Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Project Report

on

"nutri.gram"

[Course Code: COMP 308]

(For partial fulfillment of III Year/ II Semester in Computer Engineering)

Submitted By:

Sailesh Dahal (10)

Sarayu Gautam (14)

Bhabin Khadka (23)

Ashish Subedi (53)

Gaurav Singh Thagunna (55)

Submitted To

Dr. Gajendra Sharma

Department of Computer Science and Engineering

Submission Date

September 28, 2020

Bonafide Certificate

This project work on

"**nutri.gram**"

is the bonafide work of

"Sailesh Dahal, Sarayu Gautam, Bhabin Khadka, Ashish Subedi, and Gaurav Singh Thagunnna"

who carried out the project work under my supervision.

Project Supervisor

Mrs. Deni Shahi

Lecturer

Department of Computer Science and Engineering

Kathmandu University

Acknowledgment

In performing our project, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this project gives us much pleasure. We would like to show our gratitude to **Mrs. Deni Shahi, Lecturer, Kathmandu University** for giving us a good guideline for this project throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in doing this project.

We would like to thank the Department of Computer Science and Engineering (DoCSE) and the whole university for providing us a chance to work on the project. Many people, especially our classmates and team members themself, have made valuable comments and suggestions on this project which inspired us to improve our project.

Sincerely,
Sailesh Dahal (10)
Sarayu Gautam (14)
Bhabin Khadka (23)
Ashish Subedi (53)
Gaurav Singh Thagunna (55)

Abstract

From the fact which connotes the value of nutrition in our diet, springs the idea of "nutri.gram". "nutri.gram" is a mobile application that scans the food label present in the packaged food items and draws a significant conclusion from that data regarding the nutritional value any food adds to our diet, how much calorific value it provides per serving, and which nutrients are present in that food. Our project uses Flutter as a front-end framework, which uses Dart as a programming language and Node.js as server-side JavaScript runtime, MongoDB as a database, and Python for Machine Learning and Computer Vision algorithms to scan the items. We have used OpenCV as a tool for working with the data. This application functions with the information provided through the scanned image of packaged food. The main objective of this project is to provide a handy guideline for the people who are diet conscious and want to visualize the nutrients present in the food they eat. This application is targeted to anyone who wants to get insight into what they are feeding into their body. After 3 months of coding, we have completed the project. The result is a cross-platform mobile application that scans a packaged food item, displays its calorie and nutritional information, visualizes that information through graphs, maintains user profile and scan history, provides a search feature to search for information on the particular food item, and provides health tips to the users.

Keywords: *Nutrition, Flutter, Node.js, Python, Machine Learning, Computer Vision, Image Processing*

List of Figures

List of Tables

Abbreviations

Short-form	**Full form**
API	Application Programming Interface
BSD	Berkeley Software Distribution
CV	Computer Vision
DB	Database
DI	Dependency Injection
DoCSE	Department of Computer Science and Engineering
ER	Entity Relation
FFI	Foreign Function Interface
GB	Gigabytes
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
JS	JavaScript
JSON	JavaScript Object Notation
JWT	JSON Web Token
ML	Machine Learning
NoSQL	No Structured Query Language
NPM	Node Package Manager
OCR	Optical Character Recognition
ODM	Object Document Mapper
OS	Operating System
OTP	One Time Password
RAM	Random Access memory
RGB	Red Green Blue
SDK	Software Development Kit
UI	User Interface
UML	Unified Modeling Language
UX	User Experience
VPS	Virtual Private Server

WSGI	Web Server Gateway Interface

Table of Contents

Chapter 1: Introduction

This chapter discusses the background, objective, and motivation related to this project.

1.1. Background

Nutrition is the science that interprets the nutrients and other substances in food about an organism's maintenance, development, reproduction, safety, and disease. The seven main nutrient groups are carbohydrates, fats, starch, minerals, proteins, vitamins, and water. We need all kinds of nutrition to function properly and for our immunity. These days we are more dependent on packaged foods and it is natural to worry about the calorie and nutrients we are getting from those foods. Nutrition is how food affects the health of the body. When you become aware of the value any food adds to your health, you start eating healthy. You start selecting that food, even if packaged, which provides an adequate number of calories and nutrients your body requires.

Hence, we decided to build a smartphone application “nutri.gram”. The application provides an interface so that users can scan the wrapper of packed foods and get relevant information in the app regarding the nutrient and calorie content of that food product. Among the food products the user searches, the algorithm generates effective information and provides a visual display to the users incorporating various features available through the analysis of data. Moreover, this application is designed with the target of helping any diet/ health-conscious people but, anyone trying to know the nutrition/calorie present in the packaged food can use this application for their benefit.

1.2. Objectives

The main objectives of this project are:

1. To enable users to get information about the nutrients and calories they are getting from the packaged food.
2. To display graphs on their calorie and nutrition consumption with respect to time so they can easily track their diet.
3. To keep the history of the packaged food they have scanned till now along with the content present in those diets.
4. To provide health tips to better enhance the grasp of people on what type of food benefits them.
5. To provide a search feature that allows users to search the food that they want to get information about without scanning its package.

1.3. Motivation and Significance

Every one of us requires nutrients to survive. Due to the increasing business mindset, packaged foods come with more processing and less fiber. On the contrary, due to our busy schedule, we are more and more allured towards canned and packaged foods than homemade and natural ones. Sometimes it is not our choice but the compulsion to be dependent or to choose packaged food. But like a diet conscious person, one desires to know the nutrients present in the food he/she is feeding. One may want to know which food is consumed by other people having the same medical condition or similar dietary preferences. People may take fancy on knowing the calorie content of their packaged diet, or how much calories are sufficient for them as per their body structure (weight and height), and also how is the nutrient distributed in the food they eat (is it a balanced diet). To solve these issues, **nutri.gram** application is planned. We were inspired by the importance of nutrition and wanted to implement it uniquely. After the target was fixed, we started a discussion on its design, layout, features, and components. When we did some research on this idea and found that it had a notable potential, our team became confident that our idea of the project was worthy of acceptance and immediate response towards its actual development.

Chapter 2: Related Works

As the research for the project proceeded further, we went on to find some projects with goals like our own. Three such projects stuck out as noteworthy each of which is described in brief below.

2.1. MyPlate Calorie Tracker [1]

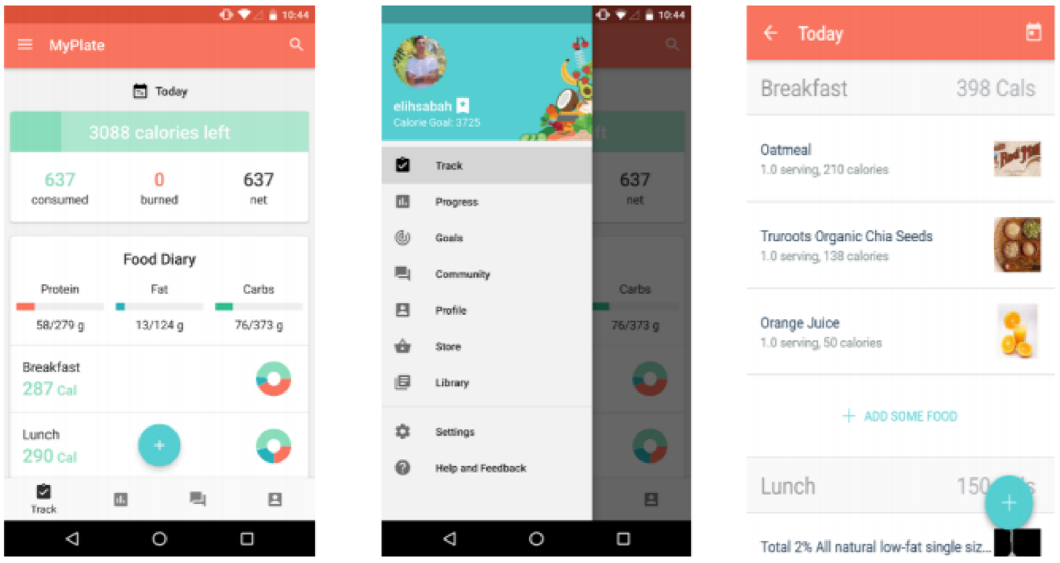


Figure 2.1.1: MyPlate Calorie Tracker

MyPlate Calorie Tracker [1], an app developed by livestrong.com, is a user-friendly app designed to help you lose weight and improve your health. This app offers easy-to-use nutritional facts, as well as personalized daily calorie goals, healthy meal plans, a barcode scanner, an extensive food database, and detailed statistics about your nutrition. This app is available on both iOS and Android. Features of this app are:

- Browse a comprehensive food database with over 2 million food items.
- Uses barcode scanners to find and track food easily
- Get a personalized daily calorie goal based on your profile information
- Keep track of weight and progress over time
- Review custom goals for your nutritional intake of protein, fiber, carbs, etc.

2.2. MyFitnessPal [2] [3]

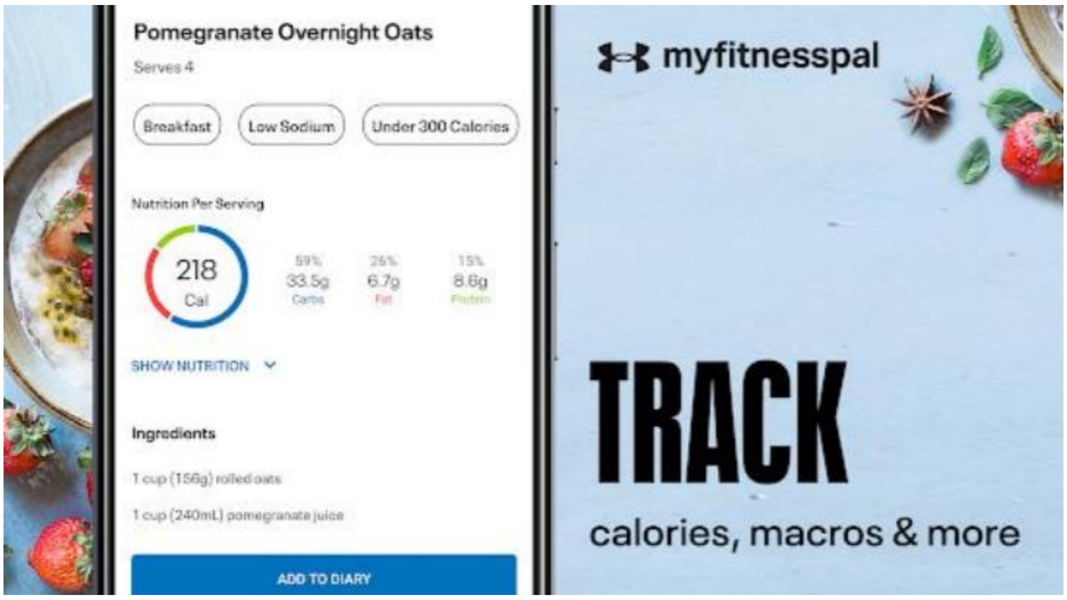


Figure 2.2.1: MyFitnessPal

MyFitnessPal [2] [3] is a smartphone app and website that tracks diet and exercise. The app uses gamification elements to motivate users. To track nutrients, users can either scan the barcodes of various food items or manually find them in the app's large pre-existing database.

The key features of MyFitnessPal are:

- Track All Nutrients like Calories, macros (carbs, fat, protein), sugar, fiber, cholesterol, vitamins, and more.
- With 11+ million foods in their database including global items and cuisines.
- Import the nutrition information for the recipes cooked.

2.3. Open Food Facts – Scan Food [4]

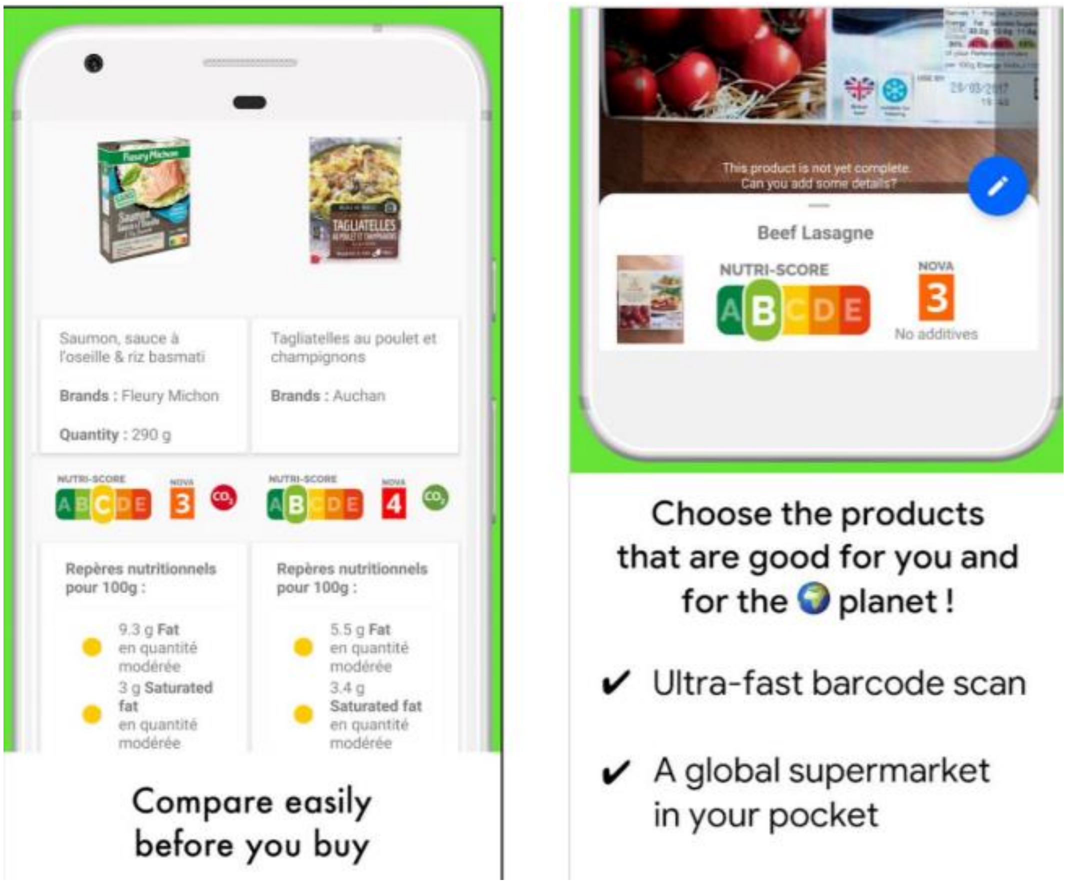

Saumon, sauce à l'oseille & riz basmati
Brands : Fleury Michon
Quantity : 290 g
Tagliatelles au poulet et champignons
Brands : Auchan
NUTRI-SCORE
NOVA
Repères nutritionnels pour 100g :
9.3 g Fat en quantité modérée
3 g Saturated fat en quantité modérée
5.5 g Fat en quantité modérée
3.4 g Saturated fat en quantité modérée
Compare easily before you buy
This product is not yet complete. Can you add some details?
Beef Lasagne
NUTRI-SCORE
NOVA
No additives
Choose the products that are good for you and for the planet !
Ultra-fast barcode scan
A global supermarket in your pocket

Figure 2.3.1: Open Food Facts- Scan food

Open Food Facts [4] is an open-source app developed by Open Food Facts. This app allows users to scan products contained in their database. In their words, "We're kind of the Wikipedia of food." It is available on both Android and iOS. It helps to choose the products good for users, contribute to food transparency, and to get the facts about food items.

The key features of Open food Facts are:

- Get facts about nutrients in common food items.
- Get Nutri-Score grade, from "A" to "E" determining the nutritional quality
- Find the carbon footprint and recycling instructions.

Although apps similar to ours are previously made, they are different in some aspects with our apps. MyPlate and MyFitnessPal deal with cooked foods, while our app is for packaged food. Though Open Food Facts - Scan Food has a feature to scan app barcodes, it doesn’t have a feature to scan food labels, which our app features machine learning and computer vision to recognize the labels with nutrients, which saves users effort to put every nutrients entry manually. Also, our app provides recommended diets from user-selected nutrients, which other users use for similar use.

Chapter 3: Design and Implementation

This chapter includes an explanation of the sequential procedure that we have performed during the project. It includes algorithms, flowcharts, system diagrams, and other analysis and design which gives a general idea about how we have approached different development procedures during the project.

3.1. Procedure

The procedure that we followed throughout the completion of this application is enlisted below:

1. Planned and designed the application along with the consideration of the required tech stack Flutter, JavaScript runtime environment: Node.js, MongoDB, OpenCV, tesseract, Python, and Flask.
2. Designed UI in Figma incorporating every page needed in the app.
3. Started simultaneously with frontend (Flutter SDK) and backend (both node server and ML server).
4. Added user authentication and displayed information per the data provided by the user to make the application user-specific.
5. Communicated frontend with backend (Flutter with node server and node with flask server).
6. Tested and debugged to avoid errors present.

As every project requires a certain level of planning, we started our project with proper planning and its design. Besides, we have selected some robust languages for development. After the completion of the prototype of our application, we have added features according to the needs of the user. Eventually, we tested and debugged the application for its implementation in the real-world scenario. **nutri.gram** application is built upon the following principles, components, and processes:

3.1.1. Image Processing [5]

3.1.1.1. Preprocessing

Preprocessing data is a crucial part when it comes to image processing. The objective of the pre-processing stage is to transform the input image into a simple, clear, and formatted output. We used tools from OpenCV [6] open-source library for this process. The preprocessing stage can be broken down into 6 stages listed below:

3.1.1.2. Resizing

The input image is resized into a height of 600px preserving its aspect ratio.

3.1.1.3. Greying and Blurring

The next step is to convert the image from RGB to grayscale as it is easier and faster to perform different operations on the grayscale image. Once the image has been greyed, we apply a Gaussian filter to smooth out any outlying pixel values.

3.1.1.4. Histogram Equalization [7]

After blurring, we apply histogram equalization to increase the contrast among different intensities within the image. This process helps to darken the borderlines and create noticeable distinction against lighter intensity background. We then apply a bilateral filter to create more uniform pixel regions. A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images.

3.1.1.5. Contour Identification

After histogram equalization, we implement the contour identification process to find the shapes present in the image. We filter out lots of contours detected during this process based on their size and location. As the label takes the majority of the image, we find external contour with 4 vertices that find the nutrition label.

3.1.1.6. Image Thresholding [8]

The extracted contour region of the image is thresholded using the adaptive Gaussian thresholding algorithm. The algorithm determines the threshold for a pixel-based on a small region around it. So, we get different thresholds for different regions of the same image which gives better results for images with varying illumination. This creates a binary image (black or white color only) which is then fed into a label extractor.

3.1.1.7. Perspective Transformation

The image clicked by the user might be skewed or wrapped. Out label extractor cannot interpret the skewed image as easily. So to fix this issue, the image has to be transformed to align the image properly. The result of this process is a vertically oriented rectangular image of just a nutrition label with a white background and black text.

3.1.1.8. Label Extractor

In this stage, the preprocessed image is fed into the OCR engine to get the output as text. For this, we use the Tesseract OCR [9] engine, which is an open-source engine. This takes input as an image and gives text as output. From this, we extract the text from the nutrition label and feed it into the post-processor for further processing and correction of errors.

3.1.1.9. Post-Processing

The objective of this stage is to identify and correct any errors present in the Tesseract output. Following are the stages of post-processing we have performed:

3.1.1.10. Decomposition of labels and values

In this step, we find the label name as well as values. As we know that nutrition labels usually consist of letters before a number is found and the value is the number in the line, we use this information to extract the label and values as well as the unit which usually comes after the value. We create a key-value pair of this information.

3.1.1.11. Keyword Matching

We determine the mapping of the keys extracted to the nutrients that are in the database. For this, we calculate the Levenshtein distance [10] between the keyword and database nutrients and select the closest matching nutrient. The Levenshtein distance [10] between two words is the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into the other.

3.1.1.12. Formatting Data

After we get all the required values, we format the data as JSON in an appropriate format that the API can send to the application.

3.1.2. Edge Detection in Flutter [11]

Edge detection is a very common task in various applications that deal with scanning of images. We have implemented an edge detection algorithm using OpenCV [6] C++ library by calling it from Dart code using a bridge called Dart FFI [12]. This lets us use the phone's camera to take a picture or choose one from the gallery, and then displays the detected edges on the picture. Since flutter is cross-platform, this feature is fully functional in both Android and iOS.

The picture taken or chosen by the user has a certain path that is forwarded to the C/C++ bridge. Then through this bridge, we call a function in OpenCV [6] to detect edges that will be returned to our Dart code. The detected edge coordinate is used to render a quadrilateral on top of the image. The rendered quadrilateral is resizable to the user’s preference if the algorithm fails to detect the edges.

3.1.3. Dependency Injection [13]

Dependency Injection [13] is a design pattern used to implement Inversion of Control. It allows the creation of dependent objects outside of the class and provides those objects to a class in different ways. Using DI, we move the creation and binding of the dependent objects of the class that depends on them. This brings a higher level of flexibility, decoupling, and easier testing.
In Flutter, the default way to provide objects/ services to widget is through InheritedWidgets. We have used dependency injection to inject the dependency of services to the implementation class. The services are used to provide the functionality to the class independent of the libraries used.
For example, Scan view uses a CameraService which in turn uses a camera package. If we need to change the camera package due to depreciation or some other reasons, we would just be changing the implementation in the service class. The ScanView is provided with the CameraService's instance using DI.

3.1.4. Stacked Architecture [14]

This is an architecture used in developing our frontend application. It mainly contains 3 parts. Views, ViewModels, and Services.
Views on top, closest to the user, ViewModels are below that taking input from views and Services below that which is what ViewModels make use of to provide functionality. This architecture provides an easy way to manage state, handle page navigation, and inversion of control. Whenever the state in the ViewModel changes, the view reacts with it resulting in an update of the view. The view has 0 business logic involved. The role of view is just to react upon the state.
The ViewModel makes use of the services. For example, ScanView has a ScanView model, which makes use of CameraService to capture, detect edges, upload, save, and all other business logic. Every ViewModel which requires navigation uses a NavigationService to navigate to and from a specific view.

3.1.5. Express Server [15]

Express [15] is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. With the help of express HTTP utility methods and middlewares, we can create a robust API that will be connected to the front-end application.

This Express server is used to handle all the incoming HTTP requests, for user authentication, MongoDB connectivity, sending image to the ML server and receiving the image scan result, and then for sending the scan result to the frontend application. The express server is also used to make database queries using Mongoose ODM [16].

3.2. ER Diagram

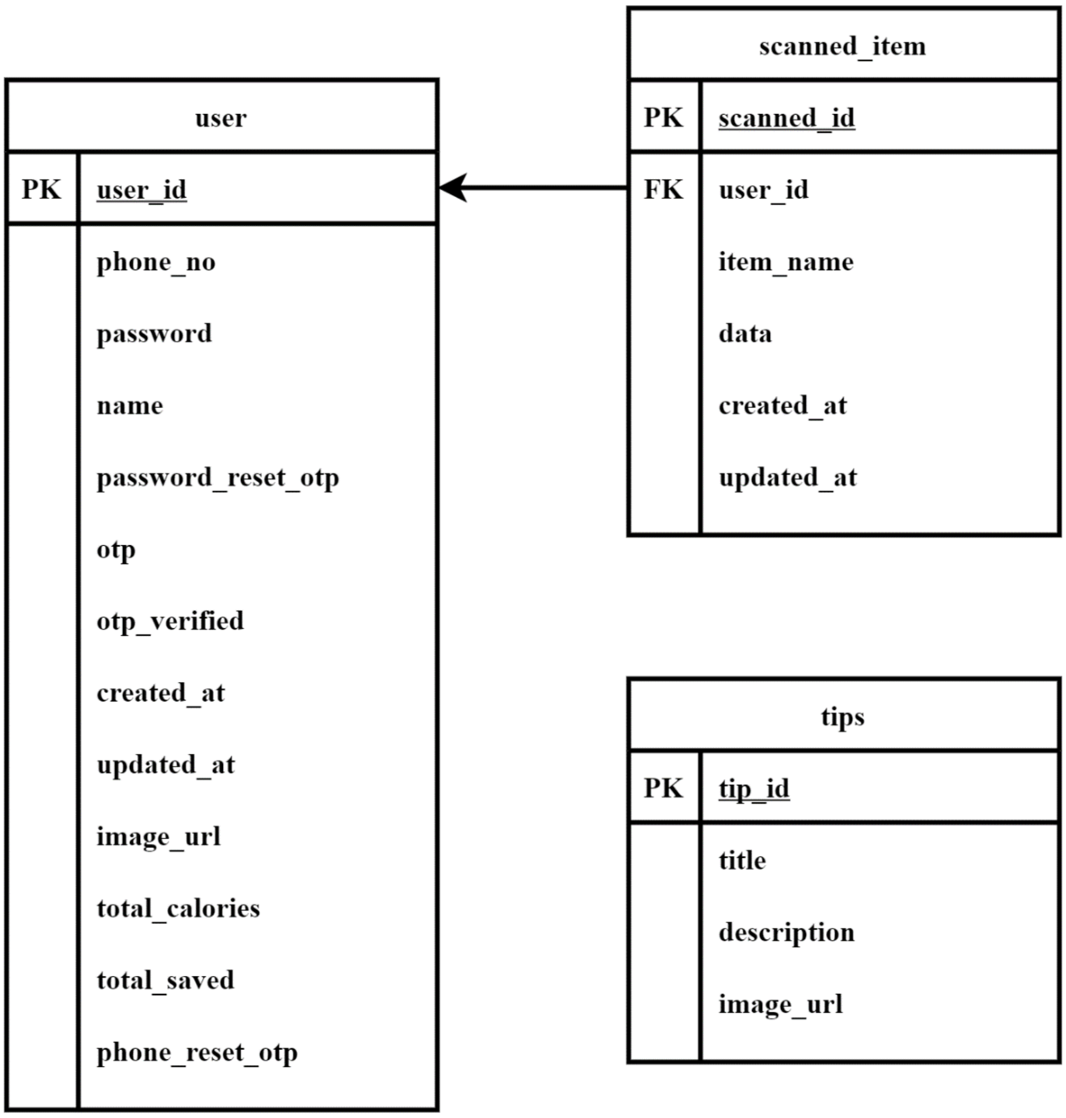


Figure 3.2.1: ER Diagram

3.3. Flowchart

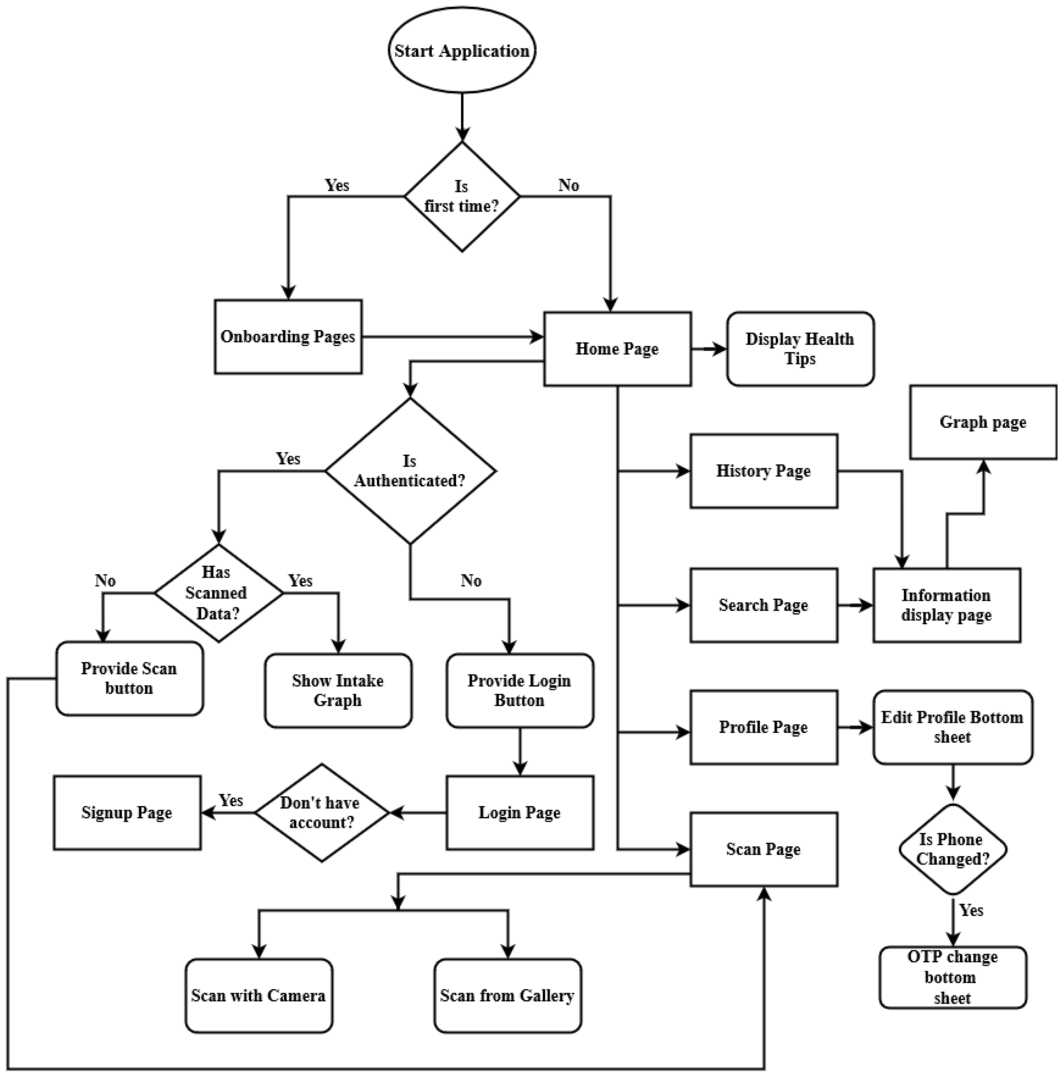


Figure 3.3.1: Flowchart

3.4. Use Case Diagram

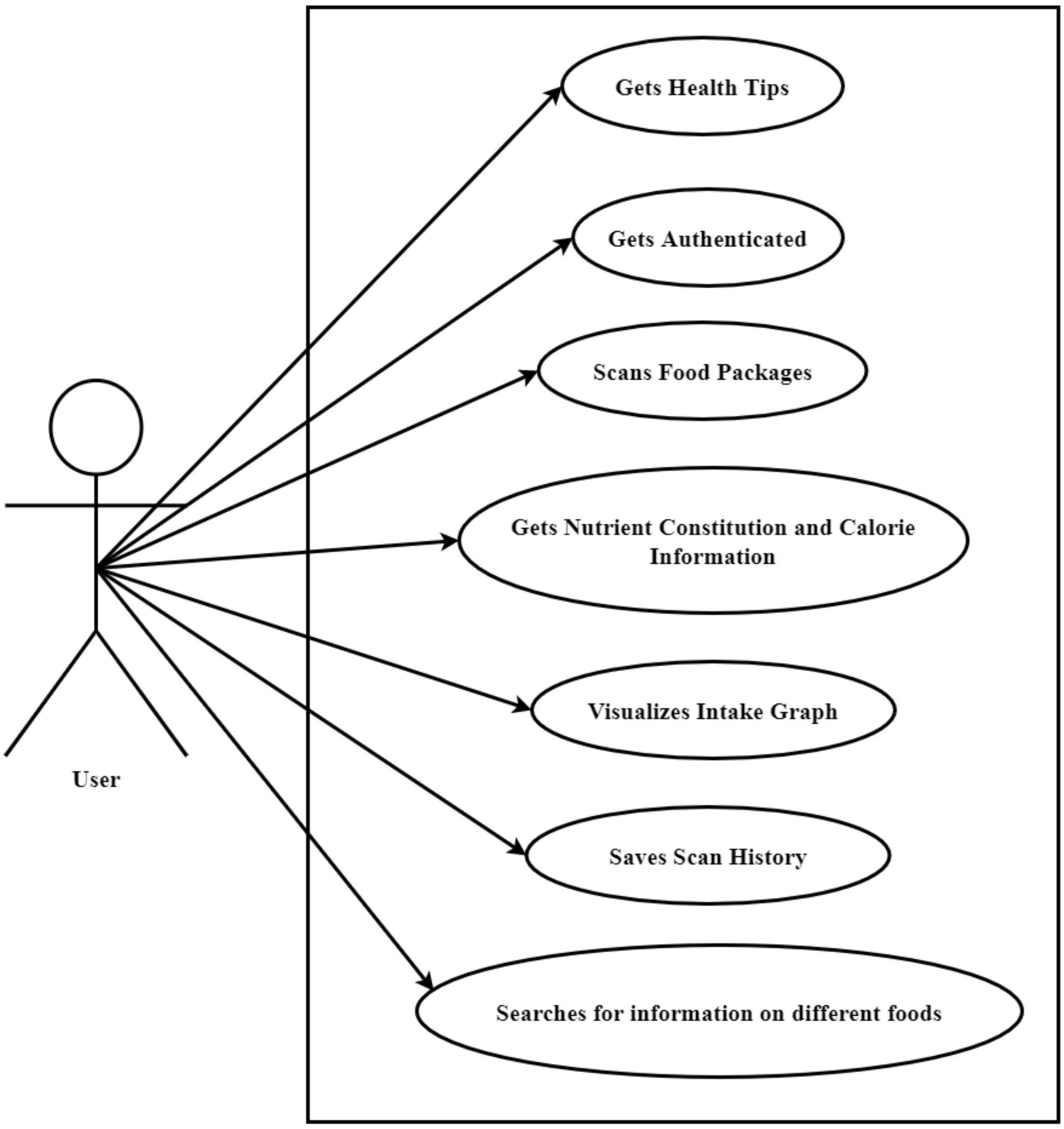


Figure 3.4.1: Use Case Diagram

3.5. Frontend Architecture Diagram

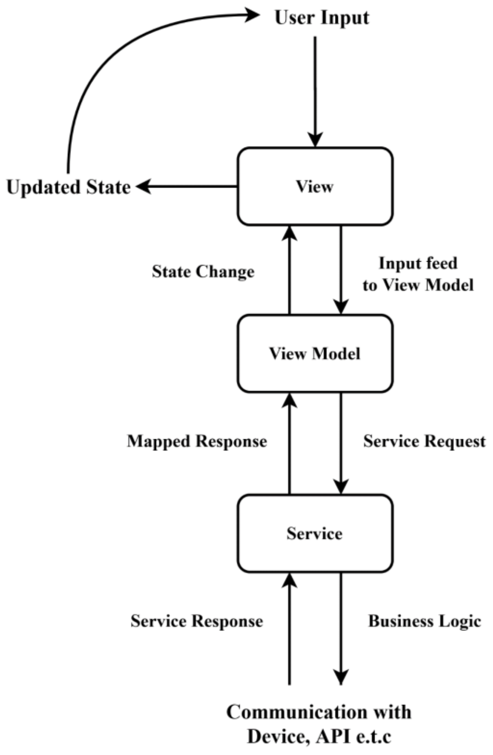


Figure 3.5.1: Frontend architecture flowchart

3.6. Edge Detection Diagram

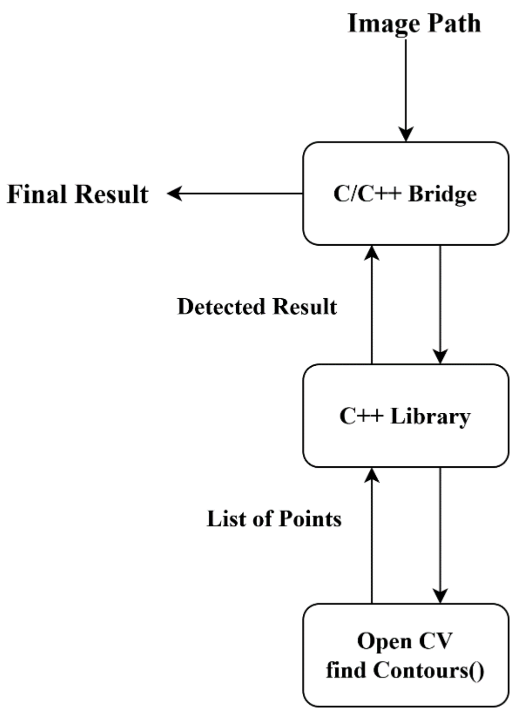


Figure 3.6.1: Edge detection flowchart

3.7. API Block Diagram

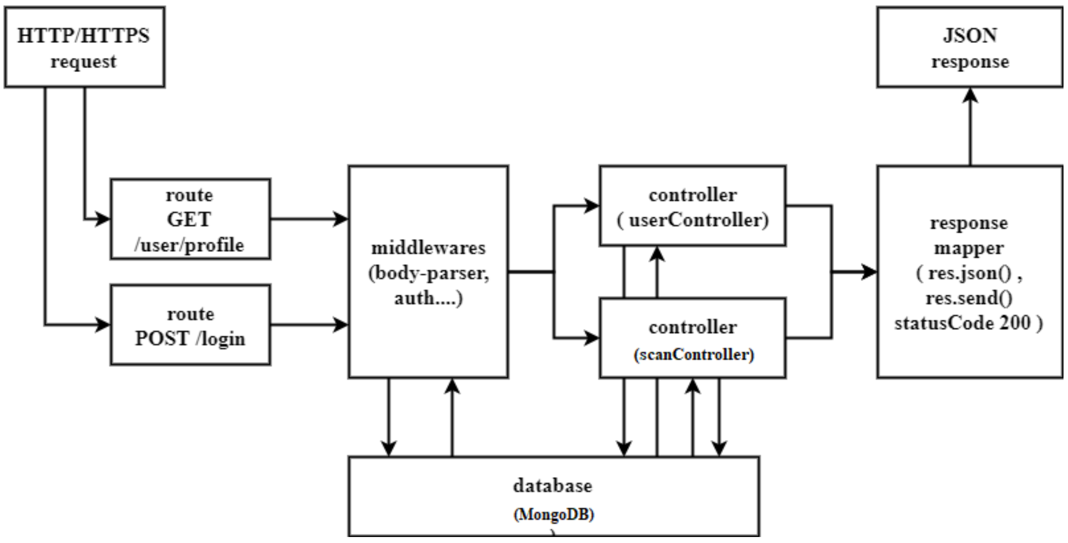

HTTP/HTTPS request
route GET /user/profile
route POST /login
middlewares (body-parser, auth....)
controller (userController)
controller (scanController)
database (MongoDB)
response mapper (res.json() , res.send() statusCode 200)
JSON response

Figure 3.7.1: API Block Diagram

3.8. System Requirement Specification

3.8.1. Software Requirement

It contains information about OS, programming platform, and other APIs. It also contains information about the technique used to store and manipulate data.

3.8.1.1. Front End Tool

The front-end tools used for this project include Flutter.

3.8.1.1.1. Flutter SDK [17]

Flutter [17] is an open-source mobile application development framework created by Google. It is used to develop applications for Android and iOS. Flutter apps are written in the Dart language and make use of many of the language's more advanced features**.**

3.8.1.2. Back End Tool

The back-end tools used for this project include Node.js, MongoDB database management system.

3.8.1.2.1. Node.js [18]

Node.js [18] is a very powerful JavaScript-based framework/platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications.

3.8.1.2.2. Bcrypt [19]

Bcrypt [19] is a NPM module. The bcrypt hashing function allows us to build a password security platform that scales with computation power and always hashes every password with a salt.

3.8.1.2.3. JWT [20]

JSON Web Token (JWT) [20]is a JSON encoded representation of a claim(s) that can be transferred between two parties. The claim is digitally signed by the issuer of the token, and the party receiving this token can later use this digital signature to prove the ownership of the claim.

3.8.1.2.4. Express [15]

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

3.8.1.2.5. OpenCV [6]

OpenCV [6] is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source BSD license.

3.8.1.2.6. MongoDB [21]

MongoDB [21] is a cross-platform document-oriented database program. Classified as a NoSQL database program. MongoDB uses JSON-like documents with optional schemas. MongoDB is faster than traditional database management systems. MongoDB is an open-source, scalable, high performance, document-oriented database.

3.8.1.2.7. Python [22]

Python is an interpreted, high-level, and general-purpose programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming.

3.8.1.2.8. Flask [23]

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

3.8.2. Development Tools

These are the tools that we used to develop this application.

3.8.2.1. Visual Studio Code [24]

Visual Studio Code [24] is a lightweight but powerful source code editor that runs on your desktop and is available for Windows, macOS, and Linux. It comes with built-in support for JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python [22], Go) and runtimes (such as .NET and Unity).

3.8.2.2. Figma [25]

Figma [25] is an interface design application that runs in the browser. Figma gives us all the tools needed for the design phase of our project, including vector tools that are capable of fully-fledged illustration, as well as prototyping capabilities.

3.8.2.3. GitHub [26]

GitHub is an open-source repository hosting service, sort of like a cloud for code. It hosts your source code projects in a variety of different programming languages and keeps track of the various changes made to every iteration. Other GitHub users can review your code and propose changes.

3.8.2.4. Terminal Window

The Terminal is an interface that allows you to access the command line from the GUI.

3.8.2.5. Diagrams.net [27]

diagrams.net [27] is a completely free online diagram editor built around JavaScript, that enables you to create flowcharts, UML, entity relation, network diagrams, mockups, and more.

3.8.2.6. Android Studio [28]

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

3.8.3. Hardware Requirements

For running backend, a VPS is recommended to run Node.js and Python.
For the smooth running of the app, it is suggested that the device has RAM greater than 2 GB.
This project requires an Android OS with an Android version greater than 4.4. This project is compatible with iOS also. An Android or iOS device with a camera is required to scan. Internet connection in the smartphone is required.

Chapter 4: Discussion on the Achievements

This chapter focuses on the challenges that we have experienced during this project. It also discusses any probable deviation from the objective, negative findings that were implemented.

4.1. Challenges

Image Processing has a lot of hype surrounding it. It needs much practice and algorithm tuning before making it work smoothly. Also, the image scanning in flutter took much time due to a lack of effective packages for our needs. And overall, there were also design and implementation challenges while doing this project. The notable ones are discussed below:

Concept-Based Challenges

- We had to change the architecture pipeline of the text detection technique as the initial technique did not produce satisfactory results.
- The design and implementation of the kind of UI / UX that we have planned for our application have been a challenging task.

Technical Challenges

- Learning new technology and implementing it in that short period is certainly difficult, especially with image processing.
- OCR is a field with lots of ongoing research. Due to this, perfect results are not obtained all the time.
- And finally, juggling time between assignments, internals, and the project was a mighty challenge as well.

4.2. Features

Despite all the aforementioned challenges, “nutri.gram” has come out to contain a few rather significant features for the users. Some of the important features are:

4.2.1. Scan feature to scan any food packaging

After the product package is scanned and information is displayed, the user has a choice to decide whether to save that scanned data or not. If saved, the scanned product can be viewed in the history page along with the scan date and a click to that product takes the user to the detailed information on that product along with the graphical visualization. The history feature is important because it saves users from scanning the same product over and over and to track their past intake to improve it.

4.2.2. History feature for saving scanned item

After the product package is scanned and information is displayed, the user has a choice to decide whether to save that scanned data or not. If saved, the scanned product can be viewed in the history page along with the scan date and a click to that product takes the user to the detailed information on that product along with the graphical visualization. The history feature is important because it saves users from scanning the same product over and over and to track their past intake to improve it.

4.2.3. Graphical visualization feature

The role of this feature is to provide a visual representation of different nutrients present in a scanned product. The users can know the composition of nutrients and calories in their diet in a more discernible way. The overall constitution of nutrients in all the scanned products can be tracked so that users can picture their intake in a better way and take action to improve their diet. This feature is present on the home page if the user has saved scanned data, in the profile page, and is displayed after each product scan in the information display page.

4.2.4. Health Tips feature

This feature provides information on how a person can remain healthy both mentally and physically. As the main theme of our project is to make people conscious of their diet and the nutrients their body is getting, we only thought that health tips would be an augmented feature to accentuate the fact that health is important. This feature is displayed on the front page with the help of a horizontally scrollable array of cards. Each card contains a health tip with its image and title. When clicked on a particular health tip, a bottom sheet pops up displaying a short description of the tip. This feature can be viewed without being logged in and makes this application more comprehensive.

4.2.5. Search feature for information on different foods

This app also consists of a search feature where the user can type the amount and name food, they want to get information about without going through the scanning process. The information about the searched product is displayed on the information display page along with the graphical representation of its nutrient content.

4.2.6. Verification feature for user information security

The user can sign up and log in using his phone number and the OTP verification code is sent to that number to maintain a secure application system.

Chapter 5: Conclusion and Recommendation

nutri.gram is a great platform for people to get information on the product they consume, about the nutritional and calorific value, and be more informed, and knowledgeable about their intake. This application will help people to be primed about nutrient details of the food they eat at the same time tracking the calorie value and visualizing it through different visual representations like graphs. While we worked hard to create this application in two months, due to time constraints there are certain limitations in the app and we expect to make some future enhancements in the coming days.

5.1. Limitations

Although we have tried our best to create a fully functional and user-friendly app, due to the time constraint of the project, there are still some significant gaping inconsistencies in it.

- The app doesn't respond without the internet or offline.
- The implementation of search features is not very efficient in our application as it is incorporated from external API.
- There is no recommended system in our application. We had planned to recommend the diets to people based on their intake data but we were not able to accomplish that due to time constraints.

5.2. Future Enhancements

Based on the limitations of the project, we intend to make these enhancements to the project to make it more deployable in the market.

- Make the app respond both offline and online.
- Now there is the only provision of scanning and getting information on packaged food. This can be enhanced to the scanning of any food item and their appropriate result.
- Image processing can be enhanced so that it can extract the information without feeding a cropped image into it.

- After the scan is complete, enable the user to edit the information in the package so that the user can add or omit a particular nutrient and visualize the result by customizing it.
- Implementation of a diet recommendation system.
- Incremental improvements to the styles, interactivity, and data security of the application.
- Possible ports to desktop and web applications since Flutter is a cross-platform software development kit.

References

[1] Leaf Group Ltd., "MyPlate Calorie Counter | Livestrong.com," 6 07 2020. [Online]. Available: https://www.livestrong.com/myplate/.

[2] PR Newswire Association LLC, "Consumer Reports Rates Diet Plans: MyFitnessPal, A Free App And Website, More Satisfying Than Weight Watchers," 4 7 2020. [Online]. Available: https://www.prnewswire.com/news-releases/consumer-reports-rates-diet-plans-myfitnesspal-a-free-app-and-website-more-satisfying-than-weight-watchers-185440082.html.

[3] Under Armour, Inc, "MyFitnessPal | MyFitnessPal.com," 3 07 2020. [Online]. Available: https://www.myfitnesspal.com/.

[4] Open Food Facts, "The Open Food Facts Team," 3 7 2020. [Online]. Available: https://world.openfoodfacts.org/who-we-are.

[5] N. Matsunaga and R. Sullivan, "Image processing for the extraction of nutritional information from food labels," *Computer Science and Engineering Senior Theses,* 6 9 2015.

[6] OpenCV, "OpenCV," 8 07 2020. [Online]. Available: https://opencv.org/.

[7] Wikipedia, "Bilateral filter - Wikipedia," 9 9 2020. [Online]. Available: https://en.wikipedia.org/wiki/Bilateral_filter.

[8] OpenCV, "OpenCV: Image Thresholding," 12 9 2020. [Online]. Available: https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html.

[9] "Tesseract documentation | Tesseract OCR," 17 9 2020. [Online]. Available: https://tesseract-ocr.github.io/.

[10] Wikipedia, "Levenshtein distance - Wikipedia," 16 9 2020. [Online]. Available: https://en.wikipedia.org/wiki/Levenshtein_distance.

[11] Flutter Clutter, "How to implement edge detection in Flutter," 4 9 2020. [Online]. Available:

https://www.flutterclutter.dev/flutter/tutorials/implementing-edge-detection-in-flutter/2020/1509/.

[12] Dart, "C interop using dart:ffi | Dart," 3 9 2020. [Online]. Available: https://dart.dev/guides/libraries/c-interop.

[13] Wikipedia, "Dependency injection - Wikipedia," 13 9 2020. [Online]. Available: https://en.wikipedia.org/wiki/Dependency_injection.

[14] filledstacks.com, "stacked | Flutter Package," 18 9 2020. [Online]. Available: https://pub.dev/packages/stacked.

[15] ExpressJS, "Express - Node.js web application framework," 18 9 2020. [Online]. Available: https://expressjs.com/.

[16] Mongoose, "Mongoose ODM v5.10.6," 18 9 2020. [Online]. Available: https://mongoosejs.com/.

[17] Google Flutter, "Flutter - Beautiful native apps in record time," 6 07 2020. [Online]. Available: https://flutter.dev/.

[18] Node.js Foundation, "Node.js," 3 07 2020. [Online]. Available: https://nodejs.org/en/.

[19] dcodelO, "bcryptjs - npm," 10 9 2020. [Online]. Available: https://www.npmjs.com/package/bcryptjs.

[20] Auth0, "JSON Web Tokens - jwt.io," 15 1 2020. [Online]. Available: https://jwt.io/.

[21] MongoDB,Inc., "The most popular database for modern apps | MongoDB," 15 9 2020. [Online]. Available: https://www.mongodb.com/.

[22] Python Software Foundation, "Our Documentation | Python.org," 21 9 2020. [Online]. Available: https://www.python.org/doc/.

[23] Pallets, "Welcome to Flask — Flask Documentation (1.1.x)," 12 9 2020. [Online]. Available: https://flask.palletsprojects.com/en/1.1.x/.

[24] Microsoft, "Visual Studio Code - Code Editing. Redefined," 5 8 2020. [Online]. Available: https://code.visualstudio.com/.

[25] Figma, "Figma," 9 2020 https://www.figma.com/login. [Online]. Available: 13.

[26] GitHub, "GitHub," 12 9 2020. [Online]. Available: https://github.com/.

[27] "Untitled Diagram - diagrams.net," 20 9 2020. [Online]. Available: https://app.diagrams.net/.

[28] Google Developers, "Download Android Studio and SDK tools | Android Developers," 12 9 2020. [Online]. Available: https://developer.android.com/studio/.

Appendix

A.1. Gantt Chart

Week	1	2	3	4	5	6	7
Planning & Preparation	■						
Work Division	■						
Coding		■	■	■	■		
Debugging & Beta-Testing						■	
Documentation							■

Table A.1.1: Gantt Chart

A.2. Workflow Chart

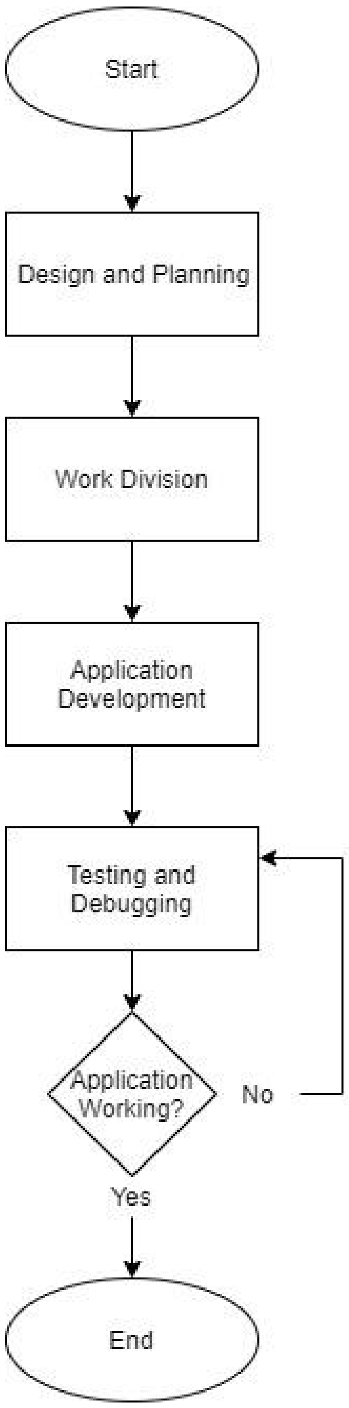
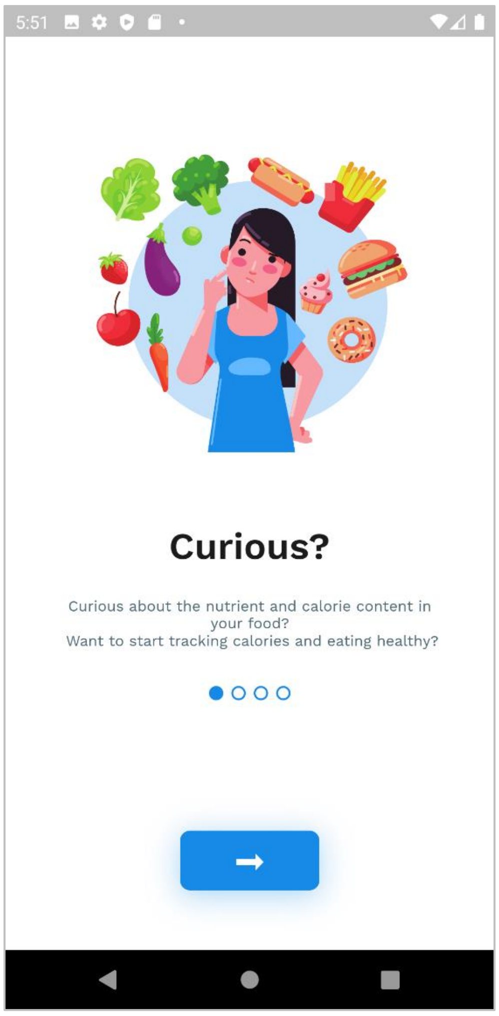


Figure A.2.1: Workflow chart

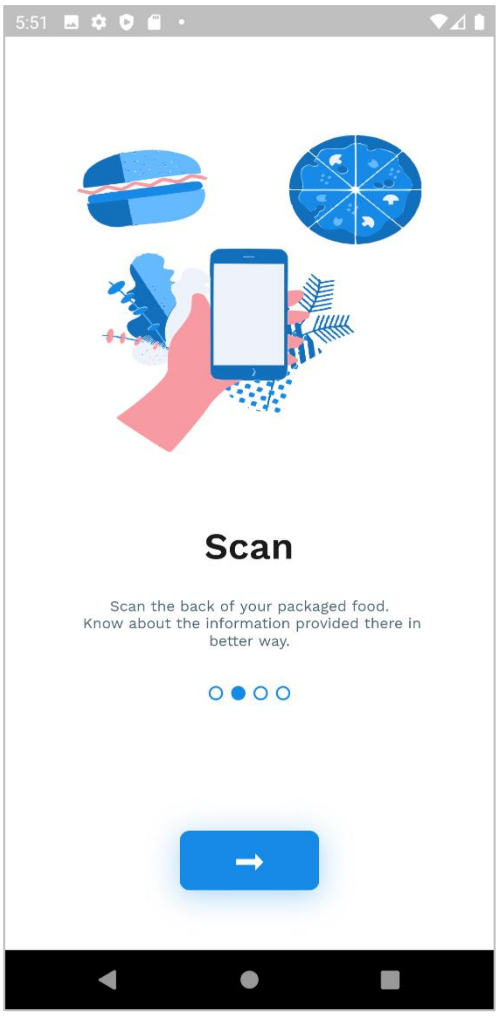
A.3. Screenshots



Curious?

Curious about the nutrient and calorie content in your food?
Want to start tracking calories and eating healthy?

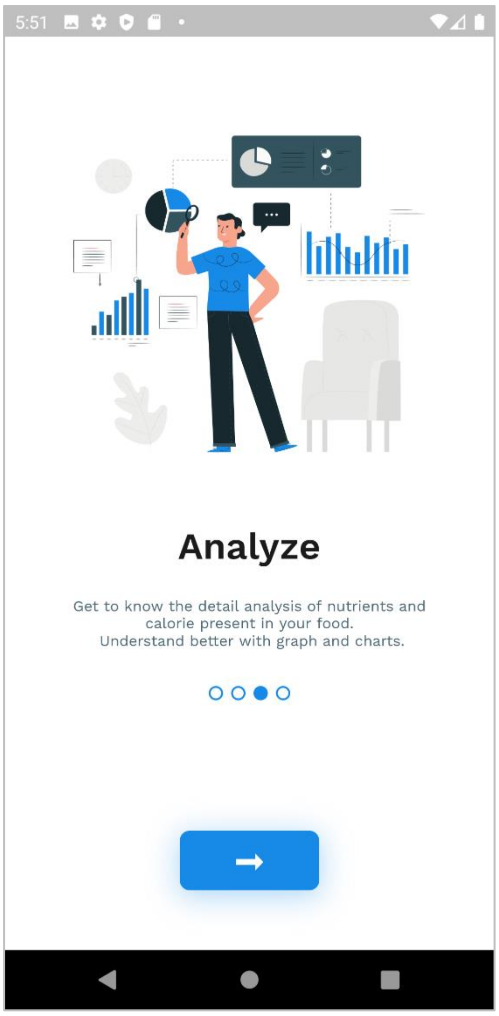
Figure A.3.2: Onboarding Page 1



Scan

Scan the back of your packaged food.
Know about the information provided there in better way.

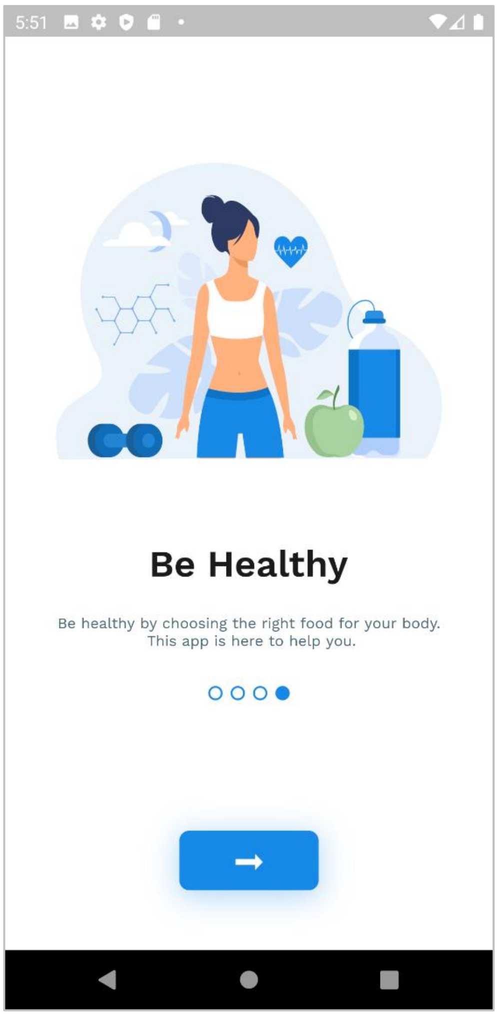
Figure A.3.1: Onboarding Page 2



Analyze

Get to know the detail analysis of nutrients and calorie present in your food.
Understand better with graph and charts.

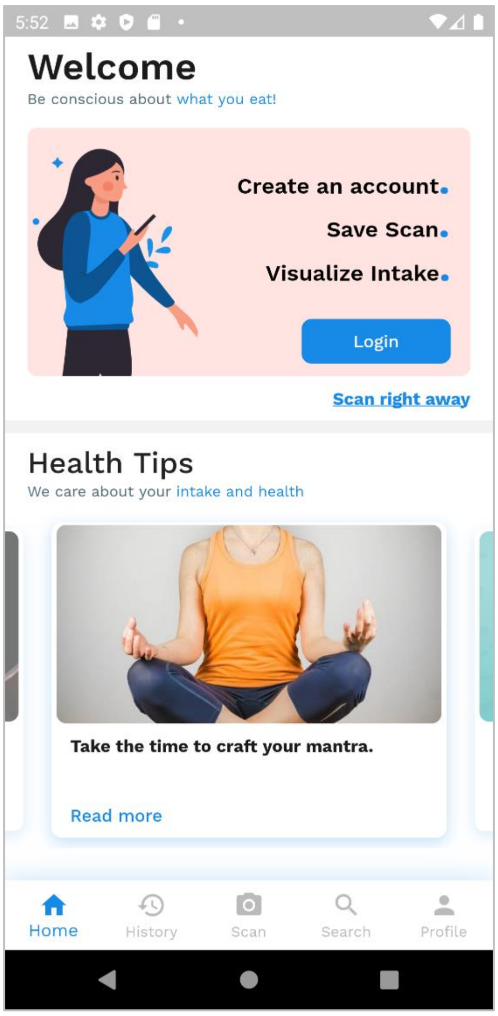
Figure A.3.4: Onboarding page 3



Be Healthy

Be healthy by choosing the right food for your body.
This app is here to help you.

Figure A.3.3: Onboarding page 4



5:52

Welcome

Be conscious about what you eat!

Create an account.

Save Scan.

Visualize Intake.

Login

Scan right away

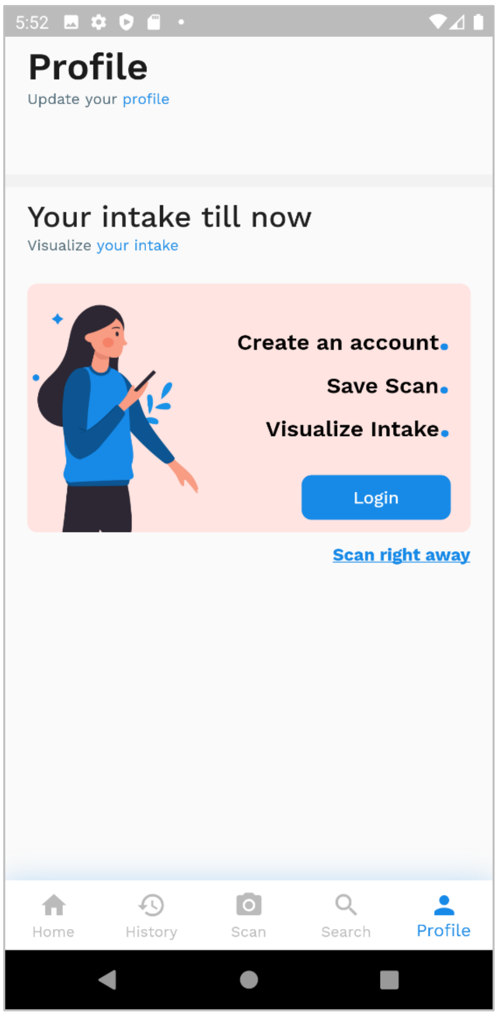
Health Tips

We care about your intake and health

Take the time to craft your mantra.

Read more

Figure A.3.6: Unauthenticated homepage



5:52

Profile

Update your profile

Your intake till now

Visualize your intake

Create an account.

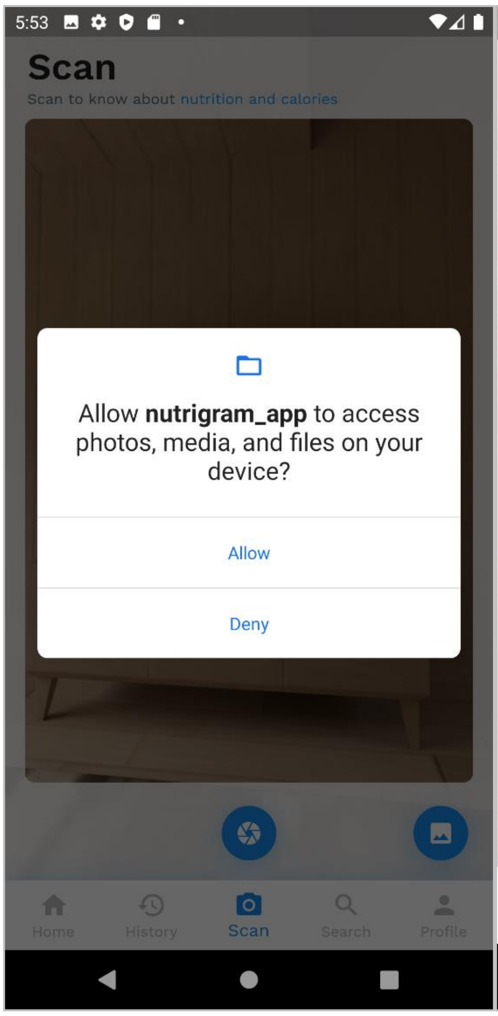
Save Scan.

Visualize Intake.

Login

Scan right away

Figure A.3.5: Unauthenticated profile page



Allow **nutrigram_app** to access photos, media, and files on your device?

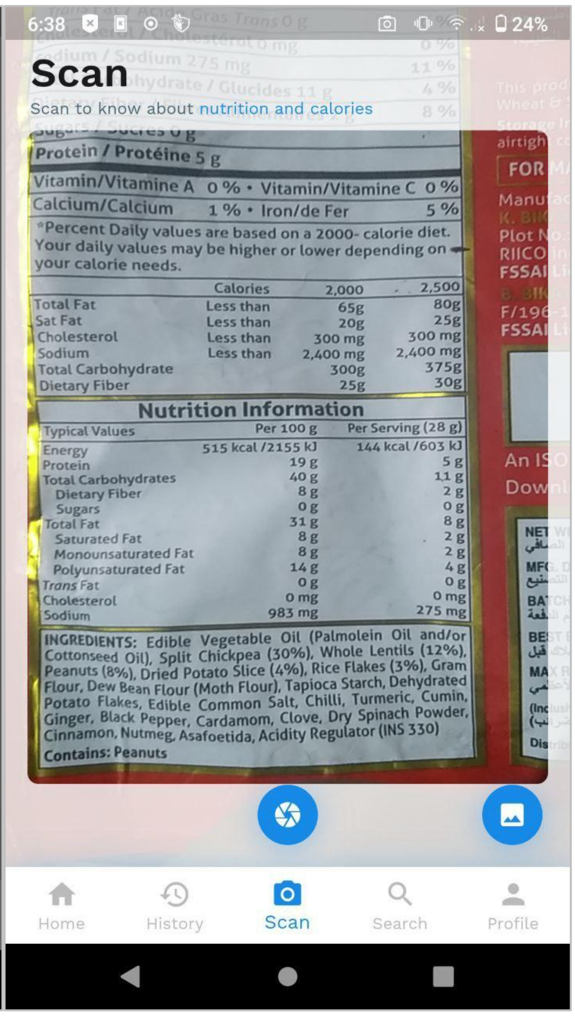
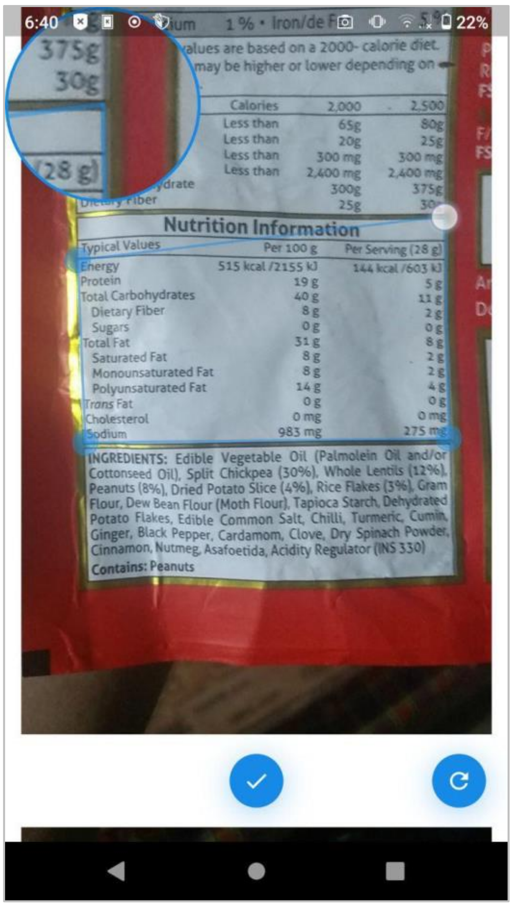


Figure A.3.8: Permission to access device camera

Figure A.3.7: Scanning product


6:40 1 % • Iron/de 22%

375g
30g
(28 g)

values are based on a 2000- calorie diet. may be higher or lower depending on

	Calories	2,000	2,500
	Less than	65g	80g
	Less than	20g	25g
	Less than	300 mg	300 mg
	Less than	2,400 mg	2,400 mg
ydrate		300g	375g
Dietary Fiber		25g	30g

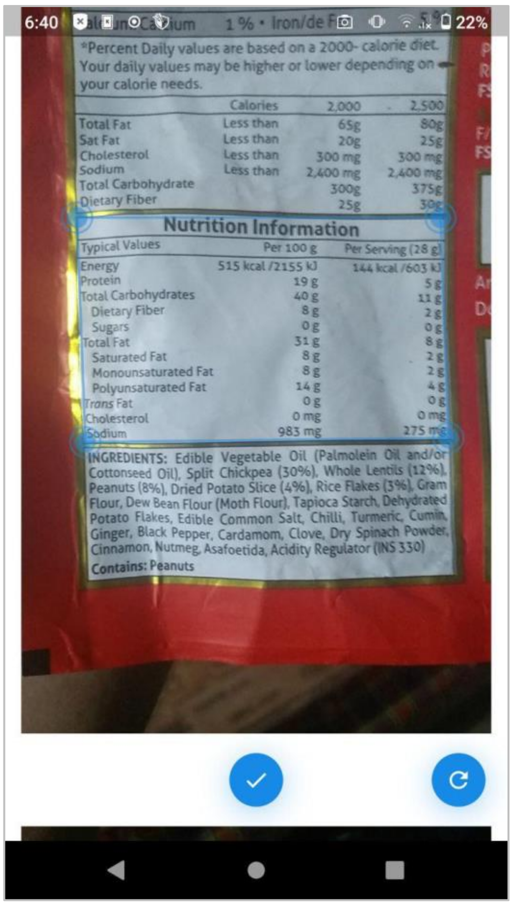
Nutrition Information

Typical Values	Per 100 g	Per Serving (28 g)
Energy	515 kcal /2155 kJ	144 kcal /603 kJ
Protein	19 g	5 g
Total Carbohydrates	40 g	11 g
Dietary Fiber	8 g	2 g
Sugars	0 g	0 g
Total Fat	31 g	8 g
Saturated Fat	8 g	2 g
Monounsaturated Fat	8 g	2 g
Polyunsaturated Fat	14 g	4 g
Trans Fat	0 g	0 g
Cholesterol	0 mg	0 mg
Sodium	983 mg	275 mg

INGREDIENTS: Edible Vegetable Oil (Palmolein Oil and/or Cottonseed Oil), Split Chickpea (30%), Whole Lentils (12%), Peanuts (8%), Dried Potato Slice (4%), Rice Flakes (3%), Gram Flour, Dew Bean Flour (Moth Flour), Tapioca Starch, Dehydrated Potato Flakes, Edible Common Salt, Chilli, Turmeric, Cumin, Ginger, Black Pepper, Cardamom, Clove, Dry Spinach Powder, Cinnamon, Nutmeg, Asafoetida, Acidity Regulator (INS 330)

Contains: Peanuts

Figure A.3.10: Correcting detected edges


6:40 1 % • Iron/de 22%

*Percent Daily values are based on a 2000- calorie diet. Your daily values may be higher or lower depending on your calorie needs.

	Calories	2,000	2,500
Total Fat	Less than	65g	80g
Sat Fat	Less than	20g	25g
Cholesterol	Less than	300 mg	300 mg
Sodium	Less than	2,400 mg	2,400 mg
Total Carbohydrate		300g	375g
Dietary Fiber		25g	30g

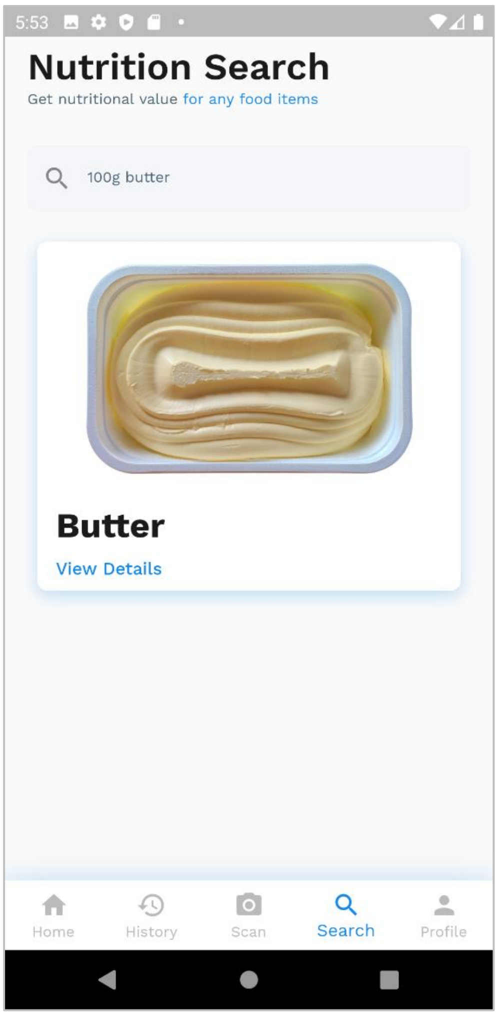
Nutrition Information

Typical Values	Per 100 g	Per Serving (28 g)
Energy	515 kcal /2155 kJ	144 kcal /603 kJ
Protein	19 g	5 g
Total Carbohydrates	40 g	11 g
Dietary Fiber	8 g	2 g
Sugars	0 g	0 g
Total Fat	31 g	8 g
Saturated Fat	8 g	2 g
Monounsaturated Fat	8 g	2 g
Polyunsaturated Fat	14 g	4 g
Trans Fat	0 g	0 g
Cholesterol	0 mg	0 mg
Sodium	983 mg	275 mg

INGREDIENTS: Edible Vegetable Oil (Palmolein Oil and/or Cottonseed Oil), Split Chickpea (30%), Whole Lentils (12%), Peanuts (8%), Dried Potato Slice (4%), Rice Flakes (3%), Gram Flour, Dew Bean Flour (Moth Flour), Tapioca Starch, Dehydrated Potato Flakes, Edible Common Salt, Chilli, Turmeric, Cumin, Ginger, Black Pepper, Cardamom, Clove, Dry Spinach Powder, Cinnamon, Nutmeg, Asafoetida, Acidity Regulator (INS 330)

Contains: Peanuts

Figure A.3.9: Detected Edges after correction



Nutrition Search

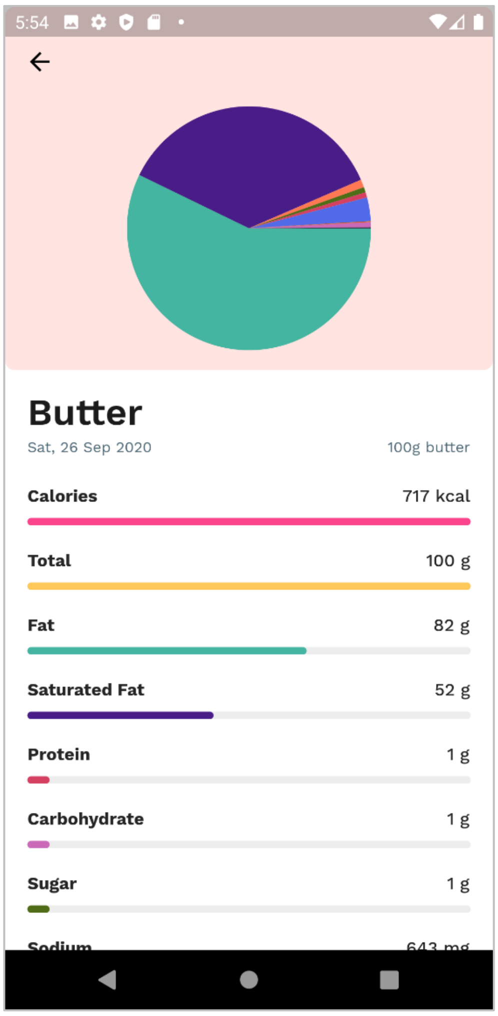
Get nutritional value for any food items

100g butter

Butter

View Details

Figure A.3.11: Food nutrition search result



Butter

Sat, 26 Sep 2020 100g butter

Calories 717 kcal

Total 100 g

Fat 82 g

Saturated Fat 52 g

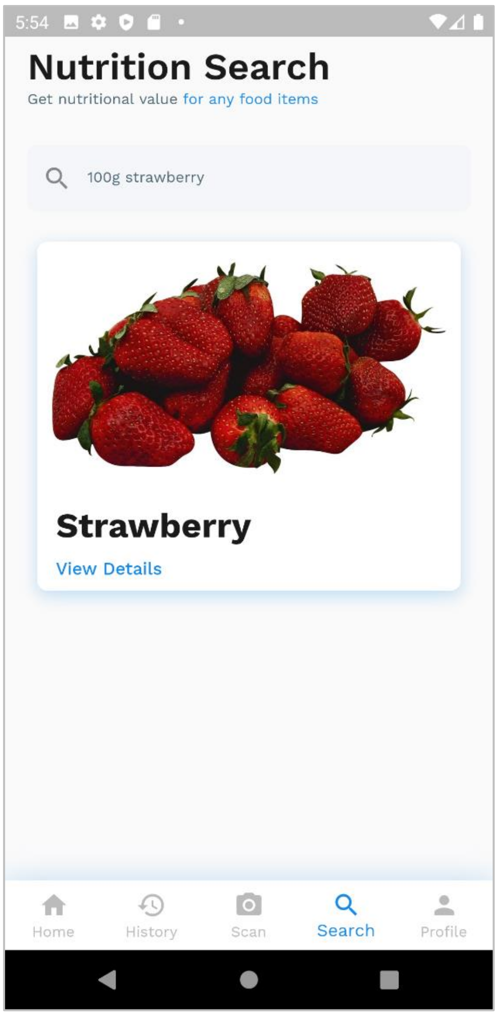
Protein 1 g

Carbohydrate 1 g

Sugar 1 g

Sodium 643 mg

Figure A.3.12: Food nutrition result for Butter (100g)



5:54

Nutrition Search

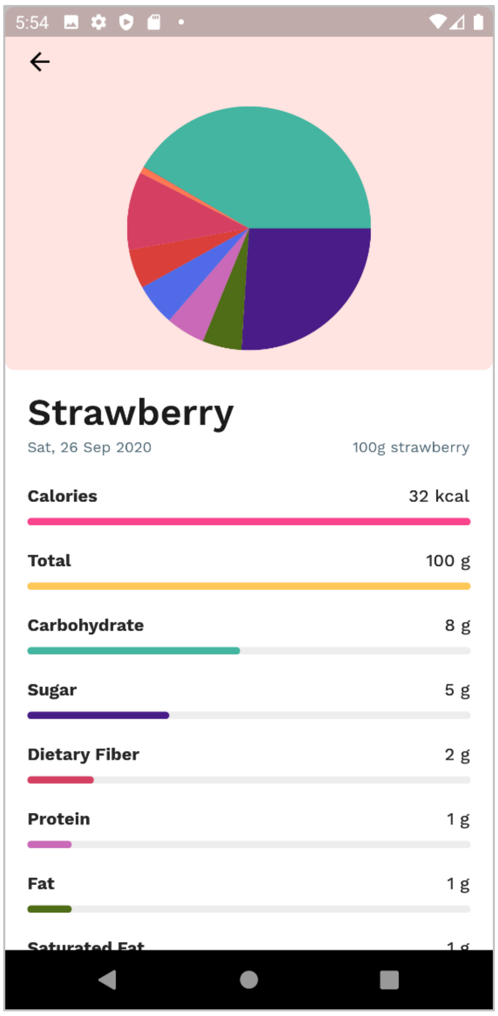
Get nutritional value for any food items

100g strawberry

Strawberry

View Details

Figure A.3.14: 100g strawberry search result



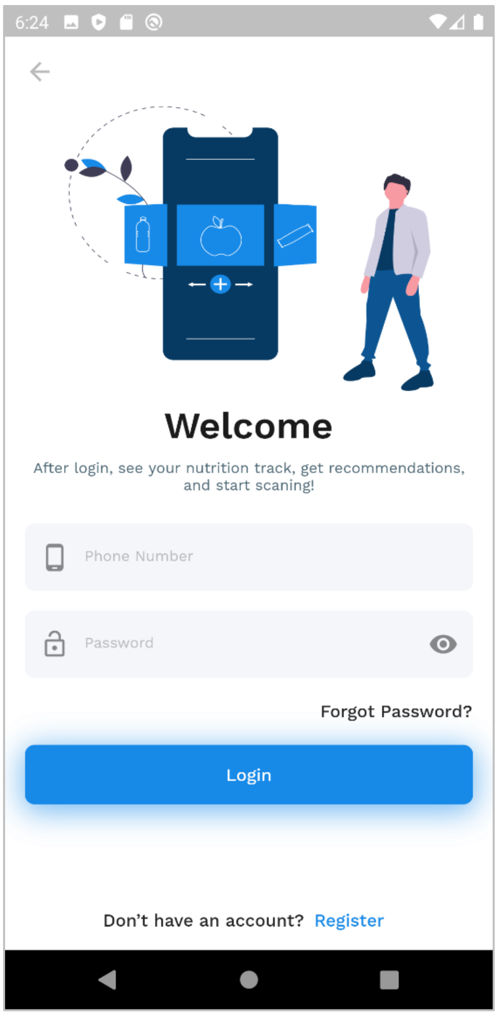
5:54

Strawberry

Sat, 26 Sep 2020 100g strawberry

Calories	32 kcal
Total	100 g
Carbohydrate	8 g
Sugar	5 g
Dietary Fiber	2 g
Protein	1 g
Fat	1 g
Saturated Fat	1 g

Figure A.3.13: Food nutrition result for strawberry (100g)


6:24

Welcome

After login, see your nutrition track, get recommendations, and start scaning!

Phone Number

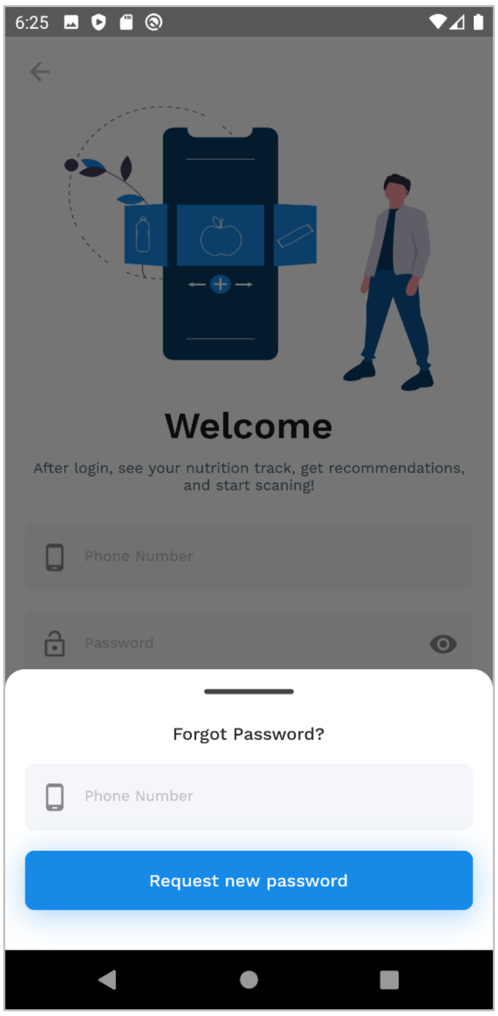
Password

Forgot Password?

Login

Don't have an account? Register

Figure A.3.16: Login page


6:25

Welcome

After login, see your nutrition track, get recommendations, and start scaning!

Phone Number

Password

Forgot Password?

Phone Number

Request new password

Figure A.3.15: Forgot password request form

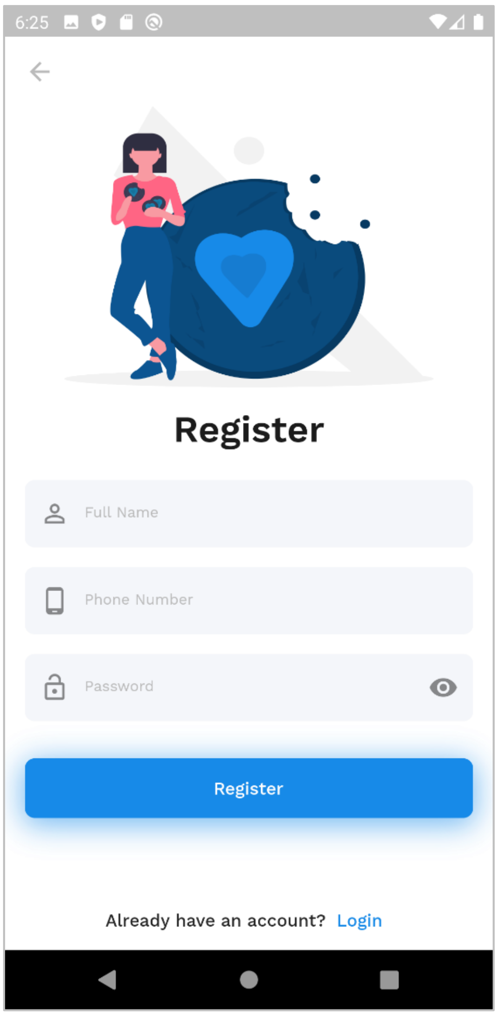
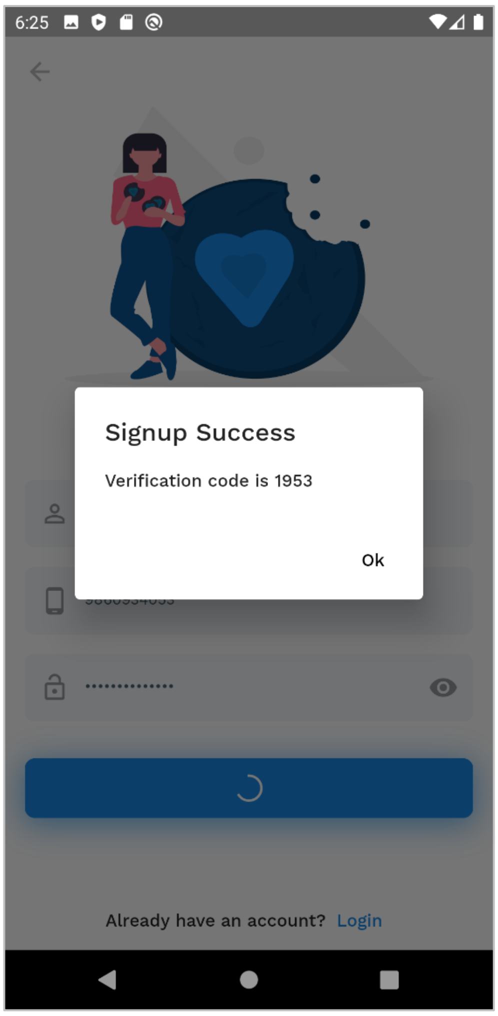


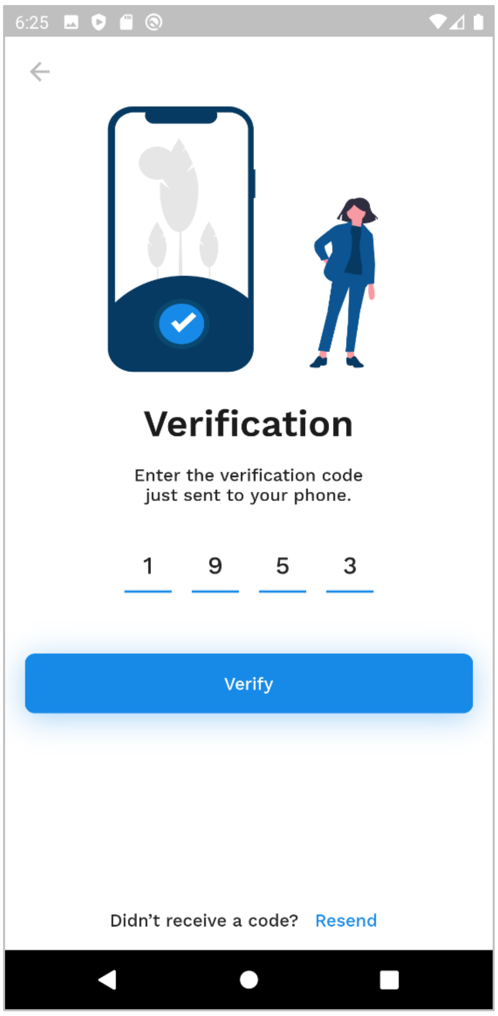
Figure A.3.18: New user registration page



Signup Success

Verification code is 1953

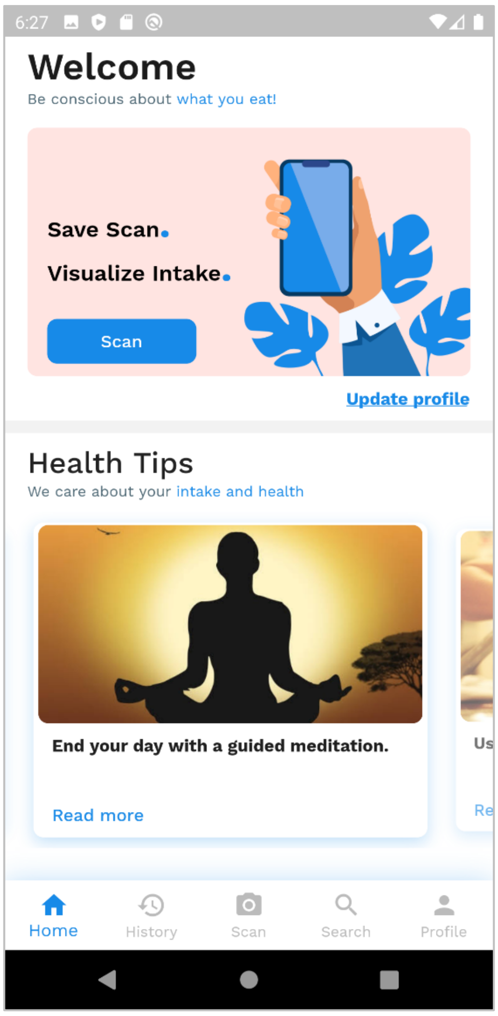
Figure A.3.17: Mock display for OTP retrieval


Verification

Enter the verification code just sent to your phone.

1 9 5 3

Figure A.3.20: New user verification using OTP


Welcome

Be conscious about what you eat!

Save Scan.

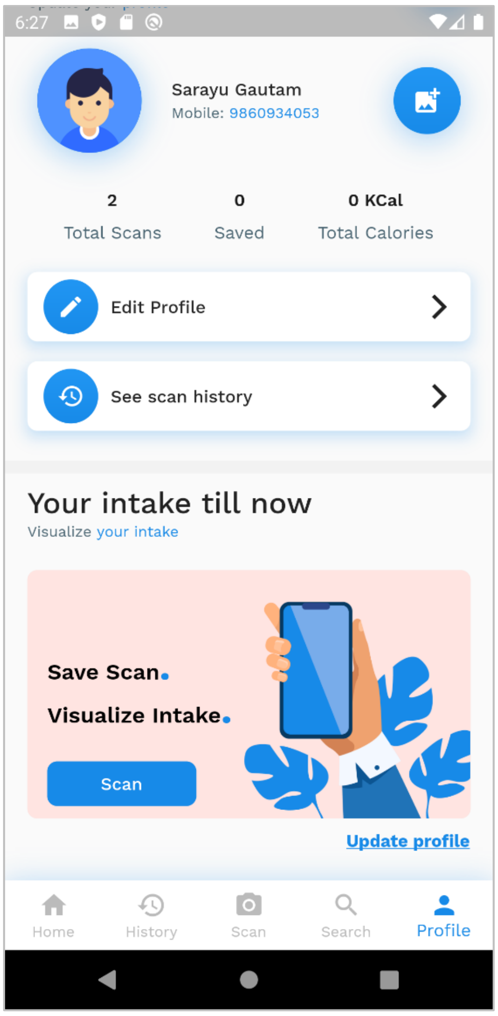
Visualize Intake.

Health Tips

We care about your intake and health

End your day with a guided meditation.

Figure A.3.19: Authenticated homepage for fresh user



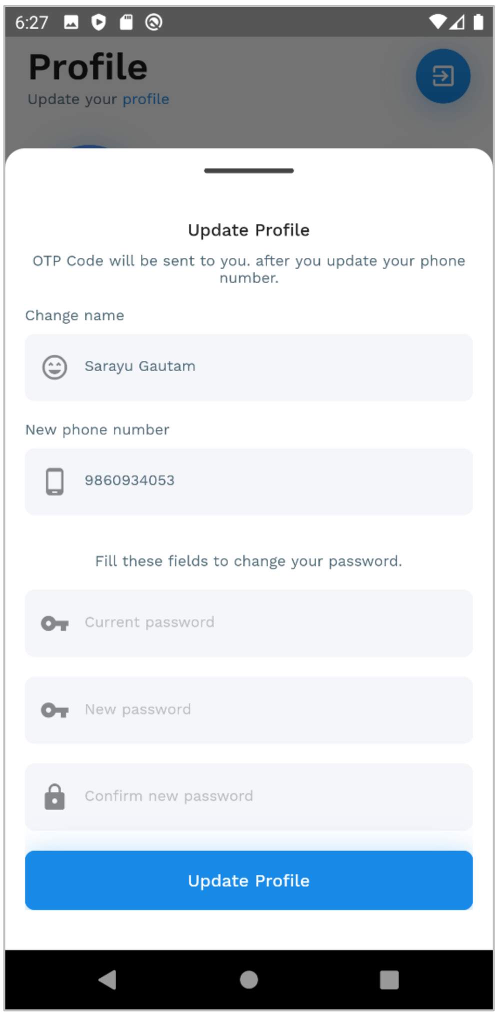
6:27

Sarayu Gautam
Mobile: 9860934053

2	0	0 KCal
Total Scans	Saved	Total Calories

Your intake till now
Visualize your intake

Figure A.3.22: Fresh User profile



6:27

Profile
Update your profile

Update Profile
OTP Code will be sent to you. after you update your phone number.

Change name: Sarayu Gautam
New phone number: 9860934053

Fill these fields to change your password.

Figure A.3.21: Update profile form

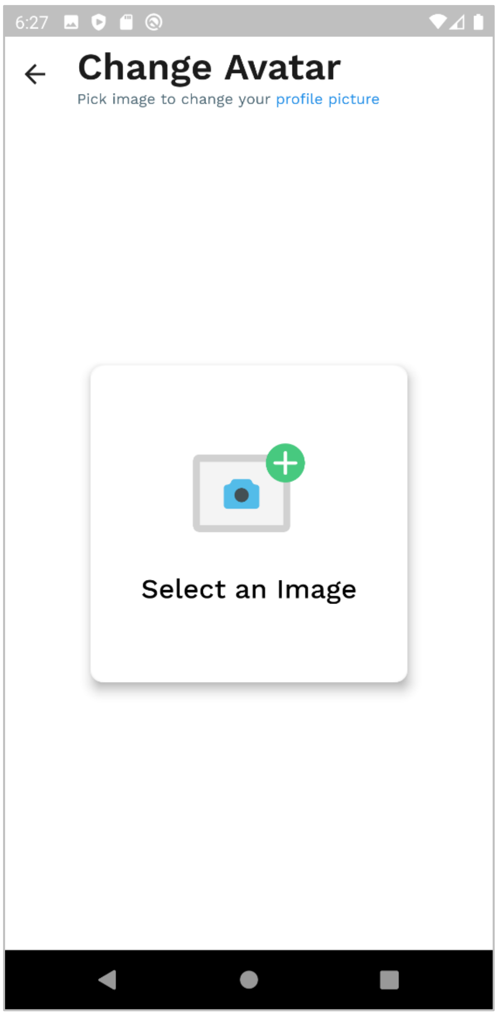


Figure A.3.24: Field for choosing a new image for a user

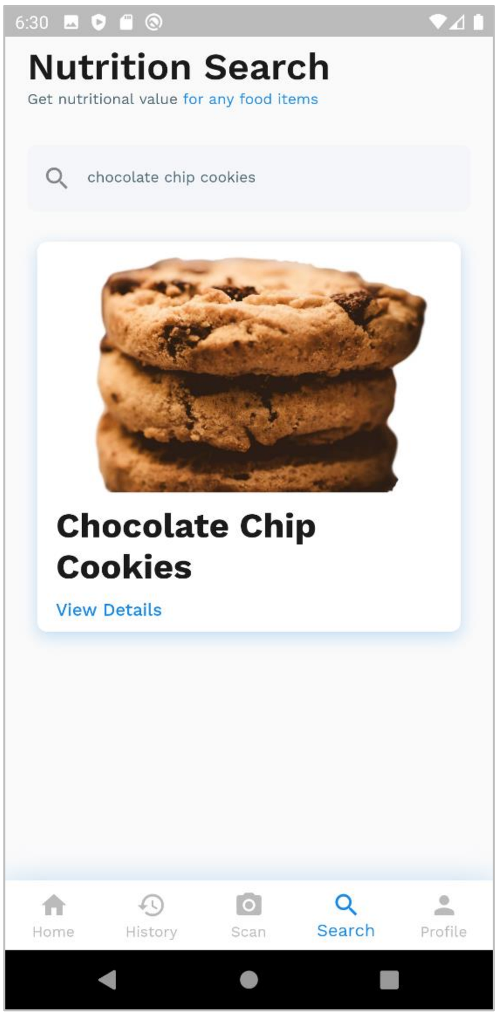
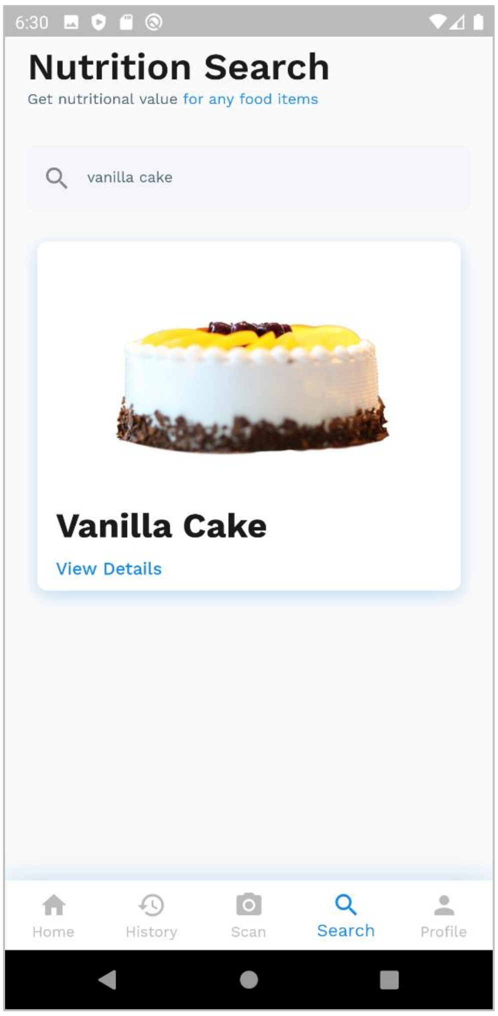


Figure A.3.23: Chocolate chip cookies search result


6:30

Nutrition Search

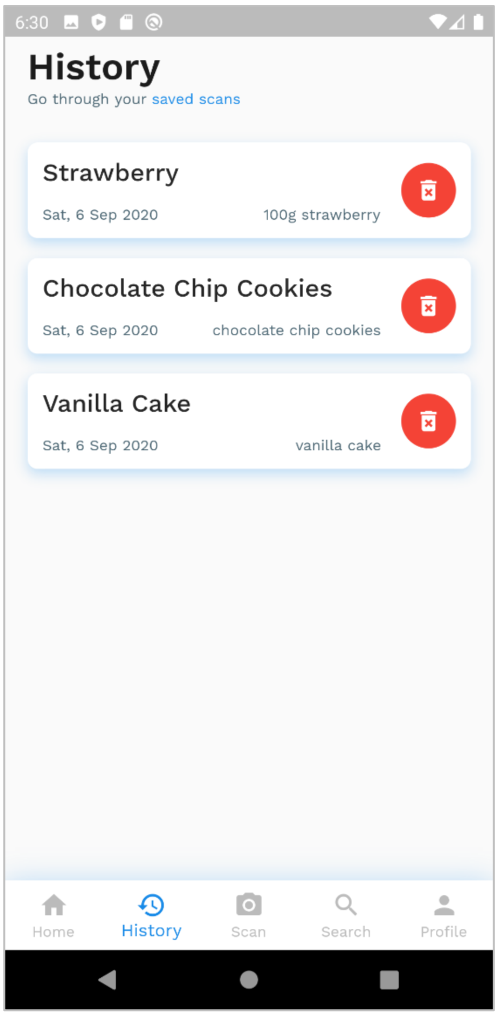
Get nutritional value for any food items

vanilla cake

Vanilla Cake

View Details

Figure A.3.26: Vanilla cake search result

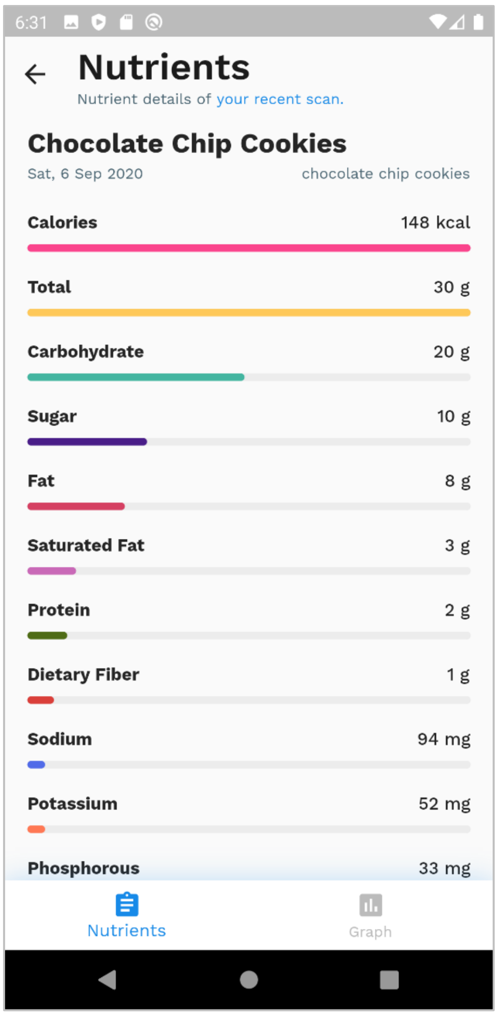

6:30

History

Go through your saved scans

Strawberry	Sat, 6 Sep 2020	100g strawberry
Chocolate Chip Cookies	Sat, 6 Sep 2020	chocolate chip cookies
Vanilla Cake	Sat, 6 Sep 2020	vanilla cake

Figure A.3.25: History page for all saved scans


6:31

Nutrients

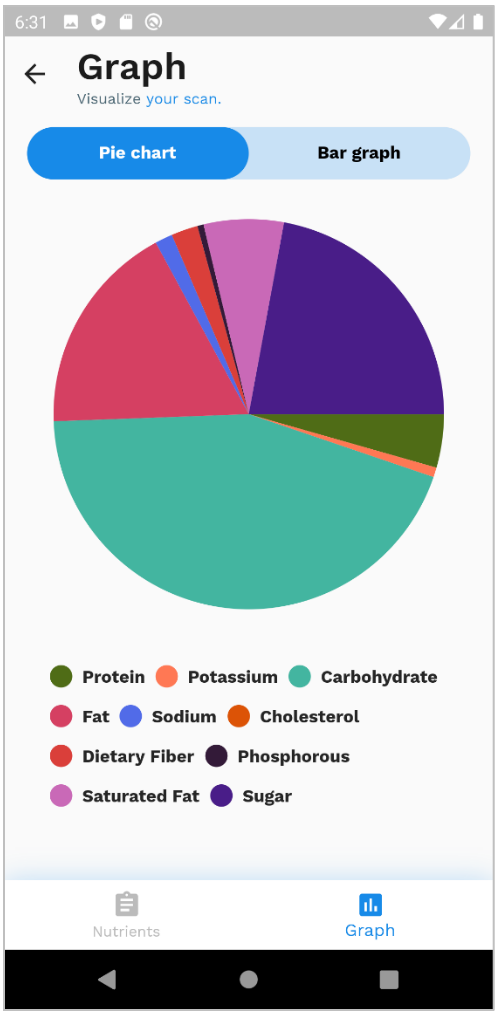
Nutrient details of your recent scan.

Chocolate Chip Cookies

Sat, 6 Sep 2020 chocolate chip cookies

Calories	148 kcal
Total	30 g
Carbohydrate	20 g
Sugar	10 g
Fat	8 g
Saturated Fat	3 g
Protein	2 g
Dietary Fiber	1 g
Sodium	94 mg
Potassium	52 mg
Phosphorous	33 mg

Figure A.3.28: Graph view for visualization of nutrients


6:31

Graph

Visualize your scan.

Figure A.3.27: Pie chart of micro and macronutrients combined

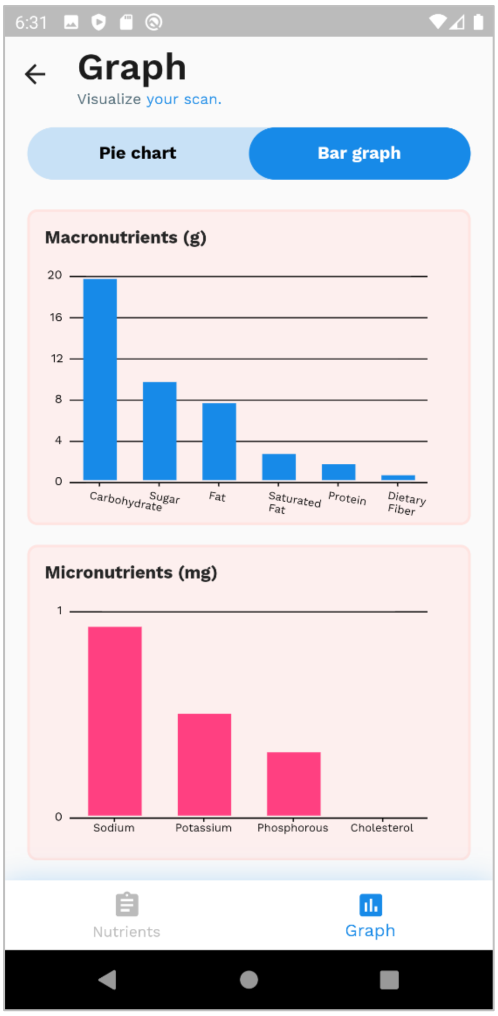


Figure A.3.29: Bar graph for micro and macronutrients

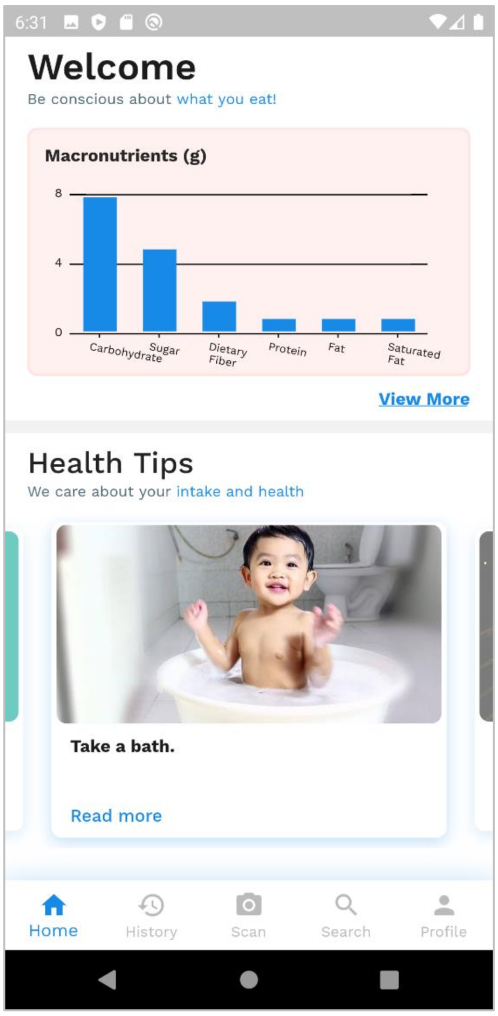
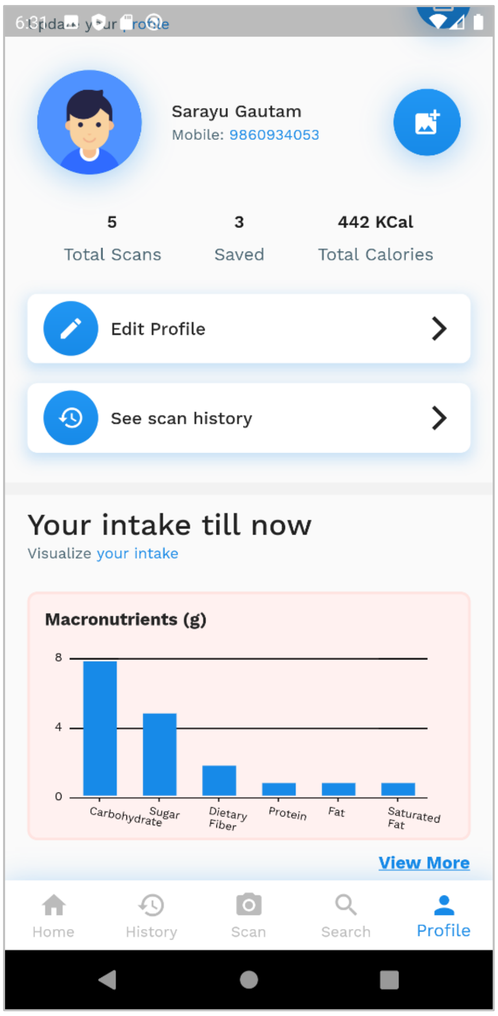


Figure A.3.30: Homepage after saving scans


Sarayu Gautam
Mobile: 9860934053

5	3	442 KCal
Total Scans	Saved	Total Calories

Edit Profile

See scan history

Your intake till now
Visualize your intake

Figure A.3.31: Profile page after saving scans