Universität Bremen

Developing a Mobile Application for Product Finding in Digital Twins of Retail Stores using Augmented Reality

Entwicklung einer mobilen Anwendung zur Produktfindung in digitalen Filialzwillingen mittels Augmented Reality

Bachelor Thesis

Toni Aleksandrov Kozarev

Supervisor:	1. Prof. Dr. Michael Beetz
	2. Dr. Thomas Barkowsky
Advisor:	Michaela Kümpel

Declaration of Authorship

I, Toni Aleksandrov Kozarev, declare that this thesis titled, „*Developing a Mobile Application for Product Finding in Digital Twins of Retail Stores using Augmented Reality*“ and the work presented in it are entirely my own. I confirm that this work was done for a research degree at the University of Bremen and I have not used any sources other than those listed in the bibliography and identified as references in „Used Resources“ section.

Bremen, October 11, 2022

Toni Aleksandrov Kozarev

Abstract

With the progress of technologies, more powerful mobile devices are created every year. Nowadays, almost all of the mobile phones support an *Augmented Reality (AR)* technology. Variety of AR applications are used to assist humans in the fulfillment of daily tasks such as visualizing interior layout for a new apartment, showing tattoos on a user's skin or navigating them while they are shopping for groceries or searching for a place in a huge building.

The aim of this thesis is to develop a *Mobile Augmented Reality (MAR)* Android application for indoor navigation to help customers find a particular product shelf in a small and closed place. This includes creating a website where the user can see all available products divided by categories in a table. The user can access the website from the mobile application and search for a certain product. The application is using a *digital twin* of the retail store to help users find the closest path from their location to the shelf where the searched product is placed. A digital twin is a virtual representation of a real object or place, in this case it is a 3D map of the store. Knowledge about all available products in the retail environment and their shelf location is obtained from a *knowledge graph*, which is a collection of organized and linked data visualized as a graph.

In the busy world that we live today, it is really difficult for people to find enough time and energy to search for the best products in a store every time they go shopping. It is shown that this AR application is less time-consuming than customers alone searching the right product for their needs. Additionally, users can be more efficient while using the application because they can plan which product to take next in their shopping cart so that they do not lose too much time walking around in the store. This can also improve the quality of work done by the workers who are not going to worry about the needs of the customers.

Lastly, it is shown that the majority of people prefer using an AR application for indoor navigation instead of a robot assistant to guide them in a retail environment.

Key words: Android application, AR Foundation (ARCore SDK), Digital Twin, Indoor Navigation, Knowledge Graph, Mobile Augmented Reality (MAR), NavMesh tool, Product Finding, Quick Response (QR) code recognition, SLAM (Simultaneous localization and mapping), Unity Engine, Website application

Contents

1 Introduction

Nowadays, the usage of AR applications is advancing rapidly and a larger number of people are integrating them into their day-to-day life [9]. This development is due to the growing interest in technology, as it has become next to impossible to imagine a person living in the developed parts of the world that does not use a mobile device for various purposes in their everyday activities. Thanks to the help of Mobile Augmented Reality (MAR), modern devices can easily present to the user 3D models or virtual information in real-time, thus facilitating an interaction with them and can be used in many use cases, such as Healthcare, Industry 4.0, Travel and Tourism, etc. [17]. Typically, *Augmented Reality* utilizes the camera of a device to show the user's real-world surroundings and to generate virtual objects in it, such as target destinations or a line which follows a path from one point to another [1].

As a result of AR technology's growing popularity, indoor navigation applications have also become more trendy [14]. Within the fast-paced world, a larger number of huge buildings are constructed every year. Thus, the need of using an indoor navigation in places such as shopping centers, airports and hypermarkets, is also on the rise [13]. Typically, such places provide people with maps at the entrance and on every level of the building. However, understanding and most importantly, remembering the maps can be very demanding and often confusing for visitors.

1.1 Motivation

The motivation behind carrying out this project is to not only develop an indoor navigation application to guide customers to a particular product shelf in a specific retail store with the usage of an AR technology but also to create a system that can be used in different shopping centers only by scanning a simple QR code to load the map layout which is designed by the help of a *digital twin*. A digital twin is a digital representation of the real world, for this thesis it will be used to portray a digital supermarket [7].

Moreover, this innovation can help the globalization of retail industry by allowing various stores to use the same application which can access the rapidly changing market inventories where the product information is stored and also increase the amount of customers in the places where it is implemented. Such an innovation is usually more straightforward and requires less time and effort on the part of visitors, as they don't need to comprehend complex maps, displayed on paper or to constantly rotate in a circle to determine their location. Additionally, workers in these places will not be distracted by people asking for directions and variety of products. This will improve the quality of their work and enable them to do more activities, like arranging products on different shelves or cleaning the place up, without being worried for the customers and their needs. The application can also be used by the employees to assist them organizing the products in the right shelves for them.

Furthermore, an AR application can significantly improve navigation in large indoor areas not just for leisure purpose, but also in the unfortunate case of emergencies, by allowing people to evacuate as quickly as possible. This can be accomplished, by the AR tracking application analyzing and providing users with the shortest and safest way out of the building, by navigating them to the closest emergency stairs or exit doors [12].

1.2 Problem statement

The idea of this thesis is to combine AR experience for finding a product in an indoor environment with the usage of digital twins for creating the maps for the retail stores and a knowledge graph for storing all the required information for the products from all of the stores at one place.

There are many similar applications (see Related Work) that achieve the idea above and can show customers the closest way to a destination which can spare them considerable amount of time and energy, by allowing them to reach the destination as fast and efficiently as possible. However, they are created only for a certain room or a building and use a separate database from where the data is accessed which make them not transferable, hence they cannot be used for different places. This thesis intends to provide a solution for these problems and in order to accomplish them, several conditions need to be met.

Essentially, for each retail store a digital twin map needs to be created. This will include all objects inside the store, such as floor, roof, walls and shelves. This will create a 3D visualization of the store which later can be used for the AR indoor navigation. A knowledge graph is used to obtain the information about the products presented in the website and their shelf placement. This term is used to describe a network of real-world entities such as objects, events, etc. and the relationship between them which looks like graph structure database [5]. The advantage of using a knowledge graph is that it can provide product information for each store and can be queried via API.

1.3 Approach

The system presented in this thesis permit a mobile application with the help of a website to perform the role of a shopping assistant for customers who need to search for a certain product in a retail store. The whole system is using sensors only from the mobile device and no other hardware is required for the application to work and the navigation to be accurate. The project consists of different components. One of them is a digital twin of the store used for a map in the application. Another one is the knowledge graph where the information for all products is stored. The store map and the product information are linked together but they can also be used separately when the location is changed by scanning a new QR code to load the next retail store the product data remains available. In order for a customer to interact with the application, an AR experience is implemented.

The Android application should work with different maps of stores and they can be loaded using a **Q**uick **R**esponse (**QR**) code recognition. Moreover, all products and their categories need to be displayed in a simple website. This website will allow the user to search for a specific product in a certain retail store and if the product is available the website will provide the user with the shelf number on which it is located. In this thesis the standardized interface for the product data is the NonfoodKG knowledge graph because it provides product information for all the existing products.

The contribution of this thesis is an AR Android application that is able to navigate a customer in a retail store to a shelf where the searched products are placed. The website created for searching shelves should be able to display all available products in the retail store and allow the customer to acquire information about the shelf location of the searched product. To evaluate the capabilities and limitations of the whole system, including the mobile application and the website application, a survey with a couple of questions was performed in the retail lab where multiple participants tested this system and a robot assistant system [2] to guide them in the right direction when looking for a product and at the end both systems were compared.

1.4 Thesis structure

This section provides an overview of the structure of the thesis. In Related Work is reviewed the literature that has influenced this paper. Afterwards, the tools and techniques used in this project are presented in Tools and Background Knowledge. They are followed by first main part Concept where the prototypes, system design, all frameworks and methods used in the paper are explained. Realization is the other main part where all components of the application and its functionality are presented. Subsequently, in Evaluation all of the data from the study is illustrated and analyzed. In the last Conclusion a summary of the paper is given and possible future work and errors in this study are discussed. All sources were last accessed on 04 October 2022.

2 Related Work

Over the last few years, Augmented Reality (AR) technology has become a viral topic in the digital world. There are many research papers which discuss different AR applications. One of the fields where AR is used a lot is indoor navigation. Several academic works focus on this subject and the following is an overview of some of those.

2.1 AR Smart Navigation System

In [14], the main goal is to find the best solution for expensive techniques like Wi-Fi Routers or BLE Beacons which are used to compute the indoor navigation in a closed place. This paper describes an easy, cost-efficient and accurate method for such an AR system using ARCore which utilizes Simultaneous Localization and Mapping (SLAM) technique. SLAM uses the camera and sensors like accelerometers and gyroscopes on the device to obtain data for the unknown surroundings of the user and his's location in it. In this research the NavMesh tool from Unity was used to define a map by choosing which area is walkable and which is restricted. This project utilizes also a QR code re-positioning. The advantage of this system is that it requires no external resources such as Internet or Bluetooth to work properly. The SLAM technique and ARCore SDK are doing the whole recognition of the location and updating the route while navigation is running. Most of the technologies used in the paper above are also used in this thesis. The system implemented in this paper has only one map created whereas the idea of the Android application developed in this work is that more digital twin maps can be loaded by scanning different QR codes.

2.2 Design of a Mobile Augmented Reality-based Indoor Navigation System

Design of a Mobile Augmented Reality-based Indoor Navigation System [13], as the title suggests, is another research based on developing an application to provide a solution for IPS and route planning in a huge university campus. To accomplish its goal, the project analyzes various pathfinding (A*, Dijkstra, etc.) and indoor positioning techniques, such as GPS, Wi-Fi-, BLE beacon, magnetic field signals, visible marker and etc.). The comparison of these positioning techniques shows which one is the most accurate, cost effective and robust for a large indoor environmental places. Most of the indoor positioning techniques above are not that accurate and have high cost or are time-consuming when used in large areas. Therefore, the application in the paper was developed using IndoorAtlas SDK, ARCore (for AR guidance for the calculated route using the built-in sensors in a mobile device to detect user's location), Android Studio IDE and a fusion of two techniques - Wi-Fi signals and magnetic field, which gave the best results for positioning technique in the enormous campus.

According to the paper above, the „Visible Marker“ technology is unreliable in large premises. The application in this work is developed for small environments, therefore the mentioned technology can prove to be useful in this situation. Furthermore, the paper specifies a number of sources, which can negatively affect the accuracy of the indoor positioning, such as light and weather. In the retail lab those factors are insignificant.

2.3 Personal and Intuitive Indoor Navigation System based on Digital Twin

In [8], the created indoor navigation system is using a digital twin of a building and QR codes attached in different places for position correction. The pathfinding algorithm used in this paper to calculate the shortest path is Dijkstra. The positioning techniques used in this paper are „static information“ (position fingerprint by Wi-Fi intensity and air pressure) combined with „dynamic information“ detected by sensors of the smartphone. The ARCore SDK is used to add an AR experience and show moving directions for the path from the entrance of the building to the destination.

In the following thesis, the Android application is created using Unity Engine and a tool called „NavMesh“ integrated in the IDE is allowing the developers to generate a walkable area in a map. This tool is using A* algorithm to calculate the path in the non restricted area from a start position to a destination. In the paper described above the pathfinding algorithm is Dijkstra and although the author is also applying ARCore SDK they are using different positioning techniques.

The application created for this thesis is also using a digital twin maps and ARCore SDK for the AR navigation but the pathfinding algorithm is different.

3 Tools

In this chapter will be presented which tools were used in creating both the website and the Android application for this thesis.

3.1 Protégé

The program „Protégé“ [6], which is a free, open-source platform (8.3), was utilized to understand the knowledge graph used in this paper. The application can open „**.owl**“ files and the user can look through the data inside them. As a result of using this software, it was planned how to divide the information that needs to be shown in the website. More details about the creation of the website will be given later on in (section 5.1).

3.2 Triply

Triply is a website on which different knowledge graphs are deployed. In this paper the integrated knowledge graph is NonfoodKG. This website allows the user to execute different SPARQL queries on the graph.

3.3 GRLC

GRLC [11] is a server which translates SPARQL queries from an API endpoint to Linked Data Web APIs (GRLC Github). In the GRLC website for the NonfoodKG knowledge graph are assign endpoints to which GET requests can be sent to acquire the necessary data.

3.4 Visual Studio Code

The IDE used for developing the website was „Visual Studio Code“ (8.3). It has many extensions which support multiple programming languages, like for example PHP and

JavaScript.

3.5 XAMPP

In order to create and test the PHP website locally, the XAMPP was used, which is a free and open-source cross-platform developed by Apache Friends (8.3).

3.6 Infinity Free

After the website is completed, it needs to be hosted so that users can access it over the Internet, in this case from the Unity Android application using a *web view* or from the browser of another device. Therefore, the hosting service of „Infinity Free" (8.3) is used, which is compatible with PHP websites. For the website to be live and the URL reachable from a browser, the user needs to issue a SSL and use it in the website. The free version of this service allows the issued SSL to function 3 months before it expires. After that, the website needs a new SSL or to renew the previous one.

3.7 Unity Engine and Visual Studio Community

Unity is a development platform for creating 2D and 3D multiplatform games and interactive experiences (8.3). Visual Studio Community is a powerful IDE used to develop programs in .NET and C# (8.3). In this project Unity was used for the implementation of the MAR Android application and Visual Studio was utilized to code the C# scripts for it. Unity is also very appropriate for the task because it has the AR Foundation package, which allows the user to work with AR features.

4 Background Knowledge

This chapter will provide the readers with the necessary background knowledge in order for them to understand easier the terms and details used in this thesis later on. At the beginning, the concept of knowledge graph is described. All the information about the products and their shelf location will be accessed from there. After that, it is important to explain what digital twins are and what their purpose is. In this work, they will be used to create a 3D map for the digital retail store. In order to implement in this project an AR experience, it was used a framework and it is necessary to explain how this framework works on a basic level and what is the technique behind it.

4.1 Knowledge Graph

A knowledge graph, visualized as a graph structure, is a network that represents real-world entities, such as objects, concepts, events or concepts and the relationships between them. This graph contains three main components - nodes, lines between the nodes and labels for the nodes and the lines. Any of the entities can be a node and the relationships between them are presented with the edges [5].

4.2 Digital Twins

A digital twin is a virtual object created to mirror an object from the real world. This object can consists of many components with various of sizes and each of them can have different functionality. This data is then replicated to the digital copy [4].

The object which needs a digital twin in this thesis is a map. This map has several walls, a floor and shelves in it. All of those components need to be reflected when creating the digital twin of the map as well their functionality. In this case, the floor is the only walkable area for the user in the map. The shelves and the walls should restrict the movement of the user.

4.3 ARCore SDK

The ARCore SDK is a platform created by Google for building AR experiences. The platform is using SLAM to estimate with the sensors of the mobile device where the phone is relatively located to the world around it. The inertial measurements from the device's IMU combined with the information from the camera can detect the position and orientation of the phone while the user is moving it. In comparison to Vuforia SDK the ARCore is free of charge and can be fully used to create a project with it.

The ARCore SDK in Unity 2020 and later is depricated and the new cross-platform framework used in Unity is AR Foundation which is available in Unity Package Manager and uses ARCore SDK to create AR applications.

5 Concept

In this chapter it will be illustrated the structure of the project. There are two applications that need to be developed in order to successfully achieve the goal of this thesis - a web application (see 5.1) and a mobile application (see 5.2).

5.1 Website for finding product shelf

For building the website PHP and JavaScript were used as programming languages, as well as HTML and CSS for the structure and layout of the webpage. In order to read the data from the NonFoodKG knowledge graph a library called „ARC2“ (see section 8.3) which supports SPARQL was used. With the help of it, the endpoint was accessed and all the information shown on the website was acquired. For the development of the website it was crucial to learn how knowledge graphs work and how to create SPARQL queries. With the understanding of these two fundamentals it was less complicated to select the required information from the ontology and use it later for the website. At the beginning, Protege helped a lot with visualizing the graph. In the picture below (see Figure 5.2) it is shown the hierarchy of the product class (highlighted item) which is the class that will be used to get all the data for the project. It can have up to four children branches in the hierarchy which are displayed on the table with four columns shown on the website. For example, in „*Products*“ there is a category „*Groceries*“ which has a subcategory „*Beverage*“ and this has a class called „*Alcoholic beverage*“, which again has subclasses such as „*Beer*“, „*Liqueur*“ or „ *Wine*“ (see Figure 5.1). Some of the categories can also have less than four children. In order to create the SPARQL query, which shows all available shelves in the retail lab, GRLC (see section 3.3) was used which has many examples of different queries. There is a query for searching the location of a detergent **/ShelfLocationForDetergent** which was similar to the query needed in this project. In order to gather the information of all available shelves in the retail lab a new query was created. A website called Triply was used to test if the new SPARQL query (see Figure 5.3) works properly.

#	Category	Under Category	Class	Under Class
7	groceries	beverage	alcoholic beverage	beer
17	groceries	beverage	alcoholic beverage	liqueur
30	groceries	beverage	alcoholic beverage	wine

Figure 5.1: Example for how all 4 categories of the „Product“ class are shown in the table

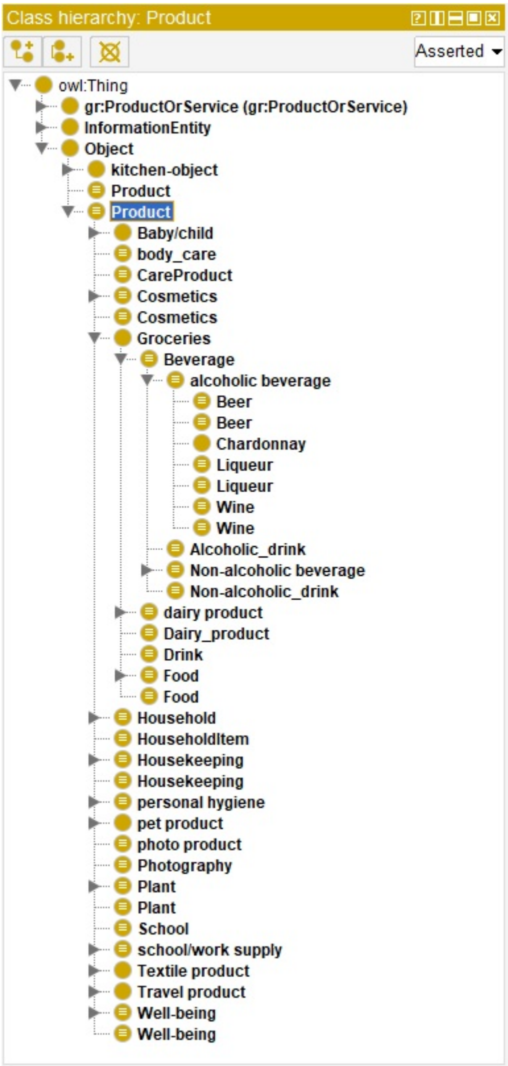


Figure 5.2: Protege

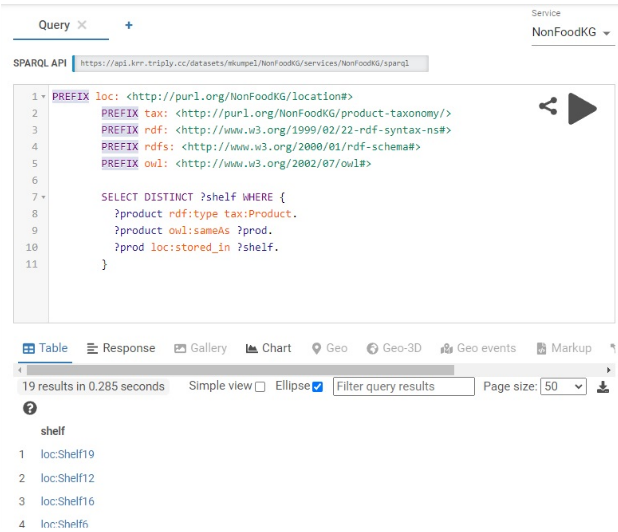


Figure 5.3: Example of SPARQL query using Triply

5.2 Android Application

In this section, the prototype of the Android application will be briefly illustrated and the concept of the whole system as well as all used resources will be described.

5.2.1 Prototype

A prototype was created only for the Android application because it is more complex than the website which consists of only one page with a table and a search bar. The prototype of the application is built in three screens. In order to navigate through the application it was necessary to create a main menu (see first screen in Figure 5.4). It was planned to contain three choices - „Scan QR-Code“, „Options“ and „Quit“. After selecting the first option the application should open the second screen shown in Figure 5.4 where the user can scan a QR code to load the map for the retail store. If the QR code is detected, it should allow the user to use the navigation (see third picture in Figure 5.4). There was no screen created for the „Options“ in „Main Menu“ because it was difficult to plan if it was needed. While developing the application, it was possible to follow the prototype example to almost full extend. There were some slight changes to the application in order to simplify the GUI and to enhance the UX. These changes will be explained later in this chapter.

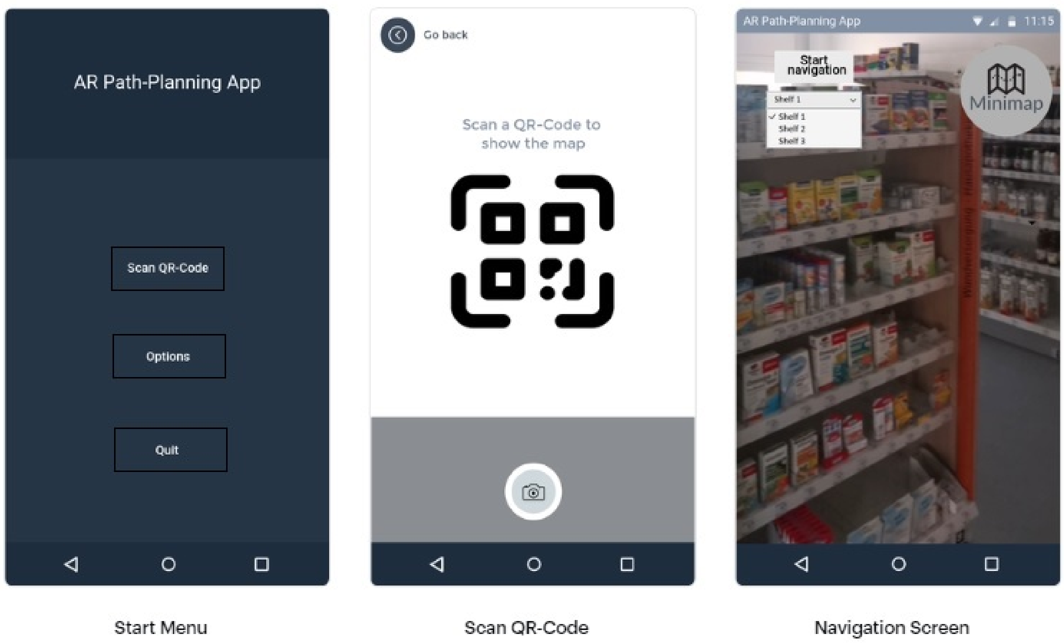


Figure 5.4: Prototype for the Android app

5.2.2 Structure

The structure of the whole project is showed below (see Figure 5.5). The website uses a knowledge graph to acquire the data required from the user. It can be opened in the browser on the computer or on the smartphone. In order to access it directly from the application it was used a webview (see the picture of the phone).

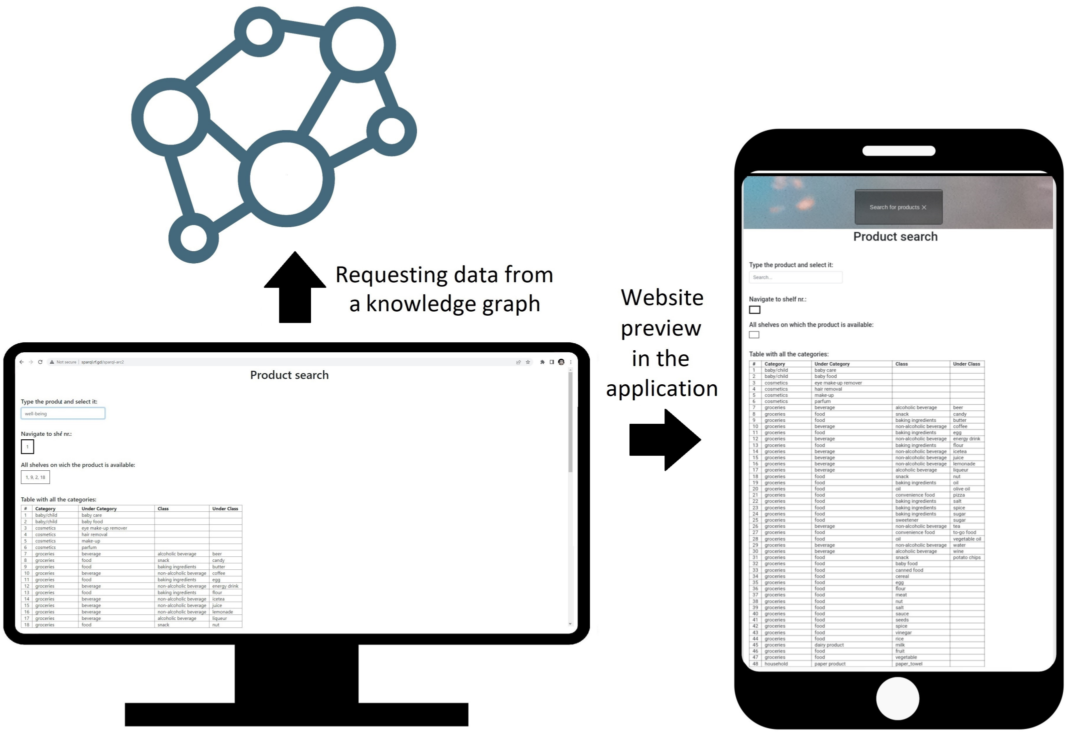


Figure 5.5: Structure

5.2.3 Functionality of the application

In order to describe the functionality of the Android application it was used an *Unified Modeling Language (UML)* activity diagram. To simplify the explanation of the following diagram (see Figure 8.1), this section was divided into four parts and each of them will represent a part of the figure mentioned above.

First part: Main Menu

If the display of the Android device has a notch (an irregular shape of the screen with a cutout for the front camera), the application will disable it when started. This will allow the user to see the entire screen of the application without an information to be cut off. The first and most simple part of the application is the main menu after the user start the application. It is shown in the figure (Figure 5.6) and has only two options, the first is to start scanning QR codes and the other one to quit the application. The middle option from the „Main Menu“ mentioned in the prototype was moved to another screen for more convenient UX.

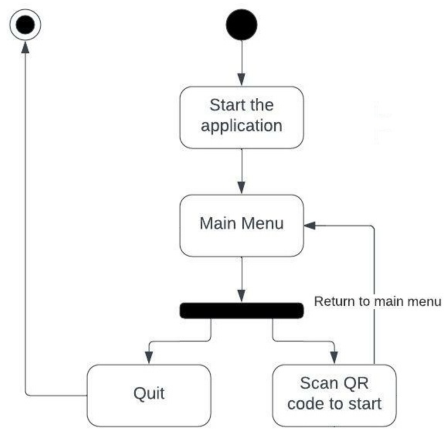


Figure 5.6: UML activity diagram (Main Menu)

Second part: Scan QR codes

In this part, it will be explained how the scanning of a QR code should work. At the initial start after the application is successfully installed on the smartphone and the user chooses the option „Scan Map QR code“ from the „Main Menu“, the device will ask the user for a permission to use the camera of the device. Without granting a permission for the camera the scanning will not be possible and hence the application will be unusable. For decoding the QR code it was used a library called „ZXing.Net“ (see zxing GitHub) which supports decoding barcodes within images. This library detects the QR code after the user takes a picture of it. If the QR code is recognized the application will load the corresponding digital twin map which is linked to this QR code and enter into navigation mode (see Figure 5.7). Otherwise, the application will throw an error message and the user can try again. The scanning process can be affected from the light or the angle so moving and focusing the camera only on the barcode can help recognizing it. The user can scan not only regular white and black QR codes but also inverted black and white as well as colorful ones if they have enough color hue difference in both colors which they have. For example, all three QR codes in the image (see Figure 8.2) should open the same map when scanned.

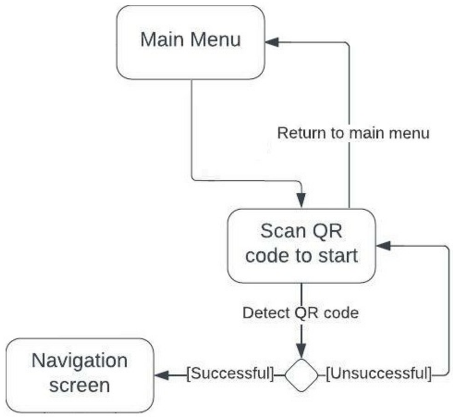


Figure 5.7: UML activity diagram (Scan QR code)

Third part: Navigation

When the user successfully scans the QR code, the map linked to this barcode will be loaded and the app will enter the navigation screen. From there the user will be able to see everything that is in the viewport of the back camera of the device. From this screen, they have multiple options besides the camera view. One of them is changing the settings of the navigation which were previously planned to be in the „Main Menu“ of the prototype and later moved to this screen for easier access. The settings and the WebView panels will be explained in details in the next section (Fourth part: Panels). At the top right corner of the screen will be placed the minimap which will be on top of the camera view and will display the position of the user in the retail store and in what direction the camera of the device is pointed. In order to detect the movement and the change of the direction the device is using a SLAM technology which is essentially integrated in the ARCore SDK. Moreover, AR Foundation framework (using ARCore SDK) is included to the project to add an AR experience to the application. The part that is shown in the Figure 5.8 is how the navigation can be used. Next to the minimap on the left side of the screen is placed one dropdown menu with all available shelves on the map as options and a button which can start or stop the navigation. There is a button to go back to the main menu in the top left corner of the screen. After clicking it the user will be asked with a dialog box to confirm their decision. Going back to the „Main Menu“ will completely close the navigation and reset the settings to the default ones if they were changed previously. When the user starts the navigation two things can happen. Firstly, the user can spot a blue dot in the middle of the left or right side of the screen that indicates in which direction the user needs to rotate in order to see the navigation line. Secondly, the navigation line, which uses AR technology, is visible from user's perspective and it can be followed to the selected destination. Once the destination is reached the user will see a green cube over the camera view showing where the searched shelf is located.

The minimap will also display the line between the user and the pin which marks the destination. If the navigation is stopped, the navigation line on the map and minimap as well as the destination pin and cube should disappear.

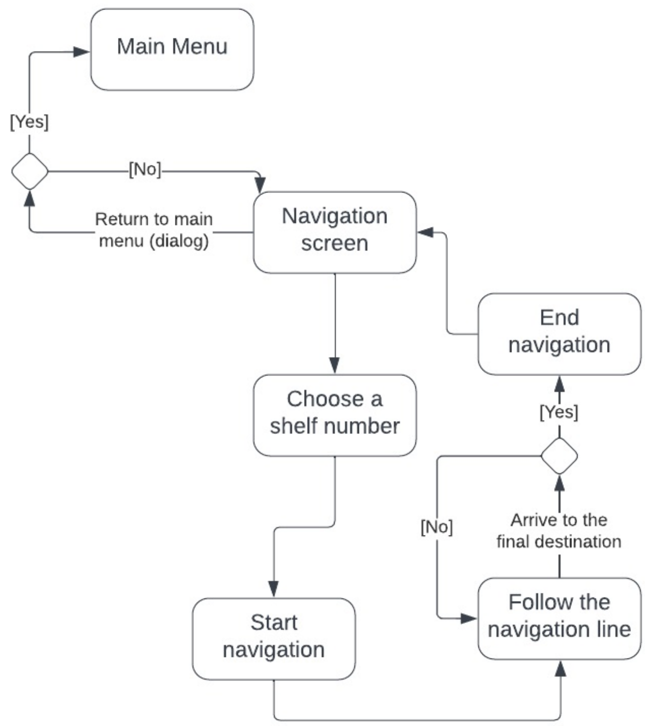


Figure 5.8: UML activity diagram (Navigation)

Fourth part: Panels

In the last part of the UML activity diagram (see Figure 5.9) are illustrated the panels for the „Settings“ and the „WebView“ used in the application. The settings panel contains of two sliders. The first one is for controlling the zoom of the minimap which changes not only how big the arrow for the directions in the middle can be but also the minimap itself as well. The second slider is to control the width of the shown navigation line. The WebView panel can only be used if the user has turned on the Wi-Fi or the mobile data on their device. This opens another smaller screen on top of the navigation screen where the website created for this project will be loaded if there is an internet connection. If not, then the WebView will be empty and no data for the products will be shown. For creating the WebView it was used a plugin but more about it will be presented in the next chapter (see Realization).

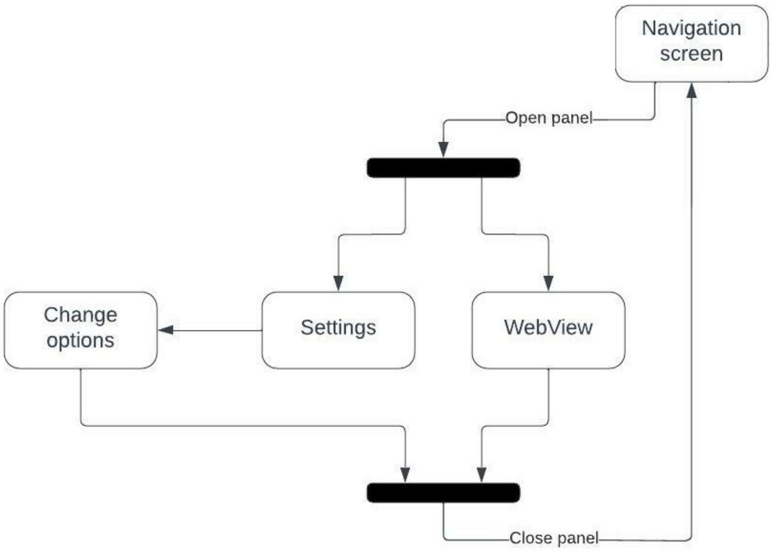


Figure 5.9: UML activity diagram (Panels)

5.2.4 Retail Lab Map

Before starting to develop the application it was necessary to take measurements of the retail lab including all the shelves. After that, a 2D floor plan (see Figure 5.10) of the room was created. The measurements of this layout were important to be completely correct in order to reduce the possibility of inaccuracy in the navigation mode when using the SLAM technology. Later the 2D layout will be used to build a 3D digital twin map (see Figure 5.11) which is an image of a prefab (reusable GameObject in Unity Engine). Moreover, the blue dot on the ground is showing from which position the user starts the application. The direction is always where the QR code is located because the user needs to scan it before the map is loaded.

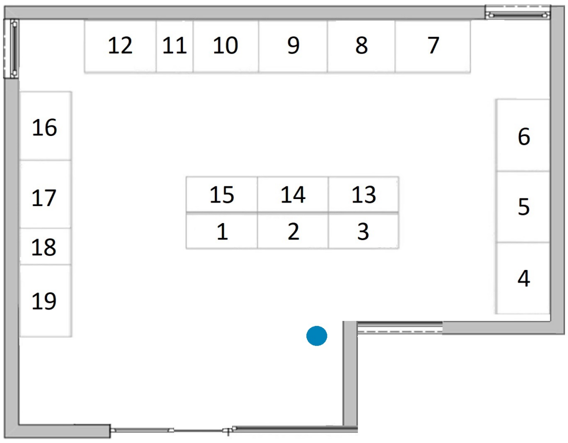


Figure 5.10: 2D layout of the retail lab with shelf numbers

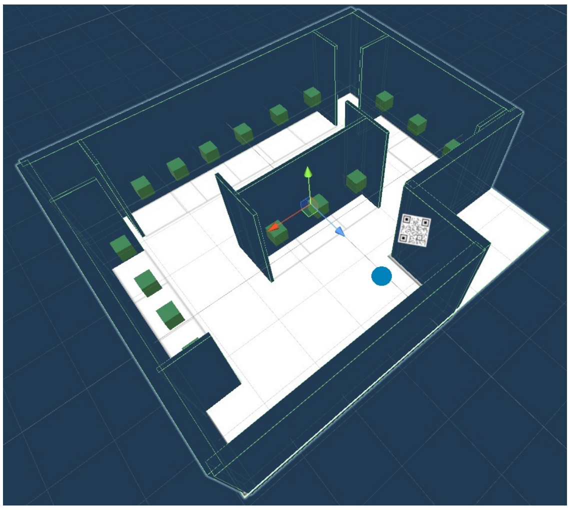


Figure 5.11: 3D digital twin of the retail lab map with 19 shelves

5.2.5 Testing Map

For creating a map it was used the same approach as explained above (see Retail Lab Map). The map of the apartment has seven destinations and is developed only for testing purposes. It was a test not only for the indoor navigation but also for the QR scanning.

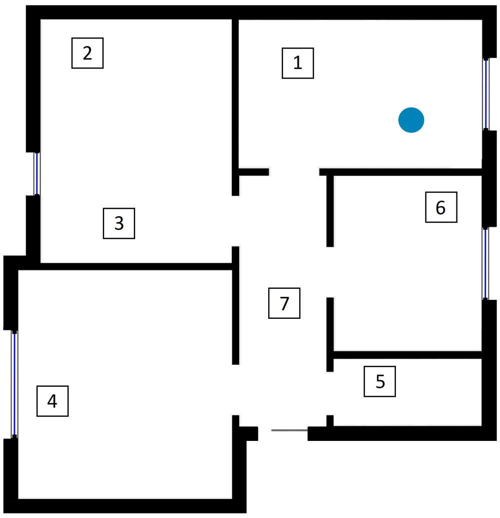


Figure 5.12: 2D layout of an apartment with numbers for all locations

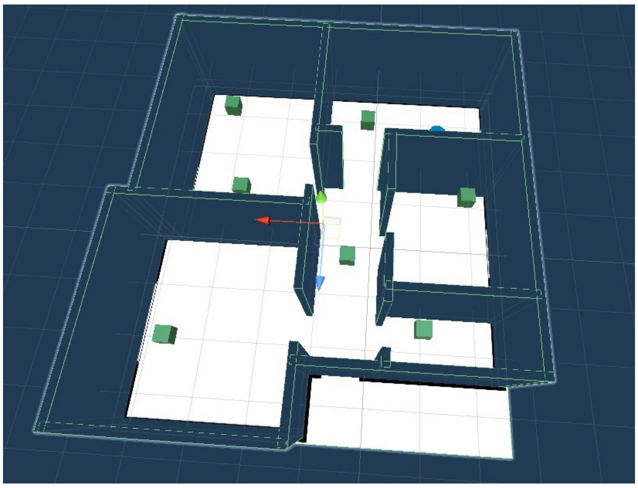


Figure 5.13: 3D digital twin of an apartment with 7 locations

6 Realization

In this chapter, it will be explained the development of the whole system including the Android application and the website application. Furthermore, it will be presented the functionality of both components and what technologies are used to implement them. The chapter will be divided in three main sections: Website Development, Android Development and at the end, it will be also discussed what Development Problems did occurred throughout the implementation of the both parts and how were they managed.

6.1 Website Development

As explained in section 5.1, the idea is to create a simple, yet reliable and user-friendly website where the user can see all available products in a table, search for a product name from a dropdown menu and gather shelf location data for it fast and efficient. In order to achieve that, firstly, it was important to decide in what programming languages this part will be developed and what technologies are needed to successfully access and obtain the information from the knowledge graph. And secondly, it will be shown how the data is displayed for the users and how the search process is working.

In [10], which is a tutorial on how to output SPARQL queries directly to a web page, is shown an example of how the data from a SPARQL endpoint is accessed and displayed on the website by using a PHP library called „ARC2“. The library should be imported into the project as followed:

```
include_once('proj/semsol/ARC2.php');
```

in order to be used. After including the library, an endpoint needs to be provided so that there is a knowledge graph that can be accessed. The endpoint for the product information is added to the project as shown below:

```
$dbpconfig = array(
    "remote_store_endpoint" => "https://api.krr.triply.cc
    /datasets/mkumpel/NonFoodKG/services/NonFoodKG/sparql",
);

$store = ARC2::getRemoteStore($dbpconfig);
```

and all the data can be queried from the variable „$*store*“. There were two SPARQL queries that were used, one for obtaining all products in the retail store and dividing them into four columns for the table, as explained in section 5.1, and the other for searching a shelf location by providing a product name. In order to allow the user to search for a product name, a search field (UI element from the navigational components) is added. Furthermore, for the functionality of the search field is used „jQuery“, a JavaScript library, to allow some additional changes. In order to search from it, the user needs to write at least two letters in the search field and a dropdown menu with maximum fifteen results will show up. If the user wants to choose an option from the dropdown menu, the only possible way is with a left mouse click on the item or if the website is open in a browser from a smartphone or directly in the application with a WebView then the way to do it is with a touch interaction. The „enter“ keyboard button is disabled. Other UI elements, such as text fields and a table, were also used to create the website. At the end, after the user searched for a product name two different results are shown. One of them is showing the users the shelf number which they should use for the navigation in the Android application and the other one is showing all shelves on which the product can be found. It is possible that a product has more than one shelf.

Throughout the whole process of creating the website, it was used XAMPP to test it locally. After the development was completed, it was essential that the website is hosted somewhere in order to be accessed from another device on the Internet. With the help of Infinity Free, a free hosting service, the website was successfully hosted (domain link) and it is accessible until 17.December 2022. After this date the SSL certificate will expire and in order for the webpage to stay online a new one will be necessary.

6.2 Android App Development

The development of the Android Application, in this section, will be divided in five parts. In the first one it is explained how the project is created, an AR experience is added and what framework is used. The second one presents the creation of both digital twin maps - the retail store and an apartment with multiple rooms. Last three parts are used to describe all of the scenes in the application - „Main Menu“, „Scan QR code“ and „Navigation mode“. The mobile device used to test the AR application is „HONOR View20“ with an Android version 10.0 (API level = 29). The smartphone supports ARCore SDK. The following link contains information about which mobile devices support ARCore SDK.

6.2.1 Unity Android Project with AR Foundation

The application was developed using Unity Engine (version 2021.2.10f1) and it is a 3D project. After creating it, the first thing was to install and enable the AR Foundation framework and ARCore XR plugin from the „Unity Registry“ using Package Manager.

Furthermore, some options in the „Build Settings“ were changed. The platform was switched from Windows to Android. In the „Project Settings“ a new tab called „XR Plug-In Management“ is shown. From there, the ARCore SDK for Android can be enabled. After that, in the „Project Settings“, Vulkan was removed from the Graphics APIs, the minimum API level was changed to Android 7.0 Nougat, scripting backend was changed to IL2CPP and the target architecture was selected to be ARM64. The default orientation was changed to be only portrait to prevent from rotating the screen while using the application. Moreover, rendering outside the safe area was disabled and if the mobile device has a notch, it will not be a problem and no information will be cut from the screen of the smartphone.

In order to successfully import the AR Foundation framework, the default main camera was removed, new „AR Session“ and „AR Session Origin“ GameObjects were added to the hierarchy of the project. The „AR Session Origin“ has a child GameObject called „AR Camera“ and this will be the camera for the AR application after deleting the main one. In the „AR Camera“ GameObject the tag can be changed to „MainCamera“, the „Audio Listener“ can be removed and „AR Pose Driver“ can be disabled because the „Tracked Pose Driver“ will be enough for this project.

6.2.2 Digital Twin Maps

For the application two digital twin maps were created - one for the retail room where all products are available and one for testing the application. Both maps have different sizes and a number of destinations. The same technique were used for creating both of them as well.

Retail Lab

The digital twin map of the retail lab is only one room and has nineteen shelves marked as destinations. In the middle of the room, a big rack with six shelves is placed and around him on almost all of the walls there are shelves (see Figure 5.10). In order to create a 3D digital twin of the room, the required measurements of the walls and shelves are taken. Multiple GameObjects are used to serve as walls and as a floor was used the 2D layout mentioned above. All of the walls have an occlusion material [16, p.184-188] and will not allow the navigation line to go through them when the indoor tracking is running. The floor and walls have different layers. Every destination of the map is marked with a green cube and a text inside it showing the number of the shelf (see Figure 8.13).

Home

The home map is created like the map of the retail lab and has only seven destinations. The area of the place is a little bit bigger and it has more rooms. Due to the restricted access to the retail lab, it was important to create another map that can be used to test the application.

6.2.3 Scene 1: Main Menu

When the user starts the application the first scene that shows up is the main menu. It differs a bit from the main menu in the prototype. In the implementation the „Settings“ option from the menu was moved to the navigation mode. There are two possibilities from the main menu - to start scanning QR codes or to quit the application. The first option will be explained in more details in the next section.

6.2.4 Scene 2: Scan QR Code

In this scene, the scanning of QR codes is implemented. For this, a library called „ZXing.Net“ is imported in the folder „Plugins/Zxing-QR-Scan-Integration/zxing.dll“. This library not only generates QR codes but it can also read them within images. In order for the users to use the back camera and make photos with it, they need to grant a permission for the camera of the device:

```
Permission.RequestUserPermission(Permission.Camera);
```

After that, the camera of the smartphone can be accessed through a „WebCamTexture“, in this case the variable is called „cameraTexture“. In order to scan the QR code it is used a function called „Decode()“ on the „IBarcodeReader“ from the „ZXing.Net“ library as shown below:

```
IBarcodeReader barcodeReader = new BarcodeReader();
Result result = barcodeReader.Decode(cameraTexture.GetPixels32(),
                cameraTexture.width, cameraTexture.height);
```

The result is used to check if the QR code is valid and can be detected. If it is not, then the user will get a message on the screen that the scanning was unsuccessful. Otherwise, the user will be send to the next and final screen of the application - to the Navigation mode.

6.2.5 Scene 3: Navigation mode

The main part of the implementation was the development of the navigation and all components to it. In this section, it will be described how the navigation works on the map and what algorithm was used to calculate the right path between the user („AR Camera") and the target, which in this case is the shelf location. Furthermore, it will be explained more details to the navigation and how the minimap was created. At the end it will be presented how the WebView panel is developed and what settings for configuration of the navigation are available to the user.

NavMesh tool

After creating both maps mentioned in subsection 6.2.2, they need to be prepared for the navigation mode. There are different algorithms for pathfinding but the algorithm used in this work was implemented in the tool NavMesh [15] and it is called **A*** [3]. In this tool various areas (e.g. Walkable, Jump, etc.) can be defined and the size of an agent (this is the user) can be modified. After configuring these settings, the tool can „bake" the environment with all walls and shelves and create an area where the agent can walk freely. The baked area for the retail map is shown in the Figure 6.1.

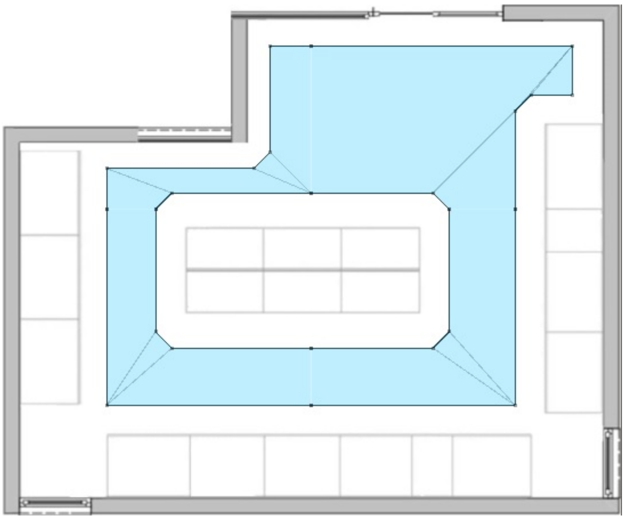


Figure 6.1: Retail lab map - baked area with the NavMesh tool

Minimap

In order to allow the user easily to check his position and direction in the map a minimap was created [16, p.169-178]. This is another camera rotated 90 degrees on the x-axis to look straight down and portrait the 2D layout plan of the map in a small circle with a border on the top right corner. An example of it can be seen in the screenshots shown in the appendix. In the minimap, the path between the user and the target is portrayed with a line. The icon shown over the target location is illustrated in Figure 6.2. This is only shown when the users have started the navigation and they are close enough to it so that it is visible on the minimap.

Figure 6.2: The minimap icon that indicates a target location

Navigation

When the user is in the navigation part of the application the screen never turns off. If users click on any of the Android buttons except the back button, because it is disabled, they will stop the application and should start over. The same will happen if the mobile device is locked (shown in the code below). This is implemented in order to prevent locking the device of minimizing the application and going back to it because after resuming the navigation the motion tracking can be lost or showing wrong information if the device was moved while the camera was not collecting information. In the following link is described all possible failures for the tracking.

```
public void OnApplicationPause(bool pauseStatus){
        if (pauseStatus) {
            Application.Quit();
    }
}
```

The navigation developed in this application is using the function „CalculatePath()“ from the *UnityEngine.AI.NavMesh* class, as shown below, to calculate the path between two points (the user - „AR Camera“ and the target object - one shelf location) and store the resulting path in a variable. The method is used in the „Update()“ method in order for the user to always stay updated when moving around the map. The *path* variable is then used to draw a „LineRenderer“ a line that indicates the closest path in the walkable area of the map.

```
public void Update() {
    NavMesh.CalculatePath(transform.position,
    positionTarget, NavMesh.AllAreas, path);
}
```

Because there is not only one target on the map, a dropdown menu with all possible shelf locations is added to the application. There is also a button which allows the user to start or stop the navigation after choosing a shelf from the dropdown menu. This is created by showing and hiding the „LineRenderer“ when the button is toggled. The starting point is always the user's location and the endpoint is always different.

When the navigation is started the dropdown menu is disabled. Moreover, if the navigation line is not visible on the screen of the device the user needs to rotate in order to see it. With the help of a pointer, implemented to show only when the navigation is running and the line is not visible, the users can see if they need to rotate left or right depends on where the pointer is shown on the screen. An example of this can be seen on the Figure 8.12.

WebView Panel

In order for the user to use a website inside the Unity Application a *webview* plugin was imported. In the plugin a sample webview was available. It had a lot of options implemented in it but some of them were unnecessary so they were modified or removed to make it more simple and intuitive for the user. The webview is always started when the navigation screen is loaded. This allows the user to directly start searching for a product. With a button the webview can be closed and everything else will be again shown (Minimap, Dropdown menu, etc.). After reopening the webview the website will be reloaded if the device has an Internet connection. If not the website will not be shown and the webview will remain white.

6.3 Development Problems

The library „ZXing.Net“ doesn’t support different colors for the QR code. Only the usual ones with white background and black squares. The solution to this problem was to add a functionality that invert the colors of the checked QR code after a photo is taken so that the white background becomes black and the black squares become white. The idea is that if the hue of the colors is different enough the program will still be able to recognize these QR codes. An example can be seen on Figure 8.2.

Moreover, returning to the main menu from the navigation screen was not working properly. The problem was that loading the scene again was not enough. In the following code is shown the correct way to load another screen and reset the „AR Session“ from the *AR Foundation* framework.

```
public void LoadSceneWithName(string scene) {
        if(arSession != null) arSession.Reset();
        SceneManager.LoadScene(scene);
}
```

Another difficult problem to find was located in the implementation of the webview plugin.

The target API used for this project was detected automatically by the Unity Engine and set to the highest installed, which in this case was the latest. As explained in this link the „android:usesCleartextTraffic“ before the API level 28 was „true“, but after that they updated it to be „false“ by default. In that respect, it was important to change a C# script file from the plugin called „UnityWebPostprocessBuild.cs“ (line 79) in order the website to be shown in the webview. For other websites that are encrypted this was not a problem but for this one the solution was to change the code of the plugin manually and add the line shown below.

```
changed = (androidManifest.SetUsesCleartextTraffic(true) || changed);
```

7 Evaluation

In this chapter, the AR Android application will be tested and evaluated. The first section is a survey which was conducted among students. The second one, was an evaluation created to compare the accuracy of the application in different environments.

7.1 Comparison between Robot and Android App

The survey was performed among 17 students at the retail lab where the products, mentioned in the knowledge graph are located.

The first part of the evaluation (see Figure 7.1) consisted of giving the participants tasks in which they needed to find a specific product. The time for each task was measured. At first (**Q1**) they searched for the product only by its name (e.g. shampoo). Then (**Q2**) they needed to find the product after receiving detailed information about it (e.g. anti dandruff shampoo). The third and forth tasks were the same, but instead of the participants searching for the product by themselves, they now used the robot assistant and its interface [2]. At this point, tasks five and six were supposed to be measured in which the same activities were to be executed using the indoor navigation application. Unfortunately, those questions were not part of the evaluation.

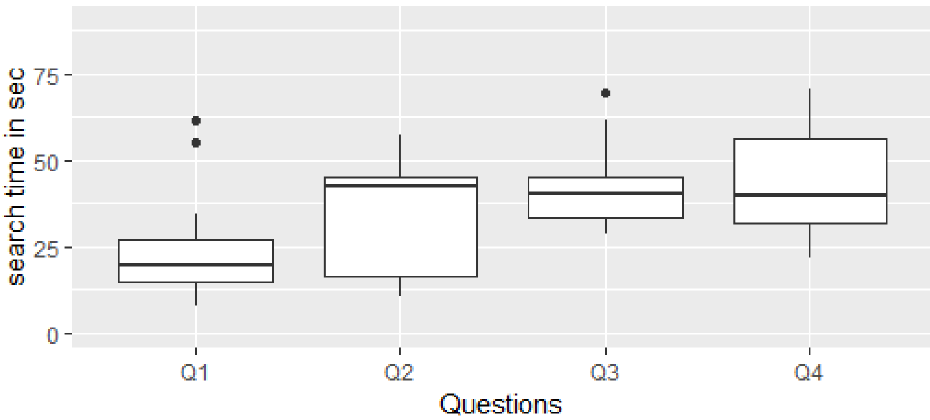


Figure 7.1: Response time for finding a product

The second and third part answered the questions „*Would you use a Shopping Assistant?*“ (Figure 7.2) and „*Would you use a Smartphone or Robot?*“ (Figure 7.3) respectively. Less than 20% of the participants were certain that they do not want the help of any shopping assistant. The remaining, more than 80%, were determined to either use it or give it a chance. Because of the high level of interest, they were asked the second question which revealed a greater preference for smartphone assistance in comparison with the robot one.

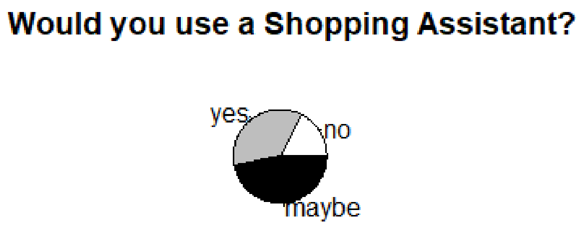


Figure 7.2: Users preference of indoor assistance in a retail store

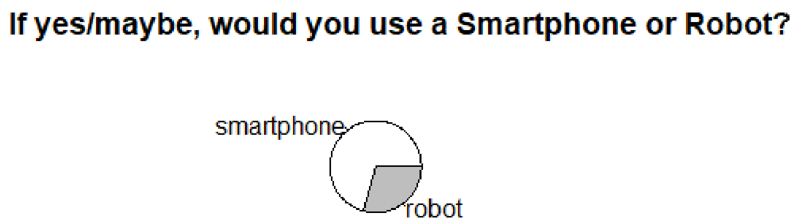


Figure 7.3: Smartphone vs Robot

7.2 Home vs Retail-Lab

In order to test the application during the development, another digital twin map was constructed replicating an actual apartment with several rooms. After the application was completed that same map was used to evaluate the accuracy of the indoor navigation system. After measuring the accuracy in both environments, the results show a better performance in favor of the retail lab map.

8 Conclusion

8.1 Summary

In this thesis, a system of android application and a website was presented. The system uses knowledge graph to access the product information in the retail map created with the help of a digital twin concept. The system was evaluated among other students and the results show that most of them prefer using an indoor navigation assistance in form of an Android application instead of a robot because it is more intuitive and accessible. Everyone can use their phones to download an application and follow the navigation by themselves. However, using a robot guidance can be a more difficult task. Robots are not everywhere and most people need to learn how to use them before they benefit from their functionality.

8.2 Future work

While the website and the Android application, created in this thesis are working properly and can be used, there are still improvements that can be done in both of them. The main one is that the application is not communicating directly with the website. Starting the navigation after receiving the shelf location from the website is performed manually by the user within the application. A possible solution would be that the user can start the navigation through the website after a product is searched and a result (shelf location) is returned. Another important improvement can be a more detailed search of products in the website. For example, searching for the cheapest product or a product that consists only of natural ingredients. Currently, there is only one POI where the user starts the application by scanning the QR code. If the retail lab has a larger area or more rooms the localization of the device can be easily lost. The solution to this problem can be adding more POIs in the store so that the user can scan the closest one and continue using the application. There are also small improvements that can make the application more user-friendly. For example, the application can stop the navigation automatically when the user is close to the target location or tilting the phone too much while using the navigation should warn the user so that the motion tracking can work properly.

8.3 Used Resources

In the section below are listed all used resources for creating this thesis along with links to them.

- NonFoodKG - https://k4r-iai.github.io/NonFoodKG/
- Protégé - https://protege.stanford.edu/products.php
- XAMPP - https://www.apachefriends.org/download.html
- Visual Studio Code - https://code.visualstudio.com/download
- Infinity Free - https://www.infinityfree.net
- Triply NonFoodKG - https://krr.triply.cc/mkumpel/NonFoodKG
- GRLC GitHub - https://github.com/CLARIAH/grlc
- GRLC NonFoodKG - https://grlc.io/api/K4R-IAI/NonFoodKG/SPARQLfiles/
- Endpoint - https://api.krr.triply.cc/datasets/mkumpel/NonFoodKG/services/NonFoodKG/sparql (this is an URI endpoint)
- Unity Engine - https://unity.com/download
- Visual Studio Community - https://visualstudio.microsoft.com/downloads
- ZXing.Net library - https://github.com/micjahn/ZXing.Net
- ARC2 library - https://github.com/semsol/arc2/
- Unity WebView - https://github.com/gree/unity-webview
- AR Foundation - https://developers.google.com/ar/develop
- App Menu Background - https://opengameart.org/sites/default/files/scifi_main_menu.jpg
- Knowledge Graph Icon - https://engineb.com/wp-content/uploads/2022/02/3-people-teal-icon-77.png

Acronyms

2D Two Dimensions

3D Three Dimensions

API Application Programming Interface

AR Augmented Reality

BLE Bluetooth Low Energy

GPS Global Positioning System

GUI Graphical User Interface

HTTP Hypertext Transfer Protocol

IDE Integrated Development Environment

IMU Inertial Measurement Unit

IPS Indoor Positioning System

MAR Mobile Augmented Reality

POI Point of Interest

RFID Radio Frequency Identification

QR Quick Response

SDK Software Development Kit

SLAM Simultaneous Localization and Mapping

SPARQL SPARQL Protocol and RDF Query Language

SSL Secure Sockets Layer

UI User Interface

UML Unified Modeling Language

URI Uniform Resource Identifier

URL Uniform Resource Locator

UX User Experience

XR Cross Reality

List of Figures

Bibliography

[1] Simran Birla, Gurveen Singh, Pratik Kumhar, Kshitij Gunjalkar, Sambhaji Sarode, Sanskar Choubey, and Mohandas Pawar. Disha-Indoor Navigation App. In *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 517–522, Greater Noida, India, December 2020. IEEE.

[2] Jonas Dech. A knowledge based robot assistant in a retail environment using external vr tracking. In *Master thesis*, 2021.

[3] Unity Documentation. Inner Workings of the Navigation System. `https://docs.unity3d.com/Manual/nav-InnerWorkings.html` (last accessed on 01 October 2022).

[4] IBM. How does a digital twin work? `https://www.ibm.com/topics/what-is-a-digital-twin` (last accessed on 04 October 2022).

[5] IBM Cloud Education. Knowledge Graph, 2021. `https://www.ibm.com/cloud/learn/knowledge-graph` (last accessed on 29 September 2022).

[6] Sanket V. Jambhulkar and S. J. Karale. Semantic web application generation using protégé tool. In *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, pages 1–5, 2016.

[7] Michaela Kümpel, Christian A. Mueller, and Michael Beetz. Semantic Digital Twins for Retail Logistics. In Michael Freitag, Herbert Kotzab, and Nicole Megow, editors, *Dynamics in Logistics*, pages 129–153. Springer International Publishing, Cham, 2021.

[8] Toru Kobayashi, Yoshifumi Onimaru, Kazuki Fukae, Tetsuo Imai, Kenichi Arai, Eisuke Nakazawa, and Tomohito Suzuki. Personal and intuitive indoor navigation system based on digital twin. In *2021 IEEE 11th International Conference on Consumer Electronics (ICCE-Berlin)*, pages 1–6, 2021.

[9] Ibrahim Alper Koc, Tacha Serif, Sezer Gören, and George Ghinea. Indoor mapping and positioning using augmented reality. In *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 335–342, 2019.

[10] Matthew Edward. Creating HTML Web Pages from Linked Data Using PHP, SPARQL, and ARC2, 2014. `http://matthewedward.blogspot.com/2014/04/create-html-web-pages-linked-data-php-sparql-arc2.html` (last accessed on 03 October 2022).

[11] Albert Meroño-Peñuela and Rinke Hoekstra. grlc Makes GitHub Taste Like Linked Data APIs. In *The Semantic Web: ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 – June 2, 2016*, pages 342–353. Springer, 2016.

[12] Gi Ho Nam, Han Sol Seo, Min Su Kim, Yun Kyoung Gwon, Cheol Min Lee, and Dong Myung Lee. AR-based Evacuation Route Guidance System in Indoor Fire Environment. In *2019 25th Asia-Pacific Conference on Communications (APCC)*, pages 316–319, Ho Chi Minh City, Vietnam, November 2019. IEEE.

[13] Xin Hui Ng and Woan Ning Lim. Design of a Mobile Augmented Reality-based Indoor Navigation System. In *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–6, Istanbul, Turkey, October 2020. IEEE.

[14] Tina Sayapogu, Kevin Dsa, and Priya Kaul. AR Smart Navigation System. In *2021 2nd International Conference for Emerging Technology (INCET)*, pages 1–4, Belagavi, India, May 2021. IEEE.

[15] Hans-Georg Schumann. *Unity-Workshop. Spiele-Welten mit Unity, MakeHuman und Blender erstellen*, pages 179–190. mitp-Verlags GmbH & Co. KG, Frechen, 2016.

[16] Carsten Seifert and Jan Wislaug. *Spiele entwickeln mit Unity 5. 2D- und 3D-Games mit Unity und C# für Desktop, Web & Mobile.* Carl Hanser Verlag München, 2017.

[17] Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, and Mika Ylianttila. A survey on mobile augmented reality with 5g mobile edge computing: Architectures, applications, and technical aspects. In *Survey on MAR with 5G MEC: Architectures, Applications, and Technical Aspects*, volume 23, pages 1160–1192, 2021.

Appendix

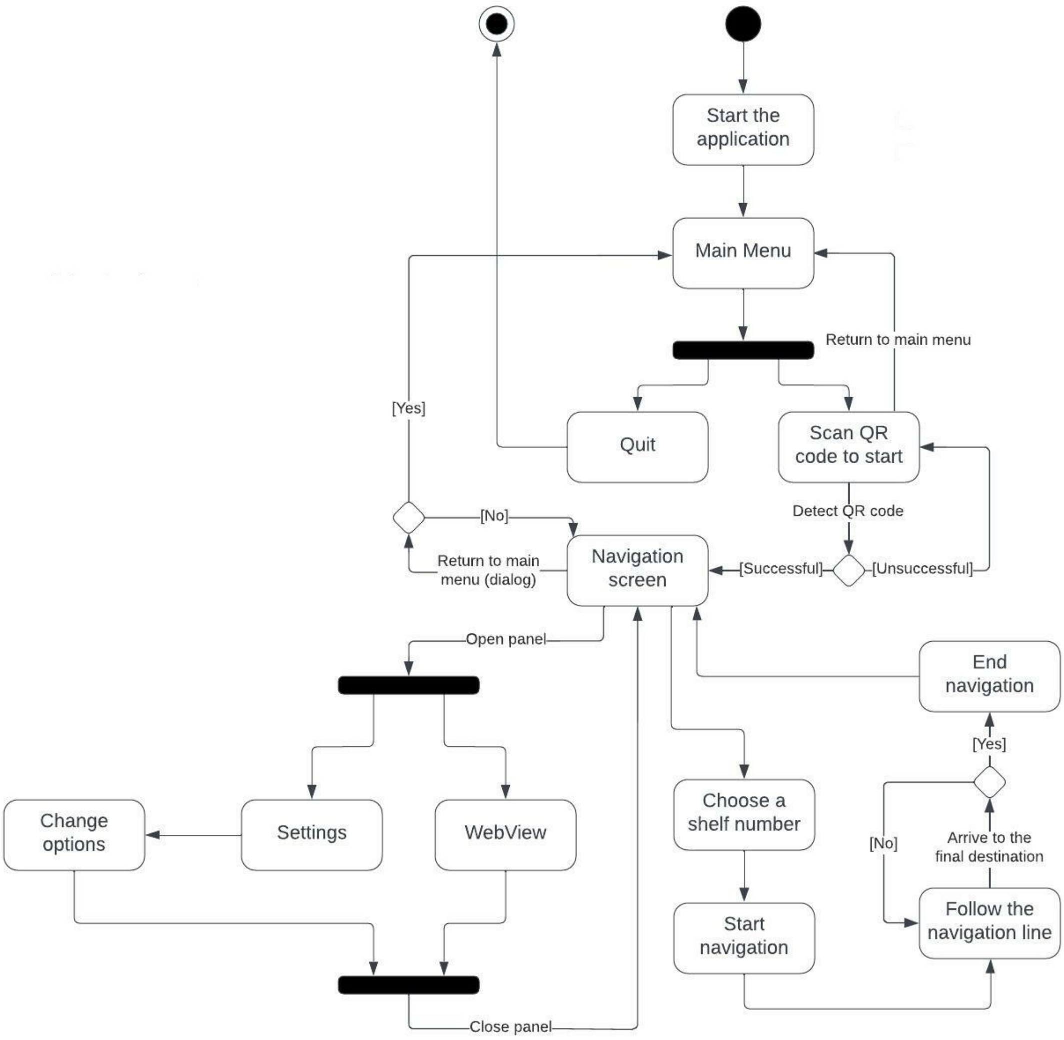


Figure 8.1: UML activity diagram for the Android application

Figure 8.2: Three QR codes for the retail lab map

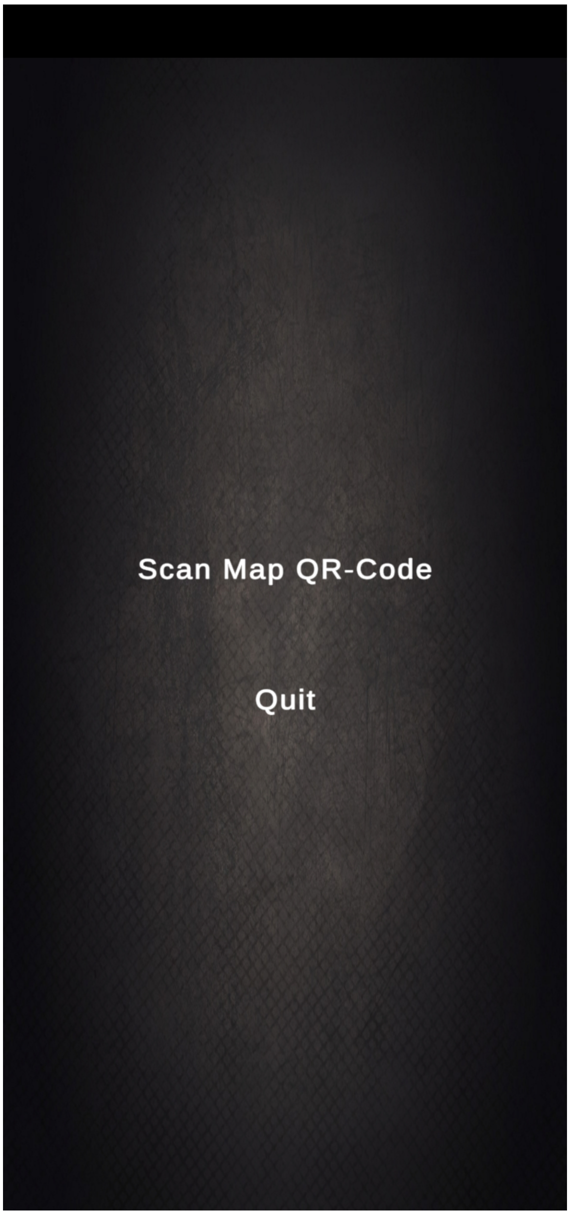


Figure 8.3: Screenshot: Main Menu

Figure 8.4: Screenshot: Scan QR Code

Figure 8.5: Screenshot: Scannning black and white QR code

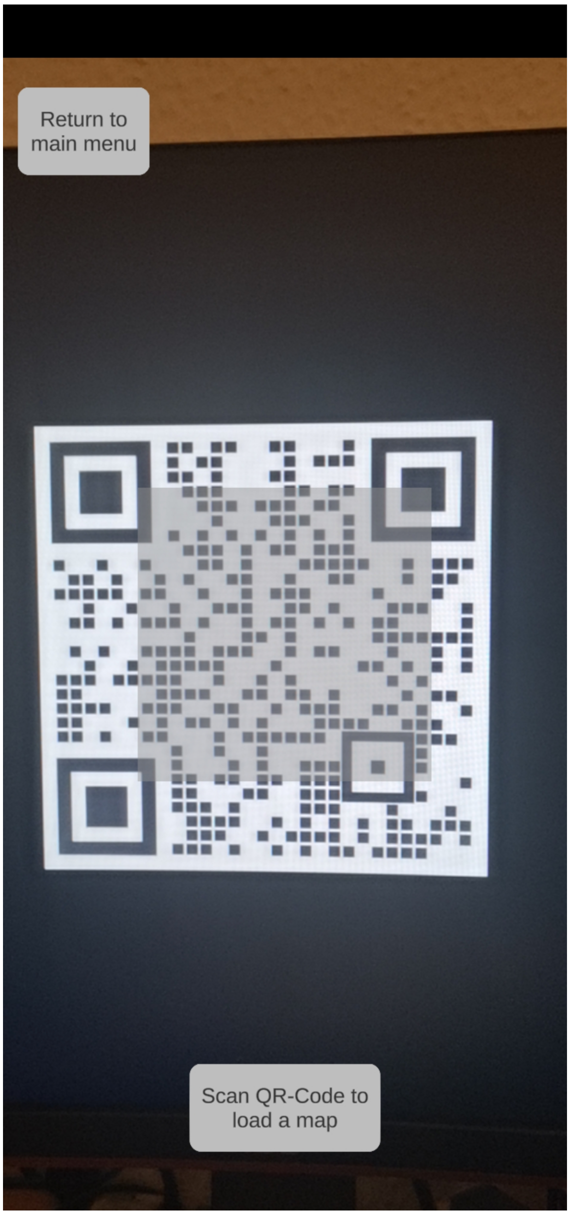


Figure 8.6: Screenshot: Scannning white and black QR code

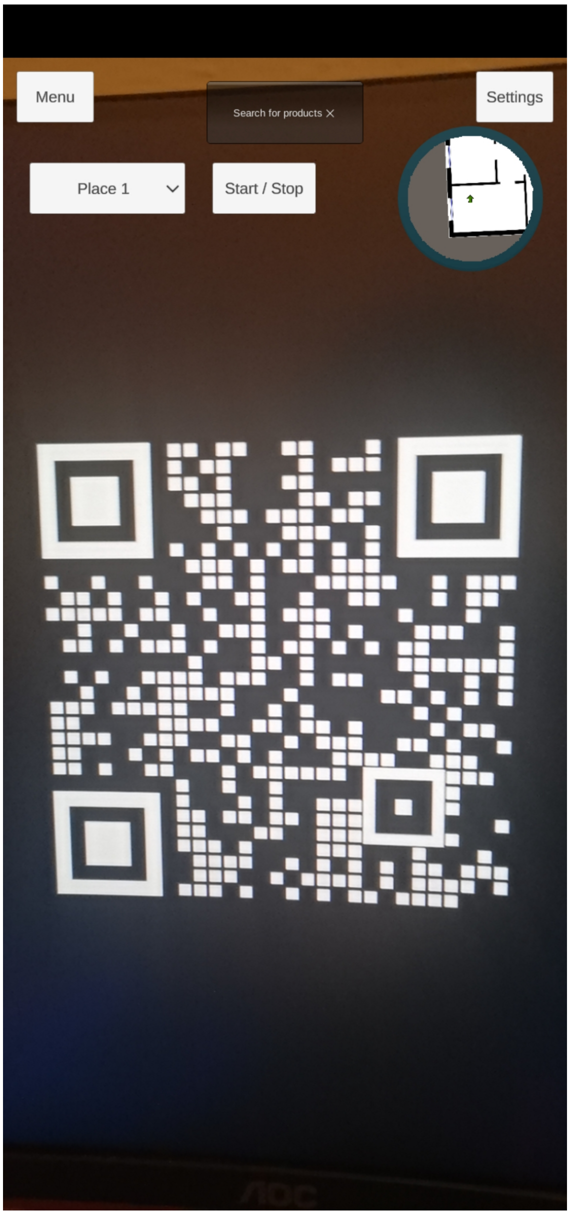
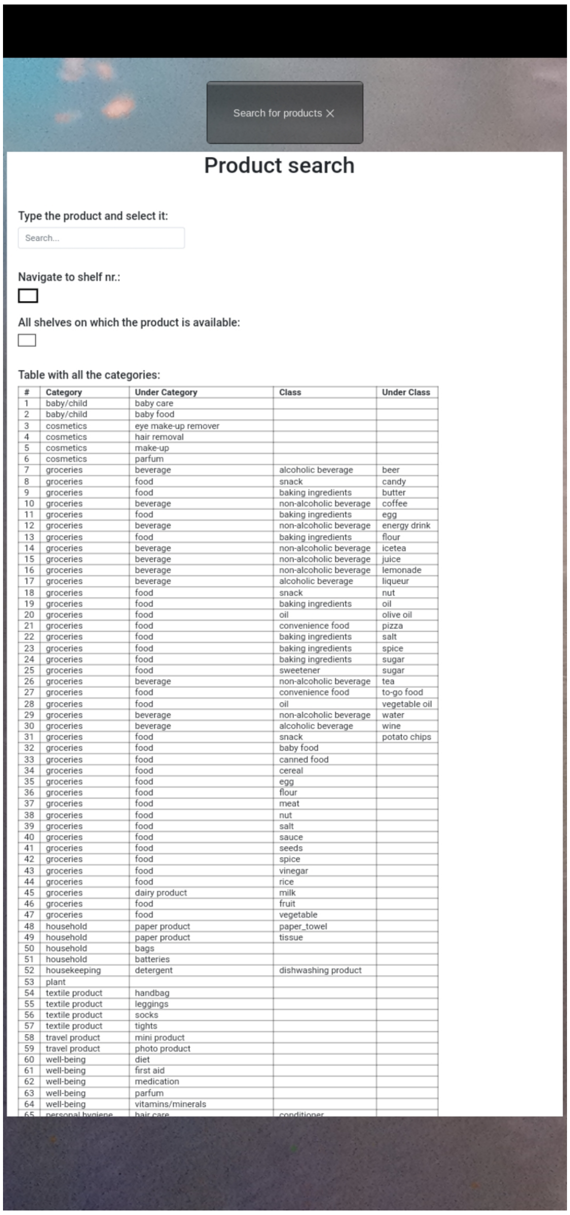


Figure 8.7: Screenshot: Navigation mode



Search for products ×

Product search

Type the product and select it:

Search...

Navigate to shelf nr.:

All shelves on which the product is available:

Table with all the categories:

#	Category	Under Category	Class	Under Class
1	baby/child	baby care		
2	baby/child	baby food		
3	cosmetics	eye make-up remover		
4	cosmetics	hair removal		
5	cosmetics	make-up		
6	cosmetics	parfum		
7	groceries	beverage	alcoholic beverage	beer
8	groceries	food	snack	candy
9	groceries	food	baking ingredients	butter
10	groceries	beverage	non-alcoholic beverage	coffee
11	groceries	food	baking ingredients	egg
12	groceries	beverage	non-alcoholic beverage	energy drink
13	groceries	food	baking ingredients	flour
14	groceries	beverage	non-alcoholic beverage	icetea
15	groceries	beverage	non-alcoholic beverage	juice
16	groceries	beverage	non-alcoholic beverage	lemonade
17	groceries	beverage	alcoholic beverage	liqueur
18	groceries	food	snack	nut
19	groceries	food	baking ingredients	oil
20	groceries	food	oil	olive oil
21	groceries	food	convenience food	pizza
22	groceries	food	baking ingredients	salt
23	groceries	food	baking ingredients	spice
24	groceries	food	baking ingredients	sugar
25	groceries	food	sweetener	sugar
26	groceries	beverage	non-alcoholic beverage	tea
27	groceries	food	convenience food	to-go food
28	groceries	food	oil	vegetable oil
29	groceries	beverage	non-alcoholic beverage	water
30	groceries	beverage	alcoholic beverage	wine
31	groceries	food	snack	potato chips
32	groceries	food	baby food	
33	groceries	food	canned food	
34	groceries	food	cereal	
35	groceries	food	egg	
36	groceries	food	flour	
37	groceries	food	meat	
38	groceries	food	nut	
39	groceries	food	salt	
40	groceries	food	sauce	
41	groceries	food	seeds	
42	groceries	food	spice	
43	groceries	food	vinegar	
44	groceries	food	rice	
45	groceries	dairy product	milk	
46	groceries	food	fruit	
47	groceries	food	vegetable	
48	household	paper product	paper_towel	
49	household	paper product	tissue	
50	household	bags		
51	household	batteries		
52	housekeeping	detergent	dishwashing product	
53	plant			
54	textile product	handbag		
55	textile product	leggings		
56	textile product	socks		
57	textile product	tights		
58	travel product	mini product		
59	travel product	photo product		
60	well-being	diet		
61	well-being	first aid		
62	well-being	medication		
63	well-being	parfum		
64	well-being	vitamins/minerals		
65	personal hygiene	hair care	conditioner	

Figure 8.8: Screenshot: WebView panel

Figure 8.9: Screenshot: Settings panel (Minimap zoom)

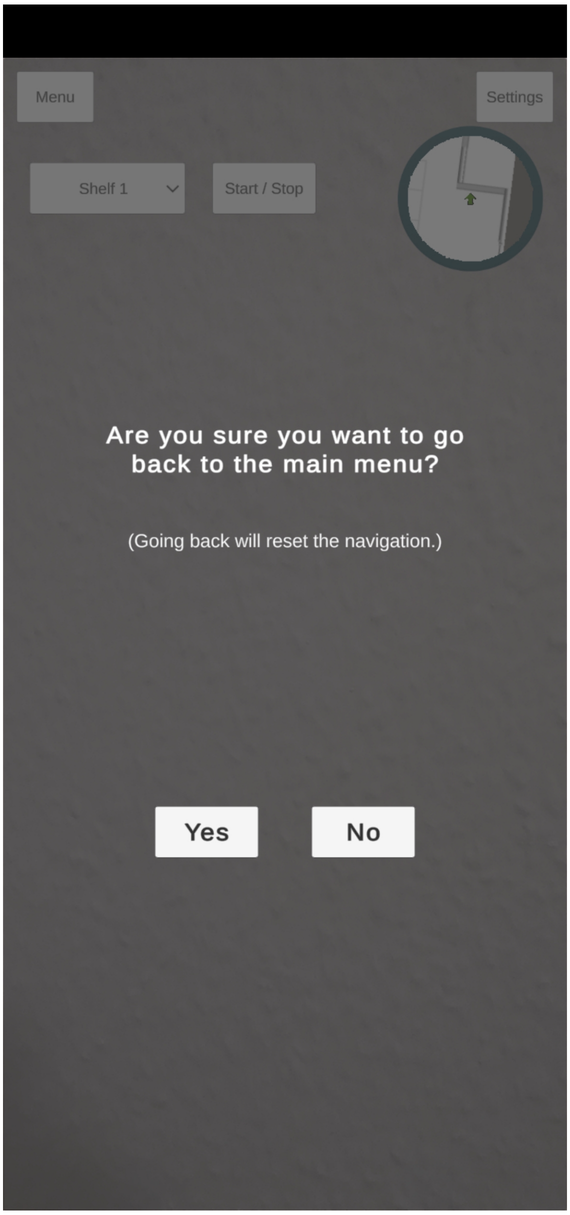


Figure 8.10: Screenshot: Dialog box to confirm returning back to main menu

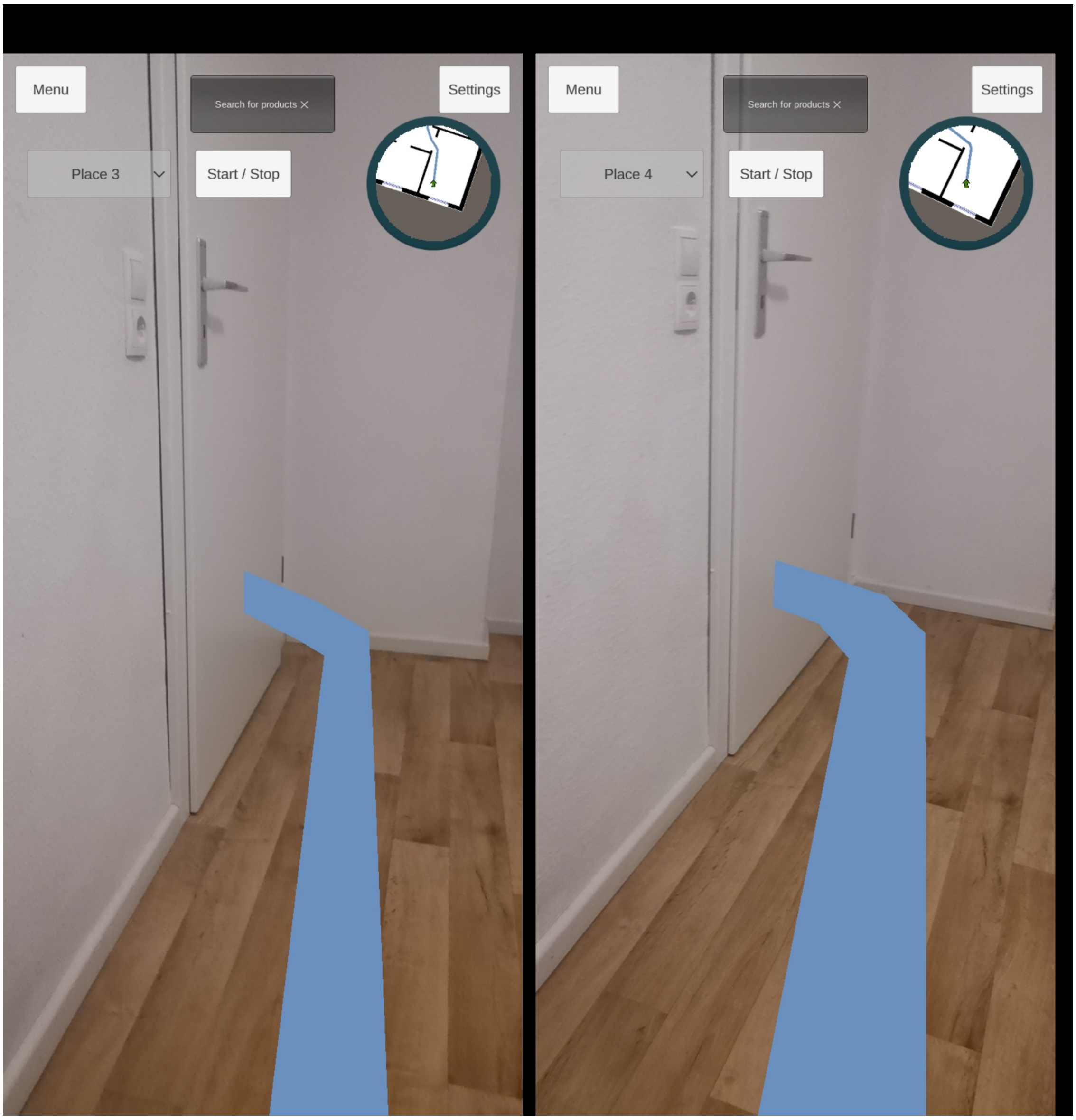


Figure 8.11: Screenshot: Settings panel (NavLine zoom)

Figure 8.12: Screenshot: Pointer for the NavLine

Figure 8.13: Screenshot: Target cube as a destination