

Development of an application for home inventory and shopping support

Alina Samatova

Supervisor: Ing. Pavel Náplava, Ph.D.
Field of study: Software engineering and technologies
May 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Samatova** Jméno: **Alina** Osobní číslo: **492213**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Vytvoření aplikace pro správu nákupů a domácích zásob

Název bakalářské práce anglicky:

Development of an application for home inventory and shopping support

Pokyny pro vypracování:

Navrhněte a v první verzi vytvořte mobilní aplikaci pro podporu nákupů a udržování domácích zásob, podporující hlasovou komunikaci mezi uživatelem a aplikací. Postupujte následovně:

- 1) Analyzujte a porovnejte existující aplikace pro podporu nákupů a udržování domácích zásob běžnými uživateli. Aplikace porovnejte a specifikujte klíčové funkčnosti, které tyto aplikace obsahují.
- 2) Prozkoumejte, zda a jak je možné tyto aplikace ovládat lidským hlasem.
- 3) Srovnejte existující hlasové asistenty, které je možné používat v mobilních zařízeních.
- 4) Na základě analýzy existujících aplikací a hlasových asistentů, navrhněte "jednoduchou" aplikaci, která bude podporovat nákupy a zjednoduší udržování domácích zásob za využití hlasového asistenta.
- 5) Vyberte vhodnou mobilní platformu a na této platformě implementujte výše specifikovanou aplikaci. Klíčovým požadavkem je ovládání aplikace hlasem.
- 6) Funkčnost a použitelnost aplikace ověřte formou uživatelského testování na vybrané skupině uživatelů.

Seznam doporučené literatury:

1. Android developers. Documentation for app developers, <https://developer.android.com/docs>
2. Google developers. Google Assistant. App Actions, <https://developers.google.com/assistant/app/overview>
3. Bc. Klusák, Michal. Nákup produktů pomocí Google Assistant. Vedoucí práce: Ing. Michal Havryluk. Praha: ČVUT, 2020. Diplomová práce (In Czech). České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra počítačů.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Náplava, Ph.D. Centrum znalostního managementu FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **02.02.2022** Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Pavel Náplava, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studentky

Acknowledgements

I would like to express my genuine gratitude to a lot of people for their support and help. First of all, I would like to thank my supervisor, Ing. Pavel Náplava, Ph.D, for his professional guidance, valuable advice, support and the time that he devoted to me and this bachelor thesis. I would also like to thank Angelina Agafonova whose assistance with the visual components was important for the success of this work. Likewise, I would like to thank the group of people who took part in the testing for their honest opinion and valuable advice. Also, I would like to thank Alsu Tasbaeva for professional language correction of my bachelor thesis. Finally, I am beyond grateful for my family, all my friends and close people who have always been there for me both in difficult and happy periods of my studies and life.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

Prague, May 15, 2022

Abstract

The main goal of this bachelor thesis is to analyze, design, implement and test the "Shopmate" application that is supporting shopping and the maintenance of the inventory, and which is being controlled by both interface and voice. Also, part of the research part of this work is the introduction to the concepts of the shopping support applications and the introduction to the voice assistants' work principles.

Keywords: inventory, shopping list, voice assistant, Google Assistant, App Actions, custom intents, Android, Java

Supervisor: Ing. Pavel Náplava, Ph.D.

Abstrakt

Hlavním cílem této bakalářské práce je zanalyzovat, navrhnout, implementovat a otestovat aplikaci „Shopmate“, která podporuje nakupování a údržbu zásob a může být ovládána přes rozhraní nebo hlasem. Součástí rešeršní části této práce je seznámení se s koncepty aplikací pro podporu nakupování a s principy fungování hlasových asistentů.

Klíčová slova: zásoby, nákupní seznam, hlasový asistent, Google Assistant, App Actions, custom intenty, Android, Java

Překlad názvu: Vytvoření aplikace pro správu nákupů a domácích zásob

Contents

1 Introduction	1
2 The goals of the thesis	3
3 The concepts of shopping support applications	5
3.1 Voice assistant	5
3.2 Inventory management	6
3.3 Shopping list management	7
3.4 Existing shopping support applications	8
3.5 The "Shopmate" application core	11
4 Voice assistants	13
4.1 Available voice assistants	13
4.2 Comparison of assistants.....	14
5 Google Assistant	17
5.1 Options to Implement Google Assistant	17
5.2 Comparison of options.....	17
5.3 Operation principles of App Actions	18
6 Analysis of the "Shopmate" application	21
6.1 Functional requirements	21
6.2 Non-functional requirements ...	23
6.3 Class diagram	24
6.4 Use cases	25
6.4.1 Actors	25
6.4.2 Use case diagram for user ...	26
6.4.3 Use case diagram for application	27
6.4.4 Use case scenarios.....	27
7 Design of the "Shopmate" application	29
7.1 Wireframes	29
7.2 Deployment diagram	34
8 Implementation of the "Shopmate" application	35
8.1 Used technologies	35
8.2 Integration of the Google Assistant to the application	36
8.2.1 Built-in intents	36
8.2.2 Custom intents	37
8.2.3 Inline Inventory.....	37
8.2.4 Similar - sounding words....	38
8.2.5 Configuration files.....	39
8.2.6 Request processing	41
8.2.7 Application logic	44
8.2.8 User interface	46
8.2.9 Problems with using the Google Assistant	50
9 Testing of the "Shopmate" application	53
9.1 Developer tests	53
9.2 Google Assistant tests	53
9.3 Testing of the Google Assistant in the real environment	55
9.4 User testing	55
9.4.1 Process of testing	56
9.4.2 Testing scenario	56
9.4.3 Tester's results	57
9.5 Testing results.....	64
10 What can be improved in the future in the "Shopmate" application	65
10.1 Improvements in the communication through the Google Assistant	65
10.2 Other improvements and new features	66
11 Conclusion	67
Resources	69
A Electronic attachments	75

Figures

3.1 Components of Voice Assistant [6]	6
3.2 Increase in the number of bought items in the inventory via assistant	7
3.3 Decrease in the number of consumed items in the inventory via voice assistant	8
3.4 Increase in the number of items in the shopping list via voice assistant	8
4.1 Share of voice assistants users in the United States in 2021, by product and age group [21]	13
5.1 Communication between the user, voice assistant, and the application.	19
6.1 Class Diagram of the "Shopmate" application	24
6.2 Actors.	25
6.3 Use case diagram for the User actor.	26
6.4 Use case diagram for the Application actor.	27
7.1 "Shopping List" page	30
7.2 "Shopping List" page, edit mode	30
7.3 "Inventory" page	31
7.4 "Add Item" page for shopping list items	31
7.5 "Add Item" page for inventory items	32
7.6 "Add Item" page for inventory items with visible critical level ...	32
7.7 "Edit Item" page for shopping list items	33
7.8 "Edit Item" page for inventory items	33
7.9 Deployment Diagram	34
8.1 Request processing	43
8.2 "Shopping List" page	46
8.3 "Inventory" page	47
8.4 "Help" page	47
8.5 "Add Item" page for shopping list items	48
8.6 "Add Item" page for inventory items	48
8.7 "Edit Item" page for shopping list items	49
8.8 "Edit Item" page for inventory items	49
8.9 Icon of the "Shopmate" application	50

Tables

3.1 Comparison of the "Inventory, shopping list" applications from the Google Play	10
4.1 Comparison of Google Assistant and Amazon Alexa	14
5.1 Ordinary Google Assistant	18
5.2 "Special" Google Assistant.	18
5.3 Comparison of App Actions and Conversational Actions	18
9.1 Results of noise tests	55
9.2 Information about the tester A .	58
9.3 Results of testing by the tester A	58
9.4 Potential improvements from testing by the tester A	59
9.5 Information about the tester B .	60
9.6 Results of testing by the tester B	60
9.7 Potential improvements from testing by the tester B.....	61
9.8 Information about the tester C .	61
9.9 Potential improvements from testing by the tester C	61
9.10 Potential improvements from testing by the tester C	62
9.11 Information about the tester D	62
9.12 Results of testing by the tester D	63
9.13 Potential improvements from testing by the tester D	63
10.1 Future improvements and features in the communication with the "Shopmate" application	65
10.2 Future improvements and features for the "Shopmate" application ...	66

Chapter 1

Introduction

Nowadays the pace of life is extremely fast: people have to complete heaps of tasks, take part in dozens of events and communicate with a great number of people every single day. Such a busy and fast life makes it easy for us to get lost in the continuous flow of information and miss the information that can be of great importance. It often happens to me that I come to the store and I don't quite remember whether I have a box of rice at home. So, I buy it just in case. Then I come home and realize that I already have 5 boxes of rice at home and don't have enough space for another one. That is why the idea of keeping the information on whole-home supplies and a shopping list in one place came to my mind. The idea is to keep this information well-organized, up-to-date, and available at all times.

In the 21st century the lives of people can be optimized and facilitated with the use of smart devices and other modern technologies. A lot of people nowadays use smartphones every day[1]: to chat with friends, surf the net, read books. Such applications as calendar and to-do list help people to save time and unload the brain for other things. So, the memorization of the shopping list and home inventory can be also delegated to the smart devices.

The goal of this bachelor thesis is to create the "Shopmate" application that can make the user's everyday shopping easier, more efficient and economical. The core idea of this application is the combination of 2 main concepts: tracing the condition of the home supplies and updating the shopping list.

Considering how the application can be the most convenient in usage, the idea of voice controlling comes to mind. Technologies have really advanced and can offer users bigger comfort and simplicity with the use of voice assistants now. The possibility of controlling the application by voice allows users to interact with an application both through the graphical interface and voice assistant. For me, it seems to be a significant part of the improvement in user's experience in such applications as "Shopmate". It is not always convenient to write down "I used 1 box of rice; 2 more boxes are left". But saying it to the voice assistant, whenever your mobile phone is near you, is easy and saves time.

Chapter 2

The goals of the thesis

The goals of this thesis are:

- to analyze and compare existing applications that support shopping, maintaining inventory and list the features that they contain (important feature to be checked: voice control of the application);
- to analyze and compare the capabilities of the different voice assistants and choose one that best suits the requirements of the application;
- to create the analysis of key features and use cases of the "Shopmate" application;
- using the results of the analysis, to design and implement the "simple" "Shopmate" application that will support shopping and the maintenance of the inventory, and which essential part is the ability to be controlled by voice;
- to test the capability and usability of the application by user testing.

The structure of this bachelor thesis corresponds to these goals.

Chapter 3

The concepts of shopping support applications

As the voice assistant is one of the core concepts of this work, at first in this chapter there are described the operational principles of the voice assistants and the fact that the voice assistant can pass user's requests to the mobile applications. Furthermore, there are discussed instruments that can be used in the sphere of the shopping support applications: the shopping lists and the inventory management, and how usage of these instruments can be simplified with the use of the voice assistants. In addition, there are presented analysis and comparison of applications that already exist in the sphere of shopping support. Finally, there is defined the core of the "Shopmate" application.

3.1 Voice assistant

This section is a compilation of information from resources: [2], [3], [4], [5].

Voice assistant (VA) is a software that serves to facilitate and speed up communication between humans and devices. Assistant can be integrated into mobile phones, tablets, smartwatches, and many other gadgets. VA can interpret human speech and generate a corresponding answer. Users can ask assistant to execute some actions or to interact with an application .

Voice assistants use speech recognition, natural language processing, and speech synthesis to fulfill users' requests - see Figure 3.1.

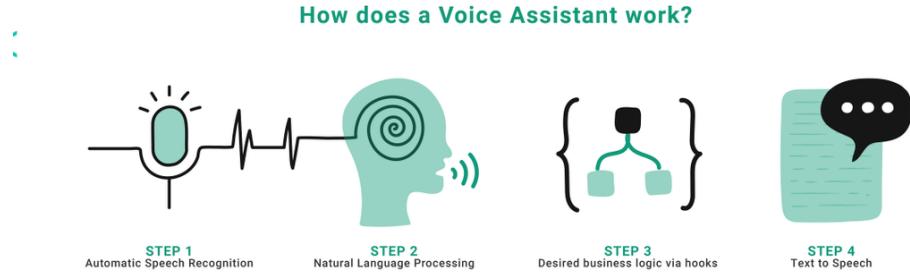


Figure 3.1: Components of Voice Assistant [6]

Communication with the voice assistant starts with the software permanently listening to everything around it and waiting for a special keyword. Once the keyword is recognized, the assistant starts to record the user's speech. Then it takes the speech, converts the signal from analog to digital, and analyzes the content trying to match it with words and phrases from the existing database. Then with the use of Natural Language Processing[7] (NLP) the assistant tries to understand the exact intention of the user. Thanks to NLP the user can say one command in different ways and it will still be recognized correctly. For instance: "Set the alarm tomorrow at 6 a.m." and "Wake me up tomorrow at 6 a.m." - will be understood the same way and it will create an alarm in the user's device set at 6 a.m. Then depending on the question or command, the assistant performs some actions and presents the result to the user. For example, it can report some information to the users, open an app, or play music. Finally, usually regardless of the command, VA gives the user some audio response as the confirmation of cooperation with the user. It takes the text response and generates voice answers with the use of speech synthesis. In order to achieve the goals of this bachelor thesis, it is important that the assistant is able to pass the recognized information to the application, so that the application does not need to be able to recognize speech, but simply process the user's query received in text format.

3.2 Inventory management

First instrument that can be used for simplifying shopping is inventory management. The term "inventory management" in this text means counting and tracking any possession that is present in the house. In other words, inventory can contain information about the kind and quantity of products present at the user's home to always know what he needs to buy and what he already has in supplies.

Usually there are two approaches to inventory management: a person can accurately keep track of everything manually or just buy random things that come to his mind. However, nowadays such processes can be optimized. The

easiest and the most effective way to optimize something is to have a smart application on the device, which can simplify and speed up routine processes.

Instead of keeping in mind or writing down on paper a lot of information (e.g.: there are 2 liters of milk at home, 1.5 more liters are bought, now there are 3.5 liters of milk at home), the user can tell it to a voice assistant that will insert, calculate and save all the information in the application - see Figure 3.2. And this data is available for the user at all times and all places where the user has the mobile phone with him/her.

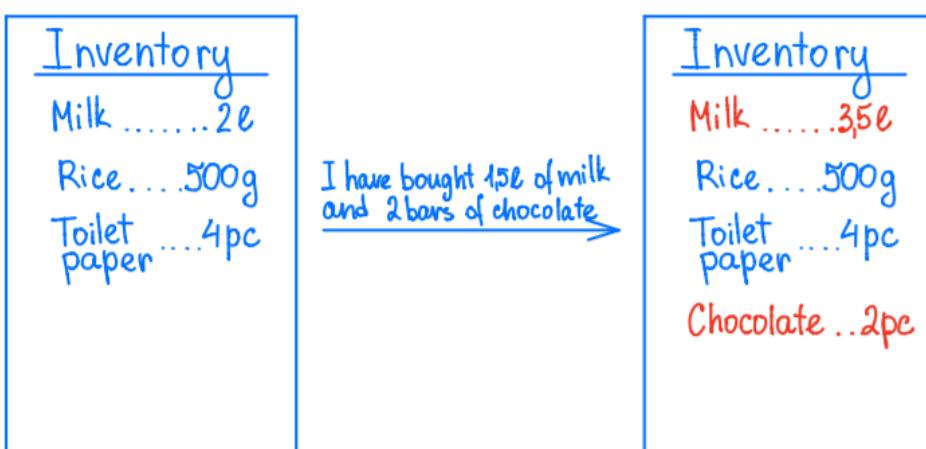


Figure 3.2: Increase in the number of bought items in the inventory via assistant

3.3 Shopping list management

Another instrument for simplifying shopping is the shopping list. It helps to always be aware of things to purchase. There can be several shopping lists. For example, 1 for a grocery store, 1 for a household shop, and 1 for a clothes shop. A shopping list can contain items and quantities. Just like with inventory management, the best implementation of this concept may be a smart application, which will help to keep all shopping lists in one place with quick access from anywhere, using just a smart device.

Everything that the user should keep in mind can be transformed so that it could be perceived by the assistant and the application. The user can set a critical level for all products. Then when the user, for example, informs the app that 1 piece of toilet paper is used, the application calculates that the number of pieces of toilet paper has reached the critical level - see Figure 3.3. The application automatically adds toilet paper to the shopping list - see Figure 3.4. So, the user does not need to make a revision of home inventory before shopping, because both an inventory and a shopping list are synchronized and therefore updated in the smart device. Consequently, it partly automatizes the shopping and home inventory routine.

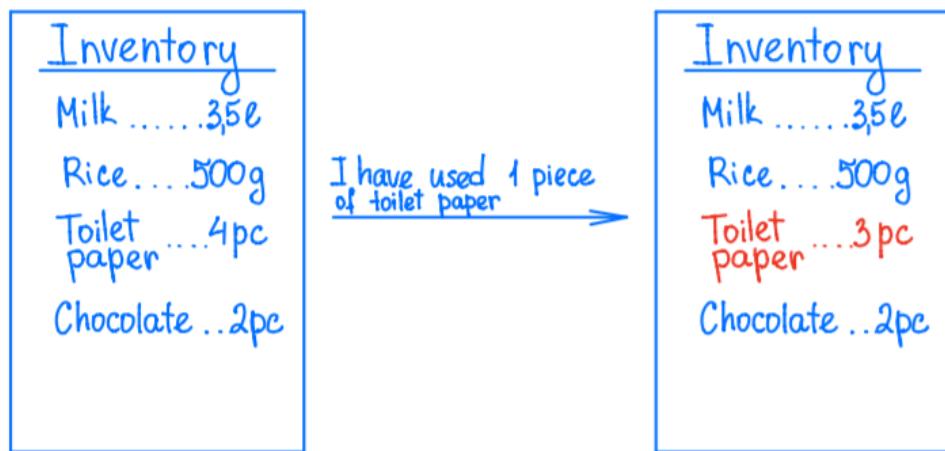


Figure 3.3: Decrease in the number of consumed items in the inventory via voice assistant

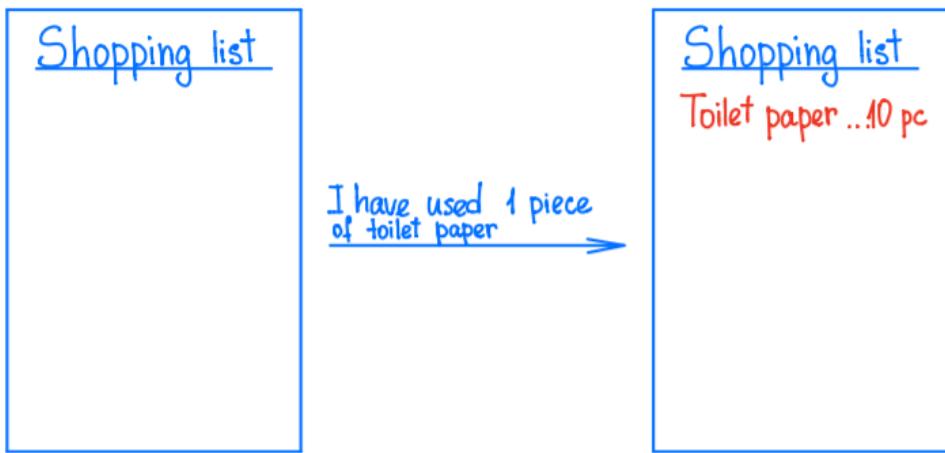


Figure 3.4: Increase in the number of items in the shopping list via voice assistant

So, such 3 instruments as inventory management, shopping lists and voice control of the application are core for shopping support applications and will be used as the main factors for searching applications which already exist in this sphere.

■ 3.4 Existing shopping support applications

Consideration of the existing applications for shopping support is important for a few reasons: to verify that development of the new "Shopmate" application is reasonable and to get inspired by these applications, their functionality, and interesting software solutions.

Search was limited to Android applications for the reasons described in

..... 3.4. Existing shopping support applications

the section 6.2. Therefore, the existing applications were searched in Google Play [8] using “inventory, shopping list” keywords. Google Play showed more than 100 applications for this request. In agreement with the supervisor, the first 10 applications were considered. To make analysis and comparison more structured, based on the previous chapter, some criteria that are sufficient for the shopping support applications were set:

- an app should have both a shopping list and inventory management parts;
- shopping list and inventory should cooperate: items can be added from inventory to shopping list - and back. At best, it should be done automatically when meeting set conditions;
- the app should provide a voice assistant which helps to use an app without typing anything into it.

So, according to these criteria, the table with a comparison of the chosen applications from the Google Play was created - see Table 3.1. Abbreviations used in the table: GA - Google Assistant, AA - Amazon Alex. Column "inv -> shl" describes if there supplement of the items from the inventory to the shopping list.

Table 3.1: Comparison of the “Inventory, shopping list” applications from the Google Play

Mº	App Name	Shopping list	Inventory	inv -> shl	Voice assistant	Other
1.	Food Checklist [9]	Yes.	Yes.	Yes, but manually.	No.	Only for food. Experation dates Bar-codes. Storage places. Sharing and synchronizing.
2.	Grocery Shopping List Listonic [10]	Yes.	No.	No.	No.	Suggest favorite products first. Sharing and synchronizing.
3.	AnyList [11]	Yes. Cate-gorization.	No.	No.	No.	Sharing and synchronizing features.
4.	Bring! Grocery Shopping List [12]	Yes. Cate-gorization.	No.	No.	Yes. GA and AA.	Sharing and synchronizing. Recipes features.
5.	Our Groceries Shopping List [13]	Yes.	No.	No.	Yes. GA and AA.	Sharing and synchronizing. Barcodes.
6.	Grocery Tracker Shopping List [14]	Yes. Cate-gorization.	Yes.	No.	No.	Shopping History. Item prices. Coupons.
7.	Shopping List [15]	Yes.	No.	No.	No.	Voice input of products. Addition of products from app’s “database”.
8.	Shopping list - My List [16]	Yes. Cate-gorization.	No.	No.	No.	Sharing and synchronizing. Discount cards. Recipes features. Item prices. Voice input of products.
9.	Shopping List [17]	Yes.	No.	No.	No.	Addition of products only from app’s “database”.
10.	Super Simple Shopping List [18]	Yes. Cate-gorization.	No.	No.	No.	Sharing and synchronizing.

Having considered all these competitive applications, it can be concluded that no application has satisfied all criteria simultaneously.

There are 2 apps that offer both shopping lists and inventory management (1st and 6th). But there is a lack of integration between these parts in the applications. In the 1st app, it is possible to add items from inventory to the shopping list, but this can be made only manually. In the 6th there is no cooperation between these 2 parts. Also, both 1st and 6th applications do not provide a voice assistant.

The 4th and the 5th apps have integration with a Voice Assistant, but none of them has inventory management features.

The 7th and the 8th applications provide only voice input, which means that the user can say the word and the word will be recognized and written down, but the interpretation of the speech, the evaluation of the user request, and generating an answer is not part of the voice input. So, voice input does not match the Voice Assistant's functionality.

With all that said, it can be concluded that no selected application fulfills the requirements to have a shopping list and inventory management simultaneously and to be managed with the voice. That is why creating the "Shopmate" application, which will fulfill these requirements, is reasonable.

Also, points from the "Other" column can be taken into consideration. These are the features that are not core for the application, but they increase the level of comfort for users. Therefore, these features will not be implemented within this bachelor thesis, but should be taken into consideration for the future extension and improvement of the "Shopmate" application.

These features are:

- possibility to share and synchronize all information;
- options to add items: with barcode, with voice input, from user's own database;
- discount cards and coupons;
- items' prices;
- shopping history;
- storage places;
- recipes' features.

3.5 The "Shopmate" application core

As it was confirmed in the previous chapter, the development of the "Shopmate" application is relevant. So in this section the core of the "Shopmate" application is stated.

The fundamental features of the "Shopmate" app are:

- management of shopping list;

3. The concepts of shopping support applications

- management of inventory;
- setting the critical level for every product;
- connection of the shopping list and inventory in the way that when the amount of product drops below set critical level in the inventory, the product is automatically added to the shopping list;
- being controlled by voice.

Based on these key features, the concept of the "Shopmate" application will be expanded and detailed further in the thesis.

Chapter 4

Voice assistants

In this chapter existing voice assistants are considered and compared in order to choose the best fit to be implemented into the “Shopmate” application.

4.1 Available voice assistants

The history of the voice assistants started in 1962 when IBM introduced the IBM shoebox - the first speech recognition machine, which recognized and responded to 16 spoken words [19]. Later in 2011, Apple introduced Siri, which became the first voice assistant for wide-audience [20]. Since then, technology has advanced and some more assistants with extended functionality were introduced. The graph below (see Figure 4.1) reflects that nowadays the most used voice assistants are Alexa from Amazon, Siri from Apple, and Google Assistant.

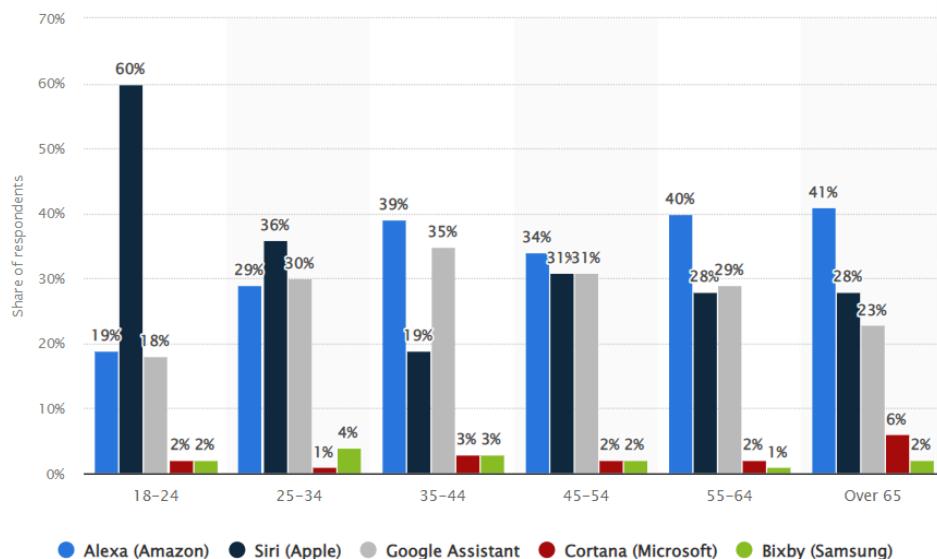


Figure 4.1: Share of voice assistants users in the United States in 2021, by product and age group [21]

As these assistants have significant preponderance over other assistants,

it can be assumed that the worldwide usage of these assistants also prevails. So, these 3 voice assistants will be further compared in order to choose one that is the best fit for the “Shopmate” app.

4.2 Comparison of assistants

Choosing one specific voice assistant can be complicated because it depends on every person’s preferences and specific usage cases that this person executes most. More broadly it also depends on the target audience: age categories, place of living etc.

As the “Shopmate” application is aimed at Android smartphones, the first criteria for the comparison are: ability to be used on smartphones and compatibility with the Android devices. Therefore, Siri from Apple should not be considered as an option for the “Shopmate” app, as it does not provide integration with Android devices. Whereas Google Assistant and Amazon Alexa are both compatible with Android smartphones.

A further important point of comparison between Google Assistant and Amazon Alexa is the fact that Google Assistant is pre-installed in all Android devices with Android 5.0+ [22], [23], while Amazon Alexa is pre-installed only in some phones from OnePlus, Sony, LG, and Motorola [24].

The information about the number of users is another important decision factor. But there is no complete information regarding that factor. While Google states that Google assistant was used by around 500 million people in 2019 [25], there are no clear statistics about the worldwide usage of Amazon Alexa.

Finally, support for languages is to be compared: Google Assistant supports 12 languages [26] and Amazon Alexa 9 languages [27].

The comparisons above can be summed up in the Table 4.1.

Table 4.1: Comparison of Google Assistant and Amazon Alexa

	Google Assistant	Amazon Alexa
Pre-installed in phones	All Android phones (Android 5+)	Phones from 4 brands
Number of users	500 million	-
Number of supported languages	12	9

According to the comparison, it can be concluded that Google Assistant is likely to have a bigger audience (even though the number of users cannot be accurately compared, Google still supports more languages). Moreover, Google Assistant requires less direct actions from the user, as the user does not need to explicitly install the Assistant. So, it can be assumed that more people will use the application with Google Assistant, than with Amazon Alexa.

4.2. Comparison of assistants

Consequently, Google Assistant is a preferable choice for the “Shopmate” app.

Chapter 5

Google Assistant

In this chapter existing options to implement Google Assistant are considered and compared, and one of them is chosen to be implemented in the “Shopmate” app. Then the operational principles of the chosen technology are described.

5.1 Options to Implement Google Assistant

From the technical side, there are 4 options for implementing Google Assistant: App Actions, Conversational Actions, Content Actions, Smart Home [28]. The Content Actions and the Smart Home options do not fit the concept of the “Shopmate” app: Content Actions is an option for web pages and Smart Home is for controlling smart home devices. So there are 2 options that may be suitable for the “Shopmate” app: App Actions and Conversational Actions.

5.2 Comparison of options

To compare App Actions and Conversational Actions, different resources were considered:

- guides from the official Google developers web [29], [30];
- codelabs from the official Google developers web [31], [32];
- discussions on StackOverflow [33].

Furthermore, 2 applications were compared: Spotify [34] and Money Manager [35]. Their voice assistants are based on App Actions and Conversational Actions respectively.

Based on the analysis of these resources, there can be pointed out the following differences between App Actions and Conversational Actions - see Table 5.3.

Table 5.3: Comparison of App Actions and Conversational Actions

App Actions	Conversational Actions
Can be used only in Android Devices.	Can be used wherever Google Assistant works.
Works inside the application. Needs this application to be installed on the device.	Works on the server. Does not need an application to be installed in the device.
Uses ordinary Google Assistant. 	Expands Google Assistant with its own “special” assistant.
After the query opens the graphical interface of the application with some feature.	Does not interact with the interface of any application.
For some applications hints with the abilities of the assistant can be found through the ordinary Google Assistant.	Can itself give hints with abilities of the assistant.

Firstly, as the “Shopmate” application is determined for Android devices, the first difference is not relevant. Secondly, as for the “Shopmate” application, it is pointless to use a voice assistant without having the application installed on the device. Finally, App Action interacts with the app interface and can directly open some features. In the case of the “Shopmate” app, it is necessary to be able to directly show the data from the application on the voice request, for example the content of the inventory list.

So, based on this comparison it was decided that App Actions is more suitable for the “Shopmate” application.

5.3 Operation principles of App Actions

This section is a compilation of information from resources: [29], [31], [36], [37].

App Actions can link finished Android Application with Assistant. It will enable users to access the app’s features without interacting with a graphical interface, but with the use of voice.

Building App Actions requires the usage of so-called built-in intents (shortly BII) [37], which reflects the most common ways people interact with the voice assistant. For example: “Open ExampleBrowser history”, “Order pasta from ExampleApp” or “Find review on ExampleHotel”.

There is a list of BII on the official website [38]. Some of these BII should be chosen in such a way that they reflect the features that would be provided to the user via voice assistant. These BII should be added to the shortcuts.xml

(configuration file in the project). For instance, to enable the application to take food orders via voice queries, the `actions.intent.ORDER_MENU_ITEM` [39] BII should be added to the `shortcuts.xml`. The intents have parameters, which are then passed to the application. For example, in the case of `actions.intent.ORDER_MENU_ITEM`, the name of the item can be stated in the request. So, when the user says “Order pasta from ExampleApp” the word “pasta” will be passed to the application as the value of the parameter “name”.

So, here is the example of the whole process of communication between the user, voice assistant, and the application:

1. The user says “Order pasta from ExampleApp”.
2. Assistant tries to match the requests with any BII that it knows.
3. In case the assistant recognizes some BII, it scans the app’s `shortcuts.xml` and looks for `actions.intent.ORDER_MENU_ITEM`.
4. If this intent is present in `shortcuts.xml`, the assistant generates an Android intent, which will launch the functionality that was requested and will pass the “name” parameter with the “pasta” value to the Android intent’s extras.
5. Application processes the request and gives the result to the user.

This process is visualized in the Figure 5.1.

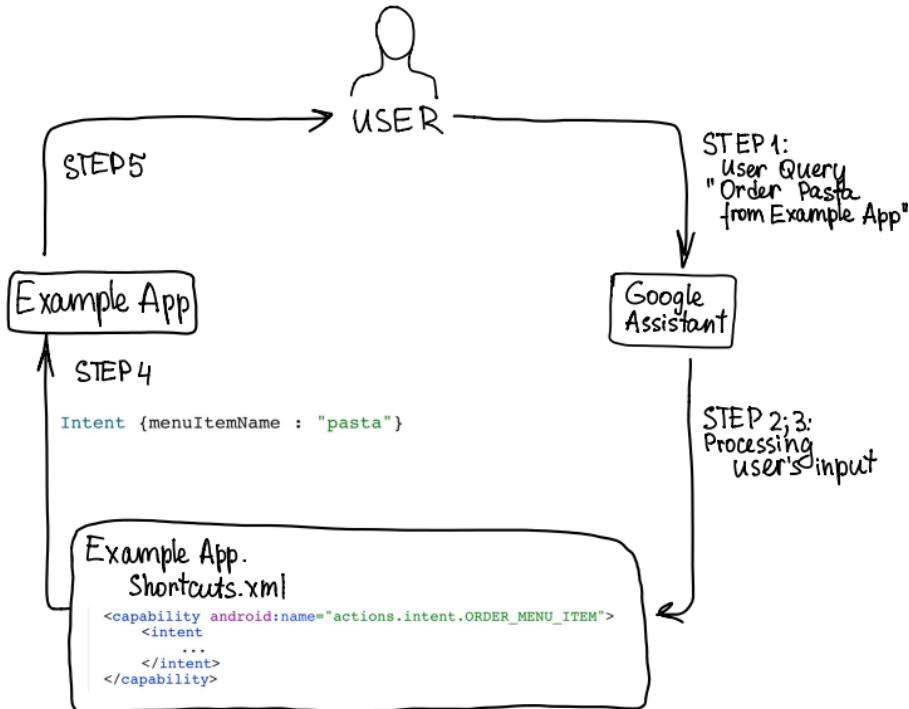


Figure 5.1: Communication between the user, voice assistant, and the application.

Chapter 6

Analysis of the "Shopmate" application

From the research in the previous chapters it follows that the "Shopmate" application's main features are managing inventory and shopping lists and being controlled by the voice. This application will be developed on Android platform and will integrate with Google Assistant via App Actions technology.

To prove the above stated, the easy proof of concept (POC) application was created as part of the project. This application allows the user to save any word to the main screen by voice. Based on the successful implementation of this POC application, it can be stated that implementation of the "Shopmate" application can be started with the use of opted technologies (Android, Google Assistant - App Actions).

So, at this point the core principles of the "Shopmate" application are stated and a more detailed analysis of this application can be started. This chapter covers the analysis of the "Shopmate" application: the functional and non-functional requirements, the class diagram, and the use case diagram along with the use cases.

6.1 Functional requirements

The functional requirements describe the specific behavior of the application [40].

As it is stated in the goals of this thesis, the "Shopmate" application should be "simple" meaning that the application will have the core functions that are necessary to fulfill the main idea of the app. At the same time, the additional advanced functions are not expected to be implemented within this thesis. The reason for this is that in the beginning, based on a "simple" application, it is needed to prove the demand for the application for users and the convenience of using such an application through the voice assistant.

Therefore, these are the functional requirements for the "Shopmate" application:

1. The application enables to add the new item (title, amount) to the shopping list (through the interface, by voice)
2. The application enables to increase and reduce the amount of the item in the shopping list (by voice)

6. Analysis of the "Shopmate" application

3. The application enables to edit the item in the shopping list (through the interface)
4. The application enables to delete the item from the shopping list (through the interface)
5. The application enables to show the list of all items in the shopping list (through the interface, by voice)
6. The application enables to add the new item (title, amount) to the cart (through the interface, by voice)
7. The application enables to increase and reduce the amount of the item in the cart (by voice, through the interface)
8. The application enables to edit the item in the cart (through the interface)
9. The application enables to delete the item from the cart (through the interface)
10. The application enables to show the list of all items in the cart (through the interface, by voice)
11. The application enables to save all items from the cart to the inventory
12. The application enables to mark item as bought/not bought in the shopping list (through the interface, by voice)
13. The application enables to add the new item (title, amount) to the inventory (through the interface, by voice)
14. The application enables to increase and reduce the amount of the item in the inventory (by voice)
15. The application enables to edit the item in the inventory (through the interface)
16. The application enables to delete the item from the inventory (through the interface)
17. The application enables to show the list of all items in the inventory (through the interface, by voice)
18. The application enables to set the critical level of the item in the inventory (through the interface)
19. The application enables to increase and decrease the critical level of the item in the inventory (through the interface)
20. The application enables to automatically increase the amount of the item in the shopping list if the amount of this item in the inventory is beyond its critical level

6.2 Non-functional requirements

The non-functional requirements describe how the application should perform [40].

Most of the non-functional requirements come out naturally from the concept of this application. However, some of them should be explained. Firstly, the selected platform on which the "Shopmate" application will run is Android. I have experience of using only Android devices, therefore the development of applications on Android is more familiar to me. Furthermore, English language was chosen as the main language of the application due to the will for the application to be available globally.

Here is the full list of the non-functional requirements for the "Shopmate" application:

1. The application must have a simple and handy interface for the users
2. The application must provide the ability to be controlled by voice and from the keyboard via interface
3. The application must integrate with the Google Assistant
4. The application must function on Android (supporting Android versions 5 and higher because of the compatibility with the App Actions[29])
5. The application must have English interface and be controlled by voice in English

6.3 Class diagram

Class diagram describes the structure of the system. It defines classes, attributes of the classes, relationships and multiplicities between classes [41]. In the Figure 6.1 the class diagram of the “Shopmate” application can be seen.

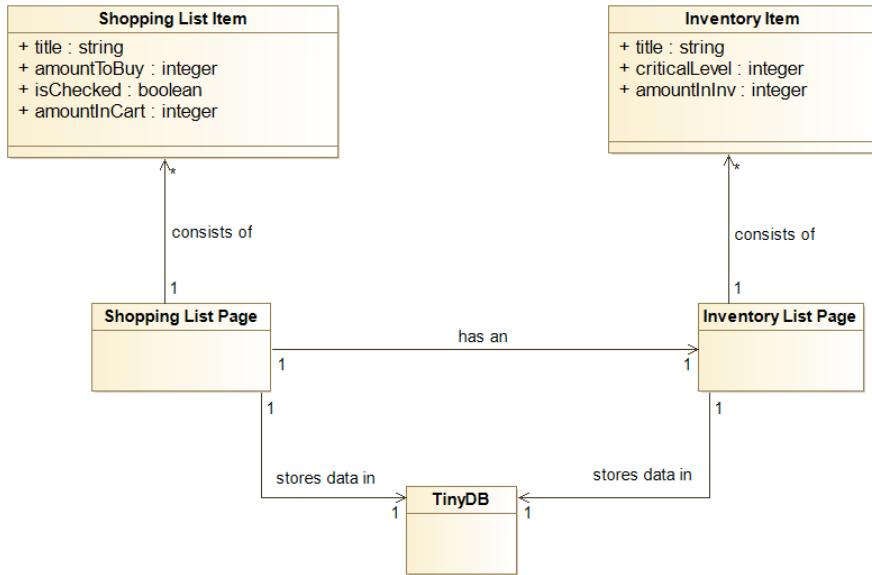


Figure 6.1: Class Diagram of the "Shopmate" application

There are two main pages in the application: shopping list and inventory list. Each of them contains a list of the relevant items.

Shopping List Item class besides the title contains information about the amount of the item that person wishes to buy at the supermarket - "amountToBuy", and the amount that person already has in the cart, when he is at the supermarket - "amountInCart". Moreover, it has the "isChecked" attribute which shows if the user checks the check-box next to this item in the shopping list or not.

Inventory Item class contains 3 attributes: "title", "amountInInv" which corresponds to the amount of this item that user has at home and the "criticalLevel" attribute that describes the minimal amount of the item that the user always wants to have at home.

Shopping List Page contains the instance of the Inventory List Page, so that they can interact with each other.

Finally, both Shopping List Page and Inventory List Page contain the instance of the TinyDB[42], in which they store data.

6.4 Use cases

Use case is a list of actions that describe the activity that is supported by the application. Use cases are created in such a way that they cover all functional requirements of the application. Actor is the external entity that uses or affects the application. Use case diagram shows relationships between actors and use cases.[43]

6.4.1 Actors

There are 2 actors in the "Shopmate" application: the User and the Application - see Figure 6.2. The user is associated with the use cases that are launched by the user. The application is associated with the use cases that are launched automatically from the other use cases.



Figure 6.2: Actors.

Due to the complexity of the use case diagram, it is divided into 2 parts according to the actors.

6.4.2 Use case diagram for user

For use cases of "User" actor - see Figure 6.3.

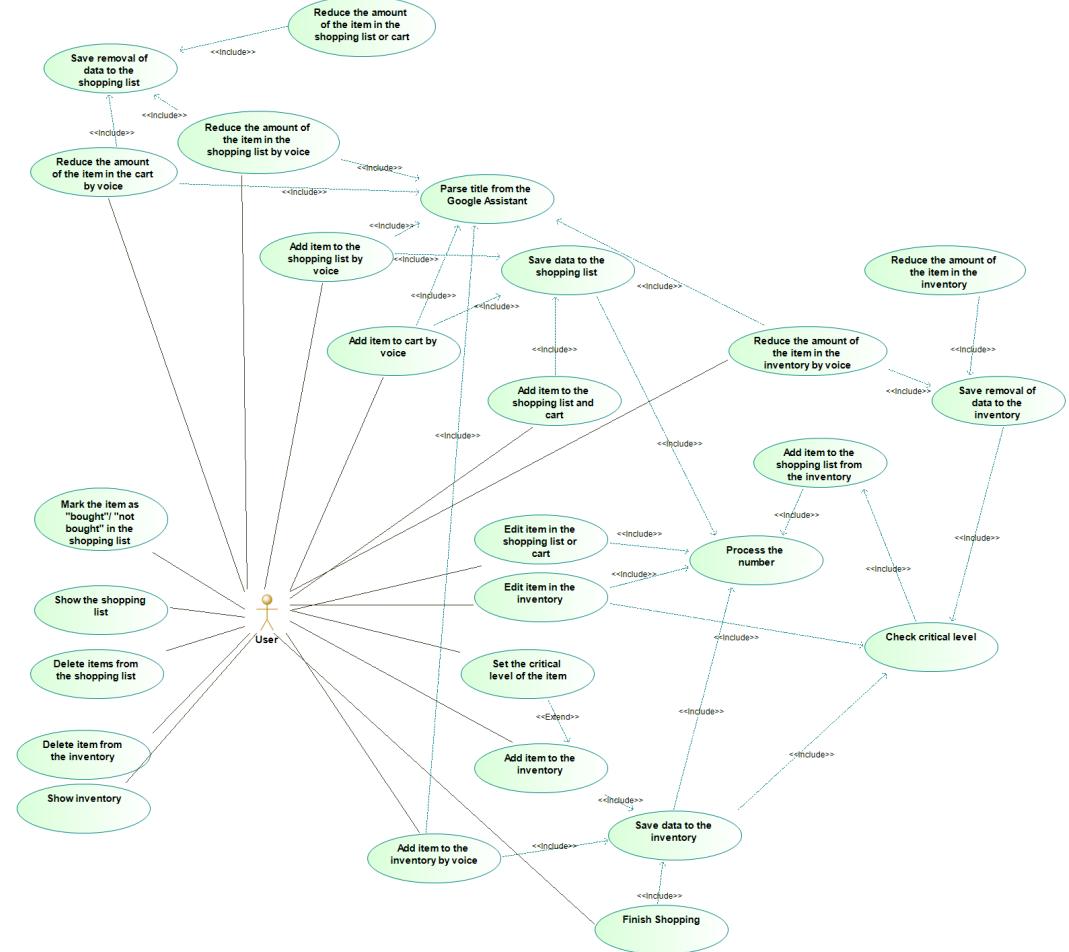


Figure 6.3: Use case diagram for the User actor.

6.4.3 Use case diagram for application

For use cases of "Application" actor - see Figure 6.4.

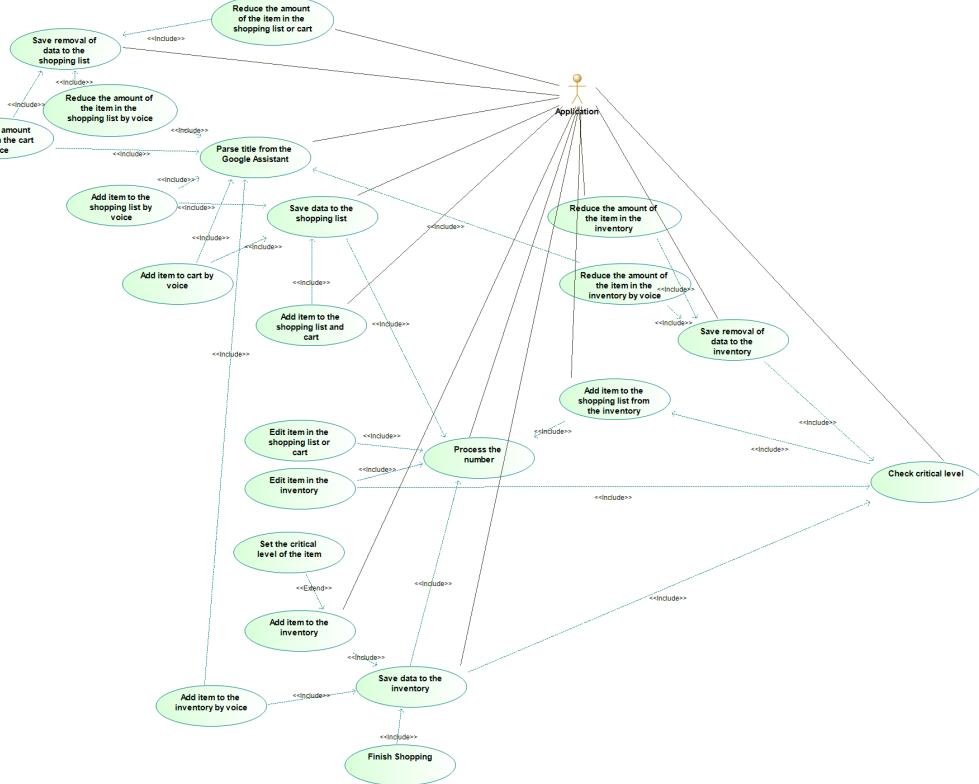


Figure 6.4: Use case diagram for the Application actor.

The whole diagram can be found in the electronic attachments A.

6.4.4 Use case scenarios

Every use case has a scenario. The scenarios of all use cases can be found in the electronic attachments A. Note: as use cases often refer to each other (with “include”, “extend” or GOTO UC), in more complex cases it is needed to state what parameters one use case passes to the other. Where it is needed, it is pointed out in round brackets after the name of the use case.

Here the example of the scenario of the one use case can be seen:

Add item to the shopping list by voice

Description: The application allows the user to add the item to the shopping list by voice.

Launch: The user says “Okay, Google..”

Scenario:

1. The user says: “Okay, Google, open \$application_name and add \$item_amount\$item_title to shopping list” (where \$application_name is replaced with the name of the application, \$item_amount is replaced with the number and \$item_title is replaced with the name)
2. The application enables to increase and reduce the amount of the item in the shopping list (by voice)
3. The application enables to edit the item in the shopping list (through the interface)
4. The application enables to delete the item from the shopping list (through the interface)
5. The application enables to show the list of all items in the shopping list (through the interface, by voice)
6. The application enables to add the new item (title, amount) to the cart (through the interface, by voice)

Chapter 7

Design of the "Shopmate" application

The design of the software is needed to suggest the conceptual solution (in software and hardware)[44]. This chapter describes the design of the "Shopmate" application along with the wireframes of the application and the deployment diagram.

7.1 Wireframes

As the "Shopmate" application's interface is supposed to be simple, the prototypes were not created. The application replaces prototypes on its own. In order to show the layout of the application, wireframes were created instead. They show the basic structure of each page of the application without detailed design [45], so that all stated above use cases are reflected in the application.

In the Figure 7.1 there is a main "Shopping List" page. This page is the first one to appear after the start of the application. Here there can be seen items along with the title, amount in cart (in the left column) and the amount to buy (in the right column). There is also a clickable check-box next to each item. There is also a plus button, which allows users to add the new item to this list. There is the "Finish Shopping" button that allows users to add all items from the cart to the inventory. Downside there is a menu that can take users to the Inventory page. There is also the "edit" icon on the right top corner that switches the shopping list into the edit mode.

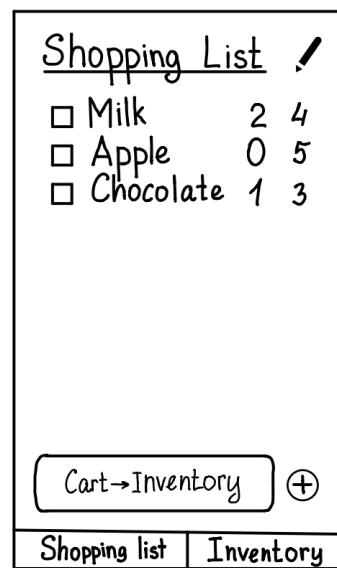


Figure 7.1: "Shopping List" page

In the Figure 7.2 there is a "Shopping List" page in the edit mode. In this mode, users can delete items by clicking the minus button next to the item. The "tick" icon on the right top corner switches the page back to the normal mode.

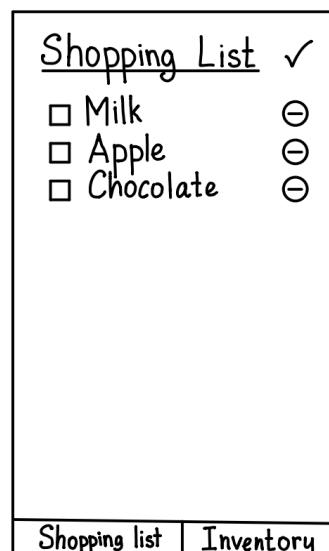


Figure 7.2: "Shopping List" page, edit mode

In the Figure 7.3 there is an "Inventory" page. In the page there are the items along with the title and amount in inventory. There is the plus button as well, that allows the user to add the new item to this list. Below there is a menu that can take users to the Shopping List page.

<u>Inventory</u>	
Cheese	2
Tea	1
+	
<u>Shopping list</u>	<u>Inventory</u>

Figure 7.3: "Inventory" page

In the Figure 7.4 there is an "Add Item" page for shopping list items. There are 3 fields for input: title, amount to buy and amount in cart. There is also an "add" button to save the item to the shopping list.

<u>Add Item</u>	
<u>Title</u>	-----
<u>Amount_to_buy</u>	-----
<u>Amount_in_cart</u>	-----
Add	

Figure 7.4: "Add Item" page for shopping list items

In the Figures 7.5 and 7.6 there is an "Add Item" page for inventory items. There are 2 fields for input: title and amount. There is a plus button that allows users to add the critical level (the collapsed version is on the 7.5 and the expanded version is on the 7.6). When the section with the critical level is expanded, there is the next input field - critical level. Finally, there is an "add" button to save the item to the shopping list.

A hand-drawn wireframe of a mobile application screen titled "Add Item". It features two input fields labeled "Title" and "Amount", each preceded by a short horizontal line. Below these fields is a button labeled "+Add critical level". At the bottom is a rectangular "Add" button.

Figure 7.5: "Add Item" page for inventory items

A hand-drawn wireframe of a mobile application screen titled "Add Item". It features three input fields labeled "Title", "Amount", and "Critical_level", each preceded by a short horizontal line. Between the first two fields is a button labeled "-Remove critical level". At the bottom is a rectangular "Add" button.

Figure 7.6: "Add Item" page for inventory items with visible critical level

In the Figure 7.7 there is an "Edit Item" page for shopping list items. There are 3 fields that can be changed: title, amount to buy and amount in cart. There is also a "save" button to save the changes to the shopping list.

The wireframe shows a rectangular form with a title 'Edit Item' at the top. Below it is a field containing the word 'Apple'. Underneath 'Apple' are two input fields: 'In cart: 0' and 'To buy: 5'. At the bottom of the form is a single button labeled 'SAVE'.

Figure 7.7: "Edit Item" page for shopping list items

In the Figure 7.8 there is an "Edit Item" page for inventory items. There are 3 fields that can be changed: title, amount and critical level. There is a "delete item" button to delete the item from the inventory. There is also a "save" button to save the changes to the shopping list.

The wireframe shows a rectangular form with a title 'Edit Item' at the top. Below it is a field containing the word 'cheese'. Underneath 'cheese' are two input fields: 'Amount: 2' and 'Critical level: 1'. At the bottom of the form are two buttons: 'DELETE ITEM' and 'SAVE'.

Figure 7.8: "Edit Item" page for inventory items

7.2 Deployment diagram

In the Figure 7.9 the deployment diagram of the full system can be seen. Deployment diagram describes the HW architecture of the system and the deployment of the SW into HW.[46]

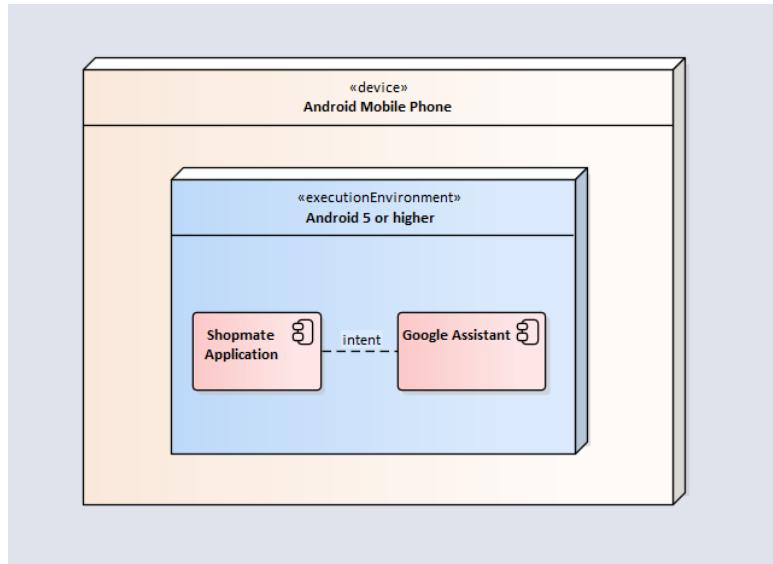


Figure 7.9: Deployment Diagram

From the diagram it is seen that the application is not directly connected to any server. The Google Assistant communicates with the application through Android intents. The data is saved inside the application, so any external database is not required either. Therefore, the application can be used in both online and offline modes. It can be used in offline mode with full range of the functionality, except the fact that it will be available only for controlling via the interface. The application should be in online mode to enable voice control of the application, as the Google Assistant requires the Internet connection.

Chapter 8

Implementation of the "Shopmate" application

This chapter describes the process of the implementation of the "Shopmate" app which was created according to standard concepts of the Android applications[47]. It covers the used technologies, the process of integration of the Google Assistant along with the difficulties that appeared in the process. The configuration files needed for Android application and integration with Google Assistant are described further. Then the process of processing the intent in the "Shopmate" application is discussed. Finally, the interface of the application is presented .

8.1 Used technologies

The technologies that were used during the development of the "Shopmate" application will be described below.

- **Operational system.** Implementation took place on Windows 10. During the development the application was tested on a physical device with Android 10.
- **Integrated development environment (IDE).** There is plenty of the IDE for Android development[48]: Android Studio [49], Eclipse [50], Visual Studio [51] etc. The Android Studio was chosen for the implementation as I was using this IDE during studies. It has an intuitive interface, provides layout editor, emulator and supports the versioned system GIT. So, it fully meets the needs for Android development.
- **Version control system.** Gitlab was chosen as the version control system since this is a popular version control system that I am used to operating both at school and private projects[52].
- **Testing tool.** Google offers the App Action testing tool[53], [54] for an easier and faster testing of the integration with the Google Assistant. This tool was used during implementation and testing of the "Shopmate" application.

- **Programming language.** Java [55] was chosen as the programming language for the application because it is one of the most popular languages, it has huge community support and good documentation [56]. Also, during my studies I got acquainted with Java the most. Due to the combination of these reasons Java was preferred to Kotlin and other programming languages.
- **Build tool.** Choosing between Maven[57] and Gradle[58] build tools, Gradle was chosen as it is officially recommended by Android [59].

Moreover, the code from open sources was used in the implementation:

- TinyDB[42] - the class that simplifies the process of saving data in Android applications.
- Class to convert words to integers[60] was used with slight changes from my side.

■ 8.2 Integration of the Google Assistant to the application

The development of the application is closely related to the implementation of the integration with the Google Assistant and there is not much information on this topic. Therefore, the implementation was realized with the method of trials and errors.

■ 8.2.1 Built-in intents

Theoretical principles of the Built-in intents [61] were described in the section 5.3. In this section the Built-in intents are described from the practical point of view. App Actions built-in intents technology was chosen as the first solution. Built-in intents are pre-defined intents, which have pre-defined fields and queries for Google Assistant. After this option was implemented and tested, it was rejected due to a few reasons. Firstly, these intents have fixed queries through which users can communicate with the application. It means that there is no possibility to enhance the user experience (e.g. adjust the request to the specific needs) by the application developers. Secondly, the faults in this technology were found during testing: some of the fields that are defined in the documentation are not supported. For example: in Create Thing BII[62] if the "name" and "description" fields are defined, the assistant can not separate one from the other and writes all the information into the "name" field. It might cause problems in processing input data. Finally, the fact that the intents are fully pre-defined means that the development is limited in collecting of input data, there is no possibility to add any custom fields or rename them. Due to these reasons, it was decided to search for other solutions.

8.2.2 Custom intents

Second chosen solution was custom intents[63]. Custom intents have some query patterns, however, everything else is to be done by the developer - specific queries, number of fields, their type and titles. It also has some limitations[64]: only certain data types are available for parameters, a maximum of two text parameters are supported per query and only the en-US locale is supported (device and Assistant language settings must both match en-US). But these limitations are not determinative for this specific application.

In custom intents user can define a few attributes for each intent:

- the intent itself
 - name
 - parameters (each may have a type: Text, Number e.t.c.)
- the queries for the intent
- the shortcuts for the specific parameter (see explanation in "Inline Inventory" subsection 8.2.3)

This solution turned out to be suitable and it is used in the final version of the “Shopmate” application. Example of implementation of the custom intent in code can be found in subsection 8.2.5.

8.2.3 Inline Inventory

As the voice assistant should simplify interacting with the application, it is important for defined voice requests to be simple and close to real-life conversation. This can be achieved by using “natural” queries and different synonyms, so the user can use the queries that are more common for him. For example: there can be not only “remove item”, but also “delete item” query, that will end up with the same result. Inline inventory can be used for these purposes.

Inline inventory[65] enables uniting different inputs, so that they are perceived as the same one. For example, words “delete” and “remove” are synonyms in context of the “Shopmate” application and can be considered as the same input. [1] Defining shortcuts is a way to implement inline inventory. In one shortcut we group all the words that should be processed as synonyms and give the same result. How it is technically realized can be seen in subsection 8.2.5.

An example of groups of the “synonymic” words from the “Shopmate” application:

- add, save
- remove, delete
- open page, show page

- cart, shopping cart, shop-cart
- to, into

8.2.4 Similar - sounding words

During development and testing it was discovered that certain words and phrases are sometimes confused by Google Assistant due to unclear or non-native pronunciation. Here is the list of corresponding words that were found during the development and testing, but there are undoubtedly more of them:

- save - safe - thief
- add - ad
- check - chirik - Czech
- show - sew - soul - so
- inventory - inventor - inventory - in Ventura
- cart - carrot - cut - card - court
- uncheck - and check - unchecked
- help - health
- two - to - too
- three - free
- four - for
- ten - turn

These words are used in the application's queries for the Google Assistant. When Google Assistant recognizes them incorrectly, then in most cases it launches a search on the net or a search in the Play Store. It can confuse the user. That is why it is important for Google Assistant to recognize them correctly to act according to the user's expectations. It was revealed that in some cases Google Assistant itself recognizes misidentification and acts accordingly. For example, even when it hears "sew" it acts like it is "show" in this query. However, to make sure that query recognition works consistently, all these groups of "similar" words are handled in the application, so as a result, all words from one group are recognized as the one right word. It is achieved by using the "Inline Inventory" described above. Similar sounding words are grouped together to give the user the right result.

This method cannot cover all situations as all the words that will be used by users cannot be predicted. Therefore, to make the application more complex and consistent neural networks and other sophisticated technologies may be used in the future.

8.2.5 Configuration files

In this subpart, there is a description of the configuration files that the Android Application set on its own, and files that make it possible for the Android application to cooperate with the Google Assistant.

AndroidManifest.xml

Essential settings of the application are set in the `AndroidManifest.xml`. The settings include name, logo, activities, and the definition of the default activity. Apart from the basic setting, there are metadata defined in this file which enable the application to use the App Actions and look for its definition in `shortcuts.xml` - see Listing 8.1.

Listing 8.1: `AndroidManifest.xml`

```

1 <meta-data
2     android:name="android.app.shortcuts"
3     android:resource="@xml/shortcuts" />

```

shortcuts.xml

`shortcuts.xml` contains intents for itself and the shortcuts for inline inventory. This is how custom intent can look like - see Listing 8.2. Intent has name (line 2), target package (line 5), target class (line 5), and parameters (lines 7 - 17). Each parameter should have the type (lines 10, 14) or its own shortcuts (in the example below the last parameter (lines 15 - 17) does not have type, because it has shortcuts described below - see Listing 8.3). Intent also has a link to its query patterns.

Listing 8.2: `Shortcuts.xml` - custom intent

```

1 <capability
2     android:name="custom.actions.intent.ADD_ITEM_TO_CART"
3     app:queryPatterns="@array/AddItemToCart">
4     <intent
5         android:targetPackage="cz.cvut.fel.bachelorapp"
6         android:targetClass="cz.cvut.fel.bachelorapp.MainActivity">
7         <parameter
8             android:name="number_of_items"
9             android:key="number_of_items"
10            android:mimeType="https://schema.org/Text" />
11         <parameter
12             android:name="add_to_cart_name"
13             android:key="add_to_cart_name"
14             android:mimeType="https://schema.org/Text" />
15         <parameter
16             android:name="action"
17             android:key="action" />
18     </intent>
19 </capability>

```

The code described below - see Listing 8.3 is the example of the inline inventory for the “action” parameter used in the above intent (lines 15 - 17). Shortcut has its ID (lines 4, 15) that is used as a value that is then provided to the intent. Shortcut also has the link to the array of values (lines 8, 19) that are grouped into this shortcut.

Listing 8.3: Shortcuts.xml - shortcuts

```

1 <shortcut
2     android:shortcutShortLabel="@string/shortcut_short_label"
3     android:shortcutLongLabel="@string/shortcut_long_label"
4     android:shortcutId="ADD">
5     <capability-binding
6         android:key="custom.actions.intent.ADD_ITEM_TO_CART">
7             <parameter-binding
8                 android:key="action"
9                 android:value="@array/add" />
10            </capability-binding>
11        </shortcut>
12
13 <shortcut
14     android:shortcutShortLabel="@string/shortcut_short_label"
15     android:shortcutLongLabel="@string/shortcut_long_label"
16     android:shortcutId="REMOVE">
17     <capability-binding
18         android:key="custom.actions.intent.ADD_ITEM_TO_CART">
19             <parameter-binding
20                 android:key="action"
21                 android:value="@array/remove" />
22            </capability-binding>
23        </shortcut>

```

arrays.xml

arrays.xml is a file that contains queries and values. As seen in the code above, the intent has `app:queryPatterns` parameter (line 3) which is linked to the array in the arrays.xml. This array can look as in the Listing 8.4.

Listing 8.4: arrays.xml - queries

```

1 <string-array name="AddItemToCart">
2     <item>$action $number_of_items $add_to_cart_name to cart</item>
3     <item>$action $number_of_items $add_to_cart_name from cart</item>
4     <item>$action all $add_to_cart_name from cart</item>
5 </string-array>

```

This is how queries along with the input parameters for Google Assistant are defined for custom intents.

Also, arrays.xml contains arrays of values for inline inventory - see Listing

8.5, for example for the “action” parameter from the previous code (8, 19):

Listing 8.5: arrays.xml - arrays of values for inline inventory

```

1 <string-array name="add">
2   <item>add</item>
3   <item>ad</item>
4   <item>save</item>
5   <item>safe</item>
6   <item>thief</item>
7 </string-array>
8 <string-array name="remove">
9   <item>remove</item>
10  <item>delete</item>
11 </string-array>
```

For example, the words “add” and “save” are synonyms in the context of this query, so they are grouped into a shortcut with the id “ADD” (line 4). Other words are the words that sound similar to “add” and “save” and may be mispronounced by the user.

■ build.gradle

This file defines the application’s build configuration on project and module levels [66]. The project-level file contains repositories and dependencies for the entire project. The module-level file “allows you to configure build settings for the specific module it is located in” [67]. Both these files mostly contain default settings. Only a few changes were made:

- ApplicationId was changed to be unique.
- minSdk was set on 25 as in earlier versions the custom shortcuts didn’t work. Although minSdk 25 still covers the majority of the devices, MinSdk 25 means that the application will be available for the Android 7.1+ and that it covers 91.3% of all devices [68], [69].
- Some dependencies for the correct work of the application were added.
- Support of vector drawables was added.

Note: in the non-functional requirements it is states that the application should work on Android 5+. Although due to the implementation reasons described above it is changed on Android 7.1+.

■ 8.2.6 Request processing

The main principles and configuration needed to integrate Google Assistant into the Android application were described above. To summarize the whole process of the user’s voice request processing, it can be described in steps on a specific example:

1. The user wants to state that he needs to buy 4 apples.

2. The user says “Okay Google, open Shopmate and add 4 apples to inventory list”.
3. Google Assistant looks for the Shopmate app’s shortcuts.xml and tries to find there any capability that matches the request stated in step 2.
4. Google Assistant finds the "custom.actions.intent.SAVE_ITEM_WITH_NUMBER" capability, that refers to the "SaveItemWithNumber" array, that contains `add $number_of_items $add_item_name to $list` query pattern.
5. This capability contains intent with 3 parameters: `number_of_items` (text type), `add_item_name` (text type), `list` (type is defined by the shortcuts). As it can be seen, the query pattern from step 4 refers to the same parameters.
6. Google Assistant matches data from the query to parameters: `number_of_items = 4`, `add_item_name = apples`.
7. `list` parameter gets value differently:
 - a. The assistant looks for the shortcuts that are connected to this parameter.
 - b. It finds the one that refers to the array with the value “inventory list”.
 - c. In this case, it is the shortcut with id "INVENTORY".
 - d. Then the "INVENTORY" value is written into the `list` parameter.
8. Google Assistant sends the intent to the “Shopmate” application with `$number_of_items`, `$add_item_name`, and `$list` parameters’ values passed as intent’s extras.
9. Then the application gets the intent and can process it. For example, the simplified version of the code that processes the intent is given - see Listing 8.6. This code mainly does 3 things: pulls out the values from the intents’ extras; validates and processes these values; depending on validation results, opens the page for manual filling of fields or passes further the data to save them.

For clarity, this process is visualized in the 8.1. Single steps respond to the relevant numbers in the picture. The color highlights are used to visually represent the parameters and parts of the voice query: green is used for the `$number_of_items` parameter, blue is used for the `$add_item_name` parameter, pink is used for the `$list` parameter, and yellow is used for the non-changing pre-defined parts of the voice query.

8.2. Integration of the Google Assistant to the application

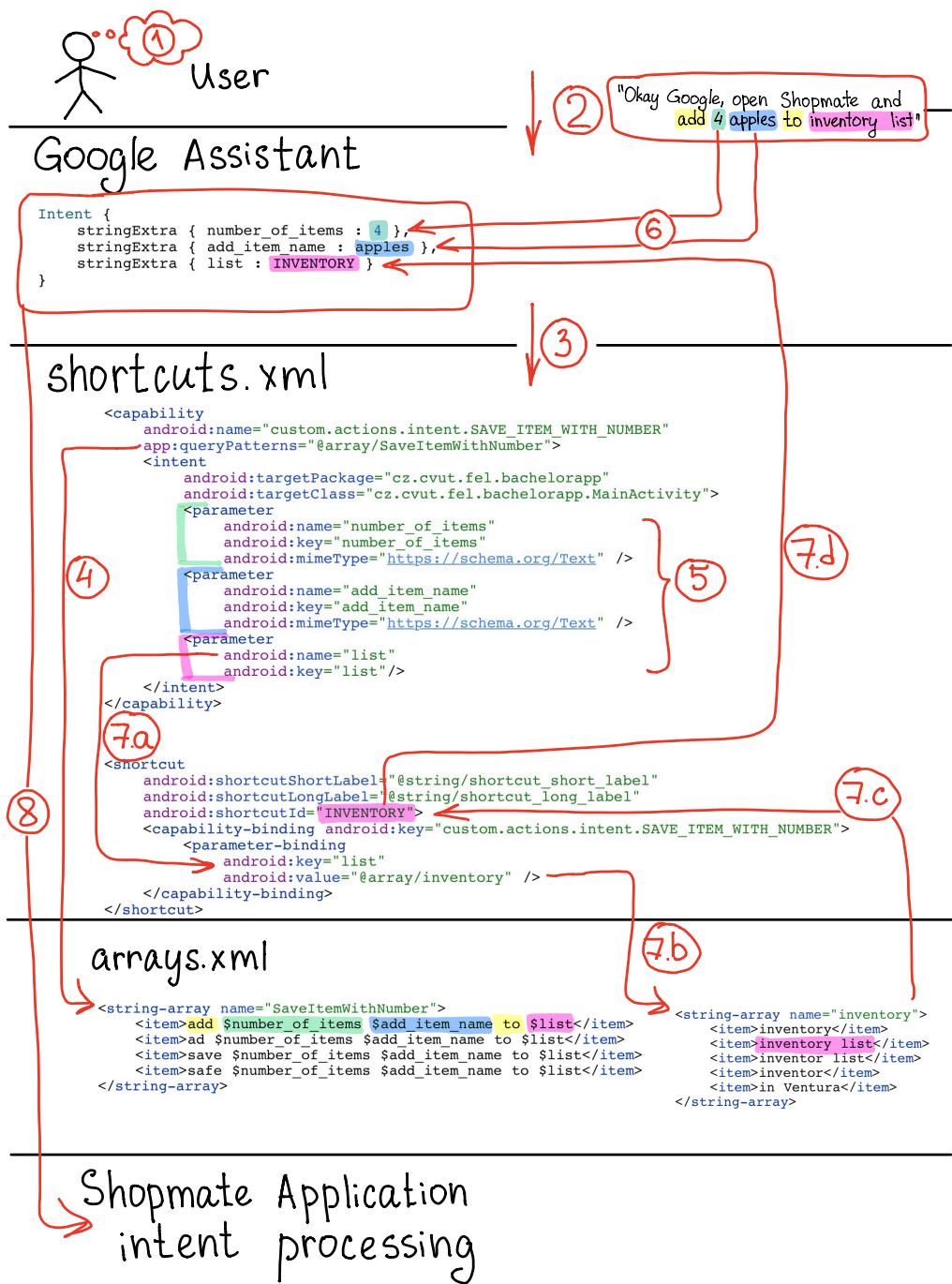


Figure 8.1: Request processing

Listing 8.6: Intent processing

```

1  if (intent.getStringExtra("list") != null &&
2      intent.getStringExtra("list").equals("INVENTORY")){
3      String new_item = intent.getStringExtra("add_item_name");
4      if (new_item != null) {
5          new_item = parseTitleFromGA(new_item);
6          if (new_item == null) {
7              goToAddItemPage("");
8          } else {
9              String number = intent.getStringExtra("number_of_items");
10             try {
11                 int num = Integer.parseInt(number);
12                 if (num < 0){
13                     throw new NumberFormatException();
14                 }
15                 addItemToList(new_item, num);
16             } catch (NumberFormatException e) {
17                 Integer num = new WordToNum(number).convert();
18                 if (num != null) {
19                     addItemToList(new_item, num);
20                 } else {
21                     goToAddItemPage(new_item);
22                 }
23             }
24         }
25     }
}

```

The whole project along with the entire source code and configuration files can be found on the <https://gitlab.com/samatali/bachelor-thesis-app> or in the electronic attachments A. APK of the "Shopmate" application can be found in the electronic attachments A as well.

8.2.7 Application logic

Big part of the application's logic is CRUD operations (create, read, update, delete) for the shopping list items and inventory list items. Most of these operations are available both via interface and voice assistant.

There are also some specific functionalities in the "Shopmate" application. All of them as well as CRUD operations are described in the use cases (see electronic attachments A) in detail, but for better orientation few features can be described here. For example, there is a "finish shopping" feature that allows users to add all items that were recorded in the user's cart in one-click (or one voice query) to the inventory or reduce their amounts in the shopping list. Moreover, the application always controls the critical level of the items in the inventory and when the amount of the item in the inventory drops below the critical level, this item is automatically added to the shopping list.

Another thing that was considered during implementation of the logic is that the titles of the items should be matched with each other correctly. So,

the titles are processed in such a way to be case insensitive ("Apple" and "apple" and "aPpLe" are considered to be equal) and to remove redundant spaces at the beginning and end of the titles ("apple" and " apple " are considered to be equal). This allows to unite the items with the same (with slight differences described above) titles and sum up their amounts instead of creating 2 different items.

The specific issue that is not covered in this work is the processing of the plural forms of the titles. As the English language has many groups of words that obey different rules and there are exception words, it is a complex issue to automate the process of the depluraling of all words. Therefore, this issue has not been solved yet, and it is recorded into the list of feature improvements that will follow in the section 10.2.

A further thing to be considered is that Google Assistant can recognize the number that the user says as the word (e.g: "twenty two"), so the application should transform it into integer 22. In order to make such an action, the code that was found on the Internet was used [60]. This code was slightly adapted to the "Shopmate" application - there were added such words as "for", "too" respectively, that are similar sounding to the "four", "two" etc. This code recognizes each separate word, converts them to the integer and then sums them up, so the amount in integer format can be saved to the application.

Some other non-standard cases are also described in the section 9.2.

The last specific thing that is worth describing is the TinyDB[42]. The "Shopmate" application does not need to store data in the external database, as the amount of the information to store is small. The use of the shared preferences[70] is sufficient. This allows to save data on the application restart or the recreation of the Android activities[71]. But saving the non-primitive types and arrays to the shared preferences is a tricky issue, that is why TinyDB was chosen as the instrument for simplifying saving data to the shared preferences. With the help of TinyDB the saving of the array of custom objects to the shared preferences is the one row of the code - see Listing 8.7.

Listing 8.7: Saving of data to the shared preferences through TinyDB

```
1 tinydb.putListObject("shoppingListItem", new
    ArrayList<>(shoppingListItemList));
```

Getting the data from the shared preferences is easy as well - see Listing 8.8.

Listing 8.8: Getting data from the shared preferences through TinyDB

```
1 List<Object> list = tinydb.getListObject("shoppingListItem",
    ShoppingListItem.class);
```

■ 8.2.8 User interface

The concept of the interface of the "Shopmate" application was chosen based on the wireframes shown above and the non-functional requirement which states that the application must be simple and handy to use. The concept of the interface is stated as the analogy to paper and pen. Light background, black text, crossing of the items, and not many details should help users to get used to the application quickly.

The appearance of the most pages reflects the corresponding wireframes, so below there are shown pages that have some differences and the description of these changes.

In the "Shopping List" page - see Figure 8.2, there appeared a new button in the left top corner. Clicking this button takes the user to the new "Help" page. There also appeared small icons on top of the "amount in cart" and "amount to buy" columns to clarify the meaning of these columns. Finally, after testing the text on the button was changed from "Cart -> Inventory" to "Finish Shopping".

The screenshot shows the "Shopping List" page. At the top right is a question mark icon. Below it is the title "Shopping List" with a refresh symbol. The main content is a table with the following data:

	🛒	📋
<input checked="" type="checkbox"/> Water	2	0
<input checked="" type="checkbox"/> Cheese	1	1
<input type="checkbox"/> Apple	0	1
<input type="checkbox"/> Grape	2	5
<input checked="" type="checkbox"/> Strawberry	5	4
<input type="checkbox"/> Bread	0	2
<input type="checkbox"/> Orange	0	5

At the bottom left is a "FINISH SHOPPING" button. At the bottom right is a plus sign icon. Navigation links "Shopping List" and "Inventory" are at the very bottom.

Figure 8.2: "Shopping List" page

In the "Inventory" page - see Figure 8.3, there also appeared the same new button in the left top corner that takes the user to the new "Help" page. Based on the testing results, it was also decided to move down the items on the inventory list that have zero amount.



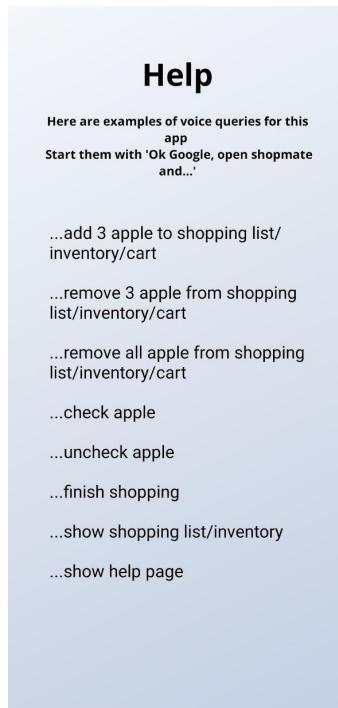
The screenshot shows a mobile application interface titled "Inventory". At the top left is a question mark icon, and at the top right is a plus sign icon. Below the title is a table listing items and their counts:

Apple	8
Water	3
Orange	1
Grape	6
Cheese	0
Ice Cream	0

At the bottom of the screen are two tabs: "Shopping List" and "Inventory", with "Inventory" being the active tab.

Figure 8.3: "Inventory" page

There is a new "Help" page - see Figure 8.4, that is created to help users to adapt to controlling the application by voice.



The screenshot shows a mobile application interface titled "Help". At the top left is a question mark icon, and at the top right is a plus sign icon. The page contains text and a list of voice queries:

Here are examples of voice queries for this app
Start them with 'Ok Google, open shopmate and...'

- ...add 3 apple to shopping list/inventory/cart
- ...remove 3 apple from shopping list/inventory/cart
- ...remove all apple from shopping list/inventory/cart
- ...check apple
- ...uncheck apple
- ...finish shopping
- ...show shopping list/inventory
- ...show help page

Figure 8.4: "Help" page

In both "Add item" pages for shopping list items - see Figure 8.5, and inventory items - see Figure 8.6, there were set placeholders for the input fields that prompt users what input and in which format is expected.

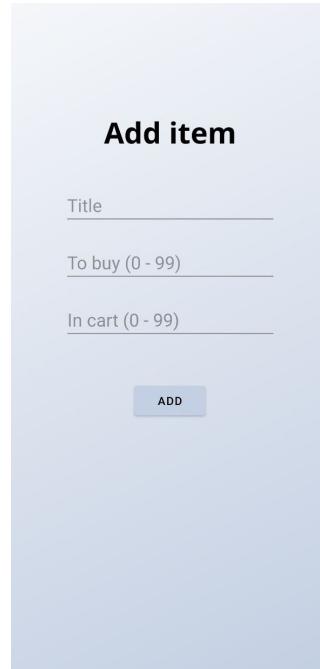


Figure 8.5: "Add Item" page for shopping list items

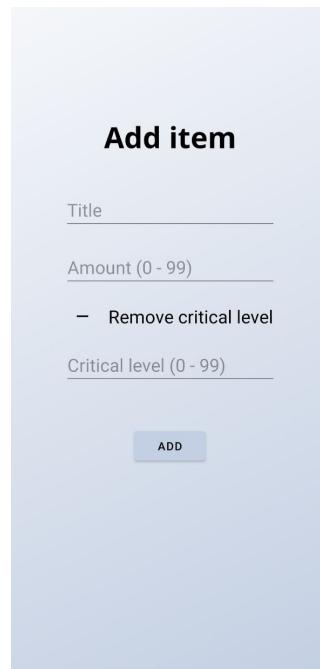


Figure 8.6: "Add Item" page for inventory items

8.2. Integration of the Google Assistant to the application

In the both "Edit item" pages for shopping list item - see Figure 8.7, and inventory item - see Figure 8.8, there are added "+" and "-" buttons, that serve for easy increasing/reducing of the amount in few points.

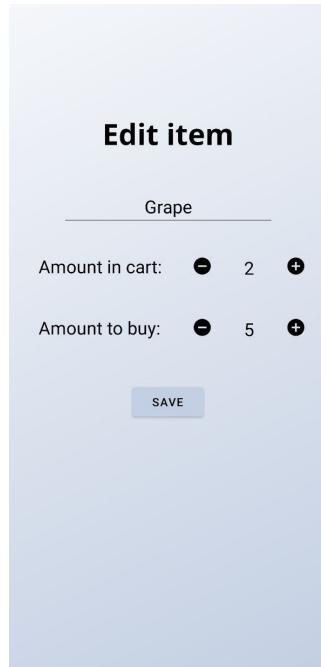


Figure 8.7: "Edit Item" page for shopping list items

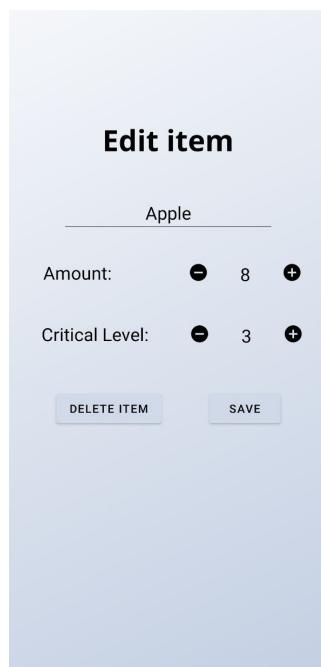


Figure 8.8: "Edit Item" page for inventory items

The icon 8.9 for the application was chosen from the open-source resource [72] and it also fits the concept of simplicity of the interface.



Figure 8.9: Icon of the "Shopmate" application

Executable APK of the "Shopmate" application and user manual can be found in the electronic attachments A.

■ 8.2.9 Problems with using the Google Assistant

During the development of the application, a certain problem with the Google Assistant became evident. On 12.04.2022 it appeared that the functionality that has already been tested was not working properly. The issue was that Google Assistant did not cooperate with the application in some instances, though it still worked in some other cases:

- when a defined query (that matches specific custom intent) was requested via Google Assistant, it was not recognized as an intent by Google Assistant and was not sent to the application. So, the application was not even launched. Instead, Google Assistant initiated a search on the Internet;
- when the same intent was tested via App Actions testing tool, everything worked correctly;
- when a query of some BII intent (not custom intent) was requested, everything functioned correctly as well.

Then it was tested on the previous git commits to make sure that there are no new changes that might impact the work of the application. The application did not function in the previous commits either.

The challenge in debugging in this case was that between sending the request to Google Assistant and getting the intent in the application there was a black box that was subservient to Google. So, the developer of the application cannot always understand the stage at which the error occurs.

After a detailed study of possible causes, it was decided to ask for assistance. Google Assistant provides contacts of Google Assistant Developer Experts [73]. One of them responded to my request. After the careful study of the code and trying different approaches that could help, he made a conclusion that everything seemed to be correct and it should have worked properly. There was an assumption that the problem could be caused by some changes on the Google side. I also tried to revert to previous versions of the Google application, but it was to no avail.

..... 8.2. *Integration of the Google Assistant to the application*

On 18.04.2022 it started to work again, but some peculiarities were evident. Some inconsistencies and limitations in the work of the application were noticed:

- Some of the queries did not work the same way as they used to.
- Sometimes requesting the same query gives the desired result only after 3-4 attempts. On the first try Google Assistant begins to browse the Internet.
- None of the queries work while the application is opened.

The first issue can be solved by slightly changing the queries. As for the other two issues, there has not been found a solution on my part yet.

Chapter 9

Testing of the "Shopmate" application

This chapter will describe all the testing that took place during the development of the "Shopmate" application. The testing included both developer testing and user testing.

9.1 Developer tests

Testing took place continuously during development. Tests have been carried out on the physical device Mi Note 10 with Android 10 and on the virtual device Pixel XL with Android 11.

Testing covered interface, the integration with the voice assistant, and back logic itself. There were processed edge cases and removed standard bugs that appear during any development.

9.2 Google Assistant tests

Validating the input from the keyboard is the standard case. For the validation of the input from the Google Assistant, it is needed to think about the issue more comprehensively. Moreover, some other non-standard situations can occur while using the integration of the Google Assistant and the "Shopmate" application. The common situations that are likely to happen were selected and tested.

The scenarios, found issues and provided solutions are described below:

1. **Scenario:** The user makes any voice query that contains the number.

Issue: The user can say or type into the Google Assistant something that will fail the parsing into the non-negative integer, for example:

- the negative number
- non-integer number
- not a valid number (e.g.: twenty thirty four)
- a word that is not the number
- any sign (e.g.: * /")

Solution: Having got the wrong number input, the application will open the page saying that the user's request was not clear and the user should fill the number field manually.

2. **Scenario:** The user makes any voice query that contains the title of the item.

Issue: The user can say or type into the Google Assistant something that will fail the validation, for example:

- digit
- any sign (e.g.: */")

Solution: Having got the wrong title input, the application will open the page saying that the user's request was not clear and the user should fill the title field manually.

3. **Scenario:** The user makes any voice query.

Issue: The Google Assistant can identify some words incorrectly.

Solution: All queries were tried 10+ times and all misidentifications were written down. These misidentifications are processed with the use of the inline inventory. As this is the most common case, there is a more detailed description in the "Similar-sounding words" section above.

4. **Scenario:** The user makes any voice query that changes the state of the application (e.g.: adding or removing items).

Issue: To distinguish if the voice query was perceived correctly by the Google Assistant and according to the user's expectations, the user must look through the whole list of items. This can take a lot of time and negate the benefits of voice control of the application.

Solution: Simple solution: The Google Assistant presents the feedback: "Opening shopmate", which means that the request is received by the Google Assistant and the request is identified as the request for the "Shopmate" application. More complex solution for the future: To give more detailed feedback, sound feedback using speech synthesis can be used. It will state precisely what the application has done. The user will be able to turn on/off this feature.

5. **Scenario:** The user makes any voice query.

Issue: The user can make a mistake and want the Google Assistant to stop processing this request and listen to the new one.

Solution: This is the issue on the Google side. The easiest way to stop Google from talking is to click the "keyboard" sign. Then say again "Ok, Google" or click the microphone sign.

9.3 Testing of the Google Assistant in the real environment

As the application will be used most likely in noisy environments, it is important to test the voice interaction in the real conditions.

For these purposes, 3 locations were chosen: supermarket, underground and tram. For each location two queries were chosen : “Open Shopmate and add two apples to the shopping list” and “Open Shopmate and check apples”. Each of these queries was repeated 5 times.

The results are as follows - see Table 9.1:

Table 9.1: Results of noise tests

Location	Right heard queries	Wrong heard queries
Supermarket	9	1
Underground	6	4
Tram	9	1

It can be concluded that due to noise and other people’s speech, sometimes the Google Assistant may not hear the whole query or may hear it with some mistakes. But all in all, the recognition works correctly in most cases, especially when it is not made in such noisy places as underground. It is also important to note that in the supermarket a lot of people were having their conversations around, but it did not interfere with the work of the Assistant, no extra words were perceived by the application in the process.

Moreover, making queries via earphones was tested (used earphones are Airpods Pro). The interaction with the Google Assistant showed the same results for testing with earphones and without them.

9.4 User testing

After the end of the main stage of the development, user testing took place. It is important to engage real potential users in the testing of the application for a few reasons:

- To look for design problems - is it clear how to use the application?
- To look for logical problems - is the application functioning according to the user’s expectations?
- To receive potential “add feature” recommendations from the user - what should/must be added to the application?
- To refer to user satisfaction - will the user use the application?

Furthermore, it is important to analyze the target audience to involve these exact people into testing. As the “Shopmate” application is conceived

for everyday usage of ordinary people, people from different age groups, with different levels of English and with different experience with modern technologies were engaged in the application testing.

9.4.1 Process of testing

1. Users have a special set of tasks prepared for them.
2. Users complete these tasks under the inspection of the observer.
3. Users are asked to vocalize their thoughts, actions, ideas, and doubts.
4. The observer records audio and takes notes during the testing.
5. The observer analyzes notes and audio records.
6. The observer makes conclusions from the testing.
7. The observer creates the list of changes that can be made to the application.

9.4.2 Testing scenario

The scenario consists of tasks and is created in such a way that all functional requirements are tested.

Tasks:

1. You want to buy 3 apples. Use the interface to state this.
2. You want to buy 7 oranges. Use the voice assistant to state this.
3. You are in the market and you want to see what you have on your shopping list. Open the shopping list using the voice assistant.
4. You are in the market and you have put 3 apples into your shopping cart. Use the interface to state that.
5. You are in the market and you have put 2 oranges into your shopping cart. Use the voice assistant to state that.
6. You came home and you want to add all the purchased items to the inventory. Do this using the interface.
7. If there are some items that you don't already need on the shopping list, delete them using the interface.
8. You don't need to buy so many oranges. Remove 1 from the shopping list using the interface.
9. You don't need to buy so many oranges. Remove 2 from the shopping list using the voice assistant.

10. You have already had 2 bananas at home. Use the interface to state this.
11. You have already had 6 oranges at home. Use the voice assistant to state this.
12. Set the minimal amount of each item that you always want to have at home using the interface.
13. Set the critical level of the banana to 5 using the interface.
14. See the shopping list to find out if you need to buy any more items.
15. You are in the market, state that you have bought all the items from the shopping list.
16. You are in the market and you see the pears on sale. You put 10 into the cart. State that using the interface.
17. You decided that you don't need so many pears and you removed 7 from the cart. State that using the interface.
18. You are in the market and you see the lemons on sale. You put 4 into the cart. State that using the voice assistant.
19. You decided that you need 1 more lemon and you put 1 more into the cart. State that using the voice assistant.
20. You came home and you want to add all the purchased items to the inventory. Do this using the voice assistant.
21. If there are some items that you don't need anymore on the shopping list, delete them using the interface.
22. You don't want oranges to have a critical level. State that using the interface.
23. You don't want apples to be recorded in the inventory anymore. State that using the interface.

9.4.3 Tester's results

Testing group consists of 4 people of age 16 - 34, of English proficiency medium to high and of experience with the modern technologies medium to high with different devices.

Tester A

For information about the tester A see Table 9.2

Table 9.2: Information about the tester A

Age	Device	Experience with the modern technologies	English proficiency
16	Xiaomi Mi8 with Android 10	Medium	High

The result of testing by the tester A can be seen in Table 9.3

Table 9.3: Results of testing by the tester A

Task	Event log
1. Task	Tries to input the title not in English. Keeps the field "cart" empty. Does not know what the "cart" field means. Sees "Inventory" in the main menu, does not know what "Inventory" means.
2. Task	Makes the wrong voice request. Makes another wrong request. Asks for help. The observer asks to look into the help page. Makes the right request, but the assistant gets 27 instead of 7. Changes the mistake via the interface.
3. Task	Looks for the right query on the help page.
4. Task	Totally deletes the apple item from the shopping list, instead of adding 3 apples to the cart. The observer explains the meaning of the cart. Adds 3 apples back to the shopping list. Adds 3 apples to the cart and removes 3 apples from the shopping cart via the edit page. The observer explains the meaning of the cart in more detail.
5. Task	Makes a wrong voice request and tries to say "Stop Google", it does not work. Makes the right voice request.
6. Task	On the first try clicks the right button.
7. Task	On the first try clicks the right buttons.
13. Task	Suggests changes: when the item in the inventory has 0 amount it might be removed from the list, but its critical level might be saved for the future.
15. Task	Looks for the button to make it in one click. Clicks the add button and the edit button - searches for the one click functionality. Does not find it. Fills it manually.
17. Task	Starts to decrease the amount one by one with the minus button. After 2 clicks just types the number with the keyboard.
18. Task	Makes the query with "shop cart" instead of "cart". Makes the query with "shopping cart" instead of "cart". Makes right query, the Google Assistant recognizes "ad" instead of "add". Gets annoyed because Google Assistant does not respond to "Okay, Google" 3 times.
19. Task	Wants to try to delete the lemon, looks for the button on the edit page.
20. Task	Is confused that the button is called "cart -> inventory", but the voice query is "finish shopping". Recommendation: should be called the same.
21. Task	Recommendation: add a feature that allows to delete all items from the shopping list in one click.

23. Task	Makes right query, the Google Assistant recognizes "in Ventura" instead of "inventory".
User satisfaction	The user finds the app useful and would use it in real life. After testing and some explanations from the observer the application seems to be intuitive in use.

For potential improvements for the application from testing by the tester A see Table 9.4

Table 9.4: Potential improvements from testing by the tester A

Suggestion	Release state of the solution
Add manual on the first start of the application to explain what features are present and to show where to look for help.	Manual is created as the external file. Feature to be implemented in the future: manual should be shown in the application on its first start.
Change the title "Inventory" to a more intuitive title.	Rejected. This word reflects the meaning of the "Inventory" part. The manual will help with the better understanding.
Find the way to stop the request, when the user begins to say the wrong request.	Feature to be implemented in the future.
When the item in the inventory has amount 0, it might be removed from the list, but its critical level might be saved for the future.	Rejected. Another feature is implemented instead: items in the inventory that have amount 0 are moved to the bottom part of the inventory list.
Add a button to indicate that everything from the shopping list is already in cart in one click (set all amountInCart on amountToBuy).	Rejected. This feature does not have an application in real life. Another feature is to be implemented in the feature instead: On the edit page of each item, there should be a button that will set the amount in cart based on the amount to buy.
Add synonyms for "cart" - "shop cart", "shopping cart".	Implemented.
Add a similar sounding word for "add" - "ad".	Implemented.
Change the place of the delete button in the shopping list page.	Rejected. It was intuitive for other users.
Set the same name for the button "cart -> inventory" and voice query "finish shopping".	Implemented. Both are set to "finish shopping".
Add a button to delete all items from the shopping list in one click.	Feature to be implemented in the future.
Add a similar sounding word for "inventory" - "in Ventura".	Implemented.

■ Tester B

For information about the tester B see Table 9.5

Table 9.5: Information about the tester B

Age	Device	Experience with the modern technologies	English proficiency
22	Oneplus Nord 2 5G with Android 11	High	Medium

The result of the testing by the tester B can be seen in Table 9.6

Table 9.6: Results of testing by the tester B

Task	Event log
2. Task	Makes the wrong voice request. The observer asks to look into the help page. Gives the right request to the Google Assistant 3 times before it is processed correctly.
4. Task	Deletes 3 apples from the shopping list and adds 3 apples to the cart.
5. Task	Says "into" instead of "to" in the request. Likes the appeared page "the request was not clear, fill in the data manually".
6. Task	On the first try clicks the right button.
7. Task	Tries clicking on the item. Tries clicking on the checkbox. Then finds the right button.
11. Task	Makes right query, the Google Assistant recognizes "at" instead of "add".
14. Task	Suggests changes: separate the amount that is added from the inventory, and the amount that is added directly to the shopping list.
15. Task	Looks for the button to make it in one click. Clicks the edit button, opens the help page, crosses out everything, clicks on the "cart" icon - searches for the one click functionality. Suggests a feature: on the edit page of each item there will be a button that will set the amount in the cart based on the amount to buy.
17. Task	Reduces the amount one by one with the minus button. Suggests UI changes: underline the input.
18. Task	Makes the wrong request. Suggests improvement: the Google Assistant may answer something like "There is nothing found in this application" instead of searching this phrase on the Internet.
19. Task	Makes right query, the Google Assistant recognizes "card" instead of "cart".

User satisfaction	The user finds the app useful and is willing to start using it from today
-------------------	---

For potential improvements for the application from testing by the tester B see Table 9.7

Table 9.7: Potential improvements from testing by the tester B

Suggestion	Release state of the solution
Add manual on the first start of the application to explain what features are present and to show where to look for the help.	Manual is created as the external file. Feature to be implemented in the future: manual should be shown in the application on its first start.
Add synonyms for “to” - “into”.	Implemented.
Add a similar-sounding word for “add” - “at”.	Implemented.
Visually separate the amount that is added from the inventory, and the amount that is added directly to the shopping list in the “Shopping List” page.	Rejected. This is a confusing feature that will overload the interface.
On the edit page of each item, there may be a button that will set the amount in cart based on the amount to buy.	Feature to be implemented in the future.
On the edit page of each item underline the numerical input. So, it will be clear that it is an editable field.	Implemented.
Add a similar-sounding word for “card” - “cart”.	Implemented.

Tester C

For information about the tester C see Table 9.8

Table 9.8: Information about the tester C

Age	Device	Experience with the modern technologies	English proficiency
18	Meizu M8 with Android 8	Medium	Medium

The result of the testing by the tester C can be seen in Table 9.9

Table 9.9: Potential improvements from testing by the tester C

Task	Event log
1. Task	Does not know what the “cart” field means.

2. Task	Does not know if he needs to reduce the amount in the shopping list based on the amount that is added to the cart.
5. Task	Makes right query, the Google Assistant recognizes "card" or "court" instead of "cart".
8. Task	Clicks edit button. Clicks the check-box. Holds down the item. Only after that does the right action - clicks on the item
12. Task	Does not know where to look for the critical level. For 10 seconds does not try anything. Tries clicking the "plus" button. Clicks the item as it should be.
17. Task	Suggests changes: instead of "plus" and "minus" buttons that allow to change the amount of the item, there might be a slider.
23. Task	Sets the amount of the item on 0 instead of completely deleting the item from the inventory.
User satisfaction	The user says that the application is OK, but some improvements may be added.

For potential improvements for the application from testing by the tester C see Table 9.10

Table 9.10: Potential improvements from testing by the tester C

Suggestion	Release state of the solution
Add manual on the first start of the application to explain what features are present and to show where to look for the help.	Manual is created as the external file. Feature to be implemented in the future: manual should be shown in the application on its first start.
Add similar-sounding words for "cart" - "court" and "card".	Implemented.
Create a slider that allows you to change the amount of the item instead of "plus" and "minus" buttons.	Rejected. Decided that it would not be convenient.

■ Tester D

For information about the tester D see Table 9.11

Table 9.11: Information about the tester D

Age	Device	Experience with the modern technologies	English proficiency
34	Mi Note 10 with Android 10	Medium	Low

The result of testing by the tester D can be seen in Table 9.12

Table 9.12: Results of testing by the tester D

Task	Event log
1. Task	Does not know what the "cart" field means. Sees "Cart" icon on the "Shopping list" page and understands the meaning of the cart.
2. Task	Makes the wrong voice request. Asks for help. The observer asks to look into the help page. Forgets what to say. Looks into the help page again. Makes the right request, but the assistant gets "and" instead of "add".
3. Task	Tries just "Open Shopmate". Then looks at the help page.
5. Task	Assistant recognizes "and" instead of "add". During the query she makes a pause until the assistant writes down all that she has told and then continues the phrase. Assistant takes it as the finish of the request and stops listening to her.
6. Task	Looks for the button. Goes to the inventory page. Clicks plus there. Returns to the shopping list page. Clicks the right button.
7. Task	Clicks the item. Does not find the needed button. Clicks the item again. Does not hit the buttons due to the long nails.
9. Task	Assistant recognizes "please" instead of "list".
10. Task	Tries to input the title not in English.
11. Task	Comes up with the query by herself, does not look into the help page.
12. Task	Clicks add button. Then clicks the right button. Is satisfied that items that are below the critical level are automatically added to the shopping list.
17. Task	Decreases the amount one by one with the minus button.
20. Task	Assistant recognizes "fees" instead of "finish".
User satisfaction	The user is satisfied with the application, but it seems that it is not easy for her to use the assistant not in her native language.

For potential improvements for the application from testing by the tester D see Table 9.13

Table 9.13: Potential improvements from testing by the tester D

Suggestion	Release state of the solution
Add manual on the first start of the application to explain what features are present and to show where to look for the help.	Manual is created as the external file. Feature to be implemented in the future: manual should be shown in the application on its first start.
Add a similar sounding word for "add" - "and".	Implemented.
Add the "Open Shopmate" query, that opens the application.	Feature to be implemented in the future.
Add a similar sounding word for "list" - "please".	Implemented.
Add a similar sounding word for "finish" - "fees".	Implemented.

9.5 Testing results

From the testing it can be concluded that the "Shopmate" application seems to be useful for users. Moreover, the application underwent some "real environment" tests, it shows that the application is adapted for use in real life. Finally, based on the developer testing, Google Assistant testing and the user testing, there can be listed features to improve or add into the "Shopmate" application. They are described in the next chapter.

Chapter 10

What can be improved in the future in the "Shopmate" application

10.1 Improvements in the communication through the Google Assistant

As the voice control is the core for comfortable usage of the "Shopmate" application, it is important to work on the improvement of this communication.

Below there is a description of the ideas of the improvements or new features based on the whole process of the development: implementation, developer and user testing. Each improvement/feature has the subjective priority (1 - the highest priority, 5 - the lowest priority), which is evaluated in terms of user-friendliness of the "Shopmate" application.

Future improvements and features for the "Shopmate" application can be seen in Table 10.1

Table 10.1: Future improvements and features in the communication with the "Shopmate" application

Improvement/feature	Priority
The voice queries must work consistently and be processed correctly by the Google Assistant from the first request that the user creates.	1
There should be the option to add/remove a few items in one voice request.	2
More specific feedback should be provided to the user when the voice request is made. It may be sound feedback with usage of speech synthesis or some visual effect. It is important for the feedback to be provided to the user concerning what precisely has been done by the application as the reaction to the voice query.	2
An option of stopping the Google Assistant from processing the wrong request by voice should be provided to the users.	2
The defined voice queries can be shortened. It will be comfortable if it is not needed to say "open Shopmate " at the beginning of each query, but only for the first time.	3

Based on the pull of users' requests, it may be beneficial to widen the list of similar-sounding words and synonyms which might be grouped together in inline inventory.	3
Other languages can be supported.	3
Having got the wrong voice queries that are addressed to the "Shopmate" application, the Google Assistant might return a response which informs the user that the request was wrong instead of searching this phrase on the Internet. The response may be as follows: "Nothing is found in this application".	4

■ 10.2 Other improvements and new features

The "Shopmate" application has all the basic expected features. However, there are still quite a lot of things that can be improved or added to the application. This list is based on the analyzes of functionalities of the similar applications, the user and developer testing, and my own ideas. Each improvement/feature has the subjective priority (1 - the highest priority, 5 - the lowest priority), which is evaluated in terms of user-friendliness of the "Shopmate" application.

Future improvements and features for the "Shopmate" application can be seen 10.2

Table 10.2: Future improvements and features for the "Shopmate" application

Improvement/feature	Priority
Plural and singular forms of the same word must be perceived by the application as the same word and be written down under the same item.	1
Add an interactive manual directly to the application.	1
The measures can be added, so that the amount of the item can be presented not only in pieces, but also in grams, liters e.t.c.	2
There can be added accounts, so that there is the possibility of having access to the inventory and the shopping list from different devices and sharing them with other people.	3
On the edit page of each item, there should be a button that will in one-click set the amount in cart based on the amount to buy.	3
Add the ability to delete all items from the shopping list in one click.	3
Add the "Open Shopmate" query, that opens the application.	3
Possibility to create several shopping lists and inventories with custom names.	3
Possibility to add items via barcodes, via voice input, from user's own database.	4
Possibility to add discount cards and coupons.	5

Chapter 11

Conclusion

The main goal of this bachelor thesis was to analyze, design, implement and test the "Shopmate" application that is supporting shopping and the maintenance of the inventory, and which is being controlled by both interface and voice. This goal has been achieved by some steps which are summarized further.

In this bachelor thesis such principles as the management of the shopping list and the inventory, the usage of the voice assistant in the applications were discussed. Having explored the market of existing applications, it may be concluded that there are no applications that implement all these concepts simultaneously. Therefore, the development of the "Shopmate" application is reasonable.

Furthermore, there were compared existing voice assistants and the App Actions from the Google Assistant which were later chosen to be integrated into the "Shopmate" application.

The analysis and the design of this application was the following step.

Based on the analysis and the design the implementation part followed further. In this part there was a description of the integration with the Google Assistant along with the difficulties that appeared in the process. After that the configuration files and the interface were described. Moreover, the summary of the whole process of processing the user's voice request was introduced in this part.

Later there was the testing part where both developer and user tests were described. There were also some special tests regarding the communication with the application via Google Assistant.

In the last part there were described things to be improved in the future versions of the application.

To sum up, it can be stated that the main goal of this bachelor thesis was fulfilled. During the work it showed that even though the concept of the application seemed to be simple at first sight, the whole process of the development was extensive and complex and there appeared some issues in the process.

As it turned out during development and testing, the integration with the Google Assistant may be not very consistent and constant as there can appear problems on the Google side, that can not be affected from our side. The

issue is that the Google Assistant does not react correctly to the voice query the first 2-3 tries, it makes the usage of the application not convenient. Voice control is the main benefit of the "Shopmate" application, so for the future development of this application it is necessary for this problem to be solved. If the Google Assistant does not work reliably with the "Shopmate" application, it could mean that it is pointless to continue on the development. In this case, the accuracy of the decisions to make the first version of the application with a simple interface and only core functionalities will be confirmed. On the other hand, the problem may be solved in the newer version of the Google Assistant or App Actions. Furthermore, entering the partnership with Google may give more benefits, more stable integration of the Google Assistant with the "Shopmate" application and more technical support from the Google side. If the problem is solved, the development should continue in the direction of working on the points described in the "What can be improved to the future" section.

As it became visible from the user testing of the "Shopmate", the application seems to be useful for potential users. Even though the "Shopmate" application on this stage faces some challenges and this is the first simple version of the application, I believe that it has big potential. As the world's level of modernization is growing at a fast tempo, people want to simplify their lives as much as possible with the use of smart devices. The popularity of the total hands-free is also growing. Therefore, the "Shopmate" application could be a very desirable and useful application in the market. So, under the condition that the integration of the Google Assistant to the "Shopmate" application becomes reliable, I plan to continue working on this application, to improve existing features and to add new ones as described above.

To conclude, in my opinion, this work is beneficial for people who want to get acquainted with the principles of the integration of the Google Assistant into the mobile application. It serves as the basic instruction for this integration, describes different options of the integration and introduces some possible risks and issues. For me, this work was beneficial as well, because I mastered the integration of the Google Assistant into the Android application from scratch, and I also deepened my knowledge and skills in programming of the mobile applications.

Resources

1. *Smartphone Usage Statistics* [online] [visited on 2022-02-20]. Available from: <https://leftronic.com/blog/smartphone-usage-statistics/>.
2. *Speech Recognition* [online] [visited on 2021-11-14]. Available from: <https://www.ibm.com/cloud/learn/speech-recognition#toc-speech-rec-GaPrtSxU>.
3. *What is a voice assistant?* [Online] [visited on 2021-11-14]. Available from: <https://alan.app/blog/voiceassistant-2/>.
4. *Voice Assistants* [online] [visited on 2021-11-14]. Available from: <https://www.slanglabs.in/voice-assistants>.
5. *What is Speech Recognition Software?* [Online] [visited on 2021-11-14]. Available from: <https://www.callrail.com/blog/speech-recognition-software/>.
6. *Voice Assistants* [online] [visited on 2021-11-14]. Available from: https://uploads-ssl.webflow.com/5d78905f61170c461a702467/5f4a233b6406281084957991_VA%20blog-03-p-1080.png.
7. *Natural Language Processing* [online] [visited on 2021-11-14]. Available from: <https://www.ibm.com/cloud/learn/natural-language-processing>.
8. *Inventory, shopping list applications, Google Play* [online] [visited on 2021-12-10]. Available from: <https://play.google.com/store/search?q=inventory%20shopping%20list&c=apps>.
9. *Food Checklist - Groceries Expiration and Shopping, Google Play* [online] [visited on 2021-12-10]. Available from: <https://play.google.com/store/apps/details?id=com.chestersw.foodlist>.
10. *Grocery Shopping List Listonic, Google Play* [online] [visited on 2021-12-10]. Available from: <https://play.google.com/store/apps/details?id=com.l>.
11. *AnyList: Grocery Shopping List Recipe Organizer, Google Play* [online] [visited on 2021-12-10]. Available from: <https://play.google.com/store/apps/details?id=com.purplecover.anylist>.

11. Conclusion

12. *Bring! Grocery Shopping List*, Google Play [online] [visited on 2021-12-10]. Available from: <https://play.google.com/store/apps/details?id=ch.publisheria.bring>.
13. *Our Groceries Shopping List*, Google Play [online] [visited on 2021-12-10]. Available from: <https://play.google.com/store/apps/details?id=com.headcode.ourgroceries>.
14. *Grocery Tracker Shopping List*, Google Play [online] [visited on 2021-12-10]. Available from: <https://play.google.com/store/apps/details?id=easicorp.gtracker>.
15. *Shopping List*, Google Play [online] [visited on 2021-12-10]. Available from: <https://play.google.com/store/apps/details?id=ru.grocerylist.android>.
16. *Shopping list - My List*, Google Play [online] [visited on 2021-12-10]. Available from: <https://play.google.com/store/apps/details?id=com.simplesmartsoft.mylist>.
17. *Shopping List*, Google Play [online] [visited on 2021-12-10]. Available from: <https://play.google.com/store/apps/details?id=com.technix.shoppinglist>.
18. *Super Simple Shopping List*, Google Play [online] [visited on 2021-12-10]. Available from: <https://play.google.com/store/apps/details?id=com.bitwize10.supersimpleshoppinglist>.
19. *IBM Shoebox* [online] [visited on 2021-11-26]. Available from: https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html.
20. *Siri* [online] [visited on 2021-11-26]. Available from: <https://www.sri.com/hoi/siri/>.
21. *Share of voice assistants users in the United States in 2021, by product and age group* [online] [visited on 2021-11-26]. Available from: <https://www.statista.com/statistics/1274429/voice-assistants-use-by-age-group-united-states/>.
22. *Set up Google Assistant on your device* [online] [visited on 2021-11-26]. Available from: https://support.google.com/assistant/answer/7172657?hl=en&ref_topic=7658198#zippy=.
23. KLUSÁK, Bc. Michal. *Nákup produktů pomocí Google Assistant*. 2020. MA thesis. CTU.
24. *Shop Alexa Built-in Phones* [online] [visited on 2021-11-26]. Available from: https://www.amazon.com/b/ref=ampd_use_topnav_desktop/ref=s9_acss_bw_cg_usetopna_md1_w?node=14613304011&pf_rd_m=ATVPDKIKX0DER&pf_rd_s=merchandised-search-top-1&pf_rd_r=5WVS6BBR8P3HV2FBKZ4M&pf_rd_t=101&pf_rd_p=491b7e28-c0f7-42fd-b921-5f079576630c&pf_rd_i=18665791011.

25. *Google just revealed that half a billion people around the world are using the Google Assistant as it battles with Amazon to conquer the smart home* [online] [visited on 2021-11-26]. Available from: <https://www.businessinsider.com/google-assistant-500-million-users-challenges-amazon-alexa-2020-1>.
26. *Change the language of Google Assistant* [online] [visited on 2021-11-26]. Available from: <https://support.google.com/googlenest/answer/7550584?co=GENIE.Platform%3DAndroid&hl=en#zippy=%2Cgoogle-home>.
27. *Develop Skills in Multiple Languages* [online] [visited on 2021-11-26]. Available from: <https://developer.amazon.com/en-US/docs/alexa/custom-skills/develop-skills-in-multiple-languages.html>.
28. *Integrate with Google Assistant* [online] [visited on 2021-12-01]. Available from: <https://developers.google.com/assistant>.
29. *App Actions overview* [online] [visited on 2021-03-27]. Available from: <https://developers.google.com/assistant/app/overview>.
30. *Conversational Actions* [online] [visited on 2021-12-01]. Available from: <https://developers.google.com/assistant/conversational/overview>.
31. *App Actions Codelabs* [online] [visited on 2021-12-01]. Available from: <https://developers.google.com/assistant/app/codelabs>.
32. *Conversational Actions Codelabs* [online] [visited on 2021-12-01]. Available from: <https://developers.google.com/assistant/conversational/codelabs>.
33. *Compare app actions with conversational actions* [online] [visited on 2021-12-02]. Available from: <https://stackoverflow.com/questions/67954451/compare-app-actions-with-conversational-actions>.
34. *Spotify: Music and Podcasts, Google Play* [online] [visited on 2021-12-02]. Available from: <https://play.google.com/store/apps/details?id=com.spotify.music>.
35. *Money Manager Expense Budget, Google Play* [online] [visited on 2021-12-02]. Available from: <https://play.google.com/store/apps/details?id=com.realbyteapps.moneymanagerfree>.
36. *Build App Actions* [online] [visited on 2021-12-13]. Available from: <https://developers.google.com/assistant/app/get-started>.
37. *App Actions built-in intents* [online] [visited on 2021-12-13]. Available from: <https://developers.google.com/assistant/app/reference/built-in-intents>.
38. *Built-in intents index* [online] [visited on 2021-12-13]. Available from: <https://developers.google.com/assistant/app/reference/built-in-intents/bii-index>.

39. *Order menu item BII* [online] [visited on 2021-12-13]. Available from: <https://developers.google.com/assistant/app/reference/built-in-intents/food-and-drink/order-menu-item>.
40. *Functional requirements VS non-functional* [online] [visited on 2022-05-04]. Available from: <https://enkonix.com/blog/functional-requirements-vs-non-functional/>.
41. *Class diagrams* [online] [visited on 2022-05-04]. Available from: <https://www.ibm.com/docs/en/rsm/7.5.0?topic=structure-class-diagrams>.
42. *TinyDB – Android-Shared-Preferences-Turbo* [online] [visited on 2022-03-19]. Available from: <https://github.com/kcochibili/TinyDB--Android-Shared-Preferences-Turbo>.
43. *UML diagramy případů užití* [online] [visited on 2022-05-04]. Available from: https://moodle.fel.cvut.cz/pluginfile.php/233850/mod_resource/content/5/P%C5%99edn%C3%A1%C5%A1ka4a5_PripadyUzitiKomplet.pdf.
44. LARMAN, Craig. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition*. Pearson, 2004.
45. *What is wireframing* [online] [visited on 2022-05-05]. Available from: <https://www.experienceux.co.uk/faqs/what-is-wireframing/>.
46. *UML diagramy případů užití* [online] [visited on 2022-05-04]. Available from: https://moodle.fel.cvut.cz/pluginfile.php/233856/mod_resource/content/1/UML%20-%20Diagram%20nasazeni.pdf.
47. *Documentation for app developers* [online] [visited on 2022-10-10]. Available from: <https://developer.android.com/docs>.
48. *Top android IDEs for developers* [online] [visited on 2022-05-01]. Available from: <https://www.developer.com/mobile/top-android-ides-for-developers/>.
49. *Android Studio* [online] [visited on 2022-05-01]. Available from: <https://developer.android.com/studio>.
50. *Eclipse* [online] [visited on 2022-05-01]. Available from: <https://www.eclipse.org/ide/>.
51. *Visual Studio* [online] [visited on 2022-05-01]. Available from: <https://visualstudio.microsoft.com/>.
52. *Gitlab* [online] [visited on 2022-05-01]. Available from: <https://gitlab.com/>.
53. *Google Assistant plugin for Android Studio* [online] [visited on 2022-05-01]. Available from: <https://developers.google.com/assistant/app/test-tool>.

54. *App Actions Test Tool* [online] [visited on 2022-05-01]. Available from: <https://plugins.jetbrains.com/plugin/12322-app-actions-test-tool>.
55. *Java* [online] [visited on 2022-05-01]. Available from: <https://www.java.com/en/>.
56. *Popularity of Programming Language* [online] [visited on 2022-05-01]. Available from: <https://pypl.github.io/PYPL.html>.
57. *Maven* [online] [visited on 2022-05-01]. Available from: <https://maven.apache.org/>.
58. *Gradle* [online] [visited on 2022-05-01]. Available from: <https://gradle.org/>.
59. *Configure your build* [online] [visited on 2022-05-01]. Available from: <https://developer.android.com/studio/build>.
60. *Converting words to numbers in Java, answer by meghana anahgem* [online] [visited on 2022-05-01]. Available from: <https://stackoverflow.com/a/66462690>.
61. *Built-in intents for App Actions* [online] [visited on 2021-12-01]. Available from: <https://developers.google.com/assistant/app/intents>.
62. *Create thing BII* [online] [visited on 2021-12-16]. Available from: <https://developers.google.com/assistant/app/reference/built-in-intents/common/create-thing>.
63. *Custom intents* [online] [visited on 2022-03-27]. Available from: <https://developers.google.com/assistant/app/custom-intents>.
64. *Custom intents - Limitations* [online] [visited on 2022-04-12]. Available from: <https://developers.google.com/assistant/app/custom-intents#limitations>.
65. *Inline Inventory* [online] [visited on 2022-03-27]. Available from: <https://developers.google.com/assistant/app/inline-inventory>.
66. *The Gradle settings file* [online] [visited on 2022-05-01]. Available from: <https://developer.android.com/studio/build#groovy>.
67. *The module-level build file* [online] [visited on 2022-05-01]. Available from: <https://developer.android.com/studio/build#module-level>.
68. *SDK Platform release notes* [online] [visited on 2022-05-01]. Available from: <https://developer.android.com/studio/releases/platforms>.
69. *Android API Levels* [online] [visited on 2022-05-01]. Available from: <https://apilevels.com/>.
70. *Shared Preferences* [online] [visited on 2022-03-18]. Available from: <https://developer.android.com/reference/android/content/SharedPreferences>.

11. Conclusion

71. *The Activity Lifecycle*. Available also from: <https://developer.android.com/guide/components/activities/activity-lifecycle>.
72. *Vector Illustration* [online] [visited on 2022-04-28]. Available from: <https://icons8.com/illustrations/illustration/inky-black-shopping-cart>.
73. *Google Assistant Developer Experts* [online] [visited on 2022-04-16]. Available from: <https://developers.google.com/assistant/community/gde>.

Appendix A

Electronic attachments

In the electronic attachments there are the following files:

- Full use case diagram: `Use Case Diagram.png`
- Scenarios of all use cases: `Use Cases.pdf`
- User manual for the "Shopmate" application: `Manual.pdf`
- Video review of the "Shopmate" application: `Shopmate.mp4`
- Source code of the "Shopmate" application: `BachelorApp.zip`
- Executable APK of the "Shopmate" application: `app-release.apk`
- Full text of this bachelor work: `Bachelor Thesis Alina Samatova.pdf`