

**NUTRITION ASSISTANT APPLICATION USING  
CLOUD TECHNOLOGY  
CLOUD APP DEVELOPMENT DOMAIN**

**TEAM ID: PNT2022TMID07434**

**A PROJECT REPORT**

*Submitted by*

<b>BURRU HEMASAI</b>	<b>(721719104014)</b>
<b>KODANDA RAMUDU</b>	<b>(721719104086)</b>
<b>CHENNA KESHA VA REDDY</b>	<b>(721719104043)</b>
<b>HARIPRAKASH</b>	<b>(721719104031)</b>

**COMPUTER SCIENCE AND ENGINEERING**

**P. A. COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(Autonomous)**

**Pollachi, Coimbatore Dt. - 642 002**



**NOVEMBER 2022**

# **P. A. COLLEGE OF ENGINEERING AND TECHNOLOGY**

## **BONAFIDE CERTIFICATE**

Certified that this project report “**NUTRITION ASSISTANT APPLICATION**” is the work of “**BURRU HEMASAI, THIMMAPURAM KODANDA RAMUDU, KONDURU CHENNA KESHA VA REDDY, HARIPRAKASH**” who carried out the project work under our supervision.

### **SIGNATURE**

Dr. D. CHITRA M.E, Ph.D.,

**Professor**

### **HEAD OF THE DEPARTMENT**

Department of Computer Science  
and Engineering

P. A. College of Engineering  
and Technology

Pollachi-642 002.

### **SIGNATURE(MENTOR)**

Dr.A.KALIAPPAN M.E,Ph.D.,

**Assistant Professor**

### **SUPERVISOR**

Department of Computer science  
and Engineering

P. A. College of Engineering  
and Technology

Pollachi-642 002.

**SIGNATURE**

**FACULTY**

**EVALUATOR**

Mr.S.SURESHKUMAR

Assistant Professor

Computer Science and

Engineering

P. A. College of

Engineering and

Technology

Submitted to the Viva- Voce Examination held on -----

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

We have immense pleasure and satisfaction in expressing our hearty thanks to our beloved Chairman **Dr. P. APPUKUTTY M.E, FIE, FIV**, for his extensive support to successfully carry out this project.

We take privilege in expressing our sincere and heartfelt thanks and gratitude to our beloved Principal **Dr. T. MANIGANDAN M.E., Ph.D.**, for providing us an opportunity to carry this project work.

We express our heartfelt thanks to **Dr. D. CHITRA M.E., Ph.D.**, Professor and Head, Department of Computer Science and Engineering for her technical guidance and support.

We are obliged to express our gratitude to our Project Mentor **Dr. A. KALIAPPAN M.E., Ph.D.**, Assistant Professor Department of Computer Science and Engineering, for his technical guidance, constructive criticism and many valuable suggestions provided throughout the project work.

We also thankful to our project Evaluator **Mr.S.SURESHKUMAR.**, Assistant Professor, Department of Computer Science and Engineering and all teaching and non-teaching faculty members of Department of Computer Science and Engineering for helping us to complete the project successfully.

Last but not least, we thank our parents and all our friends for helping us to complete our project.

# ABSTRACT

The analysis of different application features shows that reflective visual feedback has a more substantial impact on healthy behaviour than the recommender. We further identify system limitations influencing this result, such as a lack of diversity, mistrust in healthiness and personalization, real-life contexts, and personal user characteristics with a qualitative analysis of semi-structured in-depth interviews.

Our results show that Nutrilize positively affects nutritional behaviour measured by the optimal intake of each nutrient. Healthy nutrition contributes to preventing non-communicable and diet-related diseases. Recommender systems, as an integral part of mHealth technologies, address this task by supporting users with healthy food recommendations.

However, knowledge about the effects of the long-term provision of health-aware recommendations in real-life situations is limited. This study investigates the impact of a mobile, personalized recommender system named Nutrilize. Finally, we discuss general knowledge acquired on the design of personalized mobile nutrition recommendations by identifying important factors, such as the users' acceptance of the recommender's taste, health, and personalization.

# TABLE OF CONTENTS

CHAPTER NO	TITILE	PAGE.NO
	<b>ABSTRACT</b>	<b>5</b>
	<b>LIST OF FIGURES</b>	<b>9</b>
	<b>LIST OF TABLES</b>	<b>9</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>10</b>
	1.1 Project Overview	<b>10</b>
	1.2 Purpose	<b>11</b>
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>12</b>
	2.1 Existing Problem	<b>12</b>
	2.2 References	<b>12</b>
	2.3 Problem Statement Definition	<b>14</b>
<b>3</b>	<b>IDEATION &amp; PROPOSED SOLUTION</b>	<b>15</b>
	3.1 Empathy Map Canvas	<b>15</b>
	3.2 Ideation & Brainstorming	<b>16</b>
	3.3 Proposed Solution	<b>16</b>
	3.4 Problem Solution fit	<b>17</b>
<b>4</b>	<b>REQUIREMENT ANALYSIS</b>	<b>18</b>

	4.1	Functional requirement	18
	4.2	Non-Functional requirement	19
<b>5</b>		<b>PROJECT DESIGN</b>	<b>20</b>
	5.1	Data Flow Diagrams	20
	5.2	Solution & Technical Architecture	20
	5.3	User Stories	21
<b>6</b>		<b>PROJECT PLANNING &amp; SCHEDULING</b>	<b>22</b>
	6.1	Sprint Planning & Estimation	22
	<b>6.2</b>	Sprint Delivery Schedule	<b>22</b>
	<b>6.3</b>	Reports from JIRA	<b>23</b>
<b>7</b>		<b>CODING &amp; SOLUTIONING</b>	<b>24</b>
	7.1	Feature1	24
	7.2	Feature2	25
	7.3	Database Schema	39
<b>8</b>		<b>TESTING</b>	<b>40</b>
	8.1	Test cases	40
	8.2	User Acceptance Testing	41
<b>9</b>		<b>RESULTS</b>	<b>43</b>
	9.1	Performance Metrics	43

<b>10</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>	<b>45</b>
<b>11</b>	<b>CONCLUSION</b>	<b>46</b>
<b>12</b>	<b>FUTURE SCOPE</b>	<b>47</b>
<b>13</b>	<b>APPENDIX</b>	<b>48</b>

Source Code

GitHub Link & Project Demo Link

## LIST OF FIGURES

<b>FIGURE NO</b>	<b>TITLE</b>	<b>PAGE NO</b>
1.2.1	Technical Architecture	11
3.1.1	Empathy Map Canvas	15
3.2.1	Ideation&Brainstroming	16
3.4.1	Problem Solution Fit	17
4.1.1	Functional requirements	19
5.1.1	Data Flow Diagrams	20
5.2.1	Solution & Technical Architecture	20
5.3.1	User Stories	20
6.3.1	Reports from JIRA	24
7.3.1	Database Schema	39



9.1.1	Performance Metrics	44
9.1.2	Performance Metrics	45
9.1.3	Performance Metrics	45

## LIST OF TABLES

TABLE NO	TITLE	PAGE NO
3.1.1	Proposed Solution	17
4.1.1	Functional requirements	19
4.2.1	Non-Functional Requirements	20
6.2.1	Sprint Delivery Schedule	23
8.1.1	Test Scenario	40
8.2.1	User AcceptanceTesting	41

## CHAPTER 1

### 1 INTRODUCTION

#### 1.1 Project Overview

Due to the ignorance of healthy food habits, obesity rates are increasing at an alarming speed, and this is reflective of the risks to people's health. People need to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity.

However, although food packaging comes with nutrition (and calorie) labels, it's still not very convenient for people to refer to App-based nutrient dashboard systems which can analyze real time images of a meal and analyze it for nutritional content which can be very handy and improves the dietary habits, and therefore, helps in maintaining a healthy lifestyle.

**Skills Required:** IBM Cloud, HTML, Javascript, IBM Cloud Object Storage, Python-Flask, Kubernetes, Docker, IBM DB2, IBM Container Registry

->First user interacts with WEB APP to upload image

->IBM DB2 stores the user data and stores the food detail

Building a WEB APP that automatically estimates food attributes such as ingredients and nutritional value of food by classifying the input of the image.

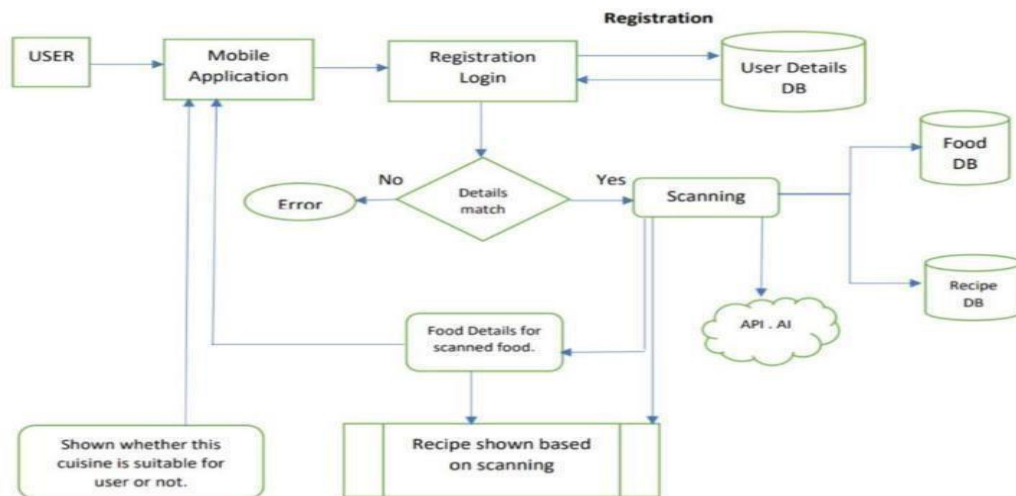
Our method employs **Clarifai's AI-Driven Food Detection Model** for food identification and Food API's to give the nutritional value of the identified food. API gets the calories of the food and those responses stored in DB2.

## 1.2 Purpose

WEB APP provides Nutritional values of image.(Nutritional values is a part of food quality is the measure of a well-balanced ratio of essential nutrients proteins, vitamins, minerals, fat, carbohydrates in item of food) those Nutritional values stored in Database.

Due to ignorance of healthy food habits, obesity rates are increasing. So, people must follow dietary plans to avoid future health problems. Our AI Driven Method analyzes the real-time images of a meal with the help of this application users are able to maintain healthy lifestyle.

This results in health among youth which can impact our future generation. The need of the hour is to understand which nutrients present in which foods.



**Figure 1.2.1 Technical Architecture**

## **CHAPTER 2**

### **LITERATURE SERVEY**

#### **2.1 Existing problem**

Healthy nutrition contributes to prevent Non communicable and diet related diseases. Recommender systems, as an integral part of mHealth technologies, address this task by supporting users with healthy food recommendations. However, knowledge about the effects of the long-term provision of health-aware recommendations in real-life situations is limited.

Our system offers automated personalized visual feedback and recommendations based on individual dietary behaviour, phenotype, and preferences. By using quantitative and qualitative measures of 34 participants during a study of 2-3 months, we provide a deeper understanding of how our nutrition application affects the user's physique, nutrition behaviour, system interactions and system perception.

Finally, we discuss general knowledge acquired on the design of personalized mobile nutrition recommendations by identifying important factors, such as the user's acceptance of the recommender's taste, health, and personalization.

#### **2.2 References**

##### **1. Predicting calorific value for mixed food using image processing**

**Author:** R. Kohila,R. Meenakumari

**Source:** Research Gate

## **Findings:**

- The image of the food is transmitted through a mobile device and it initially undergoes segmentation with Fuzzy C- means Clustering Segmentation which fixes the cluster center based on the group data unlike the K means Clustering which can be erroneous if the cluster center is not defined properly by the user.

## **2. Food Item Recognition and Intake Measurement Techniques**

**Author:**Adnan Shehzad, Nauman Zafar,Mir Hassan, Zhidong Shen

**Source:** Research Gate

## **Findings**

- The photograph then passes some preprocessing steps, and after successful segmentation, many physical features are examined such as shape and size etc.
- Also, dimensions of the food object are determined. The concluding step is then recognition along with calorie estimation
- The main aim of this review paper is to do a critical analysis of recent studies on accurate calorie estimation and food item recognition.

## **2.3 Problem Statement Definition**

Due to the ignorance of healthy food habits, Obesity rates are increasing at an alarming speed, and this is reflective of the risks to the people's health. People need to control the daily Calories intake by eating Healthier foods, which is the most basic method to avoid obesity.

However, although food packing comes with Nutrition (and Calories) labels, it's still not very convenient for people to refer to App-based Nutrient dashboard Systems which can analyze Real-time.

Images of a Meal and Analyze it for Nutritional content which can be very handy and improves the Dietary habits, and therefore, helps in maintaining a Healthy Lifestyle.

The main Objective of this project is to building a Web App that automatically estimates Food attributes such as Ingredients and Nutrition Value by classifying the input image of food .

## CHAPTER 3

# IDEATION & PROPOSED SOLUTION

### 3.1 Empathy Map Canvas

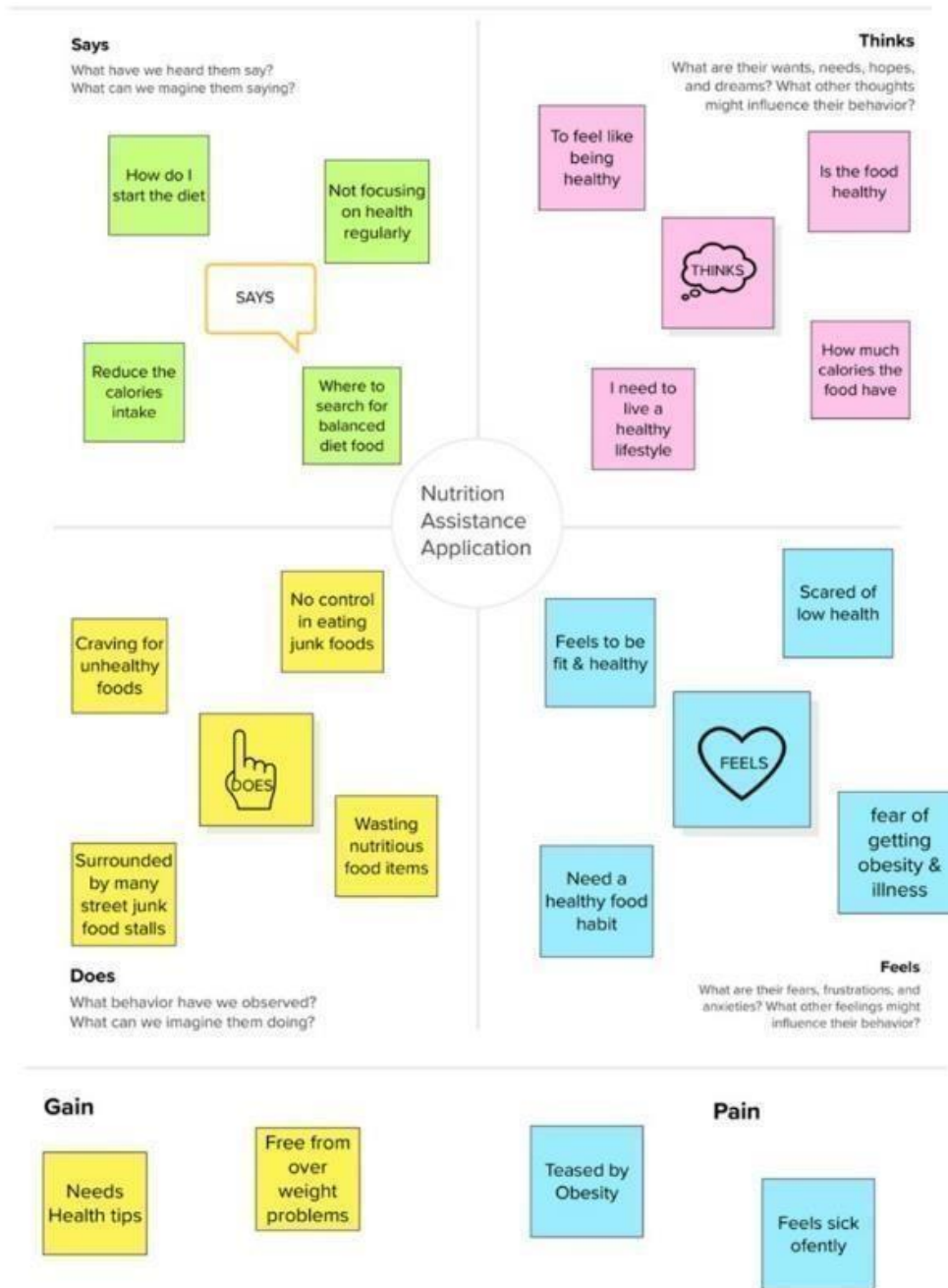


Figure 3.1.1 Empathy Map Canvas

## 3.2 Ideation&Brainstroming

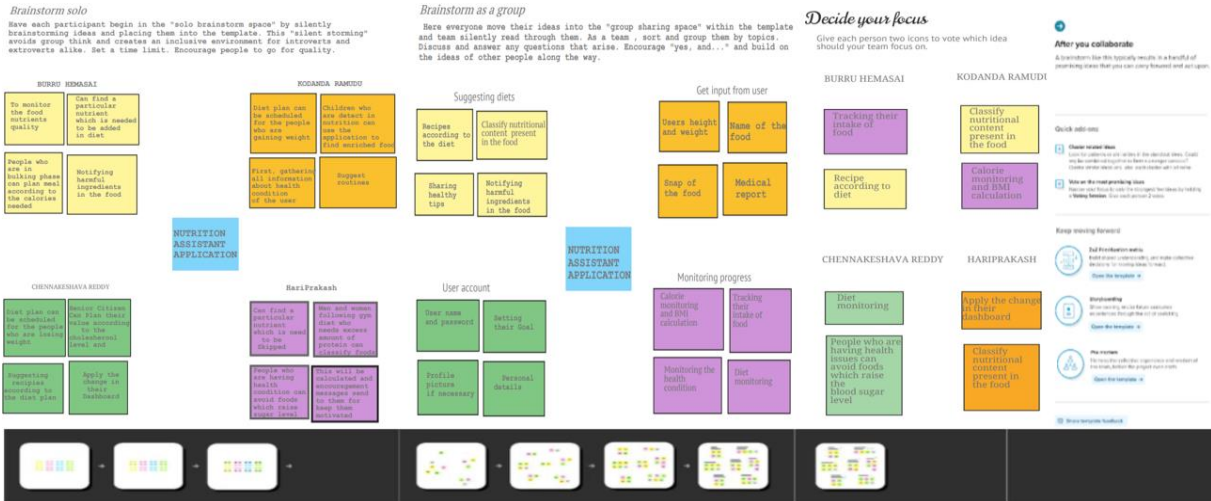


Figure 3.2.1 Ideation&Brainstroming

## 3.3 Proposed Solution

S.NO	Parameter	Description
1	Aim	Building a WEB APP, user load image to application nutritional values are stored in Database.
2	Problem Statement (problem to be solved)	App-based nutrient dashboard systems which can analyze real-time images of a meal and analyze it for nutritional content.
3	Idea / Solution description	The solution can be brought by using Clarifai's AI-Driven food detection model to obtain precise food identification and food APIs to give the nutritional value of the identified food.
4	Uniqueness	Providing a user-friendly environment to access the nutritional information about the food by Choosing from the provided list
5	Customer Satisfaction	By providing custom diet and meal plans to the user, getting user feedbacks for the product enhancement and longevity.

Table No: 3.1.1



## 3.4 Problem Solution Fit

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> <p>All age group people who are careless about their health due to their busy schedule and intake of high-calorie diet.</p>	<b>6. CUSTOMER CONSTRAINTS:</b> <span>■</span> <p>The customer should provide a clear image for knowing the nutrition content about the food. The app can't provide accurate result if the image is not clear. In some cases, the recipes may be allergic to their health.</p>	<b>5. AVAILABLE SOLUTIONS:</b> <span>■</span> <p>Although the food packaging comes with nutrition (and calorie) labels, it's still not very convenient for people to refer to App-based nutrient dashboard systems.</p>	Explore AS, differentiate
Focus on J&P, tap into BE, understand RC	<b>2. JOBS-TO-BE-DONE / PROBLEMS:</b> <span>J&amp;P</span> <p>The problem and pains of the user are obesity, fear of getting health related issues. They will get frustrated of not getting immediate result and difficult to do tedious work. Lack of confidence due to appearance.</p>	<b>9. PROBLEM ROOT CAUSE:</b> <span>RC</span> <p>It is easy to fall into a trap of eating unhealthy foods which is heavy in calories. Once the nutritional value is replaced by foods high in sugar, bad fats and salt it leads to various health issues so users need to control their daily calorie intake to lead a healthy lifestyle.</p>	<b>7. BEHAVIOUR:</b> <span>BE</span> <p>The behavioral changes in users reflect in their day-to-day life such as they will maintain a proper diet and follow the daily routine in eating and intake of healthy food. So, that it helps them to improve their health.</p>	Focus on J&P, tap into BE, understand RC
	<b>3. TRIGGERS:</b> <span>TR</span> <p>Desire to live a healthy lifestyle. By knowing the success story of people who achieved their goal. By seeing people who are fit and healthy.</p>	<b>10. YOUR SOLUTION:</b> <span>SL</span> <p>The solution is user can know the nutritional content of the food they are intaking, by taking picture of the food and uploading it in the app. Clarifai's AI-Driven Food Detection Model is used for getting accurate food identification and APIs to give the nutritional value of the identified food</p>	<b>8. CHANNELS of BEHAVIOUR:</b> <span>CH</span> <p><b>ONLINE:</b> The application provides a user-friendly environment that enables users to interact through chatbot to clarify their queries and a dashboard is displayed to know the activities</p> <p><b>OFFLINE:</b> Connecting all the users through offline meeting and giving some complimentary gifts. Conducting offline session by nutrition expert.</p>	
	<b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> <p>They scared of declining health, so they get motivated towards eating healthy foods and move to healthy lifestyle.</p>			

**Figure 3.4.1 Problem Solution Fit**

## CHAPTER 4

### REQUIREMENT ANALYSIS

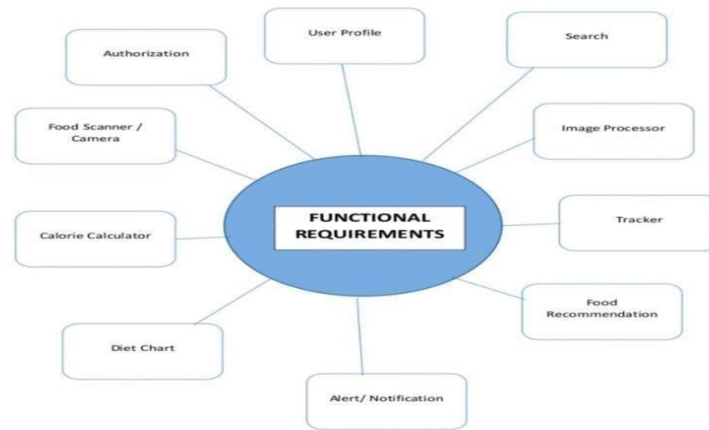
#### 4.1 Functional requirements

Functional requirements are **product features or functions that developers must implement to enable users to accomplish their tasks.**

**Table Name: Functional requirements**

S.NO	Functional requirements	Sub-Requirements
1	User Registration	-> Registration through Form -> Registration through Google -> Registration through Facebook
2	User Confirmation	-> Confirmation via Email -> Confirmation via OTP
3	User Login	-> Login with Username -> Login with Password
4	User Profile Update	-> Update user's name -> Update date of birth
5	Uploading Food Image	-> Upload from Gallery -> Capture using Camera
6	Enter Food Name	-> Type the name of the food -> Automatically it displays food name
7	Result	-> Download Results -> Share result through social media
8	Ratings and Reviews	-> Share the experiences -> Provide Feedback

**Table No: 4.1.1**



**Figure 4.1.1 Functional requirements**

## 4.2 Non-Functional Requirements

Non- Functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability.

**Table Name: Non-Functional Requirements**

S.NO	Non-Functional Requirements	
1	Usability	Provide user friendly UI simple and Intuitive design
2	Performance	The landing page supporting several users, must provide 5 sec or less response time
3	Scalability	Provide horizontal or vertical scaling for higher workloads
4	Availability	Uninterrupted services must be available all time except the time of server updation.
5	Security	Comprehensive authorization and authentication scheme for each system actor
6	Reliability	The system must perform without failure in 95% use cases

**Table No: 4.2.1**

## CHAPTER 5

### PROJECT DESIGN

#### 5.1. Data Flow Diagrams

A data flow diagram (DFD) maps out the flow of information for any process or system.

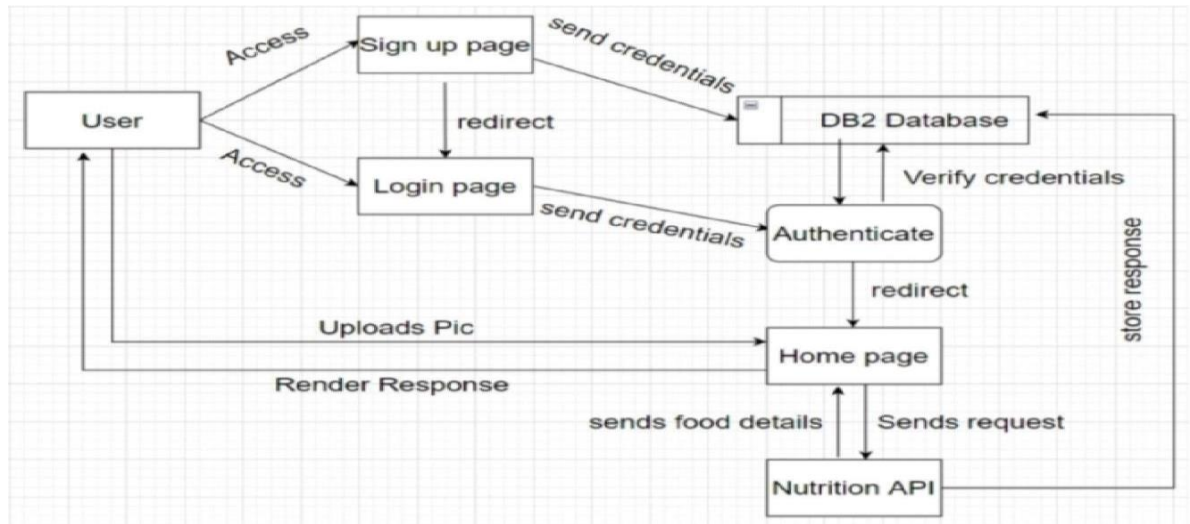


Figure 5.1.1 Data Flow Diagrams

#### 5.2 Solution & Technical Architecture

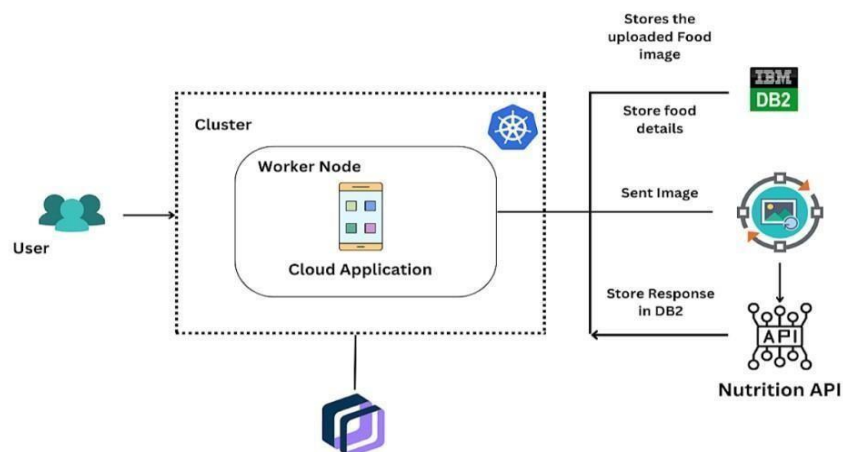


Figure 5.2.1 Solution & Technical Architecture

## 5.3 User Stories

SCENARIO	Entice	Enter	Engage
<p>Browsing, booking, attending, and rating a local city tour</p>	<p></p> <p><b>Entice</b></p> <p>How does someone initially become aware of this process?</p>	<p></p> <p><b>Enter</b></p> <p>What do people experience as they begin the process?</p>	<p></p> <p><b>Engage</b></p> <p>In the core moments in the process, what happens?</p>
<p> <b>Steps</b></p> <p>What does the person (or group) typically experience?</p>	<p><b>Knowledge through online advertisement</b></p> <p>A customer navigates to the page to take a class &amp; customer navigates to the page for taking image.</p> <p><b>To know the calorie values.</b></p> <p>people want to know the calorie value of the food they intake.</p>	<p><b>User friendly interface</b></p> <p>Customer can view the home page after login and upload the picture of the food.</p>	<p><b>Upload a image</b></p> <p>Customer wants to take a picture of the food and upload the picture to know the attributes in it.</p> <p><b>View the result.</b></p> <p>Customer can view the calorie value for the uploaded food image.</p> <p><b>Enjoy their diet.</b></p> <p>According to their BMI, customer can get a diet chart.</p>
<p> <b>Interactions</b></p> <p>What interactions do they have at each step along the way?</p> <p>» <b>People:</b> Who do they see or talk to?</p> <p>» <b>Places:</b> Where are they?</p> <p>» <b>Things:</b> What digital touchpoints or physical objects would they use?</p>	<p><b>Interaction with a web page.</b></p> <p><b>Interaction with browsers.</b></p>	<p><b>Interaction with the home page.</b></p> <p><b>Interaction with a UI Login page if they already registered.</b></p> <p><b>Interaction with a registration page if they are new user.</b></p>	<p><b>People interact with a interface to knowing about the food nutrition value nearby.</b></p> <p><b>Interacts with result page using the image uploadable user will being engage with the software.</b></p>
<p> <b>Goals &amp; motivations</b></p> <p>At each step, what is a person's primary goal or motivation? ("Help me..." or "Help me avoid...")</p>	<p><b>Help me preserve my physical wellbeing.</b></p> <p><b>Assist me with avoiding junk food.</b></p>	<p><b>Please help me to know the food's calorie count.</b></p> <p><b>Give me suggestion to maintain my diet.</b></p>	<p><b>Please assist me in learning the nutritional content of each meal.</b></p> <p><b>shows the caloric value of the uploaded picture.</b></p> <p><b>It gives diet suggestions according to the BMI given.</b></p> <p><b>Customer should follow the diet plan.</b></p>
<p> <b>Positive moments</b></p> <p>What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?</p>	<p><b>Customer feel more enjoyable and excited if it's free to use.</b></p>	<p><b>Customers are excited about the user friendly features.</b></p>	<p><b>The customer will be happy to maintain proper diet plan in their meal.</b></p> <p><b>Customer feel delighted to have a nutritional guidelines.</b></p> <p><b>User feels joyful to know the calorie value of the food they intake.</b></p>
<p> <b>Negative moments</b></p> <p>What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?</p>	<p><b>A Customer feels upset if a application charges to utilize.</b></p>	<p><b>Some users may confusing about using this application.</b></p>	<p><b>When the caloric value of the food is incorrect, it will be helpless to the user.</b></p> <p><b>If the value is inaccurate, it redirects the user's health.</b></p>
<p> <b>Areas of opportunity</b></p> <p>How might we make each step better? What ideas do we have? What have others suggested?</p>	<p><b>Easy to accessibility to all customer.</b></p> <p><b>Nutritional value estimate.</b></p>	<p><b>An user friendly interface.</b></p> <p><b>calorie counter estimation.</b></p> <p><b>Offer a food diary to let you track what you eat.</b></p>	<p><b>Give food nutrition value image processing.</b></p> <p><b>Provide customizable meal plan for an individual.</b></p> <p><b>Provide high nutrition food list.</b></p>

Figure 5.3.1 User Stories

## CHAPTER 6

### PROJECT PLANNING & SCHEDULING

#### 6.1 Sprint Planning & Estimation

Setting Up Application Environment To create lots of environment. Create or enrolment to the IBM Cloud, Docker CLI Installation, Create an account in SendGrid and Nutrition API etc., Implementing Web Application, Create a UI to interact with application.

Integrating SendGrid Service SendGrid integration with python code for include

Some RestAPI services for to give a Nutrition and calorie value. Development of App in IBM Cloud in the deploy process, the deployment in Kubernetes cluster is the major

Task before that we need to containerize the app and upload image to IBM container Registry.

#### 6.2 Sprint Delivery Schedule

**Table Name: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	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	16 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

**Table No: 6.2.1**



6.3 Reports from JIRA



Figure 6.3.1 Reports from JIRA

## CHAPTER 7

### CODING & SOLUTIONING

#### 7.1 Feature 1

**Information:** This is the mobile apps that allow you to search the content instantly anywhere, anytime. Apps are reported to be 1.5 times faster as compared to websites and perform actions at a faster pace too. Human minds are inquisitive and smartphone applications serve as an instant relief for this inquisitiveness.

#### **Instant Connection & Communication**

Mobile phones and the internet have made this world a small place. There are ample of apps that allow a person to instantly connect and communicate another person living miles away. Apps have played a vital role in helping us in maintaining a healthy work-life balance.

#### **Better Productivity**

According to a report released by Accenture, business employees acknowledge the fact that smartphone apps play a critical role in their smooth business operations. Earlier people can access professional emails and messages from office computers only.

Now with the help of various apps and an internet connection they can check their emails and messages from anywhere in the world. Even if you're holidaying with your family, you can stay connected with your business updates with just a simple tap. This will contribute to your productivity as well.



## 7.2 Feature 2

**Personal Accounts and Registration:** Personalization is the key feature of every health and nutrition app.

**Food Logging and Dashboard:** A top-notch health and nutrition app should have a duly designed dashboard and a food logging feature. You need to prepare a database consisting of all diet information in it like calories, food intake.

### Push Notifications

Push notifications hold a special relevance in smartphone apps. Install this feature to remind your users for their upcoming meal and motivate them to achieve their health and nutrition goals.

### DASHBOARD.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"> <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv
1WTRi" crossorigin="anonymous">
<script>
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle
.min.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V
8Qbsw3" crossorigin="anonymous"></script>
<link rel="stylesheet" href="{ {url_for('static',
filename='css/dashboard.css')}}">
<link rel="icon" href="{ { url_for( 'static', filename =
```

```

'src/cardiogram.png'))}>
<title>Welcome {{user}}</title>
</head>
<body>
{% if msg %}
<div class="msg bg-info" style="padding: 0px 0 0px 50px;margin: 20px
20px 0 20px;border-radius: 20px;">
<h4>{{msg}}</h4>
</div>{% endif %}
<div class="container-fluid dash">
<div class="header p-3">
<h3>&nbsp;&nbsp;&nbsp;Dashboard</h3>
<div style="display:flex; justify-content: right;align-
items:
center;">Welcome {{user}},&nbsp;&nbsp;&nbsp;<form action="" method="post"
enctype="multipart/form-data"><button type="submit"
name="logout"
class="combutton btns">Log Out</button></form></div>
</div>
</div>
<div class="container-fluid">
<div class="row rowh">
<div class="col-lg-4 colh">
<div class="maincon comcolor">
<h4>Control panal</h4>
<h5><form action="" name="deleteHistory" method="post"
enctype="multipart/form-data"><input style="width:
200px;border:0px;padding:10px 40px;border-radius:20px;" type="submit"
name="deleteHistory" value="Delete All History"></form></h5>
<table>
<tr>
<th>History</th>
</tr>
{% for row in history %}

```

```

<tr>
<td>{{
row['TIME']
}}</td>
<td><form
method="post"
action="{{url_for('homepage')}}" class="inline">
<input type="hidden"
name="extra_submit_param_view" value="{{row['NUTRITION']}}">
<button type="submit"
name="submit_param" value="submit_value" class="link-button">
View
</button>
</form>
</td>
<td><form method="post"
action="{{url_for('homepage')}}" class="inline">
<input type="hidden"
name="extra_submit_param_delete" value="{{row['TIME']}}"
<button type="submit"
name="submit_param" value="submit_value" class="link-button">
Delete
</button>
</form>
</td>
</tr>
{% endfor %}
</table>
</div>
</div>
<div class="col-lg-8 row colh">
<div class="row normsize">
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex-col">

<button class="combutton btns" onclick="setImage()"
>Clear Image</button>
</div>
</div>
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex lesssize normpadding">
<div>
<h1>Upload Image</h1>
<form action="{ {url_for('upload_file')}}" method="post"
enctype="multipart/form-data">
<input type="file
onchange="readURL(this);" name="file"> <input
style="margin: 10px 0px;"
onclick="setImage()" type="submit" value="Upload" name="upload">
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
{ % if data % }
</form>
</td>
</tr>
{ % endfor % }
</table>
</div>
</div>
<div class="col-lg-8 row colh">
<div class="row normsize">
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex-col">

```

```


<button class="combutton btns" onclick="setImage()"
>Clear Image</button>
</div>
</div>
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex lesssize normpadding">
<div>
<h1>Upload Image</h1>
<form action="{{url_for('upload_file')}}" method="post"
enctype="multipart/form-data"> nput type=file
</form>
</td>
</tr>
{% endfor %}
</table>
</div>
</div>
<div class="col-lg-8 row colh">
<div class="row normsize">
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex-col">

<button class="combutton btns" onclick="setImage()"
>Clear Image</button>
</div>
</div>
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex lesssize normpadding">
<div>
<h1>Upload Image</h1>

```

```

<form action="{ {url_for('upload_file') }}" method="post"
enctype="multipart/form-data">
<input type="file"
onchange="readURL(this);" name="file"> <input
style="margin: 10px 0px;"
onclick="setImage()" type="submit" value="Upload" name="upload">
</form>
</div>
</div>
</div>
</div>
</div>
</div>
{ % if data % }
</form>
</td>
</tr>
{ % endfor % }
</table>
</div>
</div>
<div class="col-lg-8 row colh">
<div class="row normsize">
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex-col">

<button class="combutton btns" onclick="setImage()"
>Clear Image</button>
</div>
</div>
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex lesssize normpadding">
<div>
<h1>Upload Image</h1>

```

```

<form action="{ {url_for('upload_file')}}" method="post"
enctype="multipart/form-data">
<input type="file
onchange="readURL(this);" name="file"> <input
style="margin: 10px 0px;"
onclick="setImage()" type="submit" value="Upload" name="upload">
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
{ % if data % }
</form>
</td>
</tr>
{ % endfor % }
</table>
</div>
</div>
<div class="col-lg-8 row colh">
<div class="row normsize">
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex-col">

<button class="combutton btns" onclick="setImage()"
>Clear Image</button>
</div>
</div>
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex lesssize normpadding">
<div>

```

```

<h1>Upload Image</h1>
<form action="{ {url_for('upload_file') }}" method="post"
enctype="multipart/form-data">
<input type=file
onchange="readURL(this);" name="file"> <input
style="margin: 10px 0px;"
onclick="setImage()" type=submit value=Upload name="upload">
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
{ % if data % }
</form>
</td>
</tr>
{ % endfor % }
</table>
</div>
</div>
<div class="col-lg-8 row colh">
<div class="row normsize">
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex-col">

<button class="combutton btns" onclick="setImage()"
>Clear Image</button>
</div>
</div>
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex lesssize normpadding">

```



```

<div>
<h1>Upload Image</h1>
<form action="{ {url_for('upload_file')}}" method="post"
enctype="multipart/form-data">
<input type=file
onchange="readURL(this);" name="file"> <input
style="margin: 10px 0px;"
onclick="setImage()" type=submit value=Upload name="upload">
</form>
</div>
</div>
</form>
</td>
</tr>
{ % endfor % }
</table>
</div>
</div>
<div class="col-lg-8 row colh">
<div class="row normsize">
<div class="col-lg normsize roudcorner comcolor">
<div class="complex-col">

<button class="combutton btns
</form>
</td>
</tr>
{ % endfor % }
</table>
</div>
</div>
<div class="col-lg-8 row colh">
<div class="row normsize">
<div class="col-lg normsize roudcorner comcolor">

```

```

<div class="comflex-col">

<button class="combutton btns" onclick="setImage()"
>Clear Image</button>
</div>
</div>
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex lesssize normpadding">
<div>
<h1>Upload Image</h1>
<form action="{{url_for('upload_file')}}" method="post"
enctype="multipart/form-data">
<input type="file"
onchange="readURL(this);" name="file"> <input
style="margin: 10px 0px;"
onclick="setImage()" type="submit" value="Upload" name="upload">
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<m>
</td>
</tr>
{% endfor %}
</table>
</div>
</div>
<div class="col-lg-8 row colh">
<div class="row normsize">
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex-col">

```

```


<button class="combutton btns" onclick="setImage()"
>Clear Image</button>
</div>
</div>
<div class="col-lg normsize roudcorner comcolor">
<div class="comflex lesssize normpadding">
<div>
<h1>Upload Image</h1>
<form action="{ {url_for('upload_file') }}" method="post"
enctype="multipart/form-data">
<input type=file
onchange="readURL(this);" name="file"> <input
style="margin: 10px 0px;"
onclick="setImage()" type=submit value=Upload name="upload">
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
{ % if data % }
<div class="container-fluid float">
<div class="containers floatcontainer ">
<div class="box1">
<div class="close">
<a href="{ {url_for('upload_file',methods='POST') }}"
class="closes"></a>
</div>
</div>
<div style="background-color: rgb(105, 102, 102);margin-top:
25px;font-size: 30px;font-weight: bold;padding-left: 15px;"><p>Nutrition

```

```

Facts</p></div>
<div class="box2">
<div class="bcol">
<table style="width:100%;">
<tr>
<th>Calories</th>
<th>{{ data[0] }} {{ unit[0] }}</th>
</tr>
<tr>
<th></th>
<th>Daily Value</th>
</tr>
<tr>
<th>Total Fat</th>
<th>{{ data[1] }} {{ unit[1] }}</th>
</tr>
<tr>
<td>Saturated Fat</td>
<td>{{ data[2] }} {{ unit[2] }}</td>
</tr>
<tr>
<td>Polyunsaturated Fat</td>
<td>{{ data[3] }} {{ unit[3] }}</td>
</tr>
<tr>
<td>Monounsaturated Fat</td>
<td>{{ data[4] }} {{ unit[4] }}</td>
</tr>
<tr>
<th>Cholesterol</th>
<th>{{ data[5] }} {{ unit[5] }}</th>
</tr>
<tr>
<th>Sodium</th>
<th>{{ data[6] }} {{ unit[6] }}</th>
</tr>

```

```

<tr>
<th>Potassium</th>
<th>{{ data[7] }} {{ unit[7] }}</th>
</tr>
<tr>
<th>Sugar</th>
<th>{{ data[8] }} {{ unit[8] }}</th>
</tr>
<tr>
<th>Protein</th>
<th>{{ data[9] }} {{ unit[9] }}</th>
</tr>
<tr>
<th>Carbohydrates</th>
<th>{{ data[10] }} {{ unit[10] }}</th>
</tr>
<tr>
<th>Vitamin A</th>
<th>{{ data[11] }} {{ unit[11] }}</th>
</tr>
<tr>
<th>Vitamin C</th>
<th>{{ data[12] }} {{ unit[12] }}</th>
</tr>
<tr>
<th>Vitamin D</th>
<th>{{ data[13] }} {{ unit[13] }}</th>
</tr>
<tr>
<th>Vitamin B5</th>
<th>{{ data[14] }} {{ unit[14] }}</th>
</tr>
<tr>
<th>Calcium</th>
<th>{{ data[15] }} {{ unit[15] }}</th>
</tr>

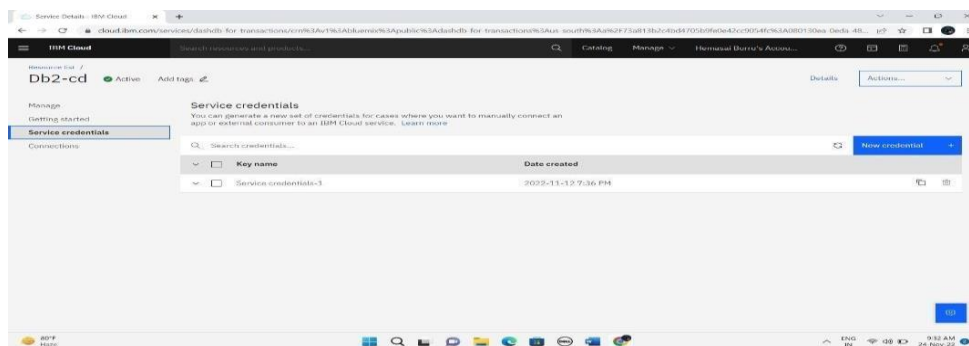
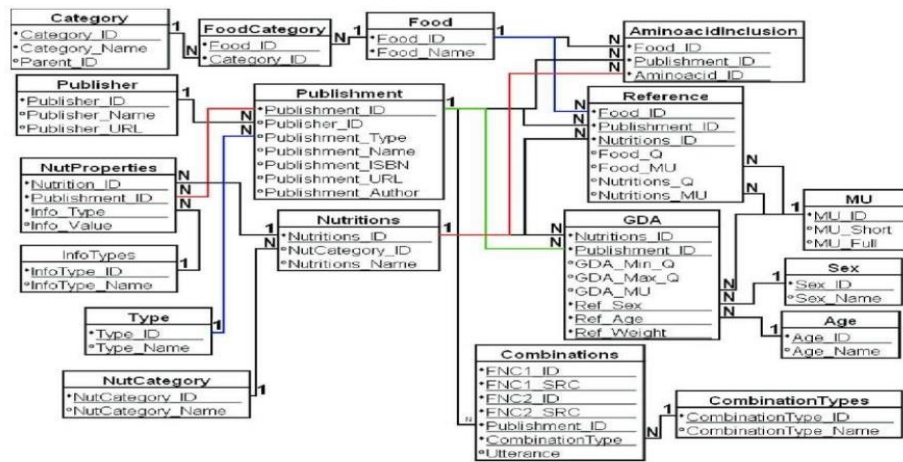
```

```

</table>
</div>
</div>
</div>
</dic>
{% endif %}
<script>
//image = document.getElementById('myImage');
function clearImage() {
image.src = "{ {url_for('static',filename='src/user.jpg')}}";
//onclick="document.getElementById('myImage').src='src/omplate.png'"
}
function setImage() {
image.src = "{ {url_for('static',filename='src/food.jpg')}}";
}
</script>
<script src="{ {url_for('static', filename='js/dashboard.js')}}">
</script>
</body>
</html>

```

## 7.3 Database Schema



### Figure 7.3.1 Database Schema

## CHAPTER 8

### TESTING

#### 8.1 Test Cases

**Table Name: Test Scenario**

	<b>Test Scenario</b>
<b>1</b>	Verify if the user is able to open and view the homepage
<b>2</b>	Verify if the user is able to interact with the elements in the homepage
<b>3</b>	Verify if the user is able to navigate to the other pages of the application from the homepage
	<b>Upload Image Page Actions</b>
<b>1</b>	User is able to upload image
<b>2</b>	User is able to submit the image and obtain results
	<b>View History of Items Related Actions</b>
<b>1</b>	User is able to view all past uploaded images
<b>2</b>	User is able to see the nutritional breakdown of the previously uploaded images
	<b>User is able to log in and sign up</b>
<b>1</b>	User is able to create an account and log in

**Table No: 8.1.1**



To log meals, users can input the food in the app, or scan the barcode of a package. In order to build a realistic initial target group, I wanted to conduct some usability tests with 5 users that regularly engage in physical activity and food tracking, including both first-time and regular users of meal planning and fitness apps.

I asked these individuals to perform tasks related to general usage of the Nutrition Coach apps (such as food logging, searching, and checking their caloric breakdown.)

## **8.2 User Acceptance Testing**

Must-have features of a diet and nutrition app I wanted to address the user pain points by including (and improving) the core features of the application. Personal profiles After downloading the app, a user needs to register and create an account.

At this stage, users should fill in personal information like name, gender, age, height, weight, food preferences, allergies, and level of physical activity. Food logging and dashboard Allowing users to analyze their eating habits. They should be able to log food and water intake and see their progress on a dashboard that can track calories, fat, protein, and carbs.

To motivate users to keep moving toward their goals, it's pertinent to deliver information on their progress toward the current goal and remind them to log what they eat. A casual dieter who does not follow a health plan regularly

Student who needs reminders, suggestions, and coaching to cook more often with fresh ingredients because they want to stay on top of their health and make it a part of their routine.

**Table Name :User AcceptanceTesting**

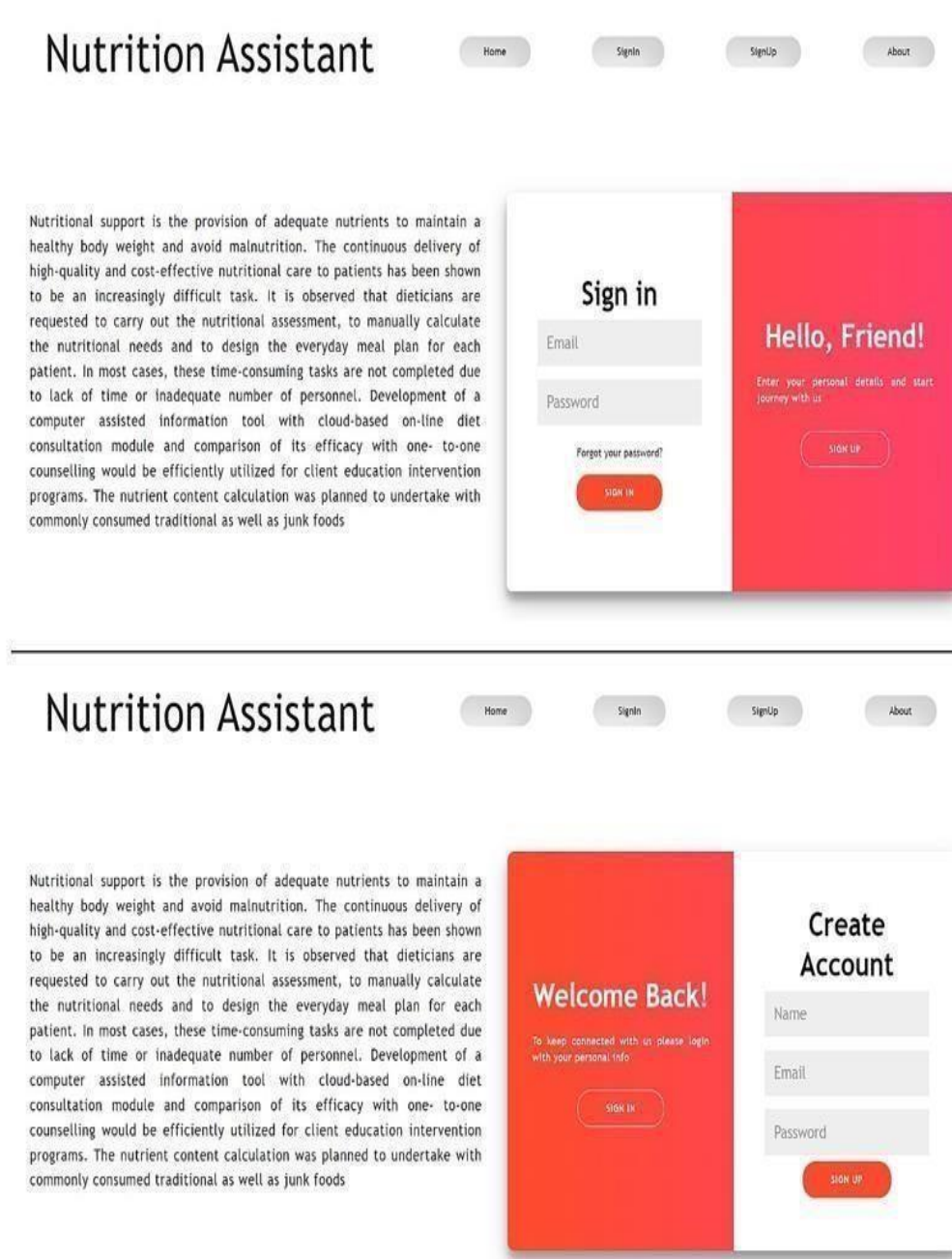
Should alert the user for their daily plan	A alarm must be set off	Passed
Should create log of their diet plan	A text file should be created with previous diet plans	Passed
Should suggest new plans	Notification rolling for important messages	Passed
During the completion of successful diet plan a new thanking window should open	A new popup should display- saying u have completed your diet plans	Passed
Nutrition amount display	Total nutrients consumed must be displayed	Passed
Logout display	Should successfully logoff the credentials	Passed

**Table No:8.2.1**

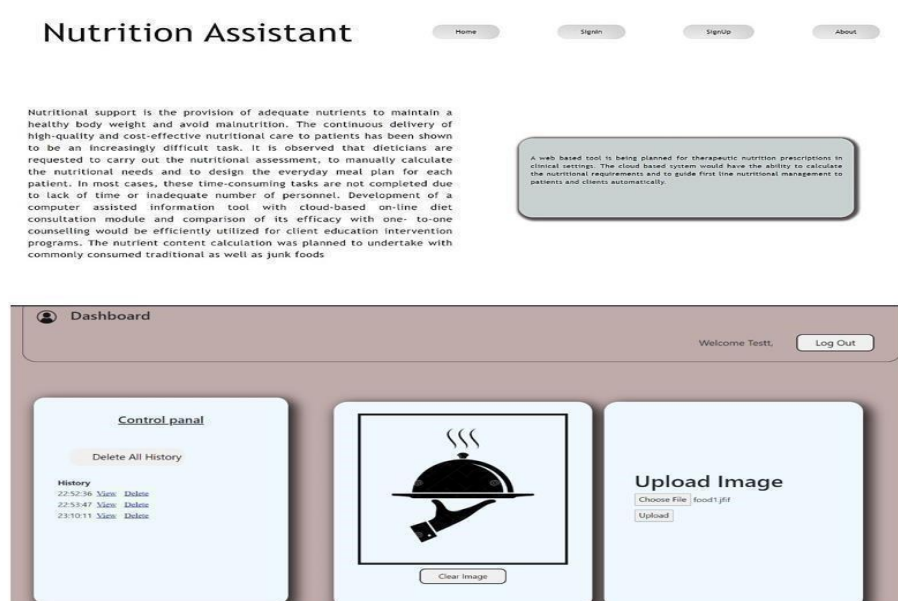
## CHAPTER 9

# RESULTS

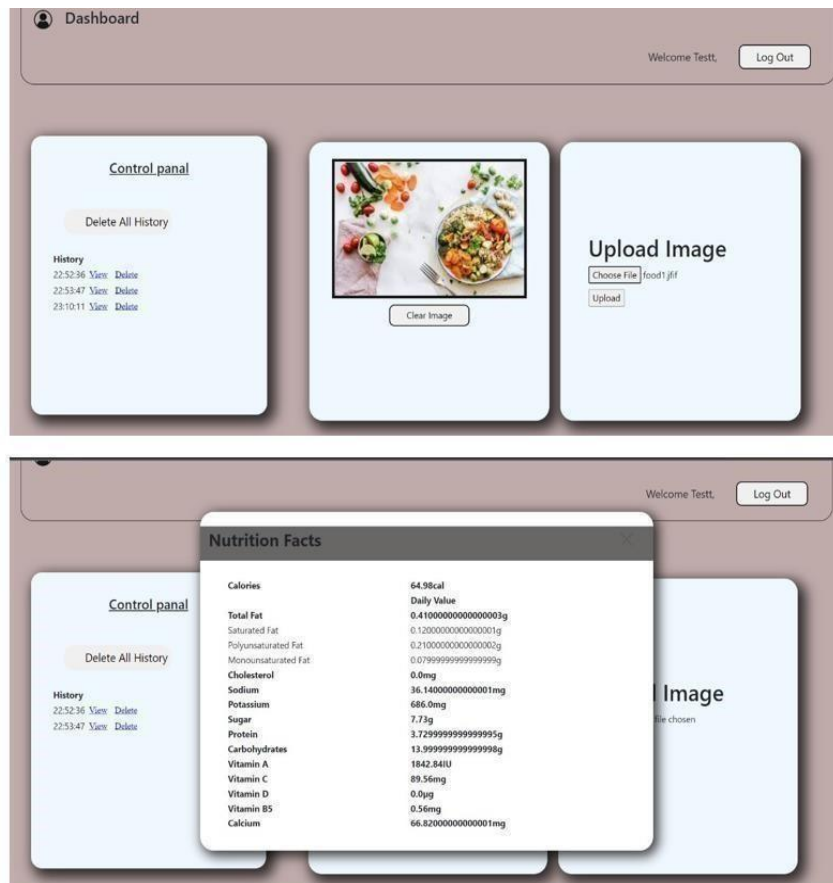
### 9.1 Performance Metrics



**Figure 9.1.1 Performance Metrics**



**Figure 9.1.2 Performance Metrics**



**Figure 9.1.3 Performance Metrics**

## **CHAPTER 10**

### **ADVANTAGES & DISADVANTAGES**

#### **Advantages**

This device is user friendly.

Its only required the image of the food.

To know the different type of nutrients present in food And also know that how much composition of the nutrients are present.

The user is now able to track his daily calorie intake.

#### **Disadvantages**

It cannot be used without an Internet Connection.

This device is not able to predict the multiple image as input.

## **CHAPTER 11**

### **CONCLUSION**

This application allows people to get to know the nutrients of foods at any time which makes it more convenient for the users. This can be scaled to include APIs that have a larger variety of foods to have it cater to larger audiences of different backgrounds and ethnicities. When choosing the right foods for yourself you should be focused on what is the healthiest choice.

Eating healthy and feeling good go hand in hand, eating better will automatically give you a better functioning body. During this assignment we were able to take a closer look at our daily eating habits. I am now more educated on the powers of food and how they control our body. I hope that people will use our application to lead a healthy life.

## **CHAPTER 12**

### **FUTURE SCOPE**

#### **1. ADDING GRAPHICAL DATA ON THE FOODS CONSUMES**

Adding a pie chart or a breakdown of what nutritional components are being consumed can give more insight into the food habits of a user. This can help the user to make changes and increase or decrease their consumption of a particular nutrient or food.

#### **2. CREATING A PERSONALIZED FOOD RECOMMENDATION SYSTEM**

Based on the previously uploaded images we can provide recommend for these kinds of foods to eat to have a balanced diet. Future goals include increasing the accuracy of our machine learning model.

Our research essentially identifies simply the nutrients, but our team members raise the bar for our project so that we also understand the ingredients and the amount of nutrients in a particular cuisine

## CHAPTER 13

### APPENDIX

#### Source code

```
from flask import Flask, render_template,
    Response, request import cv2 import datetime, time
import os, sys import numpy as np from threading
import Thread ## csv code import pandas as pd
read_file = pd.read_excel
("C:\\Users\\hemasai\\Desktop\\IBM2\\Book.xlsx")
read_file.to_csv ("Test.csv", index = None,
header=True) df =
pd.DataFrame(pd.read_csv("Test.csv"))
df.to_csv("Test.csv") df=df.set_index("Food Name")
def Nutrients(Name): name=Name
return(df.loc[(name),:]) 27 ## global capture,rec_frame,
grey, switch, neg, face, rec, out,p,d capture=0 grey=0
neg=0 face=0 switch=1 rec=0 # ML import keras
import cv2 import tensorflow as tf #import PIL.Image
load_img, img_to_array from
keras_preprocessing.image import.v2 as tf from
keras.models import load_model model
=keras.models.load_model('C:\\Users\\vimala\\Desktop\\
\\IBM2\\Daiyan.h5') import numpy as np ## import
numpy as np CATEGORIES = ['Vegetable-Fruit', 'Egg',
'Bread', 'Soup', 'Seafood', 'Meat', 'vada pav', 'Fried food',
'pizza', 'Dessert', 'Dairy product', 'Rice', 'burger',
```



```

'NoodlesPasta'] def image(path): img =
cv2.imread(path, cv2.IMREAD_GRAYSCALE)
new_arr = cv2.resize(img, (60, 60)) new_arr =
np.array(new_arr) new_arr = new_arr.reshape(-1, 60,
60, 1) return new_arr ## #make shots directory to save
pics try: os.mkdir('./shots') except OSError as error: 28
pass #instantiate flask app app = Flask(name ,
template_folder='./templates') camera =
cv2.VideoCapture(0) # def Path(d): # a=d # return a def
gen_frames(): # generate frame by frame from camera
global out, capture,rec_frame,d while True: success,
frame = camera.read() if success: if(capture): capture=0
now = datetime.datetime.now() p =
os.path.sep.join(['shots',
"shot_{ }.png".format(str(now).replace(":", ""))]
#d=("C:\\Users\\anish\\Desktop\\IBM2\\"+p)
cv2.imwrite(p, frame) d=p try: ret, buffer =
cv2.imencode('.jpg', cv2.flip(frame,1)) frame =
buffer.tobytes() yield (b'--frame\r\n' b'Content-Type:
image/jpeg\r\n\r\n' + frame + b'\r\n') except Exception
as e: pass else: pass @app.route('/') def index(): return
render_template('index.html') @app.route('/uplod') def
uplod(): return render_template('index.html')
@app.route('/video_feed') def video_feed(): return
Response(gen_frames(), mimetype='multipart/x-mixed-
replace; boundary=frame')
@app.route('/requests',methods=['POST','GET']) def

```

```

tasks(): global switch,camera if request.method ==
'POST': 29 ifrequest.form.get('click')== 'Capture':
global capture capture=1 elif request.form.get('detect')
== 'Detect': # prediction
=model.predict([image("C:\\Users\\hemasai\\Desktop\\I
BM2\\download.jfif"))] path = os.getcwd() print(d)
p=os.path.join(path, "", d ) prediction =
model.predict([image(p)])
name=(CATEGORIES[prediction.argmax()])
Product_name=name data=Nutrients(Product_name)
return
render_template('Prelect.html',name=name,data=data)
elif request.form.get('stop') == 'Stop/Start':
if(switch==1): switch=0 camera.release()
cv2.destroyAllWindows() else: camera =
cv2.VideoCapture(0) switch=1 elif
request.method=='GET': return
render_template('index.html') return
render_template('index.html') if name == ' main ':
app.run() camera.release() cv2.destroyAllWindows()

```

## **GITHUB LINK**

<https://github.com/IBM-EPBL/IBM-Project-53216-1661319134>

## **Project Demo link**

<https://www.mediafire.com/file/3q95xo38z5w98bk/Demo+Video.mp4/file>