Bilkent University
Department of Computer Engineering

Senior Design Project

Analysis and Requirements Report

CONTENTA

GROUP #T2325

Ömer Oktay Gültekin | 21901413 | oktay.gultekin@ug.bilkent.edu.tr
Oğuz Kuyucu | 21902683 | oguz.kuyucu@ug.bilkent.edu.tr
Barış Tan Ünal | 22003617 | tan.unal@ug.bilkent.edu.tr
Mert Ünlü | 22003747 | mert.unlu@ug.bilkent.edu.tr
Alperen Utku Yalçın | 22002187 | utku.yalcin@ug.bilkent.edu.tr

Supervisor: Fazlı Can
Innovation Expert: Tağmaç Topal
Instructors: Mert Bıçakçı & Atakan Erdem
08.12.2023

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.

ABSTRACT

This report delineates the foundational framework, encompassing current system evaluation, proposed system specifications, and essential considerations, for the development of the *Contenta* mobile application. Leveraging state-of-the-art image recognition technology, the application enables users to capture the ingredients section of food products using their mobile phone cameras. Through the implementation of sophisticated algorithms, the application detects potential hazards, such as allergens like aspartame, monosodium glutamate, or E621. For a personalized user experience, individuals can input details about their allergies, height, and weight, empowering the application to precisely identify allergenic ingredients and offer insights into potential health risks. Originally developed to support Turkish and/or English, the application is ready for global outreach by integrating a smooth translation API. In subsequent phases, the application holds the potential to extend its range to include products beyond its initial focus, such as cosmetics or cleaning goods. This exhaustive report outlines the vital technical specifications and features crucial for the effective development and deployment of the *Contenta* mobile application, contributing to an enriched landscape of informed content consumption and promoting health-conscious choices for users globally.

Keywords: packaged food, cosmetic products, human health, ingredient analysis, health-conscious consumption, image-processing.

TABLE OF CONTENTS

ABSTRACT.......... 2
LIST OF FIGURES.......... 4
2.0 CURRENT SYSTEM.......... 6
3.0 PROPOSED SYSTEM.......... 8
3.1 OVERVIEW.......... 8
3.1.1 SCANNING THE INGREDIENTS.......... 8
3.1.2 SCANNING THE NUTRITION FACTS.......... 9
3.1.3 INGREDIENT BLOGS.......... 10
3.2 FUNCTIONAL REQUIREMENTS.......... 10
3.3 NON-FUNCTIONAL REQUIREMENTS.......... 11
3.3.1 USER-FRIENDLINESS.......... 11
3.3.2 MAINTAINABILITY.......... 12
3.3.3 SCALABILITY.......... 12
3.3.4 PERFORMANCE.......... 13
3.3.5 PRIVACY.......... 13
3.4 PSEUDO REQUIREMENTS.......... 14
3.5 SYSTEM MODELS.......... 15
3.5.1 USE CASE MODEL.......... 15
3.5.2 USE CASE DESCRIPTIONS.......... 15
3.5.3 OBJECT AND CLASS MODEL.......... 24
3.5.4 DYNAMIC MODELS.......... 25
3.5.4.1 SCENARIOS & ACTIVITY DIAGRAMS.......... 26
3.5.4.1.1 SIGN-UP & LOGIN.......... 26
3.5.4.1.2 SCAN INGREDIENT / NUTRITION FACTS.......... 27
3.5.4.1.3 READ BLOG.......... 28
3.5.4.1.4 WRITE BLOG.......... 29
3.5.4.2 STATE MACHINE DIAGRAM.......... 30
3.5.5 USER INTERFACE.......... 31
3.5.5.1 NAVIGATIONAL PATHS.......... 31
3.5.5.2 SCREEN MOCK-UPS.......... 32
4.0 OTHER ANALYSIS ELEMENTS.......... 55
4.1. CONSIDERATION OF VARIOUS FACTORS IN ENGINEERING DESIGN.......... 55
4.1.1 CONSTRAINTS.......... 57
4.1.2 STANDARDS.......... 57
4.2 RISKS & ALTERNATIVES.......... 58
4.3 PROJECT PLAN.......... 59
4.4 ENSURING PROPER TEAMWORK.......... 64
4.5 ETHICS & PROFESSIONAL RESPONSIBILITIES.......... 65
4.6 PLANNING FOR NEW KNOWLEDGE & LEARNING STRATEGIES.......... 66
5.0 CONCLUSION.......... 67
6.0 LIST OF ABBREVIATIONS.......... 68
REFERENCES.......... 69

LIST OF FIGURES

Figure 1: Rear face of a chocolate syrup package …………………..……………………......6

Figure 2: Conceptual drawing of a cosmetic product with a focus on the ingredients ………7

Figure 3: Use Case Diagram of Contenta…………………………………….…………...…….16

Figure 4: Class Diagram of Contenta……………………..………….…………...…….22

Figure 5: Sign-up / Login Activity Diagram……..……………..………….…………...…….24

Figure 6: Scan Ingredients / Nutrition Facts Activity Diagram ………….…………...…….25

Figure 7: Read Blog Activity Diagram…….…………………..………….…………...…….26

Figure 8: Write Blog Activity Diagram…….…………………..………….…………...…….27

Figure 9: State Machine Diagram for the main functionality of a premium user…………....28

Figure 10: Navigational Path Diagram between the screens……………………………………..31

Figure 11: Login Screen……………………………...……………………………………..32

Figure 12: Sign-up Screen…………….……………...……………………………………..33

Figure 13: Blog Screen……………….……………...……………………………………..34

Figure 14: Blog Details Screen………….……………...……………………………………..35

Figure 15: Settings Screen…………….……………...……………………………………..36

Figure 16: Expert Application Screen Screen……...……...……………………………………..37

Figure 17: Cancel Expert Rights Screen……………......……………………………………..38

Figure 18: Add New Blog Screen….….………………...……………………………………..39

Figure 19: Add New Blog Details Screen...….……………………...………………………..40

Figure 20: Bookmarks Screen…….…………………...……………………….…………..41

Figure 21: Add Allergy Tab………………….………...………………………………..42

Figure 22: Add Unwanted Ingredients Tab……………...………………………………..43

Figure 23: Add Diet Choices Screen………….……...……………………………………..44

Figure 24: Allergen Details Screen……………...………...……………………………………..45

Figure 25: Unwanted Ingredients Screen………………….…………………………………..46

Figure 26: Diet Screen……….……………………...……………………………………..47

Figure 27: Take Photo Screen……..…………….…………………….…………………..48

Figure 28: Crop Image Screen……………..…………...……………………………………..49

Figure 29: Ingredient Analysis Screen…………………….…………………...…………..50

Figure 30: Nutrition Table Analysis Screen……………….…………….……...…………..51

Figure 31: History Screen……………….………………………………......…………..52

Figure 32: Admin Login Screen………………………...….……………...…………..53

Figure 33: Admin View Applications Screen…………...….……………...…………..54

Figure 34: Admin View Users Screen……………………...………………...…………..55

1.0 INTRODUCTION

Buying packaged products is commonly routine for some households, however, with increasing awareness, there comes a need to know what is within the products customers consume. When buying products, some buyers have to carefully consider which products they should and should not consume depending on the ingredients of these products. The section on the packaging for the ingredients should be read thoroughly since some ingredients that may pose a particular problem to the customer may be within this government-approved and seemingly harmless product.

Understanding the ingredients on the ingredient labels is especially challenging for customers with limited knowledge about the ingredients, their background, and potential risks, even if they pay close attention to the ingredients section. The problem particularly persists with customers with intolerance to certain ingredients who may be at severe risk if they were to consume a particular product [1]. Additionally, people who wish to avoid certain food products such as meat, pork-related food, corn syrup, gluten, or any other product the customer may wish to avoid in their daily consumption. Apart from the more commonly known ingredients, there are also some potentially risky ingredients that users may not have heard of. The user should somehow be well informed when they wish to buy a product with any risky but government-approved content, for which obtaining knowledge of every product's ingredient is particularly difficult to do while shopping in a mall on the get-go.

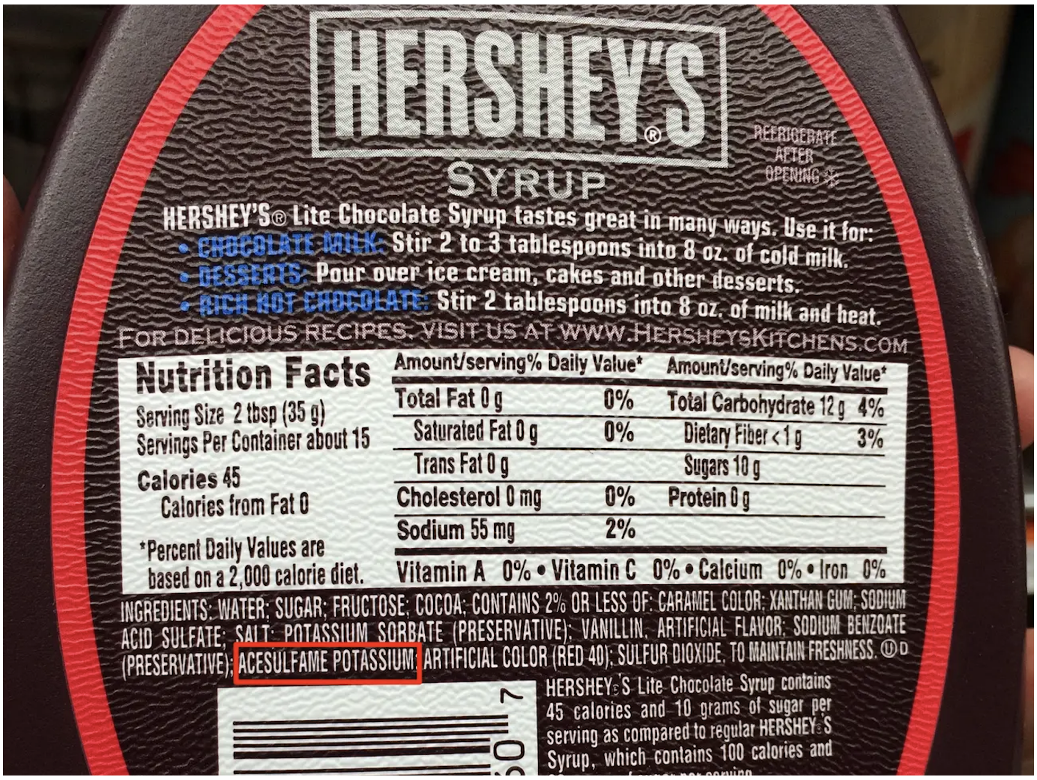

HERSHEY'S®
SYRUP
REFRIGERATE AFTER OPENING
HERSHEY'S® Lite Chocolate Syrup tastes great in many ways. Use it for:
• CHOCOLATE MILK: Stir 2 to 3 tablespoons into 8 oz. of cold milk.
• DESSERTS: Pour over ice cream, cakes and other desserts.
• RICH HOT CHOCOLATE: Stir 2 tablespoons into 8 oz. of milk and heat.
FOR DELICIOUS RECIPES, VISIT US AT WWW.HERSHEYSKITCHENS.COM
Nutrition Facts
Serving Size 2 tbsp (35 g)
Servings Per Container about 15
Calories 45
Calories from Fat 0
*Percent Daily Values are based on a 2,000 calorie diet.
Amount/serving % Daily Value*
Total Fat 0 g 0%
Saturated Fat 0 g 0%
Trans Fat 0 g
Cholesterol 0 mg 0%
Sodium 55 mg 2%
Amount/serving % Daily Value*
Total Carbohydrate 12 g 4%
Dietary Fiber < 1 g 3%
Sugars 10 g
Protein 0 g
Vitamin A 0% • Vitamin C 0% • Calcium 0% • Iron 0%
INGREDIENTS: WATER; SUGAR; FRUCTOSE; COCOA; CONTAINS 2% OR LESS OF: CARAMEL COLOR; XANTHAN GUM; SODIUM ACID SULFATE; SALT; POTASSIUM SORBATE (PRESERVATIVE); VANILLIN, ARTIFICIAL FLAVOR; SODIUM BENZOATE (PRESERVATIVE); ACESULFAME POTASSIUM; ARTIFICIAL COLOR (RED 40); SULFUR DIOXIDE, TO MAINTAIN FRESHNESS. ⓤD
HERSHEY'S Lite Chocolate Syrup contains 45 calories and 10 grams of sugar per serving as compared to regular HERSHEY'S Syrup, which contains 100 calories and

Fig. 1: Rear face of a chocolate syrup package [2].

There are many ingredients that a less informed customer may not have the necessary detailed knowledge about. The customers may buy a product after considering all of the contents they know of, and skipping the uncommonly named ones. They may assume these ingredients are necessary for the product or they may even assume that if the product is government-approved, it must be fine to consume. Assuming they even carefully read the ingredients section, customers may buy products that may threaten their own health. One possible example is “Acesulfame Potassium”, an ingredient commonly used as a sweetener, such as within the Cranberry Juice seen in Figure 2. However the ingredient is listed as a potential cancer cause, as well as listed as a health hazard to pregnant people [2][3]. Hence the ingredients similar to this should be known about and the customers should be more conscious about their decisions while consuming such products with risky ingredients. The app *Contenta* is therefore designed to provide customers with quick and efficient information, aiming to assist them in their daily product consumption by offering transparency about the contents of what they consume.

In this report, we discuss the current methods customers may rely on to check the ingredients of their products in the current system section. The description of our system, along with its functional requirements, quality attributes, pseudo requirements, and the aimed system model- the architecture of our design- are explored in depth within the proposed system section. Later, we delve into the analysis elements that are necessary for ensuring that the development process goes smoothly throughout the timeline. These analysis elements are to consider various factors in engineering design, consider the risks and alternatives, establish an approach for the project plan, find methods of ensuring proper teamwork among the development team, consider ethics and professional responsibilities, and lastly, establish a plan for using all our learned knowledge effectively.

2.0 CURRENT SYSTEM

Our domain is healthcare in packaged products and it is different from most of its competitors who are product-based. One of these competitors is *Ecomercek*, a website specializing in cosmetic product ingredient analysis. Users buy cosmetics and while buying these cosmetic products, they search for the name of the product on this website. Then, the website gives a detailed analysis of the ingredients and the users can learn what the contents are rather efficiently. This implementation is product-based in the sense that each registered product is entered into the system of *Ecomercek* when they are released to the market and any ingredient changes to products are entered into their system manually. Our product, which also aims to spread its scope on cosmetic products in its later stages, is expected to be better compared to the product-based applications for a few different reasons.

First of all, automation plays an important role in the maintainability and scalability of a system. As the number of products within the system grows and the number of products entering the market increases, the manual labor necessary to address updates on product information is expected to increase in folds. If a product changes its ingredients slightly, the product-based system will likely update its data after the item is released to the market and the difference is spotted by some regulation. Hence, there is some period for correcting the false data on the site, while the users are given false data on the contents of the items. Furthermore, the database for keeping information related to the products is expected to change and grow significantly over time, whereas the number of ingredients within the market grows and changes far more slowly, which means an ingredient-based database is expected to have far higher reliability. Another product-based application would be *Yuka* which is based on the functionality of scanning the barcode of a product and giving its contents. This and *Ecomercek* are both product-based and have the same issues of maintenance as well as the incorrectness of data. The benefit of our implementation is that it is fully automatic in finding the ingredients of a food product and uses an ingredient-based database, possibly increasing the correctness, the speed of corrections, and minimizing maintenance.

Furthermore, our application aims to speed up the shopping experience. Especially if the person is shopping within the market, they may not have much time to research each product in detail. The *Ecomercek* is a website where users find their item by searching for the item name. This process is expected to take longer time and is likely to feel more time-consuming and is not much different than searching online, as was told to us in our interviews with potential customers. In this sense, our method of scanning the ingredients section is likely to be less time-consuming and more practical in the user’s eyes. Moreover, having the application in mobile format is likely to attract more attention, rather than having it as a website.

Another application *Ingredio* which uses an ingredient-based database similar to ours is also within the market with over fifty thousand downloads. Even though this application has some functionalities we sought to make better, our application aims to add the key feature of language support, especially in the Turkish market, which from our observations, *Ingredio* lacks greatly. Furthermore, some negative comments on the *Ingredio* application seem to stem from how the application was designed, and how it has difficulty in picking up most of the ingredients. We aim to have more accurate results, a better layout for user experience, informative blogs, as well as the user's personalized allergy, unwanted ingredients, and diet choices.

3.0 PROPOSED SYSTEM

Within this section, the system of packaged food shopping with our app *Contenta* is explained in detail. The functional, nonfunctional, and pseudo requirements are determined for the app. Later, the architecture of our system and the models for our design are explained to an extent.

3.1 OVERVIEW

The main goal of Contenta is to give users fast feedback on the ingredients within their packaged products on the go. Within the app, consumers may scan the ingredients of packaged goods to obtain detailed information, especially any health concerns, about the ingredients. In addition to warning or informing users about their concerns, risks, and backgrounds; the application seeks to read, detect, and list the ingredients listed on the packages in an easily interactable format. If the user wishes to, it will be an assistant tool for users who are following any special diets where they avoid certain kinds of ingredients. The goal of the healthcare app is to deliver quick, precise results in order to help raise customer knowledge of healthcare issues, hastening the buying process for those who want to know what's within the products they use. Within the app, users can create their own profiles. Within this profile, users provide details about their allergies, potential health problems, whether there is a pregnant person in the home, whether they would prefer not to eat a particular component, etc. In addition, customers have the option to follow any diet programs available in the app. Based on the user's profile choices, the app aims to advise users on whether or not to consume a particular product, giving healthcare tips where the users have a transparent view of what they consume.

3.1.1 SCANNING THE INGREDIENTS

Once users log in to the application, they are able to click on a camera button at the bottom tab of the screen. When this button is clicked, two options show up on the screen, first for scanning the ingredients of a food product, and the second option is for scanning the nutrition facts of the product. These are separate since they are located in different parts of the packaging.

Upon clicking the scan ingredients button, the user is able to see the back camera’s view. The user then picks up the item they want to scan and takes a picture of the contents. Afterward, they crop the taken photo and adjust the cropped section so that the photo has the ingredients section. The cropping process is necessary since it increases the accuracy of the analysis results. Next, upon pressing the analyze button, the image that contains text gets processed within the application. After the analysis is complete, the contents that were extracted from the image are shown on the screen. Every piece of content is ranked in accordance with its level of safety and given a color designation (red for maybe dangerous, yellow for neutral, and green for safe to eat). Users can click on any of the ingredients listed to access additional details about the ingredient, such as its description, information about its composition, hazards, and health benefits, as well as related blogs and articles. Moreso, the app gives recommendations on whether the consumption of the product may be unsafe to the consumer (user), based on their earlier set preferences; their pre-established diet plan, unwanted ingredients, or allergies.

Fig. 2: Conceptual drawing of a cosmetic product with a focus on the ingredients section [4].

The app may misread, misinterpret, or read only a limited number of ingredients that are written on the packaging, hence the users are informed that some results may be faulty and they should check if any ingredients were skipped in the analysis process. Even though we expect the app to perform well (given our test results on text detection on packaging at the current stage), since we are dealing with the healthcare industry, it is of great importance that users are notified of any possible flaws.

3.1.2 SCANNING THE NUTRITION FACTS

If the user chooses to click on the scan nutrition facts button after clicking the camera button on the bottom tab of our app, they are able to take a photo of the nutrition facts part of the packaging to evaluate the nutritional value of a food product. The software then assesses the food items' nutritional values and displays these values on the screen. Furthermore, it calculates the nutri-score of the product with the available information. This feature is particularly useful in formatting the table in a standard format for all packages, enabling users to read the data as an easy-to-read standardized table with an easily readable font size, which is especially useful for packaging with small fonts or badly colored and formatted layouts.

3.1.3 INGREDIENT BLOGS

Healthy consumers are becoming more conscious of the contents of packaged goods, which is contributing to an increase in the number of blogs about food ingredients and their hazards. Our app also provides users with access to these helpful and educational blogs, as there is a growing need to learn from professionals about the specifics of food consumption by customers. Within the app, there exist nutrition experts who have influencing knowledge and expertise in the field of ingredients of products. These experts write blogs to the app and users can view these blogs on the blogs tab. All blogs about a certain ingredient are displayed within the ingredient information, and when the user clicks a blog, they can read further and gain further understanding. The experts within the app provide thoughts, recommendations, and details about goods and ingredients while giving customers guidance on which ingredients may be unrecommended or healthy to consume. While users are evaluating whether to buy a certain product, they can quickly access the blogs of these authors to obtain further information and viewpoints on their shopping experience.

3.2 FUNCTIONAL REQUIREMENTS

- *Sign Up:* A user can sign up with an email and password.
- *Login:* A user can log in from different devices by using email and password.
- Forgot Password: A user can get an email to change his/her password whenever he/she forgets it.
- *Change Password:* When logged in, a user can change his/her password by using the old password.
- *Add Allergy:* A user can add an allergy from the allergies list. The application will notify the user if the user scans a food that contains an ingredient that may trigger the allergy.
- *Select Language:* A user can select a language to scan. A user can change this selection at any time.
- *Scan Ingredients:* A user can scan food's ingredients and then select the desired language area. The application will detect and inform the user about the current status of food additives in the US, UK, and EU. The application will detect if there are any ingredients that may

trigger the user's allergies. The application will warn users if there are any potentially harmful substances in the product.

- *Become Premium Member:* Users can become a premium member by subscribing. Subscription fees will be determined in the later stages.
- *Terminate Subscription:* A premium user can stop his/her subscription.
- *Scan Nutrition Facts Table:* A premium member can scan the food nutrients part in the label. The application will detect the Nutri-Score (A, B, C, D, or E) of the food which will be calculated according to EU health legislation.
- *Add Unwanted Ingredient:* A user can select an additive from the list and get a warning when they scan food ingredients and the application detects the ingredient.
- *Provide Feedback:* Users can provide feedback to the developer team.
- *Delete Account:* A user can delete his/her account. His/Her personal information will be deleted in this case.
- *Read Blog:* Users can read blogs about specific ingredient(s) or nutrient facts which are written by experts.
- *Write Blog:* Experts who are confirmed by admin can write blog articles about some ingredients or nutrition facts.
- *Promote User*: An admin can promote a user to an expert.
- *Demote User*: An admin can demote an expert to a user.
- *Bookmark:* A user can bookmark a blog article.
- *Remove Bookmark:* A user can remove the bookmark of a blog article.

3.3 NON-FUNCTIONAL REQUIREMENTS

In this part, we will list the application’s non-functional requirements.

3.3.1 USER-FRIENDLINESS

The app’s users are expected to be of all ages, so the UI should be intuitive and easy to understand even in its first-ever use. Hence, it should be similar to other healthcare apps on the internet and it should have a friendly, inviting look to it- especially since it is a healthcare app. The pages should look clean and the fonts should be big enough for the users to read the ingredients. Since the fonts on packaged food products are smaller in size, we wish for our users to have a more convenient experience and prefer to scan the ingredients to a format that they can read through easily. Also, a sweet spot between easing the mechanism of processing the image and making it easy for the customer to scan it should be found. Making the user select the part that includes the ingredients after taking a picture of the ingredients would remarkably fasten the processing stage but it should not be hard for the user to do so. The user should not wait more than 5 seconds for any operation without providing any input. Furthermore, the splash onboarding screen will give a tour of the app when it is opened the first time, easing the process of understanding the app. Users will be able to swap one screen to another with at most 5 clicks. The user will clip the images' related parts by clipping, which is a similar approach to common applications like WhatsApp or Google Translate.

3.3.2 MAINTAINABILITY

The app should have as little maintenance as possible and the end result of our production should be a self-sufficient app that can continue its functionality with as little reliance on updates as possible. The app is aimed to be non-product-based, which greatly increases its maintainability in the long term. The database will avoid holding information related to the products themselves but will be keeping user-related information, ingredient information, and blog information. The process of identifying the ingredients within the product is based solely on what is written on the package itself. Therefore if the ingredients of a product change, then the ingredients label is also changed. This means our app has all the data necessary to function as needed even when a product changes completely, without having to rely on a database update whenever a change in a product's ingredients is detected. This greatly increases the sustainability of the app as well as its accuracy in providing correct healthcare advice. The information about ingredients will be changed according to new findings, which are maintained with expert blogs and feedback. User information is maintained by themselves for the most part and we keep their data only as long as the user wishes to continue using our product. Furthermore, blogs are maintained by the experts and they can edit, remove, or add blogs as they wish. If any blog is found to contain false information, experts are informed by the customers and they should revise their blogs accordingly. Any necessary action may be taken by the admins to remove & edit the blogs as well as take action against experts; warn them, or take the rights of contribution to the app - experts are fully responsible for their own misinformation and are penalized. The servers should not stay down for more than 8 hours and the users should be notified at least 24 hours before a planned maintenance. Besides, maintaining the application will require 1 software engineer because "Effective Dart" standards will be followed.

3.3.3 SCALABILITY

In our implementation we have both short-term and long-term goals. We are developing the application with the intention of adding our long-term goals such as transitioning to cosmetic products after our application works on food-related products if the application gets good attention. Since we have a road map in mind, the foundations and the database of the app will be compatible with other types of product categories and ingredient categories. We plan to support up to 10,000 users at the beginning and improve it as the number of users increases.

3.3.4 PERFORMANCE

The processing speed of the images is crucial for the user experience as it will probably be the bottleneck of the whole waiting time in the application overall. We plan to make the user's CPU run the image processing algorithms. If we were to give an approximate number from the initial tests we performed, the output should be created in 5 seconds at maximum on a Snapdragon 855 or higher. Since the users' mobile phones each have different processing speeds, although the processing time may vary, it should preferably be between 3 to 7 seconds (given 2 seconds of uncertainty to our test value of 5 seconds) for more than 90% of our users.

3.3.5 PRIVACY

The application will collect personal data that may be linked to real people, storing their preferences and healthcare information as long as they wish to. Since our application is made and maintained in Turkey, we have to store data in Turkey and delete data if a user deletes their account. The terms and conditions, as well as the privacy policy of *Contenta,* are both confirmed by the users when they wish to use the application, and the user is notified accordingly that they accept these documents fully. Regarding the regulations and laws in Turkey, data is protected and kept according to the 6698th law of the legislation law "Mevzuat", and the health-related data is kept for the reasons of health services, and any information is terminated as soon as the user wishes to deletes their account, or when there is no need for keeping any necessary user data [5]. The user has full control of their data, and we aim to make the process as transparent as possible by only storing the data the user wishes us to store, and the data will only be processed for the purposes of the app functionalities the user wishes to use, such as the calories of food they took will only be stored to calculate the calorie intake with respect to their diet and the processed data will be given to the user directly. Unless the user wishes us to store any data by providing us with the information themselves by interacting with the app's UI, no data will be stored and all unavailable data will be deleted permanently.

3.4 PSEUDO REQUIREMENTS

The pseudo requirements include information related to preferences, technologies, or anticipated future developments that may influence our project. Even though these requirements are not essential for the core functionality, it is necessary to have a guideline.

- The application will be available for both Android and IOS in the long term.
- Application will support English (certain) and Turkish (probable) at the first stage.
- Flutter will be used for frontend development.
- Firebase will be used for backend development and database management.
- Dart will be utilized as the programming language.
- For collaboration among the developers, GitHub will be used for version control.
- Jira is being used for project planning.

3.5 SYSTEM MODELS

In this section the system models including the use case model, object and class model, dynamic models, and user interfaces will be illustrated.

3.5.1 USE CASE MODEL

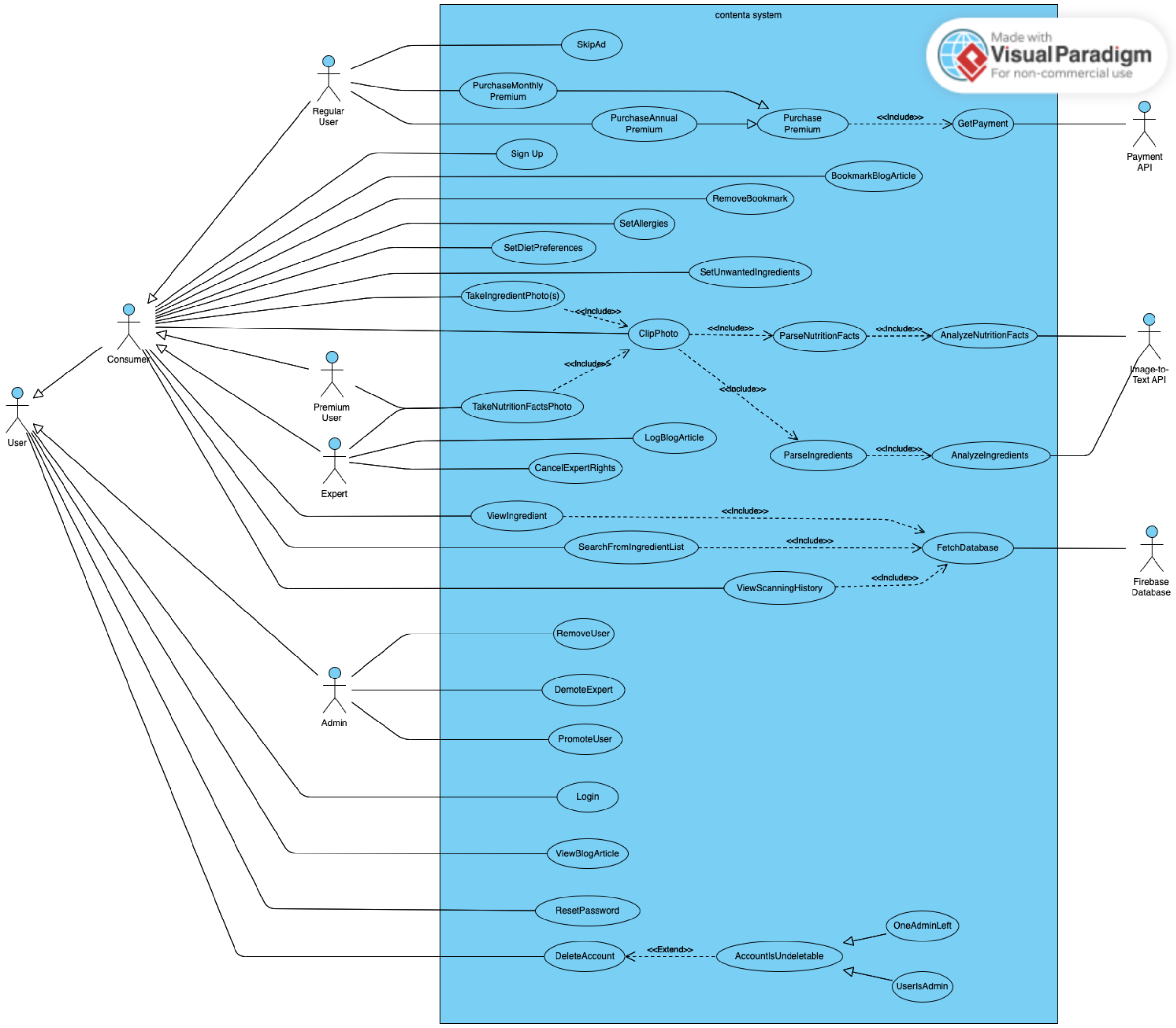


Fig. 3: Use Case Diagram.

3.5.2 USE CASE DESCRIPTIONS

Below, each use case is explained in detail with their names, participating actors, flow of events, entry and exit conditions. Trivial use case descriptions such as SkipAd, RemoveUser, RemoveBookmark etc. are omitted for simplicity.

Use Case Name: TakeIngredientPhoto(s)

Participating Actors: Consumer

Flow of Events:

1. Consumer taps to the camera icon in the middle of the dock.
2. Consumer selects the ingredient scan option.
3. Consumer takes up to 3 photos.
   3.1. If any one of the photos is out of focus,
      3.1.T. The consumer is asked to retake the photo.
      3.1.F. Else, the photo is added to the set of photos.

Entry Conditions:

- If the user is a regular user, the weekly photo limit should not be exceeded.

Exit Conditions:

- All of the photos are in-focus which will be determined by an image-blur detection algorithm.
- There are up to 3 photos on the screen that the user can navigate between.

Use Case Name: TakeNutrientFactsPhotos

Participating Actors: Premium User, Expert

Flow of Events:

1. Premium user/expert taps the camera icon in the middle of the dock.
2. Premium user/expert selects the nutrient facts option.
3. Premium user/expert takes up to 3 photos.
   3.1. If any one of the photos is out of focus,
      3.1.T. The premium user/expert is asked to retake the photo.
      3.1.F. Else, the photo is added to the set of photos.

Entry Conditions:

- Premium user/expert should be logged in.

Exit Conditions:

- All of the photos are in-focus which will be determined by an image-blur detection algorithm.
- There are up to 3 photos on the screen that the user can navigate between.

Use Case Name: ClipPhoto

Participating Actors: Consumer

Flow of Events:

1. Consumer selects the clipping tool while viewing the photo.
2. A rectangular grid is displayed for visualization of the clipping operation.
3. Consumer drags the corners of the grid in the photo.
4. Consumer taps the “analyze” button.

Entry Conditions:

- A set of photos (or a single photo) are taken and present on the screen.

Exit Conditions:

- N/A

Use Case Name: ViewIngredient

Participating Actors: Consumer

Flow of Events:

1. Consumer searches for the ingredient.
2. Consumer clicks on the possible results from the search bar.
3. Consumer navigates to the ingredients page.

Entry Conditions:

- The ingredient data is present in the database.

Exit Conditions:

- Selected ingredient's data is displayed.

Use Case Name: PurchasePremium

Participating Actors: Regular User

Flow of Events:

1. User navigates to the payment interface.
2. Selects monthly or annual membership.
3. Triggers the Payment API to get the payment.

Entry Conditions:

- User has an internet connection.

Exit Conditions:

- Payment API successfully acquired the payment.

Use Case Name: GetPayment

Participating Actors: Payment API

Flow of Events:

1. The Payment API is triggered by either subscription expiration or by a regular user manually with PurchasePremium use case.
2. Payment API checks whether there are enough resources in the credit card for the subscription. If there are enough resources,

   2.T. The credit card gets charged.

   2.F. The subscription purchase attempt fails.

Entry Conditions:

- The user's subscription has less than 3 days before expiration. *OR*
- User manually triggers the payment process. *AND*

- The user’s credit card has enough money for the respective subscription.

Exit Conditions:

- Subscription starts. *OR*
- User is displayed an error message regarding the cause of the failed attempt.

Use Case Name: SetUnwantedIngredients

Participating Actors: Consumer

Flow of Events:

1) Consumers navigate to their profile.
2) Consumers update the unwanted ingredients list.

Entry Conditions:

- Consumers have an internet connection.

Exit Conditions:

- Success return message has arrived.

Use Case Name: SetAllergies

Participating Actors: Consumer

Flow of Events:

1. Consumer goes to the allergies page.
2. Consumer selects an allergy form list.
3. Consumer clicks the button “add”.

Entry Conditions:

- Consumer should be logged in.

Exit Conditions:

- Allergy is successfully added.

Use Case Name: SetDietPreferences

Participating Actors: Consumer

Flow of Events:

1. Consumer clicks diets page.
2. Consumer selects a diet.
3. Consumer clicks set button.

Entry Conditions:

- Consumer should be logged in

Exit Conditions:

- Diet is set successfully.

Use Case Name: LogBlogArticle

Participating Actors: Expert

Flow of Events:

1. Expert chooses the related ingredients/nutrients.
2. Expert writes an article.
3. Expert submits the article.

Entry Conditions:

- Expert should be logged in.

Exit Conditions:

- Article is submitted successfully.

Use Case Name: ParseIngredients

Participating Actors: ImageToTextAPI

Flow of Events:

1. API receives the image.
2. API detects texts by computer vision algorithms.
3. API returns text.

Entry Conditions:

- User should have access to the internet.
- API should be accessible

Exit Conditions:

- Text is returned successfully.

Use Case Name: ParseNutritionFacts

Participating Actors: ImageToTextAPI

Flow of Events:

1. API receives the image.
2. API detects texts by computer vision algorithms.
3. API returns text.

Entry Conditions:

- User should have access to the internet.
- API should be accessible

Exit Conditions:

- Text is returned successfully.

Use Case Name: AnalyzeNutritionFacts

Participating Actors: ImageToTextAPI

Flow of Events:

1. API receives the text
2. API matches nutrients in text with a nutrient
3. API calculates nutri score according to a formula.
4. API returns nutri-score

Entry Conditions:

- User should have access to the internet.
- API should be accessible

Exit Conditions:

- Nutrition facts are returned successfully

Use Case Name: AnalyzeIngredients

Participating Actors: ImageToTextAPI

Flow of Events:

1. API receives the text
2. API matches ingredients in text with ingredients
3. API detects ingredients' status according to FDA's UK's or EU's regulations.
4. API returns ingredients status.

Entry Conditions:

- User should have access to the internet.
- API should be accessible

Exit Conditions:

- Ingredients' status are returned successfully.

Use Case Name: SearchFromIngredientList

Participating Actors: Consumer

Flow of Events:

1. Consumer clicks the ingredient search bar.
2. Consumer writes their search words.
3. Database fetches the list of ingredients related to the word (gets the ingredients with word substring etc.)
4. Consumer accesses the ingredients on the front end.

Entry Conditions:

- User should have access to the internet.
- Database should be accessible

Exit Conditions:

- Search for ingredients is returned successfully.

Use Case Name: ViewScanningHistory

Participating Actors: Consumer

Flow of Events:

1. Consumer opens the history page
2. Database fetches the list of ingredients of the searches of the consumer
3. Consumer views their past scans and the ingredients within these scans.

Entry Conditions:

- User should have access to the internet.
- Database should be accessible

Exit Conditions:

- History for consumer is returned successfully.

Use Case Name: FetchDatabase

Participating Actors: Firebase Database

Flow of Events:

1. A query is entered to search data.
2. The query's result is fetched from the Firebase Database.

Entry Conditions:

- User should have access to the internet.
- Database should be accessible

Exit Conditions:

- Database fetched the query successfully.

Use Case Name: Log In

Participating Actors: User

Flow of Events:

1. User opens the app
2. User enters their credentials to log in
3. User is logged in

Entry Conditions:

- User should have access to the internet.
- User must have an account already.

Exit Conditions:

- User logged in successfully.

Use Case Name: Sign Up

Participating Actors: Consumer

Flow of Events:

1. Consumer opens the app
2. Consumer enters a username, and a password
3. Consumer verifies their password
4. Consumer is signed up, and they automatically log in

Entry Conditions:

- User should have access to the internet.

Exit Conditions:

- Consumer signed up successfully.

Use Case Name: DeleteAccount

Participating Actors: User

Flow of Events:

1. User opens the settings page
2. User clicks on the delete account button
3. User clicks confirm to verify
4. If the account is undeletable (user is the last admin and there is only one admin left), this action is canceled

Entry Conditions:

- User should have access to the internet.
- User should be logged in

Exit Conditions:

- User deleted successfully.

Use Case Name: ResetPassword

Participating Actors: User

Flow of Events:

1. User opens the settings page
2. User clicks on the reset password
3. User enters the current password and the new password

Entry Conditions:

- User should have access to the internet.
- User should be logged in

Exit Conditions:

- User reset their password successfully.

Use Case Name: ViewBlogArticle

Participating Actors: User

Flow of Events:

3. User opens the list of blogs page
4. User chooses a blog
5. User views the blog page in detail

Entry Conditions:

- User should have access to the internet.
- User should be logged in
- Expert should have logged the blog article

Exit Conditions:

- Blog article is viewed successfully.

3.5.3 OBJECT AND CLASS MODEL

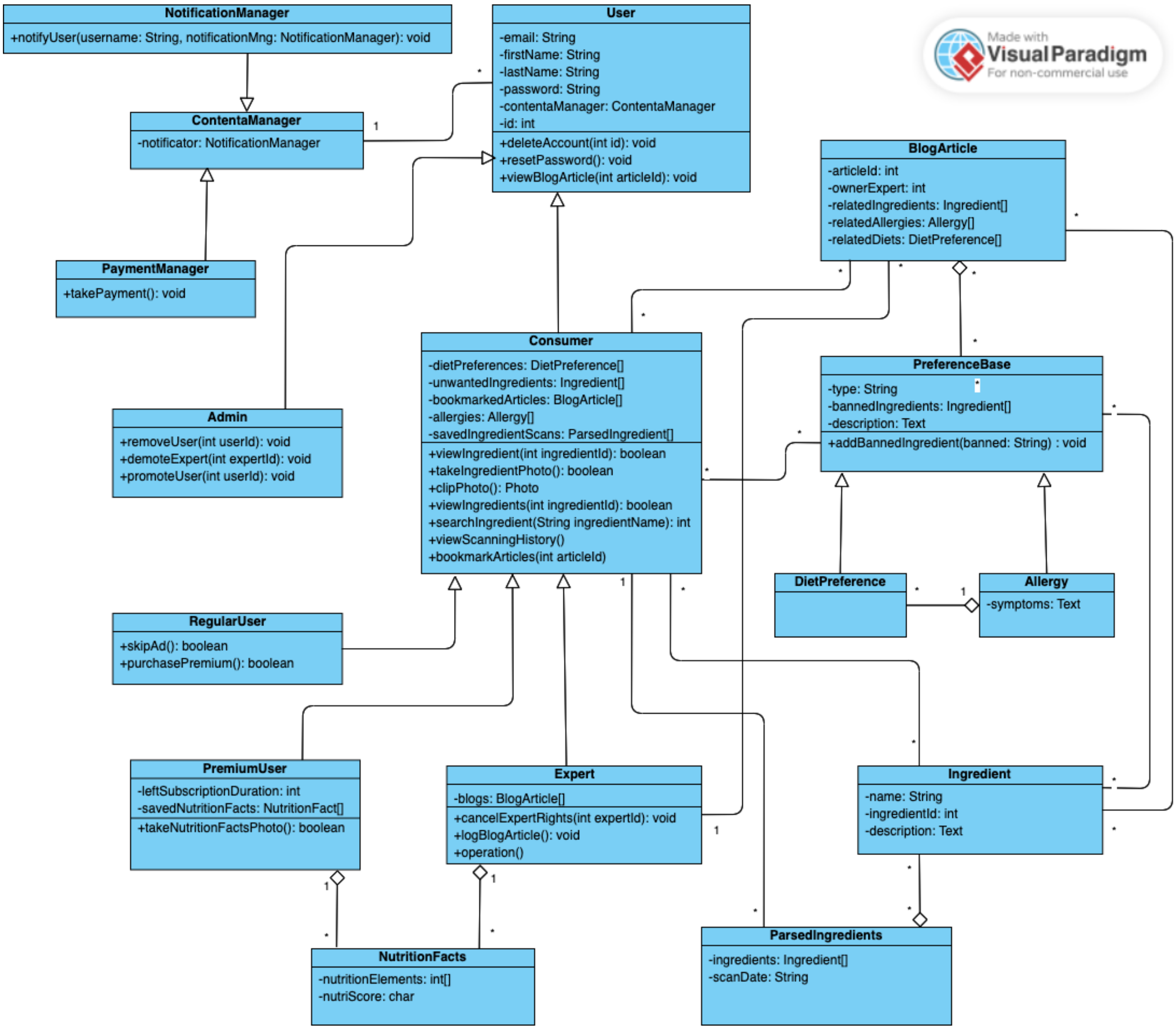

NotificationManager
+notifyUser(username: String, notificationMng: NotificationManager): void
User
-email: String
-firstName: String
-lastName: String
-password: String
-contentaManager: ContentaManager
-id: int
+deleteAccount(int id): void
+resetPassword(): void
+viewBlogArticle(int articleId): void
Made with Visual Paradigm For non-commercial use
ContentaManager
-notificator: NotificationManager
1
*
BlogArticle
-articleId: int
-ownerExpert: int
-relatedIngredients: Ingredient[]
-relatedAllergies: Allergy[]
-relatedDiets: DietPreference[]
PaymentManager
+takePayment(): void
Consumer
-dietPreferences: DietPreference[]
-unwantedIngredients: Ingredient[]
-bookmarkedArticles: BlogArticle[]
-allergies: Allergy[]
-savedIngredientScans: ParsedIngredient[]
+viewIngredient(int ingredientId): boolean
+takeIngredientPhoto(): boolean
+clipPhoto(): Photo
+viewIngredients(int ingredientId): boolean
+searchIngredient(String ingredientName): int
+viewScanningHistory()
+bookmarkArticles(int articleId)
PreferenceBase
-type: String
-bannedIngredients: Ingredient[]
-description: Text
+addBannedIngredient(banned: String) : void
Admin
+removeUser(int userId): void
+demoteExpert(int expertId): void
+promoteUser(int userId): void
DietPreference
Allergy
-symptoms: Text
RegularUser
+skipAd(): boolean
+purchasePremium(): boolean
PremiumUser
-leftSubscriptionDuration: int
-savedNutritionFacts: NutritionFact[]
+takeNutritionFactsPhoto(): boolean
Expert
-blogs: BlogArticle[]
+cancelExpertRights(int expertId): void
+logBlogArticle(): void
+operation()
Ingredient
-name: String
-ingredientId: int
-description: Text
NutritionFacts
-nutritionElements: int[]
-nutriScore: char
ParsedIngredients
-ingredients: Ingredient[]
-scanDate: String
1
*

Fig. 4: Class Diagram.

- **User**: This class represents the base user class which is the parent class of other user types. It contains the primitive user attributes and methods.

- **Consumer**: This class is representing the user types that are actually using the functionalities of the application.

- **Regular User**: It is the unpaid user type which can scan ingredients and read/save blogs articles.

- **Premium User**: This class represents the paid user type which can also scan nutrition facts tables in addition to the other regular user functionalities.

- **Admin**: Admin class, contains methods related to users.

- **Expert**: This class represents the experts that can post blog articles in our application. It is also a Consumer sub-class. It contains expert related data such as written articles.

- **Blog Article**: This class is representing the articles that are submitted by experts. Each article has an owner expert and other content related data such as related ingredients.

- **Preference Base**: This class is a parent class for Allergy and Diet Preference classes. It has a banned ingredients list and modifier method.

- **Allergy**: This class represents the Allergies in our application. Each Allergy has a symptom description as well as suggested Diet Preferences.

- **Diet Preference**: This class is also a child class of Preference Base. It represents the different diets in our app.

- **Nutrition Facts**: Nutrition Tables are represented with this class. It contains data related to the content of the table.

- **Ingredient**: Ingredient is the cell of our application. It contains the ingredient related data. It has aggregation relations to many other classes.

- **Parsed Ingredients**: This class represents the object that is constructed after the parsing process of a taken photo. It contains the parsed data.

- **Contenta Manager**: This class acts in Singleton pattern and contains other managers inside it.

- **Payment Manager**: This class acts as a middleware between users and payment API.

3.5.4 DYNAMIC MODELS

In this section, dynamic models will be illustrated. Dynamic models include activity diagrams and a state machine diagram of the main functionality of the application.

3.5.4.1 SCENARIOS & ACTIVITY DIAGRAMS

In this subsection, the scenarios are discussed with their respective activity diagrams. There are four major scenarios: Sign-up / login, scan ingredient / nutrition facts, read blog, and write blog.

3.5.4.1.1 SIGN-UP & LOGIN

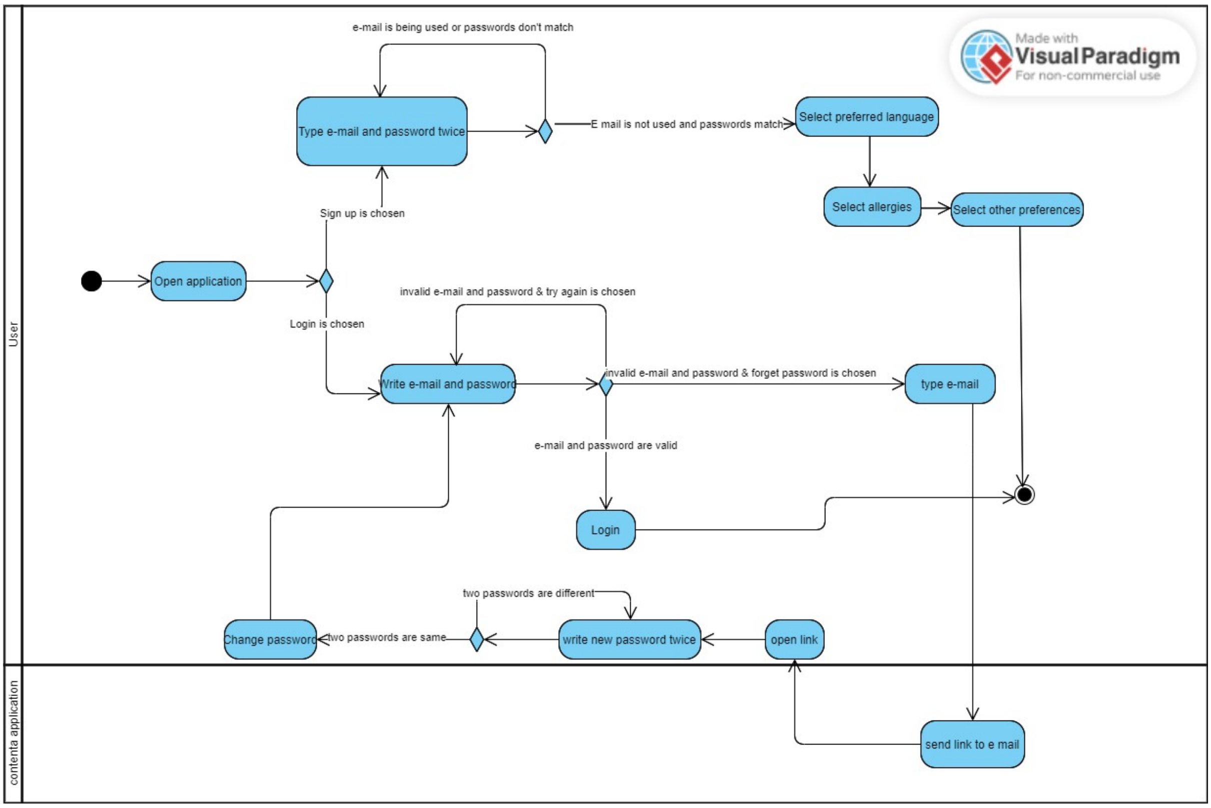

User
contenta application
e-mail is being used or passwords don't match
Type e-mail and password twice
E mail is not used and passwords match
Select preferred language
Select allergies
Select other preferences
Sign up is chosen
Open application
Login is chosen
invalid e-mail and password & try again is chosen
Write e-mail and password
invalid e-mail and password & forget password is chosen
type e-mail
e-mail and password are valid
Login
two passwords are different
Change password
two passwords are same
write new password twice
open link
send link to e mail

Fig. 5: Sign-up & Login Activity Diagram.

Users can sign-up or login. If they forget their password, they can change it.

3.5.4.1.2 SCAN INGREDIENT / NUTRITION FACTS

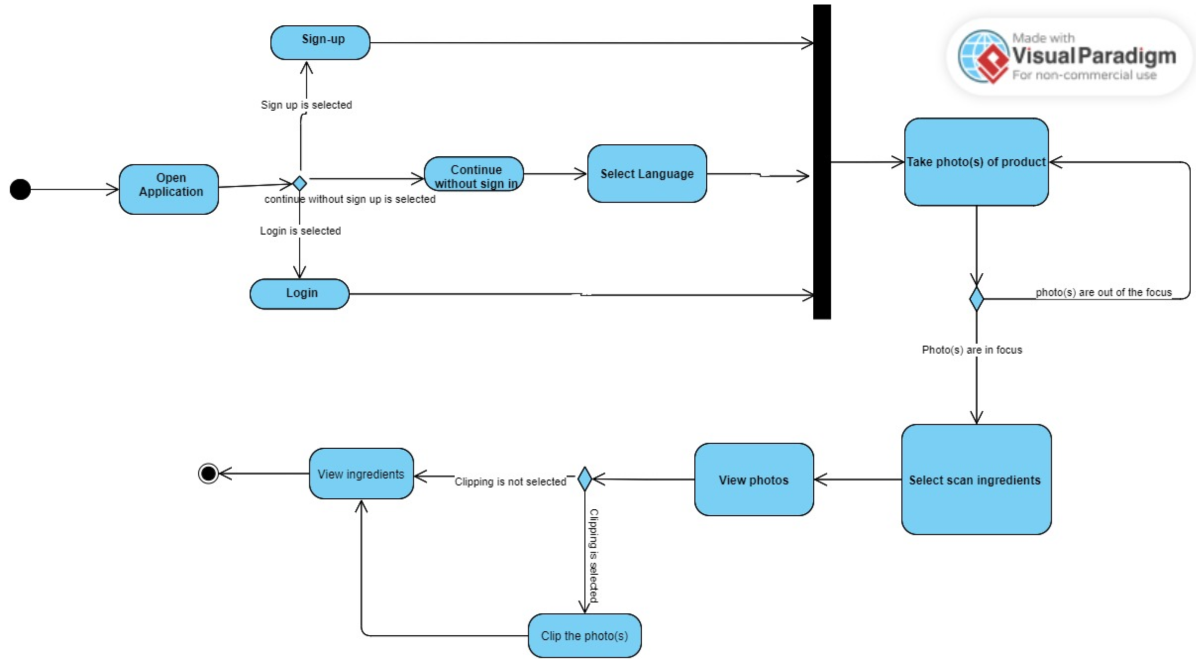


Fig. 6: Scan Ingredients / Nutrition Facts Activity Diagram.

Users can take a photo(s) of a product (depending on a product's shape, 1 photo may be enough. If a product is not rectangular, 2 or 3 photos may be enough.) and then select to scan ingredients. And then they can view photos and clip the photos if needed. Then they can see ingredients inside the product and their status according to FDA, EU and UK.

For scan nutrient facts, only "Select scan ingredients" and "View ingredients" will be different. They will be "Select scan nutrient facts" and "View nutrient facts" respectively.

3.5.4.1.3 READ BLOG

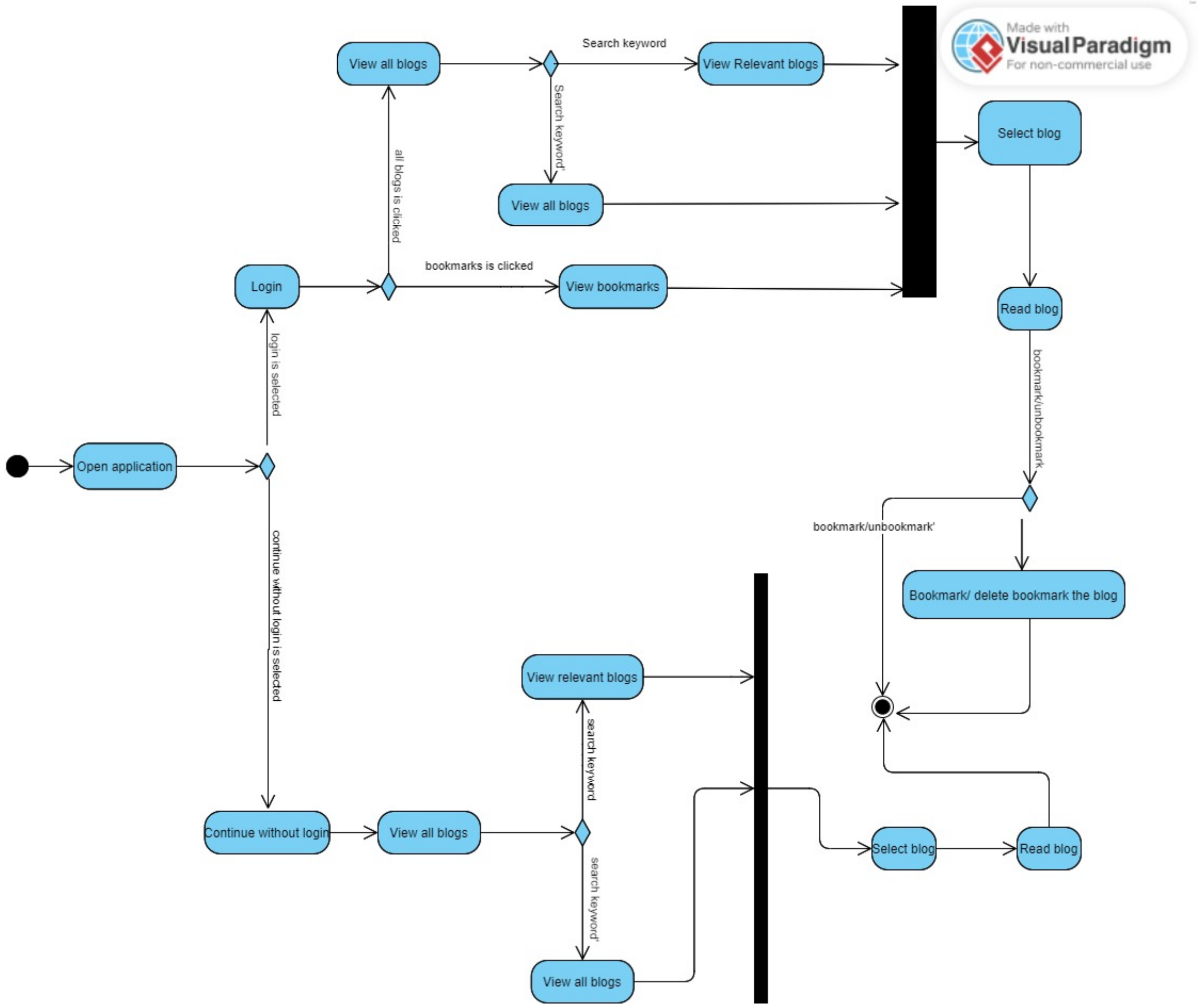


Fig. 7: Read Blog Activity Diagram.

Users can read blogs about ingredients / nutrients or diets. They can add bookmarks or delete bookmarks if they are logged in.

3.5.4.1.4 WRITE BLOG

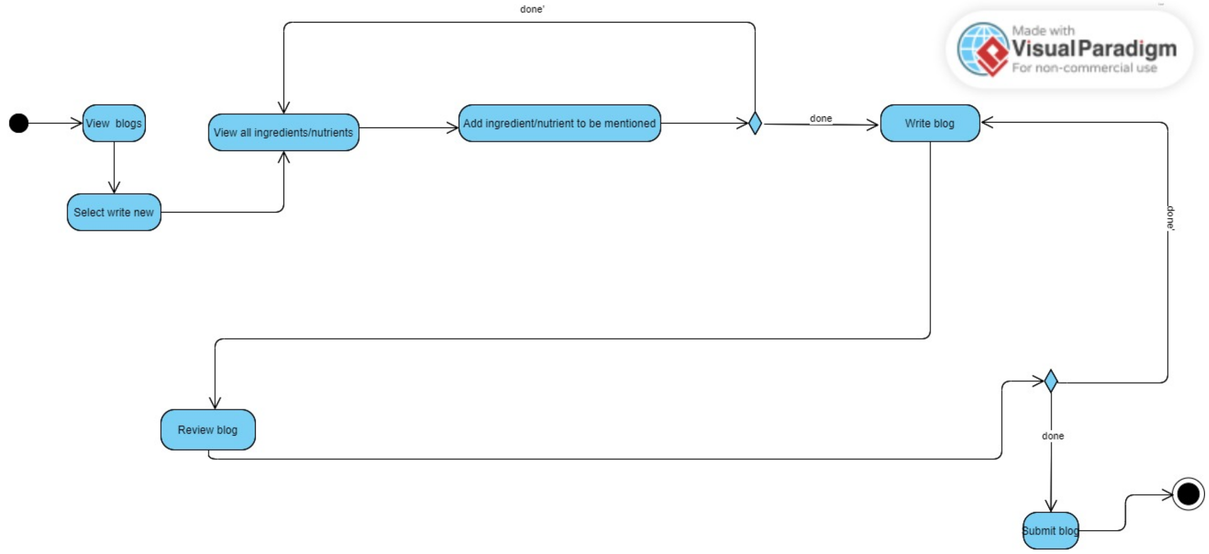

done'
Made with Visual Paradigm For non-commercial use
View blogs
Select write new
View all ingredients/nutrients
Add ingredient/nutrient to be mentioned
done
Write blog
done'
Review blog
done
Submit blog

Fig. 8: Write Blog Activity Diagram.

Experts can write blogs about nutrients/ingredients. They can mention ingredients in their blog articles.

3.5.4.2 STATE MACHINE DIAGRAM

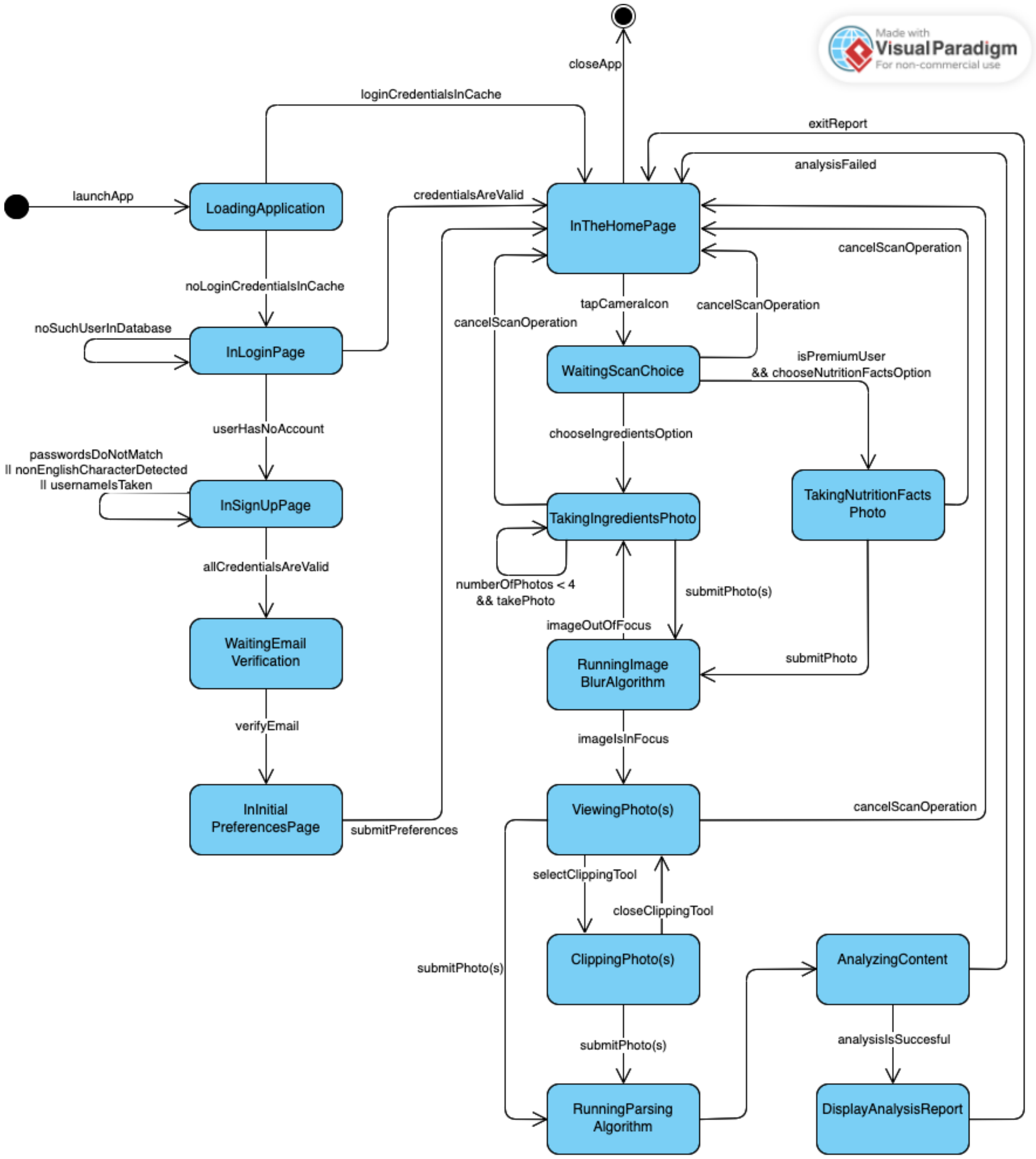


Fig. 9: State Machine Diagram for the main functionality of a premium user.

The state machine diagram for the *Contenta* depicts a user journey through the sign-up / login process and the app's core functionality which is scanning photos. Beginning with app launch, the user can either be directed to the home page if previously logged in, or navigate to the login or sign-up page. From the home page, the user can initiate a scan for nutrition facts or ingredients, leading to photo capture and image processing. Upon successful analysis, the user receives a detailed report. Additionally, the ability to view and manipulate photos, including clipping specific areas, is included within the diagram. This visual representation effectively outlines a premium user's flow within the *Contenta* mobile application.

3.5.5 USER INTERFACE

In this section, a rough sketch of the user interface will be introduced with the navigational paths followed by the screen mock-ups.

3.5.5.1 NAVIGATIONAL PATHS

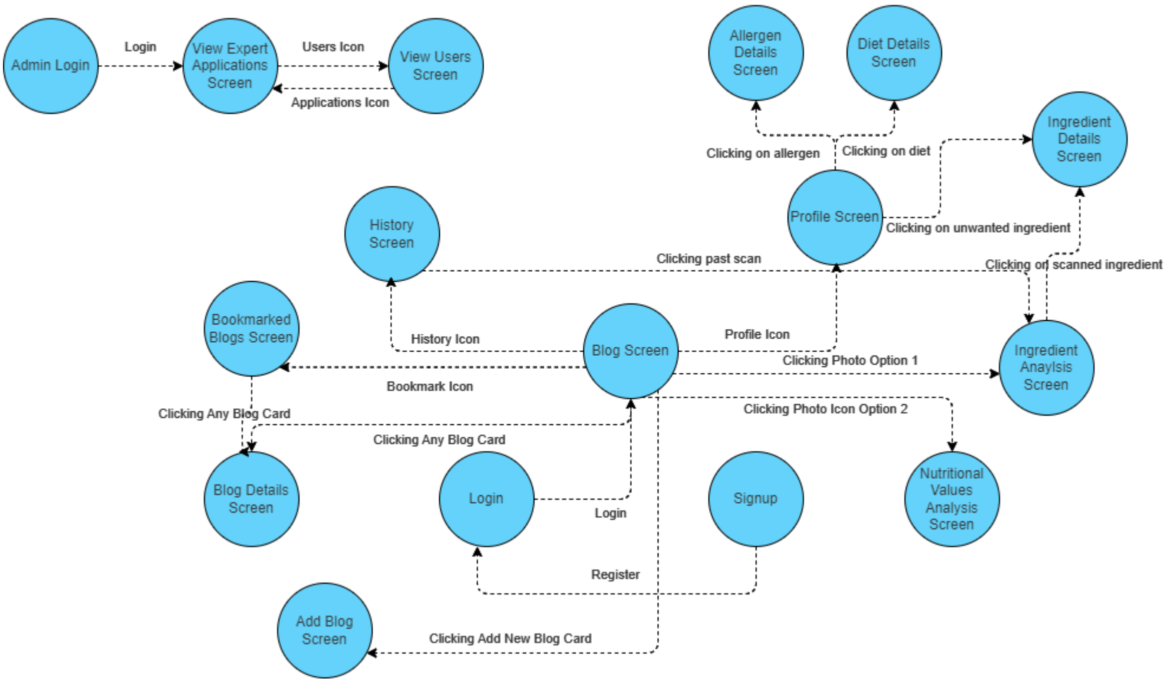
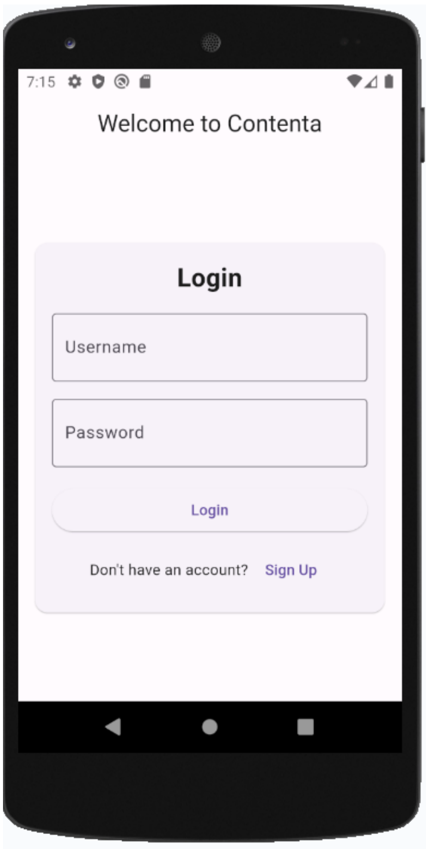


Fig. 10: Navigational Path Diagram between the screens.

3.5.5.2 SCREEN MOCK-UPS

In this section, the mock-up screens will be depicted.

3.5.5.2.1 LOGIN SCREEN



Welcome to Contenta

Don't have an account? Sign Up

Fig. 11: Login Screen.

All consumers (for consumer definition see use case diagram) login to the system using this page.

3.5.5.2.2 SIGN-UP SCREEN

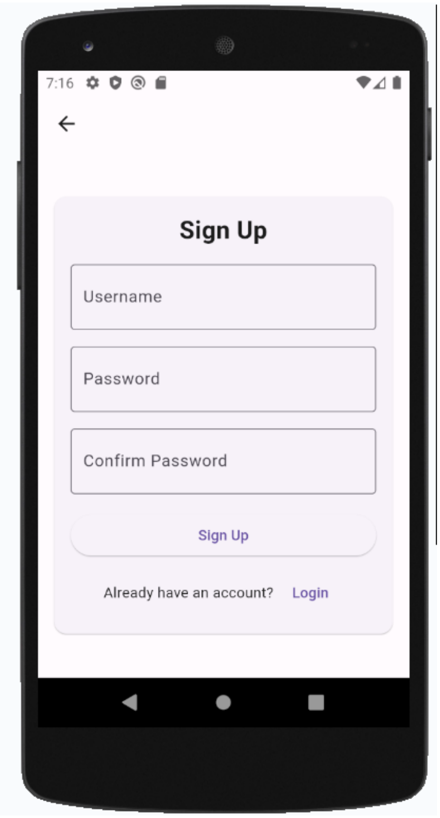


Fig. 12: Sign Up Screen.

Consumers sign up to the app by giving their username (mail) and password information.

3.5.5.2.3 BLOG SCREEN

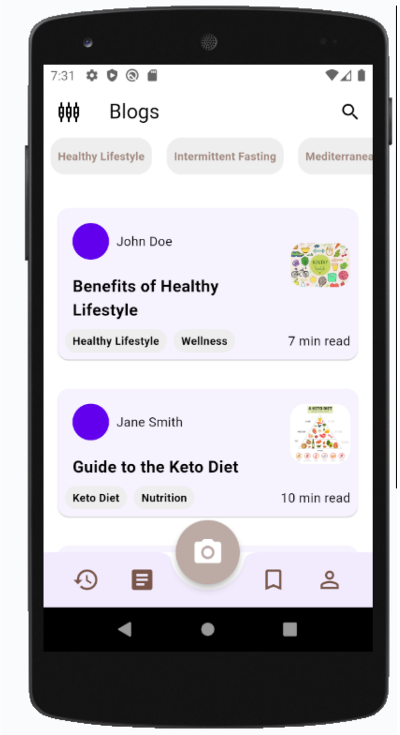
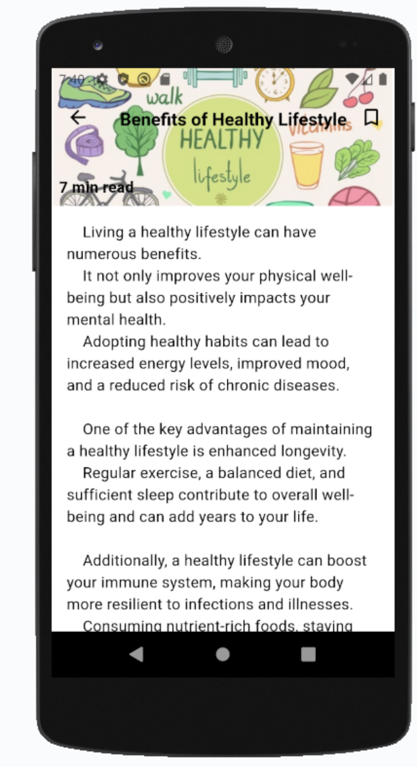


Fig. 13: Blog Screen.

Blog Screen is where the consumers can see the blogs written by experts about both dietary lifestyles and ingredient details. Consumers can filter the blogs both searching for their name or selecting a tag from tag list.

3.5.5.2.4 BLOG DETAIL SCREEN



Benefits of Healthy Lifestyle

7 min read

Living a healthy lifestyle can have numerous benefits.
It not only improves your physical well-being but also positively impacts your mental health.
Adopting healthy habits can lead to increased energy levels, improved mood, and a reduced risk of chronic diseases.

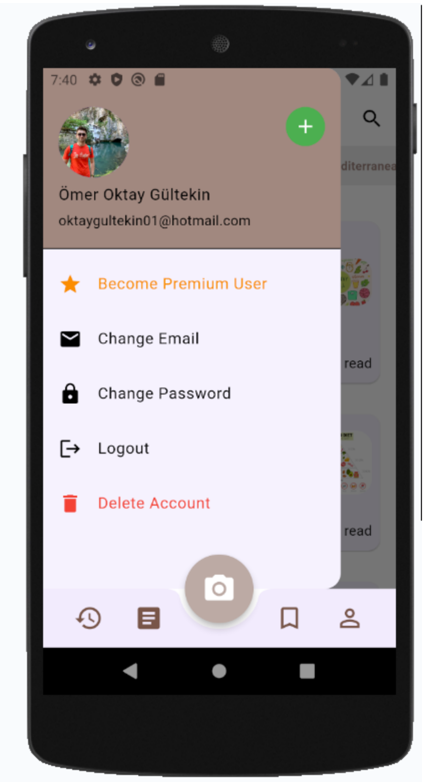
One of the key advantages of maintaining a healthy lifestyle is enhanced longevity.
Regular exercise, a balanced diet, and sufficient sleep contribute to overall well-being and can add years to your life.

Additionally, a healthy lifestyle can boost your immune system, making your body more resilient to infections and illnesses.
Consuming nutrient-rich foods, staying

Fig. 14: Blog Details Screen.

When consumers open a blog, they can see its content and they can bookmark it if they think they may need it in the future as a reference.

3.5.5.2.5 SETTINGS SCREEN



7:40

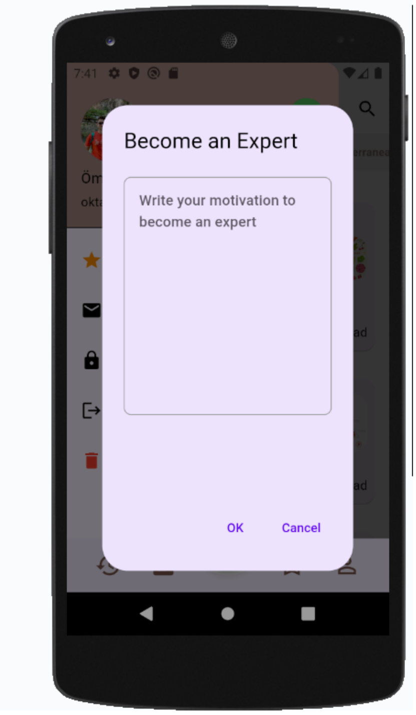
Ömer Oktay Gültekin

oktaygultekin01@hotmail.com

Fig. 15: Settings Screen.

Consumers can access this page any time by sliding the app from the left side to the right. They can change their emails, passwords, delete their account or log out. Also, they can become a premium user through this screen. Furthermore, when they click the add button at the top, They can apply to be an expert.

3.5.5.2.6 EXPERT APPLICATION SCREEN



Become an Expert

Write your motivation to become an expert

Fig. 16: Expert Application Screen.

Expert Candidates write their statement of purpose through this page. Later on, the application process will be made more complicated.

3.5.5.2.7 CANCEL EXPERT RIGHTS SCREEN

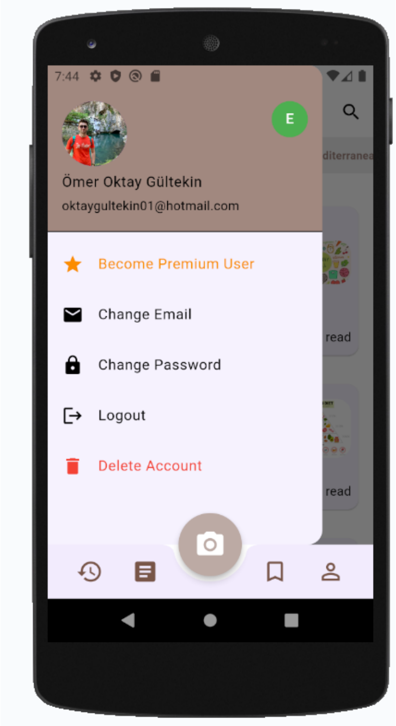


Fig. 17: Cancel Expert Rights Screen.

Expert can click the E button at the top to cancel his or her expert rights.

3.5.5.2.8 ADD NEW BLOG SCREEN

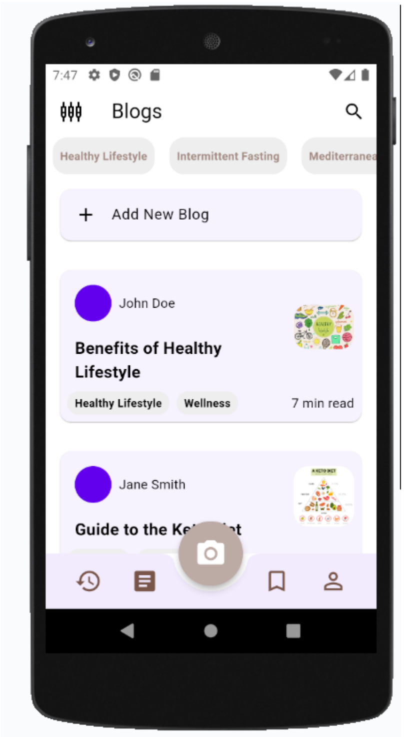
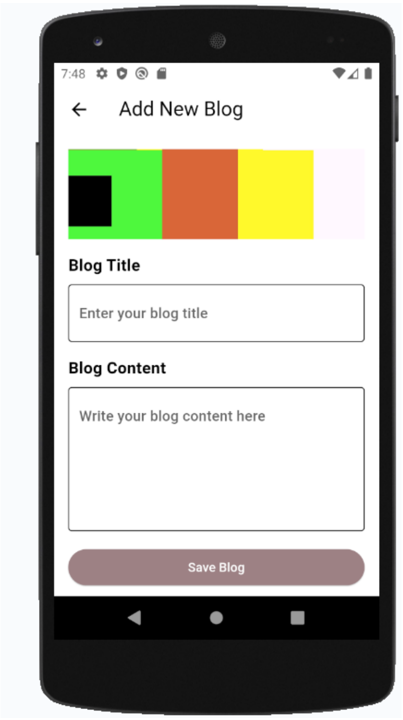


Fig. 18: Add New Blog Screen.

When a consumer becomes an expert, adding new blog options become available to her.

3.5.5.2.9 ADD NEW BLOG DETAILS SCREEN



7:48

Add New Blog

Blog Title

Enter your blog title

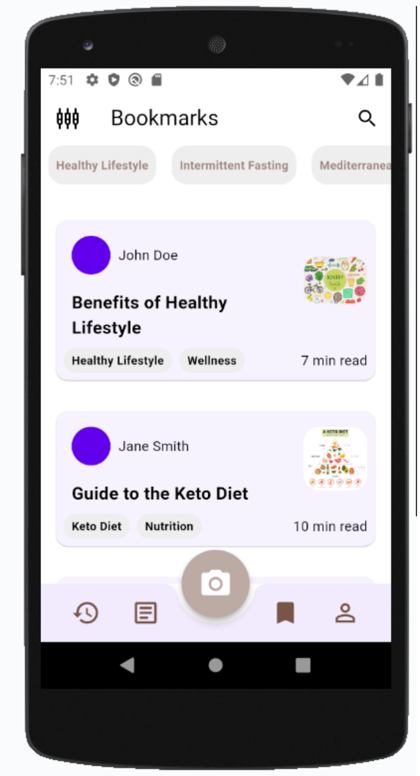
Blog Content

Write your blog content here

Fig. 19: Add New Blog Details Screen.

Then an expert adds a photo and writes the title and content to publish a blog. Save the blog button and directly publish the blog. More sophisticated blog editor will be developed in further stages.

3.5.5.2.10 BOOKMARKS SCREEN



7:51

Bookmarks

Healthy Lifestyle · Intermittent Fasting · Mediterranea

John Doe

Benefits of Healthy Lifestyle

Healthy Lifestyle · Wellness · 7 min read

Jane Smith

Guide to the Keto Diet

Keto Diet · Nutrition · 10 min read

Fig. 20: Bookmarks Screen.

Consumers can see her bookmarks directly using the bottom navigation bar.

3.5.5.2.11 PROFILE SCREEN

3.5.5.2.11.1 ADD ALLERGY TAB

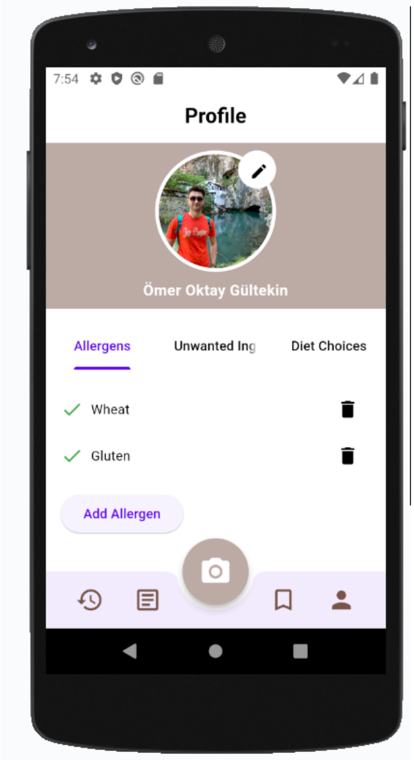


Fig. 21: Profile Screen - Add Allergy Tab.

Consumers are able to add allergens from a list of the possible allergens through their profile page. At any time, they can delete the added allergen. Also clicking to the allergen open Allergen Details Screen.

3.5.5.2.11.2 ADD UNWANTED INGREDIENTS TAB

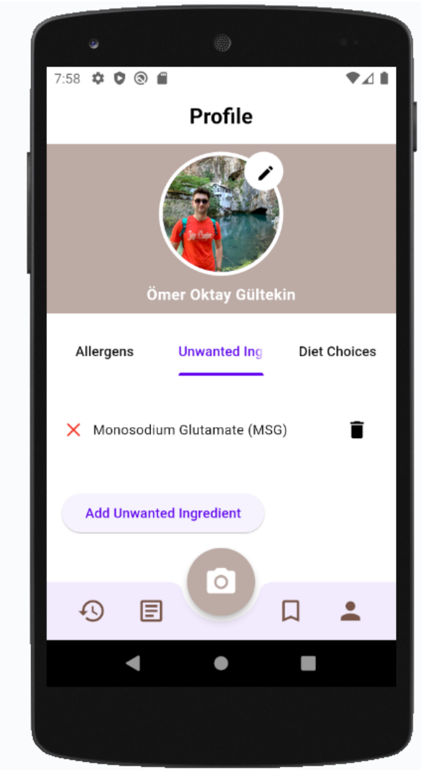


Fig. 22: Profile Screen - Add Unwanted Ingredients Tab.

Consumers are able to add unwanted ingredients from a list of the possible ingredients through their profile page. At any time, they can delete the added ingredients. Also clicking to the ingredient open the Ingredient Details Screen.

3.5.5.2.11.3 ADD DIET CHOICES TAB

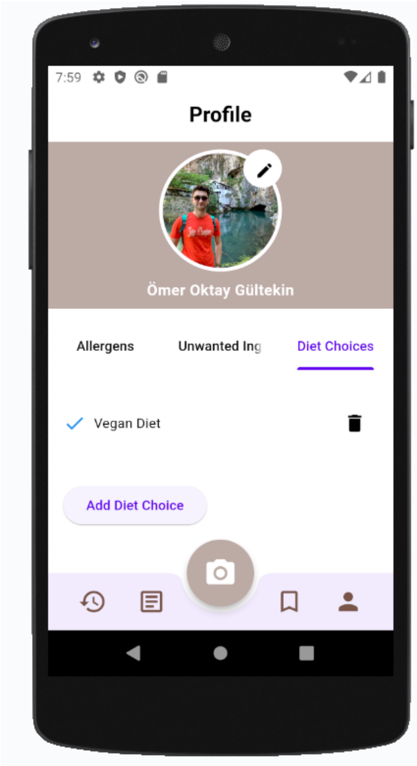
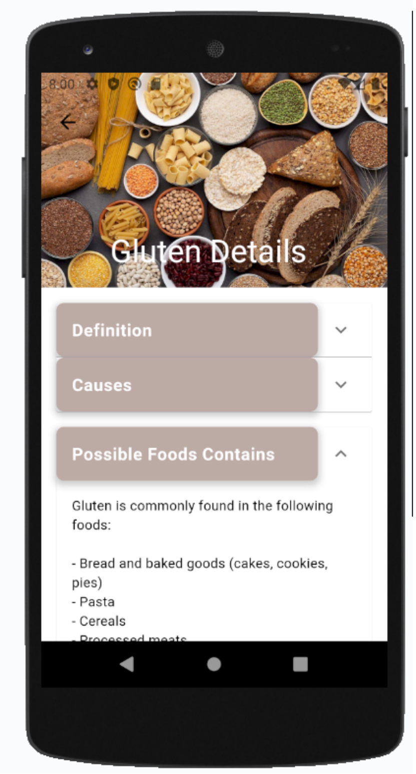


Fig. 23: Profile Screen - Add Died Choices Tab.

Consumers are able to add diet choices from a list of the possible diets through their profile page. At any time, they can delete the added diet. Also clicking to the allergen open Diet Screen.

3.5.5.2.12 ALLERGEN DETAILS SCREEN



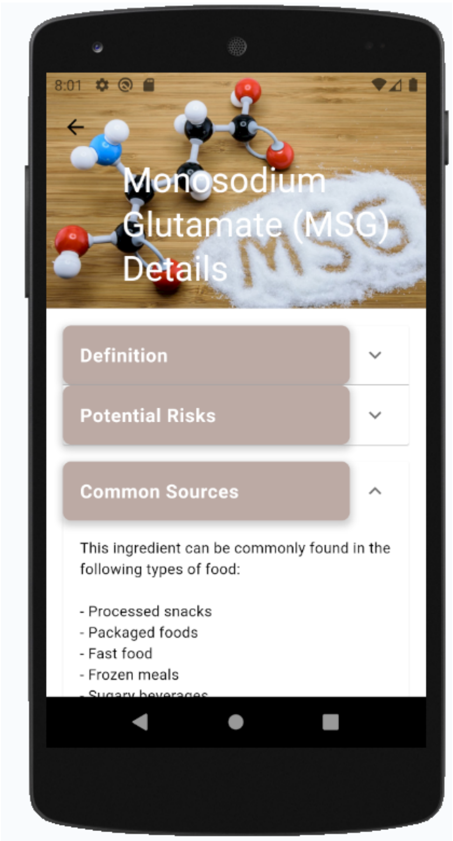
Gluten is commonly found in the following foods:

- Bread and baked goods (cakes, cookies, pies)
- Pasta
- Cereals
- Processed meats

Fig. 24: Allergen Details Screen.

Through this page, consumers see the detailed information about selected allergens which are parsed from various API's.

3.5.5.2.13 UNWANTED INGREDIENTS SCREEN



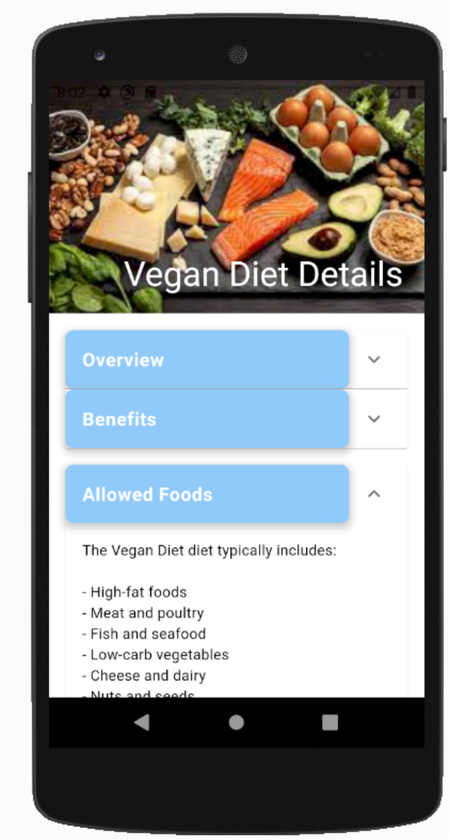
This ingredient can be commonly found in the following types of food:

- Processed snacks
- Packaged foods
- Fast food
- Frozen meals
- Sugary beverages

Fig. 25: Unwanted Ingredients Screen.

Through this page, consumers see the detailed information about selected ingredients which are parsed from various API’s.

3.5.5.2.14 DIET SCREEN



The Vegan Diet diet typically includes:

- High-fat foods
- Meat and poultry
- Fish and seafood
- Low-carb vegetables
- Cheese and dairy
- Nuts and seeds

Fig. 26: Diet Screen.

Through this page, consumers see the detailed information about selected diets which are parsed from various API’s.

3.5.5.2.15 TAKE PHOTO SCREEN

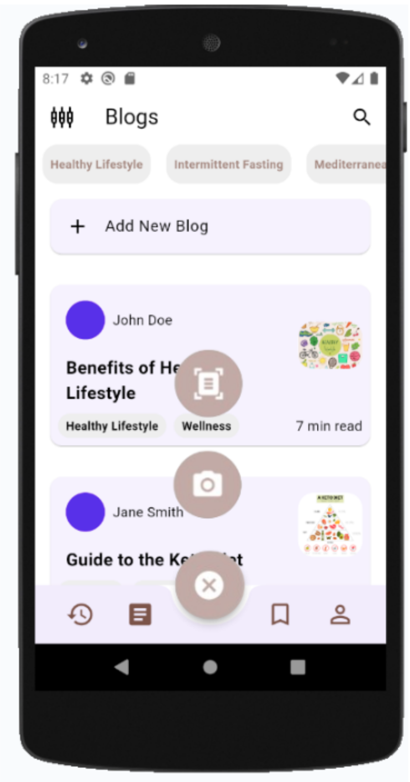


Fig. 27: Take Photo Screen.

At any time, consumers can click the camera icon at the bottom center of the screen to take a photo of a product. If they select the upper button, the app will try to parse a nutrition table. Otherwise, they select the middle button to parse the ingredients.

3.5.5.2.16 CROP IMAGE SCREEN

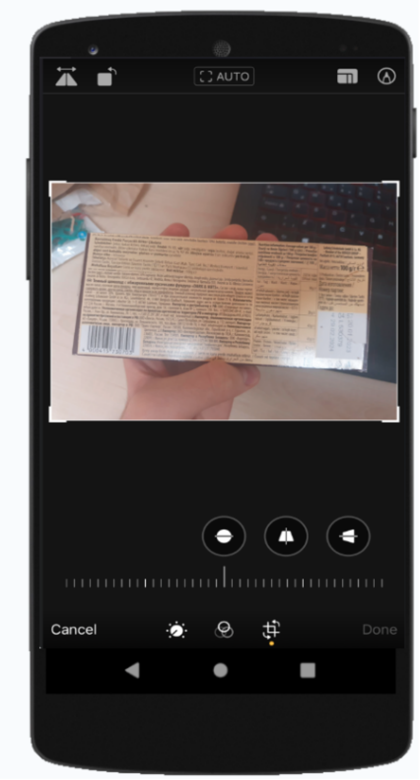
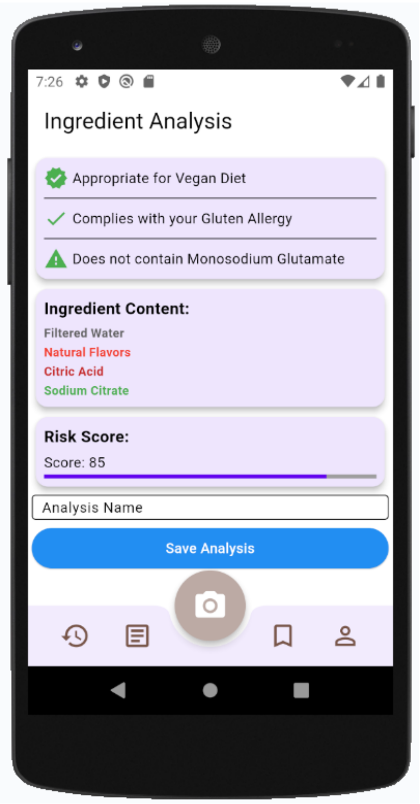


Fig. 28: Crop Image Screen.

When they click either button, the camera and then crop image screen is opened for the consumer to focus the image onto the relevant parts of the image that they want from the system to parse.

3.5.5.2.17 INGREDIENT ANALYSIS SCREEN



✓ Appropriate for Vegan Diet

✓ Complies with your Gluten Allergy

⚠ Does not contain Monosodium Glutamate

Ingredient Content:
Filtered Water
Natural Flavors
Citric Acid
Sodium Citrate

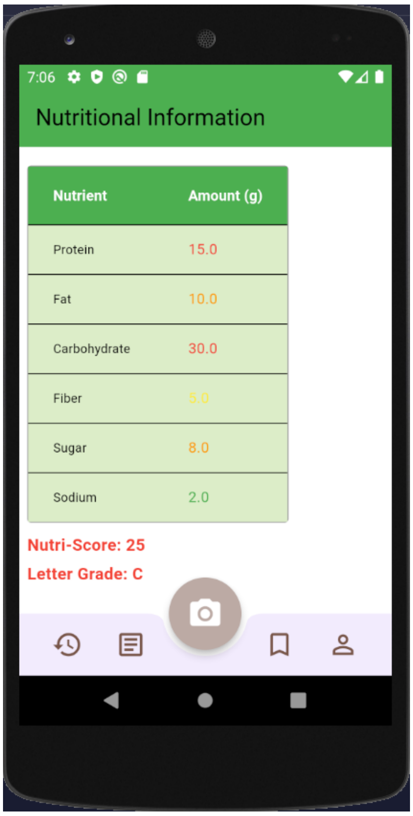
Risk Score:
Score: 85

Analysis Name

Fig. 29: Ingredient Analysis Screen.

If they are parsing the ingredient section, Ingredient Analysis Screen detects if there is anything potentially unwanted by the user by using the information given at the profile page. Also this screen gives the content of the ingredients which are all links to respected details page. Then a risk score is calculated based on the food's content. Finally, consumers can save this scan for future references.

3.5.5.2.18 NUTRITION TABLE ANALYSIS SCREEN



Nutrient	Amount (g)
Protein	15.0
Fat	10.0
Carbohydrate	30.0
Fiber	5.0
Sugar	8.0
Sodium	2.0

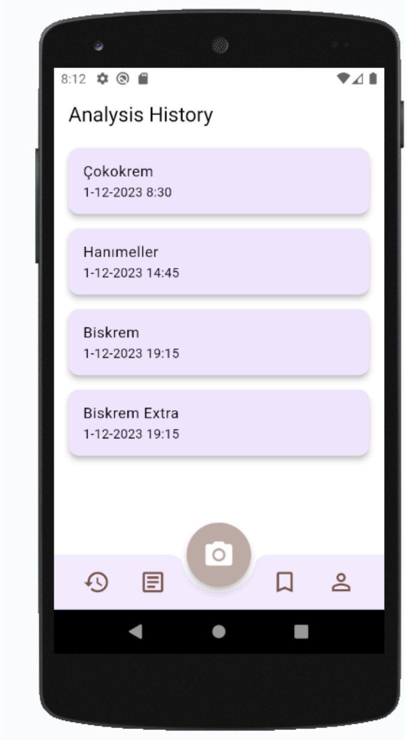
Nutri-Score: 25

Letter Grade: C

Fig. 30: Nutrition Table Analysis Screen.

If the user chooses to parse the nutrition table, the system parses the table and gives it on the screen as well as calculating its Nutri-Score value and grade.

3.5.5.2.19 HISTORY SCREEN



Analysis History

Çokokrem
1-12-2023 8:30

Hanımeller
1-12-2023 14:45

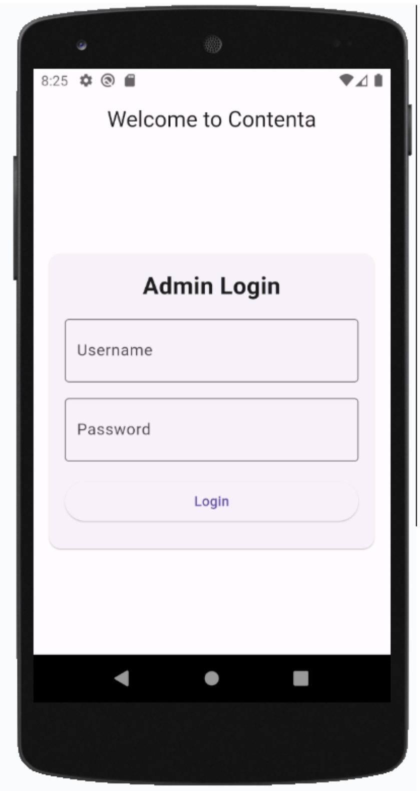
Biskrem
1-12-2023 19:15

Biskrem Extra
1-12-2023 19:15

Fig. 31: History Screen.

Consumers can see their scan history through this page. Also, they can click to a list item to open the details of the scan.

3.5.5.2.20 ADMIN LOGIN SCREEN



8:25

Welcome to Contenta

Admin Login

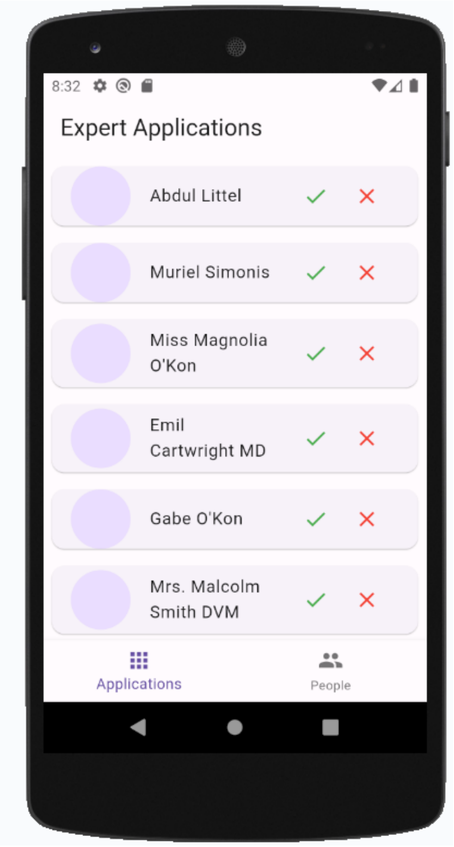
Username

Password

Fig. 32: Admin Login Screen.

Admin user login to the system by possibly using a different app so that consumers may not “accidentally” find an admin account. Note that admin users are registered to the system directly from the database.

3.5.5.2.21 ADMIN VIEW EXPERT APPLICATIONS SCREEN



Expert Applications

Abdul Littel ✓ ✕

Muriel Simonis ✓ ✕

Miss Magnolia O'Kon ✓ ✕

Emil Cartwright MD ✓ ✕

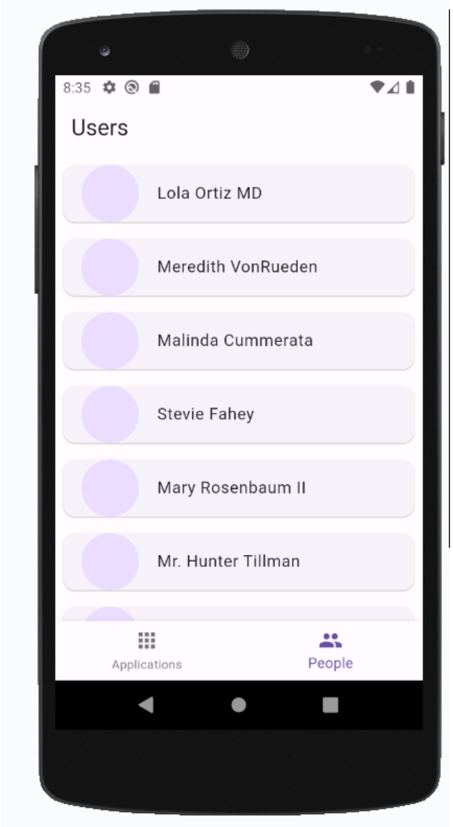
Gabe O'Kon ✓ ✕

Mrs. Malcolm Smith DVM ✓ ✕

Fig. 33: Admin View Expert Applications Screen.

Admins can see the applications of experts and clicking on an item opens the details of the application.

3.5.5.2.22 ADMIN VIEW USERS SCREEN



8:35

Users

Lola Ortiz MD

Meredith VonRueden

Malinda Cummerata

Stevie Fahey

Mary Rosenbaum II

Mr. Hunter Tillman

Fig. 34: Admin View Users Screen.

Admins can see the user list and by clicking on an item, they can do various operations such as removing users.

4.0 OTHER ANALYSIS ELEMENTS

4.1. CONSIDERATION OF VARIOUS FACTORS IN ENGINEERING DESIGN

- **Public Health (10/10):** The primary focus of *Contenta* is to promote public health and well-being through its innovative approach to content consumption. By providing users with tools to make informed and health-conscious choices about their diets and lifestyles, *Contenta* aims to contribute significantly to public health. The application's features, such as ingredient scanning and health insights, are carefully designed to positively impact users' health outcomes, earning a maximum rating of 10 in this category.

- **Global and Cultural Factors (2/10):** *Contenta*'s impact on global and cultural factors is relatively limited. While the application has the potential to transcend geographical boundaries, its core functionality is centered around health-conscious consumption, which may not be universally applicable. Cultural nuances related to dietary preferences and health practices may vary, resulting in a lower rating of 2 in terms of global and cultural factors.

- **Social Factors (3/10):** *Contenta*, not being a social media platform, has a moderate impact on social factors. It focuses on individual health rather than fostering extensive social interactions. The application's design emphasizes user privacy and personalized experiences, limiting its direct influence on social dynamics. Therefore, it receives a rating of 3 in the social factors category.

- **Environmental Factors (4/10):** While *Contenta* does not directly address environmental concerns, it indirectly supports environmental goals by encouraging users to adopt plant-based diets. The promotion of vegan or vegetarian lifestyles can contribute to reduced environmental impact, aligning with sustainability objectives. Therefore, *Contenta* receives a moderate rating of 4 in the environmental factors category.

- **Economic Factors (6/10):** *Contenta* demonstrates a significant potential to contribute to the local economy through its advertising and subscription models. By supporting local businesses and potentially generating employment opportunities in the future, Contenta aligns with economic factors. The application's ability to drive revenue and support local economies earns it a rating of 6 in the economic factors category.

In summary, *Contenta's* engineering design is deeply influenced by its core objective of promoting public health. While its impact on global and cultural factors is limited, it has moderate effects on social, environmental, and substantial effects on economic factors. The following table provides a concise overview of the considerations and their respective effects:

Table 1: Table of various factors and their effects.

Factor	**Effect Rating**
Public Health	10/10
Global and Cultural Factors	2/10
Social Factors	3/10
Environmental Factors	4/10
Economic Factors	6/10

4.1.1 CONSTRAINTS

There are lots of constraints to bear in mind while developing and maintaining *Contenta*, but the following three are probably the most major ones.

- **Legal Considerations and Privacy**: Contenta must prioritize legal considerations and privacy standards to ensure compliance with data protection regulations. As the application involves scanning and processing user-generated content, it is imperative to uphold intellectual property rights, content distribution policies, and user data privacy. Strict adherence to legal frameworks will establish trust with users and safeguard the application's reputation.

- **Marketability and User Engagement:** Ensuring Contenta's marketability involves addressing the challenge of user adoption and retention. While encouraging users to download and retain the app on their phones can be challenging, implementing engaging features, personalized content recommendations, and periodic updates can enhance user retention. Moreover, considering the marketability of a premium version with additional features can create revenue streams and further incentivize user commitment.

- **Maintainability and Database Management:** Contenta's maintainability is crucial, particularly with the potential growth in the size of the database due to the storage of users' scanning history. A robust database management system should be in place to handle the increasing volume of data efficiently. Regular maintenance procedures and optimization strategies will ensure the application's continued smooth operation, preventing performance bottlenecks and enhancing the overall user experience.

4.1.2 STANDARDS

The *Contenta* mobile application development adheres to internationally recognized engineering standards to ensure a robust and quality-driven software life cycle. The project aligns with the ISO/IEC/IEEE International Standard for Systems and Software Engineering -- Software Life Cycle Processes, emphasizing the systematic and well-defined processes essential for the development, testing, and maintenance phases.

For the formulation of software requirements, the ISO/IEEE Recommended Practice for Software Requirements Specifications serves as a foundational guide. This standard ensures the clarity, completeness, and consistency of software requirements documentation, contributing to effective communication between stakeholders and facilitating a comprehensive understanding of project specifications.

In the realm of modeling and design, the project employs the Unified Modeling Language (UML) 2.5.1, a widely accepted standard for visualizing, specifying, constructing, and documenting software artifacts. UML's standardized notation and semantics provide a universal language for the project team, fostering clear communication and promoting a shared understanding of the system architecture.

Moreover, the structure of the analysis and requirements report, including this one, aligns with the IEEE report writing guidelines. This adherence ensures a logical and coherent presentation of information, facilitating a comprehensive understanding of the project's progress and requirements. While maintaining the integrity of IEEE guidelines, a departure from the traditional two-column structure has been made to enhance readability and accessibility for a broader audience.

4.2 RISKS & ALTERNATIVES

The true motive for *Contenta* is to give customers fast and accurate results on the ingredients within a product. Here, the first risk is the app taking too much time depending on the processor of the phone. If the phone is too slow, the app may take longer than the estimated 3 to 8-second interval we had evaluated from our initial tests. The possible alternative to our design would then be to have a server process the taken image.

Another risk is the accuracy of our scans. The cropped photo section may not visibly have all the ingredients, which would be a human error and may cause the app to give a smaller number of total ingredients, perhaps not reading a crucial ingredient and stating that the product seems safe to use. This situation may particularly arise if the product has shiny packaging or round packaging, and the user takes the photo without realizing that some part of the image is unreadable, for which we plan to have the user add multiple photos and get the ingredient results accordingly. Furthermore, the scan may misread a word, mistaking it to be an ingredient it is not, or perhaps misreading an ingredient and not putting it to the list of ingredients found. From our tests, we believe that the odds of these falsehoods happening are low, although they are definitely present, and we are informing the users of the possible falsehoods, telling them to check if the found ingredients are actually within the package.

The risk of accuracy is also present in our blogs. Since the experts will be writing blogs, they must be fact-checked to ensure no false data is given to the users. The fact-checking mechanism may be difficult to follow once the system reaches a great number of blogs, and perhaps old blogs may contain misinformation. In these cases, the experts are notified by our admins and the users may also contact the system for correction. To prevent these misinformations, we give requests, warnings, as well as bans to experts who spread false blogs, thus hoping to effectively increase the expert attention on the older blogs they had published. We believe this solution would decrease the falsehood, although false blog information still remains a risk.

Table 2: Risks & effects table.

	Estimated Likelihood of Error	Effect on Project	(Alternative) Measure
Long Processing Time on Slow Phone Processors	About 1 out of 1000 users may have an outdated phone processor	The shopping process becomes slow, as the process becomes a hindrance	Putting the process step to a server (Causes the app to only work with the internet)
Scanned Ingredient Accuracy	About 1 ingredient in every 10 product scans may be missing, or false	With every misrecognized ingredient, the users may have a negative feedback on our app	Adding multiple photos of the same product's contents list
Blog Misinformation	About 1 out of 10 blogs may become false each month	Users may be given misinformation, decreasing the trustworthiness of experts and blogs, as well as the app.	Having stricter measures and policies against false blog detection and penalization

In table 2, we give the summary of this section with some estimated likelihood of errors. Do note that these estimations were purely to show the amount of error we are currently expecting from our research and these values are purely given with respect to our expectations. The actual error likelihoods can only be calculated upon the completion of the app, so these values are only to share our perceptions on the current situation at hand.

4.3 PROJECT PLAN

In the tables below, one can see the project plan until the end of the semester including the deadlines, work packages, and its leaders.

Table. 3: Project plan table.

Package Name	Date	Leader	Involved Members
Reports & Documentation	01.10.2023 – 01.05.2024	Barış Tan Ünal	All Team Members
Image to Text API Trials and Research	15.10.2023 – 22.10.2023	Ömer Oktay Gültekin	Mert Ünlü
Frontend Implementation	01.12.2023 – 01.03.2024	Ömer Oktay Gültekin	Oğuz Kuyucu, Mert Ünlü
Backend Implementation	01.12.2024 – 01.04.2024	Alperen Yalçın	Mert Ünlü, Barış Tan Ünal
API Integration of Backend and Frontend	15.12.2024 – 01.04.2024	Oğuz Kuyucu	Mert Ünlü, Ömer Oktay Gültekin, Oğuz Kuyucu
Database Implementation	01.01.2024 – 01.04.2024	Mert Ünlü	Alperen Yalçın, Barış Tan Ünal, Oğuz Kuyucu
Project Demo	10.12.2023 – 20.12.2023	Barış Tan Ünal	All Team Members
Test & Optimization	15.03.2024 – End of Semester	Alperen Yalçın	All Team Members
Project Launch	End of Semester	Mert Ünlü	All Team Members

1) Reports & Documentation

Work Package 1: Reports & Documentation
Duration: 01.10.2023 – 01.05.2024
Leader: Barış Tan Ünal **Members:** All Members
Objectives: To produce informative reports in order to keep all developers and stakeholders on the same page.
Tasks: **Task 1.1** Deliver Project Specification Report **Task 1.2** Deliver Analysis and Requirements Report **Task 1.3** Deliver Detailed Design Report **Task 1.4** Deliver Final Report
Deliverables: **D1.1** Project Specification Report **D1.2** Analysis and Requirements Report **D1.3** Detailed Design Report **D1.4** Final Report

2) **Image to Text API Trials and Research**

Work Package 2: Image to Text API Trials and Research
Duration: 15.10.2023 – 22.10.2023
Leader: Ömer Oktay Gültekin **Members:** Ömer Oktay Gültekin, Mert Ünlü
Objectives: Evaluate and select an appropriate Image to Text API for integration into the project.
Tasks: **Task 2.1** Conduct a comprehensive review of available Image to Text APIs. **Task 2.2** Perform trials with selected APIs to assess accuracy, speed, and suitability. **Task 2.3** Analyze costs and licensing terms of potential APIs.
Deliverables: **D2.1** A primitive image to text functionality that is implemented with the chosen API.

3) **Frontend Implementation**

Work Package 3: Frontend Implementation
Duration: 01.12.2023 – 01.03.2024
Leader: Ömer Oktay Gültekin **Members:** Oğuz Kuyucu, Mert Ünlü
Objectives: Develop the user interface and interactive components for the project.
Tasks: **Task 3.1** Create wireframes and design mockups for the frontend. **Task 3.2** Implement the frontend using the selected framework. **Task 3.3** Ensure responsive design for various screen sizes and devices.
Deliverables: **D3.1** Mockup Pages **D3.2** Frontend Codebase

4) **Backend Implementation**

Work Package 4: Backend Implementation
Duration: 01.12.2024 – 01.04.2024

Leader: Alperen Utku Yalçın **Members:** Mert Ünlü, Barış Tan Ünal
Objectives: Develop the server-side logic and functionalities for the project.
Tasks: **Task 4.1** Define the backend architecture and data models. **Task 4.2** Implement core functionalities and business logic. **Task 4.3** Set up server infrastructure and configure databases.
Deliverables: **D4.1** Backend Architecture and Data Models **D4.2** Backend Codebase

5) API Integration of Backend and Frontend

Work Package 5: API Integration of Backend and Frontend
Duration: 15.12.2024 – 01.04.2024
Leader: Oğuz Kuyucu **Members:** Mert Ünlü, Ömer Oktay Gültekin, Oğuz Kuyucu
Objectives: Integrate the frontend and backend components to ensure communication and functionality with the API.
Tasks: **Task 5.1** Integrate the Image to Text API with the frontend. **Task 5.2** Conduct integration tests to verify system functionality **Task 5.3** Implement user authentication and authorization mechanisms.
Deliverables: **D5.1** Accurately functioning image-to-text API **D5.2** Functioning authentication API

6) Database Implementation

Work Package 6: Database Implementation
Duration: 01.01.2024 – 01.04.2024
Leader: Mert Ünlü **Members:** Alperen Yalçın, Barış Tan Ünal, Oğuz Kuyucu
Objectives: Design and implement the database structure to efficiently store and retrieve data.
Tasks: **Task 6.1** Implement database tables and relationships. **Task 6.2** Optimize database queries for performance. **Task 6.3** Ensure data security and implement backup mechanisms.

Deliverables: **D6.1** Implemented Database **D6.2** Implemented Backup Mechanism

7) Project Demo

Work Package 7: Project Demo
Duration: 10.12.2023 – 20.12.2023
Leader: Barış Tan Ünal **Members:** All Team Members
Objectives: Prepare and conduct a demonstration of the project's features and functionalities.
Tasks: **Task 7.1** Create a demo script outlining key features to showcase. **Task 7.2** Set up demo environments for both development and presentation. **Task 7.3** Conduct a practice run-through of the demo.
Deliverables: **D7.1** Demo Script **D7.2** Demo Environment Setup Documentation

8) Test & Optimization

Work Package 8: Test & Optimization
Duration: 15.03.2024 – End of Semester
Leader: Alperen Utku Yalçın **Members:** All Team Members
Objectives: Perform thorough testing and optimize the project for performance and user experience.
Tasks: **Task 8.1** Identify and address bugs and issues. **Task 8.2** Optimize ingredient detector algorithm to increase accuracy. **Task 8.3** Optimize code and system performance. **Task 8.4** Perform usability testing and gather user feedback.
Deliverables: **D8.1** Increased User Experience **D8.2** More Accurate Detection Algorithm

9) Project Launch

Work Package 9: Project Launch
Duration: End of Semester
Leader: Mert Ünlü **Members:** All Team Members
Objectives: Prepare for the official launch of the project.
Tasks: **Task 9.1** Develop a launch plan including marketing and communication strategies. **Task 9.2** Deploy the project to the production environment. **Task 9.3** Monitor system performance and address any issues. **Task 9.4** Communicate the project launch to stakeholders and end-users.
Deliverables: **D9.1** Launch Plan

4.4 ENSURING PROPER TEAMWORK

Ensuring proper teamwork within our project is a prioritized focus, mainly due to the fact that our team is developing as smaller subgroups with different tasks throughout the development process. To be able to efficiently balance the workload and make everyone as efficient as possible, we are and will make use of specific development planning/scheduling tools to easily track our progress and to remain in contact as well as be organized as a team.

4.4.1 COLLABORATION TOOLS

GitHub: For our project, GitHub not only facilitates version control but also displays individual contributions. With regular commits, pull requests, and code reviews, we intend to closely monitor and document each team member's involvement in the project. In addition, code reviews will also be beneficial in terms of everyone's general knowledge about the project parts that they did not implement.

Jira: Recognizing the significance of comprehensive project management, we have decided to integrate Jira into our workflow. Jira provides our team with the capability of tracking issues, managing tasks, and planning the remainder of the project. With Jira’s customizable workflows and project board, we aim to streamline our development process. It is also very beneficial in terms of awareness of other team members' ongoing development activities.

4.4.2 SHARED LEADERSHIP AND INCLUSIVITY

To foster shared leadership and inclusivity, we are implementing the following strategies:

- **Regular Team Meetings:** Consistent and well-planned team meetings play a crucial role in maintaining a healthy development process. During these sessions, as a team we aim to brainstorm for a problem or help a member’s current problem. Main goal is to stay in contact and also help to solve a problem which is blocking the development process. With the regularity of these sessions, we are aiming to ensure that the team is always in contact and up to date in every part of the project, in addition, a blocking problem is brainstormed to be solved together.

- **Rotational Leadership Roles:** Leadership roles will be rotated among team members throughout the project's duration since the leaders change for each work package. It not only provides every team member with an opportunity to lead but also encourages a collaborative and inclusive environment.

- **Peer Feedback:** By regularly carrying out peer feedback sessions, we aim to achieve shared leadership and equal participation throughout the project. We aim to create an environment that encourages continuous improvement.

4.5 ETHICS & PROFESSIONAL RESPONSIBILITIES

Our dedication to protecting user privacy dominates every element of Contenta's functioning and design. We seek to establish strong data protection mechanisms in operation while users engage with the scanning feature so that their sensitive information is handled with the utmost care. In order to establish a safe environment and protect user data from any potential unwanted access or breaches, strict protocols and encryption techniques are used.

It’s also important to acknowledge that the Contenta app may not guarantee error-free results. To uphold ethical standards and prioritize user trust, Contenta will consistently display warnings and error cautions for any inaccuracies. This transparent approach not only aligns with our commitment to responsible development practices but also makes sure that users are well-informed about the limitations of the application, creating a relationship built on trust and integrity which is a high priority of our project.

4.6 PLANNING FOR NEW KNOWLEDGE & LEARNING STRATEGIES

Learning Technologies:

Our diverse technology stack includes Flutter, Firebase, Dart, GitHub, Jira, and the incorporation of APIs. To ensure comprehensive proficiency, we've structured our learning plan as follows:

- **Flutter and Dart Mastery:** While Git/GitHub is relatively more familiar to our team members, Flutter comes with a learning curve since it will be the first Flutter-based project for our team members even though we, as a team, have a mobile development background. Our objective is to achieve proficiency in Flutter for cross-platform mobile development. Dart, as the programming language of Flutter, will be mainly learned and practiced with official documentation and online tutorials such as courses from Udemy or YouTube supported by Stack Overflow examples, and practical experience during the project's development.

- **Firebase Integration:** Being aware of Firebase's importance in our project, our team is also committed to learning and implementing Firebase for the backend. In contrast to Flutter, the majority of our team has a stronger background in Firebase implementation. Therefore a faster and more generic crash-course approach will be followed. This includes understanding Firebase's real-time database capabilities, backend services and integration with Flutter.

- **API Integration:** Acknowledging the role of APIs in enhancing Contenta's functionality, we plan to deepen our understanding of API integration. This learning phase will involve exploring various API use cases, understanding their documentation, and hands-on experience in incorporating APIs within the Flutter framework for the main functionalities of our app.

Knowledge Division and Collaboration:

Our learning strategy involves a collaborative and divisional approach:

- **Flutter and Dart Language:** Team members with varying degrees of mobile development experience will focus on mastering Flutter and Dart through online courses. More experienced members will actively mentor others while they are also learning. We also aim to achieve a more efficient learning process for both more and less experienced team members.

- **Firebase and API Integration:** Recognizing the complexities involved, the team will collectively focus on Firebase and API integration. Similarly, official documentation and tutorials supported by other online resources, group discussions, and practical exercises will be utilized to ensure a shared understanding and demonstrate effective integration of these technologies. Especially, for our main API which we will use for our image-to-text functionality, the team members will update other members for every new know-how gained to achieve a cumulative know-how team-wide.

By combining online tutorials, collaborative learning, and hands-on experiences, our team aims to acquire the necessary skills, ensuring *Contenta*'s successful development and its seamless integration with APIs, Firebase, and other technologies.

5.0 CONCLUSION

In conclusion, the analysis and requirements report for the Contenta mobile application has provided a comprehensive understanding of the current system, proposed system, and various elements crucial for its successful development.

In Section 2.0, the examination of the current system highlighted existing functionalities, drawbacks, and the need for an enhanced solution. This set the stage for the proposed system, discussed in Section 3.0. The overview (Section 3.1) outlined the general framework, while Sections 3.2 and 3.3 delved into the specific functional and non-functional requirements. Noteworthy among the latter were user-friendliness (Section 3.3.1), maintainability (Section 3.3.2), scalability (Section 3.3.3), performance (Section 3.3.4), and privacy (Section 3.3.5).

The pseudo requirements (Section 3.4) addressed additional considerations, ensuring a holistic approach to system development. System models, discussed in Section 3.5, provided a detailed perspective, encompassing scenarios, use case models, object and class models, dynamic models, and user interface specifications, including navigational paths and screen mock-ups.

Moving beyond the system-centric aspects, Section 4.0 explored other critical elements. The consideration of various factors in engineering design (Section 4.1) touched upon constraints and standards, emphasizing the need for a well-defined framework. Risks and alternatives (Section 4.2) were assessed, outlining potential challenges and mitigation strategies. The project plan (Section 4.3) provided a roadmap for implementation, while aspects such as teamwork (Section 4.4), ethics and professional responsibilities (Section 4.5), and planning for new knowledge and learning strategies (Section 4.6) underscored the importance of a holistic approach.

In conclusion, the analysis and requirements report serves as a comprehensive guide for the development of the Contenta mobile application. By addressing current system limitations, proposing robust requirements, and considering diverse factors in engineering design, the report lays a solid foundation for a successful project. As the team embarks on this journey, close attention to the outlined considerations will ensure the creation of a mobile application that not only meets user expectations but also adheres to ethical and professional standards.

6.0 LIST OF ABBREVIATIONS

- AI: Artificial Intelligence
- EU: European Union
- GDPR: General Data Protection Regulation
- KVKK: Kişisel Verilerin Korunması Kanunu (Personal Data Protection Law of Türkiye)
- Nutri-Score: Nutrition Score (a.k.a 5-Colour Nutrition Label)
- UI: User Interface
- UK: United Kingdom
- FDA: Food Drug Administration

REFERENCES

[1] M. Janani, P. Selvasekaran, M. Lokanadham, and R. Chidambaram, "Food and food products associated with food allergy and food intolerance – An overview," Food Research International, vol. 138, Part B, pp. 109780, 2020, ISSN 0963-9969, [Online]. Available: https://doi.org/10.1016/j.foodres.2020.109780.

[2] Calderone, Julia. "24 foods that artificial sweeteners are hiding in". Business Insider. Accessed: Dec. 08, 2023. https://www.businessinsider.com/surprising-foods-that-artificial-sweeteners-are-hiding-in-2016-1.

[3] "Acesulfame Potassium". Center for Science in the Public Interest. Accessed: Nov. 16, 2023. https://www.cspinet.org/article/acesulfame-potassium.

[4] Lapidos, Rachel. "What Derms Want You to Know About 'Controversial' Skin-Care Ingredients". *Well+Good*. Accessed: Dec. 08, 2023. https://www.wellandgood.com/cosmetic-ingredient-review/

[5] Türkiye, TBMM. (2016, Mar. 24). Kişisel Verilerin Korunması Kanunu. Accessed: Nov. 16, 2023. https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1 &MevzuatTertip=5.