FINAL PROJECT REPORT

1. INTRODUCTION

1.1. Project Overview

Sometimes elderly people forget to take their medicine at the correct time. They also forget which medicine they should take at that time. It is also difficult for doctors/caretakers to monitor the patients around the clock. To avoid this problem, a reminder system has been proposed. 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 (Internet of Things) Device through the IBM IoT platform. The device will receive the medicine name and notify the user with voice commands.

Keywords: *IoT, Web Application, Medicine Intake, IBM Cloudant.*

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.

2. LITERATURE SURVEY

2.1. Existing problem

Elderly people do skip their medications or forget to take the medicines at the correct time and the existing solutions for this, a solution that exists is to set a reminder or using pill boxes, calendars, Personal Assistance. Though the solutions give reminders, the voice commands or assistance given by this system is more efficient.

2.2. References

- Santo, K., Chow, C. K., Thiagalingam, A., Rogers, K., Chalmers, J., & Redfern, J. (2017). MEDication reminder APPs to improve medication Adherence in Coronary Heart Disease (MedApp-CHD) Study: a randomised controlled trial protocol. BMJ open, 7(10), e017540.

- Waykole, M., Prakash, V., & Nalini, N. H. S. (2016). Ardumed-smart medicine reminder for old people. Int J Sci Eng Res, 7, 650-654.
- Al-Mahmud, O., Khan, K., Roy, R., & Alamgir, F. M. (2020, June). Internet of things (IoT) based smart health care medical box for elderly people.
- In 2020 International Conference for Emerging Technology (INCET) (pp. 1-6). IEEE.
- Kader, M. A., Uddin, M. N., Arfi, A. M., Islam, N., & Anisuzzaman, M. (2018, October). Design & implementation of an automated reminder medicine box for old people and hospitals. In 2018 International conference on innovations in science, engineering, and technology (ICISET) (pp. 390-394). IEEE.Yamamoto, Y., Huang, R., & Ma, J. (2010, November). Medicine management and medicine taking assistance system for supporting elderly care at home. In 2010 2nd International Symposium on Aware Computing (pp. 31-37). IEEE.

2.3. Problem Statement Definition

Sometimes elderly people forget to take their medicine at the correct time.

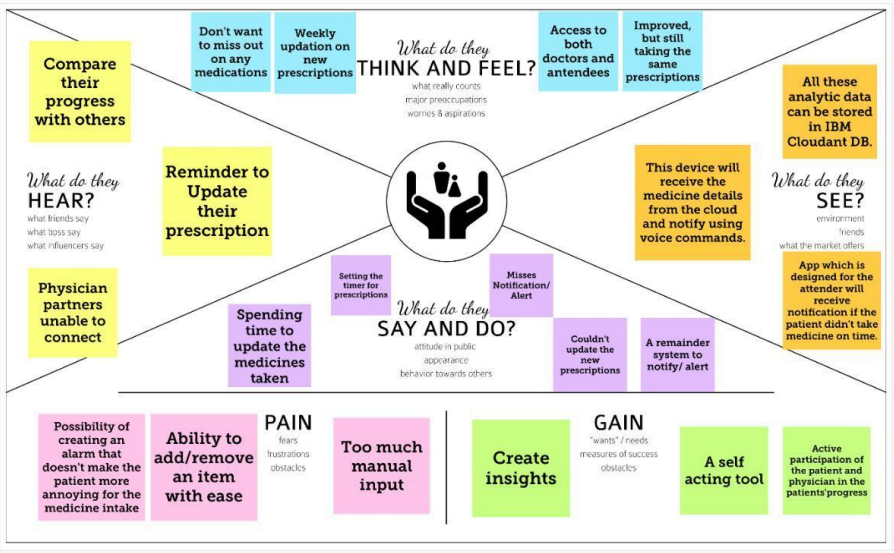
- They also forget which medicine He / She should take at that particular time.
- And it is difficult for doctors/caretakers to monitor the patients around the clock. To avoid this problem, this medicine reminder system is developed.
- An app is built for the user (caretaker) which enables him to set the desired time and medicine.

These details will be stored in the IBM Cloud-ant 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.

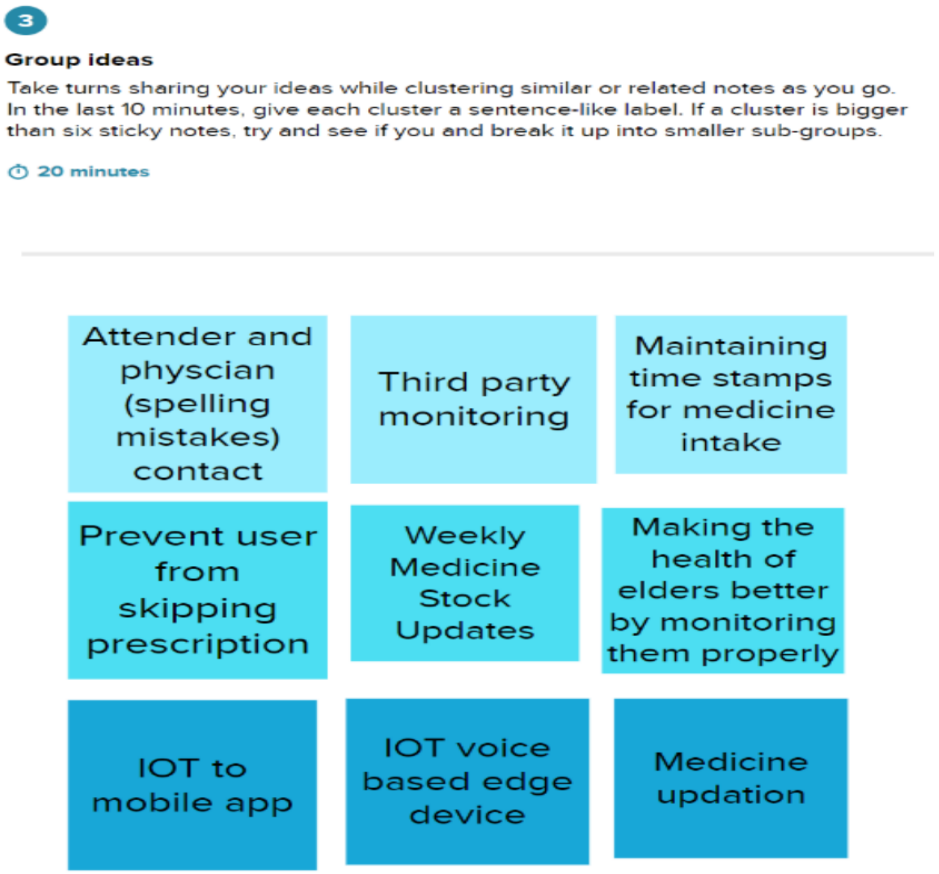
3. IDEATION & PROPOSED SOLUTION

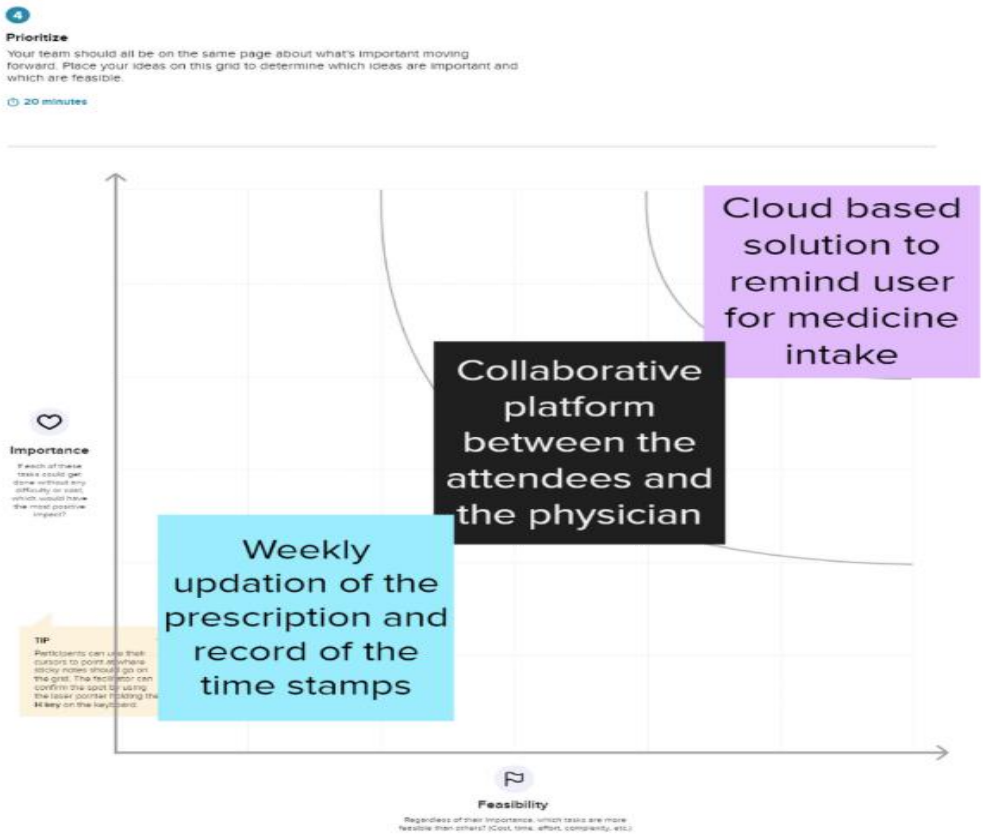
3.1. Empathy Map Canvas



Brainstorm & Idea Prioritization:

3.2. Ideation & Brainstorming





3.3. Proposed Solution

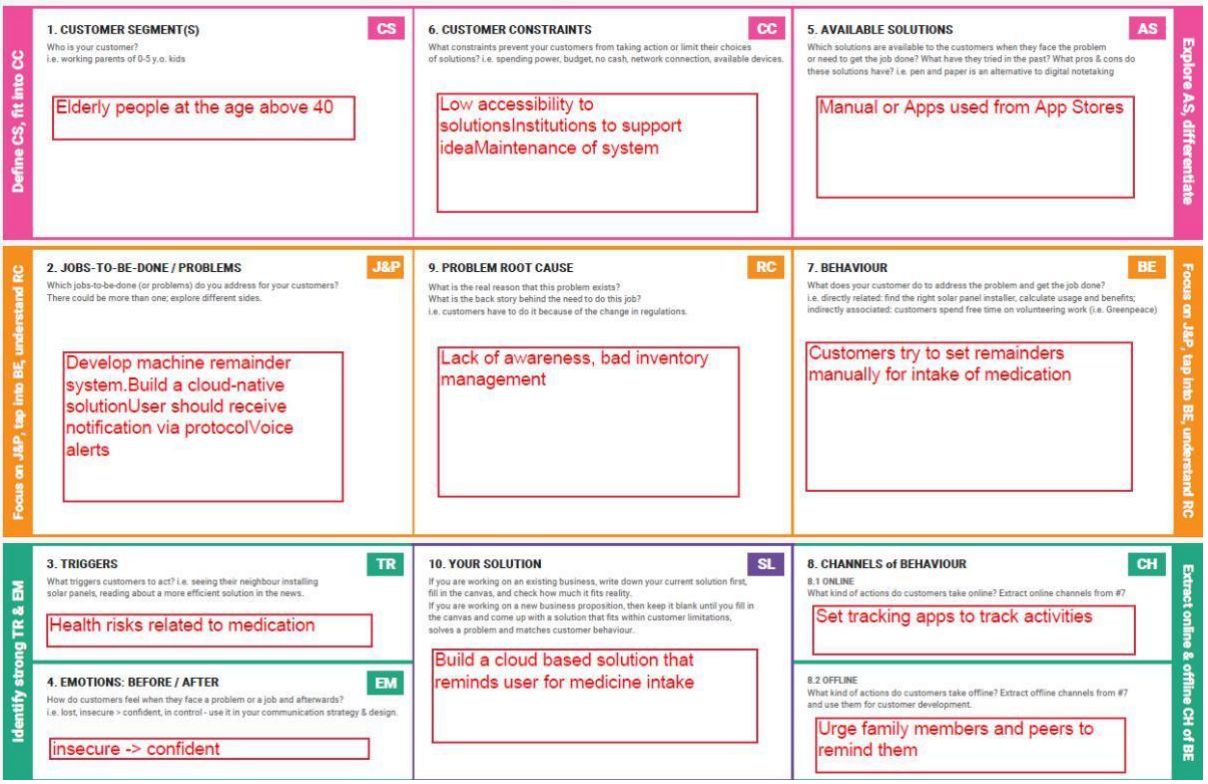
S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Personal Assistance for Seniors Who Are Self-Reliant
2.	Idea / Solution description	IoT – edge based real time cloud solution, that reminds the user to take the prescription on time.
3.	Novelty / Uniqueness	The prototype prevents the user from skipping them, by sending periodic reminders to both the attenders and the patients
4.	Social Impact / Customer Satisfaction	The client looks for a model that is feasible and easy to use. Since the ones we are targeting are the elderly, it should be a model that enables them to handle the model on their own.
5.	Business Model (Revenue Model)	The business impact of this project lies on the hands of target audiences including elderly and physicians.
6.	Scalability of the Solution	The project we are proposing is based on cloud, so the idea to expand the prototype to a next level will not involve more manpower since everything happens via cloud.

3.4. Problem Solution Fit

Here our problem is elderly people forget to take their medicine at the correct time and they forget which medicine to take at that time. It is difficult for the caretakers or doctors to oversee them at the right time every day.

Purpose:

- 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.



4. REQUIREMENT ANALYSIS

4.1. Functional Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Registration**	Registration through Form Registration through Gmail
FR-2	**User Confirmation**	Confirmation via Email Confirmation via OTP
FR-3	**Log in**	Username and Password - credentials
FR-4	**Forget Password**	Link to reset OTP via mail
FR-5	**Navigation**	Simple and easy navigation Limited gesture
FR-5	**Cloud Services**	Easy access of database

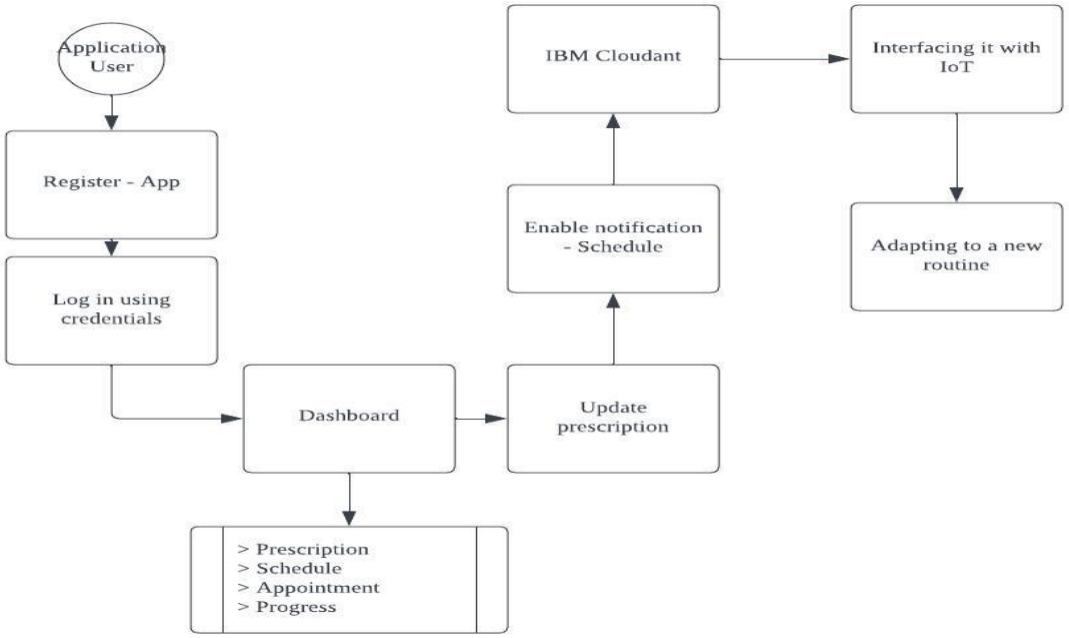
4.2. Non-Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Registration**	Registration through Form Registration through Gmail
FR-2	**User Confirmation**	Confirmation via Email Confirmation via OTP
FR-3	**Log in**	Username and Password - credentials
FR-4	**Forget Password**	Link to reset OTP via mail
FR-5	**Navigation**	Simple and easy navigation Limited gesture
FR-5	**Cloud Services**	Easy access of database

5. PROJECT DESIGN

5.1. Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

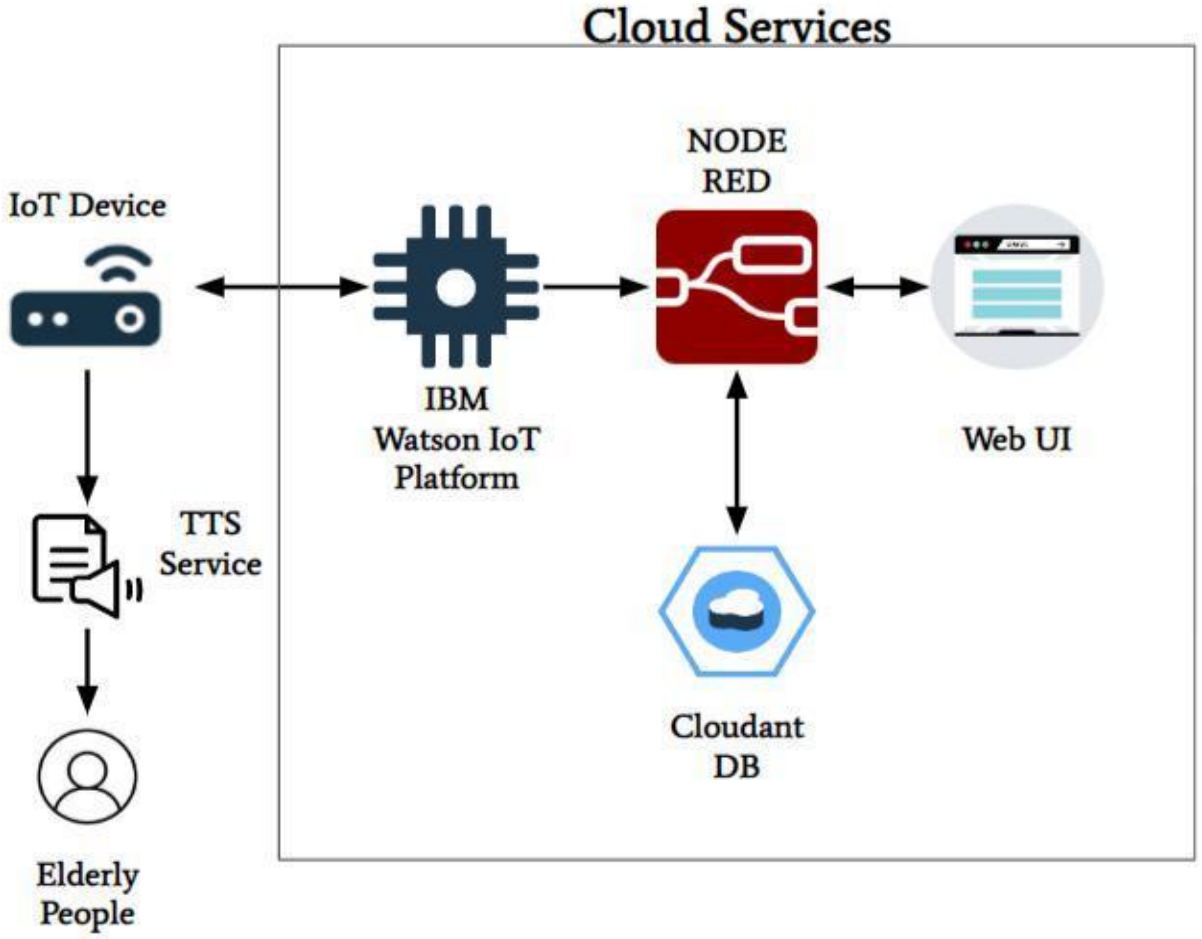


5.2. Solution & Technical Architecture

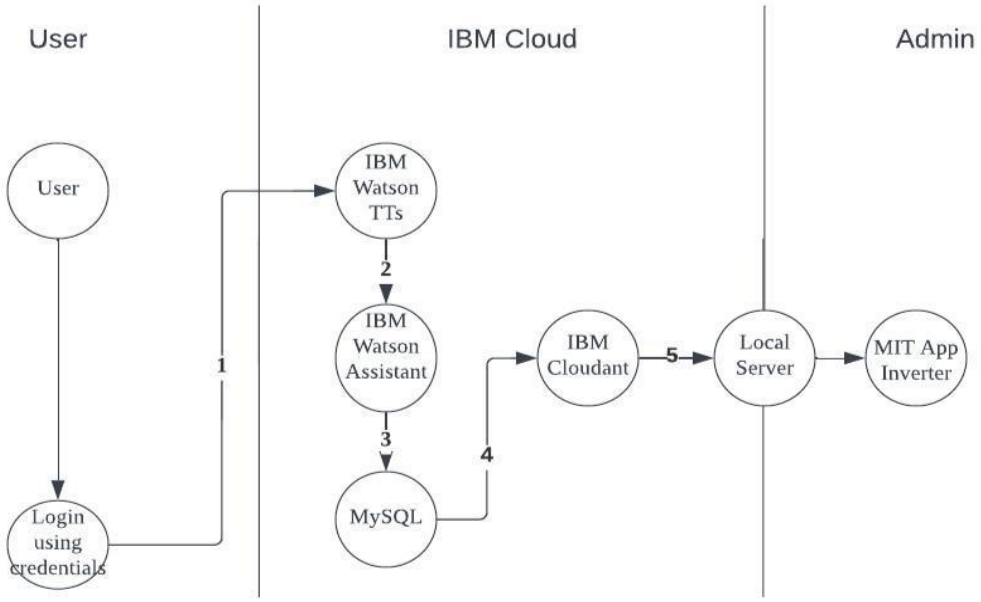
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Attenders and physicians to set remainders in accordance with the prescriptions
- Database to track the activities and timestamps
- Node-Red acts as an interface between the database and the UI
- Build a cloud native solution

Solution Architecture Diagram



Technical Architecture



5.3. User Stories

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Registration Page	IBM-5	Developing Page for getting Username & Password	5	High	Rashmi S
Sprint 1	Registration Page	IBM-6	Triggering the Verification mail	5	High	Kaiser
Sprint 1	Registration Page	IBM-7	Giving optional registration methods from Facebook and Google	5	Low	Harish L
Sprint 1	Login Page	IBM-8	Logging in using Email & Password or by other services	5	Medium	Gokul K S
Sprint 2	Dashboard	IBM-9	Dashboard for Patient's Usage	8	High	Gokul K S
Sprint 2	Dashboard	IBM-10	Dashboard for Caretaker's Usage	6	Low	Kaiser
Sprint 2	Dashboard	IBM-11	Dashboard for Physician's Usage	6	Low	Rashmi S
Sprint 3	Backend and Hardware	IBM-12	Node-Red Flow Setup	5	Medium	Gokul K S
Sprint 3	Backend and Hardware	IBM-13	IBM Watson Setup	5	High	Rashmi S
Sprint 3	Backend and Hardware	IBM-14	Cloudant DB Setup	5	High	Kaiser

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 3	Backend and Hardware	IBM-15	Device Setup	5	High	Harish L
Sprint 4	Backend and Hardware	IBM-16	Text to Speech Engine	6	High	Gokul K S
Sprint 4	Backend and Hardware	IBM-17	Enhancing the interface through reiteration (Improvising)	4	High	Kaiser A
Sprint 4	Backend and Hardware	IBM-18	Hardware Integration	10	High	Harish L Kaiser A

6. PROJECT PLANNING & SCHEDULINGS:

6.1. Sprint Planning & Estimation

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	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov2022	19 Nov 2022	20	19 Nov 2022

6.2. Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Registration Page	IBM-5	Developing Page for getting Username & Password	5	High	Rashmi S
Sprint 1	Registration Page	IBM-6	Triggering the Verification mail	5	High	Kaiser
Sprint 1	Registration Page	IBM-7	Giving optional registration methods from Facebook and Google	5	Low	Harish L
Sprint 1	Login Page	IBM-8	Logging in using Email & Password or by other services	5	Medium	Gokul K S
Sprint 2	Dashboard	IBM-9	Dashboard for Patient's Usage	8	High	Gokul K S

Sprint 2	Dashboard	IBM-10	Dashboard for Caretaker's Usage	6	Low	Kaiser
Sprint 2	Dashboard	IBM-11	Dashboard for Physician's Usage	6	Low	Rashmi S
Sprint 3	Backend and Hardware	IBM-12	Node-Red Flow Setup	5	Mediu m	Gokul K S
Sprint 3	Backend and Hardware	IBM-13	IBM Watson Setup	5	High	Rashmi S
Sprint 3	Backend and Hardware	IBM-14	Cloudant DB Setup	5	High	Kaiser
Sprint 3	Backend and Hardware	IBM-15	Device Setup	5	High	Harish L
Sprint 4	Backend and Hardware	IBM-16	Text to Speech Engine	6	High	Gokul K S
Sprint 4	Backend and Hardware	IBM-17	Enhancing the interface through reiteration (Improvising)	4	High	Kaiser A
Sprint 4	Backend and Hardware	IBM-18	Hardware Integration	10	High	Harish L Kaiser A

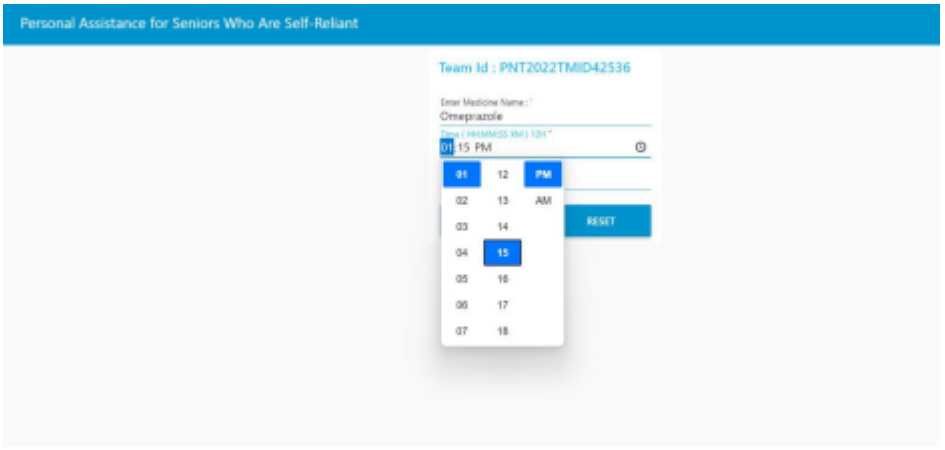
6.3. Reports from JIRA

JIRA Files

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

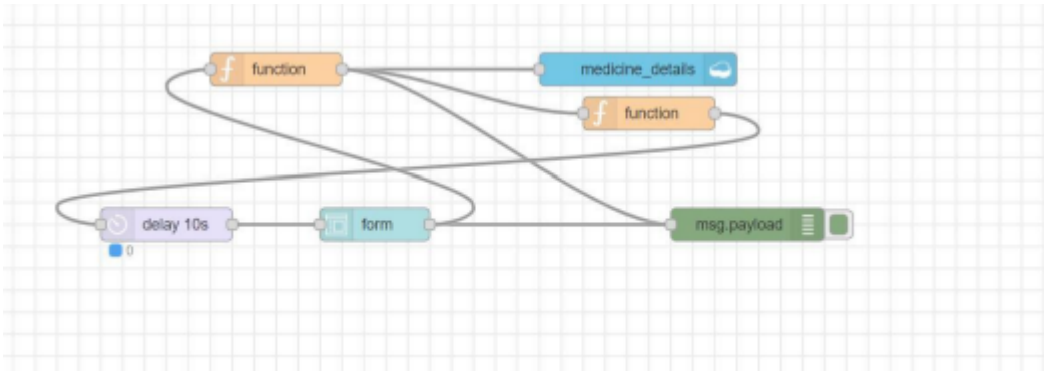
7.1. Feature 1

Flexible UI for the customer to update his schedule periodically



7.2. Feature 2

Convenient database to store the datum of the patient



8. TESTING

8.1. Test Cases

```
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
url
="https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/8e5bc662-02f5-4cc3-b2a3-27086673e789"
api ="QGXbVq1lgSFNn8_7wpT1kGVYIKCHG*NL fHnC1B8XNwj"
auth = IAMAuthenticator(api)
tts=TextToSpeechV1(authenticator=auth)
tts.set_service_url(url)
a={"Command": ["Dart","Paracitamol","Dolo 650"]}
instruction = "Please Take following Medicine."
```

```
for i in a[“commands”]:
        instruction+=i
        instruction+=” ”
with open(“./speech.mp3”,”wb”) as audio_file;

res=tts.synthesize(instruction,accept=äudio/mp3”,voice=én-us_AllisonV3Voice’)
audio_file.write(res.content)
```

8.2. User Acceptance Testing

9. RESULTS

9.1. Performance Metrics:

S. No.	Parameter	Performance
1	Response Time	0.2s (Average of 10 trials)
2	Workload	500 users (Calculated based on Cloud Space)
3	Revenue	Individual users and pharmaceutical industries
4	Efficiency	Simple and straightforward workflow, which makes the process efficient
5	Down Time	Almost no down time due to IBM cloud enabled solution

10. ADVANTAGES & DISADVANTAGES:

10.1. Advantages

- Help the elderly people to take their medicine at the correct time.
- Avoid personal assistants or caretakers needed for medically sick people.
- Cost efficient.
- Can store multiple data and many notifications can be generated.
- Since it includes voice assistance, even blind people can use our device

10.2. Disadvantages

- Makes people lazy and always making them dependent
- Requires a stable data connection

11. CONCLUSION

The Solution offers the elderly or reliable sick people assistant which reminds them of the medicines to be consumed at the particular time. Skipping tablets is never an option and may lead to serious problems if the person has a severe illness and this can be avoided. Since the cloud is integrated with the mobile application, numerous data can be fed into the database and notifications can be generated. The mobile application developed is highly customisable by the user and easy to use.

12. FUTURE SCOPE

The proposed solution can be further developed by bringing into the feature of informing the medicine name during the notification. The voice assistance which is given can be customized by adding the user's voice or the caretaker's voice. Further the mobile application can update medicines by taking voice commands as an input from the user.

13. APPENDIX

13.1. Source Code

```
import wiotp.sdk.device
import time
import random
import playsound
from datetime import datetime
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
myConfig = {
 "identity": {
 "orgId": "jwl2wf",
 "typeId": "SalmanDevice",
 "deviceId":"SalmanDevice_1"
 },
 "auth": {
 "token": "e3DwTZGGA1Y?0BD*s9"
 }
}
def myCommandCallback(cmd):
 print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
 m=cmd.data['command']
 if(m=="Medicine Taken"):
 print("Medicine Intaken\nThank You!!")
 else:
 print("*****")
 print("Take Your Medicine")
 print("******")
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
```

```
authenticator = IAMAuthenticator('mc6GkVtcmmR8o5UlAk5-jhyvsmieCN8nhJ-Xc7awmRly')
text_to_speech = TextToSpeechV1(
 authenticator=authenticator
)
text_to_speech.set_service_url('https://api.eu-gb.text-tospeech.watson.cloud.ibm.com/instances/2ff7f2d9-da46-4f46-bbda-fad6e9f83882')
now = datetime.now()
current_time = now.strftime("%H:%M")
print("Current Time =", current_time)
while True:
 with open('z.mp3','wb') as audio_file:
 audio_file.write(
 text_to_speech.synthesize(
 'take your respected medicine',
 voice='en-US_AllisonV3Voice',
 accept='audio/wav'
 ).get_result().content)
  client.publishEvent(eventId="test", msgFormat="json", data="z.mp3", qos=0, onPublish=None)
 client.commandCallback = myCommandCallback
client.disconnect()
```

GitHub:

Github Link

Project Demo Link:

Video Link