OBESITY DETECTION AND DIETARY ANALOGY USING MACHINE LEARNING APPROACH

A Major Project Report Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Mr. Ch Vinay Sai Reddy (18071A0566)
Miss. Lasya Kilari (18071A0586)
Mr. Venigandla Ruthwik Sai (18071A05B8)
Mr. Vishlavath Bhargav Aniketh (18071A05B9)
Mr. Yalakaturi Venkata Varun (18071A05C0)

Under the Guidance of

Dr. C. Kiran Mai

(Professor, Department of CSE, VNR VJIET)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses
Approved by AICTE, New Delhi, Affiliated to JNTUH
Recognized as "College with Potential for Excellence" by UGC
ISO 9001:2015 Certified, QS I GUAGE Diamond Rated
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING and TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade

NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses

Approved by AICTE, New Delhi, Affiliated to JNTUH

Recognized as "College with Potential for Excellence" by UGC

ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "**OBESITY DETECTION AND DIETARY ANALOGY USING MACHINE LEARNING**" is a bonafide work done under our supervision and is being submitted by **Mr. Ch Vinay Sai Reddy (18071A0566), Miss. Lasya Kilari (18071A0586), Mr. Venigandla Ruthwik Sai (18071A05B8), Mr. Vishlavath Bhargav Aniketh (18071A05B9), Mr. Yalakaturi Venkata Varun (18071A05C0)** in partial fulfilment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, of the VNRVJIET, Hyderabad during the academic year 2021-2022.

Certified further that to the best of our knowledge the work presented in this thesis has not been submitted to any other University or Institute for the award of any Degree or Diploma.

Dr. C. Kiran Mai
Professor &Internal Guide
CSE Department, VNR VJIET

Dr. S Nagini
Professor & HOD
CSE Department, VNR VJIET

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING and TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade

NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses

Approved by AICTE, New Delhi, Affiliated to JNTUH

Recognized as "College with Potential for Excellence" by UGC

ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We declare that the major project work entitled "**OBESITY DETECTION AND DIETARY ANALOGY USING MACHINE LEARNING**" submitted in the department of Computer Science and Engineering, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** is a bonafide record of our own work carriedout under the supervision of **Dr. C. Kiran Mai, Professor, Department of CSE, VNRVJIET.** Also, we declare that the matter embodied in thisthesis has not been submitted by us in full or in any part thereof for the award of anydegree/diploma of any other institution or university previously.

Place: Hyderabad.

Ch Vinay Sai Reddy	**Lasya Kilari**	**V Ruthwik Sai**	**V Bhargav Aniketh**	**Y V Varun**
(18071A0566)	**(**18071A0586)	(18071A05B8)	(18071A05B9)	(18071A05C0)

ACKNOWLEDGEMENT

Firstly, we would like to express our immense gratitude towards our institution VNR Vignana Jyothi Institute of Engineering and Technology, which created a great platform to attain profound technical skills in the field of Computer Science, thereby fulfilling our most cherished goal.

We are very much thankful to our Principal**, Dr. Challa Dhanunjaya Naidu** and our Head of Department, **Dr. S. Nagini,** for extending their cooperation in doing this project in stipulated time.

We extend our heartfelt thanks to our guide**, Dr. C. Kiran Mai** and the project coordinators **Mrs. K. Hari Priya** and **Dr. P. V. Siva Kumar,** for their enthusiastic guidance throughout the course of our project.

Last but not least, our appreciable obligation also goes to all the staff members of Computer Science & Engineering department and to our fellow classmates who directly or indirectly helped us.

Mr. Ch Vinay Sai Reddy (18071A0566)
Miss. Lasya Kilari (18071A0586)
Mr. V. Ruthwik Sai (18071A05B8)
Mr. V. Bhargav Aniketh (18071A05B9)
Mr. Y. V. Varun (18071A05C0)

ABSTRACT

Obesity is a disease characterized by the accumulation of excessive amounts of body fat. Apart from being a mere cosmetic concern, obesity has serious health ramifications. Obesity dramatically increases an individual's risk for a wide range of health problems such as heart disease, diabetes, high blood pressure, and certain cancers. Getting obesity in control is crucial to ensuring a healthy and disease-free lifestyle. Obesity can be caused due to numerous factors including genetic and hereditary disorders. In the case of obesity being genetic, it is much more difficult for individuals to lose weight.

This Obesity Detection and Diet Control Android app has been developed to help individuals suffering from obesity with bringing their weight under control. There are many applications that help an individual. This system accounts for various factors such as the user's BMI, Age, Gender, History of genetic diseases, any allergies they might be suffering from, their preferred activity rate, etc. Based on these factors, the individuals receive a diet plan that is tailored specifically to their needs. This android application is built using Android Studio and JAVA as a programming language. There are many applications that suggest diets to be consumed to reduce or maintain weight, but many of those are with limitations that are to be dealt with to build an efficient system to better recommend diets and measures and help people. Some limitations of these existing apps are that they can't be customizable according to our concerns on health and our allergies towards some particular food. Most generalized diet plans fail to consider genetic or hereditary disorders, apart from diet alone, one can do several exercises to lose weight efficiently, but the existing systems do not consolidate both diet and activities into a single plan. Many food habits that are considered regularly for weight maintenance might affect the health of an individual if he/she is allergic to it. Hence, those food items should have an alternative to consume.

The maintenance of existing apps is very difficult. Hence the proposed system is going to be built using multi-agent environment which takes various parameters like allergies, genetic history into consideration which helps in giving efficient suggestions and recommendations to people using the system.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Overweight and obesity are defined as abnormal or excessive fat accumulation that may impact health. Body mass index (BMI) is a simple index of weight-for-height that is commonly used to classify overweight and obesity in adults. It is defined as a person's weight in kilograms divided by the square of his height in meters (kg/m2). In 2019, an estimated 38.2 million children under the age of 5 years were overweight or obese. Globally there are more people who are obese than underweight – this occurs in every region except parts of sub-Saharan Africa and Asia. The fundamental cause of obesity and overweight is an energy imbalance between calories consumed and calories expended. Globally, there has been an increased intake of energy-dense foods that are high in fat and sugars; and an increase in physical inactivity due to the increasingly sedentary nature of many forms of work, changing modes of transportation, and increasing urbanization. Early Detection of Obesity can help individuals from suffering from major health problems and also in changing their food habits which can lead to reducing this risk. For this, an android application that helps in detecting obesity using BMI and suggesting food diet plans, explaining the health diseases that can be at risk, and suggesting certain exercises specific for reducing further risks for different BMI is built using Artificial Intelligence which works in a multi-agent environment taking various parameters into consideration that better help in recommending diets and appropriate measures hence acting as an efficient system. If an individual’s obesity condition is due to their genetic health, reducing calorie intake doesn’t work. For those people, the diet would be recommended by adding health conditions as an attribute and not just by the body pattern. In this, a database is created, and food preferences are given in accordance with Indian cuisine, and all allergies are also taken into consideration while making food recommendations. Also, apart from weight-based problems, during the course, the individual may have skin shrinking problems which are also taken into consideration and could be cured by following a particular diet. Alarms and notifications can be set to keep track and follow the diet.

CHAPTER 2
OBJECTIVE

Obesity is a disorder that could be caused due to a numerous factors including genetic disorders. Individuals suffering from obesity often have a tough time losing weight due to which it becomes easy to get disheartened and have trouble sticking to a diet plan or following an exercise regimen. To effectively bring a person's weight under control, it is imperative that we consider the genetic and physiological disorders they might suffer from. This Obesity Detection and Diet Control Android app helps provide a specialized diet and exercise plan to users that suits their needs and takes into consideration the disorders and allergies that they might be suffering from Along with providing a diet plan, this system also helps users consult with a doctor to alter or tweak their diet or exercise.

CHAPTER 3
LITERARY SURVEY

PAPER-1

Study on the Principles of the Intelligent Diet Arrangement System Based on Multi-Agent

AUTHORS:

Fule Wang, Yuan Yuan, Yu Pan, Bin Hu

ABSTRACT:

Based on the individualized requirements of the customers as regards to reasonable diet, diet medication and so on, this article simulates the nutritionist's experience by artificial intelligence, and uses Internet as the technical platform, so as to introduce to customers a scientifically and perfectly structured menu or diet medication plan based on their preference and health status by using actual-time recognition, collocation and integration. According to the complexion of the process of intelligent diet arrangement, the system referring in the paper adopts the Genetic Algorithm as the key algorithm. For increasing the efficiency of process, the system, this paper improves the Genetic Algorithm and makes adaptability perfect in the process of iterative computation, the intelligent diet management system based on Multi-agent is preliminarily achieved.

METHODOLOGY:

The Multi-agent theory and method have been applied to the catering algorithm in this paper. Data pre-processing, encoding, and nutrition control variables have been designed to improve the programmable intelligence catering process, and fitness function has been optimised.

LIMITATIONS:

Few more factors could have been taken into consideration for more accurate diet recommendation.

PAPER-2

A DASH Diet Recommendation System for Hypertensive Patients Using Machine Learning

AUTHORS:

Romeshwar Sookrah, Jaysree Devee Dhowtal, Soulakshmee Devi Nagowah

ABSTRACT:

Hypertension is becoming a serious health issue in the world. People tend to have a busy lifestyle and to adopt unhealthy diets. Due to poor eating habits, the rate of Non Communicable Diseases (NCDs) such as hypertension together with the rate of death caused by such diseases are rising. In order to promote healthy eating habits in Mauritius, the paper proposes a DASH diet recommender system that recommends healthy Mauritian diet plans to hypertensive patients. The system consists of a recommendation engine that uses techniques such as content-based filtering along with machine learning algorithms to recommend personalised diet plans to hypertensive patients based on factors such as age, user preferences about food, allergies, smoking level, alcohol level, blood pressure level and dietary intake. The system makes use of a mobile application which is handy and quick to use. Based on a survey carried out, the application has helped users to control and reduce their BP level.

METHODOLOGY:

The recommended dishes aim to help not only a hypertensive person to control his diet but also benefit a normal user to prevent him from getting health complications. The system considers a number of factors such as allergies, BP level,age, weight, smoke/alcohol intake, dietary intake and food preferences to make proper recommendations based on machine learning and content-based filtering techniques.

LIMITATIONS:

Accurate dishes and how to mix different ingredients appropriately isn't taken into consideration.

PAPER- 3

A Food Recommender System Considering Nutritional Information and User Preferences

AUTHORS:

Raciel Year Toledo, Ahmad A.Alzahrani,Luis Martinez

ABSTRACT:

The World Health Organization identifies the overall increase of noncommunicable diseases as a major issue, such as premature heart diseases, diabetes, and cancer. Unhealthy diets have been identified as the important causing factor of such diseases. In this context, personalised nutrition emerges as a new research field for providing tailored food intake advices to individuals according to their physical, physiological data, and further personal information. Specifically, in the last few years, several types of research have proposed computational models for personalised food recommendation using nutritional knowledge and user data. This paper presents a general framework for daily meal plan recommendations, incorporating as main feature the simultaneous management of nutritional-aware and preference-aware information, in contrast to the previous works which lack this global viewpoint. The proposal incorporates a pre-filtering stage that uses AHPSort as a multi-criteria decision analysis tool for filtering out foods which are not appropriate to the current user characteristics. Furthermore, it incorporates an optimization-based stage for generating a daily meal plan whose goal is the recommendation of food highly preferred by the user, not consumed recently, and satisfying his/her daily nutritional requirements. A case study is developed for testing the performance of the recommender system.

METHODOLOGY:

DIETOS is a web-based recommender system for the adaptive delivery of nutrition contents to improve the quality of life of both healthy people and individuals affected by chronic diet-related diseases. The proposed system is able to build a users health profile, and provides individualised nutritional recommendation according to the health profile.

LIMITATIONS:

Other than diet recommendation to individuals for users health profile, no other measures are recommended, that have to be followed along with diet for user health.

PAPER- 4:

Personalized Food Recommendation Using Deep Neural Network

AUTHORS:

Tossawat Mokdara, Priyakorn Pusawiro, Jaturon Harnsomburana

ABSTRACT:

Making decisions about what and where to eat is a major problem in our everyday lives due to a wide variety of ingredients, culinary styles, ethnicities, cultures, and personal tastes. Choosing the right dish at the right time seems to be a very difficult task. Thus, this research proposes an integration of a deep neural network with a recommendation system with Thai food as our test domain. The proposed model extracts interested ingredients from the set of recipes of the user's favourite dishes that is given before using the system. The features are extracted from the analysis of favourite ingredients then a user profile is evaluated by a model of deep neural network(DNN). The system also collects the history of selected dishes along with the user profile in a database. The model will predict the next dishes using a temporal prediction model on the profile and eating history. The satisfaction is analysed through experiment based on whether the user selected the recommended dishes or not which is calculated as hit ratio. In addition, the accuracy and variety of recommendations are also analysed. From the experimental results, the DNN model can predict the user profile with precision up to 90% and the accuracy of hit ratio is up to 89%.

METHODOLOGY:

The model consists of 3 layers which are fully connected Rectified Linear Unit .The input layer of this system is ingredient vector which contains categories of ingredients. Values of them are between 0 and 1 which come from ingredients quantity in the recipe. The ingredients quantity contains many kinds of quantity measurements. Thus, the system will convert those units to a common measurement which is weight in grams. After that, the system will divide ingredients into groups then calculate a summation of all weights of each group and normalise the sum of the weight of each group into a single value between 0 and 1.

LIMITATIONS:

Does not take allergies and previous health and genetic history into consideration before making recommendations.

PAPER- 5

A machine learning approach for obesity risk prediction

AUTHORS:

Faria Ferdowsy, Kazi Samsul Alam Rahi, Md. Ismail Jabiullah, Md. Tarek Habib

ABSTRACT:

In modern times, obesity has become a significant threat all over the world. Obesity means an unnatural or excessive amount of fat that is present in our bodies. People are constantly moving towards an unhealthy lifestyle, eating excessive junk food, late-night sleep, spend a long time sitting down. Adolescents are being affected because of their unconscious attitudes. It is a medical problem known as a very complex disease. It promotes the spread of complex illnesses, stroke, heart disease, liver cancer. Consequently, as an aware multitude of Bangladesh, we have to move forward to prevent this risk of obesity. The purpose of this paper is to move towards a machine-learning-based pathway for predicting the risk of obesity using machine-learning algorithms. The great thing about this paper is that people will know the risk of obesity and the reasons behind their obesity. We collect more than 1100 data from many varieties of people of different ages and collect information from both are suffering obesity and non-obesity. For this research, we apply nine prominent machine learning algorithms. We used the algorithm of k-nearest neighbour (k-NN), random forest, logistic regression, multilayer perceptron (MLP), support vector machine (SVM), naïve Bayes, adaptive boosting (ADA boosting), decision tree, and gradient boosting classifier, and we have measured the performance of each of these classifications in terms of some prominent performance metrics. From the experimental results, we determine the obesity of high, medium, and low. The Logistic Regression Algorithm achieves the highest accuracy of 97.09% as compared to the other classifiers. In addition, the gradient boosting algorithm gave the poorest accuracy of 64.08% as well as the lowest metric values.

METHODOLOGY-:

In this application, certain inputs will be taken from the user through the fill up method. The result will be resolved, dependent on the user input by deploying a machine learning classifier of logistic regression algorithm to the processed information.

LIMITATIONS:

Genetic History is not considered as a feature in the prediction which also stands to be one of the possible vital reasons for obesity.

PAPER-6

Analysis and Design for Food Planning Mobile Application

AUTHORS:

Natalia Chandra, Sonia Rapinta Manalu, Afan Galih Salman

ABSTRACT:

This study builds the system that allow users to plan their food consumption via their mobile phones.The system help user to manage and tracking history of their food consumption, choosing food that is suitable for their health, and help user to select their favorite restaurant. Methodology used in this research are analysis method, design method, and literature study. The result is a food planning mobile application based on iOS and Android platforms that help user to manage and track their food consumption, to calculate & choose balanced food that is suitable for their body.

METHODOLOGY:

Front end is intended for IOS and Android users who want to set their diet and the back end is intended for administrators, restaurants, and a nutritionist. While nutrition experts will assess the food menu has been submitted. Administrator in charge of organising the user approval for restaurant and nutritionists who have registered on this system. Finally the user can view recommendations and suggestions.

LIMITATIONS:

Not complete accuracy as energy spent by users and any disease history is not taken into consideration through input for providing recommendations.

PAPER- 7

Smart diet: A Personal Diet Consultant for Healthy Meal Planning.

AUTHORS:

Jen-Hao Hsiao and Henry Chang

ABSTRACT:

Effective personal dietary guidelines are essential for health management and preventing chronic diseases. The objective of this research is to achieve nutrient-balanced food recommendations for each individual, while considering individual's requirements at the same time. To reach this goal, we developed a location-aware interactive diet consultant named SmartDiet based on multi-objective optimization. The proposed personalized diet planning approach not only translates nutrient recommendations into realistic dish choices, but also accepts feedback from users to fine-tune their meal plans. The results showed that daily nutrition needs can be fulfilled by the designated meals, and the interactive diet planning scheme helps a user adjust the plan in an easier way. The guidelines generated by SmartDiet are expected to potentially improve the overall health and reduce the risk of chronic diseases of individuals.

METHODOLOGY:

Multi-objective diet selection method according to its knowledge engine,which stores the nutrition constraints for different subjects (e.g., normal people or diabetic patients), and the historical intake data, which records the user's historical diet habits. Once complete, a combination of available dishes and restaurants that satisfies the daily nutritional requirements and user's current locality will be sent to the user.

LIMITATIONS:

No health hazards are discussed with the user due to inefficient diet intake and no measures to prevent the same are discussed.

PAPER-8

Nutrition Diet Recommendation System Using User's Interest

AUTHORS:

Butti.Gouthami, Malige Gangappa

ABSTRACT:

Diet Recommendation system can be expected for a better solution for people's healthy eating habits. Due to a busy lifestyle, healthy eating habits are overlooked. With these unhealthy eating habits, we get health problems. This system provides functionalities to change their eating behavior in positive ways. Healthy living is the biggest factor these days. A small change can have a big impact on our health. The system is constructed on the USDA dataset, grocery data, consumed food, and user's Quetelet Index (BMI). Most recommendations suggest a proper diet plan for individual users' Quetelet Index (BMI value), and food consumed that day and our system suggests the remaining nutrition food suggestions to fill that day's dietary food recommendations. The body Quetelet Index (BMI) is used to assess the user's body fat. The user's diet recommendations are varied with the interested food of the user. Grocery data is varied with continuous data collection given by the user. This application will help users to structure diet recommendations according to various individual factors which include food suggestions, nutrition, deficiencies, Quetelet Index (BMI), and tracking of his food habits of the user.

METHODOLOGY:

USDA nutrient dataset will be used in calculating the recommended diet for the user. Grocery data set consisting of a user choice of food as intake. USDA database is responsible for maintaining nutrition factors for every individual food item. The input values will be based on the USDA id for every 100 grams. The values required in calculating BMI (body mass index) need to be given as an input which will be utilized in calculating the final diet recommendations. The second input while calculating the diet recommended for the user is based on the consumed food for that day. While calculating the diet recommendation, initially the deficit nutrition is calculated based on the food consumed for that day and sorted the input nutrients dataset based on the BMI value, and the deficit food will be filled from the sorted grocery dataset. The obesity parameter is used for food recommendations. According to obesity, the diet recommendations are calculated to control fat levels.

LIMITATIONS:

Does not different parameters like genetic history, allergies into consideration. Does not recommend preventive measures to end-users.

PAPER-9

Mobile Apps for Weight Management

AUTHORS:

Drishti P. Ghelani, Negar Naderpoor

ABSTRACT:

Over the last decade, mobile technology has emerged as a potentially useful platform to facilitate weight management and tackle the current obesity epidemic. Clinicians are being more frequently asked to give advice about the usefulness of mobile apps and many individuals have already integrated apps into their attempts to manage weight. A number of newly published studies have demonstrated promising results of mobile-based interventions for weight management across different populations, but the extent of their effectiveness remains widely debated. This narrative literature review synthesizes the latest evidence, primarily from randomized controlled trials (RCTs), regarding the clinical use of mobile applications for weight management, as well as highlight key limitations associated with their use and directions for future research and practice.

METHODOLOGY:

The dietary routine of an individual is asked where the user enters their details. BMI would be calculated based on weight and height. Now nutrition intake would be calculated and a food chart is suggested based on it. They get some choices to choose among the given options.

LIMITATIONS:

Monitoring diet is difficult, accounting for a high proportion of the attrition as each item needs to be individually entered into the application, and a greater variety in the diet requires more information to be entered, thus taking up more time.

Too little choice of pre-entered meals in their databases, which often do not include ethnic foods.

PAPER-10

Online Recommender System for Personalized Nutrition Advice

AUTHORS:

Rodrigo Zenun Franco

ABSTRACT:

The general recommendations for addressing non-communicable diseases, which are responsible for two-thirds of deaths globally, are mainly related to lifestyle changes, such as diet and physical activity. Challenges with encouraging healthy diets include gathering accurate information about dietary intake and delivering interventions that can influence behavior. Internet technologies offer excellent potential for addressing these challenges. Furthermore, personalized recommendations are more effective than general population-based recommendations at modifying health-related behavior in nutrition interventions.

The overall aim of this project is to design, develop and evaluate a recommender system able to assess dietary intake, using a validated Food Frequency Questionnaire (FFQ), and propose valid personalized nutrition advice for adults. It is investigating an effective way of providing personalized online dietary recommendations to increase diet quality at the population level and of considering an individual user's preferences, population data, and experts' knowledge in the recommendation. The system is envisaged to be a web-based service, built with commercially available technologies, scalable, replicable, inexpensive, and independent of any bespoke device (e.g. proprietary activity trackers).

METHODOLOGY:

It asks users to complete an FFQ (Questionnaire), converts the dietary intake into nutrients, calculates the components of the AHEI (e.g. vegetables, fruits, and sugar), derives a personalized nutrition recommendation based on the AHEI score, and presents the recommendation. It also calculates the ideal weight range of the participants, based on the Body Mass Index (BMI), and provides feedback on their physical activity level, based on the Baecke questionnaire.

LIMITATIONS:

It takes the information from the user and might not work effectively if the user doesn't have clear knowledge or information that is being asked.

The model only considers BMI, Physical activities separately but doesn’t take other parameters into consideration.

SNO	TITLE & AUTHORS	METHODOLOGY	LIMITATIONS
1	Study on the Principles of the Intelligent Diet Arrangement System Based on Multi-Agent-Fule Wang, Yuan Yuan, Yu Pan, Bin Hu	The Multi-agent theory and method have been applied to the catering algorithm in this paper. Data pre-processing, encoding, and nutrition control variables have been designed to improve the programmable intelligence catering process, and fitness function has been optimized.	Few more factors could have been taken into consideration for a more accurate diet recommendations.
2	A DASH Diet Recommendation System for Hypertensive Patients Using Machine Learning-Romeshwar Sookrah, Jaysree Devee Dhowtal, Soulakshmee Devi Nagowah	The recommended dishes aim to help not only a hypertensive person to control his diet but also benefit a normal user to prevent him from getting health complications. The system considers a number of factors such as allergies, BP level,age, weight, smoke/alcohol intake, dietary intake, and food preferences to make proper recommendations based on machine learning and content-based filtering techniques.	Accurate dishes and how to mix different ingredients appropriately aren't into consideration.
3	A Food Recommender System Considering Nutritional Information and User Preferences-Raciel Year Toledo, Ahmad A.Alzahrani, Luis Martinez.	DIETOS is a web-based recommender system for the adaptive delivery of nutrition content to improve the quality of life of both healthy people and individuals affected by chronic diet-related diseases. The proposed system is able to build a user's health profile and provides individualized nutritional recommendations according to the health profile.	Other than diet recommendations to individuals for users' health profiles, no other measures are recommended, that have to be followed along with diet for the user health.

4	Personalized Food Recommendation Using Deep Neural Network- Tossawat Mokdara, Priyakorn Pusawiro, Jaturon Harnsomburana	The model consists of 3 layers which are fully connected Rectified Linear Unit .The input layer of this system is ingredient vector which contains categories of ingredients. Values of them are between 0 and 1 which come from ingredients quantity in the recipe. The ingredients quantity contains many kinds of quantity measurements. Thus, the system will convert those units to a common measurement which is weight in gram. After that, the system will divide ingredients into group then calculate a summation of all weights of each group and normalize sum of the weight of each group into a single value between 0 and 1.	Does not take allergies and previous health and genetic history into consideration before making recommendations.
5	A machine Learning approach for Obesity risk Prediction-Faria Ferdowsy,Md.Tarek Habib	In this application, certain inputs will be taken from user through fill up method. The result will be resolved, dependent on the user input by deploying a machine learning classifier of logistic regression algorithm to the processed information.	Genetic History is not considered as a feature in the prediction which also stands to be one of the possible vital reasons for obesity.
6	Analysis and Design for Food Planning Mobile Application- Natalia Chandra, Sonia Rapinta Manalu, Afan Galih Salman	Front end is intended for IOS and Android users who want to set their diet and the back end is intended for administrators, restaurants, and a nutritionist. While nutrition experts will assess the food menu has been submitted. Administrator in charge of organising the user approval for restaurant and nutritionists who have registered on this system. Finally user can view recommendations and suggestions.	Not complete accuracy as energy spent by users and any disease history is not taken into consideration through input for providing recommendations.

7	Smart Diet: A personal diet consultant for healthy meal planning.- Jen-Hao Hsiao ; Henry Chang	Multi-objective diet selection method according to its knowledge engine, which stores the nutrition constraints for different subjects (e.g., normal people or diabetic patients), and the historical intake data, which records the user's historical diet habits. Once complete, a combination of available dishes and restaurants that satisfies the daily nutritional requirements and user's current locality will send to user.	No health hazards are discussed with the user due to inefficient diet intake and no measures to prevent the same are discussed.
8	Nutrition Diet Recommendation System Using User's Interest, Butti.Gouthami, Malige Gangappa	Databases are created where they provide nutrition values. Obesity is the main concern and diet is recommeded such that nutrient value doesn't exceed the daily intake based on the BMI.	Does not different parameters like genetic history, allergies into consideration. Does not recommend preventive measures to end-users.

9	Mobile Apps for Weight Management- Drishti P. Ghelani, Negar Naderpoor	The dietary routine of an individual is asked where the user enters their details. BMI would be calculated based on weight and height. Now nutrition intake would be calculated and a food chart is suggested based on it. They get some choices to choose among the given options.	Insufficient evidence for standalone reliance on application for complete weight management analysis and diet recommendation.
10	Online Recommender System for Personalized Nutrition Advice- Rodrigo Zenun Franco	It asks users to complete an FFQ (~~Questionare~~Questionnaire), converts the dietary intake into nutrients, calculates the components of the AHEI (e.g. vegetables, fruits and sugar), derives a personalized nutrition recommendation based on the AHEI score, and presents the recommendation. It also calculates the ideal weight range of the participants, based on the Body Mass Index (BMI), and provides feedback on their physical activity level, based on the Baecke questionnaire.	It takes the information from the user and might not work effectively if the user doesn't have clear knowledge or information that is being asked. The model only considers BMI, Physical activities separately but doesn't take other parameters into consideration.

CHAPTER 4
EXISTING SYSTEM

Most generalized diet plans fail to consider the genetic or hereditary disorders of an individuals. Due to this, many a times people suffering from obesity fail to get their weight under control even after following rigorous diets. It is not easy to get a diet plan that has been custom made to suit an individual's needs.

The main drawbacks of the existing system are

- Maintenance of the system is very difficult.
- There is a possibility for getting inaccurate results.
- User friendliness is very less.

CHAPTER 5
SYSTEM ANALYSIS

5.1 REQUIREMENT ANALYSIS

A condition or capability needed by a user to solve a problem or achieve an objective. The requirements are the descriptions of the system services and constraints. Requirements analysis is a very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types:

- Functional requirements
- Non-functional requirements

5.1.1 FUNCTIONAL REQUIREMENTS

5.1.1.1 DEFINITION

These are the requirements that the end-user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed, and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

5.1.1.2 SOFTWARE REQUIREMENTS

1. Laptop or PC

- Windows 7 or higher.
- Java
- Android Studio

2. Android Phone or Tablet

- Android v5.0 or Higher

5.1.2 NON-FUNCTIONAL REQUIREMENTS

5.1.2.1 DEFINITION

Non-functional requirements are characteristics or attributes of the system that can judge its operation. (or) Non-functional Specifications are the needs based on the specific criteria to evaluate the operation of the system. These requirements are collected and analyzed based on the client's needs and exceptions, security, working conditions, etc.

1. User Interface:
    a) The system shall maintain an easy-to-use interface across all functionalities and for all users
2. Scalability:
    a) The system shall be able to scale based on the number of users using the system.
3. Security:
    a) The administrative system should be protected from unauthorized access.
    b) The database should be protected from attacks and unauthorized access.
    c) The interface should be protected from attacks.
    d) All passwords should be stored as a secure hash of the administrator password.
4. Portability:
    a) The system should run on a variety of operating systems that support the Java language.
    b) The system should run on a variety of hardware.
5. Maintainability:
    a) The system should be easy to maintain.
    b) There should be a clear separation between the interface and the business logic code.
    c) There should be a clear separation between the data access objects that map the database and the business logic code.
6. Exception handling:
    a) Exceptions should be reported effectively to the user if they occur.
7. Ethics:
    a) The system shall not store or process any information about its users.

5.2 MODULES

The system comprises 2 major modules with their sub-modules as follows.

- User Module
- Doctor Module

User Module:

• Sign Up

• Sign In

• Profile

• Change Password

• Home

Options to Go Forward

A. Weight-based

B. Skin based

C. Existing Diet Plan (will not go forward if A or B is never chosen & a plan is not created)

A. Weight Based

- BMI (Age/Gender/Height/Weight)

- Genetic History [Choose from a list of Genetic Diseases/ Disorders (multi-select)]

- Allergies Choose from a list of Allergies (Allergies can be Food/ingredients/plants etc.) (multi-select)

- Activity Rate [Choose one out of 4 options (for e.g.:
None/Beginner/Moderate/Advanced) Under each option, we will give a short description
what it is.]

- SHOW BMI & BMR Level

• Diet Plan - Diet Plan

- Extra Ingredients/Foods based on BMI/Genetics - Activities you can do (Text-based)

- Disease Occurrence (Genetics/BMI based) & Prevention about it (Text-based)

B. Skin Based

1. Acne

2. Skin Shrinking

• Diet Plan

- Diet Plan (Diet Plan for the above can be static or criteria specified from the Food Database) - Activities you can do (Text-based)

C. Existing Diet Plan (Weight-based/Skin-based)

- View Diet plan & all the other things mentioned above

- Request for Customization (Request Status)

• Reminder (You will get a notification at that particular time)

- Add/Update/Delete (Plan-related or custom)

• Motivational Videos

- View motivation videos (type of videos & YouTube links from the database)

Doctor Module: (There will be only one fixed login for the doctor)

• Login

• Pending/Changed

- List of Pending Requests & change the diet plan

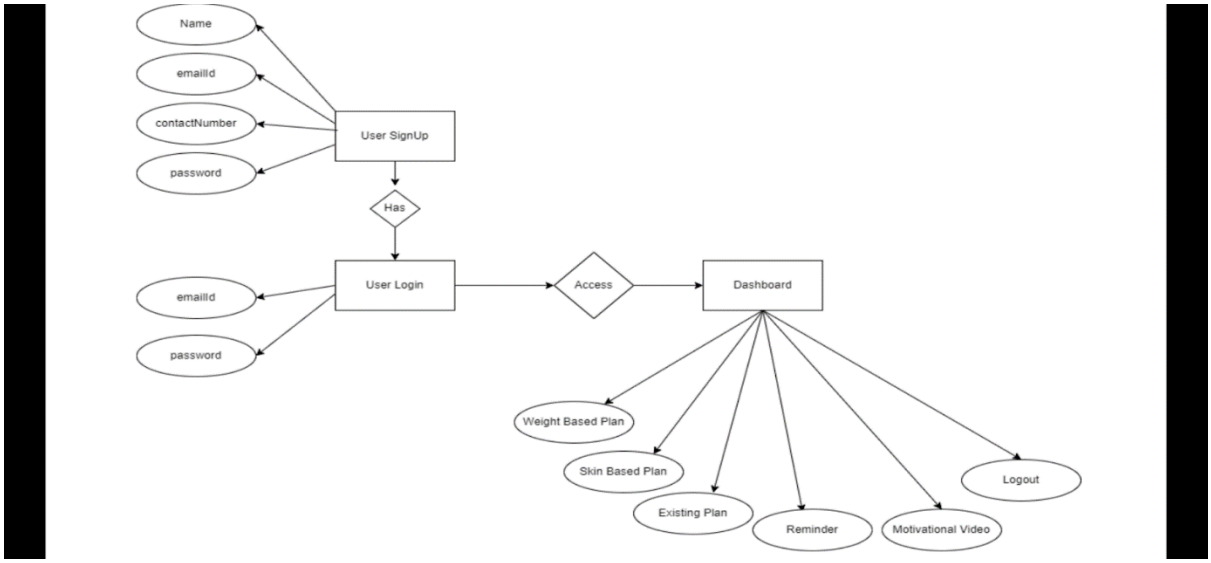
- List of Changed Diet Plans.

CHAPTER 6
SOFTWARE DESIGN

6.1 Diagrams

6.1.1 ER Diagram

An entity-relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities. ER Diagram is a type of flowchart that illustrates how “entities” such as people, objects, or concepts relate to each other within a system. They use a defined set of symbols such as rectangles, diamonds, ovals, and connecting lines to depict the interconnectedness of entities, relationships, and their attributes.



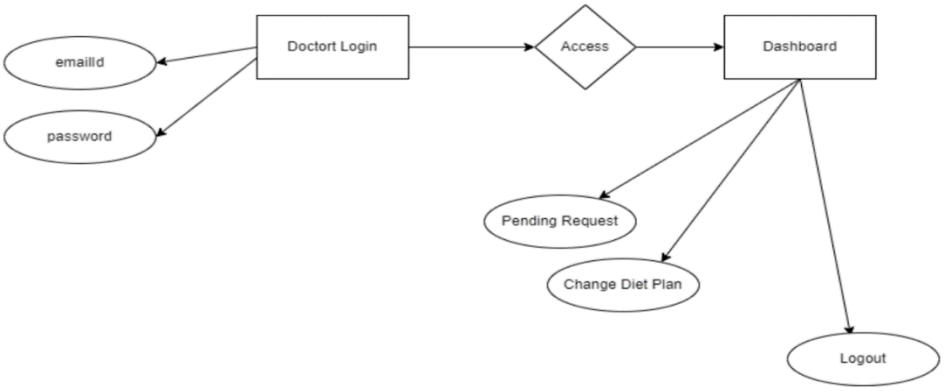


Fig 6.1.1.1: ER Diagram for User and Doctor

6.1.2 Use Case Diagram

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a website. The "actors" are people or entities operating under defined roles within the system. The following represents the use case diagram of the proposed system:

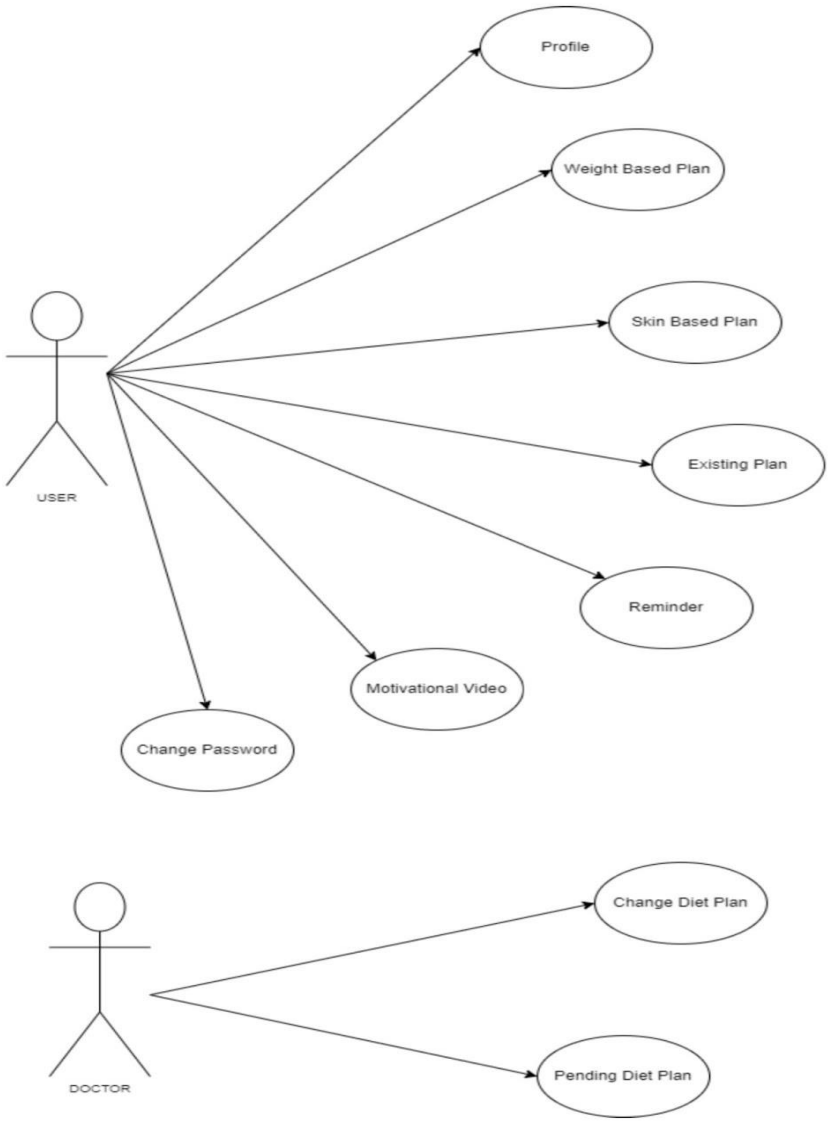
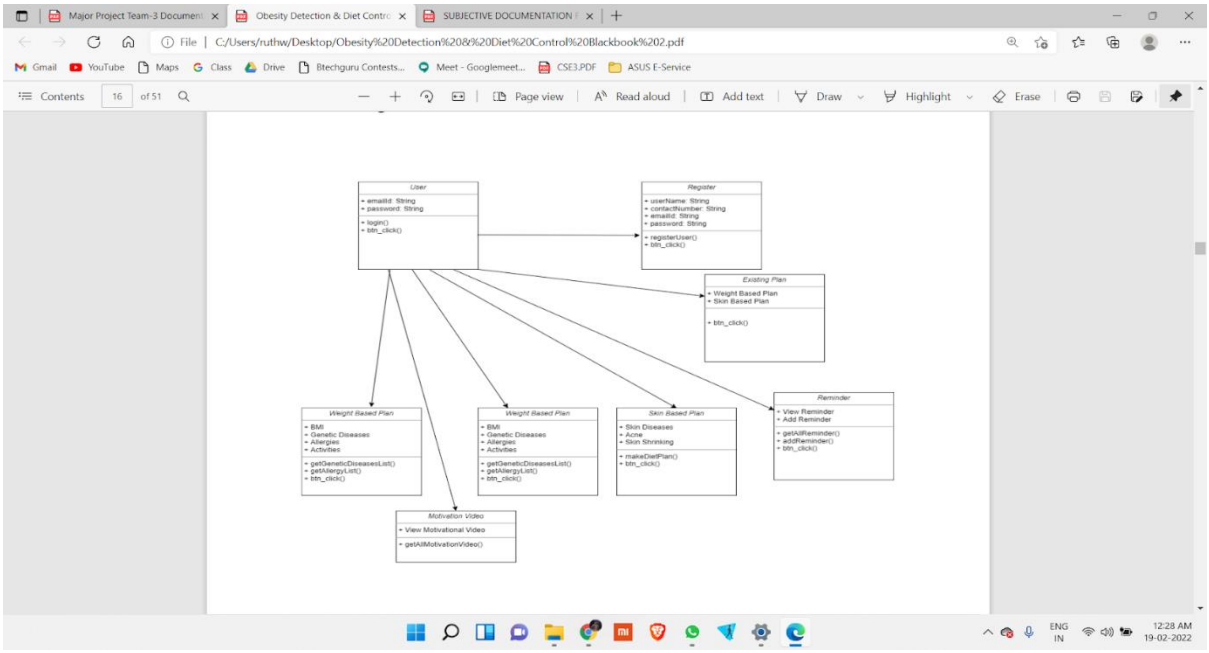


Fig 6.1.2.1: Use Case Diagrams for User and Doctor

6.1.3 Class Diagram

The class diagram is a static diagram. It represents the static view of an application. The class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. A class diagram describes the attributes and operations of a class and the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. The class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.



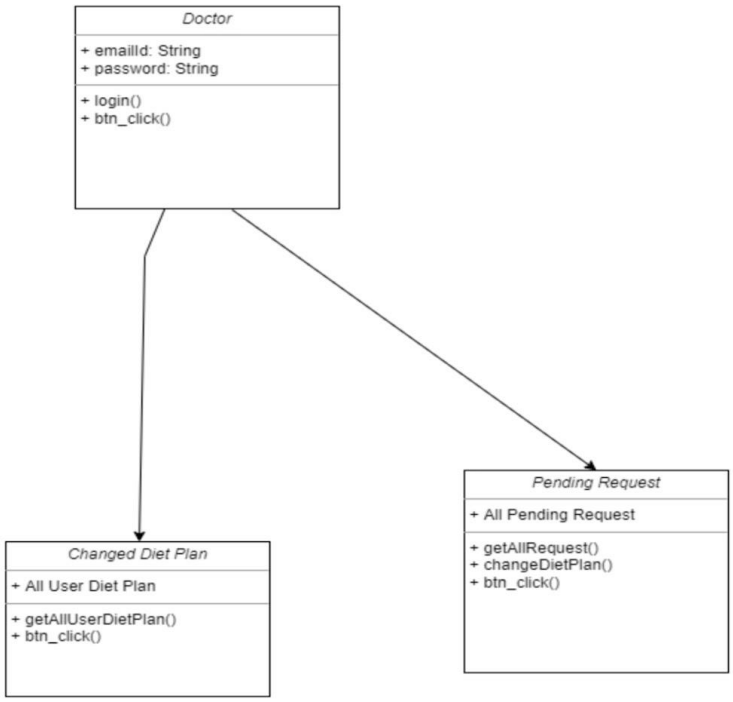


Fig 6.1.3.1: Class Diagrams for User and Doctor

6.1.4 Activity Diagram

An activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all types of flow control by using different elements such as fork, join, etc.

The basic purposes of activity diagrams are similar to the other four diagrams. It captures the dynamic behavior of the system. The other four diagrams are used to show the message flow from one object to another, but the activity diagram is used to show the message flow from one activity to another.

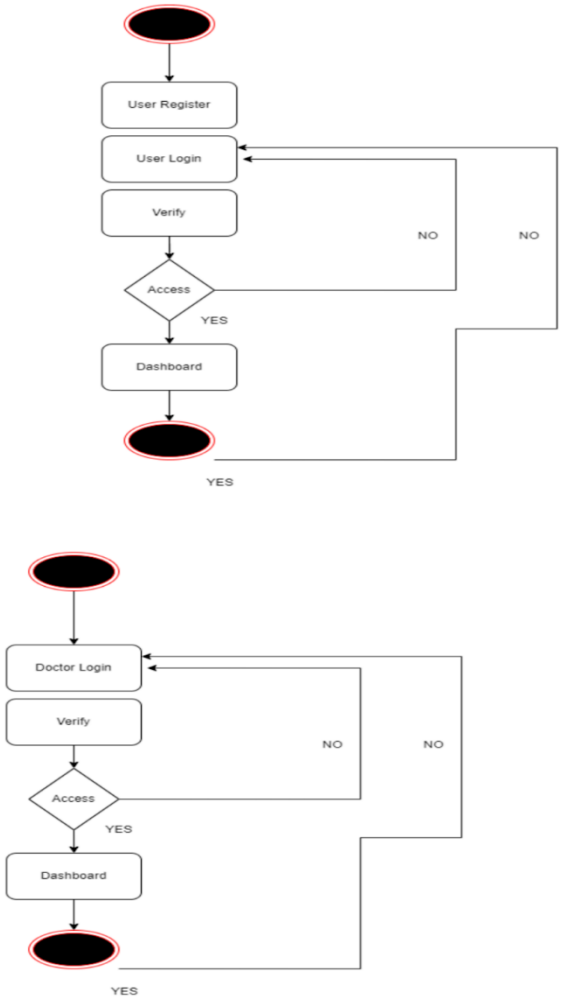


Fig 6.1.4.1: Activity Diagrams for User and Doctor

6.1.5 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

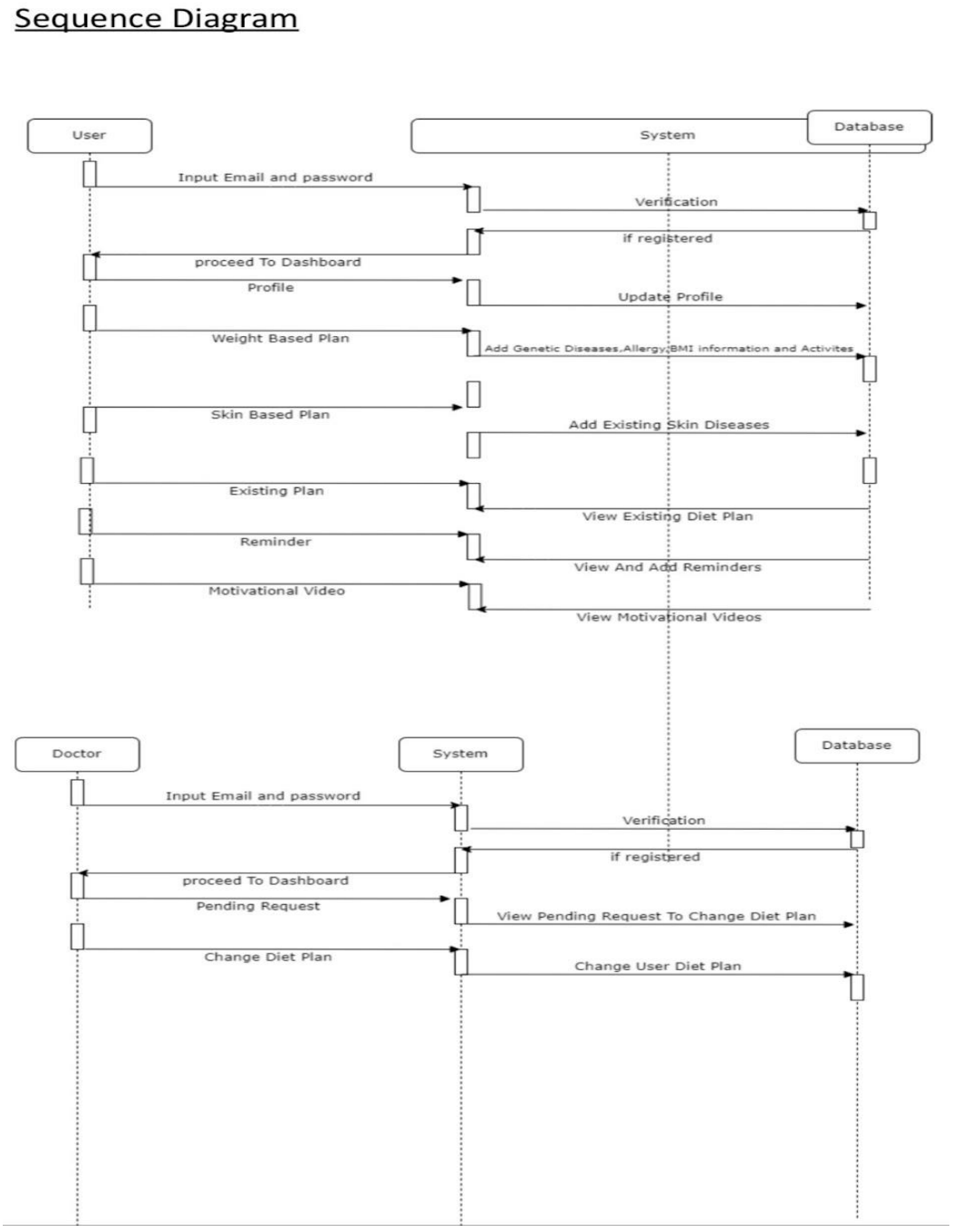


Fig 6.1.5.1: Sequence Diagrams for User and Doctor

CHAPTER 7
PROPOSED SYSTEM

- Just like a human dietician but with an addition, this system based on the Android operating system will also act like your device dietician while at the same time helping in detecting obesity through BMI calculator.
- When you go to a nutritionist, he/she will ask you details related to your body and health such as your age, height, weight and how much water you consume in a day, and how much you walk and work regularly. This artificial intelligent diet consultant will also ask you similar questions on your device through the app and you have to answer all these questions which will then generate a BMI specific to you that helps in understanding body culture and detecting early or existing obesity.
- Then this AI consultant will also advise you about what kinds of food you should intake in your diet, what you should ignore to keep yourself healthy via your diet, what health diseases you could possibly be prone to depending on your body culture, and how to prevent them, what exercises you need to follow and their importance.
- The Android application would be built using Java and all databases that contain data related to food ingredients, allergic food ingredients, preventive measures and exercises will be stored in SQL (the data related to food and diet recommendations are all taken in Indian cuisine consideration to make it more applicable to our end users). Google Cloud storage could help us access the data effectively building a larger network.
- All diet recommendations are made with detailed explanations on the nutritional content of the dish and the vitamin importance of every ingredient to make the end users better understand the significance of the diet that is being recommended. All the data is related through a multi agent environment which helps in making the application efficient and has a complete flow to make the user experience better. The food diet recommendations and preferences are refined regularly through AI to keep the application up to date to make the application efficient. Query optimization is used to best retrieve required data and help the end user by providing best recommendations.

- The Gradle build system has let us use the advanced features of android studio to build the application effective.

CHAPTER 8
IMPLEMENTATION

8.1 OVERVIEW OF TECHNOLOGIES USED

8.1.1 Introduction to Android

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

• A flexible Gradle-based build system

• A fast and feature-rich emulator

• A unified environment where you can develop for all Android devices

• Instant Run to push changes to your running app without building a new APK

• Code templates and GitHub integration to help you build common app features and import sample code • Extensive testing tools and frameworks

• Lint tools to catch performance, usability, version compatibility, and other problems

• C++ and NDK support

• Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

.

8.1.2 Project Structure

Each project in Android Studio contains one or more modules with source code files and resourses files. Types of modules include:

- Android app modules
- Library modules
- Google App Engine modules

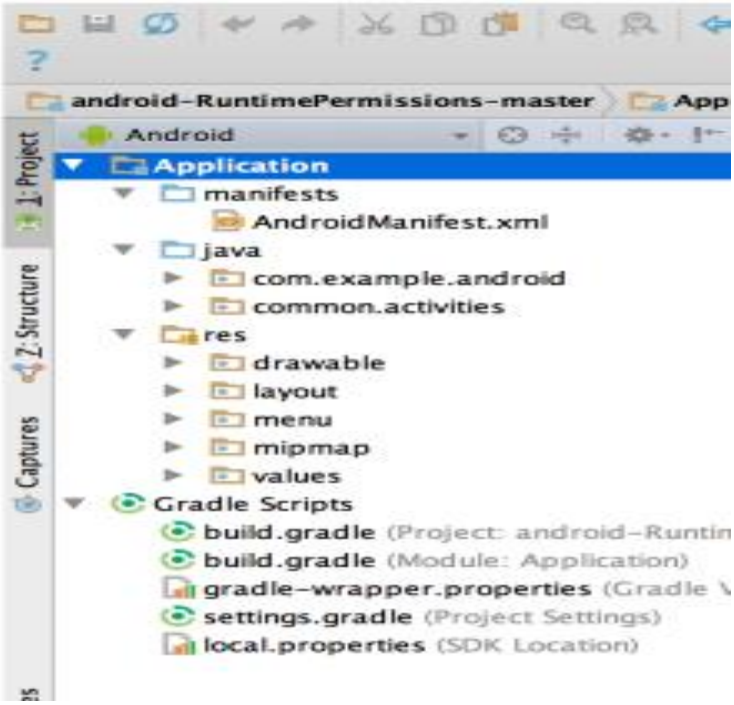


Fig 8.1.2.1: Project Structure

By default, Android Studio displays your project files in the Android project view, as shown in figure 1. This view is organized by modules to provide quick access to your project's key source files.

All the build files are visible at the top level under Gradle Scripts and each app module contains the following folders:

***Manifests**: Contains the AndroidManifest.xml file,

***Java:** Contains the lava source code files, Including Unit test code.

***Res:** Contains all non- code resources, such as XML layouts, UI strings, and

bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select Project from the Project dropdown (In figure, it's showing as Android).

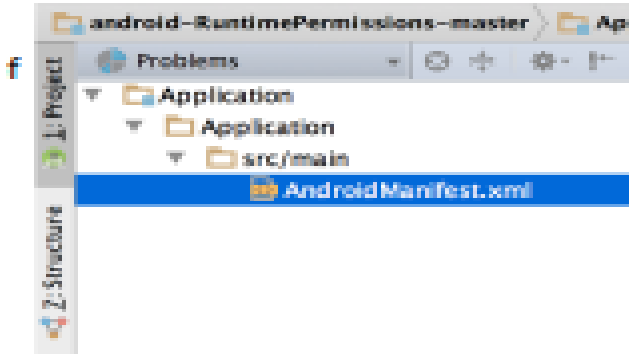


Fig 8.1.2.2: Project Dropdown

You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the Problems view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element closing tag in a layout file.

8.1.3 The User Interface:

1. The **toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
2. The **navigation bar** helps you navigate through your project and open filles for editing. It provides a more compact view of the structure visible in the Project windows.
3. The **editor window** is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
4. The **tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.

5. The **tool windows** give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.

6. The **status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

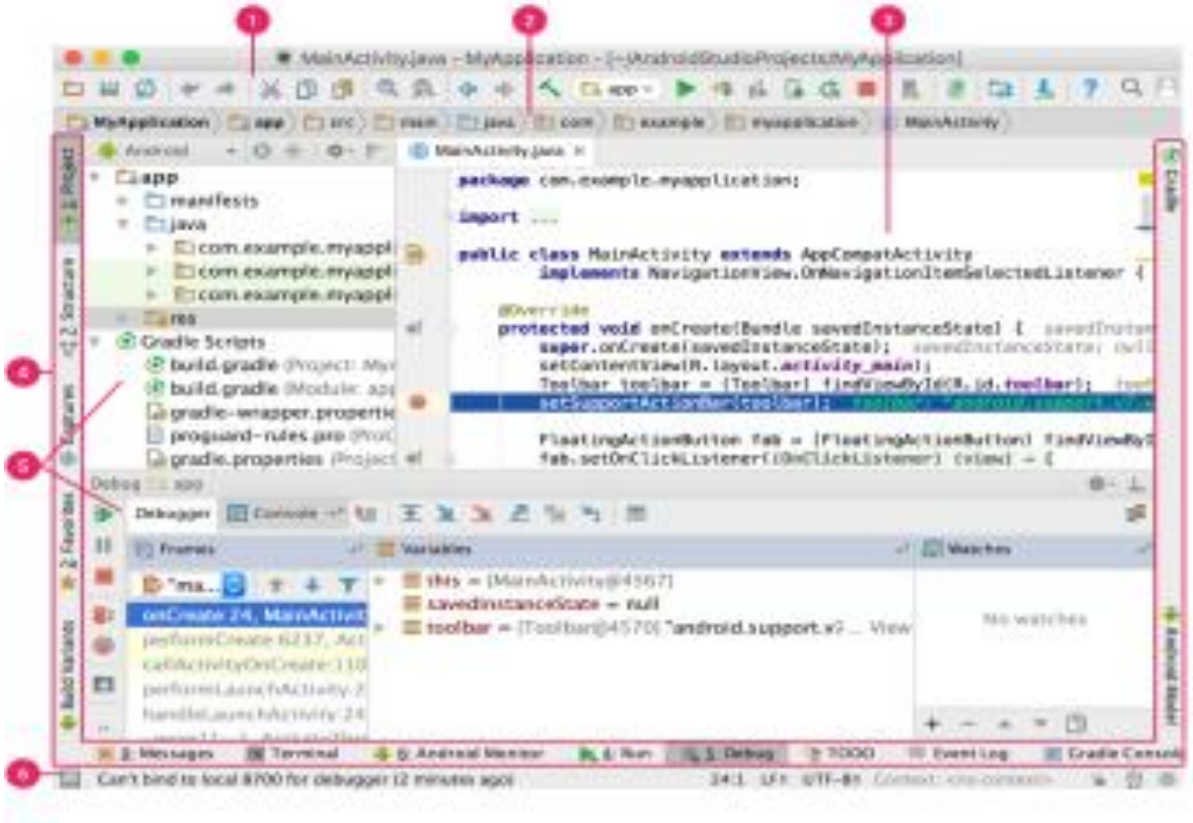


Fig 8.1.3.1: User Interface

You can organize the main window to give yourself more screen space by hiding or moving toolbars and tool windows. You can also use keyboard shortcuts to access most IDE features.

At any time, you can search across your source code, databases, actions, elements of the user interface, and so on, by double-pressing the Shift key, or clicking the magnifying glass in the upper right-hand corner of the Android Studio window. This can be very useful if, for example, you are trying to locate a particular IDE action that you have forgotten how to trigger.

8.1.4 Tool Windows:

Instead of using preset perspectives, Android Studio follows your context and automatically brings up relevant tool windows as you work. By default, the most used tool windows are pinned to the tool window bar at the edges of the application window.

- To expand or collapse a tool window, click the tool's name in the tool window bar. You can also drag, pin, unpin, attach, and detach tool windows.
- To return to the current default tool window layout, click Window > Restore Default Layout or customize your default layout by clicking Window > Store Current Layout as Default.
- To show or hide the entire tool window bar, click the window icon in the bottom left-hand corner of the Android Studio window.
- To locate a specific tool window, hover over the window icon and select the tool window from the menu.

8.1.5 Navigation:

- Here are some tips to help you move around Android Studio. Switch between your recently accessed files using the Recent Files action. Press Control+E (Command+E on a Mac) to bring up the Recent Files action. By default, the last accessed file is selected. You can also access any tool window through the left column in this action.
- View the structure of the current file using the File Structure action. Bring up the File Structure action by pressing Control+F12 (Command+F12 on a Mac). Using this action, you can quickly navigate to any part of your current file.
- Search for and navigate to a specinc class in your project using the Navigate to Class action. Bring up the action by pressing Control+N (Command+O on a Mac). Navigate to Class supports sophisticated expressions, including camel humps, paths, line navigate to, middle name matching, and many more. If you call it twice in a row, it shows you the results out of the project classes.
- Navigate to a file or folder using the Navigate to File action. Bring up the Navigate to File action by pressing Control+shift+N (Command+Shift+0 on a Mac). To search for folders rather than files, add a / at the end of your expression.

- Navigate to a method or field by name using the Navigate to Symbol action. Bring up the Navigate to Symbol action by pressing Control+Shift+Alt+N(Command+Shift+Alt+0 on a Mac).
- Find all the pieces of code referencing the class, method, field, parameter, or statement at the current cursor position by pressing Alt+F7

8.1.6 Gradle Build System:

Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by the Android plugin for Gradle.
This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. You can use the features of the build system to do the following:

- Customize, configure, and extend the build process.
- Create multiple APKs for your app, with different features using the same project and modules.
- Reuse code and resources across sourcesets.

By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are namedbuild gradle. They are plain text files that use Groovy syntax to configure the build with elements provided by the Android plugin for Gradle. Each project has one top-level build file for the entire project and separate module- level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

8.1.7 Multiple APK Support:

Multiple APK support allows you to efficiently create multiple APKs based on screen density or ABl. For example, you can create separate APKs of an app for the hdpi and mdpi

screen densities, while still considering them a single variant and allowing them to share test APK, javac, dx, and ProGuard settings.

8.1.8 Debug and Profile Tools:

Android Studio assists you in debugging and improving the performance of your code, including inline debugging and performance analysis tools.

8.1.9 Inline Debugging:

Use inline debugging to enhance your code walk-throughs in the debugger view with inline verification of references, expressions, and variable values. Inline debug information includes:

- Inline variable values
- Referring objects that reference a selected object
- Method return values
- Lamba and operator expressions
- Tooltip values

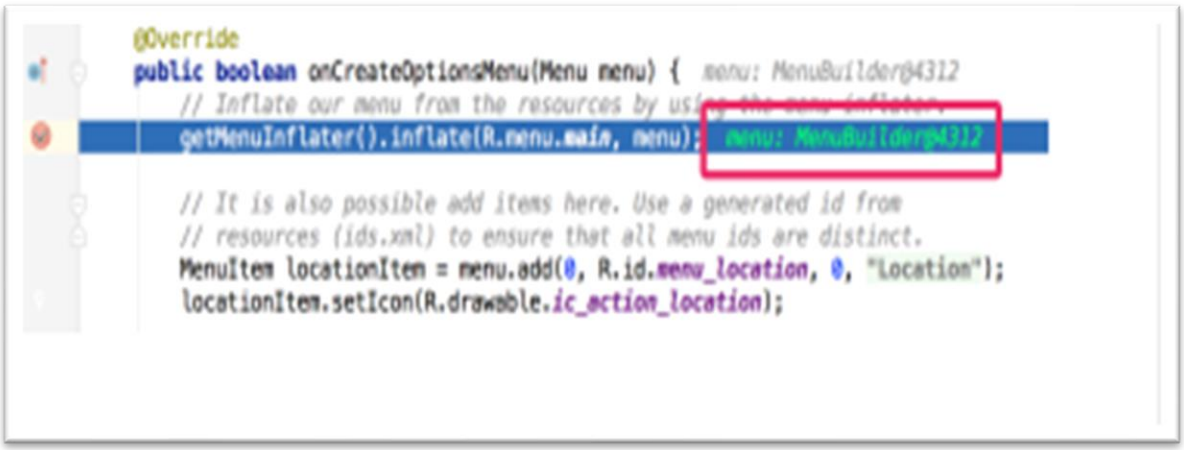


Fig 8.1.9.1: Inline Debugging

8.1.10 Performance Monitors:

Android Studio provides performance monitors so you can more easily track your app's memory and CPU usage, find deallocated objects, locate memory leaks, optimize graphics performance, and analyze network requests. With your app running on a device or emulator, open the Android Monitortool window, and then click the Monitors tab.

8.1.11 Allocation Tracker:

Android Studio allows you to track memory allocation as it monitors memory use. Tracking memory allocation allows you to monitor where objects are being allocated when you perform certain actions. Knowing these allocations enables you to optimize your app's performance and memory use by adjusting the method calls related to those actions.

8.1.12 Code Inspections:

Whenever you compile your program, Android Studio automatically runs configured Lint and other IDE inspections to help you easily identify and correct problems with the structural quality of your code.

The Lint tool checks your Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization.

Fig 8.1.12.1: Code Inspection

8.2 OBESITY DETECTION:

8.2.1 USER END:

8.2.1.1 Home Activity for User End:

Home activity has initiated the different types of plans based on weight, based on skin type and it even include the motivational vedios. Various options to check the list of Existing plans and Remainders to stay on time.

```
private void initUI(){
    getSupportActionBar().hide();

    sharedpreferences = getSharedPreferences( name: "OBESITY_DET

    mContext = this;
    homeLay = findViewById(R.id.homeLay);
    spSelection = findViewById(R.id.spSelection);
    dietRV = findViewById(R.id.dietRV);
    txtNoData = findViewById(R.id.txtNoData);
    iv_logout = findViewById(R.id.iv_logout);
    ArrayAdapter adapter = new ArrayAdapter( context: this,androi
    adapter.setDropDownViewResource(android.R.layout.simple_spi
    spSelection.setAdapter(adapter);

    spSelection.setOnItemSelectedListener(new AdapterView.OnIte
        @Override
        public void onItemSelected(AdapterView<?> parent, View
            if(spSelection.getSelectedItem().toString().equalsI
                new getUserDietPlan().execute("REQUESTED");
            }else{
                new getUserDietPlan().execute("CHANGED");
            }
        }

        @Override
```

Fig 8.2.1.1: Home Activity

8.2.1.2 Login Activity and Registration Activity for User End:

In Login Activity we check for existing user by accessing the database in background and if we found login was successful if not update to end user by worng password or ask him to register if user was new.

```
    boolean isLoggedIn = sharedpreferences.getBoolean( s: "isLoggedIn",  b: false);
    if (isLoggedIn) {
        Intent mainIntent = new Intent( packageContext: LoginActivity.this, HomeActivity.c
        LoginActivity.this.startActivity(mainIntent);
        LoginActivity.this.finish();
    } else {
        loginLay = findViewById(R.id.loginLay);
        etEmailId = findViewById(R.id.etEmailId);
        etPassword = findViewById(R.id.etPassword);
        btnLogin = findViewById(R.id.btnLogin);
        txtRegister = findViewById(R.id.txtRegister);
        txtRegister.setOnClickListener(v -> Helper.startActivity( context: this, Registra



        btnLogin.setOnClickListener(v -> {
            if (isValidate()) {
                new Login().execute(etEmailId.getText().toString(), etPassword.getText
            }
        });
    }
}
```

Fig 8.2.1.2: Login Activity and Registration Activity

8.2.1.3 Diet Plan Activity for User End:

This Activity contains the following Diet the Doctor given to the user.

```
public class DietPlanAdaptor  extends RecyclerView.Adapter<DietPlanAdaptor.ViewHolder>{
    private final Context mContext;
    private ArrayList<DietPlanDetail> dietPlanDetailArrayList = null;

    public DietPlanAdaptor(ArrayList<DietPlanDetail> dietPlanDetailArrayList, Context mContext) {
        this.dietPlanDetailArrayList = dietPlanDetailArrayList;
        this.mContext = mContext;
    }

    @NonNull
    @Override
    public DietPlanAdaptor.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        LayoutInflater layoutInflater = LayoutInflater.from(parent.getContext());
        View item = layoutInflater.inflate(R.layout.diet_plan_detail_content, parent,  attachToRoot: false);
        ViewHolder viewHolder = new ViewHolder(item);
        return viewHolder;
    }

    @Override
    public void onBindViewHolder(@NonNull DietPlanAdaptor.ViewHolder holder, int position) {
```

Fig 8.2.1.3: Diet Plan Activity

8.2.1.4 Remainder Activity and Motivational Videos Activity for User End:

As name says it was a remainder help to keep track of activities and it even has both spiritual and physical motivational videos to push forward the user.

```
import com.google.android.material.floatingactionbutton.FloatingAc
import com.google.android.material.snackbar.Snackbar;

import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;

public class ReminderActivity extends AppCompatActivity {

    RelativeLayout reminderLay;
    RecyclerView reminderRc;
    FloatingActionButton add_reminder_fab;
    Context mContext;
    SharedPreferences sharedPreferences;
    String userID;
    ArrayList<Reminder> reminderArrayList;
    TextView txtNoData;
    ReminderAdaptor.reloadUI reloadUI = id -> new deleteReminder()

    @Override
```

Fig 8.2.1.4: Remainder Activity and Motivational Videos Activity

8.2.2 DOCTOR END:

This application is for doctors to select the deit plan for user and Keep track of user activity.

8.2.2.1 Home Activity for Doctor End:

```
private void initUI(){
    getSupportActionBar().hide();

    sharedpreferences = getSharedPreferences( name: "OBESITY_DETECTION", MODE_PRIVATE);

    mContext = this;
    homeLay = findViewById(R.id.homeLay);
    spSelection = findViewById(R.id.spSelection);
    dietRV = findViewById(R.id.dietRV);
    txtNoData = findViewById(R.id.txtNoData);
    iv_logout = findViewById(R.id.iv_logout);
    ArrayAdapter adapter = new ArrayAdapter( context: this,android.R.layout.simple_spinner_item,selecti
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spSelection.setAdapter(adapter);

    spSelection.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
            if(spSelection.getSelectedItem().toString().equalsIgnoreCase( anotherString: "PENDING")){
                new getUserDietPlan().execute("REQUESTED");
            }else{
```

Fig 8.2.2.1: Home Activity

8.2.2.2 Login Activity for Doctor End:

```
private class Login extends AsyncTask<String, JSONObject, String> {
    @Override
    protected String doInBackground(String... strings) {
        String data = null;
        RestAPI restAPI = new RestAPI();
        try {
            JSONParse jp = new JSONParse();
            JSONObject json = restAPI.DLogin(strings[0], strings[1]
            data = jp.Parse(json);
        } catch (Exception e) {
            data = e.getMessage();
        }
        return data;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        Helper.showProgressDialog( context: LoginActivity.this);
    }
```

Fig 8.2.2.2: Login Activity

8.2.2.3 Deit Plan and Food Name Content Activities for Doctor End:

It has deit plans and even has food dailog box which keep tracks of fat and colostrols of food intake.

```
public void onBindViewHolder(@NonNull DietPlanAdaptor.ViewHolder holder, int position) {
    DietPlan dietPlan = dietPlanArrayList.get(position);
    holder.txtUserName.setText(dietPlan.getUserName());
    holder.txtBmi.setText("BMI Value: " + dietPlan.getBMIValue());
    holder.txtBmr.setText("BMR Value: "+dietPlan.getBMRValue());

    holder.dietPlanCard.setOnClickListener(v -> {
        Intent intent = new Intent(mContext, ChangeDietPlanActivity.class);
        intent.putExtra( name: "DIET",dietPlan);
        mContext.startActivity(intent);
    });
}

@Override
```

Fig 8.2.2.3: Deit Plan and Food Name Content Activities

8.2.2.4 RESTAPI:

```
public class RestAPI {
    private final String urlString = "http://aobesity.hostoise.com/handler1.ashx";

    private static String convertStreamToUTF8String(InputStream stream) throws IOException {
        String result = "";
        StringBuilder sb = new StringBuilder();
        try {
            InputStreamReader reader = new InputStreamReader(stream,  charsetName: "UTF-8");
            char[] buffer = new char[4096];
            int readedChars = 0;
            while (readedChars != -1) {
                readedChars = reader.read(buffer);
                if (readedChars > 0)
                    sb.append(buffer,  offset: 0, readedChars);
            }
            result = sb.toString();
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        return result;
    }
```

Fig 8.2.2.4: RestAPI

The following methods are implemented to intreact with database and other API's

- Load
- mapObject
- Ulogin
- URegister
- UgetProfile
- UpdateProfileUser
- UchangedPassword
- AddRemainder
- GetAllRemainder
- UpdateRemainder
- DeleteRemainder
- OnOffRemainder
- Add/Update/DeleteMotivationalVedio
- AddUserDiseasesAndAllergies
- getAllActivities
- getAllDieseasesOccurences

- getAllGeneticDiseases
- getAlluserGeneratedDiseases
- GetAllAlegries
- MakeDietPlan
- getDietPlan
- CheckIfUserHasDietPlan
- GetExtraIngredient
- getActivities
- getDiseasesOccurence
- RequestCustomization
- Dlogin
- GetAllFood
- GetDietPlanByStatus
- ChangeDietPlan
- GetAllType
- GetAllSkinFoodByType
- GetAllSBActivities

CHAPTER 9

TESTING

9.1 TESTING PLAN

Testing process starts with a test plan. This plan describes all of the testing related activities that must be completed, as well as the timetables, resource allocations, and testing criteria. The stated test cases are executed during unit testing, and the actual result is compared to the expected output. The test report and the error report are the final outputs of the testing phase.

The testing done here was System Testing checking whether the user requirements were satisfied. The code for the new system has been written completely using JAVA as the coding language and

Fig 9.1: Testing

Android Studio as the interface for front-end designing. using Python as the coding language and Django Framework as the interface for front-end designing. The new system has been tested well with the help of the users and all the applications have been verified from every nook and corner of the user.

Although some applications were found to be erroneous these applications have been corrected before being implemented. The flow of the forms has been found to be very much in accordance with the actual flow of data.

9.2 Levels of Testing:

To uncover the errors, present in different phases we have the concept of levels of testing. The basic levels of testing are:

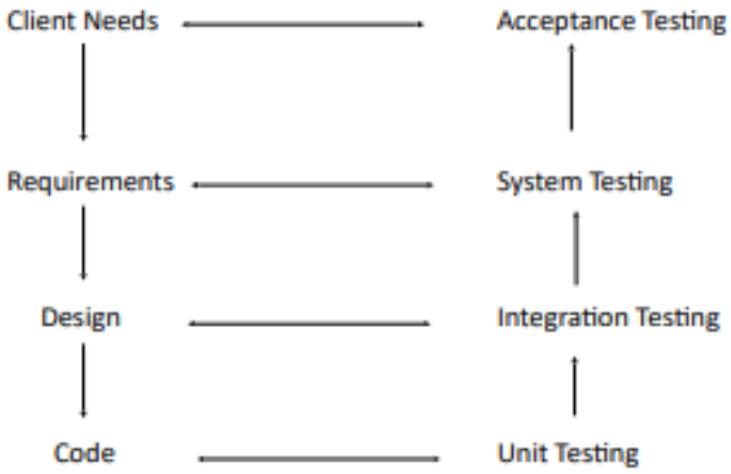


Fig 9.2: Levels of Testing

A series of testing is done for the proposed system before the system is ready for the user acceptance testing.

The steps involved in Testing are:

9.2.1 Unit Testing:

Unit testing focuses verification efforts on the smallest unit of the software design, the module. This is also known as "Module Testing". The modules are tested separately. This testing carried out during programming stage itself. In this testing each module is found to be working satisfactorily as regards to the expected output from the module.

9.2.2 Integration Testing:

Data can be grossed across an interface; one module can have adverse efforts on another. Integration testing is systematic testing for construction the program structure while at the same time conducting tests to uncover errors associated with in the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is

difficult because the isolation of cause is complicate by the vast expense of the entire program. Thus, in the integration testing stop, all the errors uncovered are corrected for the text testing steps.

9.2.3 System Testing:

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently for live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, then goal will be successfully achieved.

9.2.4 Validation Testing:

At the conclusion of integration testing software is completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software tests begins, validation test begins. Validation test can be defined in many ways. But the simple definition is that validation succeeds when the software function in a manner that can reasonably expected by the customer. After validation test has been conducted one of two possible conditions exists.

One is the function or performance characteristics confirm to specifications and are accepted and the other is deviation from specification is uncovered and a deficiency list is created. Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

9.2.5 Output Testing:

After performing validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated by the system under consideration. Here the output format is considered in two ways, one is on the screen and other is the printed format. The output format on the screen is found to be correct as the format was designed in the system designed phase according to the user needs.

For the hard copy also, the output comes as the specified requirements by the users. Hence output testing does not result any corrections in the system.

9.2.6 User Acceptance Testing:

User acceptance of a system is the key factor of the success of any system. The system under study is

tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

9.3 Test Cases:

Registration: To begin with login, user need to register by filling up basic registration details. There are multiple fields in registration page and every field has to fill by user. User cannot use character in the login id field.

Login: Login id and password are kept compulsory fields, and if the id or password doesn't match then it will show an error message.

CHAPTER 10

RESULT AND DISCUSSION

10.1 Register Page:

Here, in the register page we can create an account in the application by adding the details to proceed further.

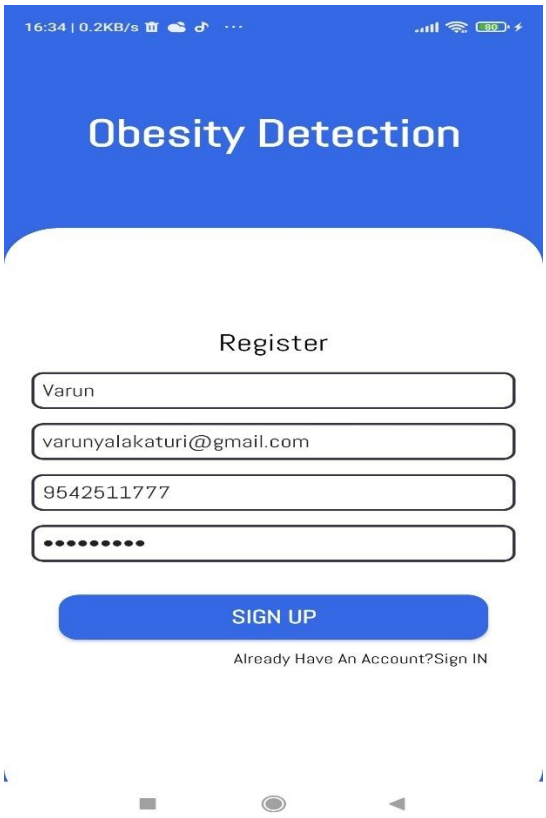


Fig 10.1: Register Page

10.2 Login Page:

Fig 10.2.1: Login Page

10.3 Add Details Page:

Fig 10.3.1: Add Details Page

10.4 Diet Plan Page:

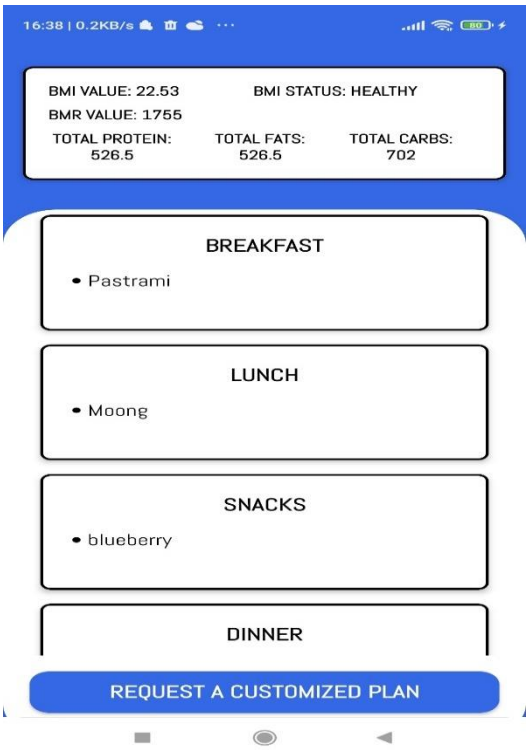


Fig 10.4.1: Diet Plan Page

10.5 Remainder Page:

Fig 10.5.1: Remainder Page

10.6 Motivational Video Page:

Fig 10.6.1: Motivational Video Page

10.7 Type of Diet Page:

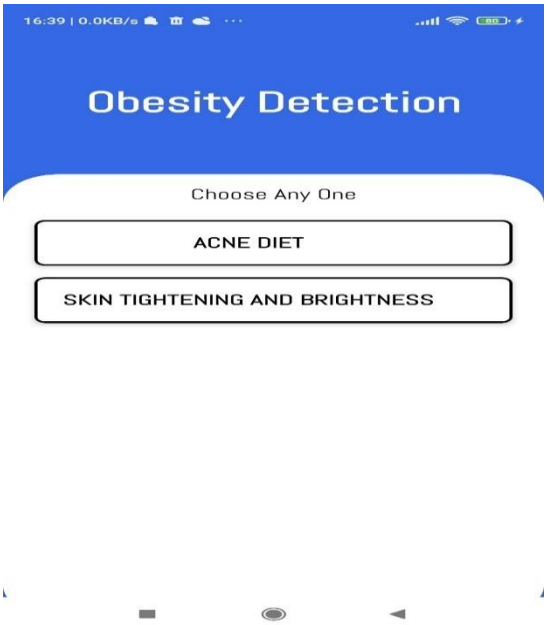


Fig 10.7.1: Type of Diet Page

10.8 Activity Page:

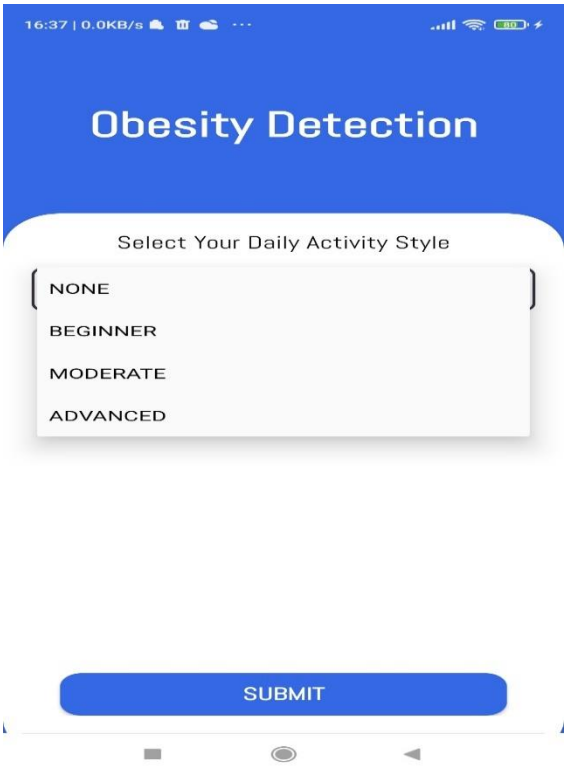


Fig 10.8.1: Activity Page

10.9 Type of Allergies Page:

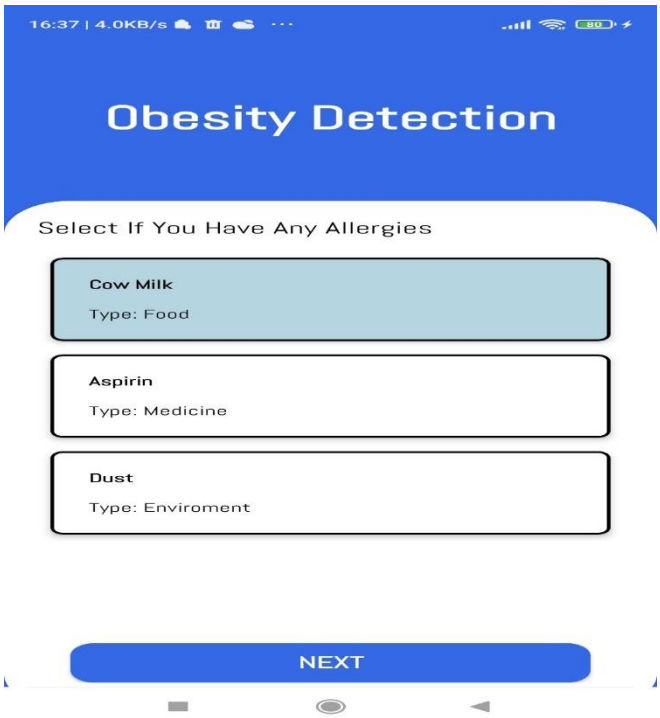


Fig 10.9.1: Type of Allergies Page

10.10 Genetic Disease Page:

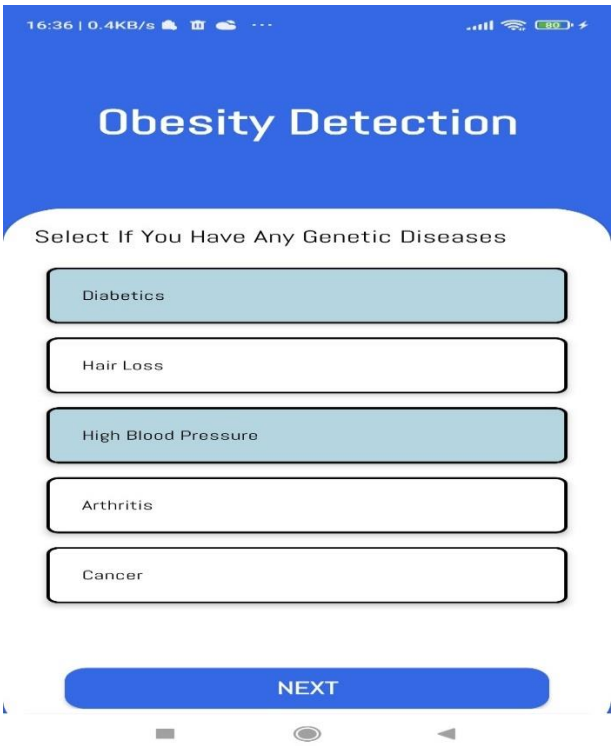


Fig 10.10.1: Genetic Disease Page

10.11 Logout Page:

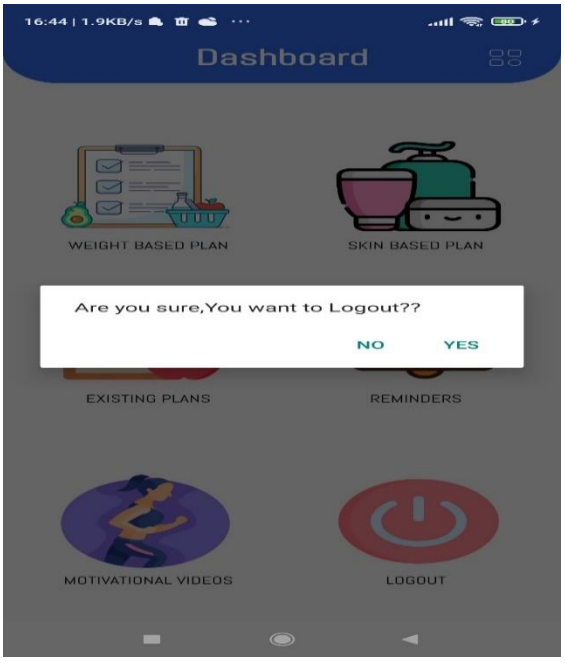


Fig 10.11.1: Logout Page

10.12 Dashboard Page:

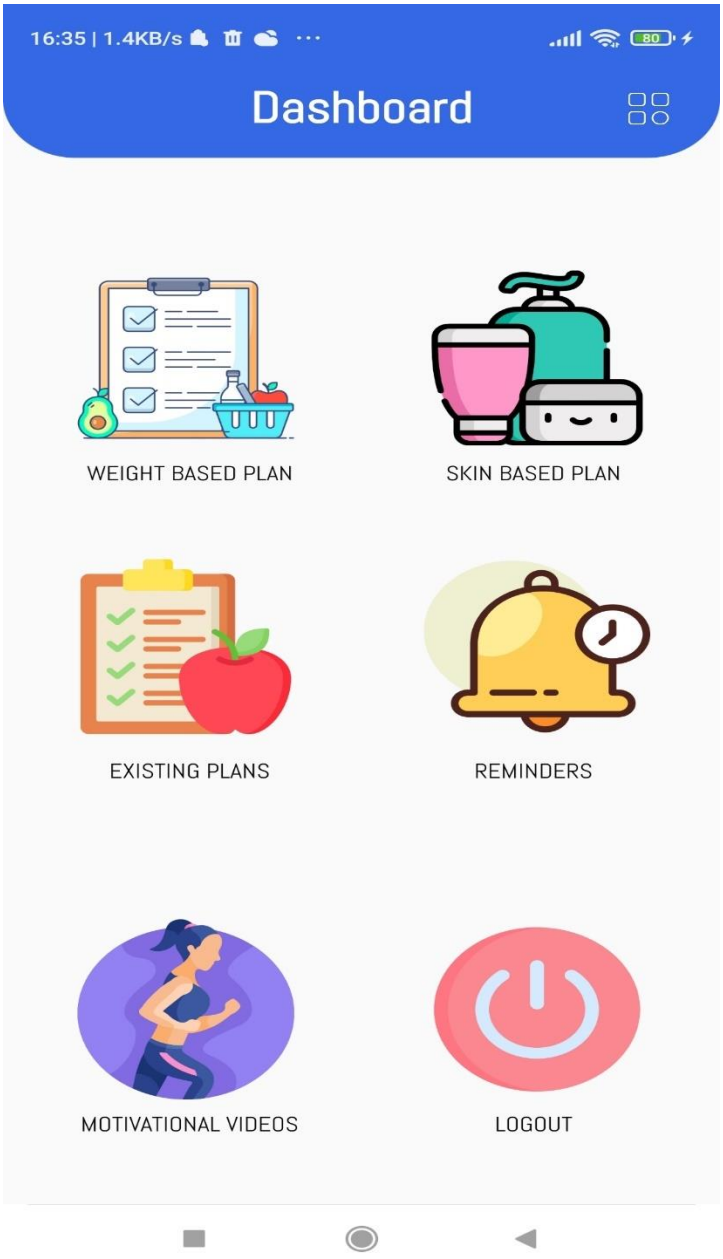


Fig 10.11.1: Dashboard Page

CHAPTER 11

CONCLUSION AND FUTURE SCOPE

11.1 CONCLUSION:

This was our project of System Design about Obesity Detection & Diet Control developed in Java programming language. The Development of this system takes a lot of efforts from us. We think this system gave a lot of satisfaction to all of us. Though every task is never said to be perfect in this development field even more improvement may be possible in this application. We learned so many things and gained a lot of knowledge about development field. We hope this will prove fruitful to us.

11.2 FUTURE SCOPE:

Some of the Advanced features are calories counter, cholesterols level checker, nutrition calculator based on machine learning algorithms. This feature helps to stabilize daily meal which show drastic results. We can also experiment some of the features like sick level test based on facial scan which suggest some health fruits or meals to get active. When a user points his smartphone camera to a plate containing his/her meal, the app rapidly recognizes its different ingredients. A calorie counter is a way to count your daily caloric intake using our easy-to-use caloric counter. Calorie counting is an easy way for you to manage your weight. If you have a daily caloric requirement that you want to meet, or you need to monitor your caloric intake, our calorie counting technique is for you.

CHAPTER 12

BIBILOGRAPHY

[1] Fule Wang, Yu Pan, Yuan Yaun, Bin Hu, “Study on the Principles of the Intelligent Diet Arrangement System Based on Multi-Agent” 2018

[2] Jen-Hao Hsiao, Henry Chang, “SmartDiet: A Personal Diet Consultant for Healthy Meal Planning” 2020

[3] Faria Ferdowsy, Kazi Samsul Alam Rahi, Md. Ismail Jabiullah, Md. Tarek Habib, “A machine learning approach for obesity risk prediction” 2021

[4] Raciel Yera Toledo, Ahmad A. Alzahrani, And Luis Martnez, “A Food Recommender System Considering Nutritional Information and User Preferences” 2019

[5] Drishti P. Ghelani, Lisa J. Moran, Cameran Johnson, Aya Mousa, Mobile Apps for Weight Management: A Review of the Latest Evidence to Inform Practice 2020.

[6] Jong Hunkin, Jung Hyun Lee, Design of Diet Recommendation System for Healthcare Service Based on User Information 2019.

[7] Priyakorn Pusawiro, Tossawat Mokdara, Personalized Food Recommendation Using Deep Neural Network 2019.

[8] Romeshwar Sookrah, Jaysree Devee, DASH Diet Recommendation system for hypertensive patients 2020.

[9] G. Agapito, M. Simeoni, B. Calabrese, I. Caré, T. Lamprinou, P. H. Guzzi, A. Pujia, G. Fuiano, and M. Cannataro, ‘‘DIETOS: A dietary recommender system for chronic diseases monitoring and management,’’ Comput. Methods Programs Biomed., vol. 153, pp. 93–104, Jan. 2018

[10] F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh, 2015, “Recommendation Systems: Principles, Methods and Evaluation”, Egyptian Informatics Journal, Vol. 16, Issue 3, November, pp. 261–273.

[11]J. Freyne and S. Berkovsky, 2010, “Intelligent Food Planning: Personalized Recipe Recommendation”, The 15th Annual Meeting of the Intelligent User Interfaces Community, 7-10 February 2010, Hong Kong, pp. 321-324.

[12] Chun-Yuen Teng, Yu-Ru Lin, Lada A. Adamic, 2012, “Recipe recommendation using ingredient networks”, 4th Annual ACM Web Science Conference, 22-24 June 2012, Illinois, pp. 298-307.

[13] M. A. El-Dosuky, M. Z. Rashad, T. T. Hamza, and A. H. EL-Bassiouny, "Food Recommendation Using Ontology and Heuristics," Advanced Machine Learning Technologies and Applications, Communications in Computer and Information Science, Vol.322, pp.423-429, 2012.

[14] C.-Y. Teng, Y.-R. Lin, and L. A. Adamic, "Recipe recommendation using ingredient networks," Proceedings of the 4th Annual ACM Web Science Conference (WebSci '12), pp.298-307, 2012

[15] D. Bianchini, V. De Antonellis, N. De Franceschi, and M. Melchiori, ‘‘PREFer: A prescription-based food recommender system.’’ Comput. Standards Inter., vol. 54, pp. 64–75, Nov. 2017.

[16] Y. van Pinxteren, G. Geleijnse, and P. Kamsteeg, "Deriving a recipe similarity measure for recommending healthful meals," Proceedings of the 16th international conference on Intelligent user interfaces (IUI '11), pp.105-114, 2011.

[17] Fang-Fei Kuo, Cheng-Te Li, Man-Kwan Shan, and Suh-Yin Lee, "Intelligent menu planning: recommending set of recipes by ingredients," Proceedings of the ACM multimedia 2012 workshop on Multimedia for cooking and eating (CEA '12), pp.1-6, 2012.