

SELF-RELIANT

(Using IoT)

A PROJECT REPORT

Submitted by

SAROJA B **311419104069**

SHALINI S **311419104072**

SHOBICA A **311419104073**

YUVATI MALAVIKAA V.S **311419104099**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

MEENAKSHI COLLEGE OF ENGINEERING, WEST K.K NAGAR

ANNA UNIVERSITY: CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report **“PERSONAL ASSISTANCE FOR SENIORS WHO ARE SELF-RELIANT”** is the bonofide work of **“SAROJA B (311419104068), SHALINI S (311419104072), SHOBICA A (311419104073), YUVATI MALAVIKAA (311419104099)”** who carried out the project work under my supervision.

INDEX

S.NO	TITLE
1.	INTRODUCTION
	1.1 Project Overview
	1.2 Purpose
2.	LITERATURE SURVEY
	2.1 Existing System
	2.2 References
	2.3 Problem Statement Definition
3.	IDEATION & PROPOSED SOLUTION
	3.1 Empathy Map Canvas
	3.2 Ideation & Brainstorming
	3.3 Proposed Solution
	3.4 Problem Solution Fit
4.	REQUIREMENT ANALYSIS
	4.1 Functional Requirement
	4.2 Non-Functional Requirement
5.	PROJECT DESIGN
	5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Burndown Chart

7. CODING & SOLUTIONING

7.1 Feature 1

7.2 Feature 2

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADAVANTAGES & DISADVANTAGES

11.CONCULSION

12.FUTURE SCOPE

13.APPENDIX

Source Code

GitHub &Project Demo Link

LIST OF FIGURES

FIG.NO	FIG.NAME
3.1.1	Empathy map canvas
3.2.1	Brain Storming
5.1.1	Data flow diagrams
5.2.1	Technical Architecture
5.3.1	User Stories
6.1.1	Sprint Planning & Estimation
6.2.1	Sprint Delivery Schedule
6.3.1	Burndown Chart

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 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

In modern society, most of the time people remain busy in their daily life schedule. It is true that they give more preference to their work than taking care of their health. Several diseases like diabetes, blood pressure is nowadays very common. Maintaining daily medication become very difficult for old people. Sometimes younger is faced with the same problem. There are many people in our family who need constant help may t be our elderly people, younger or others. But it is not

always possible for us to remind them of their medicine's dosages every time. For this purpose, there needs to be some facility for us which monitoring patient and take care. Nowadays we are all used to living technology-based life. We can use this technology in a way that will be beneficial for us. Cell phones aren't best utilized for calling but now maybe used as an ensemble of embedded sensors that together allow new packages including human services, healthcare, social networks, environmental tracking etc. Today in medical services frameworks, the usage of cell phones is turning into an expanding number of values [1]. IoT may be helpful to monitor real-time condition and IoT can be a powerful and effective paradigm to store data collected by sensors devices to the cloud. In our project, the IoT enabled device will control the overall monitoring system. And developed and android application which help patients by reminding medicine in take time and so on.

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

Patients may often fail to comply with their medication whether it was from forgetting to take the medicine, from taking medicine at the wrong time or even time taking too much medicine. 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. Although the Internet of Things can be of great benefits to healthcare, there are still major challenges to address before full-scale implementation. The threats and disadvantages of using connected devices in healthcare are as follows:

- 1. Security and Privacy:** Security and privacy remain a major concern deterring users from using IoT technology for medical purposes, as healthcare monitoring solutions have the potential to be breached or hacked. The leak of sensitive information about the patient's health and location and meddling with sensor data can have grave consequences, which would counter the benefits of IoT.
- 2. Risk of failure:** Failure or bugs in the hardware or even power failure can impact the performance of sensors and connected equipment placing

healthcare operation at risk. In addition, skipping a scheduled software update may be even more hazardous than skipping a doctor checkup.

- 3. Integration:** There's no consensus regarding IoT protocols and standards, so devices produced by different manufacturers may not work well together. The lack of uniformity prevents full-scale integration of IoT, therefore limiting its potential effectiveness.
- 4. Cost:** While IoT promises to reduce the cost of healthcare in the long-term, the cost of its implementation in hospitals and staff training is quite high.

2.2 REFERENCE

1. Tahaseen Hasrath, P. Sowmika, N. Rajendra, R. Nishma and U. Gowthami, "Automatic Medicine Dispenser", Journal of Emerging Technologies and Innovative Research (JETIR), vol. 8, no. 6, pp. e317-e323, June 2021.
2. D MohanaPriya, V Deepika, M ShanmughaPriya and Yogeswari C Sivasankari, "A Real Time Support System to Impact Medicine using smart Dispenser", IEEE ICSCAN, 2020.
3. Wissam Antoun, Ali Abdo, Suleiman Al-Yaman Abdallah Kassem, Mustapha Hamad and Chady El-Moucary, "Smart Medicine Dispenser (SMD)", 208 IEEE pp.20-23, 19 March 2019.

4. Animesh Kumar Sahu, Rohit Roy, A. Jabeena and N. Sardar Basha, “Automatic Pill Remainder for Easy Supervision”, ICISS-2017 (IEEE Conference) School of Electronics Engineering VIT University ,pp. 98-102,2017

2.3 PROBLEM STATEMENT DEFENITION

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

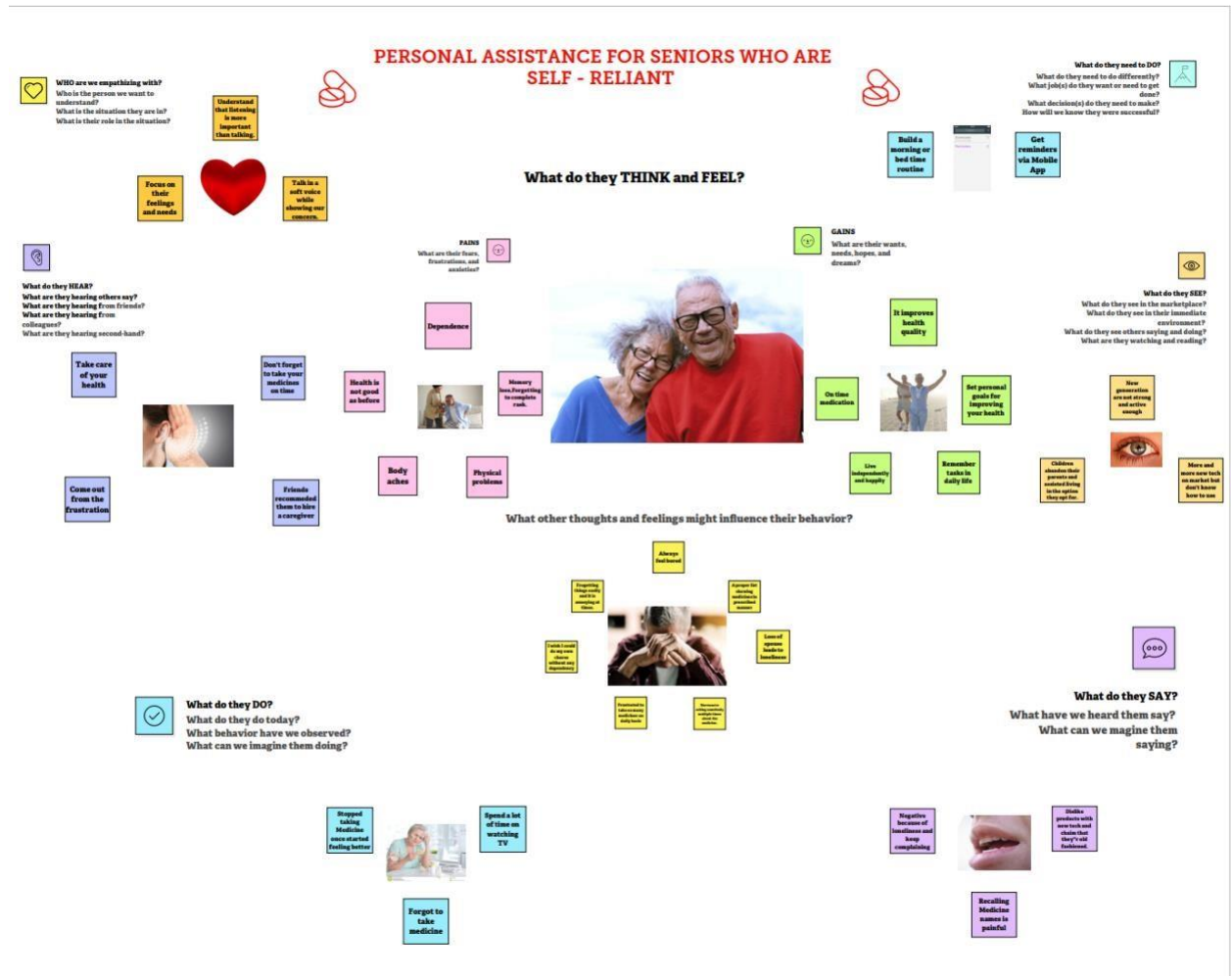
Customer Problem Statement Template:

MEDICINE REMINDER



Problem Statement (PS)	I am (User)	I am trying to	But	Because	Which makes me feel
PS-1	User	Contact someone in case of emergency	Unable to contact them at right time	No one is around me to contact someone	Helpless
PS-2	User	Take medicines on right time	Unable to take medicines on right time	Due to age factors I am forgetting to take medicines on right time	Dependent

3.1 EMPATHY MAP CANVAS




3.2 IDEATION & BRAINSTORMING

Brainstorm & Idea Prioritization Template:

Team Gathering, Collaboration and Select the Problem Statement

Template



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

⌚ 10 minutes to prepare
🕒 1 hour to collaborate
👥 2-8 people recommended

[Share template feedback](#)

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

⌚ 10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#) →

1 Define your problem statement


What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

⌚ 5 minutes

PROBLEM
How might we [your problem statement]?

Key rules of brainstorming
To run a smooth and productive session

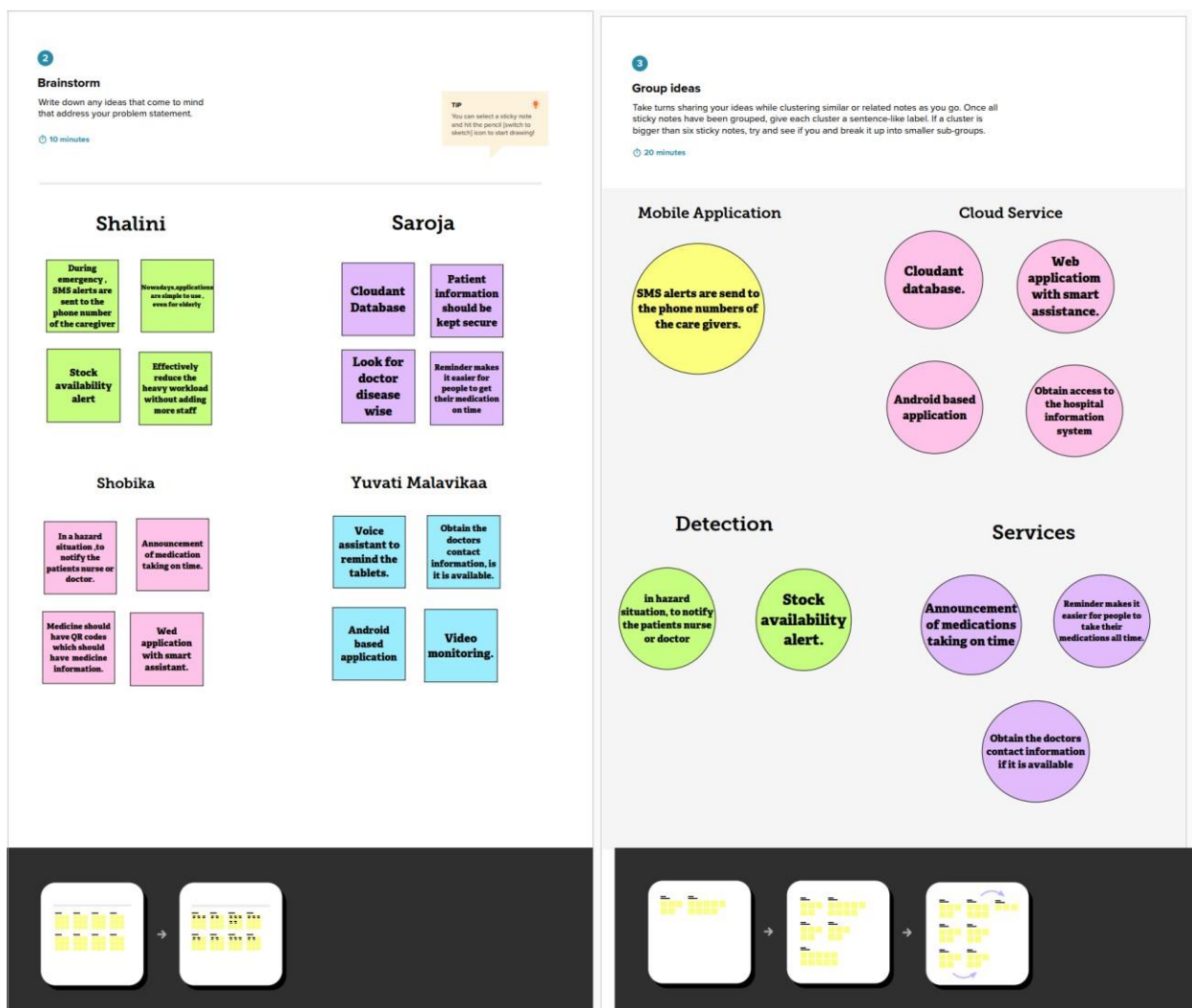
🗨️ Stay in topic.	💡 Encourage wild ideas.
🚫 Defer judgment.	👂 Listen to others.
🗣️ Go for volume.	👁️ If possible, be visual.



Need some inspiration?
See a featured version of this template to kickstart your work.

[Open example](#) →

Brainstorm, Idea Listing and Grouping



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⌚ 20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Ideas placed on the grid:

- Top-Left (High Importance, Low Feasibility):** Voice assistance to remind the tablets.
- Top-Right (High Importance, High Feasibility):** In hazard situation to notify the patient nurse or doctors. (Blue box), Reminder makes it easier for people to take their on time (Green box), Cloudant database (Green box), SMS alerts are sent to the phone number of the caregiver (Green box).
- Bottom-Left (Low Importance, Low Feasibility):** Look for doctors disease wise (Yellow box), Video monitoring (Yellow box).
- Bottom-Right (Low Importance, High Feasibility):** Stock availability alert (Blue box), Android based application. (Blue box), Announcement of medication taking on time. (Blue box), Medicine should have QR codes which should have medicine information. (Yellow box).

➔

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

- A Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
- B Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

- Strategy blueprint**
Define the components of a new idea or strategy.
[Open the template →](#)
- Customer experience journey map**
Understand customer needs, motivations, and obstacles for an experience.
[Open the template →](#)
- Strengths, weaknesses, opportunities & threats**
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
[Open the template →](#)




[Share template feedback](#)




3.3 PROPOSED SOLUTION TEMPLATE

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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.	Idea / Solution description	An app is built for the user (caretaker) which enables him/her 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.
3.	Novelty / Uniqueness	The device provides multiple ways of reminding to the patient to take medicine. Gives stock alert when medicine is low. For

		security purpose, emergency alert through SMS to the caregiver/family member is added.
4.	Social Impact / Customer Satisfaction	Our system will remind the user at the prescribed time of which medicine is to be taken in the form of mobile notification. Customers (Users) are satisfied because when they intake their medications on correct time they feel healthy.
5.	Business Model (Revenue Model)	Through are web application the revenue can be made in the form of pop-up advertisements overlay ads from third party services.
6.	Scalability of the Solution	Large number of people can be supplied with the wearable devices to ensure their safety. They can easily set their medication time in the web application

3.4 PROBLEM SOLUTION FIT

Define CS, fit into CC	1. CUSTOMER SEGMENT(S)  <p>Citizens who are in need of external support to take care of themselves for medical assistance .</p>	6. CUSTOMER CONSTRAINTS  <p>It should be present near to them. Knowing the process of using the application. Registered users can use the application</p>	5. AVAILABLE SOLUTIONS  <p>If user forget to take medicine, then the application will help them to take medicine at right time. Alerts the users by SMS notification.</p>	Explore AS, differentiate

Focus on J&P, tap into BE, understand RC	2. JOBS-TO-BE-DONE / PROBLEMS  <p>This application helps the patient to remind medicine through voice assistance. It helps the user to do their daily routine without seeking help from other people.</p>	9. PROBLEM ROOT CAUSE  <p>Side-effects affecting thinking and balance. Doctors cannot monitor the patients all the time</p>	7. BEHAVIOUR  <p>The patient need to update the information about their medication, life routines to the application.</p>	Focus on J&P, tap into BE, understand RC

Problem Solution Fit

Identify strong TR & EM	<p>3. TRIGGERS</p> <p>R</p> <p>People simply forget, skip or stop taking their medications which leads to non-adherence. Triggers helps people to integrate healthy behaviour by using technology in a very simple way.</p>	<p>10. YOUR SOLUTION</p> <p>L</p> <p>Notifying of medicines names through messages with the help of data fed from the mobile application which is initiated by web application which stores the user details.</p>	<p>8. CHANNELS</p> <p>of BEHAVIOUR</p> <p>C</p> <p>H</p> <p>ONLINE</p> <p>Users can set reminder about their medicines in online mode.</p> <p>OFFLINE</p> <p>Users get notification alert to take medications on right time in offline mode</p>
	<p>4. EMOTIONS:</p> <p>BEFORE / AFTER</p> <p>E</p> <p>Despite effective treatments, depression</p>		

may often un
recognize and
untreated. 2,3 and
many persons in the
community with
depression see a
general physician so
primary care setting is
pivotal when
considering how to
optimize the treatment
for depression and
other forms of
emotional distress in
the community

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through form. Registration g-mail.
FR-2	User Confirmation	Confirmation via e-mail. Confirmation via OTP.
FR-3	User Login	Login through registered mail ID & Password. Login through Mobile number & Password.
FR-4	Users medical information	Enter your Medicine details with date.
FR-5	Access Cloud Services	Access the cloud services with correct credit details store the details in the database. Retrieve the needed information for the user's operation.
FR-6	IOT Configuration	Access the cloud database via device manage the request & response effectively.

4.2 NON-FUNCTIONAL REQUIREMENTS

Following are the non-functional requirements of the proposed solution.

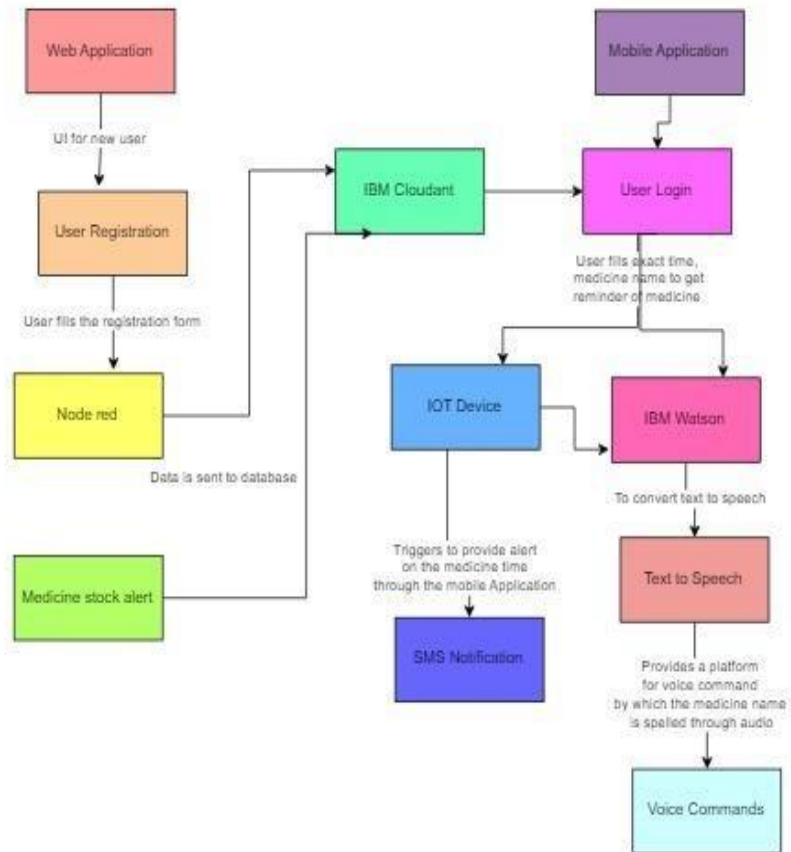
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system should be user friendly for the user's. It is used to remind the medicine names. It alerts the users through voice commands.
NFR-2	Security	The large information & the data of the users should be kept confidential.
NFR-3	Reliability	It is trust worthy because it uses trusted cloud service like IBM.
NFR-4	Performance	The voice command will be delivered exactly at the time. It is better compared to the market product.
NFR-5	Availability	Available on the mobile. It should be monitored 24*7 for the alert of the medicine.
NFR-6	Scalability	It is adaptable. Using cloud service makes the scalable higher than using traditional database.

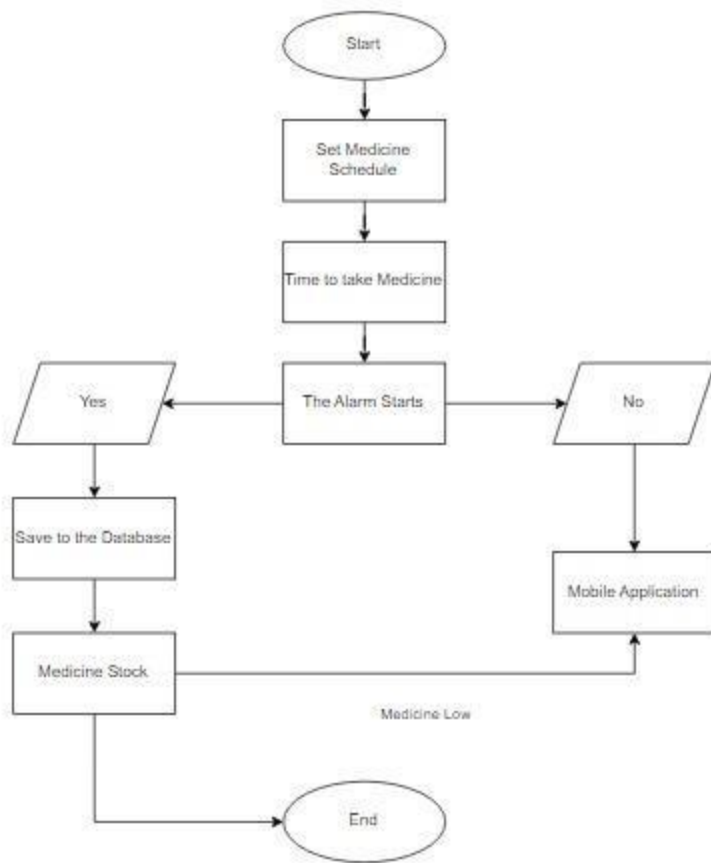
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

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

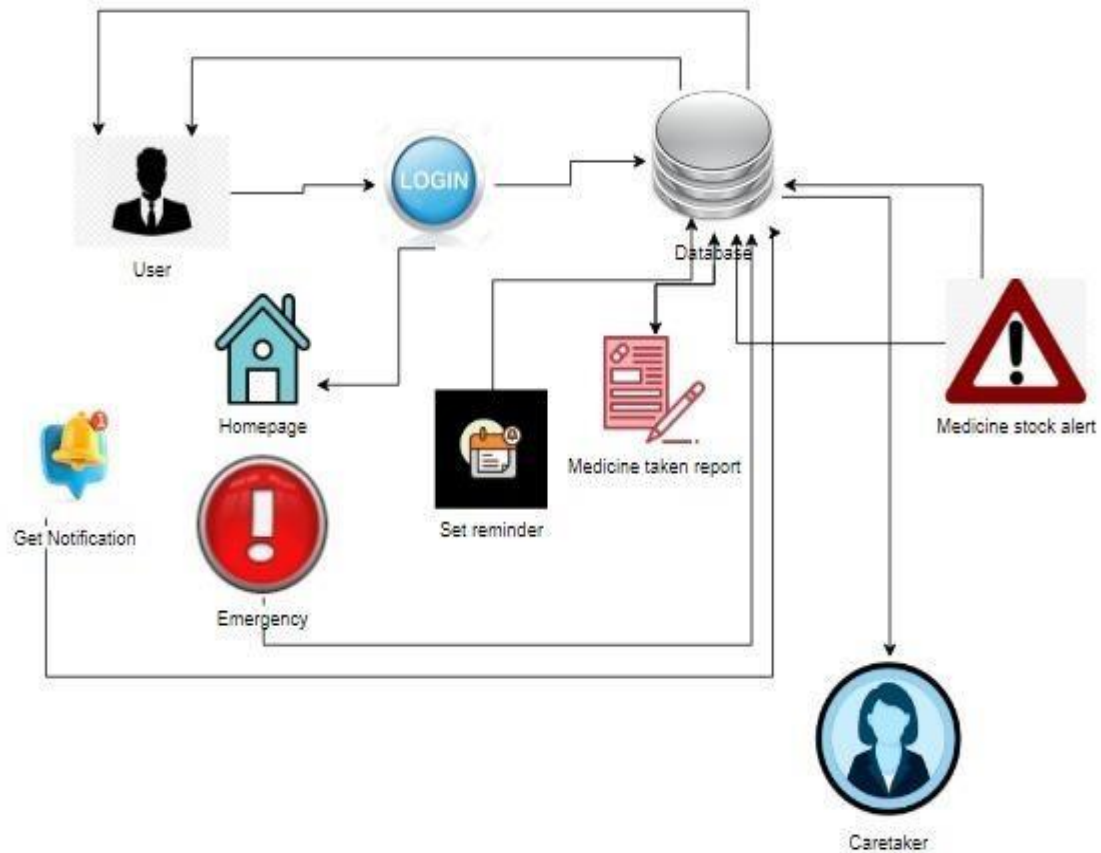
- The user should register on Medicare (web application) by using their mail ID, password and get confirmed.
- The user can set medicine name and time by med care.
- The SMS is sent to the user to notify the intake of medicine.
- The data given as input by the user on mobile application is stored in IBM Cloudant database.



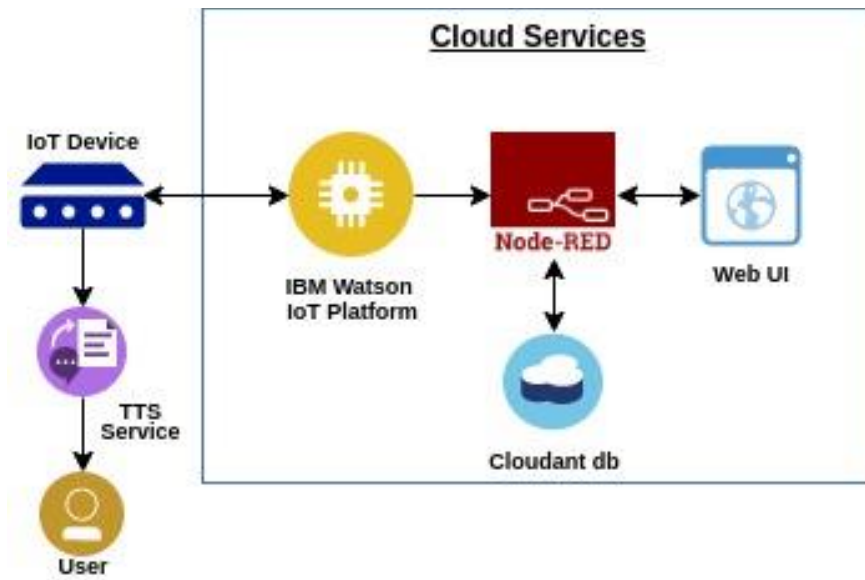


5.2 SOLUTION & TECHNICAL ARCHITECTURE

Solution Architecture Diagram:



Technical Architecture Diagram:



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User	Registration	USN - 1	As a user, I can register for the application by entering my email, password and confirming my password	I can access my account	High	Sprint - 1
User	Confirmation	USN - 2	As a user I will receive confirmation mail once I have	I can receive confirmation and click confirm	High	Sprint - 1

			registered for the application			
User (Administration)	Login	USN - 3	As a user, I can log into my application by entering email & password	I can access the app and save my medical details at any time	High	Sprint - 1
User (Senior Citizen)	Care taker	USN - 4	As a user I want to take medicines on time and monitor my health	I want to take medicines on time	High	Sprint - 1
	Smart medicine box	USN - 5	As a user, I want to take medicine on right time via SMS	I need to take medicines at exact time by SMS notification	High	Sprint - 2
	Digital med care	USN - 6	As a user, I want to take medicine on time by voice commands	I need to take medicines on time by voice commands	Medium	Sprint - 3
	Virtual medicine kit	USN - 7	As a user, my patient medication time & name should be loaded in Database	My patients medicine details should be stored in database	High	Sprint - 3
		USN - 8	As a user I want to know the stock alert of my medicines before it gets over	I need to know the stock alert	Low	Sprint - 4

User	Notification	USN - 9	As a user I should able to notify my caretaker at emergency situation	I can able to receive the notification at right time	Medium	Sprint - 4
------	--------------	---------	-----------------------------------------------------------------------	------------------------------------------------------	--------	------------

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	Registration	USN-1	As a user, I can register for the application by Entering my email, and password, and confirming my password.	5	High	Shalini S
Sprint-1	Confirmation Email	USN-2	As a user, I will receive a confirmation email once I have registered for the application	6	High	Shalini S Saroja B

Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	4	High	Shalini S Shobika A
Sprint-1		USN-4	As a user I want to take medicines on time and monitor my health	6	High	Shalini S Yuvati malavikaa VS
Sprint-2	Dashboard	USN-5	As a user I want to take medicines on time via SMS	9	High	Shalini S Saroja B

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Node RED - Cloudant DB communication	USN-6	As a user my patient medication time and database should be loaded in database	13	High	Shalini S, Shobika A
Sprint-3	Text to speech	USN-7	As a user I need to take medicines on time by voice commands	7	Medium	Shalini S Shobika A

Sprint-4	Cloudant DB communication	USN-8	As a user I want to know the stock alert of my medicines before it gets over	6	Low	Shalini S, Yuvati malavikaa VS
Sprint-2	User– Web UI interface	USN-9	The Web UI should get inputs from the user	9	High	Shalini S Saroja B
Sprint-4	Alarm	USN-10Shalini	The Alarm of the remainder should be done based on the medication time	11	High	Shalini S Yuvati malavikaa VS

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	21	6 Days	24 Oct 2022	29 Oct 2022	21	29 Oct 2022
Sprint-2	18	6 Days	31 Oct 2022	05 Nov 2022	18	05 Nov 2022
Sprint-3	20	6 Days	07 Nov2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	17	6 Days	14 Nov2022	19 Nov 2022	17	19 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)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

Average Velocity for Sprint 1 => **21/6 = 3.5 => 3 Story points per day**

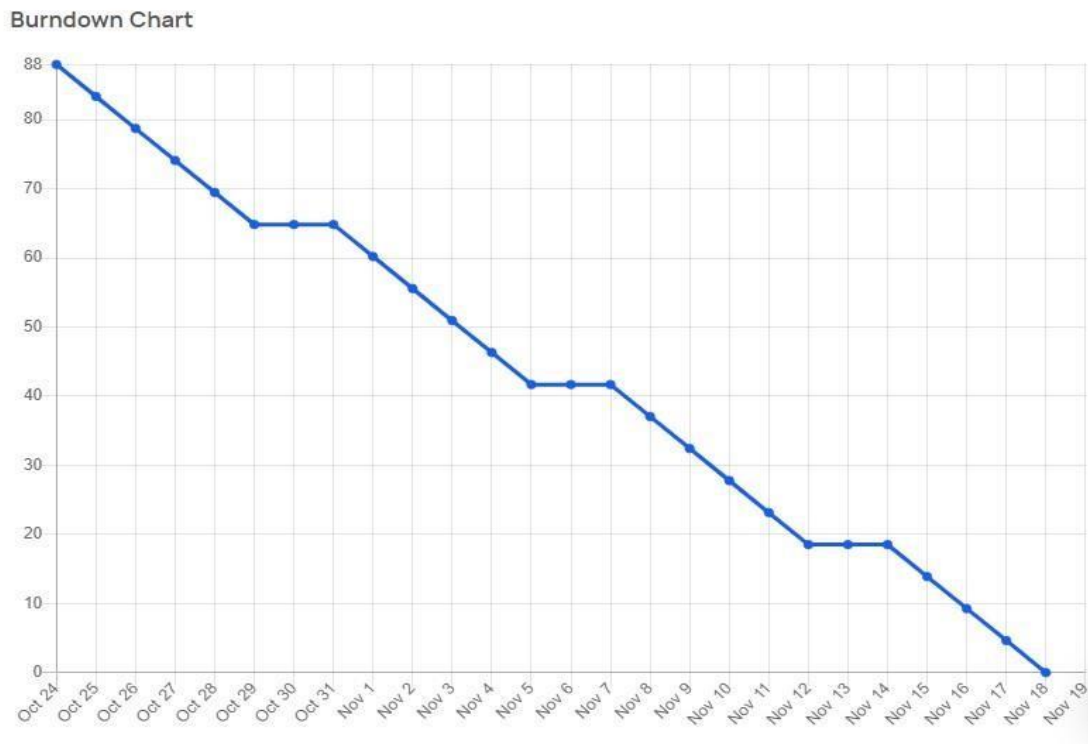
Average Velocity for Sprint 2 => **18/6 = 3 => 3 Story points per day**

Average Velocity for Sprint 3 => **20/6 = 3.3 => 3 Story points per day**

Average Velocity for Sprint 4 => **17/6 = 2.87 => 2 Story points per day**

6.3 BURNDOWN CHART

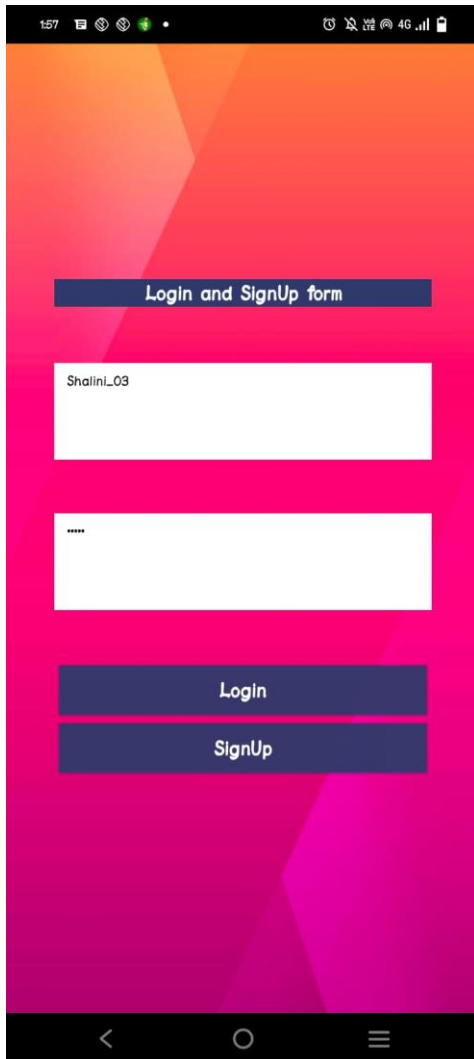
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



7. CODING AND SOLUTIONING

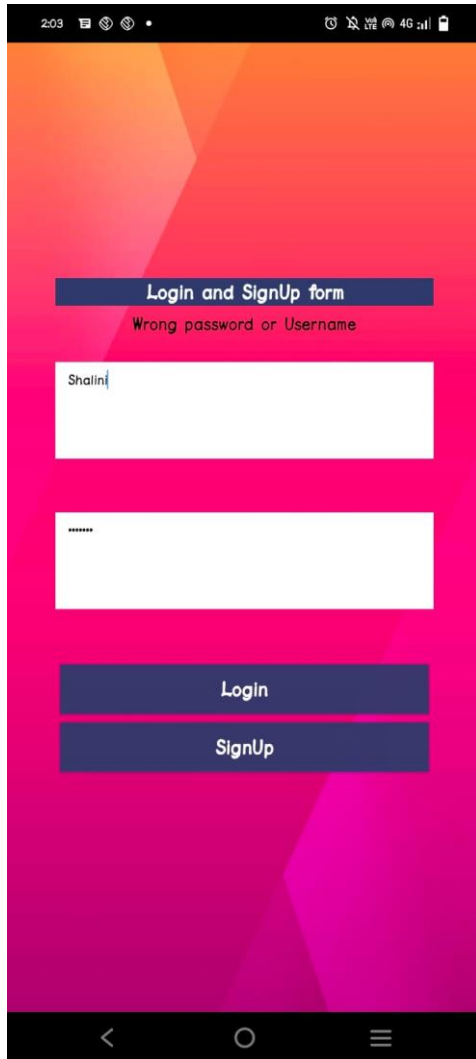
7.1 FEATURE 1:

The mobile application developed as a feature of individual login by different user.



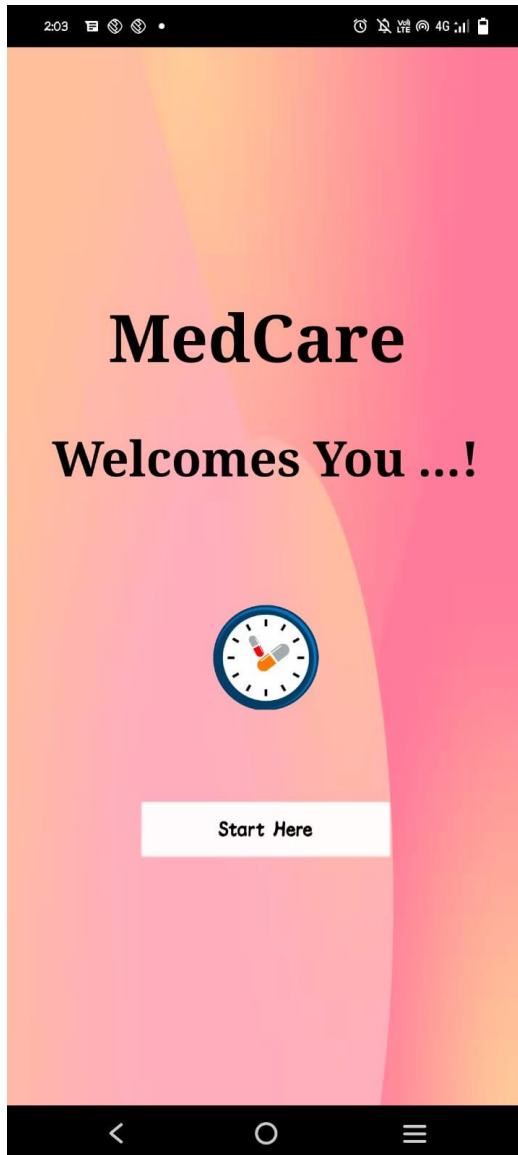
7.2 FEATURE 2:

User cannot login with Incorrect password.




7.3 FEATURE 3:

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



2:03

4G



In Each Hour

No of times

Alert Message

Set Reminder

<

○

≡

8. TESTING

8.1 TESTING

TESTCASES	PRE- CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT
Verify login with valid credential.	User should have a network connection.	1.Lunch URL. 2. Enter valid user name. 3. Enter valid password. 4. Click on login button.	User name: Shalini Password: 12345	User should be able to login successfully.
Verify login with invalid credential.	User should have a network connection.	1. Lunch URL. 2. Enter valid user name. 3. Enter invalid password. 4. Click on login button.	User name: Shalini Password: 1234567	User should not be able to login.

Update the medicine detail with time.	User should have a network connection.	1. Enter medicine detail. 2. Enter the time when the medicine have to be consumed. 3. Enter the date when the medicine have to be consumed. 4. Click on “Set Reminder”.	Medicine name: Dolo650 Time:17.00 Date:23-11-2022	User should be able to login successfully.
---------------------------------------	----------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------	--------------------------------------------

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 [Elderly people] 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	Sub Total
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 fixed	1	3	2	2	8
Totals	13	11	15	25	64

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
Login Page	5	0	0	5
Register	3	0	0	3
Home page	2	0	0	2
Node RED Dashboard	32	0	0	32
IBM Watson IOT Platform	2	0	0	2
MIT app inventor	3	0	0	3
Text To Speech	3	0	0	3

9. RESULT

9.1 PERFORMANCE METRICS

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

NFT - Detailed Test Plan								
S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff				
1	Medicine Reminder Web -UI	Stress	App Crash/ Developer team/ Site Down	Approved				
2	Medicine Reminder Web -UI	Endurance	App Crash/ Site Down	Approved				
3	Medicine Reminder Web -UI	Load	Server Crash/ Developer team/ Server Down	Approved				
End Of Test Report								
S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	Medicine Reminder Web -UI	Stress	Performance	CPU -01	GO	High Performance server	Closed	Approved
2	Medicine Reminder Web -UI	Load	Scalability	DB Storage - 01	NO-GO	MongoDB Instance for free	Closed	Approved
3	Medicine Reminder Web -UI	Endurance	Connectivity	Connection	GO	High Performance	Closed	Approved

10. ADVANTAGE & DISADVANTAGE

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 customizable 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

```
#include <SevSeg.h>
```

```
#include "Button.h"
```

```
#include "AlarmTone.h"
```

```
#include "Clock.h"
```

```
#include "config.h"
```

```
const int COLON_PIN = 13;
```

```
const int SPEAKER_PIN = A3;
```

```
Button hourButton(A0);
```

```
Button minuteButton(A1);
```

```
Button alarmButton(A2);
```

```
AlarmTone alarmTone;
```

```
Clock clock;
```

```
SevSeg sevseg;
```

```
enum DisplayState {
```

```
    DisplayClock,
```

```
    DisplayAlarmStatus,
```

```
    DisplayAlarmTime,
```

```
    DisplayAlarmActive,
```

```
    DisplaySnooze,
```

```
};
```

```
DisplayState displayState = DisplayClock;
```

```
long lastStateChange = 0;
```

```
void changeDisplayState(DisplayState newValue) {
```

```
    displayState = newValue;
```

```
    lastStateChange = millis();
```

```
}
```

```
long millisSinceStateChange() {  
  
    return millis() - lastStateChange;  
  
}
```

```
void setColon(bool value) {  
  
    digitalWrite(COLON_PIN, value ? LOW : HIGH);  
  
}
```

```
void displayTime() {  
  
    DateTime now = clock.now();  
  
    bool blinkState = now.second() % 2 == 0;  
  
    sevseg.setNumber(now.hour() * 100 + now.minute());  
  
    setColon(blinkState);  
  
}
```

```
void clockState() {
```

```
displayTime();
```

```
if (alarmButton.read() == Button::RELEASED && clock.alarmActive()) {
```

```
    // Read alarmButton has_changed() to clear its state
```

```
    alarmButton.has_changed();
```

```
    changeDisplayState(DisplayAlarmActive);
```

```
    return;
```

```
}
```

```
if (hourButton.pressed()) {
```

```
    clock.incrementHour();
```

```
}
```

```
if (minuteButton.pressed()) {
```

```
    clock.incrementMinute();
```

```
}
```

```
if (alarmButton.pressed()) {
```

```
    clock.toggleAlarm();
```

```

    changeDisplayState(DisplayAlarmStatus);

}

}

void alarmStatusState() {

    setColon(false);

    sevseg.setChars(clock.alarmEnabled() ? " on" : " off");

    if (millisSinceStateChange() > ALARM_STATUS_DISPLAY_TIME) {

        changeDisplayState(clock.alarmEnabled() ? DisplayAlarmTime : DisplayClock);

        return;

    }

}

void alarmTimeState() {

    DateTime alarm = clock.alarmTime();

    sevseg.setNumber(alarm.hour() * 100 + alarm.minute(), -1);

```



```
    if (millisSinceStateChange() > ALARM_HOUR_DISPLAY_TIME ||  
alarmButton.pressed()) {  
  
    changeDisplayState(DisplayClock);  
  
    return;  
  
}
```

```
if (hourButton.pressed()) {  
  
    clock.incrementAlarmHour();  
  
    lastStateChange = millis();  
  
}
```

```
if (minuteButton.pressed()) {  
  
    clock.incrementAlarmMinute();  
  
    lastStateChange = millis();  
  
}
```

```
if (alarmButton.pressed()) {  
  
    changeDisplayState(DisplayClock);  
  
}
```

```
}
```

```
void alarmState() {
```

```
    displayTime();
```

```
    if (alarmButton.read() == Button::RELEASED) {
```

```
        alarmTone.play();
```

```
    }
```

```
    if (alarmButton.pressed()) {
```

```
        alarmTone.stop();
```

```
    }
```

```
    if (alarmButton.released()) {
```

```
        alarmTone.stop();
```

```
        bool longPress = alarmButton.repeat_count() > 0;
```

```
        if (longPress) {
```

```
            clock.stopAlarm();
```

```
            changeDisplayState(DisplayClock);
```

```

    } else {

        clock.snooze();

        changeDisplayState(DisplaySnooze);

    }

}

}

}

void snoozeState() {

    sevseg.setChars("**");

    if (millisSinceStateChange() > SNOOZE_DISPLAY_TIME) {

        changeDisplayState(DisplayClock);

        return;

    }

}

void setup() {

    Serial.begin(115200);

```

```
clock.begin();
```

```
hourButton.begin();
```

```
hourButton.set_repeat(500, 200);
```

```
minuteButton.begin();
```

```
minuteButton.set_repeat(500, 200);
```

```
alarmButton.begin();
```

```
alarmButton.set_repeat(1000, -1);
```

```
alarmTone.begin(SPEAKER_PIN);
```

```
pinMode(COLON_PIN, OUTPUT);
```

```
byte digits = 4;
```

```

byte digitPins[] = {2, 3, 4, 5};

byte segmentPins[] = {6, 7, 8, 9, 10, 11, 12};

bool resistorsOnSegments = false;

bool updateWithDelays = false;

bool leadingZeros = true;

bool disableDecPoint = true;

sevseg.begin(DISPLAY_TYPE, digits, digitPins, segmentPins,
resistorsOnSegments,

        updateWithDelays, leadingZeros, disableDecPoint);

sevseg.setBrightness(90);

}

void loop() {

    sevseg.refreshDisplay();

    switch (displayState) {

        case DisplayClock:

```

```
clockState();
```

```
break;
```

```
case DisplayAlarmStatus:
```

```
alarmStatusState();
```

```
break;
```

```
case DisplayAlarmTime:
```

```
alarmTimeState();
```

```
break;
```

```
case DisplayAlarmActive:
```

```
alarmState();
```

```
break;
```

```
case DisplaySnooze:
```

```
snoozeState();
```

```
break;
```

```
}
```

```
}
```

Git repo link: <https://github.com/IBM-EPBL/IBM-Project-12142-1659379159>

Project Demo Link : <https://youtu.be/ZnkzqwiBMJo>