Bilkent University
Department of Computer Engineering

Senior Design Project

Detailed Design 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
15.03.2024

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 detailed design report presents a comprehensive blueprint for the development of the Contenta mobile application, which utilizes cutting-edge image recognition technology to enable users to scan the ingredients section of food products via their mobile phone cameras. By implementing advanced algorithms, the application identifies potential hazards such as allergens, including aspartame, monosodium glutamate, and E621. Users can personalize their experience by inputting details about their allergies, height, and weight, allowing the application to accurately identify allergenic ingredients and provide insights into potential health risks. Initially designed to support Turkish and/or English, the application is poised for global expansion through the integration of a seamless translation API. Future iterations may broaden the application's scope to include products beyond its initial focus, such as cosmetics or cleaning goods. This report meticulously delineates the technical specifications and features essential for the successful development and deployment of the Contenta mobile application, contributing to a more informed landscape of content consumption and promoting health-conscious choices among users worldwide.

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

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

Purchasing packaged goods is a regular practice for some residents, and as consumers' awareness grows, so does their desire to know what's in the products they consume. Some consumers have to carefully consider which products they should or should not choose when making purchases, based on their inner components. It is vital to thoroughly read the ingredients section on the label, as there may be components in this government-approved and ostensibly safe product that could cause the client special problems.

Even if customers pay great attention to the ingredients section, it might be particularly difficult for them to understand the contents on the labels that describe them if they have a limited understanding of the ingredients, their history, or potential hazards. Customers who have intolerances to specific substances, who could be seriously harmed if they were to ingest a particular product, are particularly impacted by this problem [1]. Furthermore, individuals who prefer not to consume any meat, pork-related foods, corn syrup, gluten, or any other substance on a daily basis may wish to avoid any associated ingredients. In addition to the more well-known components, there are also several potentially hazardous ones that consumers may not be aware of. It is very challenging to learn every product's ingredients when shopping in a mall, therefore users should be well-informed before purchasing any product that contains hazardous but government-approved material.

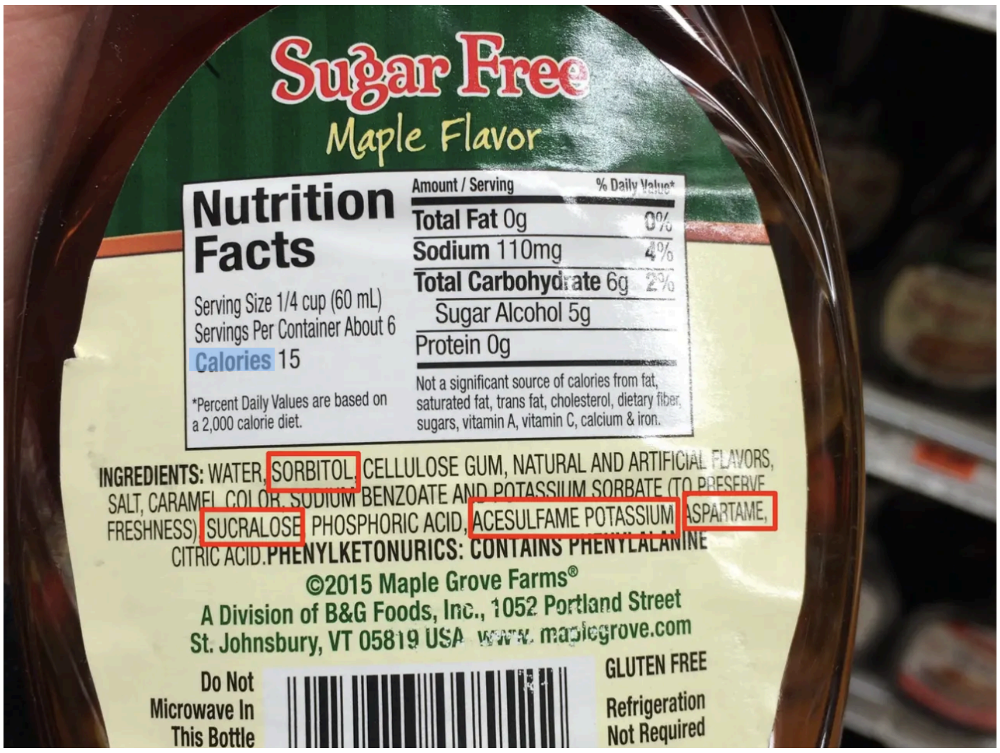
Sugar Free
Maple Flavor
Nutrition Facts
Serving Size 1/4 cup (60 mL)
Servings Per Container About 6
Calories 15
Amount / Serving
% Daily Value*
Total Fat 0g 0%
Sodium 110mg 4%
Total Carbohydrate 6g 2%
Sugar Alcohol 5g
Protein 0g
INGREDIENTS: WATER, SORBITOL, CELLULOSE GUM, NATURAL AND ARTIFICIAL FLAVORS, SALT, CARAMEL COLOR, SODIUM BENZOATE AND POTASSIUM SORBATE (TO PRESERVE FRESHNESS), SUCRALOSE, PHOSPHORIC ACID, ACESULFAME POTASSIUM, ASPARTAME, CITRIC ACID. PHENYLKETONURICS: CONTAINS PHENYLALANINE
GLUTEN FREE

Fig. 1: Rear face of a sugar-free maple flavor [2].

There are numerous ingredients that a less informed consumer would not be familiar with in-depth. On the spot, customers may choose to purchase a product after taking into account all of the contents they are aware of and ignoring the ones with unusual names. Additionally, people may think that all listed components are essential to the product, or perhaps because the government has authorized it, the food must be safe to eat. Hence, customers may purchase goods that endanger their own health, even assuming they read the ingredients section carefully. One such event may happen with the ingredient "Acesulfame Potassium," a typical sweetening component used in sugar-free flavors, as shown in Figure 2. However, after a quick search, the customers can see that the substance is labeled as a health risk to pregnant people and as a possible cause of cancer [2][3]. Certain people would most likely decide to not consume such a product after reading about its hazards. Hence, especially consumers who wish to be attentive to their intake should be more aware of substances similar to these and cautious about the items they consume that include potentially harmful ingredients. *Contenta* is therefore an app that aims to help users with their everyday product consumption by giving them transparency about the contents of the products they consume. It is designed to give users quick and efficient information.

In this report, we will first propose our system by explaining its purpose and then we will state our design goals. Then, we will go over the sample software architecture utilized by the other apps that customers use for ingredient analysis, under the current software architecture section. Next, the proposed software architecture section will delve deeper into the description of our system where we will go over how the proposed system will exist as a standalone application. Afterward, we will go over the subsystems that will serve our app. Later, test cases will show how our designed system is expected to function in certain procedures and test scenarios. Next, we will explain the constraints of various factors, as well as the standards we are utilizing in the development of our engineering design. Finally, in the teamwork details section, we will give details of our teamwork process.

1.1 PURPOSE OF THE SYSTEM

Contenta proposes to aid customers who wish to pay attention to the inner contents of the products they consume. The app is designed to serve customers while they are shopping, where the customer is able to scan the ingredients section of a product, and they can learn much about the ingredients. The customers are informed of the contents in a quick manner, and they can grasp a general idea of whether this product contains an ingredient that is potentially harmful to them while generating a general idea of what the customer would consume. Hence, the users of *Contenta* are able to have a transparent understanding of product ingredients quickly, while being able to skip the steps of having to read the ingredients sections written in small fonts. Users are able to understand all the unfamiliar chemical names or codes, what each of them does, and their potential health hazards, as well as additional resources and informative explanations.

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

1.2 DESIGN GOALS

The following subsections demonstrate the most important design goals we are trying to achieve. The main purpose and the users of the application, along with the expectations from contenta are sought while determining these items.

1.2.1 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 smallest font should be at least 11 so that hypermetropic users can use the application easily. 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.

1.2.2 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. Besides, if a user receives personal advice from his/her doctor about an ingredient, he/she can add that ingredient to the blacklist without needing any software update. 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.

1.2.3 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.

1.2.4 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 use google ML kit to process the image. If we were to give an approximate number from the initial tests we performed, the output should be created in 5 seconds on a Snapdragon 855 or higher at maximum with the internet speed minimum 10 mbit/second. Since in Turkey and in the world, users' internet speeds are different, 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.

1.2.5 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.

1.3 DEFINITIONS, ACRONYMS, AND 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

1.4 OVERVIEW

In response to consumers' increasing knowledge of and desire to understand what they consume, Contenta is a unique application that provides transparency and insight into the product's contents. The application assists users in identifying potentially hazardous compounds, dietary restrictions, and ingredient functions by easing the process of ingredient analysis through rapid scans. Because of its non-product-based approach and user-friendly design, it is more scalable and maintainable for users of all ages. Contenta promotes user privacy and compliance with Turkish data protection regulations, ensuring openness and control over personal data while utilizing cutting-edge technology for quick picture processing. Overall, Contenta facilitates informed decision-making for consumers and establishes a new standard for product awareness and consumer engagement.

2.0 CURRENT SYSTEM ARCHITECTURE

In the current market, there are two direct competitors which are aiming to inform users about packaged foods' ingredients.

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 items 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 SOFTWARE ARCHITECTURE

3.1 OVERVIEW

Contenta's primary objective is to offer users swift insights into the contents of their packaged products while they're on the move. Through the app, consumers can effortlessly scan product ingredients to access comprehensive information, particularly regarding any health implications associated with those ingredients. Beyond merely alerting or educating users about potential risks and backgrounds, the application endeavors to accurately interpret and present ingredient lists in a user-friendly format. Acting as an assistant tool, *Contenta* caters to users adhering to specialized diets by helping them avoid specific ingredients. The app's healthcare-centric aim is to promptly furnish precise information, thereby enhancing consumer awareness of health-related matters and expediting purchasing decisions for those seeking product transparency. Users have the option to create personalized profiles within the app, where they can input details such as allergies, health concerns, pregnancy status, and dietary preferences. Additionally, users can opt to follow dietary programs offered within the app. Leveraging the information provided in user profiles, the app strives to offer personalized recommendations on product consumption, accompanied by healthcare tips that empower users with clear insights into their dietary choices.

To enhance manageability and sustainability, we partitioned the system into subsystems based on their respective functionalities. Through thorough research into various software architectures and scrutiny of comparable approaches, we concluded that a three-layered architecture best meets our requirements. Additionally, we identified the necessity for a client-server architecture to extract specific data from diverse APIs and efficiently store it in databases.

3.2 SUBSYSTEM DECOMPOSITION

At the base, we have the Database Layer, which is made up of Data Access, Entity, and the core databases. The Data Access acts like a middleman, allowing for a buffer between the actual database and the rest of the app. This is key because it means that the database type doesn't limit the rest of the app's design or technology choices. Thanks to Data Access, the other layers can evolve independently, without database compatibility concerns.

Above that layer, there comes the Business Logic Layer which handles all the behind-the-scenes operations that the app requires to operate its functionality. This layer is split into separate subsystems based on the app's different features, reducing dependency across the app's structure. Each subsystem connects to the database to gather the data it needs for its specific operations. For instance, one subsystem might manage user accounts, another might handle image scanning and processing, and yet another could oversee alert notifications. This separation of concerns ensures that each piece of functionality is both self-contained and scalable.

At the top, the User Interface Layer serves as the face of the app. It's where all the user interaction happens, via pages that the app displays. Each subsystem is paired with a controller that bridges the gap between the user and the system, allowing for actions like button clicks and text input. The screens within these subsystems are designed not only to look good but also to connect smoothly with the controllers, providing a user-friendly experience that matches our project goals.

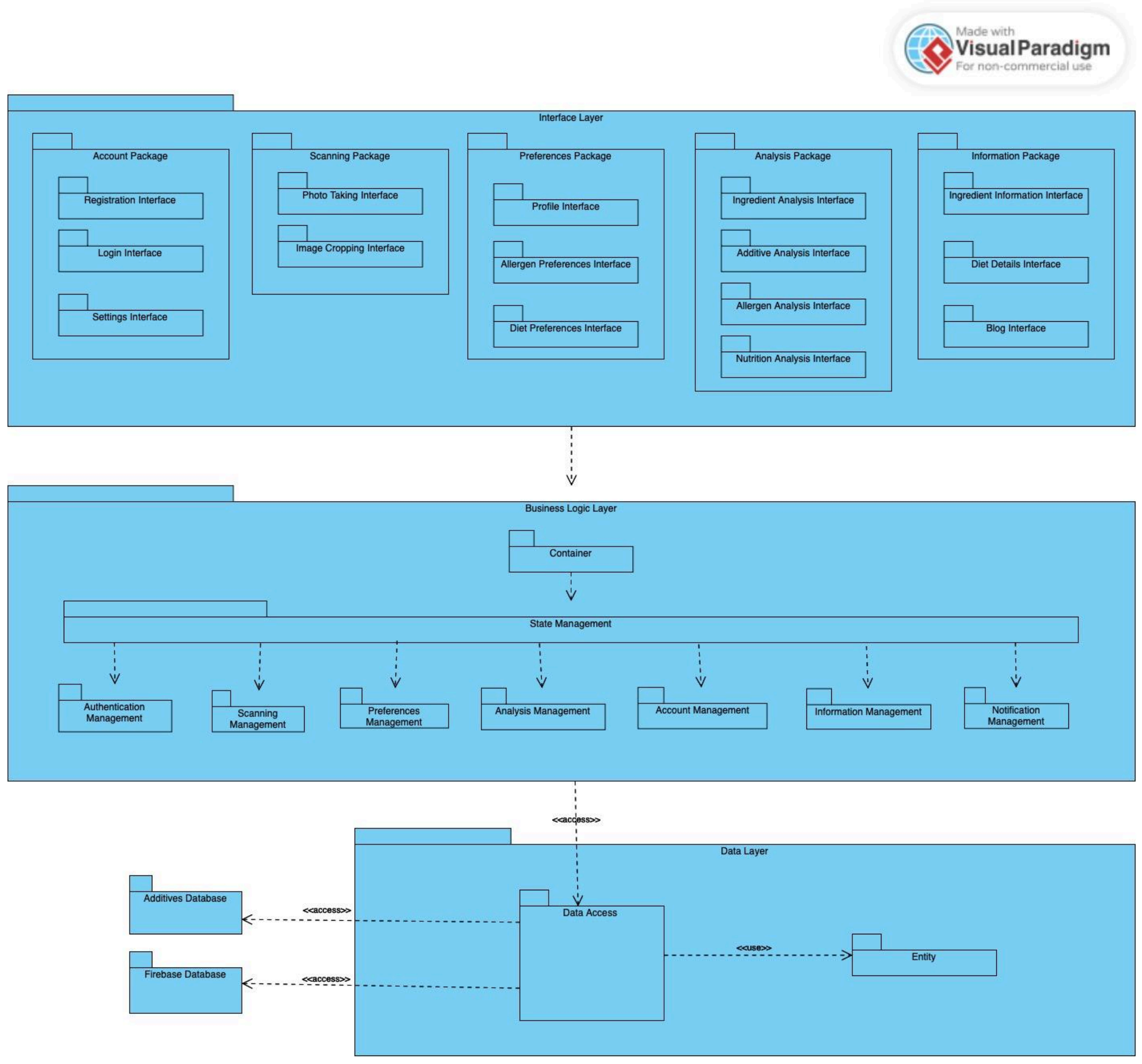

Fig. 3: Subsystem Decomposition Diagram.

3.3 PERSISTENT DATA MANAGEMENT

The persistent data of *Contenta* consists of our Firebase database that contains user information (which would be stored as a User Table with related information), blog post information (Blog Table), as well as ingredient information (Ingredient Table). Data is only manipulated when data has to be changed, and the changes, especially the user information, happen only when users wish their data to be changed. Therefore, if a user decides to delete their account, the removed user data is no longer in our database, and no extra data is held or exchanged with third parties, thus achieving some level of profound security. The data related to the ingredients we keep in our database are updated whenever a new piece of information is informed by the users and experts. Whenever a new ingredient that is not yet in our database is found, a request to add this ingredient is sent through the application and the authorities are able to make any necessary additions or changes. If some data are found to be false (even though these occurrences are rare), such as if an ingredient is later found to be dangerous in a way, then the data is changed accordingly, considering reliable references. Additionally, the blogs that we keep in our database can be edited / changed / removed by the community experts, and their newly written blogs would be submitted to the database and would appear on the blogs page. Thus, since all these data are interactively changed through the application, our community-driven healthcare application would be able to have persistent data that keeps changing with every update regarding the health industry.

3.4 HARDWARE/SOFTWARE MAPPING

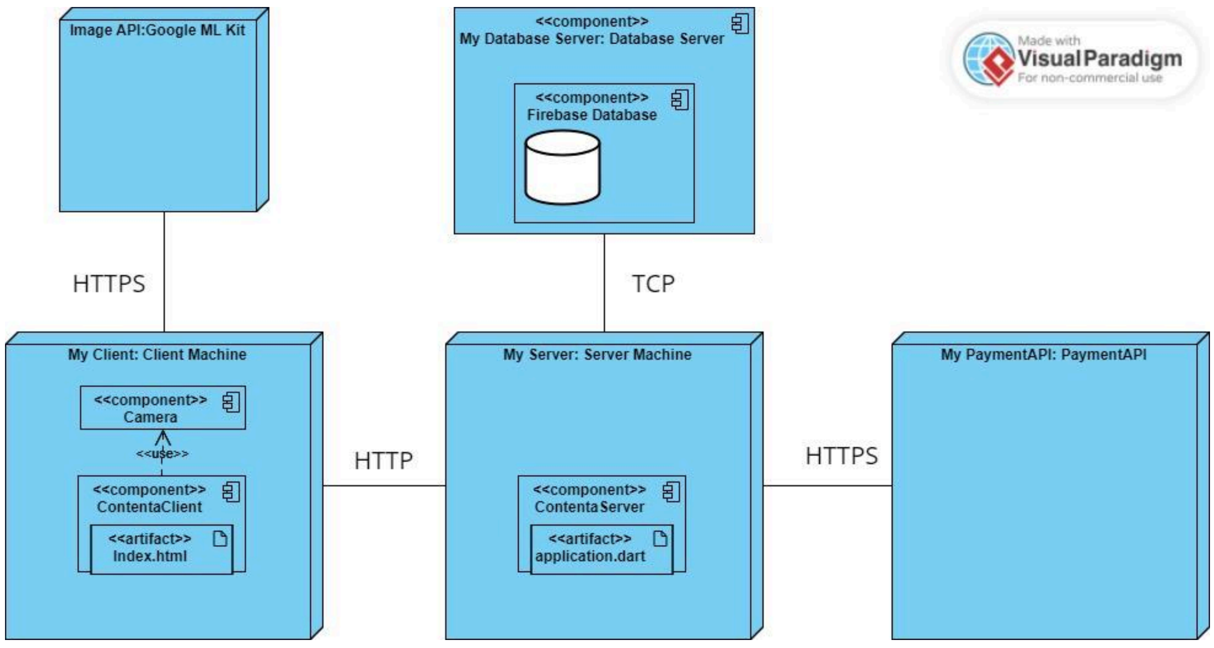

Fig. 3: Deployment Diagram.

The proposed mobile application, named *Contenta,* is a cutting-edge solution designed to simplify and enhance the experience of interpreting and interacting with various images through mobile devices. This application is designed to operate on the client side, utilizing the robust Google ML Kit as its core API for advanced image processing and recognition capabilities. By leveraging the power of Flutter for the client-side development, the application promises a responsive, efficient, and aesthetically pleasing user interface that ensures compatibility across a wide range of mobile devices.

As shown in the deployment diagram in Figure 3, Contenta mobile application sends HTTP requests from the client-side to a server. This server handles logic operations and obtains necessary data from the database, managing data storage and retrieval through the TCP protocol to ensure fast, secure, and reliable communication. Additionally, the server side communicates with a payment API with HTTPS protocol.

The Firebase database is selected for its scalability, real-time data synchronization, and ease of integration with Flutter, making it an ideal choice for managing the extensive data generated by the application. This includes user preferences, image data, and transaction histories, all stored securely and accessible in real time to provide a personalized and efficient user experience.

From a deployment perspective, the Packaged Image Reader Mobile is designed with a clear structure comprising three main components: the client application (developed in Flutter), the server, and the Firebase database. This modular approach ensures that each component can be independently developed, maintained, and scaled according to the application's evolving requirements. The contenta application will run in:

- All devices with Android 10 or higher.

To ensure flawless experience in the application, the client device should have at least:

- Processor: Snapdragon 855
- 5 mbit/second download/upload speed
- 10 megapixel rear camera

To ensure the highest level of performance and reliability, the server infrastructure is recommended to meet the following minimum hardware specifications:

- Processor: Intel Core i9-13900 H
- Memory: 32GB RAM
- Operating System Disk: SSD DRİVE: 128GB storage capacity
- Connection: 800 megabit/second internet

3.5 ACCESS CONTROL SECURITY

Given the sensitive nature of the data collected by *Contenta*, including personal information such as weight, height, allergens, and diet preferences, robust access control security measures are imperative to safeguard user privacy and adhere to regulatory requirements such as the General Data Protection Regulation (GDPR) of the EU and the Turkish Personal Data Protection Law (KVKK). To ensure compliance and protect user data from unauthorized access, *Contenta* implements a multi-layered access control framework encompassing various security mechanisms.

First and foremost, *Contenta* employs role-based access control (RBAC) to manage user permissions and restrict access to sensitive data based on predefined roles and responsibilities. Each user is assigned a specific role within the system, such as regular user, administrator, or data manager, with corresponding access privileges tailored to their role. This granular control over user access helps mitigate the risk of unauthorized data exposure and ensures that only authorized personnel can view or manipulate sensitive information.

Furthermore, *Contenta* implements strong authentication mechanisms to verify the identity of users accessing the application. This includes the use of password-based authentication augmented with two-factor authentication (2FA) for an added layer of security. By requiring users to provide multiple forms of authentication, Contenta reduces the risk of unauthorized access due to stolen or compromised credentials, enhancing the overall security posture of the application.

In addition to RBAC and authentication mechanisms, Contenta employs encryption techniques to secure data both in transit and at rest. All communications between the application and its servers are encrypted using industry-standard cryptographic protocols such as Transport Layer Security (TLS) to prevent eavesdropping and data interception. Similarly, sensitive user data stored in the application's databases is encrypted using strong encryption algorithms to protect against unauthorized access in the event of a data breach. By incorporating these access control security measures, *Contenta* demonstrates its commitment to safeguarding user privacy and ensuring compliance with regulatory standards.

4.0 SUBSYSTEM SERVICES

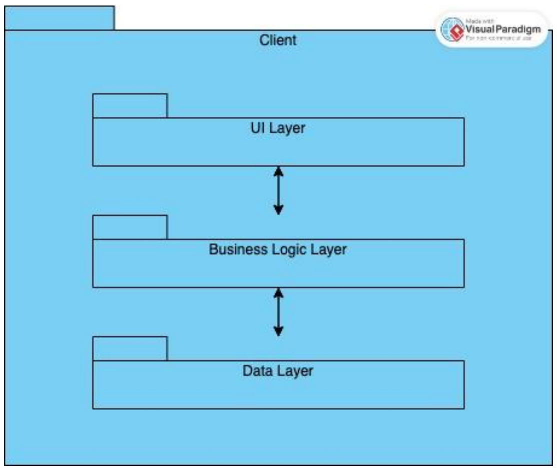

Fig. 4: Layered Decomposition Diagram.

4.1 UI LAYER

The User Interface layer is composed of 5 main packages that have various components inside. The Account Package contains all the UI components related to our user authentication and account management screens. Interfaces for user registration, user login, user related profile settings are all under this package. Another main package is Scanning Package. This package contains the interfaces responsible for capturing and processing images that are taken or uploaded by the users. The interfaces where the users engage with the photo taking and afterwards cropping to refine the visuals further are under this package. The Preferences Package is to gather and manage user preferences. This package contains interfaces that enable users to customize their profiles, specify personal allergen preferences and also set their dietary preferences. The Information Package contains interfaces that provide more educational and informative content such as details about foods, diets. This package also contains the blog interfaces.

Class	Description
Registration Interface	An interface which allows users to register.
Login Interface	An interface which allows users to login.
Setting Interface	An interface which allows users to view/change the settings.
Photo taking interface	An interface which allows users to take a photo.
Image Cropping Interface	An interface which allows users to crop an image which has just been taken.
Profile Interface	A screen where users can see their profile/information.
Allergen Preferences Interface	A screen where users can set their allergies.
Diet Preferences Interface	A screen where users can set their diet preferences.
Ingredient Analysis Interface	An interface which informs users about a product’s ingredients.
Additive Analysis Interface	An interface which informs users about a product’s additives.
Allergen Analysis Interface	An interface which informs users about ingredients/additives which may potentially trigger their allergies.
Nutrition Analysis Interface	An interface which informs users about a product’s nutri-score.

Ingredient Information Interface	A screen where users can see information about ingredients.
Diet Details Interface	A screen where users can see information about different diets.
Blog Interface	A screen where users can see/select all blogs.

4.2 BUSINESS LOGIC LAYER

The Business Logic Layer talks to the UI layer. It makes sure that the app responds well when users do something. Inside this layer, we have a 3 layered structure. Account Management, Authentication Management, Scanning Management, Preferences Management, Analysis Management, Information Management, Notification Management are the bottom components of our 3 layered structure. Each of these components are responsible for handling a different job in the flow of the app. Account Management handles everything to do with user accounts. Authentication Management checks the correctness of user login. Scanning Management works with the pictures users provide. Preferences Management remembers users profile, dietary and allergen preferences. Analysis Management handles the analysis after image upload. Information Management keeps all the detailed food and diet information. It also handles the blog part of the application. Notification Management sends users messages and other alerts. All these parts are held together under the State Management. This makes sure that the app is always showing the right thing for what the user is doing.

Class	Description
Container	A class that gives access to the user interface widgets.
State Management	Set of classes that handles the requests of the user interface layer. Moreover, It handles the interactions of the user interface and performs the application logic, using other controllers in case of need and returns the relevant state/data back to the User Interface Layer.
Account Management	A class which performs application logic of user’s account
Authentication Manager	A class which handles user’s authentication/authorization
Scanning Management	A class which performs scanning logic.

Preferences Management	A class which handles diet/allergy preferences logic.
Analysis Management	A class which performs ingredient and nutri-score analysis.
Information Management	A class which handles user's information logic.
Notification Management	A class which handles notification logic

4.3 DATA LAYER

The Data Layer is where our app stores and gets all its information that has been requested. It is made up of two databases. First one is for the additives and the other one is for everything else. The latter is stored in Firebase while the first is an external database which we reach via their API calls. When the app needs to use this information, there is a Data Access part that knows how to find and retrieve it. The Entity part is like a box in this layer that keeps the details the app needs to work properly.

Component	Description
Data Access	A class which ensures communication between databases and business logic layers.
Additive Database	A database which contains information about additives and ingredients.
Firebase Database	A database which contains all relevant information other than information about additives and ingredients.
Entity	The set of objects required for handling analysis operations through the database.

5.0 TEST CASES

This section demonstrates some crucial test cases to be performed after the implementation is completed. The success of the test cases will be stated in the final report.

5.1 FUNCTIONAL TEST CASES

Test ID: TF01

Type: Functional

Title: Registration and Login

Covered Requirement(s):

- *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 their password whenever they forget it.

Entry Criteria: The application is up and running and the user is not logged in.

Test Steps	Expected Outcome
User clicks the register button.	Registration form is displayed.
User enters the first name, last name, email, password, weight, height, diet preferences, and allergies.	Confirmation mail is sent.
User confirms the confirmation mail.	User gets registered and the main page is displayed.
User presses the sign off button from the sidebar.	User is signed off and the login page is displayed.
User enters their email and password.	User is logged in and the main page is displayed.
User presses the sign off button from the sidebar.	User is signed off and the login page is displayed.
User clicks the “forgot password” button.	A temporary password is sent to the user's mailbox.
User logs in by entering their email and the temporary password.	User is logged in and the main page is displayed.

Test ID: TF02

Type: Functional

Title: Change Password and Delete Account

Covered Requirement(s):

- *Delete Account:* A user can delete their account. Their personal information will be deleted in this case.
- *Change Password:* When logged in, a user can change their password by using the old password.

Entry Criteria: The application is up and running and the user is logged in.

Test Steps	Expected Outcome
User clicks change password button.	The user is redirected to the change password

	page which demands current password and new password.
User types old password and new password.	If the current password is wrong, an error which informs the user will appear. If current password is right, the password will be changed
User clicks delete account button.	A page which demands the current password appears.
User types his/her current password and clicks delete account button	If the password is right, the user will be asked "are you sure" If the current password is wrong, an error which informs the user will appear.
User clicks "I am sure" button	User's account along with all related information will be deleted.

Test ID: TF03

Type: Functional

Title: Premium Membership

Covered Requirement(s):

- *Become Premium Member:* Users can become a premium member by subscribing.
- *Terminate Subscription:* A premium user can stop his/her subscription.

Entry Criteria: User is logged in and is not a premium member.

Test Steps	Expected Outcome
User clicks the "become a premium member" button.	A page which informs user about the advantages of premium members will appear.
User clicks the continue button.	User is redirected to the payment page.
User types his/her credit card information.	Payment API processes the information and returns the result. If payment is successful, the membership is upgraded, if not, an error which informs the user appears.
User clicks terminate subscription button.	User is redirected to "terminate subscription" page which asks the user "are you sure"
User clicks yes.	The subscription is terminated and no more payment will be taken from the user.

Test ID: TF04

Type: Functional

Title: Add Allergy / Unwanted Ingredient

Covered Requirement(s):

- *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.
- *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.

Entry Criteria: User is logged in.

Test Steps	Expected Outcome
User clicks the add allergy button.	List of allergies will appear.
User chooses an allergy and clicks the done button.	The allergy is added.
User scans an ingredient which may potentially trigger the user's allergy/allergies.	User is warned.
User clicks the add unwanted ingredient button.	List of ingredients with a search bar appears.
User chooses an ingredient and clicks the done button.	Unwanted ingredient is added.
User scans an ingredient which has an unwanted ingredient.	User is warned.

Test ID: TF05

Type: Functional

Title: Scan Ingredients

Covered Requirement(s):

- *Scan Ingredients:* A user can scan food’s ingredients and then select the desired language area.. 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.

Entry Criteria: User is logged in.

Test Steps	Expected Outcome
User clicks the scan ingredients button.	The camera is open.
User takes a photo of the package’s ingredient part.	The image appears with a cropping option.
User crops the photo and gets rid of irrelevant	The application shows the ingredients safety

parts and clicks the done button.	level and provides all other relevant information.

Test ID: TF06

Type: Functional

Title: Scan Nutrient Facts Table

Covered Requirement(s):

- *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.

Entry Criteria: User is logged in and user has premium subscription.

Test Steps	Expected Outcome
Premium user clicks scan nutrient facts button.	The camera is open.
User takes a photo of package's nutrient information part	The image appears with a cropping option.
User crops the photo and gets rid of irrelevant parts and clicks the done button.	The application shows the Nutri-score.

Test ID: TF07

Type: Functional

Title: Promote / Demote User

Covered Requirement(s):

- *Promote User*: An admin can promote a user to an expert.
- *Demote User*: An admin can demote an expert to a user.

Entry Criteria: The user is logged in and should have admin authorization.

Test Steps	Expected Outcome
Admin clicks the view candidate experts button.	The user list which shows the users who applied to become an expert appears.
Admin selects a user and clicks the approve button.	User is promoted to an expert.
Admin clicks the view experts button.	List of experts appears.
Admin selects an expert and clicks the demote button.	Expert is demoted to user.

Test ID: TF08

Type: Functional

Title: Read Blog

Covered Requirement(s):

- *Read Blog*: Users can read blogs about specific ingredient(s) or nutrient facts which are written by experts.

Entry Criteria: The user is logged in.

Test Steps	Expected Outcome
User clicks the view blogs button.	List of blogs with a search bar appears.
User selects a blog.	The content of the blog appears.
User clicks the back button while reading a blog.	List of blogs with a search bar appears.

Test ID: TF09

Type: Functional

Title: Add / Remove Bookmark

Covered Requirement(s):

- *Bookmark:* A user can bookmark a blog article.
- *Remove Bookmark*: A user can remove the bookmark of a blog article.

Entry Criteria: User is logged in.

Test Steps	Expected Outcome
User clicks the view blogs button.	List of blogs with a search bar appears.
User selects a blog.	The content of the blog appears. At the top of the page, the bookmark icon appears. If the blog is already in the bookmarks, inside of this button will be colored, else white.
User clicks the bookmark button.	If the blog is in bookmarks, it is removed. If it is not in the bookmarks, it is added to bookmarks.

Test ID: TF10

Type: Functional

Title: Write Blog

Covered Requirement(s):

- *Write Blog*: Experts who are confirmed by admin can write blog articles about some ingredients or nutrition facts.

Entry Criteria: User is logged in and has expert authorization.

Test Steps	Expected Outcome
An expert clicks the write blog button.	A page which demands a title and text appears.
Expert types title and text and clicks the submit button.	The blog is submitted and becomes available to all users.

5.2 NON-FUNCTIONAL TEST CASES

Test ID: TN01

Type: Non-Functional

Title: Performance Testing

Covered Requirement(s):

1. *Account:* Simulate the Sign up, Log in, and Log out actions with only one as well as multiple users in the app.
2. *Scan Ingredients:* Simulate scanning on different ingredients. Simulate scanning ingredients with only one, as well as multiple users using the application, recording the times at which their results are shown on the screen.
3. *Blog Posts:* Simulate adding a blog to the system on one, as well as multiple accounts, and compare the amount of time it takes for them to show up in the blogs tab.

Entry Criteria: The application is running and the user is an expert type.

Test Steps	Expected Outcome
Sign up, log in, log out with one user, and note its time. Sign up, log in, log out with multiple users at the same time, and compare the amount of time it took.	The time taken should be close enough that the case with multiple people should be less than two times the single-person entry time.
Scan ingredients of different products and record the time it takes for various products.	The time it takes should be close enough (that the time it takes for one product should be less than two times of time it took for any other product scan).
Scan an ingredient on a single device and later scan it on multiple devices and note the time it took on both trials.	The time it takes should be relatively similar, similar to the previously expected outcomes.
Add a blog on one account, later add blogs on	The time it takes should be relatively similar.

multiple accounts, and note the time both processes take for all blog posts to appear on the blogs page.	

Test ID: TN02

Type: Non-Functional

Title: User Interface Adaptability

Covered Requirement(s):

1. *Account:* Simulate sign up, log in, sign out, delete account actions and verify that all buttons, and fields are clickable and all text fields are readable.
2. *Scanning:* Simulate scanning an ingredient action, going to different pages, and pressing all buttons.
3. *Blogs*: Simulate sending a blog, editing, and reading blog functionality.

Entry Criteria: The application is running.

Test Steps	Expected Outcome
The user Signs up, Logs in, Logs out, and deletes their account.	the UI is visible, interactable, and functional.
A customer user who is logged in, scans a product, views its contents, and presses the links or buttons.	the UI is visible, interactable, and functional.
An expert user who is logged in writes a blog, sends, and edits a blog. Then read multiple blogs to take a general look at the blogs page.	the UI is visible, interactable, and functional.

Test ID: TN03

Type: Non-Functional

Title: Data Access Security

Covered Requirement(s):

1. *Sign up:* Simulate signing up with different and same-named accounts, and simulate different password situations (acceptable, not acceptable).
2. *Log in:* Simulate logging in to multiple accounts, entering the correct/wrong password and correct/wrong username.

Entry Criteria: The application is running.

Test Steps	Expected Outcome

Sign up to account A, then try signing up with A’s credentials, then try signing up with A’s username but a different password, then sign up to B’s account.	The UI should give appropriate warnings (and not sign up) on the 2nd and 3rd cases. The application should not allow these cases. The other cases should sign users up.
Log in to account A, log out, then try logging in to account A with a false password, try logging in with a different username and right password, and log in with account B.	The UI should give appropriate warnings (and not log in) on the false password and false username cases. The application should not allow these cases. The other cases should allow log in.

6.0 OTHER ANALYSIS ELEMENTS

6.1 CONSIDERATION OF VARIOUS FACTORS IN ENGINEERING DESIGN

- **Public Health (10/10):** Contenta's main objective revolves around advancing public health and fostering well-being through its innovative content consumption approach. The platform endeavors to empower users with the means to make informed and health-conscious decisions regarding their diets and lifestyles, aiming to make a substantial contribution to overall public health. Carefully crafted features within the application, such as ingredient scanning and health insights, are geared towards positively influencing users' health outcomes, earning an impressive maximum rating of 10 in this particular category.

- **Global and Cultural Factors (2/10):** *Contenta*'s influence on global and cultural aspects remains somewhat constrained. Although the application holds the potential to extend beyond geographical boundaries, its primary focus lies in health-conscious consumption, a feature that may not universally apply. Divergent cultural nuances regarding dietary preferences and health practices may lead to variations, contributing to a lower rating of 2 in the realm of global and cultural factors.

- **Social Factors (3/10):** *Contenta*, not positioned as a social media platform, exerts a moderate influence on social factors. Its primary emphasis is on individual health rather than cultivating extensive social interactions. The application's design places a premium on user privacy and tailored experiences, constraining its direct impact on social dynamics. Consequently, it garners a rating of 3 in the social factors category.

- **Environmental Factors (4/10):** Although *Contenta* does not explicitly tackle environmental concerns, it indirectly champions environmental goals by promoting the adoption of plant-based diets among users. The advocacy for vegan or vegetarian lifestyles has the potential to contribute to a diminished environmental footprint, aligning with overarching sustainability objectives. Consequently, *Contenta* is assigned a moderate rating of 4 in the environmental factors category.

- **Economic Factors (6/10):** *Contenta* exhibits substantial potential to play a pivotal role in bolstering the local economy, primarily through its advertising and subscription models. The platform's support for local businesses and the prospective creation of employment opportunities align seamlessly with economic factors. Contenta's capacity to generate revenue and contribute to local economies warrants a commendable rating of 6 in the economic factors category.

To encapsulate, *Contenta*'s engineering design is intricately shaped by its fundamental goal of advancing public health. Although its reach on global and cultural aspects is confined, it exerts moderate influence on social and environmental factors, while demonstrating significant impact on economic considerations. The succinct table below offers a condensed overview of these considerations and their corresponding 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

6.2 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 align with data protection regulations. Given the involvement of scanning and processing user-generated content, upholding intellectual property rights, adhering to content distribution policies, and safeguarding user data privacy are imperative. Rigorous adherence to legal frameworks establishes trust with users and safeguards the application's reputation.

- **Marketability and User Engagement:** A critical challenge lies in ensuring *Contenta*'s marketability, which hinges on addressing user adoption and retention. Encouraging users to download and retain the app requires strategic implementation of engaging features, personalized content recommendations, and regular updates to enhance user retention. Exploring the marketability of a premium version with extra features can create additional revenue streams and bolster user commitment.

- **Maintainability and Database Management:** *Contenta*'s maintainability is paramount, especially considering potential database growth due to storing users' scanning history. An efficient database management system is essential to handle the expanding volume of data seamlessly. Implementing regular maintenance procedures and optimization strategies will sustain the application's smooth operation, mitigating performance bottlenecks and enhancing the overall user experience.

6.3 STANDARDS

The development of the *Contenta* mobile application strictly adheres to globally recognized engineering standards, ensuring a robust software life cycle driven by uncompromising quality. This endeavor aligns seamlessly with the ISO/IEC/IEEE International Standard for Systems and Software Engineering – Software Life Cycle Processes, placing emphasis on the need for systematic and well-defined processes throughout development, testing, and maintenance phases.

When it comes to crafting software requirements, we rely on the ISO/IEEE Recommended Practice for Software Requirements Specifications as a foundational guide. This standard not only guarantees the clarity, completeness, and consistency of software requirements documentation but also promotes effective communication among stakeholders, fostering a comprehensive understanding of project specifications.

In the domain of modeling and design, our project leverages the widely accepted Unified Modeling Language (UML) 2.5.1. This standard is instrumental in visualizing, specifying, constructing, and documenting software artifacts, providing the project team with a universal language for clear communication and a shared understanding of the system architecture.

Additionally, the structure of the analysis and requirements report, including this one, aligns with the rigorous 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 fidelity to IEEE guidelines, a departure from the traditional two-column structure has been implemented to enhance readability and accessibility for a broader audience.

7.0 TEAMWORK DETAILS

7.1 CONTRIBUTING AND FUNCTIONING EFFECTIVELY ON THE TEAM

In our project, we prioritized equal workload distribution while keeping our academic obligations in mind. Our team's understanding of the project's requirements allowed us to support each other. Scheduled and regular team meetings serve as a platform for us to do our task allocation and strategize future steps to come. After each member completes their portion of work, we review each other's contributions to ensure quality and being on the same page. To ensure a balanced contribution, we have responsibilities aligning with each team member's strengths, interests and experience. API research and integration are collaborative efforts, involving Mert, Ömer, Barış, Oğuz, Alperen. Ömer is specifically tasked with designing and implementing the AWS architecture. Mert and Alperen are focused on the front-end development by creating a user-friendly interface. The server-side development is handled together by Barış, Oğuz and Ömer. In addition, the responsibility of compiling reports is evenly distributed among all team members to be able to make sure that each person contributes to documenting our project's progress and findings. This structure does not only allow us to have a balanced workload but it also enables us to have a supportive project development environment where each and every team member can seek and offer help. It also allows us to be flexible while maintaining our individual schedules as well as maintaining collective project progress.

7.2 HELPING CREATING A COLLABORATIVE & INCLUSIVE ENVIRONMENT

In our team, we all knew each other before starting the project. This made working together easier. We like to meet in person because we think that it helps us communicate better and work together more effectively. Even when we don't have specific tasks to do, we keep in touch to make sure everyone is included and up to date.

We believe in helping each other out, not just with our own tasks but with anything the team needs. We always share updates related to our ongoing tasks to each other, whether it's good or bad, so everyone knows what's going on and can help out if needed. Also this keeps all of us aligned. In addition, with this method, we are always informed and ready to work together to solve any problems.

Even though we're friends, we make sure everyone's voice is heard and included in our project. This way, we create a friendly and supportive environment where everyone feels part of the team and motivated to contribute.

7.3 TAKING LEAD ROLE AND SHARING LEADERSHIP ON THE TEAM

In our team, we embrace a flexible approach to leadership. With this, we can leverage each member's strengths and expertise. We operate without a designated single leader. Instead, we prefer to make important decisions together during our regular team meetings. This ensures that every member's voice is heard and valued.

Our task allocation is strategically distributed based on our team members' individual skills and interests. For instance, Ömer, who has knowledge in cloud computing and AWS services, naturally took the lead in that domain. His expertise enables us to utilize the best practices and good approaches in our cloud structuring. Another example, in the front-end development part, Alperen's experience has made him the leader. His ability to design user friendly interfaces has significantly enhanced the user experience of our project, making it more engaging and accessible to users.
For the documentation and organization part, Barış stands out. He has assumed responsibility for compiling, organizing, and maintaining our project documentation. His strengths in this area ensure that our documents are comprehensive, well-organized and easy to read.

Our approach to leadership is dynamic, adjusting as the project evolves. This flexibility has not only allowed us to get better in our areas of expertise but has also contributed to a learning environment where everyone has the chance to take on new challenges and expand their skill sets as a leader. By sharing leadership based on current needs and individual strengths, we have created an efficient team dynamic.

8.0 CONCLUSION

In conclusion, the Detailed Design Report provides a comprehensive roadmap for the development of the system, addressing its current architecture, proposed enhancements, and critical design considerations.

Beginning with an overview of the system's purpose and design goals, the report delves into specific aspects such as user-friendliness, maintainability, scalability, performance, and privacy. This sets the stage for the exploration of the current system architecture and the proposed software architecture, including subsystem decomposition, data management, hardware/software mapping, and access control security.

The delineation of subsystem services across UI, business logic, and data layers further elucidates the system's structure and functionality. Test cases, both functional and non-functional, are meticulously outlined to ensure comprehensive testing and validation.

Additionally, the report considers various factors in engineering design, constraints, and standards, emphasizing the need for a holistic approach to system development. Teamwork details underscore the importance of collaboration, inclusivity, and shared leadership in achieving project success.

In summary, the Detailed Design Report serves as a valuable guide for the development team, providing insights, specifications, and guidelines necessary for the successful implementation of the system. Adherence to the outlined recommendations and considerations will facilitate the creation of a robust, efficient, and user-centric system that meets the desired objectives and standards.

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: Mar. 13, 2024. 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.