PERSONAL ASSISTANCE FOR SENIORS WHO ARE SELF-RELIANT

1. INTRODUCTION

1.1 Project Overview

Sometimes elderly people forget to take their medicine at the correct time.They also forget which medicine He / She should take at that particular time. And it is difficult for doctors/caretakers to monitor the patients around the clock. To avoid this problem, this medicine reminder system is developed. An Online web application is developed through which the caretaker can update the medication details and timing. There is a Backend to store all the medication details and time.And comes the use of an IoT device which receives the medication name at the desired time.Then the use of Text To Speech service to speak out the medicine name through the IoT device after receiving it from the web application.

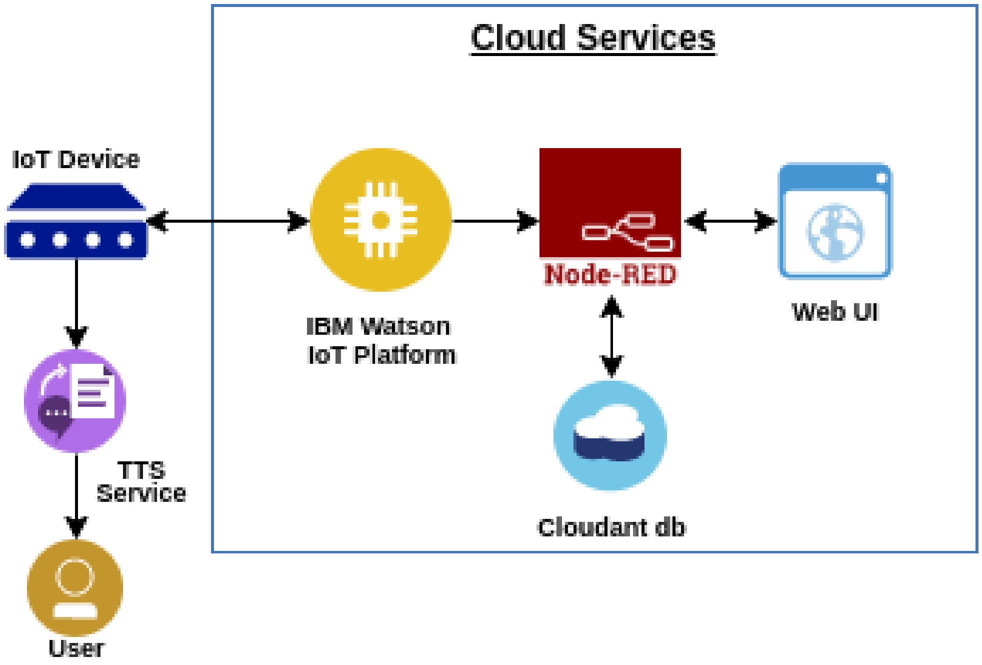


Fig 1.1.1 Technology architecture

The following cloud services are as follows,

1. IBM Watson IoT Platform
2. Node-RED Service
3. Cloudant DB
4. TTS Service

Need to have basic knowledge on the above mentioned cloud services.

1.2 Purpose

All of us have a busy hectic schedule. Today’s life is full of responsibilities and stress. So people are prone to diseases of different types and it is our duty to make ourselves stay fit and healthy. If the patient stays at home then he or she might get someone to look after him/her but when one is not at home, is out of the city or state away from home then it is hard for the family members to call them and remind them their dosage timings every time.

In our developing and technology dependent life we totally rely on gadgets especially smartphones. With this we get an opportunity to use technology in a better way so that it can be made useful to us. And it plays an important part in our daily life and helps us staying fit in many ways. The remarkable problem is that patients forget to take the proper medicines in proper proportion and in proper time. Medication adherence, which refers to the degree or extent to which a patient takes the right medication at the right time according to a doctor’s prescription, has recently emerged as a serious issue because many studies have reported that non-adherence may critically affect the patient, thereby raising medical costs.So our objective is to give reminder to the self reliant people about their medications. The patients can leave taking medicines to just our app. Whenever the time for the medicine is up, they will be notified and they only have to take their prescriptions during that time, and no other time. If implemented properly, this will drastically decrease overdose of medicines due to forgetfulness and the patients will also be reminded to take their medicines

2. LITERATURE SURVEY

2.1 Existing problem

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. Furthermore, some patients are so occupied with their day-to-day activities that they just forget to take their medications.

This is particularly true for old patients who have to take more than one medicine at more than one time in a day. Setting alarm clocks is a tedious task which patients are too lazy to set again and again. If asked about what time people have to take their medicines, many forget to answer the correct times or remember whether they have already taken the medicine in the day already

Elder people specially face this problem because of their degrading memory and in severe cases, forget that they have already taken their prescription and retake the same medicine 2 or 3 times in the same duration. This may not be harmful for lighter medicines, but for some strong and concentrated medicines, it can have further harmful effects to the body. This is exactly where our medicine reminder system can help. Our system takes up the prescription details from the user such as the duration of the prescription, the names of the medicines, the times they are to be taken and the amount of each medicine which is to be taken.

Thus the solution for the above mentioned problem is the MEDICATION REMINDER for the PEOPLE WHO ARE SELF RELIANT. It can help the users to take their prescribed medications at right time.

2.2 References

1. "Medication Adherence", available at: http://circ.ahajournals.org/content/119/23/3028.full

2. “Thinking Outside the Pillbox: A System-wide Approach to Improving Patient Medication Adherence for Chronic Disease” (2009), A NEHI Research Brief July 2009, New England Healthcare Institut.

3. Park, KeeHyun & Lim, Seung Hyeon, (2012) “Construction of a Medication Reminder Synchronization System based on Data Synchronization”, International Journal of Bio-Science and Bio-Technology, Vol.4, No. 4, pp1-10.

4. IoT based Pill reminder and monitoring system,2020 Journal of Computer science and Network security Sultan Ahmad, Tasnia Tabassum

5. Medication Reminder And Healthcare,2015 Journal of Managing Public Sector Information and Communication Technologies Deepti Kalpana Mudaliar and Palak Patel

6. Development of an Android Based Medication Reminder
2020 LAUTECH Journal of Computing and Informatics Adeyemi, T.O. Amusan, E. A.

2.3 Problem Statement Definition

Aging cannot be prevented and old age is a sensitive phase. But we can learn how to deal with arising conditions for great health of our loved ones. Elderly people need care and comfort to lead a healthy life without worries and anxiety. We cannot completely divert them from worrying but we can take care of them properly by providing proper medicines at the correct time. 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 application is built for the user (caretaker) which enables him / her to set the desired time and medicine. These details will be stored in the cloud database. If the medicine time arrives the web application will send the medicine name to the IoT Device through the IoT platform. The device will receive the medicine name and notify the user with voice commands.By providing an intimation like this, we can take care of the health of the elderly people by making them to take the right medicines at the right time.

The main aim is to ensure that the IoT-Based Smart Medicine Reminder Device will be solving problems faced by the elderly. The issues that have been identified are targeted very much to the elderly and are aimed to solve the issues faced by the elderly on a daily basis, especially with the consumption of medicine. This application was designed for the appropriate medication administration including time and dosages through: recording patient and medication data; scheduling patients' medication; and reporting medication administration on progress.

3. IDEATION AND PROPOSED SOLUTION

Fig 3.1 Medication reminder

3.1 Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours 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 person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

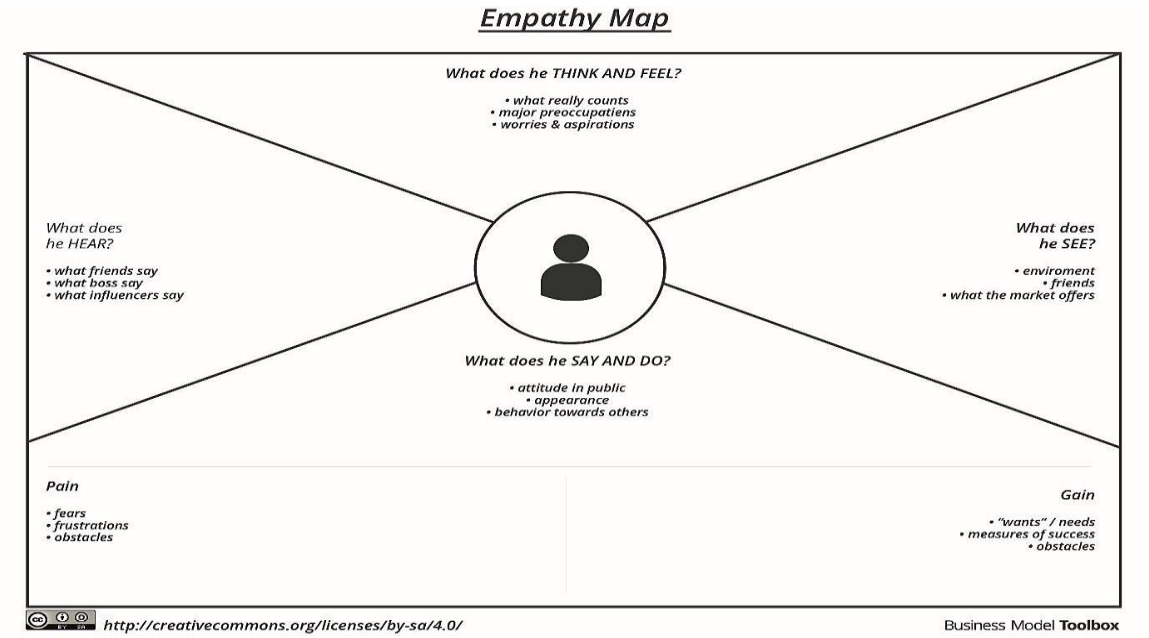
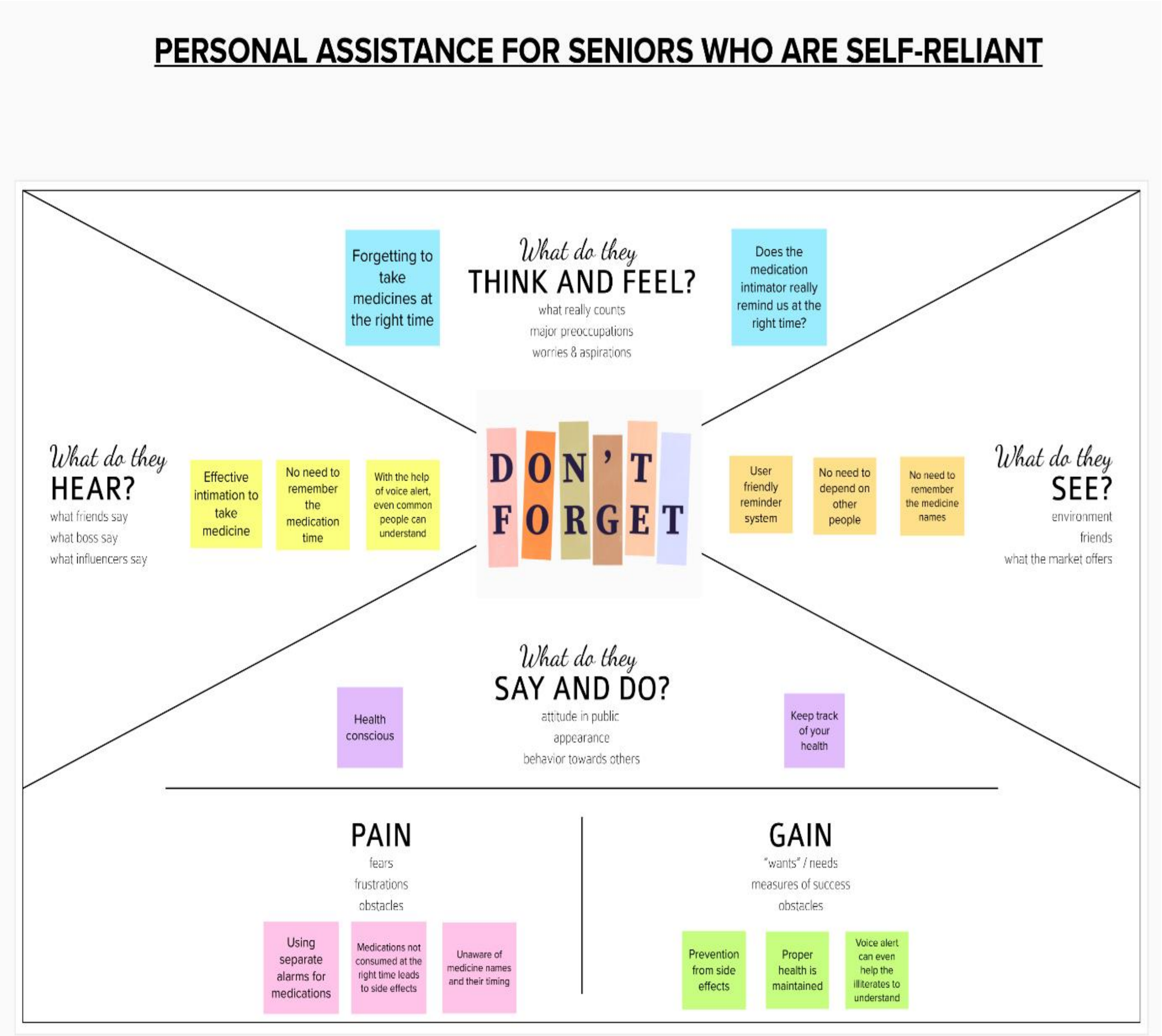


Fig 3.1.1 Empathy map-sample

Reference: https://www.mural.co/templates/empathy-map-canvas



PERSONAL ASSISTANCE FOR SENIORS WHO ARE SELF-RELIANT

What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

Forgetting to take medicines at the right time

Does the medication intimator really remind us at the right time?

What do they HEAR?
what friends say
what boss say
what influencers say

Effective intimation to take medicine

No need to remember the medication time

With the help of voice alert, even common people can understand

DON'T FORGET

User friendly reminder system

No need to depend on other people

No need to remember the medicine names

What do they SEE?
environment
friends
what the market offers

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

Health conscious

Keep track of your health

PAIN
fears
frustrations
obstacles

Using separate alarms for medications

Medications not consumed at the right time leads to side effects

Unaware of medicine names and their timing

GAIN
"wants" / needs
measures of success
obstacles

Prevention from side effects

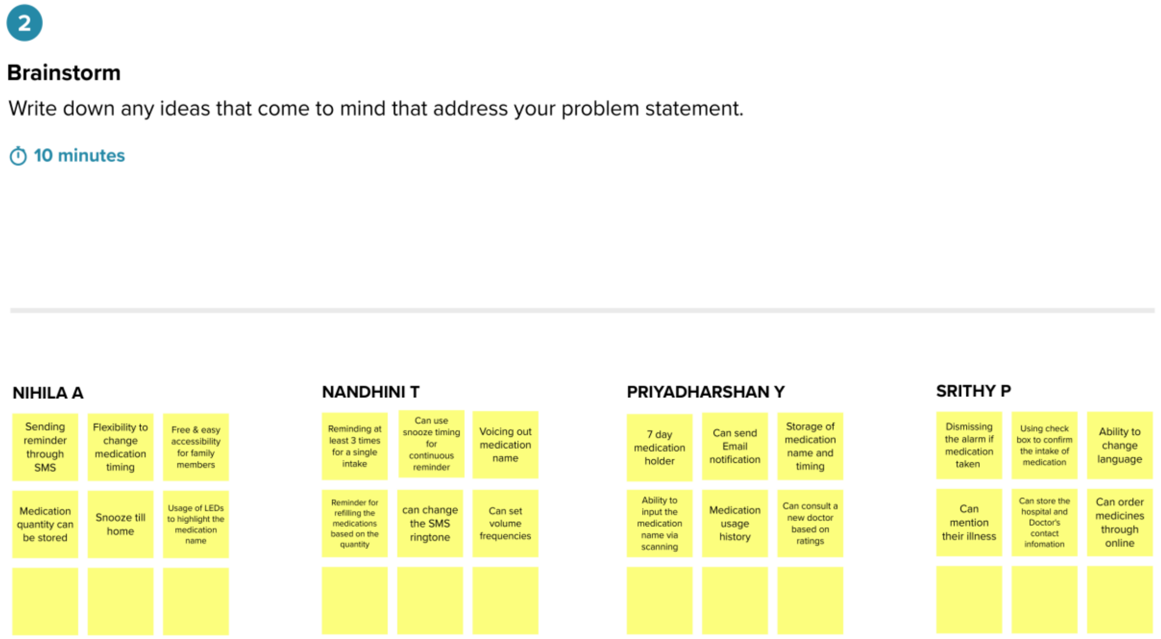
Proper health is maintained

Voice alert can even help the illiterates to understand

Fig 3.1.2 Empathy map

3.2 Ideation and Brainstorming:

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity. Brainstorming is usually conducted by getting a group of people together to come up with either general new ideas or ideas for solving a specific problem or dealing with a specific situation.



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

NIHILA A

- Sending reminder through SMS
- Flexibility to change medication timing
- Free & easy accessibility for family members
- Medication quantity can be stored
- Snooze till home
- Usage of LEDs to highlight the medication name

NANDHINI T

- Reminding at least 3 times for a single intake
- Can use snooze timing for continuous reminder
- Voicing out medication name
- Reminder for refilling the medications based on the quantity
- can change the SMS ringtone
- Can set volume frequencies

PRIYADHARSHAN Y

- 7 day medication holder
- Can send Email notification
- Storage of medication name and timing
- Ability to input the medication name via scanning
- Medication usage history
- Can consult a new doctor based on ratings

SRITHY P

- Dismissing the alarm if medication taken
- Using check box to confirm the intake of medication
- Ability to change language
- Can mention their illness
- Can store the hospital and Doctor's contact infomation
- Can order medicines through online

Fig 3.2.1 Brainstorming template

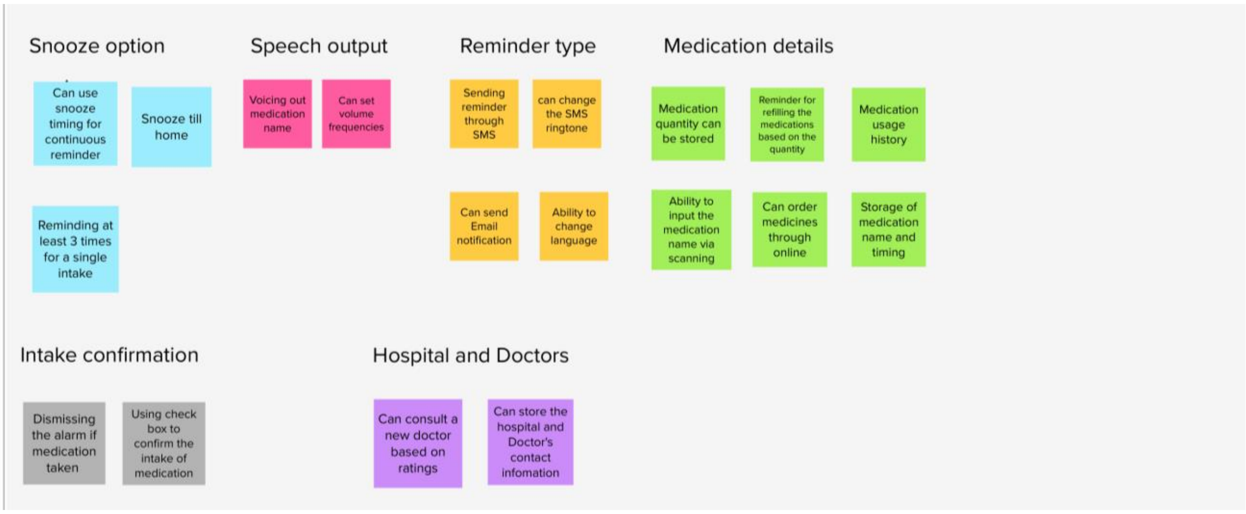

Snooze option
Can use snooze timing for continuous reminder
Snooze till home
Reminding at least 3 times for a single intake
Speech output
Voicing out medication name
Can set volume frequencies
Reminder type
Sending reminder through SMS
can change the SMS ringtone
Can send Email notification
Ability to change language
Medication details
Medication quantity can be stored
Reminder for refilling the medications based on the quantity
Medication usage history
Ability to input the medication name via scanning
Can order medicines through online
Storage of medication name and timing
Intake confirmation
Dismissing the alarm if medication taken
Using check box to confirm the intake of medication
Hospital and Doctors
Can consult a new doctor based on ratings
Can store the hospital and Doctor's contact infomation

Fig 3.2.2 - Ideas

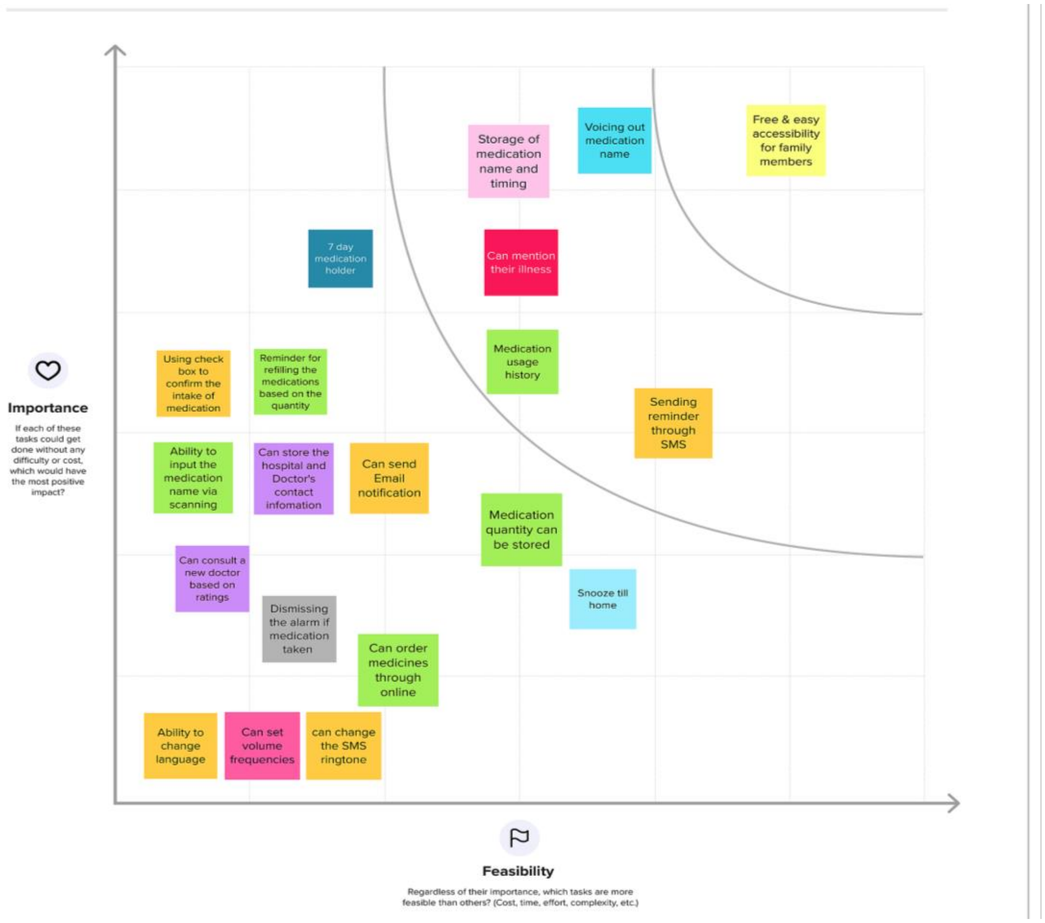
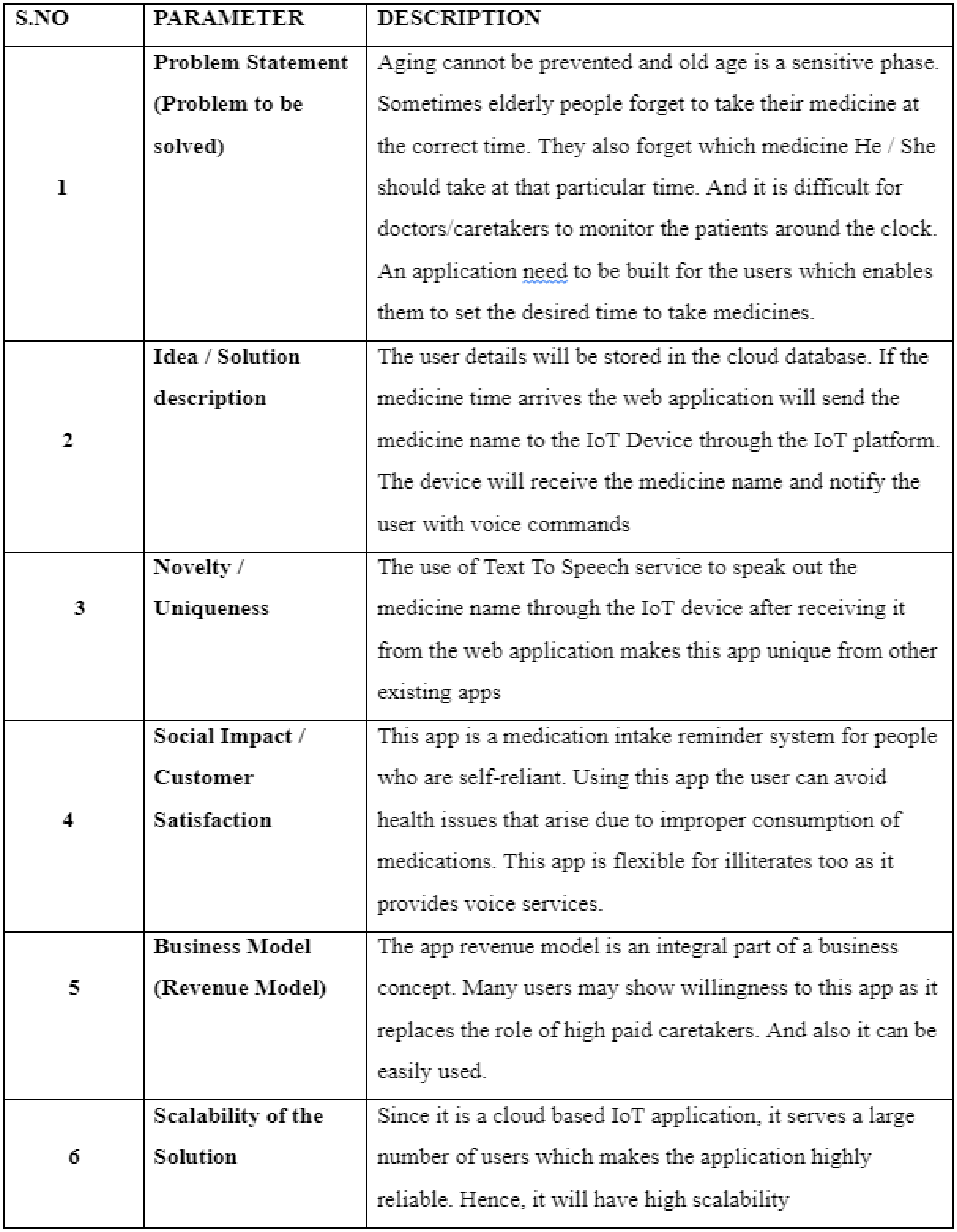

Storage of medication name and timing
Voicing out medication name
Free & easy accessibility for family members
7 day medication holder
Can mention their illness
Medication usage history
Using check box to confirm the intake of medication
Reminder for refilling the medications based on the quantity
Sending reminder through SMS
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Ability to input the medication name via scanning
Can store the hospital and Doctor's contact infomation
Can send Email notification
Medication quantity can be stored
Can consult a new doctor based on ratings
Snooze till home
Dismissing the alarm if medication taken
Can order medicines through online
Ability to change language
Can set volume frequencies
can change the SMS ringtone
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Fig 3.2.3 - Prioritization

3.3 Proposed Solution:



S.NO	PARAMETER	DESCRIPTION
1	**Problem Statement (Problem to be solved)**	Aging cannot be prevented and old age is a sensitive phase. 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. An application need to be built for the users which enables them to set the desired time to take medicines.
2	**Idea / Solution description**	The user details will be stored in the cloud database. If the medicine time arrives the web application will send the medicine name to the IoT Device through the IoT platform. The device will receive the medicine name and notify the user with voice commands
3	**Novelty / Uniqueness**	The use of Text To Speech service to speak out the medicine name through the IoT device after receiving it from the web application makes this app unique from other existing apps
4	**Social Impact / Customer Satisfaction**	This app is a medication intake reminder system for people who are self-reliant. Using this app the user can avoid health issues that arise due to improper consumption of medications. This app is flexible for illiterates too as it provides voice services.
5	**Business Model (Revenue Model)**	The app revenue model is an integral part of a business concept. Many users may show willingness to this app as it replaces the role of high paid caretakers. And also it can be easily used.
6	**Scalability of the Solution**	Since it is a cloud based IoT application, it serves a large number of users which makes the application highly reliable. Hence, it will have high scalability

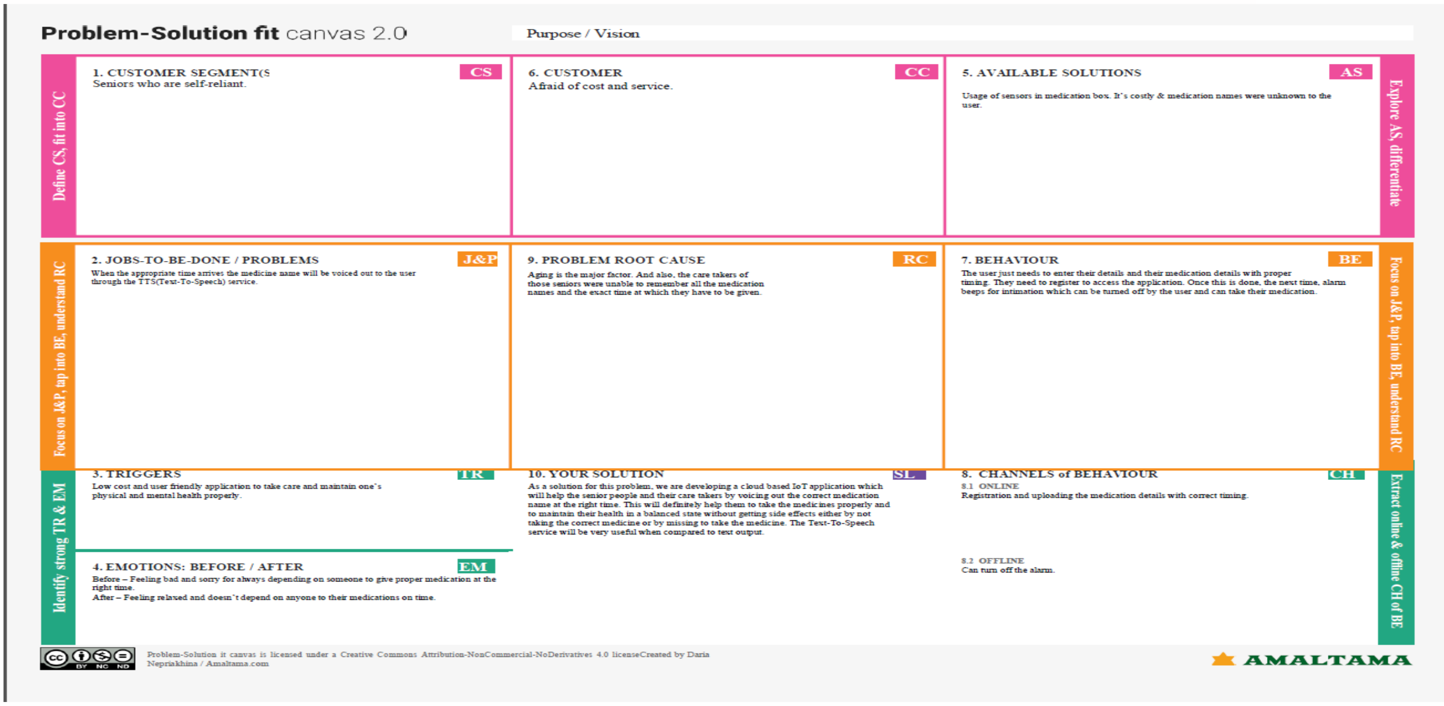
Table 3.3.1 Proposed solution table

3.4 Problem Solution fit:

The problem-solution fit is when you-

1. **Validate that the problem exists:** When you validate your problem hypothesis using real-world data and feedback. That is, you gather information from real users to determine whether or not they care about the pain point you're trying to solve.
2. **Validate that your solution solves the problem:** When you validate that the target audience appreciates the value your solution delivers to them.

The problem-solution fit precedes the product development and forms the foundation upon which a company is built. It helps you answer the basics startup-related questions before you even start your startup.



Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into CC

1. CUSTOMER SEGMENT(S CS
Seniors who are self-reliant.

6. CUSTOMER CC
Afraid of cost and service.

5. AVAILABLE SOLUTIONS AS
Usage of sensors in medication box. It's costly & medication names were unknown to the user.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P
When the appropriate time arrives the medicine name will be voiced out to the user through the TTS(Text-To-Speech) service.

9. PROBLEM ROOT CAUSE RC
Aging is the major factor. And also, the care takers of those seniors were unable to remember all the medication names and the exact time at which they have to be given.

7. BEHAVIOUR BE
The user just needs to enter their details and their medication details with proper timing. They need to register to access the application. Once this is done, the next time, alarm beeps for intimation which can be turned off by the user and can take their medication.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR
Low cost and user friendly application to take care and maintain one`s physical and mental health properly.

4. EMOTIONS: BEFORE / AFTER EM
Before – Feeling bad and sorry for always depending on someone to give proper medication at the right time.
After – Feeling relaxed and doesn`t depend on anyone to their medications on time.

10. YOUR SOLUTION SL
As a solution for this problem, we are developing a cloud based IoT application which will help the senior people and their care takers by voicing out the correct medication name at the right time. This will definitely help them to take the medicines properly and to maintain their health in a balanced state without getting side effects either by not taking the correct medicine or by missing to take the medicine. The Text-To-Speech service will be very useful when compared to text output.

8. CHANNELS of BEHAVIOUR CH
8.1 ONLINE
Registration and uploading the medication details with correct timing.

8.2 OFFLINE
Can turn off the alarm.

Extract online & offline CH of BE

Problem-Solution it canvas is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 licenseCreated by Daria Nepriakhina / Amaltama.com

AMALTAMA

Fig 3.4.1 Problem-solution fit

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	Access Cloud services	Access the cloud service with correct credentials Store the details in the database Retrieve needed information for the user's operation
FR-4	IOT configuration	Fine Tuning the IOT device based on preference Access the Cloud DB via device Manage the request and response effectively

Table 4.1.1 Functional requirements

4.2 Non-Functional requirements

FR NO.	Non functional requirement	Description
NFR-1	Usability	App can be used by anyone who has operational knowledge about internet and computer.
NFR-2	Security	For security, TFA is enabled and biometrics are also added for user safety
NFR-3	Reliability	Highly reliable since, It uses Trusted cloud services like IBM
NFR-4	Performance	Performance is better compared to other market products.
NFR-5	Availability	Available on mobile app. Web version is getting ready for next release

Table 4.2.1 Non-functional requirements

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.

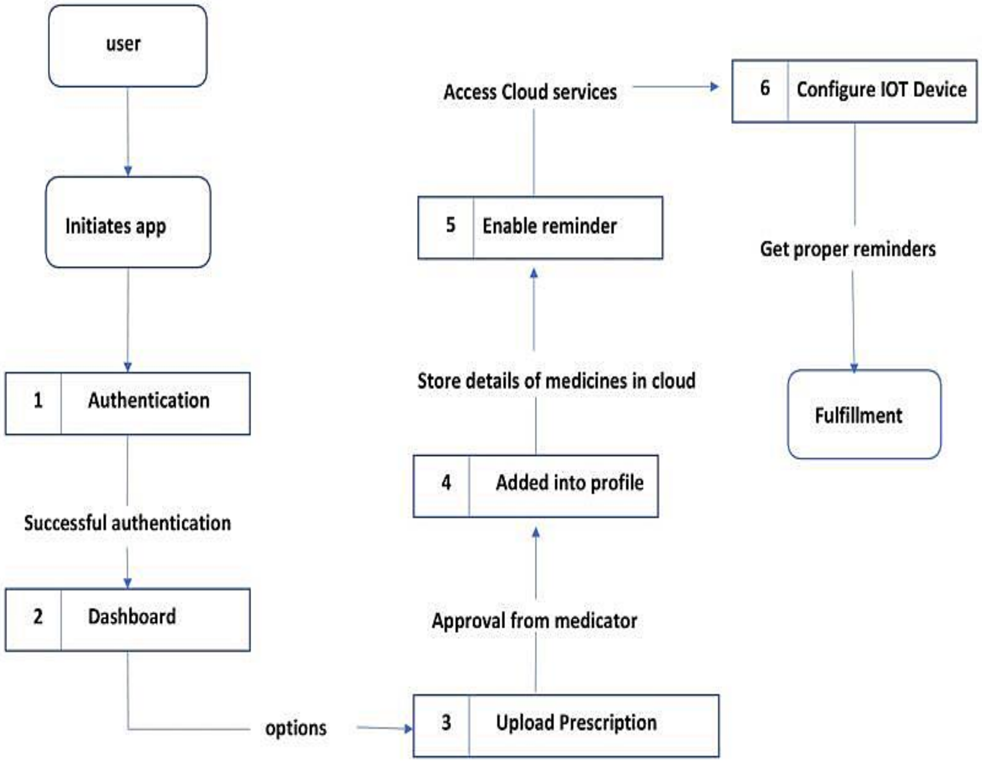


Fig 5.1.1 Data flow diagram

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.

GOALS:

1. Find the best tech solution to solve existing business problems.
2. Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
3. Define features, development phases, and solution requirements.
4. Provide specifications according to which the solution is defined, managed, and delivered.

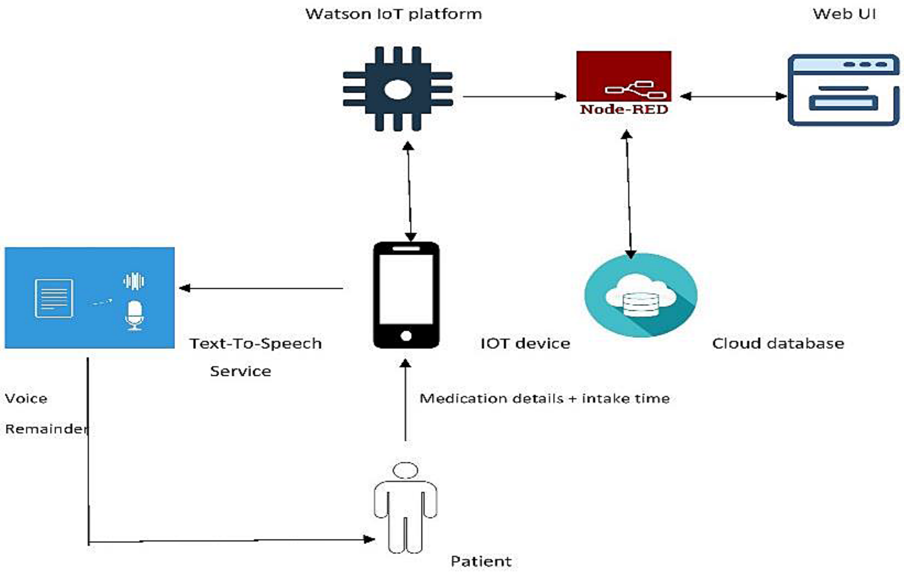


Fig 5.2.1 Solution architecture

Technical architecture—which is also often referred to as application architecture, IT architecture, business architecture, etc.—refers to creating a structured software solution that will meet the business needs and expectations while providing a strong technical plan for the growth of the software application through its lifetime. IT architecture is equally important to the business team and the information technology team. Technical architecture includes the major components of the system, their relationships, and the contracts that define the interactions between the components. The goal of technical architects is to achieve all the business needs with an application that is optimized for both performance and security.

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Reference: https://ieeexplore.ieee.org/document/9640866

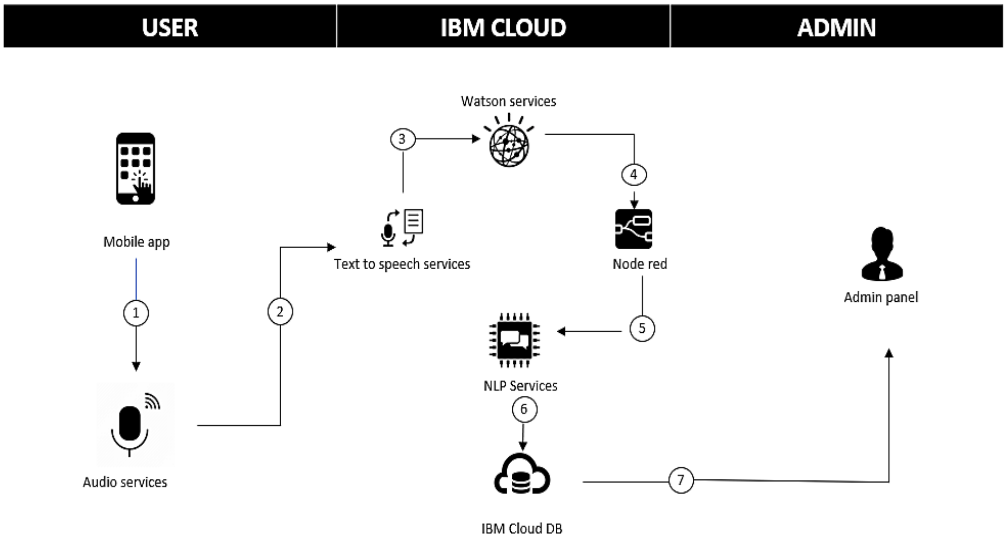
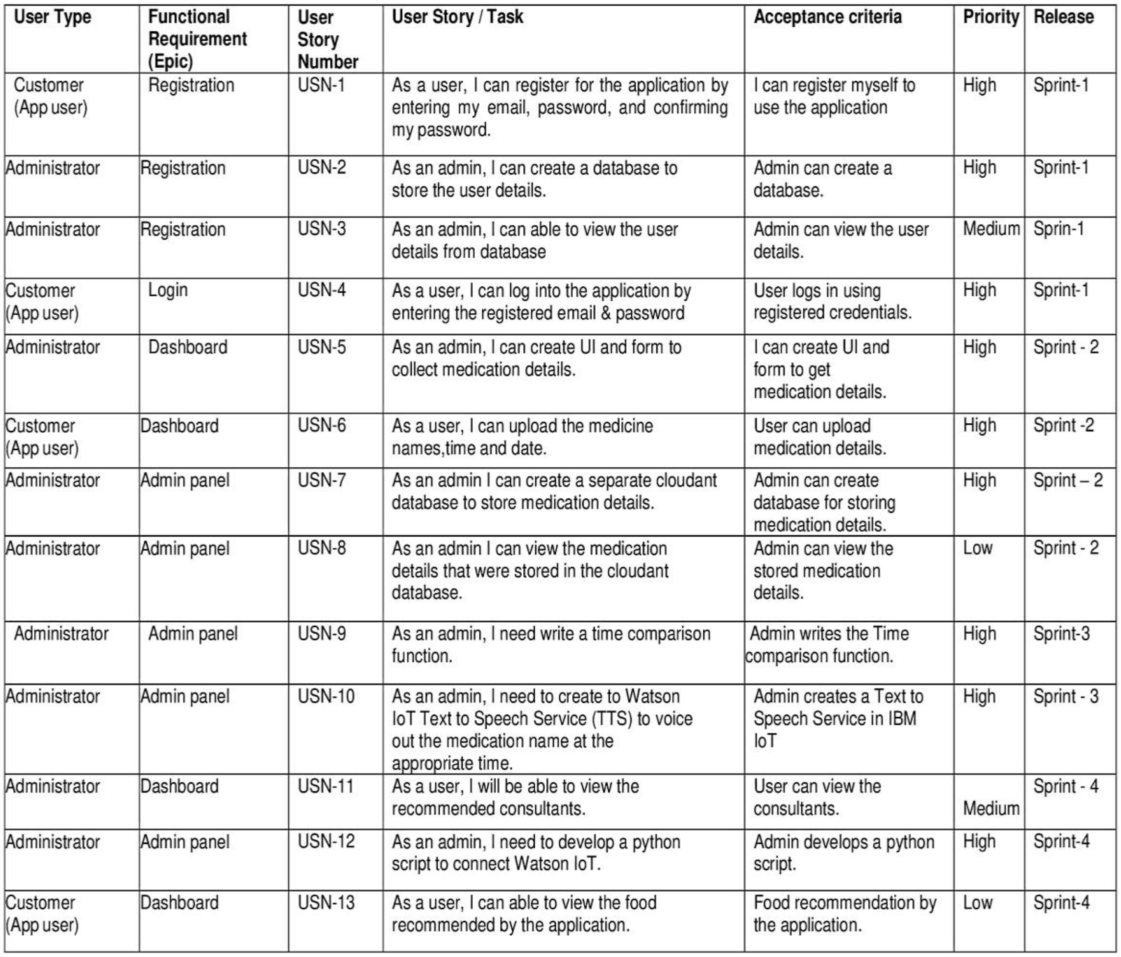


Fig 5.2.2 Technical architecture

5.3 User Stories

A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work — which drives collaboration, creativity, and a better product overall.



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (App user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can register myself to use the application	High	Sprint-1
Administrator	Registration	USN-2	As an admin, I can create a database to store the user details.	Admin can create a database.	High	Sprint-1
Administrator	Registration	USN-3	As an admin, I can able to view the user details from database	Admin can view the user details.	Medium	Sprin-1
Customer (App user)	Login	USN-4	As a user, I can log into the application by entering the registered email & password	User logs in using registered credentials.	High	Sprint-1
Administrator	Dashboard	USN-5	As an admin, I can create UI and form to collect medication details.	I can create UI and form to get medication details.	High	Sprint - 2
Customer (App user)	Dashboard	USN-6	As a user, I can upload the medicine names,time and date.	User can upload medication details.	High	Sprint -2
Administrator	Admin panel	USN-7	As an admin I can create a separate cloudant database to store medication details.	Admin can create database for storing medication details.	High	Sprint – 2
Administrator	Admin panel	USN-8	As an admin I can view the medication details that were stored in the cloudant database.	Admin can view the stored medication details.	Low	Sprint - 2
Administrator	Admin panel	USN-9	As an admin, I need write a time comparison function.	Admin writes the Time comparison function.	High	Sprint-3
Administrator	Admin panel	USN-10	As an admin, I need to create to Watson IoT Text to Speech Service (TTS) to voice out the medication name at the appropriate time.	Admin creates a Text to Speech Service in IBM IoT	High	Sprint - 3
Administrator	Dashboard	USN-11	As a user, I will be able to view the recommended consultants.	User can view the consultants.	Medium	Sprint - 4
Administrator	Admin panel	USN-12	As an admin, I need to develop a python script to connect Watson IoT.	Admin develops a python script.	High	Sprint-4
Customer (App user)	Dashboard	USN-13	As a user, I can able to view the food recommended by the application.	Food recommendation by the application.	Low	Sprint-4

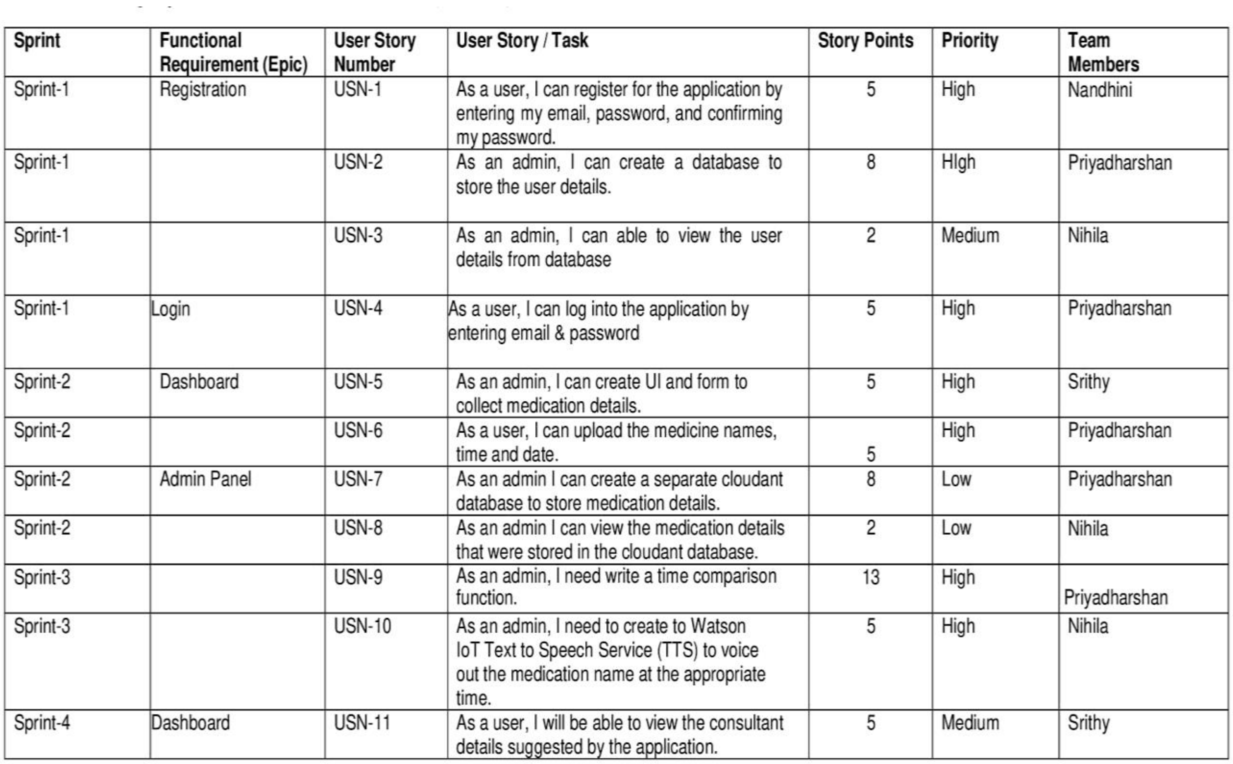
Table 5.3.1 User stories

6. PROJECT PLANNING & SCHEDULING

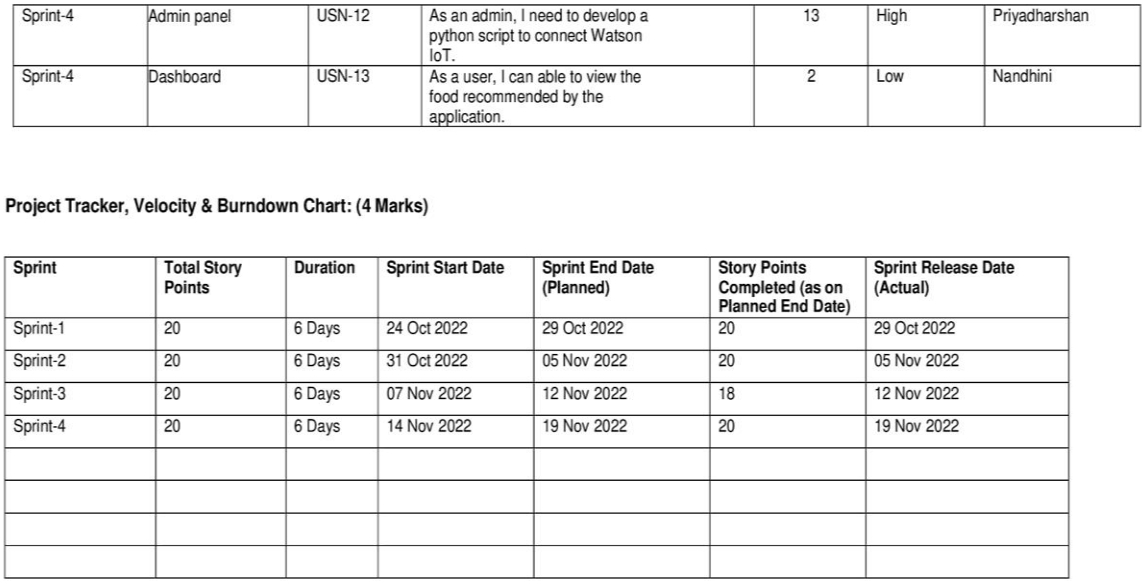
6.1 Sprint Planning & Estimation

What are the steps in sprint planning?

1. Examine team availability.
2. Establish velocity for your team.
3. Plan your sprint planning meeting.
4. Start with the big picture.
5. Present new updates, feedback, and issues.
6. Confirm team velocity and capacity.
7. Go over backlog items.
8. Determine task ownership.



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Nandhini
Sprint-1		USN-2	As an admin, I can create a database to store the user details.	8	HIgh	Priyadharshan
Sprint-1		USN-3	As an admin, I can able to view the user details from database	2	Medium	Nihila
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	5	High	Priyadharshan
Sprint-2	Dashboard	USN-5	As an admin, I can create UI and form to collect medication details.	5	High	Srithy
Sprint-2		USN-6	As a user, I can upload the medicine names, time and date.	5	High	Priyadharshan
Sprint-2	Admin Panel	USN-7	As an admin I can create a separate cloudant database to store medication details.	8	Low	Priyadharshan
Sprint-2		USN-8	As an admin I can view the medication details that were stored in the cloudant database.	2	Low	Nihila
Sprint-3		USN-9	As an admin, I need write a time comparison function.	13	High	Priyadharshan
Sprint-3		USN-10	As an admin, I need to create to Watson IoT Text to Speech Service (TTS) to voice out the medication name at the appropriate time.	5	High	Nihila
Sprint-4	Dashboard	USN-11	As a user, I will be able to view the consultant details suggested by the application.	5	Medium	Srithy



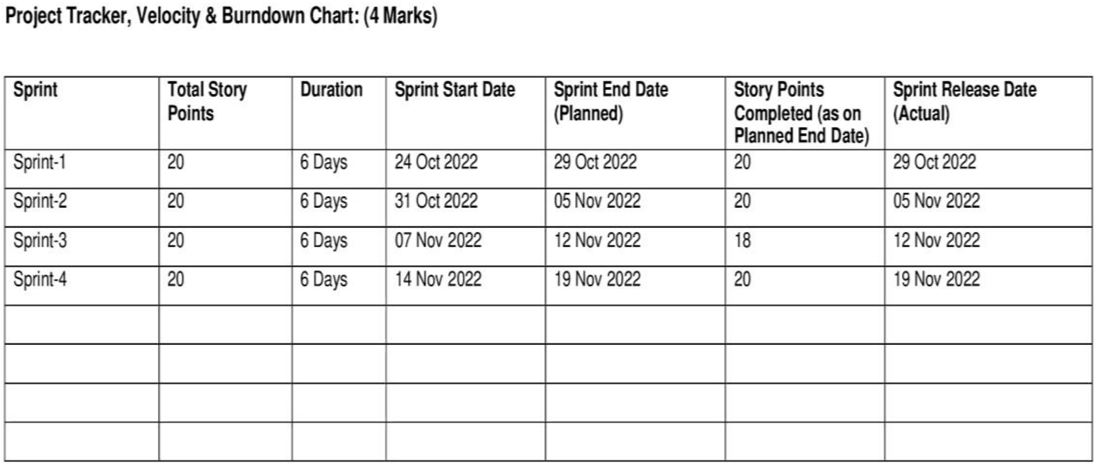
Sprint-4	Admin panel	USN-12	As an admin, I need to develop a python script to connect Watson IoT.	13	High	Priyadharshan
Sprint-4	Dashboard	USN-13	As a user, I can able to view the food recommended by the application.	2	Low	Nandhini

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	18	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Table 6.1.1 Sprint planning and estimation

6.2 Sprint Delivery Schedule



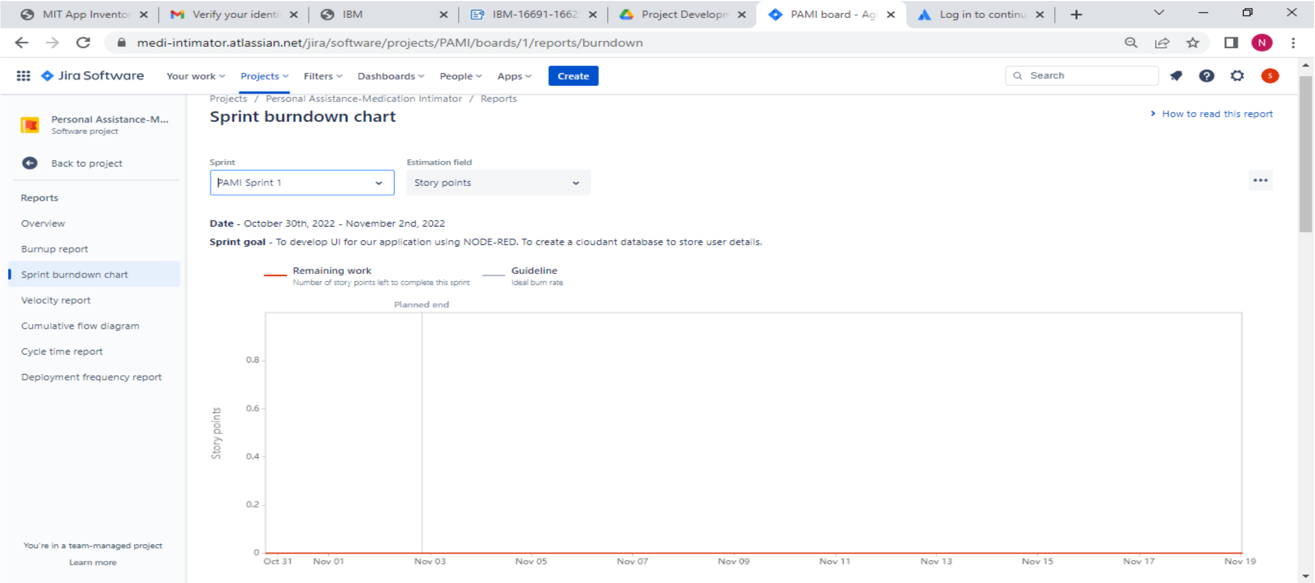
Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	18	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Table 6.2.1 Sprint delivery schedule

6.3 Reports from JIRA

SPRINT-1 REPORT



Sprint burndown chart

Sprint: PAMI Sprint 1

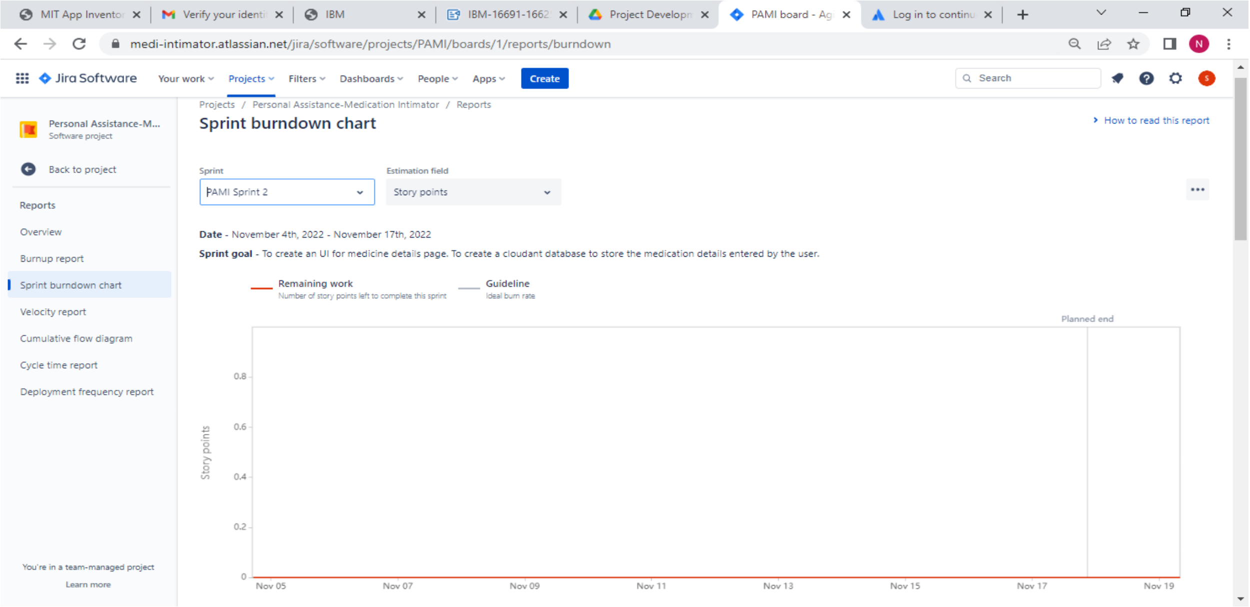
Estimation field: Story points

Date - October 30th, 2022 - November 2nd, 2022

Sprint goal - To develop UI for our application using NODE-RED. To create a cloudant database to store user details.

Fig 6.3.1 Sprint-1 report

SPRINT-2 REPORT



Sprint burndown chart

Sprint: PAMI Sprint 2

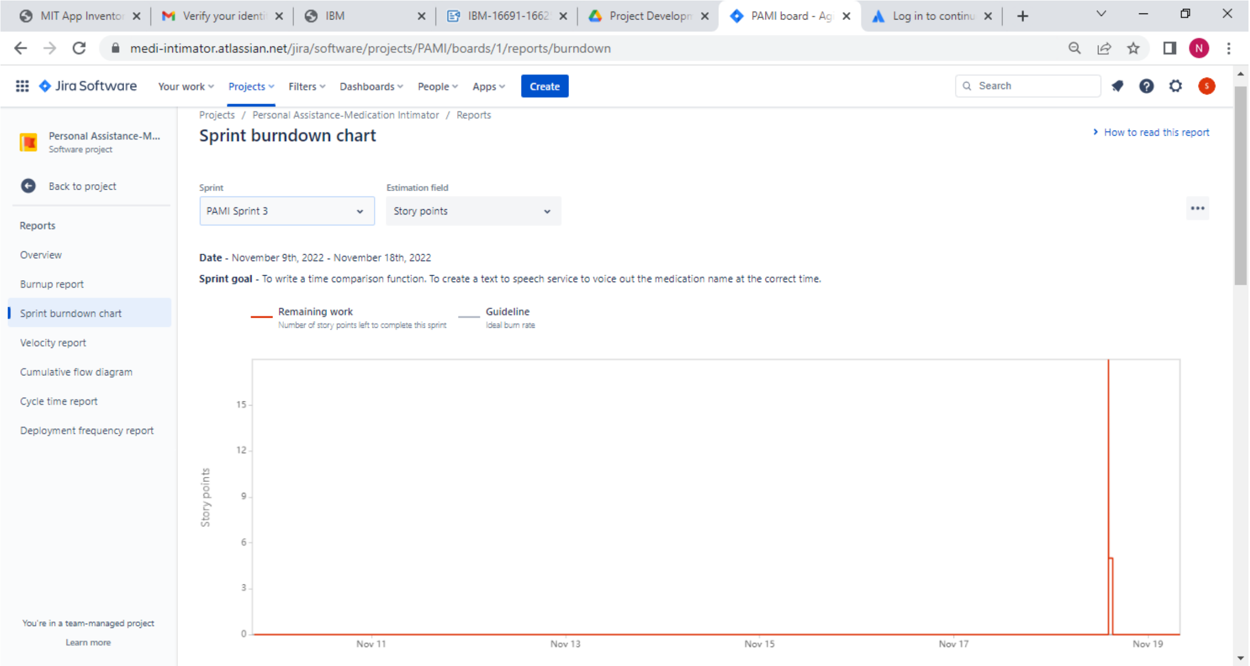
Estimation field: Story points

Date - November 4th, 2022 - November 17th, 2022

Sprint goal - To create an UI for medicine details page. To create a cloudant database to store the medication details entered by the user.

Fig 6.3.2 Sprint-2 report

SPRINT-3 REPORT



medi-intimator.atlassian.net/jira/software/projects/PAMI/boards/1/reports/burndown

Projects / Personal Assistance-Medication Intimator / Reports

Sprint burndown chart

Sprint: PAMI Sprint 3

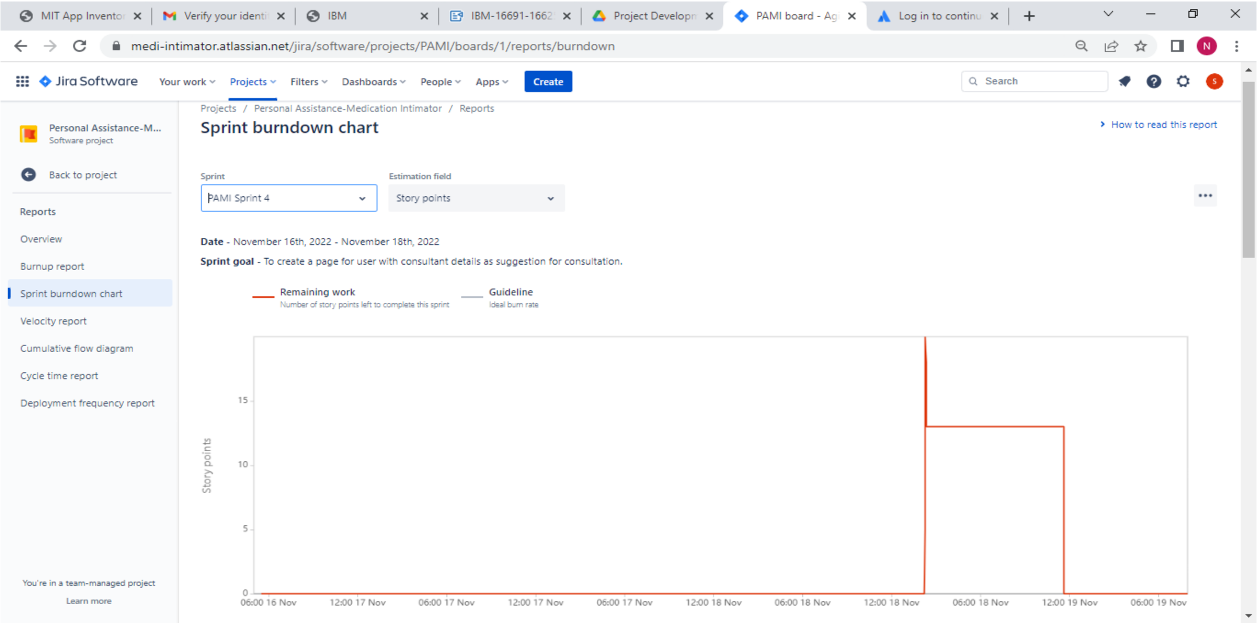
Estimation field: Story points

Date - November 9th, 2022 - November 18th, 2022

Sprint goal - To write a time comparison function. To create a text to speech service to voice out the medication name at the correct time.

Fig 6.3.3 Sprint-report 3

SPRINT-4 REPORT



medi-intimator.atlassian.net/jira/software/projects/PAMI/boards/1/reports/burndown

Projects / Personal Assistance-Medication Intimator / Reports

Sprint burndown chart

Sprint: PAMI Sprint 4

Estimation field: Story points

Date - November 16th, 2022 - November 18th, 2022

Sprint goal - To create a page for user with consultant details as suggestion for consultation.

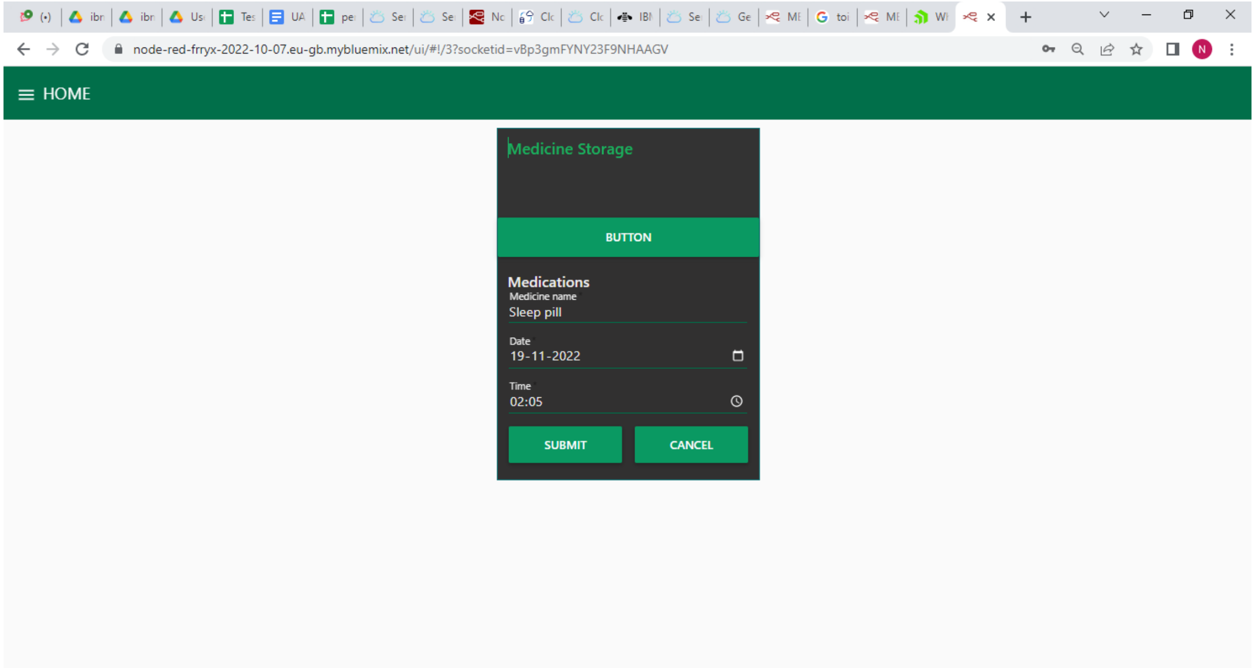
Fig 6.3.4 Sprint-report 4

7 CODING AND SOLUTIONING

7.1 Feature 1

Medication details storage:

The main aim of our application to get the medication details from user and store them in our cloudant database. The home page contains this medication details form. All these data will get stored in the "medications" database.


node-red-frryx-2022-10-07.eu-gb.mybluemix.net/ui/#!/3?socketid=vBp3gmFYNY23F9NHAAGV
HOME
Medicine Storage
BUTTON
SUBMIT
CANCEL

Medications

Medicine name
Sleep pill

Date
19-11-2022

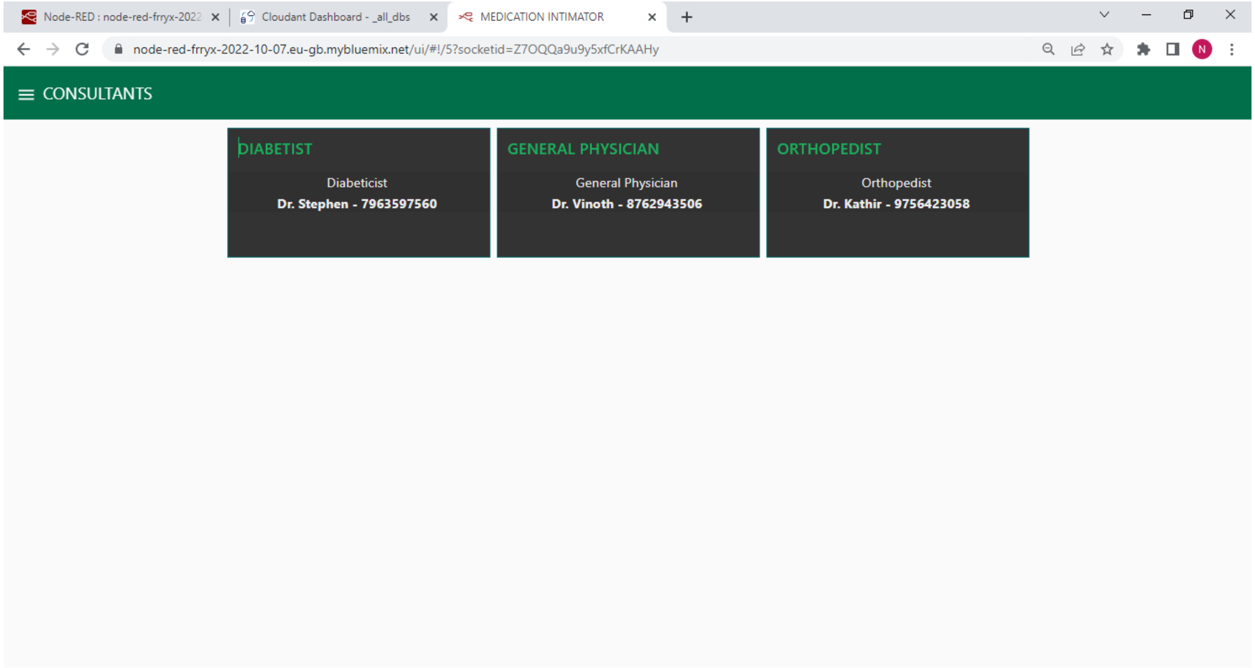
Time
02:05

Fig 7.1.1 -Medication details form

7.2 Feature 2

Consultants information and food recommendation:

This feature is an add-on to our application. In the Medication Intimator web application all the users will be able to view the counsultants page and food recommendation page. In the consultants tabn some of the counsultants from few different fields will be available. And in the What to eat tab, food will be recommended to the users in general with some specifications.


Node-RED : node-red-frryx-2022
Cloudant Dashboard - _all_dbs
MEDICATION INTIMATOR
node-red-frryx-2022-10-07.eu-gb.mybluemix.net/ui/#!/5?socketid=Z7OQQa9u9y5xfCrKAAHy
N
≡ CONSULTANTS

DIABETIST

Diabeticist
Dr. Stephen - 7963597560

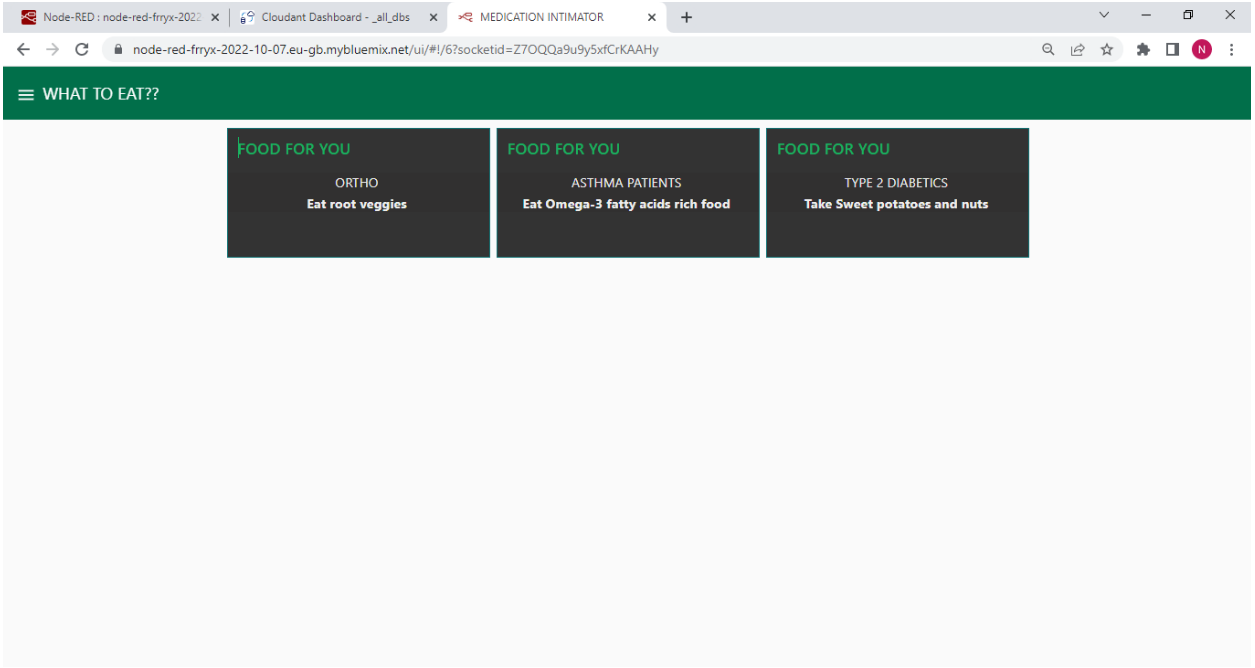
GENERAL PHYSICIAN

General Physician
Dr. Vinoth - 8762943506

ORTHOPEDIST

Orthopedist
Dr. Kathir - 9756423058

Fig 7.2.1- Consultants tab


Node-RED : node-red-frryx-2022
Cloudant Dashboard - _all_dbs
MEDICATION INTIMATOR
node-red-frryx-2022-10-07.eu-gb.mybluemix.net/ui/#!/6?socketid=Z7OQQa9u9y5xfCrKAAHy
N

WHAT TO EAT??

FOOD FOR YOU

ORTHO

Eat root veggies

FOOD FOR YOU

ASTHMA PATIENTS

Eat Omega-3 fatty acids rich food

FOOD FOR YOU

TYPE 2 DIABETICS

Take Sweet potatoes and nuts

Fig 7.2.2 - Food recommendation (What to eat) tab

CODE

NODE-RED FLOWCODE

```
[
    {
        "id": "772ba0efeb30c3c6",
        "type": "tab",
        "label": "IoT",
        "disabled": false,
        "info": "",
        "env": []
    },
    {
        "id": "74d2c457369f9f98",
        "type": "inject",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "props": [
            {
                "p": "payload"
            }
        ],
        "repeat": "",
        "crontab": "",
        "once": false,
        "onceDelay": 0.1,
        "topic": "",
        "payload": "",
        "payloadType": "date",
        "x": 100,
        "y": 60,
        "wires": [
            [
                "c8a2dc273924fe60"
            ]
        ]
    },
    {
        "id": "c8a2dc273924fe60",
        "type": "function",
        "z": "772ba0efeb30c3c6",
        "name": "timeFunction",
        "func": "\nvar 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": 250,
        "y": 40,
        "wires": [
            [
                "49828d64e64fff84"
            ]
        ]
    },
    {
        "id": "49828d64e64fff84",
        "type": "cloudant in",
        "z": "772ba0efeb30c3c6",
        "name": "medicine",
        "cloudant": "",
        "database": "medications",
        "service": "node-red-frryx-2022--cloudant-1665131455235-12133",
        "search": "_all_",
        "design": "",
        "index": "",
        "x": 420,
        "y": 40,
        "wires": [
            [
                "e83f21d1e93fe454"
            ]
        ]
    },
    {
        "id": "e83f21d1e93fe454",
        "type": "switch",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "property": "payload",
        "propertyType": "msg",
        "rules": [
            {
                "t": "eq",
                "v": "is null",
                "vt": "str"
            },
            {
                "t": "eq",
                "v": "otherwise",
                "vt": "str"
            }
        ],
        "checkall": "true",
        "repair": false,
        "outputs": 2,
        "x": 590,
        "y": 40,
        "wires": [
            [],
            [
                "032f6c976e6d7fc7"
            ]
        ]
    },
    {
        "id": "032f6c976e6d7fc7",
        "type": "function",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "func": "msg.payload = {\"command\":msg.payload.mname};\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 600,
        "y": 80,
        "wires": [
            [
                "4c341564f66c4c56",
                "c61453ce912ceec9"
            ]
        ]
    },
    {
        "id": "4c341564f66c4c56",
        "type": "debug",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 790,
        "y": 80,
        "wires": []
    },
    {
        "id": "c61453ce912ceec9",
        "type": "ibmiot out",
        "z": "772ba0efeb30c3c6",
        "authentication": "apiKey",
        "apiKey": "466840d3b27c292d",
        "outputType": "cmd",
        "deviceId": "12345",
        "deviceType": "NodeMCU",
        "eventCommandType": "cmd",
        "format": "json",
        "data": "data",
        "qos": 0,
        "name": "IBM IoT",
        "service": "registered",
        "x": 780,
        "y": 120,
        "wires": []
    },
    {
        "id": "3368bb38e9143ae9",
        "type": "function",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "func": "var d=msg.payload.date;\nvar t = msg.payload.time;\nmsg.payload = {\n    \"_id\" : d +\" \"+t,\n    \"name\" : msg.payload.mname\n}\n\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 120,
        "y": 140,
        "wires": [
            [
                "91cb4d75a43c6970",
                "afa12d0a32d700ab"
            ]
        ]
    },
    {
        "id": "2d903e59b4169b40",
        "type": "cloudant out",
        "z": "772ba0efeb30c3c6",
        "name": "medicine",
        "cloudant": "",
        "database": "medications",
        "service": "node-red-frryx-2022--cloudant-1665131455235-12133",
        "payonly": true,
        "operation": "insert",
        "x": 660,
        "y": 380,
        "wires": []
    },
    {
        "id": "91cb4d75a43c6970",
        "type": "function",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "func": "msg.payload = {\n    \"date\":\"\",\n    \"name\":\"\",\n    \"time\":\"\"\n    \n}\nreturn
msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 320,
        "y": 140,
        "wires": [
            [
                "e20ddb3e70e38323"
            ]
        ]
    },
    {
        "id": "afa12d0a32d700ab",
        "type": "debug",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 330,
        "y": 100,
        "wires": []
    },
    {
        "id": "e20ddb3e70e38323",
        "type": "delay",
        "z": "772ba0efeb30c3c6",
        "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": 180,
        "wires": [
            [
                "c0cfd54d4bad4b3f"
            ]
        ]
    },
    {
        "id": "5f34e036f99d0e8f",
        "type": "ui_text",
        "z": "772ba0efeb30c3c6",
        "group": "c65e35ff24555591",
        "order": 1,
        "width": 0,
        "height": 0,
        "name": "",
        "label": "General Physician",
        "format": "Dr. Vinoth - 8762943506",
        "layout": "col-center",
      "className": "",
      "x": 1250,
      "y": 80,
      "wires": []
    },
    {
      "id": "481c1aa6307e0d80",
      "type": "ui_text",
      "z": "772ba0efeb30c3c6",
      "group": "c8375fc927c1b4eb",
      "order": 1,
      "width": 0,
      "height": 0,
      "name": "",
      "label": "Orthopedist",
      "format": "Dr. Kathir - 9756423058",
      "layout": "col-center",
      "className": "",
      "x": 1230,
      "y": 40,
      "wires": []
    },
    {
      "id": "8e71feebb056ba63",
      "type": "ui_text",
      "z": "772ba0efeb30c3c6",
      "group": "b39b0c959f2f9d67",
      "order": 1,
      "width": 0,
      "height": 0,
      "name": "",
      "label": "Diabeticist",
      "format": "Dr. Stephen - 7963597560",
      "layout": "col-center",
      "className": "",
      "x": 1230,
      "y": 120,
      "wires": []
    },
    {
      "id": "7ba73a7fd3ccd6b7",
      "type": "ui_text",
      "z": "772ba0efeb30c3c6",
      "group": "bbd3a14fee1266d4",
      "order": 1,
      "width": 0,
    "height": 0,
    "name": "",
    "label": "ORTHO",
    "format": "Eat root veggies",
    "layout": "col-center",
    "className": "",
    "x": 1220,
    "y": 160,
    "wires": []
  },
  {
    "id": "2cc1c6b37532dde8",
    "type": "ui_text",
    "z": "772ba0efeb30c3c6",
    "group": "db51e34f28d951bc",
    "order": 1,
    "width": 0,
    "height": 0,
    "name": "",
    "label": "TYPE 2 DIABETICS",
    "format": "Take Sweet potatoes and nuts",
    "layout": "col-center",
    "className": "",
    "x": 1260,
    "y": 200,
    "wires": []
  },
  {
    "id": "de4beeb347c3bb96",
    "type": "ui_text",
    "z": "772ba0efeb30c3c6",
    "group": "1b818abb6c71b0b2",
    "order": 1,
    "width": 0,
    "height": 0,
    "name": "",
    "label": "ASTHMA PATIENTS",
    "format": "Eat Omega-3 fatty acids rich food",
    "layout": "col-center",
    "className": "",
    "x": 1260,
    "y": 240,
    "wires": []
  },
  {
    "id": "3445b1bc227426f7",
        "type": "ui_button",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "group": "4006abb73245ff99",
        "order": 2,
        "width": 0,
        "height": 0,
        "passthru": false,
        "label": "REGISTER",
        "tooltip": "",
        "color": "",
        "bgcolor": "",
        "className": "",
        "icon": "",
        "payload": "REGISTER",
        "payloadType": "str",
        "topic": "topic",
        "topicType": "msg",
        "x": 110,
        "y": 300,
        "wires": [
            [
                "0ba257e81bdad4f4"
            ]
        ]
    },
    {
        "id": "0650dadc9f98c82b",
        "type": "ui_button",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "group": "4006abb73245ff99",
        "order": 1,
        "width": 0,
        "height": 0,
        "passthru": false,
        "label": "LOGIN",
        "tooltip": "",
        "color": "",
        "bgcolor": "",
        "className": "",
        "icon": "",
        "payload": "LOGIN",
        "payloadType": "str",
        "topic": "topic",
        "topicType": "msg",
        "x": 180,
        "y": 340,
        "wires": [
            [
                "0ba257e81bdad4f4"
            ]
        ]
    },
    {
        "id": "0ba257e81bdad4f4",
        "type": "ui_ui_control",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "events": "change",
        "x": 300,
        "y": 240,
        "wires": [
            [
                "7caeeb89712cebfd"
            ]
        ]
    },
    {
        "id": "5db2df3b7f0b7f91",
        "type": "ui_form",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "label": "LoginForm",
        "group": "876611f251aa851a",
        "order": 1,
        "width": 0,
        "height": 0,
        "options": [
            {
                "label": "Username",
                "value": "uname",
                "type": "text",
                "required": true,
                "rows": null
            },
            {
                "label": "Password",
                "value": "pwd",
                "type": "password",
                "required": true,
                "rows": null
            }
        ],
        "formValue": {
            "uname": "",
            "pwd": ""
        },
        "payload": "",
        "submit": "HOME",
        "cancel": "",
        "topic": "topic",
        "topicType": "msg",
        "splitLayout": "",
        "className": "",
        "x": 610,
        "y": 320,
        "wires": [
            [
                "355425b69e2759ef",
                "99d6dbd3d7672bd2"
            ]
        ]
    },
    {
        "id": "2f8194b073afa5f3",
        "type": "ui_form",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "label": "Enter your details",
        "group": "6eae6ed0cbde419d",
        "order": 1,
        "width": 0,
        "height": 0,
        "options": [
            {
                "label": "Username",
                "value": "uname1",
                "type": "text",
                "required": true,
                "rows": null
            },
            {
                "label": "Mobile Number",
                "value": "mobnum",
                "type": "number",
                "required": true,
                "rows": null
            },
            {
                "label": "Email",
                "value": "email",
                "type": "email",
                "required": false,
                "rows": null
            },
            {
                "label": "Password",
                "value": "pwd1",
                "type": "password",
                "required": true,
                "rows": null
            }
        ],
        "formValue": {
            "uname1": "",
            "mobnum": "",
            "email": "",
            "pwd1": ""
        },
        "payload": "",
        "submit": "Register",
        "cancel": "cancel",
        "topic": "payload",
        "topicType": "msg",
        "splitLayout": "",
        "className": "",
        "x": 630,
        "y": 240,
        "wires": [
            [
                "73c07523ebdc3b84",
                "99a3a7dee08dc495",
                "5e6d20adeb894548"
            ]
        ]
    },
    {
        "id": "7caeeb89712cebfd",
        "type": "switch",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "property": "payload",
        "propertyType": "msg",
        "rules": [
            {
                "t": "eq",
                "v": "\"REGISTER\"",
                "vt": "str"
            },
            {
                "t": "eq",
                "v": "\"LOGIN\"",
                "vt": "str"
            }
        ],
        "checkall": "true",
        "repair": false,
        "outputs": 2,
        "x": 450,
        "y": 240,
        "wires": [
            [
                "2f8194b073afa5f3"
            ],
            [
                "5db2df3b7f0b7f91"
            ]
        ]
    },
    {
        "id": "73c07523ebdc3b84",
        "type": "cloudant out",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "cloudant": "",
        "database": "newusers",
        "service": "node-red-frryx-2022--cloudant-1665131455235-12133",
        "payonly": true,
        "operation": "insert",
        "x": 840,
        "y": 260,
        "wires": []
    },
    {
        "id": "99a3a7dee08dc495",
        "type": "debug",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 830,
        "y": 220,
        "wires": []
    },
    {
        "id": "c0cfd54d4bad4b3f",
        "type": "ui_form",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "label": "Medications",
        "group": "5f5120bb7e50aed4",
        "order": 0,
        "width": 0,
        "height": 0,
        "options": [
            {
                "label": "Medicine name",
                "value": "mname",
                "type": "text",
                "required": true,
                "rows": null
            },
            {
                "label": "Date",
                "value": "date",
                "type": "date",
                "required": true,
                "rows": null
            },
            {
                "label": "Time",
                "value": "time",
                "type": "time",
                "required": true,
                "rows": null
            }
        ],
        "formValue": {
            "mname": "",
            "date": "",
            "time": ""
        },
        "payload": "",
        "submit": "submit",
        "cancel": "cancel",
        "topic": "topic",
        "topicType": "msg",
        "splitLayout": "",
        "className": "",
        "x": 310,
        "y": 420,
        "wires": [
            [
                "986b8511c38f7b47",
                "bd23ee3b066ed3bd",
                "34988011c7c5ca64",
                "cc5c4fcb7d32be59",
                "3368bb38e9143ae9"
            ]
        ]
    },
    {
        "id": "986b8511c38f7b47",
        "type": "debug",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 510,
        "y": 480,
        "wires": []
    },
    {
        "id": "34988011c7c5ca64",
        "type": "function",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "func": "\nvar medicinename = {};\nmedicinename.a = \"Its time to take \"+msg.payload.mname;\nglobal.set(\"mname\", medicinename);\n\nmsg.payload = \n{\n 'medicine': msg.payload.mname,\n    'global' : global.get(\"mname\")\n};\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 120,
        "y": 480,
        "wires": [
            [
                "433f937f8962e549"
            ]
        ]
    },
    {
        "id": "433f937f8962e549",
        "type": "debug",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 310,
        "y": 480,
        "wires": []
    },
    {
        "id": "4e810742a78cb1e9",
        "type": "ui_audio",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "group": "5f5120bb7e50aed4",
        "voice": "Google UK English Female",
        "always": "",
        "x": 920,
        "y": 580,
        "wires": []
    },
    {
        "id": "a72b31b145f77ceb",
        "type": "ui_button",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "group": "5f5120bb7e50aed4",
        "order": 4,
        "width": 0,
        "height": 0,
        "passthru": true,
        "label": "button",
        "tooltip": "speech",
        "color": "",
        "bgcolor": "",
        "className": "",
        "icon": "",
        "payload": "mname.a",
        "payloadType": "global",
        "topic": "payload",
        "topicType": "msg",
        "x": 770,
        "y": 580,
        "wires": [
            [
                "4e810742a78cb1e9"
            ]
        ]
    },
    {
        "id": "b1947670a172ce2b",
        "type": "debug",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": true,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 935,
        "y": 300,
        "wires": [],
        "l": false
    },
    {
        "id": "355425b69e2759ef",
        "type": "function",
        "z": "772ba0efeb30c3c6",
        "name": "auth",
        "func": "var username =msg.payload.uname;\nvar dbusername =
msg.payload.uname1;\n\nmsg.payload = {\n    'username in form': msg.payload.uname,\n
'username in db': msg.payload.uname1\n};\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 790,
        "y": 300,
        "wires": [
            [
                "b1947670a172ce2b"
            ]
        ]
    },
    {
        "id": "99d6dbd3d7672bd2",
        "type": "ui_button",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "group": "876611f251aa851a",
        "order": 1,
        "width": 0,
        "height": 0,
        "passthru": false,
        "label": "GO TO HOME",
        "tooltip": "",
        "color": "",
        "bgcolor": "",
        "className": "",
        "icon": "",
        "payload": "HOME",
        "payloadType": "str",
        "topic": "topic",
        "topicType": "msg",
        "x": 820,
        "y": 340,
        "wires": [
            [
                "b8f1777e69b6feae"
            ]
        ]
    },
    {
        "id": "b8f1777e69b6feae",
        "type": "ui_ui_control",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "events": "all",
        "x": 120,
        "y": 420,
        "wires": [
            [
                "c0cfd54d4bad4b3f"
            ]
        ]
    },
    {
        "id": "259efeaed660d9bc",
        "type": "moment",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "topic": "timeformatter",
        "input": "payload",
        "inputType": "msg",
        "inTz": "Asia/Kolkata",
        "adjAmount": "330",
        "adjType": "minutes",
        "adjDir": "add",
        "format": "",
        "locale": "en-US",
        "output": "payload",
        "outputType": "msg",
        "outTz": "Asia/Kolkata",
        "x": 830,
        "y": 420,
        "wires": [
            [
                "4b4accae74b5c63a"
            ]
        ]
    },
    {
        "id": "bd23ee3b066ed3bd",
        "type": "function",
        "z": "772ba0efeb30c3c6",
        "name": "timeinput",
        "func": "\nvar d=msg.payload.date;\nvar t = msg.payload.time;\nmsg.payload = {\n    \"_id\" : d +\" \"+t,\n    \"date\" : d,\n    \"time\" : t,\n    \"name\" : msg.payload.mname\n}\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 500,
        "y": 420,
        "wires": [
            [
                "7be0c5b989882cf5",
                "2d903e59b4169b40"
            ]
        ]
    },
    {
        "id": "ec7f450ec4d4d23b",
        "type": "debug",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 890,
        "y": 480,
        "wires": []
    },
    {
        "id": "4b4accae74b5c63a",
        "type": "function",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "func": "var time = msg.payload;\nvar d = new Date();\n//db time\nvar dbTime= d.getTime(time);\n//current time \nvar currentTime = d.getTime();\n\n/* time compare\nmsg.payload = {\n    'db': dbTime,\n    'current':currentTime\n}*/\n\n/*if ( dbTime == currentTime)\n{\n    msg.payload.action = on;\n}\n\nelse\n{\n    msg.payload.action = off;\n    msg.payload = \"Remainder is set successfully..!\";\n}*/\n\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 720,
        "y": 480,
        "wires": [
            [
                "81ca02962a59d0cc",
                "ec7f450ec4d4d23b"
            ]
        ]
    },
    {
        "id": "81ca02962a59d0cc",
        "type": "switch",
        "z": "772ba0efeb30c3c6",
        "name": "check",
        "property": "payload",
        "propertyType": "msg",
        "rules": [
            {
                "t": "eq",
                "v": "on",
                "vt": "str"
            }
        ],
        "checkall": "true",
        "repair": false,
        "outputs": 1,
        "x": 850,
        "y": 520,
        "wires": [
            [
                "adadc83750d91014"
            ]
        ]
    },
    {
        "id": "433f832726c64e56",
        "type": "eztimer",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "debug": false,
        "autoname": "dawn",
        "tag": "eztimer",
        "topic": "",
        "suspended": false,
        "sendEventsOnSuspend": false,
        "latLongSource": "manual",
        "latLongHaZone": "zone.home",
        "lat": "20.5937",
        "lon": "78.9629",
        "timerType": "2",
        "startupMessage": true,
        "ontype": "1",
        "ontimesun": "dawn",
        "ontimetod": "17:00",
        "onpropertytype": "msg",
        "onproperty": "payload",
        "onvaluetype": "num",
        "onvalue": "1",
        "onoffset": 0,
        "onrandomoffset": 0,
        "onsuppressrepeats": false,
        "offtype": "1",
        "offtimesun": "dusk",
        "offtimetod": "dusk",
        "offduration": "00:01:00",
        "offpropertytype": "msg",
        "offproperty": "payload",
        "offvaluetype": "num",
        "offvalue": 0,
        "offoffset": 0,
        "offrandomoffset": 0,
        "offsuppressrepeats": false,
        "resend": false,
        "resendInterval": "0s",
        "mon": true,
        "tue": true,
        "wed": true,
        "thu": true,
        "fri": true,
        "sat": true,
        "sun": true,
        "x": 650,
        "y": 600,
        "wires": [
            [
                "a72b31b145f77ceb"
            ]
        ]
    },
    {
        "id": "adadc83750d91014",
        "type": "function",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "func": "msg.payload =msg.payload.action;\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 480,
        "y": 580,
        "wires": [
            [
                "433f832726c64e56",
                "4e51b2cb766fdc5f"
            ]
        ]
    },
    {
        "id": "cc5c4fcb7d32be59",
        "type": "ibmiot out",
        "z": "772ba0efeb30c3c6",
        "authentication": "apiKey",
        "apiKey": "466840d3b27c292d",
        "outputType": "cmd",
        "deviceId": "12345",
        "deviceType": "2.2.2",
        "eventCommandType": "NodeMCU",
        "format": "json",
        "data": "payload",
        "qos": 0,
        "name": "IBM IoT",
        "service": "",
        "x": 460,
        "y": 340,
        "wires": []
    },
    {
        "id": "5e6d20adeb894548",
        "type": "ui_toast",
        "z": "772ba0efeb30c3c6",
        "position": "top right",
        "displayTime": "0.2",
        "highlight": "blue",
        "sendall": true,
        "outputs": 0,
        "ok": "OK",
        "cancel": "",
        "raw": false,
        "className": "",
        "topic": "Registered..!",
        "name": "",
        "x": 850,
        "y": 180,
        "wires": []
    },
    {
        "id": "b7394dc8ea40a894",
        "type": "ui_button",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "group": "2a06614fef005d29",
        "order": 0,
        "width": 0,
        "height": 0,
        "passthru": false,
        "label": "View My Account",
        "tooltip": "",
        "color": "",
        "bgcolor": "",
        "className": "",
        "icon": "",
        "payload": "WELCOME PAGE",
        "payloadType": "str",
        "topic": "topic",
        "topicType": "msg",
        "x": 110,
        "y": 240,
        "wires": [
            [
                "3519f78386229421"
            ]
        ]
    },
    {
        "id": "3519f78386229421",
        "type": "ui_ui_control",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "events": "all",
        "x": 140,
        "y": 180,
        "wires": [
            []
        ]
    },
    {
        "id": "4e51b2cb766fdc5f",
        "type": "trigger",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "op1": "",
        "op2": "It's time to take Sleep dose",
        "op1type": "nul",
        "op2type": "str",
        "duration": "2",
        "extend": false,
        "overrideDelay": false,
        "units": "min",
        "reset": "",
        "bytopic": "all",
        "topic": "topic",
        "outputs": 1,
        "x": 690,
        "y": 660,
        "wires": [
            [
                "4e810742a78cb1e9"
            ]
        ]
    },
    {
        "id": "7be0c5b989882cf5",
        "type": "function",
        "z": "772ba0efeb30c3c6",
        "name": "",
        "func": "msg.payload = msg.payload.time;\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 660,
        "y": 420,
        "wires": [
            [
                "259efeaed660d9bc"
            ]
        ]
    },
    {
        "id": "4d529e9ae98d1130",
        "type": "ui_spacer",
        "z": "772ba0efeb30c3c6",
        "name": "spacer",
        "group": "5f5120bb7e50aed4",
        "order": 1,
        "width": 1,
        "height": 1
    },
    {
        "id": "823fcb3a0551c8a0",
        "type": "ui_spacer",
        "z": "772ba0efeb30c3c6",
        "name": "spacer",
        "group": "5f5120bb7e50aed4",
        "order": 1,
        "width": 1,
        "height": 1
    },
    {
        "id": "20f632eba8223aa2",
        "type": "ui_spacer",
        "z": "772ba0efeb30c3c6",
        "name": "spacer",
        "group": "b39b0c959f2f9d67",
        "order": 2,
        "width": 1,
        "height": 1
    },
    {
        "id": "e6a3a03a5158e28b",
        "type": "ui_spacer",
        "z": "772ba0efeb30c3c6",
        "name": "spacer",
        "group": "61e6d4c4.e7619c",
        "order": 1,
        "width": 1,
        "height": 1
    },
    {
        "id": "2744c1e60d8165fc",
        "type": "ui_spacer",
    "z": "772ba0efeb30c3c6",
    "name": "spacer",
    "group": "bbd3a14fee1266d4",
    "order": 2,
    "width": "1",
    "height": "1"
  },
  {
    "id": "8bc329d3978b20cb",
    "type": "ui_spacer",
    "z": "772ba0efeb30c3c6",
    "name": "spacer",
    "group": "1b818abb6c71b0b2",
    "order": 2,
    "width": 1,
    "height": 1
  },
  {
    "id": "d35fb745bc134200",
    "type": "ui_spacer",
    "z": "772ba0efeb30c3c6",
    "name": "spacer",
    "group": "db51e34f28d951bc",
    "order": 3,
    "width": 1,
    "height": 1
  },
  {
    "id": "f5f4bc88a9c289aa",
    "type": "ui_spacer",
    "z": "772ba0efeb30c3c6",
    "name": "spacer",
    "group": "c65e35ff24555591",
    "order": 2,
    "width": 1,
    "height": 1
  },
  {
    "id": "fd3e168ce8bffda7",
    "type": "ui_spacer",
    "z": "772ba0efeb30c3c6",
    "name": "spacer",
    "group": "c8375fc927c1b4eb",
    "order": 2,
    "width": 1,
    "height": 1
    },
    {
        "id": "466840d3b27c292d",
        "type": "ibmiot",
        "name": "Watson",
        "keepalive": "60",
        "serverName": "",
        "cleansession": false,
        "appId": "",
        "shared": false
    },
    {
        "id": "c65e35ff24555591",
        "type": "ui_group",
        "name": "GENERAL PHYSICIAN",
        "tab": "3a65061d26bf9466",
        "order": 2,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""
    },
    {
        "id": "c8375fc927c1b4eb",
        "type": "ui_group",
        "name": "ORTHOPEDIST",
        "tab": "3a65061d26bf9466",
        "order": 3,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""
    },
    {
        "id": "b39b0c959f2f9d67",
        "type": "ui_group",
        "name": "DIABETIST",
        "tab": "3a65061d26bf9466",
        "order": 1,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""
    },
    {
        "id": "bbd3a14fee1266d4",
        "type": "ui_group",
        "name": "FOOD FOR YOU",
        "tab": "05a54edb77c2b875",
        "order": 1,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""
    },
    {
        "id": "db51e34f28d951bc",
        "type": "ui_group",
        "name": "FOOD FOR YOU",
        "tab": "05a54edb77c2b875",
        "order": 3,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""
    },
    {
        "id": "1b818abb6c71b0b2",
        "type": "ui_group",
        "name": "FOOD FOR YOU",
        "tab": "05a54edb77c2b875",
        "order": 2,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""
    },
    {
        "id": "4006abb73245ff99",
        "type": "ui_group",
        "name": "New user Or Old user",
        "tab": "2863b8bb54864aa3",
        "order": 1,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""
    },
    {
        "id": "876611f251aa851a",
        "type": "ui_group",
        "name": "Login",
    "tab": "1f949d50c69b978f",
    "order": 1,
    "disp": true,
    "width": "6",
    "collapse": false,
    "className": ""
  },
  {
    "id": "6eae6ed0cbde419d",
    "type": "ui_group",
    "name": "User Details",
    "tab": "fe87a3aec4690274",
    "order": 2,
    "disp": true,
    "width": "6",
    "collapse": false,
    "className": ""
  },
  {
    "id": "5f5120bb7e50aed4",
    "type": "ui_group",
    "name": "Medicine Storage",
    "tab": "5a2367a8a72352fd",
    "order": 1,
    "disp": true,
    "width": "6",
    "collapse": false,
    "className": ""
  },
  {
    "id": "2a06614fef005d29",
    "type": "ui_group",
    "name": "My account",
    "tab": "054b3b226a06a861",
    "order": 1,
    "disp": true,
    "width": "6",
    "collapse": false,
    "className": ""
  },
  {
    "id": "61e6d4c4.e7619c",
    "type": "ui_group",
    "name": "TEST",
    "tab": "",
    "order": 1,
            "disp": true,
            "width": "5",
            "collapse": false
        },
        {
            "id": "3a65061d26bf9466",
            "type": "ui_tab",
            "name": "CONSULTANTS",
            "icon": "dashboard",
            "order": 6,
            "disabled": false,
            "hidden": false
        },
        {
            "id": "05a54edb77c2b875",
            "type": "ui_tab",
            "name": "WHAT TO EAT??",
            "icon": "dashboard",
            "order": 7,
            "disabled": false,
            "hidden": false
        },
        {
            "id": "2863b8bb54864aa3",
            "type": "ui_tab",
            "name": "WELCOME PAGE",
            "icon": "dashboard",
            "order": 2,
            "disabled": false,
            "hidden": false
        },
        {
            "id": "1f949d50c69b978f",
            "type": "ui_tab",
            "name": "LOGIN",
            "icon": "dashboard",
            "order": 4,
            "disabled": false,
            "hidden": true
        },
        {
            "id": "fe87a3aec4690274",
            "type": "ui_tab",
            "name": "REGISTER",
            "icon": "dashboard",
            "order": 3,
        "disabled": false,
        "hidden": true
    },
    {
        "id": "5a2367a8a72352fd",
        "type": "ui_tab",
        "name": "HOME",
        "icon": "dashboard",
        "order": 5,
        "disabled": false,
        "hidden": true
    },
    {
        "id": "054b3b226a06a861",
        "type": "ui_tab",
        "name": "ACCOUNT",
        "icon": "dashboard",
        "order": 1,
        "disabled": false,
        "hidden": false
    }
]
```

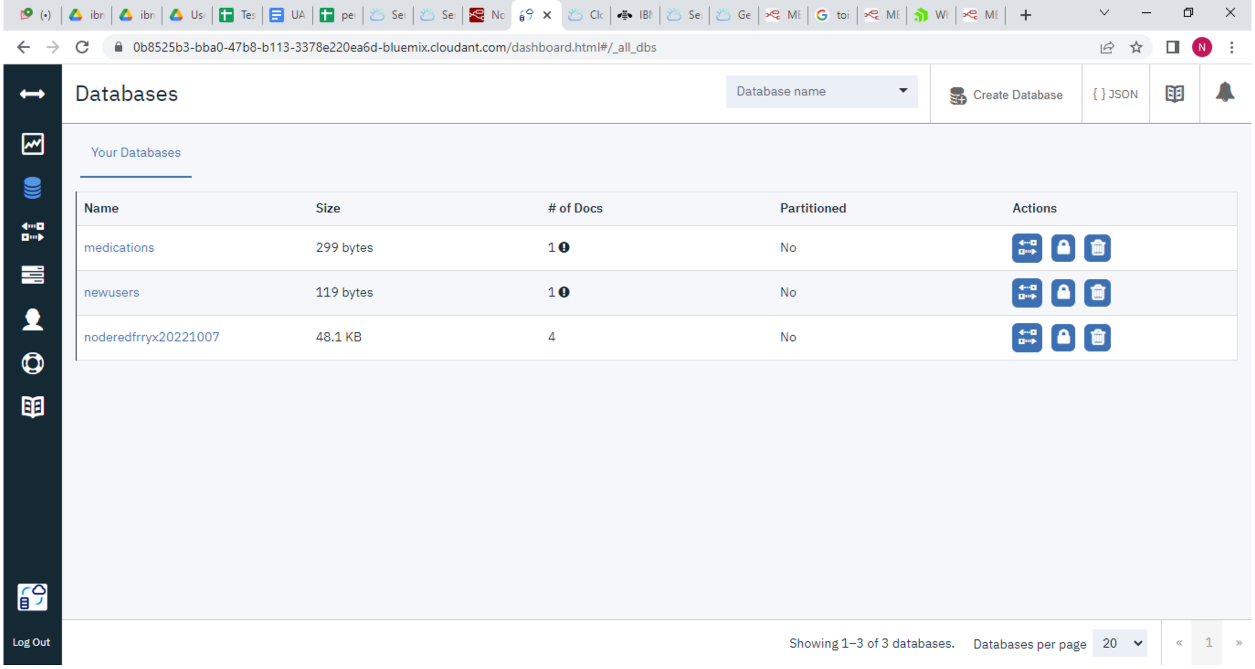
7.3 Database Schema

We used two databases to store the details that we get from the users

Database - "newusers" is used to store the details of new users.

Database - "medications" is used to store the medication details entered by the users.

The details entered by the users will get stored in the json format in both the databases.



0b8525b3-bba0-47b8-b113-3378e220ea6d-bluemix.cloudant.com/dashboard.html#/_all_dbs

Databases

Name	Size	# of Docs	Partitioned	Actions
medications	299 bytes	1 ❗	No	
newusers	119 bytes	1 ❗	No	
noderedfrryx20221007	48.1 KB	4	No	

Showing 1–3 of 3 databases. Databases per page 20 « 1 »

Fig 7.3.1 Database schema

8. TESTING

8.1 Test Cases

Feature type	Component	Test scenario	Test data	Expected result	Actual result	Status
Functional	Welcome page	user is able to see the Login and Signup popup when enters url.	https://node-red-frryx-2022-10-07.eu-gb.mybluemix.net/ui/	Login and Signup popup should display	Working as expected	Pass
UI	Home page	Verify the UI elements in Login/Signup popup	https://node-red-frryx-2022-10-07.eu-gb.mybluemix.net/ui/	Application should show below UI elements: a.email text box b.password text c.Login button with orange color d.New customer? Create acc link e.Last password? Recovery password link	Working as expected	Pass
Functional	Home page	Verify user is able to log into app with Valid credentials	Username: Krish Email:Krish@gmail.com Mobile number:7846120659 password: Testing123	User should navigate to user account homepage	The user navigated to Home page.	Pass
Functional	Login page	Verify user is able to log into app with InValid credentials	Username: Krish Email:Krish@gmail.com Mobile number:7846120659 password: Test	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass

Functional	Login page	Verify user is able to log into app with InValid credentials	Username: Krish Email:Krish@gmail.com Mobile number:7846120659 password: Test	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass

Table 8.1.1 Test cases

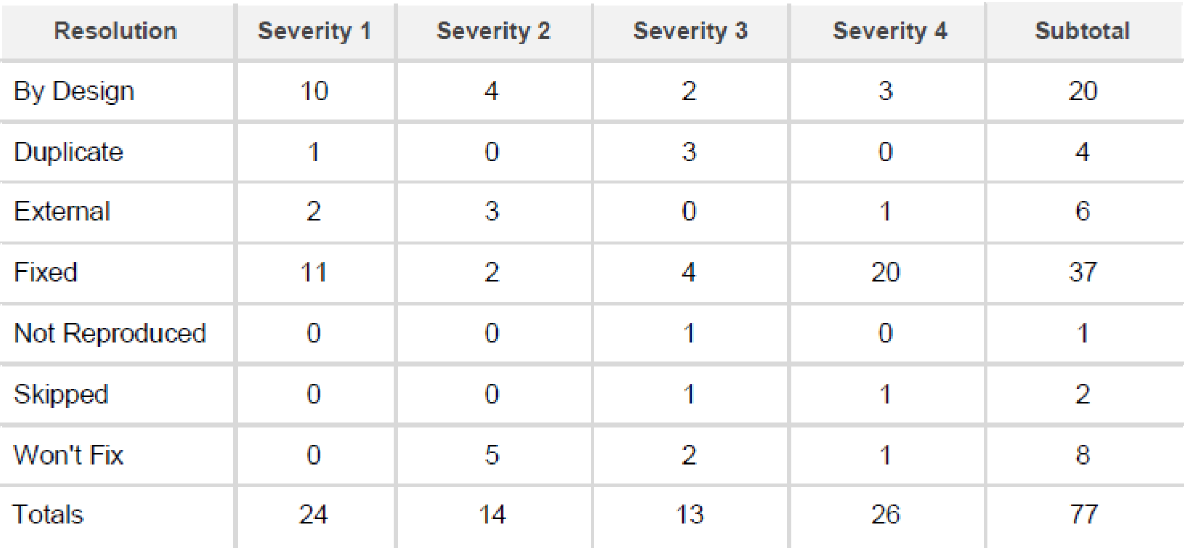
8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

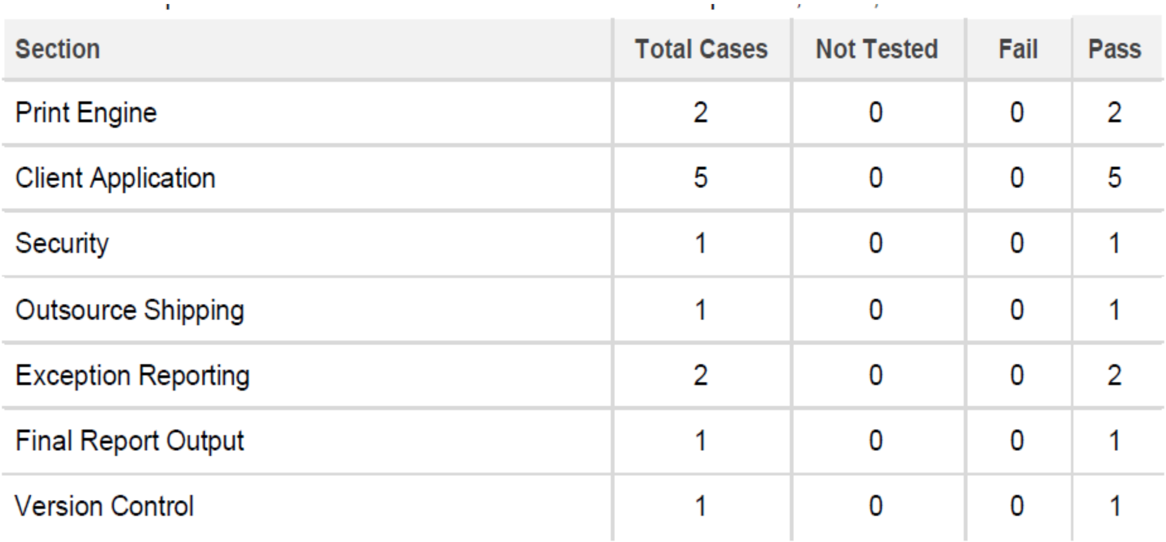


Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

Table 8.2.1 Defect analysis

3. Test Case Analysis

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



Section	Total Cases	Not Tested	Fail	Pass
Print Engine	2	0	0	2
Client Application	5	0	0	5
Security	1	0	0	1
Outsource Shipping	1	0	0	1
Exception Reporting	2	0	0	2
Final Report Output	1	0	0	1
Version Control	1	0	0	1

Table 8.2.2 Testcase analysis

9. RESULTS

This medication reminder system serves reliable reminders, has a good and easy to use user interface and supports a lot of features adhering to medicines. The details are not at all confusing and can be easily understood by the user. The best part of the application is that the details only have to be entered one time. On submitting the details once, the data is synced on all the user’s devices on which he/she is logged in. This allows for easy reminders no matter what device the user is using. The reviews on the system are overall positive and it addresses most of the flaws in the current reminder systems.

9.1 PERFORMANCE METRICS

NFT Risk Assessment

Project name	Scope/ feature	Functional changes	Hardware changes	Software changes	Impact of downtime	Load/ volume change	Risk score
Personal assistance for people who are self reliant-medication intimator	new	low	no changes	moderate	low	>5 to 10%	orange

Table 9.1.1 - NFT Risk Assessment

NFT-Detailed test plan

Project overview	NFT test approach	Assumptions/Dependencies/Risks	Approvals/signoff
Medication intimator-UI	Stress	App crash/developer team/site down	Approved
Medication intimator-UI	Endurance	App crash/site down	Approved
Medication intimator-UI	Load	Server crash/developer team/server down	Approved

Table 9.1.2 - NFT- detailed test plan

End of test report

Project overview	NFT test approach	NFR-met	Test outcome	GO/ NO-GO decision	recommend	Defects	Approval/ signoff
Medication intimator-UI	Stress	Performance	CPU-01	GO	High performance server	closed	approved
Medication intimator-UI	Load	Scalability	DB-STORAGE -01	NO-GO	IBM cloud	closed	approved
Medication intimator-UI	Endurance	Connectivity	connection	GO	High performance	closed	approved

Table 9.1.3 - End of test report

10. ADVANTAGES AND DISADVANTAGES

Advantages

1. When setting up medication reminders, users can customize their schedule. For example, users can set reminders for morning medications at a later weekends than on weekdays.

2. This medication reminder system is simple and easy to use.

3. With this app, even illiterates can use the available features and can take their medications at the right time due to text to speech service.

4. An unlimited number of medication reminders

5. Flexible scheduling for daily, weekly, or monthly dosa

Disadvantages

1. The user updates the medicines to be taken while using this medication reminder app, but the limitation is only single medicine detail can be uploaded as of now.

2.Registration notification cannot be sent to the user

11. CONCLUSION

The medication reminder system is a useful resource for those who need technological help in completing or need help in working through day-to-day tasks and taking care of their health. It is a smart and organized system that is designed to help the elderly people in our homes, but we have not put any restrictions that stop an everyday user from using the system. Anyone can need medical attention and normal people forget to take their prescriptions as well. The system will help them out in regulating their medications. It can also help a working person with a busy schedule by sending him a notification on the device he uses full day, his laptop. Thus, there is no restriction on the user base for our system.

12. FUTURE SCOPE

For future work we have decided to add a confirmation from the user whether they have taken the medicine or not. If they haven’t taken the medicine the information will be sent to the doctor and he can reschedule the further medicine reminders according to the new schedule. Overall, this is a much-needed system by the elderly people in our homes. It helps them in taking their medications in the prescribed quantity and at the prescribed time.

13. APPENDIX

13.1 SOURCE CODE

PYTHON CODE FOR CONNECTING WATSON IOT DEVICE

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

#Provide your IBM Watson Device Credentials
organization = "4htmhw"
deviceType = "Tester"
deviceId = "12345"
authMethod = "token"
authToken = "123456789"

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

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()
```

PYTHON CODE FOR TEXT TO SPEECH SERVICE

```
 import json
from ibm_watson import ApiException
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

def tts():
authenticator =
IAMAuthenticator('3UpxSyoSB7FmzX550dxWtYC5WiTZzPXpqVvhPbUjkDeL')
    text_to_speech = TextToSpeechV1(authenticator=authenticator)

text_to_speech.set_service_url('https://api.us-east.text-to speech.watson.cloud.ibm.com')

#to disable SSL verification with a service managed on IBM Cloud
text_to_speech.set_disable_ssl_verification(True)
```

JAVASCRIPT CODE FOR TIME COMPARISON

```
var dtime = msg.payload;
var ctime = new Date().toLocaleString("en-US", {timeZone: "Asia/Kolkata"});

var d = new Date(dtime);
var db_hour = d.getUTCHours();
var db_minutes = d.getUTCMinutes();

var c = new Date(ctime);
var c_hour = c.getUTCHours();
var c_minutes = c.getUTCMinutes();

if ( db_hour == c_hour && db_minutes == c_minutes)
{
  msg.payload =
   {
        "status":"on",
        "db":db_hour+":"+db_minutes,
        "current":c_hour+":"+c_minutes
   }
}

else
{
   msg.payload =
   {
       "status":"off",
       "db":db_hour+":"+db_minutes,
     "current":c_hour+":"+c_minutes
   }
}

return msg;
```

13.2 Github and Project demo link

Github link - https://github.com/IBM-EPBL/IBM-Project-16691-1659620261

Project demo link - https://vimeo.com/772834835/f6da3ce413