



# MID-TEAM REPORT

Graduation at CGI

**Name:** Yusuf Onu Ibrahim

**Student Number:** 3706036

**Study program:** Software engineering

**Location:** Venlo, The Netherlands

**University coach:** Kuiper, Matthijs M.S.

**Company supervisor:** Denis Gorianin

**Word count:** 5500

## Abstract

This mid-term report provides an update on the progress made in developing the Smart Shopper application (Graduation project). The report outlines the initial research conducted to determine the requirements and specifications of the application, as well as the methodology used to acquire and evaluate data sources. The progress made in analyzing, designing, and executing the application is also covered in the report, along with the decision to use the Ionic and Angular frameworks for its development. Finally, the report summarizes the challenges encountered during the development process, as well as plans for future work.

## Table of Contents

<b>ABSTRACT.....</b>	<b>1</b>
<b>1.INTRODUCTION.....</b>	<b>4</b>
1.1 COMPANY.....	4
1.2 CONTEXT.....	4
1.3 BACKGROUND .....	4
<b>2. PROJECT.....</b>	<b>5</b>
2.1. OBJECTIVE.....	5
2.2. SCOPE.....	5
2.3. OUT OF SCOPE .....	5
2.4. PROJECT TEAM .....	6
2.5. PROJECT MANAGEMENT.....	6
2.5 PROJECT SCHEDULE .....	7
<b>3. ANALYSIS .....</b>	<b>8</b>
3.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENT .....	8
3.2 USE CASE IDENTIFICATION .....	10
3.3 USE CASE DIAGRAM .....	10
3.4 USE CASE DESCRIPTION TABLE .....	11
3.5 ACTIVITY DIAGRAMS .....	12
3.6 SEQUENCE DIAGRAMS.....	12
<b>4. DESIGN.....</b>	<b>13</b>
4.1 WIREFRAMES .....	13
4.2 UI DESIGN.....	14
<b>5. API RESEARCH .....</b>	<b>15</b>
5.1 CONTEXT.....	15
5.2 GOAL.....	15
5.3 METHODOLOGY.....	15
5.4 API RESULT .....	16
5.5 WEB-SCRAPING .....	20
5.6 ADVANTAGES AND DISADVANTAGES.....	21
5.6.1 API.....	21
5.6.2 Web scraping .....	21
5.7. RESEARCH CONCLUSION .....	22
<b>6. IMPLEMENTATION .....</b>	<b>23</b>
6.1 CHOICE OF DEVELOPMENT FRAMEWORK.....	23
6.2 USE CASE 1 (VIEW PRODUCT) IMPLEMENTATION.....	23
<b>7.CONCLUSION .....</b>	<b>25</b>
<b>8. REFLECTION.....</b>	<b>26</b>
<b>9. REFERENCE .....</b>	<b>27</b>

## STATEMENT OF AUTHENTICITY

Issued by the FHTenL Examination Board, September 2017

I, the undersigned, hereby certify that I have compiled and written this document and the underlying work / pieces of work without assistance from anyone except the specifically assigned academic supervisor. This work is solely my own, and I am solely responsible for the content, organization, and making of this document and the underlying work / pieces of work.

I hereby acknowledge that I have read the instructions for preparation and submission of documents / pieces of work provided by my course / my academic institution, and I understand that this document and the underlying pieces of work will not be accepted for evaluation or for the award of academic credits if it is determined that they have not been prepared in compliance with those instructions and this statement of authenticity.

I further certify that I did not commit plagiarism, did neither take over nor paraphrase (digital or printed, translated or original) material (e.g. ideas, data, pieces of text, figures, diagrams, tables, recordings, videos, code, ...) produced by others without correct and complete citation and correct and complete reference of the source(s). I understand that this document and the underlying work / pieces of work will not be accepted for evaluation or for the award of academic credits if it is determined that they embody plagiarism.

Name (in capital letters): YUSUF ONU IBRAHIM

Student number: 3706036

Place / Date: Venlo/ 27-03-2023

Signature: 

# 1. Introduction

## 1.1 Company

Founded as an IT consulting on June 15, 1976, in Quebec City, CGI (Consultants to Government and Industry Incorporated) is a leading provider of IT and business consulting services globally. The company's focus is on driving results through the use of insights and data. Operating in 21 industries across 400 locations worldwide with approximately 90,000 professionals, CGI offers comprehensive, scalable, and sustainable consulting services that are informed by a global perspective and delivered locally. The company has a branch in Eindhoven located at the High-tech campus, which is where the graduation project will be carried out. The project is a smart shopper application, which aims to make grocery shopping as easy and affordable as possible for users (CGI internationals) by providing information on where to find the cheapest prices, promotions, and bulk purchase options in The Netherlands.

## 1.2 Context

This paper will address the progress made on the graduation project, which involves the development of the Smart Shopper application. It will provide an overview of the goals and objectives of the project, as well as the research conducted to inform the development process. The paper will also outline the methodology used, including the selection of the appropriate framework for development. Additionally, the paper will discuss the challenges encountered and the solutions implemented. Finally, it will provide a summary of the current status of the project and the plans for future development.

## 1.3 Background

The smart shopper application aims to address the challenge of finding the best deals on groceries, which can be a time-consuming and tedious task. Consumers often have to visit multiple stores, compare prices, and keep track of flyers and coupons to get the best deals, which can be overwhelming and discourage them from shopping around for the best prices, resulting in higher grocery costs. The smart shopper application aims to simplify the grocery shopping experience by providing information on the best deals and prices from different stores. With the application, users can easily compare prices and find the stores with the lowest prices for items on their shopping list, as well as receive notifications about sales and promotions at their preferred stores to take advantage of discounts and save money on groceries. In summary, the smart shopper application aims to address the lack of an efficient and convenient way for individuals to find the best deals on groceries, making the shopping experience more efficient and cost-effective through the use of real-time information and alerts.

## 2. Project

### 2.1. Objective

The smart shopper application will allow users(CGI internationals) to quickly and easily compare prices, by finding the stores offering the lowest prices for the items on their shopping list. Users can also receive notifications about sales and promotions at their preferred stores, enabling them to take advantage of discounts and save money on groceries. The application will also feature a shopping list function, which will enable users to create and manage their own personalized shopping lists.

The following SMART objectives will be achieved by the end of this assignment:

- Specific: Develop a mobile application that provides information(prices) on groceries at major supermarkets.
- Measurable: Test the application with a group of users to gather feedback and measure its effectiveness and usability.
- Achievable: Utilize API or other methods to access pricing information and incorporate it into the application.
- Relevant: Provide a valuable resource for individuals looking to save time and money on their grocery shopping.
- Time-bound: Complete the development of the application within the allotted time frame.

### 2.2. Scope

Scope areas	Explain scope
Business process / portfolio	Identifying the major supermarkets to be included in the app, based on factors such as market share and relevance to the target audience.
Geographical	Defining the target audience for the app, which are CGI international looking to save money in the Netherlands,
Organisational	An ambiguous software engineering student, alongside supporting employees from CGI.
Application	Outlining the features and functionality of the app, such as the ability to compare prices, create shopping lists, and receive notifications about deals and promotions.
Information	Determining the types of information to be included in the app, such as pricing data, promotions(discount), and name of store.

Table 1-Scope of project

### 2.3. Out of Scope

This project will not be delivering the following.

- Providing supermarkets outside the Netherlands
- Hosting the application on a server
- Developing a comprehensive database of all grocery products, including nutritional information and ingredients.
- Developing a loyalty program or reward system for users of the app.

## 2.4. Project team

The Project Management Team Structure consists of the following:

Name(s)	Role
Wesley Notten	Business Executive
Jelle Baede	Project Owner /User
Denis Gorianin	Supervisor/ Coach
Oussama El-Hajoui	Supervisor/ Coach

Table 2-Team

## 2.5. Project Management

Throughout the development of the Smart shopper application, an agile approach will be followed, a daily stand up and a 2-week sprint breaking down the project into smaller, more manageable tasks(use cases). This will make it easy to focus and track progress(Technology: Trello) as well as a responsive approach to changes and challenges that arose during the project. These will be done together with 4 interns and 2 coaches (Oussama El-Hajoui, Denis Gorianin).

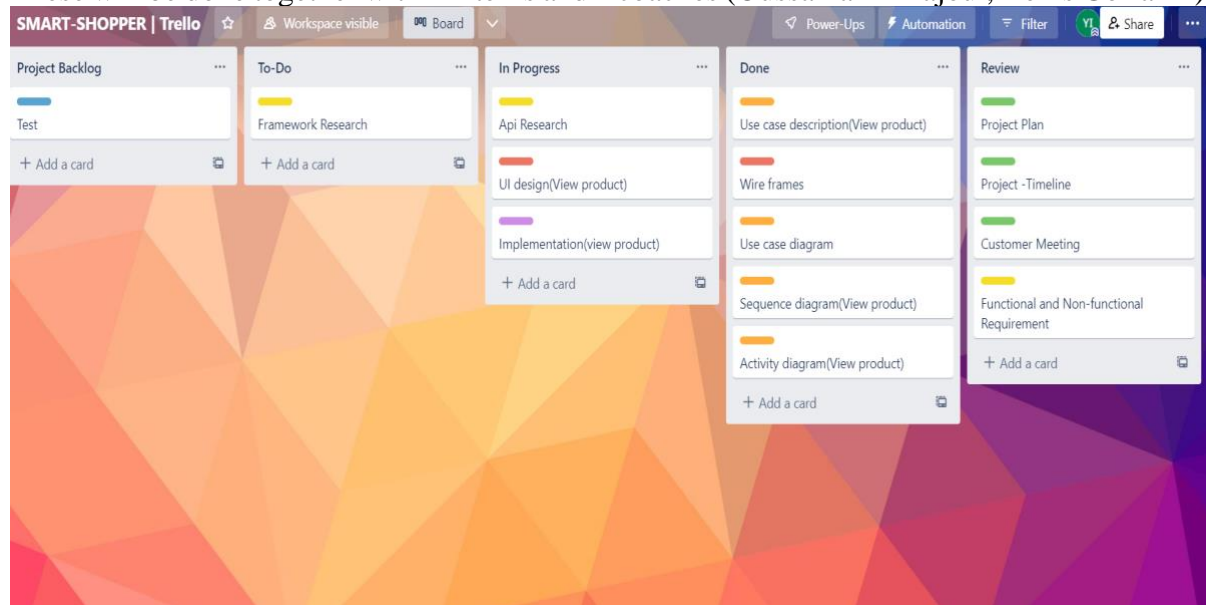


Figure 1-Personal Trello board

At the start of a sprint tasks will be prioritized using "planning poker," a technique that involved assigning numerical values to tasks based on complexity. This helped to focus on tasks following the MoSCoW principle which stands for must have, should have, could have, and would like to have.

At the end of each sprint, a sprint retrospective will be held to review progress and identify areas for improvement. This includes reflecting on what had gone well during the sprint, as well as identifying any challenges or issues that had arisen and developing strategies to address them.

Regular progress reports will also be provided to the project steering committee (PSC) at the end of each sprint. These reports included an overview of completed tasks during the sprint, any challenges or issues that had risen and how they were addressed, and an update on overall progress, including key milestones and upcoming challenges or deadlines.

## 2.5 project schedule

Milestones	Phase	Planned date	Major Deliverables
M1	Project Preparation	01-02-2023 To 01-03-2023	-Business case -Project charter -Project plan -SRS -WBS -Use case diagram -Activity diagram
M2	Design	01-03-2023 To 15-03-2023	-UI design -Wireframes -Relational model
M3	Realisation	15-03-2023 To 01-04-2023	-API Research -Database schema -Business logic and frontend
M4	Final Preparation	01-04-2023 To 15-05-2023	-Requirement tractability Metrix is fulfilled -Minimum viable product is ready
M5	Go live and Support	15-05-2023 To 01-06-2023	Working product
M7	End of Project	01-06-2023 To 15-06-2023	-Project Report -Project presentation

Table 3-Project Schedule



### 3. Analysis

The analysis phase of the Smart Shopper project consisted of several steps to establish the functional and non-functional requirements of the application. This chapter will outline the process used to collect these requirements, identify use cases, and create the required diagrams and tables.

#### 3.1 Functional and Non-functional requirement

To initiate the analysis phase, the customer was consulted to outline their needs and preferences for the Smart Shopper application. During this stage, both functional and non-functional requirements that needed to be addressed in the application's design was discussed and noted in a tabular format which can be found in the table below.

Functional requirements refer to features and capabilities that the application must possess, while non-functional requirements are constraints that impact the application's quality. For instance, functional requirements for the Smart Shopper application comprised the ability to search for products and compare prices, while non-functional requirements included security and usability considerations.

Title	Search
ID	REQ-F-1
Priority	5
Type	Functional
Description	The ability for users to search for specific grocery items
Title	Filtering and sorting
ID	REQ-F-2
Priority	3
Type	Functional
Description	The app should allow users to filter and sort search results based on factors such as store name, price, and promotions.

Title	Information on products
ID	REQ-F-3
Priority	4
Type	Functional
Description	The app should provide information on prices and at major supermarkets(at least 5 stores), with regular updates

Title	Personalization
ID	REQ-F-4
Priority	4
Type	Functional
Description	The app should allow users to create profiles and save favorite items.

Title	Integration
ID	REQ-F-5
Priority	4
Type	Functional
Description	The app should integrate with the systems and APIs used by major supermarkets to manage prices and promotions.

Title	Usability
ID	REQ-N-1
Priority	5
Type	Non-Functional
Description	The app should be easy to use and navigate, with a visually appealing and responsive user interface.

Title	Operating system
ID	REQ-N-2
Priority	5
Type	Non-Functional
Description	The app should be available to both android and IOS users.

Title	Performance
ID	REQ-N-3
Priority	4
Type	Non-Functional
Description	The app should be fast and responsive.

Title	Scalability
ID	REQ-N-4
Priority	4
Type	Non-Functional
Description	The app should be able to scale up to meet increased demand as the user base grows.current audience target 100 - 1000

Title	Security
ID	REQ-N-5
Priority	3
Type	Non-Functional
Description	The app should implement appropriate security measures to protect user data and ensure the privacy and confidentiality of user information.

Title	Reliability
ID	REQ-N-6
Priority	3
Type	Non-Functional

Description	The app should be reliable and provide consistent and accurate information to users.
-------------	--

Title	Maintainability
ID	REQ-N-7
Priority	4
Type	Non-Functional
Description	The app should be designed with maintainability in mind, with a clear separation of concerns and well-documented code.

### 3.2 Use Case Identification

After the requirements were identified, the next step was to determine the use cases for the application. Use cases refer to how a user will interact with the application to perform specific tasks. Several use cases were identified, including:

- Search for products
- Compare prices
- View product details
- Add items to Wishlist
- Create and manage user account

### 3.3 Use Case Diagram

The use cases identified earlier were represented in a use case diagram, which served as a visual depiction of the actors, use cases, and their relationships. This diagram was useful in defining the application's scope and served as a foundation for subsequent development efforts.

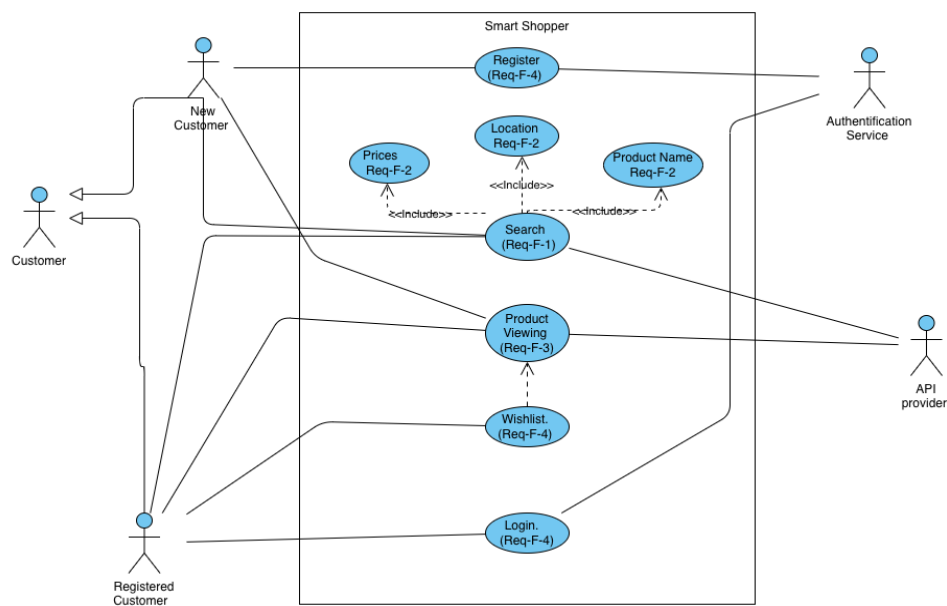


Figure 2- Use case diagram

### 3.4 Use Case Description Table

To provide more detailed information about each use case, a use case description table will be created. These table contained information such as the actors involved, preconditions, postconditions, basic flow, and alternative flow for each use case.

<b>Name</b>	View products
<b>Actor</b>	Customer
<b>Description</b>	Actor is able to view main products at the home page
<b>Pre-condition</b>	N/A
<b>Scenario</b>	<ol style="list-style-type: none"> <li>1. The actor enters to the app.</li> <li>2. The system displays a home page, displaying all the main product available</li> </ol>
<b>Result</b>	Products displayed on the home page
<b>Exceptions</b>	None
<b>Extensions</b>	None

Table 4-Use case description(View product)

### 3.5 Activity Diagrams

In the context of the Smart Shopper project, activity diagrams were created to depict the flow of actions in each use case.

Each activity diagram consists of a set of activities, which represent the actions that are performed in the use case, and transitions, which represent the flow of control between activities. An activity diagram can be thought of as a flowchart, showing the steps in a process and the order in which they occur.



Figure 3-Activity diagram(view product)

### 3.6 Sequence Diagrams

Sequence diagrams were utilized to model the interactions between the application components. For instance, a sequence diagram was created to depict the flow of events when a user view a product. This diagram displayed the user interface component sending a view request to the controller, which in turn sent a query to the database. The database then returned the queried items to the controller, which passed them back to the user interface for display to the user.

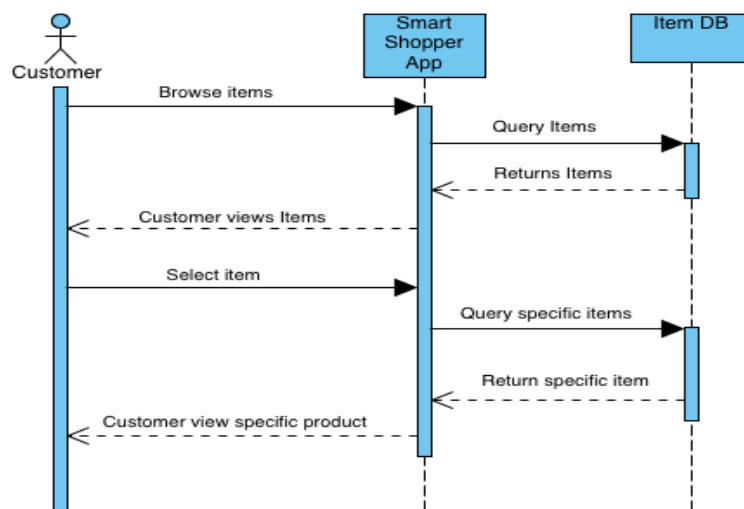


Figure 4-Sequence diagram(View Product)

Sequence diagrams are useful in identifying potential bottlenecks or areas for optimization in the system. They ensure that all components work together correctly and that the system functions as intended.

Developers, architects, and other stakeholders can collaborate in creating sequence diagrams. This process can help identify any possible issues early in the development stage and make the system more robust and efficient.

## 4. Design

The design phase of the project involved the creation of both wireframes and UI design. Wireframes were used to provide a low-fidelity representation of the user interface and layout, while UI design were used to provide a more high-fidelity representation of the visual design and branding.

### 4.1 Wireframes

The wireframes were created using visual paradigm to provide a basic layout for each page of the application, including the home screen, search results screen, and product detail screen. These wireframes were designed to provide a clear and simple user interface that would be easy to navigate and understand.

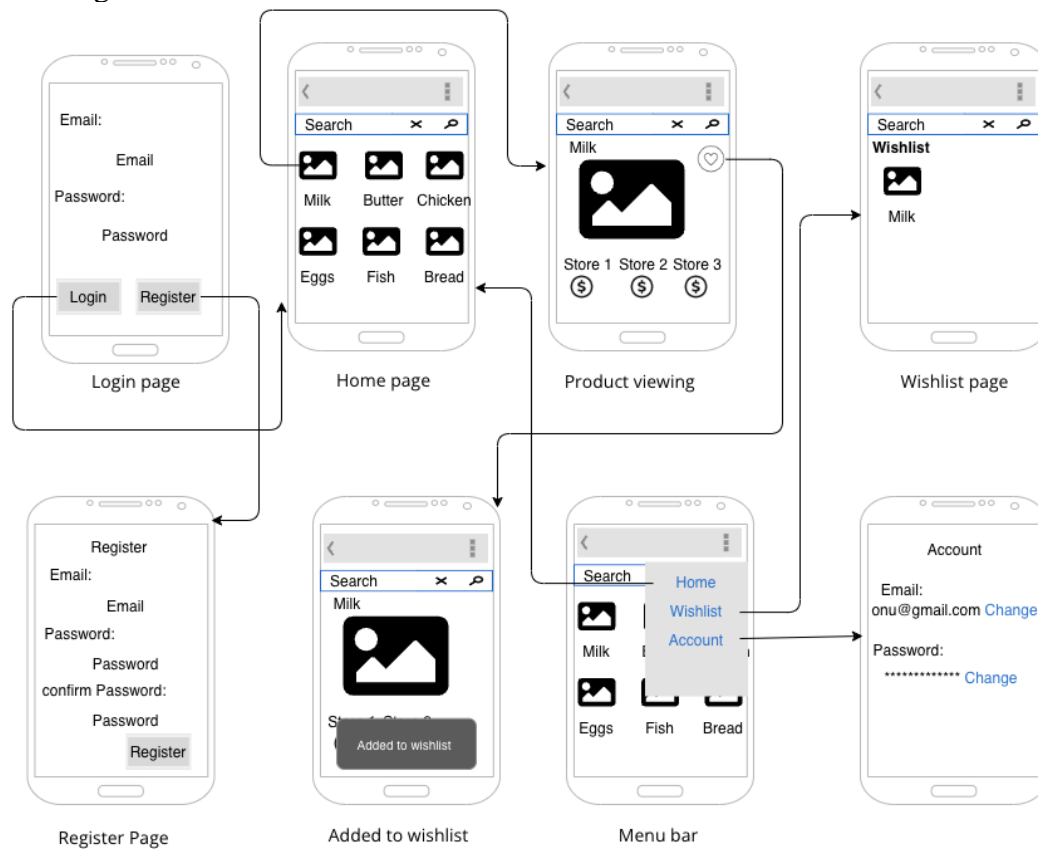


Figure 5-Wireframes

## 4.2 Ui design

UI diagrams were then created to provide a more detailed representation of the visual design of the application. These diagrams included information on typography, color schemes, and branding elements such as logos and icons. The UI diagrams were used to ensure consistency and coherence throughout the visual design of the application.

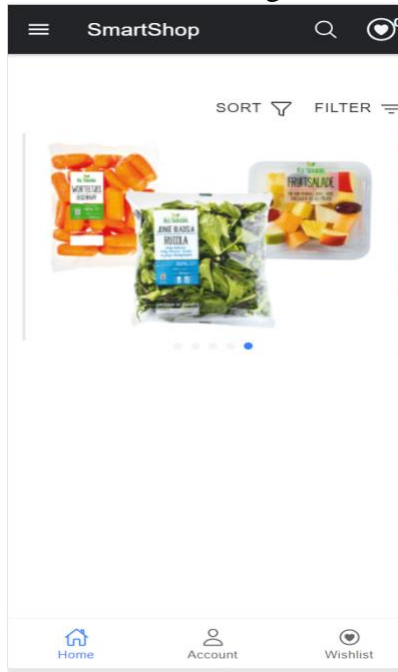


Figure 6-Ui design(view product)

UI design is an iterative process, and it's essential to involve users and stakeholders in every step of the design process to ensure the final product meets their needs and expectations. A well-designed UI can help to improve user engagement, increase user satisfaction, and ultimately drive the success of the application.

## 5. API Research

### 5.1 Context

To commence with the implementation of the project, it is crucial to conduct extensive research into available APIs and web scraping techniques that can offer data on products at various grocery stores in the Netherlands. APIs provide a more reliable and streamlined approach to acquire data, whereas web scraping offers more customization and flexibility.

The primary aim of this research is to identify APIs and web scraping techniques that can provide data on products, promotions, and prices at major grocery stores in the Netherlands. This data will be utilized to fill the database of the Smart Shopper application and offer users accurate and up-to-date information on products and prices.

This chapter explores different API options and web scraping techniques while evaluating their suitability for the project. The advantages and disadvantages of each approach are also discussed to provide recommendations on the best method(s) to use based on the research findings.

### 5.2 Goal

The goal of this research is:

1. Identify available APIs that provide data on different products at several grocery stores in the Netherlands.
2. The research will explore the possibility of generating data through web scraping.

### 5.3 Methodology

The methodology used to conduct the API research involved several steps.

Firstly, the data requirements for the Smart Shopper application were identified, specifically the need for up-to-date information on prices and promotions at major supermarkets in the Netherlands.

Furthermore, an extensive search was conducted for available APIs that provide the necessary data. This involved searching online directories of APIs, such as ProgrammableWeb and RapidAPI, as well as conducting targeted searches on search engines and social media platforms.

The APIs gathered were evaluated based on several criteria, including the availability and accuracy of data, ease of integration, cost, and support.

A comparative analysis of the APIs that met the selection criteria was conducted, to determine which one was the most suitable for the Smart Shopper application.

Lastly, the possibility of generating our own data through web scraping was considered, in case the available APIs did not meet the required standards.



Overall, the methodology used in this API research was rigorous and thorough, ensuring that the Smart Shopper application will have access to the most reliable and relevant data from the selected API.

#### 5.4 API Result

In this chapter, the results of the research into available APIs for data acquisition for the Smart Shopper application will be presented. The goal was to identify the most suitable API(s) for retrieving data on products, prices, and promotions at major grocery stores in the Netherlands.

Four different data providing APIs were evaluated, namely:

- ShoppingScraper API
- Superscanner API
- Checkjebon API
- Websrapping.Amsterdam API

To evaluate each API, the following criteria were considered:

- Data format: In what format or formats are data explorable ?
- Data coverage: Does the API provide data on a wide range of products and stores?
- Cost: What are the costs associated with using the API?

<b>API</b>	ShoppingScraper
<b>Description</b>	ShoppingScraper provides a variety of real-time endpoints for obtaining various types of data, such as product prices, specifications, and images.
<b>Link</b>	<a href="https://shoppingscraper.com/">https://shoppingscraper.com/</a>
<b>Type of data format</b>	CSV or JSON
<b>NL grocery stores list</b>	Albert Heijn Jumbo
<b>Price</b>	<p>There are three different prices:</p> <ol style="list-style-type: none"> <li>1. Hobby: 49 EUR / Month</li> <li>2. Startup: 129 EUR / Month</li> <li>3. Business: 349 EUR / Month</li> </ol> <p>Link to the detailed explanation of the prices:  <a href="https://shoppingscraper.com/pricing">https://shoppingscraper.com/pricing</a></p>

<b>Pros</b>	<ol style="list-style-type: none"> <li>1. The data extracted from various stores has the same structure.</li> <li>2. A shopping scraper that is designed to avoid captchas and other security measures in order to encounter fewer disruptions or obstacles when gathering data.</li> <li>3. The ability to request scrapping for other stores</li> </ol>
<b>Cons</b>	<ol style="list-style-type: none"> <li>1. It is not available for free.</li> <li>2. It is not possible to request that other stores be scrapped for the cheapest plan.</li> </ol>

<b>API</b>	Superscanner
<b>Description</b>	Superscanner collects price, promotion, and product information for over 4,000,000 products from more than 90 retailer websites in seven countries including The Netherlands.
<b>Link</b>	<a href="https://superscanner.nl/">https://superscanner.nl/</a>
<b>Type of data format</b>	Tabular format
<b>NL grocery stores list</b>	Albert Heijn Jumbo Makro Dirk Nettorama PicNic Plus Spar Hoogvliet
<b>Price</b>	Not indicated
<b>Pros</b>	<ol style="list-style-type: none"> <li>1. Covers a wide range of grocery stores in the Netherlands, including prices, promotions, and product information.</li> </ol>
<b>Cons</b>	<ol style="list-style-type: none"> <li>1. There is no trial period available to observe the platform or the data retrieved.</li> <li>2. The data format is not appropriate for extraction</li> </ol>

<b>API</b>	Checkjebon
<b>About</b>	Prices in supermarkets change frequently; Checkjebon.nl updates them daily and tells where to find the best deals on your groceries. Furthermore, Checkjebon.nl is an open source project, and the supermarket data used in it can be freely reused in other projects.
<b>Link</b>	<a href="https://github.com/supermarkt/checkjebon/blob/main/README.md">https://github.com/supermarkt/checkjebon/blob/main/README.md</a>
<b>Type of data format</b>	JSON
<b>NL grocery stores list</b>	<a href="#">AH</a> <a href="#">ALDI</a> - limited assortment due to not all products being available online. <a href="#">Coop</a> <a href="#">DekaMarkt</a> - Note; limited assortment due to not all products being available online. <a href="#">Dirk</a> <a href="#">Hoogvliet</a> <a href="#">Jan Linders</a> <a href="#">Jumbo</a> <a href="#">Plus</a> <a href="#">SPAR</a> <a href="#">Vomar</a>
<b>Price</b>	Free to use
<b>Pros</b>	<ol style="list-style-type: none"> <li>1. It is free to use.</li> <li>2. Covers many grocery stores in the Netherlands.</li> <li>3. The data is frequently updated.</li> </ol>
<b>Cons</b>	<ol style="list-style-type: none"> <li>1. It does not include product descriptions only store name, price and discounts.</li> <li>2. There is only one JSON file for all stores</li> </ol>

<b>API</b>	Webscrapping
------------	--------------

<b>Description</b>	Webscraping offers scrapping services to get data from the internet. They focus on gathering the correct data for price comparison, analysis, research, and monitoring.
<b>Link</b>	<a href="https://webscraping.amsterdam/supermarkt-prijzen-data/">https://webscraping.amsterdam/supermarkt-prijzen-data/</a>
<b>Type of data format</b>	EXCEL
<b>NL grocery stores list</b>	Albert Heijn Jumbo Coop Lidl Plus Ekoplaza Dirk
<b>Price</b>	Small: 450 EUR Medium: 550 EUR Large: 750 EUR For more detail: <a href="https://webscraping.amsterdam/pricing-webscraping/">https://webscraping.amsterdam/pricing-webscraping/</a>
<b>Pros</b>	<ol style="list-style-type: none"> <li>1. Provides high-quality and accurate store data.</li> <li>2. The ability to request necessary data based on the project's goal.</li> <li>3. Coverage of many grocery stores in the Netherlands</li> </ol>
<b>Cons</b>	<ol style="list-style-type: none"> <li>1. Very costly</li> <li>2. There is no option to obtain the data in JSON format.</li> </ol>

Based on the evaluation, the Checkjebon API was found to be the most suitable for the project. It provides data on products, prices, and promotions at major grocery stores in the Netherlands, and has a wide coverage of stores and products. Additionally, the API is relatively easy to use and integrates well with the application. The cost associated with using the API is reasonable and within the project's budget (Free).

The other APIs evaluated had some limitations that made them less suitable for the project. For example, the ShoppingScraper API had limited coverage of grocery stores in the Netherlands, and the Webscraping.Amsterdam API provided data in a format that was not ideal for the application as well also being the most expensive API.

## 5.5 Web-scraping

This chapter will explore web scraping as a possible method for data acquisition in the Smart Shopper application. Web scraping involves extracting data from websites by parsing HTML and other web page content. It is a customizable approach that can retrieve data from various sources, including grocery store websites.

To use web scraping in the Smart Shopper application, a script must be written to automatically retrieve product information from the websites of major grocery stores in the Netherlands. This can be accomplished using programming languages such as Python or JavaScript and libraries like BeautifulSoup or Selenium. The script must be designed to navigate the store website, identify relevant information, and store it in a format usable by the application.

```
productScraper.py > ...
1  import requests
2  from bs4 import BeautifulSoup
3  import json
4
5  # website URL
6  url = 'https://www.ah.nl/producten/aardappel-groente-fruit/groente'
7
8  html_text = requests.get(url).text
9  soup = BeautifulSoup(html_text, 'xml')
10
11 #Returns all product associated with the class name
12 products = soup.find_all('article',
13                           class_='product-card-landscape_root_LXes- product-grid-lane_gridItems_BB4h')
14
15 print(products)
16
```

Figure 7-Example of webscraping using python

Web scraping provides some advantages over using an API. It allows for more flexibility in the type and amount of data that can be retrieved. APIs often have limitations on the data they provide or charge extra fees for more extensive data. Web scraping, however, can retrieve any data publicly available on the website. Additionally, web scraping is often more cost-effective since many websites provide their data for free. Finally, web scraping can provide more accurate and up-to-date data since it retrieves data directly from the source website.

However, web scraping also has some disadvantages. Firstly, it can be a complex and time-consuming process, especially if the website structure changes frequently. Secondly, web scraping can put a strain on the website server and result in the website blocking the IP address of the scraper. Thirdly, web scraping can be less reliable than using an API since it relies on the structure of the website remaining consistent.

In conclusion, web scraping is a possible alternative to using an API for data acquisition in the Smart Shopper application. It offers more flexibility and customization options but requires more technical expertise and can be less reliable than using an API. Further evaluation of the feasibility of using web scraping and comparing it with using the Supermarket API will be performed in future stages of the project.

## 5.6 Advantages and Disadvantages.

### 5.6.1 API

Advantages	Disadvantages
APIs are often designed for developers, making them easier to use and integrate into applications.	APIs can be expensive, especially for extensive or up-to-date data.
APIs provide a more streamlined and reliable method of data acquisition compared to web scraping.	APIs may have limitations on the data they provide or charge extra fees for more extensive data.
APIs often have a wider coverage of stores and products than web scraping, which can save time and effort in data acquisition.	APIs may have restrictions on the frequency of data retrieval or the amount of data that can be retrieved in a given period of time.

### 5.6.2 Web scraping

Advantages	Disadvantages
Web scraping allows for more flexibility and customization in the type and amount of data that can be retrieved.	Web scraping can be a complex and time-consuming process (amount of data to scrape), especially if the website structure changes frequently.
Web scraping can be more cost-effective since many websites provide their data for free.	Web scraping can put a strain on the website server and can result in the website blocking the IP address of the scraper.
Web scraping can provide more accurate and up-to-date data since it retrieves data directly from the source website.	Web scraping can be less reliable than using an API since it relies on the structure of the website remaining consistent.

## 5.7. Research conclusion

After conducting research into data acquisition methods for the Smart Shopper application, it has been concluded that the Checkjebon API is the preferred option over web scraping. The main purpose of this research was to investigate the most suitable method for acquiring data for the Smart Shopper application, which required product information from at least five major grocery stores in the Netherlands.

Through investigation, it was found that web scraping offers more flexibility and customization options but would require a significant amount of time to gather data from multiple sources. Given the functional requirements of the application and the time frame of the graduation internship, it was determined that using the Checkjebon API would provide a more efficient and effective solution to obtain real-time data on products, prices, and promotions from major grocery stores in the Netherlands.

The main findings of the research demonstrate that using the Checkjebon API is a more time-effective solution that enables the allocation of more resources towards designing and implementing features that enhance the user experience, such as personalized recommendations and shopping lists.

The implications of the findings for the field of study and the broader context in which the research was conducted suggest that data acquisition methods should be carefully considered based on their efficiency, reliability, and scalability. Future research could investigate other data acquisition methods or compare the effectiveness of different methods for specific applications.

In practical terms, the use of the Checkjebon API has significant implications for the development of the Smart Shopper application, allowing for the building of a robust and functional application within the given time frame.

In conclusion, the decision to use the Checkjebon API over web scraping was made based on the time it would take to gather data from multiple sources and the available time frame for the graduation internship. This approach enables the building of a robust and functional Smart Shopper application and provides opportunities for further research into data acquisition methods and their effectiveness for specific applications.

## 6. Implementation

This chapter covers the current status on implementation of the Smart Shopper application, including the choice of development frameworks, programming languages, and other technical aspects of the implementation.

### 6.1 Choice of development framework

The non-functional requirement of the Smart Shopper application was that it should be accessible on both Android and iOS platforms. After researching various hybrid frameworks, it was determined that Ionic and Angular were the best choice for developing the application.

Ionic is a cross-platform mobile development framework that is open-source and uses Angular for building mobile applications. Angular is a JavaScript framework that is popularly used for building single-page applications. The combination of Ionic and Angular provides an efficient solution for developing hybrid mobile applications.

The decision to use Ionic and Angular was based on experience working with the framework, their ease of use and extensive collection of frameworks and plugins that make mobile app development much more accessible. Furthermore, the Angular framework provides a modular structure that is scalable, making it easier to maintain and update the application in the future.

### 6.2 Use case 1 (view product) implementation

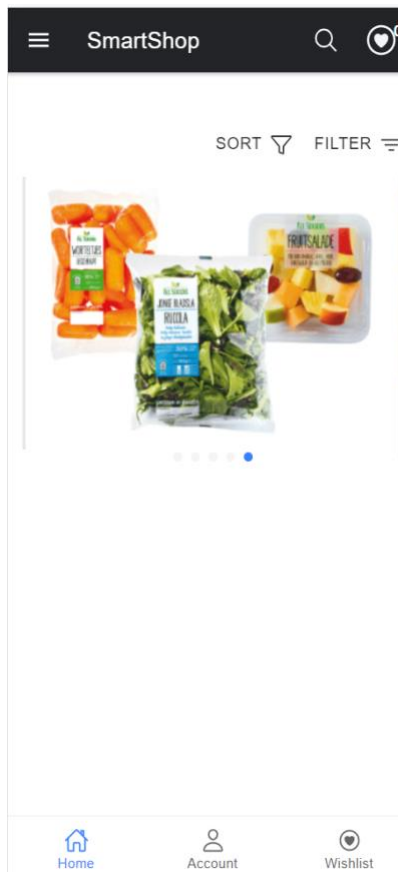
The View Product use case is a critical feature of the Smart Shopper application, which allows users to view product information such as store, prices, and images. Currently, the View Product use case is in the implementation phase.

The implementation of the View Product use case involves designing a user interface (UI) that displays top or discounted products on the home page of the application. This entails setting up a database that stores product details such as store name, price, and images.

To ensure that users can easily view and interact with product information, the UI is being designed with a user-friendly layout that provides relevant details such as price, discounts, and ratings at a glance. Additionally, the product images will be optimized to load quickly and in high quality, providing a visually appealing experience for users.

The database will be populated with product details and images from the API of choice, which will be displayed on the home page using a responsive design that adjusts to fit the screen size of the user's device. The UI will also allow users to sort and filter products by category, search for specific products, and view related products as shown in the diagram below.





```
<ion-slides [options]="sliderOptions" class="ion-padding" pager style="margin-top:4.2rem">
  <ion-slide *ngFor="let slide of sliderImages">
    <img [src]="slide+'.png'" alt="images">
  </ion-slide>
</ion-slides>

<ion-grid>
  <ion-row>
    <ion-col>
      <ion-list>
        <ion-item *ngFor="let prod of displayedList" class="item-entry" detail routerLink="/tabs/tab1/prod-detail/{{prod.id}}">
          <ion-thumbnail>
            <ion-img [src]="prod.images[0].src" alt="{{prod.name}}"></ion-img>
          </ion-thumbnail>
          <ion-label>
            <h3 class="item-name">{{prod.name}}</h3>
            <ion-text...>
          </ion-label>
        </ion-item>
      </ion-list>
    </ion-col>
  </ion-row>
</ion-grid>
</ion-content>
```

Figure 8-Home page(View product)

Overall, the implementation of the View Product use case is essential to the success of the Smart Shopper application as it enables users to make informed purchasing decisions by providing them with relevant information.

## 7.conclusion

Over the course of the graduation internship, significant progress has been made towards the development of the Smart Shopper application. The project began with a requirement gathering phase, during which functional and non-functional requirements were identified. The use cases were then identified and use case diagrams were created to provide a visual representation of the actors and relationships between them. Detailed use case descriptions were also created to provide a more comprehensive understanding of each use case. API research was also conducted to determine the most suitable method for acquiring data for the Smart Shopper application.

Through research, it was concluded that the Checkjebon API is the preferred option over web scraping for data acquisition. This decision was based on the functional requirements of the application and the time frame of the internship.

Some issues were faced during the implementation of the view product use case, such as integration with the Checkjebon API and ensuring the smooth flow of data. However, these were addressed through collaboration with other team members and consultation with the client.

Moving forward, The development of the Smart Shopper application will continue with the implementation of other use cases, including search, add to wishlist, filter, sort, and account creation. The search use case will enable users to search for products based on keywords, stores, or categories. The add to wishlist use case will allow users to add products to their wishlist, which they can refer to later when making purchasing decisions. The filter and sort use cases will enable users to refine their search results based on various criteria such as price, brand, or availability. Finally, the account creation use case will allow users to create an account and have a personal wishlist.

To ensure that the development of these use cases is aligned with the client's requirements and expectations, regular consultation with the client will be necessary. This will involve gathering feedback, addressing any concerns or issues, and providing updates on the progress of the project.

## 8. Reflection

Reflecting on the entire Smart Shopper project, valuable experience and knowledge about software development and working were gained. The internship has been an excellent opportunity to apply acquired academic studies knowledge and gain practical experience in a real-world setting.

During the first few weeks of the internship, focus was on requirement gathering, which was a critical phase in the development process. This phase helped understand the importance of communication and collaboration with the client to identify and define the functional and non-functional requirements for the application. Relevant questions, analysis of client needs, and identification of potential challenges and limitations were learned.

After defining the requirements, the project moved onto analyzing and designing the application, which was an exciting and challenging phase. Use case diagrams were created to visualize the actors and relationships between them, and then detailed use case descriptions were developed. This phase helped to understand the importance of creating a robust design that captures all the requirements and specifications of the application.

One of the most significant challenges during the development process was the research into data acquisition methods for the Smart Shopper application. The suitable method for acquiring data from various grocery stores in the Netherlands was identified. Through investigation, it was determined that using the Checkjebon API would provide a more efficient and effective solution to obtain real-time data on products, prices, and promotions. This decision was based on the functional requirements of the application and the time frame of the internship.

During the implementation phase, the responsibility was to implement the view product use case. It was a challenging task to integrate with the Checkjebon API and ensure the smooth flow of data. However, the challenges were overcome through collaboration with other members and consultation with the client.

Throughout the internship, effective communication, time management, and resource management were learned. Practical experience in software development was gained, and various software development methodologies and techniques, such as Agile development, requirement gathering, design, and implementation were applied.

In conclusion, the Smart Shopper project has been a valuable learning experience that helped develop skills and knowledge in software development. The opportunity to work on this project and learn from experienced professionals in the field is appreciated. Continuing to work on the project and applying the skills and knowledge gained from this internship to future projects is looked forward to.

## 9. Reference

- Pierre, P. 2022, 'Making Use of APIs in Your Front End', The Startup, Medium, viewed 11 January 2023, <https://medium.com/swlh/making-use-of-apis-in-your-front-end-c168e343bea3>.
- Smith, A. 2022, '6 APIs Frontend Developers Should Know', Bits and Pieces, Medium, viewed 11 January 2023, <https://blog.bitsrc.io/6-apis-frontend-developers-should-know-3a94815c53dd>.
- Superscanner n.d., Realtime prijs- en productinformatie, viewed 11 January 2023, <https://superscanner.nl/>.
- S n.d., 'GitHub - supermarket/checkjebon: Dutch supermarket prices and comparison tool', GitHub, viewed 11 January 2023, <https://github.com/supermarkt/checkjebon>.
- ShoppingScraper n.d., 'ecommerce price scraper', viewed 11 January 2023, <https://shoppingscraper.com/>.
- Sites, F. I. n.d., 'Prijzen supermarkten automatisch linken met MS Excel? | Digital Corner (DIG) - forum.fok.nl', FOK!Forum, viewed 11 January 2023, <https://forum.fok.nl/topic/2655492/1/50>.
- Webscraping Amsterdam 2023, 'Supermarkt Prijs en Product informatie voor uw bedrijf', viewed 11 January 2023, <https://webscraping.amsterdam/supermarkt-prijzen-data/>.
- Wikipedia contributors 2023, 'MIT License', Wikipedia, viewed 11 January 2023, [https://en.wikipedia.org/wiki/MIT\\_License](https://en.wikipedia.org/wiki/MIT_License).