



COMSATS University, Islamabad Pakistan

Food Explorer

By

Shamas Ul Din CIIT/SP18-BCS-150/ISB

Talal Nasir CIIT/SP18-BCS-159/ISB

Supervisor

Mr. Azfar Shakeel

Bachelor of Science in Computer Science (2018-2022)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.



COMSATS University, Islamabad Pakistan

Food Explorer

A project presented to
COMSATS University, Islamabad

In partial fulfillment
of the requirement for the degree of

Bachelor of Science in Computer Science (2018-2022)

By

Shamas Ul Din CIIT/SP18-BCS-150/ISB

Talal Nasir CIIT/SP18-BCS-159/ISB

DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Shamas Ul Din

Talal Nasir

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (CS) “Food Explorer” was developed by **Shamas Ul Din (CIIT/SP18-BCS-150)** and **Talal Nasir (CIIT/SP18-BCS-159)** under the supervision of “Mr. Azfar Shakeel” and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelor of Science in Computer Sciences.

Supervisor

External Examiner

Head of Department
(Department of Computer Science)

Executive Summary

The system under consideration, Food Explorer, serves as the complete guide to the audience having trouble in opting their daily meal over a new unfamiliar place. The system assists in searching good food in nearby places. It helps users to get quality food and good restaurants with less effort. Searching details about a restaurant and its food items is now just a few taps away from the user.

Food Explorer assists its audience in deciding what to eat, when to eat, and where to eat it, all in accordance with the personal preferences and priorities of the user, as the system offers a complete depiction about nutritional facts present in each food item, that it contains. In this way, user can have food according to their diet preset.

Food Explorer is not just an application on which users can search for food, but along with it, they can also see recipes of different food items. If they are willing to make their food at home, they can search for a food product and can know everything about it with just one tap.

Food Explorer, a cross-platform smartphone application that helps its users to find the name of a particular food, recipes about how they are made, ingredients of that food, price of that meal at a particular place around you, and also the routes and directions on the map for the selected restaurant selling a particular food item.

Moreover, say that there is a tourist unfamiliar of the culture and he cannot decide for himself because he have no knowledge of the place, he is currently at. Here, the local community will play its role to guide him about the stuff that he should try out, based on their opinions and popularity of multiple peculiar places in that location.

In addition to the above facilities, the users of this system are provided with tailored fitness goals and guidelines in the form of categorized workouts and trainings that are all in accordance with the recommendations of best health and fitness practices followed around the globe. To follow them up efficiently, users are also facilitated with various pre-cut diet plans that they can opt and race towards their health and fitness preferences.

Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Mr. Azfar Shakeel”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are grateful to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us with the values of honesty & hard work.

Shamas Ul Din

Talal Nasir

Abbreviations

| | |
|-------------|-----------------------------|
| IC | Image Classification |
| AR | Augmented Reality |
| WS | Web Scraping |
| DB | Database |
| SR | Speech Recognition |
| WD | Web Driver |
| CSV | Comma Separated Values |
| JSON | Java Script Object Notation |
| BS4 | Beautiful Soup 4 |
| TS | Tensor Flow |
| BMI | Body Mass Index |

Table of Contents

| | | |
|-------|--|-----|
| 1 | Introduction..... | 1 |
| 1.1 | Vision Statement | 1 |
| 1.2 | Related System Analysis/ Literature review | 1 |
| 1.3 | Project Deliverables | 3 |
| 1.4 | System Limitations/ Constraints | 4 |
| 1.5 | Tools and Technologies | 5 |
| 1.6 | Relevance to Course Modules..... | 8 |
| 2 | Problem Definition..... | 9 |
| 2.1 | Problem Statement | 9 |
| 2.2 | Problem Solution..... | 9 |
| 2.3 | Objectives of the Proposed System..... | 9 |
| 2.4 | Scope | 10 |
| 2.5 | Modules..... | 10 |
| 3 | Requirement Analysis..... | 13 |
| 3.1 | User Classes and Characteristics..... | 13 |
| 3.2 | Requirement Identification Technique | 14 |
| 3.3 | Functional Requirements..... | 49 |
| 3.4 | Non-Functional Requirements | 115 |
| 3.4.1 | Reliability..... | 115 |
| 3.4.2 | Usability | 115 |
| 3.4.3 | Performance | 115 |
| 3.4.4 | Security | 116 |
| 3.5 | External Interface Requirements | 116 |
| 3.5.1 | User Interfaces Requirements | 116 |
| 3.5.2 | Software interfaces..... | 116 |
| 3.5.3 | Hardware interfaces | 117 |
| 3.5.4 | Communications interfaces..... | 117 |
| 4 | Design and Architecture | 118 |
| 4.1 | Architectural Design | 118 |
| 4.2 | Design Models..... | 119 |
| 4.3 | Data Design | 130 |
| 4.3.1 | Data Dictionary | 130 |
| 4.4 | Human Interface Design..... | 132 |
| 4.4.1 | Screen Images | 132 |
| 4.4.2 | Screen Objects and Actions | 133 |

| | | |
|--------|--|-----|
| 5 | Implementation | 134 |
| 5.1 | Algorithm | 134 |
| 5.1.1 | Image Classification..... | 134 |
| 5.1.2 | Automated Web Scraping AI Bot..... | 135 |
| 5.2 | External APIs/SDKs..... | 138 |
| 5.3 | User Interface | 139 |
| 5.3.1 | Splash Screen..... | 139 |
| 5.3.2 | Sign up Screen | 139 |
| 5.3.3 | Login Screen | 140 |
| 5.3.4 | Home Screen..... | 140 |
| 5.3.5 | Restaurant Section Screen..... | 141 |
| 5.3.6 | Restaurant Location Screen | 141 |
| 5.3.7 | Restaurant Searching Screen..... | 142 |
| 5.3.8 | Restaurant Menu Screen | 142 |
| 5.3.9 | Food Searching Screen | 143 |
| 5.3.10 | Restaurant About Screen..... | 143 |
| 5.3.11 | Restaurant Feedback Screen | 144 |
| 5.3.12 | Posting Review Screen | 144 |
| 5.3.13 | Cities Section Screen | 145 |
| 5.3.14 | Cities Selection Screen | 145 |
| 5.3.15 | Workout Section Screen | 146 |
| 5.3.16 | Workout Types Screen..... | 146 |
| 5.3.17 | Exercises Details Screen..... | 147 |
| 5.3.18 | Exercise Instructions Screen | 147 |
| 5.3.19 | Diet Plan Section Screen..... | 148 |
| 5.3.20 | BMI Calculation Screen..... | 148 |
| 5.3.21 | Diet Plan Category Screen..... | 149 |
| 5.3.22 | Weekday Selection Screen..... | 149 |
| 5.3.23 | Meal Plan Contents Screen | 150 |
| 5.3.24 | Side Drawer Screen..... | 150 |
| 5.3.25 | Suggested Restaurants Screen..... | 151 |
| 5.3.26 | Change Suggestions Location Screen | 151 |
| 5.3.27 | Change Language Screen..... | 152 |
| 5.3.28 | Sample Language Screen [1/3] | 152 |
| 5.3.29 | Sample Language Screen [2/3] | 153 |
| 5.3.30 | Sample Language Screen [3/3] | 153 |
| 5.3.31 | Camera Screen | 154 |
| 5.3.32 | Gallery Screen..... | 154 |
| 5.3.33 | Login Screen | 155 |
| 5.3.34 | Dashboard Screen | 155 |

| | | |
|--------|----------------------------------|-----|
| 5.3.35 | Restaurant Screen..... | 156 |
| 5.3.36 | Profile Screen..... | 156 |
| 5.4 | Deployment | 157 |
| 6 | Testing and Evaluation | 158 |
| 6.1 | Unit Testing..... | 158 |
| 6.2 | Functional Testing..... | 177 |
| 6.3 | Integration Testing | 185 |
| 7 | Conclusion and Future Work | 188 |
| 7.1 | Conclusion..... | 188 |
| 7.2 | Future Work | 188 |
| 8 | References..... | 188 |

List of Figures

| | |
|---|-----|
| Figure 1: Use Case Diagram showing set of actions for user from home screen | 14 |
| Figure 2: Use Case Diagram explaining process of exploring restaurants | 15 |
| Figure 3: Use Case Diagram showing process of exploring workout plans and diet plans..... | 16 |
| Figure 4: Use Case Diagram showing process of exploring food products..... | 17 |
| Figure 5: Use Case Diagram showing set of actions for user from side panel | 18 |
| Figure 6: Use Case Diagram showing set of actions for admin from web app | 19 |
| Figure 7: System architecture diagram of food explorer | 118 |
| Figure 8: Activity diagram for actions performed from the home screen of mobile application | 119 |
| Figure 9: Activity diagram explaining the exploration of restaurants | 120 |
| Figure 10: Activity diagram explaining the exploration of food items | 121 |
| Figure 11: Activity diagram explaining the exploration of workout plans..... | 122 |
| Figure 12: Activity diagram explaining the exploration of diet plans | 123 |
| Figure 13: Activity diagram for set of actions performed from the side drawer | 124 |
| Figure 14: Activity diagram illustrating the exploration of cities..... | 125 |
| Figure 15: Activity diagram for roles and responsibilities of admin from web app..... | 126 |
| Figure 16: Data Flow Diagram (Level 0) | 127 |
| Figure 17: Data Flow Diagram (Level 1) | 128 |
| Figure 18: Data Flow Diagram (Level 2) | 129 |
| Figure 19: Entity Relation Diagram of Food Explorer | 131 |
| Figure 20: Splash Screen | 139 |
| Figure 21: Sign up Screen..... | 139 |
| Figure 22: Login Screen..... | 140 |
| Figure 23: Home Screen | 140 |
| Figure 24: Restaurant Section Screen | 141 |
| Figure 25: Restaurant Location Screen..... | 141 |
| Figure 26: Restaurant Searching Screen | 142 |
| Figure 27: Restaurant Menu Screen..... | 142 |
| Figure 28: Food Searching Screen..... | 143 |
| Figure 29: Restaurant About Screen | 143 |
| Figure 30: Restaurant Feedback Screen..... | 144 |
| Figure 31: Posting Review Screen | 144 |
| Figure 32: Cities Section Screen..... | 145 |
| Figure 33: Cities Selection Screen | 145 |
| Figure 34: Workout Section Screen..... | 146 |
| Figure 35: Workout Types Screen | 146 |
| Figure 36: Exercise Details Screen | 147 |
| Figure 37: Exercise Instructions Screen | 147 |

| | |
|---|-----|
| Figure 38: Diet Plan Section Screen | 148 |
| Figure 39: BMI Calculation Screen | 148 |
| Figure 40: Diet Plan Category Screen | 149 |
| Figure 41: Weekday Selection Screen | 149 |
| Figure 42: Meal Plan Contents Screen..... | 150 |
| Figure 43: Side Drawer Screen | 150 |
| Figure 44: Suggested Restaurant Screen..... | 151 |
| Figure 45: Change Suggestion Location Screen..... | 151 |
| Figure 46: Change Language Screen | 152 |
| Figure 47: Sample Language Screen [1/3]..... | 152 |
| Figure 48: Sample Language Screen [2/3]..... | 153 |
| Figure 49: Sample Language Screen [3/3]..... | 153 |
| Figure 50: Camera Screen..... | 154 |
| Figure 51: Gallery Screen | 154 |
| Figure 52: Login Screen (Web) | 155 |
| Figure 53: Dashboard Screen (Web)..... | 155 |
| Figure 54: Restaurant Screen (Web)..... | 156 |
| Figure 55: Profile Screen (Web) | 156 |

1 Introduction

Food Explorer assists the general public and helps the community to stay healthy, content, and fit. Guiding them out in an unknown place and situation; fulfilling the dire necessity of mankind. An easy to use, central platform is provided to the people for their convenience on daily basis, hence opening up a new facilitating ground in this era of advancement and technology. Food Explorer serves as the complete guide to the audience having trouble in opting their daily meal over a new unfamiliar place. The system assists in searching good food in their nearby places. It helps users to get quality food and good restaurants with less effort, along with their ratings and reviews. Furthermore, users can focus on their fitness and diet plans.

1.1 Vision Statement

Food Explorer is a platform for two types of users: general public and the system admin. The system is to mainly facilitate the general public. Using the food explorer will enable them to conveniently select their meals from the best restaurants of their choice around them. User can either search for food, or restaurant. Restaurant being selected can also be reached out by the user through the maps, using our system. Users can also opt to their desired workout plans. To further assist their goals, they can also go select the suitable diet plans to follow. System will suggest the best diet plan to user based on their Body Mass Index calculation.

Users can also have a bird's eye view of the food item before them and further dig-in to its preparation details, enabling them to virtually visualize the figures that can normally be hard to estimate.

1.2 Related System Analysis/ Literature review

Following are some of the major, somewhat related existed systems that we would like to mention here in this section because of the services they provide.

- Bixby, a virtual assistant by Samsung Electronics, uses voice commands to perform different tasks on the smartphone. One of the important features is that Bixby can scan objects to find information of that object like: similar objects, shopping, and text from that image. Bixby is not available in all countries and models. [1]
- In Samsung Galaxy S9, Samsung introduced a feature inside its camera which can scan food pictures and tell the calories of the food from that picture. This feature was not available in any of their previous product and later. [2]

- Zomato is an Indian-based restaurant aggregator system, earlier to be known as Foodie Bay, founded by two native IT graduates. Later on, Ant Financial (subsidiary of Alibaba) got major shares of it. [3]
- In Pakistan, Foodpanda is one of the major sources that provide details about different food places and items. Foodpanda is helpful to find restaurants and order various meal deals based on their own delivery system. [4]
- Foursquare is a similar application based on the global community that gives restaurants list using different filters. [5]
- Yelp is a website and application that provides user reviews about different things like restaurants, places, saloons, etc. People use Yelp to explore more options for their shopping. Yelp is an American based company. [6]

As you can see that these tick marks are all scattered in different mentioned systems and are not completely filled against any of the column. The system that we are proposing will contain all the above features, along with many more extra features and functionalities.

Table 1: Related System Analysis with proposed project solution

| Sr. No. | Characteristics | Bixby | Zomato | Foodpanda | Foursquare | Snappy Meal | Recipe |
|---------|-----------------------------|-------|--------|-----------|------------|-------------|--------|
| 1 | Food Recognition | ✓ | | | | ✓ | |
| 2 | Local Restaurants | ✓ | | ✓ | ✓ | | |
| 3 | Address of Restaurant | | ✓ | | ✓ | | |
| 4 | Nearest Restaurant | | ✓ | | | | |
| 5 | Recipe | ✓ | | | ✓ | ✓ | ✓ |
| 6 | Direction to the Restaurant | | ✓ | | ✓ | | |
| 7 | Currently Available Items | | | ✓ | | | |

Our system, Food Explorer covers up all the deficiencies defined against each of the above-mentioned systems.

1.3 Project Deliverables

The overall developments produced and to be produced in the future, that we owe delivering for this project of ours Food Explorer, are all identified and mentioned here-by.

- Documentations
 - Scope Document
 - Software Requirement Specification Document
 - Software Design Specification Document
 - Software Test Document
 - Final Report
- Mobile Application
 - Language Dictionaries
 - Food Identification
 - Cities, Restaurants, About, Foods, and Reviews implementation
 - Workouts and Diet plans
 - Route Planner
 - Application Customization
 - Profile Management
- Web Application
 - Admin Control Panel
 - System Management Dashboard
 - Portal Customization
- Image Classification Model along with the training dataset
- Flask API that communicates between ML model and Mobile App
- Project for creating language dictionaries
- Cloud Storage Database along with the data
- Web Scraper
 - ReCAPTCHA solver along with Speech Recognition script
 - Selenium Automation Bot
 - CSV Files
- AR implementation for food overview

1.4 System Limitations/ Constraints

Following are some of the limitations of Food Explorer at current stage:

LI-1: Food Explorer only classifies/identifies only a max of 5 food items at this stage, i.e., Biryani, Burger, Fries, Pizza, and Steak. This number can be increased up to any count with the help of relevant dataset and re-training of the same machine learning model.

LI-2: The total number of native languages in which the smartphone application of Food Explorer can be operated, is 5 at this stage. This number can be increased up to any count by creating the dictionary of desired language with the provided code and importing it into the project.

LI-3: The total number of cities in which our project Food Explorer is currently partially implemented, is 5. This number can be increased at any stage by using the web scraping module provided with the project.

LI-4: A total of 5 items can be identified by Food Explorer's AR module implementation at this stage. This is because that AR activities are directly loaded into the smartphone's processor and is an extensive task to be performed. Thus, this number is kept intentionally to a minimum on purpose.

1.5 Tools and Technologies

These are all of those tools and technologies that we have used for the development purpose of our project.

Table 2: Tools and Technologies for Proposed Project

| Tools And Technologies | Tools | Version | Rationale |
|------------------------------|--------------------|--------------|----------------------|
| | Visual Studio Code | 1.52.0 | IDE |
| | Firebase | 26.1.1 | DBMS |
| | PyCharm | 2019.3.4 | IDE |
| | Unity | 2020.3.20f1 | Engine |
| | Unity Hub | 2019 | Project Manager |
| | Git Hub | 2.22.6 | Code Hosting |
| | Adobe XD | 2.0.5 | Mockups Creation |
| | Draw | 1.9.12.13913 | UML Diagrams |
| | GanttPRO | 2020 | Gantt Charts |
| | MS Word | 2016 | Documentation |
| | MS Power Point | 2016 | Presentation |
| | Technology | Version | Rationale |
| | JavaScript | ES7 | Programming language |
| | React | 17.0.1 | JavaScript Library |
| | React Native | 0.63 | JavaScript Framework |
| | Expo | 40.0.0 | JavaScript Framework |
| | C# | 10.0 | Programming Language |
| | AR Foundation | 4.2.1 | Framework |
| | Python | 3.9.1 | Programming Language |
| | Beautiful Soup | 4.9.3 | Python Library |
| | Flask | 1.1.2 | Python Framework |
| | Pydub | 0.25.1 | Python Library |
| | Speech Recognition | 3.8.1 | Python Module |
| | TensorFlow | 2.0 | Python Library |
| | Selenium | 4.0 | Python Framework |
| | Google Maps | 3.43 | API |

Below here, is a further brief description against each of the tools and technologies that we have utilized and mentioned above:



JavaScript, the main programming language used for the development of our smartphone application and web application.



React (JavaScript Library), that we have used for the development of the web application.



React Native (JavaScript Framework), used for the development of our smartphone application.



Expo framework used to assist the development, building, and deployment phases of the react native application.



We used Visual Studio Code as our main IDE for coding of React and React Native, in JavaScript.



Python programming language used for the development of web scrapper, image classification, custom API, and making translation dictionaries.



TensorFlow (Python library), provided by the Google used for the machine learning model development.



Flask

Flask (Python Framework) that we have used for writing our custom API, to enable the communication between mobile app and ML model.

BeautifulSoup

Beautiful Soup (Python Library) used for scraping out webpage's content from the HTML of a website.

Pydub

Pydub is Python library that we have used for manipulating audio files (.wav format files), in Captcha solver.



SPEECH RECOGNITION

Speech Recognition is a Python module that enables machine to understand human speech and convert that speech into text, utilizing Google Speech API under the hood.



Selenium, an open-source package that we have used for automating our scraping bot, when pairing it up with chrome web driver.



We have used PyCharm as our primary IDE for the development in Python Programming Language.



C Sharp is used as the primary programming language in coding for augmented reality.



AR Foundation is used as the main framework for the development of augmented reality.



Unity Hub is used as the project manager, package installer, and version controller for the unity project, in AR implementation.



Unity Engine is selected as the platform for the production of features based on the Augmented Reality.



Google Maps

Google Maps API for showing the local maps to the user, and navigating him to other spots via that map, using geocoding and reverse geocoding.



Firebase (Firestore), a platform provided by Google, is used to create the cloud-based database of our project.



GitHub, well-known tool used for code hosting, collaboration, and version control management.



We have used Draw.io tool, for making all of the UML diagrams of our project.



We used GanttPRO tool for making the Gantt chart of our project.



Adobe XD is selected for creating the mockup screens of our project.



Microsoft Word, for documentation purpose.



Microsoft Power Point, for making presentation slides.

This concludes the bunch of technologies that we have used for the purpose of development of our project, Food Explorer.

1.6 Relevance to Course Modules

Some core features and functionalities belonging to Food Explorer are only achieved due to the sheer abstract of the courses learned throughout our academic career. The most honorable mentions among them are, Topics in Computer Science-I (Advanced Web Programming) and Mobile Application Development. Other courses that also played a vital role in the development of our project are, Web Technologies, Software Engineering Concepts and Data Communications and Computer Networks, Database Systems, Operating Systems, and last but not the least, Human Computer Interaction.

There are also some implemented core modules -like image classification, augmented reality, web scraping- that we have studied none about from our university and self-learned ourselves from external sources.

2 Problem Definition

2.1 Problem Statement

It is difficult to find the meal that is suitable for your health. Often you are not aware of the fact that the thing you're eating, whether it is healthy or not. If you want to try a specific food item at your home, that you've only eaten in a restaurant up till now, it is difficult to find the accurate recipe of it. If you want to try out some new stuff that you have only seen in some advertisement or has been recommended by someone, people mostly face problem on where to find that meal.

Allow us to paint you a scenario. For example, you're in China for a visit and there are many types of meals that are provoking your temptations, but you don't know whether it is Halal or Haram and you cannot even ask the cook because of the language barrier as you cannot speak Chinese (Mandarin). What would you do in such a scenario?

2.2 Problem Solution

The purpose of this system is to facilitate general public (specially tourists and travelers) to find their meal easily, with much convenience than it already is. This is achieved by the idea of doing less and getting more, i.e., via just scanning a food item from your smartphone's camera, the system will tell you: the name of that item, its different variants, its recipes, the complete list of nutritional facts it contains. User can also search for a specific restaurant or select one from the provided list to view the complete menu of that restaurant, along with the food products and their prices, details about that restaurant, like its address, opening time, closing time, and its rating, along the complete routeway to that restaurant from user's current location with live tracking and guidance. User can post reviews on different restaurants and can also rate them after visiting that restaurant so that it can be easy for visitors to choose the meal that is not a wastage of their time and money.

2.3 Objectives of the Proposed System

Following are the major objective of our proposed systems:

BO-1: Identification of unknown food products through food images.

BO-2: Making nearby restaurants easily available to the users.

BO-3: Making menus and rates of restaurants available to users at their homes.

BO-4: Enabling users to opt for workout plans to stay fit.

BO-5: Motivating users to eat healthy meals, in the form of tailored diet plans, according to their body type.

BO-6: Creating awareness in the users about the nutritional facts and values of the routine food they eat.

BO-7: Conveying processed and easy to prepare recipes for common food products along with their required ingredients.

BO-8: Suggesting popular nearby places to the user, according to their current location, based on the top-rated restaurants.

BO-9: Overcoming the communicational barrier, by providing multiple languages to operate.

BO-10: Guiding users to the restaurants, by providing him with appropriate routes and directions with real-time tracking.

2.4 Scope

There is a number of related applications in the selected domain but almost every one of them has either limited information on the food or the complete set of features are only available on distinct applications and platforms. Our system will serve as the central point for every type of related information about the food e.g., food identification, recipes, restaurants menus, restaurants details, directions, etc. Moreover, we will also facilitate the users with some new unique features, i.e., the food recommendation to a user based on the current most popular venues of that geological location. Moreover, all of the users will also be enabled to rate some particular place that they have already tried, so that whenever a new user, from that location, opens the application, he views the peculiar top rated food item as the suggestion for him. Users can also select from various workout plans and dietary plans that suits them.

2.5 Modules

Following are the identified modules and sub-modules that are implemented in the development of our project. Brief description against each of the module and sub module is mentioned below in the corresponding sub-section:

Module 1: Automated Web Scraping AI Bot

FE-1: Open the link of targeted webpage for a particular city, and make a list for all of its restaurants, locally, in a file.

FE-2: From the list created, sequentially open up the webpages for all of the restaurants from list.

FE-3: Against each restaurant's URL, interact with the opened webpage and collect all of the useful information. This information includes name of restaurant, their rating, their reviews, their address, meals they offer, price of their food items, and images of their food items.

FE-4: Cleanout all of the extracted data and organize it into separate files after classification.

FE-5: If Human-Verification is triggered at any stage during scraping, go to the corresponding webpage to interact with the frame of robot check.

FE-6: In case a verification challenge is raised by the Captcha, the bot automatically switches to the audio test and solves it using Human Speech Recognition.

FE-7: After successfully bypassing Captcha verification, the bot recontinues the scraping job from where it was left out.

Module 2: Image Classification

FE-1: User, using the camera of his smartphone, captures an image, which is then sent to a custom-made API that is running live on a server.

FE-2: Rather than the camera of smartphone, user can also select an image from his mobile gallery that is already saved on his local storage.

FE-3: This API that has been hit by our mobile app, has already loaded the asset package of the machine learning model in it that we have created. The sent image gets processed through that model and creates a JSON response as a result.

FE-4: This JSON response is returned and sent back to the mobile app. Based on this response, the smartphone application shows different results to the user accordingly.

Module 3: User Management

FE-1: User can operate the smartphone application in multiple languages. These languages include, English, Italian, French, German, Spanish.

FE-2: User can effectively change the language of the application at any stage during his experience.

FE-3: User can explore the cities in which he can facilitate himself with our system. This list indicates the cities in which our system is fully implemented and deployed.

FE-4: In each of the city, user can view multiple restaurants of that city, that are registered with our system.

FE-5: User can select a restaurant, and view complete menu of this restaurant, including the price tag against each of the food item of this restaurant.

FE-6: User can know all about a specific restaurant. These details include the information about the restaurant itself, containing name of the restaurant, address of the restaurant, opening time of the restaurant, closing time of the restaurant, rating of the restaurants, reviews of the restaurant, and much more.

FE-7: User can view the reviews of a restaurant, posted earlier by general public. These reviews include the name of the reviewer, rating that he posted for this restaurant, date of the review, and the comment feedback provided by the reviewer.

FE-8: User can avail the suggestions for the most popular restaurants of that area. These suggestions are based on other users reviews and restaurant's ratings.

Module 4: Navigation

FE-1: User can open geological maps against a restaurant.

FE-2: User can get different directions for a restaurant from his current location.

FE-3: User can select a route to be followed on the map.

FE-4: User will be tracked in real time at each stage of his traveling, using our system.

FE-5: User will be provided with multiple traveling routes that are optimal in different conveyance scenarios.

Module 5: Augmented Reality

FE-1: User can observe the food details directly from his smartphone camera, by placing the real-world object in the vision of the mobile camera.

FE-2: The details that are presented to the user about the food item, is basically a bird-eye view of the food product. These details include calorie count, carbohydrate measure, fat level, fiber, protein, sodium, and sugar level of that particular food item.

FE-3: User can scan a total of 5 food products from his smartphone camera. These 5 food products contain: biryani, burger, fries, pizza, and steak.

Module 6: Health and Care

FE-1: User can explore different categories of workout plans. These categories include homebased workouts, muscle building workouts, and weight losing workouts.

FE-2: In each of the above defined categories, user can select a targeted area of muscles, or the specific body part that he wants to work on.

FE-3: In each exercise type selected, user can view and read about multiple exercises and workouts provided there. These exercise details include exercise names, visual demonstration, total circuits, and reps in each circuit to be performed.

FE-4: User can calculate his body mass index using our system, that will help him know his body type and choose a diet plan according to the suggestion.

FE-5: For calculating body mass index, user will have to enter his height in feet and inches, and his body weight in kilograms.

FE-6: User can explore multiple tailored diet plans from our system. There is total 3 diet-plans offered in our system, i.e., underweight diet plan, over-weight diet plan, and balanced diet plan.

FE-7: User can view the details in the selected meal plan. These details include names of the eatable, time at which you are supposed to eat that meal, and the day on which you are to eat that.

Module 7: Admin Dashboard

FE-1: Admin can login and logout from his web application.

FE-2: Admin can register new restaurants in the system.

FE-3: Admin can view existing data of the system.

FE-4: Admin can search existing data of the system.

FE-5: Admin can update any outdated data of the system.

FE-6: Admin can delete unwanted or no longer needed entries from the system.

FE-7: Admin observe and analyze the system's growth, with respect to registered restaurants in each city.

FE-8: Admin observe and analyze the system's growth, with respect to registered users in each city.

3 Requirement Analysis

3.1 User Classes and Characteristics

There are a total of two user classes in our system, i.e., end-user and admin. Following are the pertinent characteristics of the two above mentioned classes.

Table 3: Representation of user classes and characteristics

| User class | Description |
|-----------------|--|
| End-user | An end-user is the entity for the facilitation of whom the whole system is being created. These users can view multiple cities, in which they can explore that city's various restaurants. Against each of the restaurant, every user can familiarize himself with the currently available items and their price tags with offered discounts, if any. End-users can get complete information about each of the restaurant like their name, their address, their opening and closing times, and their ratings. Users can also view feedbacks of other users on these restaurants. Users can also add their own reviews and feedbacks against these restaurants. If user likes a particular restaurant, he can also get routes and directions to go to that restaurant from his current live location. The system will track him travelling in the real-time for best of the experiences. User can explore multiple categories and sub-categories of different workout plans with our system. Along with the workout plans, users can also avail tailored diet plans and meal plans according to a specific goal and body type. To know what is best for the user, he can calculate his body mass index from our system and the system will in return suggest him with the appropriate diet plan according to his body type. Users can get suggestions from the system about the popular places and restaurants around them. User can search and completely know about some food items, just taking a picture of them from their smartphone's camera or by uploading an image from their smartphone's gallery. End-user can also operate the system in 5 different languages, i.e., English, French, German, Italian, and Spanish. |
| Admin | Admin is single personality who is responsible to undertake and look-over all the managerial tasks of the system. Admin's duties include registering new restaurants. Altering or deleting data of the pre-existing restaurants and their food items, along with their prices and discounts, if any. Admin can observe the overall stats of different entities related to the system. These entities include total end-users and restaurants, with respect to the total cities in which the system is implemented. |

3.2 Requirement Identification Technique

For our project Food Explorer, we have adopted the ‘use case’ as our requirement identification and elicitation technique. Rationale behind this decision is that the nature of our system is an end-user application, that is the case in which ‘use case’ technique is the most suitable one. This includes user case diagrams. and detailed tabular use cases. Following are the use case diagrams of our project, created according to each actor’s course of actions:

This use case diagram is smartphone application oriented in which all of the actions that can be performed by him from applications home screen, are illustrated in detail.

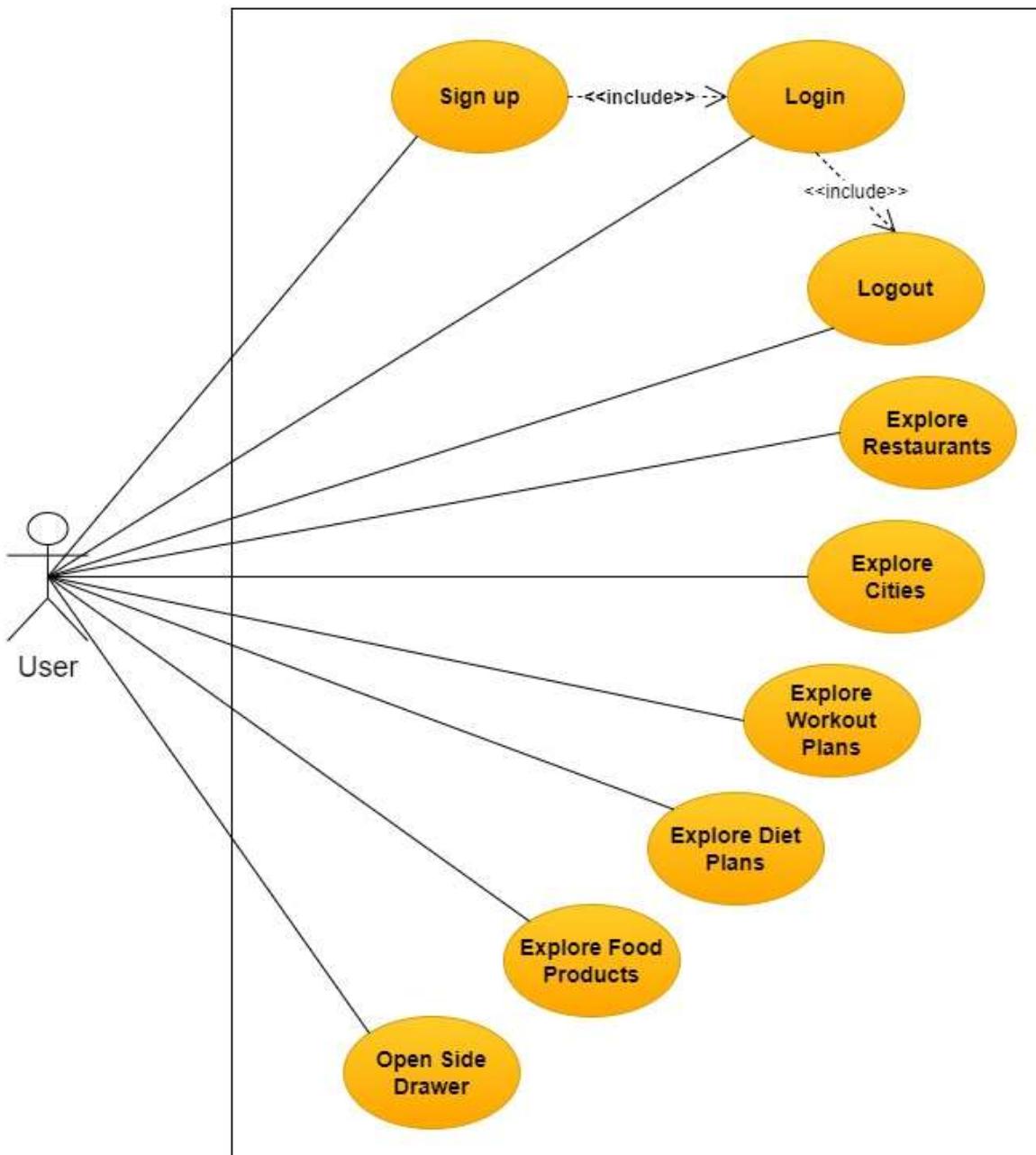


Figure 1: Use Case Diagram showing set of actions for user from home screen

This use case diagram completely explains the process of exploring a restaurant, which is an action that can be performed by end-user, using our smartphone application.

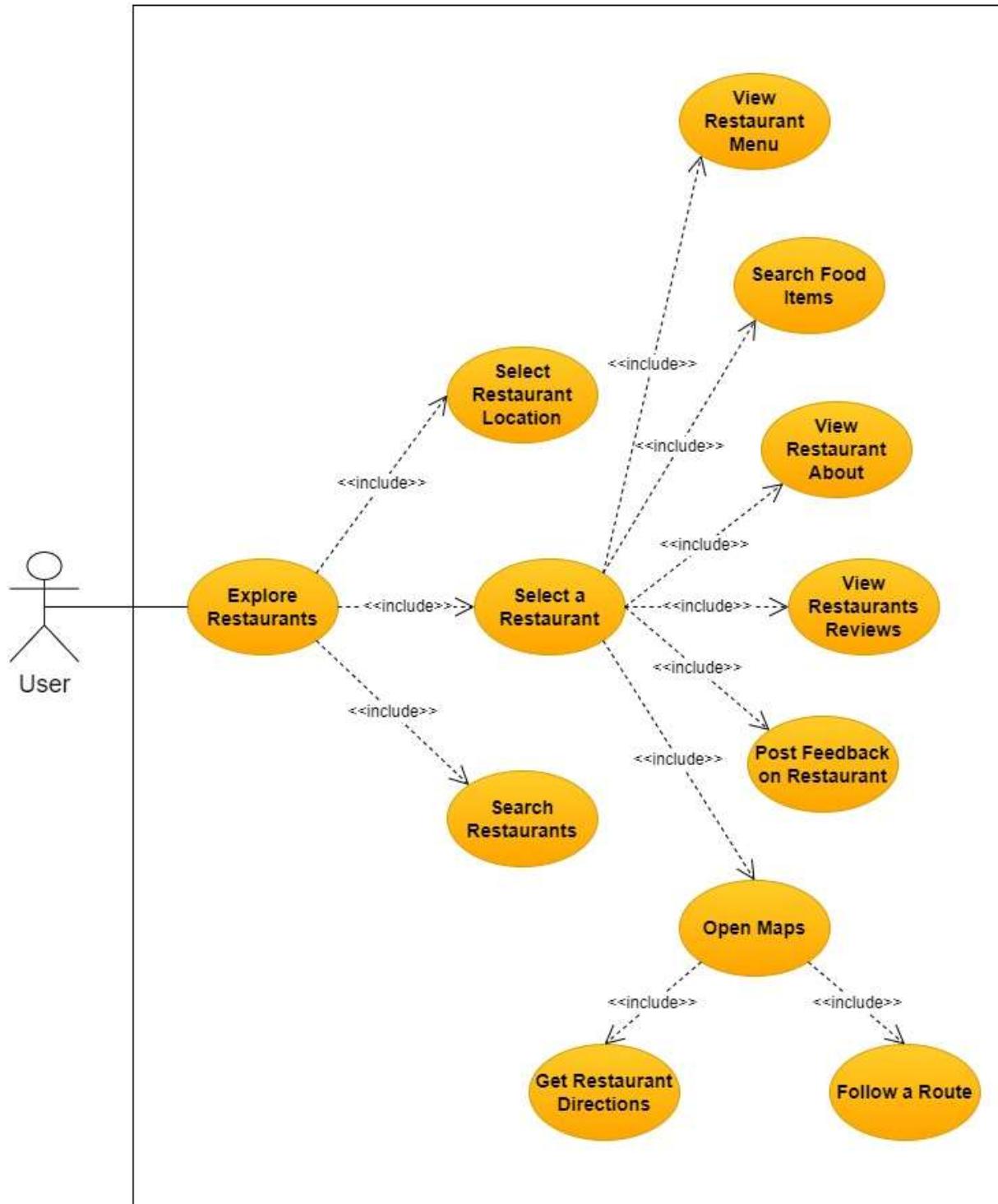


Figure 2: Use Case Diagram explaining process of exploring restaurants

This use case diagram depicts the overall activity for the module of ‘health and care’ that falls under the smartphone application and is in the control of end-user.



Figure 3: Use Case Diagram showing process of exploring workout plans and diet plans

This use case diagram is the depiction for one of the core functionalities of the system, that works with Image Classification and Augmented Reality under the hood, for the purpose of identifying food items and food products through user's smartphone's camera.

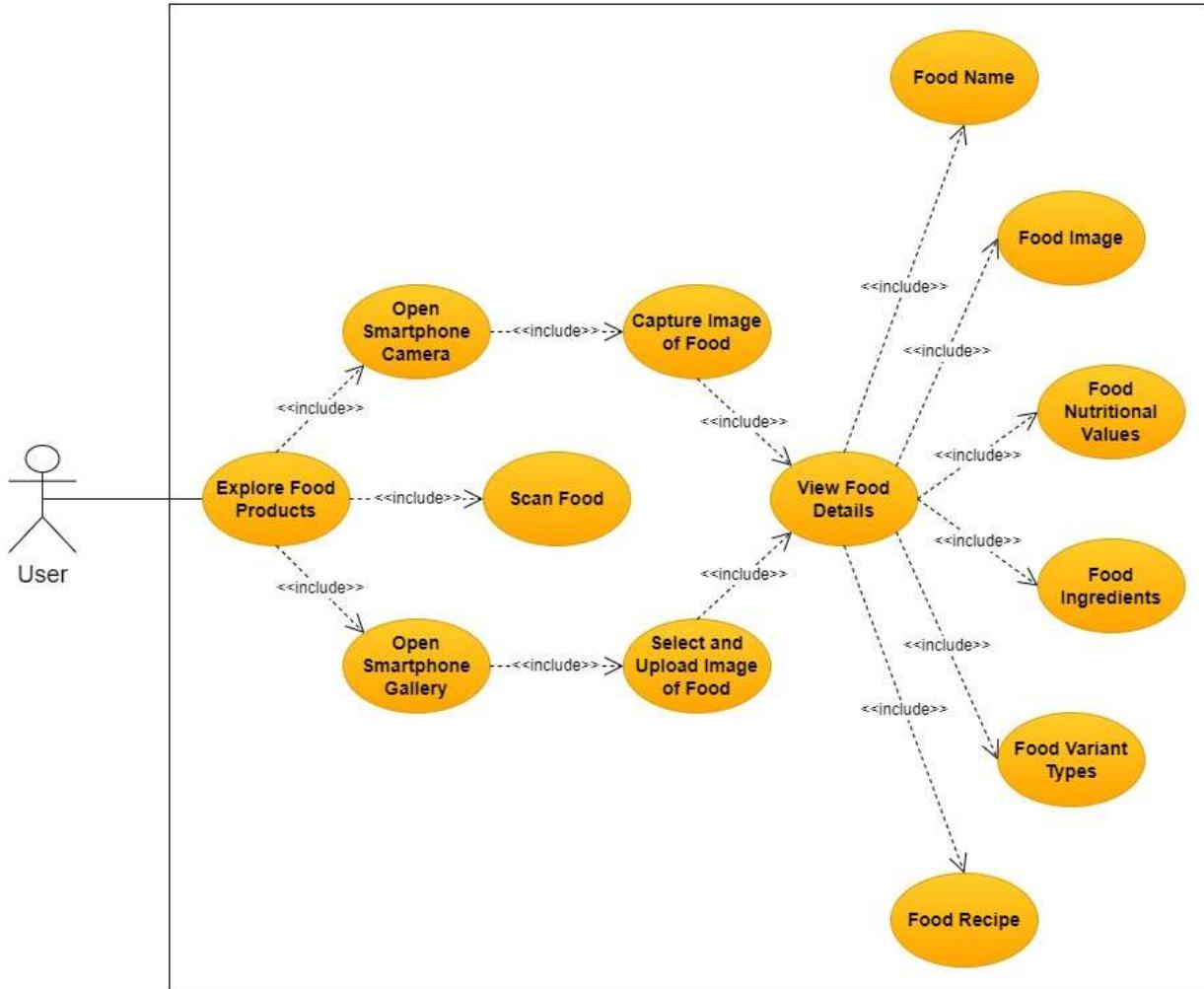


Figure 4: Use Case Diagram showing process of exploring food products

This user case diagram contains all of those set of actions that the end-user can perform from the side drawer menu of the smartphone application.

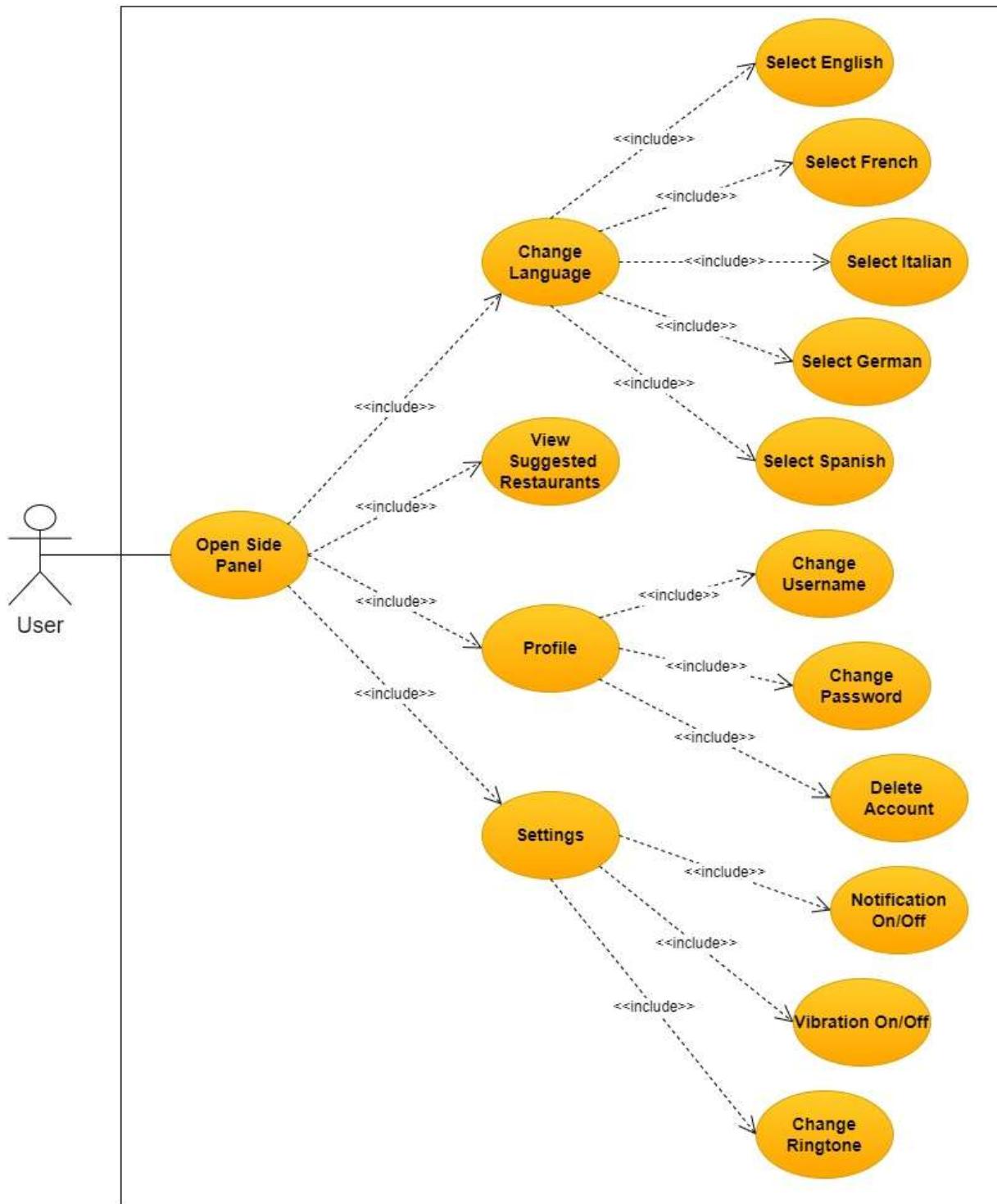


Figure 5: Use Case Diagram showing set of actions for user from side panel

This use case diagram shows all the roles and responsibilities of the admin, that he can operate from the web application of our system.



Figure 6: Use Case Diagram showing set of actions for admin from web app

Following are the identified detailed use cases of our system. Tabular representation for each of them is provided here-by:

Table 4: Sign up (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-1 |
| Use Case Name: | Signup |
| Actors: | Primary Actor: User |
| Description: | The user will be able to Register account and use the application. |
| Trigger: | User will click on “Sign Up” to start registration process. |
| Level: | High |
| Preconditions: | <ul style="list-style-type: none"> 1. Mobile application is downloaded and installed on the smartphone. 2. The smartphone must have access to active internet connection. |
| Postconditions: | User will be successfully registered into the system. |
| Normal Flow: | <ul style="list-style-type: none"> 1. User will download application. 2. User will have option to select ‘Log in’ or ‘Sign up’ option. 3. User will tap on ‘Sign up’ option. 4. User will enter asked data into the provided data fields and tap on proceed. |
| Alternative Flows: | N/A |
| Exceptions: | If user doesn’t have an active internet connection, connection error will be prompted. |
| Business Rules: | N/A |
| Assumptions: | <ul style="list-style-type: none"> 1. User should be able to create account. 2. User should be able to login and use application. |

Table 5: Login (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-2 |
| Use Case Name: | Login |
| Actors: | Primary Actor: User |
| Description: | User will be able to Login after registration. |
| Trigger: | User will open application and fill in credentials to login. |
| Level: | High |
| Preconditions: | User must have to signup first. |
| Postconditions: | User will successfully land into the system for further usage. |
| Normal Flow: | <ol style="list-style-type: none"> 1. User will download application. 2. User will run the application. 3. User will have to enter email and password. 4. User will have to confirm provided email. |
| Alternative Flows: | N/A |
| Exceptions: | If user doesn't have an active internet connection, connection error will be prompted. If user enter wrong credentials. |
| Business Rules: | N/A |
| Assumptions: | <ol style="list-style-type: none"> 1. User should be able to create account. 2. User should be able to login and use application. |

Table 6: Logout (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-3 |
| Use Case Name: | Logout |
| Actors: | Primary Actor: User |
| Description: | User can Logout account. |
| Trigger: | User will click on "Logout" from Drawer to Logout. |
| Level: | High |
| Preconditions: | User should have logged in. |
| Postconditions: | User will close and exit from the system. |
| Normal Flow: | <ol style="list-style-type: none"> 1. User will go to home screen. 2. User will tap on "Logout" button on top right corner from home screen. 3. User will confirm to Logout. |
| Alternative Flows: | N/A |
| Exceptions: | User is not logged into the system at first place |
| Business Rules: | N/A |
| Assumptions: | User is logged in and is able to Logout of account. |

Table 7: Explore Restaurants (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-4 |
| Use Case Name: | Explore Restaurants |
| Actors: | Primary Actor: User |
| Description: | User will be able to explore multiple restaurants. |
| Trigger: | User will tap on restaurant section from home screen. |
| Level: | High |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will find himself on the restaurants screen. |
| Normal Flow: | User will tap on restaurants section from the home screen. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 8: Search Restaurant (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-5 |
| Use Case Name: | Search restaurant. |
| Actors: | Primary Actor: User |
| Description: | User can search restaurant by entering name in search field. |
| Trigger: | User will click on search field on top of screen and search a restaurant. |
| Level: | Medium |
| Preconditions: | User should know the name of restaurant to be searched. |
| Postconditions: | User will successfully search the desired restaurant |
| Normal Flow: | <ol style="list-style-type: none"> 1. From the home-screen, user will tap on the restaurant section 2. From the restaurant screen, user will tap on the search field on top of the screen. 3. User will enter restaurant's name. 4. Searches are dynamically filtered out with each input. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. User entered wrong name of the restaurant. |
| Business Rules: | N/A |
| Assumptions: | User is logged in and is able to search restaurant successfully. |

Table 9: Select Restaurant Location (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-6 |
| Use Case Name: | Select Restaurant Location. |
| Actors: | Primary Actor: User |
| Description: | User will be able to select restaurant location. |
| Trigger: | User will tap on the location filter. |
| Level: | Medium |
| Preconditions: | User should be logged into the application |
| Postconditions: | User will be displayed the restaurants of the selected location. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the restaurant section. 2. From restaurant screen, user will tap on the location filter from the top of the screen. 3. User will select a location from the provided list. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. User has not tapped on the location filter. |
| Business Rules: | N/A |
| Assumptions: | N/A. |

Table 10: View Restaurant Menu (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-7 |
| Use Case Name: | View Restaurant Menu. |
| Actors: | Primary Actor: User |
| Description: | User will be able to see restaurant menu. |
| Trigger: | User will select a restaurant and can see their menu there. |
| Level: | High |
| Preconditions: | User should be logged into the application |
| Postconditions: | User will be able to see the menu of a restaurant |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the restaurant section. 2. From restaurant screen, user will tap on any of the restaurant available in the list. 3. User will be directed to that restaurant's screen. 4. On this screen, user can view this restaurant's menu. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. User has not selected a restaurant. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is able to see restaurant's menu. |

Table 11: View Restaurant Details (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-8 |
| Use Case Name: | View Restaurant's Details. |
| Actors: | Primary Actor: User |
| Description: | User will be able to view the details about a restaurant. |
| Trigger: | User will tap on the about tab from the restaurant screen. |
| Level: | Medium |
| Preconditions: | User have selected a restaurant to see details. |
| Postconditions: | User will be able to see a restaurant's details successfully. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the restaurant section. 2. From restaurant screen, user will tap on any of the restaurant available in the list. 3. User will be directed to that restaurant's screen. 4. User will tap on the about tab from this restaurant's screen. 5. User will be directed to the restaurant's details screen. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. User has not selected a restaurant. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is able to see restaurant details. |

Table 12: View Restaurant Reviews (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-9 |
| Use Case Name: | View Restaurant's Reviews. |
| Actors: | Primary Actor: User |
| Description: | User will be able to view the reviews about a restaurant. |
| Trigger: | User will tap on the reviews tab from the restaurant screen. |
| Level: | High |
| Preconditions: | User have selected a restaurant to see reviews. |
| Postconditions: | User will be able to see a restaurant's reviews successfully. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the restaurant section. 2. From restaurant screen, user will tap on any of the restaurant available in the list. 3. User will be directed to that restaurant's screen. 4. User will tap on the reviews tab from this restaurant's screen. 5. User will be directed to the restaurant's reviews screen. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. User has not selected a restaurant. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is able to see restaurant reviews. |

Table 13: Posting Review (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-10 |
| Use Case Name: | Give Feedbacks and Ratings. |
| Actors: | Primary Actor: User |
| Description: | User will be able to give ratings to restaurants and food items. |
| Trigger: | User will enter his feedback and tap on post. |
| Level: | High |
| Preconditions: | User have selected a restaurant to post review. |
| Postconditions: | User will be able to post a restaurant's reviews successfully. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the restaurant section. 2. From restaurant screen, user will tap on any of the restaurant available in the list. 3. User will be directed to that restaurant's screen. 4. User will tap on the reviews tab from this restaurant's screen. 5. User will be directed to the restaurant's reviews screen. 6. User will tap on Post Review button. 7. User will provide his feedback and hit on Post Review again. |
| Alternative Flows: | N/A |
| Exceptions: | Feedback or rating not given due to bad internet connection. |
| Business Rules: | N/A |
| Assumptions: | User is logged in and have active internet access and can give feedbacks and ratings. |

Table 14: View Restaurant Directions (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-11 |
| Use Case Name: | View Restaurant's Directions. |
| Actors: | Primary Actor: User |
| Description: | User will be able to view the directions of a restaurant. |
| Trigger: | User will tap on the 'Go to Restaurant' button in the about tab from the restaurant screen. |
| Level: | High |
| Preconditions: | User have selected a restaurant to see details. |
| Postconditions: | User will be able to see a restaurant's directions successfully. |
| Normal Flow: | <ol style="list-style-type: none"> From home screen, user will tap on the restaurant section. From restaurant screen, user will tap on any of the restaurant available in the list. User will be directed to that restaurant's screen. User will tap on the about tab from this restaurant's screen. User will tap on the Go to Restaurant button from this screen. User will be redirected to a new screen, where he can view the directions of the selected restaurant from his current location. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. User has not selected a restaurant. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection. User has granted location permission. User is able to perceive directions of restaurant on a map. |

Table 15: Explore Cities (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-12 |
| Use Case Name: | Explore Cities |
| Actors: | Primary Actor: User |
| Description: | User will be able to explore multiple cities. |
| Trigger: | User will tap on cities section from home screen. |
| Level: | Low |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will be able to view multiple cities in which he can avail our services. |
| Normal Flow: | User will tap on location section from the home screen. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 16: Explore Workout Plans (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-13 |
| Use Case Name: | Explore Workouts |
| Actors: | Primary Actor: User |
| Description: | User can familiarize himself with multiple tailored workout plans |
| Trigger: | User will tap on the “Workouts” section |
| Level: | High |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will successfully be able to get to know multiple workout plans |
| Normal Flow: | <ol style="list-style-type: none"> From the home screen, user will tap on “Workouts” section User will be directed to the corresponding screen where he can see different categories of workouts available |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 17: Explore Home Based Workout Plan (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-14 |
| Use Case Name: | Explore Home-based Workouts |
| Actors: | Primary Actor: User |
| Description: | The user will get familiar with the basic workouts that he can practice in his home, even without the need of any prior equipment. |
| Trigger: | User will tap on the “Home Based Workouts” section |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | The user will get familiar with the basic workouts that he can practice in his home, even without the need of any prior equipment. |
| Normal Flow: | <ol style="list-style-type: none"> From home screen, user will tap on the Workout section. From Workout screen, user will tap on Home Based Workout section from the list. User will be directed to the corresponding category of the workouts screen. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 18: Explore Build Muscles Workout Plan (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-15 |
| Use Case Name: | Explore Build Muscles Workouts |
| Actors: | Primary Actor: User |
| Description: | The user will get familiar with the workouts for building muscles. These exercises are mostly intensive in nature; therefore, they are not recommended for elderly people without supervision. |
| Trigger: | User will tap on the “Build Muscles” section |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | The user will get familiar with the intensive workouts that he can practice for achieving his target of building muscles. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the Workout section. 2. From Workout screen, user will tap on Build Muscles section from the list. 3. User will be directed to the corresponding category of the workouts screen. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 19: View Weight Loss Workout Plan (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-16 |
| Use Case Name: | Weight Loss Workouts |
| Actors: | Primary Actor: User |
| Description: | The user will get familiar with the workouts for weight loss. These exercises are mostly intensive in nature; therefore, they are not recommended for elderly people without supervision. |
| Trigger: | User will tap on the “Lose Weight” section |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | The user will get familiar with the intensive workouts that he can practice for achieving his target of weight loss. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the Workout section. 2. From Workout screen, user will tap on Lose Weight section from the list. 3. User will be directed to the corresponding category of the workouts screen. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 20: Calculate User BMI (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-17 |
| Use Case Name: | Calculate BMI (Body Mass Index) |
| Actors: | Primary Actor: User |
| Description: | User will be able to calculate his BMI by entering the asked details |
| Trigger: | User will tap on the section “Calculate BMI” |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | System will calculate and tell the BMI value and description to the user based on the entered figures along with the suggested diet plan |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the Diet plans section. 2. From Diet Plan screen, user will tap on Calculate BMI section from the list. 3. User will be directed to the corresponding screen. 4. User will enter his data asked, into the provided fields and tap on the Calculate BMI button. |
| Alternative Flows: | N/A |
| Exceptions: | <ol style="list-style-type: none"> 1. If user leaves mass field empty before taping “Calculate” button, he’ll be prompted to enter mass first 2. If user leaves height field empty before taping “Calculate” button, he’ll be prompted to enter height first 3. If user doesn’t fill any of the required fields before taping “Calculate” button, he’ll be prompted to enter details first |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 21: Explore Diet Plans (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-18 |
| Use Case Name: | Explore Diet Plans |
| Actors: | Primary Actor: User |
| Description: | User will explore multiple categories of diet plans and select from them based on his personal desire |
| Trigger: | User will tap on the “Diet Plans” section |
| Level: | High |
| Preconditions: | <ul style="list-style-type: none"> • Not necessarily, but the user should calculate his BMI first, so that the system can suggest him the best diet plan that is the most suitable for him • User is logged into the system |
| Postconditions: | User will be familiarized with different diet plans available, among which he can select and see their further details and descriptions. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the Diet plans section. 2. From Diet Plan screen, user will tap on Meal Plans section from the list. 3. User will be directed to the corresponding screen. 4. User will be able to see multiple categories of diet plans. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | People, registered as the users of the system will be allowed to use this feature (no admins allowed) |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 22: Explore Diet for Underweight (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-19 |
| Use Case Name: | Explore Diet Plan for Underweight |
| Actors: | Primary Actor: User |
| Description: | User will explore the diet plan to be followed that is most suitable for people of underweight body type |
| Trigger: | User will tap on the “Diet for Underweight” section |
| Level: | Medium |
| Preconditions: | User must already be in the section of “Diet Plans” |
| Postconditions: | User will be provided with complete instructions and routines that are to be followed in order to overcome his underweight |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the Diet plans section. 2. From Diet Plan screen, user will tap on Meal Plans section from the list. 3. From meal plans screen, user will tap on diet for underweight section provided in the list. 4. User will be able to see meal plans of the selected category. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | People, registered as the users of the system will be allowed to use this feature (no admins allowed) |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 23: Explore Diet to Maintain Weight (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-20 |
| Use Case Name: | Explore Diet to Maintain Weight |
| Actors: | Primary Actor: User |
| Description: | User will explore the diet plan to be followed that is most suitable to maintain their body weight |
| Trigger: | User will tap on the “Diet to maintain weight” section |
| Level: | Medium |
| Preconditions: | User must already be in the section of “Diet Plans” |
| Postconditions: | User will be provided with complete instructions and routines that are to be followed in order to maintain his body weight |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the Diet plans section. 2. From Diet Plan screen, user will tap on Meal Plans section from the list. 3. From meal plans screen, user will tap on diet to maintain weight section provided in the list. 4. User will be able to see meal plans of the selected category. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | People, registered as the users of the system will be allowed to use this feature (no admins allowed) |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 24: Explore Diet for Overweight (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-21 |
| Use Case Name: | Explore Diet for Overweight |
| Actors: | Primary Actor: User |
| Description: | User will explore the diet plan to be followed that is most suitable for people of overweight body type |
| Trigger: | User will tap on the “Diet for overweight” section |
| Level: | Medium |
| Preconditions: | User must already be in the section of “Diet Plans” |
| Postconditions: | User will be provided with complete instructions and routines that are to be followed in order to overcome his overweight |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on the Diet plans section. 2. From Diet Plan screen, user will tap on Meal Plans section from the list. 3. From meal plans screen, user will tap on diet for overweight section provided in the list. 4. User will be able to see meal plans of the selected category. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | People, registered as the users of the system will be allowed to use this feature (no admins allowed) |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 25: Explore Food Product (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-22 |
| Use Case Name: | Explore Food Product |
| Actors: | Primary Actor: User |
| Description: | User can explore food products from his smartphone, using our system |
| Trigger: | User will tap on the button in the center bottom from the home screen |
| Level: | High |
| Preconditions: | User should have logged in. |
| Postconditions: | User will be able to explore food products |
| Normal Flow: | <ol style="list-style-type: none"> 1. From the home screen, user will tap on the button found on the bottom center of the screen |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. User have not granted camera or storage permission. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is able to search restaurants. |

Table 26: Identify Food using Smartphone's Camera (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-23 |
| Use Case Name: | Identify food item using smartphone's camera. |
| Actors: | Primary Actor: User |
| Description: | User will be able to identify food item by capturing food image |
| Trigger: | User will click on capture button at the bottom of camera screen to identify food item. |
| Level: | High |
| Preconditions: | User should have functional camera in smartphone. |
| Postconditions: | User will be able to identify food item using image from his camera |
| Normal Flow: | <ol style="list-style-type: none"> 1. From the home screen, user will tap on the camera button on the bottom center of the screen 2. User will be asked to grant camera access permission, if not already granted 3. User will place the food product in the vision of his camera where that item is clearly visible. 4. Tap on the capture button that can be found on the bottom center of the camera screen. 5. Item in that image will be scanned and processed by the system, and the results will be displayed to the user. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. User has not approved camera access permission to the user. User does not have working camera in smartphone. User is unable to take image. |
| Business Rules: | N/A |
| Assumptions: | User has working camera in smartphone and is able to scan and search food item. |

Table 27: Identify Food using Image from smartphone gallery (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-24 |
| Use Case Name: | Identify food using image from smartphone gallery. |
| Actors: | Primary Actor: User |
| Description: | User will be able to search food through an image from smartphone's gallery. |
| Trigger: | User will select and upload an image from his smartphone gallery to search the food item present in that image. |
| Level: | High |
| Preconditions: | Users have to select right image to search food item |
| Postconditions: | User will be able to successfully identify the food item in that image |
| Normal Flow: | <ol style="list-style-type: none"> 1. From the home screen, user will tap on the camera button on the bottom center of the screen 2. User will be asked to grant camera access permission, if not already granted 3. User will tap on the gallery icon that can be found on the bottom left corner of the camera screen. 4. User will be asked to grant storage access permission, if not already granted. 5. User will have to select a single appropriate image file from his smartphone gallery that is now opened in front of him. 6. The finally selected image will be asked to confirm for upload, by the user, to the system. 7. Item in that image will be scanned and processed by the system, and the results will be displayed to the user. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. User does not have proper image to search food item. |
| Business Rules: | N/A |
| Assumptions: | User has proper image to search food item and is able to see food item after selecting image and searching that. |

Table 28: View Food Details (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-25 |
| Use Case Name: | View Food details. |
| Actors: | Primary Actor: User |
| Description: | User will be able to see food details after searching food items. |
| Trigger: | User will search food item by any mean and will be able to see its details. |
| Level: | High |
| Preconditions: | User should have searched food item to see its details. |
| Postconditions: | User will successfully be able to view the details of the searched food item. |
| Normal Flow: | <ol style="list-style-type: none"> 1. User will search the food item, by the any of the process illustrated in use case 23 or use case 24. 2. User will be directed to the corresponding screen containing complete details of the searched food item. 3. These details include: <ul style="list-style-type: none"> • Name of food • Image of food • Types of food • Ingredients of food • Recipe of food |
| Alternative Flows: | N/A |
| Exceptions: | User is not able to search food due to bad internet connection. User does not have a functional camera to search food. User does not have clear image of food item in his gallery to search. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is able to see food item details. |

Table 29: Open Side Drawer (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-26 |
| Use Case Name: | Open Side Drawer |
| Actors: | Primary Actor: User |
| Description: | User can open side drawer to access features in it. |
| Trigger: | User will tap on the hamburger button on the home screen. |
| Level: | High |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will be successful in opening the side drawer of the application. |
| Normal Flow: | <ol style="list-style-type: none">From the home screen, user will tap on the hamburger icon button, that can be found on the top right corner of the home screen.Side drawer will be animated open from left panel of the screen. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 30: Changing Application Language (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-27 |
| Use Case Name: | Open Side Drawer |
| Actors: | Primary Actor: User |
| Description: | User can change application's language. |
| Trigger: | User will tap on "Select Language" option. |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will be able to change application's default language. |
| Normal Flow: | <ol style="list-style-type: none">From the home screen, user will tap on the hamburger icon button, that can be found on the top right corner of the home screen.Side drawer will be animated open from left panel of the screen.From the side drawer, tap on the Select Language option.User will be redirected to the corresponding screen, from where he can choose any of the 5 available languages. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 31: Changing Profile Settings (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-28 |
| Use Case Name: | Change profile settings |
| Actors: | Primary Actor: User |
| Description: | User can change profile settings according to his desire. |
| Trigger: | User will tap on “Profile” in Drawer Menu from home screen to change profile settings. |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will be able to change profile settings according to his desire. |
| Normal Flow: | <ol style="list-style-type: none"> From home screen, user will tap on Side Drawer Menu icon. From the side drawer, user will tap on “Profile” option from the list of options. User will be directed to the profile settings screen. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 32: Change Username (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-29 |
| Use Case Name: | Change Username. |
| Actors: | Primary Actor: User |
| Description: | User can change his username. |
| Trigger: | User will go to profile settings and then can change his username. |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will successfully be able to change his username. |
| Normal Flow: | <ol style="list-style-type: none"> From the home screen, user will tap on Side Drawer icon. From the opened side drawer, user will tap on “Profile” from the list of options. From the redirected screen, user will locate and tap on the username field. User will erase his old username and type in a new one. |
| Alternative Flows: | N/A |
| Exceptions: | <ol style="list-style-type: none"> User does not have active internet connection. Username field is left empty. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 33: Change Password (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-30 |
| Use Case Name: | Change Password. |
| Actors: | Primary Actor: User |
| Description: | User can change his account password. |
| Trigger: | User will go to profile Settings and then can change account's password. |
| Level: | High |
| Preconditions: | User should remember old password. |
| Postconditions: | User will successfully change into a new password. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From the home screen, user will tap on Side Drawer icon. 2. From the opened side drawer, user will tap on "Profile" from the list of options. 3. From the redirected screen, user will locate and tap on the change password option. 4. User will be prompted to enter his old password first. 5. After successfully entering old password, user can enter and save his new password. |
| Alternative Flows: | N/A |
| Exceptions: | <ol style="list-style-type: none"> 1. Old password is incorrect. 2. New password and confirm new password do not match. |
| Business Rules: | N/A |
| Assumptions: | User have logged in and can change password successfully. |

Table 34: Delete Account (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-31 |
| Use Case Name: | Delete account |
| Actors: | Primary Actor: User |
| Description: | User can also delete his account. |
| Trigger: | User will go to profile settings and then tap on delete account. |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will have his account removed from the system. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From the home screen, user will tap on Side Drawer icon. 2. From the opened side drawer, user will tap on “Profile” from the list of options. 3. From the redirected screen, user will locate and tap on the delete account option. 4. User will be prompted to confirm his password first. 5. After successfully confirming password, user can now successfully unregister his account. |
| Alternative Flows: | N/A |
| Exceptions: | Account could not be deleted due to internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 35: Change Application Settings (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-32 |
| Use Case Name: | Change application settings. |
| Actors: | Primary Actor: User |
| Description: | User can change settings of the application. |
| Trigger: | User will click on settings from the side drawer. |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will be able to change settings of the application. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From home screen, user will tap on Side Drawer Menu icon. 2. From the side drawer, user will tap on “Settings” option from the list of options. 3. User will be directed to the application settings screen. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 36: Toggle Notifications (Mobile Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-33 |
| Use Case Name: | Toggle notifications |
| Actors: | Primary Actor: User |
| Description: | User can switch on and off notifications. |
| Trigger: | User will go to settings and then can change his notification status. |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will successfully be able to change his notification setting. |
| Normal Flow: | <ol style="list-style-type: none"> From the home screen, user will tap on Side Drawer icon. From the opened side drawer, user will tap on “Settings” from the list of options. From the redirected screen, user will locate and tap on the Notification Toggle switch button. User will successfully toggle between switching on and off his application notifications. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 37: Toggle Vibration (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-34 |
| Use Case Name: | Toggle vibration |
| Actors: | Primary Actor: User |
| Description: | User can switch on and off vibration. |
| Trigger: | User will go to settings and then can change his vibration status. |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will successfully be able to change his vibration setting. |
| Normal Flow: | <ol style="list-style-type: none"> From the home screen, user will tap on Side Drawer icon. From the opened side drawer, user will tap on “Settings” from the list of options. From the redirected screen, user will locate and tap on the Vibration Toggle switch button. User will successfully toggle between switching on and off his application vibration. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 38: Change Ringtone (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-35 |
| Use Case Name: | Change Ringtone |
| Actors: | Primary Actor: User |
| Description: | User can select ringtone of his own choice. |
| Trigger: | User will go to settings and then can change his notification ringtone. |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will successfully be able to change his notification ringtone. |
| Normal Flow: | <ol style="list-style-type: none"> From the home screen, user will tap on Side Drawer icon. From the opened side drawer, user will tap on “Settings” from the list of options. From the redirected screen, user will locate and tap on the Change Notification Ringtone option. User will select his favorite ringtone from the list and tap on apply. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 39: View Suggested Restaurants (Mobile Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-36 |
| Use Case Name: | View suggested restaurants. |
| Actors: | Primary Actor: User |
| Description: | User can view restaurants suggested by the system, based on other users’ opinions. |
| Trigger: | User will tap on view suggestions option from side drawer. |
| Level: | Medium |
| Preconditions: | User should be logged into the application. |
| Postconditions: | User will successfully view restaurants suggested by the system. |
| Normal Flow: | <ol style="list-style-type: none"> From the home screen, user will tap on Side Drawer icon. From the opened side drawer, user will tap on “View Suggestions” from the list of options. User will be redirected to the corresponding screen where he can view the restaurants suggested by the system. |
| Alternative Flows: | N/A |
| Exceptions: | User does not have active internet connection. |
| Business Rules: | N/A |
| Assumptions: | User has active internet connection and is on the home screen. |

Table 40: Login (Web Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-37 |
| Use Case Name: | Login |
| Actors: | Primary Actor: Admin |
| Description: | Admin will be able to Login into the system |
| Trigger: | Admin will open web application and fill in credentials to login. |
| Level: | High |
| Preconditions: | N/A |
| Postconditions: | Admin will be able to successfully login into the system |
| Normal Flow: | <ol style="list-style-type: none"> 1. Admin will open web application. 2. Admin will have to enter email and password. 3. Admin will click on login button 4. Admin will be successfully logged in the system |
| Alternative Flows: | N/A |
| Exceptions: | If Admin doesn't have an active internet connection, connection error will be prompted. If Admin enter wrong credentials, he'll be prompted. |
| Business Rules: | N/A |
| Assumptions: | Admin has an active internet connection. |

Table 41: Logout (Web Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-38 |
| Use Case Name: | Logout |
| Actors: | Primary Actor: Admin |
| Description: | Admin will be able to Logout into the system |
| Trigger: | Admin will click on the logout button from the home screen. |
| Level: | High |
| Preconditions: | N/A |
| Postconditions: | Admin will be able to successfully exit from the system |
| Normal Flow: | <ol style="list-style-type: none"> 1. From the home screen, admin will click on the logout button. 2. Admin will be prompted to confirm this activity 3. Upon confirmation, admin will successfully logout from the system |
| Alternative Flows: | N/A |
| Exceptions: | If Admin doesn't have an active internet connection, connection error will be prompted. |
| Business Rules: | N/A |
| Assumptions: | Admin has an active internet connection. |

Table 42: Change Profile Settings (Web Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-39 |
| Use Case Name: | Change profile settings |
| Actors: | Primary Actor: Admin |
| Description: | Admin can change profile settings according to his preferences. |
| Trigger: | Admin will click on settings icon to change settings. |
| Level: | Medium |
| Preconditions: | Admin must be logged in to the application. |
| Postconditions: | Admin will be able to change settings according to his desire. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From the home screen, admin will click on Side Menu icon. 2. Admin will click on “Settings” option from Side menu options 3. Admin will be able to change the settings from the opened page. |
| Alternative Flows: | N/A |
| Exceptions: | Admin doesn't have an active internet connection. |
| Business Rules: | N/A |
| Assumptions: | Admin has an active internet connection. |

Table 43: Change Username (Web Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-40 |
| Use Case Name: | Change username |
| Actors: | Primary Actor: Admin |
| Description: | Admin can change his username |
| Trigger: | Admin can click on settings and then select account settings. Then admin can change his username. |
| Level: | Medium |
| Preconditions: | Admin must be logged in already. |
| Postconditions: | Admin will be able to change his username. |
| Normal Flow: | <ol style="list-style-type: none"> 1. From the home screen, admin will click on Side Menu icon. 2. Admin will click on “Settings” option from Side menu options 3. From the settings page, admin will click on the username field 4. Admin will erase old username and type-in the new one and will click on apply button |
| Alternative Flows: | N/A |
| Exceptions: | Admin left out username field empty. |
| Business Rules: | N/A |
| Assumptions: | Admin has an active internet connection. |

Table 44: Change Password (Web Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-41 |
| Use Case Name: | Change Password. |
| Actors: | Primary Actor: Admin |
| Description: | Admin can change password of their account. |
| Trigger: | Admin will go to Settings and then can change his account's password. |
| Level: | High |
| Preconditions: | <ul style="list-style-type: none"> 1. Admin should have logged in. 2. Admin should remember old password. |
| Postconditions: | Admin will be able to change his account password. |
| Normal Flow: | <ul style="list-style-type: none"> 1. From the home screen, admin will click on Side Menu icon. 2. Admin will click on "Settings" option from Side menu options 3. From settings page, admin will select change password option 4. Admin will be prompted to confirm his old password first 5. After confirming old password, admin can type-in the new password and click on apply button |
| Alternative Flows: | N/A |
| Exceptions: | <ul style="list-style-type: none"> 1. Current password is incorrect. 2. New password and confirm new password do not match. |
| Business Rules: | N/A |
| Assumptions: | Admin has an active internet connection. |

Table 45: Add New Data (Web Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-42 |
| Use Case Name: | Add new data |
| Actors: | Primary Actor: Admin |
| Description: | Admin can add new data into the database to be used by the user |
| Trigger: | Admin will click on the "Add Data" button |
| Level: | High |
| Preconditions: | Admin must be logged in the system |
| Postconditions: | Admin will successfully add new data in the system |
| Normal Flow: | <ul style="list-style-type: none"> 1. Admin will click on the "Add Data" button 2. Admin will be directed to the corresponding screen 3. Admin can enter the data in the respective forms and hit enter. |
| Alternative Flows: | N/A |
| Exceptions: | If new data is not available at that moment, the admin cannot enter the data into the database |
| Business Rules: | N/A |
| Assumptions: | Admin has an active internet connection. |

Table 46: View Data (Web Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-43 |
| Use Case Name: | View data |
| Actors: | Primary Actor: Admin |
| Description: | Admin can view data from the database to is used by the system |
| Trigger: | Admin will click on the “View Data” button |
| Level: | High |
| Preconditions: | Admin must be already logged in the system |
| Postconditions: | Admin will successfully view data of the system |
| Normal Flow: | <ol style="list-style-type: none"> 1. Admin will click on the “View Data” button 2. Admin will be directed to the corresponding screen 3. Admin can choose which data he wants to view from the database. 4. The selected data will be displayed in front of the admin |
| Alternative Flows: | N/A |
| Exceptions: | If no data is already entered into the database, the admin cannot view any data |
| Business Rules: | N/A |
| Assumptions: | Admin has an active internet connection. |

Table 47: Update Data (Web Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-44 |
| Use Case Name: | Update data |
| Actors: | Primary Actor: Admin |
| Description: | Admin can update the existing data in the database to is used by the system |
| Trigger: | Admin will click on the “Update Data” button |
| Level: | High |
| Preconditions: | Admin must be already logged in the system |
| Postconditions: | Admin will successfully update data of the system |
| Normal Flow: | <ol style="list-style-type: none"> 1. Admin will click on the “Update Data” button 2. Admin will be directed to the corresponding screen 3. Admin can choose which data he wants to update in the database. 4. The selected data will be displayed in front of the admin 5. Admin will enter new data in place of the existing data 6. Old data values will be successfully replaced |
| Alternative Flows: | N/A |
| Exceptions: | If no data is already entered into the database, the admin cannot update any data |
| Business Rules: | N/A |
| Assumptions: | N/A |

Table 48: Delete Data (Web Application)

| | |
|---------------------------|--|
| Use Case ID: | UC-45 |
| Use Case Name: | Delete data |
| Actors: | Primary Actor: Admin |
| Description: | Admin can delete the existing data from the database to is used by the system |
| Trigger: | Admin will click on the “Delete Data” button |
| Level: | High |
| Preconditions: | Admin must be already logged in the system |
| Postconditions: | Admin will successfully delete data from the system |
| Normal Flow: | <ol style="list-style-type: none"> 1. Admin will click on the “Delete Data” button 2. Admin will be directed to the corresponding screen 3. Admin can choose which data he wants to delete from the database. 4. The selected data will be displayed in front of the admin 5. Admin will select the amount of data he wants to delete 6. Selected data values will be successfully deleted from the database |
| Alternative Flows: | N/A |
| Exceptions: | If no data is already entered into the database, the admin cannot delete any data |
| Business Rules: | N/A |
| Assumptions: | N/A |

Table 49: View Graphs (Web Application)

| | |
|---------------------------|---|
| Use Case ID: | UC-46 |
| Use Case Name: | View graph |
| Actors: | Primary Actor: Admin |
| Description: | Admin can view observatory data graphs on the home screen |
| Trigger: | Admin will see graphs on the home screen |
| Level: | High |
| Preconditions: | Admin must be logged in the system |
| Postconditions: | Admin will successfully view data graphs |
| Normal Flow: | Directly on the home screen, admin will see different informative graphs, that are based on the data of the system. |
| Alternative Flows: | N/A |
| Exceptions: | If no data is already entered into the database, the admin cannot view any data |
| Business Rules: | N/A |
| Assumptions: | Admin has an active internet connection. |

3.3 Functional Requirements

Following are the identified functional requirements of our system. Tabular representation for each of them is provided here-by:

Table 50: Enter Username

| | |
|----------------------|---|
| Identifier | FR-1 |
| Title | Enter Username for signup |
| Requirement | Users will be able to create account. |
| Source | Development Team |
| Rationale | This requirement will help users to create account. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 51: Enter Email

| | |
|----------------------|---|
| Identifier | FR-2 |
| Title | Enter email |
| Requirement | Users will be able to create account. |
| Source | Development Team |
| Rationale | This requirement will help users to create account. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 52: Enter Password

| | |
|----------------------|---|
| Identifier | FR-3 |
| Title | Enter Password |
| Requirement | Users will be able to create account. |
| Source | Development Team |
| Rationale | This requirement will help users to create account. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 53: Confirm Password

| | |
|----------------------|---|
| Identifier | FR-4 |
| Title | Enter confirm Password |
| Requirement | Users will be able to create account. |
| Source | Development Team |
| Rationale | This requirement will help users to create account. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 54: Tap Sign up Button

| | |
|----------------------|---|
| Identifier | FR-5 |
| Title | Press Signup Button to Sign up. |
| Requirement | Users will be able to create account. |
| Source | Development Team |
| Rationale | This requirement will help users to create account. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 55: Enter Email

| | |
|----------------------|---|
| Identifier | FR-6 |
| Title | Enter Email to Login. |
| Requirement | Users will be able to Login to the application. |
| Source | Development Team |
| Rationale | This requirement will help users to Login. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 56: Enter Password

| | |
|----------------------|---|
| Identifier | FR-7 |
| Title | Enter password to Login |
| Requirement | Users will be able to Login to the application. |
| Source | Development Team |
| Rationale | This requirement will help users to Login. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 57: Tap Login Button

| | |
|----------------------|--|
| Identifier | FR-8 |
| Title | Press Login button to Login. |
| Requirement | Users will be able to Login to the application |
| Source | Development Team |
| Rationale | This requirement will help users to Login. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 58: Tap Logout Button

| | |
|----------------------|---|
| Identifier | FR-9 |
| Title | Logout |
| Requirement | Users will be able to Logout account. |
| Source | Development Team |
| Rationale | This requirement will help users to Logout account. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 59: Confirm Logout

| | |
|----------------------|--|
| Identifier | FR-10 |
| Title | Confirm Logout |
| Requirement | Users will be able to confirm before Logout. |
| Source | Development Team |
| Rationale | This requirement will help users to confirm before logout. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 60: Explore Restaurants

| | |
|----------------------|---|
| Identifier | FR-11 |
| Title | Explore Restaurants |
| Requirement | Users will be able to explore new restaurants. |
| Source | Development Team |
| Rationale | Users will tap on the restaurants section from the home screen. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 61: Filter location

| | |
|----------------------|--|
| Identifier | FR-12 |
| Title | Filter location |
| Requirement | Users will be able to search restaurant by location. |
| Source | Development Team |
| Rationale | Users will select location from restaurant section screen, from the top. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 62: Search restaurant by name

| | |
|----------------------|---|
| Identifier | FR-13 |
| Title | Search restaurant by name |
| Requirement | Users will be able to search restaurants. |
| Source | Development Team |
| Rationale | User will click on search bar on top of restaurant section screen. User will enter restaurant's name to search. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 63: Enter text in search field

| | |
|----------------------|--|
| Identifier | FR-14 |
| Title | Enter text into the search field |
| Requirement | Users will be able to search restaurants. |
| Source | Development Team |
| Rationale | After entering restaurant's name that is to be searched, users will press 'Search' icon to search that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 64: Open a restaurant

| | |
|----------------------|---|
| Identifier | FR-15 |
| Title | Open restaurant |
| Requirement | Users will be able to open restaurants. |
| Source | Development Team |
| Rationale | User will be able to get to know more about selected restaurants. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 65: View restaurant menu

| | |
|----------------------|---|
| Identifier | FR-16 |
| Title | View restaurant menu |
| Requirement | Users will be able to open restaurant menu. |
| Source | Development Team |
| Rationale | User will be able to get to know more about selected restaurant menu. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 66: View restaurant food images

| | |
|----------------------|---|
| Identifier | FR-17 |
| Title | View restaurant food images |
| Requirement | Users will be able to view images of restaurant food. |
| Source | Development Team |
| Rationale | User will be able to view images of that restaurant food. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 67: View restaurant food description

| | |
|----------------------|--|
| Identifier | FR-18 |
| Title | View restaurant food description |
| Requirement | Users will be able to view description of restaurant food. |
| Source | Development Team |
| Rationale | User will be able to view description of that restaurant food. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 68: View restaurant food price

| | |
|----------------------|--|
| Identifier | FR-19 |
| Title | View restaurant food price |
| Requirement | Users will be able to view price of restaurant food. |
| Source | Development Team |
| Rationale | User will be able to view price of that restaurant food. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 69: View restaurant details

| | |
|----------------------|---|
| Identifier | FR-20 |
| Title | View restaurant details |
| Requirement | Users will be able to view details of restaurant. |
| Source | Development Team |
| Rationale | User will be able to view details of that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 70: View restaurant banner

| | |
|----------------------|--|
| Identifier | FR-21 |
| Title | View restaurant banner |
| Requirement | Users will be able to view banner of restaurant. |
| Source | Development Team |
| Rationale | User will be able to view banner of that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 71: View restaurant name

| | |
|----------------------|--|
| Identifier | FR-22 |
| Title | View restaurant name |
| Requirement | Users will be able to view name of restaurant. |
| Source | Development Team |
| Rationale | User will be able to view name of that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 72: View restaurant rating

| | |
|----------------------|--|
| Identifier | FR-23 |
| Title | View restaurant rating |
| Requirement | Users will be able to view rating of restaurant. |
| Source | Development Team |
| Rationale | User will be able to view rating of that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 73: View restaurant opening time

| | |
|----------------------|--|
| Identifier | FR-24 |
| Title | View restaurant opening time |
| Requirement | Users will be able to view opening time of restaurant. |
| Source | Development Team |
| Rationale | User will be able to view opening time of that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 74: View restaurant closing time

| | |
|----------------------|--|
| Identifier | FR-25 |
| Title | View restaurant closing time |
| Requirement | Users will be able to view closing time of restaurant. |
| Source | Development Team |
| Rationale | User will be able to view closing time of that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 75: View restaurant address

| | |
|----------------------|---|
| Identifier | FR-26 |
| Title | View restaurant address |
| Requirement | Users will be able to view address of restaurant. |
| Source | Development Team |
| Rationale | User will be able to view address of that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 76: View restaurant reviews

| | |
|----------------------|---|
| Identifier | FR-27 |
| Title | View restaurant reviews |
| Requirement | Users will be able to view reviews of restaurant. |
| Source | Development Team |
| Rationale | User will be able to view reviews of that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 77: View reviewer name

| | |
|----------------------|--|
| Identifier | FR-28 |
| Title | View reviewer name |
| Requirement | Users will be able to view name of reviewer. |
| Source | Development Team |
| Rationale | User will be able to view name of reviewer. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 78: View reviewer rating

| | |
|----------------------|--|
| Identifier | FR-29 |
| Title | View reviewer rating |
| Requirement | Users will be able to view rating of reviewer. |
| Source | Development Team |
| Rationale | User will be able to view rating of reviewer. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 79: View reviewer feedback

| | |
|----------------------|--|
| Identifier | FR-30 |
| Title | View reviewer rating |
| Requirement | Users will be able to view rating of reviewer. |
| Source | Development Team |
| Rationale | User will be able to view rating of reviewer. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 80: Post Feedback on Restaurant

| | |
|----------------------|---|
| Identifier | FR-31 |
| Title | Give feedbacks to restaurants |
| Requirement | Users will be able to give feedbacks to visited restaurants. |
| Source | Development Team |
| Rationale | If user has visited a restaurant, he can give feedbacks so that they can improve accordingly. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 81: Enter Review Description

| | |
|----------------------|--|
| Identifier | FR-32 |
| Title | Write feedback |
| Requirement | Users will be able to write feedback and post it. |
| Source | Development Team |
| Rationale | Users will press on feedback field where they can write their feedbacks about restaurants. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 82: Give rating to a restaurant

| | |
|----------------------|---|
| Identifier | FR-33 |
| Title | Give ratings to restaurants |
| Requirement | Users will be able to give ratings to restaurant. |
| Source | Development Team |
| Rationale | Users will go to restaurants profile and there they can give stars out of 5 to rate restaurants |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 83: Tap Post button to give feedback

| | |
|----------------------|--|
| Identifier | FR-34 |
| Title | Press post button to give feedback. |
| Requirement | Users will be able to give feedbacks. |
| Source | Development Team |
| Rationale | Users after writing feedback in the text field will click on ‘POST’ button to give feedback. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 84: Go to a restaurant

| | |
|----------------------|---|
| Identifier | FR-35 |
| Title | Go to a restaurant |
| Requirement | Users will be able to go to a restaurant. |
| Source | Development Team |
| Rationale | User will be able to go to that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 85: View restaurant directions

| | |
|----------------------|--|
| Identifier | FR-36 |
| Title | View restaurant directions |
| Requirement | Users will be able to view directions of restaurant. |
| Source | Development Team |
| Rationale | User will be able to view directions of that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 86: Select restaurant route

| | |
|----------------------|---|
| Identifier | FR-37 |
| Title | Select restaurant route |
| Requirement | Users will be able to select route of restaurant. |
| Source | Development Team |
| Rationale | User will be able to select route of that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 87: Follow restaurant route

| | |
|----------------------|---|
| Identifier | FR-38 |
| Title | Follow restaurant route |
| Requirement | Users will be able to follow route of restaurant. |
| Source | Development Team |
| Rationale | User will be able to follow route of that restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 88: Explore Cities

| | |
|----------------------|--|
| Identifier | FR-39 |
| Title | Explore Cities |
| Requirement | Users will be able to explore Cities. |
| Source | Development Team |
| Rationale | Users will tap on the Cities section from the home screen. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 89: Explore Faisalabad

| | |
|----------------------|---|
| Identifier | FR-40 |
| Title | Explore Faisalabad |
| Requirement | Users will be able to explore restaurants in Faisalabad. |
| Source | Development Team |
| Rationale | Users will tap on the Faisalabad section from the cities screen, to explore restaurants in that city. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 90: Explore Islamabad

| | |
|----------------------|--|
| Identifier | FR-41 |
| Title | Explore Islamabad |
| Requirement | Users will be able to explore restaurants in Islamabad. |
| Source | Development Team |
| Rationale | Users will tap on the Islamabad section from the cities screen, to explore restaurants in that city. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 91: Explore Karachi

| | |
|----------------------|--|
| Identifier | FR-42 |
| Title | Explore Karachi |
| Requirement | Users will be able to explore restaurants in Karachi. |
| Source | Development Team |
| Rationale | Users will tap on the Karachi section from the cities screen, to explore restaurants in that city. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 92: Explore Lahore

| | |
|----------------------|---|
| Identifier | FR-43 |
| Title | Explore Lahore |
| Requirement | Users will be able to explore restaurants in Lahore. |
| Source | Development Team |
| Rationale | Users will tap on the Lahore section from the cities screen, to explore restaurants in that city. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 93: Explore Rawalpindi

| | |
|----------------------|---|
| Identifier | FR-44 |
| Title | Explore Rawalpindi |
| Requirement | Users will be able to explore restaurants in Rawalpindi. |
| Source | Development Team |
| Rationale | Users will tap on the Rawalpindi section from the cities screen, to explore restaurants in that city. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 94: Calculate BMI

| | |
|----------------------|---|
| Identifier | FR-45 |
| Title | Calculate BMI |
| Requirement | User will be able to calculate his BMI |
| Source | Development Team |
| Rationale | After calculating the body mass index, user will be provided with the complete details and information about the different slabs of BMI and in which category he currently falls in |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 95: Enter mass in text field

| | |
|----------------------|--|
| Identifier | FR-46 |
| Title | Enter mass in the provided text field |
| Requirement | User will be able to enter his mass in the system |
| Source | Development Team |
| Rationale | User will be able to enter his mass into the provided text field in the selected metric unit for calculating his body mass index |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 96: Enter height in feet

| | |
|----------------------|--|
| Identifier | FR-47 |
| Title | Enter Height (feet) in the provided text field |
| Requirement | User will be able to enter his height in feet |
| Source | Development Team |
| Rationale | User will be able to enter his height in feet into the provided text field for calculating his body mass index |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 97: Enter height in inches

| | |
|----------------------|--|
| Identifier | FR-48 |
| Title | Enter Height (inches) in the provided text field |
| Requirement | User will be able to enter his height in inches |
| Source | Development Team |
| Rationale | User will be able to enter his height in inches into the provided text field for calculating his body mass index |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 98: Tap on calculate BMI button

| | |
|----------------------|--|
| Identifier | FR-49 |
| Title | Click on “Calculate” button |
| Requirement | User will be able to calculate his BMI |
| Source | Development Team |
| Rationale | User have already entered his mass and height in the selected units, now he can click on the “Calculate” button and find out his body mass index |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 99: View BMI details

| | |
|----------------------|---|
| Identifier | FR-50 |
| Title | BMI Details |
| Requirement | User will be provided the details of BMI and what they mean |
| Source | Development Team |
| Rationale | After calculating the body mass index, user will be provided with the complete details and information about the different slabs of BMI and in which category he currently falls in |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 100: View Suggested Diet Plan

| | |
|----------------------|--|
| Identifier | FR-51 |
| Title | BMI Details |
| Requirement | User will be suggested with a diet plan that based on his BMI |
| Source | Development Team |
| Rationale | After calculating the body mass index, user will be suggested with preferred type of diet plan |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 101: Explore diet plans

| | |
|----------------------|---|
| Identifier | FR-52 |
| Title | Explore Diet Plans |
| Requirement | To bring a proper schedule and routine in the user's life |
| Source | Development Team |
| Rationale | To provide the users a proper complete schedule to be adopted for a healthy lifestyle. If the user has some specific fitness goals, he can explore and go through some of the provided categories of diet plans from our system |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 102: Explore diet plan for underweight

| | |
|----------------------|---|
| Identifier | FR-53 |
| Title | Diet plan for under weight |
| Requirement | User will be able to explore the diet plan for underweight people |
| Source | Development Team |
| Rationale | If user's body mass index is small, or he want to gain fats and calories to have his dream body; he can explore this option |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 103: Explore diet plan for normal weight

| | |
|----------------------|--|
| Identifier | FR-54 |
| Title | Diet plan for normal weight |
| Requirement | User will be able to explore the diet plan to remain fit |
| Source | Development Team |
| Rationale | If user's body mass index is normal, or he want to keep his body fit, swift and athletic, he can explore this option |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 104: Explore diet plan for overweight

| | |
|----------------------|---|
| Identifier | FR-55 |
| Title | Diet plan for overweight |
| Requirement | User will be able to explore the diet plan for overweight people |
| Source | Development Team |
| Rationale | If user's body mass index is large, or he want to lose fats and calories to have his dream body; he can explore this option |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 105: Open up a diet plan

| | |
|----------------------|--|
| Identifier | FR-56 |
| Title | Open up a diet plan |
| Requirement | User will be able to open up and view diet plans |
| Source | Development Team |
| Rationale | User will be able to view diet plans details |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 106: Select day of week

| | |
|----------------------|---|
| Identifier | FR-57 |
| Title | Open up a diet plan for a specific day |
| Requirement | User will be able to open up and view diet plans for a specific day |
| Source | Development Team |
| Rationale | User will be able to view diet plans details for a specific day |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 107: View Meal Type

| | |
|----------------------|---|
| Identifier | FR-58 |
| Title | View meal type |
| Requirement | User will be able to view type of meal in this diet plans |
| Source | Development Team |
| Rationale | User will be able to view type of meal in this diet plans on this day |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 108: View Mealtime

| | |
|----------------------|---|
| Identifier | FR-59 |
| Title | View mealtime |
| Requirement | User will be able to view time of meal in this diet plans |
| Source | Development Team |
| Rationale | User will be able to view time of meal in this diet plans on this day |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 109: View Meal Contents

| | |
|----------------------|---|
| Identifier | FR-60 |
| Title | View meal contents |
| Requirement | User will be able to view contents of meal in this diet plans |
| Source | Development Team |
| Rationale | User will be able to view contents of meal in this diet plans on this day |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 110: Explore workout plans

| | |
|----------------------|--|
| Identifier | FR-61 |
| Title | Explore Workout Plans |
| Requirement | To bring a proper schedule and routine for workouts in the user's life |
| Source | Development Team |
| Rationale | To provide the users a proper complete schedule to be adopted for a healthy lifestyle. If the user has some specific fitness goals, he can explore and go through some of the provided categories of workout plans from our system |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 111: Explore home based workout plans

| | |
|----------------------|---|
| Identifier | FR-62 |
| Title | Home based workout plan |
| Requirement | User will be able to explore the home-based workout plans |
| Source | Development Team |
| Rationale | If the user wants to perform light workouts in his home without any prior equipment, he can explore this option |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 112: Explore muscle building workout plans

| | |
|----------------------|--|
| Identifier | FR-63 |
| Title | Muscle building workout plan |
| Requirement | User will be able to explore the muscle building workout plans |
| Source | Development Team |
| Rationale | If the user wants to perform extensive workouts in the gym with proper equipment, he can explore this option |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 113: Explore weight losing workout plans

| | |
|----------------------|--|
| Identifier | FR-64 |
| Title | Weight losing workout plan |
| Requirement | User will be able to explore the weight losing workout plans |
| Source | Development Team |
| Rationale | If the user wants to perform fat burning workouts to shape his body figure, he can explore this option |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 114: Select targeted body part

| | |
|----------------------|---|
| Identifier | FR-65 |
| Title | Open up workouts for a specific body part |
| Requirement | User will be able to open up and view workouts for a specific body part |
| Source | Development Team |
| Rationale | User will be able to view workouts details for a specific body part |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 115: Select day of week

| | |
|----------------------|---|
| Identifier | FR-66 |
| Title | Open up a workout for a specific day |
| Requirement | User will be able to open up and view workouts for a specific day |
| Source | Development Team |
| Rationale | User will be able to view workouts details for a specific day |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 116: View Exercise Name

| | |
|----------------------|---|
| Identifier | FR-67 |
| Title | View exercise name |
| Requirement | User will be able to view name of exercise in this workout plans |
| Source | Development Team |
| Rationale | User will be able to view name of exercise in this workout plan on this day |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 117: View Exercise Visual Demonstration

| | |
|----------------------|---|
| Identifier | FR-68 |
| Title | View exercise visual demonstration |
| Requirement | User will be able to view the visual demonstration of exercise in this workout plan |
| Source | Development Team |
| Rationale | User will be able to view the visual demonstration of exercise in this workout plan on this day |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 118: View Exercise Circuits

| | |
|----------------------|---|
| Identifier | FR-69 |
| Title | View exercise circuits |
| Requirement | User will be able to view circuits of exercise in this workout plans |
| Source | Development Team |
| Rationale | User will be able to view circuits of exercise in this workout plan on this day |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 119: View Exercise Repetitions

| | |
|----------------------|--|
| Identifier | FR-70 |
| Title | View exercise repetitions |
| Requirement | User will be able to view repetitions of exercise in this workout plans |
| Source | Development Team |
| Rationale | User will be able to view repetitions of exercise in this workout plan on this day |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 120: Identify food item

| | |
|----------------------|--|
| Identifier | FR-71 |
| Title | Search food by smartphone's camera |
| Requirement | Users can search food items by scanning them |
| Source | Development Team |
| Rationale | Users can scan food item to search details about that food item. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 121: Open smartphone's camera

| | |
|----------------------|--|
| Identifier | FR-72 |
| Title | Open camera |
| Requirement | Users will be able to open camera and search food item. |
| Source | Development Team |
| Rationale | Users will press camera button from the main screen of application to open camera. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 122: Capture food image

| | |
|----------------------|---|
| Identifier | FR-73 |
| Title | Scan food item |
| Requirement | User will be able to scan food item. |
| Source | Development Team |
| Rationale | Users will scan food item by placing the camera towards that food item. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 123: Select image from gallery

| | |
|----------------------|---|
| Identifier | FR-74 |
| Title | Search food by image |
| Requirement | Users will be able to search food item by image. |
| Source | Development Team |
| Rationale | Users will select image from smartphone's gallery to search food item. Users can see all details about that food item. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 124: Verify image

| | |
|----------------------|---|
| Identifier | FR-75 |
| Title | Checking image |
| Requirement | The system will check image before searching food. |
| Source | Development Team |
| Rationale | The system will check image first if food item present in that image is visible or not. Then it will search that food item and will display all the details about that food item. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 125: Process image for identification

| | |
|----------------------|---|
| Identifier | FR-76 |
| Title | Checking if food item is available or not. |
| Requirement | The system will search first if that food item is present or not. |
| Source | Development Team |
| Rationale | The system will first check if that food item is present or not. If its present, then it will find that food item will display all details about that food item. If it's not available, then it will display "Sorry! Cannot find food item.". |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 126: View food name

| | |
|----------------------|---|
| Identifier | FR-77 |
| Title | See food item name after scanning |
| Requirement | Users will be able to see food item name that is scanned from camera. |
| Source | Development Team |
| Rationale | Users will be able to find out the food name after scanning that food item. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 127: View food image

| | |
|----------------------|--|
| Identifier | FR-78 |
| Title | See food item image after scanning |
| Requirement | Users will be able to see food item image that is scanned from camera. |
| Source | Development Team |
| Rationale | Users will be able to find out the food image after scanning that food item. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 128: View food types

| | |
|----------------------|--|
| Identifier | FR-79 |
| Title | See food item types after scanning |
| Requirement | Users will be able to see food item types that is scanned from camera. |
| Source | Development Team |
| Rationale | Users will be able to find out the food types after scanning that food item. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 129: View food ingredients

| | |
|----------------------|--|
| Identifier | FR-80 |
| Title | See food item ingredients after scanning |
| Requirement | Users will be able to see food item ingredients that is scanned from camera. |
| Source | Development Team |
| Rationale | Users will be able to find out the food ingredients after scanning that food item. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 130: View food recipe

| | |
|----------------------|---|
| Identifier | FR-81 |
| Title | See food item recipe after scanning |
| Requirement | Users will be able to see food item recipe that is scanned from camera. |
| Source | Development Team |
| Rationale | Users will be able to find out the food recipe after scanning that food item. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 131: View food nutrition

| | |
|----------------------|--|
| Identifier | FR-82 |
| Title | See food item nutrition after scanning |
| Requirement | Users will be able to see food item nutrition that is scanned from camera. |
| Source | Development Team |
| Rationale | Users will be able to find out the food nutrition after scanning that food item. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 132: View food recipe

| | |
|----------------------|--|
| Identifier | FR-83 |
| Title | View food recipe |
| Requirement | Users will be able to see recipe of different food items. |
| Source | Development Team |
| Rationale | Users will search for food and can see its recipe to try that at home. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 133: Open Side Drawer

| | |
|----------------------|---|
| Identifier | FR-84 |
| Title | Open Side Drawer |
| Requirement | Users will be able to open the side drawer to access its functionalities. |
| Source | Development Team |
| Rationale | Users will be able to open the side drawer to access its functionalities. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 134: Change Application Language

| | |
|----------------------|--|
| Identifier | FR-85 |
| Title | Change application's Language |
| Requirement | Users can also change application's Language according to their native language so they can use application better. |
| Source | Development Team |
| Rationale | Users can be from different regions. There could be some users who will not be able to understand English. So, they can change language in application settings. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 135: Select English Language

| | |
|----------------------|--|
| Identifier | FR-86 |
| Title | Select English Language |
| Requirement | Users can also change to English Language according to their native language so they can use application better. |
| Source | Development Team |
| Rationale | User can change the applications language to be English. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 136: Select French Language

| | |
|----------------------|---|
| Identifier | FR-87 |
| Title | Select French Language |
| Requirement | Users can also change to French Language according to their native language so they can use application better. |
| Source | Development Team |
| Rationale | User can change the applications language to be French. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 137: Select German Language

| | |
|----------------------|---|
| Identifier | FR-88 |
| Title | Select German Language |
| Requirement | Users can also change to German Language according to their native language so they can use application better. |
| Source | Development Team |
| Rationale | User can change the applications language to be German. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 138: Select Italian Language

| | |
|----------------------|--|
| Identifier | FR-89 |
| Title | Select Italian Language |
| Requirement | Users can also change to Italian Language according to their native language so they can use application better. |
| Source | Development Team |
| Rationale | User can change the applications language to be Italian. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 139: Select Spanish Language

| | |
|----------------------|--|
| Identifier | FR-90 |
| Title | Select Spanish Language |
| Requirement | Users can also change to Spanish Language according to their native language so they can use application better. |
| Source | Development Team |
| Rationale | User can change the applications language to be Spanish. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 140: Apply New Language

| | |
|----------------------|---|
| Identifier | FR-91 |
| Title | Save settings after change language. |
| Requirement | Users will have to save settings after changing language of application. |
| Source | Development Team |
| Rationale | If users select a language by mistake that he does not know, settings for change language will not be saved until user saves it by himself. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 141: Search restaurants by suggestions

| | |
|----------------------|---|
| Identifier | FR-92 |
| Title | Search restaurants by suggestions. |
| Requirement | Users will be able to search restaurants by suggestions or by ratings given by people who have visited that restaurant. |
| Source | Development Team |
| Rationale | Users can tap to side drawer option from home screen. From the side drawer panel, user will tap on view suggested restaurant. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 142: Change Profile Settings

| | |
|----------------------|---|
| Identifier | FR-93 |
| Title | Settings |
| Requirement | Users will be able to change the settings of his profile. |
| Source | Development Team |
| Rationale | Users can change all of the settings from within the settings according to his preferences. These details include: <ul style="list-style-type: none">• Change username• Change account password• Delete account |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 143: Change Username

| | |
|----------------------|---|
| Identifier | FR-94 |
| Title | Change username |
| Requirement | User can change his username |
| Source | Development Team |
| Rationale | User can change his current username from profile settings. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 144: Change Password

| | |
|----------------------|---|
| Identifier | FR-95 |
| Title | Change Password |
| Requirement | Users can change his password |
| Source | Development Team |
| Rationale | User can change his current password from account settings. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 145: Confirm Old Password

| | |
|----------------------|--|
| Identifier | FR-96 |
| Title | Confirm Old Password |
| Requirement | User will have to confirm his old password before proceeding |
| Source | Development Team |
| Rationale | Before changing the current password of his account, the user will have to confirm his old already set password for security reasons, in order to successfully shift to new password |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 146: Enter New Password

| | |
|----------------------|--|
| Identifier | FR-97 |
| Title | Enter New Password |
| Requirement | User can enter new password to set on his account as future log in credentials |
| Source | Development Team |
| Rationale | After successfully entering the correct old password correctly, the admin can now proceed to adding the new password for his account login |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 147: Show/Hide Password

| | |
|----------------------|---|
| Identifier | FR-98 |
| Title | Show/Hide Password |
| Requirement | User can switch between show and hide password |
| Source | Development Team |
| Rationale | For security reasons, when entering the password in the text fields, the user can easily switch between hiding and showing the password in that field via a button. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 148: Apply/Discard New Password

| | |
|----------------------|---|
| Identifier | FR-99 |
| Title | Apply/Discard new password |
| Requirement | User can make the decision about changing password |
| Source | Development Team |
| Rationale | When the user confirms his old password and enters a new password for his account, he can either apply these new settings or he can also discard this new password and revert back to his old password. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 149: Delete Account

| | |
|----------------------|--|
| Identifier | FR-100 |
| Title | Delete Account |
| Requirement | Users will be able to delete account. |
| Source | Development Team |
| Rationale | If users want to delete their account, they will go to settings and can delete their account from there. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Low |

Table 150: Confirm Account Deletion

| | |
|----------------------|---|
| Identifier | FR-101 |
| Title | Confirm delete account |
| Requirement | Users will have to confirm before deleting their account. |
| Source | Development Team |
| Rationale | If any user clicked “Delete account” option by mistake. The account will not be deleted until user confirms it. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 151: Change Application Settings

| | |
|----------------------|--|
| Identifier | FR-102 |
| Title | Settings |
| Requirement | Users will be able to change the settings of application. |
| Source | Development Team |
| Rationale | Users can change all of the settings from within the settings according to his preferences. These details include: <ul style="list-style-type: none">• Toggle Notifications• Toggle Vibration• Change Ringtone |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 152: Turn on Notifications

| | |
|----------------------|--|
| Identifier | FR-103 |
| Title | Turn on Notifications |
| Requirement | Users will be able to turn on notifications for the application |
| Source | Development Team |
| Rationale | User can turn on the notification alerts for the smartphone application. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 153: Turn off Notifications

| | |
|----------------------|---|
| Identifier | FR-104 |
| Title | Turn off Notifications |
| Requirement | Users will be able to turn off notifications for the application |
| Source | Development Team |
| Rationale | User can turn off the notification alerts for the smartphone application. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 154: Turn on Vibrations

| | |
|----------------------|--|
| Identifier | FR-105 |
| Title | Turn on Vibrations |
| Requirement | Users will be able to turn on Vibrations for the application |
| Source | Development Team |
| Rationale | User can turn on the Vibrations alerts for the smartphone application. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 155: Turn off Vibrations

| | |
|----------------------|---|
| Identifier | FR-106 |
| Title | Turn off Vibrations |
| Requirement | Users will be able to turn off Vibrations for the application |
| Source | Development Team |
| Rationale | User can turn off the Vibrations alerts for the smartphone application. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 156: Change Ringtone

| | |
|----------------------|--|
| Identifier | FR-107 |
| Title | Change Ringtone |
| Requirement | Users will be able to change notification ringtone of mobile application. |
| Source | Development Team |
| Rationale | Users will have to change the notification ringtone of the smartphone application. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 157: Choose Ringtone

| | |
|----------------------|--|
| Identifier | FR-108 |
| Title | Choose Ringtone |
| Requirement | Users will be able to choose notification ringtone from the list. |
| Source | Development Team |
| Rationale | Users will have to choose the notification ringtone of the smartphone application. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 158: Apply Ringtone

| | |
|----------------------|--|
| Identifier | FR-109 |
| Title | Apply Ringtone |
| Requirement | Users will be able to apply notification ringtone selected from the list. |
| Source | Development Team |
| Rationale | Users will have to apply the selected notification ringtone on the smartphone application. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 159: Login (Web Application)

| | |
|----------------------|---|
| Identifier | FR-110 |
| Title | Login (Admin) |
| Requirement | Admin will login to the system |
| Source | Development Team |
| Rationale | To use the admin panel, admin login to the system by entering the correct credentials. Admin will enter his username or email, along with the correct password, and click on the login button |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 160: Enter Email (Web Application)

| | |
|----------------------|---|
| Identifier | FR-111 |
| Title | Enter Email to Login. |
| Requirement | Admin will be able to Login to the application. |
| Source | Development Team |
| Rationale | This requirement will help admin to Login. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 161: Enter Password (Web Application)

| | |
|----------------------|---|
| Identifier | FR-112 |
| Title | Enter password to Login |
| Requirement | Admin will be able to Login to the application. |
| Source | Development Team |
| Rationale | This requirement will help admin to Login. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 162: Click Login Button

| | |
|----------------------|--|
| Identifier | FR-113 |
| Title | Click Login button to Login. |
| Requirement | Admin will be able to Login to the application |
| Source | Development Team |
| Rationale | This requirement will help admin to Login. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 163: Log out (Web Application)

| | |
|----------------------|--|
| Identifier | FR-114 |
| Title | Logout (Admin) |
| Requirement | Admin will be logout to the system |
| Source | Development Team |
| Rationale | To end the session of admin panel, admin will log out from the system by clicking on the logout button |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 164: Add data

| | |
|----------------------|---|
| Identifier | FR-115 |
| Title | Add Data |
| Requirement | Admin can add data into the database |
| Source | Development Team |
| Rationale | Admin can add data for the new restaurants and meals. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 165: Add restaurant data

| | |
|----------------------|--|
| Identifier | FR-116 |
| Title | Add restaurant data |
| Requirement | Admin can add restaurant data into the database |
| Source | Development Team |
| Rationale | Admin can add restaurant data for registering new restaurants. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 166: Add food data

| | |
|----------------------|---|
| Identifier | FR-117 |
| Title | Add food data |
| Requirement | Admin can add food data into the database |
| Source | Development Team |
| Rationale | Admin can add food data for registering new food. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 167: View data

| | |
|----------------------|--|
| Identifier | FR-118 |
| Title | View Data |
| Requirement | To show data of the system to the admin |
| Source | Development Team |
| Rationale | Admin can view the data that he wants to see from the database and can make decisions based on it. He can even organize this data. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 168: View restaurant data

| | |
|----------------------|--|
| Identifier | FR-119 |
| Title | Add restaurant data |
| Requirement | Admin can view restaurant data from the database |
| Source | Development Team |
| Rationale | Admin can view restaurant data for already existing restaurants. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 169: View food data

| | |
|----------------------|--|
| Identifier | FR-120 |
| Title | Add food data |
| Requirement | Admin can view food data from the database |
| Source | Development Team |
| Rationale | Admin can view food data for already existing restaurants. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 170: Update data

| | |
|----------------------|--|
| Identifier | FR-121 |
| Title | Update Data |
| Requirement | Admin will be able to successfully update the system's data |
| Source | Development Team |
| Rationale | System's data that has already been added into the database, admin can update that data, and change it according to the business requirements. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 171: Update restaurant data

| | |
|----------------------|--|
| Identifier | FR-122 |
| Title | Update restaurant data |
| Requirement | Admin can update restaurant data from the database |
| Source | Development Team |
| Rationale | Admin can update restaurant data for already existing restaurants. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 172: Update food data

| | |
|----------------------|--|
| Identifier | FR-123 |
| Title | Update food data |
| Requirement | Admin can update food data from the database |
| Source | Development Team |
| Rationale | Admin can update food data for already existing restaurants. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 173: Delete data

| | |
|----------------------|--|
| Identifier | FR-124 |
| Title | Delete Data |
| Requirement | Admin will be able to delete data |
| Source | Development Team |
| Rationale | Admin will be able to successfully remove unwanted data entries from the database for whatever the business reasons are. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 174: Delete restaurant data

| | |
|----------------------|--|
| Identifier | FR-125 |
| Title | Delete restaurant data |
| Requirement | Admin can delete restaurant data from the database |
| Source | Development Team |
| Rationale | Admin can delete restaurant data for already existing restaurants. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 175: Delete food data

| | |
|----------------------|--|
| Identifier | FR-126 |
| Title | Delete food data |
| Requirement | Admin can delete food data from the database |
| Source | Development Team |
| Rationale | Admin can delete food data for already existing restaurants. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | High |

Table 176: Change Username (Web Application)

| | |
|----------------------|--|
| Identifier | FR-127 |
| Title | Change username |
| Requirement | Admin can change his username |
| Source | Development Team |
| Rationale | Admin can change his current username from profile settings. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 177: Change password (Web Application)

| | |
|----------------------|---|
| Identifier | FR-128 |
| Title | Change Password |
| Requirement | Admin can reset his password |
| Source | Development Team |
| Rationale | Admin can change his current password for logging in to his account of admin panel. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 178: Confirm old password (Web Application)

| | |
|----------------------|---|
| Identifier | FR-129 |
| Title | Confirm Old Password |
| Requirement | Admin will have to confirm his old password before proceeding |
| Source | Development Team |
| Rationale | Before changing the current password of his account, the admin will have to confirm his old already set password for security reasons, in order to successfully shift to new password |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 179: Add new password (Web Application)

| | |
|----------------------|--|
| Identifier | FR-130 |
| Title | Add New Password |
| Requirement | Admin can add new password to set on his account as future log in credentials |
| Source | Development Team |
| Rationale | After successfully entering the correct old password correctly, the admin can now proceed to adding the new password for his account login |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 180: Show/hide password (Web Application)

| | |
|----------------------|--|
| Identifier | FR-131 |
| Title | Show/Hide Password |
| Requirement | User can switch between show and hide password |
| Source | Development Team |
| Rationale | For security reasons, when entering the password in the text fields, the admin can easily switch between hiding and showing the password in that field via a button. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

Table 181: Apply/Discard new password (Web Application)

| | |
|----------------------|--|
| Identifier | FR-132 |
| Title | Apply/Discard new password |
| Requirement | Admin can make the decision about changing password |
| Source | Development Team |
| Rationale | When the admin confirms his old password and enters a new password for his account, he can either apply these new settings or he can also discard this new password and revert back to his old password. |
| Business Rule | N/A |
| Dependencies | N/A |
| Priority | Medium |

3.4 Non-Functional Requirements

The non-functional requirements are applied on the whole system when it is finalized. It defines the system attributes such security, reliability performance etc. There are some non-functional requirements that are listed below.

3.4.1 Reliability

Mean Time between Failures (MTBF): The software will be designed efficiently to keep the MTBF to the minimum.

Mean Time to Repair (MTTR): In case if a fatal crash encountered, it'll be dealt with immediately to keep MTTR to the minimum.

3.4.2 Usability

Time: The system is pretty well-designed and user friendly. The user will get used to use every function of system in very short time.

Functionalities: The functionalities of the system are developed in a way that they will be very easy to use. The user will get to know about every functionality really quick.

Task Time: Task completion time is very minimal as the system deals with them efficiently. The user will be very satisfied in regard to the usability of this system.

3.4.3 Performance

Response:

Average time response = 500 milliseconds, maximum response time = 3 seconds.

Throughput:

Throughput of the system is designed to be efficient for the fastest of experiences for the user.

Resource Utilization:

Required Main Memory = 50-80mb, Required Secondary Memory = 19mb.

Capacity:

Initially the capacity of our system will be 8000-10,000 user, but these numbers will be extended as with the passage of time.

Degradation modes:

The system, at each stance, tries to recover from the garbled transmission, in case if any degradation happens.

3.4.4 Security

One of the top shelf attributes that makes the backbone of a system; is its data. The data that your company creates, collects, stores, and exchanges is a valuable asset. Safeguarding it from corruption and unauthorized access by internal or external people protects your company from financial loss, reputation damage, consumer confidence disintegration, and brand erosion. [7]

The system is developed in such a way that all the technicalities of the system will be handled and dealt by the admin only, and no user can take part in any of these activities, keeping the loss risk near to absolute zero.

3.5 External Interface Requirements

Following are the complete details for all of the external interface requirements identified for our system, Food Explorer:

3.5.1 User Interfaces Requirements

Mentioned before here, are some of the identified user interface requirements of our system:

- User friendly GUI.
- Icons in drawer menu with name to improve design.
- Standards for fonts, icons, button labels, images, color schemes, gradient, form inputs,
- Responsive screens so that the system can run on any android device.
- Standard navigation back buttons, standard main screen cards should maintain throughout the system.
- In case of error message, Messages shown through Alert boxes.
- Primary color #F4511E and secondary color #E4E5E7 should maintain throughout the app.
- Language localization to entertain users from different locations.

3.5.2 Software interfaces

Mentioned before here, are some of the identified user interface requirements of our system:

SI-1: Food Explorer

SI-1.1: The Food Explorer shall be connected to firebase and database and store all the data in Cloud Firestore.

SI-1.2: The Food Explorer shall use i18n.next library to use different languages.

SI-1.3: The Food Explorer shall display different restaurants of different cities and food items of that city.

SI-1.4: Food Explorer shall use expo-location to get the user location.

SI-1.5: Food Explorer shall use Google Maps API to get directions of restaurants.

SI-1.6: Food Explorer shall use expo-camera to capture images.

SI-1.7: Food Explorer shall use IamgePicker to select image from smartphone's gallery.

SI-1.8: Food Explorer shall use Flask API for Image classification.

SI-1.9: Food Explorer shall use react-navigation to navigate user to corresponding screen.

SI-1.10: Food Explorer shall simple and similar GUI that shall be user friendly.

3.5.3 Hardware interfaces

Since the system is only for android smartphone devices, it does not need any hardware requirements.

3.5.4 Communications interfaces

CI-1: Food Explorer

CI-1.1: Food Explorer shall use expo-notifications to send notifications on smartphone.

4 Design and Architecture

4.1 Architectural Design

We have selected the Micro-services architecture for the development purpose of our project. The goal behind this intention is to separately implement and deploy the identified chunks of our project in an isolated manner. We found this pattern suitable for us since the nature of food explorer is of such kind that we can break it down into smaller independent sub-parts.

The prepared micro-services architecture diagram of our project, food explorer, is here below:

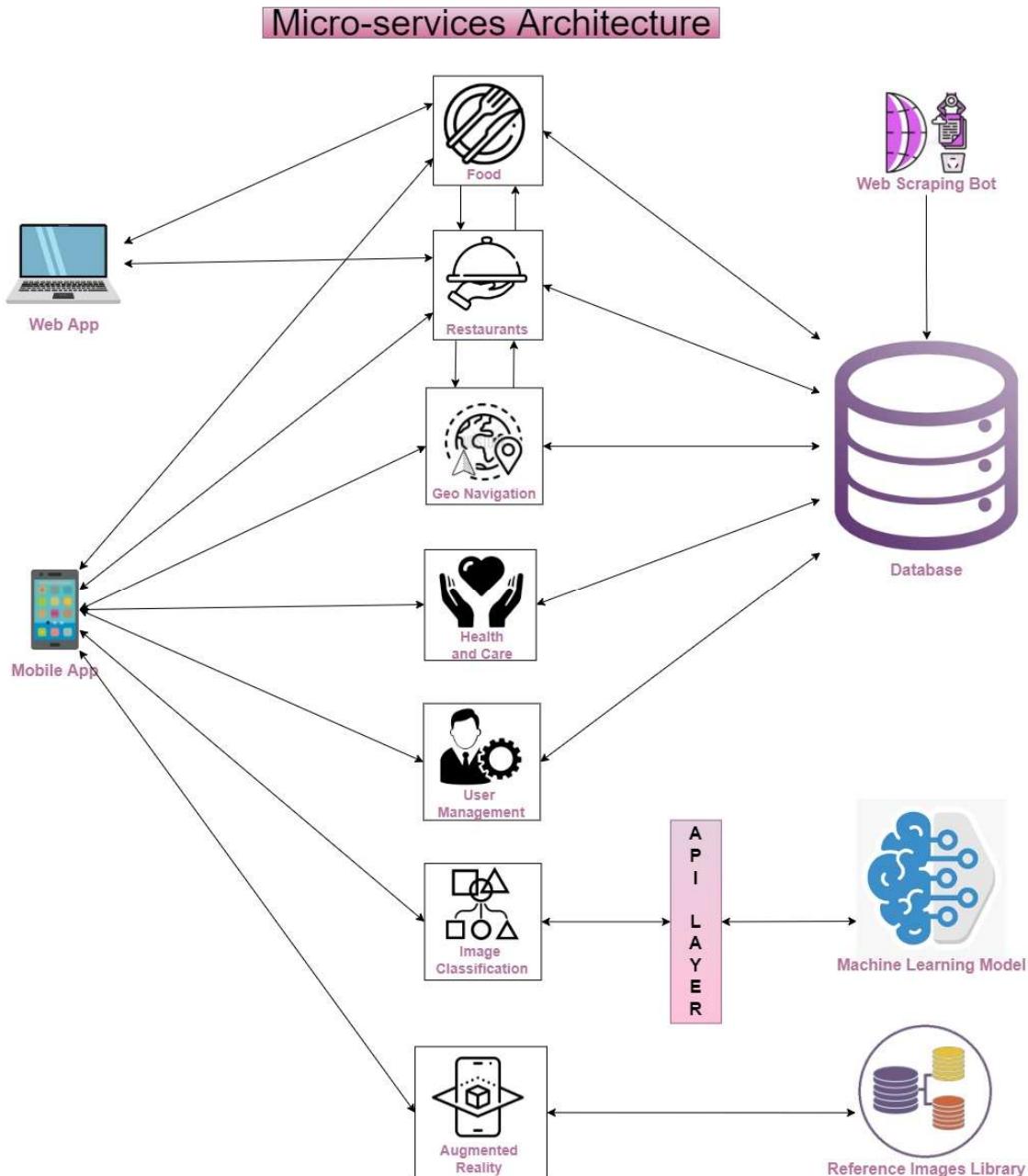


Figure 7: System architecture diagram of food explorer

4.2 Design Models

Following attached are several activity diagrams for some of the major processes of our project, provided for the sake of better process flow representation:

This activity diagram shows all of the functionality that the end-user can perform from the home screen of his smartphone application:

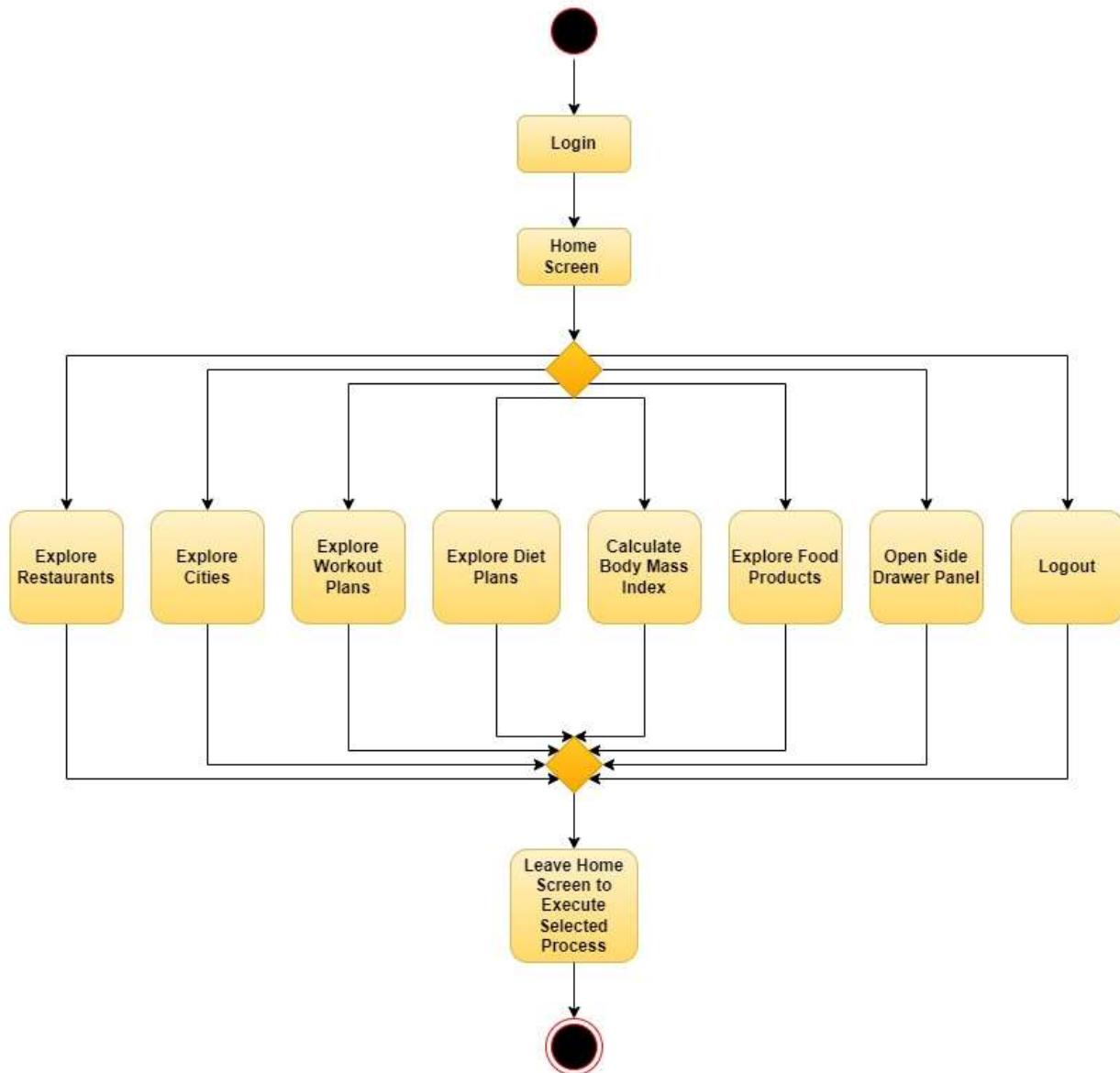


Figure 8: Activity diagram for actions performed from the home screen of mobile application

This activity diagram is the complete explanation for the process of exploring restaurants, which is the first functionality performed by the user, from the home screen of smartphone application:

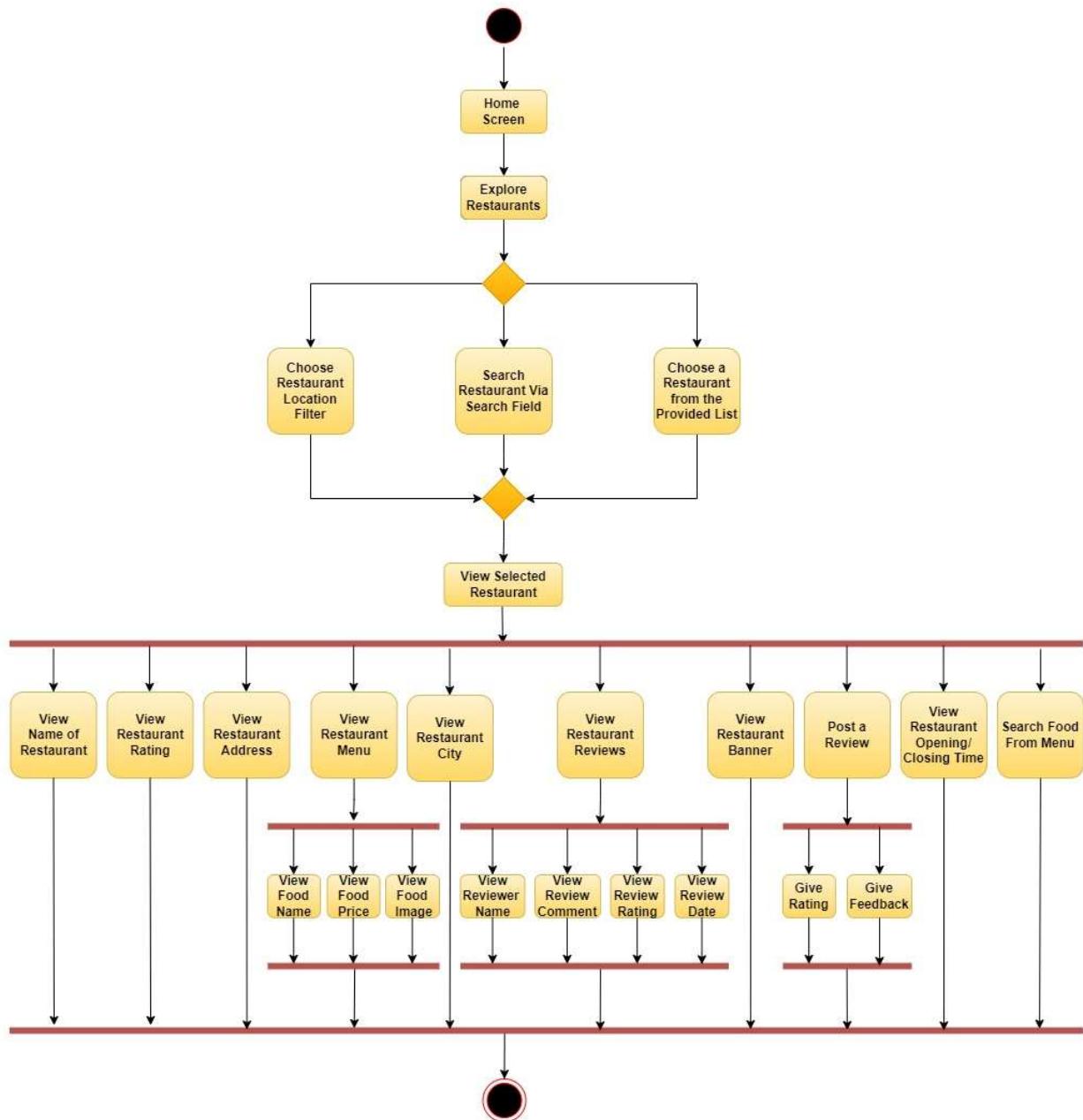


Figure 9: Activity diagram explaining the exploration of restaurants

This activity diagram is the detailed demonstration for the process of exploring food products, explored by the user, from the smartphone application:

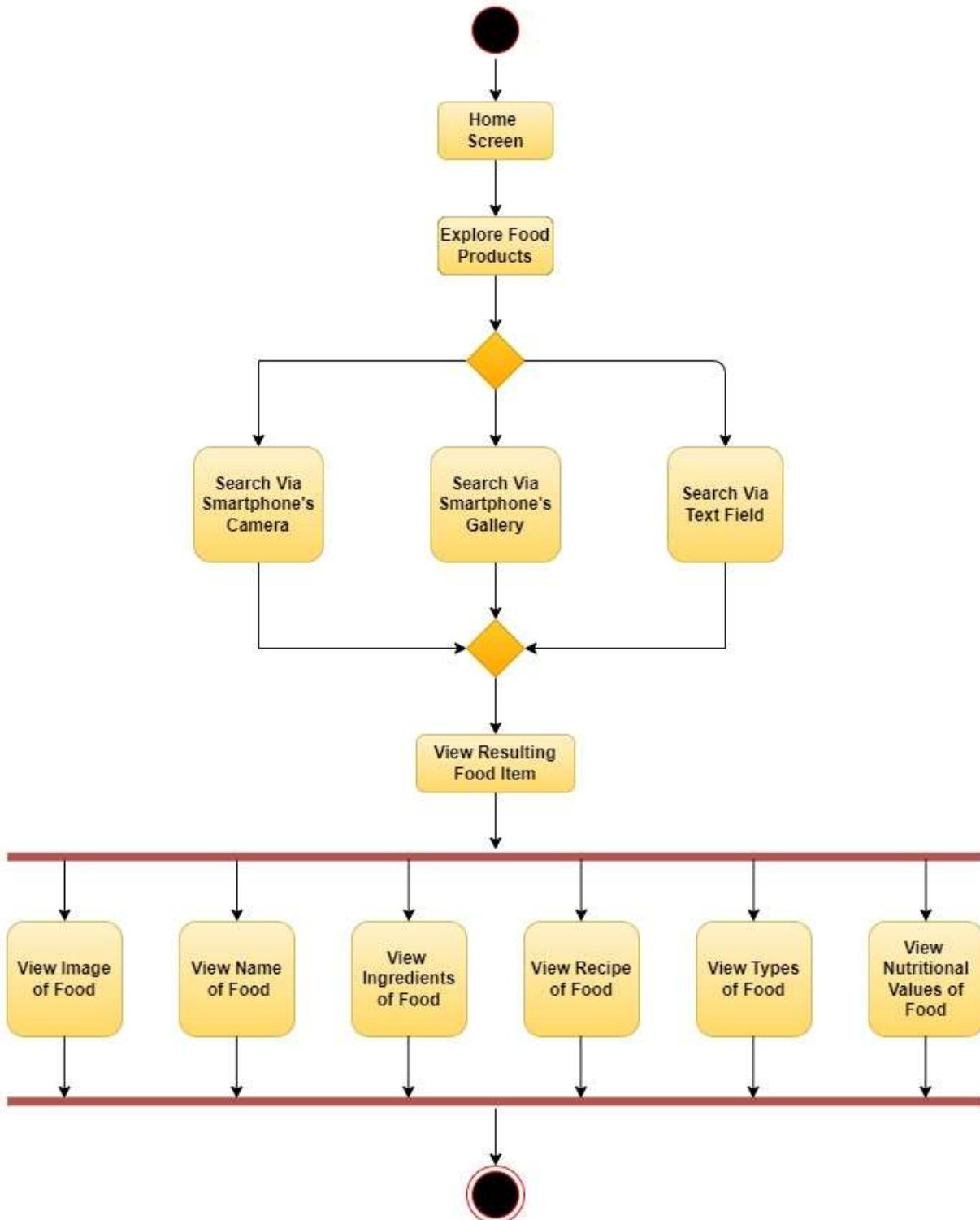


Figure 10: Activity diagram explaining the exploration of food items

This activity diagram illustrates the procedure of exploring workout plans, that are explored by the end-user, from the home screen of the smartphone application:

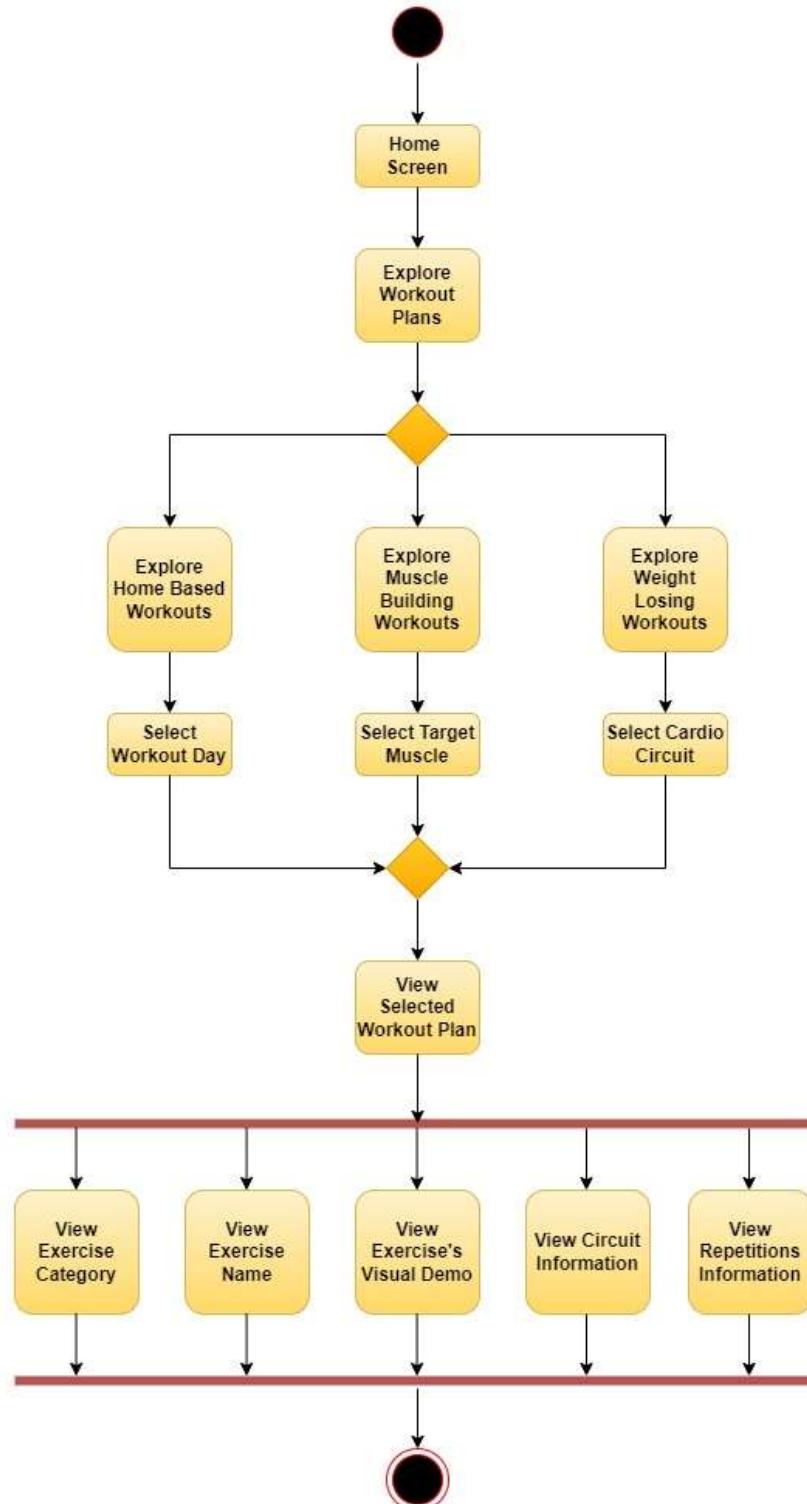


Figure 11: Activity diagram explaining the exploration of workout plans

This activity diagram illustrates the procedure of exploring different diet plans, that are explored by the end-user, from the home screen of the smartphone application:

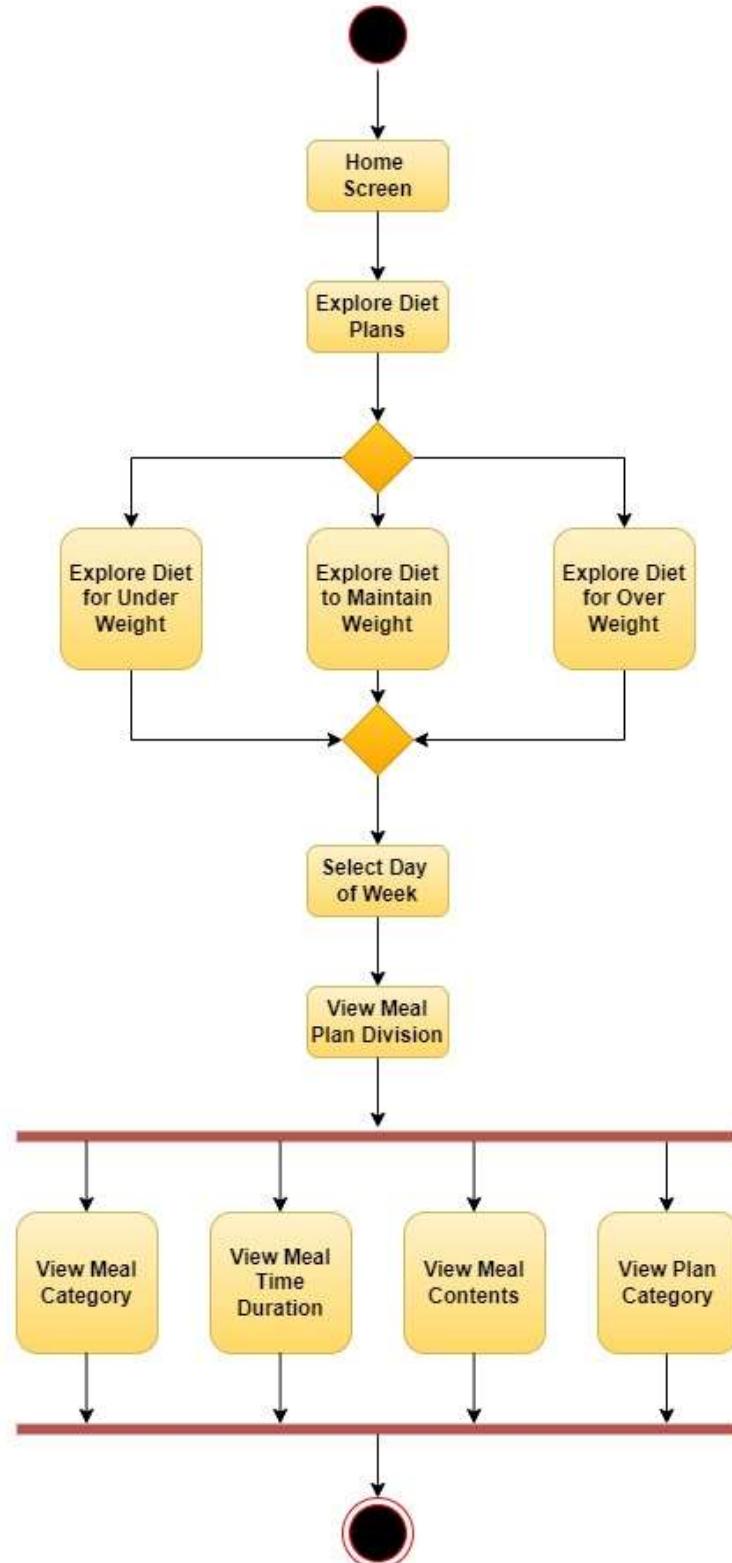


Figure 12: Activity diagram explaining the exploration of diet plans

This activity diagram is the detailed explanation for all of the process and set of actions, that the end-user can perform from the side drawer panel, which is triggered from the home screen of the smartphone application:

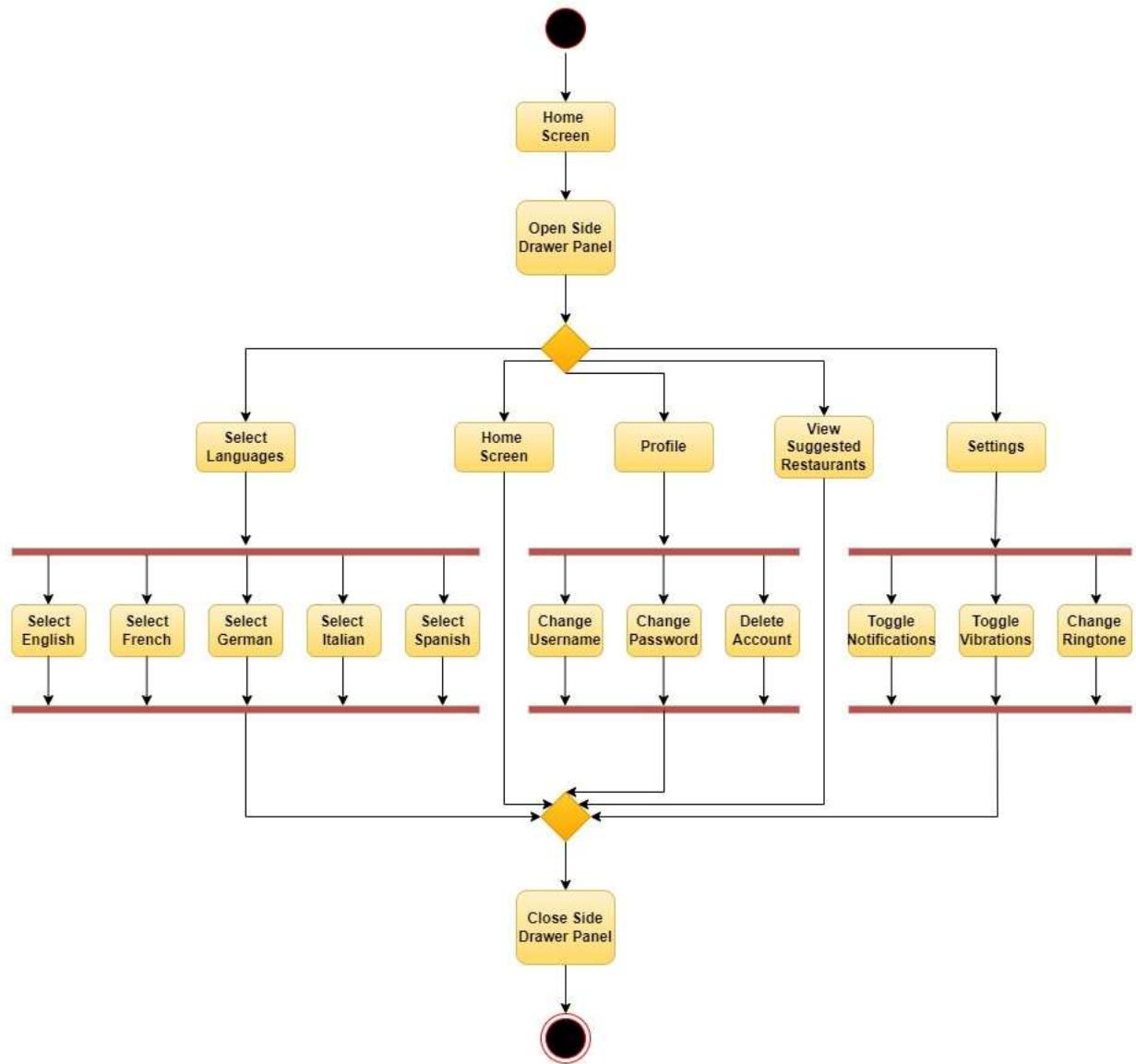


Figure 13: Activity diagram for set of actions performed from the side drawer

This activity diagram demonstrates the total number of options available to the user when he selects the option to explore cities from the home screen of the smartphone application. This section depicts the total number of cities in which our system is currently functional:

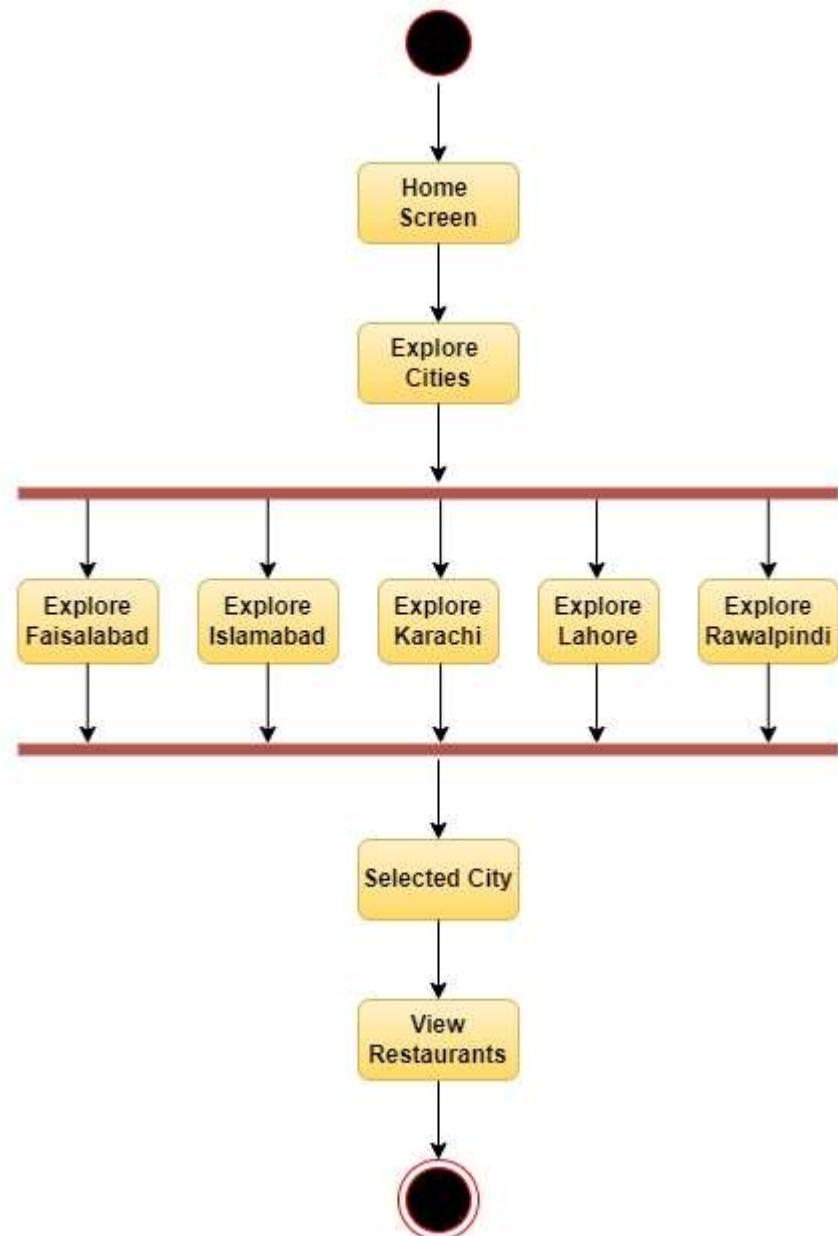


Figure 14: Activity diagram illustrating the exploration of cities

This activity diagram is the depiction for all the roles and responsibilities of admin, that falls under him and are only operable through the system's web application:

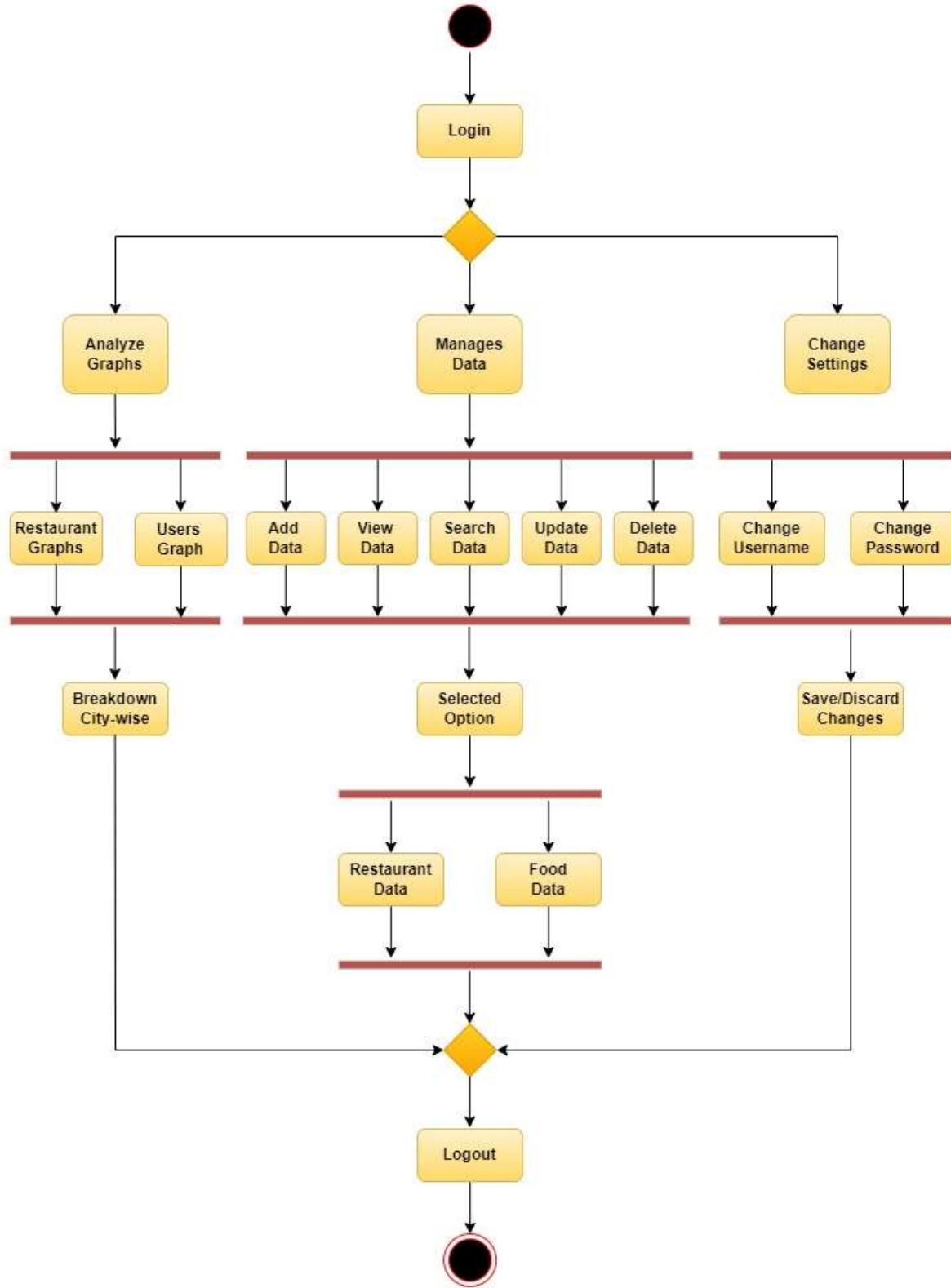


Figure 15: Activity diagram for roles and responsibilities of admin from web app

Since our project is of procedural nature, we have created the Data Flow Diagrams for our system, that are as follow:

This is the first data flow diagram of food explorer and is of Level 0, showing the overall flow of usage against the proposed system:

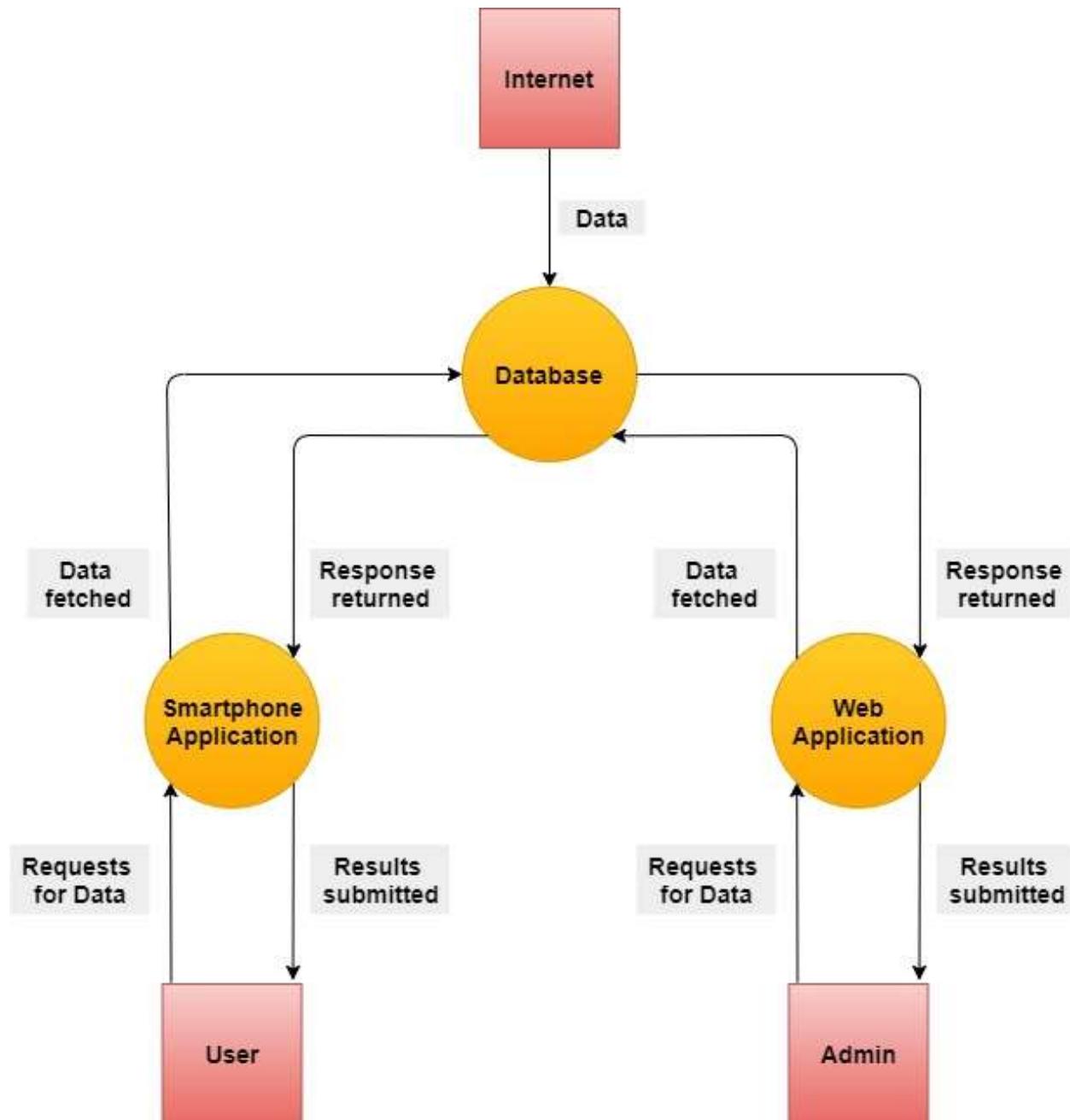


Figure 16: Data Flow Diagram (Level 0)

This is the second data flow diagram of food explorer and is of Level 1, showing the overall interconnectivity, collaboration, and working of major modules of the proposed system:

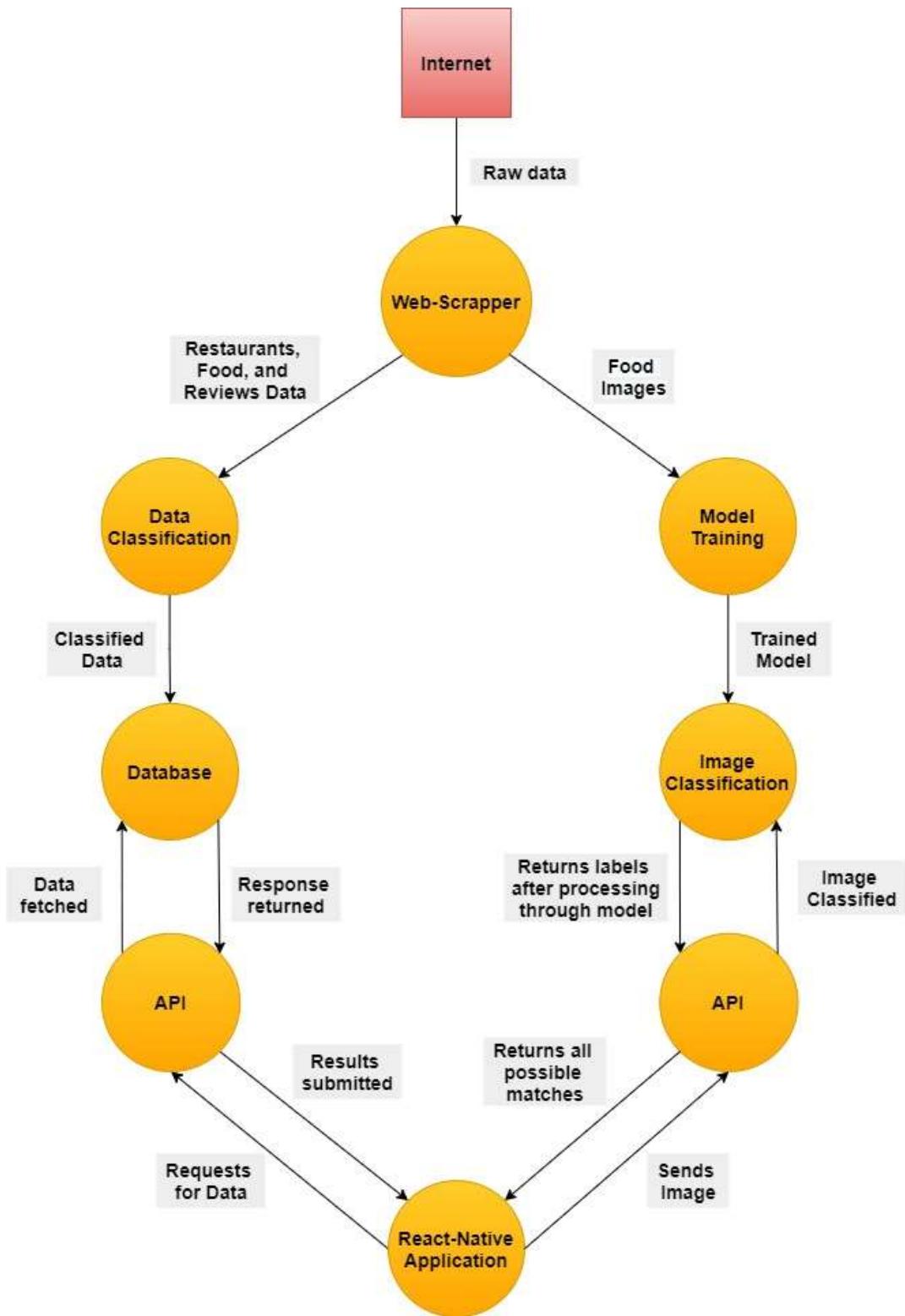


Figure 17: Data Flow Diagram (Level 1)

This is the third data flow diagram of food explorer and is of Level 2, giving detailed explanations for all of the functionalities, processes, and sub-processes of the proposed system:

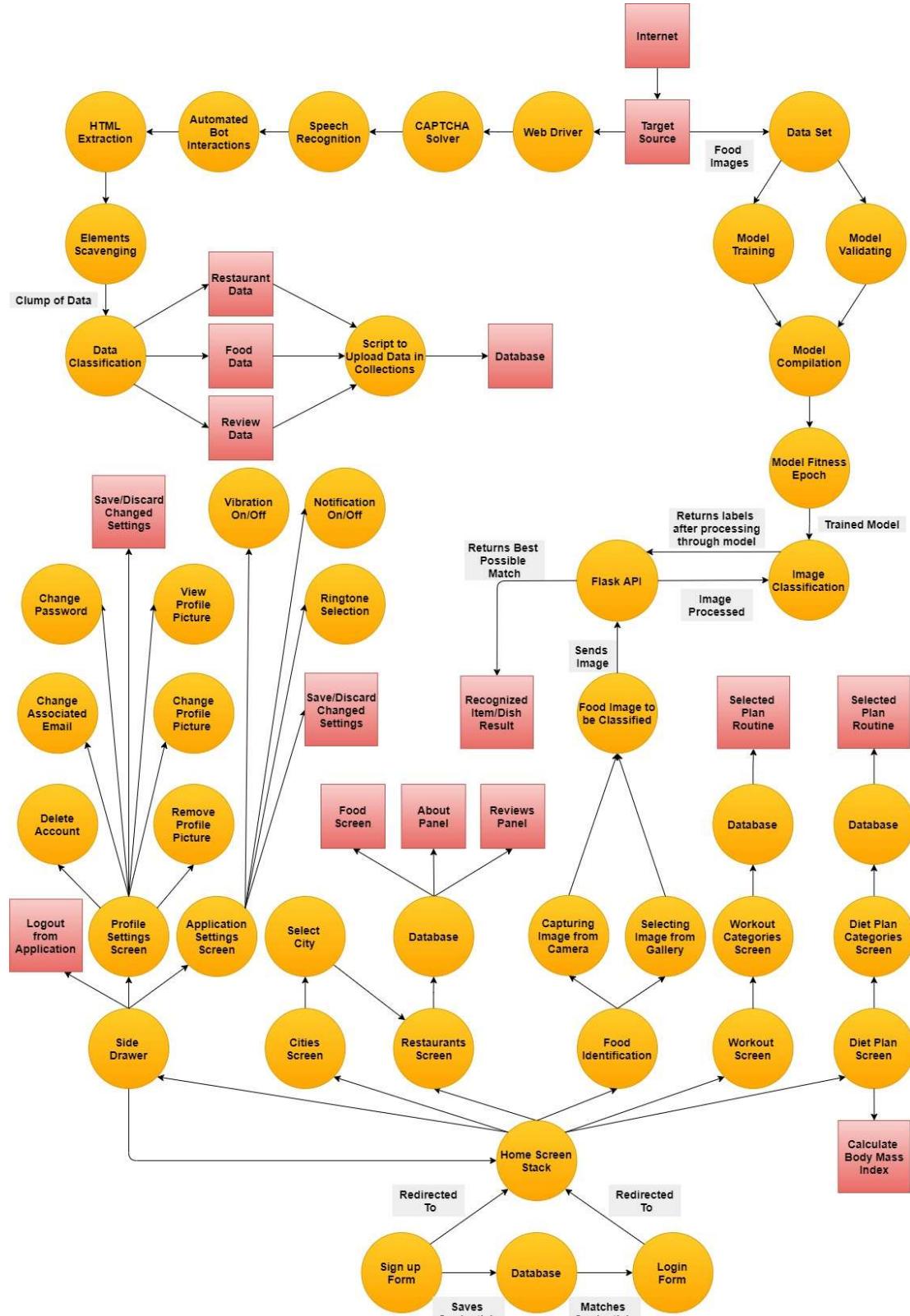


Figure 18: Data Flow Diagram (Level 2)

4.3 Data Design

We made our database using the Google's Firebase (Cloud Firestore). Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud. Like Firebase Realtime Database, it keeps your data in sync across client apps through real-time listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity. Cloud Firestore also offers seamless integration with other Firebase and Google Cloud products, including Cloud Functions. [8]

4.3.1 Data Dictionary

Following is the list of system entities, along with their brief descriptions:

- 1. Admin** The name, email and password will be stored for the only admin of the system.
- 2. Diet Plans** There is total three categories for the diet plans, where each of the plan offers includes a plan ID, name of the plan, days of week, meals time, and what to eat in a meal.
- 3. Food** All food items that are in the menu of a restaurant contains a FoodID, RestaurantID, name of that food item, price of that food item, image of that food item, and discount on that food item (if any available).
- 4. Food Details** Users can view the detailed information against a food item, which includes name of food, image of food, ingredients for preparing that food, recipe process of that food, different types of that food, nutritional facts, and values in that food.
- 5. Restaurant** Data against each of the restaurants contains restaurant name, RestaurantID, restaurant rating, restaurant address, opening time, closing time, longitude, latitude, and banner of that restaurant.
- 6. Review** Review is the community opinion that is given to a restaurant by a user, containing reviewer name, reviewed rating, review date, feedback description posted, and a unique review ID.
- 7. User** Data of the user will hold name of user, email, password, user location, and a unique id for each user of the system.
- 8. Workout Plans** This entity has three different types of workout plans. Each containing plan ID, name of the plan, image of this plan, exercises names in that plan, circuits, repetitions, and visual demonstration of these exercises.

Attached hereby is the formulated Entity Relationship Diagram (ERD) for our project, Food Explorer, according to which we have developed our system:

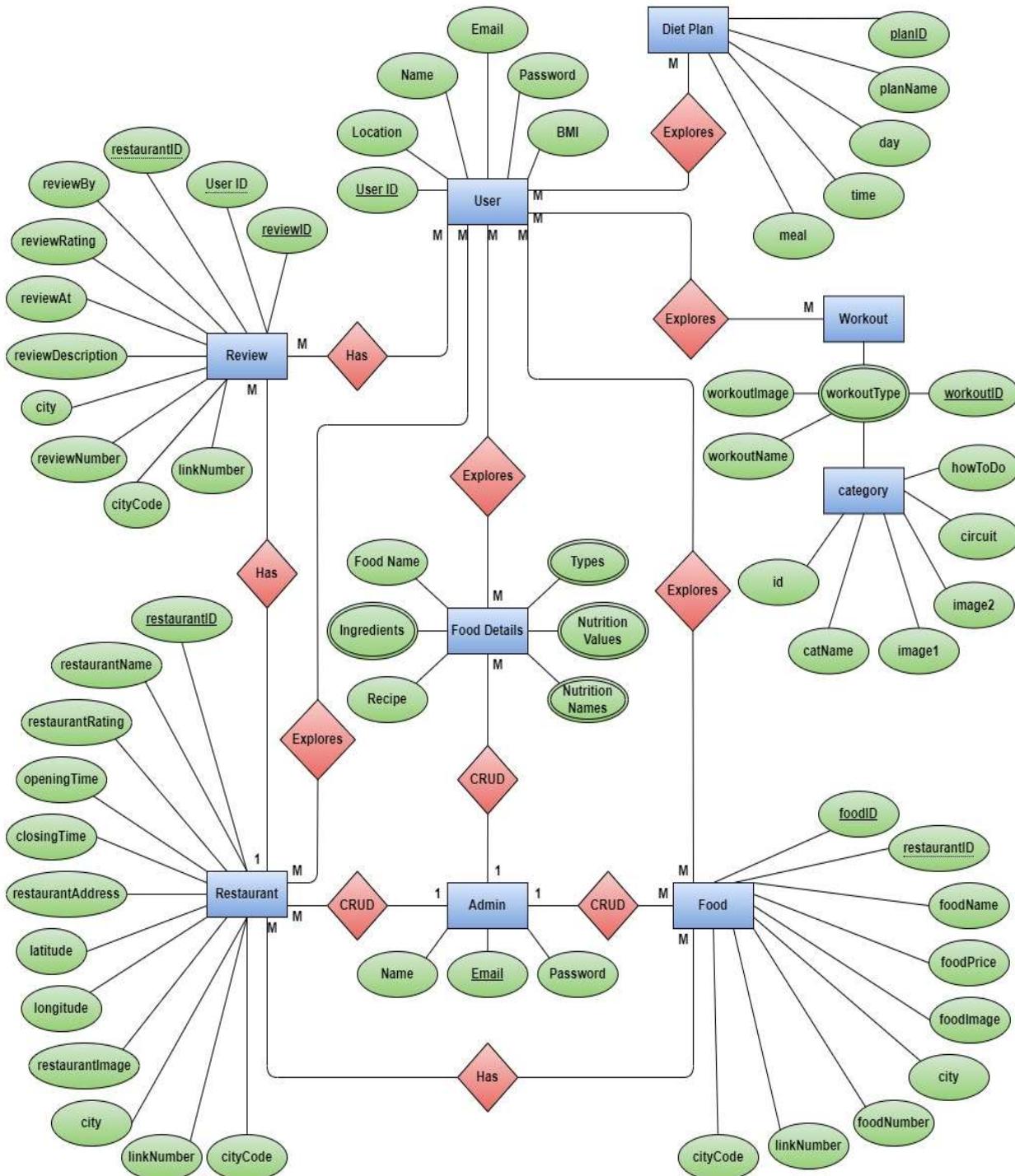


Figure 19: Entity Relation Diagram of Food Explorer

4.4 Human Interface Design

User will open the system and login screen will appear on the screen. User can login here if the account is created or user can go to sign up and can create account before using the system. When logged in, home screen will appear on the screen. On the top of screen, there is header bar from which user can toggle drawer menu. There are different options in drawer which are well defined from their names and icons. User can click on any option and will navigate to the respective screen.

On main screen, there are 4 card buttons i.e., Restaurants, Cities, Workout and Diet plans, and one bottom button which will open the device's camera if the permissions are granted. From restaurants section, user can view restaurants of different cities and food items of particular restaurants. User can also navigate to location of particular restaurant. The route will be displayed on integrated maps. User can also view reviews about particular restaurant and also can post review. User can also view restaurant's details. Cities sections have the different cities, and each city will navigate to restaurants screen which contains restaurants of that city. User can find different types of workouts in workout section. Each type of workout is further divided into categories which are sub divided in exercises. Instructions to perform exercise are also described.

4.4.1 Screen Images

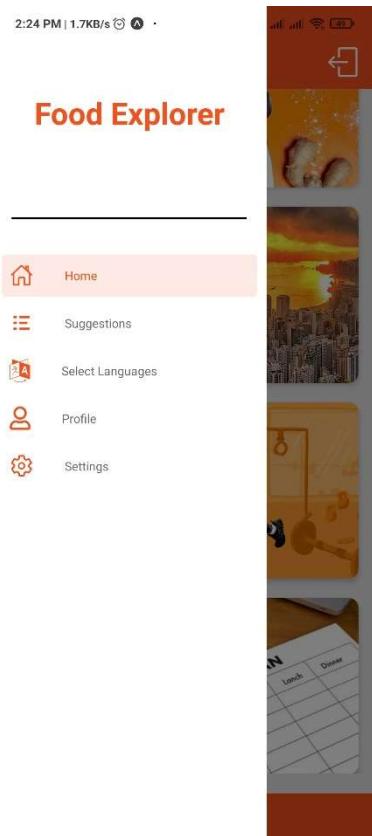
Home Screen



Camera Screen



Side Drawer Screen



4.4.2 Screen Objects and Actions

Dashboard Screen

On Dashboard screen, there are 7 objects on which some actions are performed.

- Menu sidebar: It will toggle the sidebar menu to navigate user to respective sections of the system.
- Logout: It will logout the user from the system.
- Restaurants: It will navigate the user to restaurants section.
- Cities: It will navigate the user to Cities screen.
- Workout: It will navigate the user to workout section of the system.
- Diet Plans: This section will navigate the user to daily diet plans and BMI calculator.
- Camera Scan: This is open the camera to scan or to capture image to search the food item from the gathered dataset.

Camera Screen

On camera screen, there are 3 objects on which actions are performed.

- Back button: This will navigate the user to previous screen.
- Camera Click button: This will capture the image and search the food item from the gathered dataset.
- Gallery Button: This will open the gallery to select the image to search.

Side Drawer Screen

On sidebar screen, there 6 objects to perform some actions.

- Close Drawer: To press on outside of the sidebar, the drawer menu will be closed.
- Home: Home button will navigate the use to dashboard of the system.
- Suggestions: Suggested restaurants of selected city will be displayed according to their ratings.
- Select Languages: This will navigate the user to the screen where language can be change of the system.
- Profile: This will navigate the user to edit profile screen.
- Settings: This will navigate the user to set settings of the system.

5 Implementation

5.1 Algorithm

In this project, we have yet covered multiple domains with implementation. Four of them domains fully map to two whole modules of our project. A natural language description for both of them algorithms is stated as follow, based on the actual development sequence and protocols:

5.1.1 Image Classification

Step 1: Set the path of the data set that is located in the local storage

Step 2: Check the total number of images in that data set

Step 3: Specify the size of the batch so the exact number of images are processed in one iteration

Step 4: Specify the custom image height and width for scaling

Step 5: Specify the split ratio to divide the data set

Step 6: Pass the bigger-ratio part to the training function

Step 7: Pass the other smaller-ratio part to the validation function

Step 8: Set the names of classes according to the desired classification items

Step 9: Set the normalization layer

Step 10: Map this normalization layer onto the training function

Step 11: Ready out the image batches and labels by iterating over the data set

Step 12: Perform data augmentation on the data set

We have used three layers in the data augmentation step

- Preprocessing random flip
- Preprocessing random rotation
- Preprocessing random zoom

Step 13: Define a sequential model that contains multiple processing layers in it

Step 14: Compile the above defined model with the metrics set to be as accuracy

Step 15: Once the model is compiled, specify the number of desired epochs

Step 16: Pass training data set, validation data set, and the specified epoch count to fitness train the model.

Step 17: Once the specified number of epoch cycles are finished processing, save the resulting model asset on your desired local storage path for future use.

5.1.2 Automated Web Scraping AI Bot

Step 1: Specify the web driver executable path

Step 2: Configure the web driver on this path

Step 3: Specify the target URL of the desired city for scraping

Step 4: Pass this URL in the above defined web driver

Step 5: Retrieve the HTML for the web page of the above defined target URL

Step 6: Extract the links of all of the available restaurants of against that city

Step 7: Create and store all of these links – each representing a unique restaurant – in a csv file, using csv.writer

Step 8: Create a new web driver using the same criteria defined above

Step 9: Pick up an entry from the just saved CSV and pass in into this web driver

Step 10: Identify if a Human Detection Captcha is invoked

Step 11: If Step 10 results true; keep on executing next step, otherwise jump to Step 33

Step 12: Make bot switch to the current default content and identify the frame of Captcha dialog box

Step 13: Make bot identify the checkbox of the Captcha solver checkbox

Step 14: Enable bot to click on the identified Captcha solver checkbox

Step 15: Make bot identify is a verification challenge is invoked

Step 16: If Step 15 results true, keep on executing next steps, otherwise jump to Step 33

Step 17: Make bot switch to the current default content and identify the challenge frame

Step 18: Make bot identify the audio challenge

Step 19: Enable bot to click on the identified audio challenge option

Step 20: Make bot to switch to the default content

Step 21: Once switched to audio challenge, retrieve the HTML for this web page

Step 22: Make bot find the online source of this audio file that was selected by the system for challenge solving

Step 23: Make bot download this audio file from the identified source in mp3 format extension

Step 24: Make bot load this mp3 audio file as a wav file in to the pydub.AudioSegment method to remove any possible overlapping voices and noise

Step 25: Result of the Step 24 should be exported and stored as a sample audio

Step 26: Pass this sample audio in the speech recognition method

Step 27: Result of the Step 26 returns a key. This key represents the text spoken in that audio clip of the raised human verification challenge

Step 28: Make bot identify the text field, in which Captcha expects the solution for the challenge

Step 29: Enable bot to enter the solution key typed into the identified text field

Step 30: Enable bot to hit and invoke the return key event of the keyboard

Step 30: Make bot identify the Captcha verify button

Step 31: Enable bot to click on the identified button

Step 32: Captcha will be solved now, and the original targeted webpage will be opened now, that we want to scrap

Step 33: As soon as the webpage opens, a pop-up box opens up asking for user location. Make bot identify the close button of this window

Step 34: Enable bot to click on the identified window closing button

Step 35: Calculate the number of scrolling actions required to scroll this webpage down to the very bottom by enumerating the current window resolution and the total height of our webpage

Step 36: Enable bot to scroll the webpage on the basis of exact scrolling actions required, separated by random delay intervals

Step 37: Once reached the bottom of the page, enable bot to reach once again the very top of the page

Step 38: Now retrieve the HTML of this webpage

Step 39: After retrieving the HTML, identify and scrape all those food item blocks in which food image is also posted

Step 40: Scrape the restaurant banner, restaurant title and rating of this restaurant

Step 41: Make bot identify the button that redirects the control to the about window

Step 42: Enable bot to click on the identified redirection button

Step 43: Make bot to switch to the new default content of the webpage

Step 44: Retrieve the HTML of this new window

Step 45: Scrape restaurant address, restaurant opening and closing time, restaurant latitude and longitude from this HTML

Step 46: Make bot to identify the button that will redirect the control to the reviews tab section

Step 47: Enable bot to click on the identified element

Step 48: Make bot to switch to the new default content of the webpage

Step 49: Retrieve the HTML of this new window

Step 50: Make bot to identify all the review blocks in this window

Step 51: Scrape reviewer's name, date of review, rating posted, and descriptive comment posted in this review by the reviewer

Step 52: Open and store all the details scraped about a restaurant in the respected CSV file for the restaurant data

Step 53: Open and store all the details scraped about food items in the respected CSV file for the food data

Step 54: Open and store all the details scraped about reviews in the respected CSV file for the review data

Step 55: Check if the data for desired number of restaurants has been already collected

Step 56: If Step 55 results false, keep on executing next steps, otherwise jump to Step 60

Step 57: Pick up next entry from the first CSV file, in which all of the restaurant links are stored

Step 58: Close the current driver and configure a new web driver on this new entry

Step 59: Keep on repeating from Step 9 to Step 58 until result of step 56 is returned true

Step 60: Prompt that the scraping is now finished and terminate the execution

5.2 External APIs/SDKs

Following are the APIs and packages used in our system for the purpose of its successful development:

| Name of API and version | Description of API | Purpose of usage | List down the API endpoint/function/class in which it is used |
|--------------------------------------|--|---|---|
| Expo-Camera (12.0.3) | Component that renders camera in android phone | To capture images | CameraScreen() |
| Expo-Image Picker (11.0.3) | Let us to pick image from gallery | To pick image from gallery | CameraScreen() |
| Expo-Localization (11.0.0) | Change languages in application | To change languages in application | LanguageText() |
| Location (13.0.4) | Can get user location | To obtain user current location | RestaurantDetail() |
| React-native-async-storage (1.15.11) | Uses to store and get data from device storage | To store and to get data | AsyncStorageHelper() |
| Expo-notification (0.13.3) | Send notifications to device | Send notifications and perform action on notifications | Settings() |
| i18n-js (3.8.0) | Translate the text into different languages | Use to translate languages | LanguageText() |
| React-native-maps (0.28.1) | To display maps | To show maps and display directions of restaurants from user's location | RestaurantDetail() |

5.3 User Interface

Following attached are actual screenshots of the user interface of the developed system:

5.3.1 Splash Screen

Splash screen is the first screen in our smartphone application that pops up right after the application is launched by the user.



Figure 20: Splash Screen

5.3.2 Sign up Screen

Sign up screen provides the form of registration to the user, through which he can affiliate himself with the system.

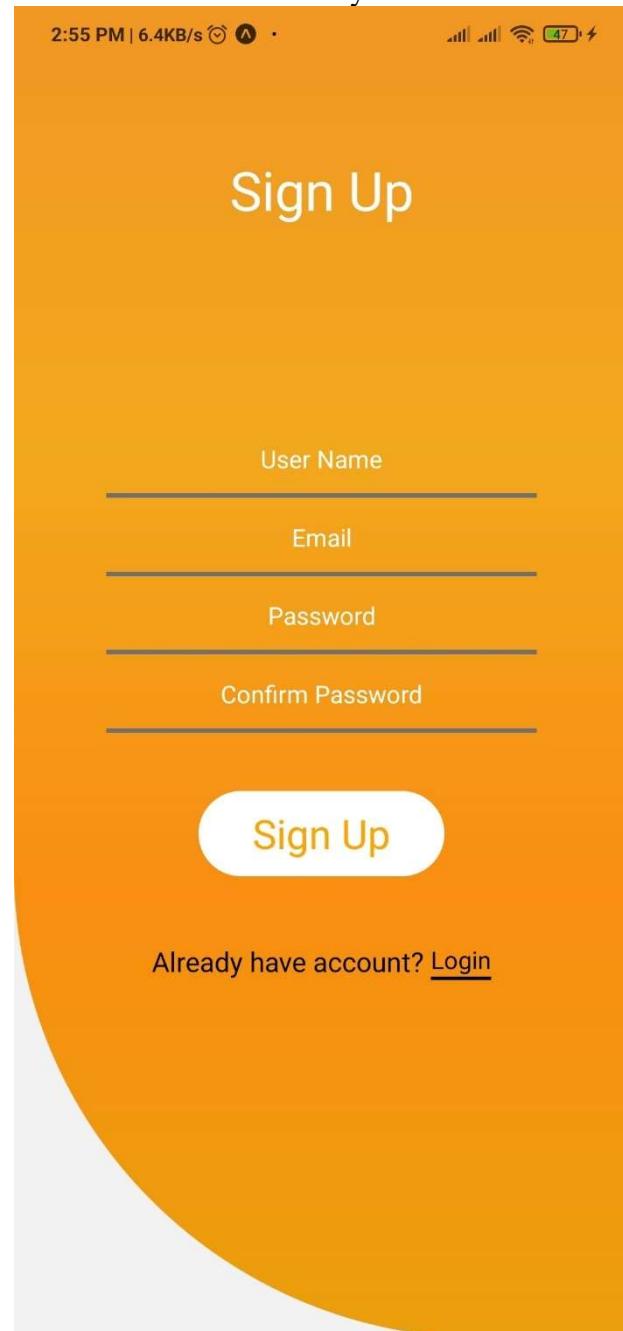


Figure 21: Sign up Screen

5.3.3 Login Screen

Login screen is the gateway of access for the users that are already registered with the system.

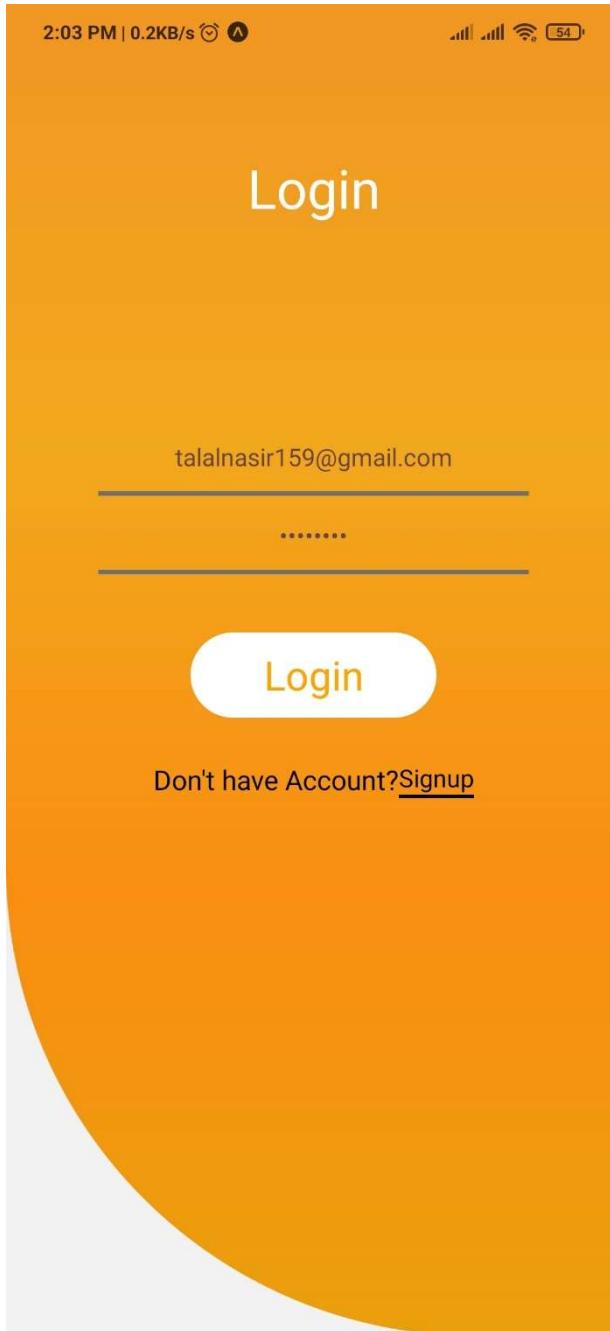


Figure 22: Login Screen

5.3.4 Home Screen

From the home screen, user has the access and can operate to almost all the main functionalities provided by the system.

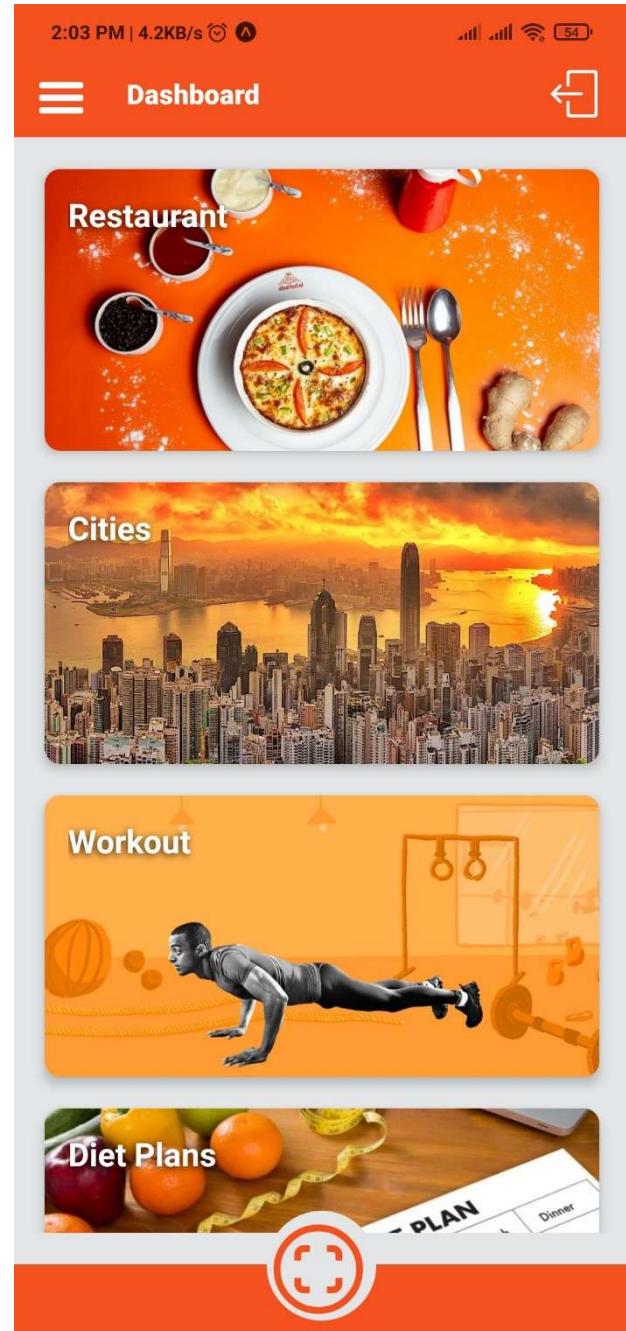


Figure 23: Home Screen

5.3.5 Restaurant Section Screen

On this screen, user can view, search, and select all of the restaurants existing in a city. Moreover, user can change the search location of restaurants as well.

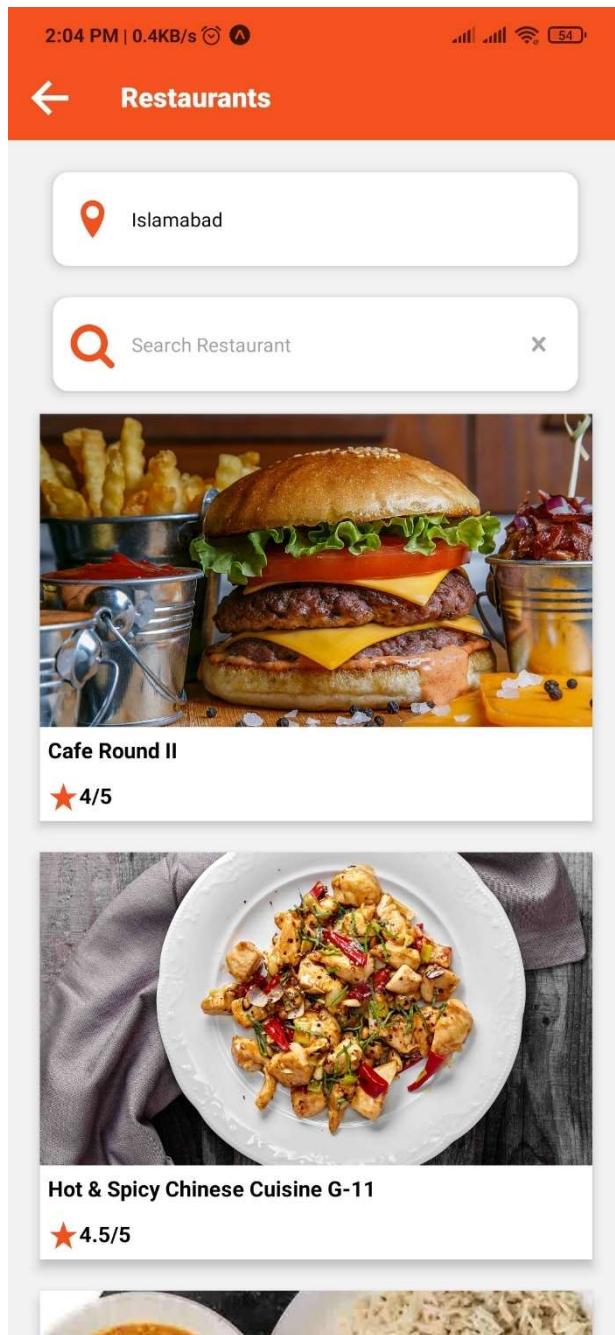


Figure 24: Restaurant Section Screen

5.3.6 Restaurant Location Screen

User can change the location of the restaurants. Such that, using this option, the resulting restaurants will be from another city, that is selected from the list of provided options.

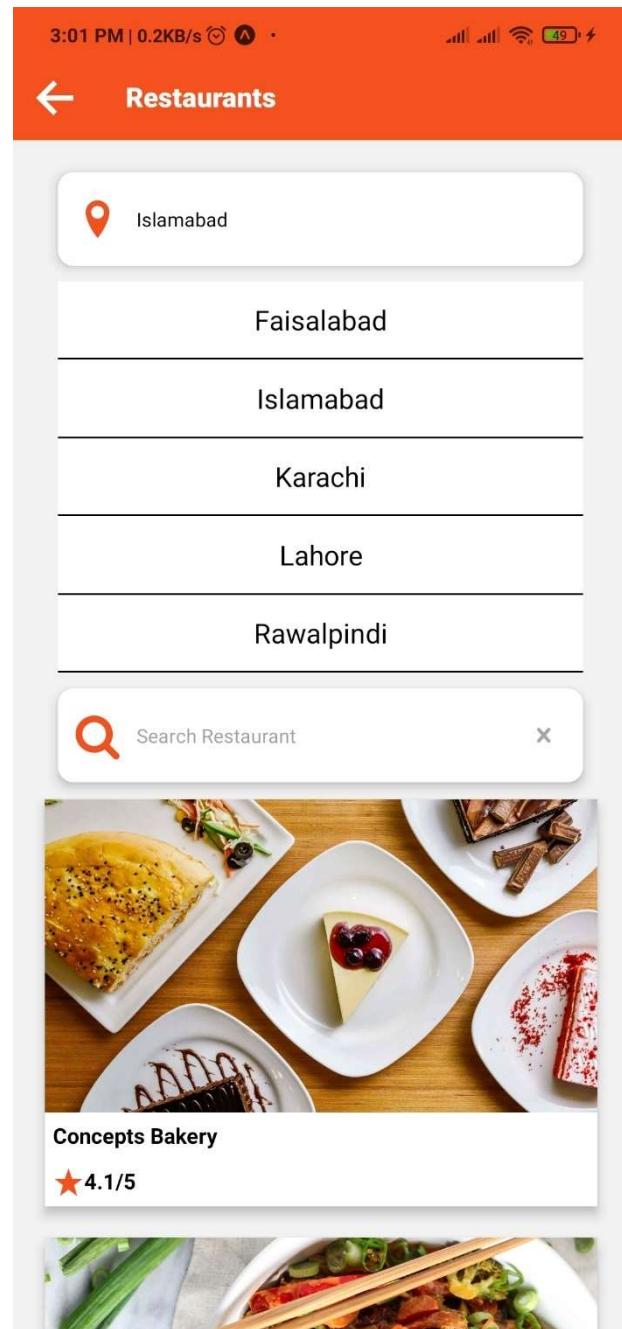


Figure 25: Restaurant Location Screen

5.3.7 Restaurant Searching Screen

Dynamic search is utilized for this functionality where the results are filtered out with each and every input of the user.

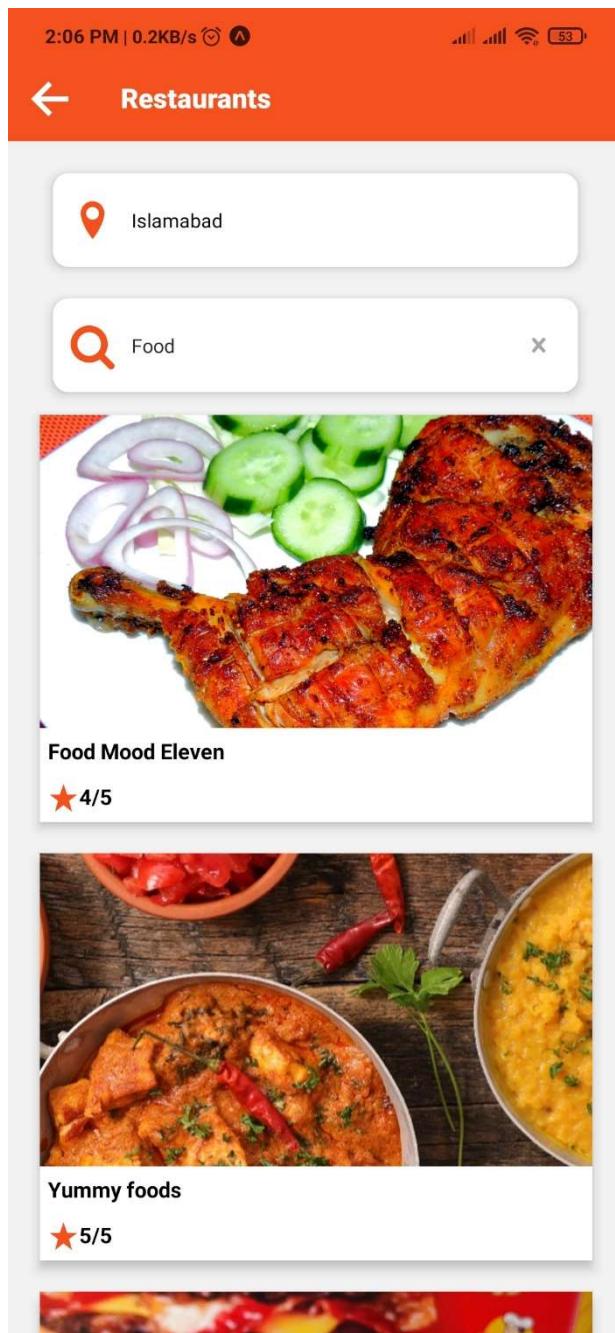


Figure 26: Restaurant Searching Screen

5.3.8 Restaurant Menu Screen

Tapping on any of the restaurant will bring forth the corresponding screen where the menu of that restaurant is made visible at first.

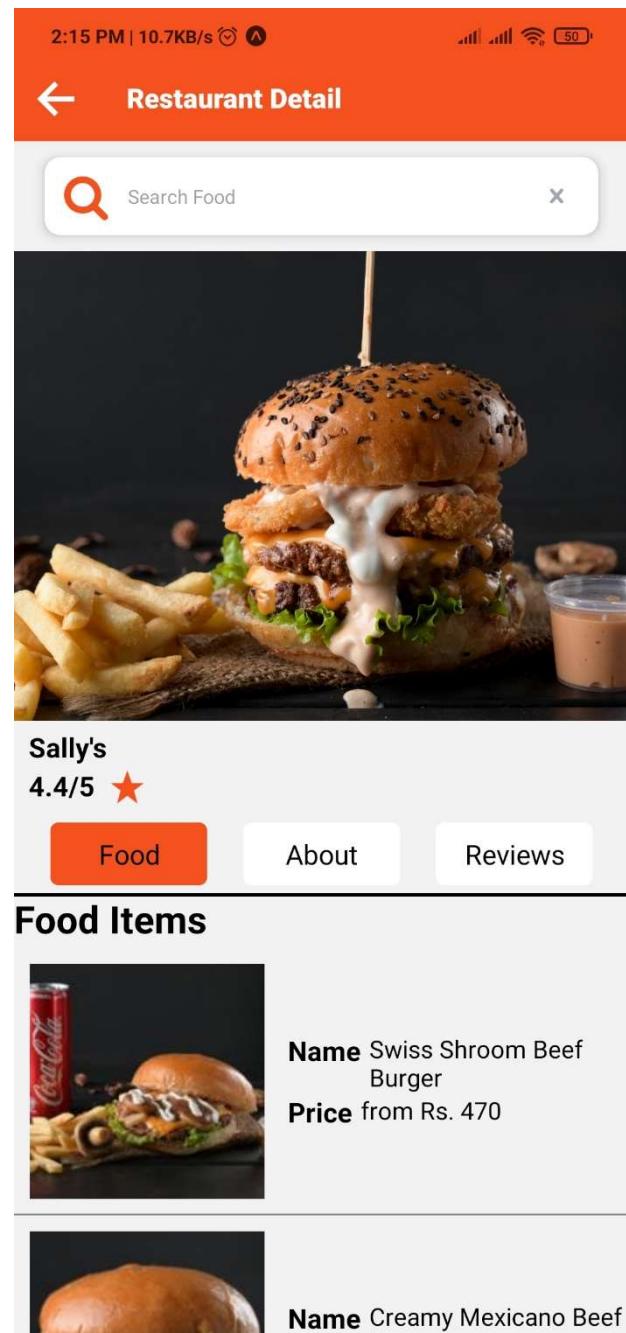
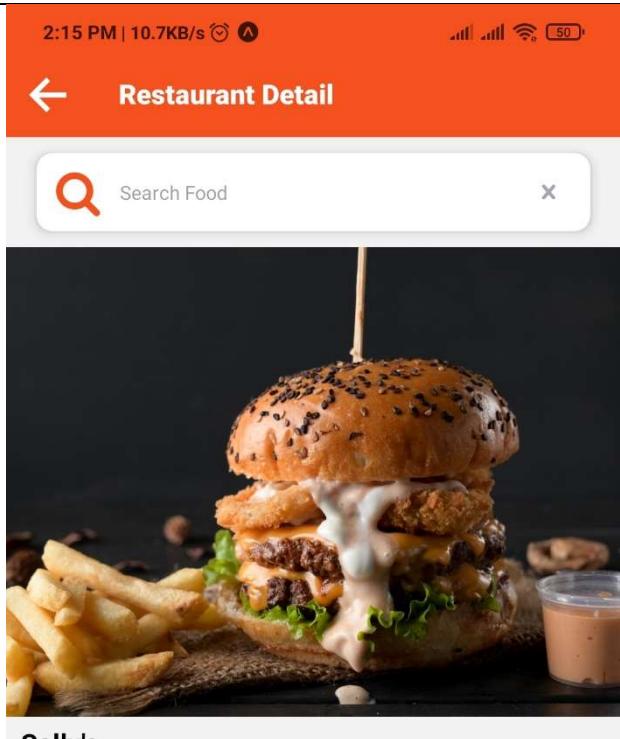


Figure 27: Restaurant Menu Screen

5.3.9 Food Searching Screen



Food Items

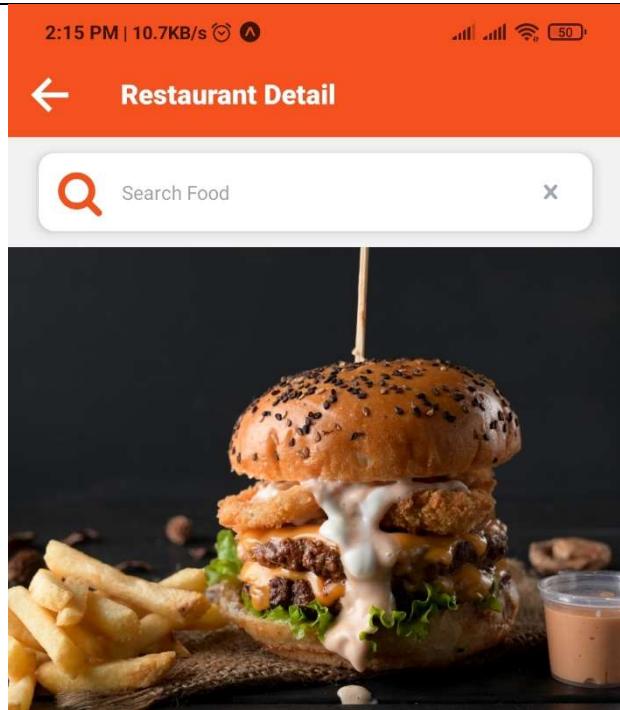


Name Swiss Shroom Beef Burger

Price from Rs. 470

Figure 28: Food Searching Screen

5.3.10 Restaurant About Screen



Food Items



Name Swiss Shroom Beef Burger

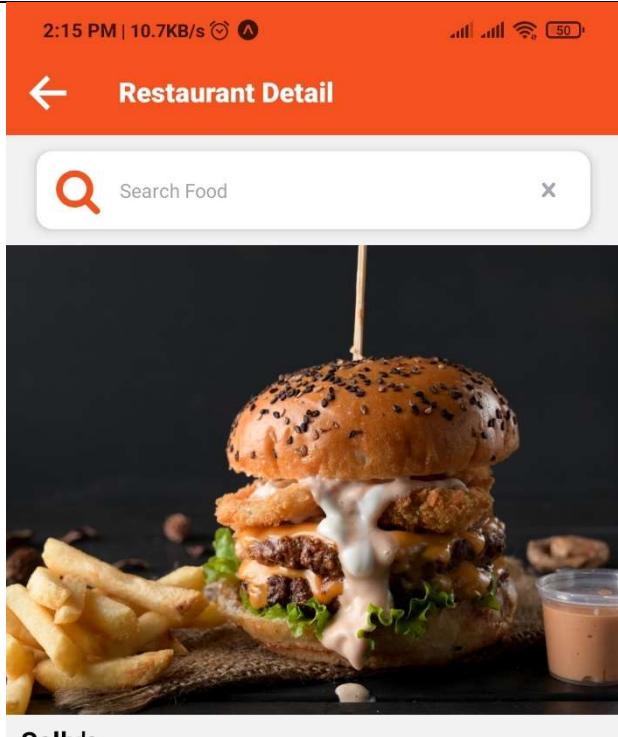
Price from Rs. 470



Name Creamy Mexicano Beef

Figure 29: Restaurant About Screen

5.3.11 Restaurant Feedback Screen



Food Items

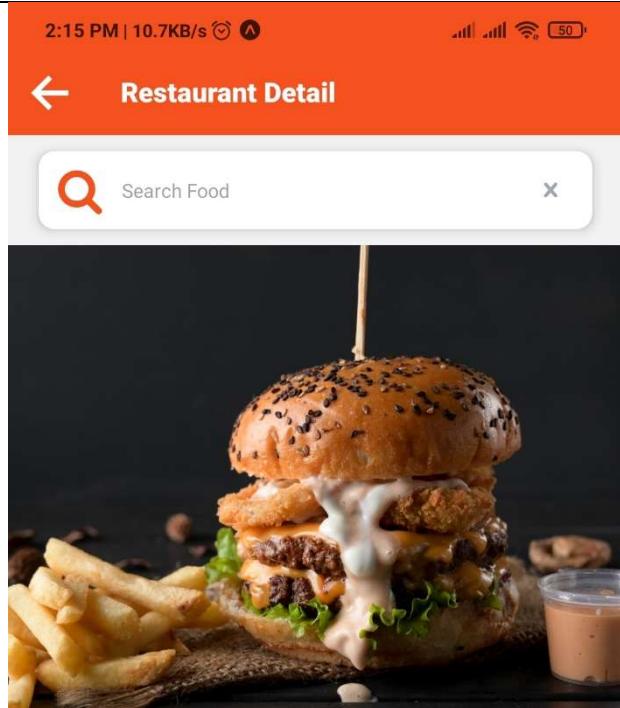


Name Swiss Shroom Beef Burger

Price from Rs. 470

Figure 30: Restaurant Feedback Screen

5.3.12 Posting Review Screen



Food Items



Name Swiss Shroom Beef Burger

Price from Rs. 470

Figure 31: Posting Review Screen

5.3.13 Cities Section Screen

Cities section is opened from home screen, that displays user the total cities in which he can be facilitated with the services of our system.



Figure 32: Cities Section Screen

5.3.14 Cities Selection Screen

Selecting a city from the city section screen will open up a new screen that contains a list of all of the restaurants from the selected city.

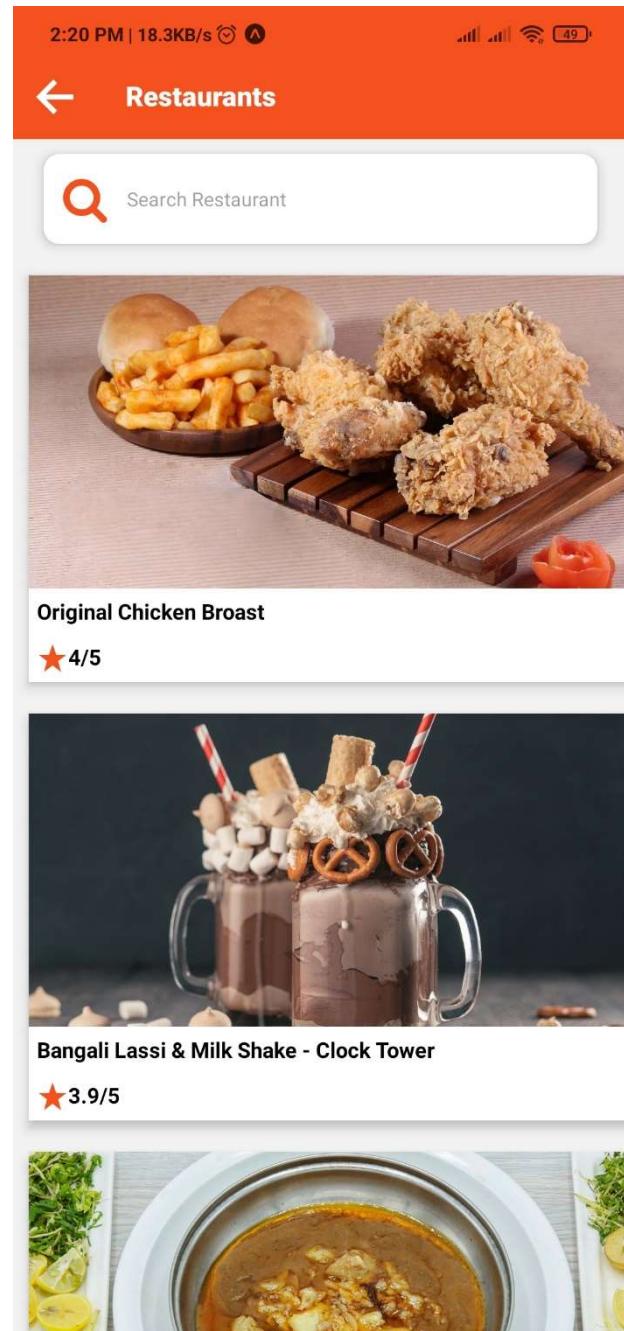


Figure 33: Cities Selection Screen

5.3.15 Workout Section Screen

Workout section is triggered from the home screen. This section displays different types of categories to the user that he can explore.

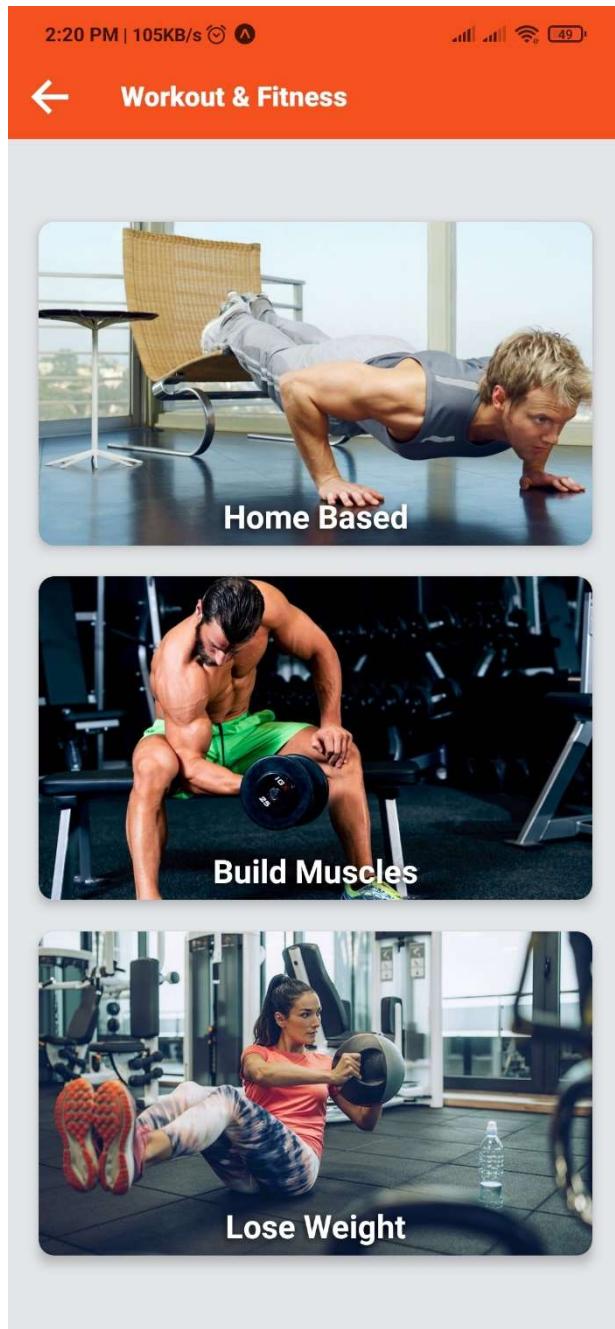


Figure 34: Workout Section Screen

5.3.16 Workout Types Screen

Each selected workout category contains multiple types of workouts. User can select any of the provided types to explore them.

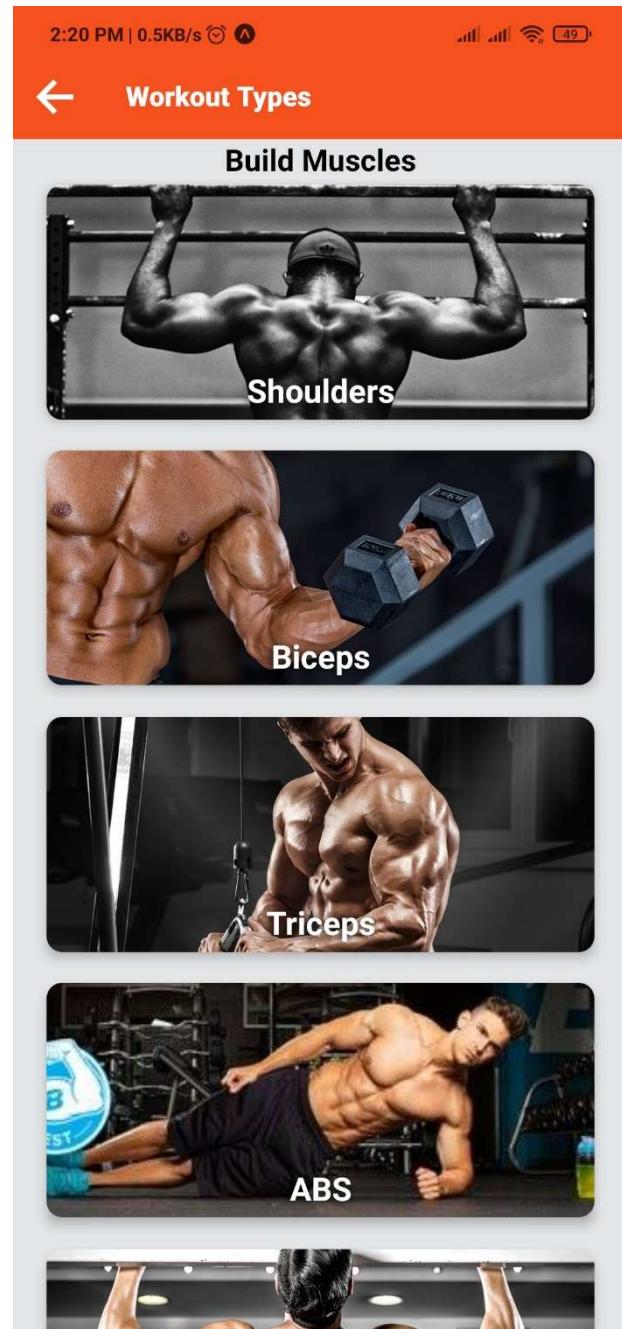


Figure 35: Workout Types Screen

5.3.17 Exercises Details Screen

Selecting any of the workout type will open a corresponding screen where different exercises with their name, visual demonstration, circuit, and repetition information are visible.

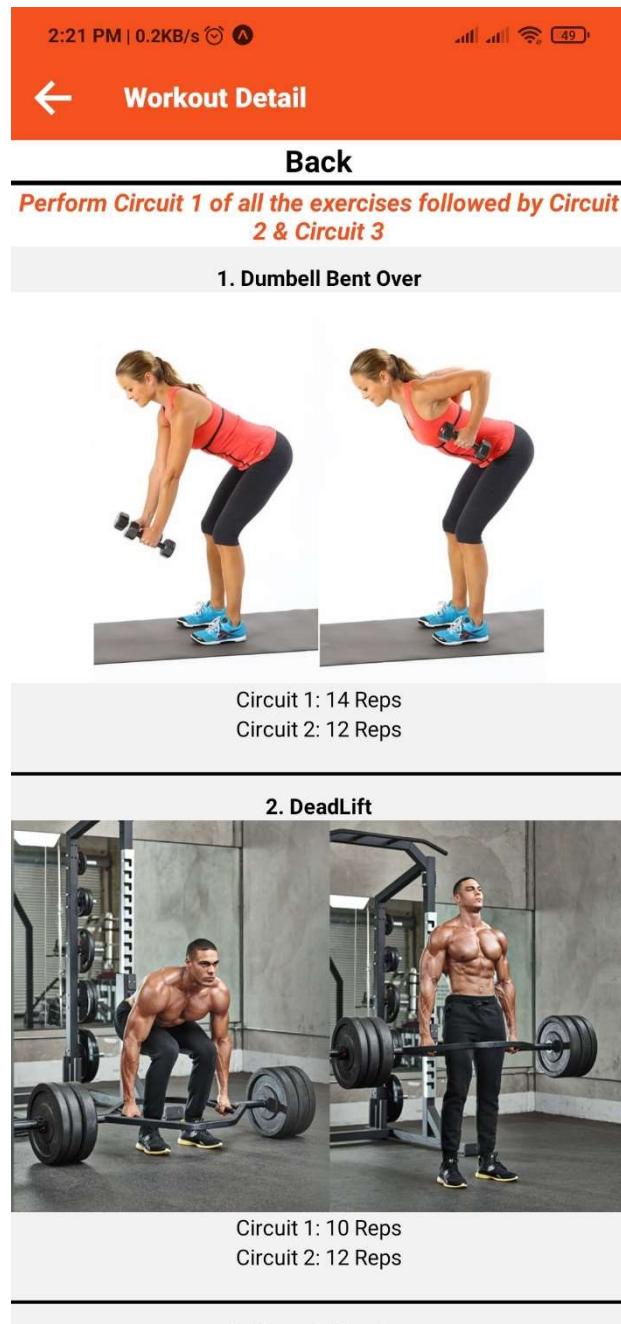


Figure 36: Exercise Details Screen

5.3.18 Exercise Instructions Screen

Exercise instruction screen is opened whenever user taps on a particular exercise. This screen displays a complete instruction manual to the user, on how to perform this selected exercise.

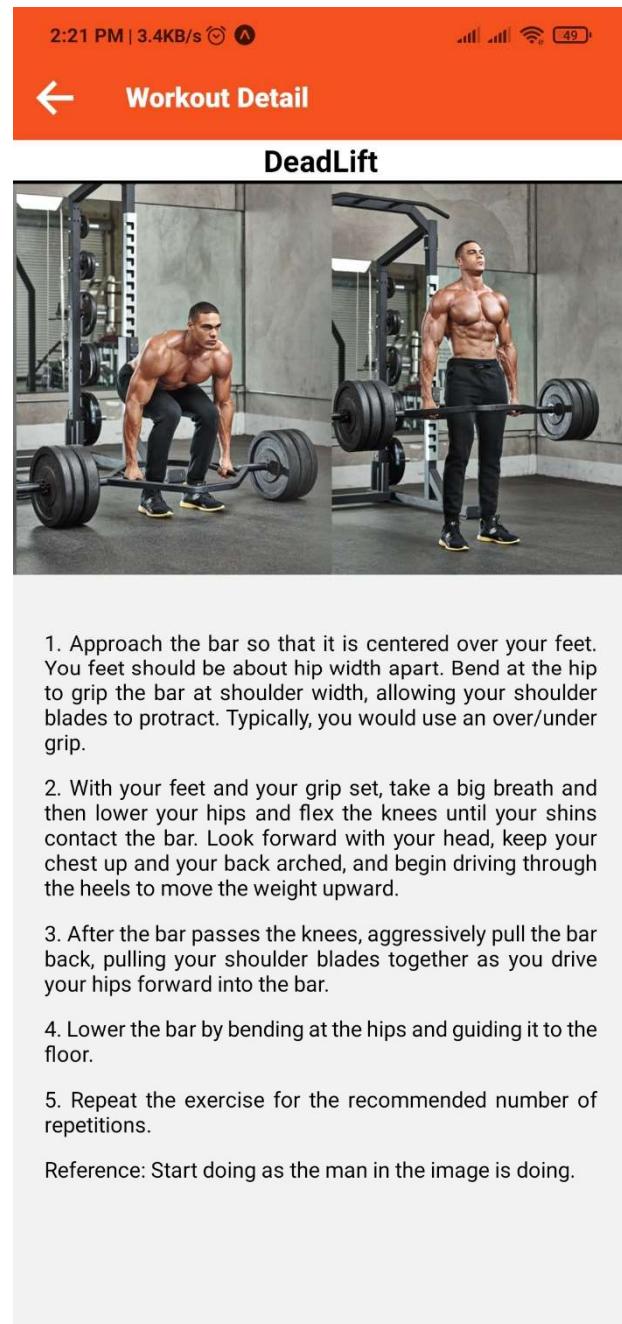


Figure 37: Exercise Instructions Screen

5.3.19 Diet Plan Section Screen

Diet plan section opened by tapping on it from the home screen. This screen contains two subsections where user can calculate his body mass index or explore different meal plans.

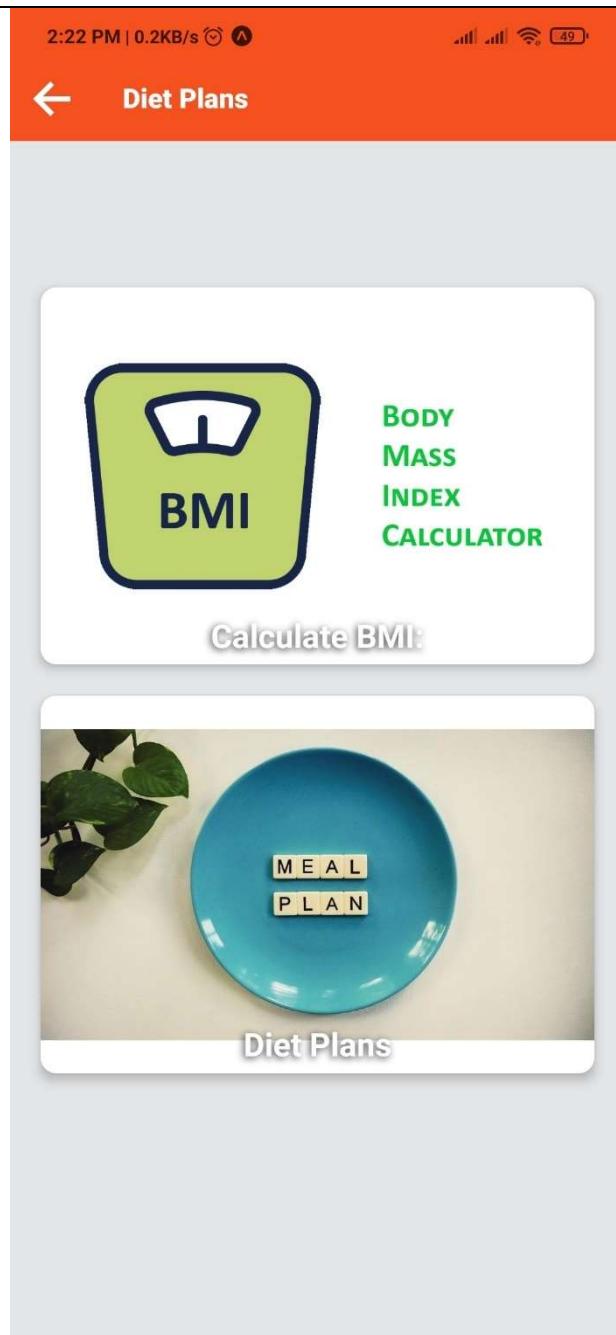


Figure 38: Diet Plan Section Screen

5.3.20 BMI Calculation Screen

A user can calculate his body mass index by accurately fulfilling the provided self-explanatory form.

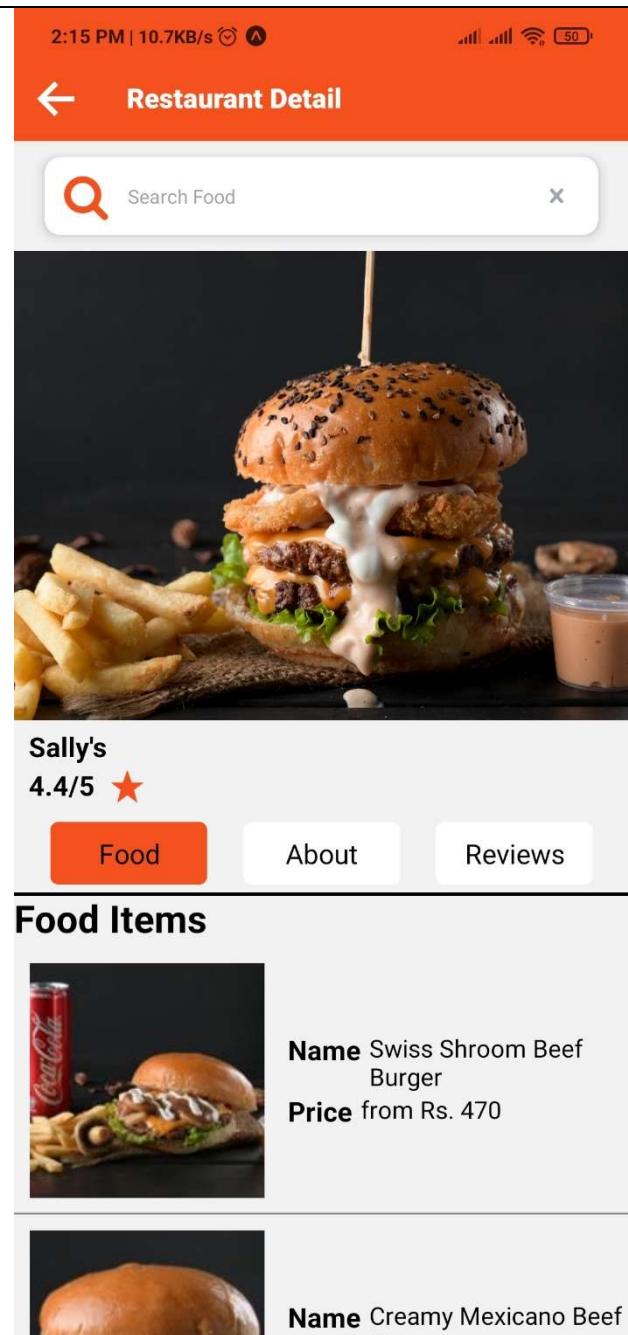


Figure 39: BMI Calculation Screen

5.3.21 Diet Plan Category Screen

This screen is opened from diet plan section screen, and it contains different types of diet plans that the user can explore and adopt.

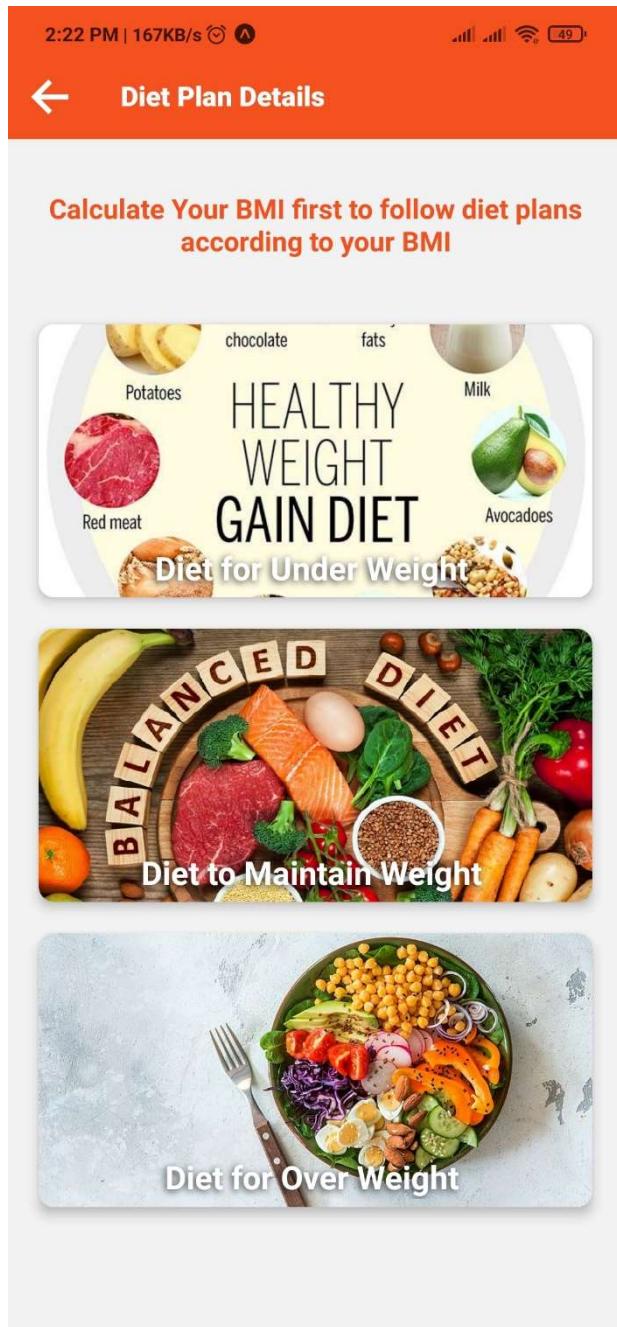


Figure 40: Diet Plan Category Screen

5.3.22 Weekday Selection Screen

Whichever diet plan category screen is selected by the user, the very next screen will ask the user to select the day of his meal.

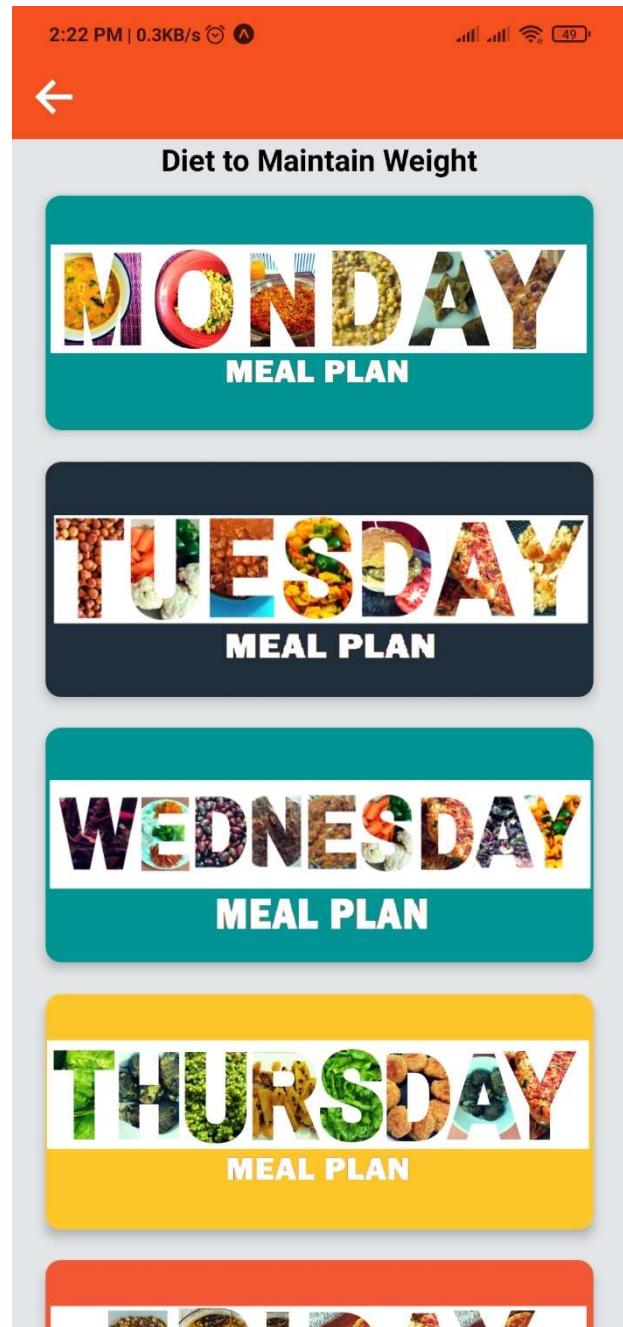


Figure 41: Weekday Selection Screen

5.3.23 Meal Plan Contents Screen

Upon selection of the day, user will be displayed with different types of meals that he will have to adopt on that day, along with the permitted durations and chosen edibles.

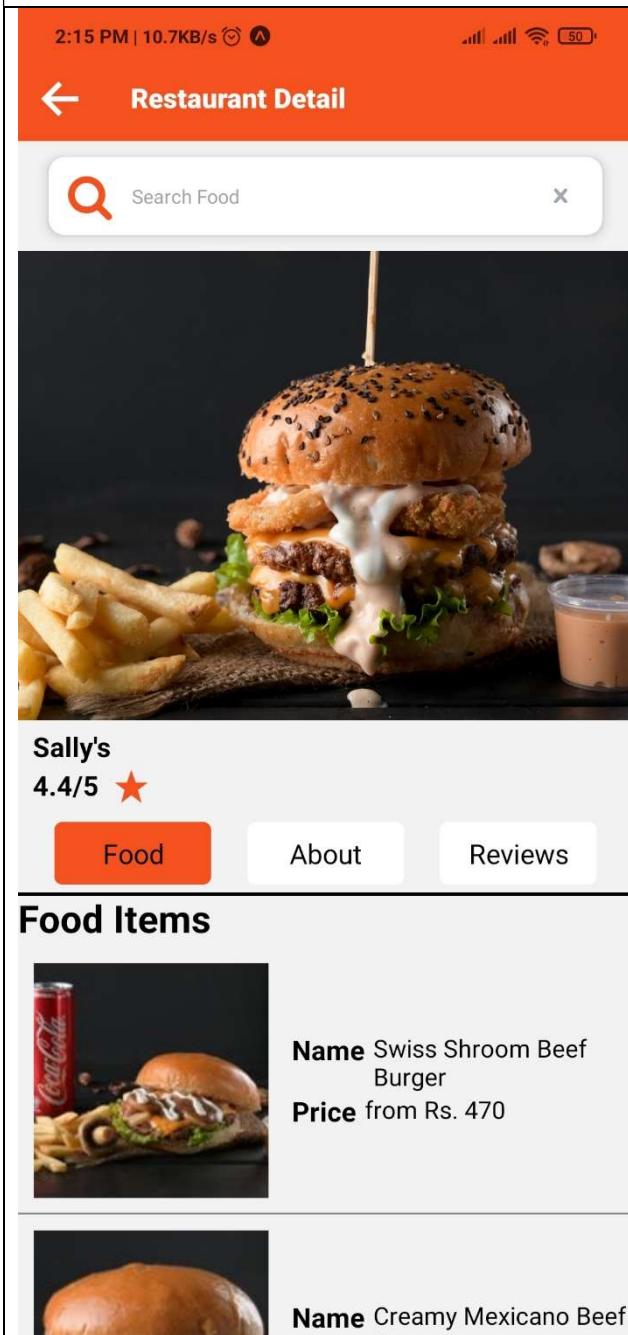


Figure 42: Meal Plan Contents Screen

5.3.24 Side Drawer Screen

From the home screen, tapping on the top left corner will animate forth the side drawer panel that contains menu for accessing multiple of functionalities.

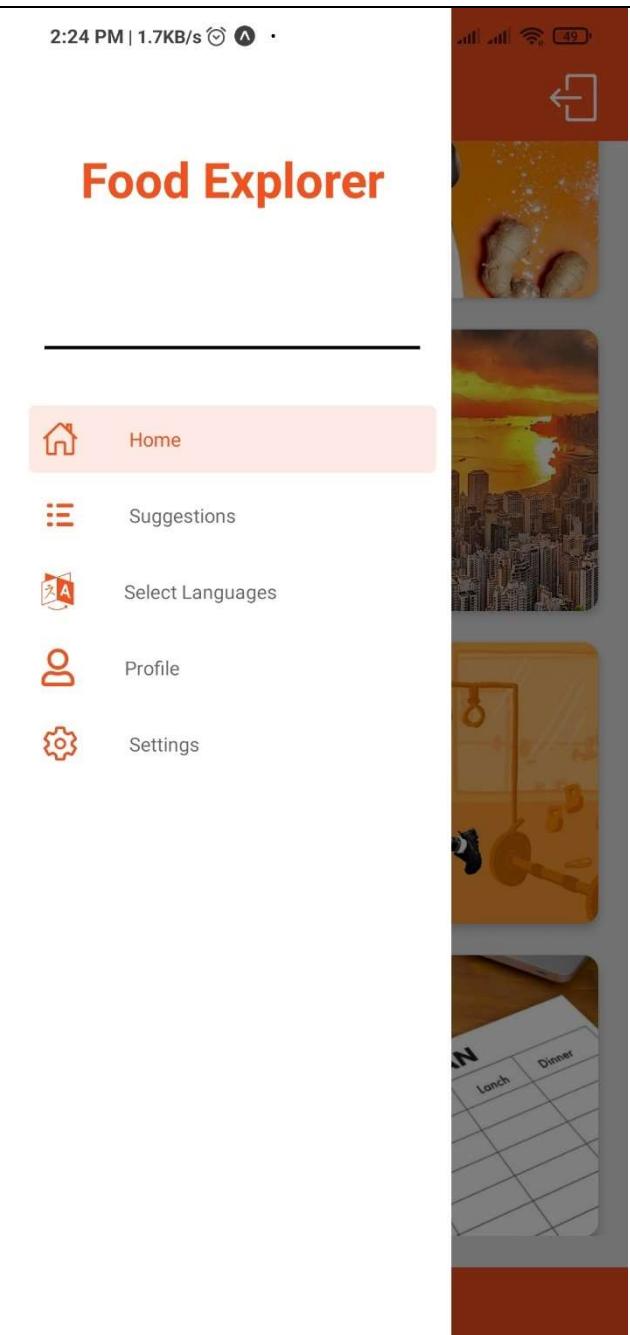


Figure 43: Side Drawer Screen

5.3.25 Suggested Restaurants Screen

From the side drawer, user can tap on the Suggestions option to view the most favored restaurants in a city by other users.

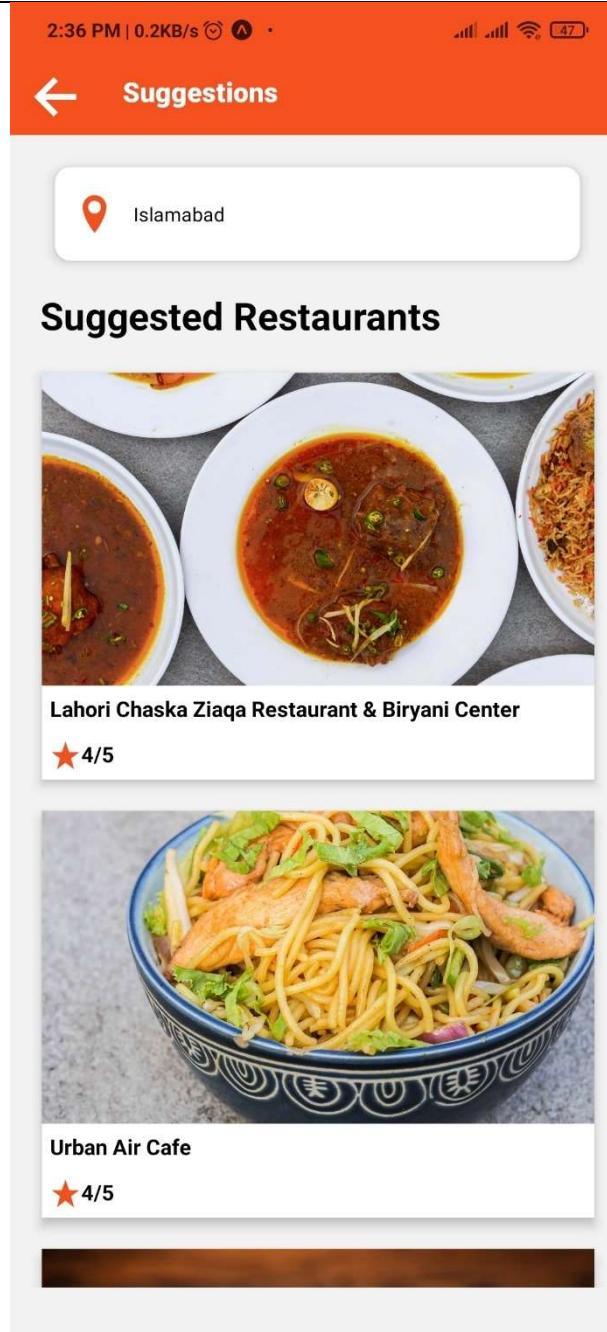


Figure 44: Suggested Restaurant Screen

5.3.26 Change Suggestions Location Screen

User can change the location of the restaurants that he wants to get suggested from. Using this option will display corresponding places.

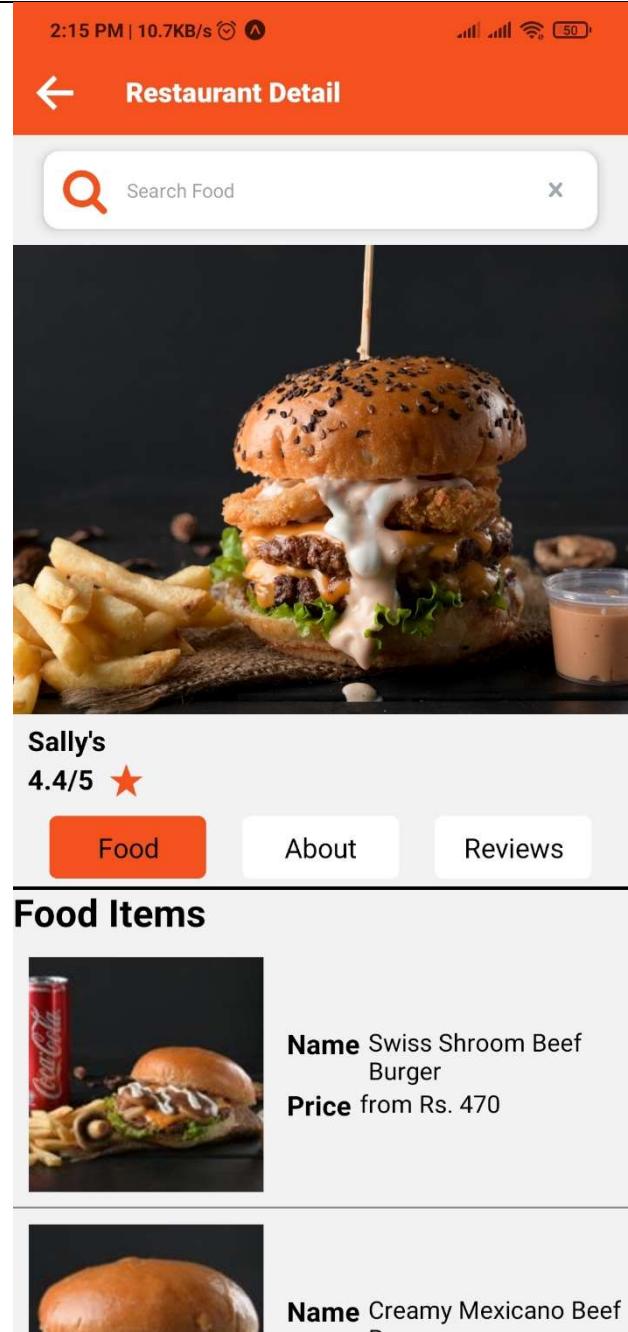


Figure 45: Change Suggestion Location Screen

5.3.27 Change Language Screen

Opening from the side drawer, this screen offers the user to select any language from the provided list to instantly change it.

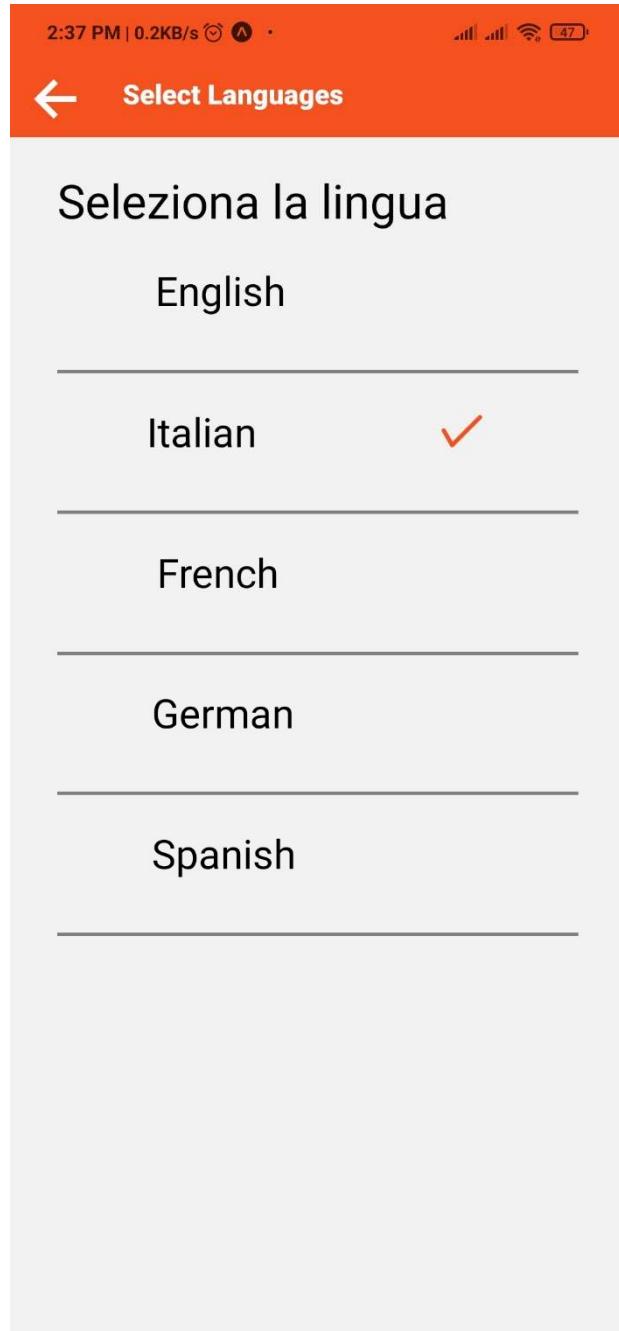


Figure 46: Change Language Screen

5.3.28 Sample Language Screen [1/3]

This screen is the first sample and is just to show the example of how the system will look under the Italian language.

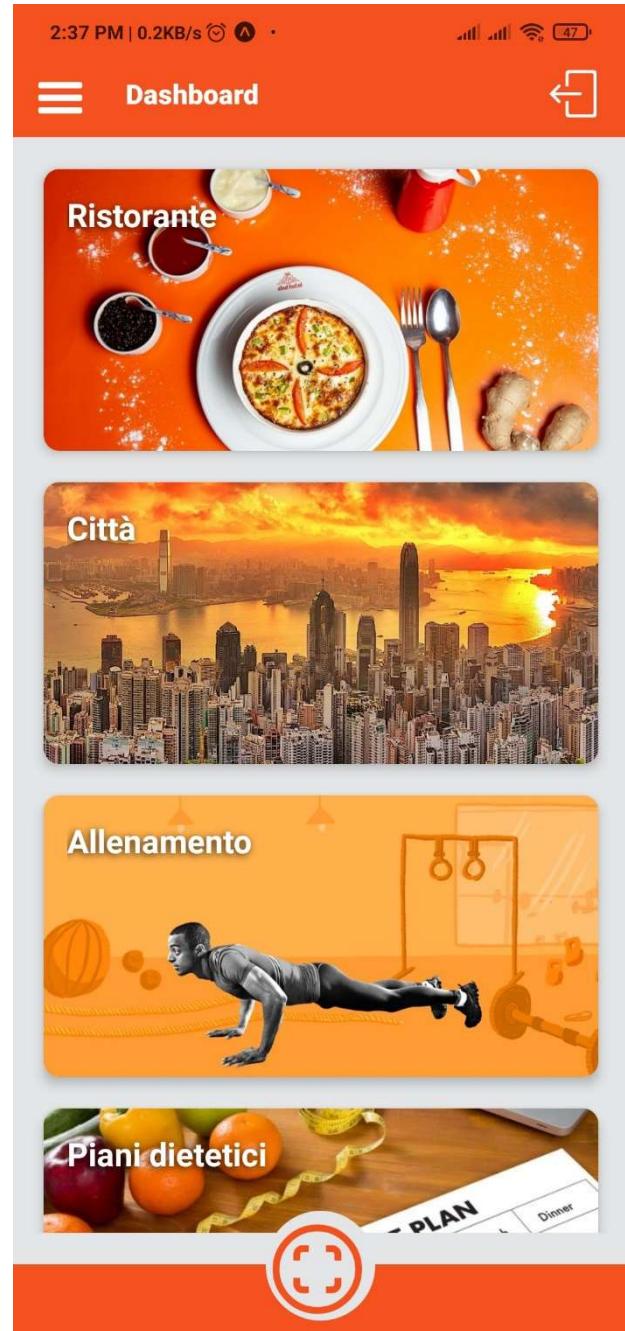


Figure 47: Sample Language Screen [1/3]

5.3.29 Sample Language Screen [2/3]

This screen is the second sample and is just to show the example of how the system will look under the Italian language.



Figure 48: Sample Language Screen [2/3]

5.3.30 Sample Language Screen [3/3]

This screen is the third sample and is just to show the example of how the system will look under the Italian language.

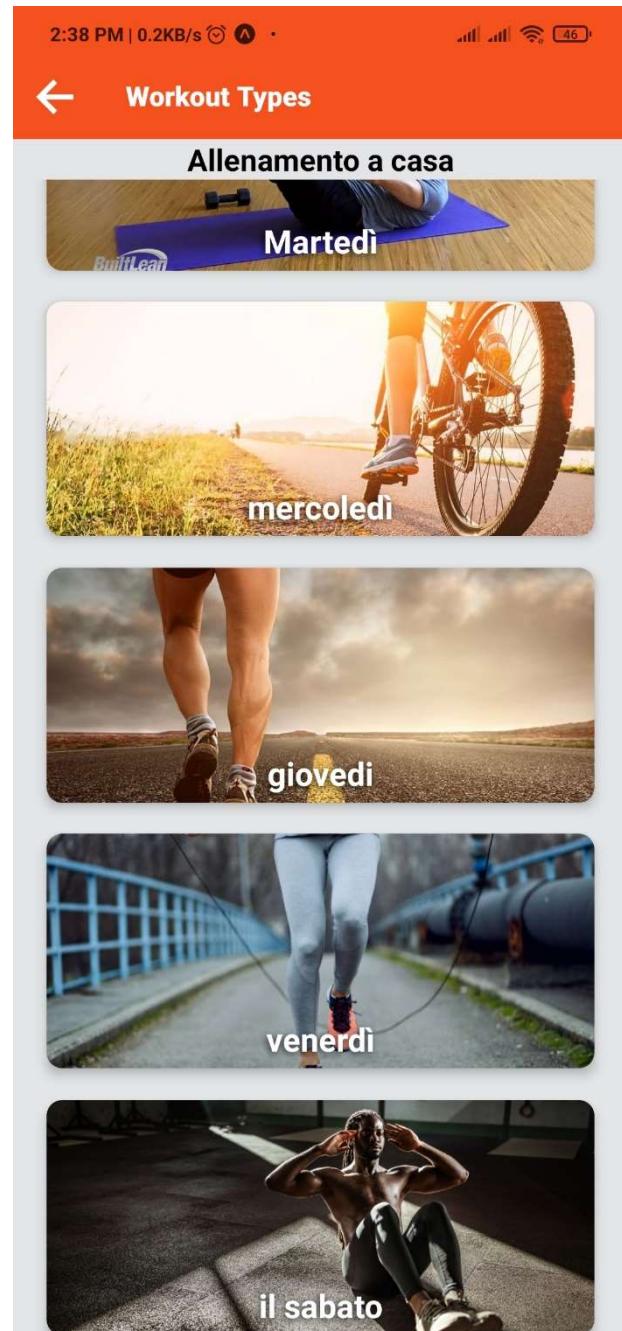


Figure 49: Sample Language Screen [3/3]

5.3.31 Camera Screen

Tapping on the bottom center camera icon from the home screen will open the smartphone camera after user grants the camera access permission.



Figure 50: Camera Screen

5.3.32 Gallery Screen

Tapping on the bottom right gallery icon button from camera screen will open the smartphone gallery after user grants the storage access permission.

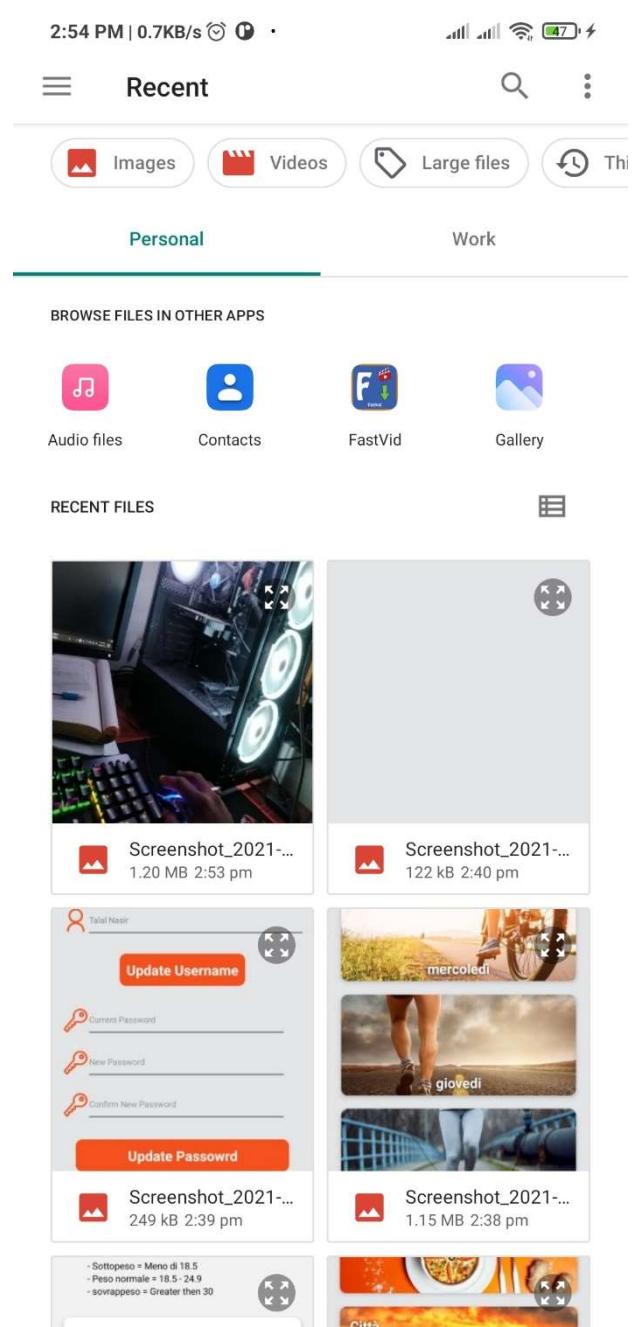


Figure 51: Gallery Screen

5.3.33 Login Screen

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

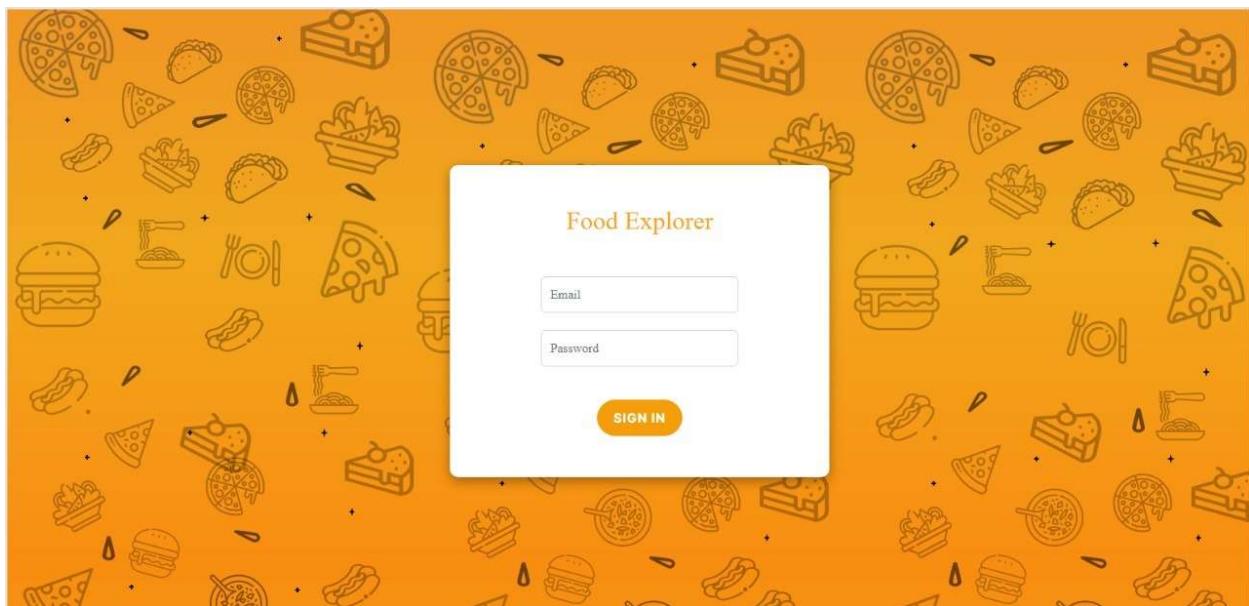


Figure 52: Login Screen (Web)

5.3.34 Dashboard Screen

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

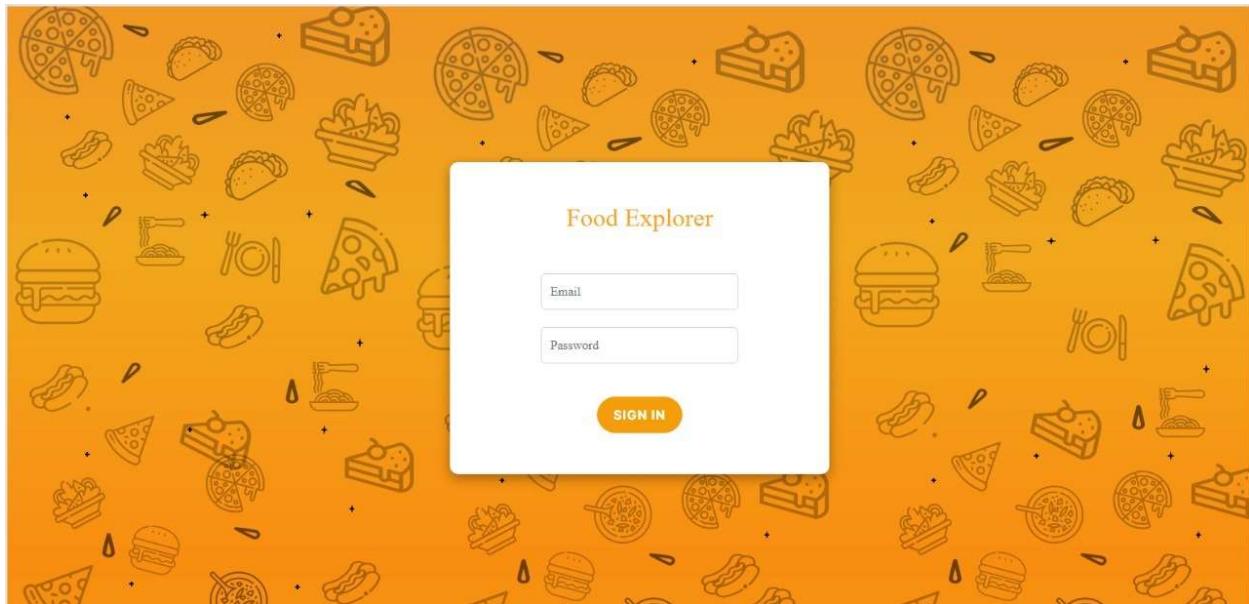


Figure 53: Dashboard Screen (Web)

5.3.35 Restaurant Screen

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

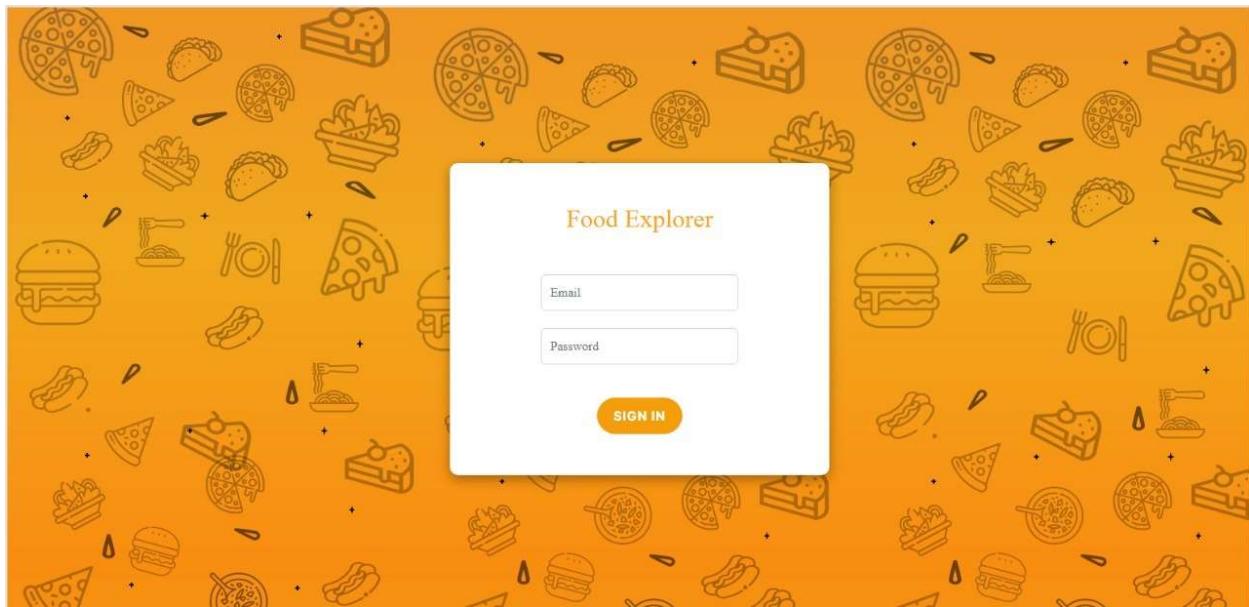


Figure 54: Restaurant Screen (Web)

5.3.36 Profile Screen

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

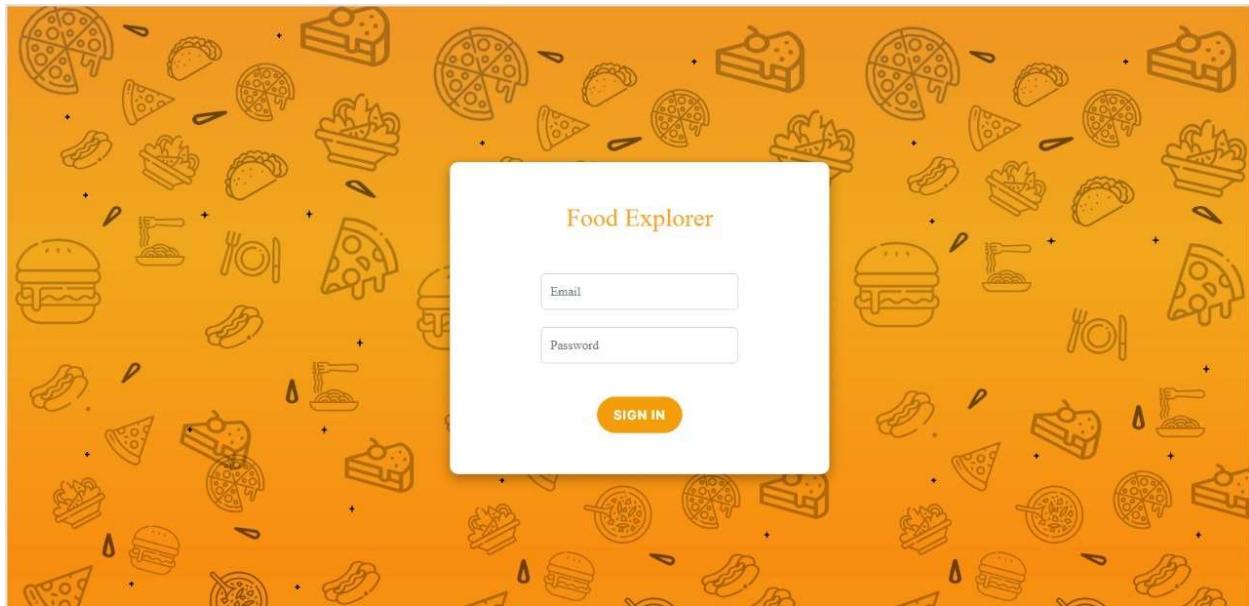


Figure 55: Profile Screen (Web)

5.4 Deployment

We have two applications in our system: a mobile app for end-user, and a web app for the admin. Both of these applications are duly deployed by the development team and hosted on two different platforms, i.e., our web application and custom API have been deployed on Heroku, and the smartphone application is deployed on Expo.dev. Details for both of them is given below:

Heroku Hosting:

We have worked with Heroku's 'Deploying with Git' variant for deploying our web application on Heroku platform. This option for deploying undergo the following process in the same chronology:

- Installing Git and Heroku Command Line Interface
- Track our application in Git
- Creating a remote repository on Heroku and renaming it as desire
- Pushing code from our Git's main and/or non-main branches on this newly created Heroku remote branch
- Defining a custom pipeline for building our code and deploying

Expo Hosting:

The smartphone application of food explorer is developed in JavaScript's React Native using the Expo CLI platform. The deployment of this smartphone application is handled using the same Expo platform mentioned earlier, with the help of Expo.dev to be explicit. A brief explanation of the process is as follow:

- Upon the development completion, from the project's CMD, we run the command of building the project's APK
- This command initiates the process of generating the smartphone application's installer APK file.
- Once this process is finished, the mentioned APK is automatically uploaded by the platform at their native servers.
- A URL is then passed down to the account owner, through which he can directly download that APK file into his smartphone for the installation purpose.

6 Testing and Evaluation

Following are all of the test cases performed on our project Food Explorer, upon the stage of its 100 percent completion:

6.1 Unit Testing

Unit Testing 1: Sign up as end-user with valid and invalid conventions

Testing Objective: To ensure the signup functionality is working correctly for all valid and invalid inputs.

Table 182: Test case 01

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|---|--|--------|
| 1. | Filling signup form with properly formatted email and password | Email: khokhar@gmail.com Password: 1234abcd | User will be successfully registered with the system | Pass |
| 2. | Filling signup form with faulty email but correct password | Email: khokhar@gmail OR Khokhar.com Password: 1234abcd | User will not be able to register with system | Pass |
| 3. | Filling signup form with proper email but without password | Email: khokhar@gmail.com Password: | User will not be able to register with system | Pass |
| 4. | Filling signup form with proper email but faulty password | Email: khokhar@gmail.com Password: abc | User will not be able to register with system | Pass |

Unit Testing 2: Login as end-user with valid and invalid credentials

Testing Objective: To ensure the login form is working correctly with valid and invalid credentials/inputs.

Table 183: Test case 02

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|--|---|--------|
| 1. | Filling login form with correct email and password | Email: khokhar@gmail.com Password: 1234abcd | User will successfully login the system | Pass |
| 2. | Filling login form with faulty email but correct password | Email: khokhar@gmail OR Khokhar.com Password: 1234abcd | User will not be able to login the system | Pass |
| 3. | Filling signup form with proper email but without password | Email: khokhar@gmail.com Password: | User will not be able to login the system | Pass |
| 4. | Filling signup form with proper email but faulty password | Email: khokhar@gmail.com Password: abc | User will not be able to login the system | Pass |

Unit Testing 3: To verify the process of viewing cities

Testing Objective: To ensure the screen for viewing cities loads up with respective data.

Table 184: Test case 03

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|--------------------------------------|---|--------|
| 1. | From home screen of mobile app, tap on the block that states ‘View Cities’ | Tapping on the ‘View Cities’ section | User will be able to successfully view all of the cities on the directed screen | Pass |

Unit Testing 4: To verify the functionality of selecting a city

Testing Objective: To ensure the functionality for selecting a city is working correctly and the relative screen properly loads up.

Table 185: Test case 04

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|--|--|--------|
| 1. | From within the ‘View Cities’ screen, tap on any of the displayed city | Tapping on one of the displayed cities | User will be able to successfully view all of the restaurants of the selected city | Pass |

Unit Testing 5: To verify the process of viewing restaurants

Testing Objective: To ensure the process of viewing restaurants is working correctly and the relative screen properly loads up with respective data.

Table 186: Test case 05

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|---|--|--------|
| 1. | From home screen of mobile app, tap on the block that states ‘View Restaurants’ | Tapping on the ‘View Restaurants’ section | User will be able to successfully view all of the restaurants on the directed screen | Pass |

Unit Testing 6: To verify the functionality of selecting restaurant location

Testing Objective: To ensure the feature of selecting restaurant location is working correctly and the relative data properly loads.

Table 187: Test case 06

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|---|--|--------|
| 1. | From within the ‘View Restaurants’ screen, tap on any of the available location from the dropdown list on the top | Tapping on one of the displayed locations | User will be able to successfully view all of the restaurants of the selected location | Pass |

Unit Testing 7: To verify the functionality of selecting a restaurant

Testing Objective: To ensure the feature of selecting a restaurant is working correctly, and the relative screen properly loads up.

Table 188: Test case 07

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|---|--|--------|
| 1. | From within the ‘View Restaurants’ screen, tap on any of the available restaurant from the screen | Tapping on one of the displayed restaurants | User will be successfully redirected to the selected restaurant’s screen | Pass |

Unit Testing 8: To verify the process of viewing food items

Testing Objective: To ensure the functionality of viewing food items is working correctly and all of the selected restaurant’s food items are displaying on the opened-up screen.

Table 189: Test case 08

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|---|--|--------|
| 1. | After selecting a restaurant from the list of restaurants, all the food items against that restaurant are displayed | Tapping on one of the displayed restaurants | User will be successfully able to view all of the food items of a restaurant | Pass |

Unit Testing 9: To verify the process of viewing restaurant details and information

Testing Objective: To ensure the functionality of viewing restaurant details and information is working correctly.

Table 190: Test case 09

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|---|---|--------|
| 1. | After selecting a restaurant from the list of restaurants, all the details of this restaurant are visible | Tapping on one of the displayed restaurants | User will be successfully able to view all of the details about this restaurant | Pass |

Unit Testing 10: Viewing reviews and feedbacks against a restaurant

Testing Objective: To ensure the functionality of viewing reviews and feedback is working as expected.

Table 191: Test case 10

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|---|---|--------|
| 1. | After selecting a restaurant from the list of restaurants, tap on the reviews tab to open the respective section | Tapping on the 'Reviews' tab on the restaurant screen | User will be successfully able to view all reviews and feedbacks of this restaurant | Pass |

Unit Testing 11: Posting feedback against a restaurant

Testing Objective: To ensure the functionality of posting feedback is working correctly.

Table 192: Test case 11

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|---|---|--------|
| 1. | Filling out feedback form with all of the required fields | Rating: 3/5 Stars Review: Great Food | User will be successfully able to post a review for this restaurant | Pass |
| 2. | Filling out feedback form with Rating, but not Review | Rating: 1/5 Stars Review: | User will not be able to post feedback for this restaurant | Pass |
| 3. | Filling out feedback form with Review, but not Rating | Rating: Review: Worst Taste | User will not be able to post feedback for this restaurant | Pass |

Unit Testing 12: Searching a restaurant**Testing Objective:** To ensure the feature of searching a restaurant is working correctly.**Table 193: Test case 12**

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|--|---|--------|
| 1. | From within the ‘View Restaurants’ screen, tap on the search restaurant bar on the top of the screen, and enter name of a restaurant | Tapping on the search bar Type: Fresh Sip | Searched restaurant is successfully appeared in front of the user on the screen | Pass |
| 2. | From within the ‘View Restaurants’ screen, tap on the search restaurant bar on the top of the screen, but do not enter name of any restaurant | Tapping on the search bar Type: | No restaurant will appear before user on the screen | Pass |

Unit Testing 13: View Workout Plans**Testing Objective:** To ensure the functionality of viewing Workout Plans is working correctly.**Table 194: Test case 13**

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|--------------------------------------|--|--------|
| 1. | Selecting the section of Workout Plans from the home screen of the application | Tapping on the Workout Plans section | User will be successfully able to view all the categories of Workout Plans | Pass |

Unit Testing 14: Selecting Workout Category

Testing Objective: To ensure the functionality of selecting a Workout category is working correctly.

Table 195: Test case 14

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|--|--|--------|
| 1. | Selecting one of the categories from the workout plans list displayed on the screen | Tapping on one of the Workout Category from the displayed list | User will be successfully able to view all the exercise types of selected Workout category | Pass |

Unit Testing 15: Selecting Exercise Type

Testing Objective: To ensure the functionality of selecting an Exercise Type is working correctly.

Table 196: Test case 15

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|--|--|--------|
| 1. | Selecting one of the exercise types from all of the available types of exercises | Tapping on one of the exercise types from the displayed list | User will successfully open the selected exercise type | Pass |

Unit Testing 16: Viewing Exercises details

Testing Objective: To ensure the functionality of Viewing Exercises details is working correctly.

Table 197: Test case 16

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|--|--|--------|
| 1. | Selecting one of the exercise types from all of the available types of exercises | Tapping on one of the exercise types from the displayed list | User will be successfully able to view all the exercises details of selected exercise type | Pass |

Unit Testing 17: View Diet Plans

Testing Objective: To ensure the functionality of viewing Diet Plans is working correctly.

Table 198: Test case 17

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|-----------------------------------|---|--------|
| 1. | Selecting the section of Diet Plans from the home screen of the application | Tapping on the Diet Plans section | User will be successfully able to view all the categories of Diet Plans | Pass |

Unit Testing 18: Selecting Diet Category**Testing Objective:** To ensure the functionality of selecting a Diet category is working correctly.**Table 199: Test case 18**

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|---|---|--------|
| 1. | Selecting one of the categories from the Diet plans list displayed on the screen | Tapping on one of the Diet Categories from the displayed list | User will be successfully able to open the selected Diet Category | Pass |

Unit Testing 19: Viewing Meal plan of a Diet Category**Testing Objective:** To ensure the functionality of Viewing Meal plan of a Diet Category is working correctly.**Table 200: Test case 19**

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|---|--|--------|
| 1. | Selecting one of the Diet types from all of the available categories of Diet | Tapping on one of the Diet categories from the displayed list | User will be successfully able to view the complete meal plan proposed against this Diet plan category | Pass |

Unit Testing 20: Calculate Body Mass Index

Testing Objective: To ensure the functionality of calculating Body Mass Index is working correctly.

Table 201: Test case 20

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|---|--|--------|
| 1. | Filling up the BMI calculation form by entering all of the required data | Weight (kg): 72 Height (feet): 5 Height (inches): 6 | User will be successfully able to get the calculated values of his BMI | Pass |
| 2. | Filling Up BMI calculation form by entering Weight value, but not height value | Weight (kg): 67 Height (feet): Height (inches): | User will not be able to successfully get the calculated values of his BMI | Pass |
| 3. | Filling Up BMI calculation form by entering Height value, but not weight value | Weight (kg): Height (feet): 6 Height (inches): 2 | User will not be able to successfully get the calculated values of his BMI | Pass |

Unit Testing 21: View Recommended Diet Plan

Testing Objective: To ensure the functionality of viewing Recommended Diet Plan is working correctly.

Table 202: Test case 21

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|-----------------------------------|--|--------|
| 1. | The user must first calculate the BMI of him on the application. After that, select the section of Diet Plans from the home screen of the application | Tapping on the Diet Plans section | User will be successfully able to view the recommended categories of Diet Plan on the screen | Pass |

Unit Testing 22: Viewing Suggestions

Testing Objective: To ensure the feature for viewing Suggestions is working correctly.

Table 203: Test case 22

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|---|--|--------|
| 1. | From home screen of mobile app, tap on the block that states ‘View Suggestions’ | Tapping on the ‘View Suggestions’ section | User will be able to successfully view all of the suggested restaurants on the directed screen | Pass |

Unit Testing 23: Selecting suggestion location

Testing Objective: To ensure the feature of selecting suggestion location is working correctly.

Table 204: Test case 23

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|---|--|--------|
| 1. | From within the ‘View Suggestions’ screen, tap on any of the available location from the drop-down list on the top | Tapping on one of the displayed locations | User will be able to successfully view all of the suggested restaurants of the selected location | Pass |

Unit Testing 24: Explore Suggested Restaurants**Testing Objective:** To ensure the feature of exploring suggested restaurants is working.**Table 205: Test case 24**

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|---|--|--------|
| 1. | From the list of suggested restaurants on the screen, select one of the displayed restaurants | Tapping on one of the displayed restaurants on suggestions screen | User will be able to successfully open and view that particular restaurant complete data on his screen | Pass |

Unit Testing 25: Open Smartphone's Camera**Testing Objective:** To ensure the feature of opening smartphone's camera is working correctly.**Table 206: Test case 25**

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|---|---|--------|
| 1. | From the Home Screen, tap on the camera icon button that can be found on the bottom of the screen | Tapping on camera button from the home screen | User will be able to successfully open the camera of his smartphone | Pass |

Unit Testing 26: Click a Picture from Smartphone's Camera

Testing Objective: To ensure the feature of taking picture from smartphone's camera works.

Table 207: Test case 26

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|---|---|--------|
| 1. | After opening the camera, tap on the shutter/capture icon – that can be found on the bottom of the screen - to click a picture | Tapping on camera button from the home screen | User will be able to successfully open the camera of his smartphone | Pass |

Unit Testing 27: Forward selected picture to API for classification

Testing Objective: To ensure that the picture is grabbed by the API for processing, is working correctly.

Table 208: Test case 27

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|---|---|--------|
| 1. | After successfully taking the picture, identify object button appeared on the screen to initiate the process | Tapping on identify object button after capturing the picture | Image will be successfully transferred to the backend API | Pass |

Unit Testing 28: Displaying identified object to the user

Testing Objective: To ensure that the results are displayed to the user correctly.

Table 209: Test case 28

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|-------------------------------|---|--------|
| 1. | After sending the image to API, grab its response and extract the label of image from it, then display it to the user | Image sent to the backend API | Object in the image will be successfully identified and displayed to the user | Pass |

Unit Testing 29: Opening side drawer

Testing Objective: To ensure that the side drawer is opened and is displayed to the user correctly.

Table 210: Test case 29

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|---------------------------------------|---|--------|
| 1. | From the home screen, tap on the hamburger icon button, that can be found on the top left corner of the screen, to open the side drawer | Tapping on the ham burger icon button | Side drawer will be opened as panel on the left-hand side of the screen | Pass |

Unit Testing 30: Opening account settings from side drawer

Testing Objective: To ensure that the account settings are being opening from side drawer correctly.

Table 211: Test case 30

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|--|---|--------|
| 1. | From the side drawer, tap on the 'Account Settings' text hyperlink | Tapping on the 'Account Settings' text | User will be redirected the Account Settings Screen | Pass |

Unit Testing 31: Opening application settings from side drawer

Testing Objective: To ensure that the application settings are being opening from side drawer correctly.

Table 212: Test case 31

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|--|---|--------|
| 1. | From the side drawer, tap on the 'Application Settings' text hyperlink | Tapping on the 'Application Settings' text | User will be redirected the Application Settings Screen | Pass |

Unit Testing 32: Logging out from application through side drawer

Testing Objective: To ensure that the user is logging out from application through side drawer properly.

Table 213: Test case 32

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|------------------------------|--|--------|
| 1. | From the side drawer, tap on the ‘Logout’ text hyperlink | Tapping on the ‘Logout’ text | User will successfully logout from the application | Pass |

Unit Testing 33: View suggestion from side drawer

Testing Objective: To ensure that the user can view suggestions from side drawer correctly.

Table 214: Test case 33

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|--|---|--------|
| 1. | From the side drawer, tap on the ‘View Suggestions’ text hyperlink | Tapping on the ‘View Suggestions’ text | User will be redirected to the Suggestions Screen | Pass |

Unit Testing 34: Change application language from side drawer

Testing Objective: To ensure that the user can change application language from side drawer correctly.

Table 215: Test case 34

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|---|---|--|--------|
| 1. | From the side drawer, tap on the ‘Change Application Language’ text hyperlink | Tapping on the ‘Change Application Language’ text | User will be redirected the language changing screen | Pass |

Unit Testing 35: Change application settings from side drawer

Testing Objective: To ensure that the user can change application settings from side drawer correctly.

Table 216: Test case 35

| No. | Test Case/Test Script | Attribute and value | Expected Result | Result |
|-----|--|--------------------------------|--|--------|
| 1. | From the side drawer, tap on the ‘Settings’ text hyperlink | Tapping on the ‘Settings’ text | User will be redirected the settings changing screen | Pass |

6.2 Functional Testing

Functional Testing 1: Testing the process of exploring restaurants

Objective: To ensure that the corresponding data of restaurants successfully loads onto the relative screen throughout the process

Table 217: Functional Testing 01

| No. | Test case/Test script | Attribute and value | Expected result | Actual result | Result |
|-----|---|---|--|---|--------|
| 1. | Opening restaurant section | Tapping on the restaurant block from home screen | Restaurant screen opens up with restaurant data | Restaurant screen opened correctly, and the data of all restaurants was visible | Pass |
| 2. | Changing restaurant location | Tapping on the location bar on top of restaurant screen | A list drops down containing various locations throughout the country | The drop-down list was visible and different locations were seen in it | Pass |
| 3. | Selecting a location for restaurant results | Tapping on one of the restaurant locations from the visible list | The resulting restaurants are now from the selected location | The restaurants visible on the screen were now the restaurants from the selected restaurants | Pass |
| 4. | Searching a restaurant | Tapping on the search bar on top of the restaurant screen just below the location bar | The search bar is now in active state and is ready to accept text in it, and the keyboard pops up | The search is in active state now and user can type restaurant name in it using the appeared keyboard | Pass |
| 5. | Typing name of a restaurant | Typing text into the search bar that's associated with a restaurant name | The search results are filtered dynamically as the user types more and more text into the search bar | Dynamic search is working as expected and the results are filtered out with each and every input | Pass |
| 6. | Selecting a restaurant | Tapping on one of the restaurants form the appeared list of results | The selected restaurant's screen will open, holding the data of that restaurant | The corresponding restaurant screen opens up with appropriate data | Pass |

| | | | | | |
|-----|-------------------------------------|---|--|--|------|
| 7. | View restaurant information/details | Tapping on the 'About' tab on the restaurant screen | The about screen of this restaurant loads up showing the restaurant opening and closing time, restaurant name, rating, restaurant banner, and restaurant's complete address | The mentioned screen successfully appears on the application, holding all of the expected data in it | Pass |
| 8. | View restaurant menu | Tapping on the 'Food' tab on the restaurant screen | The food screen of this restaurant loads up showing the complete menu, including the food name, food image, and the food price | The mentioned screen is successfully loaded in the application, along with the specified data | Pass |
| 9. | View restaurant reviews | Tapping on the 'Reviews' tab on the restaurant screen | The reviews screen of this restaurant loads up showing all of the reviews of this restaurant's reviews, including the reviewer's name, date of the review, feedback comment posted in the review, and the rating given | The mentioned screen is successfully loaded onto the screen with the appropriate data for the reviews of this restaurant | Pass |
| 10. | Post a review | From the restaurant's reviews section, tapping on the 'Post Review' button, write your review and tap on 'Post Review' button again | Take rating from the user in 5 stars format and a feedback comment in the text field provided, then posting this review among the already posted reviews | The review given by the user is successfully posted on the system and can be viewed among the other reviews, posted by other users | Pass |

Functional Testing 2: Testing the process of exploring workout plans

Objective: To ensure that the corresponding data of workout plans successfully loads onto the relative screen throughout the process

Table 218: Functional Testing 02

| No. | Test case/Test script | Attribute and value | Expected result | Actual result | Result |
|-----|--------------------------------------|---|--|--|--------|
| 1. | Opening workout section | Tapping on the ‘Workout’ block from home screen | Workout screen opens up, containing different categories of workout plans | Workout screen opens up, containing different categories of workout plans | Pass |
| 2. | Exploring home based workout plan | Tapping on the section stating ‘Home Based’ | User is redirected to a new screen, for the home-based workout category | User is redirected to a new screen, for the home-based workout category | Pass |
| 3. | Selecting day of workout | Tapping on any day from the list of days | User is redirected to a new screen where he is shown the exercises to be performed on the selected day | User is redirected to a new screen where he is shown the exercises to be performed on the selected day | Pass |
| 4. | Exploring build muscles workout plan | Tapping on the section stating ‘Build Muscles’ | User is redirected to a new screen, for the muscles-building workout category | User is redirected to a new screen, for the muscles-building workout category | Pass |
| 5. | Selecting targeted body part | Tapping on the part of the body from the provided options | User is redirected to a new screen, where he can view the exercises focused on the muscles of the selected body part | User is redirected to a new screen, where he can view the exercises focused on the muscles of the selected body part | Pass |
| 6. | Exploring weight losing workout plan | Tapping on the section stating ‘Lose Weight’ | User is redirected to a new screen, for the weight losing workout category | User is redirected to a new screen, for the weight losing workout category | Pass |

| | | | | | |
|----|--------------------------|---|--|--|------|
| 7. | Select circuit type | Tapping on one of the provided circuits available in this category of workout | User will be redirected to the corresponding screen on which he is shown the related exercises belonging to the selected circuit | User will be redirected to the corresponding screen on which he is shown the related exercises belonging to the selected circuit | Pass |
| 8. | View set of exercises | Tapping on any one of the exercise types of any category | User is displayed with many exercises on his screen, where he can view exercise name, exercise visual demonstration, exercise circuits, and exercise repetitions for each of the exercise | User is displayed with many exercises on his screen, where he can view exercise name, exercise visual demonstration, exercise circuits, and exercise repetitions for each of the exercise | Pass |
| 9. | Viewing exercise details | Tapping on any exercise from the exercise set | User will be able to view complete details of the selected exercise. These details include exercise name, exercise visual demonstration, and the complete steps and guidelines for performing this exercise, including the postures and positionings | User will be able to view complete details of the selected exercise. These details include exercise name, exercise visual demonstration, and the complete steps and guidelines for performing this exercise, including the postures and positionings | Pass |

Functional Testing 3: Testing the process of exploring diet plans

Objective: To ensure that the corresponding data of diet plans successfully loads onto the relative screen throughout the process

Table 219: Functional Testing 03

| No. | Test case/Test script | Attribute and value | Expected result | Actual result | Result |
|-----|-------------------------------------|--|---|--|--------|
| 1. | Opening diet plans section | Tapping on the ‘Diet Plans’ block from home screen | Diet plans screen opens up, containing different categories of diet plans | Diet plans screen opens up, containing different categories of diet plans | Pass |
| 2. | Explore diet for under weight | Tapping on the diet plan stating ‘Diet for Under Weight’ | User is redirected to a new screen, for the under-weight diet category | User is redirected to a new screen, for the under-weight diet category | Pass |
| 3. | Explore diet for maintaining weight | Tapping on the diet plan stating ‘Diet to Maintain Weight’ | User is redirected to a new screen, for the weight maintaining diet category | User is redirected to a new screen, for the weight maintaining diet category | Pass |
| 4. | Explore diet for overweight | Tapping on the diet plan stating ‘Diet for Overweight’ | User is redirected to a new screen, for the overweight diet category | User is redirected to a new screen, for the overweight diet category | Pass |
| 5. | Selecting day of week for diet | Tapping on any of the day of week from the list | User is redirected to a new screen corresponding to the weekday | User is redirected to a new screen corresponding to the day of week | Pass |
| 6. | View meal plan details | Tapping on any of the day of week from the list | On this screen for the said day, user can view the meal category, mealtime duration, and the contents to eat in this meal | On this screen for the said day, user can view the meal category, mealtime duration, and the contents to eat | Pass |
| 7. | Calculate body mass index | Tapping on the block stating, ‘Calculate BMI’ and entering the asked details | User will be able to calculate and view his BMI and will be suggested with a diet plan | User will be able to calculate and view his BMI and is suggested with a diet plan | Pass |

Functional Testing 4: Testing the process of exploring food products

Objective: To ensure that the corresponding data of food products successfully loads onto the relative screen throughout the process

Table 220: Functional Testing 04

| No. | Test case/Test script | Attribute and value | Expected result | Actual result | Result |
|-----|--|--|---|--|--------|
| 1. | Initiating the process for exploring food products | Tapping on the scanning icon button from the bottom of the home screen | User is redirected to a new screen where he is asked for the permission to grant camera access | User is redirected to a new screen where he is asked for the permission to grant camera access | Pass |
| 2. | Granting camera access permission | Tapping on the ‘While using app’ button from the permission dialog | Being the permission granted, the camera of the smartphone is now functional for usage | Being the permission granted, the camera of the smartphone is now functional for usage | Pass |
| 3. | Denying camera access permission | Tapping on the ‘Deny’ button from the permission dialog | Being the permission denied, the camera of the smartphone will not be accessed and used by the system | Being the permission denied, the camera of the smartphone is not accessed and used by the system | Pass |
| 4. | Capturing image of food item | Tapping on the shutter icon from the bottom center of the camera screen | Image of the food item is successfully captured and is ready for processing | Image of the food item is successfully captured and is ready | Pass |
| 5. | Opening smartphone gallery to select an image | Tapping on the gallery icon button from the bottom right corner of the camera screen | User is redirected to a new screen where he is asked for the permission to grant storage access | User is redirected to a new screen where he is asked for the permission to grant storage access | Pass |
| 6. | Granting storage access permission | Tapping on the ‘While using app’ button from the permission dialog | Being the permission granted, the gallery of the smartphone will be opened for usage | Being the permission granted, the gallery is opened on screen | Pass |

| | | | | | |
|-----|----------------------------------|--|---|---|------|
| 7. | Denying camera access permission | Tapping on the ‘Deny’ button from the permission dialog | Being the permission denied, the storage of the smartphone will not be accessed, and gallery will not be opened | Being the permission denied, the storage of the smartphone is not accessed, and gallery is not opened | Pass |
| 8. | Selecting image of food item | Tapping on any of the image from the gallery opened on the screen | Selected image successfully loads up on full screen for further operations | Selected image successfully loads up on full screen for further operations | Pass |
| 9. | Crop selected image | Scaling up/down the grid box over the image to place the food item in the center of vision | The image selected will be cropped according to the resized grid and the cropped image part will be considered only | The image selected is cropped according to the resized grid and the cropped image part is considered only | Pass |
| 10. | Finalize image | Tapping on the tick mark from the top right corner of the cropping screen | The selected image, after cropping, is grasped, and forwarded to the system for processing | The selected image, after cropping, is grasped, and forwarded to the system for processing | Pass |
| 11. | View food details | Processing the selected image | After successful processing, user is redirected to a new screen where he can view that food name, food image, food types, food ingredients, food nutritional facts values, and the recipe of that food product. | After successful processing, user is redirected to a new screen where he can view that food name, food image, food types, food ingredients, food nutritional facts values, and the recipe of that food product. | Pass |

Functional Testing 5: Testing the process of changing system's language

Objective: To ensure that the system successfully loads up with the selected language throughout the application

Table 221: Functional Testing 05

| No. | Test case/Test script | Attribute and value | Expected result | Actual result | Result |
|-----|-------------------------|--|---|--|--------|
| 1. | Open side drawer panel | Tapping on the hamburger icon from the top left corner of the home screen | Side drawer panel is animated from the left-hand side of the screen, on to the home screen of the application | Side drawer panel is animated from the left-hand side of the screen, on the home screen of the application | Pass |
| 2. | Go to language settings | Tapping on the option stating 'Change Language' from the list of options in the side drawer menu | User will be redirected to another screen where he can view the different languages in which he can use the app | User is redirected to another screen where he views the different languages in which he can use the app | Pass |
| 3. | Select Italian Language | Tap on the block stating 'Italian' | The language off the entire application will be instantly changed to Italian language | The language off the entire application is instantly changed to Italian language | Pass |
| 4. | Select French Language | Tap on the block stating 'French' | The language off the entire application will be instantly changed to French language | The language off the entire application is instantly changed to French language | Pass |
| 5. | Select German Language | Tap on the block stating 'German' | The language off the entire application will be instantly changed to German language | The language off the entire application is instantly changed to German language | Pass |
| 6. | Select Spanish Language | Tap on the block stating 'Spanish' | The language off the entire application will be instantly changed to Spanish language | The language off the entire application is instantly changed to Spanish language | Pass |

6.3 Integration Testing

Table 221: Integration Testing

| No. | Test Case/ Test script | Attribute and value | Expected Result | Actual Result | Result |
|-----|---|---|--|------------------|--------|
| 1. | Login as mobile application user | Email: khokhar@gmail.com Password: 1234abcd | User is successfully logged in the system and is redirected to the mobile application's home screen | As Expected | Pass |
| 2. | View different cities in which Food Explorer is offering services | From the home screen, tap on the section that states 'Cities' | User is successfully redirected to the corresponding screen on which he can view all of the available cities | As Expected | Pass |
| 3. | View all restaurants of a city | From the cities screen, tap on any one of the available cities in the list | User is successfully redirected to the corresponding screen on which he can view all of the available restaurants of selected city | As Expected | Pass |
| 4. | Search a restaurant | From the restaurant screen, tap on the search bar on top of the screen and enter name of a restaurant | Results are loaded dynamically as the user continues typing, and at the end, sees his corresponding search results | As Expected | Pass |
| 5. | View menu offered by a restaurant | From the restaurant screen, tap on any one of the available restaurants in the list | User is successfully redirected to the corresponding screen on which he views the complete menu along with price list of the selected restaurant | As Expected | Pass |

| | | | | | |
|-----|--|---|---|-------------|------|
| 6. | View details of restaurant itself | From a restaurant's screen, tap on the tab with the title of 'About' | User views the complete details of the restaurant on the loaded section | As Expected | Pass |
| 7. | View feedbacks of a restaurant | From a restaurant's screen, tap on the tab with the title of 'Reviews' | User views all of the feedbacks posted against that restaurant on the loaded section | As Expected | Pass |
| 8. | Post review on restaurant | From the review tab of the restaurant screen, tap on post review | User will provide rating, and a feedback comment in the provided fields and hit on the post review button to add his review | As Expected | Pass |
| 9. | View workout plans | From home screen, tap on the section that states 'Workout Plans' | User is successfully redirected to the corresponding screen where he views different categories of workouts | As Expected | Pass |
| 10. | View different exercise types against a workout plan | From the workout screen, tap on any one of the workout categories available in the list | User is successfully redirected to the corresponding screen where he views exercise types of the selected plan | As Expected | Pass |
| 11. | View exercises details of a workout category | From the exercise type screen, tap on any one of the exercise types available in the list | User is successfully redirected to the corresponding screen where he views the complete details and demonstration of the selected exercise type | As Expected | Pass |
| 12. | View diet plans | From home screen, tap on the section that states 'Diet Plans' | User is successfully redirected to the corresponding screen where he views different categories of diet | As Expected | Pass |

| | | | | | |
|-----|---------------------------------------|---|---|-------------|------|
| 13. | View meal plans | From the diet plans screen, tap on any one of the diet categories available in the list | User is successfully redirected to the corresponding screen where he views the complete meal plan division of the selected diet plan | As Expected | Pass |
| 14. | Explore food products | From the home screen, tap on the camera icon button from bottom center of the screen | Camera of smartphone will be opened after granting the camera access permission by the user | As Expected | Pass |
| 15. | Capture food picture | Tap on the shutter icon button from the bottom center of camera screen | Image of the food item is captured and transferred to the system for processing and identification | As Expected | Pass |
| 16. | Open smartphone gallery | Tap on the gallery icon button from the bottom right corner of the camera screen | Gallery of the smartphone will be opened after granting the storage access permission by the user | As Expected | Pass |
| 17. | Selecting an image for identification | From the smartphone gallery, tap on an image | User will be able to send the selected image to the system for processing after he crops and finalize the image that he selected from his smartphone gallery | As Expected | Pass |
| 18. | View food details | Getting an image from the user for the system by any of the two means defined above | User will be able to successfully view all of the details about the searched food item on the redirected screen. These details include food name, image, ingredients, types, recipes, and nutrition values. | As Expected | Pass |

7 Conclusion and Future Work

7.1 Conclusion

The food explorer will be assisting the general public and helping the community to stay healthy, content, fit, and social. Guiding them out in an unknown place and situation; fulfilling the dire necessity of mankind. An easy to use, central platform is provided to the people for their convenience on daily basis, hence opening up a new facilitating ground in this era of advancement and technology. Food Explorer serves as the complete guide to the audience having trouble in opting their daily meal over a new unfamiliar place. The system assists in searching good food in their nearby places and guiding them out to those places. It helps users to get quality food and good restaurants with less effort, along with their ratings and reviews and bunch of other useful information. Furthermore, users can focus on their fitness and health by opting suitable workouts and supportive diet plans at the same time.

7.2 Future Work

In the future, the system admin can enlarge the scale of the system however he likes. The admin can do this by increasing the numbers of servicing cities, identifiable food products, and supported languages. Currently the count for each of the mentioned entity is 5. The servicing cities are currently Faisalabad, Islamabad, Karachi, Lahore , and Rawalpindi. The identifiable food items are currently biryani, burger, fries, pizza, and steak. The currently supported languages are English, French, German, Italian, and Spanish.

8 References

- [1] Samsung, "Bixby," 2020. [Online]. Available: <https://www.samsung.com/us/explore/bixby/>.
- [2] K. Strick, "Tech," Evening Standard, 26 April 2018. [Online]. Available: <https://www.standard.co.uk/tech/hack-snacks-samsung-galaxy-s9-smartphone-calorie-counter-a3824491.html>.
- [3] I. Kundu, "India Today," News, 28 June 2020. [Online]. Available: <https://www.indiatoday.in/india/story/kolkata-zomato-delivery-boys-quit-job-protesting-chinese-investment-in-company-1694690-2020-06-27>.
- [4] Foodpanda, Foodpanda, 2020. [Online]. Available: <https://www.foodpanda.pk/>.
- [5] Foursquare, Foursquare, 2020. [Online]. Available: <https://foursquare.com/>.
- [6] B. H, "Yelp," Yelp, 2020. [Online]. Available: <https://www.yelp.com>.
- [7] "Data Security," Looker, [Online]. Available: <https://looker.com/definitions/data-security>.
- [8] Google, "Cloud Firestore Documentaion," 14 May 2021. [Online]. Available: <https://firebase.google.com/docs/firestore>.