**DETECTING PARKINSON’S DISEASE USING MACHINE**

**LEARNING**

**IBM-Project-8946-1658939044**

**NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READLINESS FOR INNOVATION, EMPLOYNMENT AND ENTERPRENEURSHIP**

**A PROJECT REPORT**

**Submitted By**

**C.KARTHIKEYAN - 921319205057**

**M.MATHESH - 921319205071**

**B.RAJESH KUMAR - 921319205109**

**A.LOKHU PRASANTH - 921319205065**

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

***of***

**BACHELOR OF TECHNOLOGY**

**in**

**INFORMATION TECHNOLOGY**

**PSNA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**DINDIGUL 624622**

**TABLES OF CONTENTS**

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

**Parkinson's Disease Detection**

**Project Report**

**Introduction**

**Project Overview:**

Parkinson’s disease (PD) is a neurodegenerative movement disease where the symptoms gradually develop to start with a slight tremor in one hand and a feeling of stiffness in the body and it becomes worse over time.

At present there is no conclusive result for this disease by non-specialist clinicians, particularly in the early stage of the disease where identification of the symptoms is very difficult. The disease is majorly said to be affecting the individuals who are living in village areas with their respective ages between 40 and 50.

Parkinson's disease detection system has been designed to detect the Parkinson's disease in a patient given their hand drawn spiral or wave images. The system is built using fundamental concepts of Data analytics and Computer Vision that are trained to differentiate between healthy and Parkinson hand drawn images.

**Purpose:**

Lack of adequate knowledge poses a barrier in the provision of appropriate treatment and care for individuals with Parkinson’s Disease. Parkinson's disease affects over 6 million people worldwide. There is no proper testing procedure defined to detect the disease as the disease examination varies at different instances of the medical operation. We propose our model to detect the disease at very less error rate.

**Literature Survey:**

**Existing problem:**

Due to insufficient resources and awareness in underdeveloped countries, proper and timely PD detection is highly challenged. Besides, all PD patients’ symptoms are neither the same nor they all become pronounced at the same stage of the illness. Therefore, this work aims to combine more than one symptom by collecting data and detecting PD with the help of a cloud-based machine learning system for monitoring the PD patients in the developing countries.

**References:**

1. Anitha R, Nandhini T, Sathish Raj S, Nikitha V, “Early detection of Parkinson’s Disease using Machine Learning”, 2020
2. Md. SakiburRahman Sajal, Md. Tanvir Ehsan, Ravi VaidyaNathan, Shouyan Wang, Tpu Aziz and Khondaker Abdullah Al Mamun, “Telemonitoring

Parkinson’s Disease using Machine Learning by combining Tremor and Voice Analysis”, 2020

1. Jaichandran R, Leelavathy S , Usha Kiruthika S , Goutham Krishna , Mevin John Mathew and Jomon Baiju, “Machine Learning technique based

Parkinson’s Diseases Detection from Spiral and Voice Inputs”, 2020

1. Radouani Laila, Lagdali Salwa, Rziza Mohammed, “Detection of voice impairment for Parkinson's disease using machine learning tools”, 2021
2. C K Gomathy, B.Varshini, B.Varsha, B.Dheeraj Kumar Reddy, “The

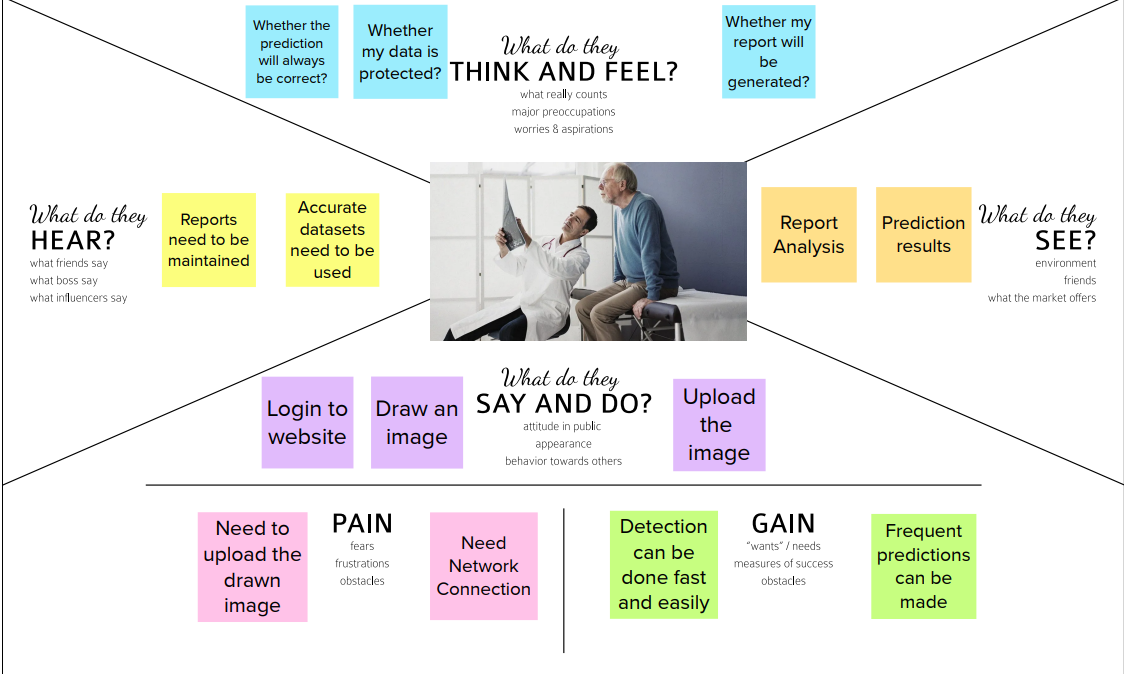
Parkinson’s Disease Detection using Machine Learning Techniques.”, 2021

**Problem Statement Definition:**

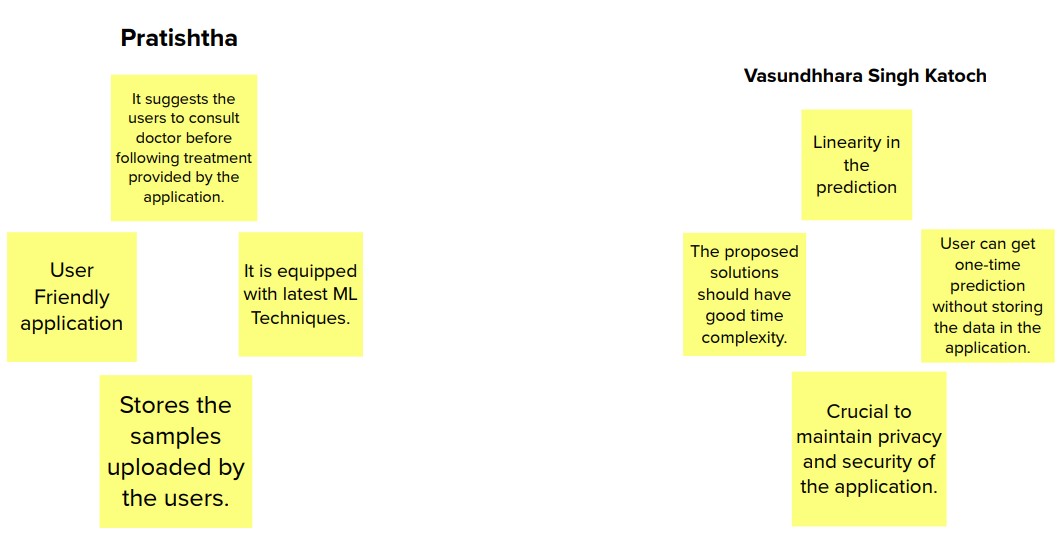
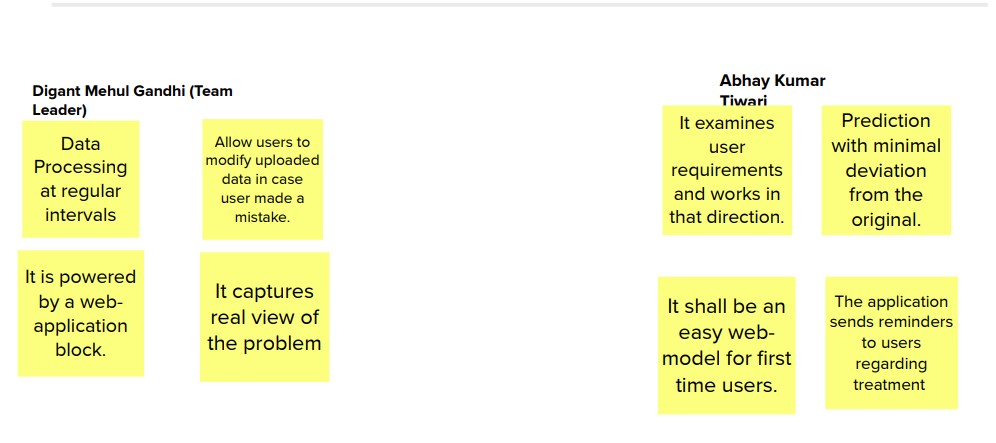
By processing the handdrawn spiral and wave images of the patients we can create a model to learn the difference between healthy and Parkinson affected drawing patterns. The patients provides their handdrawn image and the our machine learrning model predicts whether the patient is affected by Parkinson's disease.

**Ideation and Proposed Solution:**

**Empathy map canvas:**



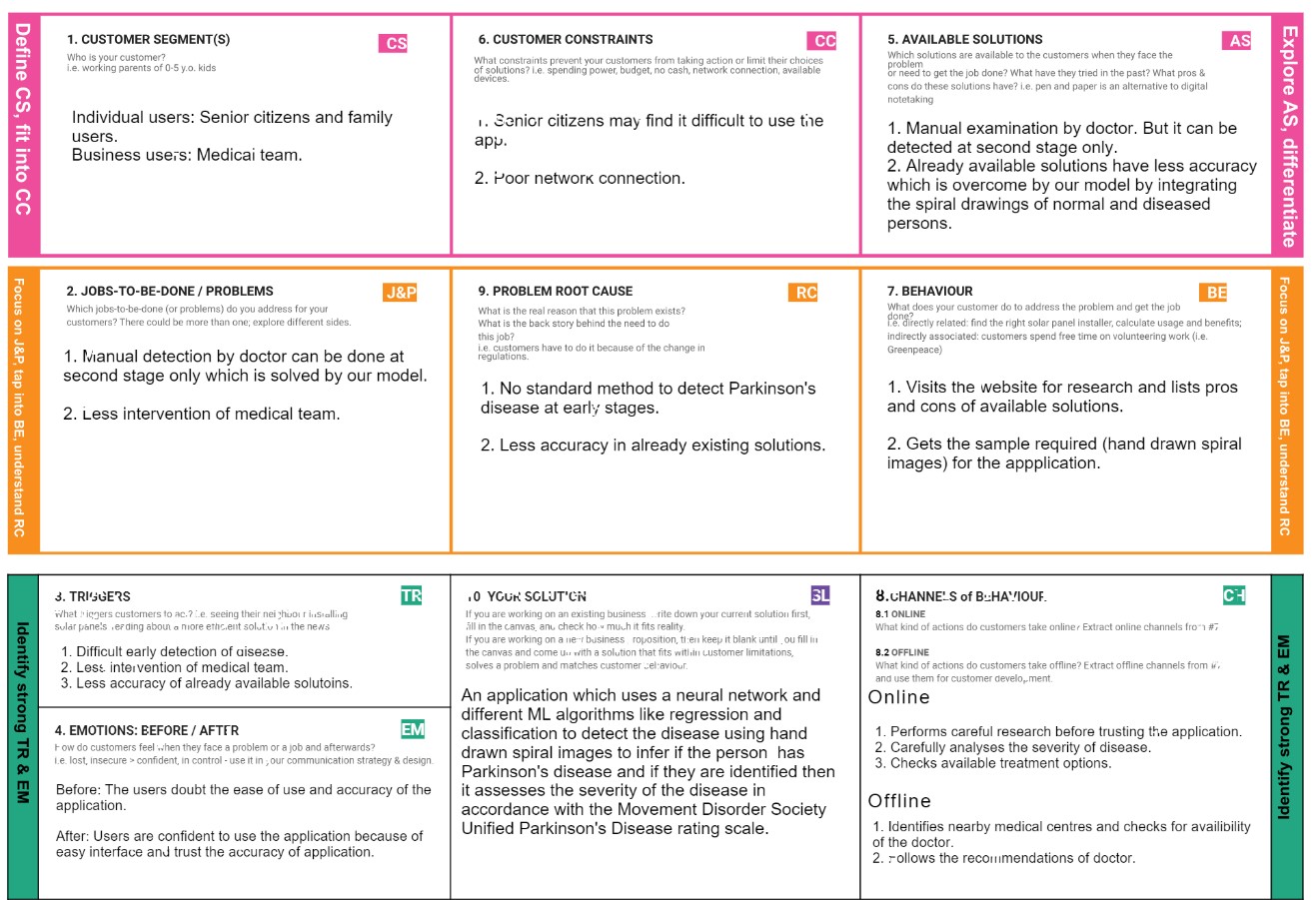
**Ideation and Brainstorming:**



**Proposed Solution:**

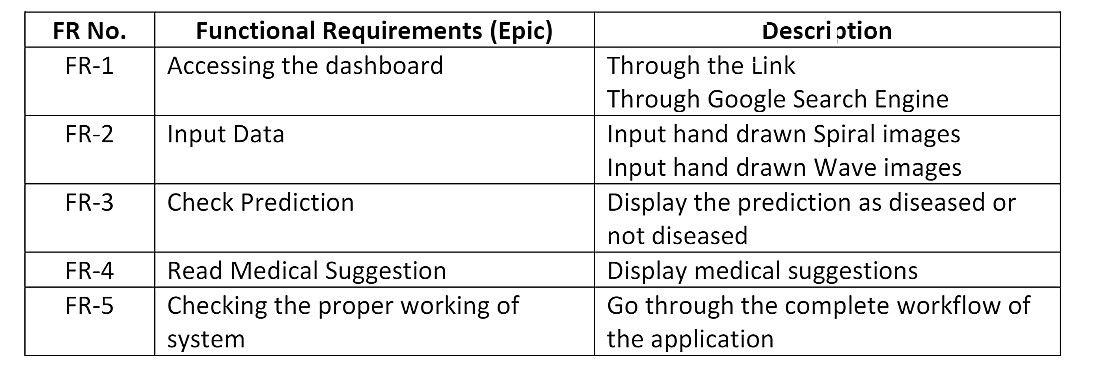
|  |  |
| --- | --- |
| Idea / Solution description | Our model processes the hand drawn spiral and wave images using a neural network that infers whether the patient has Parkinson's disease. |
| Novelty / Uniqueness | This application offers medical advice and solutions as the next step after user is confirmed based on the presence of Parkinson’s disease. This can be used direct by medical team for analysing and offering the solutions at much positive scaling time. |

**Problem Solution fit:**

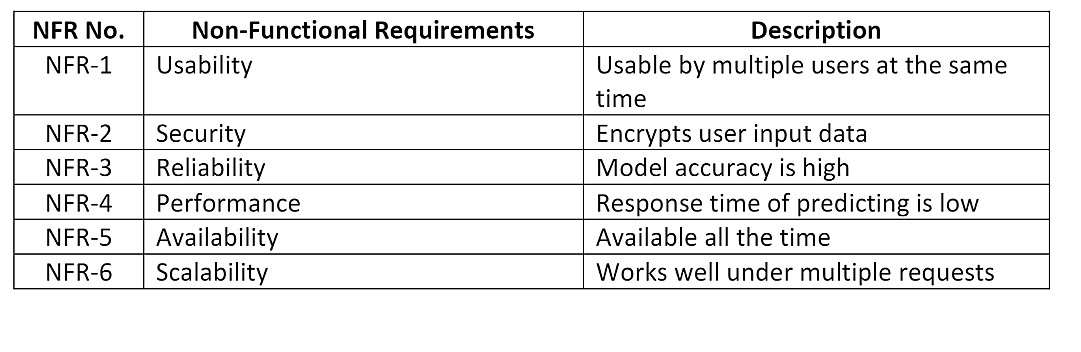


**Requirement Analysis:**

**Functional Requirements:**

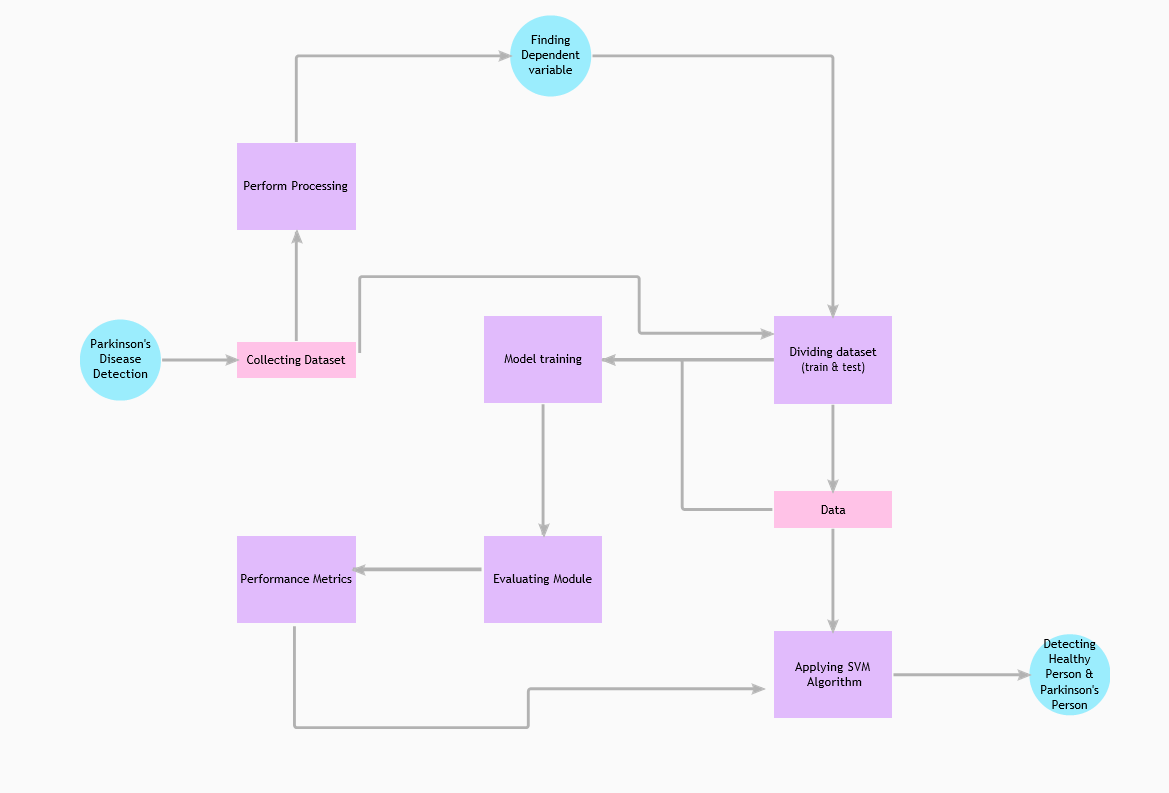


**Non-functional Requirements:**

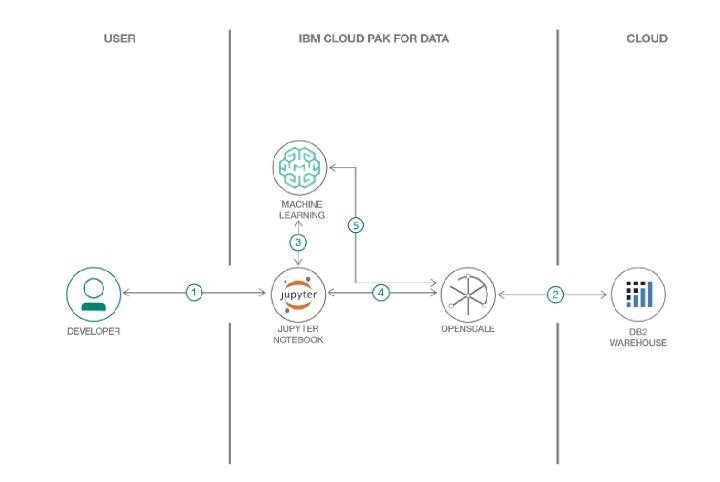


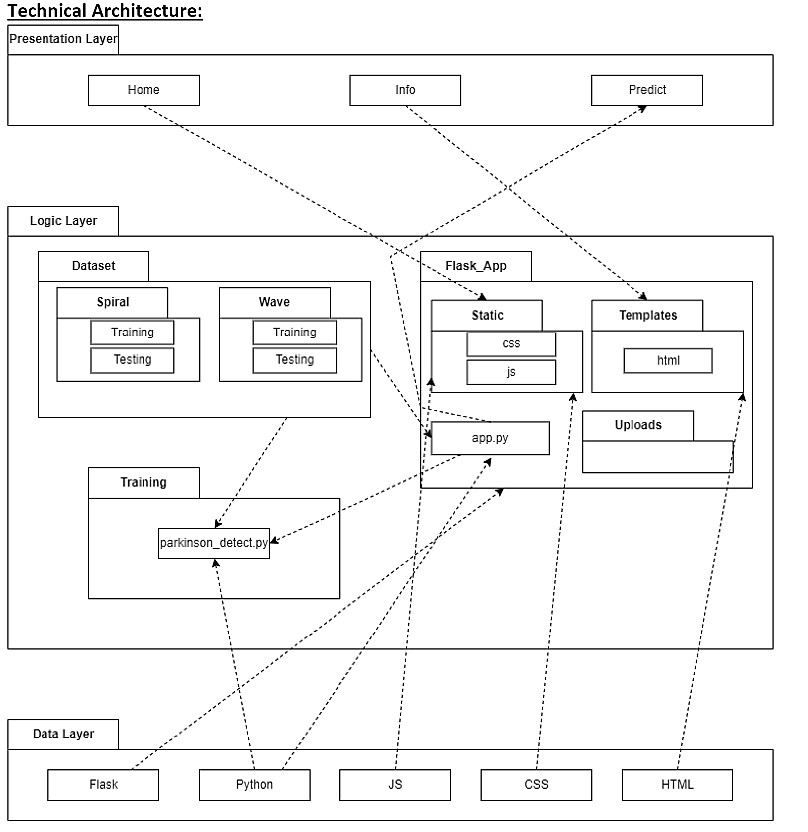
**Project Design:**

**Data flow diagrams:**

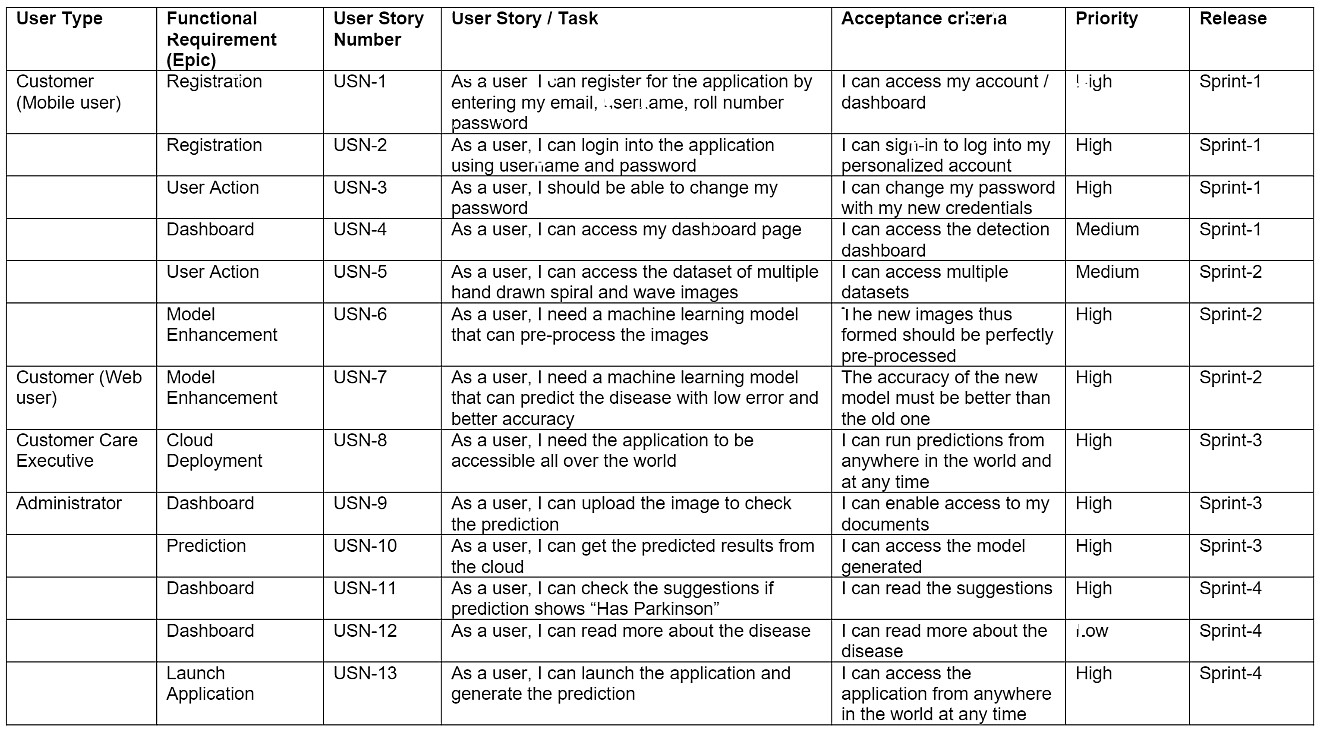


**Solution and technical architecture:**





**User Stories:**



**Project Planning and Scheduling:**

**Sprint Planning and Estimation:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement**  **(Epic)** | **User**  **Story**  **Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint1 | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | 3 | High | Karthikeyan C  Mathesh M  Rajesh Kumar B  Lokhu Prasanth A |
| Sprint1 |  | USN-2 | As a user, I will receive confirmation email once I have registered for the application | 2 | High | Karthikeyan C  Mathesh M  Rajesh Kumar B  Lokhu Prasanth A |
| Sprint2 |  | USN-3 | As a user, I can register for the application through  Facebook | 3 | Low | Karthikeyan C  Mathesh M  Rajesh Kumar B  Lokhu Prasanth A |
| Sprint2 |  | USN-4 | As a user, I can register for the application through  Gmail | 3 | Medium | Karthikeyan C  Mathesh M  Rajesh Kumar B  Lokhu Prasanth A |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint2 | Login | USN-5 | As a user, I can log into the application by entering email & password | 3 | High | Karthikeyan C  Mathesh M  Rajesh Kumar B  Lokhu Prasanth A |
| Sprint3 | Dashboard | USN-6 | As a user, I can upload my images and get my details. | 3 | High | Karthikeyan C  Mathesh M  Rajesh Kumar B  Lokhu Prasanth A |
| Sprint1 | Logout | USN-7 | As a user I can logout successfully. | 2 | Medium | Karthikeyan C  Mathesh M  Rajesh Kumar B  Lokhu Prasanth A |
| Sprint4 | Feedback | USN-8 | A customer care executive, I can able to interact with all the customer and get their feedback which is used to enhance the scope of the project. | 2 | Medium | Karthikeyan C  Mathesh M  Rajesh Kumar B  Lokhu Prasanth A |
| **Sprint** | **Functional**  **Requirement**  **(Epic)** | **User**  **Story**  **Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
|  |  |  |  |  |  |  |
| Sprint3 | Image processing localization | USN-9 | The uploaded image is preprocessed and fed into trained model. | 3 | High | Karthikeyan C  Mathesh M  Rajesh Kumar B  Lokhu Prasanth A |
| Sprint4 | Classification and prediction | USN-9 | The model classifies and predicts the type of disease. | 3 | High | Karthikeyan C  Mathesh M  Rajesh Kumar B  Lokhu Prasanth A |
| Sprint4 | Report generation | USN-10 | Based on the prediction of Parkinson’s disease, the health care is generated to provide the feedback. | 2 | Medium | Karthikeyan C  Mathesh M  Rajesh Kumar B  Lokhu Prasanth A |

**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)** |
| Sprint1 | 20 | 6 Days | 20 Oct  2022 | 26 Oct 2022 | 20 | 26 Oct 2022 |
| Sprint2 | 20 | 6 Days | 27 Oct  2022 | 02 Nov 2022 | 20 | 31 O ct  2022 |
| Sprint3 | 20 | 6 Days | 02 Nov  2022 | 08 Nov 2022 | 20 | 06 Nov 2022 |
| Sprint4 | 20 | 6 Days | 08 Nov  2022 | 14 Nov 2022 | 20 | 08 Nov 2022 |

**Coding and Solutioning:**

**Register and Login:**

In the home page, the links for login and register are available. If the user is new then he/she can register for a new account using the register button. If the user has already registered, he/she can login using his username and password.

**home.html** <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-

ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='styles.css')

}}" />

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

<style>

.bg-img{

background-image: url("{{ url\_for('static',filename='pd.jpg') }}");

}

</style>

</head>

<body>

<div class="col-md-8">

{% with messages = get\_flashed\_messages(with\_categories=true) %}

{% if messages %}

{% for category, message in messages %}

<div class="alert alert-{{category}}">

{{ message }}

</div>

{% endfor %}

{% endif %}

{% endwith %}

{% block content %} {% endblock %}

</div>

<div class="bg-img">

<div class="topnav">

<div class="topnav-right">

<a href="{{ url\_for('login') }}">Login Page</a>

<a href="{{ url\_for('register') }}">Register Page</a>

<a href="{{ url\_for('update') }}">Update Password</a>

</div>

</div>

</div>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script

src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js" integrity="sha384-

UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1

" crossorigin="anonymous"></script>

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js" integrity="sha384-

JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

</body>

</html>

# login.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-

ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

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

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

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

<style> body{

background-image: url("{{ url\_for('static',filename='login.jpg') }}"); min-height: 520px;

}

</style>

</head>

<body>

<div class="text">

<div class="col-md-8">

{% with messages = get\_flashed\_messages(with\_categories=true) %}

{% if messages %}

{% for category, message in messages %}

<div class="alert alert-{{category}}">

{{ message }}

</div>

{% endfor %}

{% endif %}

{% endwith %}

{% block content %} {% endblock %}

</div>

<h1 style="font-family: Copperplate, fantasy;">Enter Login Details</h1><br>

<form method="POST" action="">

{{ form.hidden\_tag() }}

<fieldset class="form-group">

<div>

{{ form.username.label(class="form-control-label") }}

{% if form.username.errors %}

{{ form.username(class="form-control form-control-lg is-invalid") }}

<div class="invalid-feedback">

{% for error in form.username.errors %}

<span>{{ error }}</span>

{% endfor %}

</div>

{% else %}

{{ form.username(class="form-control form-control-lg") }} {% endif %}

</div>

<div>

{{ form.password.label(class="form-control-label") }}

{% if form.password.errors %}

{{ form.password(class="form-control form-control-lg is-invalid") }}

<div class="invalid-feedback">

{% for error in form.password.errors %}

<span>{{ error }}</span>

{% endfor %}

</div>

{% else %}

{{ form.password(class="form-control form-control-lg") }} {% endif %}

</div>

</fieldset>

<div class="form-group">

{{ form.submit(class="btn btn-outline-info") }}

</div>

<small class="text-muted ml-2">

<a href='{{url\_for('register')}}'>Do not have an account? Sign Up?</a> </small>

</form>

</div>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script

src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js" integrity="sha384-

UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1

" crossorigin="anonymous"></script>

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js" integrity="sha384-

JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

</body>

</html>

**register.html** <!DOCTYPE html>

<html lang="en">

<head>

<link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

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

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

}}" />

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<style> body{

background-image: url("{{ url\_for('static',filename='login.jpg') }}"); min-height: 520px;

}

</style>

</head>

<body>

<div class="text" style="margin-top: 50px;padding: 30px 30px;">

<h1 style="font-family: Copperplate, fantasy;">Register With Us</h1> <br>

<!-- <form method="POST" action="">

{{ form.hidden\_tag() }}

{{ form.email }} <br> <br>

{{ form.username }} <br> <br>

{{ form.rollnumber }} <br> <br>

{{ form.password }} <br> <br>

{{ form.submit }} <br> <br>

</form> -->

<form method="POST" action="">

{{ form.hidden\_tag() }}

<fieldset class="form-group">

<!-- <legend class="border-bottom mb-4">Registration Page</legend> -->

<div>

{{ form.email.label(class="form-control-label") }}

{% if form.email.errors %}

{{ form.email(class="form-control form-control-lg is-invalid") }}

<div class="invalid-feedback">

{% for error in form.email.errors %}

<span>{{ error }}</span>

{% endfor %}

</div>

{% else %}

{{ form.email(class="form-control form-control-lg") }}

{% endif %}

</div>

<div>

{{ form.username.label(class="form-control-label") }}

{% if form.username.errors %}

{{ form.username(class="form-control form-control-lg is-invalid") }}

<div class="invalid-feedback">

{% for error in form.username.errors %}

<span>{{ error }}</span>

{% endfor %}

</div>

{% else %}

{{ form.username(class="form-control form-control-lg") }} {% endif %}

</div>

<div>

{{ form.rollnumber.label(class="form-control-label") }}

{% if form.rollnumber.errors %}

{{ form.rollnumber(class="form-control form-control-lg is-invalid") }}

<div class="invalid-feedback">

{% for error in form.rollnumber.errors %}

<span>{{ error }}</span>

{% endfor %}

</div>

{% else %}

{{ form.rollnumber(class="form-control form-control-lg") }} {% endif %}

</div>

<div>

{{ form.password.label(class="form-control-label") }}

{% if form.password.errors %}

{{ form.password(class="form-control form-control-lg is-invalid") }}

<div class="invalid-feedback">

{% for error in form.password.errors %}

<span>{{ error }}</span>

{% endfor %}

</div>

{% else %}

{{ form.password(class="form-control form-control-lg") }}

{% endif %}

</div>

</fieldset>

<div class="form-group">

{{ form.submit(class="btn btn-outline-info") }}

</div>

<small class="text-muted ml-2">

<a href="{{ url\_for('login') }}">Already have an account? Log In</a>

</small>

</form>

</div>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script

src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js" integrity="sha384-

UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1

" crossorigin="anonymous"></script>

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js" integrity="sha384-

JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

</body>

</html>

**Prediction:**

After logging in user can test if he/she has pakinson's disease by uploading handdrawn spiral or wave image. The model predicts whether the user has Parkinson's disease. If the user has Parkinson's disease the application offers the user medical suggestions and healthy diets.

# predict.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<style>

body{

background-image: url("{{ url\_for('static', filename='predict.jpg') }}"); background-repeat: no-repeat;

background-size: cover; height: 100%;

}

.text{ text-align: center; align-items: center; justify-content: center; position: absolute;

top: 25%;; bottom: 25%;

left: 0;

right: 0; margin: auto; font-weight: bold; color:aliceblue;

background-color: rgb(0,0,0); /\* Fallback color \*/ background-color: rgba(0,0,0, 0.7);

font-weight: bold; border: 3px solid #f1f1f1;

width: 40%;

} a{

text-decoration: none;

color:aliceblue;

}

a:hover {

text-decoration: underline;

}

</style>

<title>Prediction</title>

</head>

<body>

<div class="text">

<h1>The predicted result is:</h1>

<h1>{{predict}}</h1>

<a href="/welcome">Click here to go Back to the Dashboard</a> </div>

<script src="{{ url\_for('static', filename='confetti.js') }}"></script>

<script> const start = () => { setTimeout(function() { confetti.start()

}, 1000); // 1000 is time that after 1 second start the confetti ( 1000 = 1 sec)

};

// for stopping the confetti

const stop = () => { setTimeout(function() { confetti.stop()

}, 5000); // 5000 is time that after 5 second stop the confetti ( 5000 = 5 sec)

};

// after this here we are calling both the function so it works start(); stop();

</script>

</body>

</html>

# app.py (prediction part)

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

f=request.files['file'] #requesting the file

basepath=os.path.dirname(\_\_file\_\_)#storing the file directory

filepath=os.path.join(basepath, "uploads", f.filename)#storing the file in uploads folder

f.save(filepath)#saving the file #Loading the saved model

print("[INFO] Loading model...")

dataset = request.form['dataset'] if dataset=='spiral':

m="C:/Users/Digant Gandhi/OneDrive/Desktop/Sprint4/parkinson.pkl" else:

m="C:/Users/Digant Gandhi/OneDrive/Desktop/Sprint4/parkinson\_wave.pkl" model = joblib.load(m) image = cv2.imread(filepath) output = image.copy()

output = cv2.resize(output, (128, 128))

# pre-process the image in the same manner we did earlier image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

image = cv2.resize(image, (200, 200))

image = cv2.threshold (image, 0, 255, cv2.THRESH\_BINARY\_INV | cv2.THRESH\_OTSU)[1]

features= quantify\_image(image) res=model.predict([features]) if(res[0]):

value="Parkinson" return redirect(url\_for('suggestion')) else:

value="Healthy"

return render\_template('predict.html',predict=value)

**Database Schema:**

id INTEGER PRIMARY KEY AUTOINCREMENT,

email TEXT NOT NULL, username TEXT NOT NULL, roll\_number INTEGER NOT NULL, pass\_word TEXT NOT NULL

**Testing:**

**Testcases:**

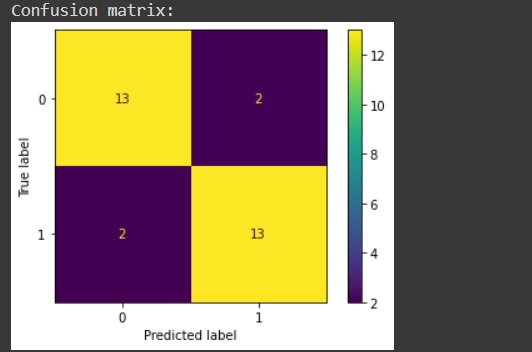
In order to test the functioning of our model, we collected a sample of Parkinson's disease and healthy handdrawn images. We tested our model against them to check if it detected the images accurately.

**User Acceptance Testing:**

The application performs as expected by detecting whether the patient has Parkinson's disease. All the other functionalities such as Login, Register, Update Password etc ae working as expected.

**Results:**

**Performance Metrics:**



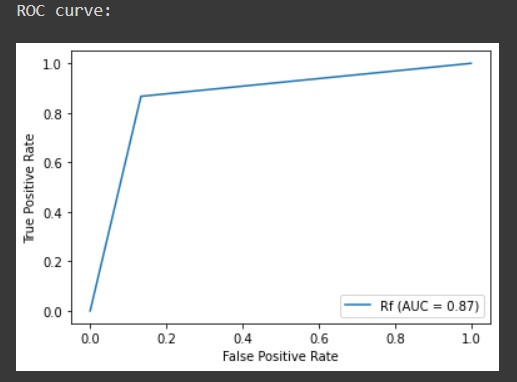
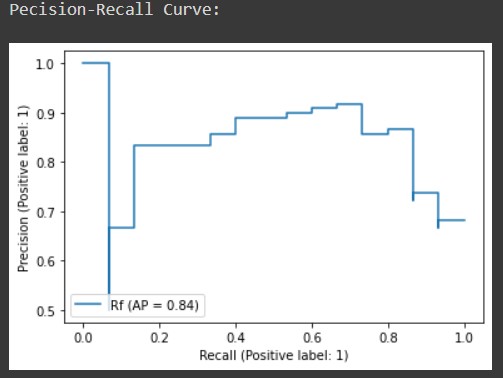
Accuracy: 0.8666666666666667

Precision: 0.8666666666666667

Recall: 0.8666666666666667

Specificity0.8666666666666667

F1 score: 0.8666666666666667



**Advantages and Disadvantages:**

**Advantages:**

* Easily accessible
* Application stays active 24X7
* Predictions are highly accurate
* User friendly and provides necessary information about the disease such as symptoms and causes.
* Provides medical suggestions along with results for those affected with the disease

**Disadvantages:**

* May not work properly on huge load (i.e. large number of requests per second)

**Conclusion:**

We have developed a web application that will help the patients to check whether they have Parkinson's disease. Thus, our application prevents expenditure on testing the disease and helps people with poor economic backgrounds. It also provides medical suggestions to those who are affected by the disease.

**Future Work:**

In future, the work can be extended to not only predict the disease but also to find out the severity of the disease. According to the severity of the disease necessary medical suggestions and medications can be provided.

**Appendix:**

**Project Demo Link:**

<https://drive.google.com/file/d/151CBaKnxuiy_O64a1_Pyg0EOnrjtgsom/view?usp=share_link>

**Source Code:**

<https://github.com/IBM-EPBL/IBM-Project-8946-1658939044>