PROJECT REPORT

Personal Assistance for Seniors Who Are Self-Reliant

1.Introduction

1.1 Project Overview

1.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.

2.If the medicine time arrives the web application will send the medicine name to the IOT Device through the IBM IOT platform.

3.The device will receive the medicine name and notify the user with voice commands.

1.2 Purpose

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

2.They also forget which medicine He/she should take at that particular time.

3.And it is difficult for doctors/caretakers to monitor the patients around the clock. To avoid this problem, this medicine remainder system is developed.

2.Literature Survey

2.1 Existing Problem

Elderly people let slip the medications at the correct time and the existing solutions for this problem is setting remainders or using pill boxes ,calendars ,Personal Assistance. Though the solutions give remainders, the voice commands or assistance given by this system is more efficient.

2.2 References

1.Visual Health Remainder: A remainder for Medication Intake and Measuring Blood Pressure to Support Elderly People; Rene Baranyi; Sascha Rainer; Stefan Schlossarek; Nadja Lederer; Thomas Grechenig.

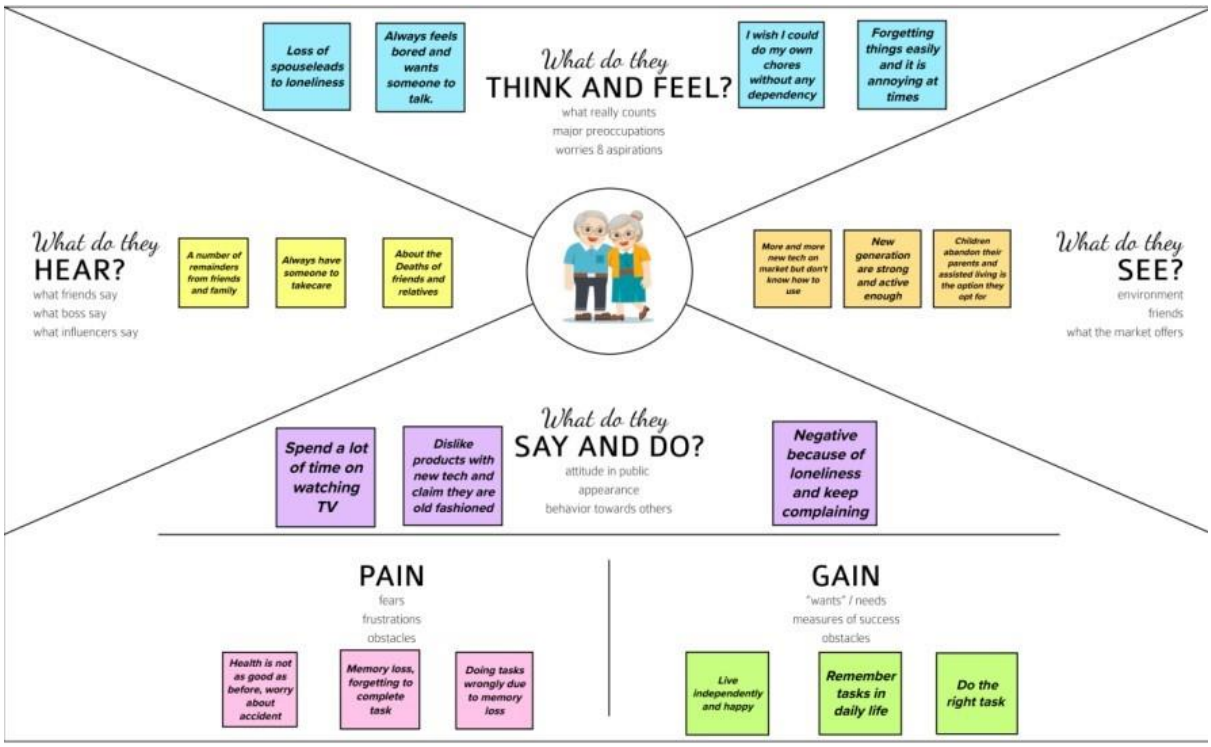
2.Cloud Computing based Medical Assistance & Pill Remainder; A.Chinnasamy; Ram Prasad J; Syed Rafeeq Ahmed; Akash S.

2.3 Problem Statement Definition

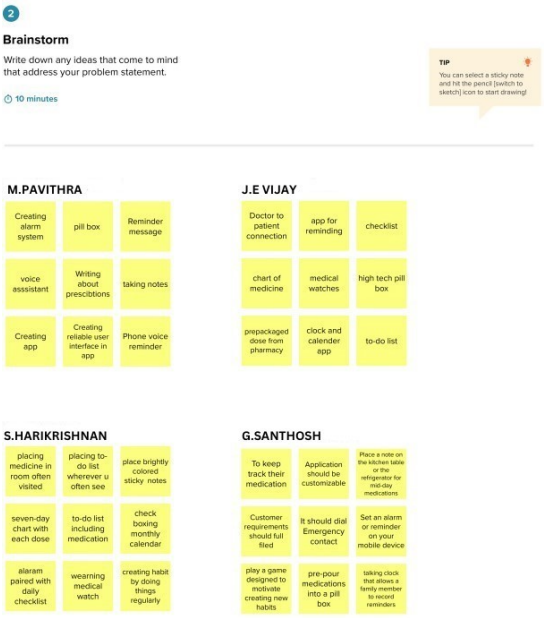
Skipping medicines can be serious for some health conditions. Sometimes elderly people forget to take their medicine at the correct time.They also forget which medicine one should take at that particular time and it is difficult for doctors/caretakers to monitor the patients around the clock.

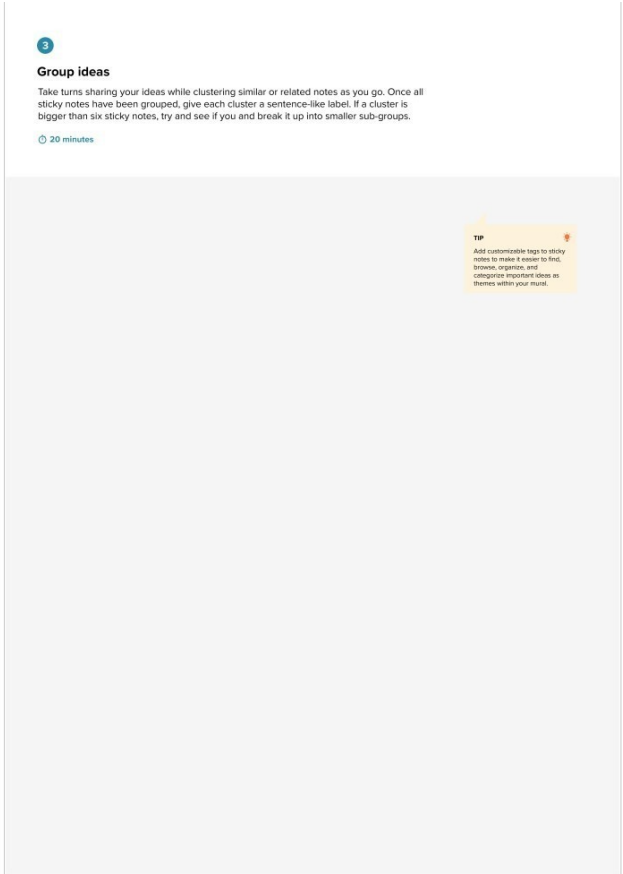
3.Ideation and Proposed Solution

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming

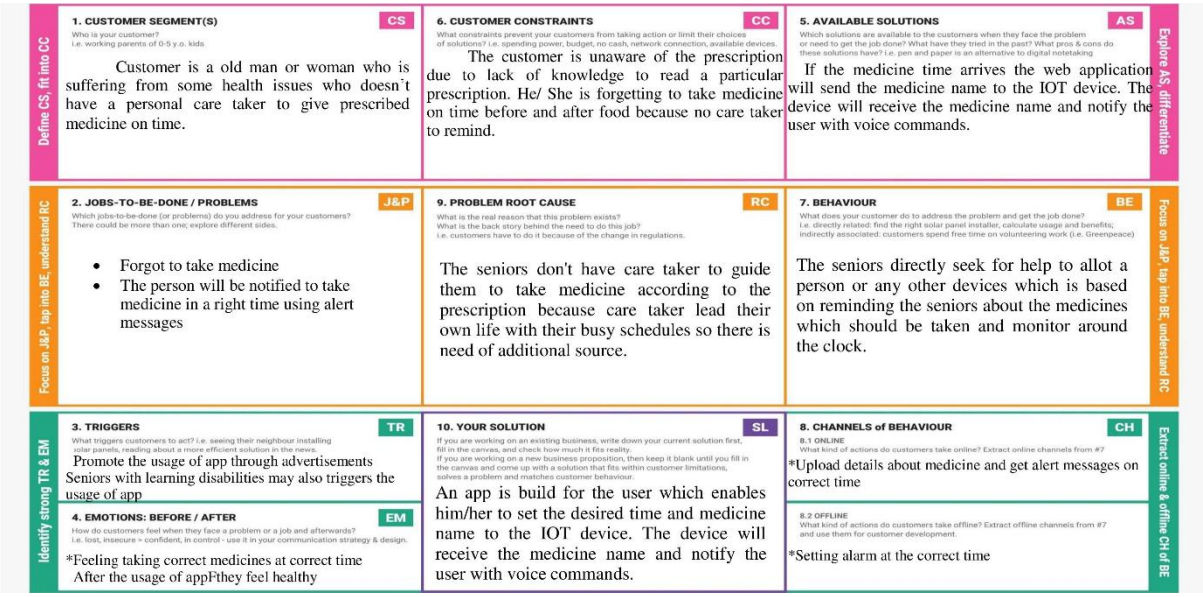




3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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 theirmedicine schedules, can easily discover pharmacies and clinics near them, and can be directed through the app by their loved One's if necessary.
3.	Novelty / Uniqueness	My research began with a series of inquiries directed at a variety of people in order to havea 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 persons ,keeping the user and their perspective in mind.
5.	Business Model (Revenue Model)	By using the model, we can collect basic and some medical information about the persona that helps us in showing relevant and profitable advertisements.
6.	Scalability of the Solution	As the model is integrated with cloud software, we can update the user experience without reinstalling a model and the persona can keep a Reminder up to year.

3.4 Problem Solution Fit



4.Requirement Analysis

4.1 Functional Requirements

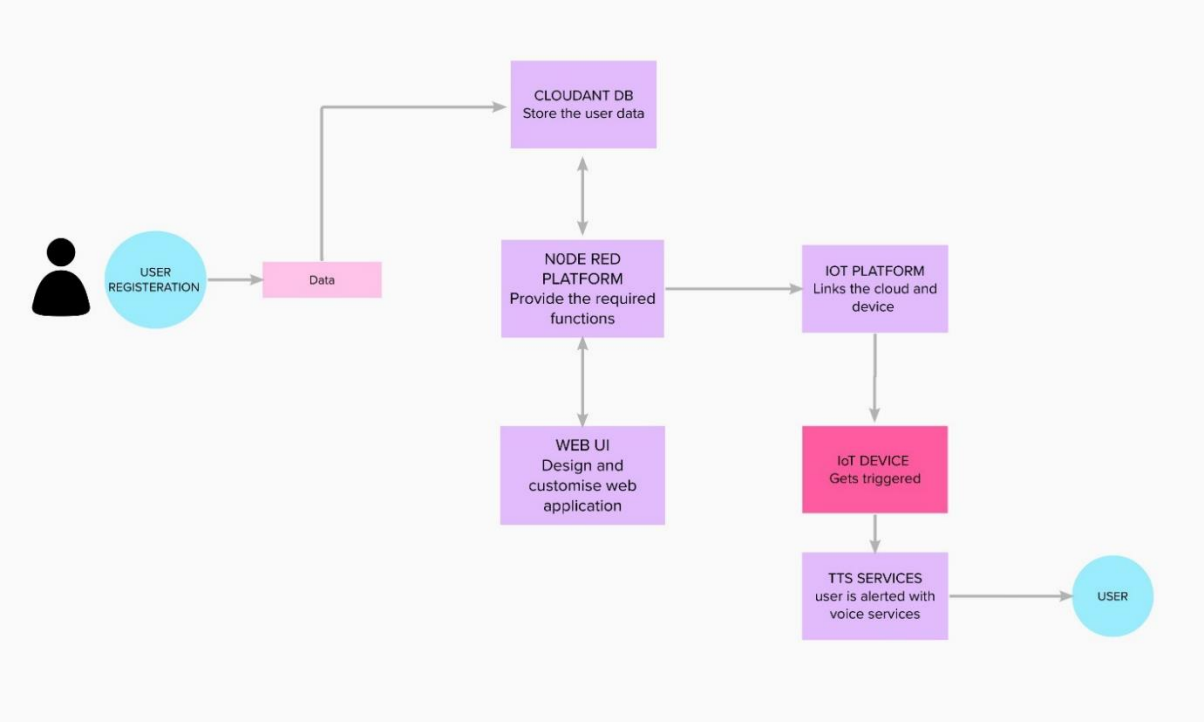
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration has been done through the form in our application.
FR-2	User Confirmation	Confirmation has been done within our application.
FR-3	Data management	All the data are stored in the cloud and retrieved when it is needed.
FR-4	Internet Connectivity	Users should have a stable internet connection to access the Application.
FR-5	User Input management	All the user's data are gotten with the help of a text field in the dashboard in the app.
FR-6	Acknowledgement	All the data are stored in the cloud via the app and acknowledgment will be given to the user.

4.2 Non-Functional Requirements

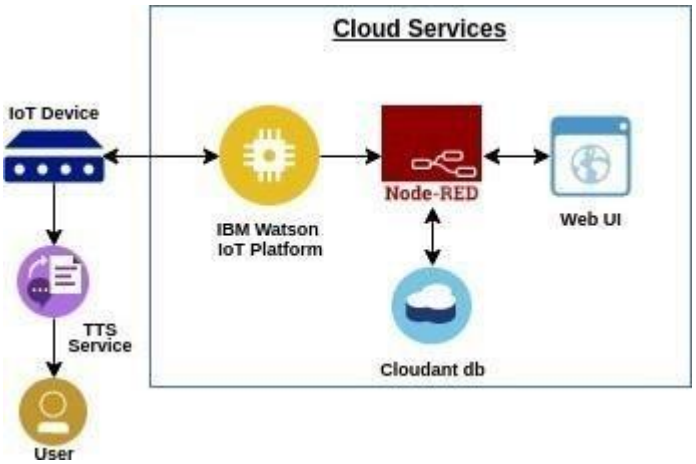
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Smart medicine box usability is the characteristics of the user that facilitate. Use, to make it easier for the user to perceive the information presented by the user interface, to understand and decide based on that information.
NFR-2	**Security**	Smart medicine box, like other computer systems, can be vulnerable to security breaches, potentially, impacting the safety and effectiveness of the device.
NFR-3	**Reliability**	The probability of medicine box will perform a required function without failure under stated conditions for a specific period.
NFR-4	**Performance**	Medical device testing is the process of demonstrating that the device will reliably and safely perform in use.
NFR-5	**Availability**	Medicine box is availability over all the conditions of weather and atmosphere pressure and be carried out with us.
NFR-6	**Scalability**	In feature we can upgrade the smart medicine box to the health care assistant to monitor our health care and book appointments to doctor.

5.Project Design

5.1 Data Flow Diagrams



5.2 Technical Architecture



5.3 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 my tablets on time by voice command	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 in database for upcoming week	My patient medication time and prescription should be in database list	low	Sprint-4

Customer (Disabled People's)	Smart medicine box	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

6.Project Planning and Scheduling

6.1 Sprint Planning and Estimation

Sprint	Functional Requirements(Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Buzzer (Set Alarm)	USN-1	As a user, I can set an alarm to alerting a medicine through medicine remainder system	10	High	Pavithra.M Vijay J.E Harikrishnan.S Santhosh.G
Sprint-1	Form	USN-2	As a user, I can Activate and Deactivate the alarm	10	High	Pavithra.M Vijay J.E Harikrishnan.S Santhosh.G
Sprint-2	Cloudant	USN-3	As a user once I can the set the alarm then I gets the notification	1	Low	Pavithra.M Vijay J.E Harikrishnan.S Santhosh.G
Sprint-2	Task	USN-4	Create All the Required Specifications (Node Red, IoT Device, Cloud, etc.)	19	High	Pavithra.M Vijay J.E Harikrishnan.S Santhosh.G
Sprint-3	Task	USN-5	Connecting the Node red, cloud, IoT	18	High	Pavithra.M Vijay J.E

			Device, Web UI, IoT Watson, TTS Service,			Harikrishnan.S Santhosh.G
Sprint-3	Cloud	USN-6	As a user, I can store the name of the medicine with its description	2	Medium	Pavithra.M Vijay J.E Harikrishnan.S Santhosh.G
Sprint-4	Device	USN-7	As a user, they can also help large hospitals and clinics manage their inventory more effectively	10	Low	Pavithra.M Vijay J.E Harikrishnan.S Santhosh.G
Sprint-4	Device, Cloud	USN-8	As a user, they used for keeping the record in medicine details the reminding the schedule of medicine. We have used the IoT enabled arduino device for monitoring the system	10	High	Pavithra.M Vijay J.E Harikrishnan.S Santhosh.G

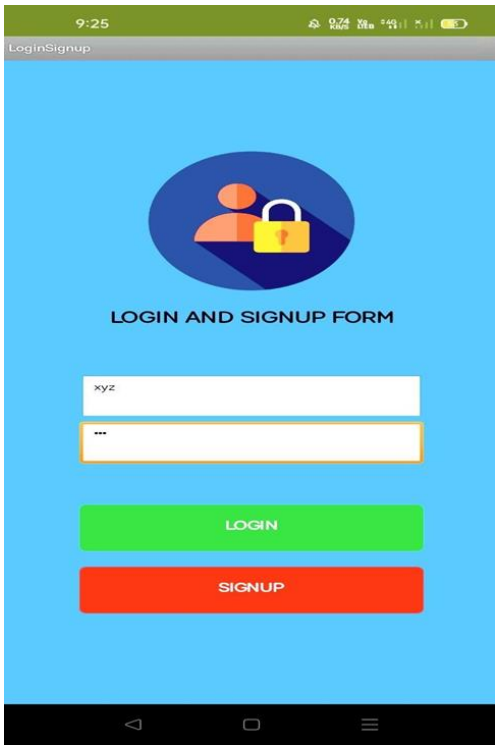
6.2 Sprint Delivery Schedule

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	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 Nov 2022	12 Nov 2022	20	13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	20 Nov 2022

7.Coding and Solutioning

7.1 Feature 1

The mobile application developed has a feature of individual login by different users.

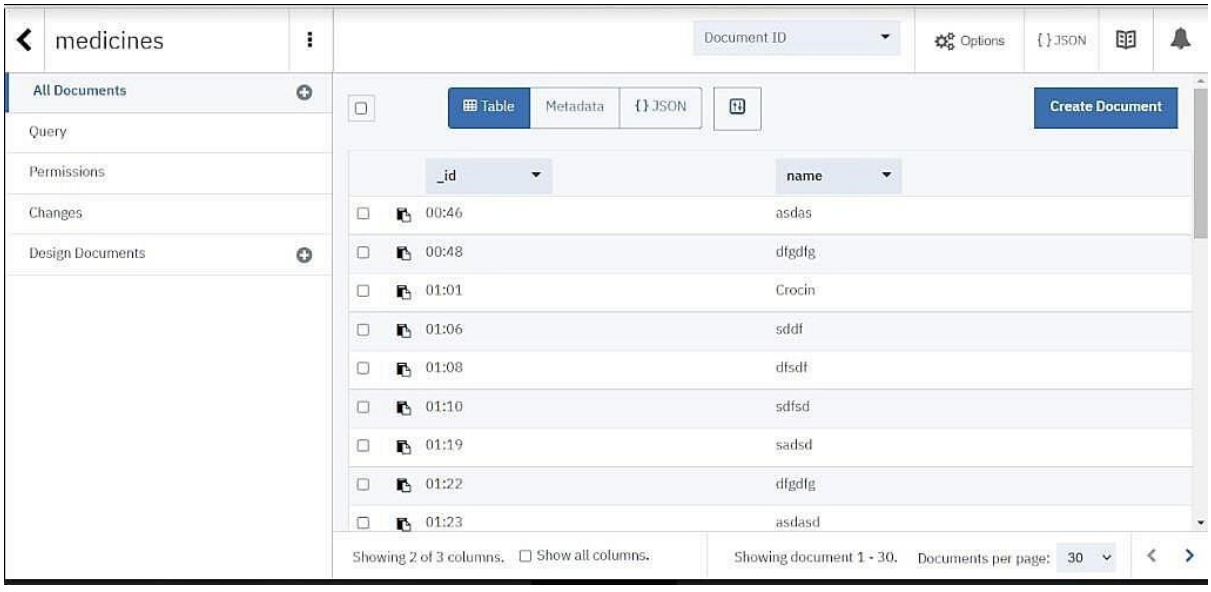


7.2 Feature 2

The mobile application also has the feature of uploading medicine names in the cloud.

7.3 Feature 3

The project includes a cloud database system.



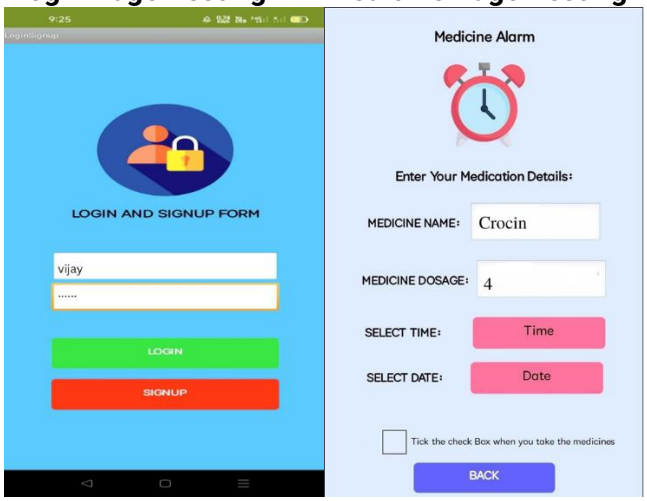
8.Testing

8.1 Test Cases

Test case	Precondition	Test steps	Test data	Expected result
Verify login with valid credentials	User should have a network connection	1. Launch URL 2. Enter valid username. 3. Enter valid password. 4. Click on the “Login” button.	Username: vijay Password: 1234@!	Users should be able to login successfully.
Verify login with invalid credentials	User should have a network connection	1. Launch URL 2. Enter valid username. 3. Enter invalid password. 4. Click on the “Login” button.	Username: vijay Password: 1234@!	Users should not be able to login.
Update the medicine name with the time.	User should have a network connection	1. Enter valid medicine name. 2. Enter the time when the medicine has to be consumed. 3. Click on the “Submit” button.	Medicine Name: Crocin Medicine Time: 20.00	Users should be able to update it successfully.

8.2 User Acceptance Testing

Login Page Testing **Medicine Page 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 and Disadvantages

Advantages

1.Help the elderly people to take their medicine at the correct time.
2.Avoid personal assistants or caretakers needed for medically sick people.
3.Cost efficient.
4.Can store multiple data and many notifications can be generated.

5.Since it includes voice assistance, even blind people can use our device.

Disadvantages

1.Makes people lethargic and makes them dependent always on others.
2.Requires a stable internet connection.

11.Conclusion

The project offers the elderly or medically sick people a personal assistant which reminds them of the medicines to be consumed at the particular time. Skipping tablets 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 applications, 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 project can be further developed by bringing into the feature of informing the medicine name during the notification.The voice assistants 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

Source code:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include <LiquidCrystal_I2C.h>
#define LED 2
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);


//-------credentials of IBM Accounts------

#define ORG "ok5c7o"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "ESP32"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "LC!x?+V9etumdVMaSR"     //Token
String data3="";


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
LiquidCrystal_I2C lcd(0x27,16,2);

//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
```

```
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential
void setup()// configureing the ESP32
{
  Serial.begin(115200);
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{
  if (!client.loop()) {
    mqttconnect();
  }
}

/*.....retrieving to Cloud.....*/


void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
```

```
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }

  Serial.println("Please take "+ data3);
  if(data3 != "")
  {
    lcd.init();
    lcd.print("Take"+ data3);

digitalWrite(LED,HIGH);
delay(20000);
digitalWrite(LED,LOW);
  }
  else
  {
digitalWrite(LED,LOW);

  }
data3="";
}
```

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-37511-1660310975
Project Demo Link: https://youtu.be/F-YXtcxXoKY