PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview

Food is essential for human life and has been the concern of many healthcare conventions. Nowadays new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintain a healthy diet. Nutritional analysis is the process of determining the nutritional content of food. It is a vital part of analytical chemistry that provides information about the chemical composition, processing, quality control and contamination of food.

The main aim of the project is to building a model which is used for classifying the fruit depends on the different characteristics like colour, shape, texture etc. Here the user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).

1.2 Purpose

The purpose of the project is to build a model to classify the fruit depending on different characteristics like colour, shape, texture etc. The user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image and display the nutrition content present in the fruit.

2. LITERATURE SURVEY

2.1 Existing problem

Although the present generation are becoming more heath conscious, they find it difficult to remember the nutrient content of each and every food item. Also it be time consuming to search the nutrient content of every food item before they are eaten. Hence, this project proposes a solution where just capturing the image would list or display its nutrient content.

2.2 References

- C. L. Ogden, M. D. Carroll, B. K. Kit, and K. M. Flegal, "Prevalence of childhood and adult obesity in the United States, 2011-2012," Jama, vol. 311, pp. 806-814, 2014.
- G. H. Beaton, J. Milner, P. Corey, V. McGuire, M. Cousins, E. Stewart, et al., "Sources of variance in 24-hour dietary recall data: implications for nutrition study design and interpretation," American Journal of Clinical Nutrition, vol. 32, 1979.
- J. Cade, R. Thompson, V. Burley, and D. Warm, "Development, validation and utilisation of food-frequency questionnaires–a review," Public health nutrition, vol. 5, pp. 567-587, 2002.
- Y. Matsuda and K. Yanai, "Multiple-food recognition considering co-occurrence employing manifold ranking," in Pattern Recognition (ICPR), 2012 21st International Conference on, 2012, pp. 2017-2020.

- S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, pp. 2249-2256.
- F. Zhu, M. Bosch, I. Woo, S. Kim, C. J. Boushey, D. S. Ebert, et al., "The use of mobile devices in aiding dietary assessment and evaluation," Selected Topics in Signal Processing, IEEE Journal of, vol. 4, pp. 756-766, 2010.
- B. L. Daugherty, T. E. Schap, R. Ettienne-Gittens, F. M. Zhu, M. Bosch, E. J. Delp, et al., "Novel technologies for assessing dietary intake: evaluating the usability of a mobile telephone food record among adults and adolescents," Journal of medical Internet research, vol. 14, 2012.
- C. Xu, Y. He, N. Khannan, A. Parra, C. Boushey, and E. Delp, "Image-based food volume estimation," in Proceedings of the 5th international workshop on Multimedia for cooking & eating activities, 2013, pp. 75-80.
- N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol. 15, pp. 1929-1958, 2014.
- K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
- M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in Computer vision–ECCV 2014, ed: Springer, 2014, pp. 818-833.
- C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going Deeper With Convolutions," in Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, 2015.
- M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in Computer vision–ECCV 2014, ed: Springer, 2014, pp. 818-833.
- C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going Deeper With Convolutions," in Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, 2015.
- K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv preprint arXiv:1512.03385, 2015.
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE vol. 86, pp. 2278-2324, 1998.
- V. Stantchev, A. Barnawi, S. Ghulam, J. Schubert, and G. Tamm, "Smart Items, Fog and Cloud Computing as Enablers of Servitization in Healthcare," Sensors & Transducers, vol. 185, p. 121, 2015.
- Y. Cao, C. Liu, B. Liu, M. J. Brunette, N. Zhang, T. Sun, et al., "Improving Tuberculosis Diagnostics using Deep Learning and Mobile Health Technologies among Resource-poor and Marginalized Communities," in IEEE Conference on Connected Health: Applications, Systems and Engineering Technologies, 2016.
- Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, et al., "Caffe: Convolutional architecture for fast feature embedding," in Proceedings of the ACM International Conference on Multimedia, 2014, pp. 675-678.

- J. Oh, S. Hwang, Y. Cao, W. Tavanapong, D. Liu, J. Wong, et al., "Measuring objective quality of colonoscopy," IEEE Transactions on Biomedical Engineering, vol. 56, pp. 2190-2196, September 2009.
- R. C. Gonzalez and R. E. Woods, Digital Image Processing: Prentice Hall, 2008.
- K. He and J. Sun, "Convolutional neural networks at constrained time cost," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5353-5360.

2.3 Problem Statement Definition

To develop a nutrient analyzer that displays the nutrient content present in the food item by capturing it.

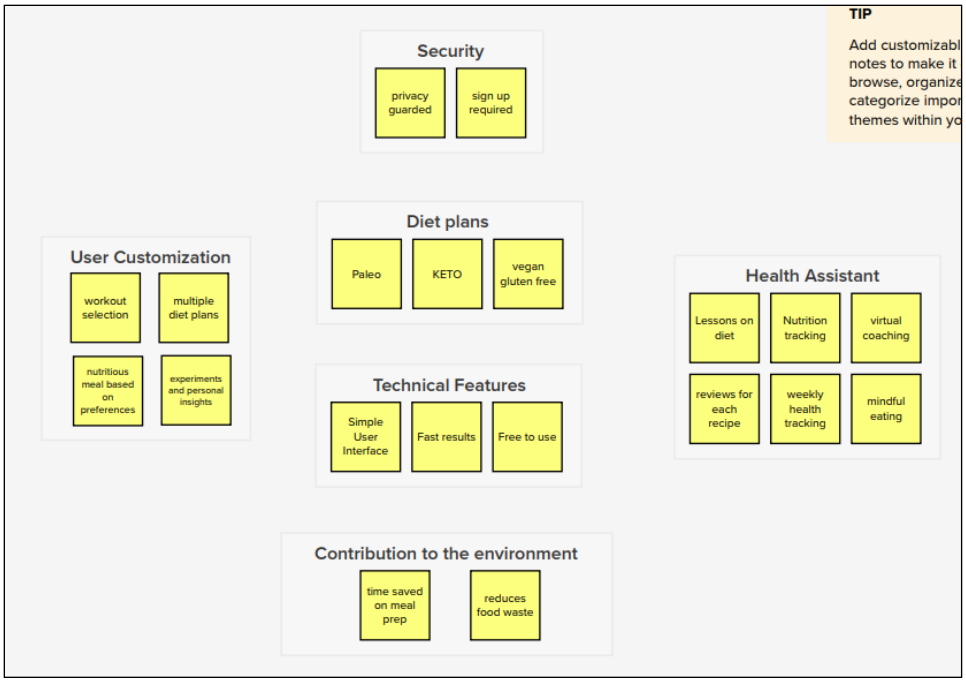
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

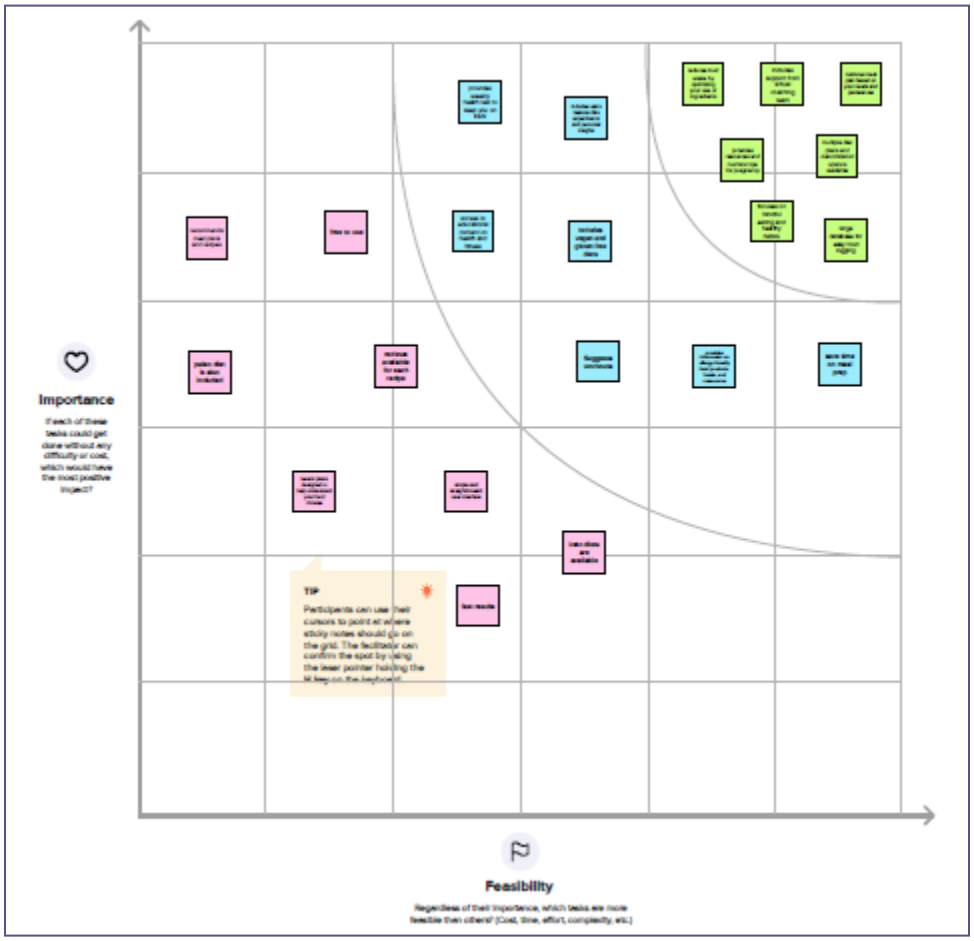
3.2 Ideation & Brainstorming

Problem Statement:

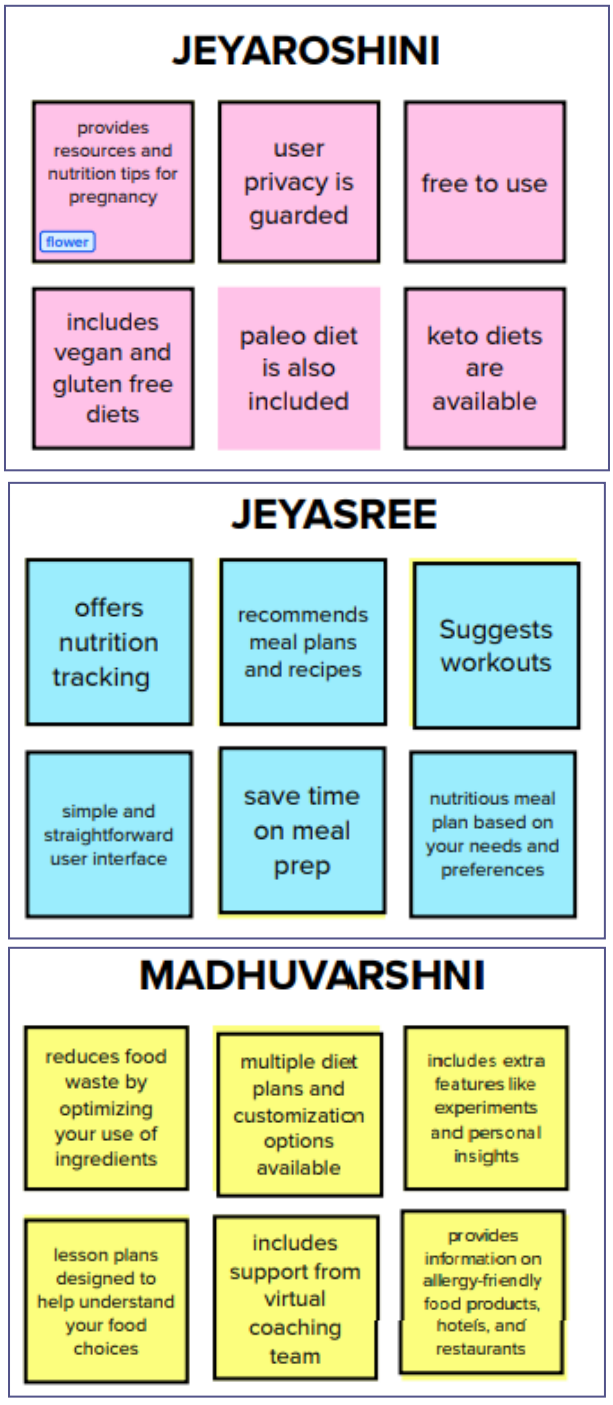
To develop a nutrient analyzer that displays the nutrient content present in the food item by capturing it.

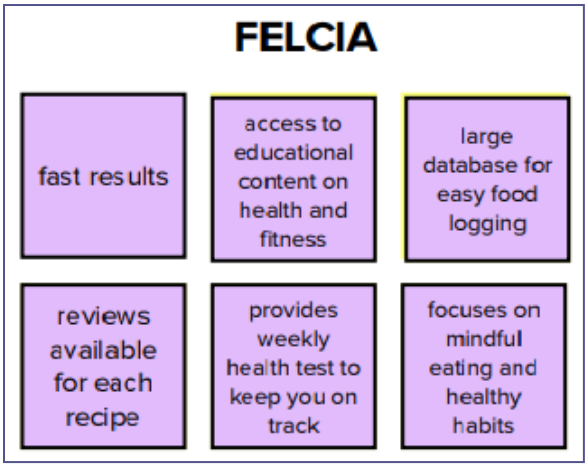


Idea prioritization



Brainstorming





3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The main aim of the project is to building a model which is used for classifying the fruit depends on the different characteristics like colour, shape, texture etc. The user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).
2.	Idea / Solution description	The user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.). The user can select the fruit accordingto his/her calorie requirement.
3.	Novelty / Uniqueness	The developed application would identify the fruit brought under the lens and then give the nutritional value of that particular fruit based on real time recognition rather than the user feeding data into the application. A weekly report is provided to monitor the progress of the user to check the analyzing view level.

4.	Social Impact / Customer Satisfaction	The nutritional information on food services could be part of a public health policy against the increasing rate of obesity. This application provides nutritional analysis of each of the recognized fruit or food item andthe user can select his/her desired item. The application is characterized as an easy-to- use, comprehensive, and useful tool.

3.4 Problem Solution fit

1. CUSTOMER

Who is your customer?

Any individual above the age of 18 will be having the need or a purpose to use the app. The individual

6. CUSTOMER

What constraints prevent your customers from taking action or limit theirchoices

Network connection would be an import factor as there will be a need to preprocess the captured data and display the result to the user.In some cases internet connection will also be necessary. The quality of the user's

5. AVAILABLE

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutionshave? i.e. pen

The user can go to an elevated area or to a spotwith more air space. In cases of low camera quality, the user

2. JOBS-TO-BE-DONE / PROBLEMS

- **The customer can obtain any nutritional information about the fooditem under the camera.**

9. PROBLEM ROOT CAUSE

What is the real reason

- **The customer is health conscious anddecides to start leading a healthy lifestyle.**
- **The customer wants to**

7. BEHAVIOUR

i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

3. TRIGGERS

TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

- **The customer might have reached an age where tracking his diet would seem essential.**
- **The customer would have become health conscious leading to monitoring his intake of calories.**

4. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

- **The customer becomes more confident about himself after they start using the app.**
- **They become more cautious with their eating habits.**
- **They can see a visible difference in their health graph.**

10. YOUR SOLUTION

SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

- **The customer needs to scan his/her desired food item.**
- **The item of interest is recognized using image recognition algorithms.**
- **Once the image has been recognized, the image is identified using existing food image with the help of deep learning algorithms and the nutritional content of that particular scanned image is returned to the customer.**

8. CHANNELS of BEHAVIOUR

CH

8.1 ONLINE

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

- **The customer can get extra information via online.**
- **The customer can write feedbacks online in case of any misshapen of the app.**

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

- **The customer learns to build a healthy lifestyle by keeping track of his/her calories.**
- **This helps in development of oneself towards a productive lifestyle.**

4. **REQUIREMENT ANALYSIS**

4.1 Functional requirement

<table>
<tr><th>FR NO.</th><th>FUNCTIONAL REQUIREMENTS(EPIC)</th><th>SUB REQUIREMENT(STORY/SUBTASK)</th></tr>
<tr><td>FR-1</td><td>USER REGISTRATION</td><td>-Registration through Gmail

-Registration through Mobile Number

-Registration through Face-book</td></tr>
<tr><td>FR-2</td><td>USER CONFIRMATION</td><td>-Confirmation via Email

-Confirmation via OTP</td></tr>
<tr><td>FR-3</td><td>USER DETAILS</td><td>PERSONAL DETAILS FOOD DETAILS<table><tr><td>Age</td><td>Food</td></tr><tr><td>Height</td><td>Recipe</td></tr><tr><td>Weight</td><td>Added ingredients</td></tr><tr><td>Diseases if any</td><td>Age</td></tr><tr><td>Conditions is any</td><td></td></tr><tr><td>Allergies is any</td><td></td></tr></table></td></tr>
<tr><td>FR-4</td><td>USER REQUIREMENTS</td><td>-The user simply inputs your recipe ingredients and amounts. The software will instantly produce an accurate readout of your dish in terms of nutritional analysis in a readable format that consumers are familiar with.

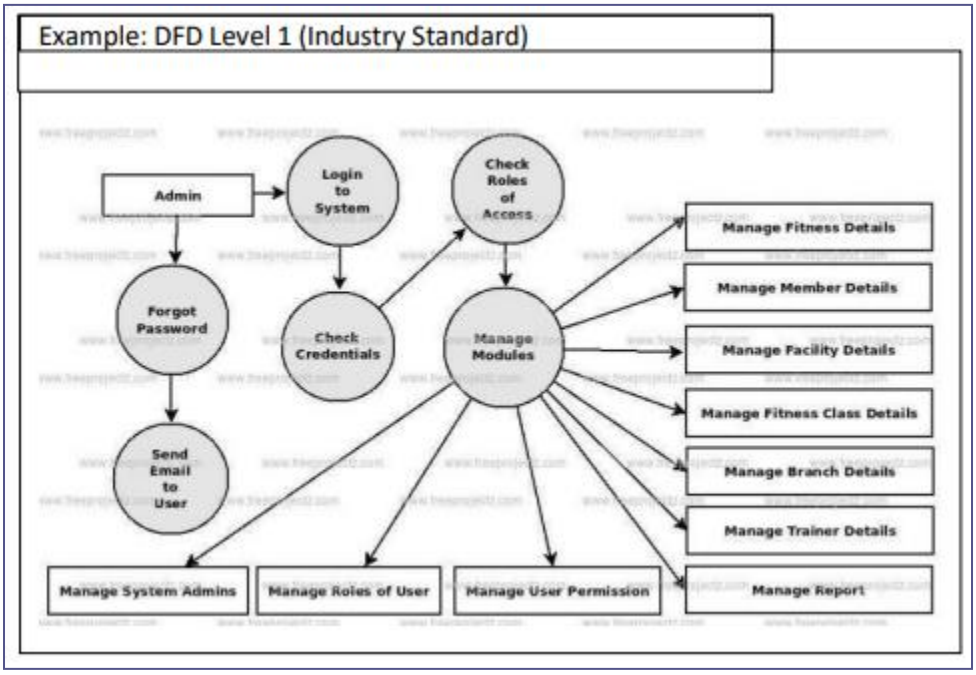
-With already given details the system can alert the consumer if any content of their allergies ,it can alert the consumer</td></tr>
</table>

4.2 Non-Functional requirements

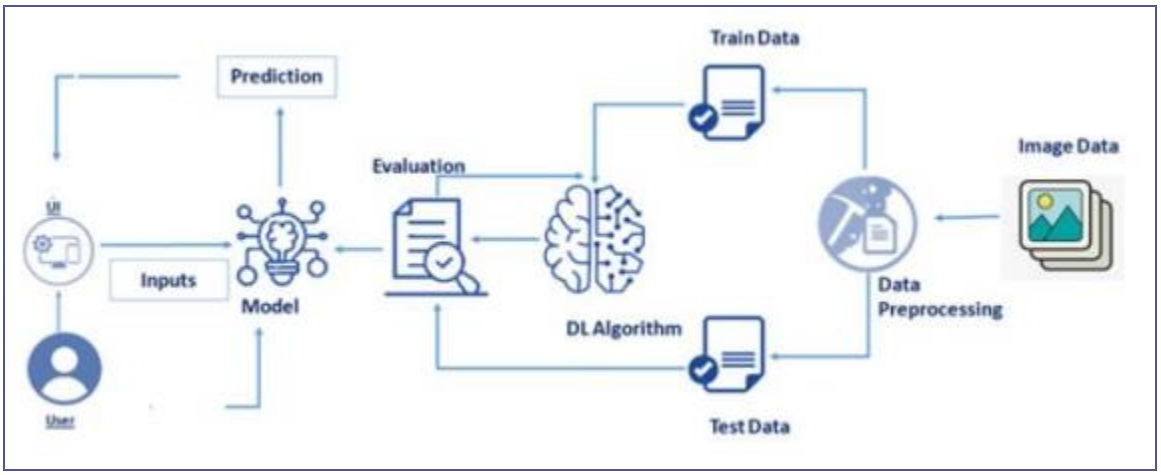
FR.NO	NON-FUNCTIONAL REQUIREMENTS	DESCRIPTION
NFR-1	USABILITY	• No training is required to access the Nutrition Analyzer. • The results should be loaded within30 seconds. • It should be user friendly and comfortable. • It should be simple and easy to use. • The results should be self explanatory so that it can be understood by common people.
NFR-2	SECURITY	• AI powered nutrition analyzer for fitness should contain more securityin which our data which entered or maintained should be more security. • With the help of the username and password it provides more securityin which it can access more securable and the data are private. • It should be social-economic which should access to sufficient and safeto use. • It is Important that the AI powered nutrition analyzer for fitness provides should Must reliable.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-2
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-2
	Dashboard	USN-6	As a user, I can select the activity I wish to do in the application.	Desired activity is opted	High	Sprint-3
	Food recognition	USN-7	The food item under the lens is recognised	Food item is recognised	High	Sprint-4
	Nutrient Content	USN-8	Once the food item is recognised, its nutrient content is displayed to the user.	Nutrient content is displayed	High	Sprint-5
Administrator	Updation	USN-9	The food dataset is updated by the administrator periodically.		High	Sprint-6

6. PROJECT PLANNING & SCHEDULING

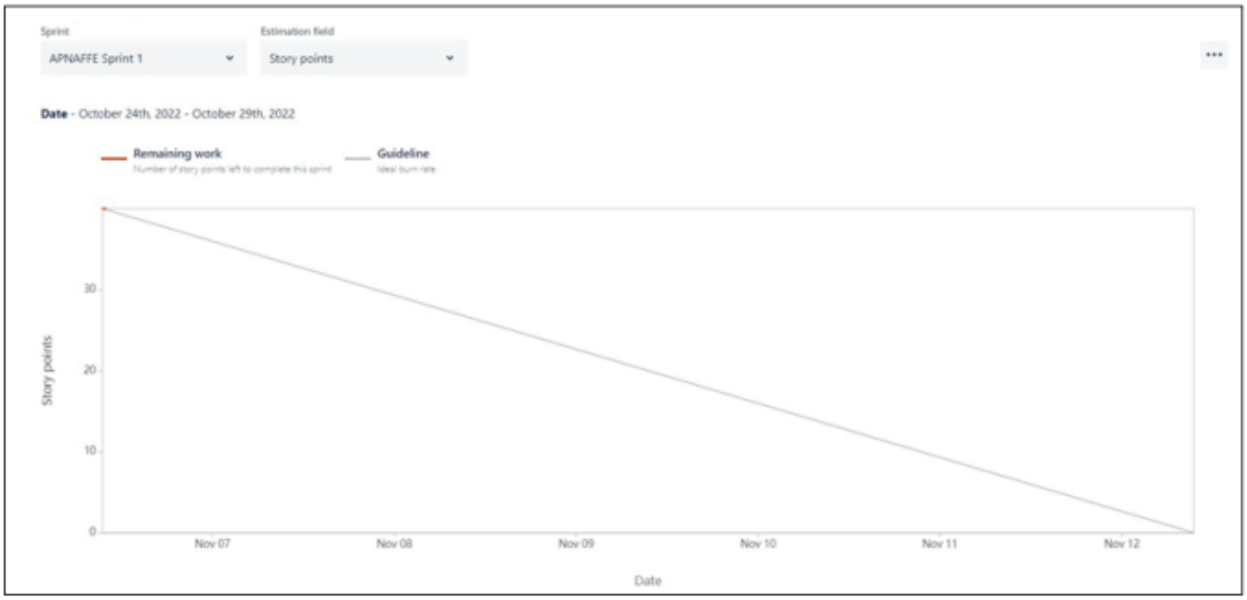
6.1 Sprint Planning & Estimation

Sprint	Functional Requirement(Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Dataset - Collecting images of food items apples,banana, orange,pineapple, watermelon for analysis	3	High	Jeyasree, Jeyaroshini
Sprint-1	Image Preprocessing	USN-2	Image data augmentation - Increasing the amount of data bygenerating new data points from existing data	3	Medium	Felcia, Madhuvarshni
Sprint-1		USN-3	Image Data Generator Class - Used for getting the input of theoriginal data	3	Medium	Jeyasree, Jeyaroshini
Sprint-1		USN-4	Applying image data generator functionality to trainset and testset	3	High	Felcia, Madhuvarshni
Sprint-2	Modelling Phase	USN-5	Defining the model architecture - Building the model using deeplearning approach and adding CNN layers	4	High	Jeyasree, Jeyaroshini
Sprint-2		USN-6	Training , saving, testing and predicting the model	4	High	Felcia, Madhuvarshni
Sprint-3	HTML Page Creation	USN-7	Home page creation - It shows options of the application	4	Medium	Jeyasree, Jeyaroshini
Sprint-3		USN-8	User Input and Prediction Page Creation - It is for the user to feedthe input images and display predicted output	4	Medium	Felcia, Madhuvarshni
Sprint-4	Application Phase	USN-9	Building the python code and importing the flask module into theproject	10	High	Jeyasree, Jeyaroshini
Sprint-4		USN-10	Importing the flask module into the project and perform routingthe HTML pages	10	High	Felcia, Madhuvarshni
Sprint-4	Deployment Phase	USN-11	Cloud deployment – Deployment of application byusing IBM cloud	10	High	Jeyasree, Jeyaroshini
Sprint-4	Testing Phase	USN-12	Checking usability and accessibility and performance	10	High	Felcia, Madhuvarshni

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	12	6 Days	24 Oct 2022	29 Oct 2022		
Sprint-2	8	4 Days	30 Oct 2022	02 Nov 2022		
Sprint-3	8	3 Days	03 Nov 2022	05 Nov 2022		
Sprint-4	40	7 Days	06 Nov 2022	12 Nov 2022		

6.3 Reports from JIRA



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

7.1 Feature 1:

Preparation of DATASET for identification of fruits:

```
# Unzipping the dataset
!unzip '/content/Dataset.zip'

Image preprocessing
#Importing The ImageDataGenerator Library
from keras.preprocessing.image import ImageDataGenerator

#Configure ImageDataGenerator Class
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zo
om_range=0.2,horizontal_flip=True)      test_datagen=ImageDataGenerator(rescale=1./255)

#Applying Image DataGenerator Functionality To Trainset And Testset

x_train = train_datagen.flow_from_directory(
r'/content/Dataset/TRAIN_SET',
target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='spars    e')
#Applying Image DataGenerator Functionality To Testset
x_test = test_datagen.flow_from_directory(
 r'/content/Dataset/TEST_SET',
target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='spars    e')

#checking the number of classes
print(x_train.class_indices)
from collections import Counter as c
```

```
c(x_train .labels)
```

Feature 2:
Accuracy using CNN

```
# Initializing the CNN
classifier = Sequential()

# First convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activatio       n='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Second convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), activation='relu'))

# input_shape is going to be the pooled feature maps from the previo       us convolution layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Flattening the layers
classifier.add(Flatten())
```

Feature 3:
Usage of dense layers

```
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=5, activation='softmax'))
```

8. TESTING

8.1 Test Cases

Section	TotalCases	Not Tested	Fail	Pass
PrintEngine	0	0	0	0
ClientApplication	51	0	0	51
Security	2	0	0	2
ExceptionReporting	9	0	0	9
FinalReportOutput	4	0	0	4

8.2 User Acceptance Testing

Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	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'tFix	0	5	2	1	8
Totals	24	14	13	26	77

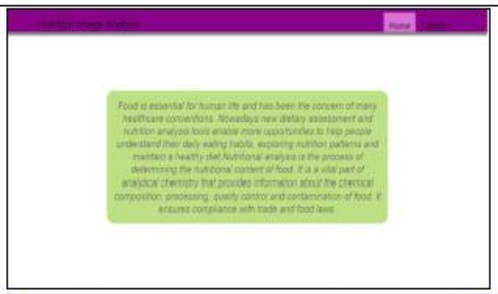
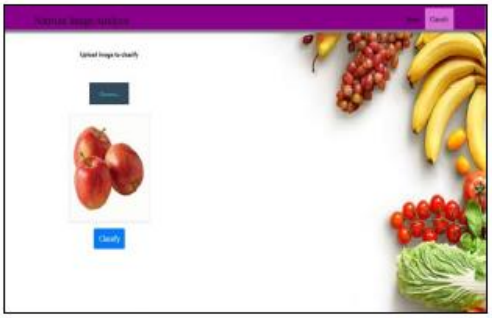
TestCaseAnalysis

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

Section	TotalCases	Not Tested	Fail	Pass
PrintEngine	0	0	0	0
ClientApplication	51	0	0	51
Security	2	0	0	2
ExceptionReporting	9	0	0	9
FinalReportOutput	4	0	0	4

8. RESULTS

8.1 Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Model Summary		  
2.	Accuracy	Training Accuracy – 90% Validation Accuracy -89%	/usr/local/lib/python3.7/dist-packages/ipyke """Entry point for launching an IPython ke Epoch 1/20 129/129 [==============================] - 2 Epoch 2/20 129/129 [==============================] - 3 Epoch 3/20 129/129 [==============================] - 3 Epoch 4/20 129/129 [==============================] - 3 Epoch 5/20 129/129 [==============================] - 3

9. ADVANTAGES & DISADVANTAGES

Advantages:

- Time-efficient
- easy to use
- instant nutrient information is displayed

Disadvantages:

- Dataset might be faulty
- Image might not be captured properly.

10. CONCLUSION

In this paper, we have developed a practical food recognition system for nutritional analysis. The key technique in this paper includes: the CNN-based food image recognition algorithms. The image is captured by the user and the corresponding nutritional information is displayed to the user instantly.

11. FUTURE SCOPE

The project can be further developed to include all kinds of food items and also provide a calorie estimation at the end of each week. The app can also suggest to either include or exclude a particular nutrient in the daily diet of the user.

12. APPENDIX

Source Code

Data Collection

Download the dataset here

```
[ ]  # Unzipping the dataset
     !unzip '/content/Dataset.zip'
```

Image Preprocessing

```
[ ]  #Importing The ImageDataGenerator Library
     from keras.preprocessing.image import ImageDataGenerator
```

Image Data Augmentation

```
[ ]  #Configure ImageDataGenerator Class
     train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_
     test_datagen=ImageDataGenerator(rescale=1./255)
```

Applying Image DataGenerator Functionality To Train And Testset

```
[ ]  #Applying Image DataGenerator Functionality To Trainset And Testset
     x_train = train_datagen.flow_from_directory(
         r'/content/Dataset/TRAIN_SET',
         target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse'
     #Applying Image DataGenerator Functionality To Testset
     x_test = test_datagen.flow_from_directory(
         r'/content/Dataset/TEST_SET',
         target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse'
```

```
Found 4118 images belonging to 5 classes.
Found 929 images belonging to 5 classes.
```

```
#checking the number of classes
print(x_train.class_indices)
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

```
#checking the number of classes
print(x_test.class_indices)
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

```
from collections import Counter as c
c(x_train .labels)
```

```
Counter({0: 995, 1: 1354, 2: 1019, 3: 275, 4: 475})
```

Model Building

1. Importing The Model Building Libraries

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout
```

2. Initializing The Model

```
model = Sequential()
```

3. Adding CNN Layers

```
[ ]  # Initializing the CNN
     classifier = Sequential()

     # First convolution layer and pooling
     classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation=
     classifier.add(MaxPooling2D(pool_size=(2, 2)))

     # Second convolution layer and pooling
     classifier.add(Conv2D(32, (3, 3), activation='relu'))

     # input_shape is going to be the pooled feature maps from the previous
     classifier.add(MaxPooling2D(pool_size=(2, 2)))

     # Flattening the layers
     classifier.add(Flatten())
```

4. Adding Dense Layers

```
[ ]  classifier.add(Dense(units=128, activation='relu'))
     classifier.add(Dense(units=5, activation='softmax'))
```

```
[ ]  #summary of our model
     classifier.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten (Flatten)           (None, 6272)              0
```

5. Configure The Learning Process

```
[ ]  # Compiling the CNN
     # categorical_crossentropy for more than 2
     classifier.compile(optimizer='adam', loss='sparse_categorical_crossentr
```

6. Train The Model

```
[ ]  #Fitting the model
     classifier.fit_generator(generator=x_train,steps_per_epoch = len(x_trai
```

7. Saving The Model

```
[ ]  classifier.save('nutrition.h5')
```

8. Testing The Model

```
[ ]  #Predict the results
     from tensorflow.keras.models import load_model
     from keras.preprocessing import image
     model = load_model("nutrition.h5")
```

```
from tensorflow.keras.utils import img_to_array
#loading of the image
img = load_img(r'/content/Sample_Images/Test_Image1.jpg',grayscale=Fals
#image to array
x = img_to_array(img)
#changing the shape
x = np.expand_dims(x,axis = 0)
predict_x=model.predict(x)
classes_x=np.argmax(predict_x,axis=-1)
classes_x
```

```
1/1 [==============================] - 0s 18ms/step
array([0])
```

```
[ ] index=['APPLES', 'BANANA', 'ORANGE','PINEAPPLE','WATERMELON']
    result=str(index[classes_x[0]])
    result
```

```
'APPLES'
```

Dataset

https://drive.google.com/file/d/1QuloJ3EXnUTrP-jV6rJu3Y0xeR9QXkIF/view?usp=share_link

https://drive.google.com/file/d/1Ml19s8nVw2VZ7SphCVYH9ez4agwLPCZq/view?usp=share_link

https://drive.google.com/file/d/1GqYkgQdG8pakGK6dj0nE-uOw2926t5n9/view?usp=share_link

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-36193-1660293378

Project Demo Link

https://drive.google.com/file/d/1qTQ0VR8EUJqSgIYZxGGPAiP9oRCStjLc/view?usp=share_link