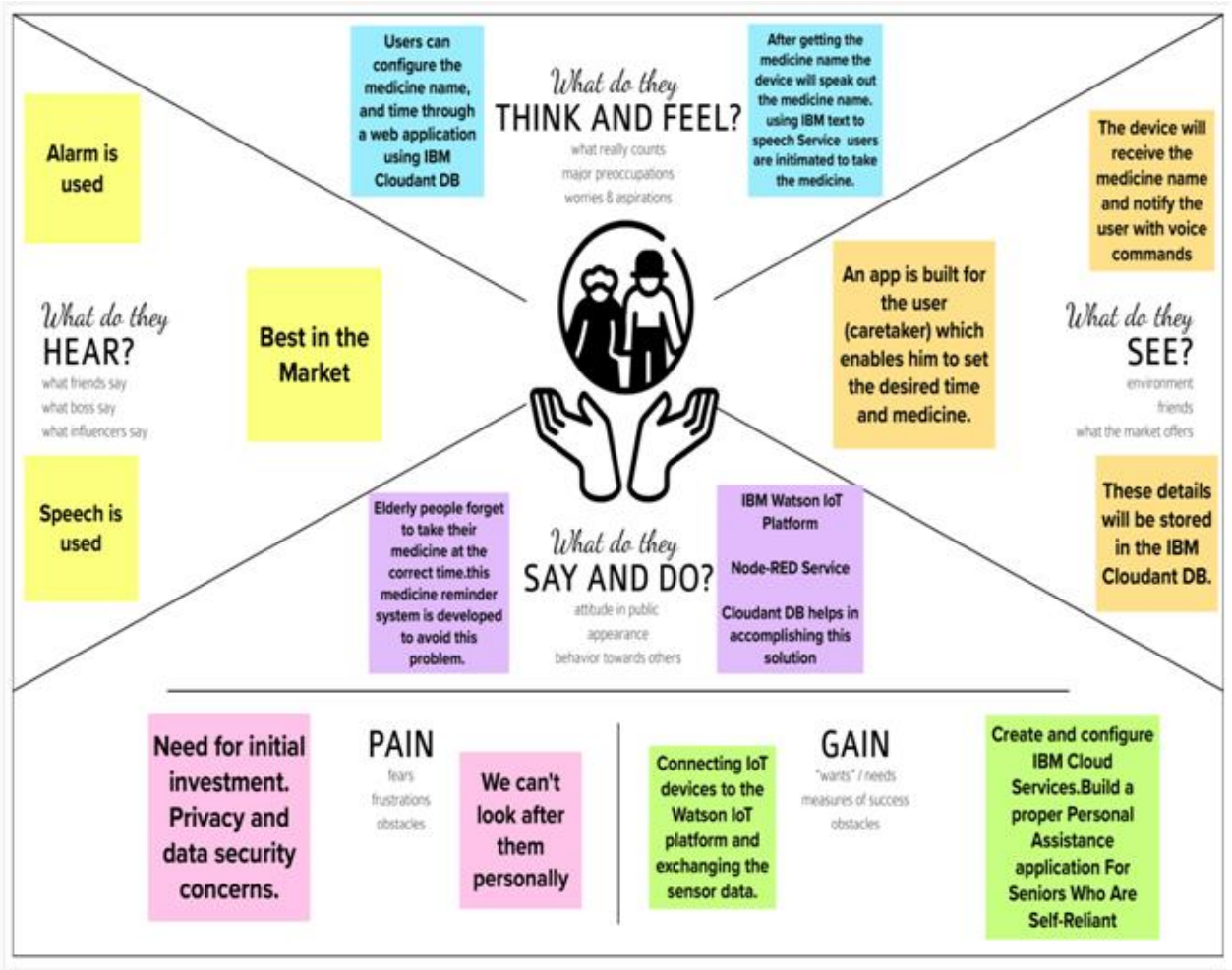
An app is built for the user (caretaker) which enables him to set the desired time and medicine. These details will be stored in the IBM Cloudant DB.If the medicine time arrives the web application will send the medicine name to the IoT Device through the IBM IoT platform.The device will receive the medicine name and notify the user with voice commands.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

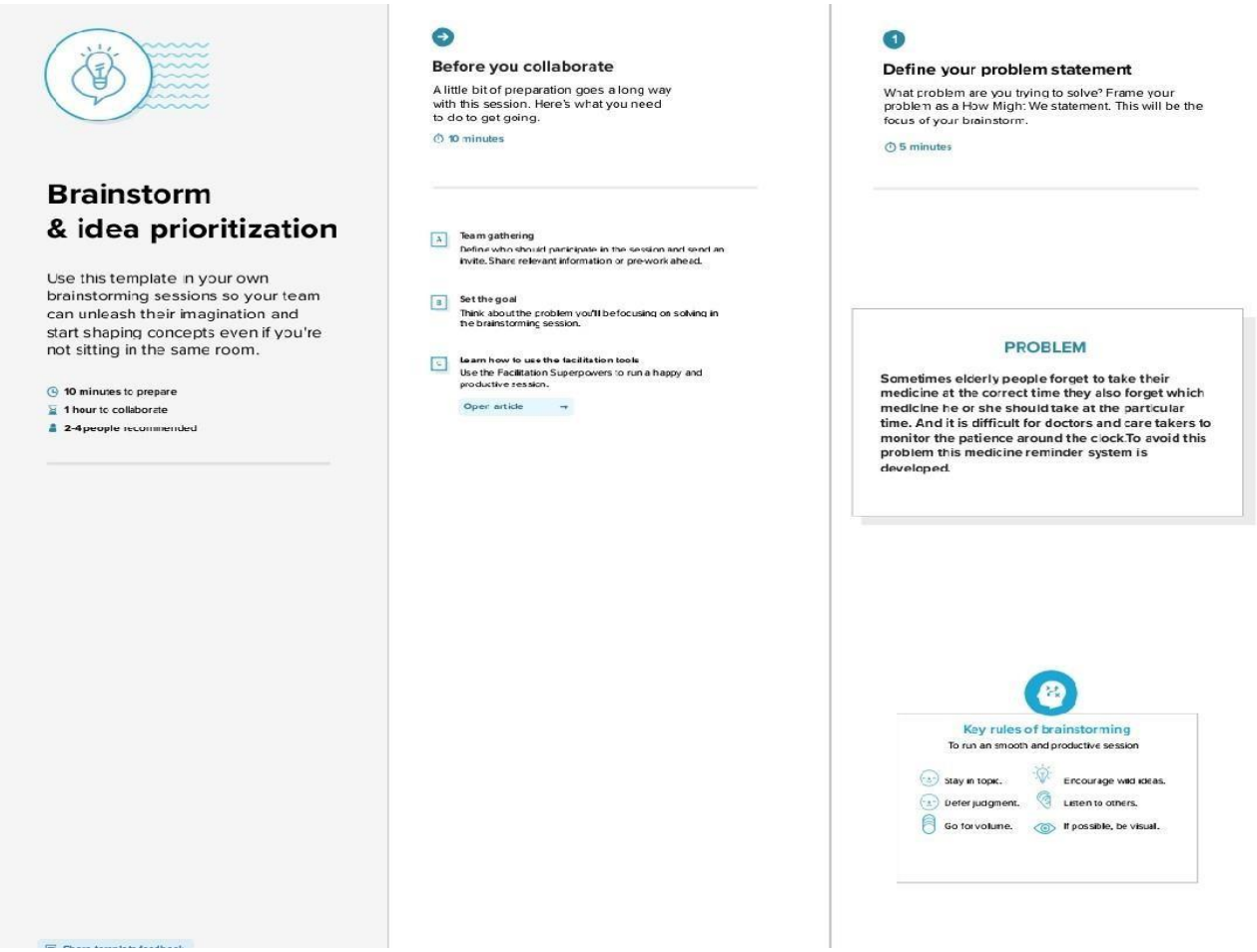
3.1. EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the personwho is experiencing it. The exercise of creating the map helps participants consider thingsfrom the user's perspective along with his or her goals and challenges.

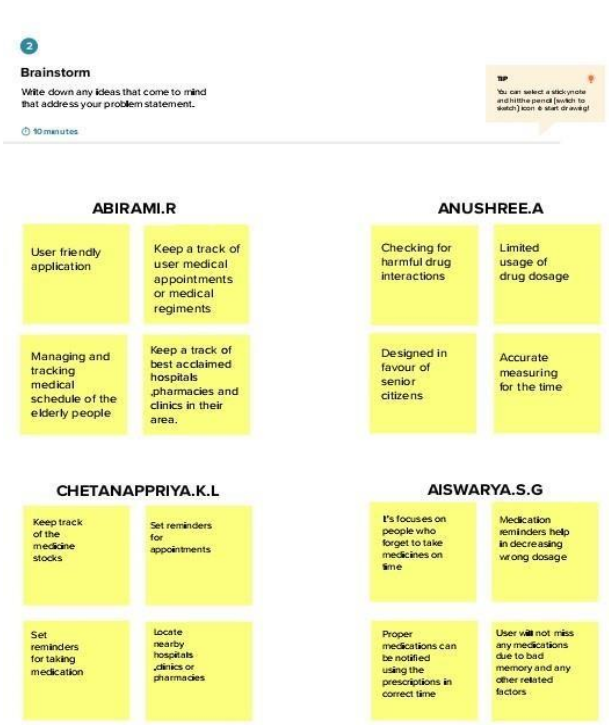


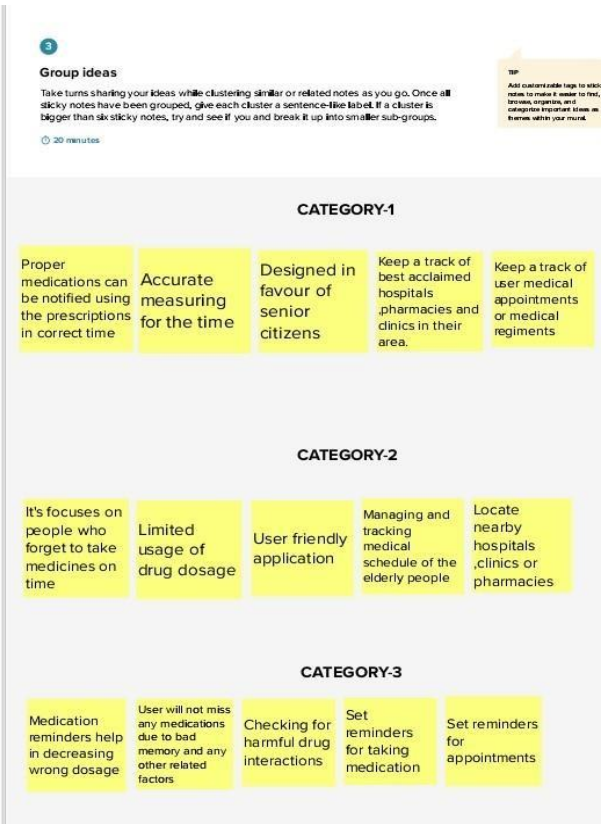
3.2. IDEATION & BRAINSTROMING

Step-1: Team Gathering, Collaboration and Select the Problem Statement

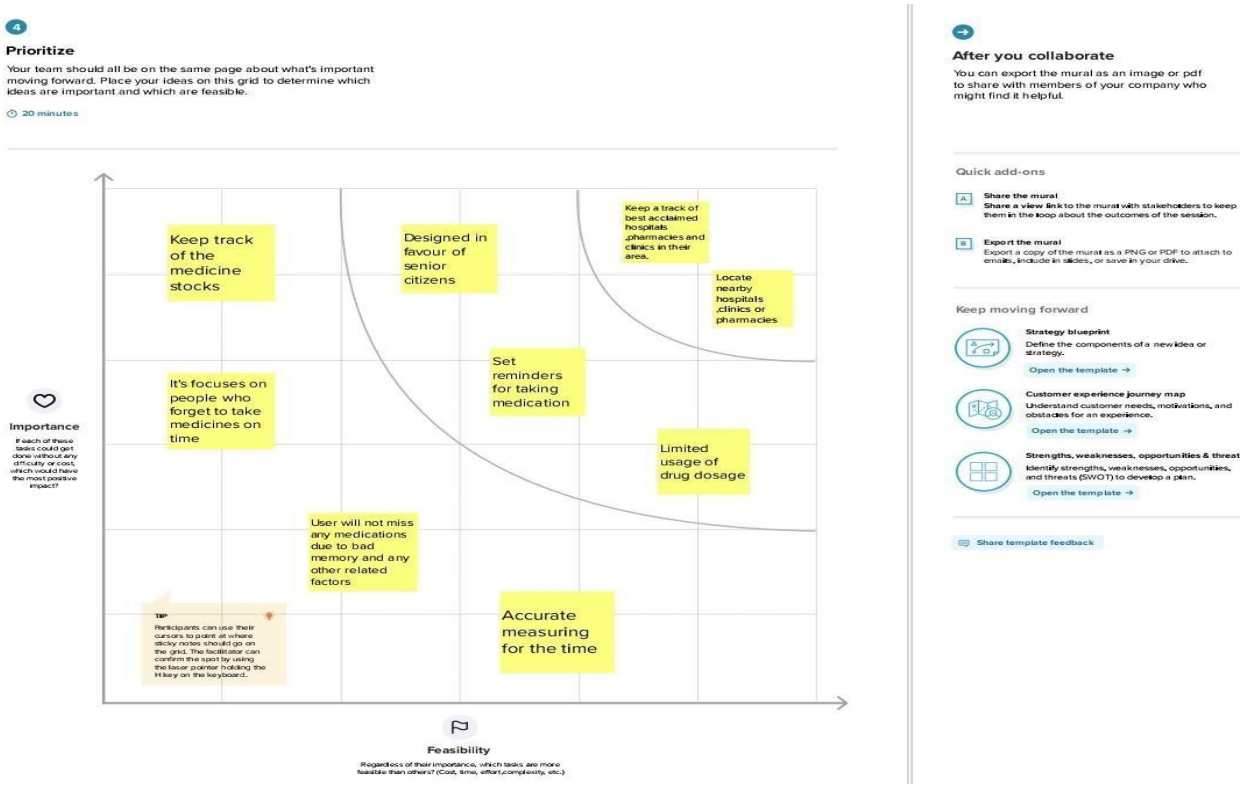


Step-2: Brainstorm, Idea Listing and Grouping





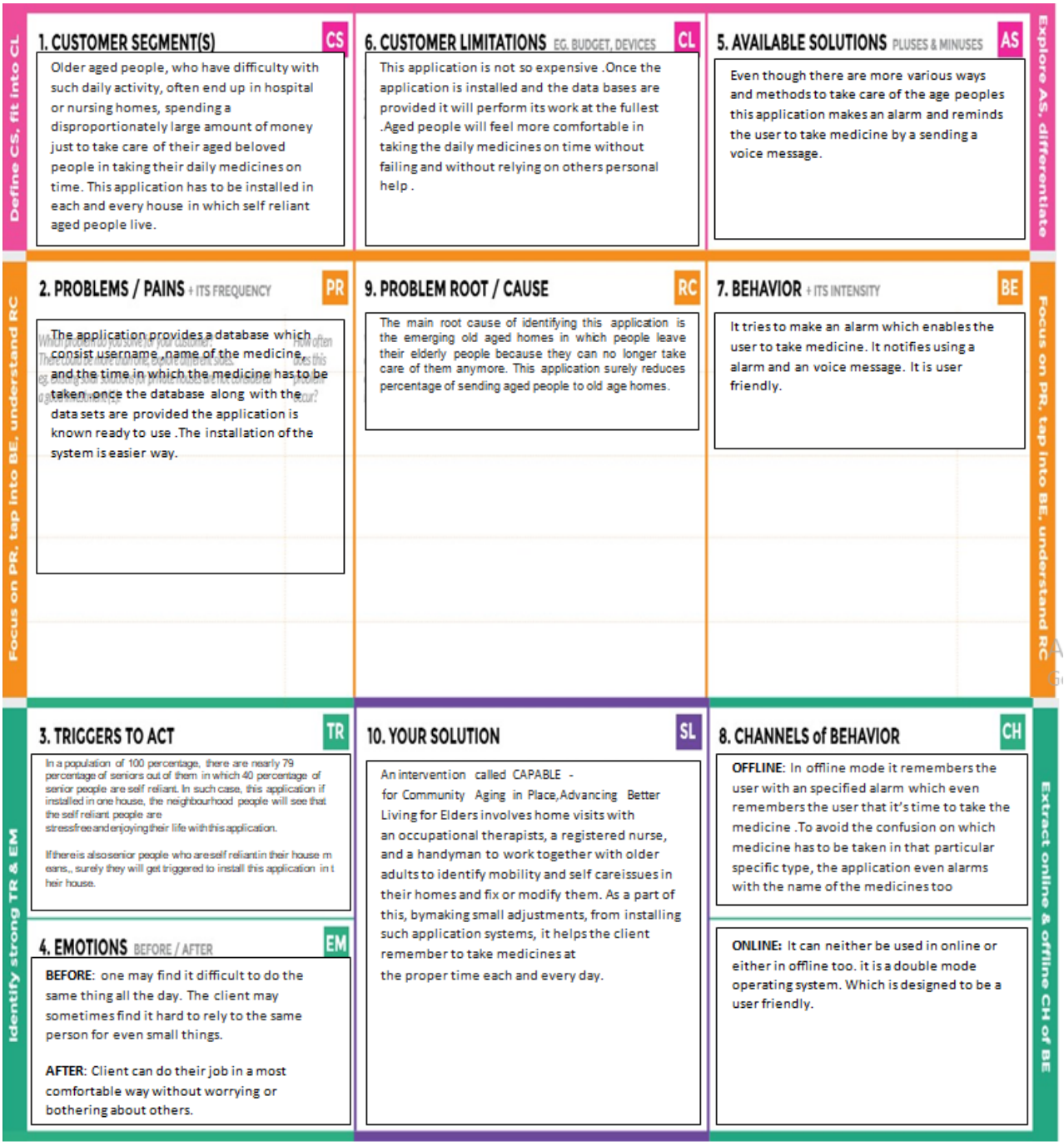
Step-3: Idea Prioritization



3.3. PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Some people find it difficult to learn new apps in this ever-expanding digital environment, and people nowadays tend to forget things more easily, such as taking their prescriptions. People need a way to remember to take their prescriptions without having to learn how to use sophisticated programs.
2.	Idea / Solution description	Create a basic, easy-to-use app so that users don't forget their medicine schedules, can easily discover pharmacies and clinics near them, and can be directed through the app by their loved ones if necessary.
3.	Novelty / Uniqueness	My research began with a series of inquiries directed at a variety of people in order to have a better understanding of their issues and demands in remembering their routines. The purpose of this study was to gain a better understanding of individuals and their needs, as well as to put them at the centre of our design process and product.
4.	Social Impact / Customer Satisfaction	I constructed these proto-personas, or names, based on the research findings from the user interview. They would be crucial to the rest of the design process. All design decisions may be assessed and re-evaluated using these personas, keeping the user and their perspective in mind.

3.4. PROBLEM SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

4.1. FUNTIONAL REQUIREMENT

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through G-mail Registration by phone number
FR-2	User Confirmation	Confirmation via Email Confirmation through SMS/Messages
FR-3	User Login (Web)	Login with registered mail id and password
FR-4	User Login (mobile app)	Login with registered mobile number and password
FR-5	User's Medical Information	In the app,enter your medicine details with date. Then set the time in the app.
FR-6	User input management	All the user's data are got with the help of a text field in the dashboard in the app.

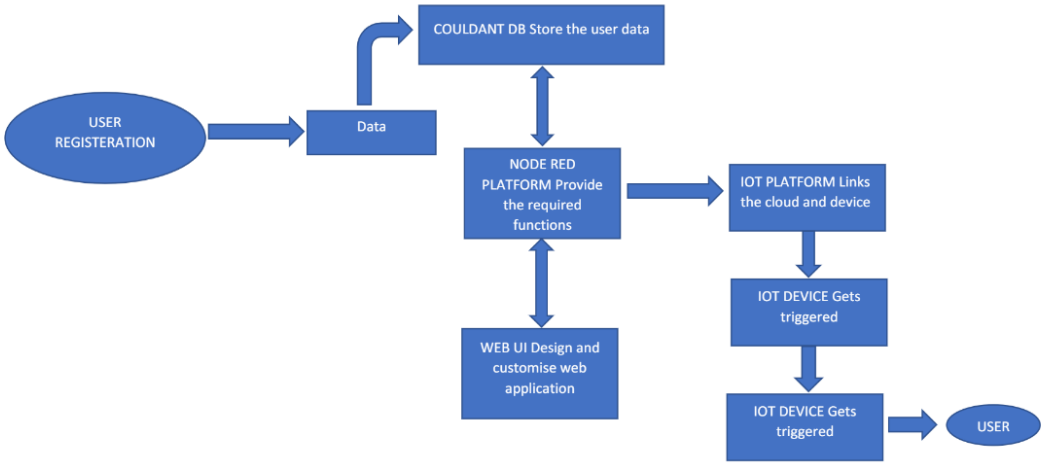
4.2. NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system should be user-friendly for the users. It is used to remind the medicine names. It alerts the users through voice commands.
NFR-2	Security	The login information should not be accessed byany other users than the respective. The data of the users should be kept confidential.
NFR-3	Reliability	It reminds on correct time The user data should be updated and examined after certain period of time.
NFR-4	Performance	The voice message will be delivered accurately to the given time. It works without any connection interruption
NFR-5	Availability	The system should be monitored 24X7 for the alert of medicines. It can be used by any registered users from any place.
NFR-6	Scalability	It is easily adaptable The device is compatible and portable The application can handle any number of registration

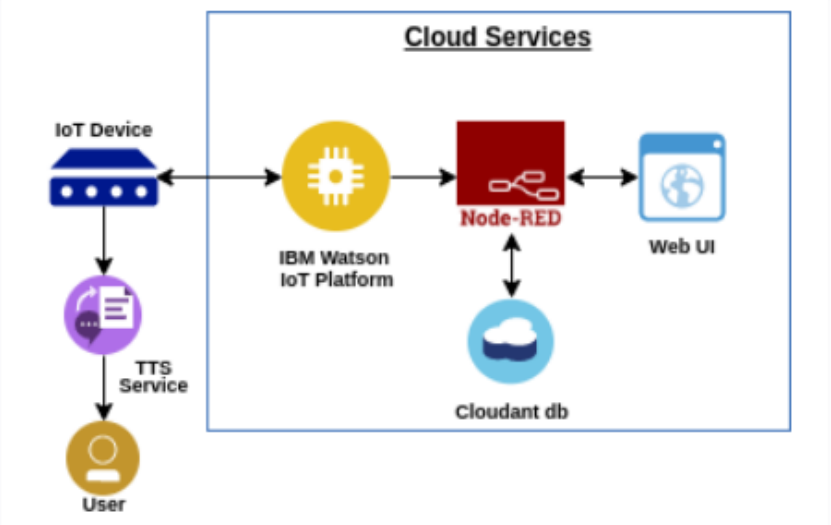
HAPTER 5

PROJECT DESIGN

5.1. DATA FLOW DIAGRAMS



5.2. SOLUTION & TECHNICAL ARCHITECTURE



5.2. USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Senior user)	caretaker	USN-1	As a user, I want to take Medicines on time and monitor my health	I want to take medicines on time	High	Sprint-1
Customer (Diabetes Patient)	Smart medicine box	USN-2	As a user, I want to take Medicines on time and monitor my health	I want to take my tablets on time by voice command	High	Sprint-1
Customer (Thyroid Patient)	Smart medicine box	USN-3	As a user, my patient needs to take medicines on time and monitoring the activity	My patient needs to take medicines on time	Medium	Sprint-2
Customer (Coma Patient)	Caretaker	USN-4	As a user, my patient needs medication time and prescription should load indatabase for upcoming week	My patient medication time and prescription should be in database list	Low	Sprint-4
Customer (Disabled *People's)*	Smart medicinebox	USN-5	As a user ,i need to take my medicine in nearby places with light notification	I need to take my medicine in nearby places with light notification	Medium	Sprint-3

CHAPTER 6

PROJECT PLANNING & SCHEDULING

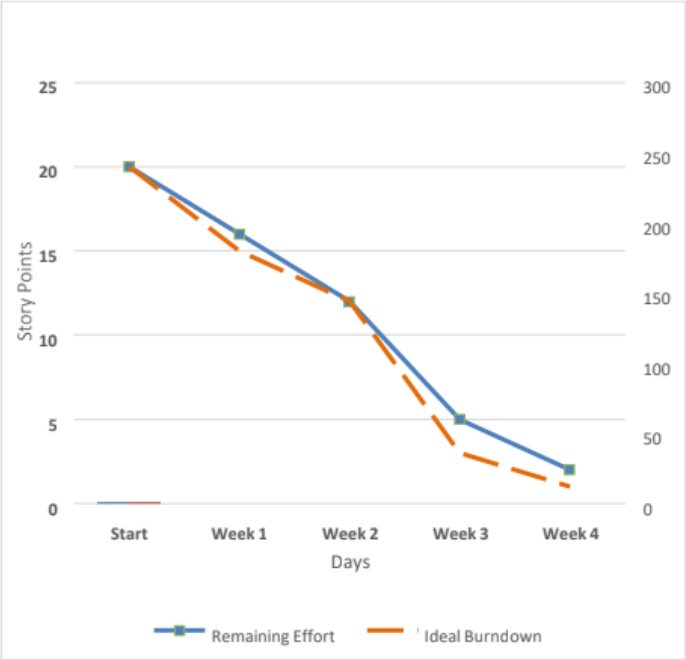
6.1. SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story points	Priority	Team Members
Sprint 1	Set Alarm	USN-1	As a user, I can set an alarm for alerting to take medicine through Medicine remainder system.	10	High	Anushree A
Sprint 1		USN-2	As a user, I can Activate and Deactivate the alarm	10	High	Abirami R
Sprint 2	Voice Assistant	USN-3	As a user once I set the alarm it will alert us voice assistant.	10	High	Chetanappriya KL
Sprint 2		USN-4	It will tell us the time and name of the medicine once the time has set.	10	High	Aiswarya SG
Sprint 3	Cloudant DB	USN-6	For storing the details of medicine reminder for which Cloud DB is used	5	low	Anushree A, Chetanappriya KL, Aiswarya SG, Abirami R
Sprint 3		USN-7	As a user, I can store the name of the medicine with timing.	10	High	Abirami R, Anushree A, Chetanappriya KL, Aiswarya SG
Sprint 4	User Friendly Application	USN-8	Our app will be companian for the senior citizen to consumes the medicines on time.	5	Low	Abirami R, Anushree A, Chetanappriya KL, Aiswarya SG
		USN-9	As a user, one need to set the medicine and time as per the instruction given by the user the voice assistan will help to take medicines on time for senior citizens.	10	High	Anushree A, Abirami R, Chetanappriya KL, Aiswarya SG

6.2. SPRINT DELIVERY SCHEDULE

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	8 days	02-11-2022	09-11-2022	20	8-11-2022
Sprint 2	10	8 days	05-11-2022	12-11-2022	10	11-11-2022
Sprint 3	20	8 days	07-11-2022	14-11-2022	20	23-11-2022
Sprint 4	10	8 days	10-11-2022	17-11-2022	10	30-11-2022

6.3. REPORTS FROM JIRA

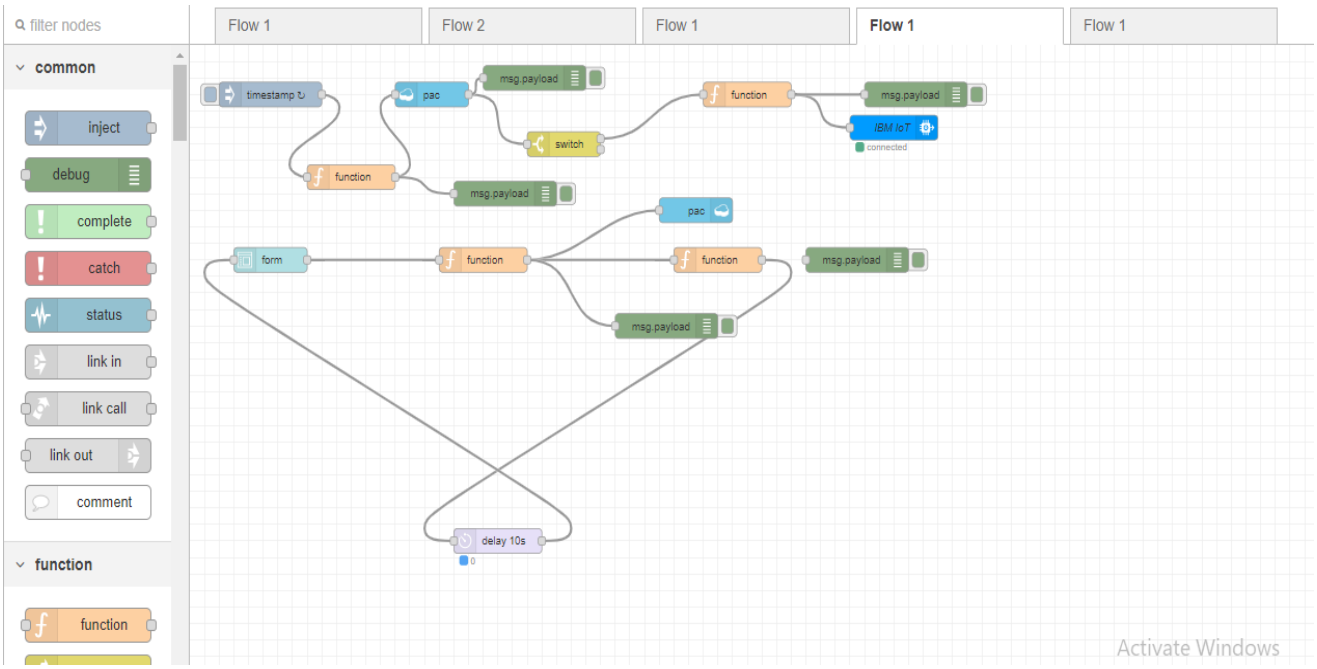


CHAPTER 7

CODING & SOLUTIONS

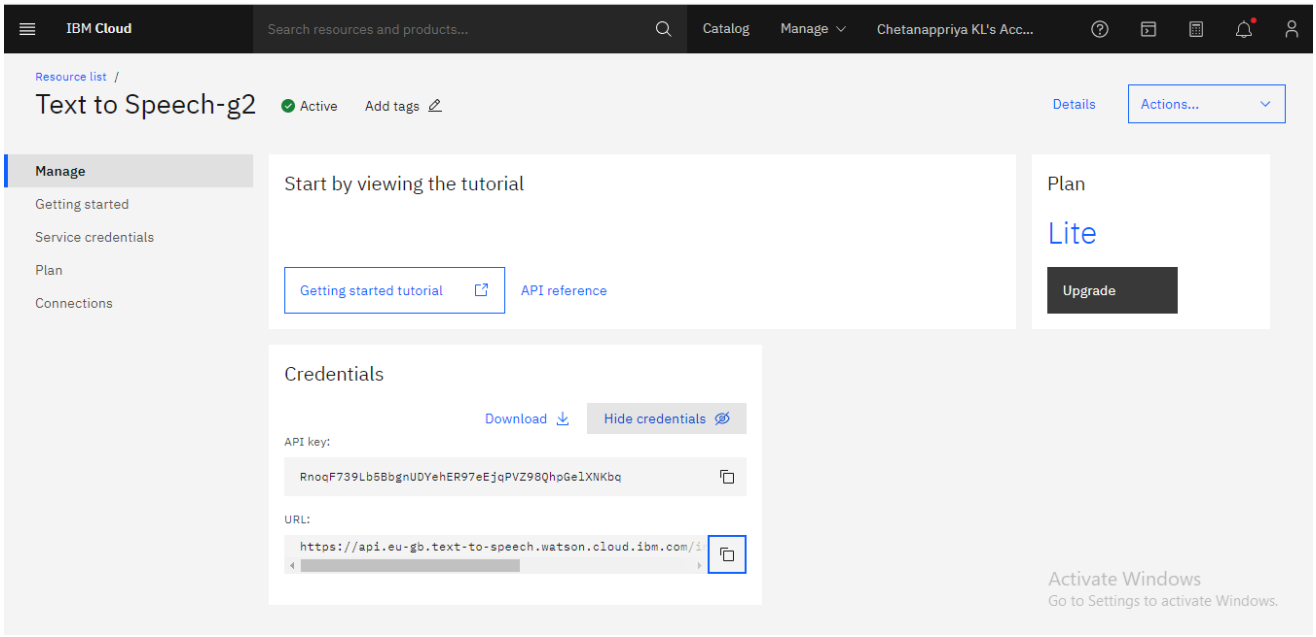
7.1. MEDICINE.PY (PYTHON SCRIPT TO RECEIVE DATA FROM NODE-RED)

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "j9acsm"
deviceType = "medicine"
deviceId = "09876"
authMethod = "token"
authToken = "12345678910"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    name=cmd.data['command']
    print ("Take medicine :" +name)

try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
        #.....................................................

except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
        sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting"
10 times
deviceCli.connect()

while True:
      #Get Sensor Data from DHT11
      deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

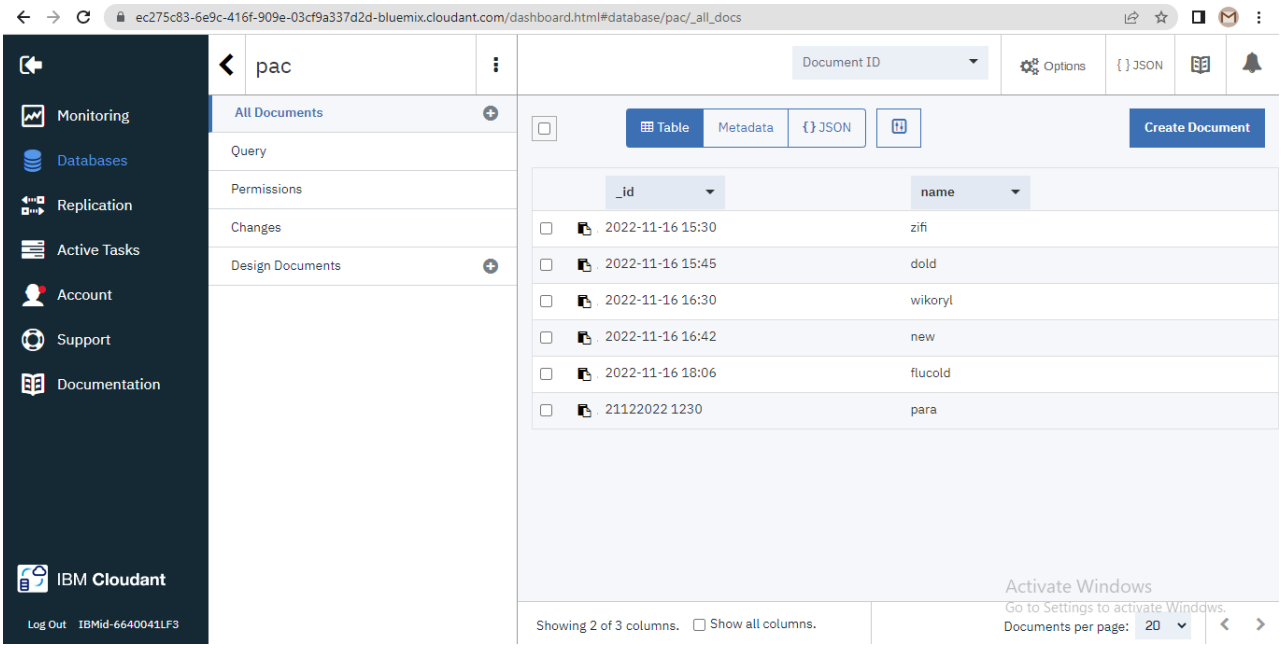


7.2. TTS.PY (PROGRAM FOR ACCESSING TEXT TO SPEECH SERVICE)

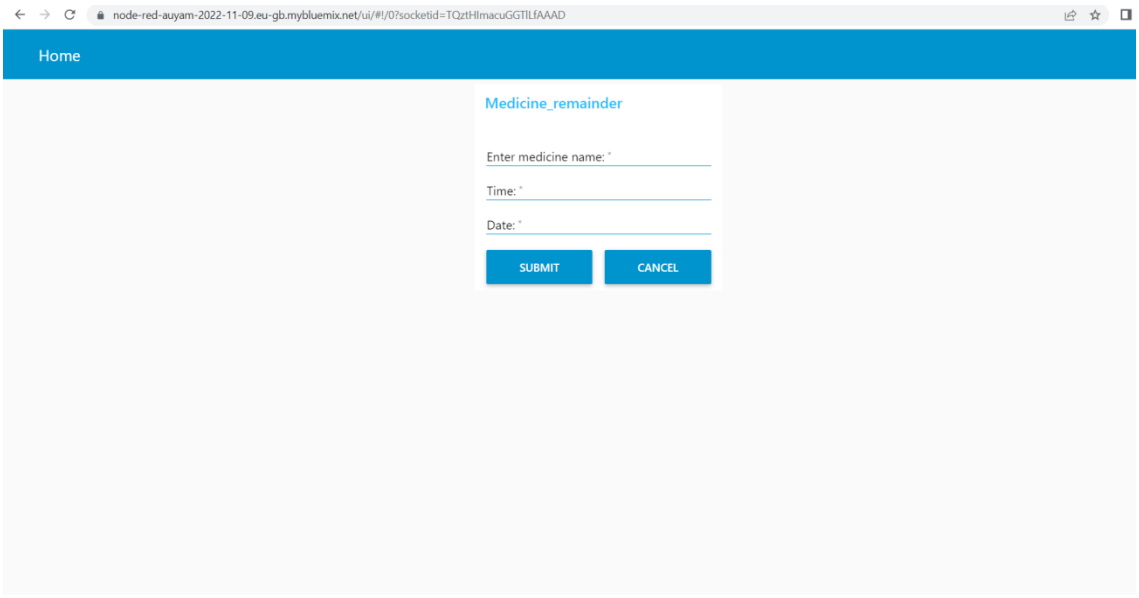
```
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
authenticator = IAMAuthenticator('RnoqF739Lb5BbgnUDYehER97eEjqPVZ98QhpGelXNKbq')
text_to_speech = TextToSpeechV1( authenticator=authenticator)

text_to_speech.set_service_url('https://api.eu-gb.text-to-
speech.watson.cloud.ibm.com/instances/f60eb2cf-126a-4812-baf3-1a3eb7324f54')
with open('hello_world.wav', 'wb') as audio_file: audio_file.write( text_to_speech.synthesize(
'its time to take ',voice='en-US_AllisonV3Voice', accept='audio/wav').get_result().content)
```



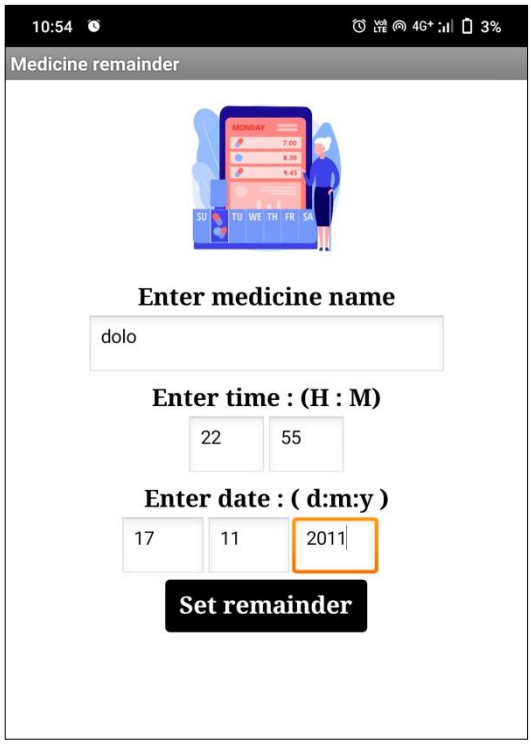
7.3. DATABASE SCHEMA

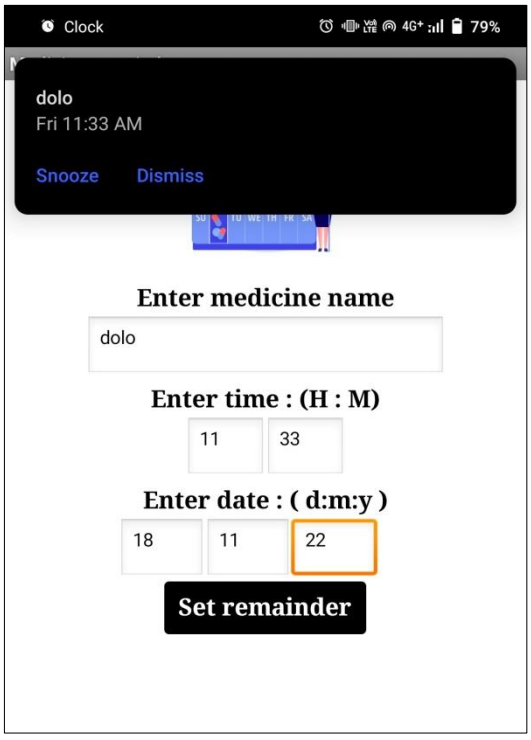


WEB USER INTERFACE:



APPLICATION:





- Voice commands are also been generated. (The generation of voice command is attached in the demo link).

CHAPTER 8

TESTING

8.1. TEST CASES

S.No	MEDICINE NAME	TIME (H:M)	Expected output	Actual Output
1	Dolo	11:20	Alarm is set	Alarm is set
2	Flucold	12:00	Alarm is set	Alarm is set
3	Wikoryl	23:54	Alarm is set	Alarm is set
4	vicks	1:20	Alarm is set	Alarm is set

8.2. USER ACCEPTANCE TESTING

- The web application will be served to the user for setting reminder.
- The user can set alarm or reminder by running the application.
- Based on the medicine name and time entered in the app , an alarm is set and a voice command is played at the entered time.

CHAPTER 9

RESULTS

9.1. PERFORMANCE METRICS

The accuracy of the app is 96.5 %.

NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Voluem Changes	Risk Score	Justification
1	Personal Assistance for Seniors Who Are Self-Reliant	New	Low	Moderate	Moderate	Low	>5 to 10%	ORANGE	As we have seen the changes

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	Medicine Reminder Web -UI	Stress	App Crash/ Developer team/ Site Down	Approved
2	Medicine Reminder Web -UI	Endurance	App Crash/ Site Down	Approved
3	Medicine Reminder Web -UI	Load	erver Crash/ Developer team/ Server Dow	Approved

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	Medicine Reminder Web -UI	Stress	Performance	CPU -01	GO	High Performance server	Closed	Approved
2	Medicine Reminder Web -UI	Load	Scalability	DB Storage - 01	NO-GO	ibm cloud	Closed	Approved
3	Medicine Reminder Web -UI	Endurance	Connectivity	Connection	GO	High Performance	Closed	Approved

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- Receiving remainders to take medications.
- Alarm is set at particular time as per the user requirement.
- Voice commands are received at particular time to take medications.
- No need of internet connection.
- Customer satisfaction.

DISADVANTAGES:

- Possibility of run time problem when user enters two medicines at the same time.
- The need for training in the operation of the new program.

CHAPTER 11
CONCLUSION

The objective of this project was to design and develop a simple, reliable, efficient, and medicine remainder system that has a precise and quick notification mechanism. Appropriate services were used to make reminder alarm and voice commands .A step-by-step approach was followed in the design of the system. The design was carried out based on the study and analysis of existing similar systems and user perceptions. A prototype of the system was implemented and tested in home and office environments. Several tests were conducted, and the results were analysed to ensure that the system produced the intended results. The system has been implemented and tested, showing satisfactory performance.

CHAPTER 12

FUTURE SCOPE

- Users:

  Single point user to multi point user.

- Health info:

  The application helps family members to keep track of the medicines.

- Alert message:

  If the medicines are out of stock it will give an alert message to the user.

- Chat with caretaker:

  Consultation can be taken online.

- Buy medicines:

  We can order medicines through nearby pharmacy

CHAPTER 13

APPENDIX

13.1. SOURCE CODE

MEDICINE.PY

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "j9acsm"
deviceType = "medicine"
deviceId = "09876"
authMethod = "token"
authToken = "12345678910"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    name=cmd.data['command']
    print ("Take medicine :" +name)

try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
        #.....................................

except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
        sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting"
10 times
deviceCli.connect()

while True:
      #Get Sensor Data from DHT11
      deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

TTS.PY

```
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
authenticator = IAMAuthenticator('RnoqF739Lb5BbgnUDYehER97eEjqPVZ98QhpGelXNKbq')
text_to_speech = TextToSpeechV1( authenticator=authenticator)

text_to_speech.set_service_url('https://api.eu-gb.text-to-
speech.watson.cloud.ibm.com/instances/f60eb2cf-126a-4812-baf3-1a3eb7324f54')
with open('hello_world.wav', 'wb') as audio_file: audio_file.write( text_to_speech.synthesize(
'its time to take ',voice='en-US_AllisonV3Voice', accept='audio/wav').get_result().content)
```

13.2. LINKS:

- **GitHub link:**

  https://github.com/IBM-EPBL/IBM-Project-17150-1659629254

- **Application link:**

  https://drive.google.com/file/d/1fzsf45YJNHOqsYw7GPvkA88bO5Dl_1ln/view?usp=sharing

- **Demo link:**