

NUTRITION ASSISTANT APPLICATION

A PROJECT REPORT

SUBMITTED BY

TEAM ID : PNT2022TMID45161

TEAM LEADER : RAMYA P

TEAM MEMBER : ALRIJWANA R

TEAM MEMBER : ASVETHASRI R

TEAM MEMBER : BOOMATHI S

TEAM MEMBER : JANANI V

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

M.A.M COLLEGE OF ENGINEERING AND TECHNOLOGY

TRICHIRAPPALLI

ANNA UNIVERSITY::CHENNAI 600 025

NOV 2022

Index

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

1. INTRODUCTION

Food is one of the most basic requirements of human life. It is often regarded as much more than a means of survival, and proper food intake is essential for human health and fitness. Our health is closely dependent on the 4 types or amount of food we intake .There are numerous fields such as sociology, psychology, nutrition sciences, and medicine in which healthy food consumption is explored .Food choices are negatively influenced by a busy lifestyle, bad habits, and low self-control . However, excessively unhealthy lifestyles and bad dietary habits, such as increased food intake with high energy and high fat, lead to various health issues. According to the World Health Organization (WHO), more than 1.9 billion adults (aged over 18) are overweight, and more than 650 million people suffer from obesity. Many chronic diseases such as hypertension, type 2 diabetes mellitus, cardiovascular disease, and stroke are linked to obesity and excess weight. This problem is becoming a significant health concern. One of the main reasons for the obesity problem is that many people follow a very unhealthy lifestyle. Their dietary habits are also unhealthy, such as increased food intake with high energy and high fat. The intake of highly caloric, inexpensive, larger portion sizes and nutrient-dense foods promoted by environmental changes, coupled with decreased physical activity, and increased sedentary behaviors, is a significant causative factor for obesity.

In recent years, the use of smartphones to track food consumption or compute the nutritional value of food's has expanded due to the increasing number of food consumption tracking and recommendation apps in the app stores, and the great potential of smartphone's to be a useful tool. Nowadays in app stores, many apps are focused on health and fitness. In the major app stores, there were 32500 mobile health apps available in 2017 and this number is continuing to rise. Apps can play an important role in simplifying the tracking of health-related behaviors and weight management. Moreover, the usage of smartphones and rapid development of artificial intelligence (AI) technologies have enabled new food identification systems for dietary assessment, which are significant for the prevention and treatment of chronic diseases such as type 2 diabetes mellitus, cardiovascular disease, and overcoming health issues such as obesity . Furthermore, food intake behaviour (e.g., assessment of calorie intake, nutritional analysis, and eating habits) can be analyzed if food items or categories are recognized. Recently, AI and machine learning based mobile food recognition methods are also being implemented. For example, He et al. (2014) used AI techniques for identifying food from an image. The bag of visual words model (BoW) has been used for representing food images as visual words distributions and the support vector machine (SVM) model has been used to classify (Farinella et al., 2014). Furthermore, Anthimopoulos et al. (2014) used SVM, artificial neural Network and random forest classifications on 5000 food images organized into 11 classes described in terms of different bag-of-features. The convolutional neural network (CNN) is also used in some studies (Christodoulidis et al., 2015; Kawano & Yanai, 2014). Ming et al. (2018) proposed a photo-based dietary tracking system that employed deep-based image recognition algorithms to recognize food and analyze nutrition. For estimating an individual's food and calorie intake, the calculation of food portion size or volume is necessary. In several studies, different types of methods (i.e., single image-based or multiple image-based) have been used for estimating food volume from food images. To achieve quantitative food intake estimation, researchers combined visual recognition and 3D reconstruction in a study.

Both Android smartphone and web-based applications are implemented to recognize food and estimate the calorific and nutritional content of foods automatically without any user input. Food recommendation is a significant domain for people as well as society. Incorporating health into recommendations is mostly a recent concern proposed integrating deep neural network with a recommendation system focusing on Thai food. It not only considers users' food choices but also pays attention to users' health. Based on individual customer behaviors, tastes, and eating history, the system will assist consumers in making food selection decisions. Besides, a food recommendation system has been built to recommend food to diabetic patients based on nutrition and food characteristics.

Reviews on various health-related apps have been conducted in many different studies. A prior study reviewed diet tracking apps common in the Apple App Store and Google Play Store analyzed the main features of the most popular nutrition apps and compared their strategies and technologies for dietary assessment and user feedback. Another study reviewed nutritional tracking mobile applications specifically for diabetes patients (Darby et al., 2016). Rivera et al. (2016) characterized the use of evidence-based methods, the participation of health care experts, and the clinical assessment of commercial smartphone applications for weight loss or weight control. In this study, we evaluated the apps from three commercial app stores – Google Play, Apple App Store, and the Microsoft Store – to evaluate food consumption tracking and recommendation apps for all users, not just diabetes patients, pregnant women, or children. To the best of our knowledge, no research has thoroughly examined the current commercial mobile app market landscape to analyze and scientifically evaluate apps linked to food consumption tracking and recommendations. The speedy growth of such apps in the app stores, and the fast acceptance of these apps by the general population necessitates an assessment of this rapidly expanding market.

In this study, we have conducted a critical review of food consumption tracking and recommendation apps accessible in the three major commercial app stores (i.e., Google Play Store, Apple App Store, and Microsoft Store). We found a total of 473 apps in our initial search; after excluding the apps based on our exclusion criteria, we finally selected 80 apps for our study. The rating tool and the rating quality of raters are examined through internal consistency, and inter- and intra-rater reliability, respectively. We also analysed the user comments from app stores to better understand users' expectations and perspectives. We also discuss the limitations of the reviewed apps and potential design considerations from the perspectives of both developers and researchers.

1.1 Project Overview

It can be very helpful and improve eating habits to develop app-based nutrient dashboard systems that can evaluate real-time photographs of meals and assess them for nutritional content. The health tracking platform must, like any other nutrition app, have a specific capability set as well as a number of fundamental elements that assist users in bettering their physical condition and set it apart from other apps currently on the market. Diet services can provide more than just calorie counting, food intake monitoring, and physical activity tracking. In addition, it offers food diaries, diet tracking, a health activity tracker, and diet plans for pregnancy, bodybuilding, and veganism. Even if the main goal was to design an app for a diet plan with proper nutrition, the platform must be adaptable to future changes and the addition of new features. The advancement of artificial intelligence (AI) and

the significant growth in the use of food consumption tracking and recommendation-related apps in the app stores have created a need for an evaluation system, as minimal information is available about the evidence-based quality and technological advancement of these apps. Electronic searches were conducted across three major app stores and the selected apps were evaluated by three independent raters. A total of 473 apps were found and 80 of them were selected for review based on inclusion and exclusion criteria. An app rating tool is devised to evaluate the selected apps. Our rating tool assesses the apps' essential features, AI based advanced functionalities and software quality characteristics required for food consumption tracking and recommendations, as well as their usefulness to general users. The rating tool's internal consistency, as well as inter- and intra-rater reliability among raters, are also calculated. Users' comments from the app stores are collected and evaluated to better understand their expectations and perspectives. Following an evaluation of the assessed applications, design considerations that emphasize automation-based approaches using artificial intelligence are proposed. According to our assessment, most mobile apps in the app stores do not satisfy the overall requirements for tracking food consumption and recommendations. "Foodvisor" is the only app that can automatically recognise food items, and compute the recommended volume and nutritional information of that food item. However, these features need to be improvised in the food consumption tracking and recommendation apps. This study provides both researchers and developers with an insight into current state-of-the-art apps and design guidelines with necessary information on essential features and software quality characteristics for designing and developing a better app.

1.2 Purpose

The overall purpose of the project is to improve food security and nutrition programming by supporting USAID/BHR/FFP and its Cooperating Sponsors. The aim of the intermediate result is to strengthen the capacity of stakeholders who analyze food and nutrition security problems, and who design projects and report their progress. Food aid programs are important because they provide nutrition directly to people, and because food aid programs are designed to complement other health and nutrition activities and common people can access this through web and its for knowing the nutrition value by scanning the food Nutrition assistants help dieticians with providing proper nutrition at healthcare facilities. They determine patients' nutritional needs, assess risk factors, and plan meals and menus. They also ensure proper sterilization of plates and utensils .To provide adequate knowledge and skills necessary for critical thinking regarding diet and health so the individual can make healthy food choices from an increasingly complex food supply. To assist the individual to identify resources for continuing access to sound food and nutrition information. A healthy diet throughout life promotes healthy pregnancy outcomes, supports normal growth, development and ageing, helps to maintain a healthy body weight, and reduces the risk of chronic disease leading to overall health and well-being.

2. LITERATURE SURVEY

2.1 Existing problem

2.1.1 Automated food recognition system that provides dietary intervention based on computer vision and machine learning

The unique feature of the system relies in the realization of real-time energy balance with the help of network simulation. Food recognition deals with the challenges in image segmentation, classification and the volume- nutrient estimation. Food segmentation is done on the food image using Otsu's segmentation. Feature extraction is done using Local Binary Patterns (LBP), colour, texture and Scale-Invariant Feature Transform (SFIT). Classification is done using SVM, Bag of Features and K Nearest Neighbours. Weight estimation is done by mapping the nutritional facts from the USDA dataset and then followed by metabolic network modelling

2.1.2 NutriTrack: Android-based food recognition app for nutrition Awareness

The researcher developed an Android-based food recognition application used as a health awareness by allowing the user to photograph food and view its nutritional content. Users are informed of their required calorie intake when the system implements the Mifflin-St Jeor method to determine daily calorie consumption. Furthermore, the researchers investigated its impact on people's health awareness regarding food nutrition using randomly selected respondents.

2.2 References

Nutrition-Reference-Center-

https://play.google.com/store/apps/details?id=com.ebsco.nurc&hl=en_IN&gl=US

Few-shot food recognition via multi-view representation learning. ACM Trans. Multimedia Comput. Commun. Appl., 16. doi:10.1145/3391624.

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 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 analyse real-time images of a meal and analyse 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 nutritional value by classifying the input image of food.

Problem statement(PS)	I am (Customer)	I'm trying to	But	Because	Which mekes me feel
PS -1	STUDENT	Find a balanced nutrition diet to loss weight	There is no balanced diet available without workout	I have no time to do workout	A best nutritional based diet plan with less workout
PS -2	ATHLETE	Choose a best nutrition plan and workout technique to increase my sprinting speed	Confused with many techniques	I want to increase my sprinting speed very much before than ever	Prefect suggestions
PS -3	PREGNANT WOWAN	Choose a yoga and healthy nutrition diet for the normal pregnancy delivery	I am not familiar with yoga and diets	I don't have any idea about it	User friendly application to choose the beginner based type of yoga,exercises, and nutrition base diet plan
PS-4	USER	Know the food nutrients	I can't predict the nutrients in food	Doesn't have an efficient system	To have a efficient systems which predict the nutrients of the food

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

		<p>load to obesity.</p> <ul style="list-style-type: none"> • To avoid that the system will detect and recognize the food and evaluate the nutrient values for certain foods. • People need to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity. • However, some food packaging has an added nutrition and calorie values, but it's not very comfortable to refer. • In this system, we applied CNN algorithm to the task of food detection and recognition through parameter optimization.
2	Idea / Solution description	<ul style="list-style-type: none"> • By scanning real-time photos of a product and examining its nutritional composition, people can quickly measure their calorie and nutrient intake, which will help them maintain healthier eating habits. Healthy eating can help avoid disease. The users of this software will receive sufficient nourishment, support

		for keeping a healthy lifestyle, and suggested diet programmes
3	Novelty / Uniqueness	<ul style="list-style-type: none"> • Clustering the peoples based on their BMI value. . A web app that can automatically estimate food attributes such as ingredients and nutrition value by classifying the input image.
4	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> • The application which brings the awareness about the obesity in between the peoples. • The Obesity rate will get reduced and people will be able to lead a healthy life. It helps to achieve and maintain a healthy weight balancing in their routine life.
5	Business Model (Revenue Model)	<ul style="list-style-type: none"> • In market, this application gives a benefit across the people health wise and economical wise. • The greatest strategy to develop this application is through social media. The use of this application will boost public confidence. It is very user-friendly, incredibly convenient, and also has a subscription if a user reaches certain services.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement(Epic)	Sub Requirement (story/sub-task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User profile completion	Get personal details like weight,height,ect
FR-4	Gathering food	Upload photo or take live photo of the meal
FR-5	Scanning and retrieving	Scanning the real time image of food. Retrieves the food details from the stored data.
FR-6	Display calorie information	Integrate and get name of the food integrate nutrition API (rapid API) to collect calorie info.

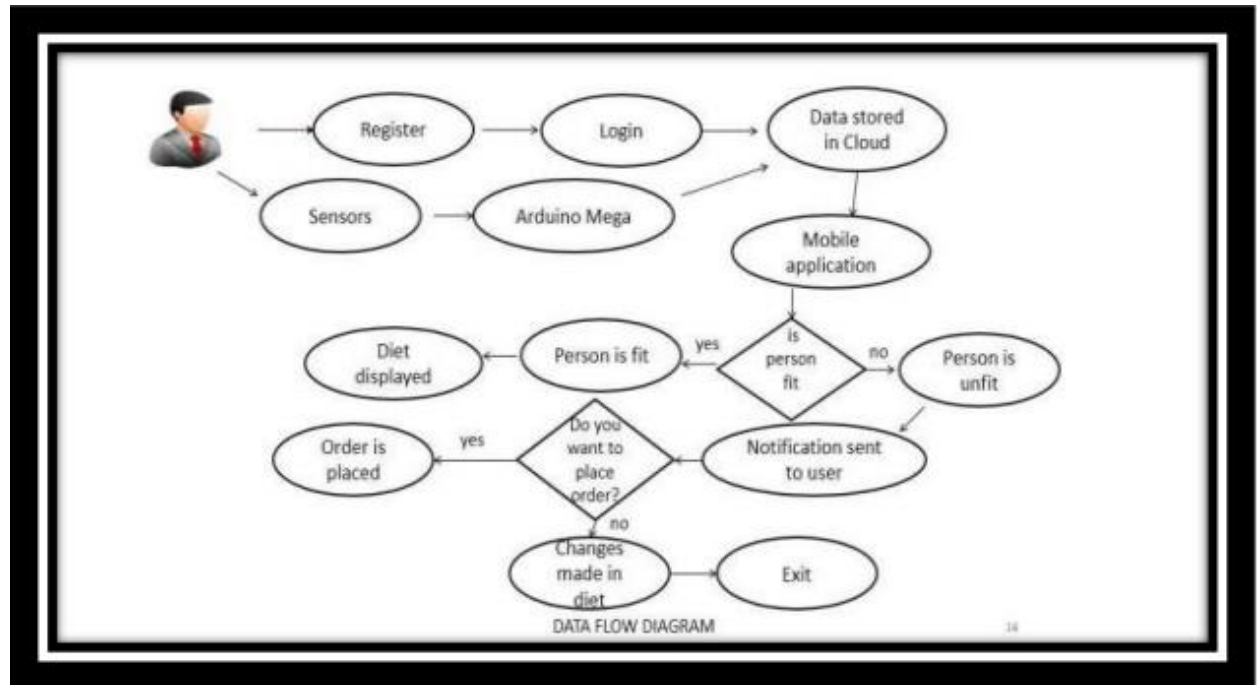
4.2 Non-Functional requirements

FR NO.	Non-Functional Requirement	Description
NFR-1	Usability	Provide user friendly UI Simple and intuitive design
NFR-2	Security	It has authorization and authentication
NFR-3	Reliability	The system most perform without failure
NFR-4	Performance	The landing page supporting several users must provide 5 seconds or less response time
NFR-5	Availability	Uninterrupted services must be available all time except the time of server updation
NFR-6	Scalability	Provide horizontal or vertical scaling for higher workloads

5. PROJECT DESIGN

5.1 Data Flow Diagrams

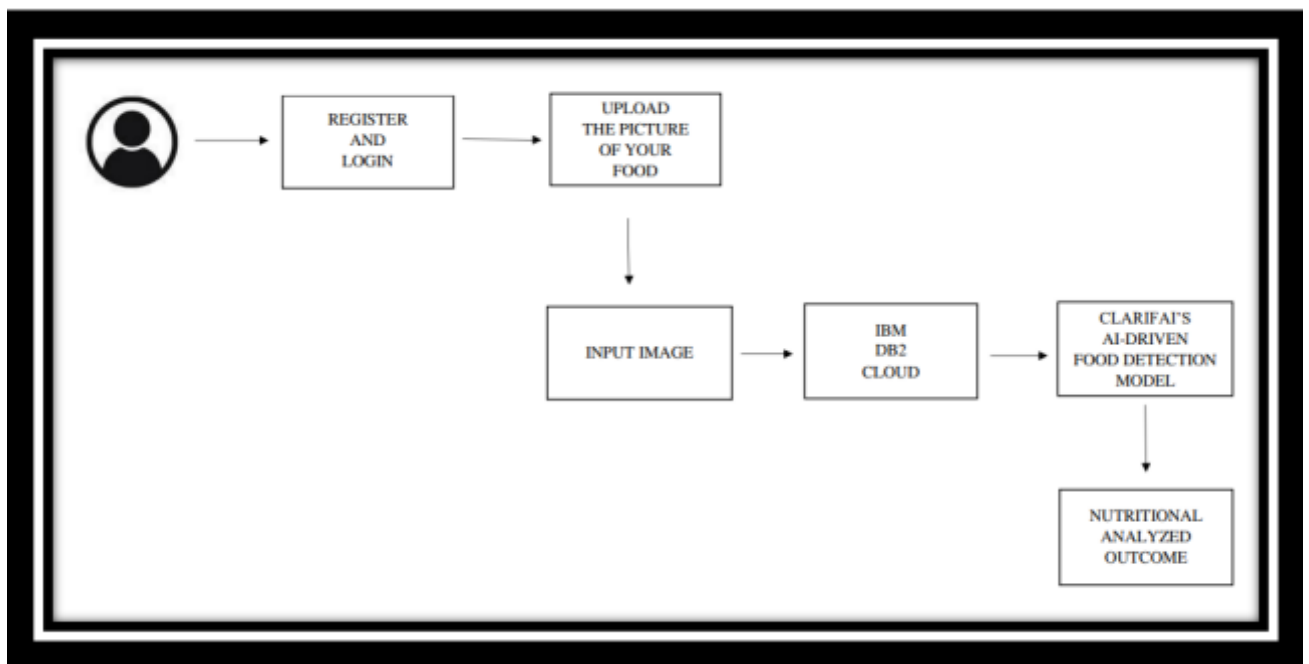
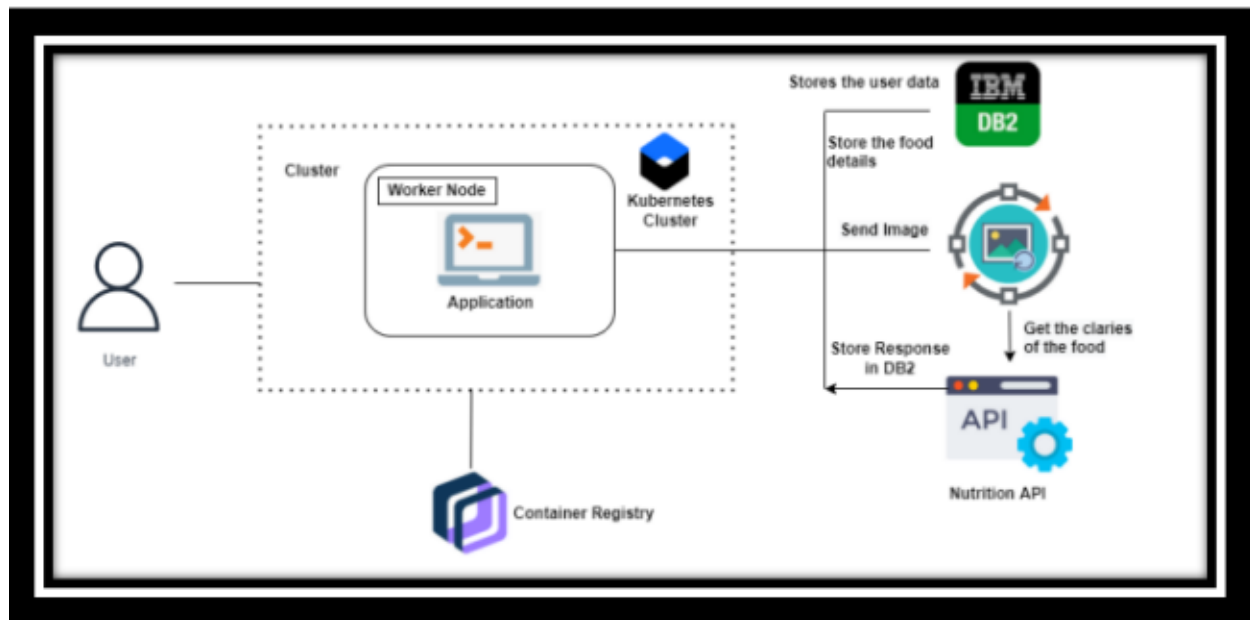
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering	I can access my account / dashboard	High	Sprint-1

			my email, password, and confirming my password.			
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Profile updation	USN-3	As a user, I have to enter my height, weight, and daily activities	I can update these information on dashboard	High	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-5	As a user, I can upload or capture live image of the meal	I can get the nutritional value of that particular meal	High	Sprint-2
Administrator	Maintain the application	USN-6	Maintaining details for users	I can access database	High	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the	2	High	ALRIJWANA R ASVETHASRI

			application by entering my email, password, and confirming my password.			R BOOMATHI S JANANI V RAMYA P
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	ALRIJWANA R ASVETHASRI R BOOMATHI S JANANI V RAMYA P
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	1	High	ALRIJWANA R ASVETHASRI R BOOMATHI S JANANI V RAMYA P
Sprint-2	User details	USN-4	As a user, I can fill the details	2	High	ALRIJWANA R ASVETHASRI R BOOMATHI S JANANI V RAMYA P
Sprint-3	Push notification	USN-5	As a user, I will search the food item	2	Medium	ALRIJWANA R ASVETHASRI R BOOMATHI S JANANI V RAMYA P
Sprint-4	Dashboard/shown the nutrition details and recipe	USN-6	As a user,I can scan and get nutrition	1	High	ALRIJWANA R ASVETHASRI

	for scanned food.		value			R BOOMATHI S JANANI V RAMYA P
--	-------------------	--	-------	--	--	---

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 1	16 Nov 2022	20	16 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)

$$AV = \text{sprint duration} / \text{velocity} = 20 / 10 = 2$$

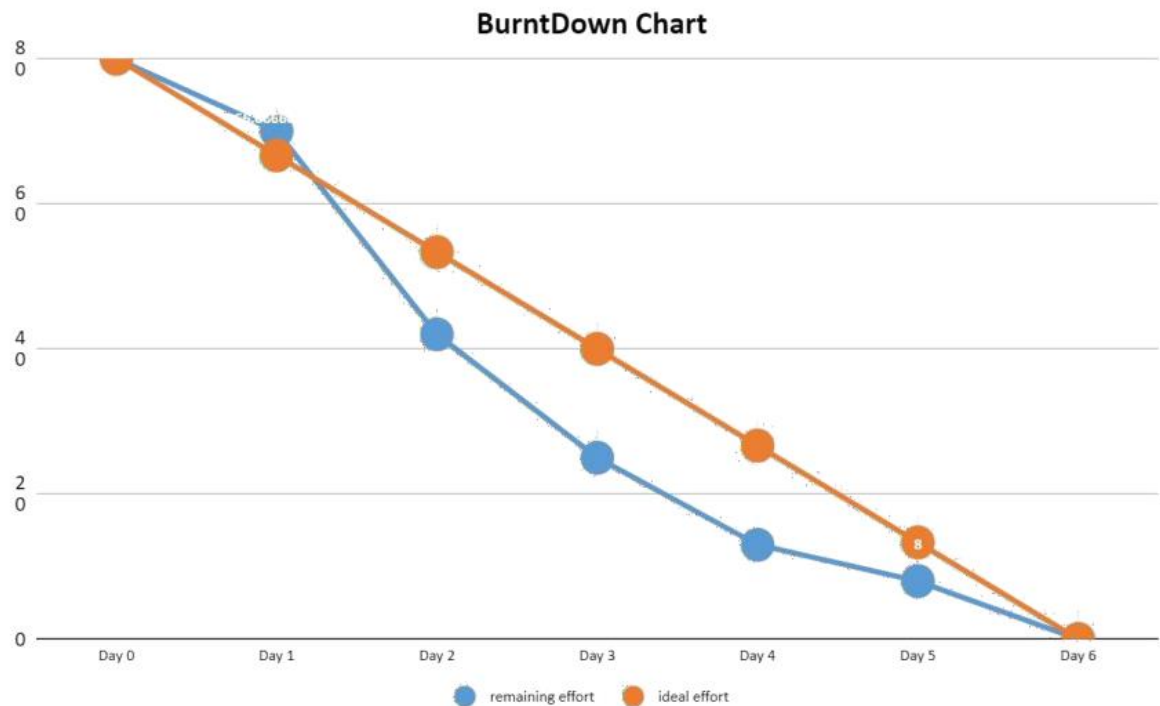
Average Velocity = Story Points per Day Sprint Duration = Number of (Duration) days per Sprint Velocity = Points per Sprint 20 AV = ≈ 4.6

Therefore, the AVERAGE VELOCITY IS 4 POINTS PER SPRINT

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software develop

	Initial Estimate	24-Oct	25-Oct	26-Oct	27-Oct	28-Oct	29-Oct
Sprint number	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
Sprint-1	20	0	10	5	3	1	1
Sprint-2	20	2	10	4	1	1	2
Sprint-3	20	5	5	5	5	0	0
Sprint-4	20	3	3	3	3	3	5
remaining effort	80	70	42	25	13	8	0
ideal effort	80	<u>66.66666667</u>	<u>53.33333333</u>	<u>40</u>	<u>26.66666667</u>	<u>13.33333333</u>	<u>0</u>

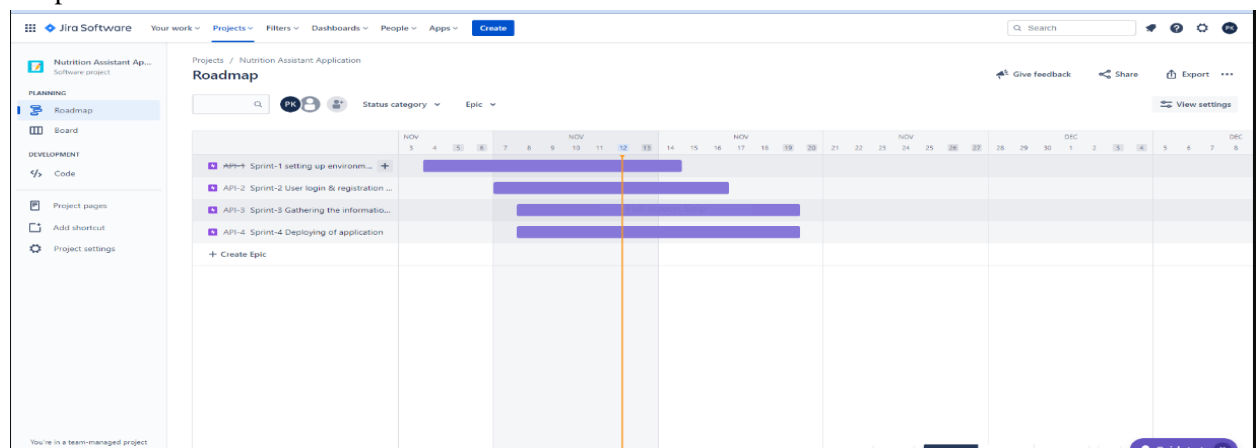


6.2 Sprint Delivery Schedule

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc	3 SEPTEMBER 2022

Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	10 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance	10 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	18 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	1 OCTOBER 2022
Solution Architecture	Prepare solution architecture document.	19 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application.	15 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review	20 OCTOBER 2022
Technology Architecture	architecture diagram.	25 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	30 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	16 NOVEMBER 2022

6.3 Reports from JIRA



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

7.1 Feature 1

Register.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="register.css">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body background="nutrition bg.png">
<div class="container">
<div class="title">Registration</div>
<div class="content">
<form action="{ {url_for('register')}}" method="POST" class="login email">
<div class="user-details">
<div class="input-box">
<span class="details">Full Name</span>
<input type="text" placeholder="Enter your name" name="fullname">
</div>
<div class="input-box">
<span class="details">Username</span>
<input type="text" placeholder="Enter your username"
name="username">
</div>
<div class="input-box">
<span class="details">Email</span>
<input type="text" placeholder="Enter your email" name="email">
</div>
<div class="input-box">
<span class="details">Phone Number</span>
<input type="text" placeholder="Enter your number"
name="phonenumber">
</div>
<div class="input-box">
<span class="details">Password</span>
<input type="password" placeholder="Enter your password"
name="passwords">
</div>
<div class="input-box">
<span class="details">Confirm Password</span>
<input type="password" placeholder="Confirm your password"
name="cpassword">
</div>
```

```

</div>
<div class="button">
  <a href="login.html"> <center>REGISTER </center></a>
<br><br>
  already registered?
  <a href="login.html"> login </a>
</div>
</form>
</div>
</div>
</div>
</body>
</html>

```

Login.html

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="register.css">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body background="nutrition bg.png">
<div class="container">
<div class="title">Login</div>
<div class="content">
<form action="{ {url_for('register')}} " method="POST" class="login email">
<div class="user-details">
<div class="input-box">
<span class="details">Username</span>
<input type="text" placeholder="Enter your username"
name="username">
</div>

<br><br>
<div class="input-box">
<span class="details">Password</span>
<input type="password" placeholder="Enter your password"
name="passwords">
</div>
</div>
<div class="button">
<a href="USER DETAILS.html">
<center>SUBMIT</center></a>
<br><br>
  not registered?

```

```
<a href="register.html"> register </a>
</div>
</form>
</div>
</div>
</body>
</html>
```

USER DETAILS.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="register.css">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body background="nutrition bg.png">
<div class="container">
<div class="title">USER DETAILS</div>
<div class="content">
<form action="{ {url_for('register')}} " method="POST" class="login email">
<div class="user-details">
<div class="input-box">
<span class="details">FULL NAME</span>
<input type="text" placeholder="Enter your name" name="fullname">
</div>
<div class="input-box">
<span class="details">HEIGHT</span>
<input type="text" placeholder="Enter your Height" name="fullname">
</div>
<div class="input-box">
<span class="details">WEIGHT</span>
<input type="text" placeholder="Enter your Weight" name="fullname">
</div>
<div class="input-box">
<span class="details">BLOOD PRESSURE </span>
<input type="text" placeholder="Enter your B.P mmHg value" name="fullname">
</div>
<div class="input-box">
<span class="details">DIABETICS </span>
<input type="text" placeholder="Enter your Diabetics mg/dl value" name="fullname">
</div>
<div class="input-box">
<span class="details">AGE</span>
<input type="text" placeholder="Enter your Age" name="fullname">
```

```

</div>
</div>
<div class="button">
<a href="dashboard.html">
<center>SUBMIT</center></a>
</div>
</form>
</div>
</div>
</body>
</html>

```

7.2 Feature 2

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 rel="stylesheet" href="static/styles.css">
  <link      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"                                integrity="sha384-
ZenH87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
  <title>Nutrition Assistant</title>
</head>

  <div class="row align-items-md-stretch">
    <div class="col-md-6 my-3">
      <div class="h-100 p-5 text-bg-dark rounded-3">
        <h2>Upload food image</h2>
        <form action = "/dashboard" method = "POST" enctype="multipart/form-data">
          <input class="my-3 form-control" type="file" name="file" required/>
          <a href="food details.html"><center>ANALYZE </center></a>

          </form>
        </div>
      </div>
    </div>
  </div>
</div>

```


LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	Login page	1.Enter URL and click go 2.Click on My Account drop down button 3.Verify login/Signup popup displayd or not	login.html	Login/Signup popup should display	Working as expected	pass				
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup	Signup page	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Signup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer?	register.html	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create	Working as expected	Pass			BUG-1234	

					Create account link e.Last password? Recovery password link		account link e.Last password? Recover y password d link						
--	--	--	--	--	--	--	--	--	--	--	--	--	--

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity2	Severity3	Severity4	Subtotal
By design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't fix	0	5	2	1	8
Totals	24	14	13	26	77

3. Test Case Analysis

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

Section	Total cases	Not tested	Fail	Pass
Print engine	7	0	0	7

Client application	51	0	0	51
Security	2	0	0	2
Outsource shipping	3	0	0	3
Exception reporting	9	0	0	9
Final report output	4	0	0	4
Version control	2	0	0	2

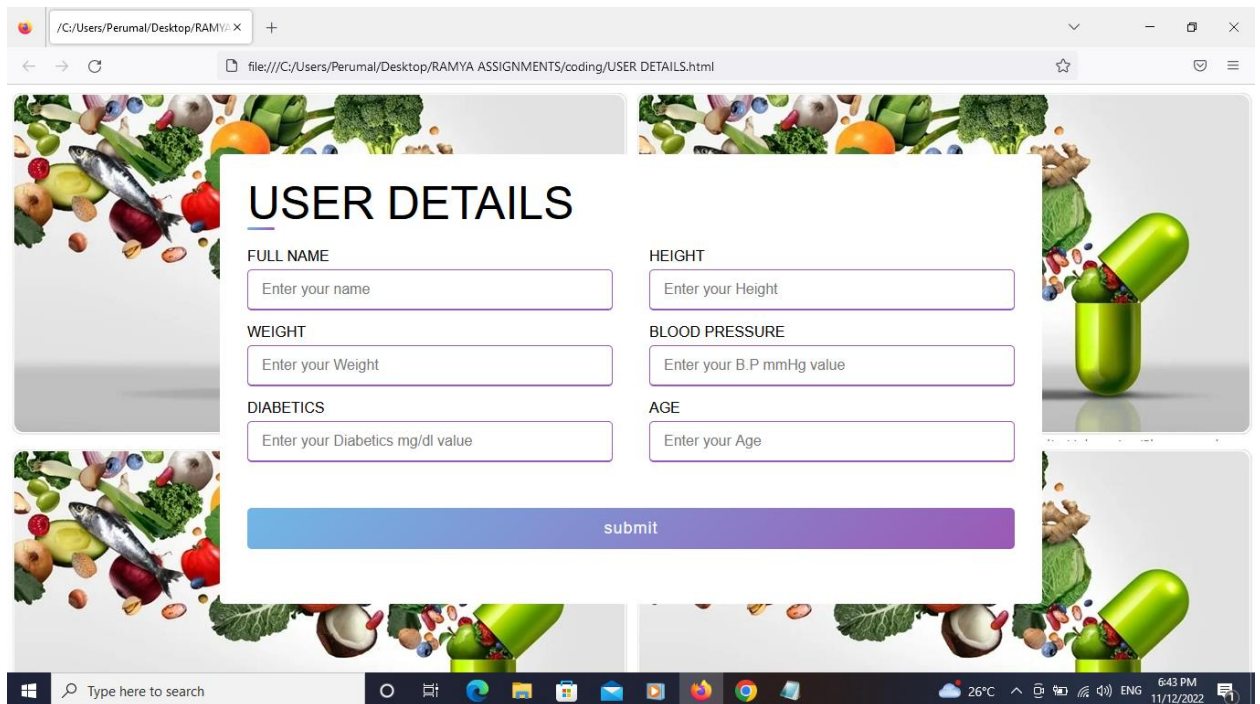
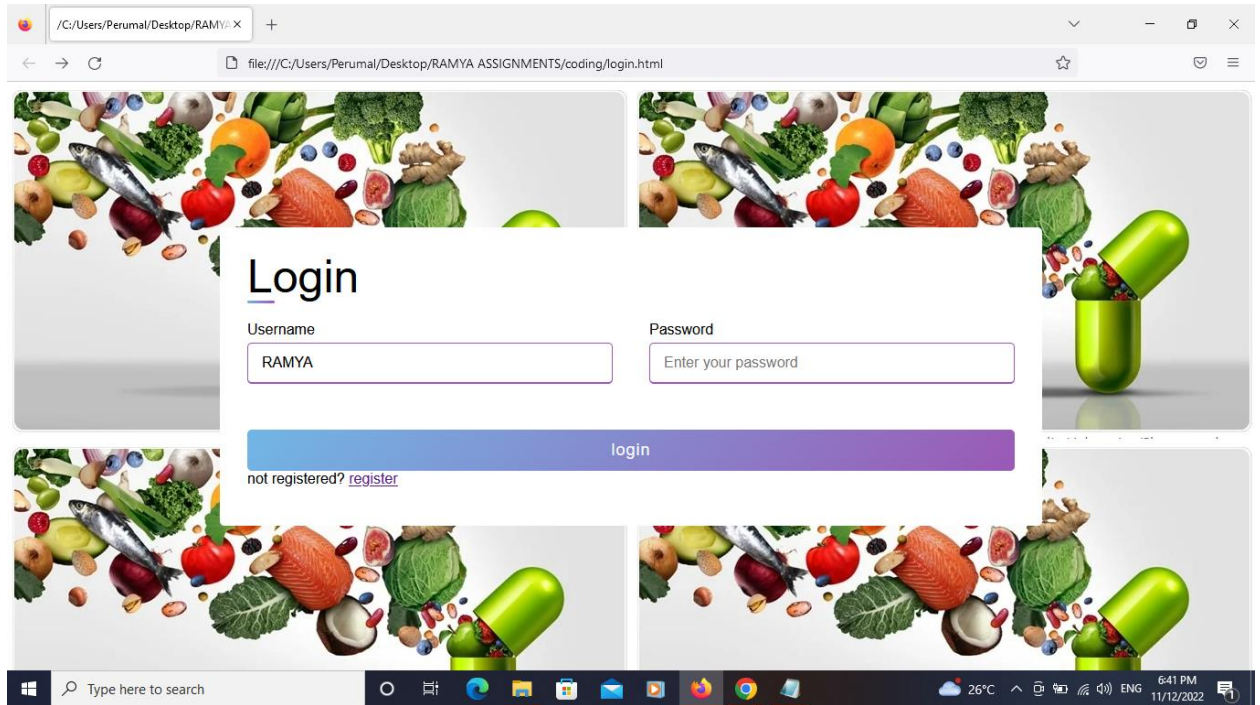
9. RESULTS

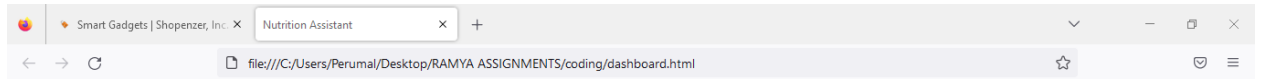
9.1 Performance Metrics

The screenshot displays a web browser window with the address bar showing the file path: `file:///C:/Users/Perumal/Desktop/RAMYA ASSIGNMENTS/coding/register.html`. The registration form is centered on the page, set against a decorative background of various fruits and vegetables. The form includes the following elements:

- Registration Title:** A large, bold heading at the top of the form.
- Input Fields:** Six text input fields arranged in two columns:
 - Full Name (placeholder: "Enter your name")
 - Username (placeholder: "Enter your username")
 - Email (placeholder: "Enter your email")
 - Phone Number (placeholder: "Enter your number")
 - Password (placeholder: "Enter your password")
 - Confirm Password (placeholder: "Confirm your password")
- Register Button:** A wide, blue button with the text "Register" centered below the input fields.
- Login Link:** A text link "already registered? [login](#)" located at the bottom left of the form.

The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system tray on the right indicates a temperature of 26°C, the date 11/12/2022, and the time 6:42 PM.





Upload food image

Browse... No file selected.

[ANALYZE](#)



Smart Gadgets | Shopenzer, Inc. x Nutrition Assistant

FOOD DETAILS

FOOD NAME: BURGER

INGREDIENTS:

- CHEESE
- ONION
- MAYONNAISE
- GROUND BEEF
- GARLIC
- SALT
- BLACK PEPPER

NUTRITION VALUE:

- CALORIES : 239
- CALORIES FROM FAT : 106
- CHOLESTEROL 54mg : 18%

10. ADVANTAGES & DISADVANTAGES

The advantages of nutrition programs are as follows:

- It provides a maintained strategy of healthy eating habits.
- It delivers information on the nutritional value of foods and how balanced and healthy eating habits are important for us.
- It limits the amount of unnecessary food such as fat that people consume a lot.
- Increase health literacy.

The disadvantages of nutrition programs are as follows:

- Sometimes it makes a level of disbalance in the balanced diet of an individual.
- It can improve the level of nutrition among individuals but delivers an inappropriate means of nutritional labeling.
- Sometimes, it is considered one of the major factors of weight gain.

11. CONCLUSION

In this study, we conducted a critical review of mobile apps from three popular app stores. Our search results identified a total of 473 related apps, from which we selected and evaluated 80 apps using our modified app rating tool. We devised this app rating tool specifically for analyzing food consumption tracking and recommendation apps by adopting and extending existing mobile app rating scales. Using this rating tool, we evaluated the selected 80 apps and analysed and identified their design faults. According to our evaluation, most of the existing mobile apps in the app stores do not meet the essential requirements for correctly tracking food consumption and recommendations.

Although a few apps had some of the expected features, none met all the required functionalities. For most of the apps, tracking information required manual data input. The databases that are used in the apps are not enriched. We also observed that there are very few evidence-based apps. Because there have been numerous studies about automatic food recognition, food portion size estimates, and nutritional value assessments, these aspects must be included in modern food consumption tracking and recommendation apps. Also, there has been much research on food recommendations but this feature is absent in most of the evaluated apps, that is why this feature needs to be included in future apps. These apps suggest diet plans, recommend foods to users, and estimate nutrient values, so an expert dietitian or nutritionist should be involved in their development. Also, enrichment of the database is required as nowadays multiple food datasets are available. Software qualities (aesthetics, general features, performance, usability) also play a vital role in commercial apps and thus developers

need to consider these matters. Nonetheless, the analysis provided here covers a variety of general quality features and specific functional features that can be used in food consumption tracking and recommendation apps to provide consumers with a realistic and evidence-based experience. Studies show how people use smartphones to improve their fitness and obesity literacy, as well as the overall status of the commercial product market for food consumption tracking and recommendation apps.

This study will open the door to future researchers who focus on the implementation, effectiveness and performance measurement of food computing apps.

12. FUTURE SCOPE

Scientific analysis of food and nutrients began during the chemical revolution in the late-18th century. Chemists in the 18th and 19th centuries experimented with different elements and food sources to develop theories of nutrition. Modern nutrition science began in the 1910s as individual micronutrients began to be identified. The first vitamin to be chemically identified was thiamine in 1926, and the role of vitamins in nutrition was studied in the following decades. The first recommended dietary allowances for humans were developed during the Great Depression and World War II. Due to its importance in human health, the study of nutrition has heavily emphasized human nutrition and agriculture, while ecology is a secondary concern .

Nutrition plays a pivotal role in leading a healthy life. It is a vital element required in every stage of life. Nutritious food intake and metabolism of nutrients are associated with the decreased risk of both infectious and non-communicable diseases. Nutritious diet is a major determinant of future health – physical, mental and social health, not merely an absence of disease.

13. APPENDIX

Source Code

```
import os, re, string, random, time, datetime, requests, sendgrid, random, flask
import ibm_db
from sendgrid.helpers.mail import *
from flask import Flask, request, render_template, flash, redirect, url_for, session
from werkzeug.utils import secure_filename
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2, resources_pb2, service_pb2_grpc
from clarifai_grpc.grpc.api.status import status_code_pb2
```

```
#####  
#####
```

```
UPLOAD_FOLDER = 'static/uploads'  
ALLOWED_EXTENSIONS = set(['png', 'jpg', 'jpeg'])  
SENDGRID_API_KEY = "SG.HwfSJ6D4Tba6O-h7fL1JlA.z2_qdNI-  
iXOhrhdzsx05PiEPj3bbNKXF_Rms0eRis4c"
```

```
app = Flask(__name__)  
app.secret_key = "bimbilikibilapi"  
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER  
app.config['MAX_CONTENT_LENGTH'] = 16 * 1024 * 1024
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-  
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;Security=SSL;P  
ROTOCOL=TCPIP;UID=pzt20234;PWD=r7CB0AmR1QtOHfR4;","","")  
#;SSLServerCertificate=DigiCertGlobalRootCA.crt
```

```
YOUR_CLARIFAI_API_KEY = "af4bc9886c744e998ee0e20f104b1518"  
YOUR_APPLICATION_ID = "test"  
SAMPLE_URL =  
"https://res.cloudinary.com/swiggy/image/upload/f_auto,q_auto,fl_lossy/nxmlubuz0b1qixa29gov  
"
```

```
metadata = (("authorization", f"Key {YOUR_CLARIFAI_API_KEY}"),)  
channel = ClarifaiChannel.get_grpc_channel()  
stub = service_pb2_grpc.V2Stub(channel)
```

```
RAPIDAPI_KEY = "74e62205b6msha6b4e69e0088de5p12c619jsn1ed9cc5e0727"
```



```
def allowed_file(filename):
    return '.' in filename and \
        filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
```

```
def sendMail(to, title, text):
    sg = sendgrid.SendGridAPIClient(api_key=SENDGRID_API_KEY)
    from_email = Email("nsnandhaa1@gmail.com")
    to_email = To(to)
    subject = title
    content = Content("text/plain", text)
    mail = Mail(from_email, to_email, subject, content)
    response = sg.client.mail.send.post(request_body=mail.get())
    print(response.status_code)
    print(response.body)
    print(response.headers)
```

```
@app.route("/forgot-pw", methods=["GET", "POST"])
def forgotpw():
    if flask.request.method == "POST":
        data = flask.request.form
        username=data['username']
        code = ''.join(random.choices(string.ascii_letters, k=6))

        sql= "SELECT * FROM users WHERE username=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        session['userid'] = account['USERID']
```

```
insert_sql = "INSERT INTO VERIFY VALUES(?,?)"
```

```
prep_stmt=ibm_db.prepare(conn, insert_sql)
```

```
ibm_db.bind_param(prepare_stmt, 1, account['USERID'])
```

```
ibm_db.bind_param(prepare_stmt, 2, code)
```

```
ibm_db.execute(prepare_stmt)
```

```
sendMail(account['EMAIL'], "Verification Code", code)
```

```
flash("We have sent a code to your registered email. please check spam folder also.")
```

```
return redirect(url_for("confirmMail"))
```

```
flash("We will send you a confirmation code to your registered email")
```

```
return render_template("forgot-pw.html")
```

```
@app.route("/confirm-mail", methods=["GET", "POST"])
```

```
def confirmMail():
```

```
    session['LoggedIn'] = False
```

```
    if flask.request.method == "POST":
```

```
        data = flask.request.form
```

```
        usercode=data['code']
```

```
        sql= "SELECT * FROM verify WHERE userid=?"
```

```
        stmt=ibm_db.prepare(conn,sql)
```

```
        ibm_db.bind_param(stmt,1,session['userid'])
```

```
        ibm_db.execute(stmt)
```

```
        verify=ibm_db.fetch_assoc(stmt)
```

```
        print(verify)
```

```
        dbcode = verify['CODE']
```

```
        if usercode == dbcode:
```

```
            session['LoggedIn'] = True
```

```
            delete_sql = "DELETE FROM verify WHERE CODE=?"
```

```
            prep_stmt=ibm_db.prepare(conn, delete_sql)
```

```
            ibm_db.bind_param(prepare_stmt, 1, dbcode)
```

```
            ibm_db.execute(prepare_stmt)
```

```
    flash("Email verified. Enter new password")
    return redirect(url_for("changepw"))
else:
    flash("Error")
    return render_template("confirm-mail")
return render_template("confirm-mail.html")
```

```
@app.route("/change-pw", methods=["GET", "POST"])
```

```
def changepw():
```

```
    if flask.request.method == "POST" and session['LoggedIn']:
```

```
        data = flask.request.form
```

```
        password=data['pw']
```

```
        sql = "UPDATE users SET PASSWORD=? WHERE USERID=?"
```

```
        prep_stmt=ibm_db.prepare(conn, sql)
```

```
        print(password, session['userid'])
```

```
        ibm_db.bind_param(prepare_stmt, 1, password)
```

```
        ibm_db.bind_param(prepare_stmt, 2, session['userid'])
```

```
        ibm_db.execute(prepare_stmt)
```

```
        flash("Password changed.")
```

```
        return redirect(url_for("login"))
```

```
    else:
```

```
        flash("verification error")
```

```
        redirect(url_for("confirmMail"))
```

```
    return render_template("change-pw.html")
```

```
@app.route("/register", methods=["GET", "POST"])
```

```
def reg():
```

```
    if flask.request.method == "POST":
```

```
        data = flask.request.form
```

```
        email=data['email']
```

```
        username=data['username']
```

```
        password=data['pw']
```

```

sql= "SELECT * FROM users WHERE username=?"
stmt=ibm_db.prepare(conn,sql)
ibm_db.bind_param(stmt,1,username)
ibm_db.execute(stmt)
account=ibm_db.fetch_assoc(stmt)
print(account)
if account:
    flash("Account already exists!")
elif not re.match(r'^@[^@]+\.[^@]+', email):
    flash("invalid email address")
elif not re.match(r'[A-Za-z0-9]+', username):
    flash("name must contain only characters and numbers")
else:
    insert_sql = "INSERT INTO users VALUES(?,?,?,?)"
    prep_stmt=ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt, 1, username)
    ibm_db.bind_param(prepare_stmt, 2, email)
    ibm_db.bind_param(prepare_stmt, 3, password)
    ibm_db.bind_param(prepare_stmt, 4, ".join(random.choices(string.ascii_letters, k=16)))
    ibm_db.execute(prepare_stmt)
    flash("logged in")

    return redirect(url_for("dashboard"))
return render_template("reg.html")

```

```

@app.route("/login", methods=["GET", "POST"])

```

```

def login():

```

```

    if flask.request.method == "POST":

```

```

        data = flask.request.form

```

```

        username=data['username']

```

```

        password=data['pw']

```

```

sql = "SELECT * FROM users WHERE username=? AND password=?"
stmt = ibm_db.prepare(conn,sql)
ibm_db.bind_param(stmt, 1, username)
ibm_db.bind_param(stmt, 2, password)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print(account)
if account:
    session['LoggedIn'] = True
    session['userid'] = account['USERID']
    session['username'] = account['USERNAME']
    userid = account['USERID']
    flash("logged in")
    return redirect(url_for("dashboard"))
else:
    flash("error")

return render_template("login.html")

```

```

@app.route("/dashboard", methods=["GET", "POST"])
def dashboard():
    global request
    if flask.request.method == "POST" and session['LoggedIn']:
        if 'file' not in flask.request.files:
            flash('No file part')
            return redirect(flask.request.url)
        file = flask.request.files['file']
        if file.filename == "":
            flash('No image selected')
            return redirect(flask.request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)

```

```

file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
flash('Image successfully uploaded')

with open(os.path.join(app.config['UPLOAD_FOLDER'], filename), "rb") as f:
    file_bytes = f.read()

request = service_pb2.PostModelOutputsRequest(
    model_id="food-item-v1-recognition",
    user_app_id=resources_pb2.UserAppIDSet(app_id=YOUR_APPLICATION_ID),
    inputs=[
        resources_pb2.Input(
            data=resources_pb2.Data(image=resources_pb2.Image(
                base64=file_bytes
            )
        )
    ],
)
response = stub.PostModelOutputs(request, metadata=metadata)

if response.status.code != status_code_pb2.SUCCESS:
    print(response)
    raise Exception(f"Request failed, status code: {response.status}")

foodname = response.outputs[0].data.concepts[0].name

ingredients = "
for concept in response.outputs[0].data.concepts:
    ingredients += f"{concept.name}: {round(concept.value, 2)}, "

nutritionValues = "
#         nutritionApiUrl         =         "https://spoonacular-recipe-food-nutrition-
v1.p.rapidapi.com/recipes/guessNutrition"
# querystring = {"title":foodname}

```

```

# headers = {
#   "X-RapidAPI-Key": RAPIDAPI_KEY,
#   "X-RapidAPI-Host": "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com"
# }

# response = requests.request("GET", nutritionApiUrl, headers=headers,
params=querystring)

# nutritions = response.text
nutritions = {
  "recipesUsed": 10,
  "calories": {
    "value": 470,
    "unit": "calories",
    "confidenceRange95Percent": {
      "min": 408.93,
      "max": 582.22
    },
    "standardDeviation": 139.8
  },
  "fat": {
    "value": 17,
    "unit": "g",
    "confidenceRange95Percent": {
      "min": 12.81,
      "max": 21.36
    },
    "standardDeviation": 6.9
  },
  "protein": {
    "value": 15,
    "unit": "g",
    "confidenceRange95Percent": {
      "min": 9.06,
      "max": 29.78
    },
  },

```

```

        "standardDeviation": 16.71
    },
    "carbs": {
        "value": 65,
        "unit": "g",
        "confidenceRange95Percent": {
            "min": 57.05,
            "max": 77.9
        },
        "standardDeviation": 16.81
    }
}

nutritions.pop('recipesUsed')
for i in nutritions:
    nutritionValues += f"{i}: {nutritions[i]['value']} {nutritions[i]['unit']}, "

sql = "INSERT INTO foods VALUES(?,?,?,?)"
stmt=ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt, 1, session['userid'])
ibm_db.bind_param(stmt, 2, datetime.datetime.now().strftime('%Y-%m-%d
%H:%M:%S'))
ibm_db.bind_param(stmt, 3, foodname)
ibm_db.bind_param(stmt, 4, ingredients)
ibm_db.bind_param(stmt, 5, nutritionValues)
ibm_db.execute(stmt)

# os.remove(os.path.join(app.config['UPLOAD_FOLDER'], filename))
return render_template("dashboard.html",
    filename = filename,
    username = session['username'],
    foodname = foodname,
    ingredients = ingredients,
    nutritionValues = nutritionValues,

```



```

    )
else:
    flash('Allowed image formats - png, jpg, jpeg')
    return redirect(flask.request.url)

elif session['LoggedIn']:
    return render_template("dashboard.html", username=session['username'])
else:
    return redirect(url_for("login"))

@app.route('/logout', methods=["GET", "POST"])
def logout():
    session.pop('LoggenIn', None)
    session.pop('userid', None)
    session.pop('username', None)
    return render_template("index.html")

@app.route('/display/<filename>', methods=["GET", "POST"])
def display(filename):
    print(filename)
    return redirect(url_for('static', filename='uploads/' + filename), code=301)

@app.route('/app', methods=["GET", "POST"])
def other():
    return render_template("index.html")

@app.route('/', methods=["GET", "POST"])
def index():
    return render_template("index.html")

```

```
if __name__ == "__main__":  
    app.run(host='0.0.0.0', port = 5000)
```

GitHub & Project Demo Link

Github link :

<https://github.com/IBM-EPBL/IBM-Project-44517-1660725000>

Project demo link:

<https://photos.app.goo.gl/56NSKNduj74g2mMN9>