**Personal Assistance for Seniors who are Self-Reliant**

**A PROJECT REPORT**

Submitted By

**Saibalaji SM (711119104051)**

**Sathish B (711119104054)**

**Shakthivel S (711119104059)**

**Sridharan R (711119104064)**

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**

**JANSONS INSTITUTE OF TECHNOLOGY, COIMBATORE**

**ANNA UNIVERSITY: CHENNAI 600 025**

**Index**

1. **INTRODUCTION**

1.1.Project Overview .............................................................. 04

1.2.Purpose ............................................................................ 04

1. **LITERATURE SURVEY**

2.1. Existing problem ............................................................ 04

2.2. References ...................................................................... 04

2.3.Problem Statement Definition ........................................ 06

1. **IDEATION & PROPOSED SOLUTION**

3.1. Empathy Map Canvas ....................................................... 06

3.2.Ideation & Brainstorming ................................................. 07

3.3. Proposed Solution ............................................................ 08

3.4. Problem Solution fit ......................................................... 09

1. **REQUIREMENT ANALYSIS**

4.1.Functional requirement ..................................................... 10

4.2. Non-Functional requirements .......................................... 10

1. **PROJECT DESIGN**

5.1. Data Flow Diagrams ......................................................... 12

5.2. Solution & Technical Architecture ................................... 13

5.3. User Stories ...................................................................... 14

1. **PROJECT PLANNING & SCHEDULING**

6.1. Sprint Planning & Estimation ........................................... 15

6.2. Sprint Delivery Schedule .................................................. 18

6.3.Reports from JIRA ........................................................... 20

1. **CODING & SOLUTIONING**

7.1. Feature 1 ........................................................................... 25

7.2. Feature 2 ........................................................................... 26

7.3. Feature 3 .......................................................................... 27

7.4. Feature 4 ........................................................................... 28

1. **TESTING**

8.1. Test Cases ......................................................................... 28

8.2. User Acceptance Testing .................................................. 30

1. **RESULTS**

9.1. Performance Metrics ........................................................ 32

**10. ADVANTAGES & DISADVANTAGES** ............................ 32

**11.CONCLUSION** ...................................................................... 33

**12. FUTURE SCOPE** .................................................................. 33

**13. APPENDIX**

Source Code .......................................................................... 34

GitHub & Project Demo Link ............................................... 48

**1.INTRODUCTION**

**1.1. PROJECT OVERVIEW**

* 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.

**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.

**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 reminders 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**

1.A. Sawand, S. Djahel, Z. Zhang, and F. Na. Multidisciplinary Approaches to Achieving Efficient and Trustworthy e Health Monitoring Systems. Commun .China (ICCC), 2014 IEEE/CIC Int. Conf., pp. 187–192, 2014.

2. D. a. Clifton, D. Wong, L. Clifton, S. Wilson, R. Way, R. Pullinger, and L. Tarassenko. A large-scale clinical validation of an integrated monitoring system in the Emergency Department. IEEE J. Biomed. Heal. Informatics vol. 17, no. 4, pp. 835–842, 2013.

3. M. Parida, H.-C.Yang, S.-W.Jheng, and C.-J. Kuo.Application of RFID Technology for In-House Drug Management System.15th Int. Conf.Network-Based Inf. Syst., pp. 577–581, 2012.

4. L. Ilkko and J. Karppinen.UbiPILL A Medicine Dose Controller of Ubiquitous Home Environment. 2009 Third Int. Conf. Mob. UbiquitousComput.Syst. Serv. Technol., pp. 329–333, 2009.

5. A. Kliem, M. Hovestadt, and O. Kao.Security and Communication Architecture for Networked Medical Devices in Mobility-Aware e Health Environments,” 2012 IEEE First Int. Conf. Mob. Serv., pp. 112–114, 2012.

6. S. T.-B. Hamida, E. Ben Hamida, B. Ahmed, and A. AbuDayya.Towards efficient and secure in-home wearable insomnia monitoring and diagnosis system.13th IEEE Int. Conf. Bioinforma. Bioeng., pp. 1–6, 2013. 7. P. Ray.Home Health Hub Internet of Things (H 3 IoT): An architectural framework for monitoring health of elderly people.Sci. Eng. Manag.Res, pp. 3–5, 2014.

8. S. Huang, H. Chang, Y. Jhu, and G. Chen.The Intelligent Pill Box - Design and Implementation.pp. 235–236, 2014.

9. F.-T. Lin, Y.-C.Kuo, J.-C.Hsieh, H.-Y.Tsai, Y.-T. Liao, and H. C. Lee A Self-powering Wireless Environment Monitoring System Using Soil Energy. IEEE Sens. J., vol. 15, no. c, pp. 1–1, 2015.

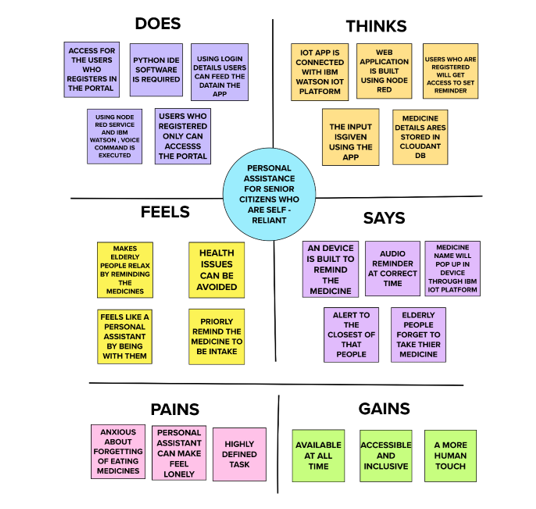
10. S. S. Al-majeed.HomeTelehealth by Internet of Things (IoT).pp. 609–613,

**2.3. PROBLEM STATEMENT DEFINITION**

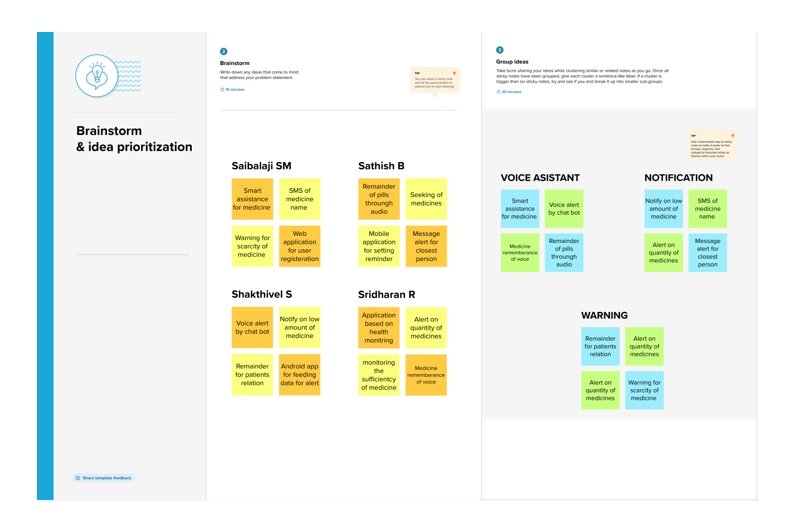
It is very difficult for the senior citizens (elder people) to remember their medicines. To avoid the skipping up the medicines,they can be remembered by using the voice commands of the medicine names at correct time specified. If the voice commands on the medicine name is not available, they are given the reminder of the medicine by SMS on their phone or to their closest person.

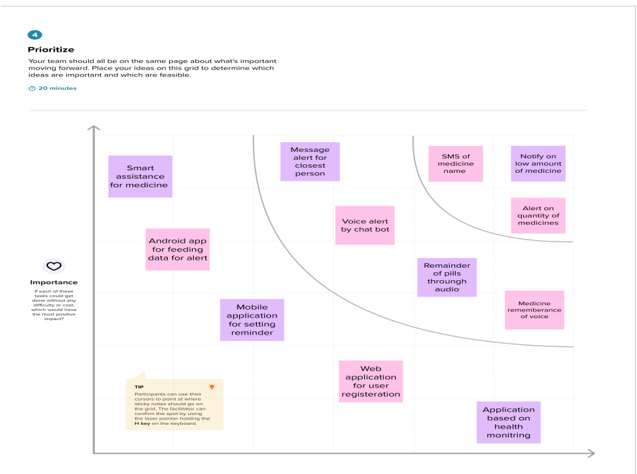
**3. IDEATION & PROPOSED SOLUTION**

**3.1. EMPATHY MAP CANVAS**



**3.2. IDEATION AND BRAINSTORMING**

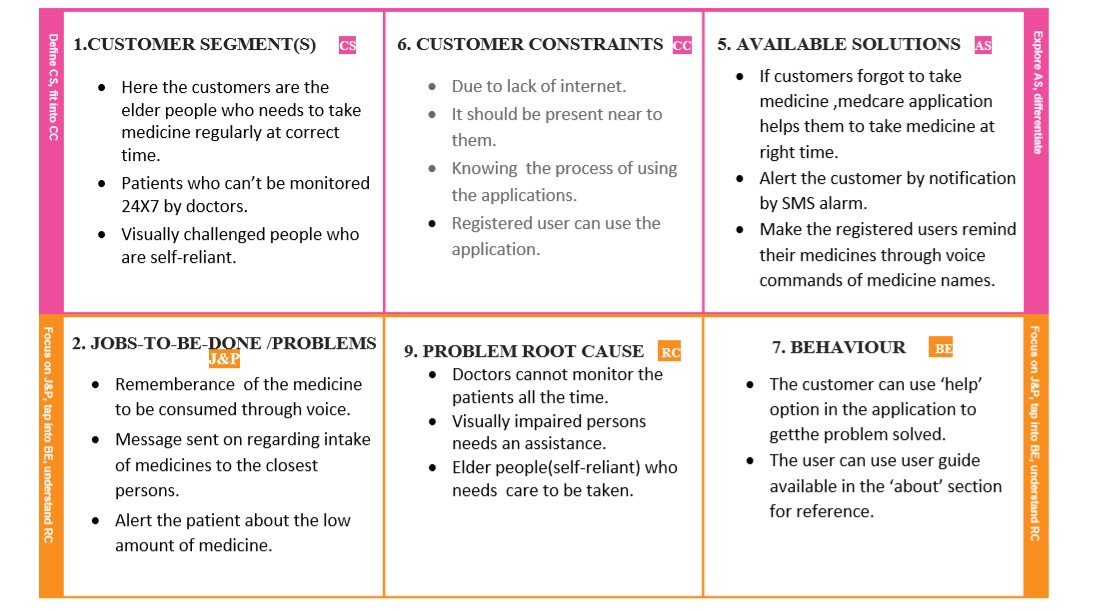


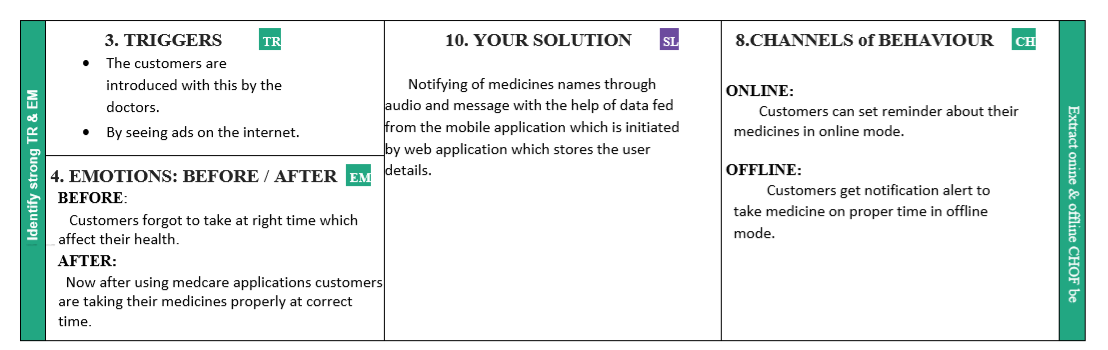


**3.3. PROPOSED SOLUTION**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Parameter** | **Description** |
| 1 | Problem Statement (Problem to be solved) | Senior citizens who are in need of medicine reminder and self-assistance because they don't want to skip their intake of medicine |
| 2 | Idea / Solution description | Creation of the web application which remind the medicine name and time through a voice alert |
| 3 | Novelty / Uniqueness | Blind people can get to know their time of taking pills |
| 4 | Social Impact / Customer Satisfaction | The users are satisfied with the proper reminder and intake of pills |
| 5 | Business Model (Revenue Model) | By our web application the revenue can be made in the form of popping up of advertisements or by overlaying add from third party services |
| 6 | Scalability of the Solution | Vast number of people who are aged can be provided with portable devices to ensure their health conditions by consuming medicines at correct time using web application |

**3.4. PROBLEM SOLUTION FIT**





**4. REQUIREMENT ANALYSIS**

**4.1. FUNCTIONAL REQUIREMENTS**

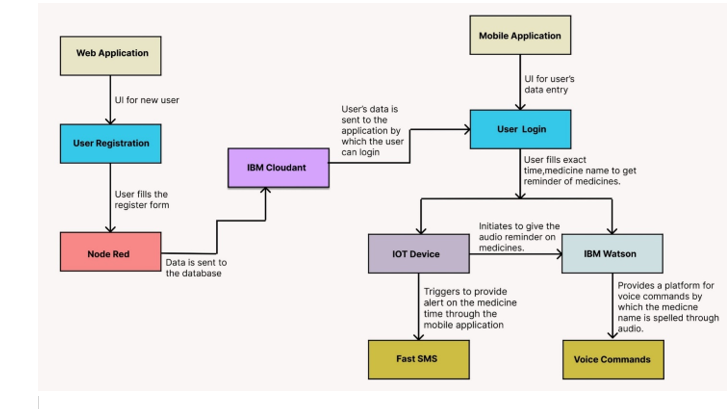
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Gmail  Registration by phone number |
| FR-2 | User Confirmation | Confirmation via Email  Confirmation throughSMS/Messages |
| FR-3 | User Login(Web) | Login withregistered mail id and password |
| FR-4 | User Login(mobile app) | Login withregistered mobile numberand password |
| FR-5 | User’s Medical Information | In the app, enter your medicine details with date. Thenset the time in the app. |

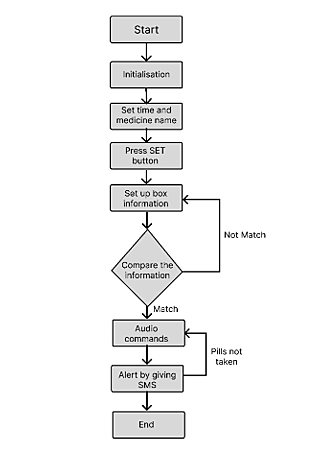
**4.2. NON-FUNCTIONAL REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | The systemshould be user-friendly for the users.It is used to remaind the medicine names.  It alertsthe users through voice commands. |
| NFR-2 | **Security** | The logininformation should notbe accessed by anyother usersthan the respective.  The dataof the usersshould be keptconfidential. |
| NFR-3 | **Reliability** | 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 giventime.  It workswithout any connection interruption |
| NFR-5 | **Availability** | The systemshould be monitored 24X7 for the alertof medicines.  It canbe used by any registered users from anyplace. |
| NFR-6 | **Scalability** | It is easily adaptable  The deviceis compatible and portable  The application can handle any number of registration. |

**5. PROJECT DESIGN**

**5.1. DATA FLOW DIAGRAMS**





**5.2. SOLUTION & TECHNICAL ARCHITECTURE**

**IOT Device:**

* Getting the information from the application about the time and name of the medicines.
* Sending an SMS to the persons.
* Gathering the user information from the web application in which the user registers.

To accomplish this, we have to complete all the activities listed below:

**Create and Configure IBM Cloud Services:**

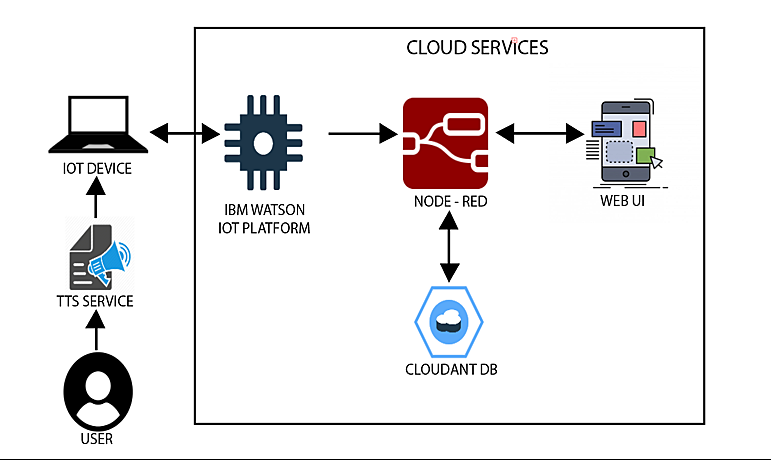
* Create IBM Watson IOT platform
* Create a device & configure the IBM IOT Platform
* Create Node-Red service
* Create a database in IBM Cloudant DB to medicine names and time.

**Develop a web application using Node-RED service:**

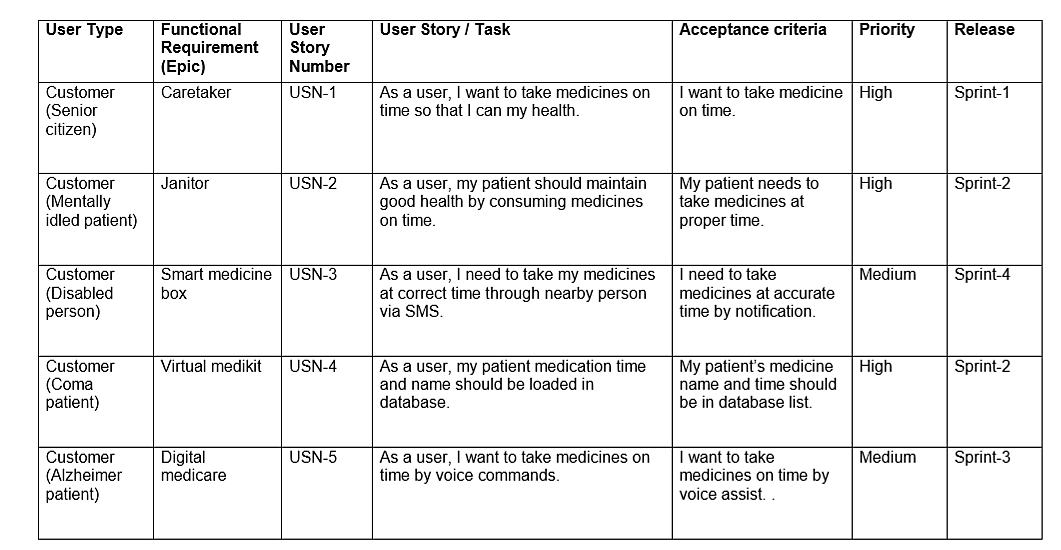
* Develop the web application using Node-RED.
* Develop a python script to publish the medicine names and time to remind details to the IBM IOT Platform.

**Develop an application:**

* Develop an application in which the user can feed the data on the medicine name and time.
* Develop an application which can transmit the signal on the reminder of the medicines at the time specified.



**5.3. USER STORIES**



**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 | Login | USN-1 | As a admin, I can log into the application by entering username & password | 5 | Medium | Sathish B |
| Sprint-1 |  | USN-2 | When the admin doesn’t enter the username it displays an error message group | 3 | Medium | Sathish B |
| Sprint-1 |  | USN-3 | When the admin doesn’t enter the password it displays an error message popup | 4 | Medium | Sathish B |
| Sprint-1 |  | USN-4 | When the admin enters the invalid credentials it displays an error popup | 5 | Medium | Shakthivel S |
| Sprint-1 |  | USN-5 | When the admin enter the correct username and password it redirects to the dashboard | 3 | High | Shakthivel S |
| Sprint-2 | Dashboard | USN-1 | Creating a Node-Red dashboard | 5 | Medium | Saibalaji SM |
| Sprint-2 |  | USN-2 | Devoloping a Node-Red to publish data to IBM cloud | 8 | High | Saibalaji SM |
| Sprint-2 |  | USN-3 | Create a register form in Node-Red | 7 | Medium | Saibalaji SM |
| Sprint-3 | Creating device | USN-1 | Creating a device in IBM Watson IOT platform | 10 | High | Shakthivel S |
| Sprint-3 | Python | USN-2 | Connect the device created in wokwi to the device created in IBM Watson IOT platform. | 10 | High | Saibalaji SM |
| Sprint-4 | MIT app inventor | USN-1 | Create a Interface for login page and Dashboard | 5 | Low | Sridharan R |
| Sprint-4 |  | USN-2 | Connect MIT app to Node Red | 5 | High | Sridharan R |
| Sprint-4 |  | USN-3 | As a user, I can keep track of the medicine time | 6 | Medium | Sridharan R |
| Sprint-4 | Alert | USN-4 | Retrieving the time from cloudant and alert the user through voice command | 4 | High | Sridharan R |

**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 | 4 Days | 31 Oct 2022 | 3 Nov 2022 | 20 | 2 Nov 2022 |
| Sprint-2 | 20 | 5 Days | 04 Nov2022 | 8 Nov 2022 | 20 | 8 Nov 2022 |
| Sprint-3 | 20 | 5 Days | 09 Nov 2022 | 13 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 4 Days | 14 Nov 2022 | 17 Nov 2022 | 20 | 18 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 = Sprint duration / Velocity

= 20 / 18

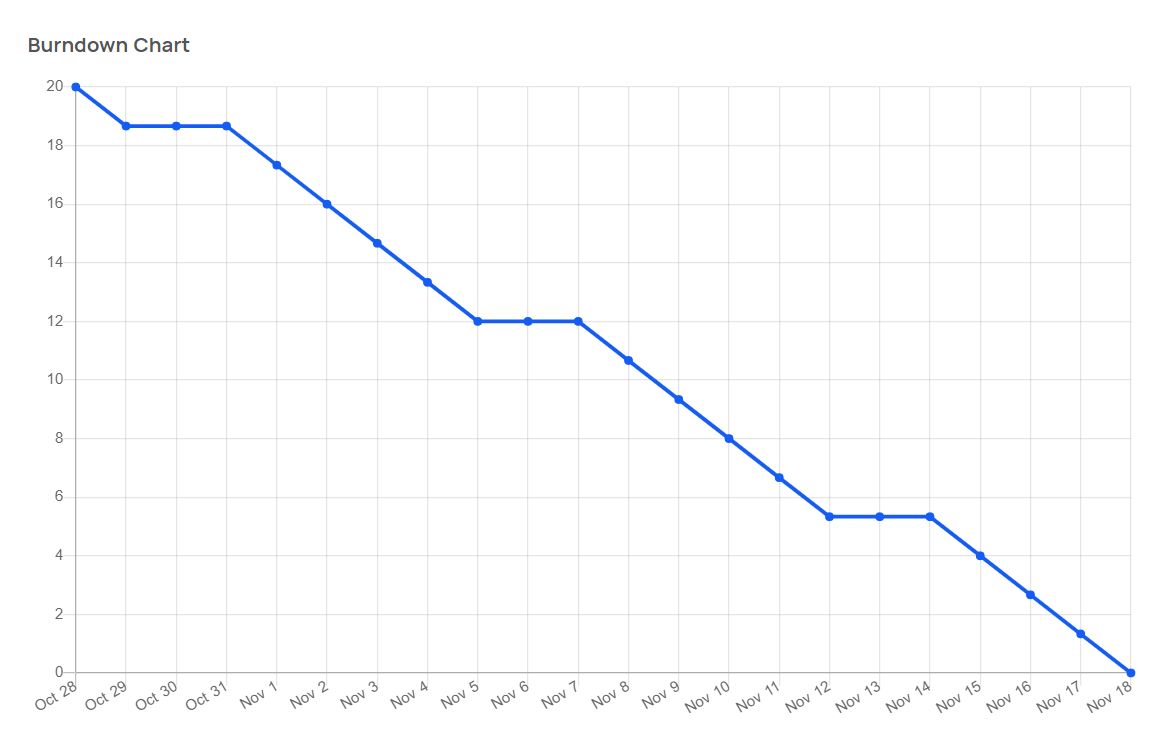
AV= 1.11

**Burndown Chart:**

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

<https://www.visual-paradigm.com/scrum/scrum-burndown-chart/>

<https://www.atlassian.com/agile/tutorials/burndown-charts>



**6.3. REPORTS FROM JIRA**

**PAFSWASR-1:**

|  |  |  |  |
| --- | --- | --- | --- |
| *[PAFSWASR-1]* [*Login page*](https://saibalaji05.atlassian.net/browse/PAFSWASR-5) *Created: 13/Nov/22 Updated: 13/Nov/22 Resolved: 13/Nov/22* | | | |
| **Status:** | Done | | |
| **Project:** | [Personal assistance for seniors wo are self-reliant](https://saibalaji05.atlassian.net/secure/BrowseProject.jspa?id=10000) | | |
| **Components:** | HTML,CSS,Javascript | | |
| **Affects versions:** | 5.0 | | |
| **Fix versions:** | 5.0 | | |
| **Type:** | Task | **Priority:** | Medium |
| **Reporter:** | [Saibalaji Sm](https://saibalaji05.atlassian.net/secure/ViewProfile.jspa?accountId=636e0946847c699ac65253f1) | **Assignee:** | [shakthivel2308](https://saibalaji05.atlassian.net/secure/ViewProfile.jspa?accountId=636e10ead7c060fdaa5c1398) |
| **Resolution:** | Done | **Votes:** | 0 |
| **Labels:** | None | | |
| **Remaining Estimate:** | 3 hours | | |
| **Time Spent:** | 21 hours | | |
| **Original estimate:** | 1 days | | |
| **Rank:** | 1 | | |
| **Sprint:** | Sprint 1 | | |

Generated at Sun Nov 13 14:17:39 UTC 2022 by Saibalaji Sm using Jira 1001.0.0-SNAPSHOT#100210-sha1:583150f45e96fe66b2cb2898eb1e9ae5719d8732.

**PAFSWASR-2:**

|  |  |  |  |
| --- | --- | --- | --- |
| *[PAFSWASR-2]* [*create a node red dashboard*](https://saibalaji05.atlassian.net/browse/PAFSWASR-9) *Created: 13/Nov/22 Updated: 13/Nov/22 Resolved: 13/Nov/22* | | | |
| **Status:** | Done | | |
| **Project:** | [Personal assistance for seniors wo are self-reliant](https://saibalaji05.atlassian.net/secure/BrowseProject.jspa?id=10000) | | |
| **Type:** | Task | **Priority:** | Medium |
| **Reporter:** | [Saibalaji Sm](https://saibalaji05.atlassian.net/secure/ViewProfile.jspa?accountId=636e0946847c699ac65253f1) | **Assignee:** | [Saibalaji Sm](https://saibalaji05.atlassian.net/secure/ViewProfile.jspa?accountId=636e0946847c699ac65253f1) |
| **Resolution:** | Done | **Votes:** | 0 |
| **Labels:** | None | | |
| **Remaining Estimate:** | 5 hours | | |
| **Time Spent:** | 28 hours | | |
| **Original estimate:** | 2 days | | |
| **Rank:** | 2 | | |
| **Sprint:** | Sprint 2 | | |

Generated at Sun Nov 13 15:53:00 UTC 2022 by Saibalaji Sm using Jira 1001.0.0-SNAPSHOT#100210-sha1:583150f45e96fe66b2cb2898eb1e9ae5719d8732.

**PAFSWASR-3:**

|  |  |  |  |
| --- | --- | --- | --- |
| *[PAFSWASR-3]* [*Create an app in MIT App Inventor for entering the details*](https://saibalaji05.atlassian.net/browse/PAFSWASR-11)*Created: 18/Nov/22 Updated: 18/Nov/22* | | | |
| **Status:** | Done | | |
| **Project:** | [Personal assistance for seniors wo are self-reliant](https://saibalaji05.atlassian.net/secure/BrowseProject.jspa?id=10000) | | |
| **Components:** | MIT App Inventor | | |
| **Affects versions:** | None | | |
| **Fix versions:** | None | | |
| **Type:** | Task | **Priority:** | Medium |
| **Reporter:** | [Saibalaji Sm](https://saibalaji05.atlassian.net/secure/ViewProfile.jspa?accountId=636e0946847c699ac65253f1) | **Assignee:** | [Sridharan R](https://saibalaji05.atlassian.net/secure/ViewProfile.jspa?accountId=636b8518c9b26a8d857c877a) |
| **Resolution:** | Done | **Votes:** | 0 |
| **Labels:** | None | | |
| **Remaining Estimate:** | 4 hours | | |
| **Time Spent:** | 15 hours | | |
| **Original estimate:** | 1 day | | |
| **Rank:** | 2 | | |
| **Sprint:** | Sprint-3 | | |

Generated at Fri Nov 18 18:26:22 UTC 2022 by Saibalaji Sm using Jira 1001.0.0-SNAPSHOT#100210-sha1:9b34d7cc56ccedf37042f403595483f2079121f4.

**PAFSWASR-4:**

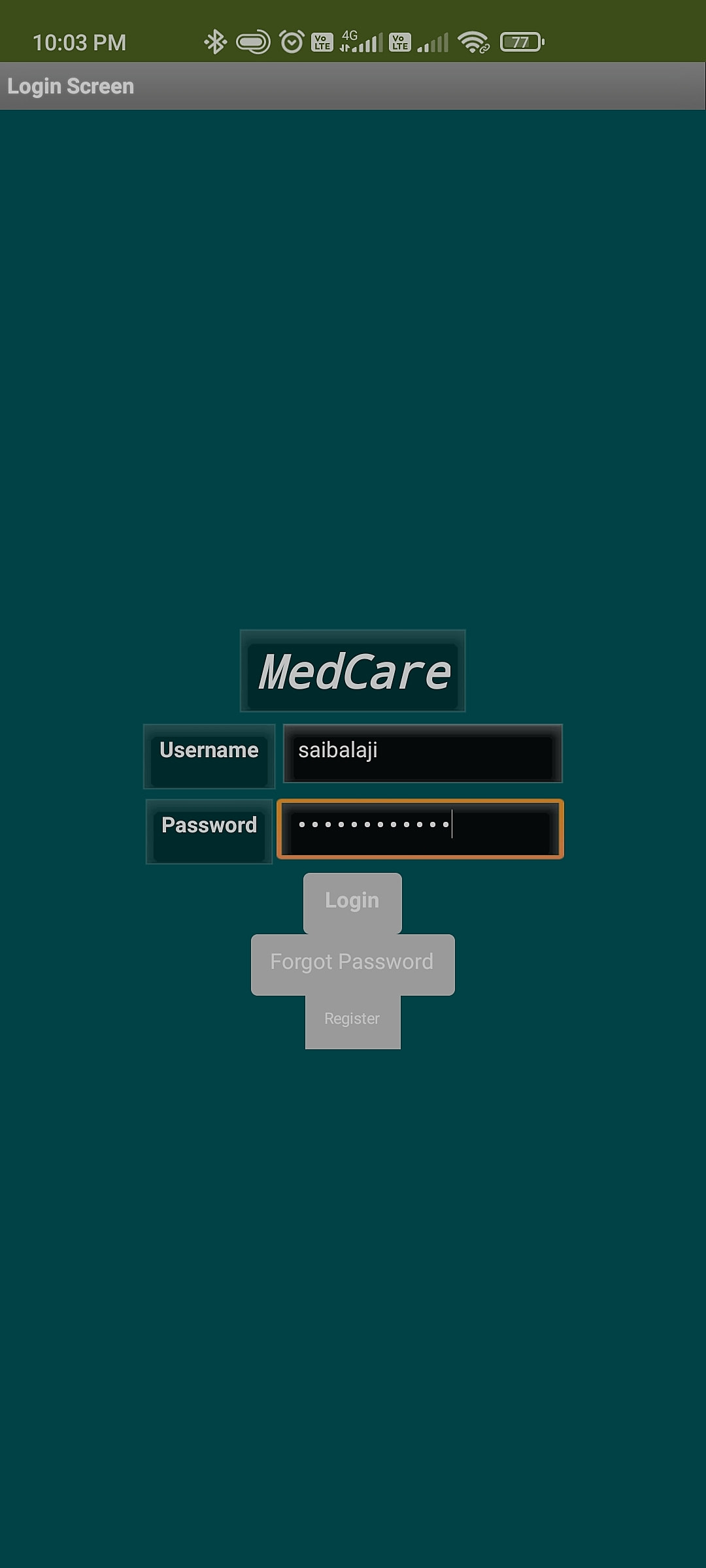
|  |  |  |  |
| --- | --- | --- | --- |
| *[PAFSWASR-4]* [*Simulation of device for medicine remainder*](https://saibalaji05.atlassian.net/browse/PAFSWASR-12) *Created: 18/Nov/22 Updated: 18/Nov/22* | | | |
| **Status:** | Done | | |
| **Project:** | [Personal assistance for seniors wo are self-reliant](https://saibalaji05.atlassian.net/secure/BrowseProject.jspa?id=10000) | | |
| **Components:** | Wokwi Simulator | | |
| **Affects versions:** | None | | |
| **Fix versions:** | None | | |
| **Type:** | Task | **Priority:** | Medium |
| **Reporter:** | [Saibalaji Sm](https://saibalaji05.atlassian.net/secure/ViewProfile.jspa?accountId=636e0946847c699ac65253f1) | **Assignee:** | [balamurugansam45](https://saibalaji05.atlassian.net/secure/ViewProfile.jspa?accountId=636e10ea3cbe3dde78b60c38) |
| **Resolution:** | Done | **Votes:** | 0 |
| **Labels:** | None | | |
| **Remaining Estimate:** | 2 hours | | |
| **Time Spent:** | 20 hours | | |
| **Original estimate:** | 22 hours | | |
| **Attachments:** | Sprint-4.pdf | | |
| **Rank:** | 1 | | |
| **Sprint:** | Sprint-4 | | |

Generated at Fri Nov 18 18:36:52 UTC 2022 by Saibalaji Sm using Jira 1001.0.0-SNAPSHOT#100210-sha1:9b34d7cc56ccedf37042f403595483f2079121f4.

**7. CODING & SOLUTIONING**

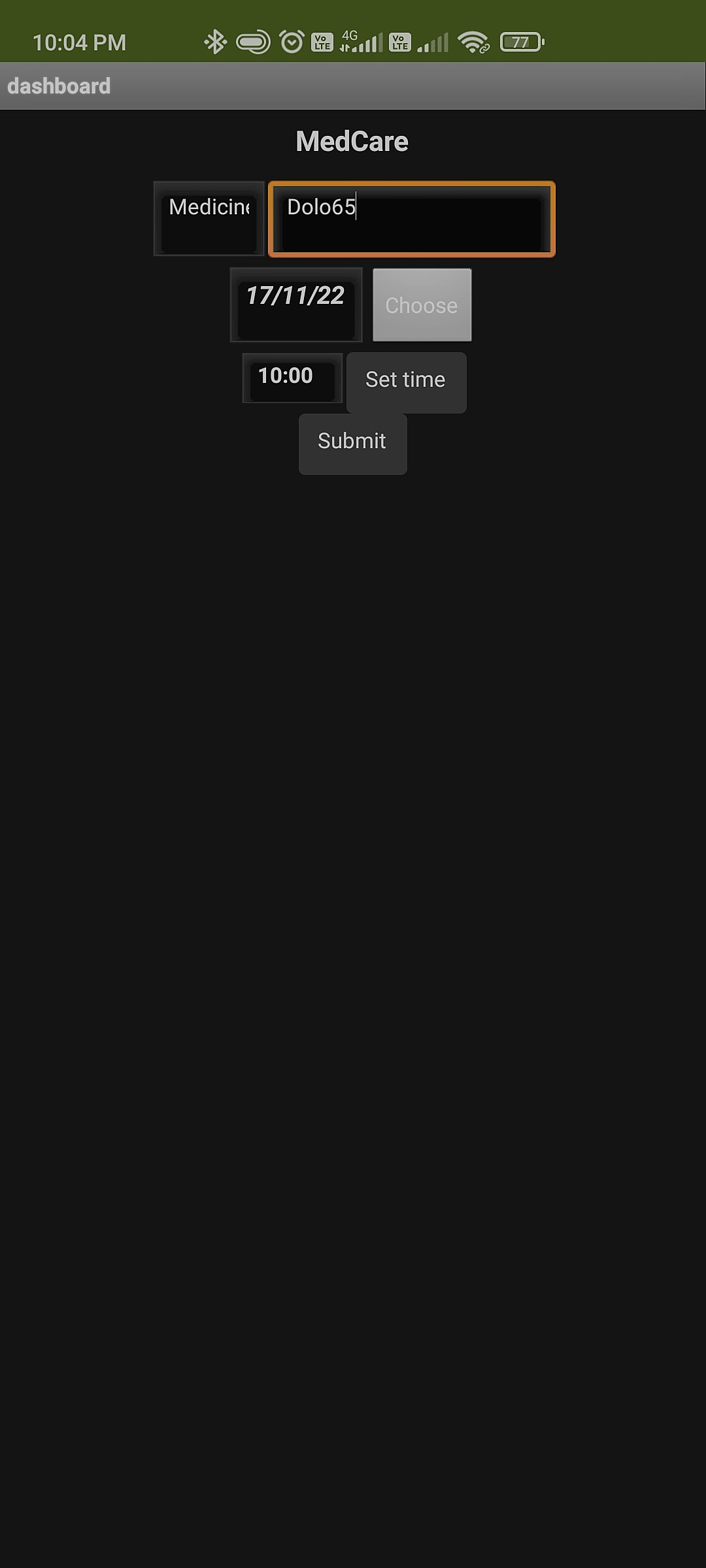
**7.1. Feature 1**

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.

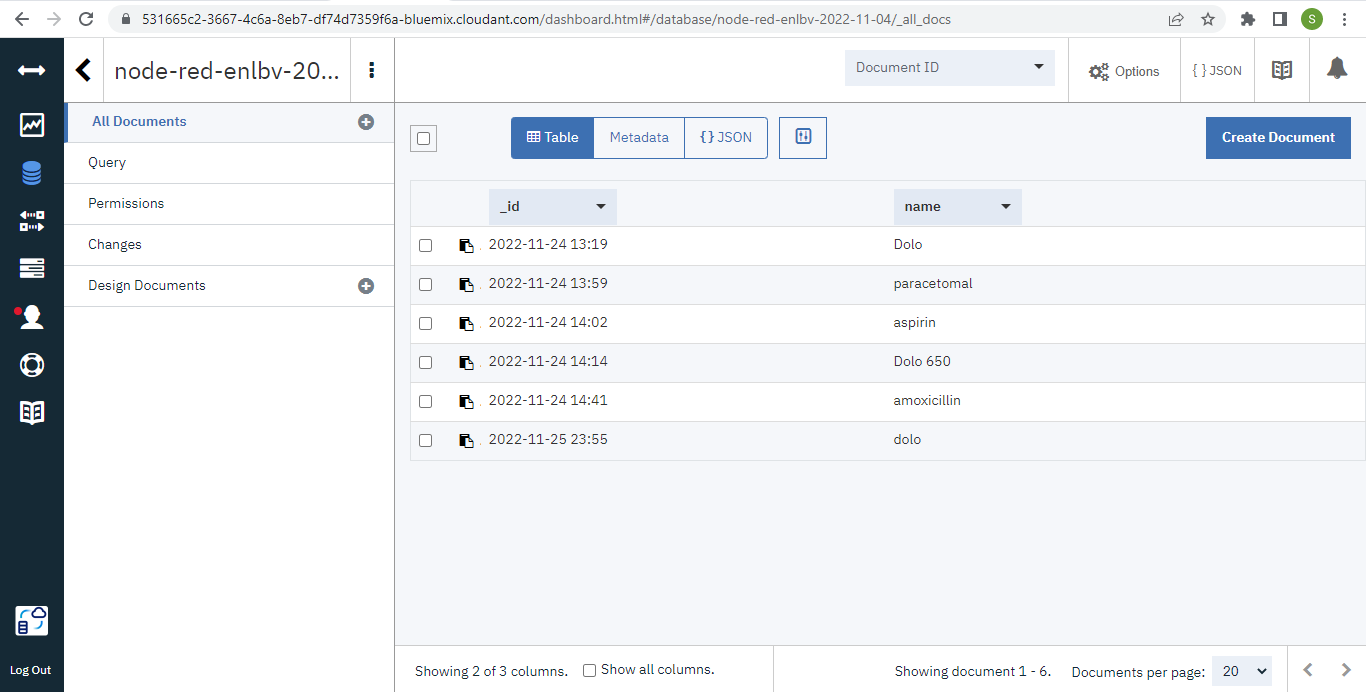
**7.3. Feature 3**

The mobile application also has the feature of registering username in the database and forgot password feature.

**7.4. Feature 4**

The project includes a cloud database system.



**8.TESTING**

**8.1. TEST CASES**

A test case is a document which has a set of conditions or actions that are performed on thesoftware application in order to verify the expected functionality of the feature.After test scripts, test cases are the second most detailed way of documenting testing work.They describe a specific idea that is to be tested, without detailing the exact steps to be taken or data to be used. For example, in a test case, you document something like ‘Test if coupons can be applied on actual price‘. This doesn’t mention how to apply the coupons or whether there are multiple ways to apply. It also doesn’t mention if the tester uses alink to apply adiscount, or enter a code, or have a customer service apply it. They give flexibility to thetester to decide how they want to execute the test.

**Test Case Format**

The primary ingredients of a test case are an ID, description, bunch of inputs, few actionablesteps, as well as expected and actual results. Let’s learn what each of them is:

* **Test Case Name**:A test case should have a name or title that is self explanatory.
* **Test Case Description:**The description should tell the tester what they’re going to test in brief.
* **PreConditions:**Any assumptions that apply to the test and any preconditionsthat must be met prior to the test being executed should be listed here.
* **Test Case Steps:**The test steps should include the necessary data and information on how to execute the test. The steps should be clear and brief, without leaving out essential facts.
* **Test Data:**It’s important to select a data set that gives sufficient coverage.Select a data set that specifies not only the positive scenarios but negative ones as well.
* **Expected Result**:The expected results tell the tester what they should experience as a result of the test steps.
* **Actual Result:**They specifies how the application actually behaved while test cases were being executed.
* **Comments:**Any other useful information such as screenshots that tester want’s to specify can be included here.

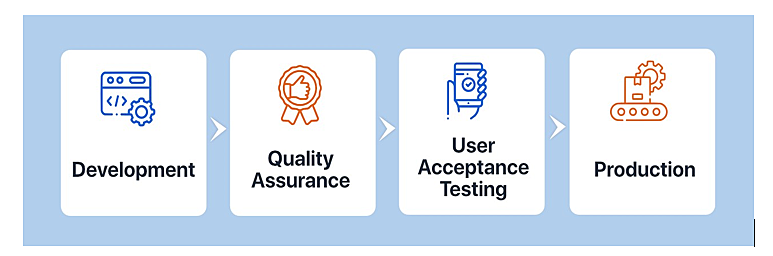
**8.2. USER ACCEPTANCE TESTING**

**1.Purpose of Document**

The main Purpose of UAT is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is kind of black box testing where two or more end-users will be involved.

UAT is performed by :

* Client
* End users



# 2.Defect Analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | **Severity 1** | **Severity 2** | **Severity 3** | **Severity 4** | **Subtotal** |
| By Design | 4 | 3 | 2 | 1 | 10 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| External | 2 | 2 | 1 | 1 | 6 |
| Fixed | 4 | 3 | 5 | 19 | 31 |
| Not Reproduced | 1 | 0 | 1 | 1 | 3 |
| Skipped | 0 | 0 | 1 | 1 | 2 |
| Won't Fix | 1 | 3 | 2 | 2 | 8 |
| Totals | 13 | 11 | 15 | 25 | 64 |

# 

# 

# 3.Test Case Analysis:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **TotalCases** | **Not Tested** | **Fail** | **Pass** |
| Login Page | 5 | 0 | 0 | 5 |
| Node Red Dashboard | 32 | 0 | 0 | 32 |
| IBM Watson IOT platform | 2 | 0 | 0 | 2 |
| MIT App Inventor | 3 | 0 | 0 | 3 |

**9. RESULTS**

**9.1. PERFORMANCE METRICS**

These metrics are used to track and measure the effectiveness and profitability of various projects. Each stage of the project is tracked and measured against the goals that the project set out to achieve. The data compiled from the metrics can be used to plan future projects and gives insight on how to make projects more efficient.

**10.ADVANTAGES & DISADVANTAGES**

**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.

**Disadvantages**

➢ Makes people lethargic and makes them dependent always on others.

➢ 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 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 project 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**

**Source Code**

**Device Simulation:**

#include <WiFi.h>//library for wifi

#include <PubSubClient.h>//library for MQtt

#include <LiquidCrystal\_I2C.h>

#include "DHT.h"// Library for dht11

#define DHTPIN 15 // what pin we're connected to

#define DHTTYPE DHT11 // define type of sensor DHT 11

#define LED 2

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht connected

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

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

#define ORG "kizp10"//IBM ORGANITION ID

#define DEVICE\_TYPE "IOTdevice"//Device type mentioned in ibm watson IOT Platform

#define DEVICE\_ID "1234567890"//Device ID mentioned in ibm watson IOT Platform

#define TOKEN "1234567890" //Token

String data3="";

int buzz= 13;

//-------- 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 your medicines”);

if(data3 != "")

{

lcd.init();

lcd.print("Its time for your medicine”);

digitalWrite(LED,HIGH);

delay(20000);

digitalWrite(LED,LOW);

}

else

{

digitalWrite(LED,LOW);

}

data3="";

}

**Database connection:**

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "kizp10"

deviceType = "IOTdevice"

deviceId = "1234567890"

authMethod = "token"

authToken = "1234567890"

# Initialize GPIO

def myCommandCallback(cmd):

print("Command received: %s" % cmd.data['command'])

status=cmd.data['command']

if status=="lighton":

print ("led is on")

elif status == "lightoff":

print ("led is off")

else :

print ("please send proper command")

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

temp=random.randint(90,110)

Humid=random.randint(60,100)

data = { 'temp' : temp, 'Humid': Humid }

#print data

def myOnPublishCallback():

print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "to IBM Watson")

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF")

time.sleep(10)

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()

**Text-to-Speech:**

from ibm\_watson import TextToSpeechV1

from ibm\_cloud\_sdk\_core.authenticators import IAMAuthenticator

authenticator = IAMAuthenticator('KSTdsMPsUS62SL58EqzaZbAFtEW2JlggKYHUI-NKLuvx') text\_to\_speech = TextToSpeechV1(

authenticator=authenticator

)

text\_to\_speech.set\_service\_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/10758658-1ffd-49e5-ae59-ffb2aaa3b131')

with open('Medicine.wav', 'wb') as audio\_file:

audio\_file.write(

text\_to\_speech.synthesize(

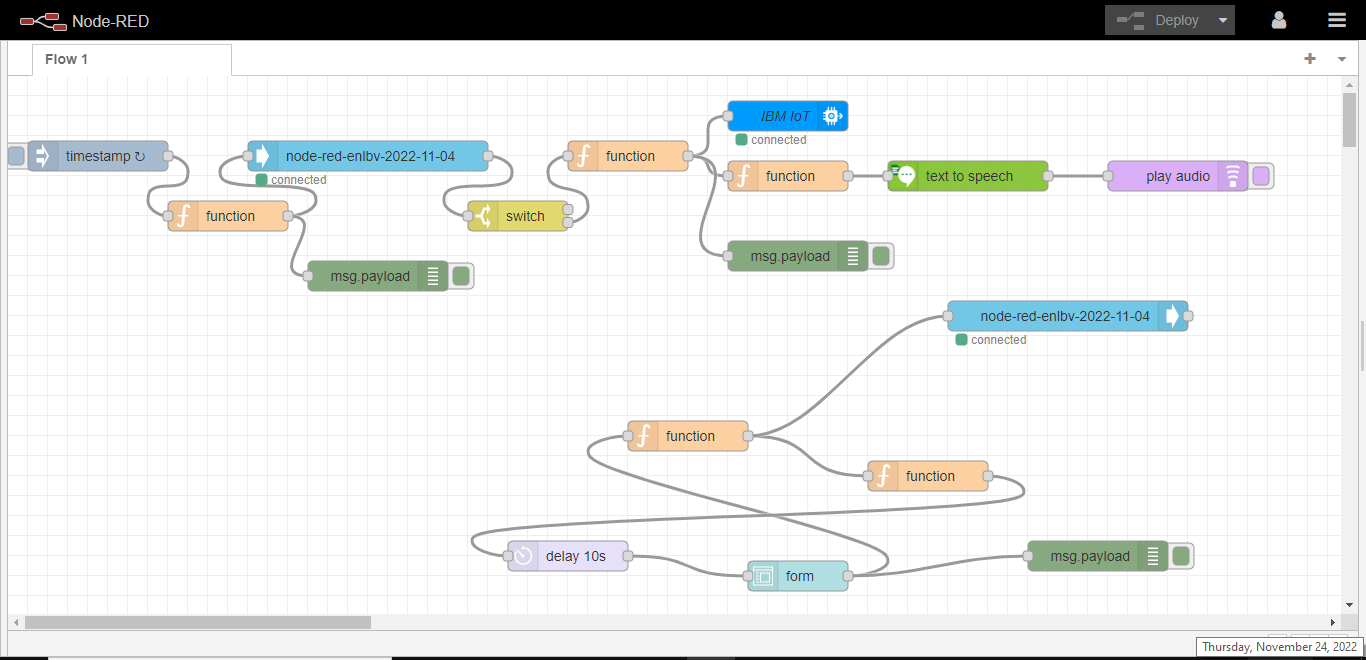
'Its time for your medicine',

voice='en-US\_AllisonV3Voice',

accept='audio/wav'

).get\_result().content)

**Node-red Flows:**



**Flows.json:**

[{"id":"b3626964d3ca3efc","type":"tab","label":"Flow 1","disabled":false,"info":"","env":[]},

{"id":"c6377dfe10ccb6c5","type":"inject","z":"b3626964d3ca3efc","name":"","props":[{"p":"payload"},{"p":"topic","vt":"str"}],"repeat":"10","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":90,"y":80,"wires":[["0ed2fb66b7d4a708"]]},

{"id":"0ed2fb66b7d4a708","type":"function","z":"b3626964d3ca3efc","name":"","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":220,"y":140,"wires":[["967607fc0feb4a2a","3bd136d585ee16b8"]]},

{"id":"967607fc0feb4a2a","type":"cloudantplus in","z":"b3626964d3ca3efc","name":"","cloudant":"e8e674ee5e0cb282","database":"node-red-enlbv-2022-11-04","service":"\_ext\_","search":"\_id\_","design":"","index":"","x":360,"y":80,"wires":[["040493930a2a7155"]]},

{"id":"040493930a2a7155","type":"switch","z":"b3626964d3ca3efc","name":"","property":"payload","propertyType":"msg","rules":[{"t":"null"},{"t":"else"}],"checkall":"true","repair":false,"outputs":2,"x":510,"y":140,"wires":[[],["36431bf85c31c423"]]},{"id":"36431bf85c31c423","type":"function","z":"b3626964d3ca3efc","name":"","func":"msg.payload=msg.payload.name\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":620,"y":80,"wires":[["dd5339bb0bbbe713","bf8eab40f9f68f2c","c56549b8d6ada0a4"]]},{"id":"dd5339bb0bbbe713","type":"function","z":"b3626964d3ca3efc","name":"","func":"var st={\"please take \":msg.payload}\nmsg.payload=JSON.stringify(st)\nmsg.payload=msg.payload.replace(':','');\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":780,"y":100,"wires":[["e81c06869a189846"]]},

{"id":"bf8eab40f9f68f2c","type":"ibmiot out","z":"b3626964d3ca3efc","authentication":"apiKey","apiKey":"c61d91360b8924ac","outputType":"cmd","deviceId":"1234567890","deviceType":"IOTdevice","eventCommandType":"command","format":"String","data":"1","qos":0,"name":"IBM IoT","service":"registered","x":780,"y":40,"wires":[]},

{"id":"e81c06869a189846","type":"watson-text-to-speech","z":"b3626964d3ca3efc","name":"","lang":"en-US","langhidden":"en-US","langcustom":"NoCustomisationSetting","langcustomhidden":"","voice":"en-US\_LisaExpressive","voicehidden":"en-US\_LisaExpressive","format":"audio/wav","password":"","apikey":"KSTdsMPsUS62SL58EqzaZbAFtEW2JlggKYHUI-NKLuvx","payload-response":true,"service-endpoint":"https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/10758658-1ffd-49e5-ae59-ffb2aaa3b131","x":960,"y":100,"wires":[["2dc705e9ed1b3ab2"]]},

{"id":"c56549b8d6ada0a4","type":"debug","z":"b3626964d3ca3efc","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","statusVal":"","statusType":"auto","x":790,"y":180,"wires":[]},{"id":"93e7a7e9e47c5383","type":"ui\_form","z":"b3626964d3ca3efc","name":"","label":"","group":"30afb0f3e06414d4","order":0,"width":0,"height":0,"options":[{"label":"Medicine Name","value":"name","type":"text","required":true,"rows":null},{"label":"Time","value":"time","type":"time","required":true,"rows":null},

{"label":"Date","value":"date","type":"date","required":true,"rows":null}],"formValue":{"name":"","time":"","date":""},

"payload":"","submit":"submit","cancel":"cancel","topic":"topic","topicType":"msg","splitLayout":"","className":"","x":790,"y":500,"wires":[["2e1eabb204c5f845","7fb1124b295547bd"]]},{"id":"1d9998b27568617e","type":"delay","z":"b3626964d3ca3efc","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":560,"y":480,"wires":[["93e7a7e9e47c5383"]]},

{"id":"2e1eabb204c5f845","type":"function","z":"b3626964d3ca3efc","name":"","func":"var d=msg.payload.date;\nvar t=msg.payload.time;\n\nvar date = d.slice(0,10)\nvar time=t.slice(10,25)\nvar hit=new Date(date+time)\nhit.setDate(hit.getDate() + 1);\nvar utc=hit.getTime()+(hit.getTimezoneOffset()\*60000);\nvar offset=5.5\nnewDate = new Date(utc+(3600000\*offset));\nvar n =newDate.toISOString()\nvar da = n.slice(0,10)\nvar ti=n.slice(11,16)\nmsg.payload={\n \"\_id\": da+\" \"+ti,\n \"name\": msg.payload.name\n}\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":680,"y":360,"wires":[["331a1eab63e02fac","51db7213a201c04f"]]},

{"id":"331a1eab63e02fac","type":"function","z":"b3626964d3ca3efc","name":"","func":"msg.payload={\n \"date\":\"\",\n \"time\":\"\",\n \"name\":\"\",\n}\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":920,"y":400,"wires":[["1d9998b27568617e"]]},

{"id":"7fb1124b295547bd","type":"debug","z":"b3626964d3ca3efc",

"name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","statusVal":"","statusType":"auto","x":1090,"y":480,"wires":[]},{"id":"7b28d944264a04aa","type":"cloudantplus in","z":"b3626964d3ca3efc","name":"",

"cloudant":"e8e674ee5e0cb282","database":"node-red-enlbv-2022-11-04","service":"\_ext\_","search":"\_id\_","design":"",

"index":"","x":1100,"y":320,"wires":[[]]},{"id":"3bd136d585ee16b8",

"type":"debug","z":"b3626964d3ca3efc","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false",

"statusVal":"","statusType":"auto","x":370,"y":200,"wires":[]},{"id":"51db7213a201c04f","type":"cloudantplus out","z":"b3626964d3ca3efc","name":"","cloudant":"e8e674ee5e0cb282","database":"node-red-enlbv-2022-11-04","service":"\_ext\_","payonly":true,"operation":"insert","x":1060,"y":240,"wires":[[]]},{"id":"2dc705e9ed1b3ab2","type":"play audio","z":"b3626964d3ca3efc","name":"",

"voice":"","x":1170,"y":100,"wires":[]},

{"id":"e8e674ee5e0cb282","type":"cloudantplus",

"host":"https://apikey-v2-2jzy07gxh6foo2jhn5tfo1k8c12ueqn3weg9kolpkm2n:7b9e69c73c1ff8711d0f323f05376bbd@531665c2-3667-4c6a-8eb7-df74d7359f6a-bluemix.cloudantnosqldb.appdomain.cloud","name":"","useapikey":false},

{"id":"c61d91360b8924ac","type":"ibmiot","name":"device api","keepalive":"60","serverName":""

,"cleansession":true,"appId":"","shared":false},

{"id":"30afb0f3e06414d4","type":"ui\_group","name":"Medicine details","tab":"4c03f41d0461d64d","order":1,"disp":true,"width":"6","collapse":false,"className":""},

{"id":"4c03f41d0461d64d","type":"ui\_tab","name":"Medicine details","icon":"dashboard","disabled":false,"hidden":false}]

**GitHub link:** <https://github.com/IBM-EPBL/IBM-Project-18767-1659689727>

**Demo link:** [**https://www.youtube.com/watch?v=rk7FFGiRycI**](https://www.youtube.com/watch?v=rk7FFGiRycI)