M.KUMARASAMY
COLLEGE OF ENGINEERING
NAAC Accredited Autonomous Institution
Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 & ISO 14001:2015 Certified Institution
Thalavapalayam, Karur - 639 113.

A Project Report

on

PERSONAL ASSISTANCE FOR SENIORS WHO ARE SELF-RELIANT

Submitted in partial fulfilment for the award of the degree1

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of

Mrs. C. SELVARATHI M.E.,
Assistant Professor/CSE

Submitted by

TEAM ID: PNT2022TMID15483

927619BCS4070 – MANOJ. S A

927619BCS4006 – AKASH. K

927619BCS4012 – ARUNPRASATH. T

927619BCS4075 – MOHAMAD SULAIMAN. S

927619BCS4107 – SANTHOSH KUMAR. K

NAALAIYA THIRAN – EXPERIENTIAL PROJECT BASED LEARNING INITIATIVE

18CSE040L - PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTERPRENURSHIP

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
M.KUMARASAMY COLLEGE OF ENGINEERING, KARUR
(Autonomous)
Karur - 639 113

November, 2022

ABSTRACT

This is a Web based application in which an automatic alert ringing system is implemented. It focuses on doctor and patient interaction. Patients need not remember their medicine dosage timings as they can set an alert on their dosage timings. The alert can be set for multiple medicines and timings including date, time and medicine description. A voice commands notification will be sent to them through email or message inside the system preferably chosen by the patients. The system focuses on easy navigation and good user interface. Many such Medicine Reminder Systems have been developed where a new hardware is required but in our work we have made an attempt to develop a system which is economical, time-saving and supports medicine adherence.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This is a Web based application in which an automatic alert ringing system is implemented. It focuses on doctor and patient interaction. Patients need not remember their medicine dosage timings as they can set an alert on their dosage timings. The alert can be set for multiple medicines and timings including date, time and medicine description. A voice commands notification will be sent to them through email or message inside the system preferably chosen by the patients. The system focuses on easy navigation and good user interface. Many such Medicine Reminder Systems have been developed where a new hardware is required but in our work we have made an attempt to develop a system which is economical, time-saving and supports medicine adherence.

1.1 PROJECT OVERVIEW

In the contemporary day life style people have no time to spend with their failing such a busy life it‘s difficult to keep an isolated day out of their busy schedule for the doctor for Consistent medical checkups and taking medicines at time. There is a necessity for new idea and technology which helps in saving their time. Medicine Reminder application Project is a native python application meant to aid the forgetful and busy with remembering to take their daily medicine. It is designed for users who need a little help keeping track of their medicine schedule and who are dedicated to keeping the schedule. The application allows the user to store pill objects and multiple alerts for those pills. Alerts have one time of day and can occur on multiple days of the week. The user is able to view their pills in a today view and can select date to view medicines. The remarkable problem is that patients forget to take the proper medicines in proper proportion and in proper time. medicines adherence, which refers to the degree or extent to which a patient takes the right medicines at the right time according to a doctor’s prescription, has recently emerged as a serious issue because many studies have reported that non-adherence may critically affect the patient, thereby raising medical costs. medicines non adherence is a common, complex, and costly problem that contributes to poor treatment outcomes and consumes health care resources. So we are introducing an this application whose objective is to remind the patients of their dosage timings through Alarm Ringing system so that they can stay fit and healthy. This application focuses on the people who forget to take medicines on time. It can be helpful in defence sector and can spread health care awareness. It is life-saving and time saving application which is easy to use and provides a good user interface.

1.2 PURPOSE

“Personal Assistance for Seniors Who Are Self-Reliant” is a process of managing the information of the patients and medicines. This system will replace the manual maintenance system and will help the authorities to keep track of the patient’s details. The proposed model enables users to improve health related risks and reduce health care costs by reminding to take medicines at time, collecting, recording and analyzing data in real time efficiently. With the help of this proposal the time of both patients and doctors are saved and doctors can also help in emergency scenario as much as possible.

CHAPTER 2

LITERATURE REVIEW

2.1 EXISTING PROBLEM

2.1.1 TITLE: SMART MEDICINE REMINDER BOX

AUTHOR: Sanjay Bhati,Harshid Soni

In day-to-day life most of the people need to take medicines which was not there in past couple of years and the reason behind this is diseases are increasing in large amount. So sooner or later many people come in contact with these diseases. Some diseases are temporary diseases while many are permanent life threatening diseases. Life threatening diseases gets mixes with the human body in such a way that they can't leave the body ever and they increases in rapid time. Life span of humans became less because of such diseases and to overcome or to live a better life we need to take medicines regularly and also in large amount. We need to be in advice of Doctor who tells us to take desired pills in desired way so that patients face problems like forgetting pills to take at right time and also when Doctor changes the prescription of medicine patients have to remember the new schedule of medicine. This problem of forgetting to take pills at right time, taking wrong medicines and accidentally taking of expired medicine causes health issues of patient and this leads to suffer from unhealthy life. Our project is to made Arduino-Uno based Smart medicine box which uses Real time clock. The new awaited feature in our project is our system is sensible that patient has taken medicine or not and thus the patient can't postpone the time on which he needs to take pills. It is compulsory for the patient to take pills from the box at the right time otherwise our systems continues to make large sound until the medicine is taken out from the box. This notification feature adds life years to the patient and thus this thing is not available in any device which is the necessity for present days.

2.1.2 TITLE: ARDUINO BASED MEDICINE REMINDER

AUTHOR: Urvashi Sharma, Chetna Chauhan, Himani Sharma,Anjali Sharma

A large number of patients fail to comply with their prescribed medication schedules. This can cause disease complications, lower quality of life, and even mortality. To overcome these issues the automatic medicine reminder is used. This system is very simple to operate and update therefore a person of any age group can use it. This system can also be useful in hospital where number of patients is present and sometimes it is difficult to remember the medicine and dosage by the staff. So this system with some updates can also be used in hospitals. The problem such as 1) maintaining the regularity of prescribed dosage is difficult to be remembered in busy schedule 2) remembering the name of medicine to be taken is really difficult 3) due to above two reasons the patient's life can get more complicated. These above problems are arising to everyone due to non adherence of medications. Therefore, there is a growing need and urgency for in-home healthcare devices and technologies in order to provide patients with the electronic tools to support medication self-management.

2.1.3 TITLE: ALARM SYSTEM FOR MEDICINE REMINDER BASED ON MICROCONTROLLER

AUTHOR: Ni Ni San Hlaing, San San Naing

A large number of patients fail to comply with their prescribed medication schedules. This causes disease complications, lower quality of life and even mortality. To overcome these issues the automatic medicine reminder is used. This system is very simple to operate and update therefore a person of any age group uses it. This system is also useful in hospital where number of patients is present and sometimes it is difficult to remember the medicine and dosage by the staff. So this system with some updates is also used in hospitals. The problem such as maintaining the regularity of prescribed dosage is difficult to be remembered in busy schedule, remembering the name of medicine to be taken is really difficult and due to above two reasons the patient's life can get more complicated. These above problems are arising to everyone due to non-adherence of medications. Therefore, there is a growing need and urgency for in-home healthcare devices and technologies in order to provide patients with the electronic tools to support medication self-management.

2.1.4 TITLE: RASBERRY PI MEDICINE REMINDER E-MAIL ALERT USING IOT

AUTHOR: CH. Santhosha Kumara, Dr. V. Gouthama

In day to day life, people face trouble remembering the pills that need to be consumed. Many times the problem is that the time required to take the medicine is not printed on the packaging of medicine. People also have a habit of forgetting to take pills.

Sometimes medicines go beyond their expiry date. Pervasive healthcare is the next generation form of healthcare and distributed, patient-centric and self-managed care. It is an alternative to the traditional hospitalized, staff and professional-managed care. Pervasive healthcare is based on emerging technologies like the Internet-of-Things (IoT). Health informatics is a developing area to advance health problems, mainly deals with the acquisition, transmission, processing, storage, retrieval, and use of different types of information in the health field. The main acquisition technologies of health information are sensing. Most patients with chronic diseases need to take medications over a prolonged period of time in order to stabilize their conditions. Ensuring that the patients consume the right medication at the appropriate time becomes crucial. This project deals with the time at which the patient needs to take pills. The timing is set to the system initially and it can be changed by the patient according to his/her requirement. The system will start an alarm at that particular time. To make the user-friendly system, a LCD and a keypad is connected to the system, this helps to change pill timings. After having pills, the user will have to put the number of pills removed from the box. When the number of pills left reach a certain number an order for that particular pill is sent by the system to a medical shop through GSM system. Therefore, it is helpful to user to get the pill at particular time and avoid confusion among pills.

2.1.5 TITLE: DEVELOPMENT OF AUTOMATIC REMINDER SYSTEM FOR GERIATRIC MEDICIN INTAKE

AUTHOR:Aisyah Rahimi, Hamimi Zakri, Azira Khalil

Geriatrics is referred to individuals aged from 50 years old and above that require particular regard to their healthcare problem. They are also defined as aged individuals receiving special care to improve their lives . This group of individuals is often involved with medicine intake in their daily life. Their medication needs can be due to various diseases, such as hypertension, diabetes mellitus, Alzheimer's, stroke, and others. Statistic from recent studies shows that approximately 80% of elderly in the United States of American have at least one chronic disease, while 77% of them have two chronic diseases per individual. These diseases can cause almost death factors to the geriatrics every year. Meanwhile, diabetes and hypertension are common diseases associated with geriatrics, where 23% of Americans are affected by diabetes, and 90% of elderly Americans are at risk for hypertension. According to , getting older means that we are more exposed to healthcare problems, and only 41% of elderly aged 65 are free from any diseases. Geriatrics need medication to improve their health and help them in their daily life. The functionality of this project is to help geriatrics in taking their medicine at the exact time. This project also allows geriatrics to consume which medicine at a particular time.

2.2 REFERENCES

1. Sanjay Bhati,Harshid Soni, Smart Medicine Reminder Box, 2017
2. Urvashi Sharma, Chetna Chauhan , Arduino Based Medicine Reminder, 2016
3. Ni Ni San Hlaing, Alarm System For Medicine Reminder Based On Microcontroller, 2019
4. Santhosha Kumara, Dr. V. Gouthama, Rasberry Pi Medicine Reminder E-Mail Alert Using Iot, 2019
5. Aisyah Rahimi, Hamimi Zakri, Development Of Automatic Reminder System For Geriatric Medicin Intake, 2021

2.3 PROBLEM STATEMENT DEFINITION

The existing system requires a lot of manual work which results in taking more time than it should. The operations like updating and synchronizing data are also done manually in the existing system that is not automated and again time-consuming process. These practices are not at all reliable as the one wrong entry can take a lot of time in detection and then there is a correction. Humans are prone to errors and can mistakes often unless it has some inbuilt programs which can take check the input and save from error. The current system and compare with the proposed it is far behind. Every work in the existing is manual and done on the paper.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

This planning of empathy map canvas helps in maintaining all aspects of the idea at the planning stage.

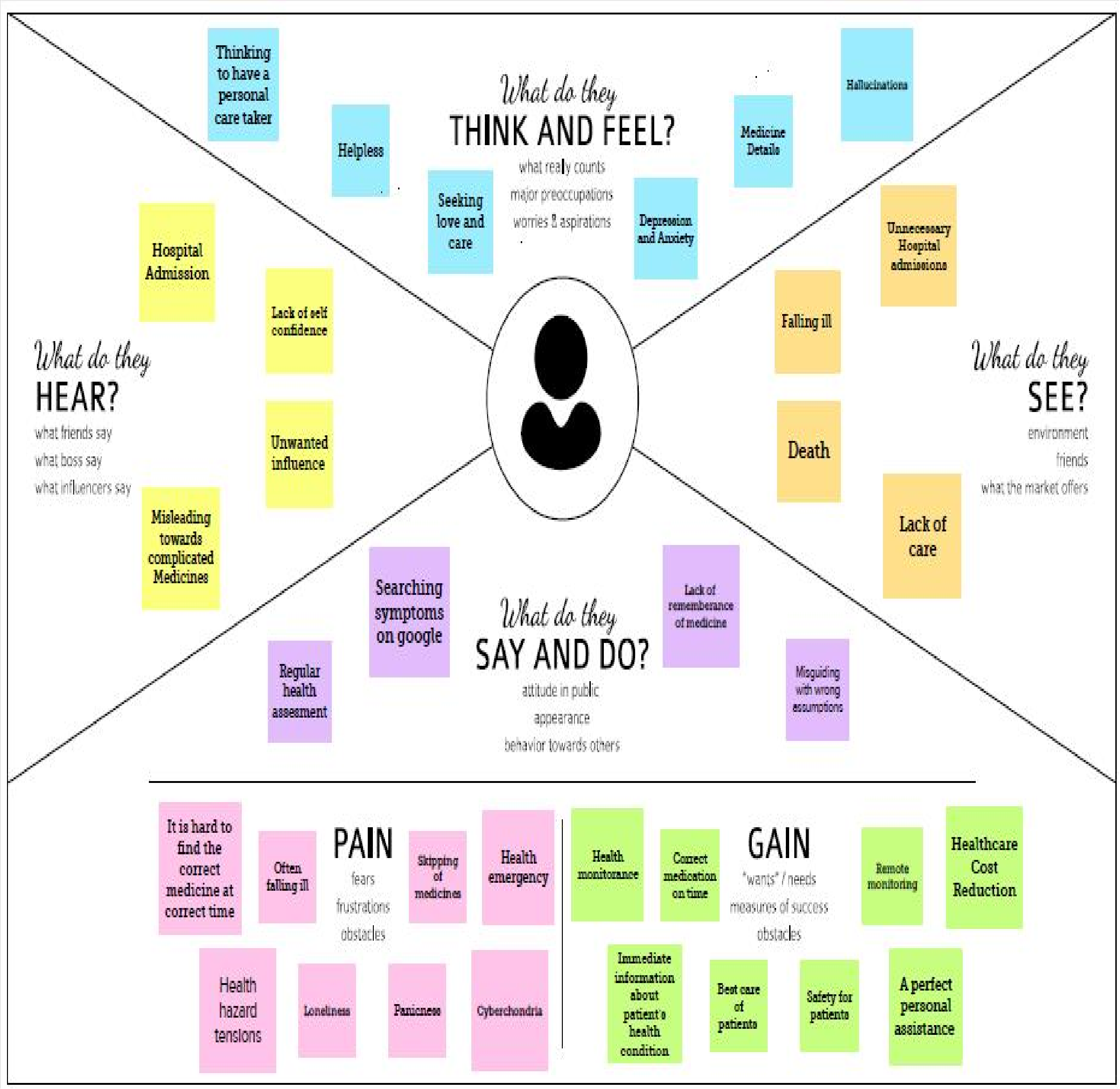
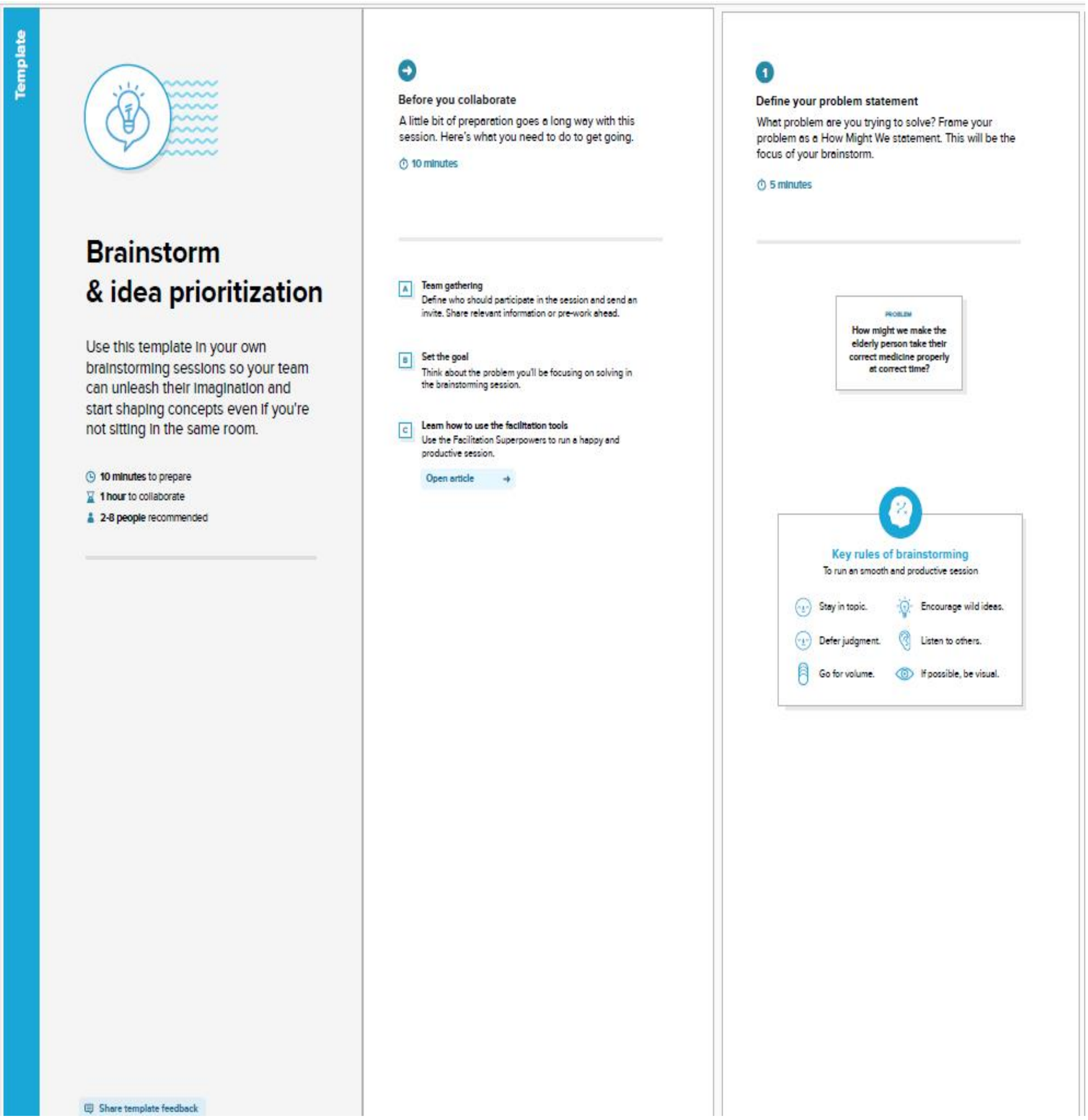


Figure 3.1.1 Empathy Map Canvas

3.2 IDEATION & BRAINSTORMING

Team Gathering, Collaboration and Select the Problem Statement.

Step 1:



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

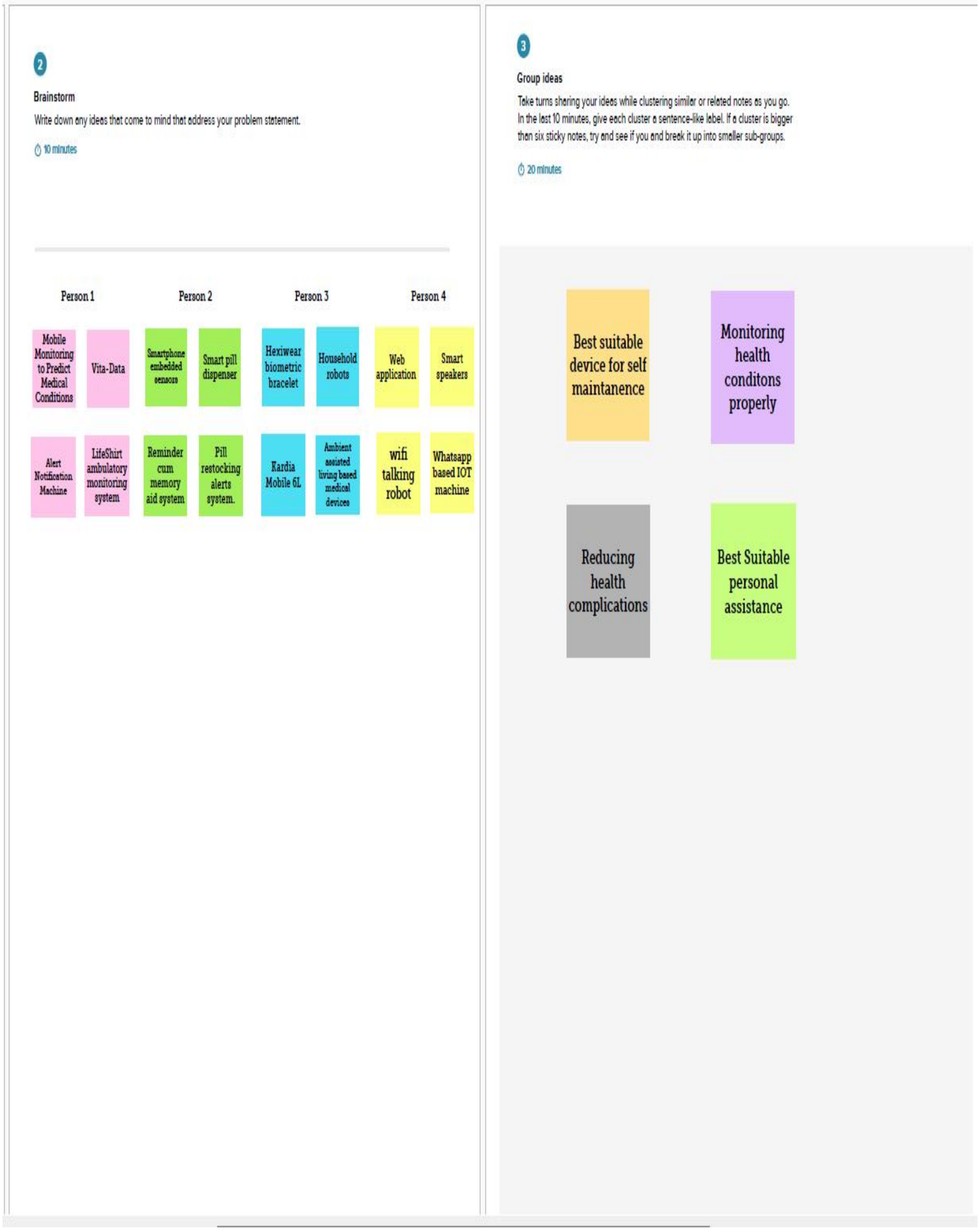
PROBLEM
How might we make the elderly person take their correct medicine properly at correct time?

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Figure 3.2.1 Brainstorming

Step 2:



2

Brainstorm

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

10 minutes

Person 1

- Mobile Monitoring to Predict Medical Conditions
- Vita-Data
- Alert Notification Machine
- LifeShirt ambulatory monitoring system

Person 2

- Smartphone embedded sensors
- Smart pill dispenser
- Reminder cum memory aid system
- Pill restocking alerts system.

Person 3

- Hexiwear biometric bracelet
- Household robots
- Kardia Mobile 6L
- Ambient assisted living based medical devices

Person 4

- Web application
- Smart speakers
- wifi talking robot
- Whatsapp based IOT machine

3

Group ideas

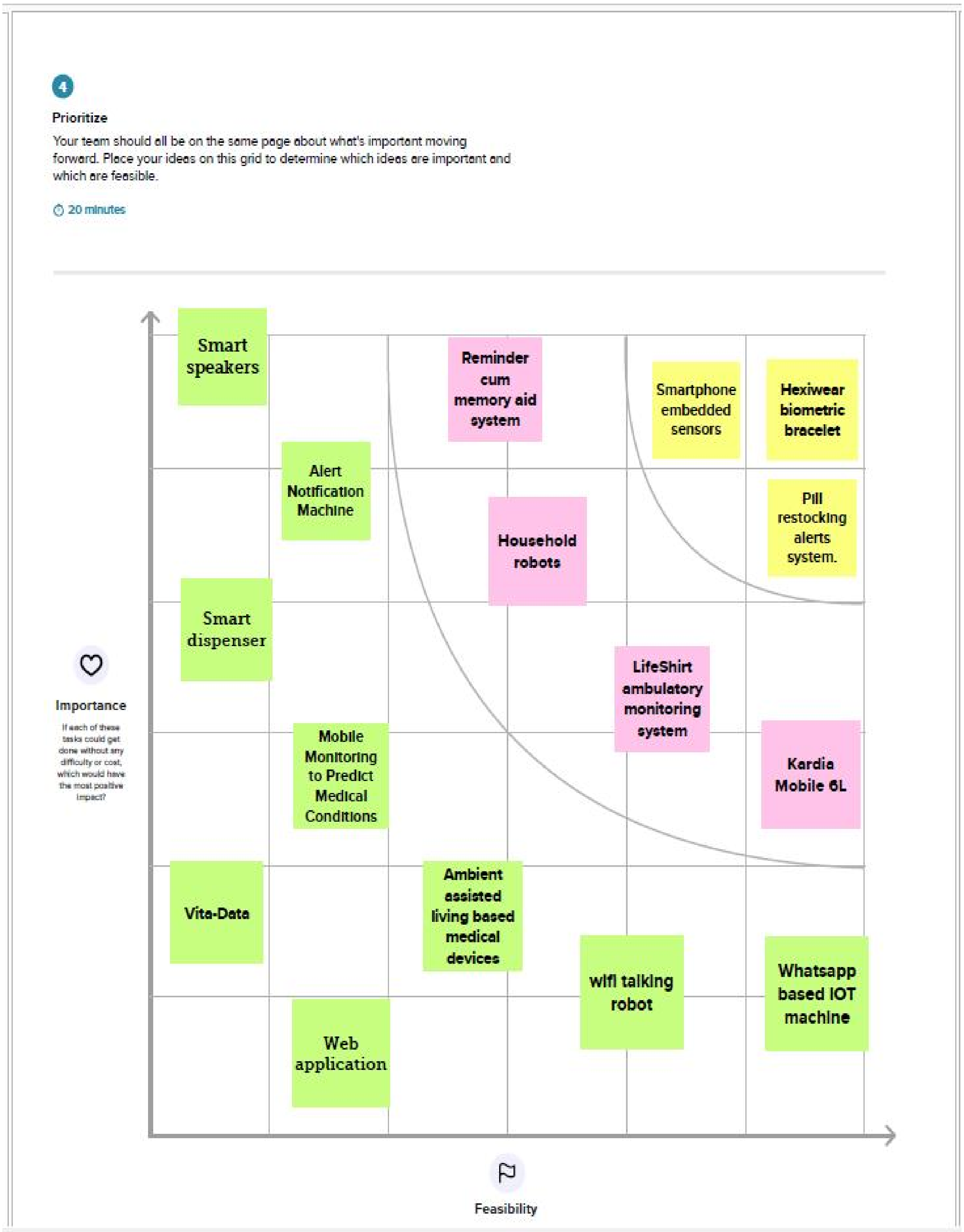
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

- Best suitable device for self maintanence
- Monitoring health conditons properly
- Reducing health complications
- Best Suitable personal assistance

Figure 3.2.2 Idea Listing and Grouping

Step 3:



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?

Smart speakers
Reminder cum memory aid system
Smartphone embedded sensors
Hexiwear biometric bracelet
Alert Notification Machine
Household robots
Pill restocking alerts system.
Smart dispenser
LifeShirt ambulatory monitoring system
Mobile Monitoring to Predict Medical Conditions
Kardia Mobile 6L
Vita-Data
Ambient assisted living based medical devices
wifi talking robot
Whatsapp based IOT machine
Web application
Feasibility

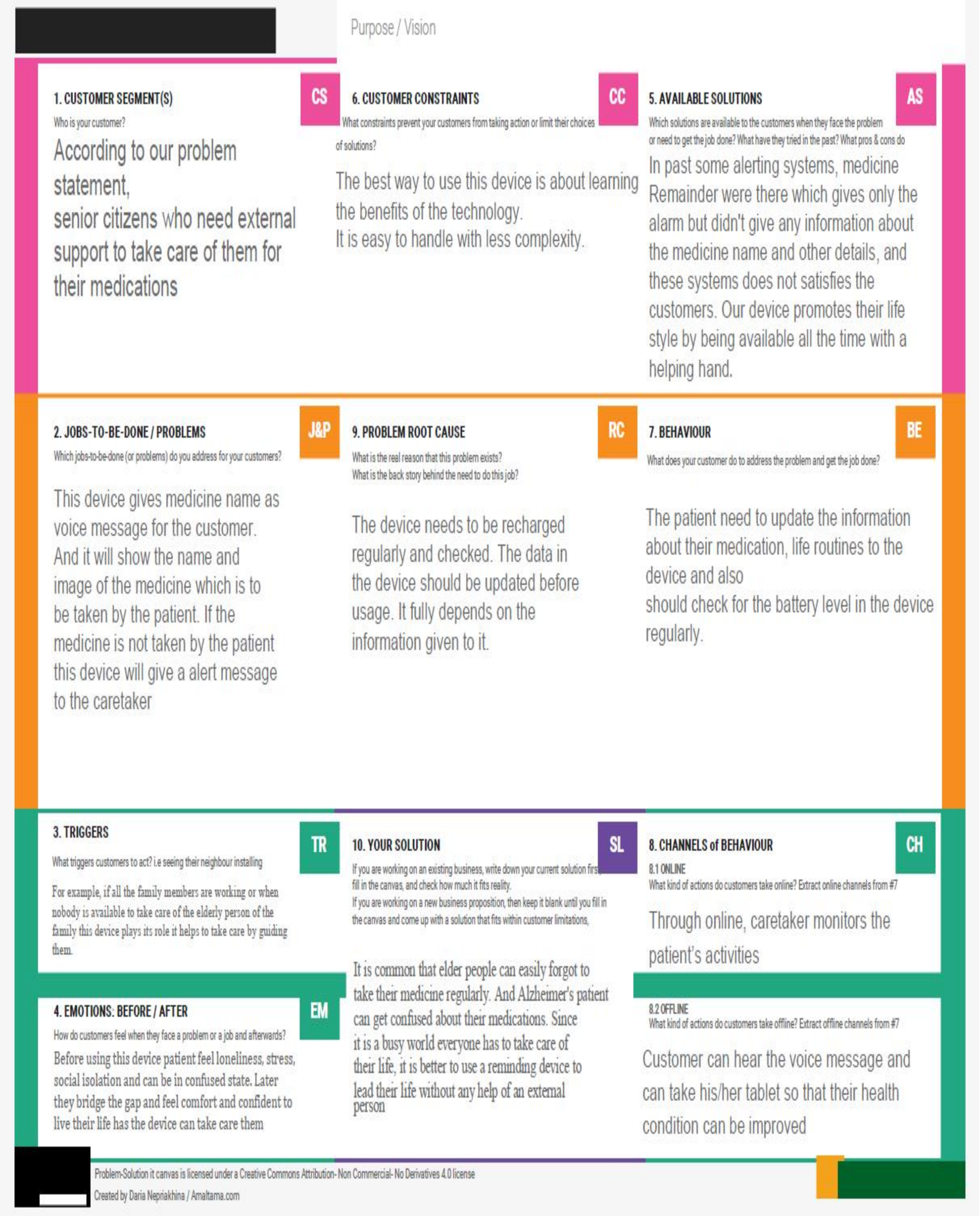
Figure 3.2.3 Idea Prioritization

3.3 PROPOSED SOLUTION

S.No.	PARAMETERS	DESCRIPTION
1.	Problem Statement (Problem to be solved)	In modern society, busy life has made people forget many things in day to day life. The older adults and the people victims of chronicle diseases who need to take the medicines timely without missing have dementia, forgetting things in their daily routine. This application is used to remind tablets on time.
2.	Idea / Solution description	The Medicine reminder system consists of a pillbox provided with a set of Compartments. It is designed in such a way that normal people can use it easily for their medication. The pill box's control system consists of LEDs for giving visual alerts to the patient for medicine. There is a Speaker in the system which alerts the patient in audio form. The alert will be given in the form of SMS to the patient's caretaker.
3.	Novelty / Uniqueness	Going across many reviews on this project, we can conclude that no technology can replace a personal care taker. Still the technologies are trying to invent a system that replaces a personal care taker by providing friendly relationship with patients.
4.	Social Impact / Customer Satisfaction	Going across many reviews on this project, we can conclude that no technology can replace a personal care taker. Still the technologies are trying to invent a system that replaces a personal care taker by providing friendly relationship with patients.
5.	Business Model (Revenue Model)	This application can be given to people who are in need to take care of elderly people by giving medicines on time.It can also be given in subscription.

Table 3.3 Proposed Solution

3.4 PROBLEM SOLUTION FIT



[redacted]

Purpose / Vision

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

According to our problem statement,
senior citizens who need external support to take care of them for their medications

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

The best way to use this device is about learning the benefits of the technology.
It is easy to handle with less complexity.

5. AVAILABLE SOLUTIONS AS

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

In past some alerting systems, medicine Remainder were there which gives only the alarm but didn't give any information about the medicine name and other details, and these systems does not satisfies the customers. Our device promotes their life style by being available all the time with a helping hand.

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers?

This device gives medicine name as voice message for the customer. And it will show the name and image of the medicine which is to be taken by the patient. If the medicine is not taken by the patient this device will give a alert message to the caretaker

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?

The device needs to be recharged regularly and checked. The data in the device should be updated before usage. It fully depends on the information given to it.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

The patient need to update the information about their medication, life routines to the device and also
should check for the battery level in the device regularly.

3. TRIGGERS TR

What triggers customers to act? i.e seeing their neighbour installing

For example, if all the family members are working or when nobody is available to take care of the elderly person of the family this device plays its role it helps to take care by guiding them.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?

Before using this device patient feel loneliness, stress, social isolation and can be in confused state. Later they bridge the gap and feel comfort and confident to live their life has the device can take care them

10. YOUR SOLUTION SL

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,

It is common that elder people can easily forgot to take their medicine regularly. And Alzheimer's patient can get confused about their medications. Since it is a busy world everyone has to take care of their life, it is better to use a reminding device to lead their life without any help of an external person

8. CHANNELS of BEHAVIOUR CH

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

Through online, caretaker monitors the patient's activities

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7

Customer can hear the voice message and can take his/her tablet so that their health condition can be improved

[redacted]

Created by Daria Nepriakhina / Amaltama.com

Figure 3.4.1 Problem Solution Fit

CHAPTER-4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

- **Admin Interface**

  This module used to interact with application that admin has unique username and password. Authorized admin only can access the application. Admin has the responsibility to maintain the application.

- **Patient Interface**

  Thus module for parents to create a account and login account. Patients have to create a new account for using this application. After registration patients get username and password. Registered patients only can use this application.

- **Set alert module**

  It helps in reminding about the medicines. User can add details of his dosage schedules. Using the date field one can enter the starting and ending dates between which he has to take medicines. The time field shows the time of dosage and on that time the alarm will get rung. The user can add the description of the medicine, including name, purpose and other related description. All the information will be saved in the database.

- **Voice based Reminder Alert**

  A regulator is a particular type of clock used for calculating specific time intervals. This timer dynamically monitors the time details. This timer event will activate when ever Time will match with user Set Time interval. Once Timer event call this application produce voice based Medicine remainder alert to elder people effectively.

4.2 NON FUNCTIONAL REQUIREMENTS

Usability

The system shall allow the users to access the system with pc using web application. The system uses a web application as an interface. The system is user friendly which makes the system easy

Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Scalability

Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

Security

A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied.

Performance

The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the requestsubmittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 secondsto appear on the screen.

Reliability

The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week. 24 hours a day.

CHAPTER-5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

DATA FLOW DIAGRAM

A data-flow diagram is a visual representation of how data moves through a system or a process (usually an information system). The DFD additionally gives details about each entity's inputs and outputs as well as the process itself. A data-flow diagram lacks control flow, loops, and decision-making processes. Using a flowchart, certain operations depending on the data may be depicted.

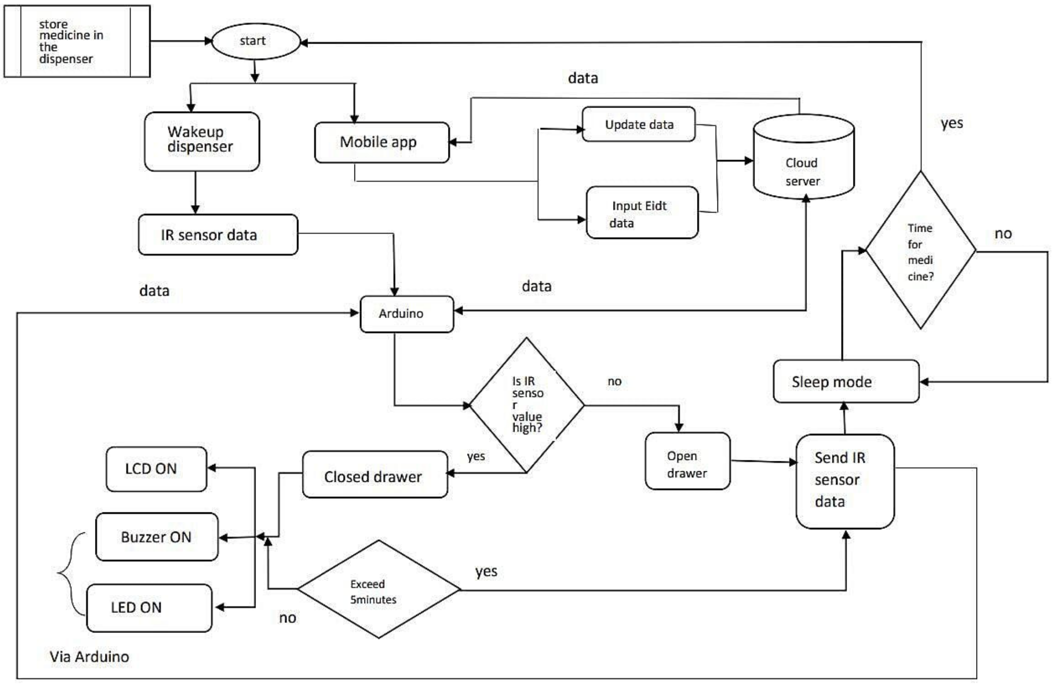


Figure 5.1.1 Data Flow Diagram

Total Data Flow project overview

LEVEL 0

It is also known as a context diagram. It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

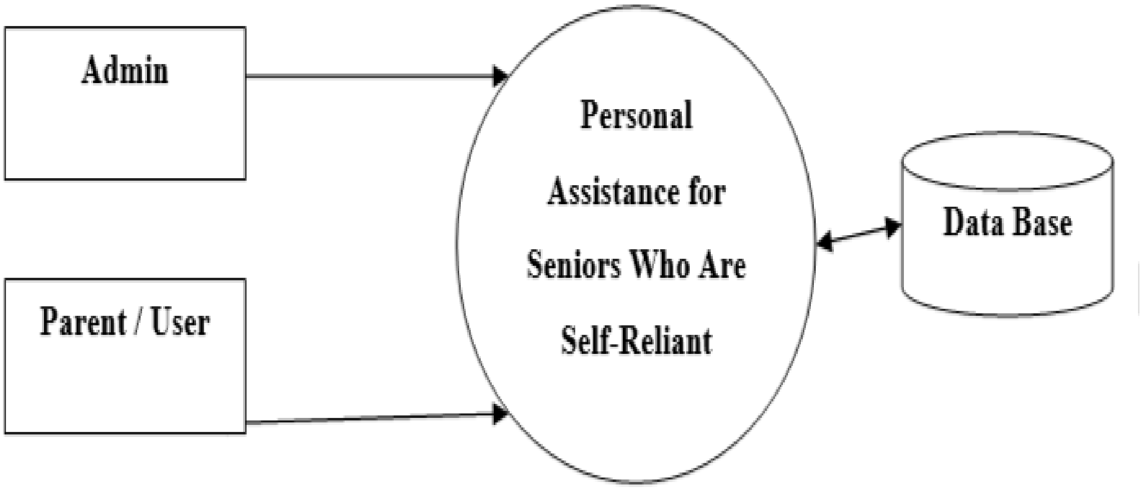


Figure 5.1.2 Data Flow

LEVEL 1

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into sub processes.

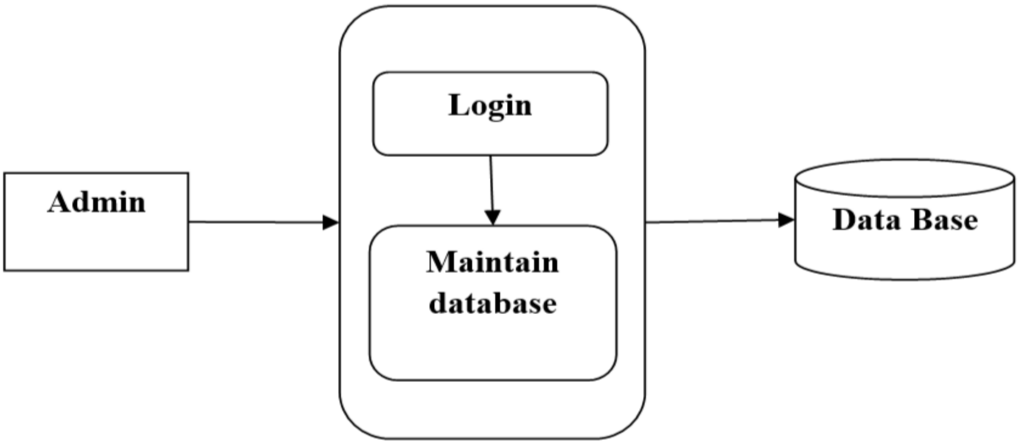


Figure 5.1.3 Admin Login

LEVEL 2

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.

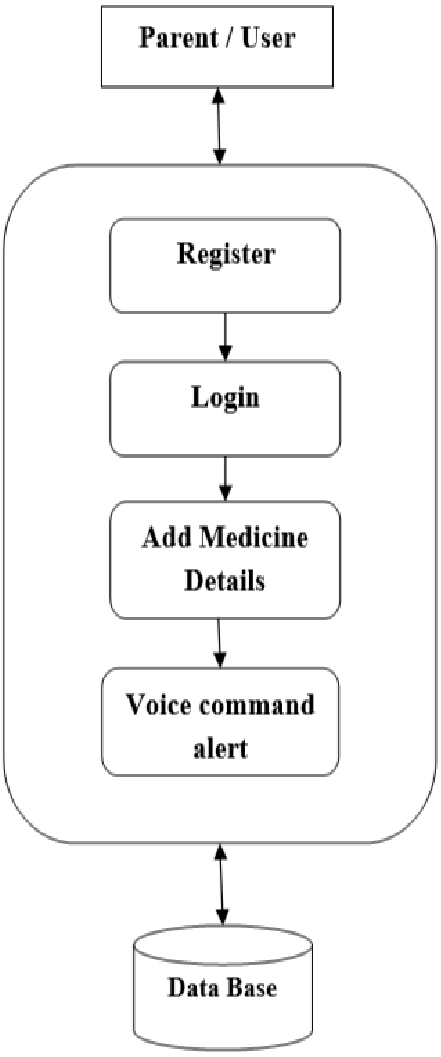


Figure 5.1.4 User Login

5.2 SOLUTION & TECHNICAL ARCHITECTURE

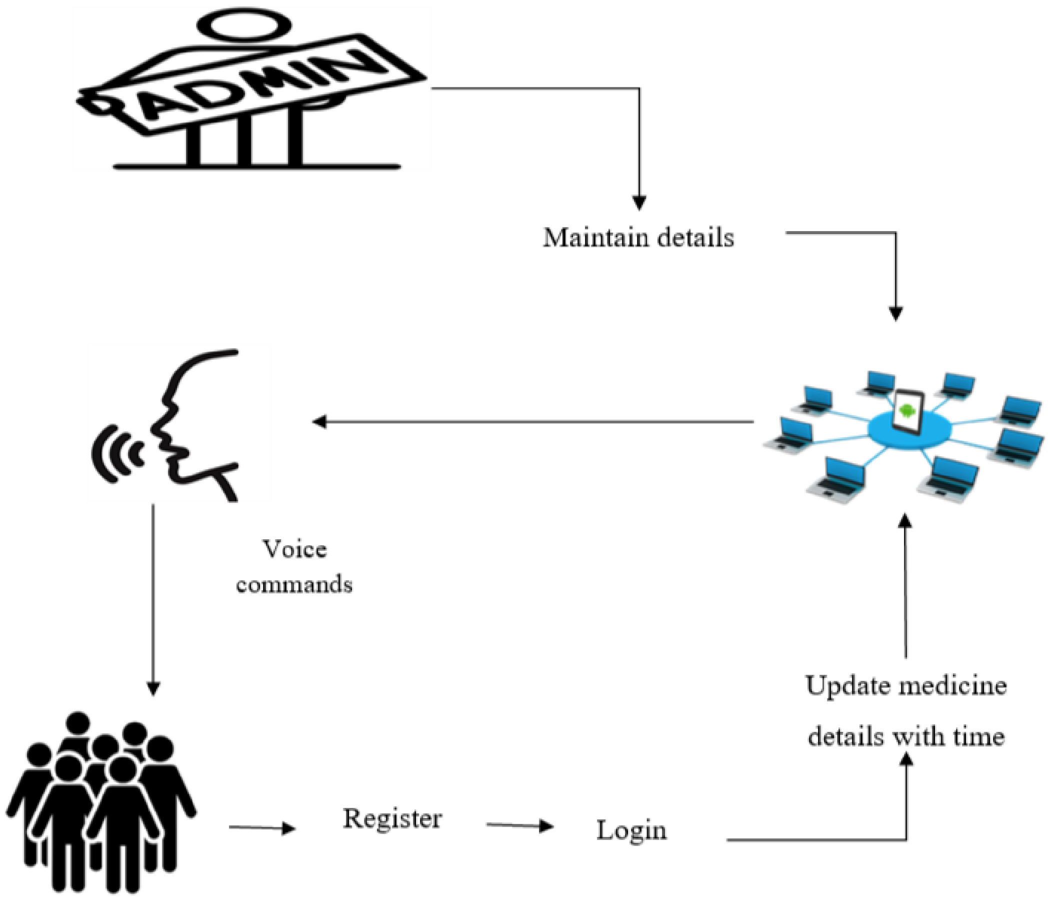


Figure 5.2.1 Solution & Technical Architecture

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	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 can 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 receive 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-2
		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	Log into the application by entering email & password	I can use the credential using my email and password	High	Sprint-1

Administrator	security	USN-6	Administrator Security.	I can provide security.	High	Sprint-1

Table 5.3 User Stories

CHAPTER-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	CUSTOMER REGISTRATION	USN-1	As a User, I can register for the application by entering my mail, password and confirming my password.	3	High	Manoj & Mohamed Sulaiman
Sprint-2	AUTHORIZATION	USN-2	As a user, I will receive confirmation email once I have registered for the application	2	Medium	Santhosh Kumar & Arun
Sprint-3	USER INTERFACE	USN-3	Using Mobile application it is easy receive an alert when the medicine is missed to take and also giving correct medicines at correct time.	3	High	Manoj ,Akash & Mohamed Sulaiman
Sprint-4	SYSTEM DESIGN	USN-4	Uses cloud database to store medicinal reports. Connecting API to the cloud and mobile application. Connecting an IOT device to the cloud.	3	High	Manoj, Akash, Santhosh Kumar, Mohamed Sulaiman , Arun

Table 6.1.1 Sprint Planning & Estimation

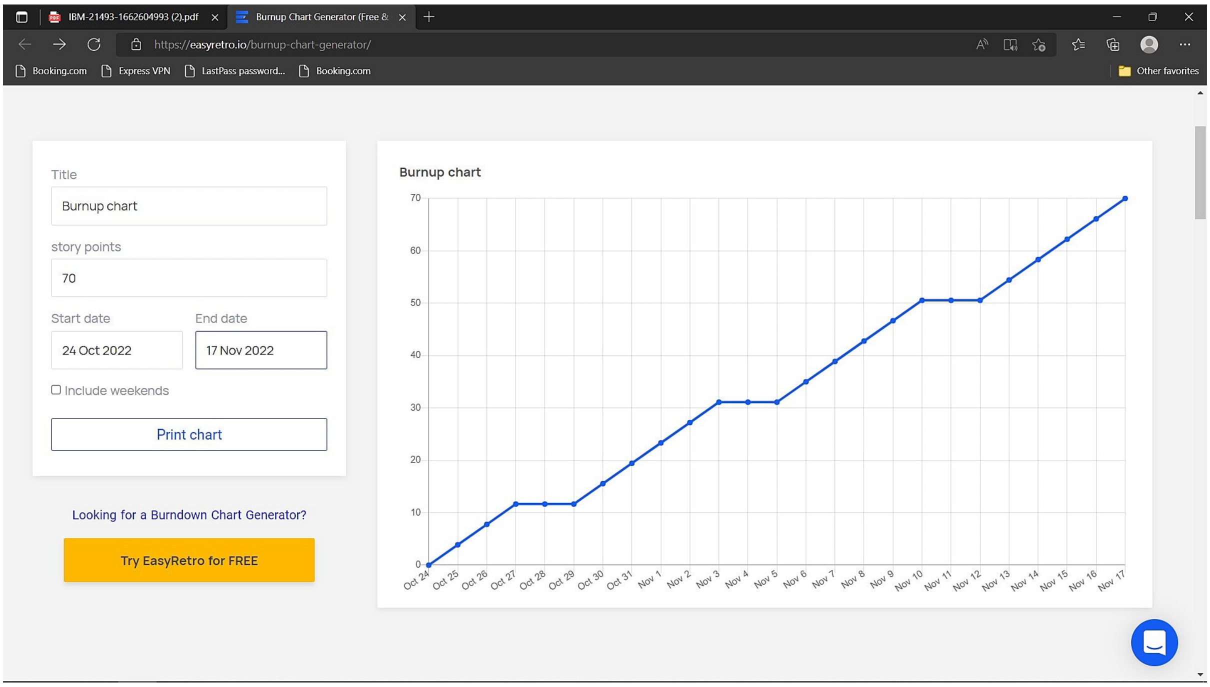
6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	5 Days	24 Oct 2022	28 Oct 2022	15	28 Oct 2022
Sprint-2	10	4 Days	29 Oct 2022	01 Nov 2022	10	01 Nov 2022
Sprint-3	20	6 Days	02 Nov 2022	07 Nov 2022	20	07 Nov 2022
Sprint-4	25	10 Days	08 Nov 2022	17 Nov 2022	25	17 Nov 2022

Table 6.2.1 Sprint Delivery Schedule

6.3 REPORTS FROM JIRA

Burnup Chart



IBM-21493-1662604993 (2).pdf

Burnup Chart Generator (Free &

https://easyretro.io/burnup-chart-generator/

Booking.com Express VPN LastPass password... Booking.com Other favorites

Title

Burnup chart

story points

70

Start date

24 Oct 2022

End date

17 Nov 2022

□ Include weekends

Print chart

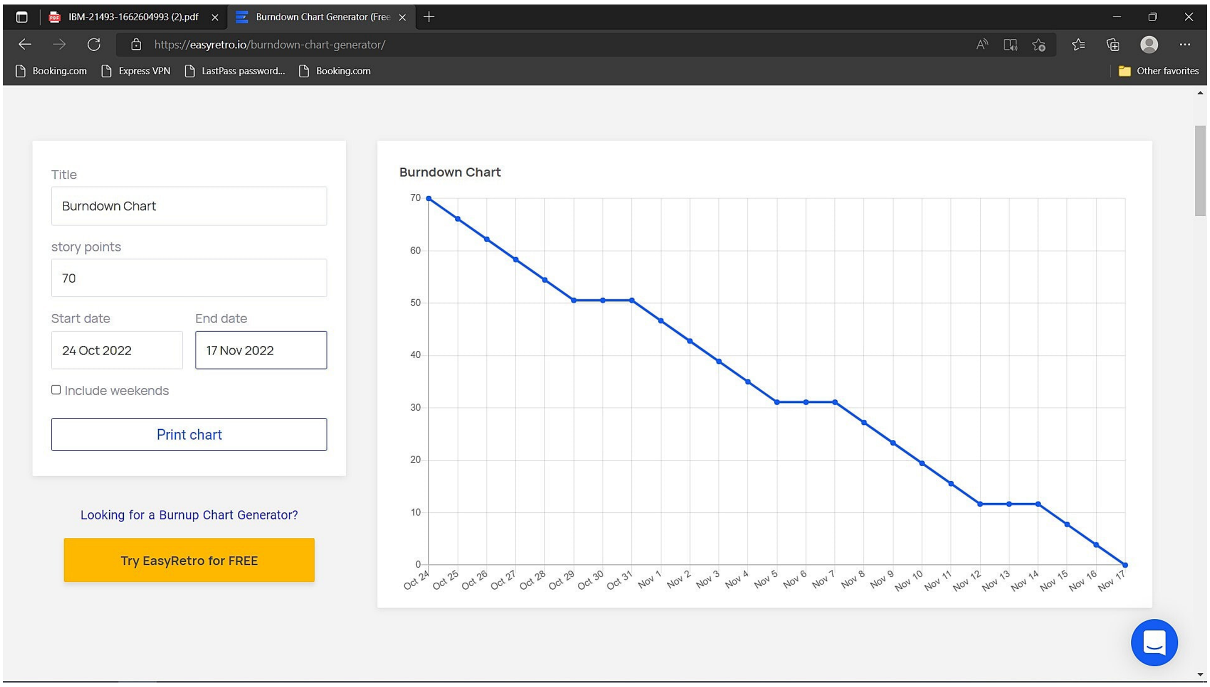
Looking for a Burndown Chart Generator?

Try EasyRetro for FREE

Figure 6.3.1 Burnup Chart

It displays the scope of a project and the work completed

Burndown Chart



IBM-21493-1662604993 (2).pdf
Burndown Chart Generator (Free
https://easyretro.io/burndown-chart-generator/
Booking.com
Express VPN
LastPass password...
Booking.com
Other favorites

Title

Burndown Chart

story points

70

Start date: 24 Oct 2022

End date: 17 Nov 2022

□ Include weekends

Figure 6.3.2 Burndown Chart

It shows work that has been completed in an sprint, and the total work remaining.

CHAPTER-7

CODING & SOLUTIONING

7.1 FEATURE 1

The main accountable feature is that the deployment is the separate admin login to maintain the activity of the users and remainder management.

```
    @app.route("/adminlogin", methods=['GET', 'POST'])
    def adminlogin():
        error = None
        if request.method == 'POST':
           if request.form['uname'] == 'admin' and request.form['password'] == 'admin':

               conn = ibm_db.connect(dsn, "", "")
               pd_conn = ibm_db_dbi.Connection(conn)
               selectQuery = "SELECT * FROM regtb  "
               dataframe = pandas.read_sql(selectQuery, pd_conn)
               dataframe.to_sql('booktb1', con=engine, if_exists='append')
               data = engine.execute("SELECT * FROM booktb1").fetchall()

               return render_template('AdminHome.html', data=data)

           else:
return render_template('index.html', error=error)
```

This feature is completely isolated from the normal user perspective that the admin can have a only access to login and maintain the activity of the process and take note of each user and remind them for the medicine.

7.2 FEATURE 2

The second feature is related to the user that is the cluster of activities that can be performed by the user like user registration and user login.

```
@app.route("/newuser", methods=['GET', 'POST'])
 def newuser():
    if request.method == 'POST':

        name1 = request.form['name']
        gender1 = request.form['gender']
        Age = request.form['age']
        email = request.form['email']
        pnumber = request.form['phone']
        address = request.form['address']

        uname = request.form['uname @app.route("/newuser", methods=['GET', 'POST'])
 def newuser():
    if request.method == 'POST':

        name1 = request.form['name']
        gender1 = request.form['gender']
        Age = request.form['age']
        email = request.form['email']
        pnumber = request.form['phone']
        address = request.form['address']

        uname = request.form['uname']
        password = request.form['psw']
        conn = ibm_db.connect(dsn, "", "")

        insertQuery = "INSERT INTO regtb VALUES ('" + name1 + "','" + gender1 + "','" + Age
+ "','" + email + "','" + pnumber + "','" + address + "','" + uname + "','" + password + "')"
        insert_table = ibm_db.exec_immediate(conn, insertQuery)
        # return 'file register successfully'


   return render_template('UserLogin.html')
```

After the login then forward to the login the page and then customise feature for the use and the admin can notice the actives of the user by the admin login and take care of each and every user.

CHAPTER 8

TESTING

8.1 TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on "HOW" to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.

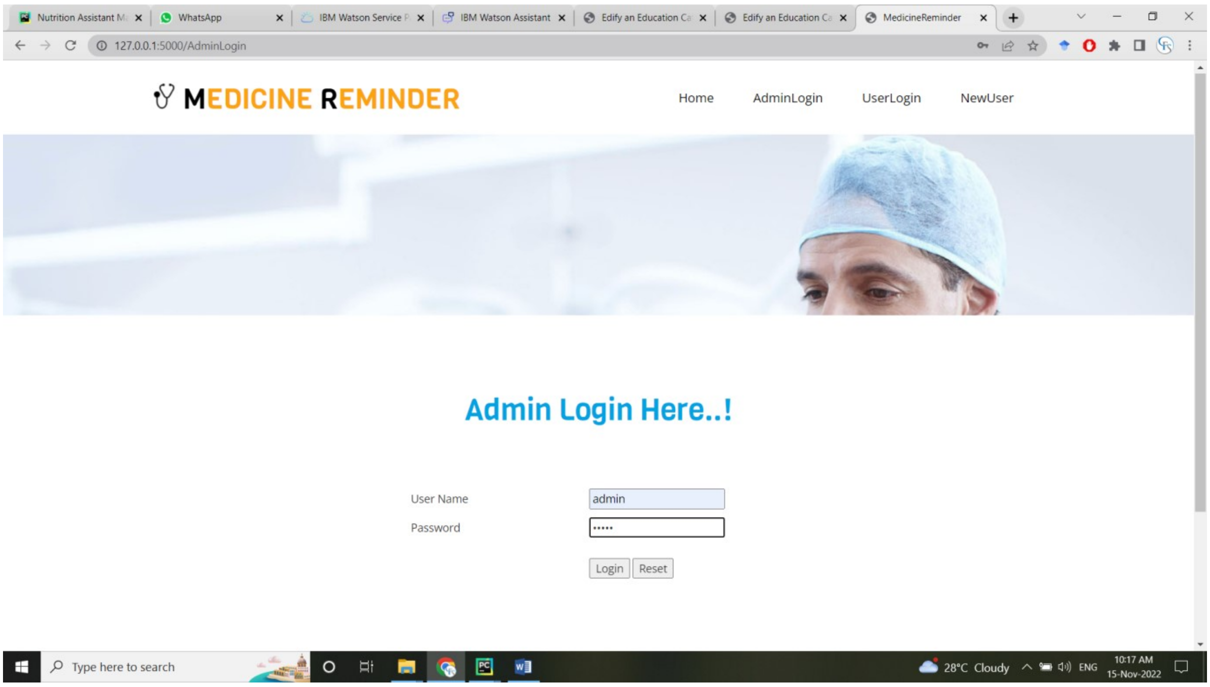
Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

S.NO	**Scenario**	**Input**	**Excepted output**	**Actual output**
1	Admin Login Form	User name and password	Login	Login success.
2	Add medicine	Medicine details	Added successfully	Medicine details stored in database.

Table 8.1 Test Cases

8.2 USER ACCEPTANCE TESTING



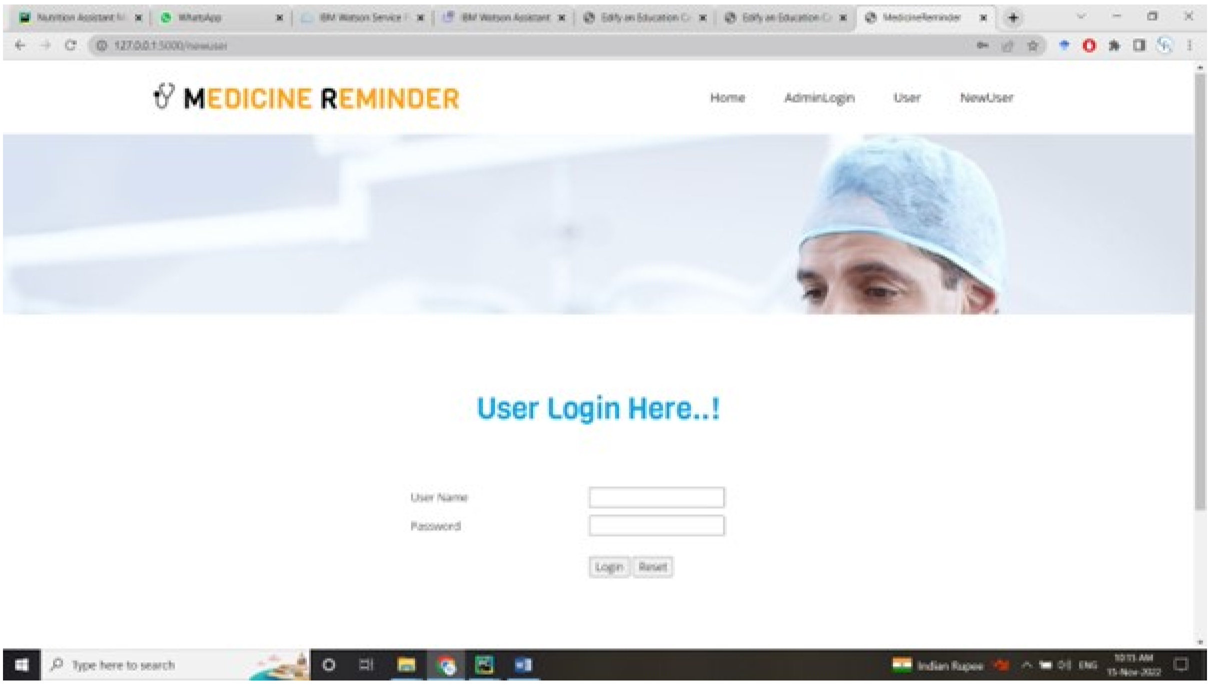
Admin Login Here..!

User Name: admin

Password: •••••

Figure 8.2.1 User Acceptance Testing

Admin login where the admin can able to login



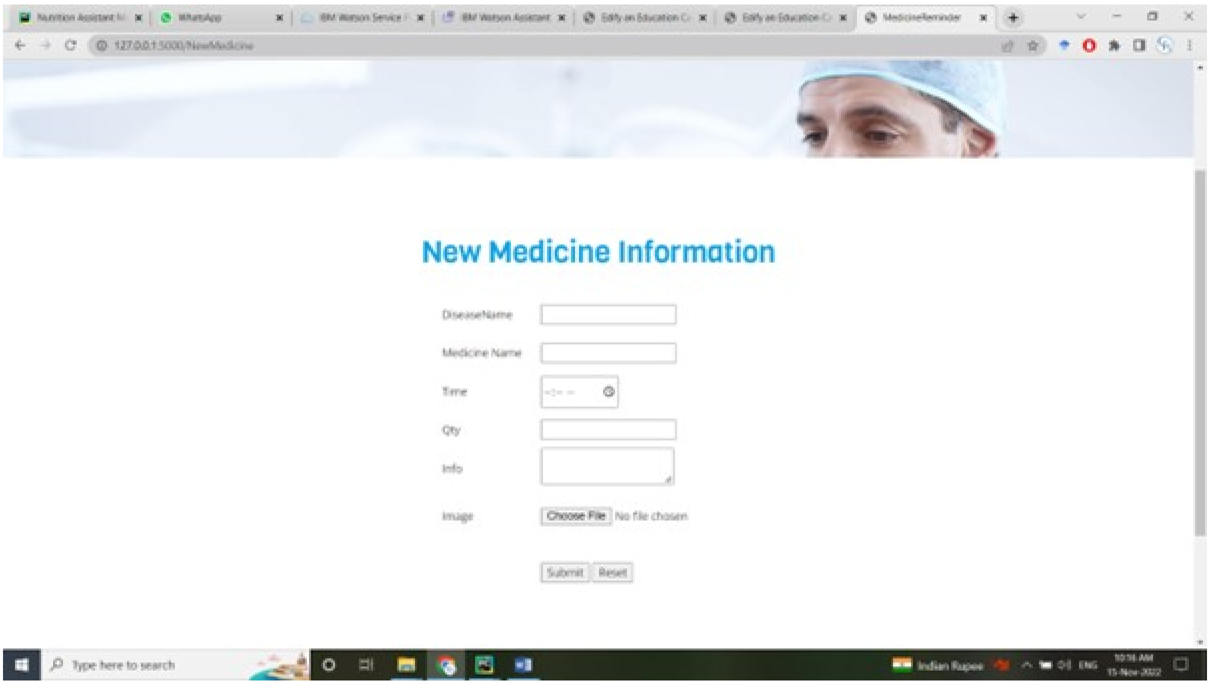
User Login Here..!

User Name

Password

Figure 8.2.2 User Login

User can login to set remainder for the Medicine



New Medicine Information

DiseaseName

Medicine Name

Time --:-- --

Qty

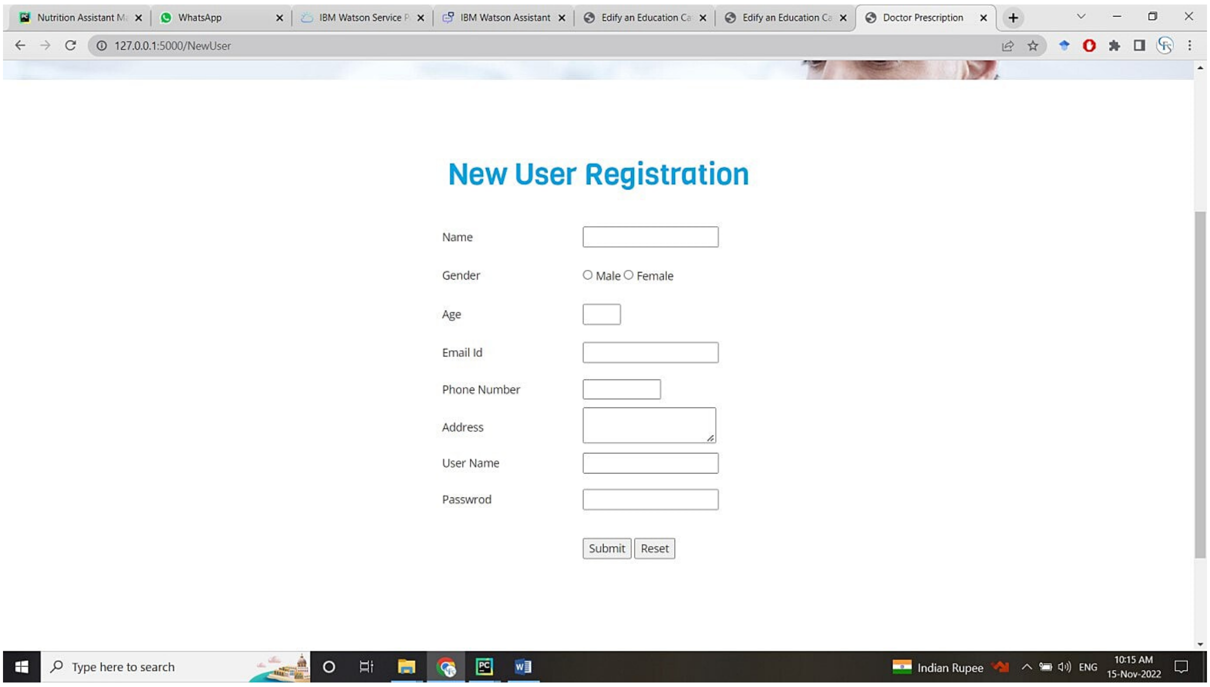
Info

Image Choose File No file chosen

Submit Reset

Figure 8.2.3 New Medicine form

The user can add the new medicine information after login



New User Registration

Name

Gender ○ Male ○ Female

Age

Email Id

Phone Number

Address

User Name

Passwrod

Submit Reset

Figure 8.2.4 User Registration

New user can register with this option

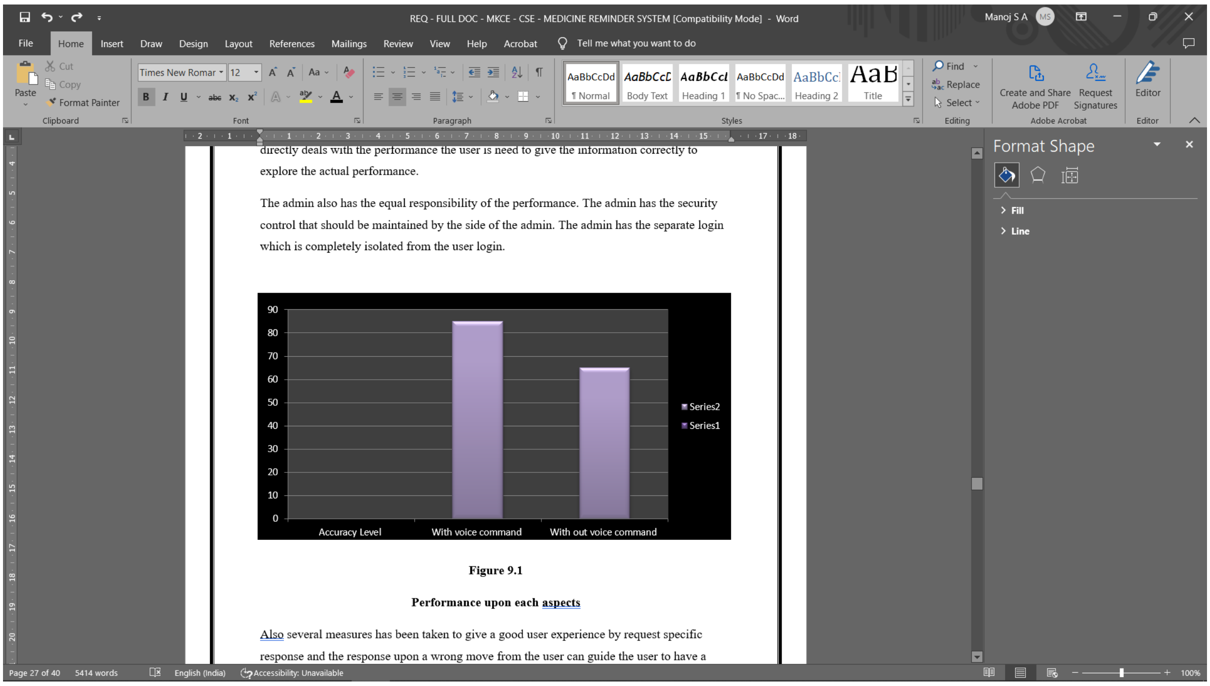
CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

The remainder is applied and work up on the detail entered by the user and then the input is directly deals with the performance the user is need to give the information correctly to explore the actual performance.

The admin also has the equal responsibility of the performance. The admin has the security control that should be maintained by the side of the admin. The admin has the separate login which is completely isolated from the user login.



directly deals with the performance the user is need to give the information correctly to explore the actual performance.

The admin also has the equal responsibility of the performance. The admin has the security control that should be maintained by the side of the admin. The admin has the separate login which is completely isolated from the user login.

Figure 9.1

Performance upon each aspects

Also several measures has been taken to give a good user experience by request specific response and the response upon a wrong move from the user can guide the user to have a

Figure 9.1 Performance Metrics

Performance upon each aspects

Also several measures has been taken to give a good user experience by request specific response and the response upon a wrong move from the user can guide the user to have a better user experience and to have a good performance.

CHAPTER 10
ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Reduce the time.
- Workload and manpower is reduced.
- Voice commands Sent to appropriate persons
- Computational complexity is less
- System also provides facility to update the medicines information.
- System provides facility to users can get the alert notification.
- This system is providing more memory for the users to maintain data.

DISADVANTAGES

- The manual system is more error prone.
- Reminders cannot be set automatically. There is a need for manual work in setting the reminder.
- A lot of time is consumed in manually setting the reminders.
- The possibility exists for the existing systems to hang down due to the manual work involved.

CHAPTER 10

CONCLUSION

This project entitled as “Personal Assistance for Seniors Who Are Self-Reliant” for Patients has been developed to satisfy all the proposed requirements. The process of recording details about medicine information is more simple and easy. The system reduces the possibility of errors to a great extent and maintains the data in an efficient manner. Through this voice assistant, we have automated various services using a single Line command. We aim to make this project a complete server assistant and make it smart enough to act as a replacement for a general server administration. This application is simplify the task of admin and patients. It helps to optimize the work of patient. Installation of the application in the system is quite simple and more useful to patients. The coding is done in a simplified and easy to understandable manner so that other team trying to enhance the project can do so without facing much difficulty. The documentation will also assist in the process as it has also been carried out in a simplified and concise way.

CHAPTER 12

FUTURE SCOPE

In future we can develop this project in IOT with android application like Secure doctor and patients communication technology. Bridging the gap between the physical and digital world has become an important aspect of the progressive world. The internet in hand with the emerging technologies play a vital role in this process. The internet is the common force that connects all technological devices in the current day and age. Any physical device that can be connected to the internet and be used to share and collect data can be termed under the ‘Internet of Things (IoT)’. The future of IoT lies in the possibility to transform something as small as a chip or even an entire city into a working IoT device.

CHAPTER 13

APPENDIX

SOURCE CODE

```
from flask import Flask, render_template, flash, request, session
from flask import render_template, redirect, url_for, request
import datetime

import ibm_db
import pandas
import ibm_db_dbi
from sqlalchemy import create_engine

engine = create_engine('sqlite://',
echo = False)

dsn_hostname = "b0aebb68-94fa-46ec-a1fc-
1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud"
dsn_uid = "zht77907"
dsn_pwd = "12CV4yAJK0mxMABC"

dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "BLUDB"
dsn_port = "31249"
dsn_protocol = "TCPIP"
dsn_security = "SSL"

dsn = (
"DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port,
dsn_protocol, dsn_uid, dsn_pwd,dsn_security)

try:
    conn = ibm_db.connect(dsn, "", "")
print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on host: ",
dsn_hostname)

except:
print ("Unable to connect: ", ibm_db.conn_errormsg() )
app = Flask(__name__)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'



@app.route("/")
def homepage():
return render_template('index.html')
@app.route("/Home")
def Home():
return render_template('index.html')
@app.route("/AdminLogin")
def AdminLogin():
return render_template('AdminLogin.html')



@app.route("/UserLogin")
def UserLogin():
```

```
return render_template('UserLogin.html')
@app.route("/NewUser")
def NewUser():
return render_template('NewUser.html')

@app.route("/NewMedicine")
def NewMedicine():
return render_template('NewMedicines.html')

@app.route("/adminlogin", methods=['GET', 'POST'])
def adminlogin():
    error = None
    if request.method == 'POST':
if request.form['uname'] == 'admin' and request.form['password'] == 'admin':

           conn = ibm_db.connect(dsn, "", "")
           pd_conn = ibm_db_dbi.Connection(conn)
           selectQuery = "SELECT * FROM regtb  "
dataframe = pandas.read_sql(selectQuery, pd_conn)
           dataframe.to_sql('booktb1', con=engine, if_exists='append')
           data = engine.execute("SELECT * FROM booktb1").fetchall()

return render_template('AdminHome.html', data=data)

else:
return render_template('index.html', error=error)

@app.route("/newmedi", methods=['GET', 'POST'])
def newmedi():
if request.method == 'POST':
        uname = session['uname']
        dname = request.form['dname']
        mname = request.form['mname']
        tim = request.form['tim']
        qty = request.form['qty']

        info = request.form['info']
        file = request.files['file']
        file.save("static/upload/" + file.filename)



        conn = ibm_db.connect(dsn, "", "")

        insertQuery =  "INSERT INTO meditb VALUES ('"+ uname +"','" + dname + "','" +
mname + "','" + tim + "','" + qty + "','" + info + "','" + file.filename + "')"
insert_table = ibm_db.exec_immediate(conn, insertQuery)



        alert = 'Medicine info saved successfully'
return render_template('goback.html', data=alert)

@app.route("/MedicineInfo")
def MedicineInfo():




    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM meditb  "
dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('booktb1', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM booktb1").fetchall()
```

```
return render_template('MedicinesInfo.html', data=data)


@app.route("/newuser", methods=['GET', 'POST'])
def newuser():
if request.method == 'POST':

        name1 = request.form['name']
        gender1 = request.form['gender']
        Age = request.form['age']
        email = request.form['email']
        pnumber = request.form['phone']
        address = request.form['address']

        uname = request.form['uname']
        password = request.form['psw']
        conn = ibm_db.connect(dsn, "", "")

        insertQuery = "INSERT INTO regtb VALUES ('" + name1 + "','" + gender1 + "','" + Age
+ "','" + email + "','" + pnumber + "','" + address + "','" + uname + "','" + password + "')"
insert_table = ibm_db.exec_immediate(conn, insertQuery)
# return 'file register successfully'


return render_template('UserLogin.html')


@app.route("/userlogin", methods=['GET', 'POST'])
def userlogin():
error = None
    if request.method == 'POST':
```

```
    username = request.form['uname']
    password = request.form['password']
    session['uname'] = request.form['uname']



    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * from regtb where UserName='" + username + "' and
password='" + password + "'"
dataframe = pandas.read_sql(selectQuery, pd_conn)
if dataframe.empty:
      data1 = 'Username or Password is wrong'
return render_template('goback.html', data=data1)
else:
print("Login")
      selectQuery = "SELECT * from regtb where UserName='" + username + "' and
password='" + password + "'"
dataframe = pandas.read_sql(selectQuery, pd_conn)

      dataframe.to_sql('Employee_Data',
con=engine,
if_exists='append')

# run a sql query
print(engine.execute("SELECT * FROM Employee_Data").fetchall())

return render_template('UserHome.html', data=engine.execute("SELECT * FROM
Employee_Data").fetchall())

@app.route("/UserHome")
def UserHome():
    username = session['uname']

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM regtb where username='" + username + "'"
dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('booktb1', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM booktb1").fetchall()

return render_template('UserHome.html', data=data)

@app.route("/Alert")
def Alert():

from datetime import datetime
import pandas as pd
# Get current time in local timezone
current_time = datetime.now()
print('Current timestamp: ', current_time.strftime('%H:%M:%S'))

    dd1 = current_time.strftime('%H:%M:%S')

    n = 15
# Add 2 minutes to datetime object containing current time
future_time = current_time + pd.DateOffset(minutes=n)
print('Future Time (2 minutes from now ): ', future_time)
# Convert datetime object to string in specific format
future_time_str = future_time.strftime('%H:%M:%S')
print('Future Time as string object: ', future_time_str)

    dd2 = future_time_str
```

```
username = session['uname']



    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM meditb  where Time between '"+ dd1 +"' and '"+ dd2+"'"
dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('booktb1', con=engine, if_exists='append')

    data = engine.execute("SELECT * FROM booktb1  ").fetchall()

for item1 in data:
uname = item1[1]
        uname =item1[1]
dname = item1[2]
        mname = item1[3]
tim =item1[4]
        qty = item1[5]

from gtts import gTTS

# This module is imported so that we can
        # play the converted audio
import os

# The text that you want to convert to audio
mytext = 'Medicine Name ' + mname + ' Quantity'+ qty

# Language in which you want to convert
language = 'en'
```

```
# Passing the text and language to the engine,
        # here we have marked slow=False. Which tells
        # the module that the converted audio should
        # have a high speed
myobj = gTTS(text=mytext, lang=language, slow=False)

# Saving the converted audio in a mp3 file named
        # welcome
myobj.save("welcome.mp3")
        os.system("welcome.mp3")

        selectQuery = "SELECT  *  FROM  regtb where Username='" + uname + "'"
dataframe = pandas.read_sql(selectQuery, pd_conn)

        dataframe.to_sql('regtb', con=engine, if_exists='append')
        data2 = engine.execute("SELECT * FROM regtb").fetchall()
for item in data2:
email = item[4]

return render_template('Alert.html', data=data)

@app.route("/Remove", methods=['GET'])
def Remove():


    unmae = request.args.get('id')
    mname = request.args.get('mname')
    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    insertQuery = "Delete from meditb  where  username='"+ unmae +"' and
MediName='"+mname +"'"
```

```
insert_table = ibm_db.exec_immediate(conn, insertQuery)



    selectQuery = "SELECT * from meditb "
dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data',
con=engine,
if_exists='append')

# run a sql query
print(engine.execute("SELECT * FROM Employee_Data").fetchall())

return render_template('MedicinesInfo.html', data=engine.execute("SELECT * FROM Employee_Data").fetchall())

if __name__ == '__main__':
    app.run(debug=True, use_reloader=True)
```

GITHUB & PROJECT DEMO LINK

Github :

https://github.com/IBM-EPBL/IBM-Project-21459-1659780556

Demo Vedio :

https://drive.google.com/drive/folders/1jclF9wCV3LXHaXGmm2X6HM5UVDbJRd2T?usp=sharing