

PROJECT REPORT
ON
PRATIBHOJANA: AN ALARM BASED
PERSONALIZED NUTRITION RECOMMENDATION
SYSTEM

SUBMITTED TO THE
UNIVERSITY OF MUMBAI

FOR THE DEGREE OF
BACHELOR OF ENGINEERING
IN
COMPUTER ENGINEERING 2015-2016

Submitted by

A.J.Sudhakaran (Roll No.01)
Stacy Bardeskar (Roll No.04)
Mohit Deorukhkar (Roll No.15)

Supervisor (s):

Ms. Ditty Varghese
Mr. Amiya Kumar Tripathy

DEPARTMENT OF COMPUTER ENGINEERING

DON BOSCO INSTITUTE OF TECHNOLOGY
(ENGINEERING COLLEGE)
PREMIER AUTOMOBILES ROAD, KURLA (W) – 400070

PRATIBHOJANA: AN ALARM BASED PERSONALIZED NUTRITION RECOMMENDATION SYSTEM

Submitted in partial fulfilment of the requirement for the Degree of
**BACHELOR OF ENGINEERING IN COMPUTER
ENGINEERING**

by

A.J.Sudhakaran Roll No.01

Stacy Bardeskar Roll No.04

Mohit Deorukhkar Roll No.15

Supervisor (s):

Ms. Ditty Varghese

Mr. Amiya Kumar Tripathy



DEPARTMENT OF COMPUTER ENGINEERING

**DON BOSCO INSTITUTE OF TECHNOLOGY
(ENGINEERING COLLEGE) PREMIER AUTOMOBILES ROAD,
KURLA (W) – 400070**

2015-2016

CERTIFICATE

This is to certify that the project report entitled “**Pratibhojana: An alarm based personalized nutrition recommendation system**” is a bonafide work
of

A.J.Sudhakaran Roll No.01

Stacy Bardeskar Roll No.04

Mohit Deorukhkar Roll No.15

submitted to the University of Mumbai in partial fulfillment of the requirement
for the award of the Degree of Bachelor of Engineering in the Computer
Engineering

Mr. Amiya Kumar Tripathy
Supervisor

Ms. Ditty Varghese
Co-Supervisor

Ms. Nilakshi Joshi
Head of Department

Dr. P. Nambiar
Principal

Project Report Approval for B. E.

This project report entitled **Pratibhojana: An alarm based personalized nutrition recommendation system** by **A.J.Sudhakaran, Stacy Bardeskar and Mohit Deorukhkar** is approved for the degree of Computer Engineering course of Bachelor of Engineering.

Examiners

- 1.-----
- 2.-----

Date:

Place:

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1. A.J.Sudhakaran, 01

(Name of student, Roll no.)

(Signature)

2. Stacy Bardeskar, 04

(Name of student, Roll no.)

(Signature)

3. Mohit Deorukhkar, 15

(Name of student, Roll no.)

(Signature)

Date:

ABSTRACT

Good health is paramount especially when we have women suffering from diseases such as anemia, osteoporosis and heart disease in India. There is lack of awareness amongst older women about importance of good nutrition which is based on the appropriate, balanced intake and utilization of nutrients. Deficiencies in nutrition lead to long-term damage to both individuals and society. The proposed system aims to design a personalized diet recommendation system supported with an alarm reminder for women in the age group of 40-50. With the help of this system, women can gain access to diet recipes based on some standard inputs. The system thereby relieves women from the burden of having to identify the right diet for their condition and will thereby help improve the nutritional status of households. Investing in nutrition in women is vital because it reduces healthcare costs, improves productivity and economic growth, and promotes education, intellectual capacity, and social development for present and future generations.

CONTENTS

• ABSTRACT	i
• CONTENTS	ii
• LIST OF FIGURES	iv
• LIST OF TABLES	v
• ABBREVIATIONS	vi
1. Introduction	
1.1. Introduction	1
1.2. Aim and Objectives	2
1.3. Organization of report	3
2. Literature Survey	4
3. Problem Statement	
3.1. Problem Statement	6
3.2. Scope	7
4. System Analysis	
4.1. Existing System	9
4.2. Proposed System	11
4.3. Analysis	11
4.4. Hardware and Software details	12
4.5. Methodology	15

4.6. Design details	16
4.7. Formulae and algorithm	17
5. Implementation	
5.1. Module I	19
5.2. Module II	20
6. Results and Discussion	23
7. Project timeline and task distribution	33
8. Conclusion	35
9. References	36
Appendix	38
Acknowledgement	92

LIST OF FIGURES

Section	Name	Pg. No.
4.5	System Flowchart	15
4.6.1	System Architecture	16
5.1	System GUI Design	19
5.2.1	Backend Module	20
5.2.2	Database Design	21
5.2.3	Ingredient Table	22
6.1	Loading page	23
6.2	Home page	24
6.3	Fragment Slider	24
6.4	User Information	25
6.5	Selection of Diseases	26
6.6	Selection of Allergies	26
6.7	Recall to know diet history	27
6.8	Type of Ingredients in recall	27
6.9	Ingredients in recall	28
6.10	Ingredients quantity in recall	29
6.11	User profile page (Personal)	29
6.12	User profile page (Diet)	30
6.13	User profile page (Nutritional)	30
6.14	Recommendation page	31
6.15	Recommendation page (Setting Alarm)	31

6.16	Recommendation page (Alarm clock pop-up)	32
6.17	Recommendation page (with Alarm set)	32

LIST OF TABLES

Table No.	Table Title	Page No.
1	Project Timeline	33
2	Task Distribution	34

ABBREVIATION

Sr. No	Abbreviation	Full form
1	BMI	Body Mass Index
2	BMR	Basic Metabolic Rate

1. INTRODUCTION

1.1 Introduction

Nutrition focuses on how to protect the body from disease with a healthy diet. Human body receives nutrition from the food consumed. The major classes of nutrients are: carbohydrates, fats, minerals, protein, vitamins, fiber and water. All these nutrients are necessary for human body but in varying quantities. Poor health or nutritional deficiency occurs due to inadequate amount of a particular nutrient or group of nutrients.

For a woman, working at home, taking care of her children and the elderly as well as working outside the home can greatly affect her physical and mental well being. Nowadays, working women tend to neglect their health. Fast life and unhealthy eating habits are leading to various health problems. Due to these changing trends, illnesses that used to usually occur during later stages of life are now arriving at an early age. And as a result women are facing a constant pressure of being health conscious. Due to this they feel the urge to obtain a quick access to information with respect to their health and are therefore resorting to the internet for the same. Also there are many women who lack appropriate knowledge of the right nutrition needed for the body.

Anemia and osteoporosis are the health issues commonly prevalent amongst women. Cultural factors in India deprive the female child nutritious food and that is the reason why Indian women are mostly anemic and prone to osteoporosis in old age. It is a major concern of health problems related to ageing amongst women. Particularly at the onset of ageing which is probably at 40, women are easily prone to numerous ailments such as heart diseases, menopause and others. Indian middle class women often tend to ignore their health and visit a doctor only when certain disease symptoms occur, which later might turn out to be incurable. Doctors and nutritionists have thus mentioned lack of awareness amongst women as the major cause which leads them to greater disease states. Prevention is better than cure. Therefore, it is best if women adopt a healthy lifestyle which could help them keep such health issues at bay.

To promote a healthy lifestyle by correcting diet now-a-days we have a lot of information available on the internet. But this can also be confusing at times. Also mobile being a handy source of information the availability of nutrition apps these days has greatly increased.

But despite all this the information available is not crisp in various ways as they are not age specific, they do not cater to the Indian population, they do not take the diet history of a person into account and also the diet recommended does not consider the availability of ingredients.

Therefore a system was proposed that would enable women to gain access to diet recipes based on some standard inputs. The proposed system is designed to take in general inputs such as the height, age, weight, type of lifestyle and all the allergies and disease inputs specific to a person. Based on these inputs, the system recommends diet recipes for each meal to be eaten at the appropriate time selected. Amongst all the categories of individuals, the ones that require utmost attention are women particularly in the age group of 40 to 50 years as they are most likely prone to diseases. The proposed system aims at providing good nourishment to these individuals. An additional advantage of the proposed system is the alarm reminder for food. With the help of this system, women will be able to obtain proper diet recommendations at the right time, thus helping them eat proper food as per their body condition and nutritional needs.

1.2 Aim and Objectives

The proposed system aims to develop a portable personalized nutrition recommendation with alarm reminder by taking the basic inputs from the user and recommending the desired dishes according to the input, for Indian women in the age group of 40-50 years.

Women tend to neglect their health in both quality and quantity. A healthy woman keeps the family healthy and as a result the aim is to provide diet recipes that will keep all the general problems away and provide fitness and good health.

Main objective is to provide women with healthy nutrition based on their level of illness and allergies supported by an alarm reminder for their medicines and food.

1.3 Organization of Report

First chapter of the report contains Introduction to this project which explain why this project is undertaken and the concepts associated with it. Second chapter of the report

consists of Literature survey where key for improvements in existing technologies is determined. Chapter 3 explains problem statement, Chapter 4 deals with analysis and designing of the system. Chapter 5 deals with the implementation. Chapter 6 discusses results in details. Rest of the report deals with conclusion and future scope.

2. LITERATURE SURVEY

India has been growing steadily richer in recent years, but it still has more malnourished people, especially children, than any other country [7] A big, nationwide study from 2005 and 2006, the National Family Health Survey (NFHS), found that 42.5% of children under five years old were underweight. The region with the next highest proportion of underweight children is Africa, with an average of 21%. Another measure of malnutrition is stunting, when children are unusually short for their age.

The UN estimates that 2.1 million Indian children die before reaching the age of 5 every year— four every minute - mostly from preventable illnesses such as diarrhea, typhoid, malaria, measles and pneumonia[6]Every day, 1,000 Indian children die because of diarrhea alone. Overall, girls and boys are about equally likely to be undernourished. Under-nutrition is higher in rural areas and is strongly correlated with the level of maternal education showing a two-fold difference between non-educated mothers and 10-year and above educated mothers. This may be linked to a stark difference in access to a nutritious diet and complementary feeding at 6-9 months. Most children under age three are anemic (79.2%). The prevalence is slightly higher in rural areas and among non-educated mothers. High prevalence of anemia may be linked to poor variety of diet, poor hygienic conditions and limited access to iron supplementation. Only 55.8% of children aged 6-9 months receive solid or semisolid food and breast milk. Although the percentage is significantly lower among non-educated mothers and in rural areas, the prevalence in urban areas and among well-educated mothers is still less than 70% making complementary feeding a high-priority to be addressed. 33% of married women and 28% of men are too thin, according to the body mass index (BMI), an indicator derived from height and weight measurements. Underweight is most common among the poor, the rural population, adults who have no education and scheduled castes and scheduled tribes. Health problems related to diet, including obesity and cancer, are important concerns in the current society [3] Diet programs keep and cause weight loss over short, medium, or long term. However, to maintain balanced body energy, a frequent physical exercise is required [3] To achieve diet management and healthy diet, many non-commercial and commercial solutions have been proposed in the format of PC-based and smartphone based apps to help users [2] There are

many available proposed systems which are successfully working in market currently which helped our system form its core features.

3. PROBLEM STATEMENT

3.1. Problem Statement

In India, there is a lack of appropriate knowledge of the right nutrition amongst older women. Women in India face a multitude of health problems, which ultimately affect the aggregate economy's output.

Malnutrition, defined as ill health caused by deficiencies of calories, protein, vitamins, and minerals interacting with infections and other poor health and social conditions, saps the strength and well-being of millions of women and adolescent girls around the world. Due to the low social status of Indian women, their diet often lacks in both quality and quantity. Women who suffer malnutrition are less likely to have healthy babies. Malnutrition and anemia are common among Indian adults. Both malnutrition and anemia have increased among women since 1998-99.

- 33% of married women are too thin, according to the body mass index (BMI).
- Overweight and obesity, the other side of malnutrition, is a growing problem in India, affecting almost 15% of women. Overweight and obesity are most common in urban areas, in wealthier households, and among older adults.
- 56.2% of women suffer from anemia, and have lower than normal levels of blood hemoglobin.
- Low intakes of dietary iron and calcium are common in women.

Adequate nutrition, a fundamental cornerstone of any individual's health, is especially critical for women because inadequate nutrition wreaks havoc not only on women's own health but also on the health of their children. Children of malnourished women are more likely to face cognitive impairments, short stature, lower resistance to infections, and a higher risk of disease and death throughout their lives.

Women, throughout most of the world, have the major responsibility for their families' nutrition. Their own nutrition is often impaired, under the social and biological stresses they face. Developments that improve women's position in society are likely to improve nutrition overall, and are essential for this. Equally, any activities aimed at preventing malnutrition depend substantially on women's activities, indeed on their empowerment. At the same

time, more attention must be paid to improving women's own nutritional status – perhaps an under-recognized problem – and many of the necessary technologies are now well known. Finally, it is becoming increasingly clear that protecting women's nutrition, notably during pregnancy and lactation but in fact throughout the life-cycle, is necessary to safeguard the nutrition of infants, children and indeed future generations.

Based on these statistics it can be concluded that Indian Women are the ones that require utmost care.

Our food choices each day affect our health — how we feel today, tomorrow, and in the future. Prevention is better than cure. The earlier a person starts to eat a healthy and balanced diet, the more he or she will stay healthy. In order for our bodies to function properly and stay healthy, it is important that we follow a good nutritious diet which will not only prevent the risk of chronic diseases but also promote overall health.

In order to combat this we proposed an alarm based personalized nutrition recommendation system which will recommend diet recipes for Indian Women between 40-50 years based on some standard inputs.

3.2. Scope

This project aims at creating an alarm based personalized nutrition recommendation system. Modules of the app will include user input, database creation and storage, recommendation of dishes, alarm based system and testing and debugging.

The system targets users who are Indian Women in the age group of 40-50 years.

Upon completion, the new android portable application will include the following:

- Section for basic inputs for Indian Women such as, name, age, gender, height, and weight as well as certain common allergies and diseases prevalent amongst them. This will be followed by a recall activity to know their diet history. Based on these inputs, a 5- time meal plan will be suggested and the alarm will have to be set for the same. At the specified timing, the alarm will ring reminding the user to eat the selected dish.
- Simple language will be used with ingredients of the dishes commonly available at any Indian household.

- The app combines the features of a nutrition and alarm based system.

4. SYSTEM ANALYSIS

4.1. Existing System

Recently, the number of people who use recipe sites is increasing [4] Since professional knowledge of nutrition is necessary to decide daily recipes for improving health, it may be difficult for general users who do not have such knowledge to search for the right recipes. We review four systems that are related to our system. These are [3] SapoFitness: A Mobile Health Application for Dietary Evaluation [4] Implementation of a Goal-Oriented Recipe Recommendation System Providing Nutrition Information, [2] SmartDiet: A Personal Diet Consultant for Healthy Meal Planning.

In the first system (that is the[3]SapoFitness: A Mobile Health Application for Dietary Evaluation) Bruno M. Silva the main author presents SapoFitness, a mobile health application for a dietary evaluation and the implementation of challenges, alerts, and constantly motivates the user to use the system and keep the diet plan. SapoFitness is customized to its user keeping a daily record of his/her food intake and daily exercise. The main goal of this application is to offer a motivation tool for weight reduction and increase physical activity. SapoFitness includes the ability to share personal achievements with social networks, a very intuitive human-device interaction and control weight, applying not only to control obesity but also to malnutrition problems. The application offers a continuous alert system activity, sending alerts/messages concerning the user diet program taking into account also his/her physical activity. SapoFitness is a challenged mobile application that delivers the action to the user, motivating for a healthier life style. The main goal of this application is to offer a motivation tool for weight reduction and increase physical activity. SapoFitness is customized to its user keeping a daily record of his/her food intake and daily exercise[3]The user enters all the necessary information such as his/her height, weight, age, and sex, for determining the BMI (body mass index) and the maximum daily calories must consume. Thus, the system automatically sets a sort of user profile and the system basically tells the user if he/she is or not on overweight and some more information surrounding, such as weight target, date for the purpose, and calories to consume. The application offers a continuous alert system activity, sending alerts/messages concerning the user diet program taking into account also his/her physical activity.

Followed by the second and most impressive system which we came across and which is currently active in japan (that is [4] Implementation of a Goal-Oriented Recipe

Recommendation System Providing Nutrition Information) impressed us to choose our core scope specifically as a recommendation system. In this system, Tsuguya UETA the author of this system proposes a goal-oriented recipe recommendation system that utilizes information about nutrition on the Internet. Their system enables users without knowledge about nutrition to search easily for recipes with natural language to improve specific health conditions. The natural language includes “I want to cure my acne” and “I want to recover from my fatigue”. UETA continues by saying “We compared the results of our system to the results we obtained by calculating the nutrient information manually. Evaluation was done on 1000 dishes. We measured the effectiveness of the system using F-Measure and the average F-measure was 0.64 respectively”. Based on the input of a particular diseased condition, the system enables users without knowledge about nutrition to search easily for recipes. The most impressive field of this system was on the real time basis it gives recipes which cures our everyday health problems through suggestion of food recommendation. The dishes in this system are only Japan based and in their own national language as this system is just available for their own nation.

The third system (that is [2] SmartDiet: A Personal Diet Consultant for Healthy Meal Planning) is a location-aware interactive diet consultant named SmartDiet based on the multi-objective optimization. Jen-Hao Hsiao, the main author of this paper says that, the objective of their research was to achieve nutrient-balanced food recommendations for each individual, while considering individual's requirements at the same time. To reach this goal, they developed a location-aware interactive diet consultant named SmartDiet based on the multi-objective optimization. The proposed personalized diet planning approach not only translates nutrient recommendations into realistic dish choices, but also accepts feedbacks 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.

All the above three systems are quite impressive as far as technology is concerned which helped us come to a solution for our problem statement and well defined scope of our system.

4.2. Proposed System

The proposed system is an android app which aims to provide nutrition recommendations to Indian Women between 40-50 years of age. An added advantage to the system is the alarm based reminder to help the user to take the meals on time followed by medicines if any.

The system relieves general users from the burden of having to identify the right recipes for their condition. The system not only outputs recipes that address the user's health issues, but at the same time, users can search for recipes to improve their health without needing a sophisticated knowledge of nutrition.

The system configuration consist of five steps. First, the system asks for the required biomedical input from the user. Here, the user needs to input his/her name, biological details for calculation of his/her BMI and also the illness he/she is having along with allergies if any. The second step which is known as recall activity, in which user inputs her diet history i.e. what she has been eating since yesterday, from which the system calculates the nutritional intake information. In the third stage, which is the recommendation stage, the system asks the user for the selection of meal for which she need recipes. Our system has 5 categories for meal selection, namely- Breakfast, Morning snacks, Lunch, Evening snacks, Dinner. After the selection of meal, the system processes the input and applies data mining algorithm on the built-in database. The fourth stage is an output phase where the system displays a list of recommended dishes to the user. The user can choose any of these dishes and view their recipe. The final stage of the system is the alarm. Here, the user inputs a specific time along with any medicine name (optional) to set a reminder for the meal he/she has chosen and medicine if any.

4.3. Analysis

Various analysis has been carried out in the project in each of the modules. Also references were studied to understand the required and necessary terms required for the implementation of the system.

Implementation of a Goal-Oriented Recipe Recommendation System Providing Nutrition Information [4] analysis suggested that based on a diseased condition, appropriate food recipes can be recommended to an individual by focusing on the nutrients needed for the

respective condition. This helps cure the condition faster and thereby helps in improving the overall health by correcting the diet consumed.

Survey of general physicians helped us in identifying the most common allergies and illnesses prevalent amongst women in the age group of 40-50 years and also how there is lack of awareness amongst them.

Consultation with a nutritionist helped in knowing the actual methodology which doctors and dieticians use to recommend diet to their patients. The idea of inclusion of diet history so as to know the diet lifestyle of a person came from this.

Research suggested that there is no nutrition app combining the features of a recommendation as well as an alarm reminder system. And this led to the inclusion of an alarm reminder system with voice output which was integrated with the app. The user can now select the dishes from the ones suggested and set an alarm reminder for the same. The voice output specifying the dish to be eaten makes it easier for the user to identify the purpose of the alarm.

The system stands unique from other diet recommendation systems in a way that it combines an alarm based reminder system and a diet recommendation system that recommends purely Indian dishes whose ingredients are commonly available at every household. Also the information provided is crisp and is available as an android app.

4.4. Hardware and software details

Software Details:

- **Android Studio IDE** is the official IDE for Android application development, based on IntelliJ IDEA which offers:
 - Flexible Gradle-based build system
 - Build variants and multiple APK file generation
 - Code templates to help you build common app features
 - Rich layout editor with support for drag and drop theme editing
 - Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine
 - Shows the most important source directories at the top level of the module hierarchy.

- Groups the build files for all modules in a common folder.
 - Groups all the manifest files for each module in a common folder.
 - Shows resource files from all Gradle source sets.
 - Group resource files for different locales, orientations, and screen types in a single group per resource type.
- **SQLite Database:**
 - Simple, easy to use API.
 - Zero-configuration - no setup or administration needed.
 - A complete database is stored in a single **cross-platform** disk file. Great for use as an application file format.
 - Follows **ACID** properties (Atomic, Consistent, Isolated and Durable).
 - Comes with a standalone command-line interface (CLI) client that can be used to administer SQLite databases.
 - Full SQL implementation with advanced features like partial indexes and common table expressions.
 - Supports terabyte-sized databases and gigabyte-sized strings and blobs. (See [limits.html](#).)
 - Small code footprint: less than 500KiB fully configured or much less with optional features omitted.
 - Programming Language: **XML, Java, SQL**

XML:

 - XML separates data from HTML
 - XML simplifies data sharing
 - XML simplifies data transport
 - XML simplifies Platform change
 - XML increases data availability
 - XML can be used to create new internet languages

JAVA

 - Object oriented
 - Syntax based on C++

- It has removed many confusing and/or rarely used features.
- Automatic Garbage Collection
- Platform Independent
- Multi-thread: A thread is like a separate program, executing concurrently
- Architecture-neutral: There is no implementation dependent features

SQL

- SQL is an ANSI and ISO standard computer language for creating and manipulating databases.
- SQL allows the user to create, update, delete, and retrieve data from a database.
- SQL is very simple and easy to learn.
- SQL works with database programs like DB2, Oracle, MS Access, Sybase, MS SQL Server.
- High Speed retrieval of data
- Less Coding Required
- Well Defined Standards Exist for SQL

Android Version: **Ice Cream Sandwich** or Above

Hardware Details:

As the Android app works offline, it involves lot of processing due to which the following hardware specification are meant to be satisfied for the app to run smoothly.

- Smart phone required by end user: **Dual Core Processor**
- Minimum Internal Memory: **4GB**
- RAM Memory: **1GB**
- CPU required for development: **Pentium i3 processor**
- Disk Space: **500GB**

4.5. Methodology

Flowchart depicting systematic steps followed in developing the system is given below.

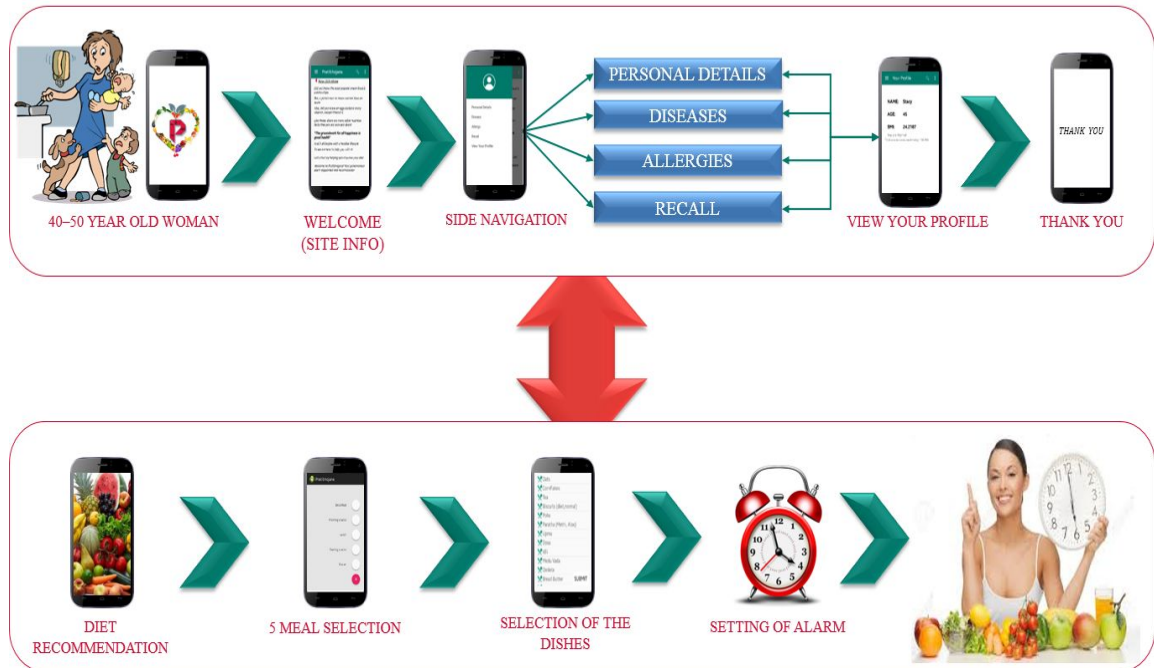


Figure 4.5 System Flowchart

As shown in the Fig. the system starts by asking the user that is Indian Women between 40 to 50 years of age, to give biomedical input namely age, height, weight and their cuisine choice, which is vegetarian or non-vegetarian. In the next step, the app asks for the name and levels of the disease and allergies if any, which the woman is suffering from. Followed with basic inputs the user is next asked to give their diet history in recall activity in which they input what they have eaten the previous day, which helps us to know their regular diet. After completion of the input procedure, the system applies data mining algorithm to offline database to display the recommended list of dishes. The user has a free choice to choose any of the dishes that are recommended to them. After selecting the dish, the app shows an option for reminder for the intake of the dish and also for medicine if any.

4.6. Design details

4.6.1. Architecture

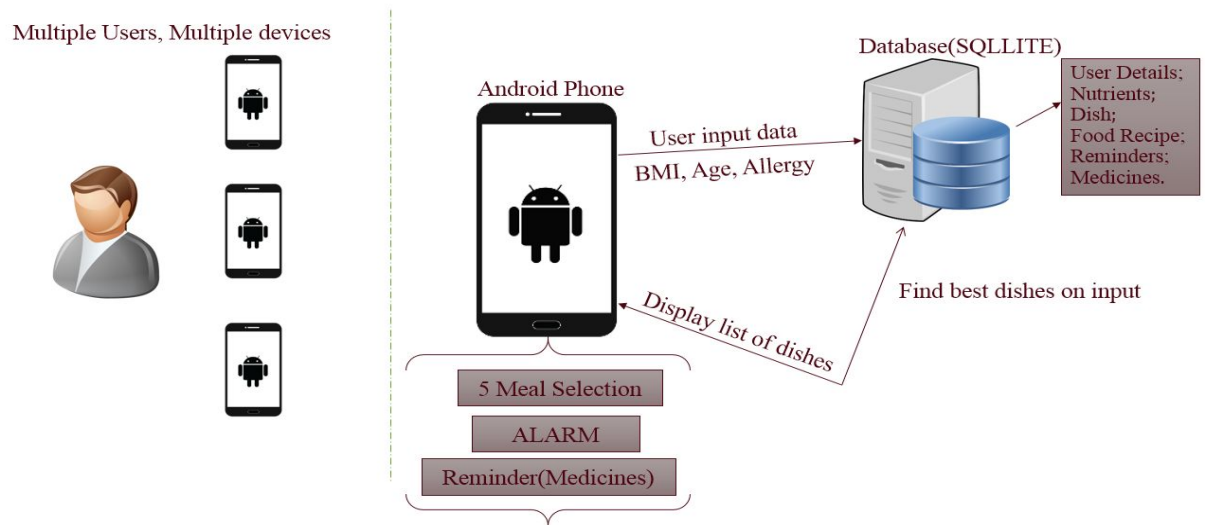


Figure 4.6.1 System Architecture.

The Android App supports multiple users and can be used simultaneously in multiple devices. As shown, the system asks for the required biomedical input from the user. Here, the user needs to input her name, biological details for calculation of her BMI and also the illness she is having along with allergies if any. The app will provide a five meal selection for which the user needs recipes. Our system has 5 categories for meal selection, namely- Breakfast, Morning snacks, Lunch, Evening snacks, Dinner. Then, the system processes the input and applies data mining algorithm on the built-in SQLite database. The next stage is the output phase where the system displays a list of recommended dishes to the user. The user can choose any of these dishes and view their recipe. The final stage of the system is the alarm. Here, the user inputs a specific time along with any medicine name (optional) to set a reminder for the meal she has chosen and medicine if any.

4.7. Formulae and algorithm

Consultation with a nutritionist helped in the inclusion of diet history as part of the recommendation because that is how they begin their recommendation to patients visiting them.

- **BMI is calculated in the following way:**

$$\text{BMI} = \text{Weight (kg)} / \text{Height (m}^2\text{)}$$

After calculation of BMI, the value falls into one of the three categories;

Underweight; Normal; Overweight

- **Calorie needs for the day is calculated in the following way:**

$$\text{Calorie needs} = \text{BMR} * \text{Activity Factor}$$

$$\text{BMR} = (10 \times \text{weight (kgs)}) + (6.25 \times \text{height (cm)}) - (5 \times \text{age (years)}) - 161$$

Activity Factor:

- 1) Sedentary: 1.2
- 2) Moderately active: 1.55
- 3) Active: 1.725

- **Amount of nutrients consumed are calculated in the following way:**

$$\text{Nutrient consumed} = \text{Multiplication factor} * \text{Nutrients contained in 100 grams/ 1tsp}$$

- **Recommendation of dishes:**

According to the user's BMI category and diseases, the dishes for recommendation are identified. In the database, each disease has a set of dishes for recommendation. If the user has more than one disease, the common dishes from each of these sets are selected.

These set of common dishes are compared with allergy provided by the user. The dishes which contain ingredients that the user is allergic to, are removed from the list, to get the final set of dishes for recommendation.

- **Algorithm**

Step1: Initialize array 'list'. If there is more than one disease go to step 2, else go to Step4.

Step2: Find array list of dishes for a disease and compare with array 'list'. Store the common dishes from both these arrays in the 'list' itself. Go to Step3.

Step3: Repeat Step2 for all diseases. Then go to Step5.

Step4: Find set of dishes for the given single disease and store them in array 'list'. Then go to Step5.

Step5: Compare array 'list' with the allergies of the user. Remove those allergic dishes from 'list' and display the final set of dishes to the user as recommendation.

5. IMPLEMENTATION

5.1. Module-I Data collection and design of GUI

Data collection first involved conducting a survey of general physicians and nutritionists to find out the most commonly existing diseases and allergies amongst women in the age group of 40-50 years. This gave the list of allergies and diseases which we included in the system. From the internet, the most common levels of the diseases was found out and considered. Also one factor that was highlighted was that BMI plays an important role when it comes to diet recommendation because it gives the amount of body fat. We thereby included it in the system thus providing users an insight of their BMI value and which category they belong to, namely; underweight, normal or overweight. GUI design included:

- The main GUI of the system was developed in the first module.
- The GUI of the system helps the user to provide input to the system, navigate to different activity and get diet recommendation from the system.
- The GUI was designed using XML coding with an option for drag/drop of GUI components present in Android Studio IDE.

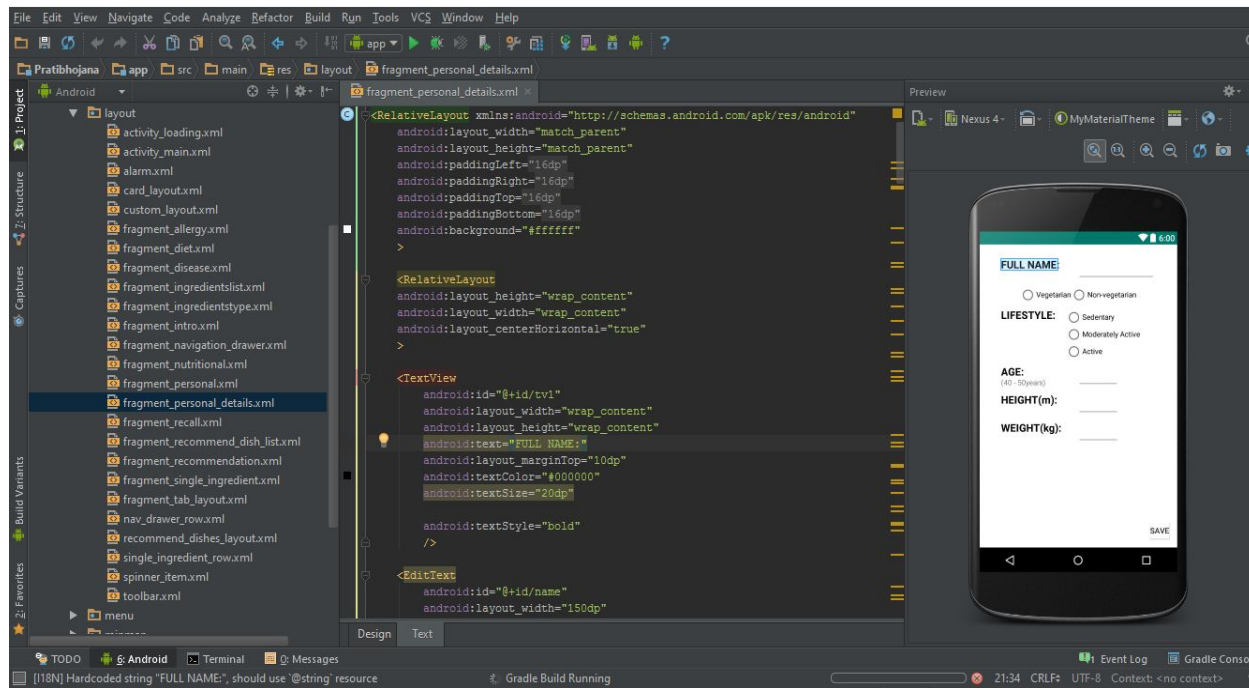


Figure 5.1: System GUI Design

5.2. Module-II Back-End Coding and Database Design

- The main objective of this module is to integrate the GUI of the app with the database present in the backend.
- This module is implemented using Java.
- Here the input taken from the user is saved in the database tables.
- Calculation of BMI value and its classification is done here.
- The constraint validation of the input also takes place here.
- With the help of this module, output of the system (Diet recommendation) is found based on the user input.

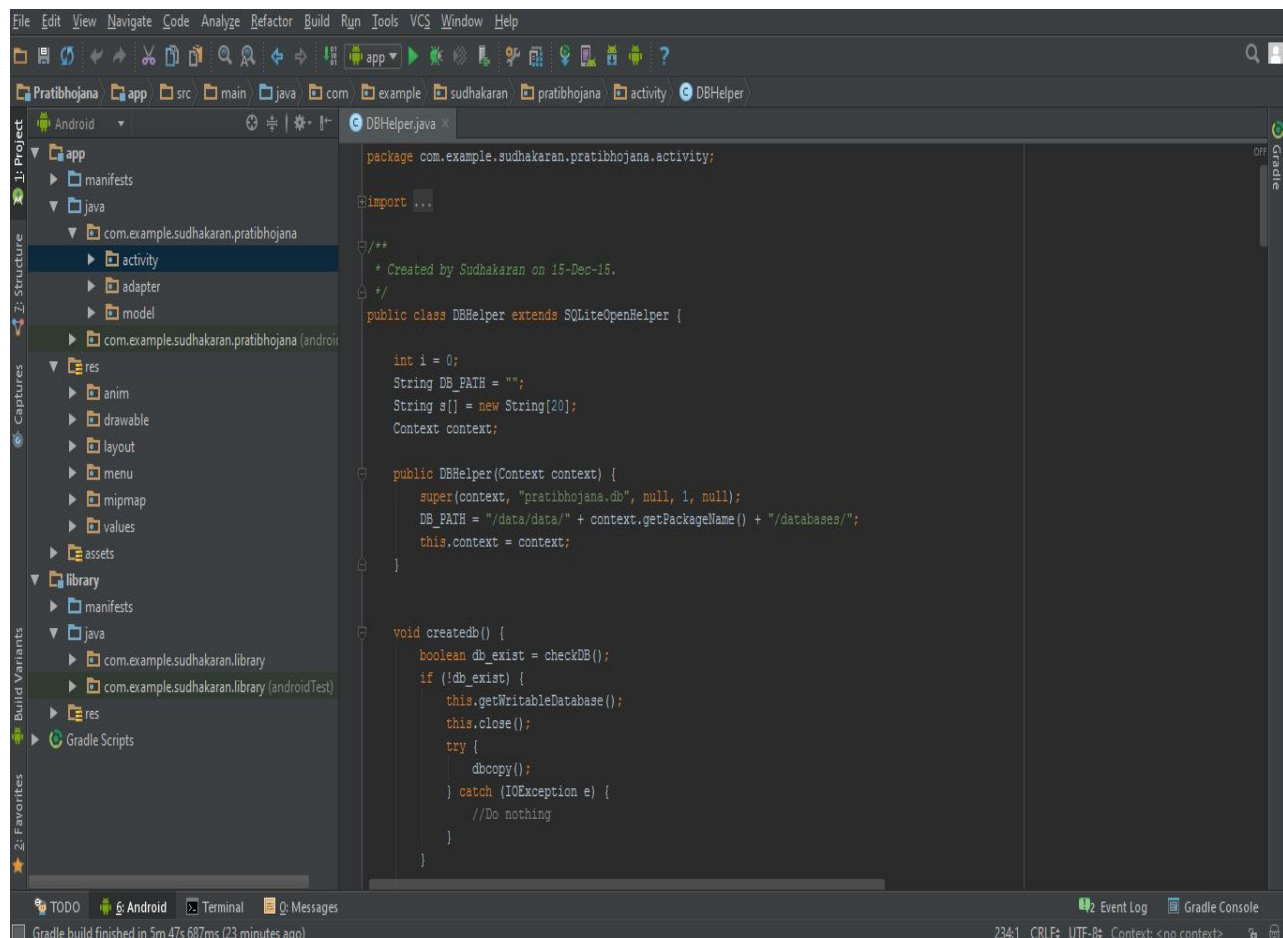


Figure 5.2.1: Backend Module

5.2.1. Database Design

- The main objective of this module is to save the necessary data for the system along with user data.
- This module is implemented using GUI based software SQLite Browser.
- The database file also has a facility for auto-upgradation in case of app updating.

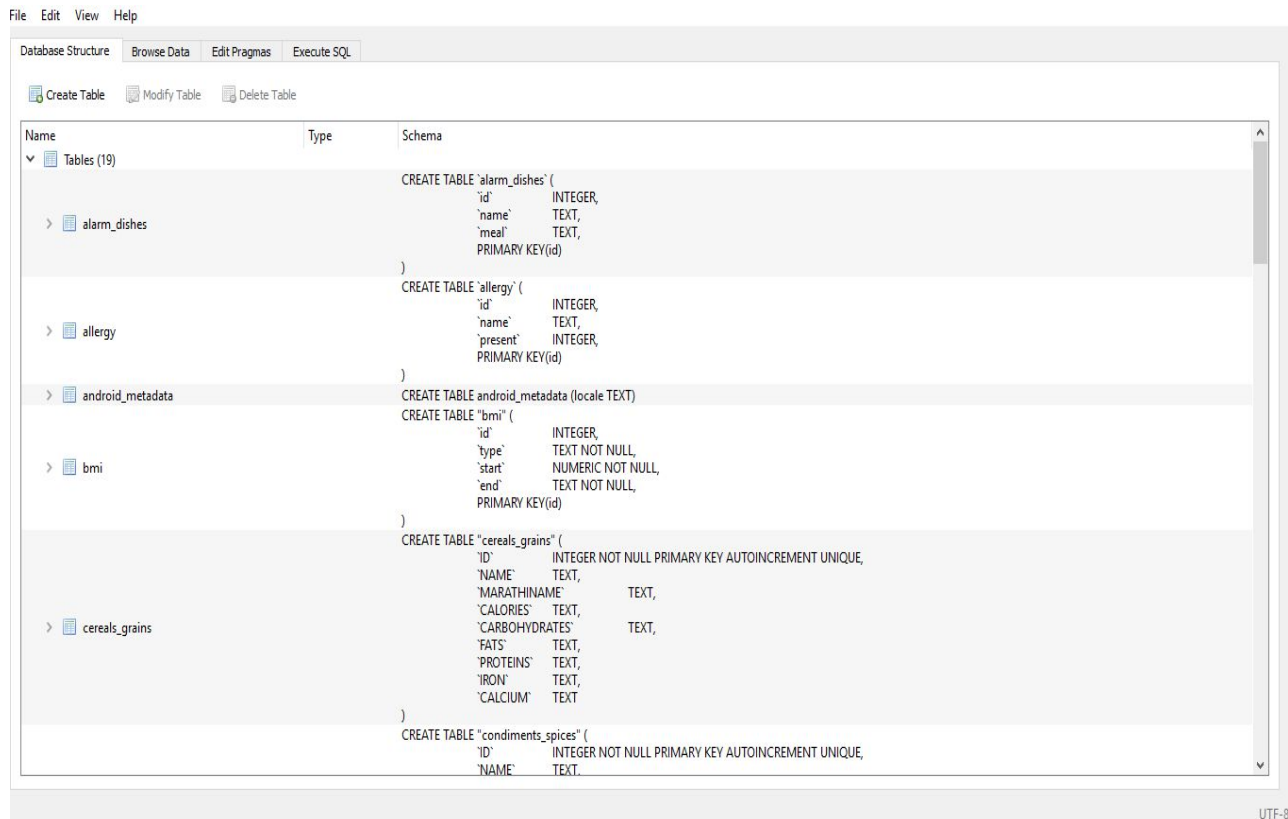


Figure 5.2.2: Database Design

Nutritional Information:

Any food item that we eat, consists of ingredients that fall under the nine categories of food; namely;

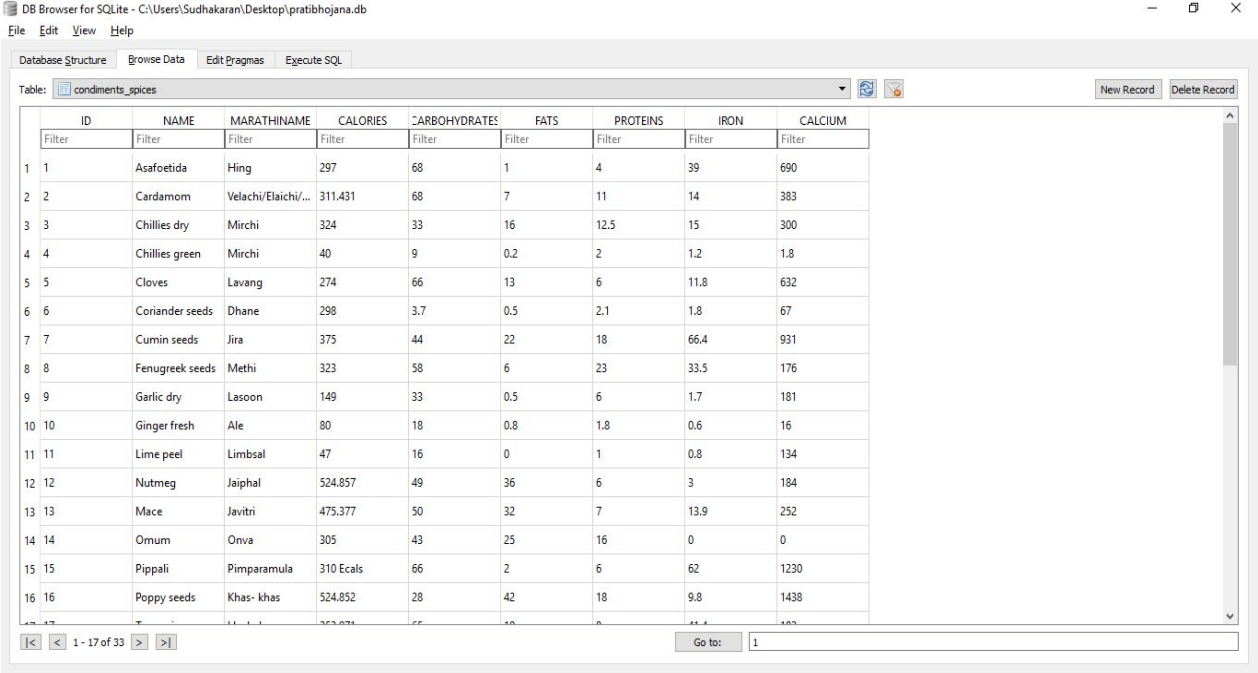
1. Cereals and grains
2. Pulses and legumes
3. Vegetables
4. Fruits

5. Fats, sugars and edible oils
6. Condiments and spices
7. Nuts
8. Dairy products
9. Meat and seafood.

The nutritional part of the database consists of these nine tables, one for each category of food. Amongst all the ingredients available and used, the ones included in the ingredient tables were those that are commonly available in an Indian household.

Each table consists of ingredient names with the amount of nutrients (proteins, carbohydrates, fats, iron and calcium) contained in them in a given specific quantity. This information was collected from standard nutrition websites such as WHO and others.

Also the amount of calories obtained from every ingredient has an entry in the ingredient table. This data was used to calculate the amount of calories consumed and nutrients contained in the diet of the user the previous day which helps in getting an idea of the diet lifestyle.



DB Browser for SQLite - C:\Users\Sudhakaran\Desktop\pratibhojana.db

File Edit View Help

Database Structure Browse Data Edit Pragmas Execute SQL

Table: condiments_spices

ID	NAME	MARATHINAME	CALORIES	CARBOHYDRATE	FATS	PROTEINS	IRON	CALCIUM
1	Asafoetida	Hing	297	68	1	4	39	690
2	Cardamom	Velachi/Elaichi/...	311.431	68	7	11	14	383
3	Chillies dry	Mirchi	324	33	16	12.5	15	300
4	Chillies green	Mirchi	40	9	0.2	2	1.2	1.8
5	Cloves	Lavang	274	66	13	6	11.8	632
6	Coriander seeds	Dhane	298	3.7	0.5	2.1	1.8	67
7	Cumin seeds	Jira	375	44	22	18	66.4	931
8	Fenugreek seeds	Methi	323	58	6	23	33.5	176
9	Garlic dry	Lasoon	149	33	0.5	6	1.7	181
10	Ginger fresh	Ale	80	18	0.8	1.8	0.6	16
11	Lime peel	Limbsal	47	16	0	1	0.8	134
12	Nutmeg	Jaiphal	524.857	49	36	6	3	184
13	Mace	Javitri	475.377	50	32	7	13.9	252
14	Onum	Onva	305	43	25	16	0	0
15	Pippali	Pimparamula	310	66	2	6	62	1230
16	Poppy seeds	Khas- khas	524.852	28	42	18	9.8	1438

Go to: 1

UTF-8

Figure 5.2.3: Ingredient Table

6. RESULTS



Figure 6.1 Loading page

Figure 6.1 shows the first activity of the proposed app which consists of the official logo and title of the app. It displays the name of the proposed system which is Pratibhojana. The word Pratibhojana is derived from the Sanskrit word prescribed diet. Just as a doctor prescribes medicines to patients, in the same way, the proposed system aims at recommending healthy nutritious food to women based on some standard inputs. The proposed system enables women without knowledge about good nutrition to search for simple Indian recipes filtered on the basis of some standard input conditions taken by the system.

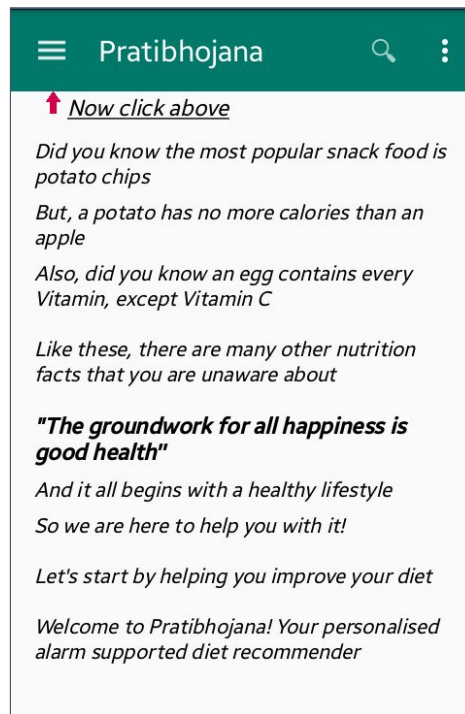


Figure 6.2 Home page

The above figure is the Home page of our app which tells what our app is about and what it does. It also indicates where to click to get started with the app.

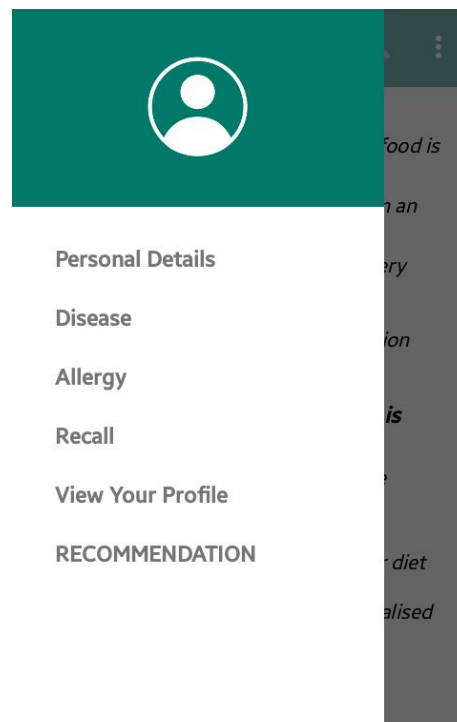


Figure 6.3 Fragment Slider

The above slider helps the user to navigate to the desired page. The pages can either ask the user to give her medical data or recommend dishes with the facility to set alarm. The slider can be opened from any of the pages by simply clicking the toolbar located at the top left corner of the screen.

The screenshot shows a mobile application interface for 'Personal Details'. At the top is a teal header bar with a hamburger menu icon, the text 'Personal Details', a magnifying glass icon, and a vertical ellipsis icon. Below the header, the form consists of several sections: 'FULL NAME:' followed by a text input field; a radio button selection for 'Vegetarian' and 'Non-vegetarian'; 'LIFESTYLE:' followed by radio button selections for 'Sedentary', 'Moderately Active', and 'Active'; 'AGE:' followed by a text input field and the text '(40 - 50years)' below it; 'HEIGHT(m):' followed by a text input field; and 'WEIGHT(kg):' followed by a text input field. At the bottom right of the form is a 'SAVE' button.

Figure 6.4 User Information

Figure 6.4 asks the user to input her name, cuisine preference, age, height and weight. The age of the user should be in the age group of 40-50 years as the app is intended for this age group only. The height and weight specified by the user must be in metres and kilograms respectively. Usually from the height and weight, the BMI is calculated which is then used for deriving certain health results. All the inputs must be properly filled. On clicking the SAVE button all the information given by the user are inserted into the database after the above constraints are satisfied successfully.

Menu Disease Search

Please select a disease/s if any from the below mentioned list:

- ☐ Anemia
- ☐ Diabetes
- ☒ Thyroid

Choose your level..
 Hypothyroidism
 Hyperthyroidism
- ☐ Cholesterol
- ☐ Blood Pressure
- ☐ Osteoarthritis
- ☐ None of these

SAVE

Figure 6.5 Selection of Diseases

Figure 6.5 shows the second set of input categories which are illnesses. This list is based on the survey of doctors conducted to find the most common illnesses prevalent amongst women in the age group of 40-50 years. The user is expected to select the allergy names that she might be suffering from else, none of these option.

Menu Allergy Search

Please select an allergy/ies if any from the below mentioned list:

- ☐ Sea Food
- ☒ Nuts
- ☒ Milk
- ☐ Acidity
- ☐ None of these

SAVE

Figure 6.6 Selection of allergies

The above figure is the allergy page. Here the user inputs the allergy/ies if any. After clicking the SAVE button, the allergies will be recorded in the database. These food containing these allergic ingredients will be avoided during recommendation to the user.



Figure 6.7 Recall to know diet history

The above recall page asks the user what she has eaten the previous day. The search bar at the top of the page helps the user to search and select the dishes efficiently. On selecting a dish, the below page is opened.

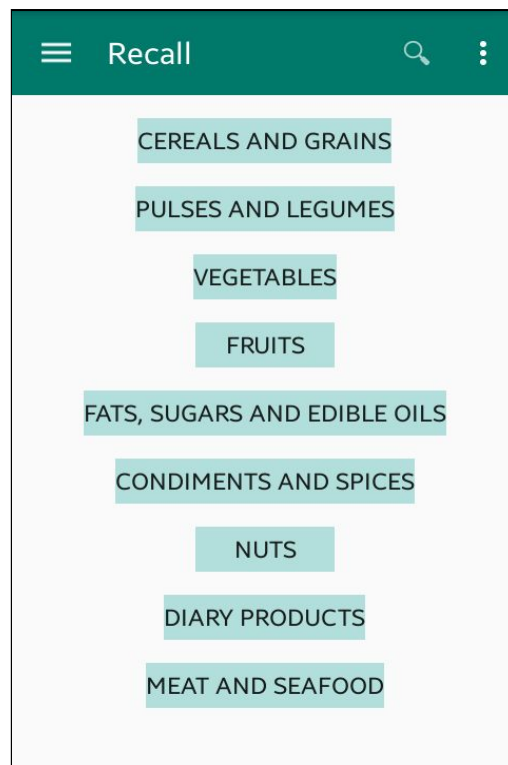


Figure 6.8 Type of ingredients in the recall

The above page asks the user to choose the category of ingredients which were used to prepare the dishes the previous day.

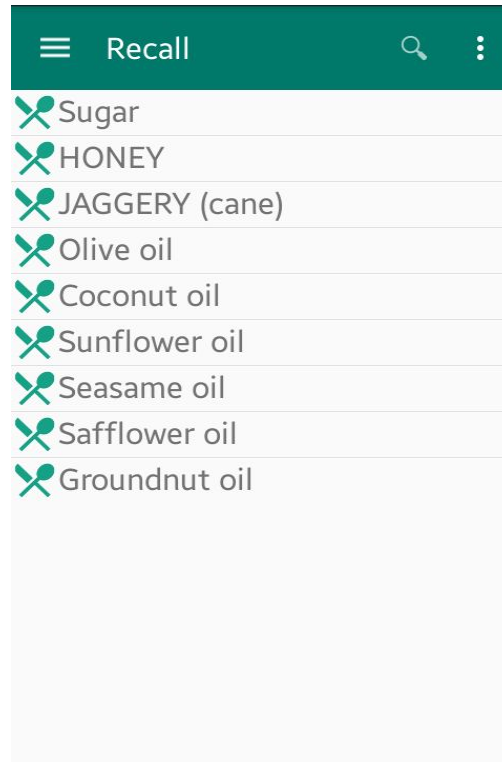


Figure 6.9 Ingredients in the recall

The above page shows the list of ingredients for the chosen ingredient type. The user can select the ingredients that has been put to prepare the dish.

Figure 6.10 Ingredients quantity in the recall

After choosing the appropriate ingredient, the user is asked its amount she has put to prepare the dish. The page also displays the calorific content of the ingredient per 100g. After user inputs the amount, the total calories is calculated and displayed below.

Figure 6.11 User Profile Page (Personal)

The personal tab in the user profile page displays the name, age, BMI, diseases and allergies entered by the user.

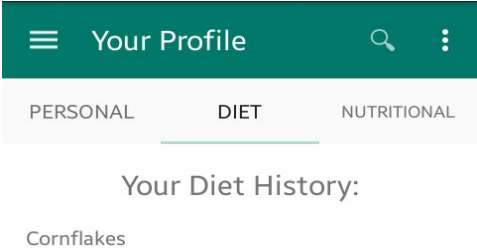


Figure 6.12 User Profile Page (Diet)

The diet tab in the user profile page displays the dishes eaten by the user the previous day.

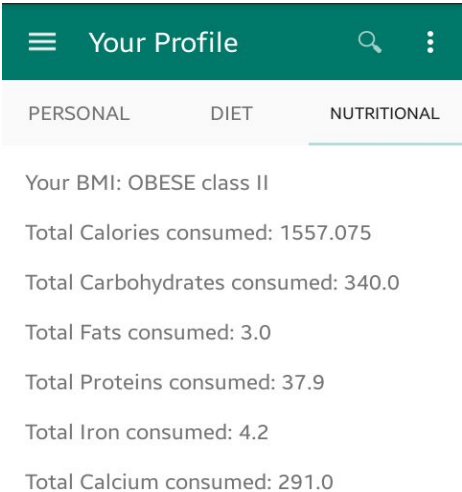


Figure 6.13 User Profile Page (Nutritional)

The nutritional tab in the user profile page displays the BMI category of the user along with total calories, carbohydrates, proteins, fats, iron and calcium consumed by user the previous day.

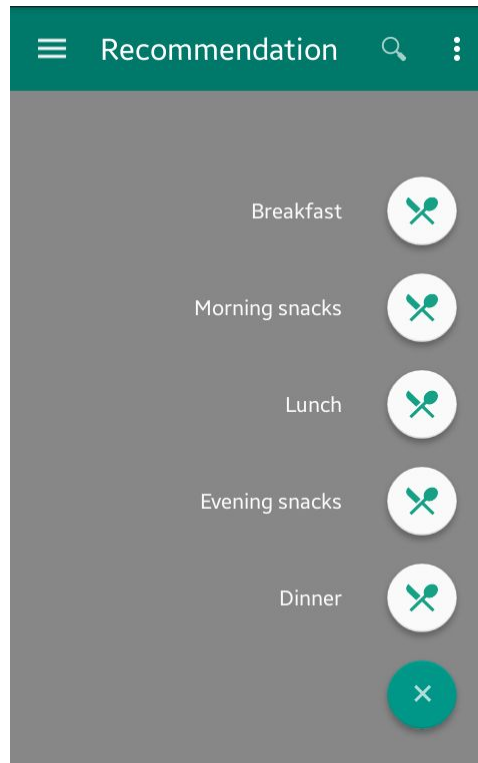


Figure 6.14 Recommendation page

In the above page the user can choose any of the five meals for dish recommendation. After the final step, that is, setting alarm is done, the name of the chosen dish is also displayed over here.

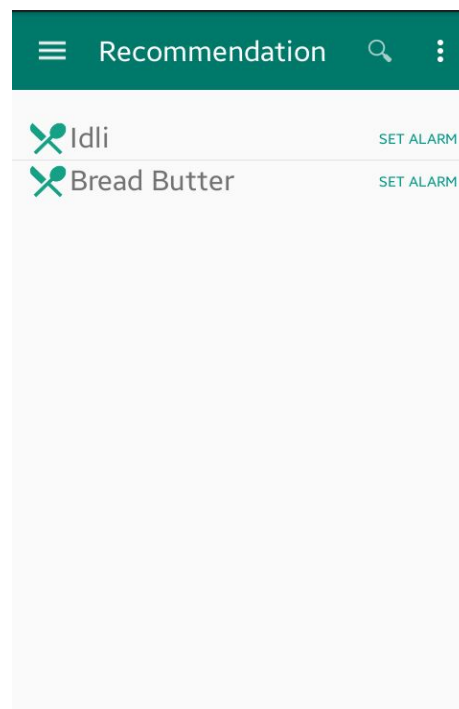


Figure 6.15 Recommendation page (Setting alarm)

This page is responsible for recommending the appropriate dish for the user with respect to the meal chosen. It also provides an option of setting alarm for the favourite dish of the user. If the user clicks the SET ALARM button, below pop-up window is opened.

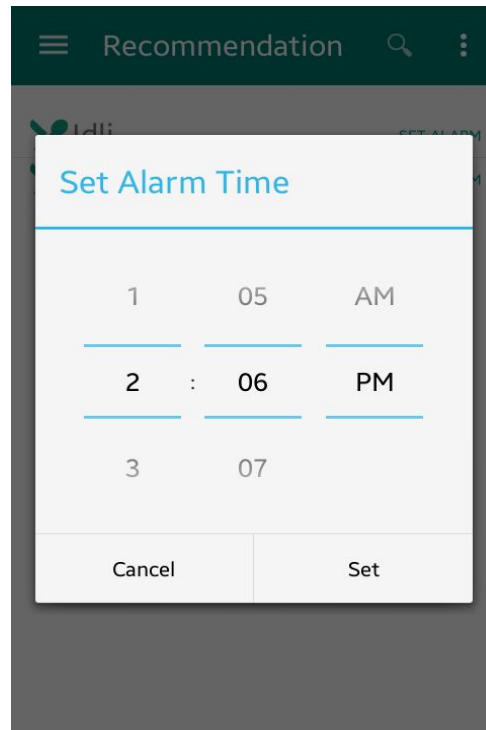


Figure 6.16 Recommendation page (Alarm clock pop-up)

After clicking on the SET ALARM button for any dish, the above pop-up window opens to set the alarm time for the chosen dish.

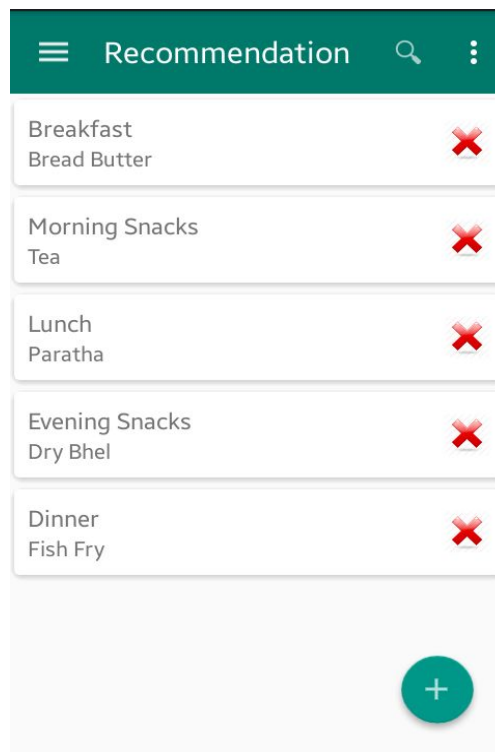


Figure 6.17 Recommendation page (With Alarm set)

The above figure shows the recommendation page after alarms have been set for a dish in each meal category.

7. PROJECT TIMELINE AND TASK DISTRIBUTION

SR. NO.	MONTH	DETAILS OF IMPLEMENTATION
1.	June	Studied various papers and existing nutrition systems to find out missing parameters in them.
2.	July	Establishment and initiation of project plan.
3.	August	Research including consultation with a nutritionist for the input parameters to be considered for diet recommendation.
4.	September	Development of alarm reminder system including voice output.
5.	October	IEEE paper and Project Report A completion.
6.	December	Survey of general physicians to find out the most common allergies and illnesses amongst women in the age group of 40-50 years.
7.	January	Development and integration of Material Design for the User Interface of app.
8.	February	Information gathering and corresponding designing of SQLite database.
9.	March	Constraint validation, working on the errors in the system, minor bug fixing.

Table 1: Project Timeline

Tasks	A.J.Sudhakaran	Stacy Bardeskar	Mohit Deorukhkar
Initiate project		✓	✓
Establish Project Plan	✓	✓	
Information Gathering		✓	✓
Survey Of Doctors	✓	✓	
Material Designing (UI)			✓
Database	✓	✓	
Backend Coding	✓		✓
Documentation		✓	✓
Presentation	✓	✓	✓

Table 2: Task Distribution

8. CONCLUSION

The survey of general physicians threw light that there is lack of awareness amongst women about the importance of nutrition particularly when they enter the age group of 40-50 where they are most likely to develop unhealthy conditions. The importance of nutrition needs to be realized by every woman. Necessary changes in the lifestyle of every woman has to be incorporated in order to ensure a healthy living. A healthy living comes from a healthy diet. For this reason, women need to get some sort of guidance for the right diet for their condition. And to support that, this project described an alarm based personalized nutrition recommendation system for women in the age group of 40-50 years.

Good nutrition starts with the basics: a well-rounded diet consisting of whole grains, fresh fruits and vegetables, healthy fats, and high quality sources of protein. These kinds of foods provide women with plenty of energy, the means for lifelong weight control, and the key ingredients for looking and feeling great at any age. Instead of obsessing over specific foods or nutrients, it's the overall eating pattern that's most important. Diet has a major effect on an individual's food cravings, stress levels, and energy throughout the day. By making smart food choices and developing healthy eating habits, it's much easier to stay slim, control cravings, and feel energetic all day long.

In this way Pratibhojana aims at playing a good nutritional role in the lives of women in the age group of 40-50 by improving their daily diet. The system relieves users the burden of having to identify the right diet as per their body conditions and nutritional needs and will save them the time of browsing the internet for appropriate diet recipes. An additional advantage of the proposed system is the alarm reminder for food. Therefore, this proposed system, is a fine attempt to build an efficient diet recommendation system that provides home made Indian recipes that will help improve the nutritional status of households. Future work may include the alarm reminder system being made multipurpose and with a better voice output. Also the app can be made online in future to incorporate more amount of data. Another task of future research is to make the system for infants as well as the elderly.

Since, Life can be hectic, and sometimes it's hard to take the time to make healthy food choices. But making wise food choices—along with regular physical activity—can offer big benefits, now and in the future.

9. REFERENCES

- [1] Developing a Nutrition and Diet Expert System Prototype, CONFERENCE PAPER
· JUNE 2013 [online]. Available:
http://www.researchgate.net/publication/244993766_Developing_a_Nutrition_and_Diet_Expert_System_Prototype [Accessed: September 2015]
- [2] Jen-Hao Hsiao; Chang, H., "SmartDiet: A personal diet consultant for healthy meal planning," in Computer-Based Medical Systems (CBMS), 2010 IEEE 23rd International Symposium on, vol., no., pp.421-425, 12-15 Oct. 2010.
- [3] Silva, B.M.; Lopes, I.M.; Rodrigues, J.J.P.C.; Ray, P., "SapoFitness: A mobile health application for dietary evaluation," in e-Health Networking Applications and Services (Healthcom), 2011 13th IEEE International Conference on, vol., no., pp.375-380, 13-15 June 2011.
- [4] Ueta, T.; Iwakami, M.; Ito, T., "Implementation of a Goal-Oriented Recipe Recommendation System Providing Nutrition Information," in Technologies and Applications of Artificial Intelligence (TAAI), 2011 International Conference on , vol., no., pp.183-188, 11-13 Nov. 2011.
- [5] Ziyu Lv; Feng Xia; Guowei Wu; Lin Yao; Zhikui Chen, "iCare: A Mobile Health Monitoring System for the Elderly," in Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom) , vol., no., pp.699-705, 18-20 Dec. 2010.
- [6] Mother, Infant and Young Child Nutrition and Malnutrition [online]. Available: <http://motherchildnutrition.org> [Accessed: August 2015]
- [7] India's Rapid Survey on Children, 2013-14 [online]. Available:
<http://www.economist.com/blogs/graphicdetail/2015/07/daily-chart-0> [Accessed: September 2015]
- [8] [Nutritionssocietyindia.org](http://www.nutritionssocietyindia.org), (2016). *Nutrition Society of India*. [online] Available at: <http://www.nutritionssocietyindia.org/> [Accessed 15 Dec. 2015].

[9] Nutritive Value of Indian Foods (NVIF) By C. Gopalan, B. v. Rama Sastri & S.C. Balasubramanian, Revised & Updated (1989) by B.S. narasinga Rao, Y.G. Deosthala & K.C. Pant.

APPENDIX

Source code

Main Activity

XML code:

```
<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/drawer_layout"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
```

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">
```

```
<LinearLayout
  android:id="@+id/container_toolbar"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:orientation="vertical">
```

```
<include
  android:id="@+id/toolbar"
  layout="@layout/toolbar" />
```

```
</LinearLayout>
```

```
<FrameLayout
  android:id="@+id/container_body"
  android:layout_width="fill_parent"
  android:layout_height="0dp"
  android:layout_weight="1" />
```

```
</LinearLayout>
```

```
<fragment
    android:id="@+id/fragment_navigation_drawer"
    android:name="com.example.sudhakaran.pratibhojana.activity.FragmentDrawer"
    android:layout_width="@dimen/nav_drawer_width"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    app:layout="@layout/fragment_navigation_drawer"
    tools:layout="@layout/fragment_navigation_drawer" />
```

```
</android.support.v4.widget.DrawerLayout>
```

JAVA code:

```
package com.example.sudhakaran.pratibhojana.activity;
```

```
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.app.TimePickerDialog;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TimePicker;
import android.widget.Toast;
```

```
import com.example.sudhakaran.pratibhojana.R;
```

```

import com.example.sudhakaran.pratibhojana.model.AlarmReceiver;

import java.util.Calendar;

public class MainActivity extends AppCompatActivity implements
FragmentManager.FragmentDrawerListener,fragmentlistener {

    private static String TAG = MainActivity.class.getSimpleName();

    private Toolbar mToolbar;
    private FragmentDrawer drawerFragment;
    static int RQS_code = 0;
    static int CRQS_code = 0;
    Fragment fragment=null;
    String title,speech;
    FragmentManager fragmentManager = getSupportFragmentManager();
    PersonalDetailsFragment personalDetailsFragment;
    DiseaseFragment diseaseFragment;
    AllergyFragment allergyFragment;
    RecallFragment recallFragment;
    RecommendationFragment recommendationFragment;
    IntroFragment introFragment;
    TimePickerDialog timePickerDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mToolbar = (Toolbar) findViewById(R.id.toolbar);

        setSupportActionBar(mToolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        personalDetailsFragment = new PersonalDetailsFragment(getBaseContext());
        diseaseFragment = new DiseaseFragment(getBaseContext());
        allergyFragment = new AllergyFragment(getBaseContext());
        recallFragment = new RecallFragment(this,getBaseContext());
        recommendationFragment = new RecommendationFragment(this,getBaseContext());
        introFragment = new IntroFragment(getBaseContext());

```

```

        drawerFragment = (FragmentManager)

getSupportFragmentManager().findFragmentById(R.id.fragment_navigation_drawer);
        drawerFragment.setUp(R.id.fragment_navigation_drawer, (DrawerLayout)
findViewById(R.id.drawer_layout), mToolbar);
        drawerFragment.setDrawerListener(this);

        // display the first navigation drawer view on app launch
        displayView(6);
    }

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    if(id == R.id.action_search){
        Toast.makeText(getApplicationContext(), "Search action is selected!",
Toast.LENGTH_SHORT).show();
        return true;
    }

    return super.onOptionsItemSelected(item);
}

```

```

@Override
public void onDrawerItemSelected(View view, int position) {
    displayView(position);
}

private void displayView(int position) {
    fragment = null;
    title = getString(R.string.app_name);
    switch (position) {
        case 0:
            fragment = personalDetailsFragment;
            title = getString(R.string.title_personaldetails);
            break;
        case 1:
            fragment = diseaseFragment;
            title = getString(R.string.title_disease);
            break;
        case 2:
            fragment = allergyFragment;
            title = getString(R.string.title_allergy);
            break;
        case 3:
            fragment = recallFragment;
            title = getString(R.string.title_recall);
            break;
        case 4:
            fragment = new TabLayoutFragment(getBaseContext());
            title = getString(R.string.title_yourprofile);
            break;
        case 5:
            fragment = new RecommendationFragment(this, getBaseContext());
            title = getString(R.string.title_recommendation);
            break;
        case 6:
            fragment = introFragment;
            title = getString(R.string.app_name);
            break;

        default:
            break;
    }
}

```

```

    }

    if (fragment != null) {
        setView();
    }
}

@Override
public void setingredientlist(String type,String dish) {
    fragment = new IngredientslistFragment(getBaseContext(),type,dish,this);
    title = getString(R.string.title_recall);
    setView();
}

@Override
public void setcategory(String dish) {
    fragment = new IngredientstypeFragment(this, getBaseContext(),dish);
    title = getString(R.string.title_recall);
    setView();
}

@Override
public void setsingleingredient(String type,String ingredient,String dish) {
    fragment = new SingleIngredientFragment(getBaseContext(),type,ingredient,dish);
    title = getString(R.string.title_recall);
    setView();
}

@Override
public void setrecommenddishes(String meal, int RQS_code) {
    fragment = new RecommendDishListFragment(getBaseContext(),meal,this);
    this.RQS_code = RQS_code;
    setView();
}

@Override
public void setrecommend() {
    fragment = new RecommendationFragment(this, getBaseContext());
    setView();
}

```

```

public void openTimePickerDialog(boolean is24r, String speech) {
    Calendar calendar = Calendar.getInstance();
    this.speech = speech;
    timePickerDialog = new TimePickerDialog(
        this,
        onTimeSetListener,
        calendar.get(Calendar.HOUR_OF_DAY),
        calendar.get(Calendar.MINUTE),
        is24r);
    timePickerDialog.setTitle("Set Alarm Time");

    timePickerDialog.show();

}

TimePickerDialog.OnTimeSetListener onTimeSetListener
    = new TimePickerDialog.OnTimeSetListener() {

    @Override
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {

        Calendar calNow = Calendar.getInstance();
        Calendar calSet = (Calendar) calNow.clone();

        calSet.set(Calendar.HOUR_OF_DAY, hourOfDay);
        calSet.set(Calendar.MINUTE, minute);
        calSet.set(Calendar.SECOND, 0);
        calSet.set(Calendar.MILLISECOND, 0);

        if (calSet.compareTo(calNow) <= 0) {
            //Today Set time passed, count to tomorrow
            calSet.add(Calendar.DATE, 1);
        }

        setAlarm(calSet, speech);
    }
};

@Override

```

```

public void setAlarm(Calendar targetCal, String speech) {
    Intent intent = new Intent(getBaseContext(), AlarmReceiver.class);
    String meal="";
    if(RQS_code==1)
        meal="Breakfast";
    else if(RQS_code == 2)
        meal = "Morning Snacks";
    else if(RQS_code == 3)
        meal = "Lunch";
    else if(RQS_code == 4)
        meal = "Evening Snacks";
    else if(RQS_code == 5)
        meal = "Dinner";
    intent.putExtra("speech", speech);
    intent.putExtra("meal", meal);
    PendingIntent pendingIntent = PendingIntent.getBroadcast(getBaseContext(),
RQS_code, intent, 0);
    AlarmManager alarmManager = (AlarmManager)
getSystemService(Context.ALARM_SERVICE);
    alarmManager.set(AlarmManager.RTC_WAKEUP, targetCal.getTimeInMillis(),
pendingIntent);
    Toast.makeText(this,"Alarm set!",Toast.LENGTH_SHORT).show();
    setrecommend();
}

public void cancelAlarm(String meal, String speech){

    if(meal.equals("Breakfast"))
        CRQS_code = 1;
    else if(meal.equals("Morning Snacks"))
        CRQS_code = 2;
    else if(meal.equals("Lunch"))
        CRQS_code = 3;
    else if(meal.equals("Evening Snacks"))
        CRQS_code = 4;
    else if(meal.equals("Dinner"))
        CRQS_code = 5;

    Intent intent = new Intent(getBaseContext(), AlarmReceiver.class);
    intent.putExtra("speech", speech);

```



```

        intent.putExtra("meal", meal);
        PendingIntent pendingIntent = PendingIntent.getBroadcast(getBaseContext(),
CRQS_code, intent, 0);

        AlarmManager alarmManager =
(AlarmManager)getSystemService(Context.ALARM_SERVICE);
        alarmManager.cancel(pendingIntent);
        Toast.makeText(this,"Alarm Cancelled!",Toast.LENGTH_SHORT).show();
    }

    public void setView(){
        FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
        //fragmentTransaction.setCustomAnimations(R.anim.animation,R.anim.animation2);
        fragmentTransaction.replace(R.id.container_body, fragment);
        fragmentTransaction.addToBackStack(null);
        fragmentTransaction.commit();
    }

    public void setActionBarTitle(int title)
    {
        getSupportActionBar().setTitle(title);
    }
}

```

Database connectivity code

```

package com.example.sudhakaran.pratibhojana.activity;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import com.example.sudhakaran.pratibhojana.model.AlarmDishes;

import java.io.File;
import java.io.FileOutputStream;

```

```

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;

/**
 * Created by Sudhakaran on 15-Dec-15.
 */
public class DBHelper extends SQLiteOpenHelper {

    int i = 0;
    String DB_PATH = "";
    String s[] = new String[20];
    Context context;

    public DBHelper(Context context) {
        super(context, "pratibhojana.db", null, 1, null);
        DB_PATH = "/data/data/" + context.getPackageName() + "/databases/";
        this.context = context;
    }

    void createdb() {
        boolean db_exist = checkDB();
        if (!db_exist) {
            this.getWritableDatabase();
            this.close();
            try {
                dbcopy();
            } catch (IOException e) {
                //Do nothing
            }
        }
    }

    boolean checkDB() {
        File dbfile = new File(DB_PATH + "pratibhojana.db");
        return dbfile.exists();
    }

```

```
}
```

```
@Override  
public void onCreate(SQLiteDatabase db) {  
    //db.execSQL("Create table family (id integer primary key, name text);");  
}
```

```
@Override  
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
  
}
```

```
public void dbcopy() throws IOException {  
    OutputStream os = new FileOutputStream(DB_PATH + "pratibhojana.db");  
    InputStream is = context.getAssets().open("pratibhojana.db");  
    byte[] buffer = new byte[1024];  
    int length;  
    while ((length = is.read(buffer)) > 0) {  
        os.write(buffer, 0, length);  
    }  
    os.flush();  
    is.close();  
    os.close();  
}
```

```
public void insertuserdata(String name, int age, String cuisine, float bmi) {  
    createdb();  
    SQLiteDatabase db = this.getWritableDatabase();  
    ContentValues values = new ContentValues();  
    values.put("user_id", 1);  
    values.put("user_name", name);  
    values.put("user_age", age);  
    values.put("user_bmi", bmi);  
    values.put("bmi_classification", getbmitype(bmi));  
    values.put("user_cuisine", cuisine);  
    db.update("USER_DETAILS", values, "USER_ID=1", null);  
}
```

```

        //db.execSQL("INSERT OR REPLACE INTO USER_DETAILS (USER_ID,
        USER_NAME, USER_AGE, USER_CUISINE, USER_BMI, BMI_CLASSIFICATION)
        VALUES(" + 1 + ",\"" + name + "\", " + age + ", " + bmi + "\",\"" + getbmitype(bmi) + "\"");

```

```

    }

```

```

    public Cursor getuserdetails() {
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor c = db.rawQuery("select * from user_details where user_id=1", null);
        return c;
    }

```

```

    public Cursor getingredients(String type) {
        SQLiteDatabase db = getWritableDatabase();
        Cursor c = db.rawQuery("select name from " + type, null);
        return c;
    }

```

```

    public Cursor getrecalldishes() {
        SQLiteDatabase db = getWritableDatabase();
        Cursor c = db.rawQuery("select name from dishes", null);
        return c;
    }

```

```

    public String getbmitype(float bmi) {
        SQLiteDatabase db = this.getWritableDatabase();
        String type = "";
        float value;

```

```

        for (int id = 8; id > 0; id--) {
            Cursor c = db.rawQuery("select start from bmi where id=" + id, null);
            c.moveToFirst();
            value = Float.parseFloat(c.getString(0));
            if (bmi > value) {
                Cursor d = db.rawQuery("select type from bmi where id=" + id, null);
                d.moveToFirst();
                type = d.getString(0);
                break;
            }
        }

```

```

    }

    return type;

}

public void deletedata(int id) {
    createdb();
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete("family", "id = " + id, null);
}

public String[] display() {
    i = 0;
    createdb();
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor c = db.rawQuery("Select * from family", null);
    c.moveToFirst();
    while (!c.isAfterLast() && i < 20) {
        s[i] = c.getString(0) + " " + c.getString(1);
        i++;
        c.moveToNext();
    }
    c.close();
    while (i < 20) {
        s[i] = "";
        i++;
    }
    return s;
}

public Cursor getnutrientvalues(String ingredienttype, String ingredient) {
    SQLiteDatabase db = getWritableDatabase();
    Cursor c = db.rawQuery("Select calories, carbohydrates, fats, proteins, iron, calcium
from " + ingredienttype + " where name= '" + ingredient + "'", null);

    return c;
}

```

```

    public void insertuserrecall(String ingredient, String dish, Float calories, Float carbs,
Float fats, Float proteins, Float iron, Float calcium) {
        SQLiteDatabase db = getWritableDatabase();
        db.execSQL("INSERT OR REPLACE INTO USER_DAY (INGREDIENTS, DISH,
CALORIES, CARBOHYDRATES, FATS, PROTEINS, IRON, CALCIUM)
VALUES('"+ingredient+"','"+dish+"','"+calories+"','"+carbs+"','"+fats+"','"+proteins+"','"+
iron+"','"+calcium+"')");
    }

```

```

    public Cursor gettotalcalories() {
        SQLiteDatabase db = getWritableDatabase();
        float total = 0.0f;
        Cursor c = db.rawQuery("Select calories,carbohydrates,fats,proteins,iron,calcium from
user_day", null);
        return c;
    }

```

```

    public void insertdisease(int id, int present, String level) {
        SQLiteDatabase db = getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("ID", id);
        values.put("PRESENT", present);
        values.put("LEVEL", level);
        db.update("DISEASE", values, "id=" + id, null);
    }

```

```

    public void insertallergy(int id, int present) {
        SQLiteDatabase db = getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("ID", id);
        values.put("PRESENT", present);
        db.update("ALLERGY", values, "id=" + id, null);
    }

```

```

    public ArrayList<String> getdiseases() {
        ArrayList<String> list = new ArrayList<String>();
        SQLiteDatabase db = getWritableDatabase();
        Cursor c = db.rawQuery("Select name from disease where present=1", null);
        c.moveToFirst();
        while (!c.isAfterLast()) {

```

```

        list.add(c.getString(0));
        c.moveToNext();
    }
    return list;
}

```

```

public ArrayList<String> getallergies() {
    ArrayList<String> list = new ArrayList<String>();
    SQLiteDatabase db = getWritableDatabase();
    Cursor c = db.rawQuery("Select name from allergy where present=1", null);
    c.moveToFirst();
    while (!c.isAfterLast()) {
        list.add(c.getString(0));
        c.moveToNext();
    }
    return list;
}

```

```

public ArrayList<String> getrecommenddishes(String meal, String disease) {
    ArrayList<String> list = new ArrayList<>();
    SQLiteDatabase db = getWritableDatabase();
    //Cursor c = db.rawQuery("Select name from recommend_dishes where (meal='" +
meal + "') AND (disease='" + disease + "')", null);
    Cursor c = db.query("recommend_dishes",new String[]{"name"},"meal=? and
disease=?", new String[] {meal,disease},null,null,null);
    c.moveToFirst();
    while (!c.isAfterLast()) {
        list.add(c.getString(0));
        c.moveToNext();
    }
    return list;
}

```

```

public ArrayList<String> getrecommenddishes(String meal) {
    ArrayList<String> list = new ArrayList<>();
    SQLiteDatabase db = getWritableDatabase();
    Cursor c = db.rawQuery("Select distinct name from recommend_dishes where
meal=\'" + meal + "\'", null);
}

```

```

        c.moveToFirst();
        while (!c.isAfterLast()) {
            list.add(c.getString(0));
            c.moveToNext();
        }
        return list;
    }

    public void insertalarmdish(String dish, String meal) {
        SQLiteDatabase db = getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("name", dish);
        db.update("alarm_dishes", values, "meal=\"\" + meal + \"\"", null);
    }

    public ArrayList<AlarmDishes> getalarmdishes() {
        ArrayList<AlarmDishes> list = new ArrayList<>();
        SQLiteDatabase db = getWritableDatabase();
        Cursor c;
        c = db.rawQuery("Select name, meal from alarm_dishes", null);
        c.moveToFirst();
        while (!c.isAfterLast()) {
            if (!c.getString(0).equals("") && c.getString(0) != null) {
                AlarmDishes ad = new AlarmDishes(c.getString(0), c.getString(1));
                list.add(ad);
            }
            c.moveToNext();
        }
        return list;
    }

    public void deletealarm(String meal)
    {
        SQLiteDatabase db = getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("name", "");
        db.update("alarm_dishes", values, "meal=\"\" + meal + \"\"", null);
    }

```



```

public String getuserdaydishes()
{
    SQLiteDatabase db = getWritableDatabase();
    Cursor c;
    String str="";
    c = db.rawQuery("Select distinct dish from user_day",null);
    c.moveToFirst();
    while (!c.isAfterLast())
    {
        str = str + c.getString(0) + " ";
        c.moveToNext();
    }
    return str;
}
}

```

Fragment Navigation Drawer

XML code:

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/white">

```

```

    <RelativeLayout
        android:id="@+id/nav_header_container"
        android:layout_width="match_parent"
        android:layout_height="140dp"
        android:layout_alignParentTop="true"
        android:background="@color/colorPrimaryDark">

```

```

    <ImageView
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:src="@drawable/ic_profile"

```

```

        android:scaleType="fitCenter"
        android:layout_centerInParent="true" />

</RelativeLayout>

<android.support.v7.widget.RecyclerView
    android:id="@+id/drawerList"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/nav_header_container"
    android:layout_marginTop="15dp" />

</RelativeLayout>

```

JAVA code:

```

package com.example.sudhakaran.pratibhojana.activity;

import android.content.Context;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.view.GestureDetector;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;

import java.util.ArrayList;

```

```

import java.util.List;
import com.example.sudhakaran.pratibhojana.R;
import com.example.sudhakaran.pratibhojana.adapter.NavigationDrawerAdapter;
import com.example.sudhakaran.pratibhojana.model.NavDrawerItem;

public class FragmentDrawer extends Fragment {

    private static String TAG = FragmentDrawer.class.getSimpleName();

    private RecyclerView recyclerView;
    private ActionBarDrawerToggle mDrawerToggle;
    private DrawerLayout mDrawerLayout;
    private NavigationDrawerAdapter adapter;
    private View containerView;
    private static String[] titles = null;
    private FragmentDrawerListener drawerListener;

    public FragmentDrawer() {

    }

    public void setDrawerListener(FragmentDrawerListener listener) {
        this.drawerListener = listener;
    }

    public static List<NavDrawerItem> getData() {
        List<NavDrawerItem> data = new ArrayList<>();

        // preparing navigation drawer items
        for (int i = 0; i < titles.length; i++) {
            NavDrawerItem navItem = new NavDrawerItem();
            navItem.setTitle(titles[i]);
            data.add(navItem);
        }
        return data;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);

// drawer labels
titles = getActivity().getResources().getStringArray(R.array.nav_drawer_labels);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                          Bundle savedInstanceState) {
    // Inflating view layout
    View layout = inflater.inflate(R.layout.fragment_navigation_drawer, container, false);
    recyclerView = (RecyclerView) layout.findViewById(R.id.drawerList);

    adapter = new NavigationDrawerAdapter(getActivity(), getData());
    recyclerView.setAdapter(adapter);
    recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));

    recyclerView.addOnItemTouchListener(new RecyclerViewTouchListener(getActivity(),
recyclerView, new ClickListener() {
        @Override
        public void onClick(View view, int position) {
            drawerListener.onDrawerItemSelected(view, position);
            mDrawerLayout.closeDrawer(containerView);
        }

        @Override
        public void onLongClick(View view, int position) {

        }
    }));

    return layout;
}

public void setUp(int fragmentId, DrawerLayout drawerLayout, final Toolbar toolbar) {
    containerView = getActivity().findViewById(fragmentId);
    mDrawerLayout = drawerLayout;
}

```

```

        mDrawerToggle = new ActionBarDrawerToggle(getActivity(), drawerLayout, toolbar,
R.string.drawer_open, R.string.drawer_close) {
    @Override
    public void onDrawerOpened(View drawerView) {
        super.onDrawerOpened(drawerView);
        getActivity().invalidateOptionsMenu();
    }

    @Override
    public void onDrawerClosed(View drawerView) {
        super.onDrawerClosed(drawerView);
        getActivity().invalidateOptionsMenu();
    }

    @Override
    public void onDrawerSlide(View drawerView, float slideOffset) {
        super.onDrawerSlide(drawerView, slideOffset);
        toolbar.setAlpha(1 - slideOffset / 2);
    }
};

mDrawerLayout.setDrawerListener(mDrawerToggle);
mDrawerLayout.post(new Runnable() {
    @Override
    public void run() {
        mDrawerToggle.syncState();
    }
});

}

public static interface ClickListener {
    public void onClick(View view, int position);

    public void onLongClick(View view, int position);
}

static class RecyclerViewTouchListener implements RecyclerView.OnItemTouchListener {

    private GestureDetector gestureDetector;

```

```

private ClickListener clickListener;

    public RecyclerViewTouchListener(Context context, final RecyclerView recyclerView,
final ClickListener clickListener) {
        this.clickListener = clickListener;
        gestureDetector = new GestureDetector(context, new
GestureDetector.SimpleOnGestureListener() {
            @Override
            public boolean onSingleTapUp(MotionEvent e) {
                return true;
            }

            @Override
            public void onLongPress(MotionEvent e) {
                View child = recyclerView.findViewById(e.getX(), e.getY());
                if (child != null && clickListener != null) {
                    clickListener.onLongClick(child, recyclerView.getChildPosition(child));
                }
            }
        });
    }

    @Override
    public boolean onInterceptTouchEvent(RecyclerView rv, MotionEvent e) {

        View child = rv.findViewById(e.getX(), e.getY());
        if (child != null && clickListener != null && gestureDetector.onTouchEvent(e)) {
            clickListener.onClick(child, rv.getChildPosition(child));
        }
        return false;
    }

    @Override
    public void onTouchEvent(RecyclerView rv, MotionEvent e) {
    }

    @Override
    public void onRequestDisallowInterceptTouchEvent(boolean disallowIntercept) {

    }

```

```

    }

    public interface FragmentDrawerListener {
        public void onDrawerItemSelected(View view, int position);
    }
}

```

Fragment Listener

```

package com.example.sudhakaran.pratibhojana.activity;

import java.util.Calendar;

public interface fragmentlistener {
    public void setingredientlist(String type,String dish);
    public void setcategory(String dish);
    public void setsingleingredient(String type,String ingredient,String dish);
    public void setrecommenddishes(String meal, int RQS_code);
    public void setrecommend();
    public void setAlarm(Calendar targetCal,String speech);
    public void cancelAlarm(String meal, String speech);
    public void openTimePickerDialog(boolean is24r, String speech);

}

```

Personal Details Fragment

XML code:

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"

```

```
android:background="#ffffff"
```

```
>
```

```
<RelativeLayout
```

```
android:layout_height="wrap_content"
```

```
android:layout_width="wrap_content"
```

```
android:layout_centerHorizontal="true"
```

```
>
```

```
<TextView
```

```
android:id="@+id/tv1"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="FULL NAME:"
```

```
android:layout_marginTop="10dp"
```

```
android:textColor="#000000"
```

```
android:textSize="20dp"
```

```
android:textStyle="bold"
```

```
/>
```

```
<EditText
```

```
android:id="@+id/name"
```

```
android:layout_width="150dp"
```

```
android:layout_alignTop="@+id/tv1"
```

```
android:layout_toRightOf="@id/tv1"
```

```
android:layout_marginLeft="35dp"
```

```
android:layout_height="wrap_content"
```

```
/>
```

```
<RadioGroup
```

```
android:id="@+id/rg1"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_below="@id/name"
```

```
android:orientation="horizontal"
```

```
android:layout_centerHorizontal="true"
```

```
android:layout_marginTop="10dp"
```


>

```
<RadioButton
    android:id="@+id/veg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Vegetarian"
/>
```

```
<RadioButton
    android:id="@+id/nveg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Non-vegetarian"

/>
```

```
</RadioGroup>
```

```
<TextView
    android:id="@+id/tv5"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="LIFESTYLE:"
    android:textColor="#000000"
    android:textSize="20dp"
    android:layout_marginTop="10dp"
    android:textStyle="bold"
    android:layout_below="@id/rg1"
/>
```

```
<RadioGroup
    android:id="@+id/rg2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/rg1"
    android:layout_toRightOf="@id/tv5"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp"

>
```

```
<RadioButton
    android:id="@+id/sedentary"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sedentary"
/>
```

```
<RadioButton
    android:id="@+id/moderate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Moderately Active"
/>
```

```
<RadioButton
    android:id="@+id/active"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Active"
/>
```

```
</RadioGroup>
```

```
<TextView
    android:id="@+id/tv2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="AGE:"
    android:textColor="#000000"
    android:textSize="20dp"
    android:layout_marginTop="10dp"
    android:textStyle="bold"
    android:layout_below="@+id/rg2"
/>
```

```
<TextView
    android:id="@+id/info"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```

        android:text="(40 - 50years)"
        android:layout_below="@id/tv2"
        android:layout_alignLeft="@id/tv2"/>

<!--<RadioGroup
    android:id="@+id/rg2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_toRightOf="@id/tv2"
    android:layout_below="@+id/rg1"
    android:layout_alignLeft="@id/name"
    android:layout_marginTop="20dp"
    >
    <RadioButton
        android:id="@+id/rb1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="40-44 years"
        android:textSize="15dp"/>
    <RadioButton
        android:id="@+id/rb2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="15dp"
        android:text="45-50 years"/>

</RadioGroup> -->

<EditText
    android:id="@+id/age"
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@id/name"
    android:layout_marginTop="10dp"
    android:layout_toRightOf="@id/tv2"
    android:layout_below="@id/rg2"
    />

```

```
<TextView
    android:id="@+id/tv3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="HEIGHT(m):"
    android:textSize="20dp"
    android:textStyle="bold"
    android:textColor="#000000"
    android:layout_marginTop="10dp"
    android:layout_below="@id/age"
/>
```

```
<EditText
    android:id="@+id/height"
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/age"
    android:layout_marginTop="10dp"
    android:layout_alignLeft="@id/name"
/>
```

```
<TextView
    android:id="@+id/tv4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="WEIGHT(kg):"
    android:textColor="#000000"
    android:textSize="20dp"
    android:textStyle="bold"
    android:layout_marginTop="10dp"
    android:layout_below="@id/height"
/>
```

```
<EditText
    android:id="@+id/weight"
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@id/name"
    android:layout_marginTop="10dp"
    android:layout_below="@id/height"
```

```

/>

<TextView
    android:id="@+id/displaybmi"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/weight"
    android:textSize="15dp"
/>

<TextView
    android:id="@+id/type"
    android:layout_below="@id/displaybmi"
    android:layout_centerHorizontal="true"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

</RelativeLayout>

<Button
    android:id="@+id/save"
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:text="SAVE"
    android:background="@color/windowBackground"
    android:padding="0dp"
    android:minWidth="0dp"
    android:textSize="15dp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
/>

</RelativeLayout>

```

JAVA code:

```
package com.example.sudhakaran.pratibhojana.activity;
```

```

import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.text.Editable;
import android.text.InputType;
import android.text.TextWatcher;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.example.sudhakaran.pratibhojana.R;

public class PersonalDetailsFragment extends Fragment {

    Button save;
    TextView displaybmi, type;
    EditText weight, height, name, age1;
    RadioGroup rg1;
    float w,h,bmi;
    int age;
    int radio_id;
    String fname="",cuisine="";
    ArithmeticException a;
    DBHelper dbh ;
    Context context;

    public PersonalDetailsFragment(Context context) {
        // Required empty public constructor
        this.context = context;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {

```

```

        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_personal_details, container, false);

        ((MainActivity) getActivity()).setActionBarTitle(R.string.title_personaldetails);

        w=0.0f;
        h=0.0f;
        bmi=0.0f;
        displaybmi = (TextView) view.findViewById(R.id.displaybmi);
        displaybmi.setVisibility(View.INVISIBLE);
        rg1 = (RadioGroup) view.findViewById(R.id.rg1);
        dbh = new DBHelper(context);
        weight = (EditText) view.findViewById(R.id.weight);
        weight.setInputType(InputType.TYPE_CLASS_NUMBER |
        InputType.TYPE_NUMBER_FLAG_DECIMAL);
        height = (EditText) view.findViewById(R.id.height);
        height.setInputType(InputType.TYPE_CLASS_NUMBER |
        InputType.TYPE_NUMBER_FLAG_DECIMAL);
        save = (Button) view.findViewById(R.id.save);
        age1 = (EditText) view.findViewById(R.id.age);
        age1.setInputType(InputType.TYPE_CLASS_NUMBER);
        name = (EditText) view.findViewById(R.id.name);
        name.setInputType(InputType.TYPE_CLASS_TEXT);
        type = (TextView) view.findViewById(R.id.type);
        a = new ArithmeticException();

        weight.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {

            }

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {

```

```

type.setText("");

try
{
    w = Float.valueOf(weight.getText().toString());
    h = Float.valueOf(height.getText().toString());
    bmi = w/(h*h);
}
catch (Exception e)
{
    bmi= 0.0f;
}

if(bmi<100 && bmi>0) {
    displaybmi.setText("Your BMI is: " + bmi);
    displaybmi.setVisibility(View.VISIBLE);
}
else
    displaybmi.setVisibility(View.INVISIBLE);

/* if(w==0.0f || h==0.0f) {
    displaybmi.setVisibility(View.INVISIBLE);
}
else
    displaybmi.setVisibility(View.VISIBLE); */
}

@Override
public void afterTextChanged(Editable s) {

}
});

height.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {

    }
}

```



```

@Override
public void onTextChanged(CharSequence s, int start, int before, int count) {

    type.setText("");

    try {
        w = Float.valueOf(weight.getText().toString());
        h = Float.valueOf(height.getText().toString());
        bmi = w / (h * h);
    } catch (Exception e) {
        bmi = 0.0f;
    }

    if (bmi < 100 && bmi > 0) {
        displaybmi.setText("Your BMI is: " + bmi);
        displaybmi.setVisibility(View.VISIBLE);
    } else
        displaybmi.setVisibility(View.INVISIBLE);

    /*if(w==0.0f || h==0.0f) {
        displaybmi.setVisibility(View.INVISIBLE);
    }
    else
        displaybmi.setVisibility(View.VISIBLE); */
}

@Override
public void afterTextChanged(Editable s) {

    }
});

save.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        try {

            fname = name.getText().toString();
            age = Integer.parseInt(age1.getText().toString());

```

```

        radio_id = rg1.getCheckedRadioButtonId();

        if(radio_id!=-1) {
            if (bmi < 100 && bmi > 0) {
                if (age >= 40 && age <= 50) {
                    if(radio_id==R.id.veg)
                        cuisine="veg";
                    else
                        cuisine="nonveg";
                    dbh.insertuserdata(fname, age, cuisine, bmi);
                    type.setText(dbh.getbmitype(bmi));
                    Toast.makeText(context, "Saved Successfully!",
Toast.LENGTH_SHORT).show();
                } else
                    throw a;
            } else {
                type.setText("Invalid BMI");
                throw a;
            }
        } else throw a;

    } catch (Exception e) {
        Toast.makeText(context, "Invalid input, check again!",
Toast.LENGTH_SHORT).show();
    }
}

});

return view;
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);

}

@Override
public void onDetach() {

```

```

        super.onDetach();

    }

}

```

Recall / Diet history Fragment

XML code:

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
        android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:id="@+id/relative_layout"
    tools:context="com.example.sudhakaran.pratibhojana.recall">

    <EditText
        android:id="@+id/search"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Select your dishes"
    />

    <ListView
        android:id="@+id/lv"
        android:layout_below="@id/search"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </ListView>

    <Button
        android:id="@+id/submit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"

```

```

        android:text="SUBMIT"
        android:textColor="#000000"
        android:background="@color/windowBackground"
        android:textSize="20dp"
    />

</RelativeLayout>

```

JAVA code:

```

package com.example.sudhakaran.pratibhojana.activity;

import android.app.Activity;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.database.Cursor;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.speech.tts.TextToSpeech;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.AlphaAnimation;
import android.view.animation.Animation;
import android.view.animation.AnimationSet;
import android.view.animation.TranslateAnimation;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.RelativeLayout;
import android.widget.Toast;

import com.example.sudhakaran.pratibhojana.R;

```

```

import com.example.sudhakaran.pratibhojana.adapter.Custom_adapter;
import com.example.sudhakaran.pratibhojana.model.RecallDishes;

import java.util.ArrayList;
import java.util.Locale;

public class RecallFragment extends Fragment implements TextToSpeech.OnInitListener {

    Context context;
    Button submit;
    EditText search;
    ListView lv;
    RelativeLayout layout;

    TextToSpeech tts;
    Animation in;
    TranslateAnimation slide;
    AnimationSet set;
    BroadcastReceiver mBroadcastReceiver;
    fragmentlistener listener;
    DBHelper dbh;
    Cursor c;

    public RecallFragment(fragmentlistener listener, Context context) {
        // Required empty public constructor
        this.listener = listener;
        this.context = context;
        dbh = new DBHelper(context);
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        c = dbh.getrecalldishes();
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {

```

```

// Inflate the layout for this fragment
View view = inflater.inflate(R.layout.fragment_recall, container, false);

((MainActivity) getActivity()).setActionBarTitle(R.string.title_recall);

tts = new TextToSpeech(context, this);
    Toast.makeText(context, "NOW IT'S TIME TO KNOW YOUR DIET HISTORY",
Toast.LENGTH_LONG).show();

ArrayList<RecallDishes> arraylist = new ArrayList <RecallDishes>();

if(c.moveToFirst()) {
    do {
        RecallDishes rd = new RecallDishes(c.getString(0));
        arraylist.add(rd);
        c.moveToNext();
    } while(!c.isAfterLast());
}

submit = (Button) view.findViewById(R.id.submit);
lv = (ListView) view.findViewById(R.id.lv);
lv.setVisibility(View.INVISIBLE);
search = (EditText) view.findViewById(R.id.search);
search.setVisibility(View.INVISIBLE);

layout = (RelativeLayout) view.findViewById(R.id.relative_layout);

in = new AlphaAnimation(0.0f, 1.0f);
in.setDuration(1500);

        /*slide = new TranslateAnimation(Animation.RELATIVE_TO_PARENT,
0.0f, Animation.RELATIVE_TO_PARENT, 0.0f, Animation.RELATIVE_TO_PARENT, 1.0
f, Animation.RELATIVE_TO_PARENT, 0.0f);
        slide.setDuration(1100);
        slide.setFillAfter(true);

set = new AnimationSet(false);
set.addAnimation(in);
set.addAnimation(slide); */

```

```

        final Custom_adapter customAdapter = new
Custom_adapter(context,arraylist,listener);
        lv.setAdapter(customAdapter);

search.postDelayed(new Runnable() {
    public void run() {
        search.startAnimation(in);
        lv.startAnimation(in);
        search.setVisibility(View.VISIBLE);
        lv.setVisibility(View.VISIBLE);
    }
}, 1000);

// Capture Text in EditText
search.addTextChangedListener(new TextWatcher() {

    @Override
    public void afterTextChanged(Editable arg0) {
        // TODO Auto-generated method stub
        String text = search.getText().toString().toLowerCase(Locale.getDefault());
        customAdapter.filter(text);
    }

    @Override
    public void beforeTextChanged(CharSequence arg0, int arg1,
        int arg2, int arg3) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onTextChanged(CharSequence arg0, int arg1, int arg2,
        int arg3) {
        // TODO Auto-generated method stub
    }
});

submit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(context,"Saved!",Toast.LENGTH_SHORT).show();
    }
});

```

```

    }
    });
    return view;

}

@Override
public void onInit(int status) {
    if (status == TextToSpeech.SUCCESS) {

        int result = tts.setLanguage(Locale.US);

        if (result == TextToSpeech.LANG_MISSING_DATA
            || result == TextToSpeech.LANG_NOT_SUPPORTED) {
            Log.e("TTS", "This Language is not supported");
        } else {
            tts.speak("Could you please tell us what you had yesterday",
TextToSpeech.QUEUE_FLUSH, null);
        }

    } else {
        Log.e("TTS", "Initilization Failed!");
    }
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);

}

@Override
public void onDetach() {
    super.onDetach();
}

}

```


Recommendation Fragment

XML code:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:fab="http://schemas.android.com/apk/res-auto"
    android:id="@+id/fl"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
    <android.support.v7.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/rv"
    />
```

```
    <View
        android:id="@+id/shadowView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#75000000"
        android:visibility="gone" />
```

```
    <com.example.sudhakaran.library.FloatingActionsMenu
        android:id="@+id/multiple_actions"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        fab:fab_addButtonColorNormal="@color/colorPrimary"
        fab:fab_addButtonColorPressed="@color/colorPrimaryDark"
        fab:fab_addButtonPlusIconColor="@color/colorAccent"
        fab:fab_labelStyle="@style/menu_labels_style"
        android:layout_marginBottom="16dp"
        android:layout_marginRight="16dp"
```

```

android:layout_marginEnd="16dp">

<com.example.sudhakaran.library.FloatingActionButton
    android:id="@+id/breakfast"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    fab:fab_colorNormal="@color/white"
    fab:fab_title="Breakfast"
    fab:fab_icon="@drawable/dish"
    fab:fab_colorPressed="@color/white_pressed"/>

<com.example.sudhakaran.library.FloatingActionButton
    android:id="@+id/morningsnacks"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    fab:fab_colorNormal="@color/white"
    fab:fab_title="Morning snacks"
    fab:fab_icon="@drawable/dish"
    fab:fab_colorPressed="@color/white_pressed"/>

<com.example.sudhakaran.library.FloatingActionButton
    android:id="@+id/lunch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    fab:fab_colorNormal="@color/white"
    fab:fab_title="Lunch"
    fab:fab_icon="@drawable/dish"
    fab:fab_colorPressed="@color/white_pressed"/>

<com.example.sudhakaran.library.FloatingActionButton
    android:id="@+id/eveningsnacks"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    fab:fab_colorNormal="@color/white"
    fab:fab_title="Evening snacks"
    fab:fab_icon="@drawable/dish"
    fab:fab_colorPressed="@color/white_pressed"/>

<com.example.sudhakaran.library.FloatingActionButton
    android:id="@+id/dinner"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        fab:fab_colorNormal="@color/white"
        fab:fab_title="Dinner"
        fab:fab_icon="@drawable/dish"
        fab:fab_colorPressed="@color/white_pressed"/>

</com.example.sudhakaran.library.FloatingActionsMenu>

</RelativeLayout>

```

JAVA code:

```

package com.example.sudhakaran.pratibhojana.activity;

import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import com.example.sudhakaran.library.FloatingActionButton;
import com.example.sudhakaran.library.FloatingActionsMenu;
import com.example.sudhakaran.pratibhojana.R;
import com.example.sudhakaran.pratibhojana.adapter.RVcard_adapter;
import com.example.sudhakaran.pratibhojana.model.AlarmDishes;

import java.util.ArrayList;
import java.util.List;

public class RecommendationFragment extends Fragment {

    Context context;
    fragmentlistener listener;
    List<AlarmDishes> alarmDishesList = new ArrayList<>();

```

```

DBHelper dbh;

public RecommendationFragment(fragmentlistener listener, Context context) {
    this.context = context;
    this.listener = listener;
    dbh = new DBHelper(context);
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                          Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    final View view = inflater.inflate(R.layout.fragment_recommendation, container,
false);

    ((MainActivity) getActivity()).setActionBarTitle(R.string.title_recommendation);

    alarmDishesList.clear();
    alarmDishesList = dbh.getalarmdishes();

    RecyclerView rv = (RecyclerView) view.findViewById(R.id.rv);
    final View shadowview = view.findViewById(R.id.shadowView);
    LinearLayoutManager llm = new LinearLayoutManager(context);
    rv.setLayoutManager(llm);

    RVcard_adapter rvcard_adapter = new RVcard_adapter(context, alarmDishesList,
listener);
    rv.setAdapter(rvcard_adapter);

    final FloatingActionsMenu menuMultipleActions = (FloatingActionsMenu)
view.findViewById(R.id.multiple_actions);

    menuMultipleActions.setOnFloatingActionsMenuUpdateListener(new
FloatingActionsMenu.OnFloatingActionsMenuUpdateListener() {
        @Override

```

```

    public void onMenuExpanded() {
        shadowview.setVisibility(View.VISIBLE);
    }

    @Override
    public void onMenuCollapsed() {
        shadowview.setVisibility(View.GONE);
    }
});

```

```

        FloatingActionButton breakfast = (FloatingActionButton)
view.findViewById(R.id.breakfast);
        FloatingActionButton morningsnacks = (FloatingActionButton)
view.findViewById(R.id.morningsnacks);
        FloatingActionButton lunch = (FloatingActionButton)
view.findViewById(R.id.lunch);
        FloatingActionButton eveningsnacks = (FloatingActionButton)
view.findViewById(R.id.eveningsnacks);
        FloatingActionButton dinner = (FloatingActionButton)
view.findViewById(R.id.dinner);

```

```

breakfast.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        listener.setrecommenddishes("Breakfast",1);
        menuMultipleActions.collapse();
    }
});

```

```

morningsnacks.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        listener.setrecommenddishes("Morning Snacks",2);
        menuMultipleActions.collapse();
    }
});

```

```

lunch.setOnClickListener(new View.OnClickListener() {
    @Override

```

```

        public void onClick(View v) {

            listener.setrecommenddishes("Lunch",3);
            menuMultipleActions.collapse();
        }
    });

    eveningsnacks.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            listener.setrecommenddishes("Evening Snacks",4);
            menuMultipleActions.collapse();
        }
    });

    dinner.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            listener.setrecommenddishes("Dinner",5);
            menuMultipleActions.collapse();
        }
    });

    return view;
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
}

@Override
public void onDetach() {
    super.onDetach();
}
}

```

ACKNOWLEDGEMENT

Success of project like this required a large amount of patience along with team work. This would not have been possible without the support shown by our team.

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

As a team we have put in efforts to complete the first half of this project. However, it would not have been possible without the kind support and help of many individuals and our college. We would like to extend our sincere thanks to all of them.

We would like to show our greatest appreciation to Prof. Dr. Amiya Kumar Tripathy and Prof. Ms. Ditty Varghese for their guidance and constant supervision in providing necessary information regarding the project & also for their support in carrying out this project work.

Special thanks to nutritionist Dr. Prajakta Sarmalkar for her tremendous support and help.

We would also like to thank our project co-ordinator Mrs. Uma Sahu for providing us with regular inputs about documentation and project timeline. Thank you all for helping us achieve this.

1. A.J.Sudhakaran, 01

(Name of student, Roll no.)

(Signature)

2. Stacy Bardeskar, 04

(Name of student, Roll no.)

(Signature)

3. Mohit Deorukhkar, 15

(Name of student, Roll no.)

(Signature)

Date: