Bilkent University

Department of Computer Engineering

Senior Design Project

ShopCart

Project Analysis Report

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

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

Analysis Report Nov 15, 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.

1. Introduction

People have been going shopping to meet their needs. To survive, we all need to purchase food and beverages. Cooking, storing food in refrigerators, and ordering meals are all common activities. When ingredients are used for such purposes, they are consumed and eventually run out. When the refrigerator is nearly empty, people typically make a shopping list and purchase the items on the list. However, they may be unable to recall all of the items that need to be purchased from the market, and they frequently forget to include some of the depleted items on the shopping list.

ShopCart is a mobile application that allows users to keep track of depleted foods, ingredients, and purchase reminders. The information includes a product's monthly consumption frequency, the number of days it is out of stock, and so on. Before throwing away a depleted product, users scan its barcode, manually enter its name, or scan a photo of it to save the data. The depleted product will be automatically added to the shopping list and its numeric data, depletion time, and other information will be preserved as a result of these methods. If they live together, multiple users will have profiles and form a community. This feature allows everyone in a group to add and remove data. Users can also edit the list after purchasing it or choose not to purchase it again. The goal of this app is to make people's lives easier and more predictable. The quality of life improves when people's living standards are stable and their eating habits are protected.

In this paper, we seek to present an overall analysis of the system. First, existing systems will be discussed. After that, a general description of the proposed system, ShopCart, will be presented, with a focus on its distinguishing features. A list of functional, non-functional, and pseudo-requirements will follow. Following that, the system models of our system will be investigated. Scenarios will be created based on generic use-cases that will be depicted with diagrams. Object and class models, as well as dynamic models like activity, state, and sequence diagrams, will be discussed. Screen mockups and their explanations will be shown. Finally, we'll talk about other possible project analysis elements at the end of the report.

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.
- Does not have diet management as a functionality
- The app has an integrated system named Banabi that helps users to order groceries.

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 System

3.1. Overview

ShopCart is simply 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.

3.2. Functional Requirements

3.2.1. Login Screen

This should be the first screen the user encounters when logging into the application.

- The user must be prompted for a username and password.
- “Forgot your password?” even though the user has forgotten his password. There should be a button and this button should direct the user to the required page.
- For users who have no account or want to open a new account, there should be a “Create New Account” button and this button should direct the user to the required page.

3.2.2. Register Screen

- The user must enter the required information for registration.
- Users should enter the "Community Code" on this screen if they intend to use a shared shopping cart.

3.2.3. Add Product to Cart

In this area, the user can add products to his cart in many ways. Depending on the support provided by the technology used in the product, the user can choose the method. This preference may vary depending on the condition of the product and environmental conditions. The 5 types used in adding products are:

- Adding the product by scanning its packaging:
  - Must be able to read the brand and product name on the packaging
  - Must have the ability to add weight/unit
- Adding by reading Barcode/QR Code
- Adding with voice detection
- Manual Typing

3.2.4. My/Our Cart

- There should be a list of products listed inside this car display.
- The list should include the properties of the products (product name, product quantity).
- There should be buttons for manually removing products from the list.
- There should be a button series where the purchase amount of the products can be increased or decreased.
  - The first of these is if the user no longer wants to buy the product: the product quantity should be reduced and the product data should be removed from the list.
  - Second, if the user has already purchased: the product quantity will be reduced and the product data will be kept in the database as the day and quantity received.
- The user should be able to have more than one shopping cart.
- The location data of the selected place (home/office) for the user group or individual user will be retrieved. When users leave home with location services turned on, they will receive shopping list notifications.

3.2.5. Pricing

- After obtaining the necessary data from APIs with product information and prices, the user should be offered the most appropriate pricing.

- Price comparison: The difference between the average market price of the products and the best price we provide should be shown, and the user should be shown how much he has saved.

3.2.6. Statistics

- Individual and community data on the consumption habits of users should be shown.
- Data should be visualized in charts and pie charts with weekly and monthly periods.

3.2.7. Credits

- This section should include the names and pictures of the app developers

3.2.8. Quit

- It must allow the user to exit the game.

3.3. Non-Functional Requirements

3.3.1. User Interface and Human Factors

In the mobile application to be developed:

- An interface with easy-to-use components and vivid colors that will not disturb the user should be used.
- The number of components on the main screen and on the screens directed after it should not exceed seven to make the application more understandable and user-friendly.
- 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.

3.3.2. Application Content

- The application to be made should solve a problem in daily life or suggest a much more practical solution than the ongoing habits.

- The **usability** of the application should be increased to provide a better experience with the sounds and routers to be used in its content.

3.3.3. 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.

3.3.4. Supportability

- Must support both mainstream mobile platforms (iOS and Android)
- Must be able to read in accordance with various barcode systems.
- Must be able to integrate various cards into the system successfully.
- Should contain all possible order materials in the household.

3.3.5. Efficiency

- Users should be able to order the products they want in a much shorter time than normal purchases.
- Users should be able to control the deficiencies in the household automatically, rather than using traditional methods.
- The application should predict orders according to user habits and shorten the order and thinking process.

3.3.6. Extendibility

- By using the data entered by the users, an automatic system based on user habits can be created with the help of data training and artificial intelligence.
- Our application, which is mostly aimed at grocery shopping, can be easily adapted to many sectors where we can obtain data.
- There may be an updated version where the special offers and discounts are notified in real time and the shopping cart is arranged accordingly.

3.4. Pseudo Requirements

- The application will be for a mobile app.
- It will be developed by using React/React Native framework.
- In the backend part, we consider using the Python language and related libraries.
- The app will have a database for user and product datasets.
- The app will have a client-server architecture.
- We will use Git for version controlling and tracking.

3.5. System Models

3.5.1. Scenarios

3.5.1.1. Scenario 1

Use Case: Enroll ShopCart

Primary Actor: User

Entry Condition: User is on “Enroll ShopCart” page

Exit Condition: User either presses SignUp button or “Go to Login page” button.

Main Flow of Events:

User enters full name, username, email on “Enroll ShopCart” page.

User sets home location

Users can enter “Community Code” and set the avatar optionally.

User clicks on the Sign Up button

3.5.1.2. Scenario 2

Use Case: Add product by Scanning Image

Primary Actor: User

Entry Condition: User is on “Add Product” page

Exit Condition: User either adds the product by clicking the “Add” button or cancels by clicking on the “Backward” button.

Main Flow of Events:

User clicks on the “Scan Product Image” button on the “Add Product” page.

System opens the camera in order to scan the product to be added.

User clicks on the confirm button

System asks user to check and edit the name of the product if needed and enter the product amount

User adds the product to the shopping list by clicking the “Add” button or cancels by clicking on the “Backward” button.

3.5.1.3. Scenario 3

Use Case: Add product by Scanning Barcode

Primary Actor: User

Entry Condition: User is on “Add Product” page

Exit Condition: User either adds the product by clicking the “Add” button or cancels by clicking on the “Backward” button.

Main Flow of Events:

User clicks on the “Scan Barcode” button on the “Add Product” page.

System opens the camera in order to scan the barcode of the product to be added.

User clicks on the confirm button

System asks user to check and edit the name of the product if needed and enter the product amount

User adds the product to the shopping list by clicking the “Add” button or cancels by clicking on the “Backward” button.

3.5.1.4. Scenario 4

Use Case: Add product by Voice Detection

Primary Actor: User

Entry Condition: User is on “Add Product” page

Exit Condition: User either adds the product by clicking the “Add” button or cancels by clicking on the “Backward” button.

Main Flow of Events:

User clicks on the “Talk To Me” button on the “Add Product” page.

System opens “Talk To Me” page

User presses and holds voice button while saying product name

System asks user to check and edit the name of the product if needed and enter the product amount

User adds the product to the shopping list by clicking the “Add” button or cancels by clicking on the “Backward” button.

3.5.1.5. Scenario 5

Use Case: Add product by Typing

Primary Actor: User

Entry Condition: User is on “Add Product” page

Exit Condition: User either adds the product by clicking the “Add” button or cancels by clicking on the “Backward” button.

Main Flow of Events:

User clicks on the “Type Product name” button on the “Add Product” page.

System opens “Type Product Name” page

User types name of the product and amount

User adds the product to the shopping list by clicking the “Add” button or cancels by clicking on the “Backward” button.

3.5.1.6. Scenario 6

Use Case: Changing product amount on “My Shopping List”

Primary Actor: User

Entry Condition: User is on Shopping Lists page

Exit Condition: User clicks “Backward” button.

Main Flow of Events:

User clicks on the “My shopping List” button on the Shopping Lists page.

System opens “My Shopping List” page

User presses the green button to increase and the red button to decrease the amount of the product.

User presses the “Backward” button.

3.5.1.7. Scenario 7

Use Case: Viewing Statistics

Primary Actor: User

Entry Condition: User is on Main Menu

Exit Condition: User clicks “Backward” button.

Main Flow of Events:

User clicks on the “My Statistics” button on the Main Menu page.

System opens “Your Statistics” page

User presses the “Backward” button.

3.5.1.8. Scenario 8

Use Case: Viewing Best Prices

Primary Actor: User

Entry Condition: User is on “My Shopping List” page

Exit Condition: User clicks “Backward” button.

Main Flow of Events:

User clicks on the “Show Best Prices” button on the My Shopping List page.

System opens “Best Prices” page

User presses the “Backward” button.

3.5.1.9. Scenario 9

Use Case: Joining Community

Primary Actor: User

Entry Condition: User is on Main Menu

Exit Condition: User clicks “Backward” button.

Main Flow of Events:

User clicks on the “Community” button on the Main Menu.

System opens “Community” page

User enters Community code and presses “Join Community” button

System shows the name of the new community on the Community page

User presses the “Backward” button.

3.5.1.10. Scenario 10

Use Case: Changing Home Address

Primary Actor: User

Entry Condition: User is on Main Menu

Exit Condition: User clicks “Backward” button.

Main Flow of Events:

User clicks on the “Edit my profile” button on the Main Menu.

System opens “Edit Profile” page

User presses the Change my Home address

System opens current location of the user

User confirms the current location

System sets the location and opens Edit My Profile page again

User presses the “Backward” button.

3.5.2. Use Case Model

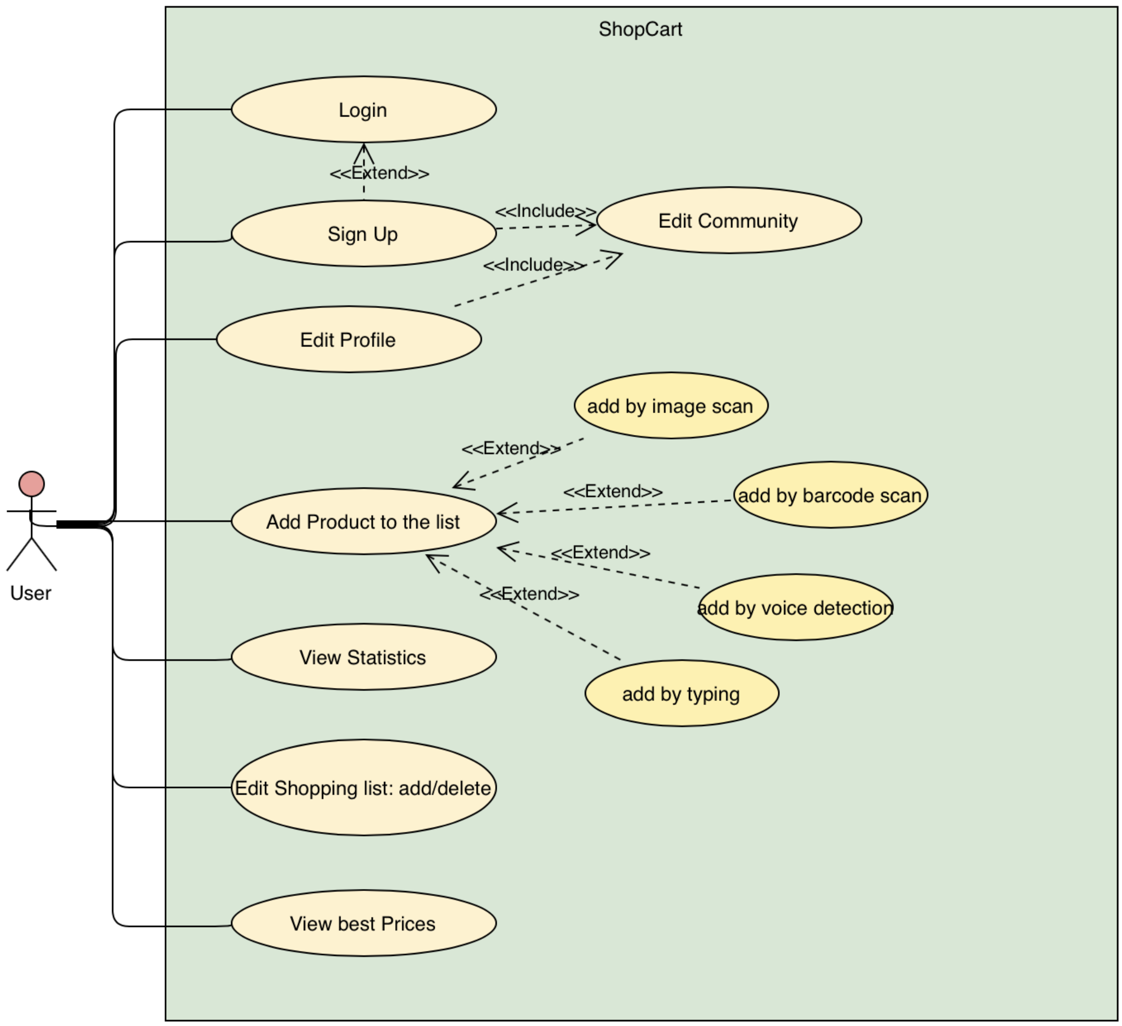


Figure 1: Use Case Model of the ShopCart application

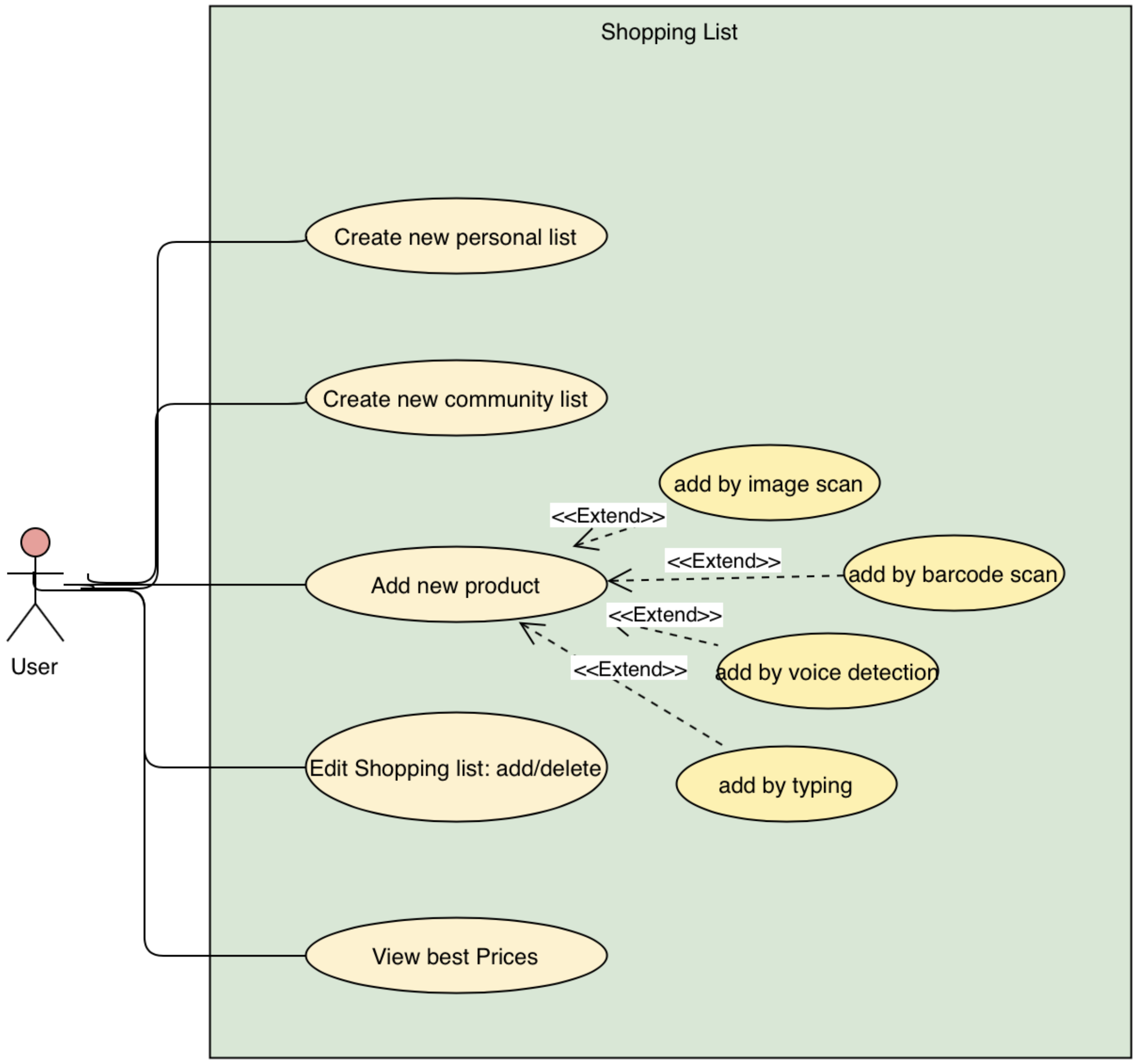
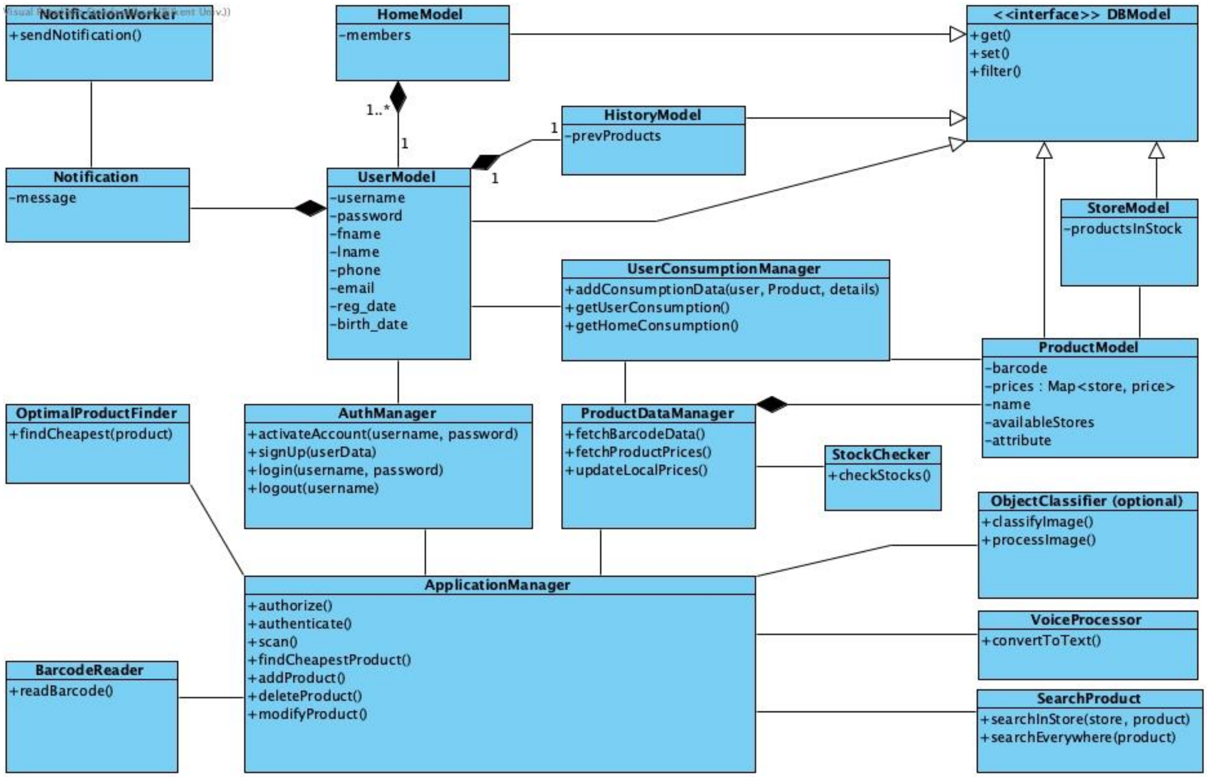


Figure 2: Use Case Model of the “Shopping List”

3.5.3. Object Class Model



- **DBModel Interface:**

Interface for our database models

- **UserModel Class:**

Represents app user.

- **StoreModel Class:**

Represents different stores which sell products.

- **ProductModel Class:**

Represents products and contains information about where that product is available in what quantity.

- **HistoryModel Class:**

Will contain the consumption history of each user.

- **HomeModel Class:**

Represents a “home” where several users may reside.

- **UserConsumptionManager Class:**

Manages the consumptions of users such as adding and getting consumption data.

- **ProductDataManager Class:**

Fetches data of products.

- **ObjectClassifier Class:**

Classifies images of products.

- **VoiceProcessor Class:**

Converts voice recording of the user to text for searching for products.

- **SearchProduct Class:**

Provides the functionalities of searching for a product.

- **BarcodeReader Class:**

Converts barcode image to string format.

- **OptimalProductFinder Class:**

Finds the product with the least price among several stores.

- **AuthManager Class:**

Manages authentication and authorization.

- **Notification Class:**

Individual user’s notifications.

- **NotificationWorker Class:**

Worker for sending the notifications.

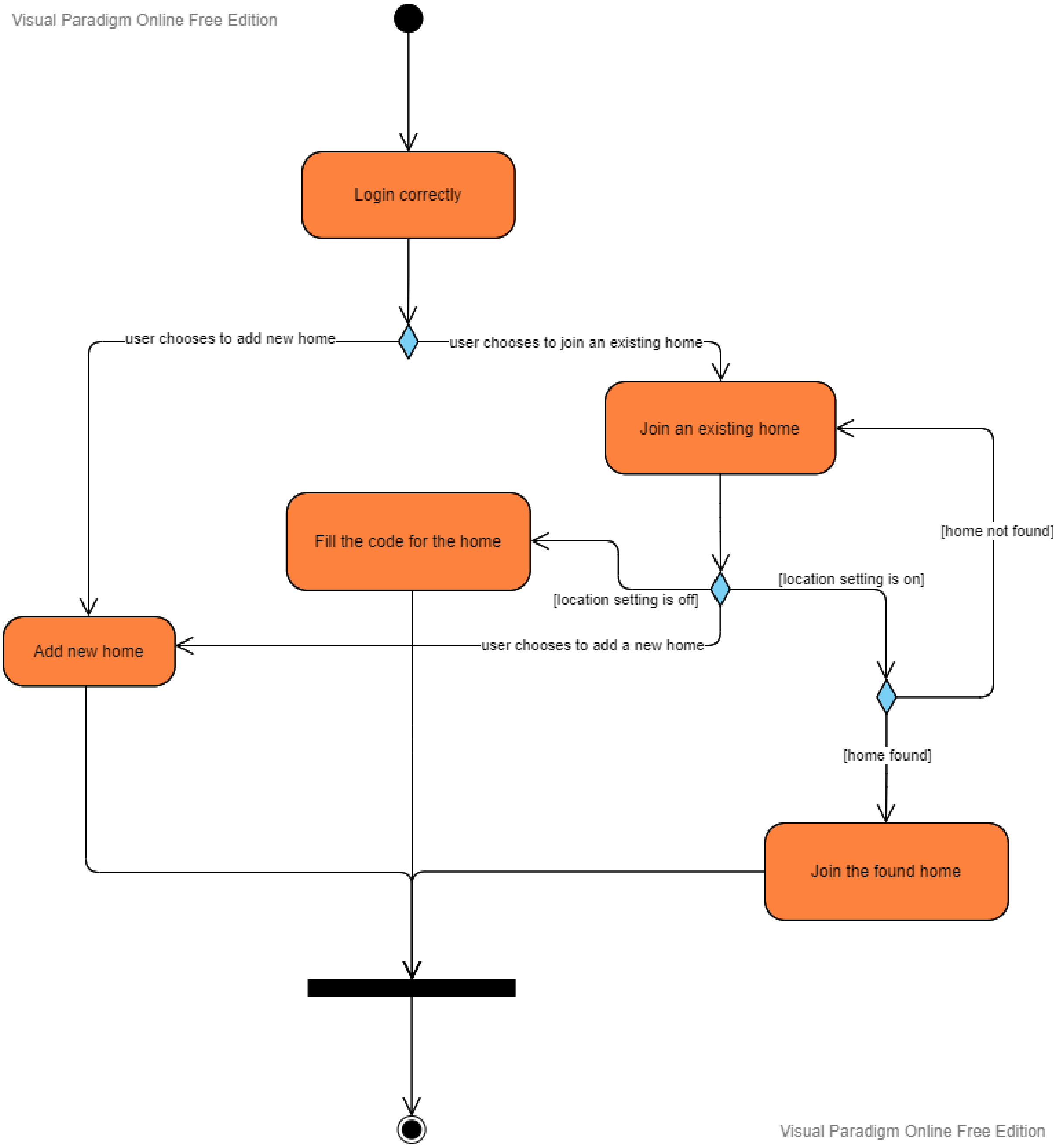
- **StockChecker Class:**

Checks for the stocks in a store.

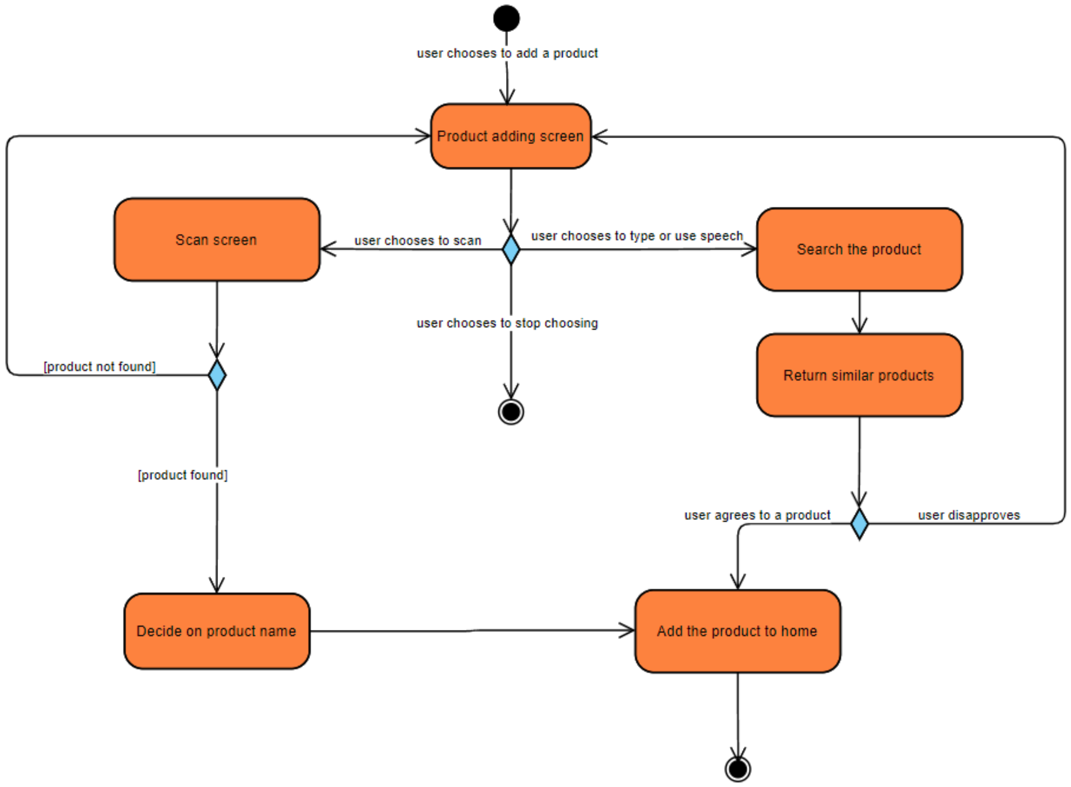
3.5.4. Dynamic Models

3.5.4.1. Activity Diagrams

Login process

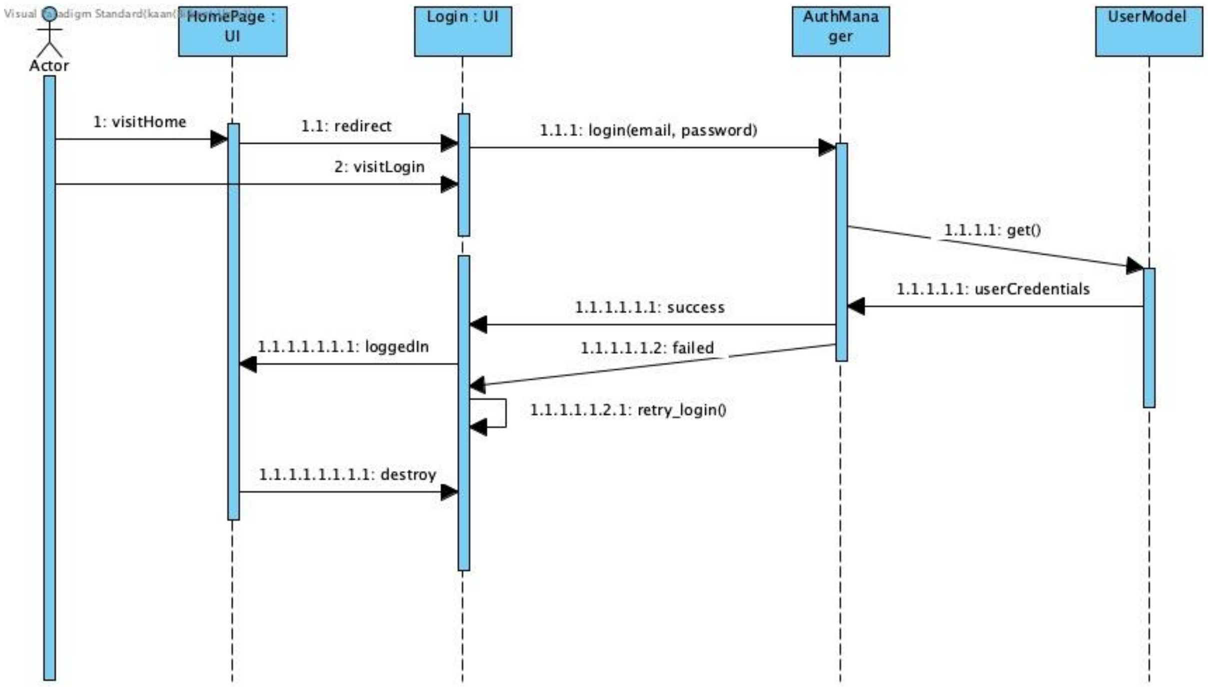

Visual Paradigm Online Free Edition
Login correctly
user chooses to add new home
user chooses to join an existing home
Join an existing home
Fill the code for the home
[home not found]
[location setting is on]
[location setting is off]
Add new home
user chooses to add a new home
[home found]
Join the found home
Visual Paradigm Online Free Edition

Adding a product to home

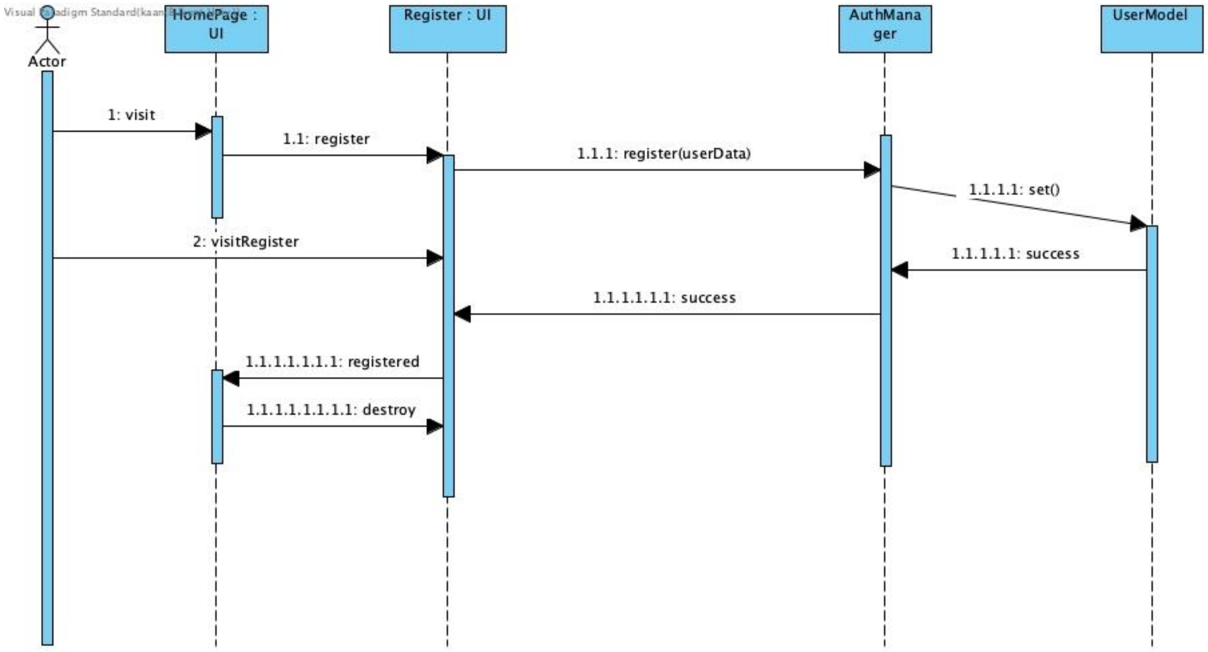


3.5.4.2. Sequence Diagrams

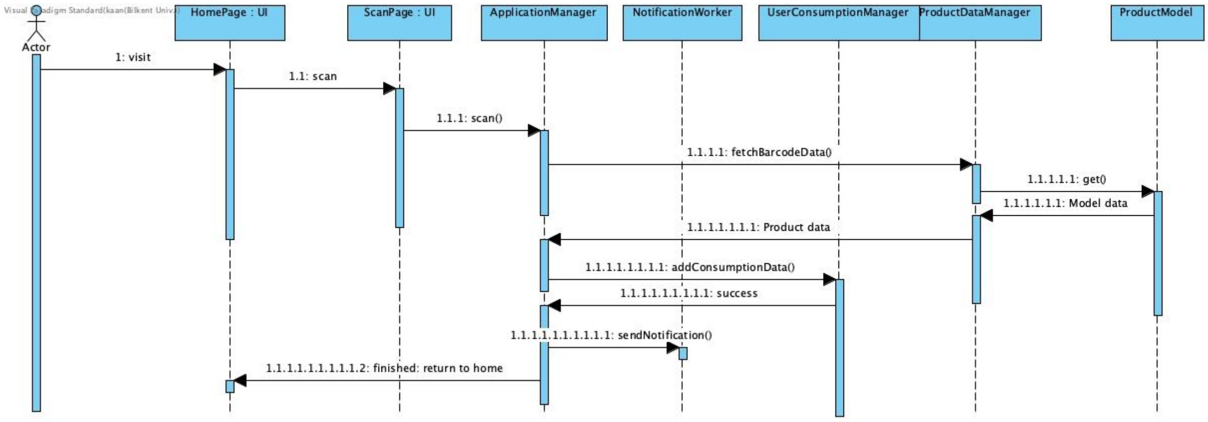
Login:



Register:



Add Consumed Product:


Visual Paradigm Standard(kaan(Bilkent Univ.))
Actor
HomePage : UI
ScanPage : UI
ApplicationManager
NotificationWorker
UserConsumptionManager
ProductDataManager
ProductModel
1: visit
1.1: scan
1.1.1: scan()
1.1.1.1: fetchBarcodeData()
1.1.1.1.1: get()
1.1.1.1.1.1: Model data
1.1.1.1.1.1.1: Product data
1.1.1.1.1.1.1.1: addConsumptionData()
1.1.1.1.1.1.1.1.1: success
1.1.1.1.1.1.1.1.1.1: sendNotification()
1.1.1.1.1.1.1.1.1.2: finished: return to home

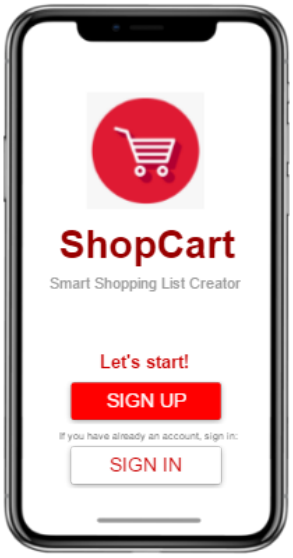
3.5.5. User Interface

3.5.5.1. Mockups

The following mockups have been drawn to give an idea, although the GUI preferences will not be exactly as seen:

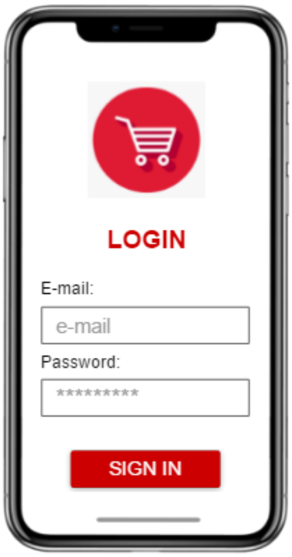
3.5.5.2. Welcome Screen

This screen is the user welcome screen. Here the user can login if registered, register if not.


ShopCart
Smart Shopping List Creator
Let's start!
SIGN UP
If you have already an account, sign in:
SIGN IN

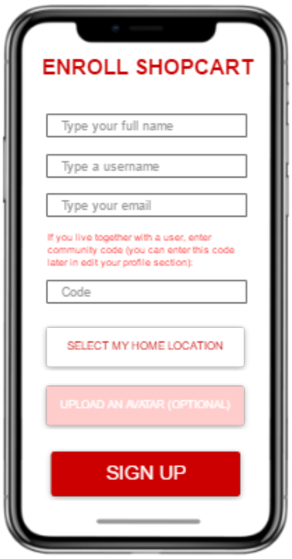
3.5.5.3. Login Screen

The user can log in to the application by entering their email address/username and password on this screen.



3.5.5.4. Register Screen

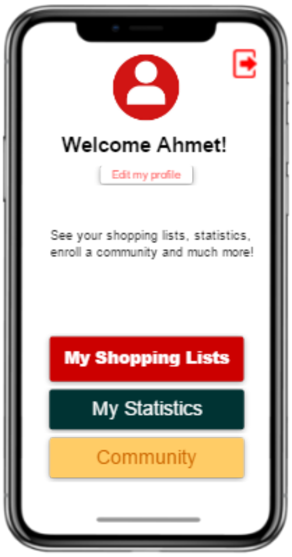
This screen is the registration screen. From here, the user is faced with a screen where he/she will enter his/her name, user name, e-mail and if he/she wants to use the application together, the code of an already existing group. Here he will define the location of his home, either automatically or manually, for use in the application. It is also available on this screen, where he will optionally choose to add his own avatar to the application.



If you live together with a user, enter community code (you can enter this code later in edit your profile section):

3.5.5.5. Main Screen

This screen is the screen that welcomes us after successfully logging in on the application login screen. From this screen, "My Shopping List", "Statistics" and the Communities where the user is can be accessed. You can also access the profile editing section here.



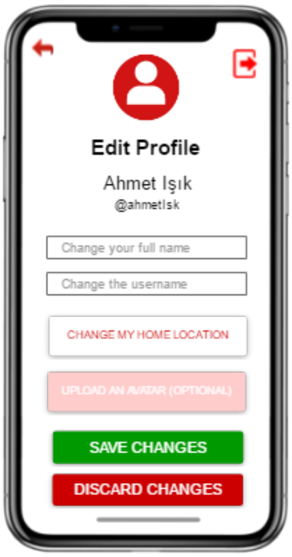
Welcome Ahmet!

Edit my profile

See your shopping lists, statistics, enroll a community and much more!

3.5.5.6. Edit Profile Screen

As the name suggests, the user can edit their own information on this screen.



Edit Profile

Ahmet Işık

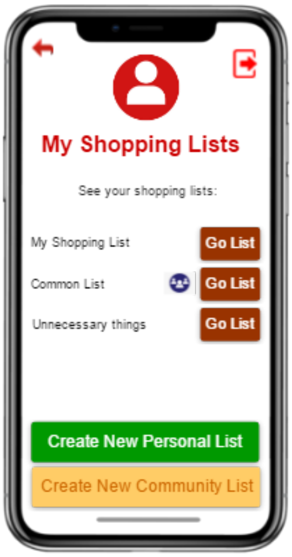
@ahmetIsk

Change your full name

Change the username

3.5.5.7. Shopping List Screen

On this screen, lists of your own and the groups you belong to are displayed. Adding new lists is also done here.



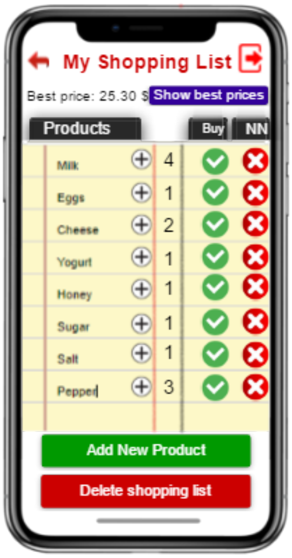
My Shopping Lists

See your shopping lists:

My Shopping List	Go List
Common List	Go List
Unnecessary things	Go List

3.5.5.7.1. Inside a Shopping List

The shopping list is displayed on this screen. Additions can be made to the list. The best prices for the products on the list can be viewed by clicking on the "Show best prices" button.



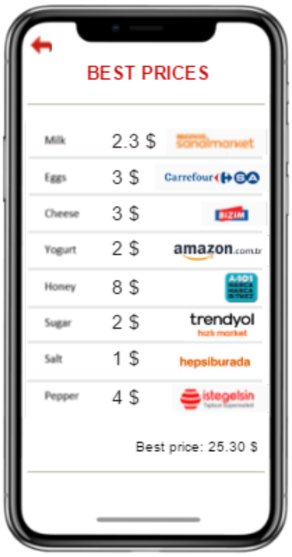
My Shopping List

Best price: 25.30 $ Show best prices

Products		Buy	NN
Milk	4	✓	✗
Eggs	1	✓	✗
Cheese	2	✓	✗
Yogurt	1	✓	✗
Honey	1	✓	✗
Sugar	1	✓	✗
Salt	1	✓	✗
Pepper	3	✓	✗

3.5.5.8. Prices Screen

Various prices drawn from market APIs are displayed on this screen. The user can examine the prices and choose the market they want. The application will suggest the user the most suitable market by default.



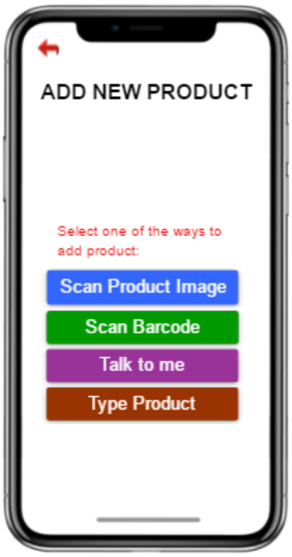
BEST PRICES

Milk	2.3 $	sanalmarket
Eggs	3 $	Carrefour SA
Cheese	3 $	BIZIM
Yogurt	2 $	amazon.com.tr
Honey	8 $	A101
Sugar	2 $	trendyol hızlı market
Salt	1 $	hepsiburada
Pepper	4 $	istegelsin

Best price: 25.30 $

3.5.5.9. Add a Product Screen

Under the "Add a product" option, there will be 4 options as we have planned right now. These will be image scanning, barcode/QR code reading, voice command, and manual typing.

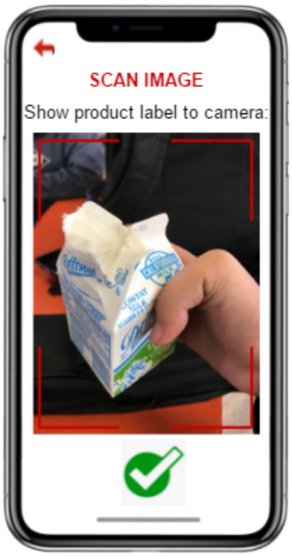


ADD NEW PRODUCT

Select one of the ways to add product:

3.5.5.10. Scan Image Screen

When you bring the product you want to read closer to the camera and view it at an appropriate angle, the application should be able to recognize the product and add it to your list.

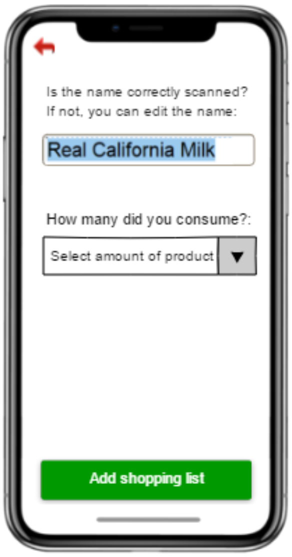


SCAN IMAGE

Show product label to camera:

3.5.5.11. Edit Item Screen

On the next screen, there is an optional text correction option and the desired number of entries is provided.



Is the name correctly scanned?
If not, you can edit the name:

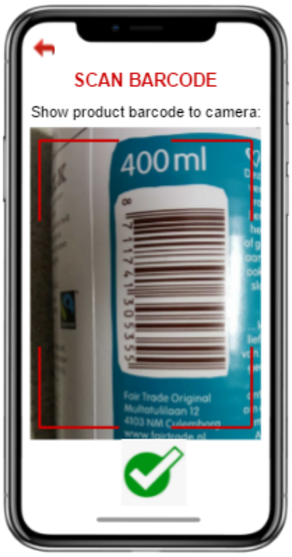
Real California Milk

How many did you consume?:

Select amount of product ▼

3.5.5.12. Scan Barcode/QR Code Screen

The user quickly identifies the product to be consumed or consumed from the barcode/QR code and ensures that the new product is added to the shopping cart by the application.

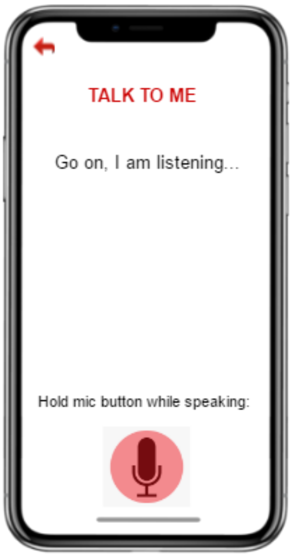


SCAN BARCODE

Show product barcode to camera:

3.5.5.13. Voice Recognition Screen

The user can enter the item using the voice input of his/her phone.



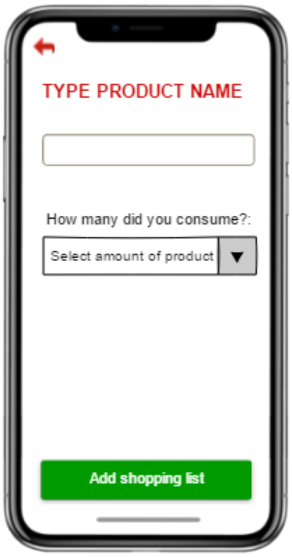
TALK TO ME

Go on, I am listening...

Hold mic button while speaking:

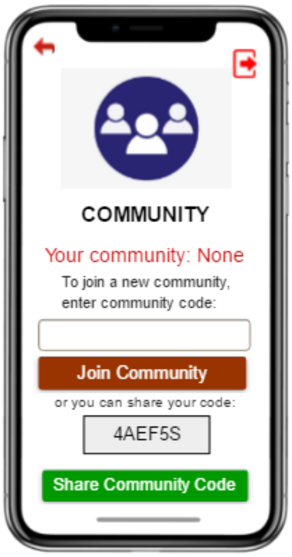
3.5.5.14. Manuel Typing Screen

In cases where all other item entry ways do not work, the traditional way of manual typing can be used.



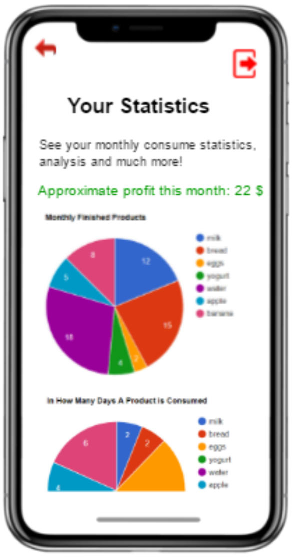
3.5.5.15. Community Screen

On this screen, the user can enter a code to join the desired community or generate a code to share with other users.


COMMUNITY
Your community: None
To join a new community, enter community code:
Join Community
or you can share your code:
4AEF5S
Share Community Code

3.5.5.16. Statistics Screen

One of the most important elements of the application is the statistics that appear on this screen. By examining the information, graphics and pie charts on this screen, the user can examine the user habits about himself and his community and take action accordingly.



Your Statistics

See your monthly consume statistics, analysis and much more!

Approximate profit this month: 22 $

4. Other Analysis Elements

4.1. 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: Because of 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.

4.2. Risks and Alternatives

Following team talks, we identified three potential risks that could arise in our project and devised a plan B for each of them to address those risks.

The first danger we saw was the project's scope being too broad. Throughout the project, all team members will be required to learn numerous new concepts. We have a limited amount of time, therefore this procedure must be completed quickly. However, some of the expertise that the project will require may be more difficult to come by. This could create delays in the project's development. That is why we have chosen to divide the different aspects of the project between our teams.

The inadequacies in the schedule are the second risk. In software development, estimating the time required for a task is difficult. Underestimating the time necessary for activity leads future tasks to be delayed. As a result, the entire project plan may need to be revised. We may avoid such problems by using an agile development technique. As a result, estimations for shorter durations are generated based on feedback, and future tasks are selected based on the completion of present ones.

The absence of a member is the third risk. At the moment, we are dealing with a pandemic. Our members might contract the virus, or they can be quarantined. Members of the team can also fail or withdraw from the course. In such circumstances, the workforce is downsized, and those who remain are given more work than before. Changing the intended features or spreading the work among the remaining members are two options.

	Likelihood	Effect on the project	B Plan Summary
Scope of the Project Being Broad	Medium	Aren't enough resources to implement all of the planned features.	Implementing an app which is mostly concerned about keeping track of the groceries, not diet management.
Schedule Flaws	High	Future planning that is inaccurate.	Agile method and planning for shorter periods according to feedback.
Absence of a Member	Medium	Workforce getting downsized causes people to do more work.	Reassign the work to the remaining team members.

Table 2: Risks

4.3. Project Plan

A planning activity is shaped with respect to our project goals and plans. This planning activity consists of work packages (WPs) that are assigned to each of the project members. According to the plan, leaders and members are assigned to work each WP within extendable deadlines. The work consists of tasks and deliverables for the WPs given below.

WP#	**Work package title**	**Leader**	**Members involved**
WP1	Project Specifications Report (Done)	Ahmet	Everyone
WP2	Project Website(Done)	Ahmet	-
WP3	Analysis Report	Furkan	Everyone
WP4	Image Processing Learning and Practices	Kaan	Ravan, Furkan
WP5	GUI Implementation	Ahmet	Mehmet, Furkan
WP6	Backend Implementation	Kaan	Ahmet, Furkan
WP7	High-level Design Report	Ravan	Everyone
WP8	Voice Processing Learning and Practices	Mehmet	Kaan, Ahmet
WP9	Barcode Data Collection	Furkan	Mehmet, Ravan
WP10	Low-level Design Report	Kaan	Everyone
WP11	Testing	Ravan	Everyone
WP12	Final Report	Mehmet	Everyone
WP13	Final Presentation	Ahmet	Everyone

Table 3: List of work packages

WP 1: *Project Specifications Report*			
Start date: *16.09.2021* **End date:** *11.10.2021*			
Leader:	*Ahmet*	**Members involved:**	*Everyone*
Objectives: *Creation of project specifications report. Declaring the constraints of the system. Reporting the functional and non-functional requirements*			
Tasks: ***Task 1.1 Constraints*** **:** *Determining on the constraints of the system. These constraints include implementation, economic, social, technical, ethical etc. constraints.* ***Task 1.2 Functional Requirements:*** *Deciding on the functional requirements of the system.* ***Task 1.3 Non-functional Requirements*****:** *Deciding on the non-functional requirements of the system. These include the security, usability, performance etc. non-functional requirements...*			

Deliverables

D1.1: *Project Specifications Report*

WP 2: *Creating a Website*

Start date: *3.10.2021* **End date:** *11.10.2021*

Leader:	*Ahmet*	**Members involved:**	-

Objectives: *To create a website that will be used to provide information and documentation about our project.*

Tasks:

Task 2.1 Picking a Host **:** *Deciding where to host our website.*

Task 2.2 Picking a Theme: *Picking an appropriate and easy to use theme for our website.*

Task 2.3 Creating a Source Code: *Coding the website to contain necessary information for the project*

…

Deliverables

D2.1: *A fully functioning, informative website that displays the documentation of our project, our team members and our project information.*

WP 3: *Analysis Report*

Start date: *11.10.2021* **End date:** *15.11.2021*

Leader:	*Furkan*	**Members involved:**	*Everyone*

Objectives: *Creating the analysis report of the project. Researching the current systems and describing the proposed system using the requirements of the system and models. Creating suitable scenarios and use cases. Examining the requirements and displaying the product's visualization. Inspecting other analysis factors such as social, economical etc.*

Tasks:

Task 3.1 Current Systems Research**:** *Researching the current systems. Comparing them and discovering their purposes.*

Task 3.2 Proposed System Description **:** *Describing the proposed system and comparing it with current systems. Modeling the system using diagrams and mockups.*

Task 3.3 Inspecting other analysis factors**:** *Analyzing the effects of the system on the social groups.*

Deliverables

D3.1: *Analysis Report*

WP 4: *Image Processing Learning and Practices*

Start date: *15.11.2021* **End date:** *15.12.2021*			
Leader:	*Kaan*	**Members involved:**	*Ravan, Furkan*
Objectives: *Implementing an object scanner system that reads the sentences or words on the product and presents the result as an editable text*			
Tasks: ***Task 4.1 Learning Word Detection in Images:*** *Learning the image processing structure and its implementation which is word detection. Practising in sample cases and analysis of sample codes.* ***Task 4.2 Applying Word Detector:*** *Applying the word detection & reading from the image scan and creating an editable text on the app.* ***Task 4.3 Object Detection:*** *Using YOLOv3 in Darknet Convolutional Neural Network Framework in order to detect objects like refrigerator products in camera to be able to detect shown products and their names*			

Deliverables ***D4.1:*** *Fully functioning image scanner that detects the name of the products kept towards the camera*			
WP 5: *GUI Implementation*			
Start date: *15.11.2021* **End date:** *10.02.2022*			
Leader:	*Ahmet*	**Members involved:**	*Mehmet, Furkan*
Objectives: *Implementing the mobile application using Javascript with React Native and creating a useful and interactive interface.*			
Tasks: ***Task 5.1 Starting the Project*****:** *Starting a new React Native project and creating all pages with corresponding routing.* ***Task 5.2 Implementing All PAges*****:** *Implementing required components excluding the image and barcode scan functionality.* ***Task 5.3 Integrating with the Database*** **:** *Establishing connection with database for profile info, product information, community and shopping list products.* ***Task 5.4 Implementing Image and Barcode Scan*****:** *The process will be conducted with the correlation of the backend. The result will be displayed on the corresponding screen.*			
Deliverables ***D5.1:*** *Mobile application that provides all the functionality promised in Functional Requirements section*			
WP 6: *Backend Implementation*			
Start date: *25.11.2021* **End date:** *01.02.2021*			
Leader:	*Kaan*	**Members involved:**	*Ahmet, Furkan*
Objectives: *Implementing the mobile application's logic using Django and implementing the database using SQLite.*			
Tasks: ***Task 6.1 Implement project structure*****:** *Implementing the backend side with Django.* ***Task 6.2 Creating API Endpoints*****:** *Coding API endpoints to provide communication between frontend side..* ***Task 6.3 Creating Database*****:** *Creating the database of the app to get and store the data, integrating the*			

data with the backend side.			
Deliverables ***D6.1:*** *Logic of the application with endpoints and the database*			
WP 7: *High-level Design Report*			
Start date: *01.12.2021* **End date:** *24.12.2021*			
Leader:	*Ravan*	**Members involved:**	*Everyone*
Objectives: *Creating a high-level design report of the project. Analyzing the high level system structures, subsystems and data handling in the project, as well as considerations about the cultural, global , economical, social aspects of our high level design.*			
Tasks: ***Task 7.1 Subsystem Decomposition:*** *Creating a high level design of our project systems and subsystems.* ***Task 7.2 Data Management* :** *Figuring out how each subsystem will communicate with each other, on top of how to store the data.* ***Task 7.3 Inspecting other analysis elements* :** *Inspecting our high level design in terms of global , social, economic and cultural elements.*			
Deliverables ***D7.1:*** *High-level Design Report*			
WP 8: *Voice Processing Learning and Practices*			
Start date: *15.12.2021* **End date:** *15.01.2022*			
Leader:	*Mehmet*	**Members involved:**	*Kaan, Ahmet*
Objectives: *Implementing a voice listener system that listens the sentences from the user about the product and presents the result as an editable text*			
Tasks: ***Task 8.1 Learning Voice Detection:*** *Learning the voice processing structure and its implementation which is word detection. Practising in sample cases and analysis of sample codes.* ***Task 8.2 Applying Voice Word Detector:*** *Applying the voice detection & listening from the user and creating an editable text on the app.*			
Deliverables ***D8.1:*** *Fully functioning voice detector that detects the name of the products from the user's voice*			
WP 9: *Barcode Data Collection*			
Start date: *15.12.2021* **End date:** *01.03.2022*			
Leader:	*Furkan*	**Members involved:**	*Mehmet, Ravan*
Objectives: *Collecting the barcode data of the products that can be added to our shopping lists*			
Tasks:			

Task 9.1 Collecting the barcode data in markets: *Communicating with the responsible parts of the markets and getting the barcode data to store our database and detect the products later.* ***Task 9.2 Implementing the barcode data and pairing them with products*** : *Creating the connection between barcodes and products and storing them at our database.*			
Deliverables ***D9.1:*** *Barcode scanner product addition feature*			
WP 10: *Low-level Design Report*			
Start date: *01.02.2022* **End date:** *15.02.2022*			
Leader:	*Kaan*	**Members involved:**	*Everyone*
Objectives: *Creating the low-level design report of the project..*			
Tasks: ***Task 10.1 Design Choices***: *Analyzing design trade-offs and compliance with engineering standards.* ***Task 10.2 Packages*** : *Creating low-level design of our project and identifying subsystem packages, creating package diagrams of subsystems.* ***Task 10.3 Class Interfaces*** : *Describe class attributes, method signatures, interfaces in detail*			
Deliverables ***D10.1:*** *Low-level Design Report*			
WP 11: *Testing*			
Start date: *25.03.2022* **End date:** *10.04.2022*			
Leader:	*Ravan*	**Members involved:**	*Everyone*
Objectives: *Testing the implemented parts, and their integrity.*			
Tasks: ***Task 11.1 Testing the GUI***: *Testing the user interface in different mobile OS versions.* ***Task 11.2 Testing Image Processor:*** *Testing the image scanner and the success rate of the word guessing.* ***Task 11.3 Testing Voice Detector***: *Testing the voice detector's success rate with different voices. .* ***Task 11.4 Testing the overall integrity:*** *Testing the integrity between different parts of the application.*			
Deliverables ***D11.1:*** *New bugs & problems that require our solutions*			
WP 12: *Final Report*			
Start date:. *01.04.2022* **End date:** *26.04.2022*			
Leader:	*Mehmet*	**Members involved:**	*Everyone*

Objectives: *Creating the final report of the project, which includes the revised content from analysis and design reports, details about our implementation and test, and a maintenance plan.*			
Tasks: ***Task 12.1 Include Revised Information*****:** *Analyse and revise the information from the previous two reports and add them to the final report.* ***Task 12.2 Implementation and Test Details*** **:** *Report the final details of the implementation of the program and the methods and results of our testing processes.* ***Task 12.3 Maintenance Plan*** **:** *Conduct a plan to maintain the program in the future.*			
Deliverables ***D12.1:*** *Final Report*			
WP 13: *Final Presentation*			
Start date: *01.05.2022* **End date:** *05.05.2022*			
Leader:	*Ahmet*	**Members involved:**	*Everyone*
Objectives: *Preparing the presentation of the overall project and the demo of the finished product.*			
Tasks: ***Task 13.1 Share the presentation among teammates*****:** *Sharing the workload for the presentation among team members.* ***Task 13.2 Creating a speech*****:** *Prepare a speech for the final presentation.* ***Task 13.3 Creating the slides*** **:** *Creating the slides for the final presentation* ***Task 13.4 Creating a demo:*** *Preparing a demo scenario to show and market our product*			
Deliverables ***D13.1:*** *Slides for the final presentation* ***D13.2:*** *Speech for the final presentation* ***D13.3:*** *Demo for the final presentation.*			

4.4. Ensuring Proper Teamwork

Every team member will be involved at each step of the project. We will make decisions democratically in order to ensure inclusive and collaborative teamwork. We think that the main reason for a successful project is effective team work. For this reason we will not make anyone responsible for the certain aspect of the project alone. To maintain this we will circulate duties between all team members and this will make everybody share the leadership and everybody will have enough knowledge about each part of the project. So, the leadership of the project will be shared among every team member. In order to be able to

support such an environment we will need a project management tool and our team decided to use GitHub as the primary project management tool, which all of us are familiar with.
By the help of this environment each team member will have a chance to learn leadership. Although each team member has priorities and practices at different job kits, tasks will be displayed according to them.
We expect everyone will make almost the same contribution for the implementation of the project by the help of GitHub. We also expect everybody will actively participate in meetings, including real-life and online. If any team member fails to contribute as much as others then we will meet in order to motivate these members.

4.5. Ethics and Professional Responsibilities

It can be said as a Global Impact of the ShopCart, by the help of this application, shopping behaviors of the users can be improved. For example, users will not go shopping as long as they need to, they will be aware of the amount they consume and this will prevent wasting the products.

In order to show the Economic Impact of ShopCart we can say that it will help users spend less by making them aware of the amount of the products they consume in a certain amount of time and showing the cheapest prices. Also, our application will be free.

Our application will also have an impact on Environment. By changing the shopping behaviors of the customers it will reduce the waste of the products and this will directly affect the environment.

For the Societal Impacts of the application, no personal data will be shared with third party companies including shopping behaviors and their statistics.

- Because of the piracy problem the source code of the project will be private.
- In order to keep the progress of the project, our team will be meeting twice a week. According to the schedule of the project, our meetings will be Scrum meetings [10]
- Each team member will be assigned a role in order to apply the divide and conquer strategy. [11].
- For the ShopCart project, we will use General Data Protection Regulation (GDPR) [12] and any personal data about users’ shopping behavior will not be shared with third-party companies, it will be anonymous. Codes of Ethics will be applied in our project [13,14,15].

4.6. Planning for New Knowledge and Learning Strategies

During the development of this project we will need these skills:

- Back-end development
- Front-end development
- Android app development
- iOS app development
- Image analysis

We will need to learn new technologies for this project, including image analysis, machine learning, new frameworks and libraries. Courses that we learned in Bilkent, interviews with our innovation expert, online research and learning from team members will help us in this process.

5. Glossary

Machine Learning: “study of computer algorithms that can improve automatically through experience and by the use of data [1]”

Computer Vision: “a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images [2]”

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 [4].

DLib: C++ library containing machine learning algorithms and tools [5].

Django: High-level Python web framework used for both frontend and backend development [6].

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” [7].

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: (cimri.com and akakce.com)

AWS: Amazon Web Services. Claimed to be the world's most comprehensive and broadly adopted cloud platform [8].

S3 Bucket: Simple storage service available in AWS.

QuaggaJS: An advanced barcode-scanner written in Javascript and TypeScript [9].

API: Application Programming Interface.

References

[1]“Machine learning,” *Wikipedia*, 04-Nov-2021. [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning. [Accessed: 14-Nov-2021].

[2]“What is Computer Vision?,” *IBM*. [Online]. Available: https://www.ibm.com/topics/computer-vision. [Accessed: 14-Nov-2021].

[3] “Tensorflow,” *TensorFlow*. [Online]. Available: https://www.tensorflow.org/. [Accessed: 14-Nov-2021].

[4] “MySQL documentation,” *MySQL*. [Online]. Available: https://dev.mysql.com/doc/. [Accessed: 14-Nov-2021].

[5] “Dlib C++ library,” *dlib C++ Library*. [Online]. Available: http://dlib.net/. [Accessed: 14-Nov-2021].

[6] *Django*. [Online]. Available: https://www.djangoproject.com/. [Accessed: 14-Nov-2021].

[7] *Sqlite*. [Online]. Available: https://www.sqlite.org/index.html. [Accessed: 14-Nov-2021].

[8] What is *AWS*. [Online]. Available: https://aws.amazon.com/what-is-aws/. [Accessed: 14-Nov-2021].

[9] Ericblade, “Ericblade/Quagga2: An advanced barcode-scanner written in Javascript and typescript - continuation from https://github.com/serratus/quaggajs,” *GitHub*. [Online]. Available: https://github.com/ericblade/quagga2. [Accessed: 14-Nov-2021].

[10]]“Scrum - what it is, how it works, and why it's awesome,” Atlassian, 2020. [Online]. Available: https://www.atlassian.com/agile/scrum. [Accessed: 10- Nov- 2021].

[11] “Approaches to software development,” OpenLearn, 2020. [Online]. Available: https://www.open.edu/openlearn/science-maths-technology/approaches-software-development/content-section-1.5. [Accessed: 10- Nov- 2021].

[12] “What is GDPR, the EU’s new data protection law? ” GDPR, 2021 Available: https://gdpr.eu/what-is-gdpr/ [Accessed: 10- Oct- 2021]

[13] “Code of Ethics | National Society of Professional Engineers,” Nspe.org, 2020. [Online]. Available: https://www.nspe.org/resources/ethics/code-ethics. [Accessed: 10- Nov- 2021].

[14] “How Codes of Ethics Work,” Investopedia, 2020. [Online]. Available: https://www.investopedia.com/terms/c/code-of-ethics.asp#:~:text=A%20code%20of%20ethics%20is,business%20honestly%20and%20with%20integrity.&text=A%20code%20of%20ethics%20also,an%20employee%20code%20of%20conduct. [Accessed: 10- Nov- 2021].

[15] “Official Legal Text,” General Data Protection Regulation (GDPR), 2019. [Online]. Available: https://gdpr-info.eu/.[Accessed: 10- Nov- 2021].