FOOD DIET PLAN

(React Native)

By

MUHAMMAD ZIA UL REHMAN

(2016-ARID-2256)

Bachelor of Science in Computer Science

(BSCS)

Barani Institute of Information Technology PMAS
Arid Agriculture University Rawalpindi

November 2020

FOOD DIET PLAN
(React Native)

A report submitted in partial fulfillment of the
requirements for the degree of
Bachelors of Science in Computer Science

Submitted By
MUHAMMAD ZIA UL REHMAN
(2016-ARID-2256)

Supervised by

MS. ZER AFSHAN GOHAR

Barani Institute of Information Technology PMAS
Arid Agriculture University Rawalpindi

November 2020

CERTIFICATE

It is certified that the contents and form of thesis entitled **Food Diet Plan (React Native)** submitted by **Muhammad Zia Ul Rehman** have been found satisfactory for the requirement of the degree

SUPERVISORY COMMITTEE

PROJECT SUPERVISOR:

Ms. Zer afshan Gohar

REPORT COORDINATOR:

Mr.Muhammd Khalid

WRITEUP COMMITTEE HEAD:

Ms. Noor ul Ain

ACKNOWLEDGEMENT

First, I owe gratitude to ALLAH Almighty, the most merciful and compassionate most gracious and beneficial who enabled me to accomplish the task. I feel honors to express heartiest feelings to director BIIT, **Dr. Muhammad Jamil Sawar** for his guidance and kind supervision. I am very thankful to my all teachers especially **Ma'am Zer afshan Gohar**, all credit goes to them for my current and future work in this field.

Furthermore, I am extremely grateful to all my official and unofficial teachers from class one to this Masters of in Computer Science everyone is exceptional, excellent and superior. Words cannot say the gratitude that for my parents whose affection and prayers has always been the key to my success. Sincere thanks to all family members and loved ones for every possible support. I am thankful to all of my old and new friends for their help and moral support, during my life, their thinking gives me the strength and power. They are so many that it will take another write-up for only mentioning their names.

Muhammad Zia Ul Rehman

DEDICATION

In the name of Almighty ALLAH, the most beneficent and merciful who gave me strength and knowledge to complete this project. His guidance and blessing have always been a source of encouragement for me. Secondly, I would like to express my special thanks of gratitude to my teachers who gave me the golden opportunity to do this wonderful project, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

Finally, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

ABSTRACT

The title of my project is FOOD DIET PLAN. It is an React Native application. The basic purpose of developing this application is to maintain diet plan of user according to their weight. This app has feature of adding new food. App deals on the basis of ingredients. User can activate and deactivate his plan. App also provides the feature of editing and deleting the plan. So basically, this app will help user to maintain their diet plan according to their weight.

Most people are concern about what they eat everyday. People tend to search information about the things that they eat or what related to their diet. Nowadays, there are systems which help people to manage their diet. However, the systems are not specifically focusing on the user's own needs. Different people will require different dietary needs or nutrition for their body. This is due to variety of weight, daily activity level and so on. So they need something that will help them with their own personal body needs. The system that has been developed is an Online Diet Planning System. This web-based application system will help user to get information about their dietary intake. There will be dietary recommendations for the user based on what they need. It is more focusing on the user with his or her own needs rather than just giving the general recommendations to all users.

Table of Contents

Contents **Page no**

CERTIFICATE ..... **i**
ACKNOWLEDGEMENT ..... **ii**
DEDICATION ..... **iii**
ABSTRACT ..... **iv**
LIST OF FIGURES ..... **vii**
LIST OF TABLE ..... **viii**
LIST OF ANNEXURES ..... **ix**
CHAPTER 1 ..... **1**
1.1 Introduction 1
1.2 Problem Statement 1
1.3 Proposed Solution 2
1.4 Project Scope 2
1.5 Project Features 2
1.6 Objectives 2
CHAPTER 2 ..... **4**
PROJECT BACKGROUND **4**
2.1 Introduction 4
2.2 Related Project and Research Article 4
2.3 Related Project Detail 5
2.4 Related Software in Market 8
CHAPTER 3 ..... **9**
CONCEPTUAL DESIGN **9**
3.1 Introduction 9
3.2 Requirement Elicitation 9
3.3 Requirement Specification 9
3.3.1 Functional Requirements 9
3.3.2 Non-Functional Requirements 9

3.3.3 Domain Requirements 10
3.4 Requirement Modeling 10
3.5 Data Flow Diagram 10
3.5.1 Data Flow Diagram Level 0 10
3.5.2 Data Flow Diagram Level 1 10
3.6 Database Design 11
3.6.1 Entity relationship Diagram 13
3.7 Logical Design 13
3.7.1 Conceptual Diagram 14
CHAPTER 4 15
IMPLEMENTATION 15
4.1 Introduction 15
4.2 Tools and Technologies 15
4.3 Pseudo Code 15
4.4 Graphical User interface 21
CHAPTER 5 25
CONCLUSION 25
5.1 Introduction 25
5.2 Concluding Remarks 25
5.2 Future Directions 25
5.3 Limitations 25
REFERENCES 26
ANNEXURE 27

LIST OF FIGURES

Figure No **Page no**

Figure 2.1: Diet Plan for 30 Days 6
Figure 2.2: One day normal diet 7
Figure 2.3: Vegetarian diet 7
Figure 2.4: Water for single day 8
Figure 3.1: Data Flow Diagram – Level 0 and 1 11
Figure 3.2: Entity Relationship Diagram 13
Figure 3.3: Representation of Conceptual Diagram 13
Figure 4.1: Home Screen Code 16
Figure 4.2: Home Screen 16
Figure 4.3: All Plans Screen Code 17
Figure 4.4: All Plans Screen 17
Figure 4.5: Today meal Code 18
Figure 4.6 (a, b): Empty and Loaded Today meals Screens 18
Figure 4.7: Active and Inactive Plans Screen Code 19
Figure 4.8: Active and Inactive Plans Screen 19
Figure 4.9: Create Plan Screen Code 20
Figure 4.10: Create New Plan Screen 20
Figure 4.11 (a,b,c): adding food to plan screen 23
Figure 4.12: Add Food Item Screens 24

LIST OF TABLE

Table no **Page No**

Table 3.1: Create plan 11

Table 3.2: Meal Item 12

Table 3.3: Food Item Calories 12

Table 3.4: Food Item Ingrediants 12

LIST OF ANNEXURES

Annexure **Page No**

A. Home Screen 27
B. User Dash Board..... 27
C. Ingredient Screen..... 27
D. Food Item Screen..... 28

CHAPTER 1

1.1 Introduction

Food Diet Plan is an ANDROID and iOS-based app. It is designed for users to maintain their diet plan according to their weight. User can make diet plan for multiple days.the app alows the user to set food for multiple times of day like breakfast, lunch, supper and dinner, there are multiple categories to select from like fruits and vegetables, the app uses a calories check to make sure the user does not eat over calories limit, the calories can be set by the user like if the user wants to reduce weight he can set low calories limit in his plan or if the user wants to increase weight he can increase calories limit, the app has a very friendly user interface that any new user can easily understand and operate.

Food Diet Plan includes:

- Saving Plan
- Adding new Food
- Activation / Deactivation / Deletion of plans
- Edit plan
- Copy plan

1.2 Problem Statement

The main problem is that people facing weight gaining issues and want to maintain the weight and diet.

People have no ideas to manage the diet according to the meal times and people may not know how much calories to take on daily basis whether they are losing or gaining weight .

There is also no guidance and proper instructions to follow the chart.

The app can be used very easily, it has self explanatory buttons and labels that tell the users what to do so the user's task becomes easier.

1.3 Proposed Solution

- Food diet plan is very convenient to implement easy to understand and also user friendly.
- The need of designing such software is to provide users a better way of scheduling the daily food calories.
- It reduces the human effort to memorize his /her daily diet plan .

1.4 Project Scope

Food Based Diet Plan app has broader scope because every user who want to maintain their diet plan will be able to use it.there are too many people trying to gain our lose weight, through this app they can easily set the calories limit according to their needs, the app is very easy to operate and user friendly.

1.5 Project Features

- Creating of Diet Plan.
- Adding new Food.
- Saved Plan Viewing.
- Delete Food Item and Plan.
- Calories Counter.
- Active and Inactive Plan

1.6 Objectives

- User can create his diet plan.
- App provides facility of Activation / Deactivation and Deletion of saved plans.
- User can add new Food to app.
- User can set calories limit.

- The food is Categorized making it easier to find.
- App enables user to handle Diet Plan on the basis of calories.
- It reduces the human effort to memorize his /her daily diet plan.

In precise, we discuss about the introduction of over project. This application is designed by covering all the basic aspects of food diet plan, and our app contain multiple items with calories that helps to user to understand easily.it also contain the feature of ingredients of every food item.

.

CHAPTER 2

PROJECT BACKGROUND

2.1 Introduction

This chapter includes the discussion about previous project by our campus students and related project in the market.Basically this project is already exist in market and our departments but we add some extra features like calories of every item with ingredients.

2.2 Related Project and Research Article

Application those are already exits having same functionality but in this application some extra features has been added like Calories Based Some application those work same like this application don't have this feature. The most common application is Calorie Counter.

Various studies have shown that many of us eat much more than we think we do we often underestimate the number of calories and food that we put into our bodies. It is also interesting to see how much we eat from each food group and how certain not-so-good things like sweets can add up.

‘Calorie Counter' as an app that keeps track of number of calories user consumes each day. It also keeps track of food, exercise and weight. This app is customized for each country's food and brands. It's a food diary to plan and keep track of what you're eating an exercise diary to record all the calories you burn and a journal to record all your progress.

It offers an easy to use food diary for user to track as well as plan in advance what you're going to eat. Image recognition of food and meals make it even easier to add what you're eating, with a community that's keen to advise on how best to proceed. A weight Tracking tool, along with barcode scanning, rounds off the package. Each time you access the app, it will tell you your progress i.e. how many percent you have hit of your daily target or if you have exceeded it. Even though it might not be 100% accurate, Calorie Counter gives you a rough overview and makes

you more aware of what you are doing to your body. It offers an easy to use food diary for user to track as well as plan in advance what you're going to eat. This app is customized for each country's food.

2.3 Related Project Detail

2.3.1 Problem Statement

People nowadays are very concern about their health. They care about what they eat. However, most of them lack knowledge about how to choose and what kind of food servings that is suitable for their body. This is because, different people will have different needs of dietary .due to variety of age, weight, height, gender, and lifestyle. Male needs different types of food compared to female, a sedentary activity level person need different types of food compared to a heavy activity level person, and this go so on and so far. In this modern day, people live in a hectic world and always on the go. So, some of them just do not have the time to go very details about what they should eat. They just eat without knowing what is actually suitable for their body. In this case, they need something that can help them quickly without having to cost them much in their eating habit. They need something that can help them quickly like in just a few clicks and we make food diet plan app for those people.

2.3.2 Scope of releated project

System will be used by people who want to know what type of food serving that are suitable for them and want to manage their diet better. It can also be used by people who works relate to the diet field. This will include the public, the expert dietitian/nutritionist, medical/health staff, individuals who is involved in campaign or talk to promote health and diet and so on.

This system will use web and android based application.

2.3.3 Tools of releated project

- Visual Studio code
- SQL
- Web Services
- Android Studio

2.3.4 Features of releated project

There four main groups of foods - carbohydrate, fruits and vegetables, protein, and milk and dairy products. For each groups, there are many different types of foods. For example, in carbohydrates group there are variety of foods such as rice, noodles, bread, and cereals. It is recommended all over the world take all the variety of food as different types of food offer different combinations of energy (calories) and nutrients (carbohydrates, proteins, fats, vitaminsand fiber).

2.3.5 Related Software Screens

2.3.5.1 30 Days diet plan

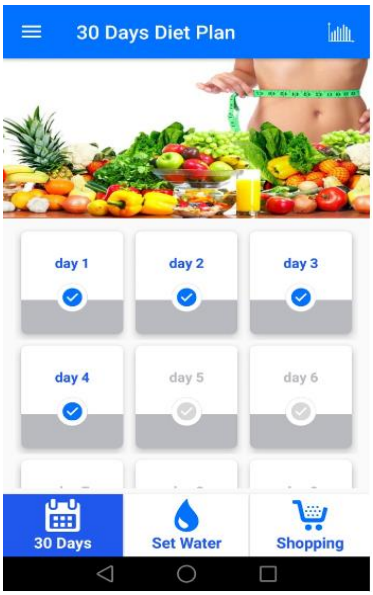


Figure 2.1: Diet Plan for 30 Days

The Figure 2.1 contains 30 days diet plan. User tap on days card and view breakfast, dinner and lunch of that screen and total calories of one day.

2.3.5.2 One day normal diet

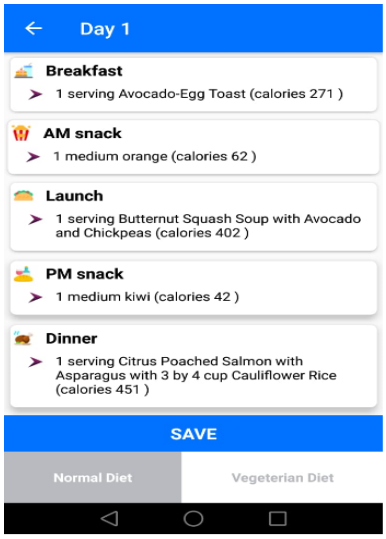


Figure 2.2: One day normal diet

The Figure 2.2 contains Breakfast , Lunch , PM Snack and Dinner with total calories and we show normal diet in this screen.

2.3.5.3 Vegetarian diet

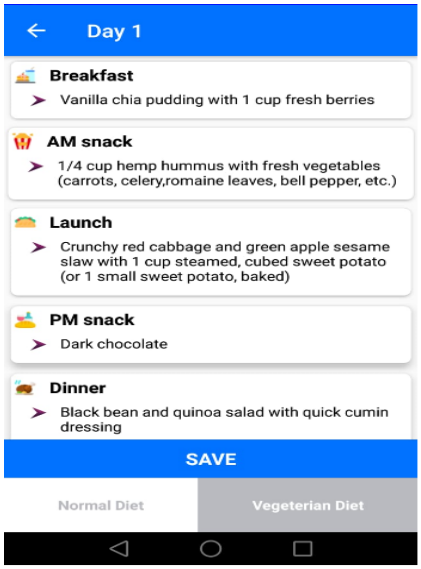


Figure 2.3: Vegetarian diet

The Figure 2.3 contains Breakfast , Lunch , PM Snack and Dinner with total calories and this screens only show vegetarian diet.

2.3.5.4 Water for single day

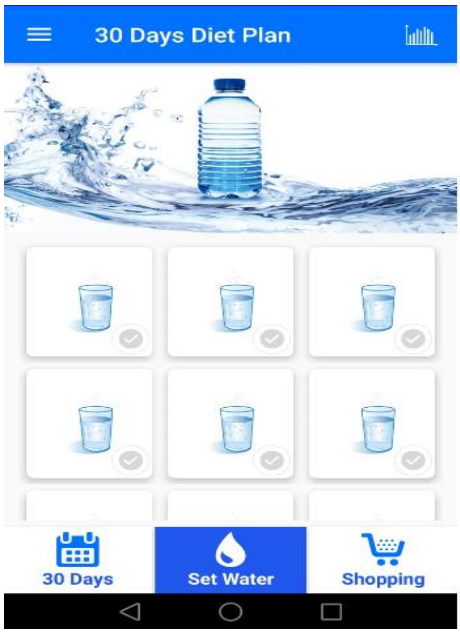


Figure 2.4: Water for single day

2.4 Related Software in Market

- Lose it
- Fat Secret

Application those are already exits having some functionality but in this application some extra features has been added like "Food Diet Plan With Calories and Ingredients". Some application those work same like this application don't have this feature. This application also has additional feature of setting levels of ingredients.

In precise, we discuss the interface of previous project. The purpose of this interface is only for learning point of view. User can understand without deep knowledge .In this chapter we can show all the design of application step by step, these apps does not contain the calories calculate feature.

CHAPTER 3

CONCEPTUAL DESIGN

3.1 Introduction

This chapter includes the discussion about the sources of requirements. The main functional and non-functional requirements including how the requirements model in the form of DFD can be validated. Conceptual diagram is design to understand the application requirement.

3.2 Requirement Elicitation

Food Diet Planes React Native based application that create Diet Plan for user who want to gain or loss or maintain their weight. This project provides an easy way to make diet plans for users who want to gain or loss or maintain their weight.

3.3 Requirement Specification

In Food Diet Plan user can insert new food in the database. User creates diet plan that also saves in database then he can view/activate his any plan.

3.3.1 Functional Requirements

It is critical to identify your platform and functional requirements. Functional requirement will specify a behavior or function, for example: User can insert Food with calories and saved that in database. The user can view his Diet Plan.

3.3.2 Non-Functional Requirements

Non-functional requirement specifies how the system behaves in terms of constraints or prerequisites. Food Diet Plan provides ability to add new Food if not available in app.

3.3.3 Domain Requirements

It is the requirement that comes from the application domain of the system that reflects the characteristics of that domain. The domain requirement of this system should concern about the requirements that reflect characteristic application. Like Android and iOS device etc.

3.4 Requirement Modeling

Multiple users can make plans to this application User selects calories and selects recommended food for that plan will be saved in database. Any user can view the saved plans.

3.5 Data Flow Diagram

3.5.1 Data Flow Diagram Level 0

Data flow diagram represents the flow of data between system and entities. As described below how user use this application. Context level represents the entire software system as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

3.5.2 Data Flow Diagram Level 1

In level 1 of Data Flow Diagram, user request to create plan and store this plan in database and return data from database to user. In next section of database user will view plan and and create day by day meal and show data from database.

Level 0 and 1:

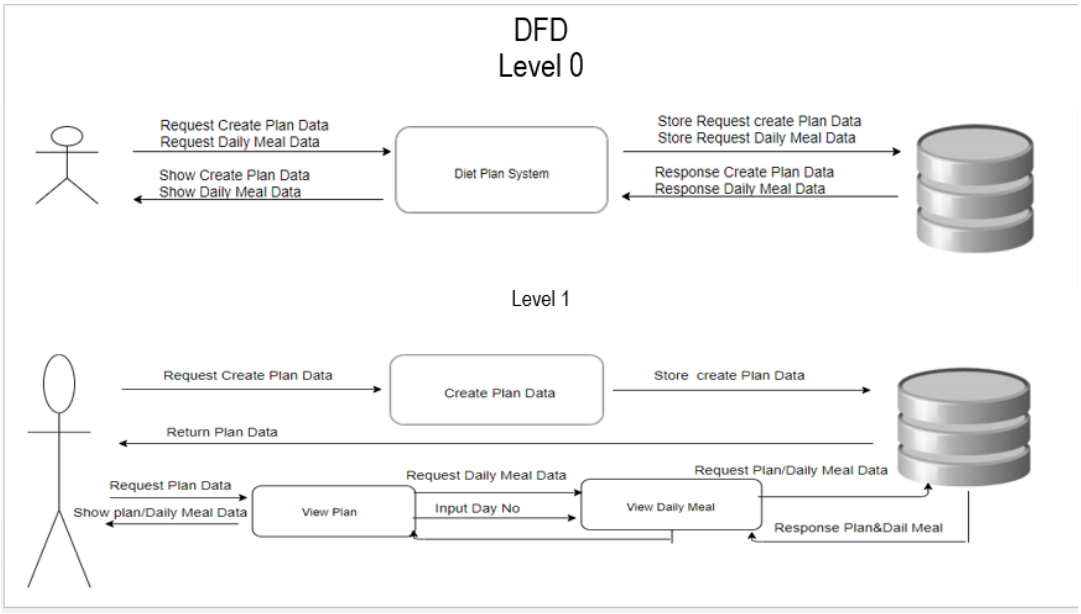


Figure 3.1: Data Flow Diagram – Level 0 and 1

Approved By: Ma'am Zer Afshan Gohar

3.6 Database Design

In database design database connections are discussed.Before starting this project we create ERD and DFD of this application . After that create database and database table using SQLITE tool. All database table given below.

PlanId	Title	Calories	ActiveStatus	TotalDays	datetime
Filter	Filter	Filter	Filter	Filter	Filter
4	Weight gain diet	2000	0	3	Mon Sep 18 2...
18	Vegetarian diet	2000	0	4	Mon Sep 20 2...
19	Weight loss diet	1600	1	5	Mon Sep 20 2...

Table 3.1: Create plan

DetailId	DayNo	MealTime	Item_Name	Quantity	PlanId	DetailPicture	DetailCalories
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
16	1	Breakfast	Apple,Peach,...	1,1,1,1,1	19	/9j/4AAQSkZJ...	52,39,60 ,89,67
17	2	Breakfast	Apple,Peach,...	1,1,1,1,1	19	/9j/4AAQSkZJ...	52,39,60 ,89,67
18	1	Lunch	Broccoli,Cucu...	1,1,1,1,1,1	19	/9j/4AAQSkZJ...	34,16,26,18,2...
19	2	Lunch	Broccoli,Cucu...	1,1,1,1,1,1	19	/9j/4AAQSkZJ...	34,16,26,18,2...
20	1	Dinner	Baked milk,Co...	1,1,1,1,1,1	19	/9j/4AAQSkZJ...	51,98,342,12...
21	2	Dinner	Baked milk,Co...	1,1,1,1,1,1	19	/9j/4AAQSkZJ...	51,98,342,12...
22	3	Breakfast	Chocolate mil...	1,1,1,1	19	/9j/4AAQSkZJ...	546,42,40,8
23	4	Breakfast	Chocolate mil...	1,1,1,1	19	/9j/4AAQSkZJ...	546,42,40,8
24	3	Lunch	Cucumber,Gr...	1,1,1,1,1	19	/9j/4AAQSkZJ...	16,20 ,77,25,25
25	4	Lunch	Cucumber,Gr...	1,1,1,1,1	19	/9j/4AAQSkZJ...	16,20 ,77,25,25
26	3	Dinner	Apple,Peach,...	1,1,1,1,1	19	/9j/4AAQSkZJ...	52,39,60 ,89,67
27	4	Dinner	Apple,Peach,...	1,1,1,1,1	19	/9j/4AAQSkZJ...	52,39,60 ,89,67
28	5	Breakfast	Cream Chees...	1,1,1,1	19	/9j/4AAQSkZJ...	342,122,95,2
29	5	Lunch	Areca Nut,Dri...	1,1,1	19	/9j/4AAQSkZJ...	342 ,241 ,249
30	5	Dinner	Tomato,Brocc...	1,1,1,1,1,1	19	/9j/4AAQSkZJ...	18,34,16,26,2...

Table 3.2: Meal Item

Id	ItemName	Picture	ItemCategory	ItemCalories	Sugar	Protien	Sodium	Fat	Carbs
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	Apple	/9j/4AAQSkZJ...	Fresh Fruit	52	10	0.3	0.001	0.2	14
2	Peach	/9j/4AAQSkZJ...	Fresh Fruit	39	8	0.9	0	0.3	10
3	Mango	/9j/4AAQSkZJ...	Fresh Fruit	60	14	0.8	0.001	0.4	15
4	Banana	/9j/4AAQSkZJ...	Fresh Fruit	89	12	1.1	0.001	0.3	23
5	Grapes	/9j/4AAQSkZJ...	Fresh Fruit	67	16	0.6	0.002	0.4	17
6	Tomato	/9j/4AAQSkZJ...	Vegetable	18	2.6	0.9	0.005	0.6	3
7	Broccoli	/9j/4AAQSkZJ...	Vegetable	34	1.7	1.9	0.033	0.4	5
8	Cucumber	/9j/4AAQSkZJ...	Vegetable	16	1.7	0.8	0.002	0	3.1
9	Pumpkin	/9j/4AAQSkZJ...	Vegetable	26	2.8	1	0.1	0.1	7
10	Tomato	/9j/4AAQSkZJ...	Vegetable	18	2.6	0.9	0.005	5	3.9
11	Carrot	/9j/4AAQSkZJ...	Vegetable	28	6.6	0.6	0.042	0.3	5.7
12	Cauliflower	/9j/4AAQSkZJ...	Vegetable	25	1.91	1.9	0.0003	0.3	5
13	Cabbage	/9j/4AAQSkZJ...	Vegetable	25	3.2	1.3	0.018	0.1	6
14	Eggplant	/9j/4AAQSkZJ...	Vegetable	25	3.5	1	0.02	0.2	6

Table 3.3: Food Item Calories

IngId	FoodItemRef	Unit	Protein	Carbs
Filter	Filter	Filter	Filter	Filter
1	Egg	g	8.3	1.1
2	Apple	g	0.3	13.8
3	Banana	g	1.1	23
4	Apple	g	8.3	1.1
5	Apple	g	8.3	1.1
6	tomato	g	0.9	3.9

Table 3.4: Food Item Ingrediants

3.6.1 Entity relationship Diagram

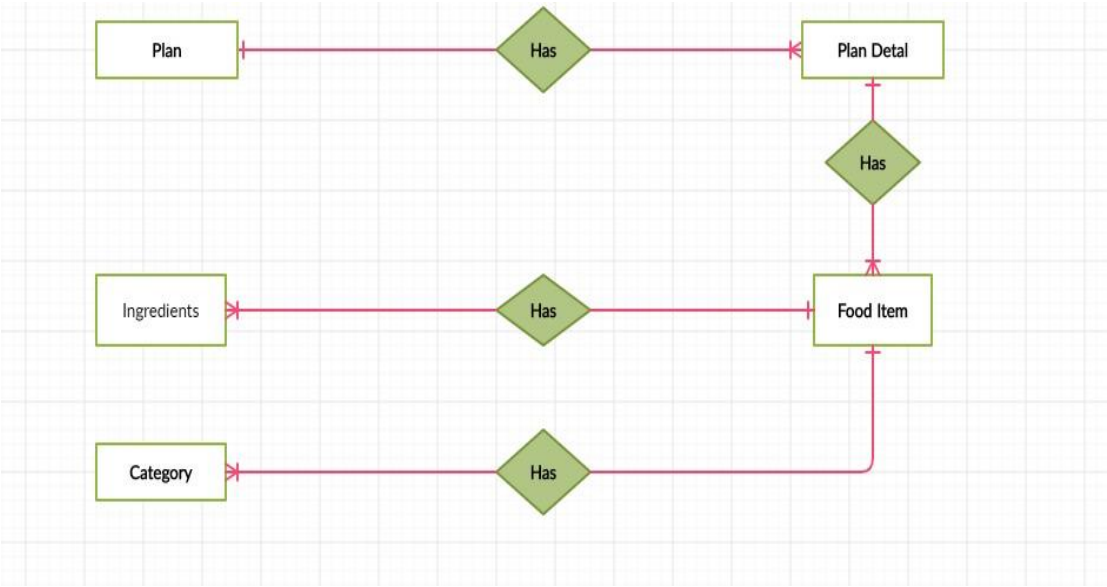


Figure 3.2: Entity Relationship Diagram

Approved By: Ma'am Zer Afshan Gohar

This diagram shows all the table relation between them, and shows a complete mechanism between each other. Every table has some relation which can be done one by one, many to many or many to one.

3.7 Logical Design

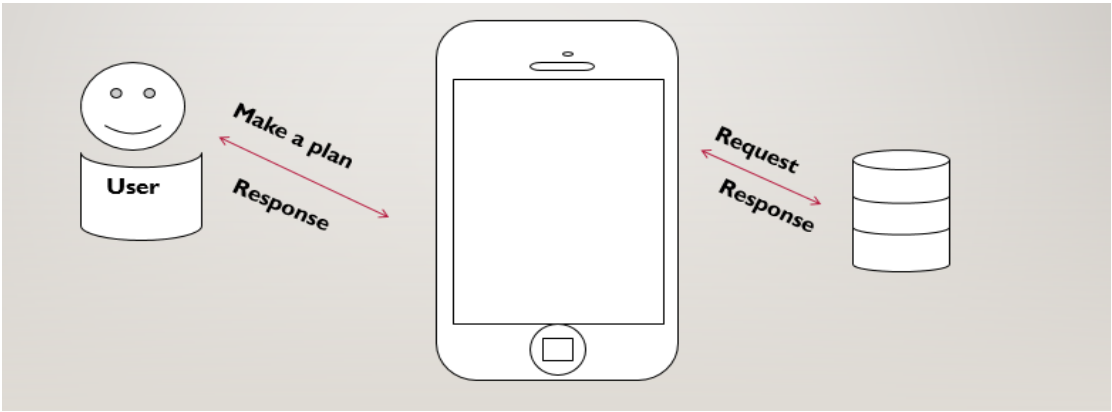


Figure 3.3: Representation of Conceptual Diagram

Approved By: Ma'am Zer Afshan Gohar

3.7.1 Conceptual Diagram

A conceptual diagram is essentially an illustration depicting the arrangement and relationships of key attributes within a system by using a variety of appropriate symbols that are easily understood. Put simply, conceptual diagrams are "thought drawings. An unfamiliar user can easily understand the working of system with the help of conceptual diagram. Conceptual diagram shows complete working of system. In this system user can use application on Android and iOS platform and access database. The diagram shows all the table relation between them, and shows a complete mechanism between each other.

In precise, we discuss about the sources of requirements. The main functional and non-functional requirements including how the requirements model in the form of DFD can be validated. Conceptual diagram is design to understand the application requirement.

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

This chapter includes the discussion about the Tools and Technologies which have been used for designing of this application. It also includes how the application has been designed through coding. Also, this chapter includes the discussion about graphical user interface and this can be shown through project screen shots.

4.2 Tools and Technologies

- Visual Studio code
- SQLite
- Android Studio

4.3 Pseudo Code

Home Screen

```
If (usertap=viewplan)
   {        Display [view plan]}
   Elseif  (usertap=viewallplan)
            {
                    Display[all plan]
             }
   Elseif(usertab=createplan)
            {
                    Display[create plan screen]
            }
   Else{Display("Do nothing")}
```

```
return (
  <View style={styles.container}>
    <View>
      <StatusBar
        barStyle="light-content"
        // dark-content, light-content and default
        hidden={false}

        backgroundColor="#595a4e"
        translucent={false}
        networkActivityIndicatorVisible={true}
      />
    </View>

    <View style={styles.cardcoverview} >
      <Card style={styles.cardcovertitle}>

        <Card.Cover source={require('./components/Icon/harvest.png')} style={styles.cardimagetitle} />

      </Card>
    </View>
    <View style={{flexDirection:"row" ,margin:20,marginTop:90,justifyContent:"space-between",backgroun
    <TouchableHighlight>
    <View style={styles.cardcoverViewBtn}>

      <Card onPress={() => this.props.navigation.navigate('CreateNewPlan')} style={styles.cardcover}>
        <Card.Cover source={require('./components/Icon/nutrition.png')} style={styles.cardimagefirst}
        <Text style={styles.cardText}>Create/Active Plan</Text>
      </Card>

    </View>
    </TouchableHighlight>
```

Figure 4.1: Home Screen Code

Output:

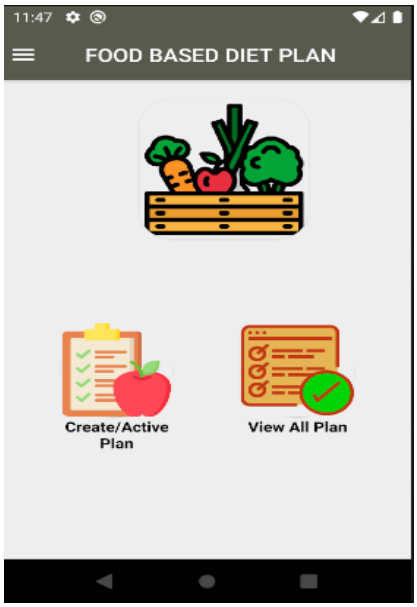


Figure 4.2: Home Screen

This is “Home Screen” In this screen user can do following tasks, create new plan and view all plans and user also have authority to activate or deactivate any plan.

All Plans Screen

If (Plans = exist in database)

{Display Plans}

Else {Do nothing}

```
  return (
    <View style={{ backgroundColor: '#c1ccc7',flex:1 }}>
      <FlatList
        data={this.state.FlatListItems}
        ItemSeparatorComponent={this.ListViewItemSeparator}
        keyExtractor={(item, index) => index.toString()}
        renderItem={({ item }) => (


          <View key={item.PlanId} style={{ backgroundColor: '#4b636e', padding: 5, borderRadius: 8,
            <Text style={styles.title}>Name: {item.Title}</Text>
            <Text style={styles.title}>Calories: {item.Calories}</Text>
            <Text style={styles.title}>Days: {item.TotalDays}</Text>
            <Text style={styles.title}>Status: {item.ActiveStatus?'Active':'Inactive'}</Text>

          </View>

        )}
      />
    </View>
  );
  }
}
```

Figure 4.3: All Plans Screen Code

Output:

Figure 4.4: All Plans Screen

This is “All plan screen” In this screen user can just view all plans .

User Dash Board

If (Selected Plan = exists in database)

{Display Selected Plan Foods}

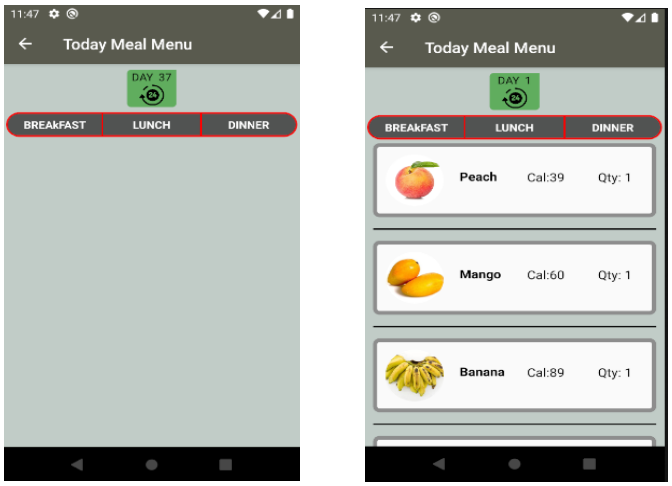
Else {Do nothing}

```
return (
  <ScrollView style={{ backgroundColor: '#c1ccc7' }}>
    <View style={styles.today}>
      <Text style={styles.daytoday}>DAY  {this.state.Day}</Text>
      <Image source={require('./components/Icon/today.png')} style={{ width: 30, height: 30, marginLeft: 15, backgroundColor: '#60ad5e' }} />
    </View>
    <View style={styles.MainContainer}>
      <FlatList
        data={this.state.FlatListItems}
        keyExtractor={(item, index) => index.toString()}
        renderItem={
          ({ item }) => (
            <View>
              <View style={styles.buttonstyle}>

                <View style={styles.BreakFast}>
                  <TouchableOpacity onPress={this.BreakFast.bind(this)} style={styles.appButtonContainerbreakfast}>
                    <Text style={styles.BreakFasttext}>BREAkFAST</Text>
                  </TouchableOpacity>
                </View>
                <View style={styles.Lunch}>
                  <TouchableOpacity onPress={this.Lunch.bind(this)} style={styles.appButtonContainerlunch}>
                    <Text style={styles.BreakFasttext}>LUNCH</Text>
                  </TouchableOpacity>
                </View>
                <View style={styles.Dinner}>
                  <TouchableOpacity onPress={this.Dinner.bind(this)} style={styles.appButtonContainerdinner}>
                    <Text style={styles.BreakFasttext}>DINNER</Text>
                  </TouchableOpacity>
                </View>
              </View>
```

Figure 4.5: Today meal Code

Output:



(a) (b)

Figure 4.6 (a, b): Empty and Loaded Today meals Screens

This is “Today meals Screens” In this screen user can follow one day meal like breakfast,lunch,dinner with item caloreis and quantity .

Active and Inactive Plans

If (Plan = active in database)

{Display Active plan}

Else {Do nothing}

```
db.transaction(tx => {
  tx.executeSql('SELECT * FROM Plan', [], (tx, results) => {
    var temp = [];
    for (let i = 0; i < results.rows.length; ++i) {
      temp.push(results.rows.item(i));
    }
    this.setState({
      ViewAllPlan: temp,
      // datetime: temp[4].datetime,
    });
  });
});
///////////==========================================================Date Time==================================
db.transaction(tx => {
  tx.executeSql('SELECT * FROM Plan WHERE ActiveStatus = 1', [], (tx, results) => {
    var temp = [];
    for (let i = 0; i < results.rows.length; ++i) {
      temp.push(results.rows.item(i));
    }
    this.setState({
      datetime: temp[0].datetime,
      expirydate: temp[0].TotalDays,
      expiryPlan: temp[0].PlanId,
      AStatus:temp[0].ActiveStatus,
    });
  });
});
```

Figure 4.7: Active and Inactive Plans Screen Code

Output:

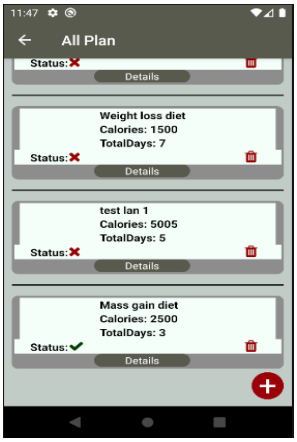


Figure 4.8: Active and Inactive Plans Screen

This is “Active and Inactive Plans Screen” In this screen user user view active plan and change plan active to inactive or inactive to active.If user tap on details button user switch to today meal screen. User can also delete any diet plan.

Create Plan

If (Plan != exists in database)

{create new plan}

Else {Do nothing}

```
return (
  <View style={styles.container}>
    <Modal
      animationType={"fade"}
      transparent={true}
      visible={this.state.isVisible}
      onRequestClose={() => { console.log("Modal has been closed.") }}>
      {

        <View style={{ backgroundColor: '#c1ccc7', flex: 0.95, height: 400, width: 340, margin: 10, marginTop: 20, borderWidth: 5, borderRadi
          <View style={{ flexDirection: "row", justifyContent: "space-between", }}>
            <TouchableOpacity>
              <Text style={styles.createplan}>CreatePlan</Text>
            </TouchableOpacity>

            <TouchableOpacity onPress={() => {
              this.setState({ isVisible: !this.state.isVisible })
            }} >

              <Text style={styles.createplan}><Icon name="close" size={20} /></Text>

            </TouchableOpacity>
          </View>
          <ScrollView keyboardShouldPersistTaps="handled">
            <KeyboardAvoidingView
              behavior="padding"
              style={{ flex: 1, justifyContent: 'space-between' }}>
              <MyTextInput
                placeholder="Enter Title"
                onChangeText={Title => this.setState({ Title })}
                style={{ padding: 10 }}
              />
```

Figure 4.9: Create Plan Screen Code

Output:

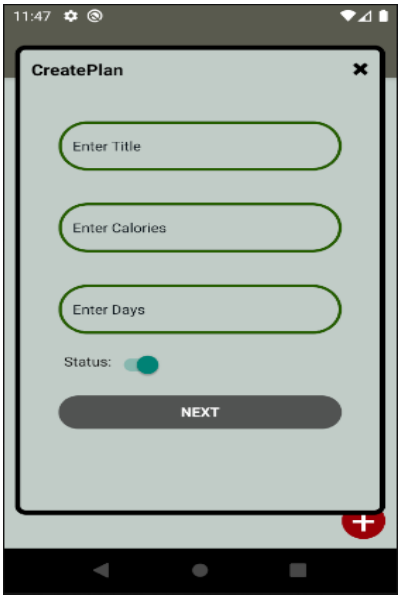


Figure 4.10: Create New Plan Screen

4.4 Graphical User interface

Following are the screenshots of application with their description.

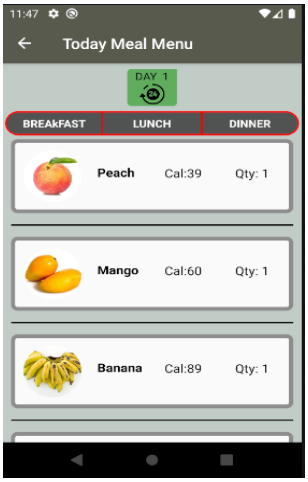


Figure 4.11: Single Day Food Item

In Figure 4.11 User have different tabs.User can view breakfast, lunch and dinner according to days. If user select day 1 user view day one meals and user can change days from days button.

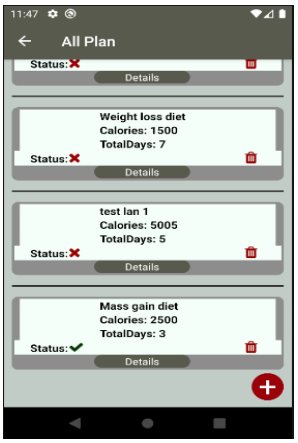


Figure 4.12: All Plan Screen

In Figure 4.12 User have different buttons. If user click on details button than user switch to single day meals screen and if user click on status button than user can change status active to inactive. User also delete plan in this screen .

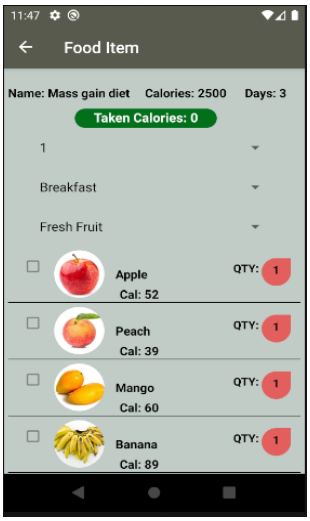


Figure 4.13: Food Item Screen

In Figure 4.13 User have different food items amd many dropdowns in this screen. User can change days no and meal category and food item category. User can select food items in this screen.

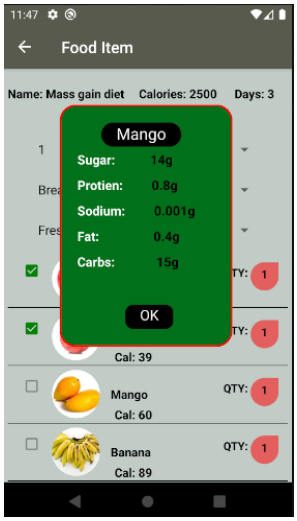
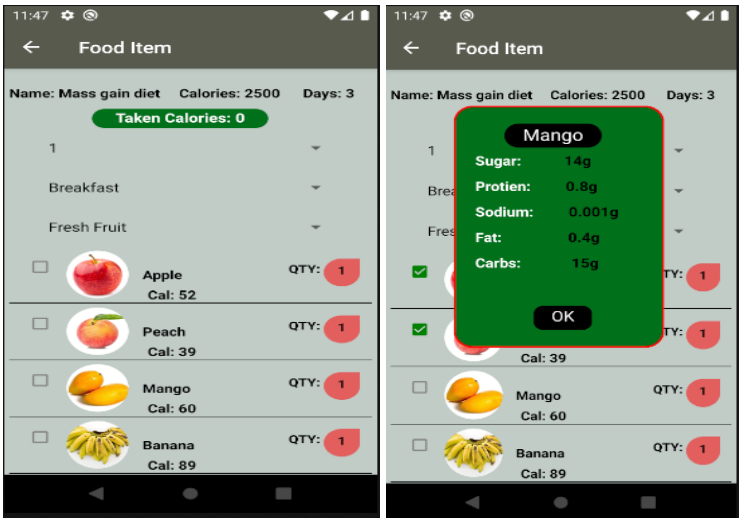


Figure 4.14: Ingredient Screen

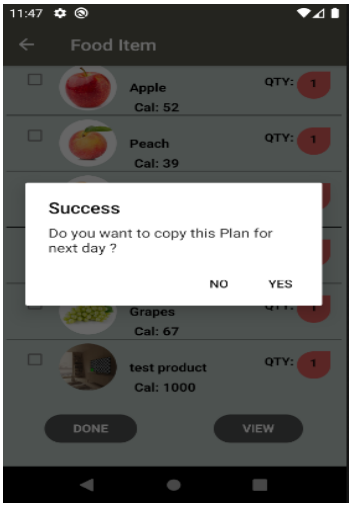
In Figure 4.14 user tap on list and dialog box will show with food item ingredients.

Adding food to Plan Screens

In this Screen the user can enter his/her preferred foods in the selected plan, this screen lets the user select the plan day, food to eat, food quantity and meal time, the user can also copy his plan to next day or any other day. The screen uses calories, sodium and sugar checks so they don't get above recommended limit. On top there is representation of total calories taken for today.



(a) (b)



(c)

Figure 4.15 (a,b,c): adding food to plan screen

Add New Food Item Screen

In this Screen user can add new food item with ingredients and images.

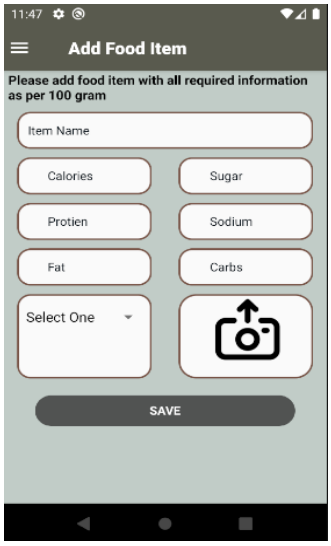


Figure 4.16: Add Food Item Screens

In precise, we discuss about the tools and technologies and the application coding and the design are discussed step by step. The implementation is based on some tools and technology.Everything mentioned above like sedu code of screens and code of screens and screens design also attached.

CHAPTER 5

CONCLUSION

5.1 Introduction

In this chapter we discuss the remark about the application and discuss the future direction and limitation of application that the user are used in future need.

5.2 Concluding Remarks

This project is successfully completed and met the requirements and objectives. The application covers all the major modules which are used to fulfill the requirements and facilities and also provide effective and efficient platform for users.The goals of Food Diet Plan are to let user create diet plan according to their weight. App is giving ability to deal the food on the basis of ingredients and calories. This App is also winging the ability to activate and deactivate plan .

5.2 Future Directions

This project has very vast scope in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in term of expansion. The record of user health at the end of diet plan will make to know whether the diet plan suits him or not. We will also connect this to web servers so that user can share his plan and diet with other users so that everyone can get benefit according to his requirement.

5.3 Limitations

This software application is easy to use and can install in Android/IPhone supported mobile/devices. Users have to just download and install it and use it without any limitations, the only thing is that users have a strong internet connection and know how to work with applications.

REFERENCES

[1] React Native Elements, Cross Platform and easy to use components

[2] React Navigation, Routing and Navigation in native apps

[3] React Native Papers, Standard quality Material Designs

[4] Github, Contributing to open source community

[5] Aboutreact.com.

ANNEXURE

Annexure A:

Home Screen

Step 1: Start

Step 2: Create variables to store user input

Step 3: Clear the variable in case it is not empty

Step 4: Ask the user to for input

Step 5: Store the response in variables

Step 6: Check the stored response to validate input

Step 7: not valid? Go back to step 3

Step 8: End

Annexure B:

User Dash Board

Step 1: Start

Step 2: Create variable for show food item and day no

Step 3: Clear the food item array in case it is not empty

Step 4: Check day no

Step 5: Show the response

Step 6: Check the stored response to validate input

Step 7: End

Annexure C:

Ingredient Screen

Step 1: Start

Step 2: Create variables to store ingredients

Step 3: Clear the variable in case it is not empty

Step 4: Ask the user to upload image of food item

Step 5: Ask the user to for input

Step 6: Store the response in variables

Step 7: Check the stored response is already exist or not

Step 8: Not valid? Go back to step 3

Step 9: End

Annexure D:

Food Item Screen

Step 1: Start

Step 2: Create variables to store food items

Step 3: Clear the variable in case it is not empty

Step 4: Ask the user to add food item

Step 5: Store the response in variables

Step 7: Check the stored response is valid

Step 8: Not valid? Go back to step 3

Step 9: End