Bilkent University

Department of Computer Engineering

Senior Design Project

InPackt

Analysis Report

November 15, 2021

Şükrü Can Erçoban - 21601437

Kadir Mert Laleci - 21601960

Vural Doğan Akoğlu - 21602479

Kutsal Bekçi - 21602163

Fadime Selcen Kaya - 21801731

Supervisor: Asst. Prof. Dr. Hamdi Dibeklioğlu
Jury Members: Asst. Prof. Dr. Shervin Arashloo, Asst. Prof. Dr. Hamdi Dibeklioğlu

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

1. Introduction

After the industrial revolution, the number of packaged products has increased drastically. Nowadays markets are full of packaged products and comparing the products is a time consuming activity in our lives. Reading the table of contents for each product we bought, understanding chemicals in the product and deciding which one to put into the shopping cart became a growing problem as the chemicals and number of packaged products increased.

Health is without question the single-most important value in life. Without sound health, nothing else can be important. Therefore, great importance must be given to how we fuel our bodies. Many food items, especially packaged ones, contain additives that the average person is not even able to pronounce, let alone know the effects. People need to be educated about these ingredients in order to make informed choices about their most basic need.

Based on these issues, InPackt will offer a quick and healthy shopping experience and help to compare between products. Thanks to algorithms and developing technologies, we can compare and decide the best products for us without looking at their table of contents directly. The user only needs to point their phone camera at the product and InPackt will take care of everything from there. It will display to the user the ingredients, packaging information, dangerous content warnings as well as similar product recommendations based on user preferences.

In this report, a brief system overview, requirements and constraints will be explained. The details of the application design will be illustrated with use-case, activity, state and sequence diagrams. In addition, user interfaces will be demonstrated within the mockups. Finally, the road map of the project will be explained by dividing tasks into work packages.

2. Current System

Open Food Facts is an open-source, worldwide, collaborative project that aims to store information about branded food such as its ingredients, nutrition facts, and packaging, organized and divided by country [1]. In fact, since it is the largest and most comprehensive database available to us for our country, we will use this project as a source for our database. The app of this project includes a barcode scanner for identifying a product, whereas our project will use real-time object recognition to identify the product. Furthermore, it does not include personalized recommendations or specific needs. Also, as the name suggests, this project only focuses on food items, whereas ours has a larger scope.

3. Proposed System

3.1 Overview

InPackt will be a mobile application developed for android users and will be available in Google Play Store. InPackt will enable users to eliminate risky products specifically catered for them by scanning the product’s package to see the harmful ingredients and further information about the contents. Aforementioned products can be of the types food, and if time and resources allow us, cosmetics, cleaning products or supplements.

The determination of the “risky” product or ingredient will be customized depending on the choice of the user. Customization can include specific unwanted ingredients, allergants as well as products that are gluten or lactose free, vegan, vegetarian or halal. The goal of our project is to provide an easy way to see the ingredients of the market products and shorten the decision process of choosing between products by individualized ingredient detector. A user will be able to look at alternative products to the one that they scanned depending on their customization criteria.

InPackt will also present the container type and information of any product with a package to further inform the user. If they want, the users will be able to factor in the sustainability and the recyclability of the packaging in their recommendations.

In the product information page, among the ingredients and container type, there will be calorie specifications as well if the product is a food, and the user will be able to choose to be recommended lower-calorie items.

InPackt will utilize real-time object recognition in order to identify the product. The user will point their camera toward the product and the recognized item will pop up on the screen. If time and efficiency allow us, we want to implement an Augmented Reality feature so that the instant that the app recognizes a product, it is displayed on the user’s screen on the product. The object recognition is the main focus, and if it fails, the writings of the package will be read to match with a product. If that fails as well, the options of voice command, barcode scanner or manual product entry will be displayed.

The project will definitely be available for users in Turkey, in Turkish and English, and if the time and technology is sufficient, be available for users worldwide in

English. The products will need to be searched in their own language, however apart from that the app can be used in both languages.

3.2 Functional Requirements

3.2.1 System

The system should:

- Ask permission to access the camera, microphone, local storage, and internet connection of the user's mobile device.
- Provide an option to create an account to save the user's preferences and personalized choices to the cloud.
- Provide an option to continue without creating an account and save user’s preferences and personalized choices in their local storage.
- Display the focused object apart from surrounding objects on the screen.
- Provide an option to scan the product's barcode.
- Provide an option to search the product manually.
- Provide an option to search the product via voice.
- Send the recognized object picture or scanned barcode to the backend server for processing.
- Find the product from the database and receive its contents.
- Compare the contents of the product with the user's disallowed list.
- Display ingredients of the product and warnings if there are any unhealthy ingredients, ingredients from the user's disallowed list and package material .

- Suggest similar products without the ingredients from the disallowed list according to the user's personal choices.
- Provide options that the users can personalize their preferences such as vegetarian, vegan, alcohol-free, halal, lactose intolerance and gluten intolerance.
- Provide options to the users to customize default lists of vegetarian, vegan, alcohol-free, halal, package material, lactose intolerance and gluten intolerance.
- Provide an option to see the general ingredients list to choose and add to the disallowed list.
- Provide an option to remove ingredients from the disallowed list.
- Provide an option to suggest a product to add with ingredients information and send the suggestion to approval.

If time allows, the system should:

- Display warnings and ingredients on focused objects in the camera view above the product.

3.2.2 User

The user can:

- Allow or deny permissions to access camera, microphone, local storage, and internet connection of the mobile device.
- Create an account to save their preferences and personalized choices to the cloud.
- Continue without creating an account and save their preferences and personalized choices to the cloud.
- Select a product to focus on the camera view.

- Scan the barcode of the product.
- Search the product manually.
- Search the product via voice.
- View its ingredients as a list.
- View warnings next to the risky ingredient.
- View similar products if the product has an ingredient from a disallowed list.
- View general ingredient list.
- Personalize their preferences such as vegetarian, vegan, alcohol-free, halal, lactose intolerance, package material and gluten intolerance.
- Customize default lists of vegetarian, vegan, alcohol-free, halal, lactose intolerance and gluten intolerance.
- Choose ingredients to add to the disallowed list.
- Remove ingredients from the disallowed list.
- Suggest a product with ingredients information.

If time allows, the user can:

- View warnings and ingredients on focused objects in the camera view above the product.

3.3 Nonfunctional Requirements

The non-functional requirements are explained below in seperate headers.

3.3.1 Availability

- InPackt application should be available 24 hours a day, 7 days a week for all users who have devices with iOS or Android operating systems.

- InPackt application should be available mainly for users who live in Turkey, and ideally worldwide.

3.3.2 Accuracy

- InPackt application should be able to display the ingredients of the product scanned by user with at least %90 accuracy.
- InPackt application should be %90 accurate when recommending products according to the personal characteristic specified by the user while creating a profile for the product scanned by the user.

3.3.3 Response Time

- InPackt application should display the ingredient of the product in less than 3 seconds after the user scans.
- InPackt application should display the alternative products in less than 3 seconds.

3.3.4 Extendibility

- The design of the InPackt application should be extensible for future updates covering additional features according to the needs of the users.
- The user data of the InPackt application should be stored to be extensible for a Web application or other platforms.

3.3.5 Scalability

- InPackt application database should be scalable according to the increased number of users.Initially, at least 100 different devices can access simultaneously.

3.3.6 Privacy

- User information should not be stored and used outside the user’s device.
- User information should be encrypted to store their own devices.
- The InPackt application should ask the user’s permission for the alternative product recommendation system according to the private user information that has been created by the user.

3.3.7 License

- All used databases, libraries and the like must be licenced.

3.3.8 Maintainability

- The InPackt application should have a design that will allow the maintainability of the system without any problems with future updates and ensuring that some changed parts are not affected to others functionality.

3.3.9 Usability

- The application aims to present to the user what is in the content of the product that the user wants to buy, whether it is suitable for the profile created by the user, and if it is not suitable, what other alternative products are available to the user in a fast, easy and reliable manner.

- Tutorial and guideline for how to use the InPackt app is available within the app.

3.3.10 Reliability

- The information displayed on the application must be accurate.
- There must be a brief information page about most used and unknown harmful ingredients to be displayed when there is no internet connection for informing users when they need to be informed.

3.4 Pseudo Requirements

3.4.1 Language Constraints

- Our application will support both Turkish and English languages.

3.4.2 Economic Constraints

- Github is used to maintain the website without any cost.
- For version control and code sharing Github will be used which is free to use.
- Free APIs will be used for recognition.

- Open Food Facts is an open-source and collaborative project which allows the use of its database for free.
- Publishing apps on Google Play Store requires a one-time payment of 27 US and publishing an application on App Store requires a subscription fee of 99 USD each year.
- Open-source libraries will be used.

3.4.3 Implementation Constraints

- Jira and GitHub will be used for project management, source control and code review.
- The frontend will be designed by Flutter because the application will be on both IOS and android.
- Object Recognition will be implemented based on TensorFlow Lite [2].
- The backend will be implemented by Python [3] because we will use libraries for python and google APIs
- The product data and object images which are used for object training are stored in PostgreSQL [4].
- The product data will be obtained from the source by parsing.
- To get a high percentage of correct recognition, the training object data will be compared by Google image search by Google Cloud [5] or Google's Custom Search JSON API [6] .

3.4.4 Social Constraints

- Understandable language must be used to describe the harmful ingredients in order to not misguide anyone.

3.4.5 Ethical Constraints

- The application will have a minimized, optimally no, error rate for allergens in order to protect users from harmful, potentially deadly, products. The information and database used must be accurate, flawless and up to date.
- In order to not affect the market competition, a new product should be added to the database as soon as possible.

3.4.6 Health and Safety Constraints

- Because of the ongoing pandemic, ideally the users should be able to identify a product in front of them without touching the package.
- The application will be customizable according to the user, since understanding of a healthy product may differ for each user. So, the recommendation system must be unbiased.

3.4.7 Sustainability Constraints

- The products must be updated and added regularly for long term use.
- Users should be able to give feedback or add non-existent products in the database.
- Any errors or bugs must be fixed as soon as possible.

3.5 System Models

3.5.1 Scenarios

Scenario 1	Sign up
Actor	User
Entry condition	User clicks the sign up button
Exit condition	1. Sign up process is successful. 2. User clicks the back button.
Flow of event	1. User enters the required information for login. 2. User completes the sign up process. 2.1 If the process is successful, the user is directed to the home page. 2.2 If the process fails, the user enters again.

Scenario 2	Login
Actor	User
Entry condition	User opens the application and clicks the login button.
Exit condition	1. Login process is successful. 2. User clicks the back button.
Flow of event	1. User enters the required information which is email and password for login. 2. User completes the login process. 2.1 If the process is successful, the user is directed to the home page. 2.2 If the process fails, the user enters again.

Scenario 3	Forgot password
Actor	User
Entry condition	User clicks forgot password button in login page
Exit condition	1. User’s password is reset 2. User clicks the back button.

Flow of event	1. User enters his/her email 2. User choose new password

Scenario 4	Scan a product
Actor	User
Entry condition	User enters scanning page from home page
Exit condition	1. The app redirects to the ingredient detail page 2. User clicks the back button.
Flow of event	1. User enables phone's camera 2. Scan the product 3. User views scanned product ingredients

Scenario 5	View product ingredient
Actor	User
Entry condition	1. User scans a product from scanning page with camera 2. User searches a product from searching page
Exit condition	User clicks the back button.
Flow of event	1. User scans or searches a product 2. The product emerges with its image and ingredient information

Scenario 6	Scan barcode
Actor	User
Entry condition	User choose option for scanning barcode instead of scan product
Exit condition	1. The app redirects to the ingredient detail page 2. User clicks the back button.
Flow of event	1. User enables phone's camera 2. Scan the barcode of product 3. User views scanned product ingredients

Scenario 7	Search a product
Actor	User
Entry condition	1. User enters searching page from home page 2. User write the searched word in searching bar
Exit condition	1. The app redirects to the product detail page 2. User clicks the back button.
Flow of event	1. User encounters with listed products according to search 2. User selects a product to see details

Scenario 8	Search an ingredient
Actor	User
Entry condition	1. User enters searching page from home page 2. User write the searched word in searching bar
Exit condition	1. The app redirects to the ingredient detail page 2. User clicks the back button.
Flow of event	1. User encounters with listed ingredient according to search 2. User selects a ingredient to see details

Scenario 9	Make a product favorite
Actor	User
Entry condition	User clicks add favorite button in product detail
Exit condition	User selects the product as favorite
Flow of event	1. User access the product detail by scanning or searching 2. User add the product to the favorite list

Scenario 10	Make a product unwanted
Actor	User
Entry condition	User clicks add unwanted button in product detail

Exit condition	User selects the product as unwanted
Flow of event	1. User access the product detail by scanning or searching 2. User add the product to the unwanted list

Scenario 11	Make an ingredient favorite
Actor	User
Entry condition	User clicks add favorite button in ingredient detail
Exit condition	User selects the ingredient as favorite
Flow of event	1. User access the ingredient detail by scanning or searching 2. User add the ingredient to the favorite list

Scenario 12	Make an ingredient unwanted
Actor	User
Entry condition	User clicks add unwanted button in ingredient detail
Exit condition	User selects the ingredient as unwanted
Flow of event	1. User access the ingredient detail by scanning or searching 2. User add the ingredient to the unwanted list

Scenario 13	Create/Edit profile choices
Actor	User
Entry condition	User clicks the user profile button on the home page.
Exit condition	1. Save the changes on the profile. 2. User clicks the back button.
Flow of event	1. User enters profile 2. User determines his/her profile choices as vegetarian, vegan, alcohol-free, halal, package material, lactose intolerance and gluten intolerance. 3. User save the changes.

Scenario 14	View/Edit favorite products
Actor	User
Entry condition	User clicks favorite products button in user profile
Exit condition	User clicks the back button.
Flow of event	1. User enters his/her profile and clicks favorite products 2. User sees his/her favorite products from the list 3. User can remove products from the list or clear the entire list, depending on his/her choice. 4. User save the changes.

Scenario 15	View/Edit favorite ingredients
Actor	User
Entry condition	User clicks favorite ingredients button in user profile
Exit condition	User clicks the back button.
Flow of event	1. User enters his/her profile and clicks favorite ingredients 2. User sees his/her favorite ingredients from the list 3. User can remove ingredients from the list or clear the entire list, depending on his/her choice. 4. User save the changes.

Scenario 16	View/Edit unwanted products
Actor	User
Entry condition	User clicks unwanted products button in user profile
Exit condition	User clicks the back button.
Flow of event	1. User enters his/her profile and clicks unwanted products 2. User sees his/her unwanted products from the list 3. User can remove products from the list or clear the entire list, depending on his/her choice. 4. User save the changes.

Scenario 17	View/Edit unwanted ingredients
Actor	User
Entry condition	User clicks unwanted ingredients button in user profile
Exit condition	User clicks the back button.
Flow of event	5. User enters his/her profile and clicks unwanted ingredients 6. User sees his/her unwanted ingredients from the list 7. User can remove ingredients from the list or clear the entire list, depending on his/her choice. 8. User save the changes.

Scenario 18	View previous scans
Actor	User
Entry condition	User clicks past scans button in home page
Exit condition	User clicks the back button.
Flow of event	1. User login the app and enter past scans page 2. User encounters with previous scans

Scenario 19	View suggestion
Actor	User
Entry condition	User scans a product from scanning page with camera and views product ingredient
Exit condition	User clicks the back button.
Flow of event	1. User scans or searches a product and its ingredient information come up 2. Suggested product emerges as pop up according to preference list properties

Scenario 20	View information
Actor	User
Entry condition	User click information button in side menu

Exit condition	User clicks the back button.
Flow of event	1. User clicks information button in side menu 2. User sees a text about how to use Inpackt app 3. User sees developers

Scenario 21	Change settings
Actor	User
Entry condition	User click change settings button in home page
Exit condition	User clicks the back button.
Flow of event	1. User clicks change settings button in home page 2. User can change font size 3. User can change themes as dark or light

Scenario 22	Change account settings
Actor	User
Entry condition	User click change account settings button in setting page
Exit condition	User clicks the back button.
Flow of event	1. User click change account settings button in setting page 2. User can change profile name 3. User can clear everything as reset all choices 4. User can delete his/her account

Scenario 23	Log out
Actor	User
Entry condition	User clicks the log out button
Exit condition	1. User logged out
Flow of event	1. User log out from the application

3.5.2 Use Case Model

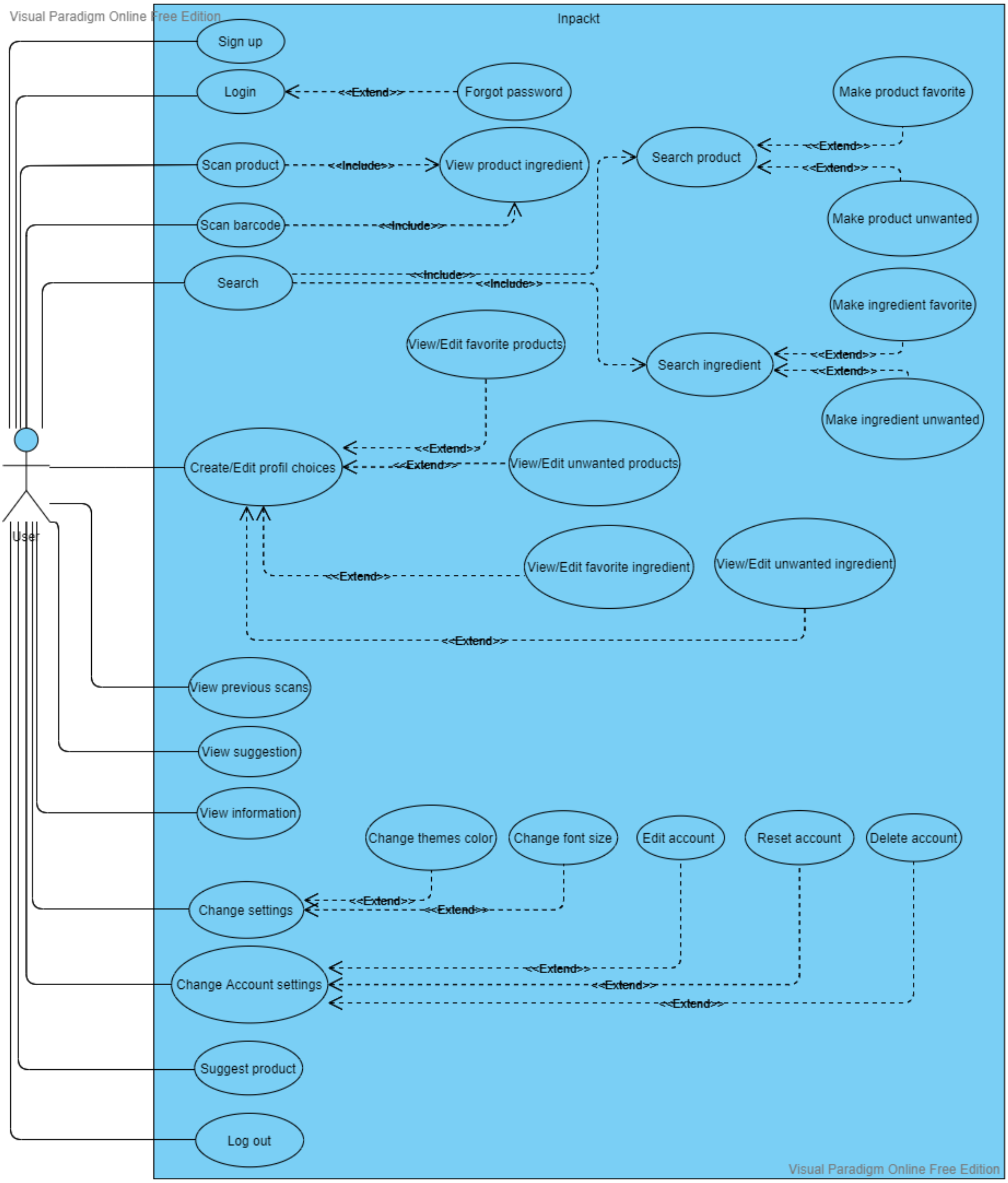


Figure 1: Use Case Diagram

3.5.3 Object and Class Model

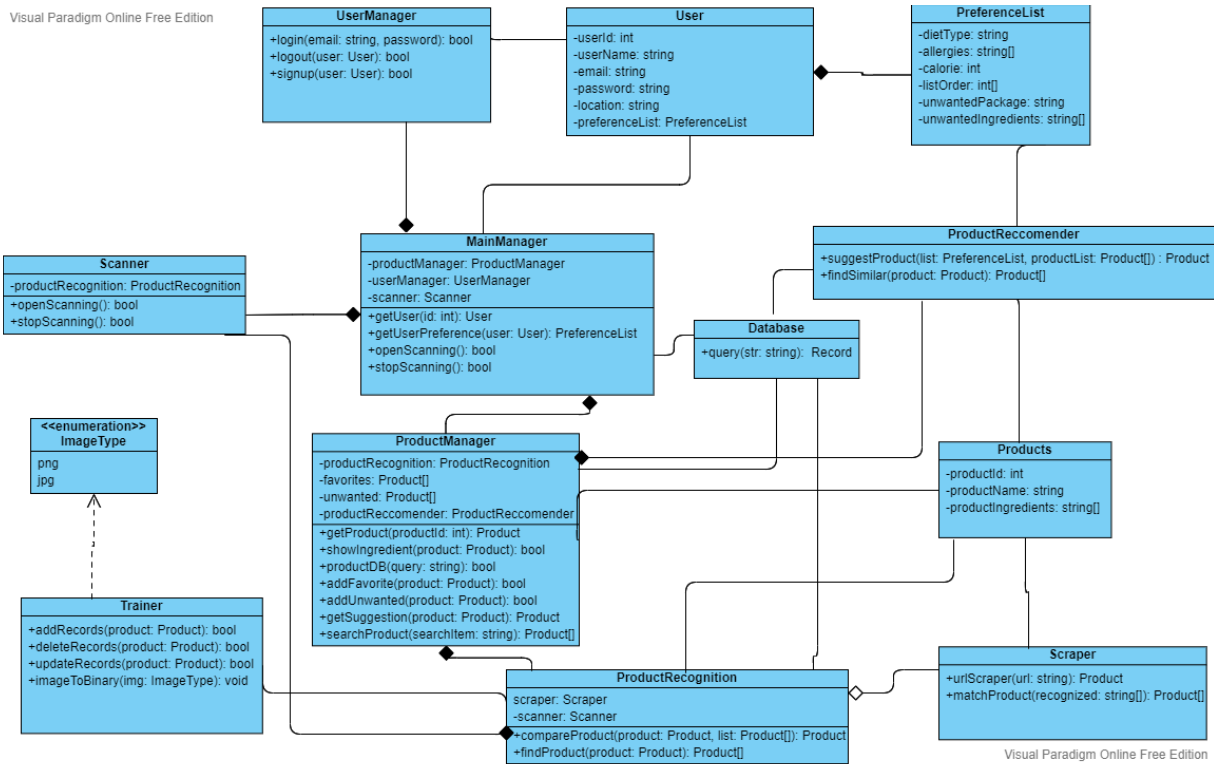

Visual Paradigm Online Free Edition
UserManager
+login(email: string, password): bool
+logout(user: User): bool
+signup(user: User): bool
User
-userId: int
-userName: string
-email: string
-password: string
-location: string
-preferenceList: PreferenceList
PreferenceList
-dietType: string
-allergies: string[]
-calorie: int
-listOrder: int[]
-unwantedPackage: string
-unwantedIngredients: string[]
Scanner
-productRecognition: ProductRecognition
+openScanning(): bool
+stopScanning(): bool
MainManager
-productManager: ProductManager
-userManager: UserManager
-scanner: Scanner
+getUser(id: int): User
+getUserPreference(user: User): PreferenceList
+openScanning(): bool
+stopScanning(): bool
ProductReccomender
+suggestProduct(list: PreferenceList, productList: Product[]) : Product
+findSimilar(product: Product): Product[]
Database
+query(str: string): Record
<<enumeration>>
ImageType
png
jpg
ProductManager
-productRecognition: ProductRecognition
-favorites: Product[]
-unwanted: Product[]
-productReccomender: ProductReccomender
+getProduct(productId: int): Product
+showIngredient(product: Product): bool
+productDB(query: string): bool
+addFavorite(product: Product): bool
+addUnwanted(product: Product): bool
+getSuggestion(product: Product): Product
+searchProduct(searchItem: string): Product[]
Products
-productId: int
-productName: string
-productIngredients: string[]
Trainer
+addRecords(product: Product): bool
+deleteRecords(product: Product): bool
+updateRecords(product: Product): bool
+imageToBinary(img: ImageType): void
ProductRecognition
scraper: Scraper
-scanner: Scanner
+compareProduct(product: Product, list: Product[]): Product
+findProduct(product: Product): Product[]
Scraper
+urlScraper(url: string): Product
+matchProduct(recognized: string[]): Product[]
Visual Paradigm Online Free Edition

Figure 2: Class Diagram

3.5.4 Dynamic Models

3.5.4.1 Activity Diagram

3.5.4.1.1 Overall Activity Diagram

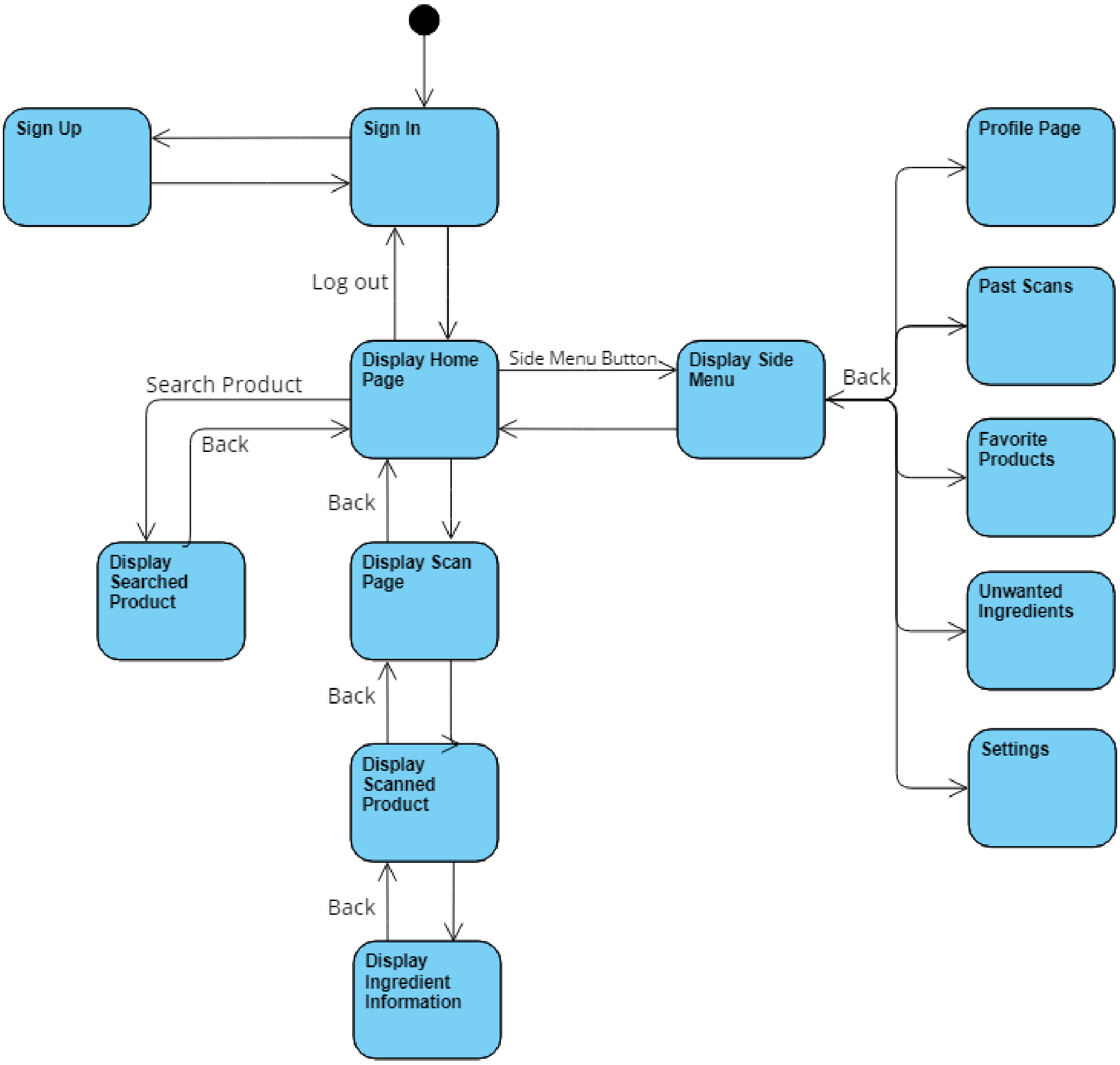


Figure 3: Overall Activity Diagram

3.5.4.1.2 Scan Product Activity Diagram

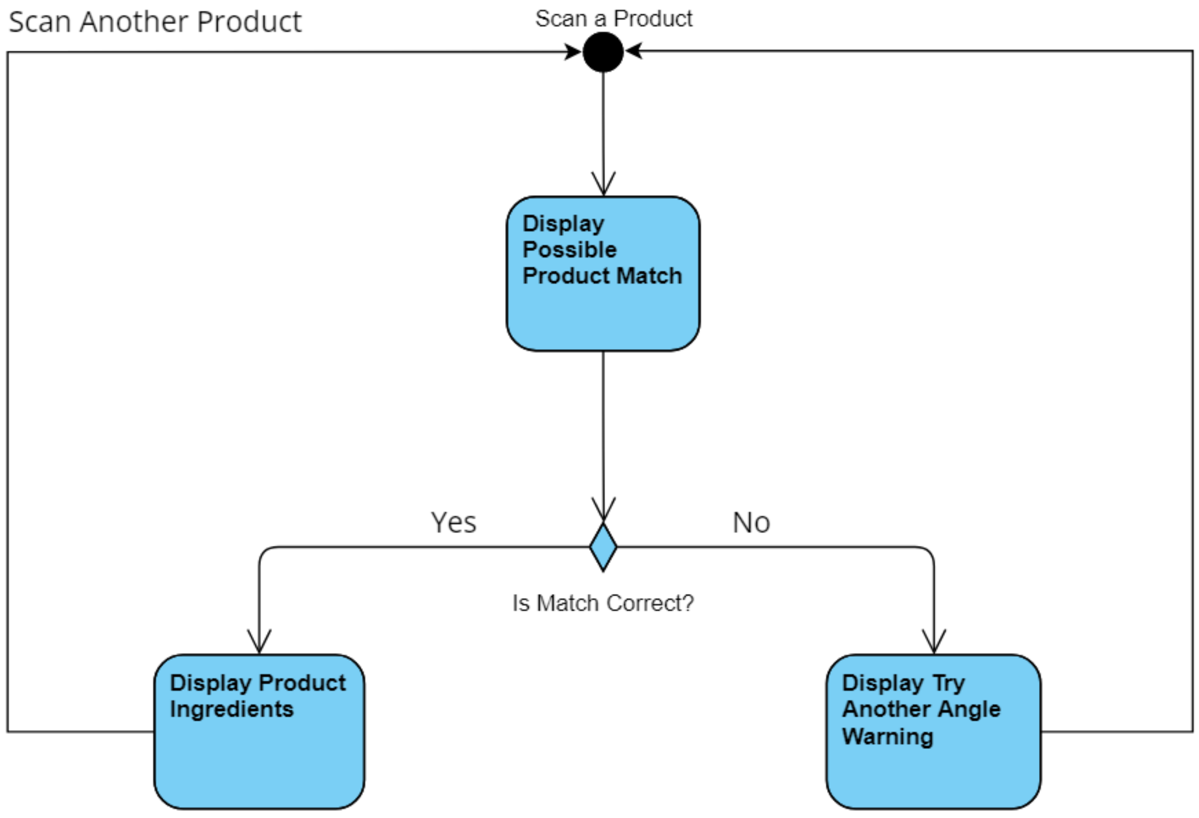

Scan Another Product
Scan a Product
Display Possible Product Match
Yes
No
Is Match Correct?
Display Product Ingredients
Display Try Another Angle Warning

Figure 4: Activity Diagram of Scan Another Product

3.5.4.2 State Diagram

First diagram shows the scanning state diagram which starts when the scanning request is received from the user. If the product can be recognisable by the application, the app moves to the product found state that shows the recognised product to the user. Otherwise, the app moves to product not found state and comes back to scanning for another try. If the identified product is confirmed by the user, the app moves to the display state that shows the ingredients of the identified product. In this state, the app offers two options to the user, one for end scanning and other for showing the package recycling information of the product. If the user requests for the package information, the app moves to the show package information state where the user can end the scanning process or request to scan another product.

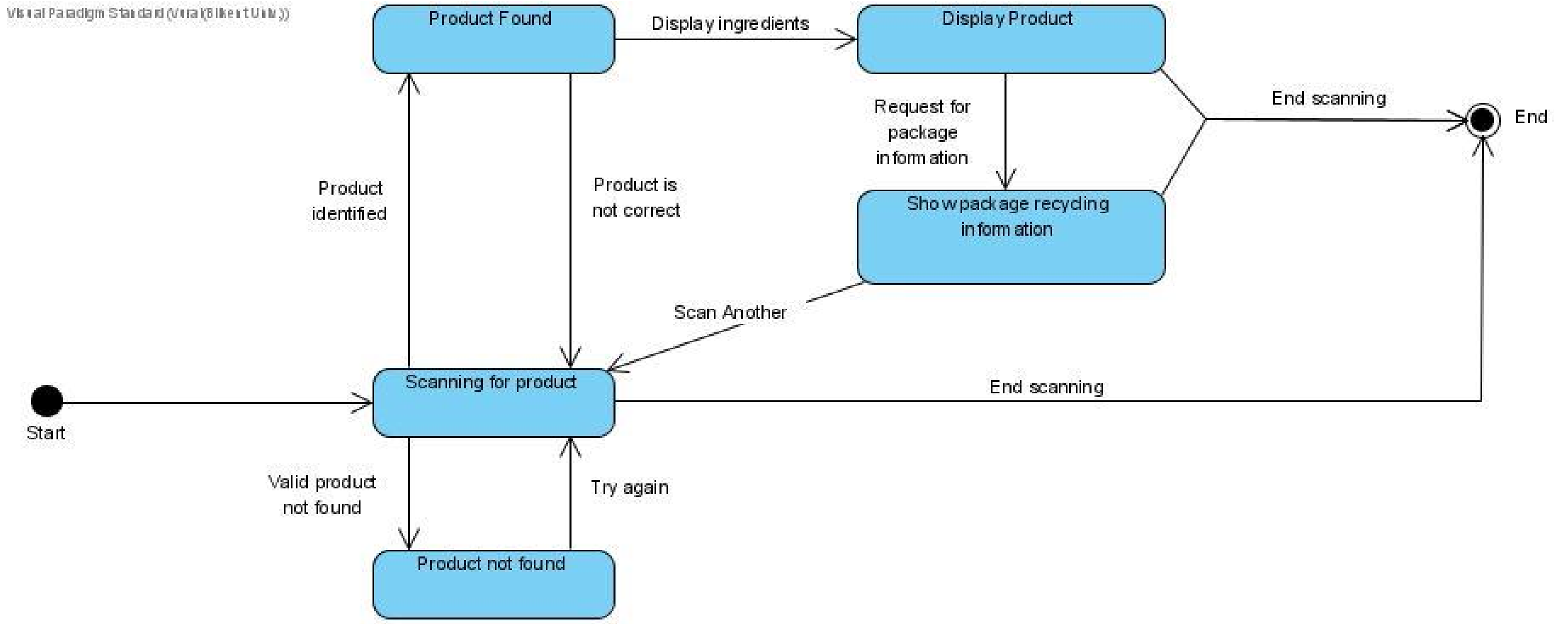


Figure 5: State Diagram

Second diagram shows the communication between the database and the app. When a product has been identified, the application requests the ingredient information from the database. The database returns the ingredient information and the app comes back to idle state for new requests until the application is closed by the user.

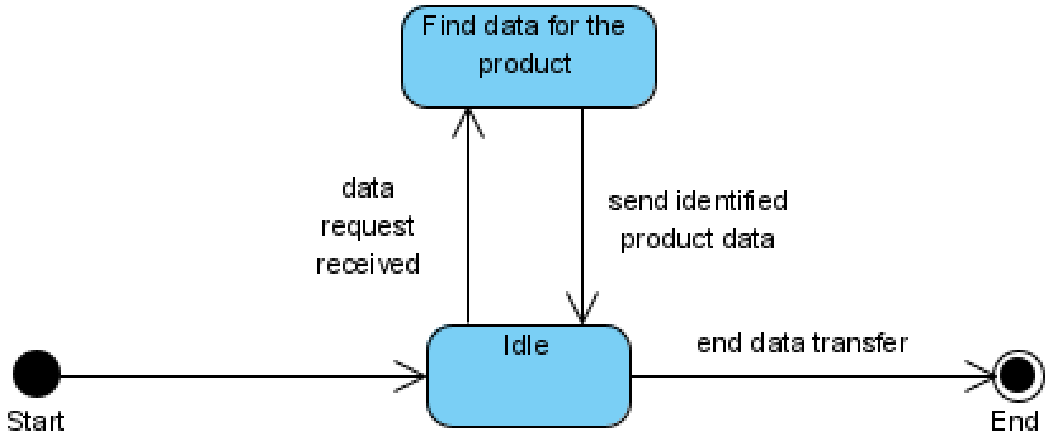


Figure 6: State Diagram of application request

Third diagram shows the product identification process in the backend application. The backend application has two states. In idle state the application waits for a product identification request and when received the request, the application goes to identify image state. After the identification of the image the identified product will be sent to the application and goes back to the idle state until the app is closed.

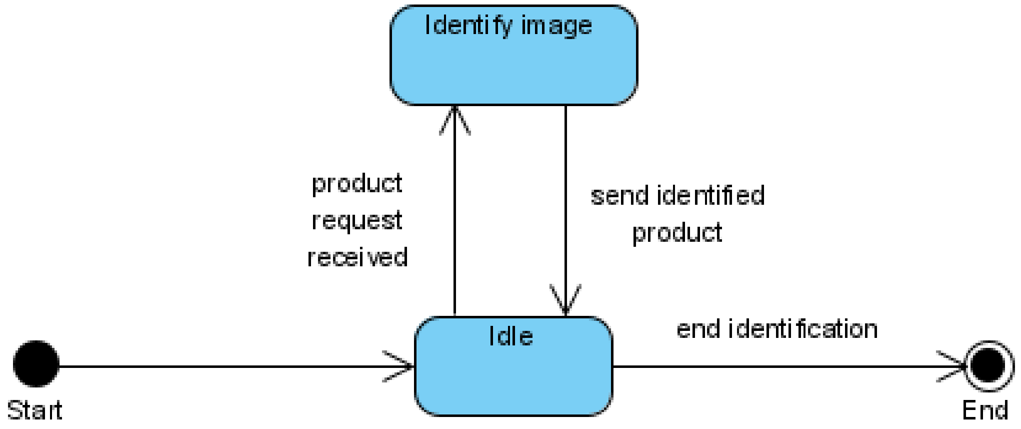


Figure 7: State Diagram of Identify Image

3.5.4.3 Sequence Diagrams

3.5.4.3.1 Scan Product

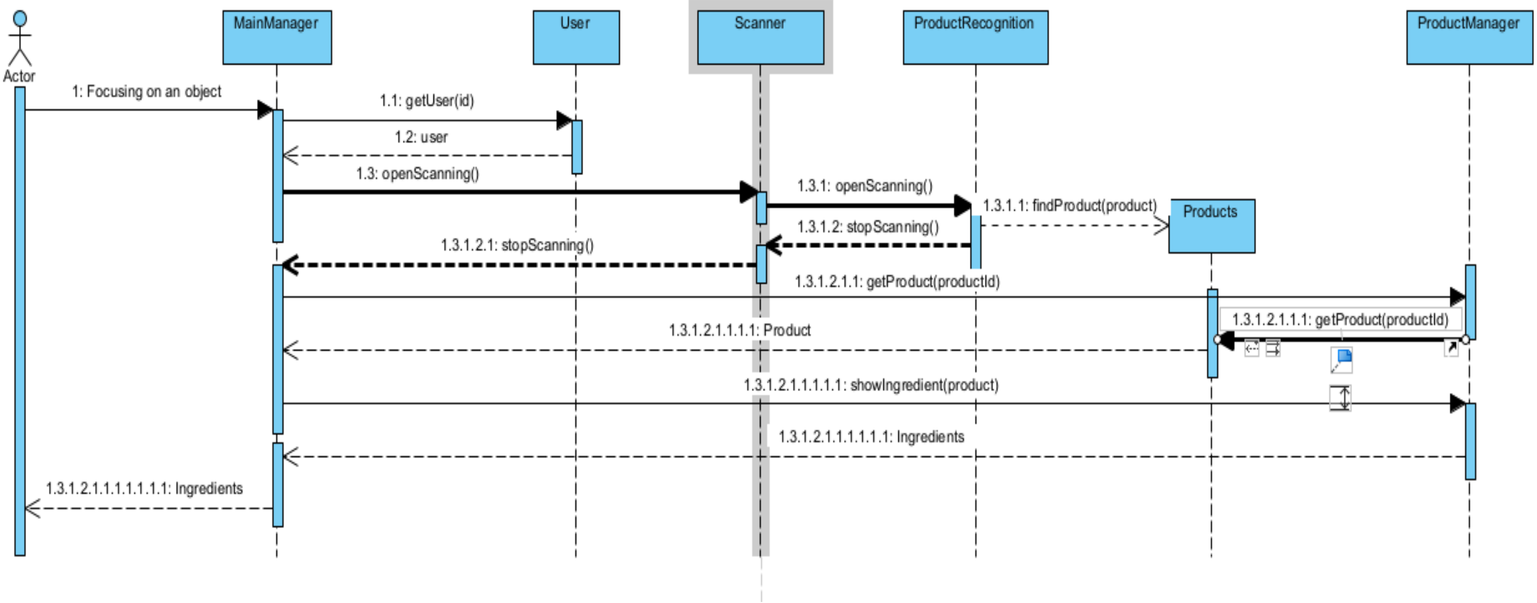

Actor
MainManager
User
Scanner
ProductRecognition
ProductManager
1: Focusing on an object
1.1: getUser(id)
1.2: user
1.3: openScanning()
1.3.1: openScanning()
1.3.1.1: findProduct(product)
Products
1.3.1.2: stopScanning()
1.3.1.2.1: stopScanning()
1.3.1.2.1.1: getProduct(productId)
1.3.1.2.1.1.1: getProduct(productId)
1.3.1.2.1.1.1.1: Product
1.3.1.2.1.1.1.1.1: showIngredient(product)
1.3.1.2.1.1.1.1.1.1: Ingredients
1.3.1.2.1.1.1.1.1.1.1: Ingredients

Figure 8: Sequence Diagram of Scanning a product

When the user focuses an object with the camera of the application, InPackt catches the image and gets the user id at the same time. While focusing, image and user information will be passed MainManager and triggers openScanning(). openScanner() triggers Scanner and ProductRecognition to find the product from the database. After finding the product MainManager asks product to ProductManager and gets the product with getProduct(product). After getting the product, MainManager asks for ingredients to ProductManager with showIngredients(product). It returns Ingredients information to the user.

3.5.4.3.2 View Suggestions

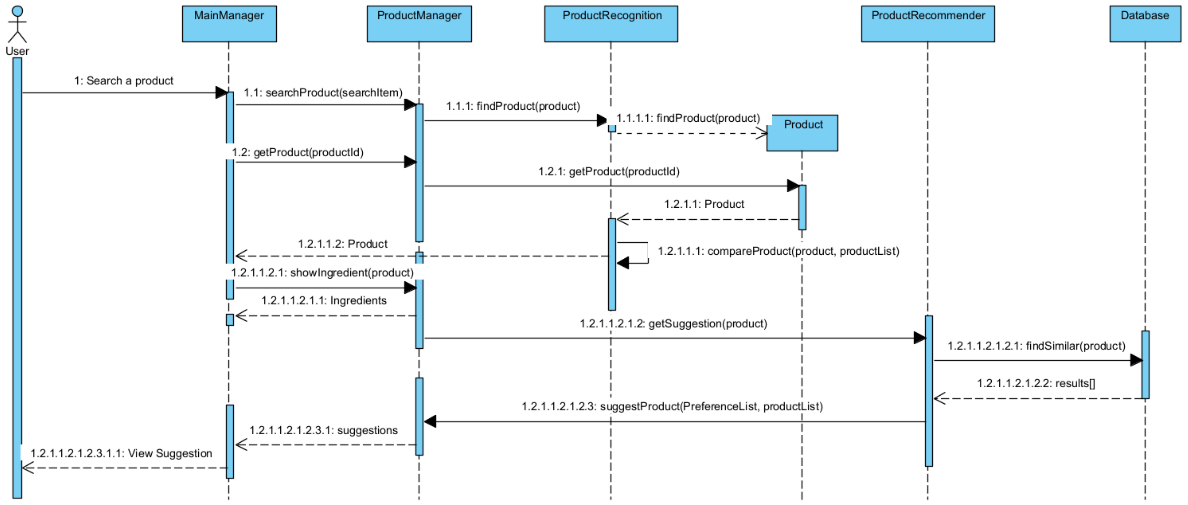

User
MainManager
ProductManager
ProductRecognition
ProductRecommender
Database
Product
1: Search a product
1.1: searchProduct(searchItem)
1.1.1: findProduct(product)
1.1.1.1: findProduct(product)
1.2: getProduct(productId)
1.2.1: getProduct(productId)
1.2.1.1: Product
1.2.1.1.1: compareProduct(product, productList)
1.2.1.1.2: Product
1.2.1.1.2.1: showIngredient(product)
1.2.1.1.2.1.1: Ingredients
1.2.1.1.2.1.2: getSuggestion(product)
1.2.1.1.2.1.2.1: findSimilar(product)
1.2.1.1.2.1.2.2: results[]
1.2.1.1.2.1.2.3: suggestProduct(PreferenceList, productList)
1.2.1.1.2.1.2.3.1: suggestions
1.2.1.1.2.1.2.3.1.1: View Suggestion

Figure 9: Sequence Diagram of Viewing suggestions

When the user searches a product, MainManager triggers searchProduct(searchItem) to ProductManager. ProductManager triggers findProduct(product) to ProductManager for finding the product from the database. After finding the product, It compares the other products with ProductManager and sends it to MainManager. MainManager shows ingredients and triggers getSuggestion(product) to ProductRecommender. ProductRecommender triggers findSimilar(product) to find similar products from the database using kNN algorithm. Database sends results and ProductRecommender lists suggested products with suggestProduct(PreferenceList, productList). After that, it sends suggestions to the Main Manager and user.

3.5.5 User Interface

The Mock-Ups of the user interface are given below in the relevant subtitles. The figures were created using moqups.com.

3.5.5.1 Welcome Page

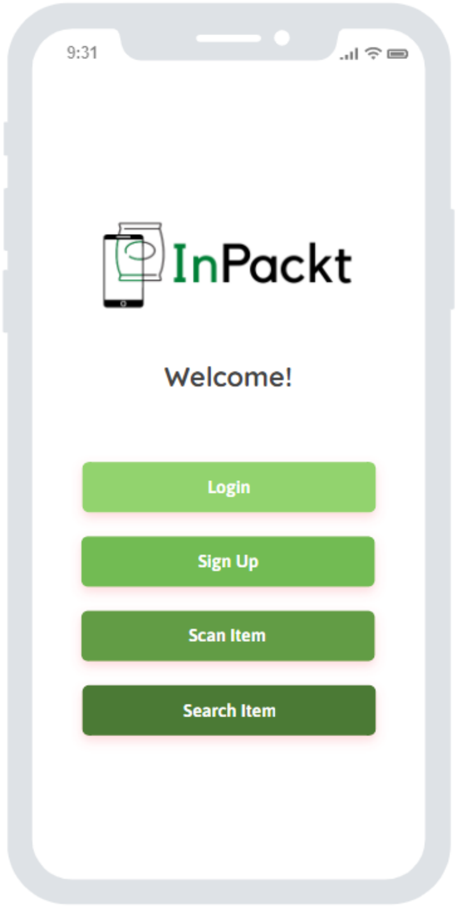


Figure 10: Mock-Up of Welcome Page

The users will be presented with this Welcome Page when they first enter the app, or enter the app when logged out. The activity options that the user can choose from are login, sign up, scan an item or search an item. We wanted to give the user the flexibility of choosing to not create an account in order to see the information in the database, however an account is necessary for preference selection and personalized recommendations which are explained below as well as the pages that these buttons forward the user to.

3.5.5.2 Login Page

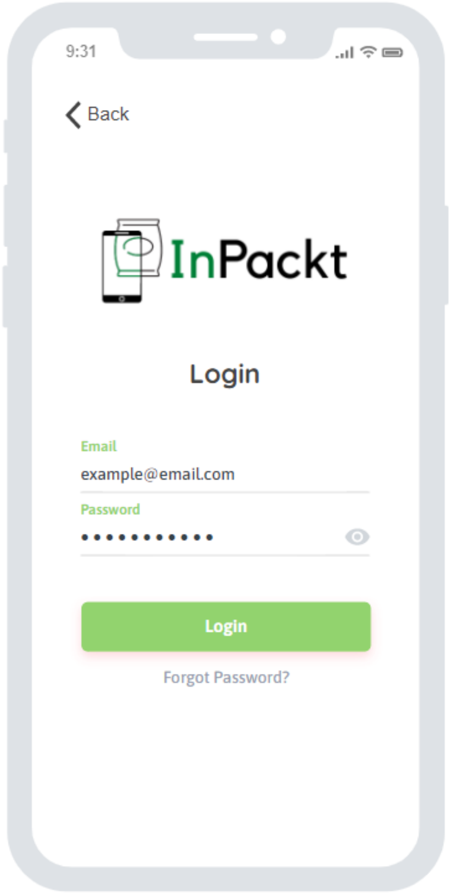
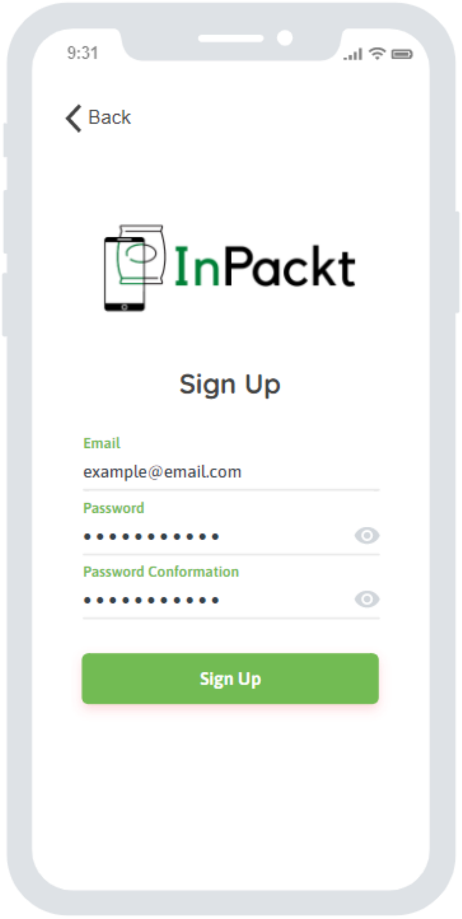


Figure 11: Mock-Up of Login Page

The user can access this page from the Welcome Page. In order to Login, they need to provide the account information (email and password) of an already created account. Otherwise, the user will be presented with an error pop up specifying whether the email or password is incorrect, and will be given the option to reenter the information. The user can also select the Forgot Password option which forwards the user to another page explained below. After logging in, the user will be directed to the Home Page, as explained below.

3.5.5.3 Sign Up Page



Email
example@email.com
Password
•••••••••••
Password Conformation
•••••••••••

Figure 12: Mock-Up of Sign Up Page

The users can Sign Up on this page with only a valid email and password. For security reasons, there will be a password confirmation input as well. The users can access this page from the Welcome Page and afterwards, they will be directed to the Home Page. New users will be assigned a random username and an empty picture which they can change any time from the Account Setting Page. By clicking the back button, the users are forwarded to the Welcome Page.

3.5.5.4 Forgot Password Page

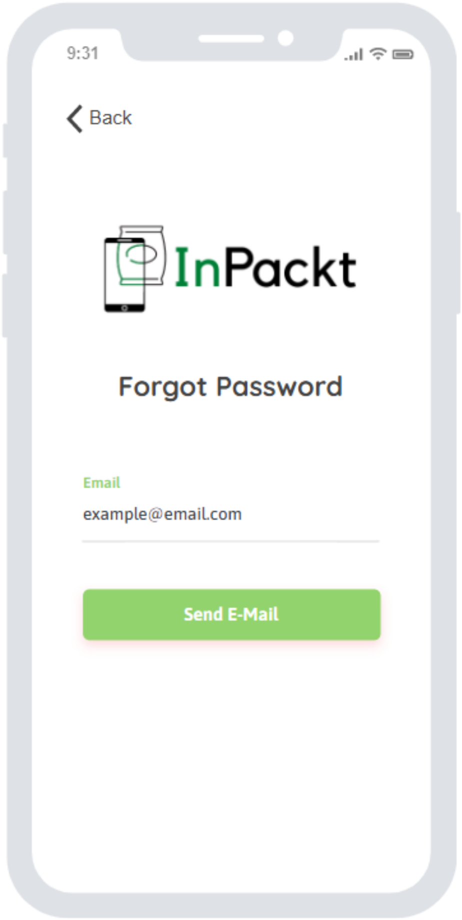


Figure 13: Mock-Up of Forgot Password Page

The users will be able to access the Forgot Password Page from the Login Page given that they have clicked the Forgot Password? prompt. Here, they need to enter a valid email address which is registered within our system in order to get an email with password reset instructions for security. If the email entered is not valid, an error message will pop up and the user will be given the option to re-enter their email. By clicking the back button, the users are forwarded to the Welcome Page.

3.5.5.5 Home Page

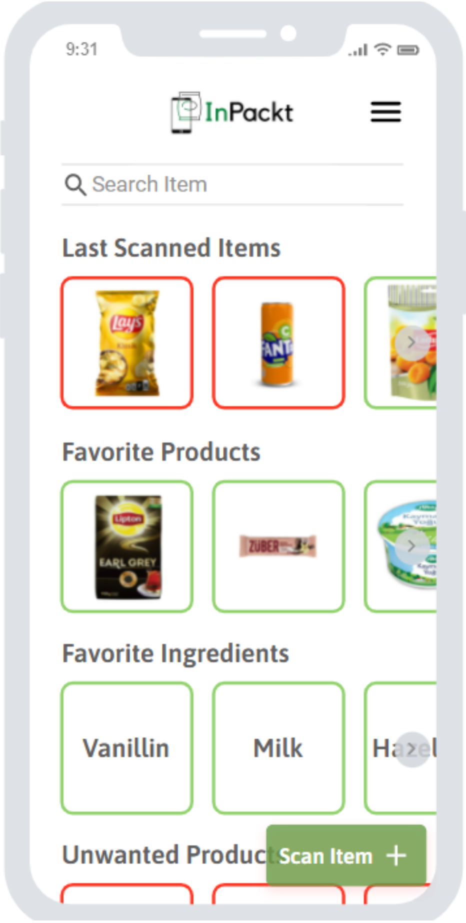


Figure 14: Mock-Up of Home Page

This Page can be accessed from Login and Sign Up pages if the user was not logged in when opening the app, or just by opening the app if the user was logged in. Here, the user gets a lot of options for action to choose from. They can browse their past scanned items and their preferences of favorite products, favorite ingredients, unwanted products and unwanted ingredients by scrolling to the sides on the related cards. They can also click the scan item button on the bottom right and be directed to the Scan Product Page. From the top bar, they can search an item (product or ingredient) manually for convenience. Also, by clicking the top right button they can open the Side Menu.

3.5.5.6 Side Menu

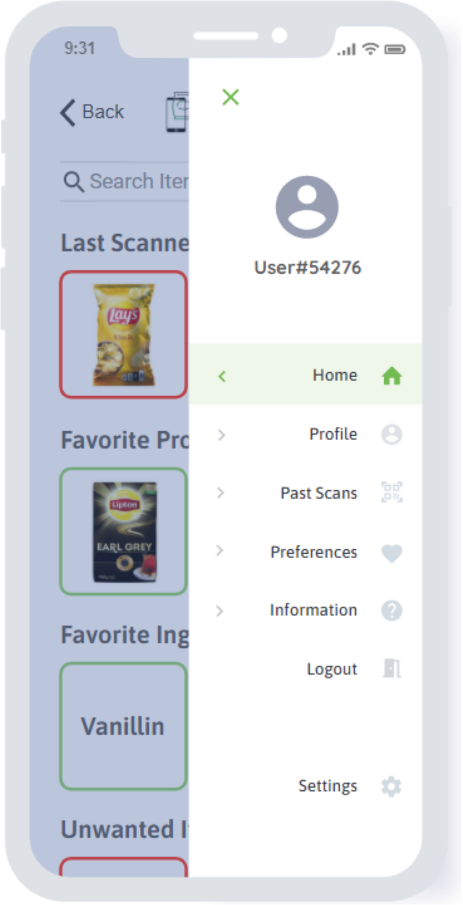


Figure 15: Mock-Up of Side Menu

This Menu can be accessed from any main page of the application. It is the main form of navigation and provides easy access to the main pages. The accessible pages are in order: home, profile, past scans, preferences, information and settings. The users are also presented with the option to Logout from this Side Menu. The chosen or assigned profile picture and username is also displayed in this menu. To exit, they can either click on the “x” on the top left of the menu or click anywhere off the menu.

3.5.5.7 Product Scanning Page

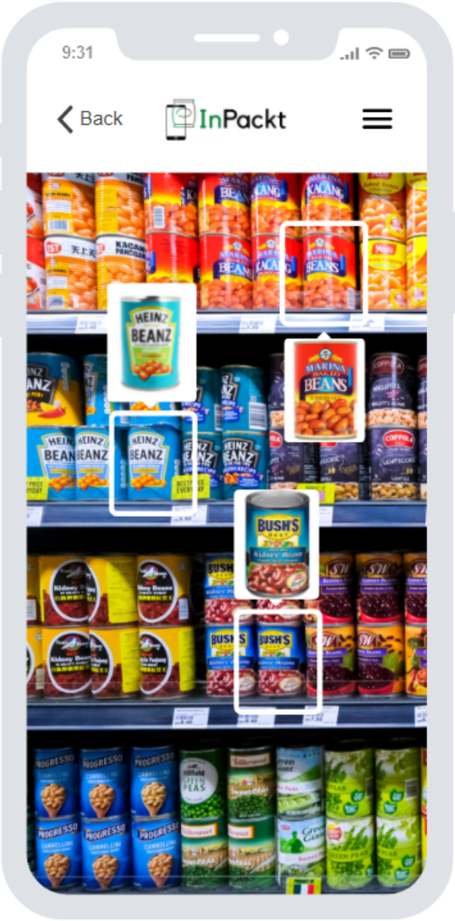


Figure 16: Mock-Up of Product Scanning Page

The user can access the Product Scan Page from the Welcome Page, Home Page, Past Scans Page and any of the preference pages (favorite products, etc.). The camera opens in this page if the access is allowed. If not, a pop up asking for access to the camera is shown. The users can scan an item by just pointing their cameras toward the item and the recognized product is shown either above or below the item depending on the page placement. They can go back to their previous page with the back button or click the side menu button for the menu.

3.5.5.8 Barcode Scanning Page

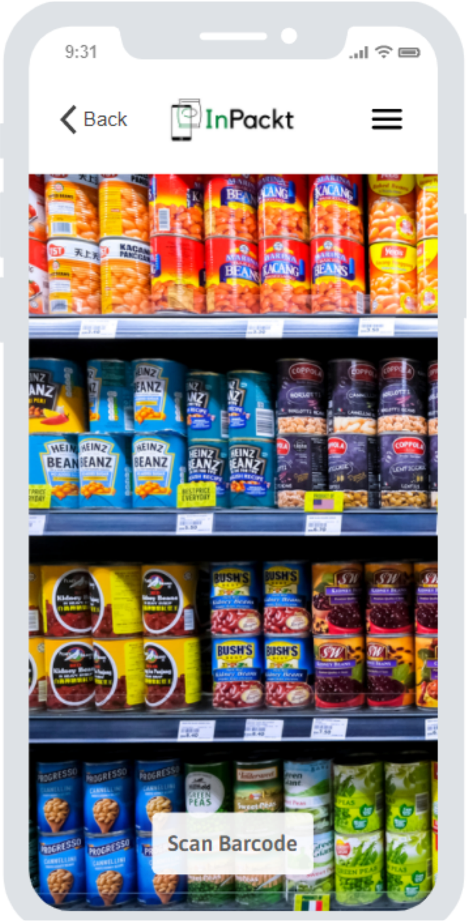


Figure 17: Mock-Up of Barcode Scanning Page

This page can only be accessed from the Product Scanning Page if and only if the application is not able to recognize any product for a certain number of seconds (5, to be exact). After the time has passed in the Product Scanning Page, the pop up on the bottom middle of Figure 17 is displayed which will direct the user to scan the barcode of the item if they please. The user, again, can go back to their previous page with the back button or click the side menu button for the menu.

3.5.5.9 Switch to Search from Scan Page

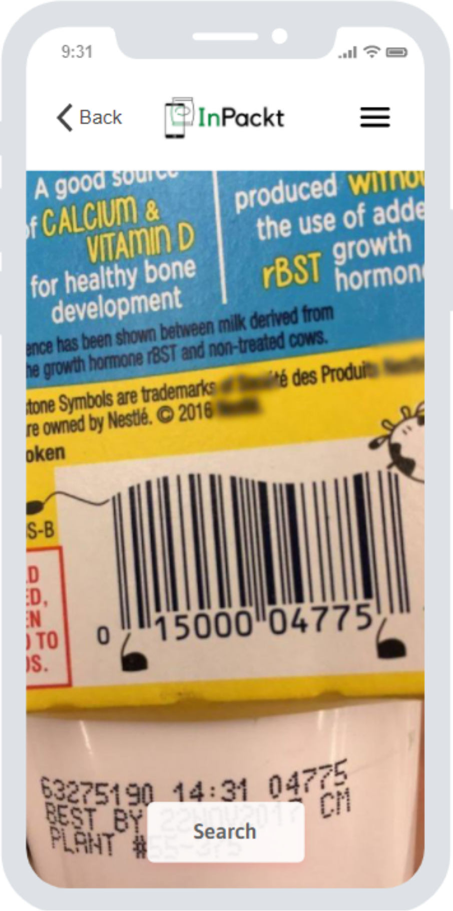
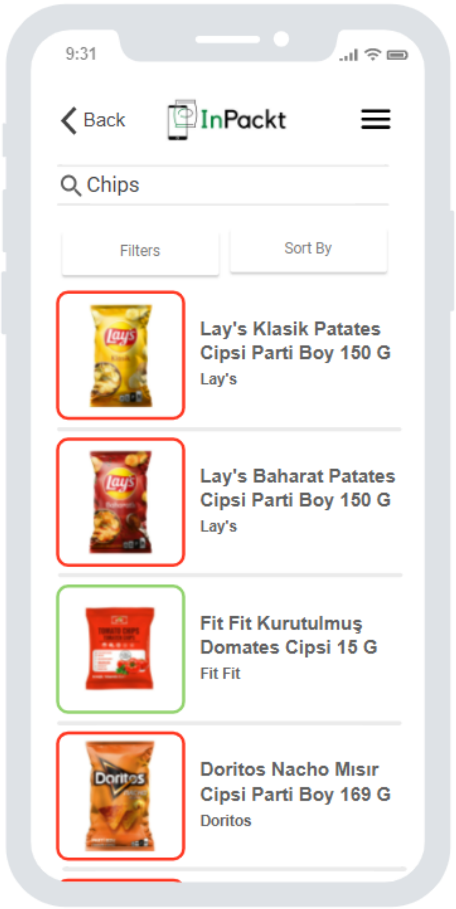


Figure 18: Mock-Up of Switch to Search from Scan Page

If the user is unable to scan the barcode of the product in the Barcode Scanning Page, they will be directed to this page after 5 more seconds. Here, a button called "Search" pops up which will direct users to the search page explained below. The user, again, can go back to their previous page with the back button or click the side menu button for the menu.

3.5.5.10 Search Page



Lay's Klasik Patates Cipsi Parti Boy 150 G
Lay's

Lay's Baharat Patates Cipsi Parti Boy 150 G
Lay's

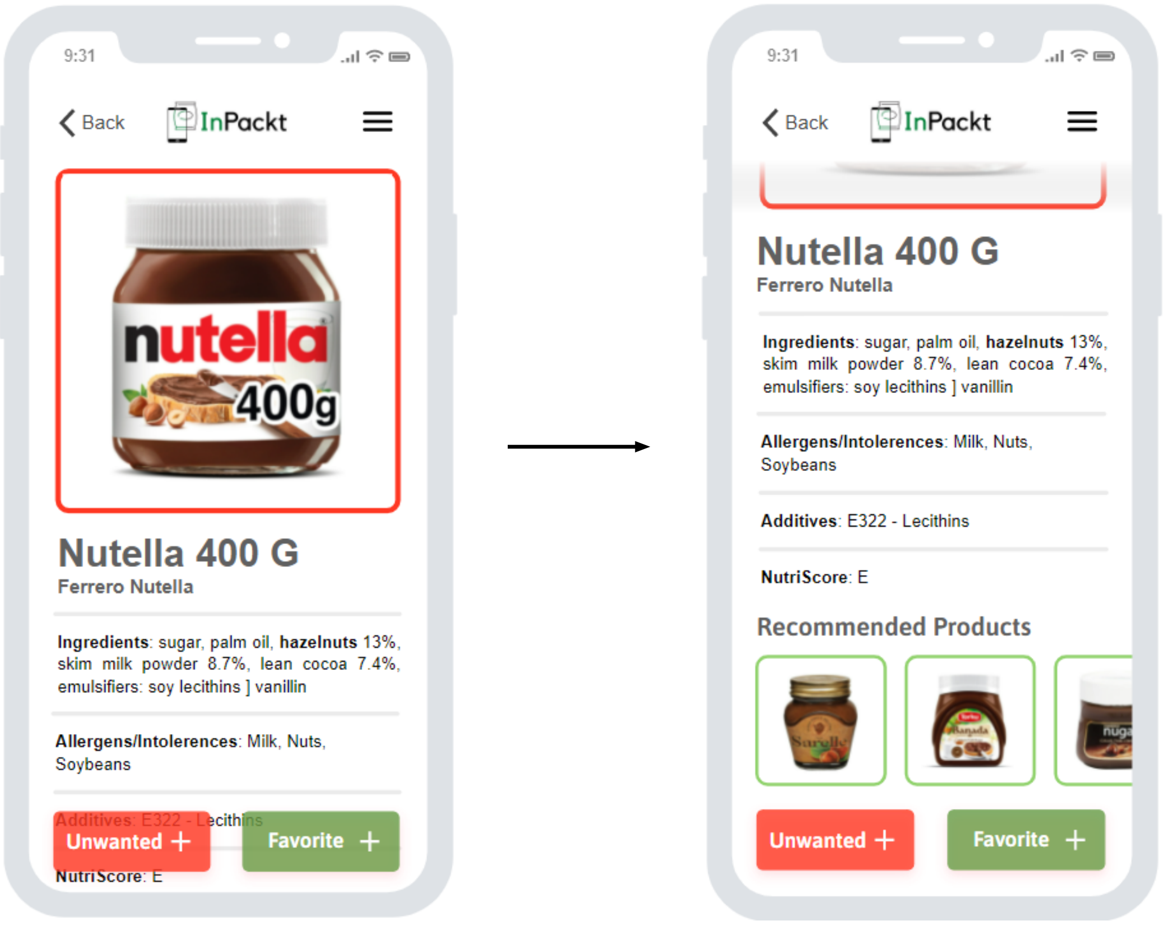
Fit Fit Kurutulmuş Domates Cipsi 15 G
Fit Fit

Doritos Nacho Mısır Cipsi Parti Boy 169 G
Doritos

Figure 19: Mock-Up of Search Page

The item search page can be accessed from various pages such as Welcome Page, Home Page and any of the preference pages' pop ups (explained below in the Add New Favorite Product section). The user can put the input text on the top search bar and hit enter to get the items. They can filter and sort the products to their own wishes with the buttons below the search bar. The users can go back with the back button and access the side menu as well.

3.5.5.11 Product Page



Nutella 400 G

Ferrero Nutella

Ingredients: sugar, palm oil, **hazelnuts** 13%, skim milk powder 8.7%, lean cocoa 7.4%, emulsifiers: soy lecithins] vanillin

Allergens/Intolerences: Milk, Nuts, Soybeans

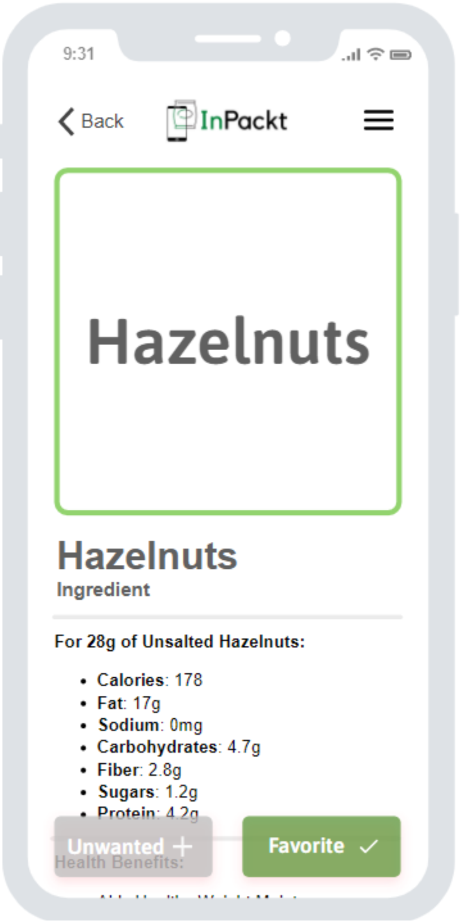
Additives: E322 - Lecithins

NutriScore: E

Figure 20: Mock-Ups of Product Page

The product page can be accessed from anywhere the user clicks on a product from. Here, the information of the product is shown from the database. The user can add the product to their unwanted or favorite lists (not both) by clicking on the related button on the bottom. Scrolling down on the page reveals the recommended products (related to the product displayed on the page) to the user based on the personal preferences such as favorites and unwanteds. The users can go back with the back button and access the side menu as well.

3.5.5.12 Ingredient Page



For 28g of Unsalted Hazelnuts:

- Calories: 178
- Fat: 17g
- Sodium: 0mg
- Carbohydrates: 4.7g
- Fiber: 2.8g
- Sugars: 1.2g
- Protein: 4.2g

Health Benefits:

Figure 21: Mock-Up of Ingredient Page

The product page can be accessed from anywhere the user clicks on an ingredient from. The relevant information is displayed regarding the ingredient, and the user can add it to their favorite or unwanted list. If the product is already in either one of the lists, the "+" sign becomes a check on that button and the other button is greyed out for understandability. The user can remove the product from the list by clicking on the colored and checked button. The users can go back with the back button and access the side menu as well.

3.5.5.13 Profile Page

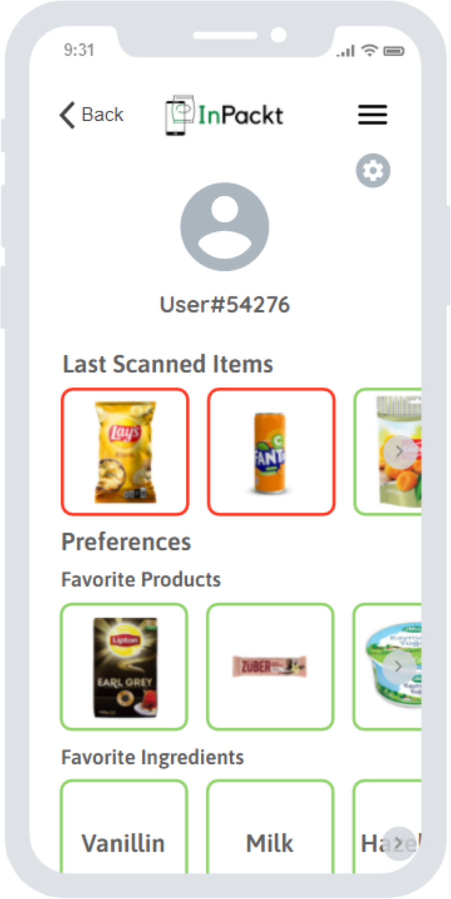


Figure 22: Mock-Up of Profile Page

The profile page can be accessed from the Side Menu on any page with access to the side menu button. Here, the users can view their profile picture, last scanned items and preferences such as favorite products, favorite ingredients, unwanted products and unwanted ingredients. The users can access the Account Settings Page explained below by clicking on the gear icon below the side menu button. They can go back by clicking the back button or view the side menu by clicking the related button.

3.5.5.14 Past Scans Page

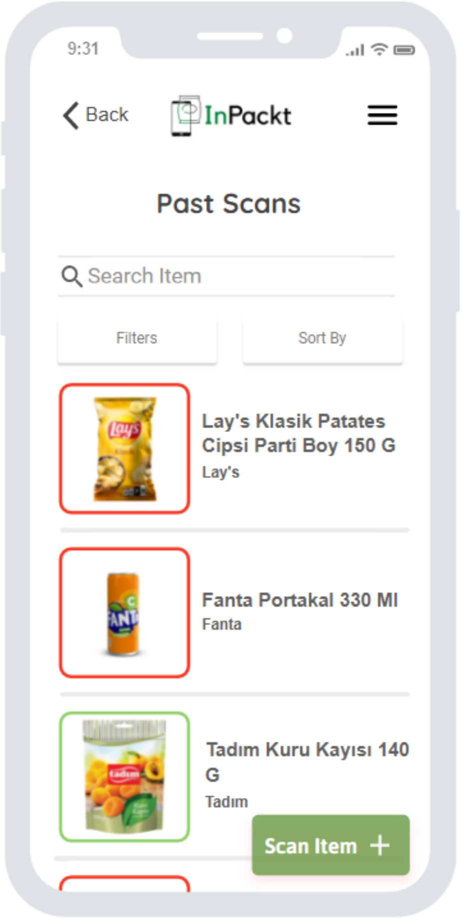


Figure 23: Mock-Up of Past Scans Page

The Past Scans Page can be accessed from any page with the side menu button, since it is present in the side menu. Here, all items the user scanned in the past are displayed. An item can be removed by swiping to the left with confirmation pop up. The items can be searched, filtered or sorted with the related action. The user can also choose to add another item to the list by clicking the scan item button on the lower right. The users can go back to their previous page by clicking the back button or view the side menu by clicking the related button.

3.5.5.15 Preferences Page

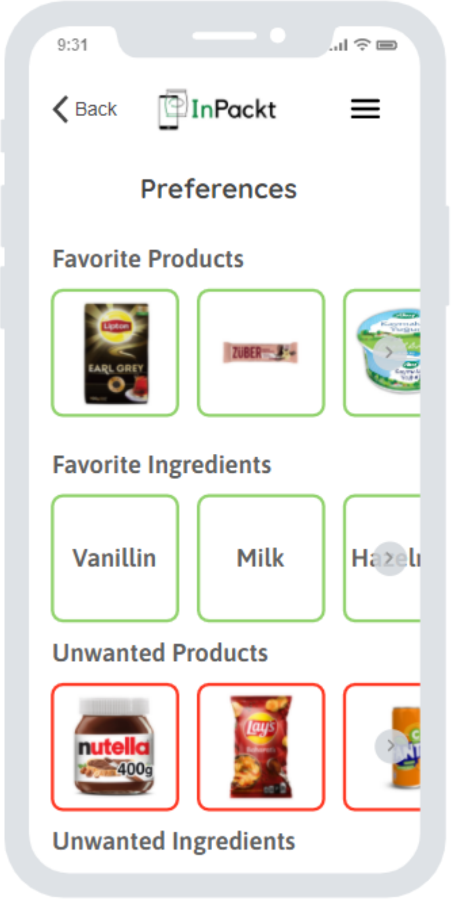


Figure 24: Mock-Up of Preferences Page

The preferences page displays the user based choices such as favorite products and so on. This page can be accessed from the side menu as well from any main page of the app. The user can click on any subheader (e.g. Favorite Ingredients) to go to that specific preference's page which is explained below. The users can go back to their previous page by clicking the back button or view the side menu by clicking the related button.

3.5.5.16 Favorite Products Page

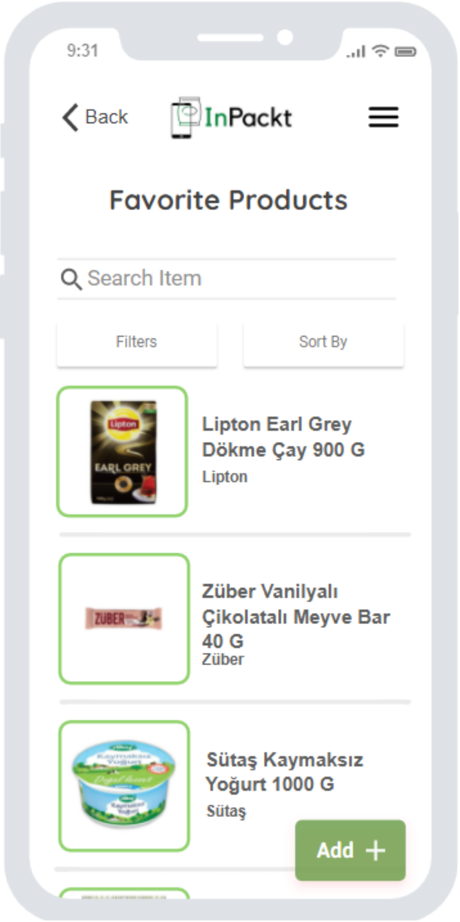

9:31
Back
InPackt
Favorite Products
Search Item
Filters
Sort By
Lipton Earl Grey Dökme Çay 900 G
Lipton
Züber Vanilyalı Çikolatalı Meyve Bar 40 G
Züber
Sütaş Kaymaksız Yoğurt 1000 G
Sütaş
Add +

Figure 25: Mock-Up of Favorite Products Page

This is one of the specific preference pages (total of 4) that the user has access to. This page can be accessed from the Home Page, Profile Page or the Preferences Page. Here, the items can be searched, filtered or sorted as well. The other preference pages are omitted from the mock-ups for avoiding repetition since they all basically look the same. The user can add a product to the specific list by the add button on the lower right. The button pops up a menu explained in the next section. The users can go back to their previous page by clicking the back button or view the side menu by clicking the related button.

3.5.5.17 Add New Favorite Pop Up

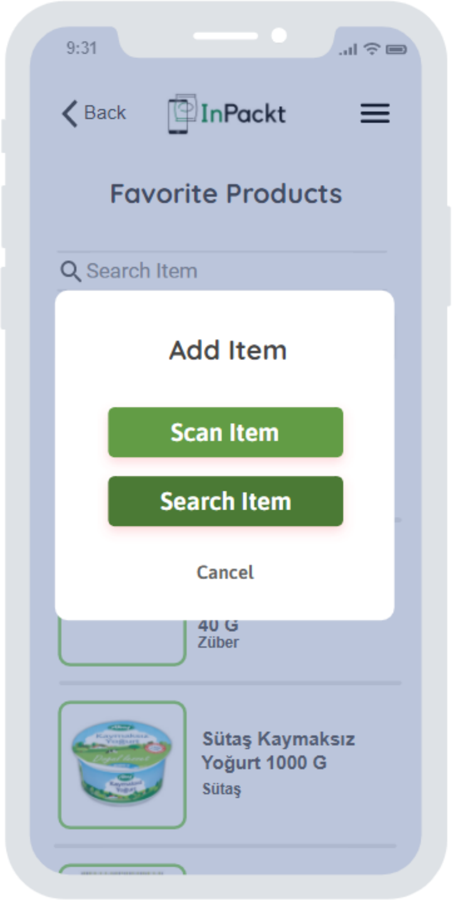
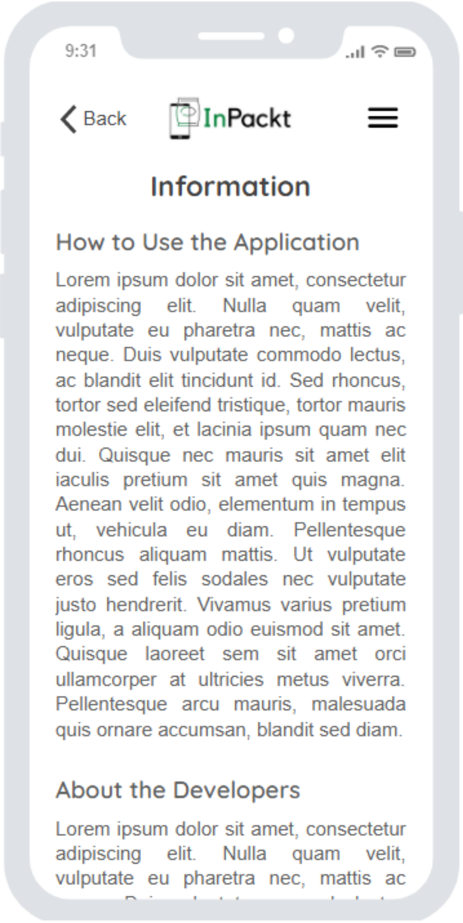


Figure 26: Mock-Up of Add New Favorite Pop Up

This pop up can be accessed from any of the four preference pages. It's main functionality is to choose between scanning or searching for a product to be added to the mentioned preference list. The user can close this pop up by clicking the cancel prompt on the bottom of the pop up.

3.5.5.18 Information Page



Information

How to Use the Application

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque rhoncus aliquam mattis. Ut vulputate eros sed felis sodales nec vulputate justo hendrerit. Vivamus varius pretium ligula, a aliquam odio euismod sit amet. Quisque laoreet sem sit amet orci ullamcorper at ultricies metus viverra. Pellentesque arcu mauris, malesuada quis ornare accumsan, blandit sed diam.

About the Developers

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac

Figure 27: Mock-Up of Information Page

This page can be accessed from anywhere with the side menu button since it is in the menu. It displays information such as how to use the app and who the developers are. For mock-up purposes, this page is not fully filled and a placeholder text is put in its place. The users can go back to their previous page by clicking the back button or view the side menu by clicking the related button.

3.5.5.19 Settings Page

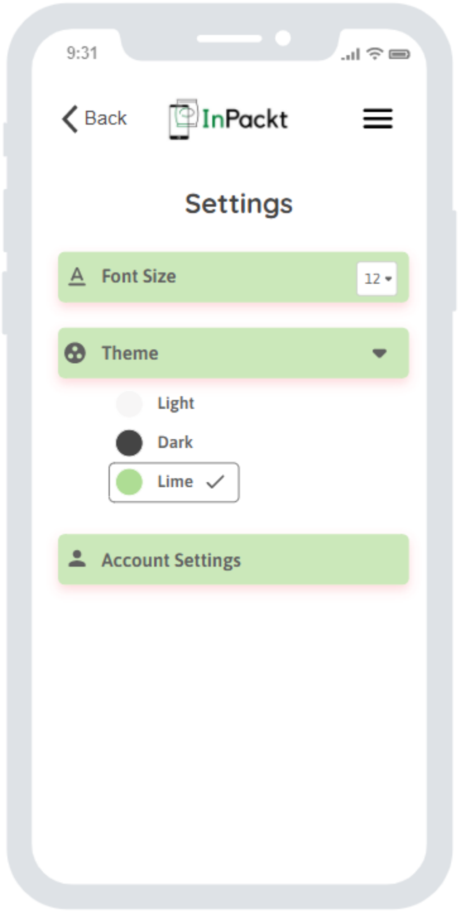
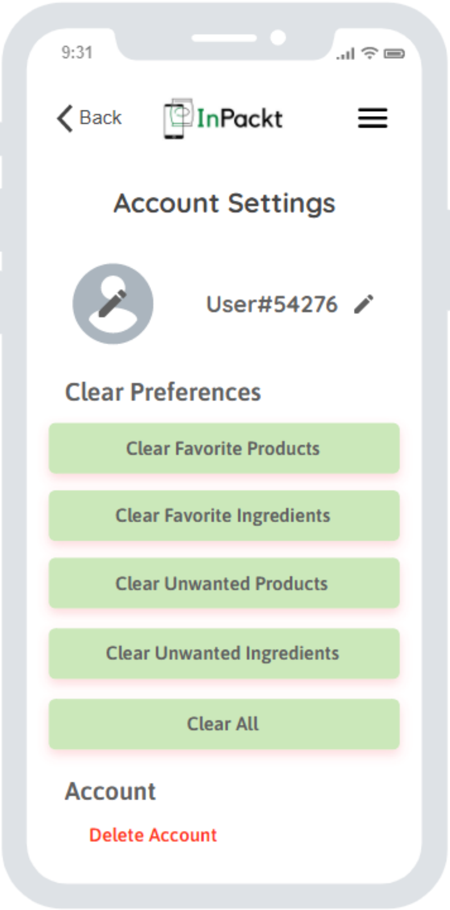


Figure 28: Mock-Up of Settings Page

This page can be accessed from the side menu as well, from most of the main pages of the application. The font size and color theme of the app can be changed here. The account settings page explained below is also accessible through here. The users can go back to their previous page by clicking the back button or view the side menu by clicking the related button.

3.5.5.20 Account Settings Page



9:31

Back

InPackt

Account Settings

User#54276

Clear Preferences

Clear Favorite Products

Clear Favorite Ingredients

Clear Unwanted Products

Clear Unwanted Ingredients

Clear All

Account

Delete Account

Figure 29: Mock-Up of Account SettingsPage

This page can be accessed from the Profile Page or the Settings Page. The profile picture and username can be changed through the edit icons presented on/next to the related fields. The preferences can be completely cleared on each specific preference or as a whole. The account can be deleted as well. The clearance and deletion will pop up a confirmation prompt from the user to take place. The users can go back to their previous page by clicking the back button or view the side menu by clicking the related button.

4. Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

As the name of the application implies, our project has an impact on many keystones of human life such as health, safety, welfare, global, cultural, social, environmental and economical. Effects of the algorithm in various categories are discussed below.

4.1.1 Public Health Factors

One of the main uses of the InPackt is to show the users unhealthy ingredients and help users to avoid preferable ingredients. Therefore, public health will be enhanced by InPackt as the user scanning products.

4.1.2 Public Safety Factors

The required data which are username, email password will not be shared anywhere and the safety of the information will be considered. The required data will be used for storing the preference list of the user.

4.1.3 Public Welfare Factors

InPackt doesn't have negative impact on public welfare. On the other hand, it can impact positively because InPackt will guide the users to do better shopping for themselves. Moreover, it will be a free application. Therefore, the application might increase public welfare with its guidance and affordability.

4.1.4 Global Factors

InPackt will focus on as many markets as possible in the world, so products will not be bought only from Turkey. Moreover, the application language will be English and this helps to be adapted easily to other countries. Therefore, by focusing on global market and main language of the application, InPackt has a global perspective as well.

4.1.5 Cultural Factors

InPackt will not be affected by cultural factors because it does not have cultural features to affect.

4.1.6 Social Factors

InPackt interacts directly with users instead of other users like social networks. Therefore, InPackt will not be affected by social factors.

4.1.7 Environmental Factors

InPackt will show the product packaging type which will be related to recycling materials and harmful material for the environment. Therefore, InPackt increase the awareness of the environmental issues and also it increases the sustainability of the products.

In the table below, each factor's levels are indicated and each factor is evaluated between 0 and 10.

	Level
Public Health Factors	9
Public Safety Factors	8
Public Welfare Factors	6
Global Factors	4
Cultural Factors	0
Social Factors	0
Environmental Factors	5

Table 1: The factors and its levels

4.2 Risks and Alternatives

The risks that may arise during project development and the plan B's given that the risk occurs are explained below.

4.2.1 Database Collapse

Database systems may collapse due to unexpected human made errors, or overloads. This may cause malfunctioning in our application since the application highly depends on the information in the database. Although the possibility of a database crash is not high, if a collapse occurs postgresql has a restoration function that saves the collapsed data[7]. The regular system backups is another option to use if any collapse occurs in the database.

4.2.2 Ingredient Change of a Specific Product

The production companies may change the ingredients of their products without our attention. This may cause some minor problems such as eating some unwanted ingredients or some major complications such as death from allergy to the new ingredient that the user may not realise. Even though the possibility of this risk occuring is low, since the outcome of it can be extreme, there must be a precaution so that the users will see the correct ingredients of the products. The alternative plan is to make sure the database is regularly and consistently updated through both the developers and the community, so that the information is accurate. If the user sees an incorrect ingredient list for a product, they will be able to send feedback with the correct information.

4.2.3 API Crash

The object recognition API may crash. The possibility of occurrence for this risk is low, depending on the use rate of the object recognition API that the application uses. A user information message will be displayed about this crash and users will be directed to search a product page. Users can continue to use the app via direct search feature.

4.2.4 Recognition Mismatch

The object recognition may not always work as intended when a user scans an item. Especially for products with similar packaging, the wrong item may be displayed to the user. The possibility of occurrence for this risk is medium, depending on our implementation of object recognition. In order to ensure the user is getting the right product on their screen, there must be an approval step in the object recognition phase where the user either accepts or denies a product displayed to them. This will help us to understand where the error lies in the object recognition system and try to fix it as soon as possible.

4.2.5 Unfamiliarity with the Technology

The major functionalities in the project require deep knowledge in some important areas of computer science such as computer vision and machine learning. However, since we are taking the courses related to these topics at the same time as the project, relying on school courses is not enough. The possibility of this risk occuring is high since none of us have previous experience in these fields. We need to learn these topics as fast, complete and through as possible. To achieve this, we need to spend some hours on learning both individually and together to succeed on the project. The faster the learning commences, the faster the project can progress.

4.2.6 Design and Architecture Inefficiency

The design and architectural structure of the project may not always work at the desired speed and efficiency. The probability of such a risk occurring in the tests we will carry out during the project stages is at an medium level. If such a situation is encountered, we can list the effects on the user as follows: the user may reach the content information of the product scanned late, or if the product does not fit the user criteria, the recommendations to be presented to the user by the application may reach the user much later than the requested time. These will cause the application to move away from the principle of facilitating the user's life and reducing the usability of the application. If we encounter such a problem during the testing stages of the application, we develop a new design and architectural structure as an alternative plan and make the application work more effectively.

4.2.7 Time Shortage

Due to some unexpected errors or bugs on the key parts of the project, the time shortage can occur for the incomplete parts. To cope with the unexpected errors, our team can save some time from minor features of the project that do not affect the functioning of the application directly.

Risk	**Possibility of Occurrence**	**Effect of Risk**	**Plan B**
Database Collapse	Medium	The user cannot access the information about products.	Backup database
Ingredient Change of a Specific Product	Low	The user encounters incorrect ingredients of products.	Updated product information via user feedback
API Crash	Low	The user is unable to scan the product.	Inform users about current error
Recognition Mismatch	Medium	The user accesses information of a different product than the one they scanned.	User approval of the product
Unfamiliarity with the Technology	High	The progress of the project may be stalled which may prevent the project from being completed in the required time.	Spend individual and collaborative learning hours on unfamiliar technology.
Design and Architecture Inefficiency	Medium	The user receives information with time-delay.	Changing the design and architecture depends on the search on the Internet

Time Shortage	Medium	Some functions of the project may not be complete, which causes the project to not work fully functional.	Planned, strict working hours on the project to complete significant requirements

Table 2: Risks, Possibility of Occurrence, Risk's Effects and Plan B's.

4.3 Project Plan

Below you can see various tables that you will make.

WP#	Work package title	Leader	Members involved
WP1	Analysis and High-Level Design Decisions	Kadir Mert Laleci	Everybody
WP2	Database Development and Data Processing	Fadime Selcen Kaya	Everybody
WP3	Object Recognition Development	Şükrü Can Erçoban	Everybody
WP4	Machine Learning For Product Suggestion	Kutsal Bekçi	Everybody
WP5	Low-Level Design	Kadir Mert Laleci	Everybody
WP6	User Interface - Demo	Vural Doğan Akoğlu	Everybody

Table 3: List of work packages

<table>
<tr><td colspan="4"><b>WP 1:</b> Analysis and High-Level Design Decisions</td></tr>
<tr><td colspan="4"><b>Start date:</b> <i>01.11.2021</i> <b>End date:</b> <i>24.12.2021</i></td></tr>
<tr><td><b>Leader:</b></td><td>Kadir Mert Laleci</td><td><b>Members involved:</b></td><td>Fadime Selcen Kaya
Şükrü Can Erçoban
Kutsal Bekçi
Vural Doğan Akoğlu</td></tr>
<tr><td colspan="4"><b>Objectives:</b> Flutter will be learned and libraries to be used will be determined</td></tr>
<tr><td colspan="4"><b>Tasks:</b>
<b>Task 1.1 Analysis Report:</b> Determining the basic parts of the project such as the functions of the project, necessary diagrams, mockups that give an idea of how I will look when finished, the work distribution of the project
<b>Task 1.2 High-Level Design Report :</b> Converting the analysis report into a system design model. Examining the architectural structure of a similar project and determining the strategies and design goals for the hardware-software environments where our innovative project will work.</td></tr>
<tr><td colspan="4"><b>Deliverables</b>
<b>D1.1:</b> Analysis Report:
<b>D1.2:</b> High-Level Design Report</td></tr>
</table>

Table 4: Work package 1

WP 2: Database Development and Data Processing			
Start date: *25/11/2021* **End date:** *27/12/2021*			
Leader:	*Fadime Selcen Kaya*	**Members involved:**	Şükrü Can Erçoban Kadir Mert Laleci Kutsal Bekçi Vural Doğan Akoğlu
Objectives: *Processing of data from our main data source, Open Food Facts. Creating a functional and accurate database with as many products as available. Creating a web scraper that completes the database with needed information.*			
Tasks: ***Task 2.1 Process Data*****:** *Process the data obtained from Open Food Facts into a unified form* ***Task 2.2 Create Database*****:** *Create the database and add items processed from sources* ***Task 2.3 Web Scraping*****:** *Complete the database as much as possible with web scraping*			
Deliverables ***D2.1:*** *Database* ***D2.2:*** *Web Scraper*			

Table 5: Work package 2

WP 3: Object Recognition Development			
Start date: 25/11/2021 **End date:** 27/12/2021			
Leader:	*Şükrü Can Erçoban*	**Members involved:**	Fadime Selcen Kaya Kadir Mert Laleci Kutsal Bekçi Vural Doğan Akoğlu
Objectives: One of the main functionality of the application is scanning a product and the products can be packaged products in the markets. The application will detect and understand the product from the image and it will show its contents. Detected product will be compared from the database and the web search. To detect products accurately, we should train the recognition engine with a variety of products. To test the model, we will implement a basic interface.			
Tasks: **Task 3.1** Development **Task 3.2** Train the model **Task 3.3** Demo			
Deliverables **D3.1:** Detect a product accurately **D3.2:** Performance of the Detection			

Table 6: Work package 3

WP 4: Machine Learning For Product Suggestion			
Start date: *15/02/2022* **End date:** *15/04/2022*			
Leader:	*Kutsal Bekçi*	**Members involved:**	Şükrü Can Erçoban Kadir Mert Laleci Fadime Selcen Kaya Vural Doğan Akoğlu
Objectives: *It is the next stage after product recognition. If there is a prohibited nutrition, InPackt suggests similar products according to user preferences. There will be a kNN algorithm to suggest products.*			
Tasks: ***Task 4.1 Implement*****:** *Implementing kNN algorithm* ***Task 4.2 Train set*** **:** *Training the data sets* ***Task 4.3 Validation*** **:** *Validation step* ***Task 4.3 Test Set*** **:** *Testing the data sets*			
Deliverables ***D4.1:*** *Suggestions*			

Table 7: Work package 4

WP 5: Low-Level Design Decisions			
Start date: 02.11.2021 **End date:** 15.03.2021			
Leader:	Kadir Mert Laleci	**Members involved:**	Fadime Selcen Kaya Şükrü Can Erçoban Kutsal Bekçi Vural Doğan Akoğlu
Objectives: Completed low-level design report and upload within the specified time.			
Tasks: **Task 5.1** Low-level Design Report **:** Focusing on the low-level design of the project such as object design trade off, class interfaces, packages			
Deliverables **D5.1:** Low-level Design Report			

Table 8: Work package 5

WP 6: User Interface and Demo			
Start date: *01/02/2022* **End date:** *15/03/2022*			
Leader:	*Vural Doğan Akoğlu*	**Members involved:**	Fadime Selcen Kaya Kadir Mert Laleci Kutsal Bekçi Şükrü Can Erçoban
Objectives: *Design and finalise the user interface of the application that is easy to use and simple. First aim is to create the same interface as mock ups and finalise our design with the experiments. In the end, the application will be connected to the backend and ready for the demo.*			
Tasks: ***Task 6.1 Creation of the Sign up - Sign in Screens*** ***Task 6.2 Creation of the Main Screen*** ***Task 6.3 Creation of the Scan Product Interface - Product Ingredient Interface*** ***Task 6.4 Creation of the User Profiles*** ***Task 6.5 Experiments - Demo:*** *In the end, everybody will use the app for a while. Depending on the experiences, the app will be finalized and become ready for the demo.*			
Deliverables ***D6.1:*** *User Interface for Sign-up, Sign-in screens and connection with the database.* ***D6.2:*** *User Interface for the main screen and connection between the pages.* ***D6.3:*** *User Interface for the Scan product connected with the recognition API.* ***D6.4:*** *User profile pages that show the information and preferences of the user.* ***D6.5:*** *User interface of the app that is ready for the demo.*			

Table 9: Work package 6

4.4 Ensuring Proper Teamwork

Ensuring teamwork is a substantial point during project's development. All team members agreed to use development environments. There are four platforms that are used by the team.

- Jira Software: Jira Software is used in order to create and assign tasks to team members and keep track of the progress of tasks. It is also used for keeping track of the deadlines and the performance of the team members in terms of fulfilling tasks before deadlines.
- Github: Github is used to create the project's website and repository. Every team member has a Github account and can contribute to the project's repository. There are many branches to work on different parts of the project and a main branch for approved parts of the project.
- Discord: Discord is used for meetings and working together. It allows sharing screens therefore team members can easily work with seeing other team members' screens.
- Whatsapp: Whatsapp is used for continuous communication between team members wherever they are. It is used for setting meetings time, discussing the project when a member is unable to connect discord.

These are the platforms that team members can use. To ensure proper teamwork, a team member who failed or overdue a task will be questioned about the reasons for failure in a supportive way and other team members will motivate and help the member who failed.

4.5 Ethics and Professional Responsibilities

There are many ethical issues of InPackt and there are solutions to prevent ethical issues that may arise. First of all, the very first goal of the InPackt is to access ingredients of the product in an easy way and if there is any nutrition prohibited by the user in the product, suggest a similar product without that nutrition. Food companies might consider advertising on the app and want to feature their own product. To prevent this situation, user specification which is determined by user’s choices is added to the application. Therefore, when a product is suggested, it is a proper product in terms of the user's choices.

In terms of health InPackt have an impact on being more healthy since the application allows that the choice of banning harmful ingredients. If the use of the application becomes widespread, it is possible to affect food companies to produce more healthy food.

4.6 Planning for New Knowledge and Learning Strategies

There are some technological issues that the team will need while developing the application. These are:

- Machine Learning
- Object Recognition
- Image Processing
- UI/UX Design
- Flutter Development

These are some of the technologies that will be acquired as new skills. The learning strategy is getting support from online courses, reviewing articles, and doing a lot of practice. To achieve these, Udemy and Youtube will be used by the team as sources of courses.

5. References

[1] “The Open Food Facts Team.” Open Food Facts. Accessed October 09, 2021 https://world.openfoodfacts.org/who-we-are.

[2] “TensorFlow Lite” TensorFlow. Accessed October 9, 2021. https://www.tensorflow.org/lite.

[3] “Python” Python. Accessed October 9, 2021. https://www.python.org.

[4] “The World's Most Advanced Open Source Relational Database.” PostgreSQL. Accessed October 9, 2021. https://www.postgresql.org/.

[5] “Vision API Product Search” Google. Accessed October 11, 2021 https://cloud.google.com/vision/product-search/docs

[6] “Custom Search JSON API | Programmable Search Engine.” Google. Google. Accessed October 11, 2021. https://developers.google.com/custom-search/v1/overview

[7] “CollabNet. user information service” PostgreSQL. Accessed Nov 15, 2021. http://help.collab.net/index.jsp?topic=/teamforge163/action/pgsqlrestore.html