1. INTRODUCTION

1.1 PROJECT OVERVIEW

The Basic idea behind this project is to help the senior citizens to take their medicine at correct to maintain their health through medication. Usually senior citizens have memor lapses which could make the task of taking medcines difficult to cope up. To avoid this problem, we make a medical remainder system is developed, where the user(caretaker) could set the desired time and medicine for the patient. These details are stored in IBM cloud through IBM IoT platform where device will receive the command and alerts the person through voice commands to take the medicine.

1.2 PURPOSE

"Personal Assistant for Seniors who are Self-Reliant" project makes us understand folloowing sectors :

- Gain knowledge of Watson Iot Platform
- Connecting IoT devices to the Watson IoT platform and exchanging the sensor data
- Gain knowledge on IBM Text to Speech Service
- Explore python client libraries of Watson IoT platform
- Gain knowledge on IBM CLoudant DB
- Creating a Web Application that interacts with IoT device

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Many older people demonstrated desire to cope with their illness and maintain independence, however, environmental factors interfered with these effort :

1. Lack of professional advice on self-healthcare strategies
2. Poor communication between caretakers and seniors
3. Lack of information on services such as care pathways

A gap in knowledge was also identified about and support needed for seniors citizens to overcome these problems.

2.2 REFERENCES

TITLE	AUTHORS	DESCRIPTIONS
International Journal of Research in Engineering, Science and Management Volume-1, Issue-10, October-2018 Smart Medicine BoxUsing IOT	Sarvesh Kulkarni Shreyas kandgule Vaibhav Katkar Shrutika Tone P. A. Chadchankar	Alerts patient to take medicine on time by storing present time inRTC module. So at the time of taking medicine system generateNotification sound and display the light in certain pill boxes. There are 21 Sub-Compartment each having separate LED and sensors to indicate which medicines need to be taken and checkifpatient taken medicine or alert doctor and siblings using GSM module by sending SMS.

International journalof advance research,ideas and innovations in technology ISSN: 2454-132X (Volume 5, Issue 3, May, 2019) Medismart: Betterhealth with iot basedmed box	Vaibhavi G. Raut Tanaya Patil Praparna Moharana Shantanu Ghanekar Swati A. Joshi	MediSmart is an IOT based smart medicine box which reminds thepatient to take their prescribed medicine. It provides an androidapplication where user needs to fill and update their personal detailsand the details of medicine prescriptions in each field via anAndroid app. This Android Application then updates the database and saves the inputs given by the user. It is now connected to the Arduino through Wi-Fi Connection by ESP8266 Wi-Fi module viaTCP protocol. On the scheduled time, the Arduino receives requests via TCP protocol and then sends signals to the other components of the Medicine Box rendering the alarm system to setoff. The Alarm System provides both audio and visual aid to guidethe end user. There are LED lights that are placed on the box .It alerts the end user if medicines have not been consumed by making use of Infra-Red sensor and determining if the medicine has been consumed or it will notifies the user via an Android application that medicine has not been taken.
International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020	Harshitha V Sandeep K Swasthika Jain T J	This system uses IR (Infra-Red) sensor, camera, and RFID tags to count the number of pills inside a tray which are interfaced with the Arduino UNO.The RFID stickers fastened on each tablet sheet will be scanned by using the RFID, camera and IR sensor. The sensors will be giving the count of pills inside the box periodically for every 5 to 6 hours. The timings for the intake of medicines by the patient will be set for the device using RTC (Real Time Clock).When the time for the medication to the patient has come the timerset for the device using RTC will be sending notification using buzzer and GSM to the patientand their familymembers. Thenotification will be sent to themobile device and smartwatch

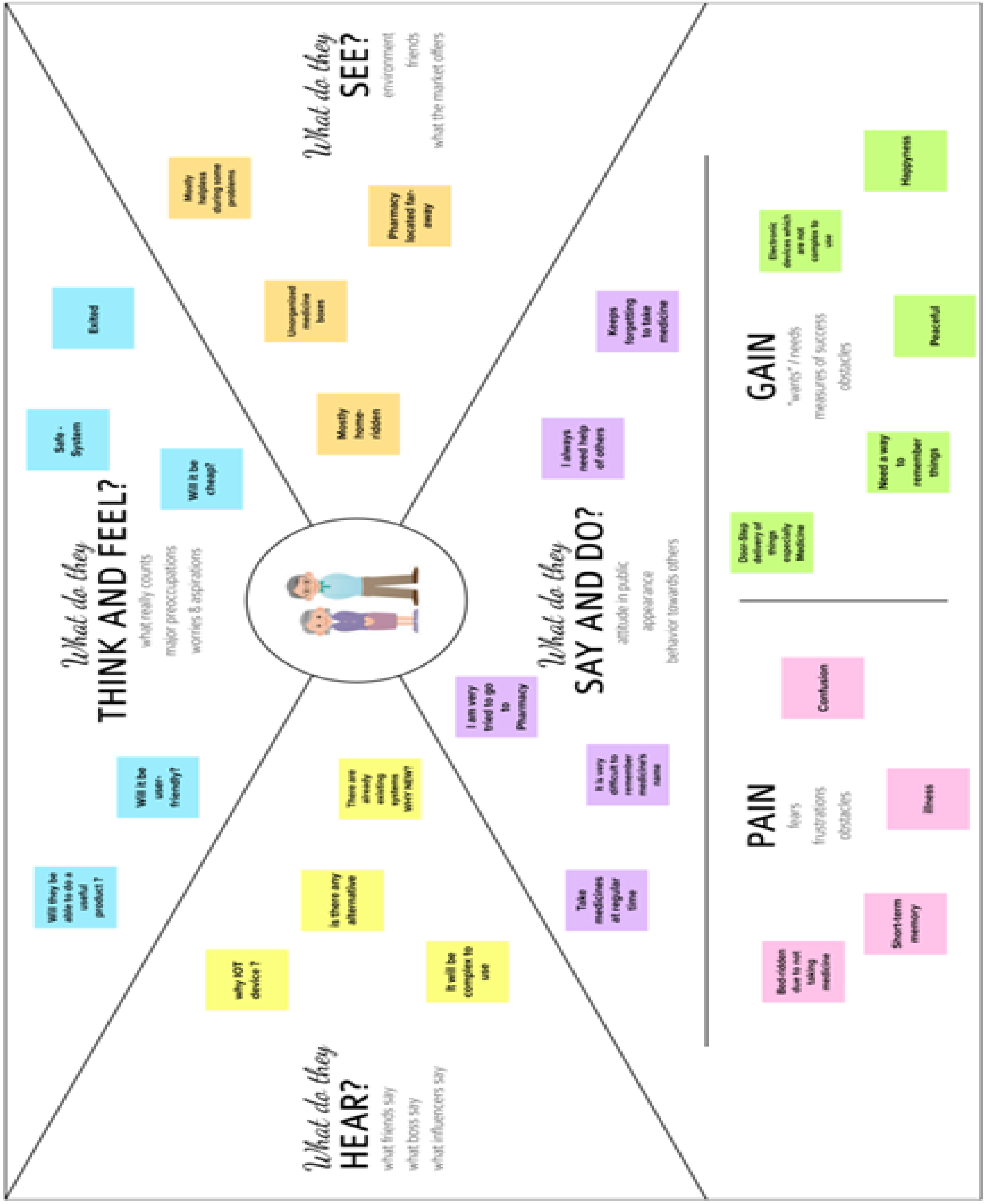
An Interactive PillBox using IoT		which is connected to the device. By using RFID tags the pills which are taken can be identified whether the patient has taken correct medicine or not at a prescribed time. The data will be updated into the web browser using the Wi-Fi module. After completely taking the medicines over days/months the device will be fixing an appointment with the doctor automatically by sendinga message using the GSM module and also convey the same to themedical shops to deliver the required medicines to patients addressor to the hospitals where the patient stays. The same thing will be displayedon the 16x2 LCD.

2.3 PROBLEM STATEMENT DEFINITION

The older people have unmet care needed to overcome their health issue and this review also emphasized the importance of developing care models and support services based around the needs of the older people.

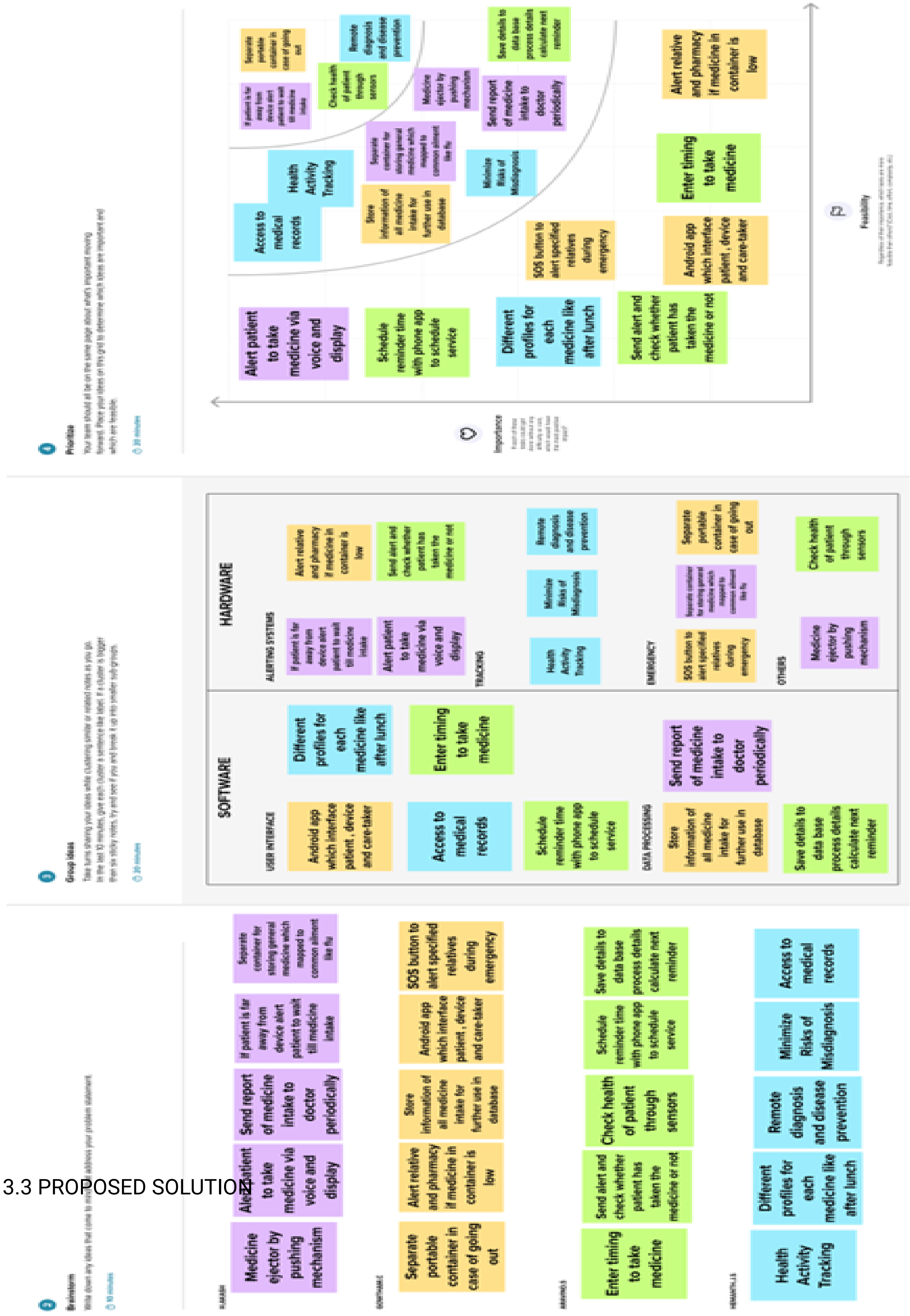
3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS


What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Exited
Safe - System
Will it be cheap?
Will it be user-friendly?
Will they be able to do a useful product ?
What do they
SEE?
environment
friends
what the market offers
Mostly helpless during some problems
Pharmacy located far-away
Unorganized medicine boxes
Mostly home-ridden
What do they
HEAR?
what friends say
what boss say
what influencers say
There are already existing systems WHY NEW?
Is there any alternative
why IOT device ?
It will be complex to use
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Keeps forgetting to take medicine
I always need help of others
I am very tried to go to Pharmacy
It is very difficult to remember medicine's name
Take medicines at regular time
GAIN
"wants" / needs
measures of success
obstacles
Happyness
Electronic devices which are not complex to use
Peaceful
Need a way to remember things
Door-Step delivery of things especially Medicine
PAIN
fears
frustrations
obstacles
Confusion
Illness
Short-term memory
Bed-ridden due to not taking medicine

3.2 IDEATION & BRAINSTORMING

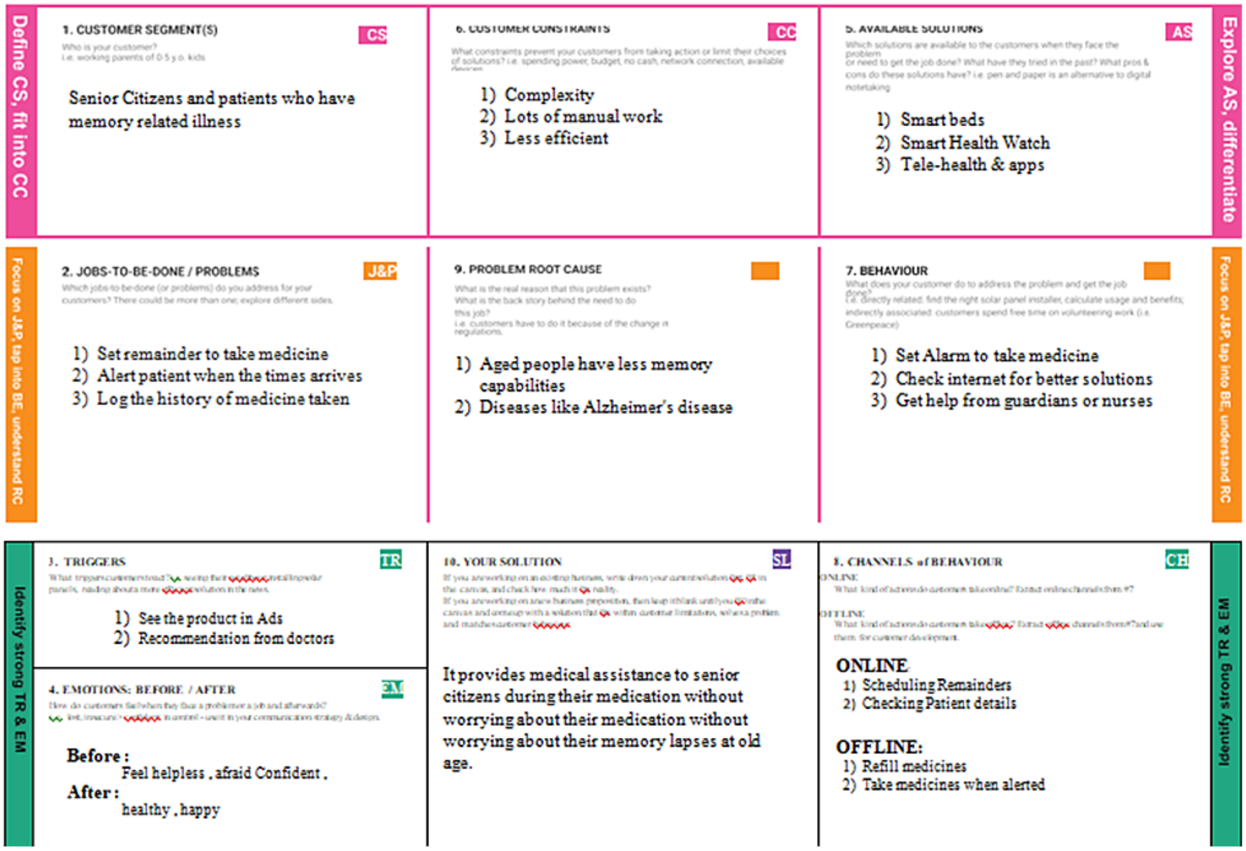
3.3 PROPOSED SOLUTION


Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
Medicine ejector by pushing mechanism
Alert patient to take medicine via voice and display
Send report of medicine intake to doctor periodically
If patient is far away from device alert patient to wait till medicine intake
Separate container for storing general medicine which mapped to common ailment like flu
Separate portable container in case of going out
Alert relative and pharmacy if medicine in container is low
Store information of all medicine intake for further use in database
Android app which interface patient, device and care-taker
SOS button to alert specified relatives during emergency
Enter timing to take medicine
Send alert and check whether patient has taken the medicine or not
Check health of patient through sensors
Schedule reminder time with phone app to schedule service
Save details to data base process details calculate next reminder
Health Activity Tracking
Different profiles for each medicine like after lunch
Remote diagnosis and disease prevention
Minimize Risks of Misdiagnosis
Access to medical records
Group ideas
20 minutes
SOFTWARE
USER INTERFACE
Android app which interface patient, device and care-taker
Access to medical records
Schedule reminder time with phone app to schedule service
Different profiles for each medicine like after lunch
Enter timing to take medicine
DATA PROCESSING
Store information of all medicine intake for further use in database
Save details to data base process details calculate next reminder
Send report of medicine intake to doctor periodically
HARDWARE
ALERTING SYSTEMS
If patient is far away from device alert patient to wait till medicine intake
Alert patient to take medicine via voice and display
Alert relative and pharmacy if medicine in container is low
Send alert and check whether patient has taken the medicine or not
TRACKING
Health Activity Tracking
Minimize Risks of Misdiagnosis
Remote diagnosis and disease prevention
EMERGENCY
SOS button to alert specified relatives during emergency
Separate container for storing general medicine which mapped to common ailment like flu
Separate portable container in case of going out
OTHERS
Medicine ejector by pushing mechanism
Check health of patient through sensors
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
Feasibility
Alert patient to take medicine via voice and display
Schedule reminder time with phone app to schedule service
Different profiles for each medicine like after lunch
Send alert and check whether patient has taken the medicine or not
Access to medical records
Health Activity Tracking
Store information of all medicine intake for further use in database
Separate container for storing general medicine which mapped to common ailment like flu
Minimize Risks of Misdiagnosis
SOS button to alert specified relatives during emergency
Android app which interface patient, device and care-taker
Enter timing to take medicine
If patient is far away from device alert patient to wait till medicine intake
Separate portable container in case of going out
Check health of patient through sensors
Remote diagnosis and disease prevention
Medicine ejector by pushing mechanism
Send report of medicine intake to doctor periodically
Save details to data base process details calculate next reminder
Alert relative and pharmacy if medicine in container is low

S.No.	**Parameter**	**Description**
1.	Problem Statement (Problem to be solved)	This project deals with the problems of elderly people who forget to take their medicine at the correct time which affects their medical diagnosis and it is difficult to monitor the patients for a whole day. From this project, we create an app built for the user(caretaker) to set the desired time and medicine which help the patients(Elderly people) to receive medicine at the correct time.
2.	Idea / Solution description	The key research objectives are as follows: • The proposed app would be able to provide the Personal Assistance for Seniors who are self-reliant in taking medicines as per the doctor consultant using IOT (Internet of Things). • The Proposed app consists of TTS(Text to Speech) service, IOT(Internet of Things) device and Cloud Services. • In the proposed app, the caretaker will be able to set the desire time and medicine for the patient through cloud services. • If the medicine time arrives the web application will send the medicine name to the IOT device and alert the user through voice commands.
3.	Novelty / Uniqueness	In our system we are going to implement automatic medicine ejection mechanism which reduces the no of compartments to hold the medicine and makes it easy to take medicine.
4.	Social Impact / Customer Satisfaction	They have the potential to improve the sustainability of health care of senior citizens around the globe. Elder people will not lose their independence and provides relief for family caregivers. It provides personalized care, comfort and convenience for the senior citizens.

5.	Business Model (Revenue Model)	The system is profitable in health sector. Subscribtion services can be implemented to have a special delivery of medicine in a monthly basis and app can implement a shop to purchase medicine directly .
6.	Scalability of the Solution	The system is open to lot of improvements for the future such as smart band to monitor the patient health continuously and recommend medicine if any illness is detected. Also can book appointments to doctor automatically on a regular basis.

3.4 PROPOSED SOLUTION FIT


Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Identify strong TR & EM
CS
CC
AS
J&P
TR
EM
SL
CH

1. CUSTOMER SEGMENT(S)

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Senior Citizens and patients who have memory related illness

6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

1) Complexity
2) Lots of manual work
3) Less efficient

5. AVAILABLE SOLUTIONS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

1) Smart beds
2) Smart Health Watch
3) Tele-health & apps

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

1) Set remainder to take medicine
2) Alert patient when the times arrives
3) Log the history of medicine taken

9. PROBLEM ROOT CAUSE

What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

1) Aged people have less memory capabilities
2) Diseases like Alzheimer's disease

7. BEHAVIOUR

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

1) Set Alarm to take medicine
2) Check internet for better solutions
3) Get help from guardians or nurses

3. TRIGGERS

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

1) See the product in Ads
2) Recommendation from doctors

4. EMOTIONS: BEFORE / AFTER

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before :
Feel helpless , afraid Confident ,
After :
healthy , happy

10. YOUR SOLUTION

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

It provides medical assistance to senior citizens during their medication without worrying about their medication without worrying about their memory lapses at old age.

8. CHANNELS of BEHAVIOUR

ONLINE
What kind of actions do customers take online? Extract online channels from #7

OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

ONLINE
1) Scheduling Remainders
2) Checking Patient details

OFFLINE:
1) Refill medicines
2) Take medicines when alerted

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution :

FR No.	Functional Requirement(Epic)	Sub Requirement(Story/Sub-Task)
FR-1	User Registration	User registration through gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Schedule Medicine In-take paramaters	● Set Medicine Name ● Set Time ● Set before/after lunch
FR-4	Alert End Users	● Alert Patient to take medicine at appropriate time ● Alert Care-taker about medicine in-take status ● Alert Care-taker, if medicines low on storage.

4.2 NON-FUNCTIONAL REQUIREMENT

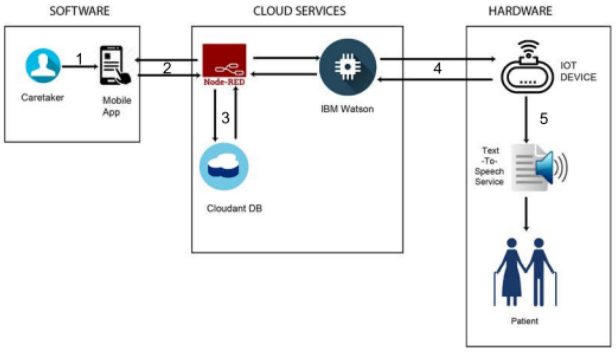
Following are the non-functional requirements of the proposed solution :

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Shoud be usefull for all patients to take medicines irrespective of other requirement
NFR-2	Security	All datas of patients should be well secured from attacks
NFR-3	Reliability	Should remind patients to take medicines without any delay
NFR-4	Performance	Performance should be smooth and easy to user interface
NFR-5	Availability	It should connect patient and care-taker irrespective of distance
NFR-6	Scalability	should be developed with modules easy to scale in future

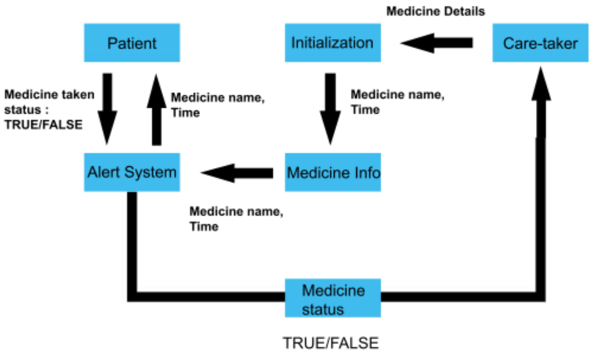
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

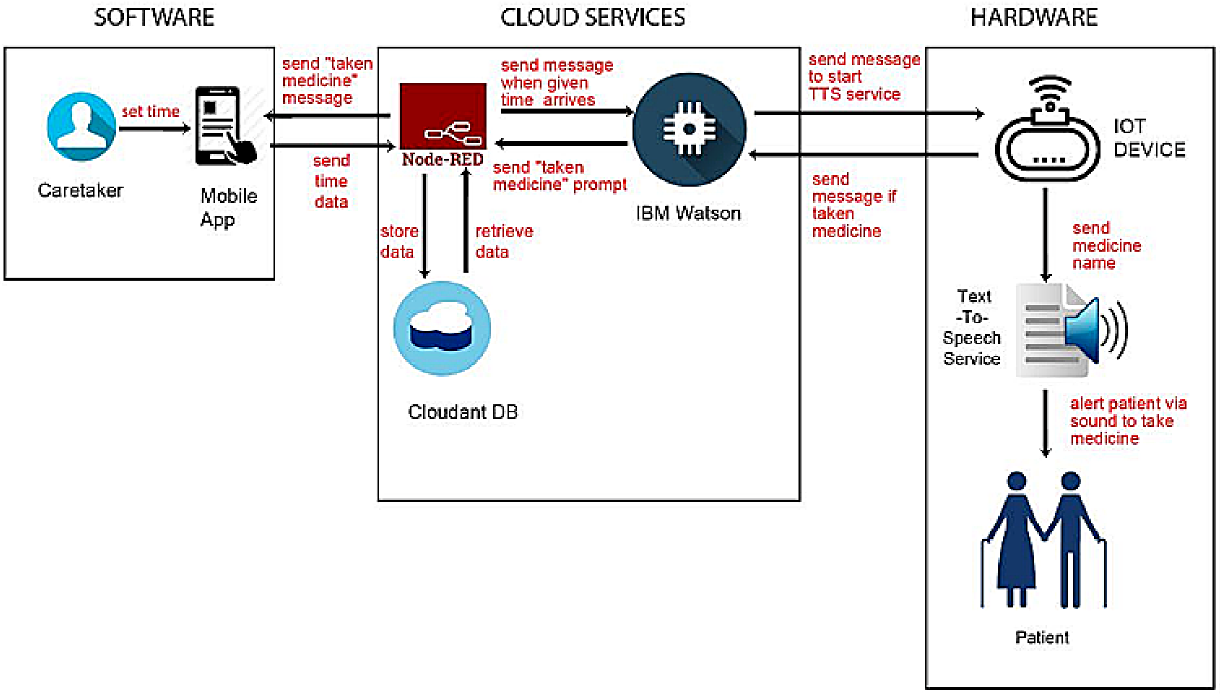
A Data Flow Diagram(DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



1) Care-taker gives login details and medicines details to the mobile app.
2) Data travels to and fro with Node-Red
3) Cloudant DB stores all data of medicine details
4) IBM Watson connects Software and hardware side
5) IOT device gets data of medicine and alerts the patient



5.2 SOLUTION & TECHNICAL ARCHITECTURE


SOFTWARE
CLOUD SERVICES
HARDWARE
set time
Caretaker
Mobile App
send "taken medicine" message
send time data
Node-RED
store data
retrieve data
Cloudant DB
send message when given time arrives
send "taken medicine" prompt
IBM Watson
send message to start TTS service
send message if taken medicine
IOT DEVICE
send medicine name
Text -To- Speech Service
alert patient via sound to take medicine
Patient

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story No.	User Story/Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password and confirming my password	I cn access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can recieve confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with facebook login	Low	Sprint-4
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account/dashboard	High	Sprint-1
	Dashboard	USN-6	As a user, I can see all the current scedules	I can see all the details of medicine times and count	High	Sprint-2
		USN-7	As a user, I have menu to	I can add new shedules for	High	Sprint-2

			add new schedules for medicines	medicines		
		USN-8	As a user, I can get alert messages if medicine not taken before specified time	I can get alert messages if medicine not taken before specified time	Medium	Sprint-3
	Logout	USN-9	As a user, I can logout of the application	I can logout of application	High	Sprint-1
Customer Care Executive	Login	USN-10	As a executive, I can log into my application using my employee id and password	I can access my account/dashboard	Medium	Sprint-4
	Dashboard	USN-11	As a executive, I can retrieve customer details by putting in username	I can get customer details	Medium	Sprint-4
		USN-12	As a executive, I can raise a complaint of customer and log the details to Admin	I can raise complaint	Low	Sprint-4
Administrator	Login	USN-10	As a admin, I can log into my application using my admin id and password	I can access my account/dashboard	High	Sprint-3
	Dashboard	USN-11	As a admin, I can access all database and make changes	I can access database	Medium	Sprint-3

			if required			
		USN-12	As a admin, I can push update to application	I can update application	Low	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	**Functional Requirement (Epic)**	**User Story No.**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**

Sprint-1	Registration	USN-1	A user can register their application usingemail address.	10	Low	Hemanth.J.S S.Aravind
Sprint-1		USN-2	User willreceive a confirmation mail to checkthe authentication of the user.	10	Medium	H.Akash C.Gowtham
Sprint-2	Login	USN-3	To login into the application, user can use email and password which they have used to register	5	High	H.Akash Hemanth.J.S
Sprint-2		USN-4	Login can alsobe done through facebook account by the user.	5	High	C.Gowtham S.Aravind
Sprint-2		USN-5	User can register their medicines andalarmsas required for the patient	2	Medium	S.Aravind H.Akash
Sprint-3	Dashboard	USN-6	As a user, we could observe all the current schedules for medications.	6	Medium	Hemanth.J.S C.Gowtham

Sprint-3		USN-7	Caretaker could alterthe medication schedule as per the requirement	2	Low	S.Aravind C.Gowtham
Sprint-3		USN-8	User(caretaker) can retrieve the details of themedication given to patients	5	High	H.Akash Hemanth.J.S.
Sprint-3	Alert System	USN-9	User’s list of scheduled medicine are sent tothe IOT device through Text-to-Speech Device.	5	Low	H.Akash S.Aravind
Sprint-3		USN-10	Through Cloud service the IOT device fedwith details of medicine for the patients.	5	Medium	C.Gowtham Hemanth.J.S
Sprint-3		USN-11	It alerts the patient through the IOT device.	10	High	H.Akash Hemanth.J.S
Sprint-4	Logout	USN-12	User can logout of the application.	10	High	C.Gowtham S.Aravind H.Akash

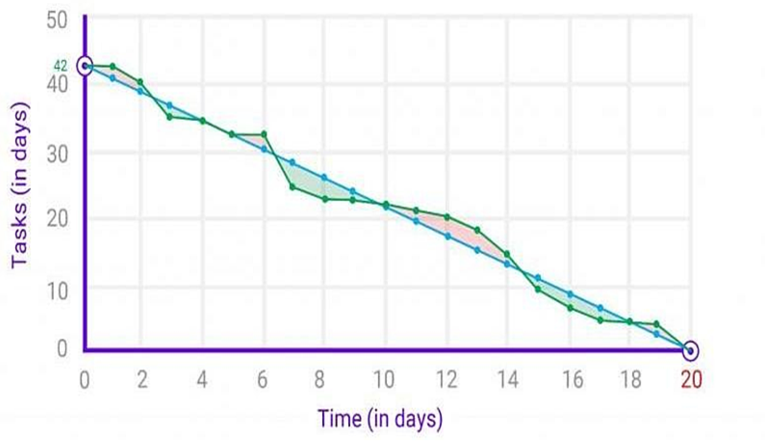
6.2 SPRINT DELIVERY SCHEDULE

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as onPlanned End Date)**	**Sprint Release Date(Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA

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.

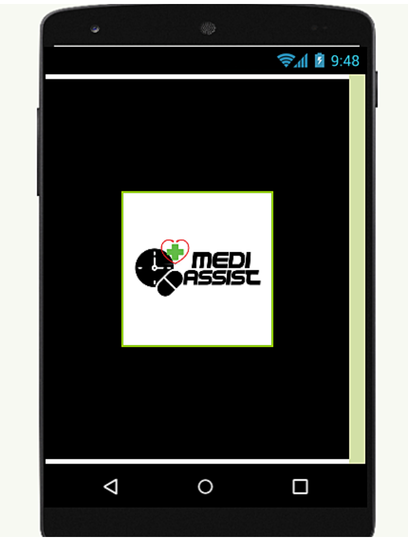


7. CODING & SOLUTIONING

7.1 FEATURE 1

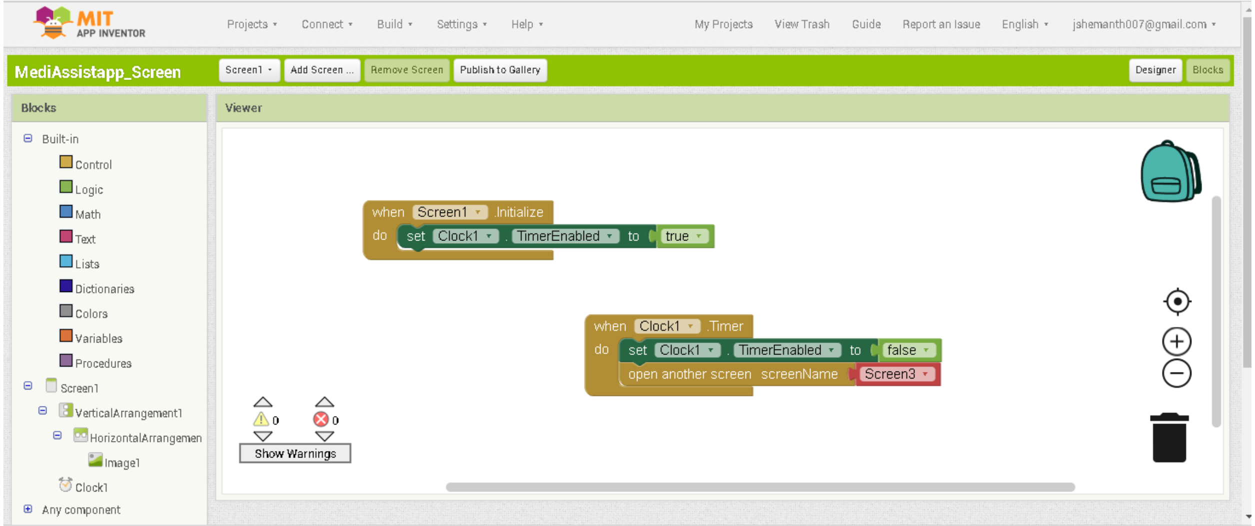
In this Project, we have createad a Personal assistance application called "Medi-Assist App" using MIT App Inventor. The app was built with total nine screens, where each screen have their unquie functions.

SCREEN 1 :



This screen 1 works as a welcome screen for the " Medi-Assist App" where we have inserted the application logo to the screen.

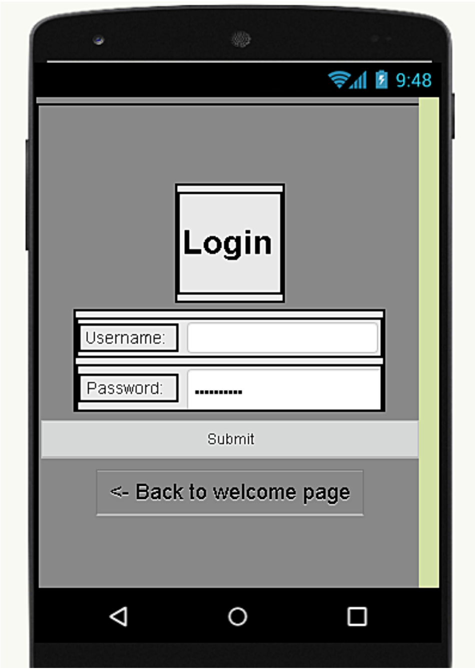
Design Block :



```
when Screen1 .Initialize
do  set Clock1 . TimerEnabled to true

when Clock1 .Timer
do  set Clock1 . TimerEnabled to false
    open another screen  screenName  Screen3
```

SCREEN 2 :

In Screen 2, we have implemented the fuctions to login into the application with a username and password, which stores the data in IBM cloud. Screen also option to go back to the welcome page where you could Login and SignUp.



9:48

Login

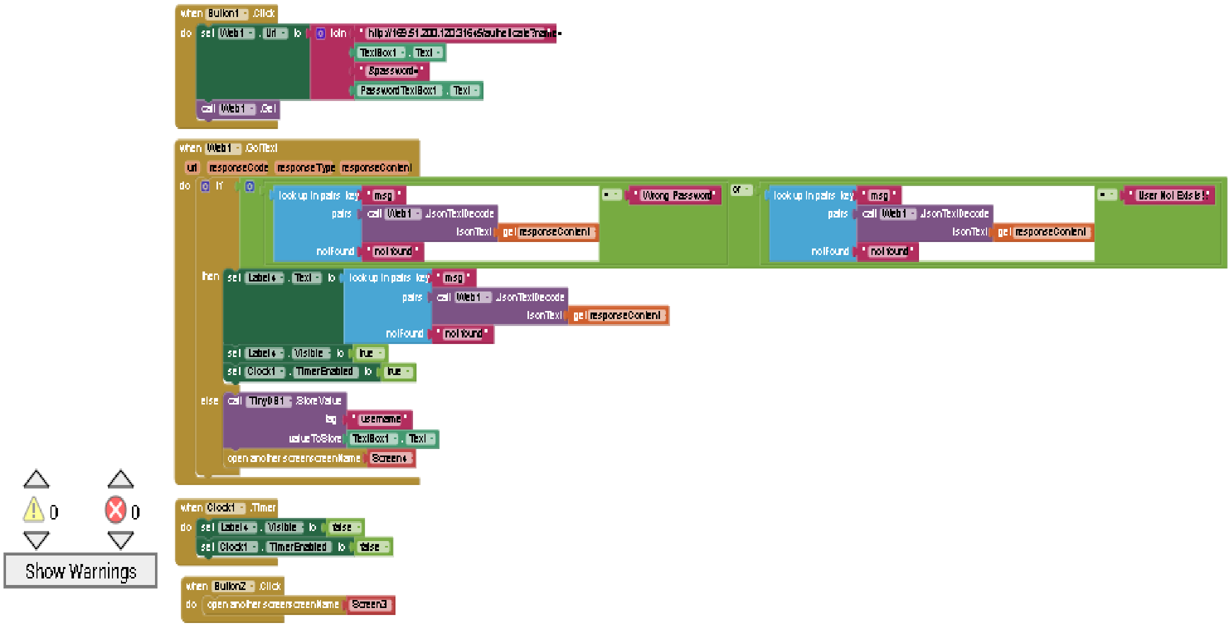
Username:

Password: ....

Submit

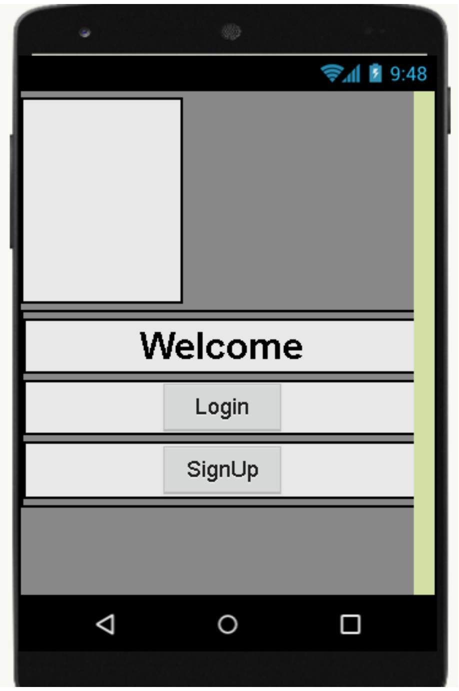
<- Back to welcome page

Design Block :



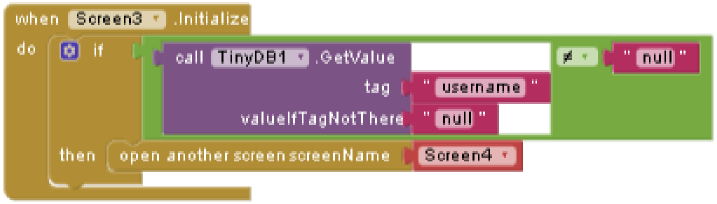
```
when Button1 .Click
do  set Web1 . Url to  join  "http://169.51.200.120:31645/authenticate?name="
                             TextBox1 . Text
                             "&password="
                             PasswordTextBox1 . Text
    call Web1 .Get

when Web1 .GotText
 url  responseCode  responseType  responseContent
do  if  look up in pairs key "msg"
                       pairs call Web1 .JsonTextDecode
                                        jsonText get responseContent
                    notFound "not found"
        = "Wrong Password"
        or
        look up in pairs key "msg"
                       pairs call Web1 .JsonTextDecode
                                        jsonText get responseContent
                    notFound "not found"
        = "User Not Exists!"
    then set Label4 . Text to  look up in pairs key "msg"
                                             pairs call Web1 .JsonTextDecode
                                                              jsonText get responseContent
                                          notFound "not found"
         set Label4 . Visible to true
         set Clock1 . TimerEnabled to true
    else call TinyDB1 .StoreValue
                            tag "username"
                     valueToStore TextBox1 . Text
         open another screen screenName "Screen4"

when Clock1 .Timer
do  set Label4 . Visible to false
    set Clock1 . TimerEnabled to false

when Button2 .Click
do  open another screen screenName "Screen3"
```

SCREEN 3 :

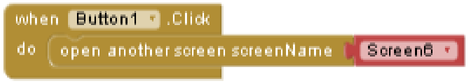


In Screen 3, we have built the welcome page of the "Medi-Assist App" where Login and Sign Up for the application is developed. New user should SignUp and existing user could Login directly.

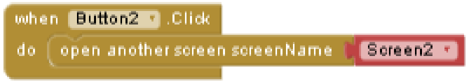
Design Block :



```
when Screen3 .Initialize
do  if   call TinyDB1 .GetValue
                         tag  " username "
            valueIfTagNotThere  " null "      ≠   " null "
    then  open another screen screenName  Screen4
```

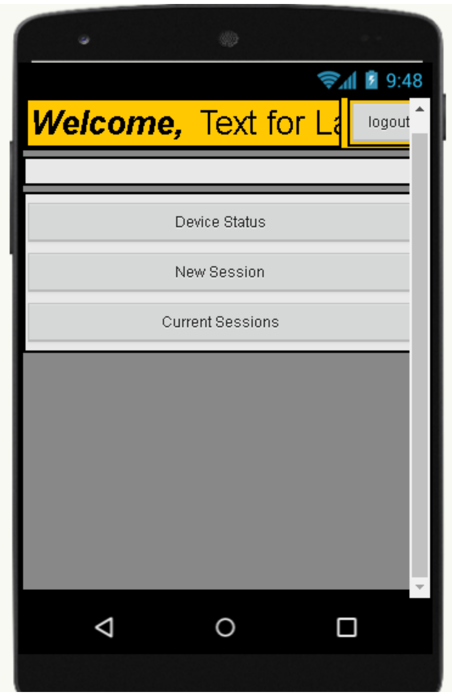


```
when Button1 .Click
do  open another screen screenName  Screen6
```



```
when Button2 .Click
do  open another screen screenName  Screen2
```

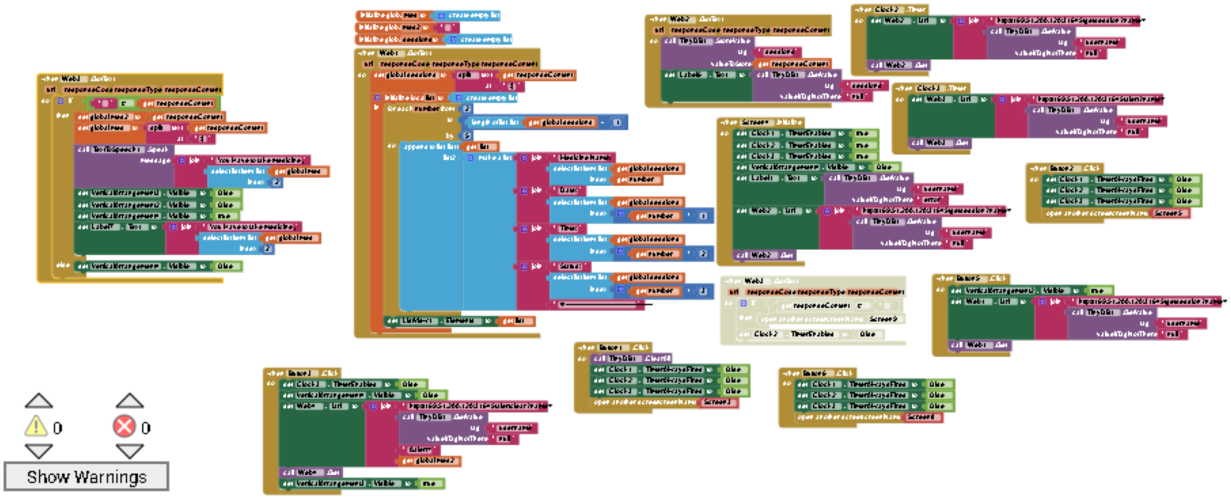
SCREEN 4 :



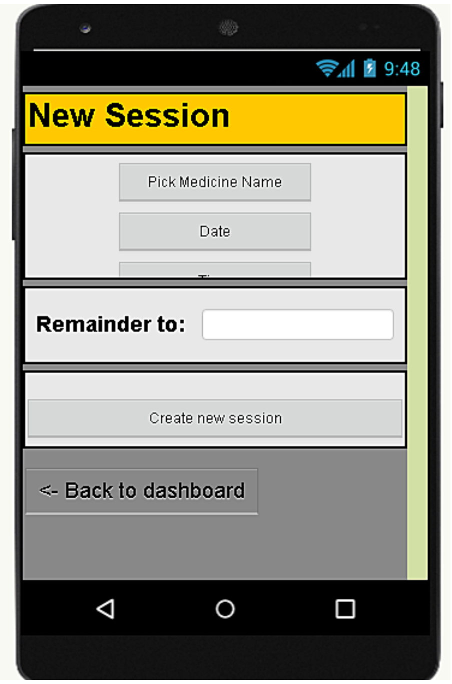
In Screen 4, "Medi-Assist App" provided with the dashboard. The Dashboard screen has four functions which are Device Status, New Session, Current Session and Logout.

- Device Status : It provides the status of the IoT device working.
- New Session : It enable us to create new session for taking medicine by the user.
- Current Session : It shows us the existing sessions in this account.
- Logout : it enable us to logout of the account.

Design Block :



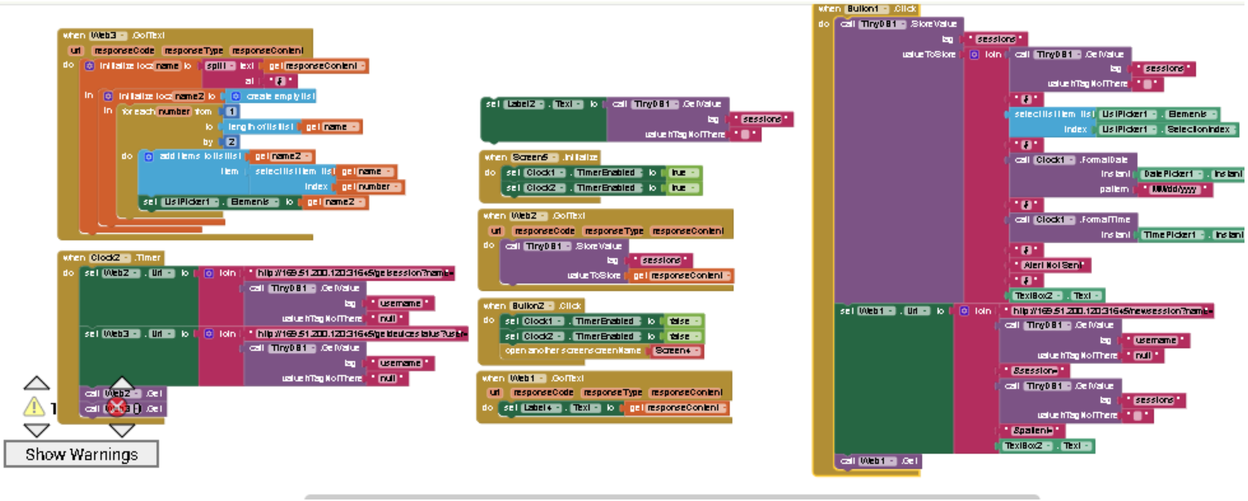
SCREEN 5 :



In Screen 5, If we click New Session in Dashboard(Screen 4) then it directs the user to the screen t where, the user can pick medicine name and date for the medicine taking person. Through ' Remainder to ' tab, you could remainder that person.

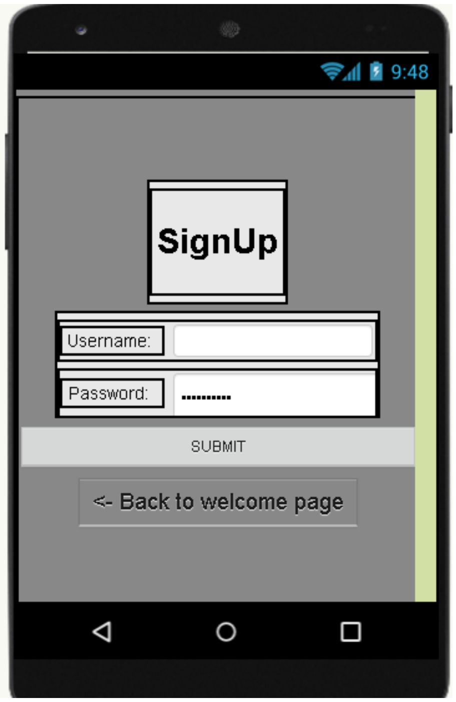
These created session are stored in IBM cloud database and can be viewed through 'Current Sessions' tab in the Dashboard(Screen 4).

Design Block :



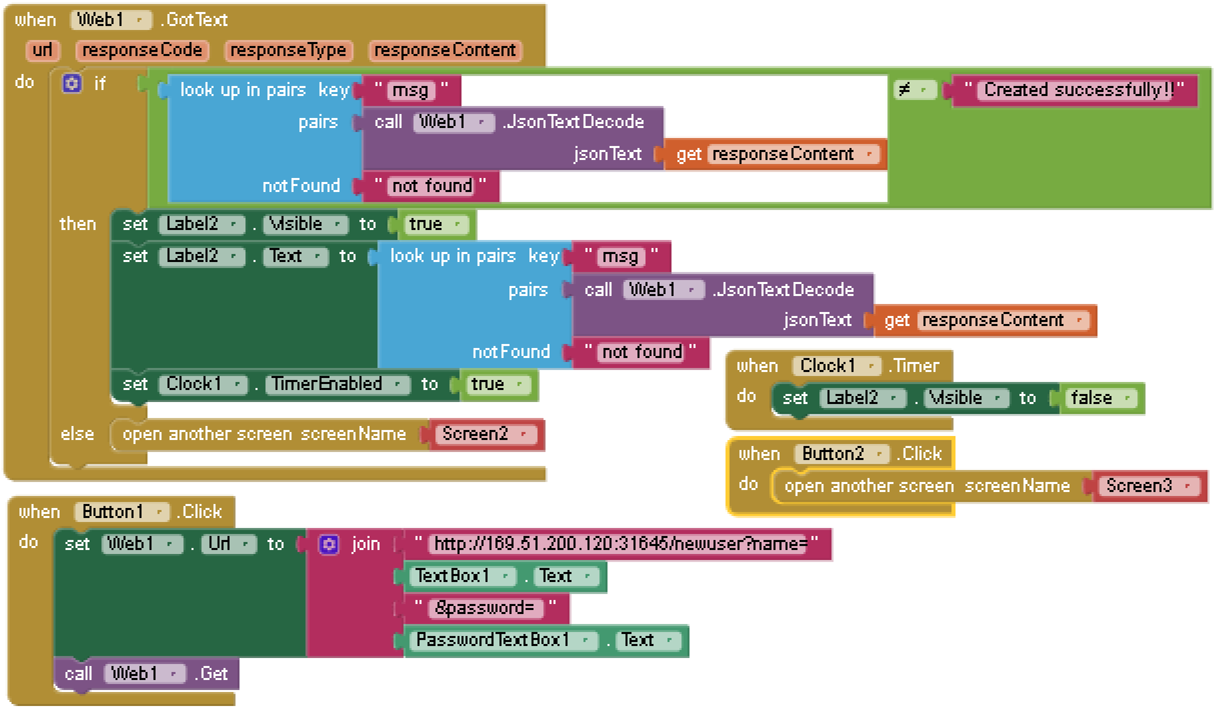
```
when Web3 .GotText
  url  responseCode  responseType  responseContent
do  initialize local name to  split  text  get responseContent
                                      at  " # "
    in  initialize local name2 to  create empty list
        in  for each number from  1
                            to  length of list list  get name
                            by  2
            do  add items to list list  get name2
                                  item  select list item  list  get name
                                                          index  get number
            set ListPicker1 . Elements to  get name2

when Clock2 .Timer
do  set Web2 . Url to  join  " http://169.51.200.120:31645/getsession?name= "
                             call TinyDB1 .GetValue
                                        tag  " username "
                                        valueIfTagNotThere  " null "
    set Web3 . Url to  join  " http://169.51.200.120:31645/getdevicestatus?user= "
                             call TinyDB1 .GetValue
                                        tag  " username "
                                        valueIfTagNotThere  " null "
    call Web2 .Get
    call Web3 .Get

set Label2 . Text to  call TinyDB1 .GetValue
                                 tag  " sessions "
                                 valueIfTagNotThere  "  "

when Screen6 .Initialize
do  set Clock1 . TimerEnabled to  true
    set Clock2 . TimerEnabled to  true

when Web2 .GotText
  url  responseCode  responseType  responseContent
do  call TinyDB1 .StoreValue
               tag  " sessions "
               valueToStore  get responseContent

when Button2 .Click
do  set Clock1 . TimerEnabled to  false
    set Clock2 . TimerEnabled to  false
    open another screen screenName  Screen4

when Web1 .GotText
  url  responseCode  responseType  responseContent
do  set Label4 . Text to  get responseContent

when Button1 .Click
do  call TinyDB1 .StoreValue
               tag  " sessions "
               valueToStore  join  call TinyDB1 .GetValue
                                              tag  " sessions "
                                              valueIfTagNotThere  "  "
                                   " # "
                                   select list item  list  ListPicker1 . Elements
                                                     index  ListPicker1 . SelectionIndex
                                   " # "
                                   call Clock1 .FormatDate
                                               instant  DatePicker1 . Instant
                                               pattern  " MM/dd/yyyy "
                                   " # "
                                   call Clock1 .FormatTime
                                               instant  TimePicker1 . Instant
                                   " # "
                                   " Alert Not Sent "
                                   " # "
                                   TextBox2 . Text
    set Web1 . Url to  join  " http://169.51.200.120:31645/newsession?name= "
                             call TinyDB1 .GetValue
                                        tag  " username "
                                        valueIfTagNotThere  " null "
                             " &session= "
                             call TinyDB1 .GetValue
                                        tag  " sessions "
                                        valueIfTagNotThere  "  "
                             " &patient= "
                             TextBox2 . Text
    call Web1 .Get
```

SCREEN 6 :



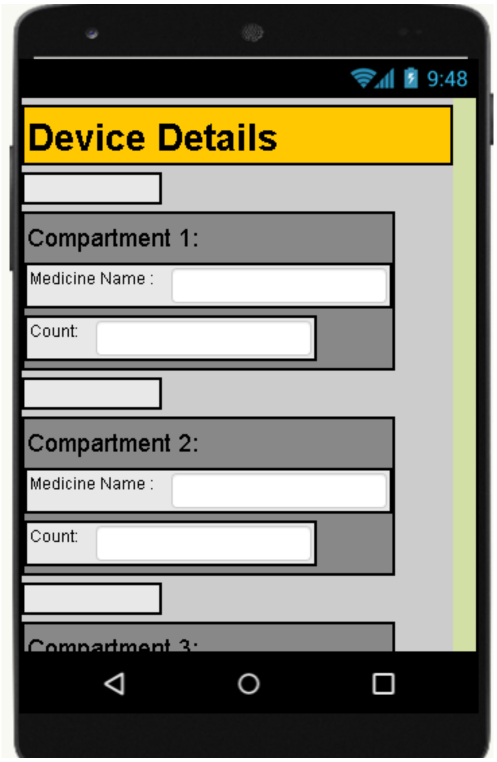
In Screen 6, This is the Sign Up page for the "Medi-Assist App" which is used to create a new database for the user to operate.

Design Blocks :



```
when Web1 .GotText
  url  responseCode  responseType  responseContent
do  if  look up in pairs key " msg "
              pairs  call Web1 .JsonTextDecode
                         jsonText  get responseContent
              notFound " not found "
        ≠  " Created successfully!! "
    then  set Label2 . Visible to  true
          set Label2 . Text to  look up in pairs key " msg "
                                   pairs  call Web1 .JsonTextDecode
                                              jsonText  get responseContent
                                   notFound " not found "
          set Clock1 . TimerEnabled to  true
    else  open another screen screenName  Screen2

when Clock1 .Timer
do  set Label2 . Visible to  false

when Button2 .Click
do  open another screen  screenName  Screen3

when Button1 .Click
do  set Web1 . Url to  join  " http://169.51.200.120:31645/newuser?name= "
                             TextBox1 . Text
                             " &password= "
                             PasswordTextBox1 . Text
    call Web1 .Get
```

SCREEN 7:



9:48

Device Details

Compartment 1:

Medicine Name :

Count:

Compartment 2:

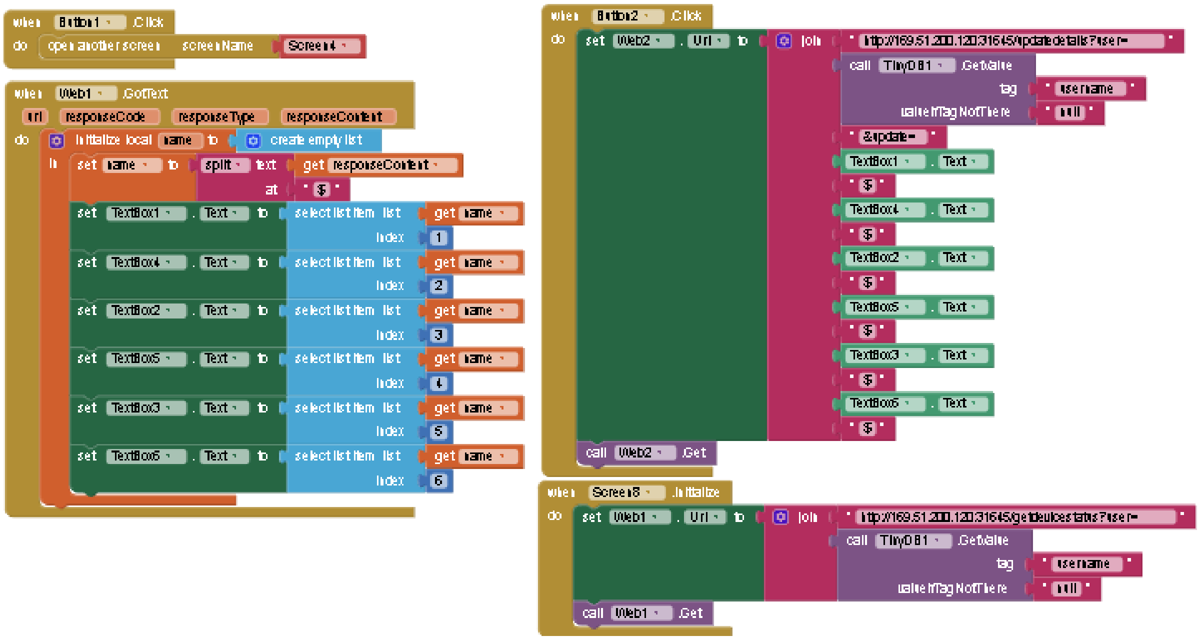
Medicine Name :

Count:

Compartment 3:

In Screen 7, We could store data about medicines with name and total count of the medicine to manage the medicine for the patient.

Design Block :



```
when Button1 .Click
do  open another screen  screenName  "Screen4"

when Web1 .GotText
  url  responseCode  responseType  responseContent
do  initialize local name to  create empty list
    in  set name to  split  text  get responseContent
                            at  "$"
        set TextBox1 . Text to  select list item  list  get name
                                                  index  1
        set TextBox4 . Text to  select list item  list  get name
                                                  index  2
        set TextBox2 . Text to  select list item  list  get name
                                                  index  3
        set TextBox5 . Text to  select list item  list  get name
                                                  index  4
        set TextBox3 . Text to  select list item  list  get name
                                                  index  5
        set TextBox6 . Text to  select list item  list  get name
                                                  index  6

when Button2 .Click
do  set Web2 . Url to  join  "http://169.51.200.120:31645/updatedetails?user="
                             call TinyDB1 .GetValue
                                  tag  "username"
                                  valueIfTagNotThere  "null"
                             "&update="
                             TextBox1 . Text
                             "$"
                             TextBox4 . Text
                             "$"
                             TextBox2 . Text
                             "$"
                             TextBox5 . Text
                             "$"
                             TextBox3 . Text
                             "$"
                             TextBox6 . Text
                             "$"
    call Web2 .Get

when Screen8 .Initialize
do  set Web1 . Url to  join  "http://169.51.200.120:31645/getdevicestatus?user="
                             call TinyDB1 .GetValue
                                  tag  "username"
                                  valueIfTagNotThere  "null"
    call Web1 .Get
```

7.2 FEATURE 2

In this part, we have created a circuit of the IoT device which is connected through Node Red to the IBM cloud database and Medi-Assist app which is created through MIT App Inventor.

Following program is done in the Wokwi simulator for IoT device :

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include<ESP32Servo.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 4);

#define LED 2

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

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

#define ORG "gzcaq8"//IBM ORGANITION ID
#define DEVICE_TYPE "IOT"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "47"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
float h, t;

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

//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefined client id by passing parameter like server id,portand
wificredential

char compartment_med[3][20];
char compartment_count[3][20],alertArr[20];
String username="akash",devicedetails,alert;
int f=0;


void setup()// configureing the ESP32
{
  Serial.begin(115200);
  pinMode(LED,OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(15, OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
  //PublishData("getdetail");
  //PublishData(compartment);
  PublishData("getdetail");
  lcd.begin(20,4);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(5, 1);
  lcd.print("MediAssist");
  lcd.setCursor(6, 2);
  lcd.print("Welcome!");
  delay(2000);
  lcd.clear();
}

void loop()// Recursive Function
{
  //PublishData(compartment);
  PublishData("getdetail");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Username:"+username);
  //lcd.setCursor(0,1);
  //lcd.print("Medicine : Count");
  //lcd.setCursor(0,2);
  for(int i =0;i<2;i++)
  {
    /*lcd.print(compartment_med[i]);
    lcd.print(":");
    lcd.print(compartment_count[i]);
    lcd.print(" ");*/
    devicedetails += compartment_med[i];
    devicedetails += "$";
    devicedetails += compartment_count[i];
    devicedetails += "$";
  }
  if(f==0)
  {
    digitalWrite(4, LOW);
    digitalWrite(2, LOW);
    digitalWrite(15, LOW);
    lcd.setCursor(0,1);
  lcd.print(compartment_med[0]);
  lcd.print(":");
  lcd.print(compartment_count[0]);
  lcd.setCursor(0,2);
  lcd.print(compartment_med[1]);
  lcd.print(":");
  lcd.print(compartment_count[1]);
  lcd.setCursor(0,3);
  lcd.print(compartment_med[2]);
  lcd.print(":");
  lcd.print(compartment_count[2]);
  }
  else if(f==1)
  {
    lcd.setCursor(0,1);
    lcd.print("You have to take ");
    lcd.setCursor(0,2);
    lcd.print("medicine ,");
    lcd.setCursor(0,3);
    lcd.print(alert);
    for(int j=0;j<alert.length();j++)
    {
      alertArr[j]=alert[j];
    }
    Serial.println(alertArr);
    for(int i=0;i<3;i++)
    {
      if(strcmp(compartment_med[i],alertArr)==0)
      {
        if(i==0){
          digitalWrite(15, HIGH);
        }
        else if(i==1){
          digitalWrite(2, HIGH);
        }
        else if(i==2){
          digitalWrite(4, HIGH);
        }
      }
    }
  }
```

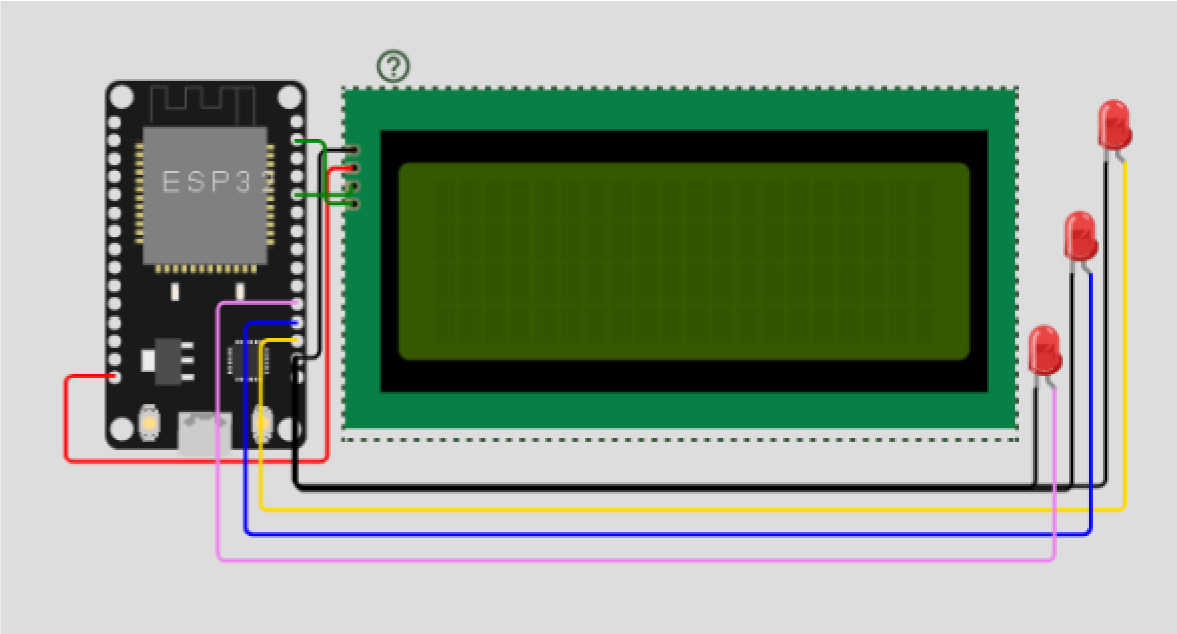
```
  //Serial.println(devicedetails);

  devicedetails = "";

  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}

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

void PublishData(String comp) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload="";
  if(comp=="getdetail")
  {
    payload="{\"Command\":";
    payload+="\"";
    payload+=comp;
    payload+="\"";
    payload += " ,";
    payload += "\"user\":";
    payload += "\"";
    payload += username;
    payload += "\"";
    payload += " }";
  }
  else {
  payload = "{\"Medicine\":";
  payload += "\"";
  payload += comp ;
  payload += "\"";
  payload += " ,";
  payload += "\"User\":";
  payload += "\"";
  payload += username;
  payload += "\",";
  payload += "\"Command\":";
  payload += "\"senddetail\"";
  payload += " }";
  }


  Serial.print("Sending payload: ");
  Serial.println(payload);

   if (client.publish(publishTopic, (char*) payload.c_str()))
{
  Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish failed
  }
else
 {
    Serial.println("Publish failed");
  }
 }

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

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

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

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

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

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
```

```
for (int i = 0; i < payloadLength; i++) {
  //Serial.print((char)payload[i]);
  data3 += (char)payload[i];
}

Serial.println(data3);
  char arr[6][20];
int k=0;
for(int i=0 ;i<6;i++)
{
  if(k<=data3.length()){
    for(int j=0 ;j<20;j++)
  {
    if(data3[k]!='$'){
      arr[i][j] = data3[k++];
  }
  else {
    arr[i][j]='\0';
    k++;
    break;
  }
  }
}
else
break;
}

Serial.println(arr[0]);
if(strcmp(arr[0],"clearAlert")==0)
{
  f=0;
}
else if(strcmp(arr[0],"ALERT")==0){
  f=1;
  alert=arr[1];
}
```

```
  else {
     k=0;
    for(int i=0;i<3;i++)
    {
      strcpy(compartment_med[i] , arr[k++]);
      strcpy(compartment_count[i] , arr[k++]);
    }
  }
  data3="";
}
```

CIRCUIT OF IOT DEVICE USING WOKWI SIMULATOR :

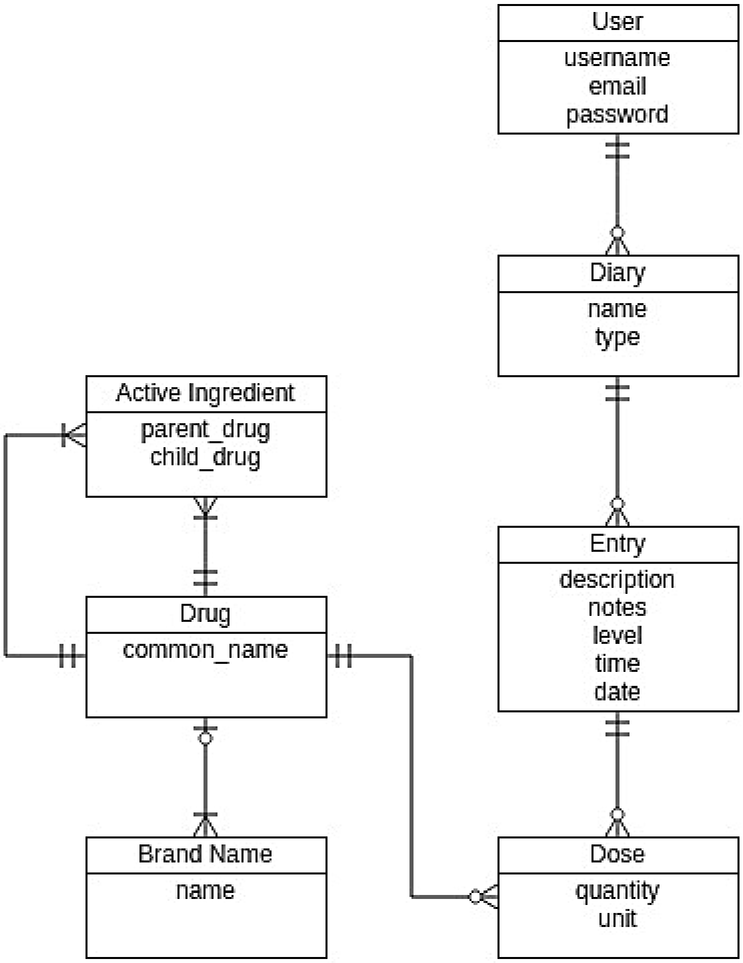


- **End devices** : Physical objects are the “things” that form the IoT. They can be small as the tip of the pin and huge as a dragline: any material object can be transformed into a connected device with corresponding elements (sensors, drives, and relevant software).
- **Software** : The IoT software provides connectivity with clouds, data collection, integration of devices, and real-time data analysis. Data visualization capabilities along with user interfaces also belong to the IoT

software.

- **Communication channels** : Choosing an appropriate communication solution is critical to building IoT systems. The selected technology determines the ways of receiving and sending data to/from the IBM Cloud.
- **Platforms**. IoT devices are able to inform users about the status of sessions. All the sensory data is collected, analyzed, and represented to users through the IoT platforms.

7.3 DATABASE SCHEMA



8. TESTING

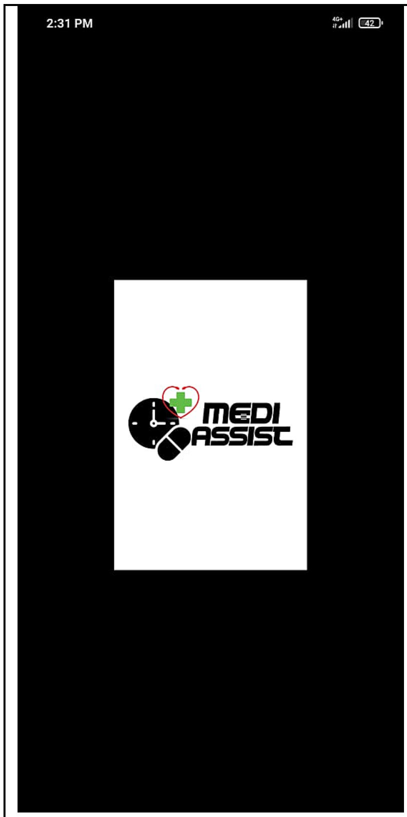
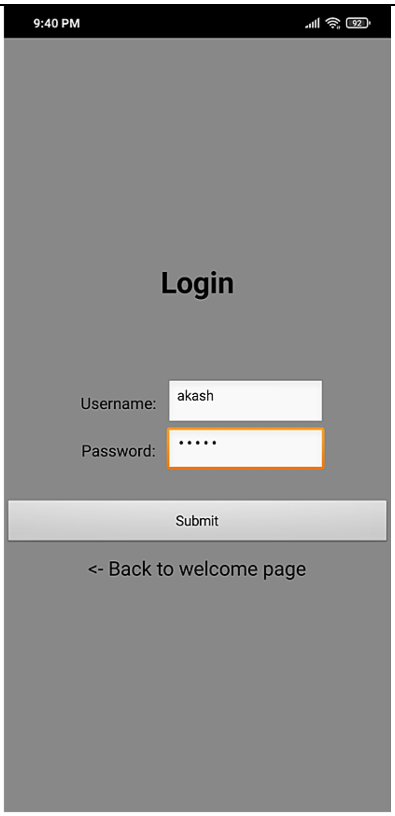
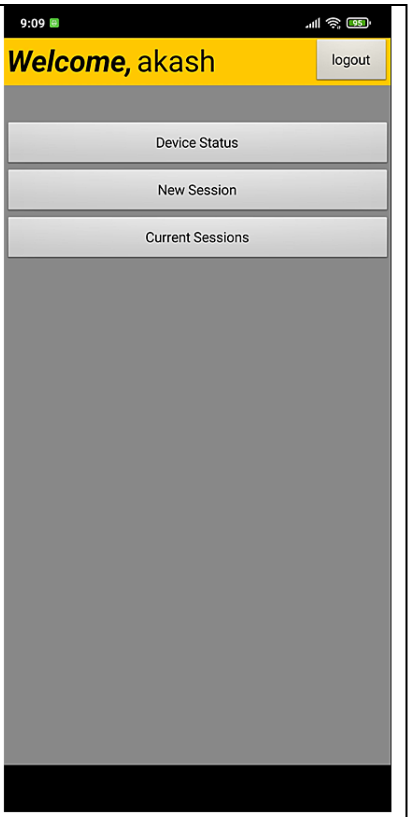
8.1 TEST CASES

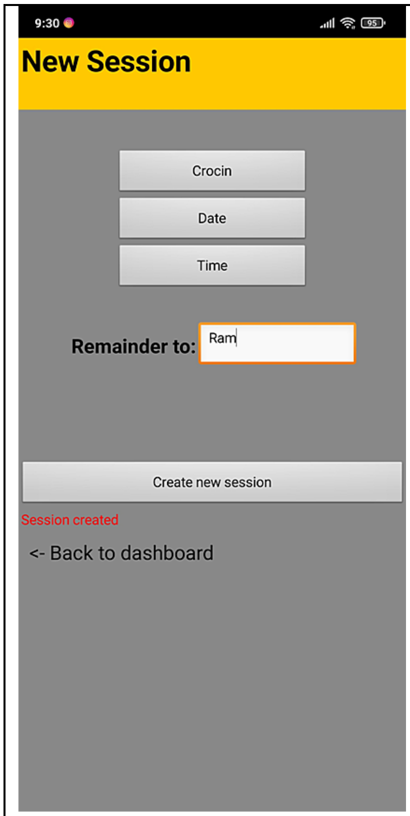
Through the series of test cases, we were able to achieve the expected result through trial and error method.

SCENARIO	TEST STEP	EXPECTED RESULT	OUTCOME
REGISTRATION	A user can register their application using email address.	Email address of the user should be registered.	Email address registered
LOGIN	To login into the application, user can use email and password which they have used toregister	User should be able to login into Medi-Assist app.	User able to login into the application
CREATE SESSIONS	User can register their medicines and alarmsas required for the patient	In the dashboard, they can click new sessions to register their medicines.	User able to create sessions
SESSION STATUS	As a user, we could observe all the current schedules for medications.	By clicking current sessions, application should show the current schedules	User able to see the current schedules by clicking current sessions
ACCESS DATA	User(caretaker) can retrieve the details of the medication given to patients	Application should allow user to know the status of the schedules	User able to see the status of the schedules
CONNECTION WITH IoT DEVICE	User's list of scheduled medicine are sent to the IOT device through Text-to-Speech Device.	Application should connect with desired IoT device and give alarm for the person to take medicine.	IoT device connected with the application
CONNECTION	Through Cloud service	IBM Cloud should	Through Cloud, the

OF IoT DEVICE WITH IBM CLOUD	the IOT device fedwith details of medicine for the patients.	provide the medicine name and time of alarm to the IoT device	details of medicine is delivered to the IoT device
ALARM	It alerts the patient through the IOT device.	IoT device should alert the person at the scheduled time	IoT device alerts through text to speech
LOGOUT	User can logout of the application.	User should be able to logout of the application.	User able to logout from the application

OUTCOME FROM TEST CASES :

 	 Login Username: akash Password: ••••• Submit <- Back to welcome page	 Welcome, akash logout Device Status New Session Current Sessions
1	2	3



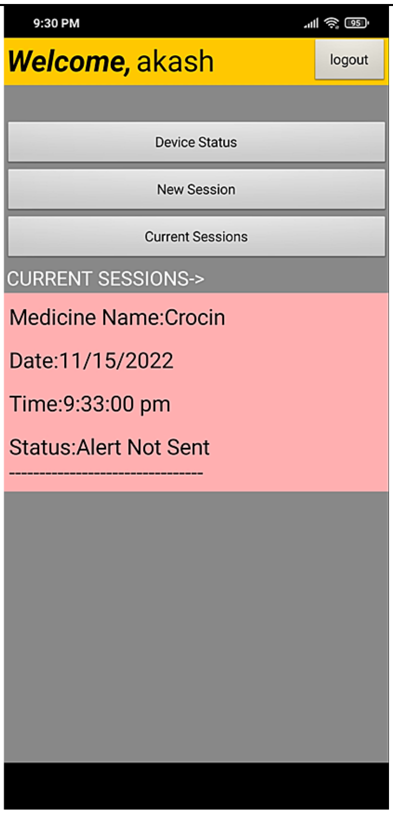
New Session

Remainder to: Ram

Session created

<- Back to dashboard

4



Welcome, akash

CURRENT SESSIONS->

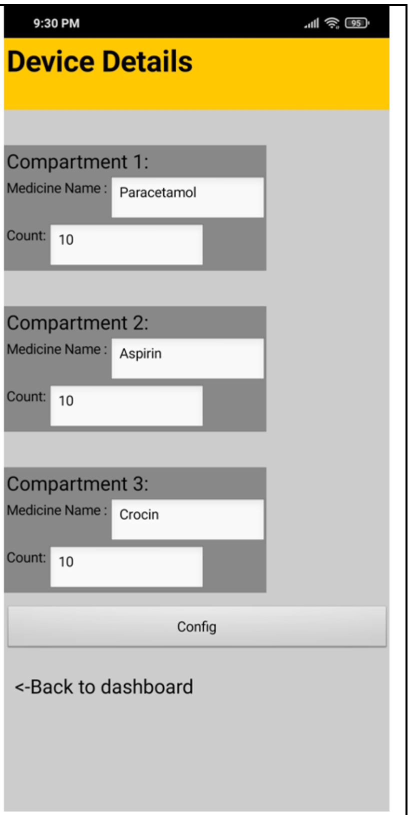
Medicine Name:Crocin

Date:11/15/2022

Time:9:33:00 pm

Status:Alert Not Sent

5



Device Details

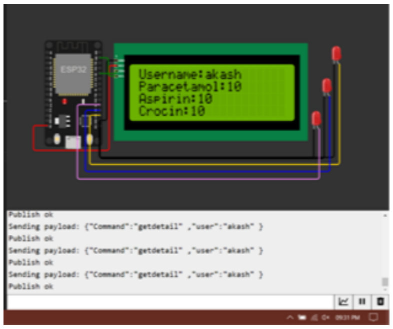
Compartment 1:
Medicine Name : Paracetamol
Count: 10

Compartment 2:
Medicine Name : Aspirin
Count: 10

Compartment 3:
Medicine Name : Crocin
Count: 10

<-Back to dashboard

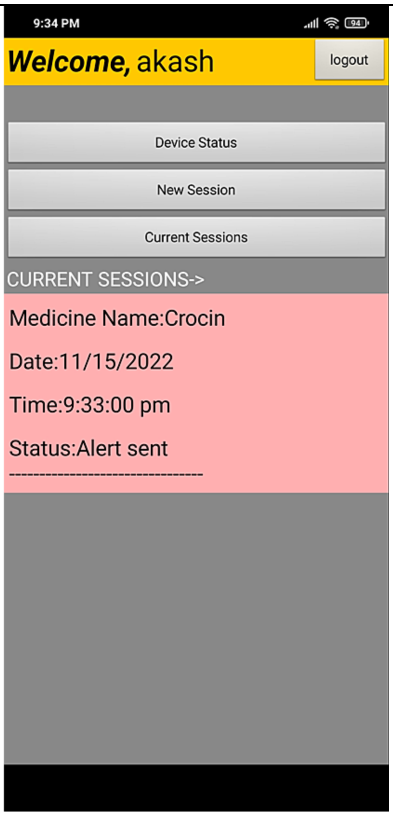
6



```
Username:akash
Paracetamol:10
Aspirin:10
Crocin:10
```

```
Publish ok
Sending payload: {"Command":"getdetail" ,"user":"akash" }
Publish ok
Sending payload: {"Command":"getdetail" ,"user":"akash" }
Publish ok
Sending payload: {"Command":"getdetail" ,"user":"akash" }
Publish ok
```

7



Welcome, akash

CURRENT SESSIONS->

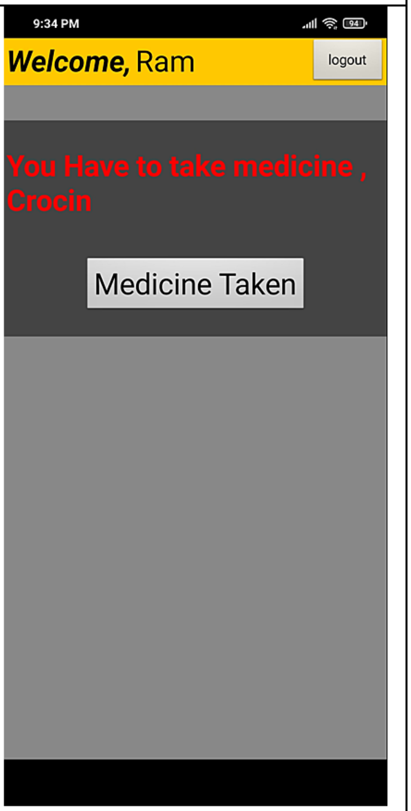
Medicine Name:Crocin

Date:11/15/2022

Time:9:33:00 pm

Status:Alert sent

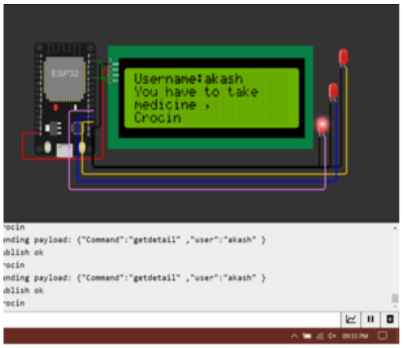
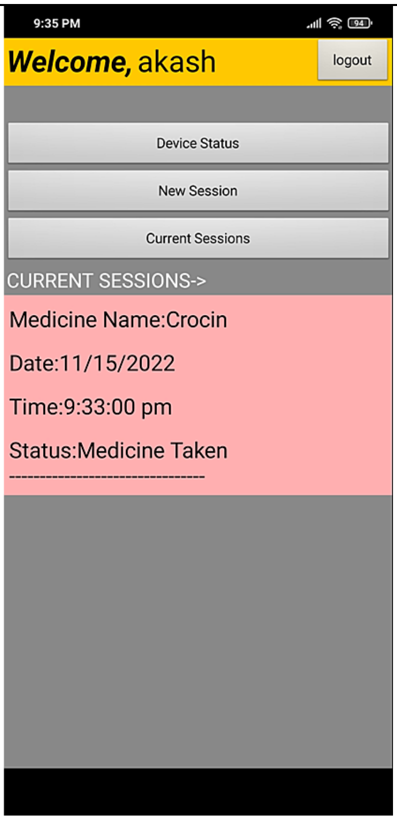
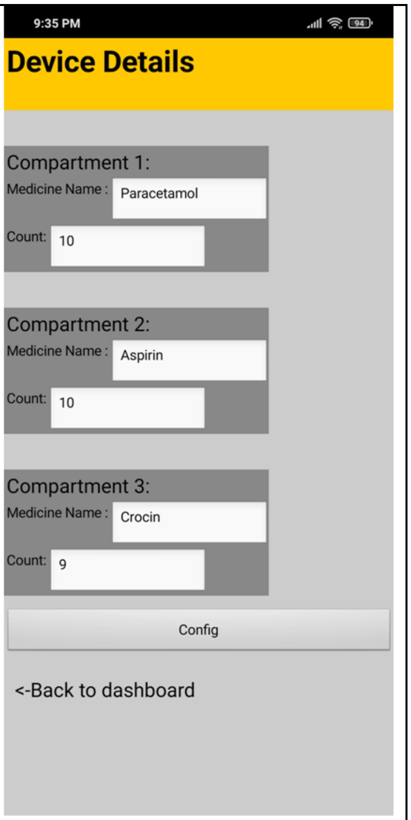
8



Welcome, Ram

You Have to take medicine , Crocin

9


Username:akash
You have to take
medicine ,
Crocin
`rocin`
`nding payload: {"Command":"getdetail" ,"user":"akash" }`
`blish ok`
`rocin`
`nding payload: {"Command":"getdetail" ,"user":"akash" }`
`blish ok`
`rocin`
 	
	
	 9:35 PM
	94
	logout
	Device Status
	New Session
	Current Sessions
	Welcome, akash
	CURRENT SESSIONS->
	Medicine Name:Crocin
	Date:11/15/2022
	Time:9:33:00 pm
	Status:Medicine Taken
	-------------------------------- 	
		
		
		Device Details
		Compartment 1:
		Medicine Name : Paracetamol
		Count: 10
		Compartment 2:
		Medicine Name : Aspirin
		Count: 10
		Compartment 3:
		Medicine Name : Crocin
		Count: 9

10	11	12

8.2 USER ACCEPTANCE TESTING

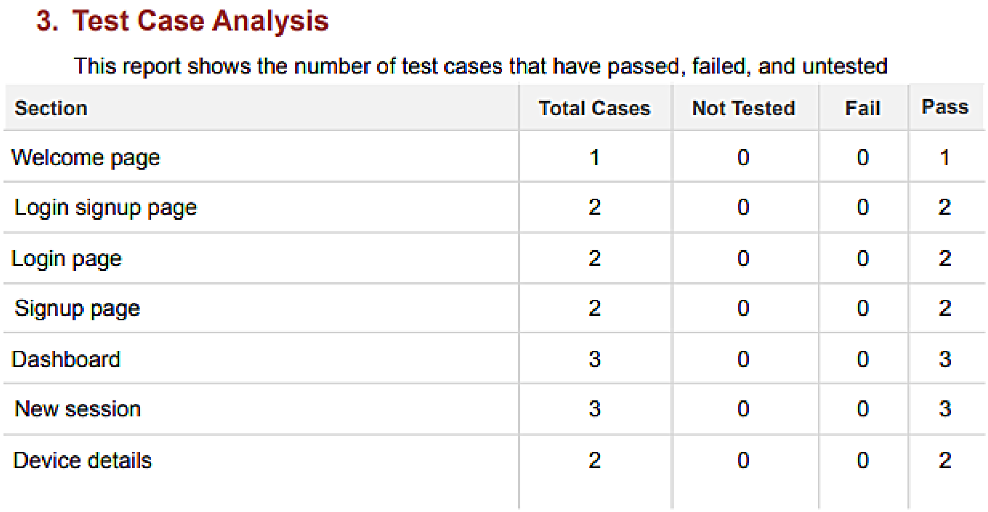
1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the GetMeds application which was developed for the project Personal assistance for senior citizens who are self-reliant 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	5	2	2	3	12
Duplicate	1	0	5	0	6
External	2	1	0	1	4
Fixed	6	1	3	10	20
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	1	2	2	1	6
Totals	15	6	14	16	51



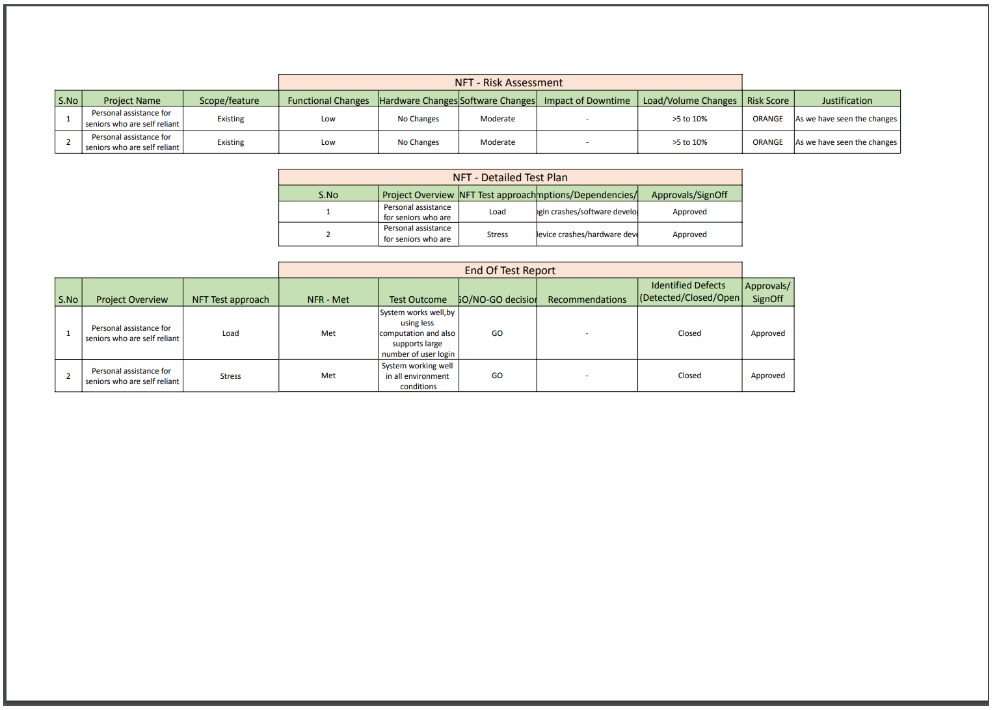
3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Welcome page	1	0	0	1
Login signup page	2	0	0	2
Login page	2	0	0	2
Signup page	2	0	0	2
Dashboard	3	0	0	3
New session	3	0	0	3
Device details	2	0	0	2

9. RESULTS

9.1 PERFORMANCE METRICS



			NFT - Risk Assessment						
S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1	Personal assistance for seniors who are self reliant	Existing	Low	No Changes	Moderate	-	>5 to 10%	ORANGE	As we have seen the changes
2	Personal assistance for seniors who are self reliant	Existing	Low	No Changes	Moderate	-	>5 to 10%	ORANGE	As we have seen the changes

NFT - Detailed Test Plan				
S.No	Project Overview	NFT Test approach	mptions/Dependencies/	Approvals/SignOff
1	Personal assistance for seniors who are	Load	gin crashes/software develo	Approved
2	Personal assistance for seniors who are	Stress	levice crashes/hardware dev	Approved

			End Of Test Report					
S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	O/NO-GO decisio	Recommendations	Identified Defects (Detected/Closed/Open	Approvals/ SignOff
1	Personal assistance for seniors who are self reliant	Load	Met	System works well,by using less computation and also supports large number of user login	GO	-	Closed	Approved
2	Personal assistance for seniors who are self reliant	Stress	Met	System working well in all environment conditions	GO	-	Closed	Approved

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES :

- Improved Patient's Diagnostics and Treatment with IoT
- Enhanced Patient Satisfaction and Engagement
- Support for Remote Monitoring for Chronic Diseases with connected devices
- Improving Patient Care
- Decreases cost while improving Patient care outcomes

DISADVANTAGES :

- Cyber security risks in Healthcare
- Lack of Empathy in Doctor and Patient Interaction
- Frustration with Poor implementation
- Risk of alteration of device functionality
- Chances of Miscommunication

11. CONCLUSION

IoT changes the way the facilities are delivered to the healthcare industry. These technologies improves the medical health of a person, causing a larger effect by bringing together minor changes.

We can consider an IoT unit as a device with a sensor that can interact with the physical world and send information to the Internet. All these Iot based healthcare devices can communicate with each other to take important actions that would provide timely help to almost all-aged people especially senior citizens. After collecting passive data, IoT healthcare devices would send this critical information to the cloud so that doctors can act upon it.

Elderly people who use this Medical Assistance using IoT device which have lot of potential to make our seniors more self-reliant without worrying about their oldage medical illnesses like memory lapses. Thus, IoT-based healthcare services not only improve a patient’s health and help in critical situations but also the productivity of health employees and healthcare organization's workflows.

12. FUTURE SCOPE

The Personal Assitance for Healthcare using IoT device is being used for some great work in the healthcare industry. From monitoring the patients to virtually assisting them, mobile applications of IoT in healthcare are immensely vast and diverse. IoT devices collect massive amounts of data about patients’ illnesses that might take days to collect manually. The data collected through these devices can be used for statistical study. Medical IoT devices and applications can gather vital data and transfer it to doctors and health personnel for real-time tracking. These mobile applications and IoT devices can also send notifications regarding a patient’s critical conditions irrespective of place, and time. There are various ways to use "Medi-Assist" like application :

- Treatments with reduced errors
- Cost reduction in treatments
- Availability of specialist in remote locations
- Improves elderly people's health care

The need for IoT in healthcare is huge and it can help the industry immensely. It can reach every patient from all over the world and connect doctors with patients. There is no denying that IoT has already made a huge impact and is only set to grow further.

13. APPENDIX

SOURCE CODE
GITHUB & PROJECT DEMO LINK

https://wokwi.com/projects/347400182013362771

https://github.com/IBM-EPBL/IBM-Project-19750-1659705714

https://drive.google.com/file/d/16jaOxHxej1Ib1KmIl8C5CLBeFKxHtnp5/view?usp=sharing