IOT BASED PERSONAL ASSISSTANCE FOR SENIOR

WHO ARE SELF-RELIANT

## A PRORJECT REPORT

***Submitted by***

|  |  |
| --- | --- |
| **PREETHI J** | **(721719106045)** |
| **RASIHA R** | **(721719106048)** |
| **SADHURYA R** | **(721719106049)** |
| **SHOBA P** | **(721719106058)** |

***in partial fulfillment for the award of the degree of***

# BACHELOR OF ENGINEERING

***in***

**ELECTRONICS AND COMMUNICATION ENGINEERING**



# P. A. COLLEGE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Pollachi, Coimbatore Dt - 642 002

DECEMBER 2022

# P. A. COLLEGE OF ENGINEERING AND TECHNOLOGY BONAFIDE CERTIFICATE

Certified that this project report **“IOT BASED PERSONAL ASSISSTANCE FOR SENIOR WHO ARE SELF-RELIANT”** is the bonafide work of **PREETHI J (7217106045), RASIHA R (721719106048), SADHURYS R (721719106049), SHOBA P (721719106058)** who carried out the project report under my supervision.

## Signature of the HoD Signature of the Mentor

Dr. L. Murali, M.E., Ph.D., Mr. M. Madhumalini, M.E.,

Associate Professor and Head, Associate Professor,

Department of Electronics and Department of Electronics and

Communication Engineering. Communication Engineering.

Submitted for the Viva-voce examination held on ……………….

……………………….. ………………………….

## Internal Examiner External Examiner

**ACKNOWLEDGEMENT**

With genuine humility, we are obediently thankful to God almighty praise and glory is to him, for all his uncountable bounties and guidance.

We express our profound gratitude to our Respected Chairman **Dr. P. Appukutty, M.E., F.I.E., F.I.V.,** for giving this opportunity to pursue this course.

At this pleasing moment of having successfully completed the project work, we wish to acknowledge our sincere gratitude and heartful thanks to our respected Principal **Dr. T. Manigandan, M.E., Ph.D.,** for having given us the adequate support and opportunity for completing this project work successfully.

We express our deep sense of sincere thanks to **Dr. L. Murali, M.E., Ph.D.,** Head of the Department, who has been a spark for enlightening our knowledge.

Our profound gratitude goes to our project mentor **Mrs. M. Madhumalini**., Associate Professor, **Mr. R. Vishnu Varadhan**., Assistant Professor, Department of ECE, for their valuable guidance and generous help.

We express our heartiest and sincere thanks to our project evaluators **Dr. K. S. Neelu Kumari**,Associate Professor, **Mr. N. Sarveswaran.,** Assistant Professor, Department of ECE, for their valuable suggestion and guidance throughout the entire course of the work. We extend our deepest thank to all staff members of Department of ECE for their help and assistance during the project.

Our humble gratitude and heartiest thanks goes to our family member and friends for their encouragement and support throughout the course of this project.

ABSTRACT

In this report a remote health monitoring system focusing on the general wellbeing of elderly people. In this work, a flexible and extensive digital platform for Smart Homes is presented, exploiting the most advanced technologies of the Internet of Things, such as Radio Frequency Identification, wearable electronics, Wireless Sensor Networks, and Artificial Intelligence. The wearable gadget is run by a microprocessor and it will gather information using sensors. By using GSM and GPS modules the patient's relatives, as well as an appointed doctor, can monitor real-time activities, see the condition of the patient remotely, receive emergency alerts and track location through an android app. The core purpose of the paper is to build a low budget monitoring system for elderly health care, developed focusing on user compatibility and reliability.

## TABLE OF CONTENTS

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER NO.** | **TITLE**  **ABSTRACT** | **PAGE NO.**   |  | | --- | |  | |
| **1.** | **INTRODUCTION** |  |
|  | 1.1 Project Overview | 1 |
|  | 1.2 Purpose | 1 |
| **2.** | **LITERATURE SURVEY** |  |
|  | 2.1 Existing problem | 2 |
|  | 2.2 References | 2 |
|  | 2.3 Problem Statement Definition | 3 |
| **3.** | **IDEATION & PROPOSED SOLUTION** |  |
|  | 3.1 Empathy Map Canvas | 4 |
|  | 3.2 Ideation & Brainstorming | 5 |
|  | 3.3 Proposed Solution | 6 |
|  | 3.4 Problem Solution fit | 7 |
| **4.** | **REQUIREMENT ANALYSIS** |  |
|  | 4.1 Functional requirement | 8 |
|  | 4.2 Non-Functional requirements | 9 |
| **5.** | **PROJECT DESIGN** |  |
|  | 5.1 Data Flow Diagrams | 10 |
|  | 5.2 Solution & Technical Architecture | 11 |
|  | 5.3 User Stories | 12 |
| **6.** | **PROJECT PLANNING & SCHEDULING** |  |
|  | 6.1 Sprint Planning & Estimation | 13 |
|  | 6.2 Sprint Delivery Schedule | 14 |
|  | 6.3 Reports from JIRA | 16 |
| **7.** | **CODING & SOLUTIONING** |  |
|  | 7.1 Feature 1 | 17 |
|  | 7.2 Feature 2 | 19 |
|  | 7.3 Database Schema | 21 |
| **8.** | **TESTING** |  |
|  | 8.1 Test Cases | 23 |
|  | 8.2 User Acceptance Testing | 25 |
| **9.** | **RESULTS** |  |
|  | 9.1 Performance Metrics | 27 |
| **10.** | **CONCLUSION** | 29 |
| **11.** | **FUTURE SCOPE** | 29 |
| **12.** | **APPENDIX** |  |
|  | A1. Source Code | 31 |
|  | A2. GitHub & Project Demo Link | 31 |

## 

**INTRODUCTION**

* 1. **PROJECT OVERVIEW**
     + An app is built for the user (caretaker) which enables him to set the desired time and medicine. These details will be stored in the IBM Cloudant DB.
     + If the medicine time arrives the web application will send the medicine name to the IOT Device through the IBM IOT platform.
     + The device will receive the medicine name and notify the user with voice commands.
  2. **PURPOSE**
     + Sometimes elderly people forget to take their medicine at the correct time.
     + They also forget which medicine He / She should take at that particular time.
     + And it is difficult for doctors/caretakers to monitor the patients around the clock. To avoid this problem, this medicine reminder system is developed.

1. **LITERATURE SURVEY**
   1. **EXISTING PROBLEM**

In this study, review provided an overview of the areas that older adults living at home with chronic conditions in the UK might need care and support with. It was clear from the evidence reviewed that older adults living with chronic conditions are faced with some challenges in their social lives, psychological health, and activities related to self-care, domestic lives and mobility. It was also clear that despite these challenges older adults valued independence and demonstrated a desire to cope with their illness. However, lack of professional support and barriers associated with some services interfered with these efforts, highlighting the fact that many services and care delivery models are still not based on the needs of older adult. Thus, these findings reinforced the importance of tailoring interventions and support services that take into consideration the needs of older adults.

* 1. **REFERENCES**

**[1]** [**Shawn Radcliffe**](https://www.healthline.com/authors/shawn-radcliffe)**(2014)** - How Seniors Can Live Independently forLonger**:** As the number of American seniors increases, more are turning to medical devices and assistance from caregivers to help them live independently, a movement called aging in place.

**URL:**

<https://www.healthline.com/health-news/aging-seniors-can-stay-at-home-longer-but-need-help-121213>

# [2] [Dennis Paulino](https://www.researchgate.net/profile/Dennis-Paulino) , [Hugo Paredes](https://www.researchgate.net/profile/Hugo-Paredes) (2017) - Using Intelligent Personal Assistants to Strengthen the Elderlies’ Social Bonds.

# URL:

# <https://www.researchgate.net/publication/317237193_Using_Intelligent_Personal_Assistants_to_Strengthen_the_Elderlies'_Social_Bonds>.

# [3][Angayarkanni Veeraputhiran](https://www.researchgate.net/profile/Angayarkanni-Veeraputhiran-2) , [Radha Sankararajan](https://www.researchgate.net/profile/Radha-Sankararajan)- Feature based fall detection system for elders using compressed sensing in WVSN

# URL:

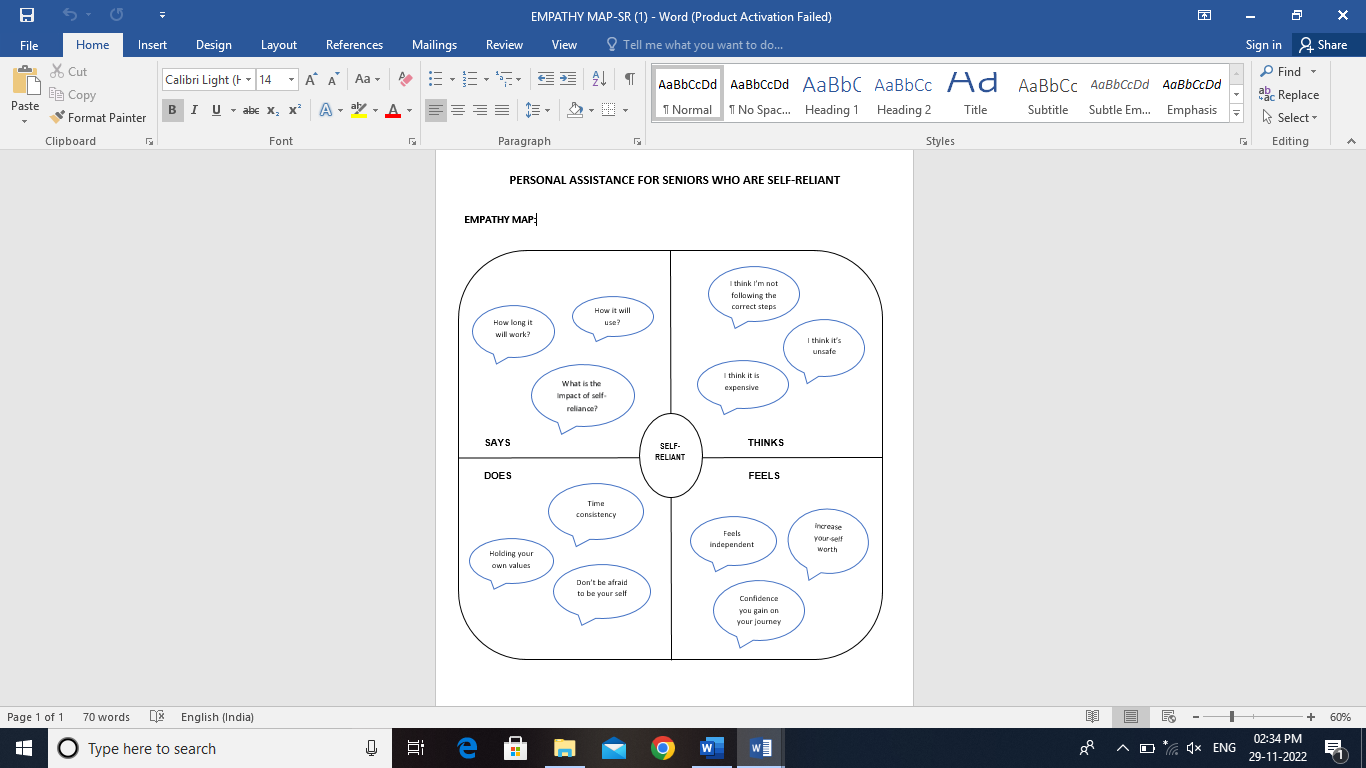
# <https://www.researchgate.net/publication/318466757_Feature_based_fall_detection_system_for_elders_using_compressed_sensing_in_WVSN>

* 1. **PROBLEM STATEMENT DEFINITION**

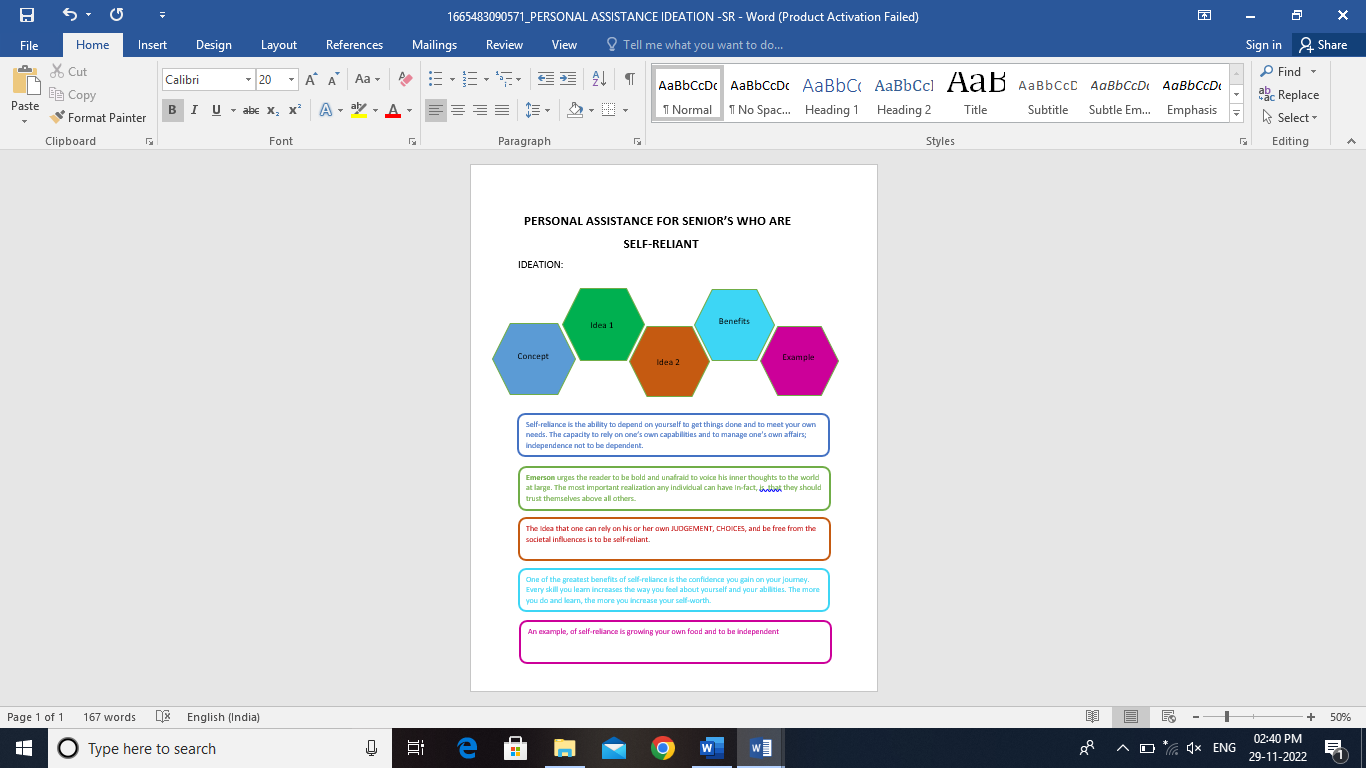
Some people find it difficult to learn new apps in this ever-expanding digital environment, and people nowadays tend to forget things more easily, such as taking their prescriptions. People need a way to remember to take their prescriptions without having to learn how to use sophisticated programs.

Elderly people let slip the medications at the correct time and the existing solutions for this problem is setting reminders or using pill boxes, calendars, Personal Assistance. Though the solutions give reminders, the voice commands or assistance given by this system is more efficient.

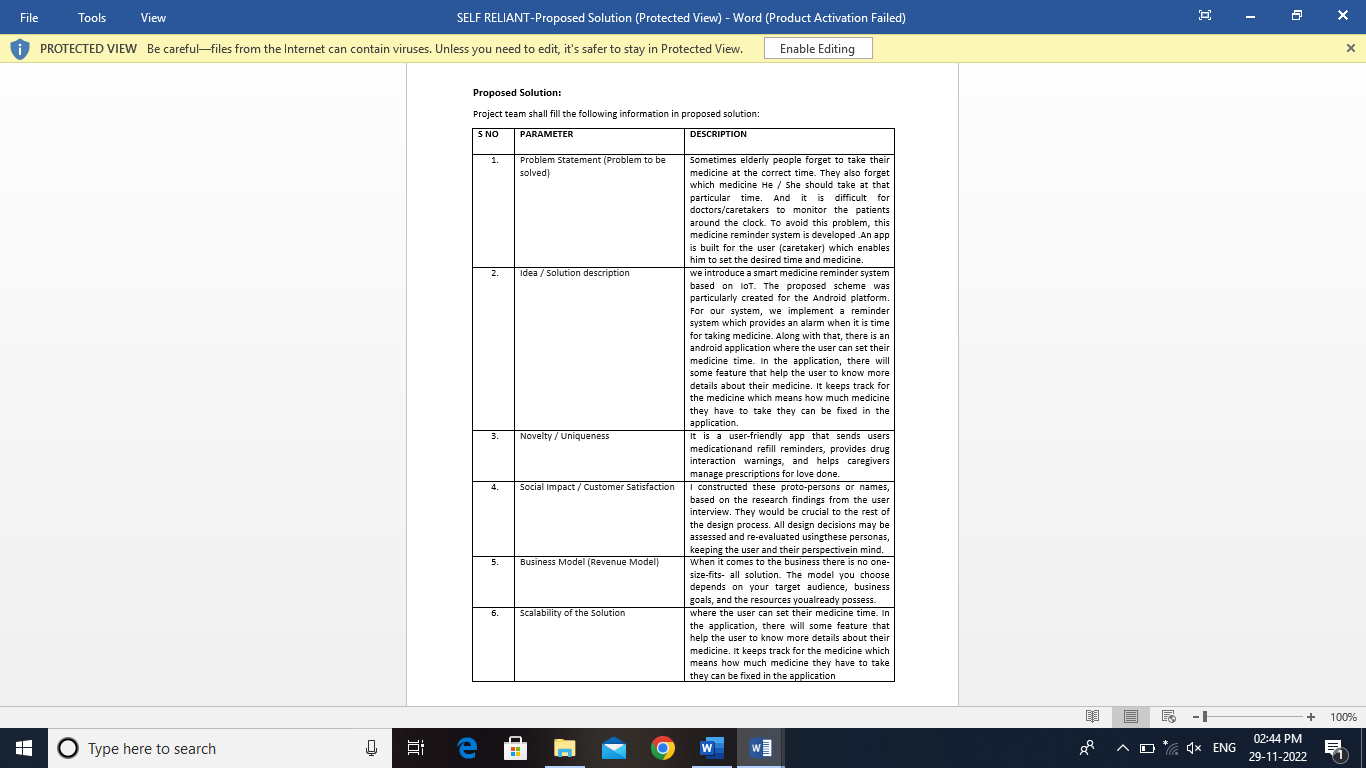
1. **IDEATION & PROPOSED SOLUTION**
   1. **EMPATHY MAP CANVAS**



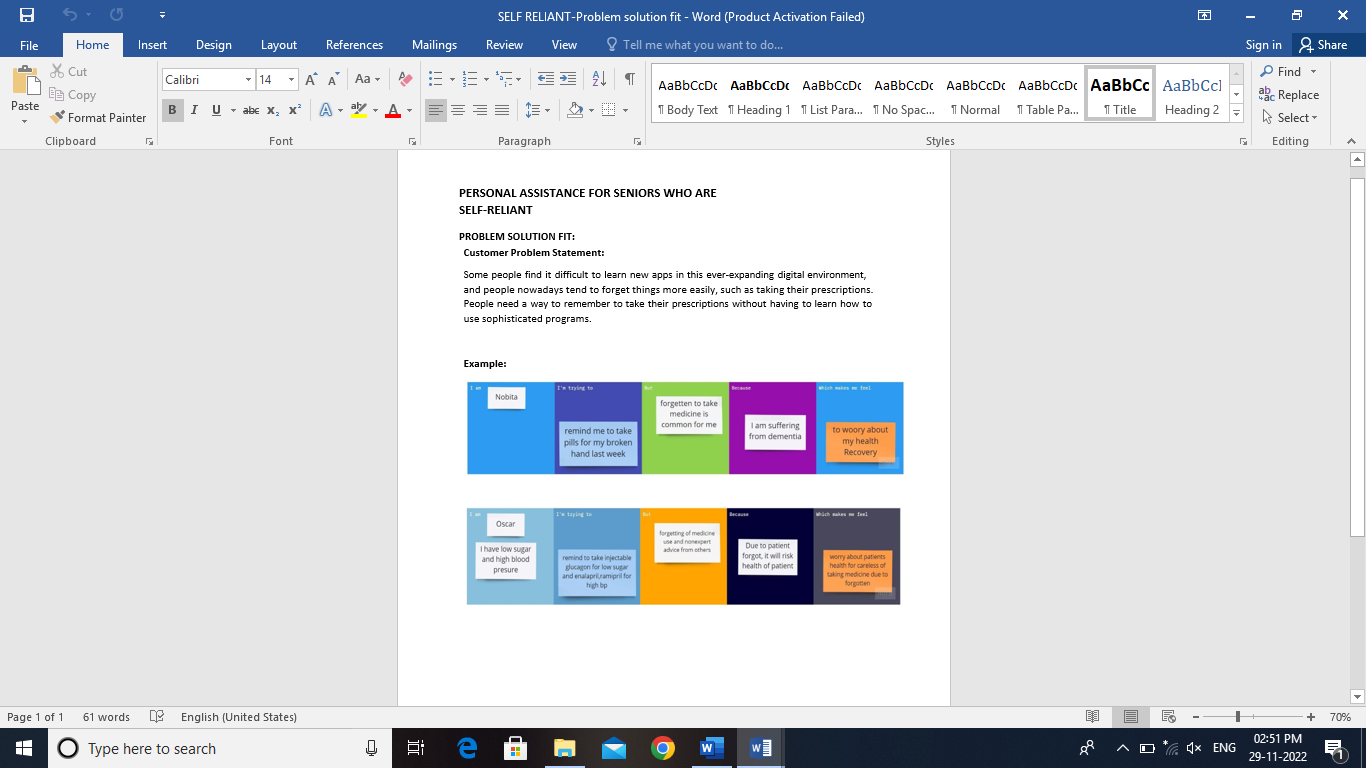
* 1. **IDEATION AND BRAINSTORMING**



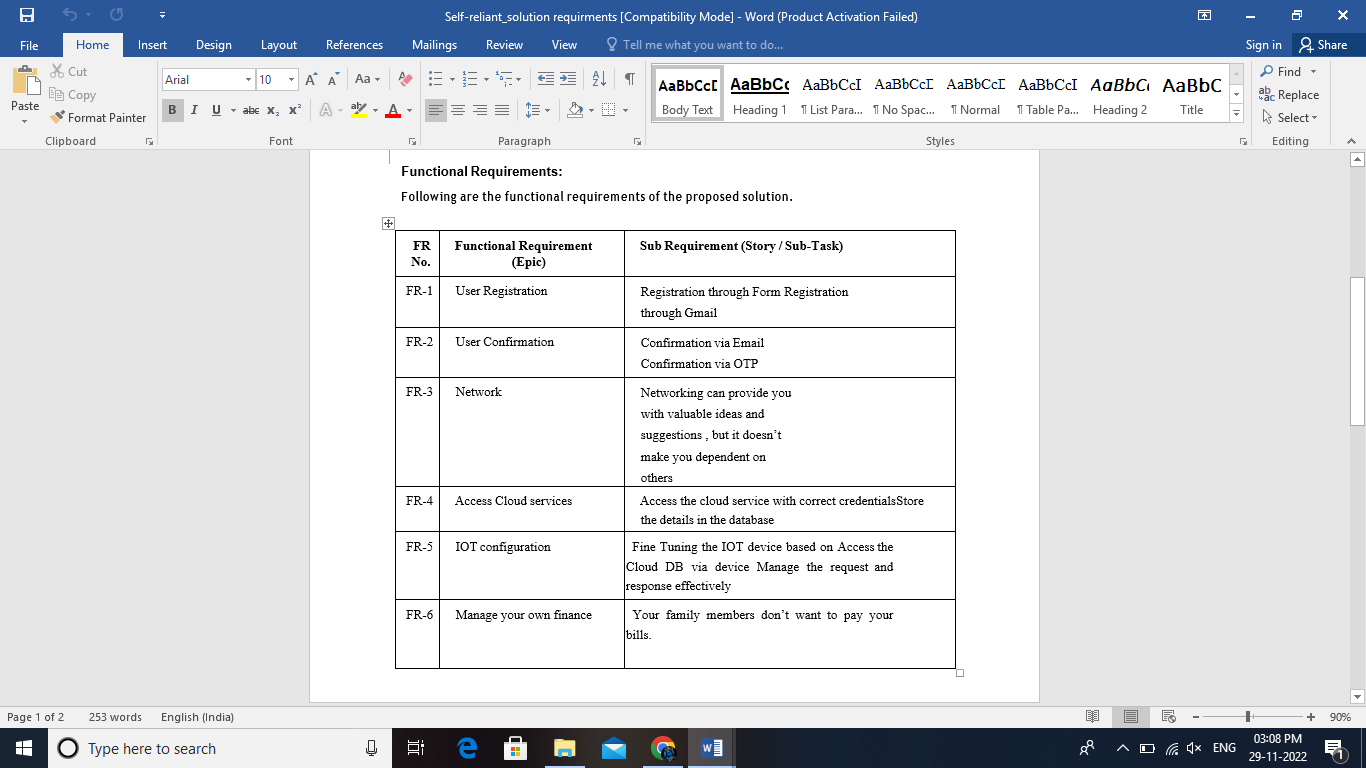
* 1. **PROPOSED SOLUTION**



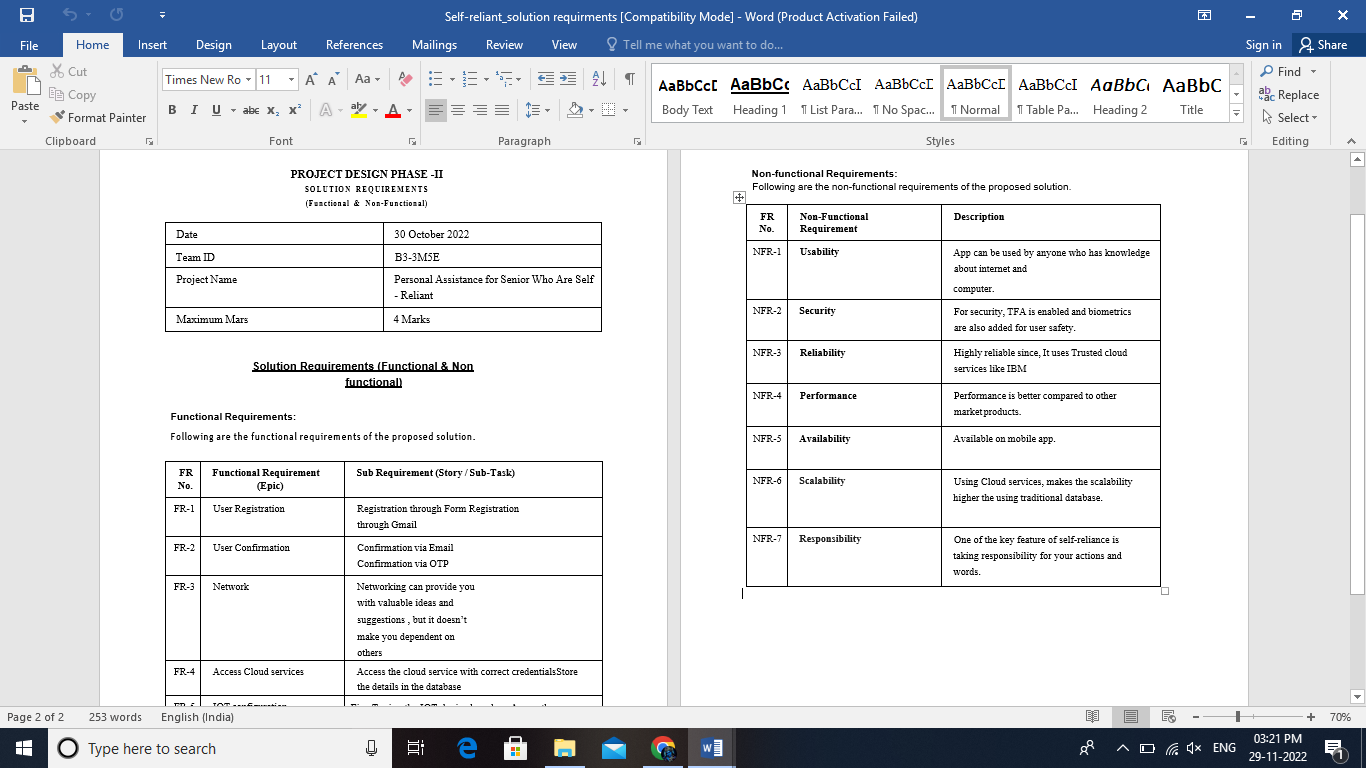
* 1. **PROBLEM SOLUTION FIT**

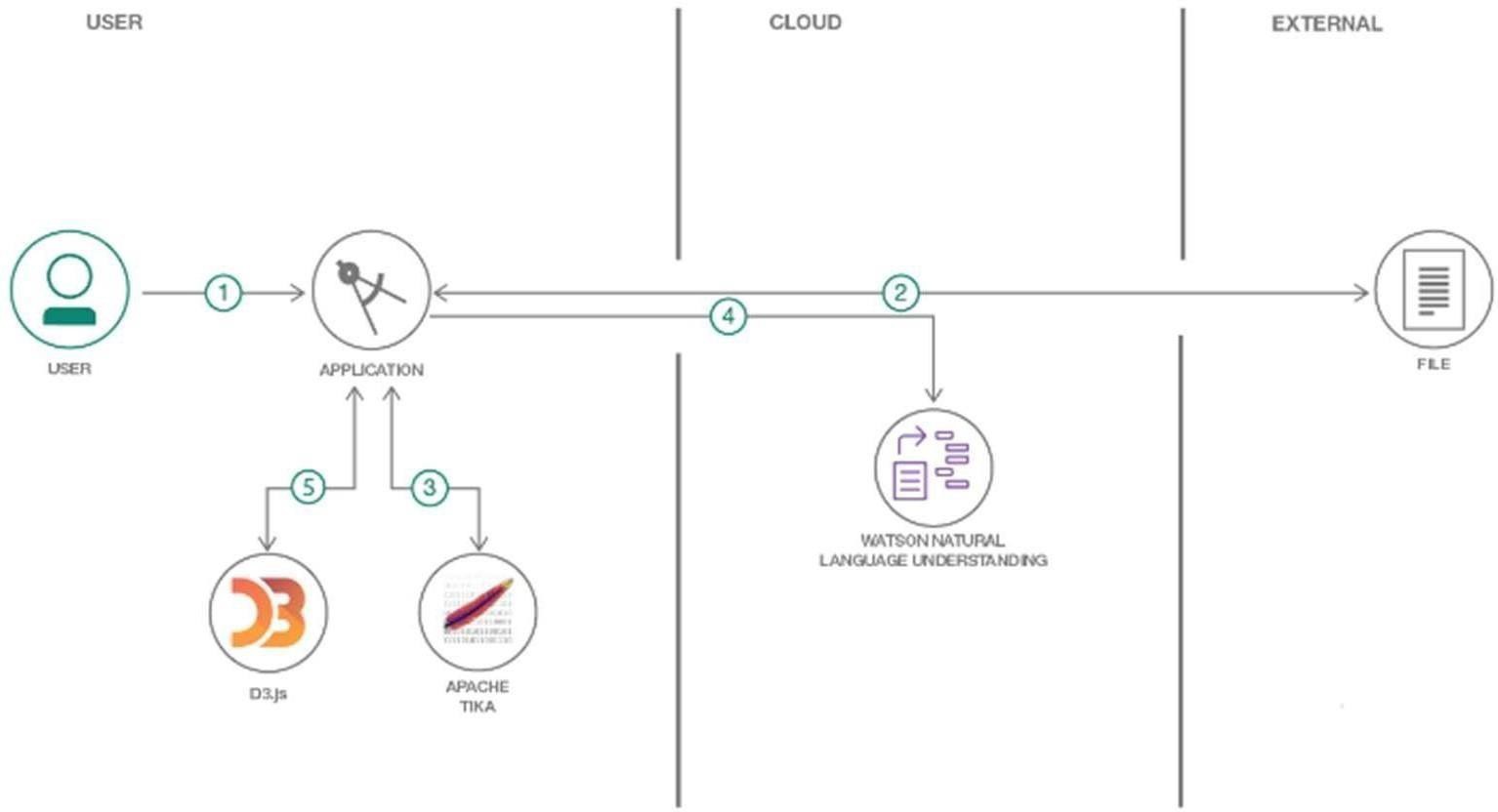


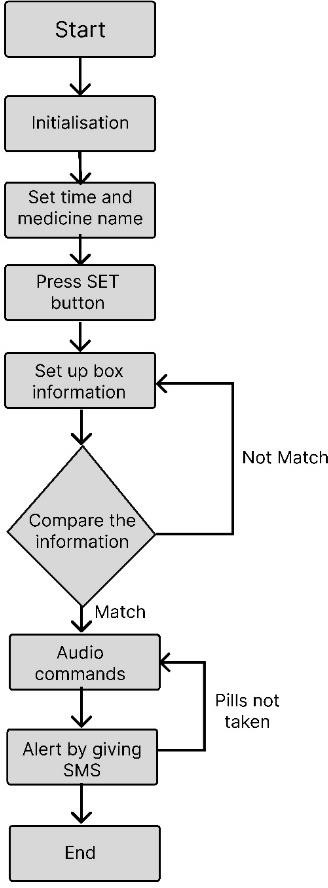
1. **REQUIREMENT ANALYSIS**
   1. **FUNCTIONAL REQUIREMENTS**



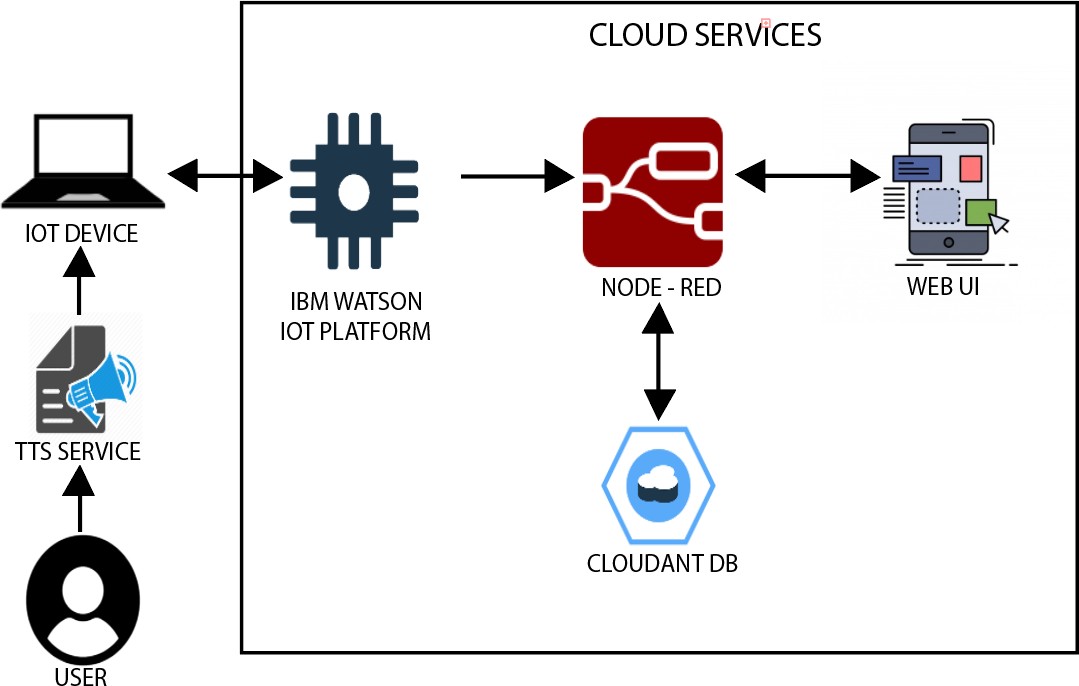
* 1. **NON-FUNCTIONAL REQUIREMENTS**

**v**

1. **PROJECT DESIGN**
   1. **DATA FLOW DIAGRAM**



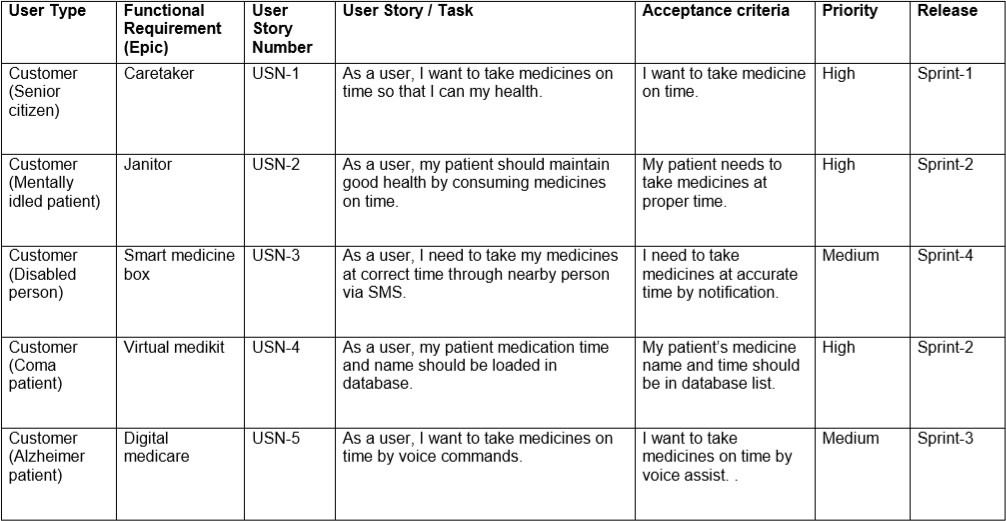
* 1. **SOLUTION & TECHNICAL ARCHITECTURE**



**COMPONENTS AND TECHNOLOGIES**

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Component | Description | Technology |
| 1.  2. | **User Interface** | **Mobile App** | **HTML, CSS,**  **JavaScript**  **Python** |
| **Application Logic-1** | **Mobile App to enter the Medicine**  **Details weekly** |
| 3. | **Application Logic-2** | **Gets the medication data from database** | **IBM Watson IoT API**  **Call data** |
| 4. | **Application Logic-3** | **Converts the text to speech to pronunciation for the user** | **IBM Watson**  **Assistant** |
| 5. | **Database** | **Medication time and tablets name on daily and** | **MySQL** |
| 6. | **Cloud Database** | **Call the data IBM Cloud is used and user login credentials** | **IBM DB2, IBM**  **Cloud** |
| 7. | **File Storage** | **App code and IOT credentials are stored and API keys** | **IBM Block Storage** |
| 8. | **External API-1** | **To get the medicine box status**  **Open or not** | **IBM box status API** |

* 1. **USER STORIES**



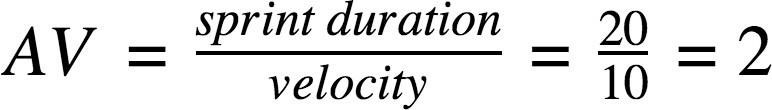
**6 PROJECT PLANNING & SCHEDULING**

* 1. **SPRINT PLANNING & ESTIMATION**

|  |  |  |  |
| --- | --- | --- | --- |
| SNO | TITLE | DESCRIPTION | DATE |
| 1 | Literature Survey & Information Gathering | Literature survey on the selected project & gathering information by referring to technical papers, research publications etc. | 3/10/2022 |
| 2 | Prepare Empath Map | Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements | 4/10/2022 |
| 3 | Ideation | List them by organizing the brainstorming session and prioritize the top 3 ideas based on feasibility & importance. | 6/10/2022 |
| 4 | Proposed Solution | Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc. | 9/10/2022 |
| 5 | Problem Solution Fit | Prepare problem - solution fit document | 13/10/2022 |

## 6.2. SPRINT DELIVERY SCHEDULE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-1 | Registration | USN-1 | As a user, I can register for the application by  entering my email, and password, and confirming my password. | 3 | High | Preethi. J |
| Sprint-1 | Confirmation Email | USN-2 | As a user, I will receive a confirmation email once I have registered for the application | 4 | High | Rasiha.R |
| Sprint-1 | Authentication | USN-3 | As a user, I can register for the application through Gmail and mobile app. | 4 | Medium | Shoba.P |
| Sprint-1 | Login | USN-4 | As a user, I can log into the application by entering email & password | 3 | High | Sadhurya.R |
| Sprint-1 | Dashboard | USN-5 | As a user, I need to be able to view the functions that I can perform | 4 | High | Rasiha.R  Shoba.P |
| Sprint-2 | Notification | USN-1 | As a user, I should be able to notify my parent and guardian in emergency situations | 1  0 | High | Preethi.J  Sadhurya.R |
| Sprint-2 | Store data | USN-2 | As a user, I need to continuously store my location data into the database. | 1  0 | Medium | Preethi.J |
| Sprint-3 | Communication | USN-3,1 | I should be able to communicate with user | 6 | Low | Rasiha.R |
| Sprint-3 | IOT Device – Watson communication | USN-1,4 | The data from IOT device should reach IBM Cloud | 7 | Medium | Preethi.J |
| Sprint-3 | Node RED - CLOUDANT DB communication | USN-5,2 | The data stored in IBM Cloud should be properly integrated with CLOUDANT DB | 7 | High | Rasiha.R |
| Sprint-4 | User – Web UI interface | USN-1,4 | The Web UI should get inputs from the user | 6 | High | Sadhurya.R  Shoba.P |
| Sprint-4 | Alarm | USN-2,3,5 | The Alarm of the remainder should be done based on the medication time | 7 | High | Rasiha.R  Preethi.J |



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

**Velocity:**

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)



**25**



300



**20**



250

200



**15**



150



**10**



100



50



**Start**

**Week**

**1**



**Week**

**2**

Days



**Week**

**3**



**Week**

**4**



Remaining

Effort

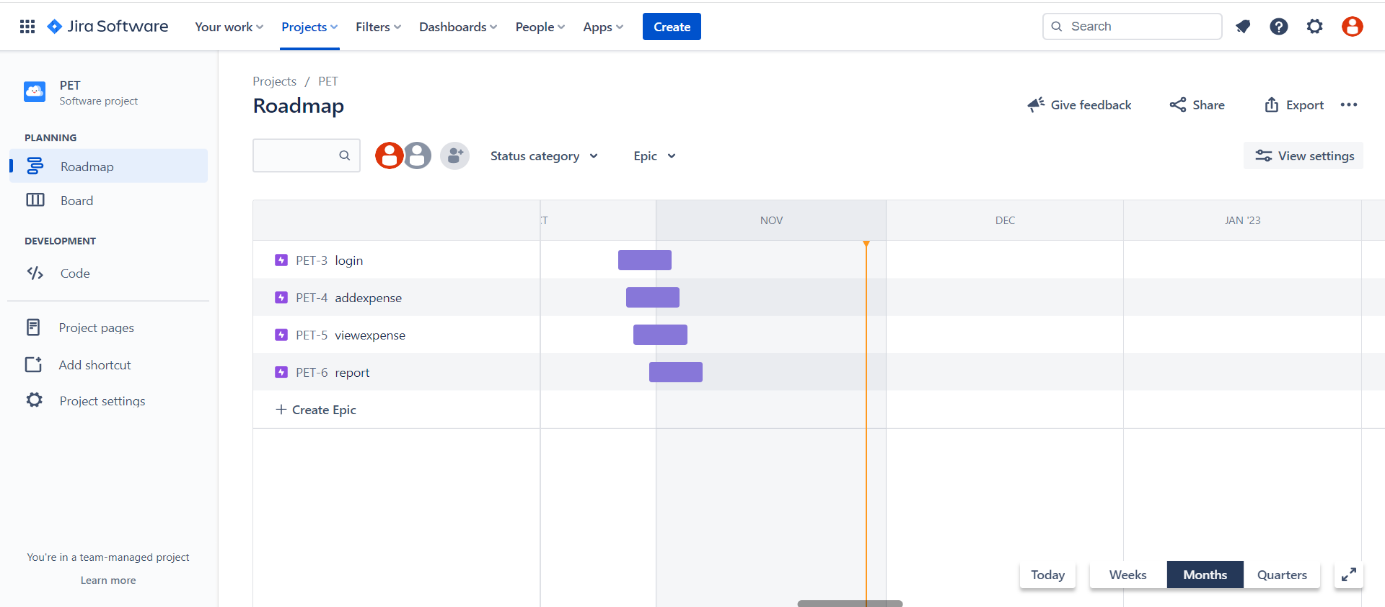


Ideal

Burndown

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

**6.3. EPORTS FROM JIRA**



**7. CODING & SOLUTIONING**

**7.1 Feature 1**

- **Login.html**

<!DOCTYPE html>

<html lang="en" dir="ltr">

   <head>

      <meta charset="utf-8">

      <title>Personal Expense Tracker</title>

      <link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

   </head>

   <body>

      <div class="wrapper">

         <div class="title-text">

            <div class="title login">

               Login Form

            </div>

            <div class="title signup">

               Signup Form

            </div>

         </div>

         <div class="form-container">

            <div class="slide-controls">

               <input type="radio" name="slide" id="login">

               <input type="radio" name="slide" id="signup">

               <label for="login" class="slide login">Login</label>

               <label for="signup" class="slide signup">Signup</label>

               <div class="slider-tab"></div>

            </div>

            <div class="form-inner">

               <form action="#" class="login">

                  <div class="field">

                     <input type="text" placeholder="Email Address" name="email" required>

                  </div>

                  <div class="field">

                     <input type="password" placeholder="Password" name = "password" required>

<div class="pass-link">

                     <a href="#">Forgot password?</a>

                  </div>

                  <div class="field btn">

                     <div class="btn-layer"></div>

                     <input type="submit" value="Login">

                  </div>

                  <div class="signup-link">

                     Not a member? <a href="">Signup now</a>

                  </div>

               </form>

               <form action="http://localhost:8080/IP/A" method="post"class="signup">

                  <div class="field">

                     <input type="text" placeholder="Email Address" name = "email" required>

                  </div>

                  <div class="field">

                     <input type="password" placeholder="Password" name = "password" required>

                  </div>

                  <div class="field">

                     <input type="password" placeholder="Confirm password" name = "cpassword" required>

                  </div>

                  <div class="field btn">

                     <div class="btn-layer"></div>

                     <input type="submit" value="Signup">

                  </div>

               </form>

            </div>

         </div>

      </div>

      <script>

         const loginText = document.querySelector(".title-text .login");

         const loginForm = document.querySelector("form.login");

         const loginBtn = document.querySelector("label.login");

         const signupBtn = document.querySelector("label.signup");

         const signupLink = document.querySelector("form .signup-link a");

         signupBtn.onclick = (()=>{

           loginForm.style.marginLeft = "-50%";

           loginText.style.marginLeft = "-50%";

         });

         loginBtn.onclick = (()=>{ loginForm.style.marginLeft = "0%";

           loginText.style.marginLeft = "0%";

         });

         signupLink.onclick = (()=>{

           signupBtn.click();

           return false;

         });

      </script>

   </body>

</html>

**7.2. Figure 2**

**AddExpense.html**

{% extends "layout.html" %}

{% block styles %}

        <link href="/static/css/expenses.css" rel="stylesheet">

{% endblock %}

{% block title %}

    Add Expenses

{% endblock %}

{% block main %}

    <h1>Add Expenses</h1>

    <br>

    <p><small>Tip: Delete a row by clicking/tapping the row number in the '#' column and then clicking/tapping the 'Delete Row' button. Must have at least 2 rows.</small></p>

    <form action="/addexpenses" method="post" id="expenseForm" autocomplete="off">

        <input type="hidden" name="csrf\_token" value="{{ csrf\_token() }}" />

        <div class="table-responsive-sm">

        <table id="expenseTable" class="table table-hover">

          <thead>

            <tr>

              <th scope="col">#</th>

              <th scope="col">Description</th>

              <th scope="col">Category</th>

              <th scope="col">Date</th>

              <th scope="col">Payer</th>

              <th scope="col">Amount</th>

            </tr>

          </thead>

          <tbody>

            <tr>

              <td onclick="selectRow(this);">1 <i class="far fa-hand-pointer"></i><br>

              </td>

              <td>

                  <textarea class="form-control-sm" name="description.1" form="expenseForm" required maxlength="200" autofocus></textarea>

              </td>

              <td>

                  <select id="category" class="form-control-sm" name="category.1" form="expenseForm" required>

                    {% for category in categories %}

                      <option value="{{ category['name'] }}">{{ category['name'] }}</option>

                    {% endfor %}

                  </select>

              </td>

              <td>

                  <input type="date" class="form-control-sm" name="date.1" form='expenseForm' required value="{{ date }}">

              </td>

              <td>

                  <select class="form-control-sm" name="payer.1" form="expenseForm" required>

                    <option value="Self">Self</option>

                    {% for payer in payers %}

                    <option value="{{ payer['name'] }}">{{ payer['name'] }}</option>

                    {% endfor %}

                  </select>

              </td>

              <td>

                  <!-- Modified RegEx from Gary @ StackOverflow: https://stackoverflow.com/a/16242575 -->

                  <!-- TODO: change input type from 'text' to 'number' to remove the need for regex? Look at my implementation on budgets.html and use the same thing here -->

                  <input type="text" class="form-control-sm" name="amount.1" form='expenseForm' size="10" placeholder="$" required maxlength="10" pattern="(?=.\*?\d)^(([1-9]\d{0,2}(\d{3})\*)|\d+)?(\.\d{1,2})?$" title="Format must be currency value without dollar sign or commas e.g. 1, 2.50, 1500.75">

              </td>

            </tr>

          </tbody>

        </table>

        </div>

        <br>

        <button type="button" id="btnNewRow" class="btn btn-primary" onclick="addRow();">New Row</button>

        <button type="button" id="btnDeleteRow" class="btn btn-danger" onclick="removeRow(selectedRow);" disabled>Delete Row</button>

        <button type="submit" id="btnExpense" class="btn btn-success">Save</button>

    </form>

    <br>

    <script src="/static/js/addexpenses.js"></script>

    <script>

      var categoryData = JSON.stringify({{ categories | tojson }});

      var dateData = "{{ date }}";

      var payersData = JSON.stringify({{ payers | tojson }});

      loadData(categoryData, dateData, payersData);

    </script>

{% endblock %}

**7.3. Database:**

CREATE FUNCTION FUNCTION1 (VARNAME VARCHAR(128))

RETURNS INTEGER

NO EXTERNAL ACTION

F1: BEGIN ATOMIC

-- #####################################################################

-- # Returns count of all tables created by and VARNAME

-- #####################################################################

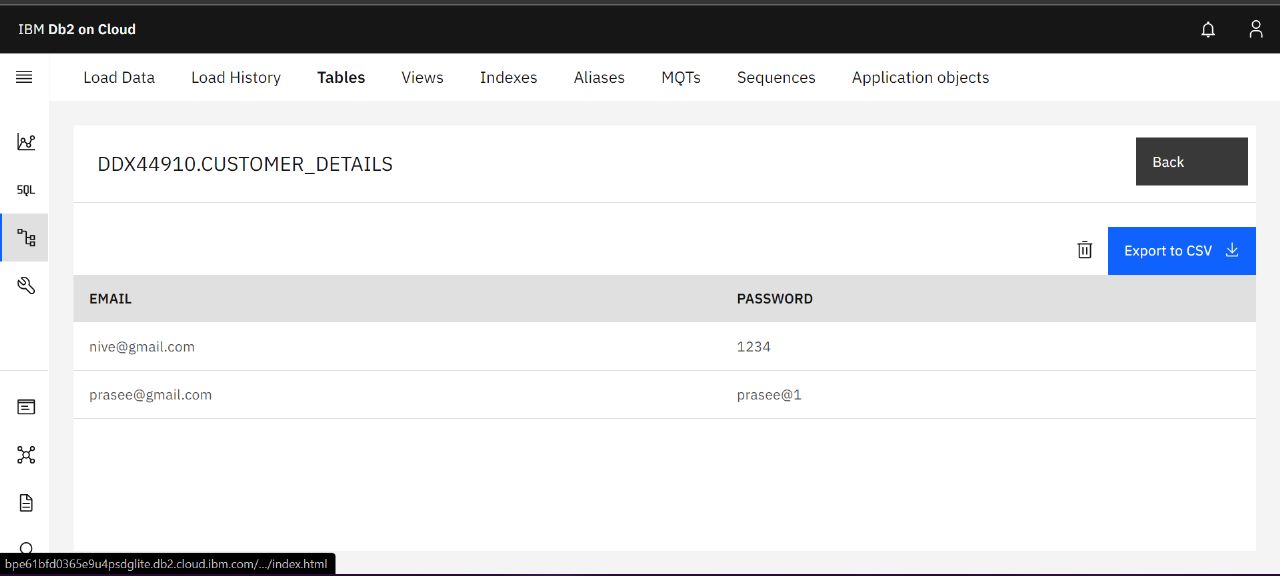
RETURN

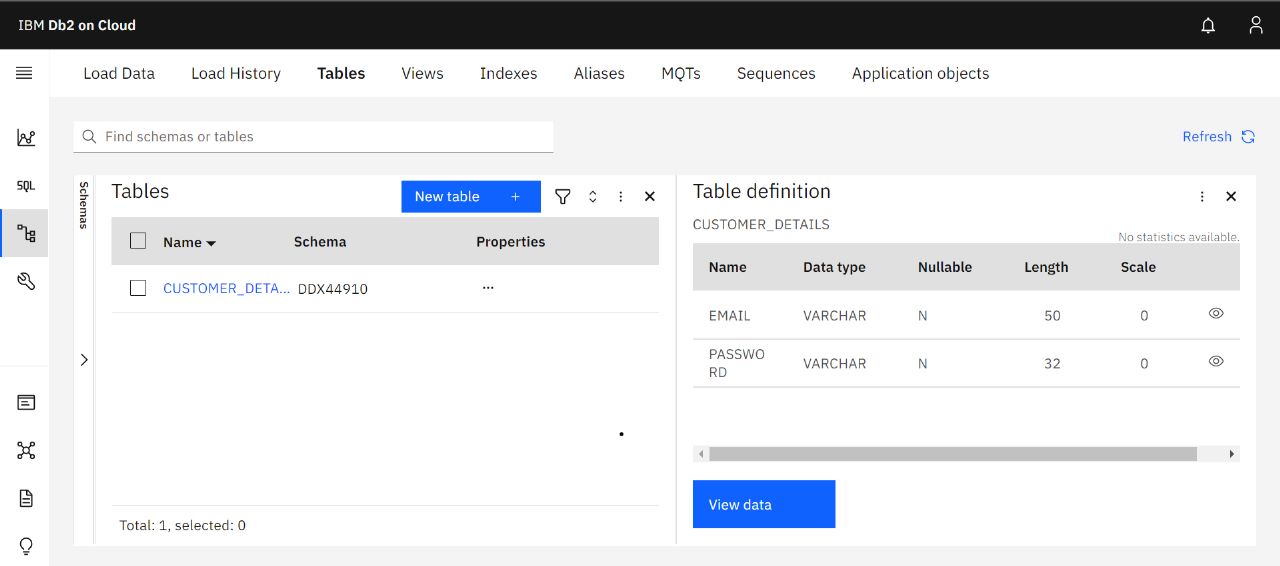
SELECT COUNT(\*)

FROM SYSIBM.SYSTABLES

WHERE CREATOR = '' AND NAME = VARNAME@INNER\_TERMINATOR

END@OUTER\_TERMINATOR





**8. TESTING**

## TEST CASES

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

## Test Case Format

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

* **Test Case Name**: A test case should have a name or title that is self- explanatory.
* **Test Case Description:** The description should tell the tester what they’re going to test in brief.
* **Pre-Conditions:** Any assumptions that apply to the test and any

pre-c ondition that must be met prior to the test being executed should be listed here.

* **Test Case Steps:** The test steps should include the necessary data and information on how to execute the test. The steps should be clear and brief, without leaving out essential facts.
* **Test Data:** It’s important to select a data set that gives sufficient coverage. Select a data set that specifies not only the positive scenarios b ut negative ones as well.
* **Expected Result**: The expected results tell the tester what they should experience as a result of the test steps.
* **Actual Result:** They specifies how the application actually behaved while test cases were being executed.
* **Comments:** Any other useful information such as screenshots that t ester want’s to specify can be included here.

**8.2. USER ACCEPTANCE TESTING**

**1.Purpose of Document**

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

UAT is performed by:

* + - * Client
      * End users



# 2.Defect Analysis

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

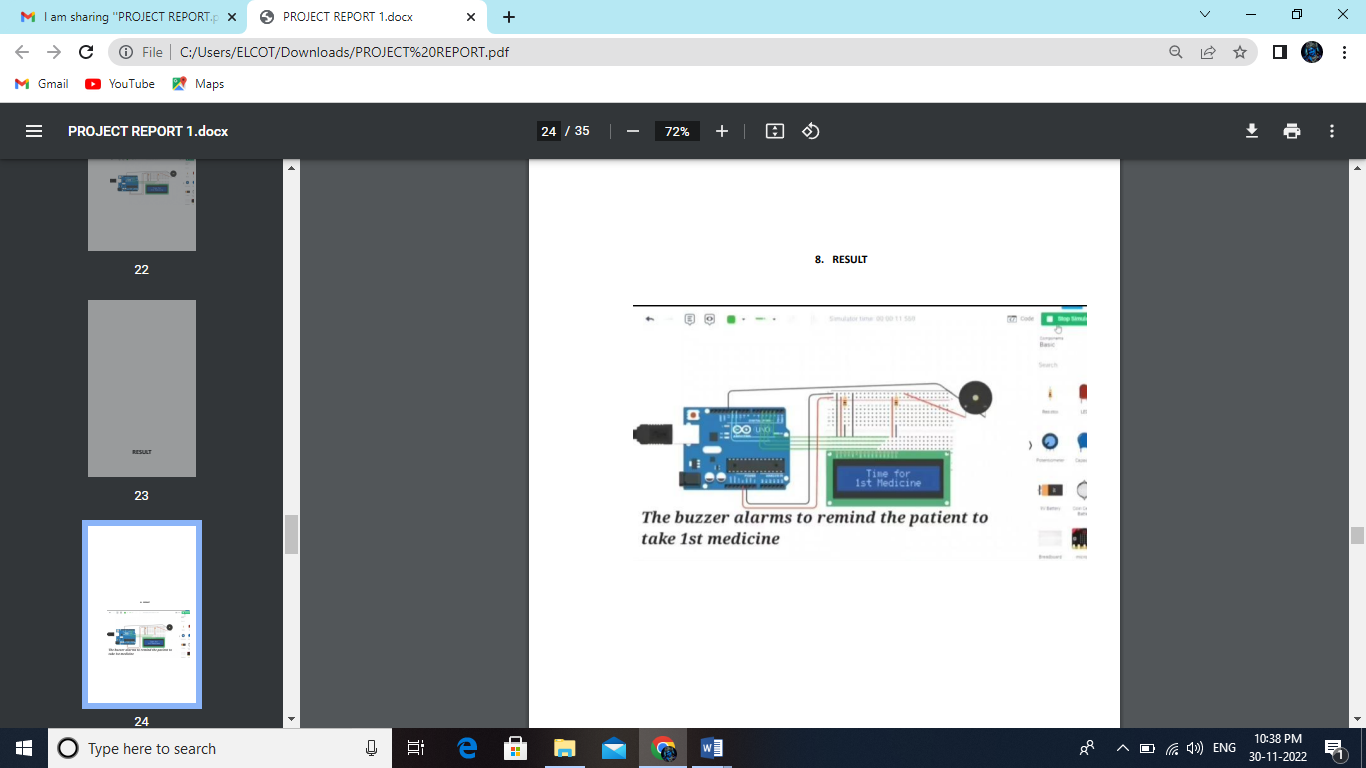
**3.Test Case Analysis:**

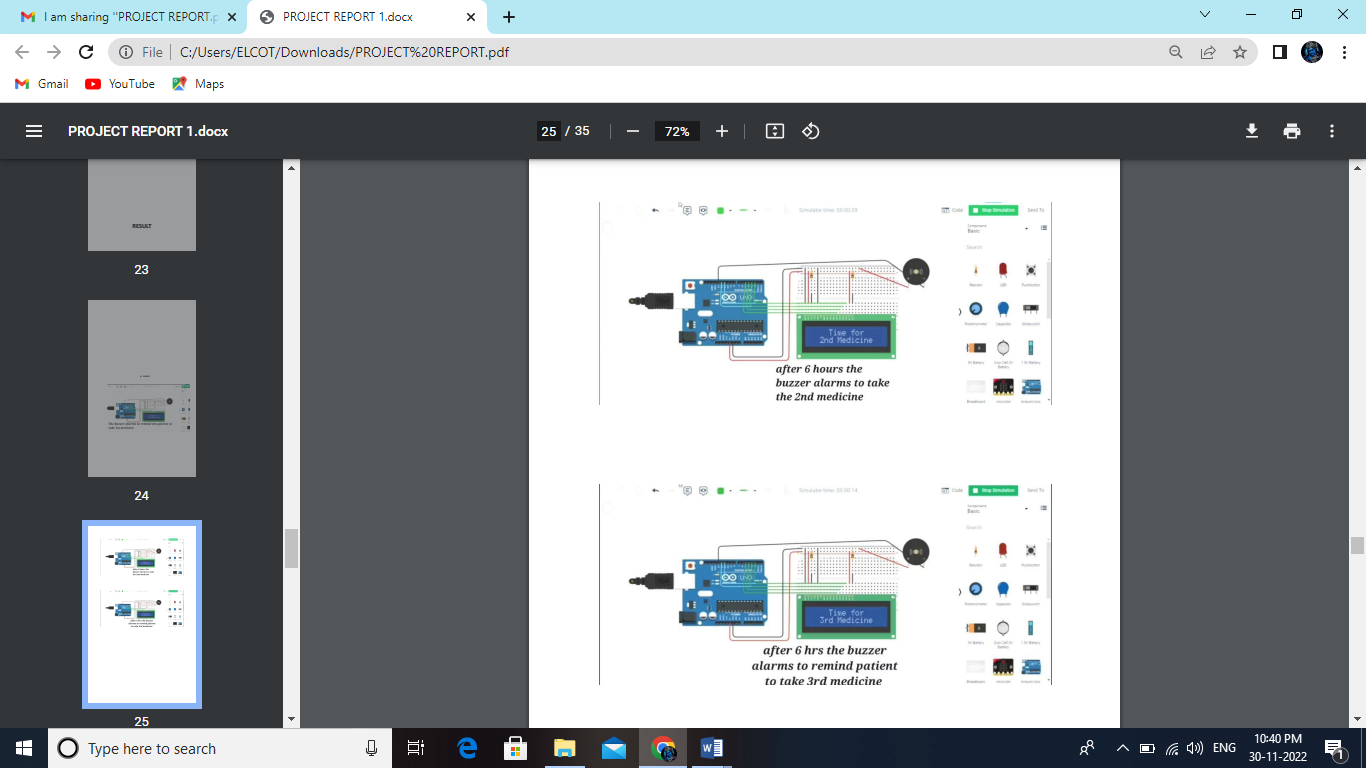
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Total Cases** | **Not Tested** | **Fail** | **Pass** |
| Login Page | 5 | 0 | 0 | 5 |
| Node Red Dashboard | 32 | 0 | 0 | 32 |
| IBM Watson IOT platform | 2 | 0 | 0 | 2 |
| MIT App Inventor | 3 | 0 | 0 | 3 |

**9.RESULTS**

* 1. **PERFORMANCE METRICS**

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





* 1. **ADVANTAGES & DISADVANTAGES**

**Advantages**

* Help the elderly people to take their medicine at the correct time.
* Avoid personal assistants or caretakers needed for medically sick people.
* Cost efficient.
* Can store multiple data and many notifications can be generated.
* Since it includes voice assistance, even blind people can use our device.

**Disadvantages**

* Makes people lethargic and makes them dependent always on others.
* Requires a stable internet connection.

**10.CONCLUSION**

The project offers the elderly or medically sick people a personal assistant which reminds them of the medicines to be consumed at the particular time. Skipping tablets may lead to serious problems if the person has a severe illness and this can be avoided. Since the cloud is integrated with the mobile application, numerous data can be fed into the database and notifications can be generated. The mobile application developed is highly customizable by the user and easy to use.

**11.FUTURE SCOPE**

The project can be further developed by bringing into the feature of informing the medicine name during the notification. The voice assistance which is given can be customized by adding the user's voice or the caretaker’s voice. Further the mobile application can update medicines by taking voice commands as an input from the user.

**12. APPENDIX:**

**12.1. SOURCE CODE**

#include <LiquidCrystal.h>

#define D4 5

#define D5 4

#define D6 3

#define D7 2

#define E 11

#define RS 12

int buzz= 13;

LiquidCrystal LCD(12, 11, 5, 4, 3, 2);

void setup()

{

pinMode(buzz, OUTPUT);

// set up the LCD's number of columns and rows:

LCD.begin(16, 2);

}

void loop()

{

LCD.setCursor(4, 0);

LCD.print("STAY HEALTHY");

LCD.setCursor(2, 1);

LCD.print("GET WELL SOON");

delay(10000);

LCD.clear();

LCD.setCursor(4, 0);

LCD.print("Time for");

LCD.setCursor(2, 1);

LCD.print("1st Medicine");

digitalWrite(buzz, HIGH);

delay(200000);

digitalWrite(buzz, LOW);

LCD.clear();

delay(36000000);

LCD.setCursor(4, 0);

LCD.print("STAY HEALTHY");

LCD.setCursor(2, 1);

LCD.print("GET WELL SOON");

delay(10000);

LCD.clear();

LCD.setCursor(4, 0);

LCD.print("Time for");

LCD.setCursor(2, 1);

LCD.print("2nd Medicine");

digitalWrite(buzz, HIGH);

delay(200000);

digitalWrite(buzz, LOW);

LCD.clear();

delay(36000000);

LCD.setCursor(4, 0);

LCD.print("STAY HEALTHY");

LCD.setCursor(2, 1);

LCD.print("GET WELL SOON");

delay(10000);

LCD.clear();

LCD.setCursor(4, 0);

LCD.print("Time for");

LCD.setCursor(2, 1);

LCD.print("3rd Medicine");

digitalWrite(buzz, HIGH);

delay(200000);

digitalWrite(buzz, LOW);

LCD.clear();

delay(36000000);

}

**12.2. GITHUB & PROJECT DEMO LINK**

GitHub link: [IBM-EPBL/IBM-Project-53095-1661313857](https://github.com/IBM-EPBL/IBM-Project-53095-1661313857)

Demo link: <https://youtu.be/mIlIMZeMF_U>