



Bilkent University

---

Department of Computer Engineering

# High Level Design Report

*ShopCart*

Ahmet Kaan Uğuralp, Ahmet Işık, Furkan Ahi, Mehmet Yaylacı, Ravan Aliyev

**Supervisor:** Asst. Prof. Dr. A. Ercüment Çiçek

**Jury Members:** Asst. Prof. Dr. A. Ercüment Çiçek, Asst. Prof. Dr. Shervin Arashloo, Asst. Prof. Dr. Hamdi Dibeklioğlu

High Level Design Report Dec 24, 2021

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

# Table of Contents

<b>1 Introduction</b>	<b>4</b>
1.1 Purpose of the system	4
1.2 Design Goals	5
1.2.1 Security	5
1.2.2 Usability	5
1.2.3 Performance	5
1.2.4 Scalability	5
1.2.5 Reliability	5
1.3 Definitions, acronyms and abbreviations:	6
1.4 Overview	6
<b>2 Current System</b>	<b>7</b>
2.1 YemekSepeti	7
2.2 Banabi	7
2.3 Migros: Sanal Market	7
2.4 MyFitnessPal	8
2.5 LifeSum	8
2.6 ShopWell	8
<b>3 Proposed software architecture</b>	<b>9</b>
3.1 Overview	9
3.2 System Decomposition	9
3.3 Hardware/software mapping	10
3.4 Persistent data management	11
3.5 Access control and security	11
3.6 Global software control	11
3.7 Boundary conditions	12
3.7.1 Initialization	12

3.7.2 Termination	12
3.7.3 Failure	12
<b>4 Subsystem services</b>	<b>13</b>
4.1 Client	13
4.1.1 View	13
Login Page	13
User Page	13
Register Page	13
Products Page	13
Barcode Page	13
4.1.2 Controller Subsystem	14
Barcode Reader	14
Authentication	14
Products Manager	14
4.2 Server	14
4.2.1 Store Manager Subsystem	14
4.2.2 Home Manager Subsystem	14
4.2.3 Auth Manager Subsystem	14
4.2.4 Products Manager Subsystem	14
<b>5 Consideration of Various Factors in Engineering Design</b>	<b>15</b>
<b>6 Teamwork Details</b>	<b>16</b>
6.1 Contributing and functioning effectively on the team	16
6.2 Helping creating a collaborative and inclusive environment	16
6.3 Taking lead role and sharing leadership on the team	16
<b>7 Glossary</b>	<b>17</b>
<b>References</b>	<b>18</b>

# 1 Introduction

People have been going shopping in order to meet their needs. We all need to purchase food and beverages to survive. Storing the purchased food in refrigerators, cooking and ordering meals are all common activities. When ingredients are used for these purposes they run out eventually. People make the shopping list when the refrigerator is nearly empty. However, they may forget some of the products and this causes them to go shopping again. Since food prices keep rising, smart shopping becomes more challenging. Smart shopping means knowing what to buy and when. [1] Keypoint here is knowing the basics before going shopping, like what you need, the amount of each food needed and where you can find these products with the lowest price.

ShopCart is a mobile application which helps users to keep track of the products that run out and purchase reminders. The information about these products includes their monthly consumption frequency, the number of the days that it has been run out, and so on. So this application helps consumers to be a smart shopper by showing when each product is out of stock, amount of the needed product, which market offers the lowest price for each product and calculating the total spending on food. Before throwing away the depleted product, users scan its barcode, enter its name manually or scan a photo in order to create a shopping list. After the product is added to the shopping list, its numeric data depletion time and other information will be preserved as a result of these methods. Users can create a “community” on the app if they live together. By the help of this feature users living together can create a shopping list of the community and each member can add and remove products from this list. Users can also edit the list after purchasing it to choose not to purchase it again. The goal of this app is to make people’s lives easier by improving their smart shopping practices. Our shopping practices have big implications for our health, our society and the environment and this app helps to raise living standards by improving consumption practices. [2]

In this paper, we provide the high level design of the paper. Firstly, the purpose of the system will be discussed and design goals will be described, later on the current systems will be examined. After that, system architecture of the ShopCart application will be described and subsystem decomposition and deployment diagrams and their explanations will be given. Afterwards, different factors that can affect our system will be discussed. Lastly, teamwork details of our team will be described.

## 1.1 Purpose of the system

ShopCart is a mobile application that helps to improve users’ shopping practices. This application makes smart shopping much easier by helping keep the data of which products users consume, amount of the product, when these products run out. In addition to these features, ShopCart also shows the price list of each product on the available markets nearby. Our system uses image analysis and machine learning algorithms to satisfy these goals in order to offer features that we feel other applications miss.

## **1.2 Design Goals**

### **1.2.1 Security**

- Users personal information like email, username, and password and their shopping behaviors will not be shared with any 3rd party companies. This data should be kept according to General Data Protection. [3]

### **1.2.2 Usability**

- Application will be available for both operating systems: iOS and Android
- The number of components on any screen should not exceed seven in order not to confuse the user.
- Labels of tappable components, such as button names and screen labels, should be self-explanatory.
- It should have an interface that users can easily use without the need for an extra learning process. Potential users should be able to use it with their current application usage information.

### **1.2.3 Performance**

- Screen loading and response times from the server side should not take more than one second.
- Creation of the community shopping list should not take more than 3 seconds.

### **1.2.4 Scalability**

- Application must be scalable in order to support large amount of users and communities with different sizes

### **1.2.5 Reliability**

Users can make some mistakes while using mobile applications:

- To avoid these, confirmation pop-ups should appear on many transition screens and confirmation processes.
- Operations other than the user's request should not occur, except where otherwise permitted.

## 1.3 Definitions, acronyms and abbreviations:

- **TA:** Teaching Assistant
- **UI:** User Interface
- **API:** Application Programming Interface
- **Server:** Houses the database and is the backend of the system. All logical operations on the data are done here.
- **DB:** Database

## 1.4 Overview

ShopCart is an application that allows you to track grocery shopping and order habits. The main purpose of the application is to make daily grocery shopping easier and more effective and to make it easier for people to follow their spending categorically. While doing all this, it also aims to provide a potential financial benefit by considering the economy of the home and the individual.

What distinguishes ShopCart from other grocery shopping applications is that it is more user-oriented and suitable for common use compared to its counterparts. Many shopping applications give results according to user searches and list them according to the results. In ShopCart, on the other hand, in addition to all these, the user can use the application both as the needs of his/her household arise, and when s/he comes to his/her mind and enters own needs into the application.

ShopCart consists of two main elements. These main elements also have many subheadings. These will be explained as appropriate. The first of the main elements is that the user can add the product in his/her household to his/her shopping cart as soon as he/she uses it or whenever he/she wants. The diversity in “adding to shopping list” feature is the most important feature that distinguishes ShopCart from other traditional applications. Barcode/QR code reading, product detection from product package, voice product recognition system are examples of varieties in the product adding phase. The striking point of this main element is that the product addition feature is not ordinary. Contrarily, the products are designed to reach the most suitable basket amount by comparing the prices of many companies, financially, for the benefit of the user. The second main element is the graphics of consumer habits that change depending on what the users do and consumption. By looking at these graphs, users will have the opportunity to examine their individual and household needs and consumption habits and will be able to shop accordingly. In addition, users will be able to learn how much profit they made on average at the end of the transactions from this part of the application.

## 2 Current System

Even though many apps are helping customers with ordering food and buying groceries, the apps in the Turkish market do not have the functionality of diet management. We will introduce some apps from the Turkish market and some apps from foreign markets. The apps that have the functionality of diet management are not dominant in the market as the food ordering apps are relatively more dominant and harder to compete with.

### 2.1 YemekSepeti

- YemekSepeti is an app for ordering food.
- The company mostly operates in Turkey and has a dominant position in the market.
- The app has an integrated system named Banabi that helps users to order groceries.
- Does not have diet management as a functionality

### 2.2 Banabi

- Lets users order groceries online.
- This app has created a campaign, the first order is 20 liras cheaper.
- Users can save their credit cards.
- Users can see the number of groceries they have ordered, but there is not any statistical information for the user to see.

### 2.3 Migros: Sanal Market

- Lets users order groceries online.
- The service is fast, if the user orders early the order will be delivered in a day.
- The order process can be checked on a map.
- This app lacks diet-management functionality

## 2.4 MyFitnessPal

- This is an app designed for diet-management
- This app has a large food database and claims to have information about 11 million different foods.
- The food information consists of calorie count, nutrition information, and serving size.
- The users can track their dietary habits and calculate various statistics about their foods' nutritious content.
- The app lets users add certain foods.
- However, the app does not have the functionality of suggesting what the users should buy for groceries.

## 2.5 LifeSum

- This app is designed for people who want to lose weight and track their progress.
- The app has the functionality of creating dietary programs specially designed for each user.
- The app has barcode scanning functionality which helps users to get nutritional information about the products they buy.
- The app also tracks the water intake of a user and makes suggestions about water drinking habits.

## 2.6 ShopWell

- This app is designed for diet-management
- The app has the functionality of creating dietary programs specially designed for each user.
- The app also has a list of food allergies and certain foods that have ingredients that are allergen for some users. Users can select these allergies and get a warning when they try to find products that are allergic to them.
- The app does not have the functionality of grocery management.



## 3 Proposed software architecture

### 3.1 Overview

*ShopCart* will consist of two main layers, the first one is the client side where the users will be able to launch the web application in order to join the session. The other one is the server side which handles the comparisons, barcode scanning and data retrieval from the database. The server will be configured using Django. The code will run on Android and IOS phones and the server will be run on a local computer. Last but not least, the database will be on SQLite and barcode-scanning will be done using “expo-barcode-scanner” library. The client and server layers will communicate with each other through sockets and HTTP requests, handled by various controllers.

### 3.2 System Decomposition

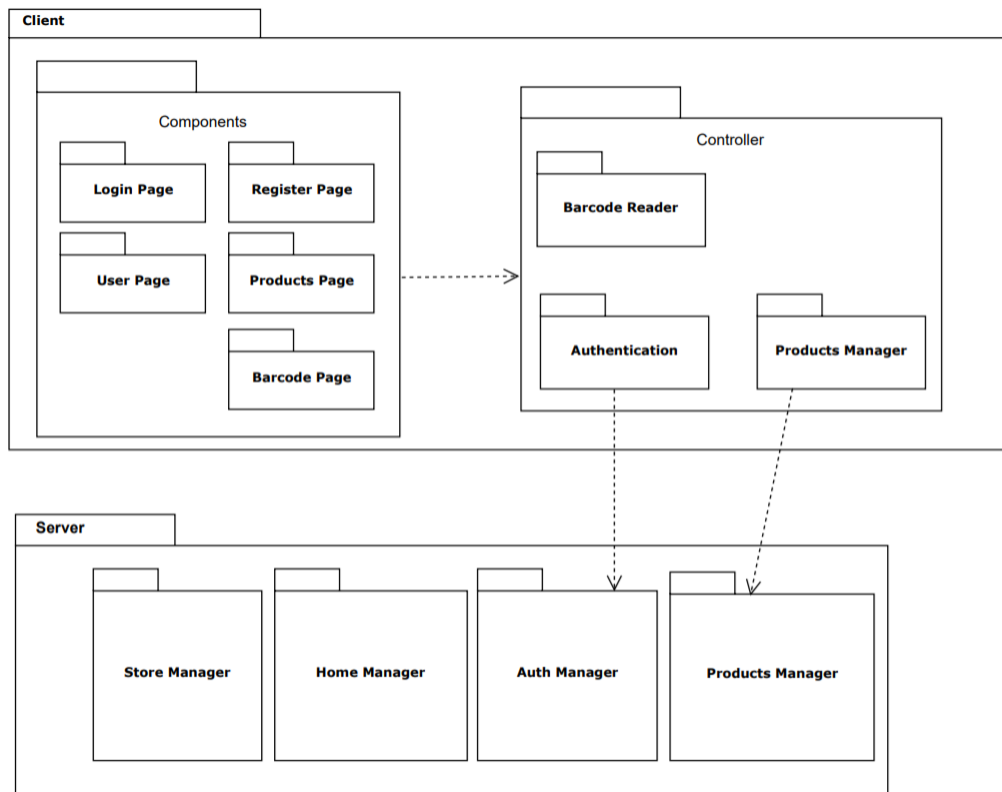


Figure 1: Subsystem Decomposition Diagram

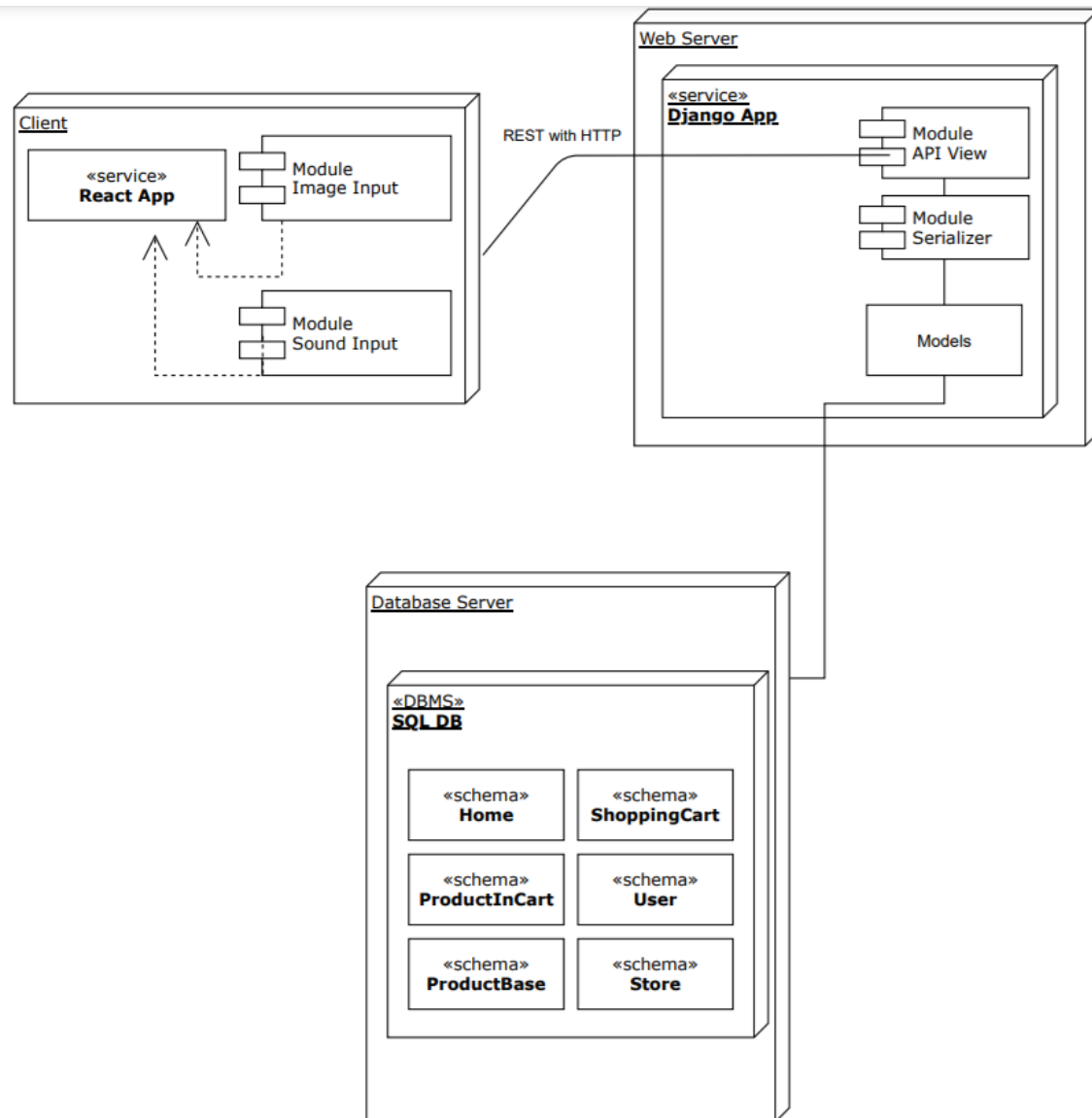


Figure 2: Deployment Diagram

### 3.3 Hardware/software mapping

ShopCart will be a mobile application developed with React Native. The user will not need an extra device other than a mobile phone to use the application. The application, which will be multiplatform, will support both Android and iOS platforms. The application will be able to use the phone's camera so that it will be able to scan images from the environment. Since the data coming from the camera will be processed either in the form of reading barcode / QR Code or scanning the product image, it is planned to send the data obtained to the necessary servers and receive their answers. In the application, mobile internet networks will be used and data exchange will be made over the internet. Thanks to the data we obtain through various APIs, comparisons to the products

determined by the user will be made on the backend of the application, the results will be shared with the user and the necessary information will be stored in the database. The application, which also supports the shared use of multiple people, will use a synchronized database and common network for this.

### **3.4 Persistent data management**

The data and other user inputs from the camera are the main drivers of the application. The data obtained from Expo-barcode-scanner framework ShopCart also receives data from the APIs of famous e-commerce sites for price tracking, such as Hepsiburada, Trendyol, n11, and compares the data of the markets and returns the most appropriate output to the user. [4][5][6] We also use SQL injection protection, CSS protection features of the Django framework so that data can be stored securely. We use the JWT token for authentication and periodically refresh these tokens. [7] The application will also have a permanent data store because the users' usernames, passwords, product baskets, group information etc. will be stored in this warehouse. On the contrary, we found it pointless to keep the pictures and barcode/QR Codes taken by the users while using the application in any storage, and we do not intend to store this data in any way. As a result, we will store the above-mentioned information in a database using SQLite, using the Django framework.

### **3.5 Access control and security**

In case the product is published, all users with market access will be able to use the application. Therefore, the only actors of the application are the users, as well as the administrators necessary for the operation of the application. Administrators do not have direct access to user data that may contain sensitive content such as passwords. Also, if they are not a group, a user cannot view the contents of the other user's shopping cart. Only group members can check and view their group and individual product baskets.

In addition to the above, it should be said that while developing the project, the General Data Protection Regulation (GDPR) is and will be adhered to. Following the GDPR regulations, we will encrypt personal data before storing it, making it very hard for others and even administrators to access and read the data. Thus, it is aimed to keep user security at a high level. [8]

### **3.6 Global software control**

ShopCart will have an event driven control system. It is of great importance in terms of user experience that multiple users can access and use the application smoothly and smoothly. This can be done using event handlers. The application uses event handlers so that each request and command asynchronously triggers a section in the source code so that users do not have to wait for other requests to complete. When a request arrives, the required subsystem receives it and processes and handles it as needed. The application server, which is developed by using Django REST Framework, is constantly on the alert to receive requests from users. The received request is noticed by React by

decoding the barcode/QR Code on the frontend, and the detected number is sent to the backend, allowing the necessary information to be called. [9] While these are happening, the system collects information from various APIs to reflect the various prices of the returned data to the user and makes it ready to be reflected to the user.

## **3.7 Boundary conditions**

Shopcart has 3 main boundary conditions. These are: starting the application, terminating the application, and encountering a failure while the application is running.

### **3.7.1 Initialization**

The user is expected to have an internet connection and to have both Android and iOS mobile devices with minimum system requirements supported by the technologies used in application development. These Android and iOS versions will be revealed towards the end of the app development process. Apart from this, the mobile device is expected to have a camera and to have minimum RAM and storage space information, which will be specified later. Thus, the user will be able to run the application on his mobile device in real time and without any problems. In addition, all users are expected to have their own account and log into the application with this account. Also, if the user has checked the "Remember Me" checkbox while logging out, the application will automatically launch with their account logged in.

### **3.7.2 Termination**

The user can exit the application instantly in any way supported by the hardware, for example by pressing the "Home" button. In addition, more securely, you can use the Logout button on the main screen to log out of the in-app account, shown as a door icon. After the user exits the program, all processes created on the server are promptly terminated, and the current information is not saved unless otherwise commanded (for example, if the add product to the list button is not pressed). In addition, if the user has selected the "remember me" option when logging out of the application, the information is stored for later use. However, unsaved transactions are not recorded and remembered because the connection to the server will be lost and they are not written to the database.

### **3.7.3 Failure**

If the user's internet connection is poor or lost, the application stops working, an error pop-up is reflected on the user's screen, and the program exits, as this will make it impossible to access the application servers and information flow from the APIs.

If the application closes unexpectedly, for example due to excessive memory usage or any phone-related problem, the user's access to the application is lost, but the user information is retained until the next successful start of the application.

If one or more of the APIs used in the application are out-dated, malfunctions may occur in the operation of the application or correct information may not be provided to the user.

Due to database errors, malfunctions may be noticed in some functions such as entry/exit, list registration.

## **4 Subsystem services**

In this section, we will explain the two major components of the system: Client and Database Server.

### **4.1 Client**

Client has two subsystems: Image Input Subsystem and Sound Import Subsystem. These subsystems are for importing various inputs about the products to search for.

#### **4.1.1 View**

View subsystem works as a bridge between the user and the other subsystems. Users requests and actions are transferred to the Controller subsystem via View subsystem. View subsystem consists of the following UI components:

##### **Login Page**

Login page helps users to login to the system. If they did not create an account the user should go to the register page.

##### **User Page**

This page has the information of the user. Users can change their information here.

##### **Register Page**

This page helps users to create new accounts.

##### **Products Page**

Products page shows the products added by the users. This page has a button that enables users to add new products. Users can also delete products here.

##### **Barcode Page**

This page has the functionality of scanning barcodes. This page opens the camera of the phone and there is also a button that uploads photos from the users gallery.

### **4.1.2 Controller Subsystem**

Controller subsystem connects the view subsystem to the other subsystems. Controller subsystem consists of these subsystems:

#### **Barcode Reader**

Barcode reader takes photos from users using the barcode page. Barcode reader is able to decipher the barcodes and send this information to the database.

#### **Authentication**

Authentication subsystem takes information from the user and connects this information to the auth manager subsystem.

#### **Products Manager**

This subsystem takes information from the user and sends this information to the product manager of the server.

## **4.2 Server**

Server handles all API requests coming from the client.

### **4.2.1 Store Manager Subsystem**

Responsible for keeping the “Store” models updated by communicating with different stores’ API’s by getting stock and price data.

### **4.2.2 Home Manager Subsystem**

Responsible for managing the models related to Home functionalities. These functionalities are updating the home shopping cart, adding/removing users from the home, creating or deleting a home.

### **4.2.3 Auth Manager Subsystem**

Responsible for login/register processes and the accessibilities of API endpoints.

### **4.2.4 Products Manager Subsystem**

Responsible for Product-related changes. Will run frequent cron jobs to update the prices and add new products into our database.

## 5 Consideration of Various Factors in Engineering Design

### **Sustainability**

Sustainability is the motivation of our project. Our resources are limited, and the total population of the world is ever-growing. This means fewer resources for each individual. Our project aims that the users buy what they need and do not waste grocery resources.

### **Public Safety**

Since the continuing epidemic, it is critical for public safety to remain isolated. Our team is unable to meet in the same setting as frequently as we would want as a result of this element, making the development process more difficult in terms of cooperation and teamwork.

### **Technological Factors**

Technology is always changing, and this is something we need to keep in mind for our project. So that our work does not become obsolete, we must maintain track of this evolution and incorporate it into our project. We must also keep an eye out for comparable items and modify our application accordingly for our product to be useful.

### **Cultural Factors**

Cuisine may differ from culture to culture. These differences can affect some predetermined cuisines and recipes added to the system. People from different cultures might view these recipes as unfamiliar.

### **Social Factors**

Users must communicate with the program in a language that they understand. We need to explore translating our application into multiple languages to make it available worldwide. Our application will be in English by default, which might be a problem given that English is spoken by around 20% of the world's population.

	Effect level	Effect
Sustainability	10	As our resources are getting scarce, sustainability helps reduce the negative effects.
Public safety	5	Making cooperation and teamwork harder for development due to isolation.
Technological factors	10	Change in design for developments of technology we utilize and the existence of similar applications.
Cultural factors	6	Change in the accuracy of image processing because of cultural differences in appearance.
Social factors	6	The language barrier between users and the application.

Table 1: Factors that can affect analysis and design.

## 6 Teamwork Details

### 6.1 Contributing and functioning effectively on the team

Workload of the project is shared among the team members in a way that each member is responsible for what they are capable of. This division helps team members work effectively and prevents the delays in the project. It also affects the motivation of team members by increasing their confidence. Because this division helps them not to feel like a failure.

### 6.2 Helping creating a collaborative and inclusive environment

Team members of the group created an inclusive environment by using communication tools to inform the others and discuss the work. In order to keep the progress of the project, our team meets twice a week. We discuss problems and progress of the project and each team member is involved in the decision making about the project. Github helped our team to collaborate easily on the project by providing ease to save our progress on the project and share with other team members.



## 6.3 Taking lead role and sharing leadership on the team

Workload of this project is divided into main work packages and detailed information was given on Analysis Report about each package. Leadership roles are shared on the team members fairly to keep the balance. This division also helps to improve the leadership skills of each team member. Leaders of each work package lead their team by deciding the action plan to fulfill the requirements of the work package and make the final decisions on debates.

## 7 Glossary

**Machine Learning:** “Study of computer algorithms that can improve automatically through experience and by the use of data [10]”

**Computer Vision:** “a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images [11]”

**API:** Application Programming Interface.

### Third Party Application

**Tensorflow:** E2E open source Machine Learning (ML) framework, used to architect and deploy ML applications [3].

**MySQL:** World’s most popular open source database [12].

**DLib:** C++ library containing machine learning algorithms and tools [13].

**Django:** High-level Python web framework used for both frontend and backend development [14].

**React Native:** A popular JavaScript-based mobile app framework that allows you to build natively-rendered mobile apps for iOS and Android. The framework lets you create an application for various platforms by using the same codebase.

**SQLite:** Claims to be the “most used database engine in the world” [15].

**GitHub:** A code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

**OpenCV:** Open source library used for computer vision applications.

**Agile Development:** software development methodologies centered round the idea of iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.

**External Api services:** ([hepsiburada.com/api](https://hepsiburada.com/api) and [trendyol.com/api](https://trendyol.com/api))

**AWS:** Amazon Web Services. Claimed to be the world's most comprehensive and broadly adopted cloud platform [16].

**S3 Bucket:** Simple storage service available in AWS.

**Expo-barcode-scanner:** An advanced barcode-scanner written in Javascript and TypeScript [6].

# References

- [1] "Grocery Guide", American Heart Association, 2020, [Online] Available: <https://www.heart.org/-/media/aha/recipe/pdf-files/grocery-guide-english-shopping-budget.pdf?la=e>. [Accessed: 24-Dec-2021].
- [2] "Food consumption in UK", Rand Corporation, 2020, [Online] Available: [https://www.rand.org/content/dam/rand/pubs/research\\_reports/RR4300/RR4379/RAND\\_RR4379.pdf](https://www.rand.org/content/dam/rand/pubs/research_reports/RR4300/RR4379/RAND_RR4379.pdf). [Accessed: 24-Dec-2021].
- [3] "Official Legal Text," General Data Protection Regulation (GDPR), 2019. [Online]. Available: <https://gdpr-info.eu/>. [Accessed: 24-Dec-2021].
- [4] hepsiburada developers: Başlangıç, <https://developers.hepsiburada.com/?docs=dokuman/baslangic>. [Accessed: 24-Dec-2021].
- [5] Giriş | Marketplace Entegrasyon Dökümanları, <https://developers.trendyol.com/tr>. [Accessed: 24-Dec-2021].
- [6] "BarCodeScanner." Expo Documentation, <https://docs.expo.dev/versions/latest/sdk/bar-code-scanner/>. [Accessed: 24-Dec-2021].
- [7] "What is SQL Injection (SQLi) and How to Prevent Attacks." Acunetix, <https://www.acunetix.com/websitesecurity/sql-injection/>. [Accessed: 24-Dec-2021].
- [8] General Data Protection Regulation (GDPR) – Official Legal Text, <https://gdpr-info.eu/>. [Accessed: 24-Dec-2021].
- [9] Django REST framework: Home, <https://www.django-rest-framework.org/>. Accessed 24 December 2021.
- [10] "Machine learning," Wikipedia, 04-Nov-2021. [Online]. Available: [https://en.wikipedia.org/wiki/Machine\\_learning](https://en.wikipedia.org/wiki/Machine_learning). [Accessed: 24-Dec-2021].
- [11] "What is Computer Vision?," IBM. [Online]. Available: <https://www.ibm.com/topics/computer-vision>. [Accessed: 24-Dec-2021].
- [12] "Tensorflow," TensorFlow. [Online]. Available: <https://www.tensorflow.org/>. [Accessed: 24-Dec-2021].
- [13] "MySQL documentation," MySQL. [Online]. Available: <https://dev.mysql.com/doc/>. [Accessed: 24-Dec-2021].

- [14] Django. [Online]. Available: <https://www.djangoproject.com/>. [Accessed: 24-Dec-2021].
- [15] Sqlite. [Online]. Available: <https://www.sqlite.org/index.html>. [Accessed: 24-Dec-2021].
- [16] "What is AWS". [Online]. Available: <https://aws.amazon.com/what-is-aws/>. [Accessed: 24-Dec-2021].