

**DEVELOPING A MOBILE APPLICATION THAT ENABLES
CONSUMERS TO REDUCE THEIR GROCERY EXPENDITURES IN PORTUGAL:
THE BUSINESS CASE OF SUPERPOUPA**

P. van Es – Developing a User Centric Mobile Application

By

Pepijn Van Es – 49522
Nova School of Business and Economics

A thesis submitted for a master's degree in
Impact Entrepreneurship and Innovation

Advised by H.M. Aguiar

January 27th, 2023

1.1.1 Abstract

This report investigates the potential of using a newly developed mobile app, SuperPoupa, to assist Portuguese consumers in mitigating the financial burden posed by increasing food prices caused by global events. Surveys have determined a willingness to adopt SuperPoupa into consumers' grocery habits, showing signs for an entrepreneurial opportunity. Core to the competitive edge of SuperPoupa is the self-build price-optimization algorithm to find the cheapest grocery basket. The report demonstrates the technical architecture and development process of the app's front- and back-end. Further deployment, research and development is necessary to evaluate the viability of SuperPoupa on larger scales.

Keywords:

Minimum Viable Product, Low-Code Mobile App Development, Back-End & Algorithm Development, Smart Grocery List App,

TABLE OF CONTENTS

1.1.1	Abstract	2
1.1.2	List of Figures	5
1.1.3	List of Tables	6
1.1.4	List of Abbreviations	7
1.1.5	Access to SuperPoupa.....	8
	<i>Executive Summary (Group Part)</i>	<i>9</i>
2	<i>Introduction (Group Part).....</i>	<i>11</i>
2.1	Contextual Background	11
2.2	Problem definition	12
2.3	Objectives and Structure of this Report.....	12
3	<i>Analysis and Diagnosis (Group Part)</i>	<i>13</i>
3.1	Literature Study	13
3.1.1	Economic Sentiment Influenced by COVID-19 and the War in Ukraine	13
3.1.2	The Effect of Inflation on the Food Industry	15
3.1.3	The Effect of Food Inflation on Middle-Class Households.....	16
3.1.4	The Effect of Food Inflation on Low-Income Households	18
3.2	Empirical Research	19
3.2.1	Research Instrument	20
3.2.2	The Perception of Inflation and Its Effect on the Reduction of Real Net Income	20
3.2.3	Users' Willingness to Adapt New Technologies for Reduced Grocery Spending.....	21
3.2.4	Assessing User's desired Features and their Willingness-To-Pay for it	21
4	<i>Solution Proposal</i>	<i>22</i>
4.1	Opportunity Identification (Individual Part: Pepijn van Es).....	22
4.2	Solution Introduction (Individual Part: Stefanie Neumann)	24
4.2.1	Business Model.....	24
4.2.2	Challenges of the Business Environment	25
4.2.3	Business Model Canvas	26
4.2.4	Competitive Analysis (Individual Part: Stefanie Neumann)	35
5	<i>Solution Development (Group Part)</i>	<i>38</i>
5.1	Back-end Development.....	39
5.1.1	Building a modular back-end using Python (Individual Part: Antonius Struk)	39
5.1.2	The Architecture Model of SuperPoupa.....	42
5.1.3	From Architecture to Algorithm and Data Model	44
5.1.4	The Data Model.....	46
5.1.5	The “cheapest path” Algorithm Development	50
5.1.6	Flask REST API (Antonius Struk).....	59

5.1.7	External Database – Implementing Azure PostgreSQL (Group Part)	61
5.1.8	Data Acquisition of Portuguese Supermarket Data (Antonius Struk)	62
5.1.9	Limitations and Missing Technology (Group Part)	65
5.2	Front-End Development (Group Part)	71
5.2.1	Technical Landscape for Building Mobile Apps.....	71
5.2.2	User Experience (UX) Design (Pepijn Van Es)	76
5.2.3	Main Feature Development of SuperPoupa (Adriano Villa Bascón).....	77
5.2.4	User-Feedback (Pepijn Van Es)	90
6	<i>Evaluation of SuperPoupa's Impact Potential (Group Part)</i>	<i>94</i>
7	<i>Conclusion (Group Contribution)</i>	<i>96</i>
8	<i>References</i>	<i>98</i>
9	<i>Appendices</i>	<i>112</i>
9.1	Customer Survey	112
9.2	Customer Survey Results English	114
9.3	Customer Survey Results Portuguese	120
9.4	Competitive Analysis	126
9.5	Customer Feedback Survey	127
9.6	User Feedback Results	128
9.7	Initial load_data() function loading CSV files	129
9.8	Commented Code API & Algorithm in Python	129
9.9	Process Flow Between Block 1, Flask API, and Block 2	9
9.10	Pricing Strategies of Different Applications	10

1.1.2 List of Figures

FIGURE 1 - PERCENTAGE OF PROGRAMMING LANGUAGE COMMUNITY QUESTIONS ON STACK OVERFLOW....	ERROR! BOOKMARK NOT DEFINED.
FIGURE 2 - HIGH LEVEL SUPERPOUPA SYSTEM ARCHITECTURE	44
FIGURE 3 - SUPERPOUPA'S ENTITY RELATIONSHIP MODEL AFTER CHEN 1976	48
FIGURE 4 - ENTITY RELATIONSHIP DIAGRAM	49
FIGURE 5. SETS OF PRODUCT PRICE COMBINATIONS	51
FIGURE 6 - EXPONENTIAL GROWTH OF COMBINATIONS	53
FIGURE 7 - ALGORITHMIC DECISION TREE FOR CHOOSING SUPERMARKETS	58
FIGURE 8 - WEB SCRAPING PROCESS (PERSSON 2019)	64
FIGURE 9 - FEATURE ENGINEERING PROCESS	ERROR! BOOKMARK NOT DEFINED.
FIGURE 10 - SERVICE STUDIO GRAPHICAL USER INTERFACE	75
FIGURE 11 - UX WIREFRAMES MADE IN FIGMA	76
FIGURE 12 – LONGER VERSIONS OF USER NARRATIVES.....	ERROR! BOOKMARK NOT DEFINED.
FIGURE 13 - SHORTER VERSIONS OF USER NARRATIVES	ERROR! BOOKMARK NOT DEFINED.
FIGURE 14 – LONGER VERSIONS OF USER NARRATIVES CREATED FOR SUPERPOUPA	ERROR! BOOKMARK NOT DEFINED.
FIGURE 15 – LONGER VERSIONS OF USER NARRATIVES CREATED FOR SUPERPOUPA	ERROR! BOOKMARK NOT DEFINED.
FIGURE 16 - PRIORITIZATION QUADRANTS FOR SUPERPOUPA.....	ERROR! BOOKMARK NOT DEFINED.
FIGURE 17 - SUPERPOUPA'S FIRST AND SECOND VERSION	ERROR!
FIGURE 18 - PRIORITIZATION QUADRANTS FOR SUPERPOUPA	ERROR! BOOKMARK NOT DEFINED.
FIGURE 19 - SUPERPOUPA'S FIRST AND SECOND VERSION	ERROR! BOOKMARK NOT DEFINED.

FIGURE 20 - SUPERPOUPA'S CURRENT VERSION. FROM LEFT TO RIGHT: DROPDOWN SEARCH BAR AND CATEGORY LIST, SUBCATEGORY POP-UP, API LIST SPLIT BY SUPERMARKET, FINISH SHOPPING POP-UP, PROFILE WITH RECENT TRIPS	FIGURE 21
- SUPERPOUPA'S FIRST AND SECOND VERSION	ERROR! BOOKMARK NOT DEFINED.
FIGURE 22 - SUPERPOUPA'S CURRENT VERSION. FROM LEFT TO RIGHT: DROPDOWN SEARCH BAR AND CATEGORY LIST, SUBCATEGORY POP-UP, API LIST SPLIT BY SUPERMARKET, FINISH SHOPPING POP-UP, PROFILE WITH RECENT TRIPS.....	83
FIGURE 23 - PROCESS FLOW BETWEEN BLOCK 1, FLASK API, AND BLOCK 2	FIGURE 24 - SUPERPOUPA'S CURRENT VERSION. FROM LEFT TO RIGHT: DROPDOWN SEARCH BAR AND CATEGORY LIST, SUBCATEGORY POP-UP, API LIST SPLIT BY SUPERMARKET, FINISH SHOPPING POP-UP, PROFILE WITH RECENT TRIPS.....
FIGURE 25 - PROCESS FLOW BETWEEN BLOCK 1, FLASK API, AND BLOCK 2	84
FIGURE 26 - PROCESS FLOW BETWEEN BLOCK 1, FLASK API, AND BLOCK 2	ERROR! BOOKMARK NOT DEFINED.

1.1.3 List of Tables

TABLE 1 - DATA MODEL CHARACTERISTICS	ERROR! BOOKMARK NOT DEFINED.
TABLE 2 - SORTED PRICE VARIATION FOR COFFEE	55
TABLE 3 - SORTED PRICE VARIATION FOR MILK.....	55
TABLE 4 - SORTED PRICE VARIATION FOR CHICKEN.....	ERROR! BOOKMARK NOT DEFINED.
TABLE 5 - SORTED PRICE VARIATION FOR BEER	ERROR! BOOKMARK NOT DEFINED.
TABLE 6 - CHEAPEST (MINIMUM) PRODUCTS PER UNIQUE SUBCATEGORY ACROSS SUPERMARKETS	56
TABLE 7 - MINIMAL PRICE COMPARISON FOR CHICKEN, BEER, MILK, AND COFFEE	56
TABLE 8 - ALL POSSIBLE CHEAPEST PATHS	57
TABLE 9 - CHOSEN CATEGORIES	68
TABLE 10 - GROUPING SUPERMARKETS' ORIGINAL CATEGORIES.....	69
TABLE 11 - SPLITTING CONSTRUCTED CATEGORIES	70
TABLE 12 - STANDARDIZING PRICE FEATURES	ERROR! BOOKMARK NOT DEFINED.
TABLE 13 - ENCODING NOMINAL DATA	71

1.1.4 List of Abbreviations

Abbreviation	Definition
API	Application Programming Interface
CAGR	Compound Annual Growth Rate
COVID-19	Coronavirus Disease 2019
E.g.	Example Given
Et al.	Et Alia (and others)
Ibid.	Ibidem (of the same place)
i.e.	id est (that is)
IBM	International Business Machines
LCNC	Low-Code and No-Code
MVP	Minimum Viable Product
SPSS	Statistical Package for the Social Sciences
UI	User Interface
UX	User Experience

1.1.5 Access to SuperPoupa

For accessing and testing the mobile application using Android OS scan the QR code below. You will then be forwarded to a screen that allows you to install an .apk file on your device.

Native Platforms

Generate a mobile app that can be distributed through app stores and can use native capabilities.



Scan the code to install the Android or iOS app on your device.
You can also get the build file [from the download page](#).
In some instances we automatically generate a new build of your mobile app - [see when this happens](#).

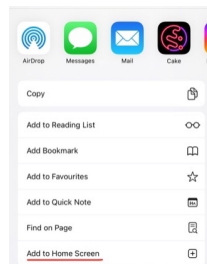
For accessing and testing the mobile application using iOS scan the QR code below. A new browser tab will open on your device. You can either use the app there or add it to your home screen.

Progressive Web App (PWA)

Mobile-optimized apps that work on any device and can be installed from the browser.



Scan the code to access your app and add it to your device's home screen.
You can also [click here to open in the browser](#)



Once the app is opened, you will need to register to access all its functions. After the registration has taken place, you will proceed to the login screen. Once logged in you should be able to see the start screen from which you can use all current features of SuperPoupa.

The currently available categories to search for are: *Água; Azeite; Banana; Batata; Brócolos; Carne de Vaca; Cebola; Cerveja; Couve Flor; Curgete; Cogumelos; Frango; Kiwi; Leite; Limão; Maçã; Manteiga; Manteiga de Amendoim; Ovos; Papel Higiênico; Pepino; Pimento; Tomate*

Following this link, a screen recording on how to use the application can be watched for further instructions: <https://drive.google.com/file/d/1555EepgHTmhWkf1txrG5Hs9kc-re-KaW/view>

EXECUTIVE SUMMARY (GROUP PART)

The global food crisis, characterized by rising food prices and disruptions to supply chains, has sparked growing concerns. These developments have contributed to a cost-of-living crisis in Europe, where stagnant wages and rising costs for essential goods and services have left many households struggling to afford their basic needs. In response to the below-displayed problem statement, this report demonstrates the potential for and the development process of a mobile application that allows consumers to find the best value-for-money grocery items and how this presents a viable business opportunity.

"Currently, consumers in Portugal experience an unprecedented spike in food prices due to inflation, leading to an increased loss of their net real income."

The literature revealed that COVID-19 and the war in Ukraine in rapid succession weighed heavily on global food production. Consequently, Portugal's food inflation reached 18,9%, equalling Europe's average, representing the fastest rise on record. As 97% of Europeans saw their grocery bills increase more rapidly, it created a new market of both lower- and middle-class households who, in response, are forced to reduce discretionary spending and prioritize essential products' price-quality ratio. Moreover, the lingering energy crises and Portugal's recent extreme weather conditions question the notion of food prices levelling out soon. As such, many believe that the identified market and the number of Portuguese households facing acute food insecurity will continue to increase over time.

The quantitative analysis of the Portuguese market showed that 93% of consumers feel the recent price surges impacted their spending potential. Additionally, 85% indicated their already conscious and directed effort to compare grocery prices before purchasing. As two-thirds of these consumers use their mobile devices to do so, the threshold for incorporating a

mobile solution into daily life was lowered. Also, participants who did not compare prices blamed this on the absence of effective solutions allowing them to, although 79% indicated their willingness to adopt such a solution, should it be available. As such, an opportunity presented itself with a digital application that would help the growing market identify the best value-for-money groceries.

This application, which was named SuperPoupa, limited users' product choice to categories (e.g., Milk) and subcategories (e.g., Skimmed Milk) when creating their shopping list, whereafter it would locate the cheapest items across Portugal's various supermarkets. The resulting list is what was coined "the cheapest path". This simplified two-step list-building approach played into the fact that many consumers now disregard the brand and opt for the best price-quality ratios. The solution's development was split into two sections: the back end, including the algorithm's programming in Python and the data model's architectural design, and the front end, including basic wireframe modelling and translating these into high-fidelity functional interfaces using OutSystems.

The resulting Minimum Viable Product, or MVP, was field-tested by 27 users. The findings suggest that inflation adversely affects consumers and reveals their positive attitude towards using SuperPoupa, due to its simplicity and potential for significant time and money savings. The approach taken by SuperPoupa thereby offers an entrepreneurial opportunity to provide real value to consumers in Portugal. In the future, introducing new features could further expand the target audience, address sustainability, health, and ethical considerations, and protect the application's ability to solidify its roots in the Portuguese grocery application market. However, further research and widespread implementation of SuperPoupa will be necessary to evaluate its potential and financial viability fully.

2 INTRODUCTION (GROUP PART)

2.1 Contextual Background

Concerns about a potential food crisis have been spreading worldwide over the past two years as global food prices have risen to alarming levels (Vos et al. 2022). The accumulation of multiple economic shocks, combined with extreme weather events, has disrupted global supply chains, exacerbating existing price trends and resulting in a European "cost-of-living crisis" (Tetlow 2022; Patrick and Pybus 2022). This crisis places millions of households under financial strain as prices for essential necessities spike while nominal wages remain stagnant not adjusting to the inflationary price environment (Ibid.). Moreover, due to lingering energy crises, the rising disparity is predicted to widen as time progresses (World Economic Forum 2022). Given that lower-income households spend as much as one-third of their disposable income on groceries (Wesley and Peterson 2022) the role of groceries in the development of living standards has become increasingly important: rising food prices have already caused the global number of people facing acute food insecurity to more than double to 345 million since 2019 (World Food Programme 2022). The ongoing deterioration of economic conditions in Europe suggests that living standards are likely to be further negatively impacted, emphasizing the necessity of prompt implementation of solutions (Patrick and Pybus 2022).

Information transparency—referring to the degree of visibility and accessibility of information (Zhu 2002) – delivered through technological advancements could provide support to consumers by allowing them to make more informed decisions, potentially lessening the financial strain caused by the cost-of-living crisis. Thus, this report investigates the potential of a mobile application, hereafter referred to as *SuperPoupa*, to assist consumers in Portugal in mitigating the effects of escalating food prices. Based on extracting price-data from three major

supermarket-chains in Portugal (*Continente, Pingo Doce, Intermarché*) SuperPoupa allows consumers to get the cheapest product combination of an inserted shopping list using a price-optimizing algorithm developed for this project. The application is therefore the proposed solution to the increasing challenges various households will prospectively face with the emerging new economic climate.

2.2 Problem definition

This report examines the difficulties posed by the current economic situation to customers and investigates how the implementation of a mobile application can enable customers to identify best value-for-money grocery items, thus helping them to adjust to the new economic environment. As such, a problem statement based on the “Five-W” methodology (Jia et al. 2016) helped to form a definition acting as a focal point for the research process and solution design of this work project.

“Currently (*when*), consumers (*who*) in Portugal (*where*) experience an unprecedented spike in food prices due to inflation (*why*), leading to an increased loss of their net real income (*what*).”

2.3 Objectives and Structure of this Report

As touched upon, this report aims to demonstrate that the identified problem gives way to a viable business opportunity. Thus, it begins by macro-analysing the influence of recent geopolitical prior events on the food industry. Then, it further supports the hypothesis that recent food inflation has caused many consumers to adjust their spending habits to mitigate its effects.

Secondly, the report will investigate the business opportunity that has arisen from the current gaps in the market and the shift in the economic climate. This section will introduce the

proposed solution SuperPoupa: a mobile application that allows consumers to get the best value-for-money grocery items. This report narrowed its focus on critical elements of the application's business model by drawing on insights gathered from the consumer survey in accordance with or contrasting literature findings. Accordingly, it provides a comprehensive overview of the solution's technical and conceptual development processes, including the instruments used. Lastly, while testing the application's first iteration, later called Minimum Viable Product (MVP), a product evaluation tool will assess how effectively users perceived the suggested solution to address the overarching problem. The report's conclusion will list its key findings and provide a prognosis for the future.

3 ANALYSIS AND DIAGNOSIS (GROUP PART)

3.1 Literature Study

It is essential to cast a view of the recent past to assess how cascading geopolitical events shaped the current economic climate surrounding the identified problem. The following section therefore explores how COVID-19 disrupted the global economy and analyses how the war on Ukraine prevented a post-pandemic economic recovery. The effects of resulting global inflationary trends on the Portuguese food industry will be analysed. Finally, the effects of the rising food prices and the behavioural response to it of both lower- and middle-class households will be covered.

3.1.1 Economic Sentiment Influenced by COVID-19 and the War in Ukraine

COVID-19 sent shock waves through the world economy, causing the past century's largest global economic crisis (WorldBank 2022). It did so by disrupting supply chains and crippling industries; thus, millions were left without employment (Dua et al. 2020). As is often the

case, lower-income households were stuck hardest (CBPP 2021). Besides mortality rates, these households disproportionally experienced food and housing insecurity due to supply-demand imbalances considering their limited budget to compete with other consumers (Parker et al. 2020).

As soon as the pandemic halted and governments alleviated peoples' and businesses' restrictions, governments and industries attempted to curb the pandemic's economic fallout (Hanselman 2022). They identified and supported initiatives that drove the economic recovery, promoted domestic spending, and protected local businesses from foreign competitors (Ibid.). Even so, the pandemic had positive repercussions. For instance, the virtual space of transactions and communication flourished into a novel economic asset, especially in third-world countries (Shang et al. 2021). Additionally, the pandemic forcibly drove entire industries, such as e-learning, logistics, and entertainment, to new levels of efficiency (Jensen 2020).

However, before the economy had time to think in future terms, Russia invaded Ukraine in February of this year (Kirby 2022). The invasion prevented a fragile post-pandemic economy from recovering and added to the supply shocks that had already struck the global supply chain (United Nations 2022). On an industry level, the war's effects propagated far and wide through trade, commodity markets, and financial relationships (Gourinchas 2022). Concerning the food industry, Russia and Ukraine exported about 30% of the world's wheat and 20% of the world's corn (Nelson 2022), reducing supplies significantly (Green 2022). So far in 2022, their prices jumped 37% and 21%, respectively, after already rising more than a fifth in 2021 (Al Jazeera 2022). As a result, households that struggled to come by during the pandemic were now faced with soaring food and energy prices (Smith and Bailie 2022).

Consequently, according to the International Monetary Fund and the United Nations, many will experience 2023 as a recession (Ward-Glenton 2022). While sizeable, they expect

these effects not to be large enough to completely derail global recovery from the pandemic (Ibid.). Historically, however, central reserves associate these two events' increased geopolitical risk with accelerated inflationary shifts worldwide (Caldara et al. 2022).

3.1.2 The Effect of Inflation on the Food Industry

Portugal's current annual inflation rate is projected to increase to 10.2% in 2022, whilst general food prices jumped 18.9%, representing peaks since May 1992 and June 1990, respectively (Trading Economics 2022). Moreover, although Brussels predicts the inflation rate to go down in 2023 (Euronews 2022a), the lingering energy crisis is expected to keep eroding Portuguese households' purchasing power and weigh heavily on food production and, thus, consumption (Ibid.). To make matters worse, Portuguese farmers recently experienced more severe and frequent droughts and extreme weather scenarios (Euronews 2022b), as well as surging prices of fertilizer, of which Russia exports one-third globally (Smith and Bailie 2022). These compounding hurdles facing the food industry's supply chain challenge the notion that food prices will decrease or even level out any time soon. In turn, it raises the question of how one can genuinely expect inflation rates to restore, given the food industry's supposed central role (Wallace 2022).

That said, it is interesting to consider the disparity in how social and economic turbulence affects different facets of society. It is well-established that low-income consumers are hit hardest for reasons discussed previously (Parker et al. 2020; Smith and Bailie 2022). Looking at the supply side, however, corporations seem to tell a much different story (Schweizer 2022). When faced with inflation, they have the luxury of choosing between different approaches that mitigate their exposure to financial risk (Ibid.). First, they can shield consumers and take a hit on their margins. Second, they can pass through and evenly share the pain with customers. Alternatively, they put an additional markup above and beyond the cost increase rate, padding their

margins at customers' expense (Ibid.). Unfortunately for the customer, companies often choose the latter, with half of consumer price inflation caused by this profit-driven approach (Bivens 2022).

As a result, market leaders are breaking year-over-year profit records in numerous industries. For example, Cargill, a commodity conglomerate, saw their most profitable year in 2021 (Ezell-Thomas 2022), while Coca-Cola, Hershey's, PepsiCo, and Mondelez, surpassed earnings estimates between 7% and 16% (Schweizer 2022). Also, when comparing recent price and wage increases, there is little to no evidence of a correlation between the acceleration in price inflation and wage growth in said industries (Schweitzer and Khattar 2022). Even so, three-quarters of middle-class households claim their wages are falling behind inflation (Uhl 2022). The Bureau of Labor Statistics affirms this claim in documenting inflation outpacing nominal wage growth for 18 months straight (Richter 2022).

Meanwhile, this year, the average CEO-to-worker pay ratio was 324 to 1, an increase of 23% from 2019, which doubled the inflation rate (Schweizer 2022). However, the above makes it seem as if corporations rationalize from an opportunistic point of view, perceiving recent crises as chances to line their pockets. Hence, instead of using their vast wealth and resources to shield their clientele from adversity, they proactively pass the largest burden to those with the most difficulty dealing with it (Bivens 2022).

3.1.3 The Effect of Food Inflation on Middle-Class Households

Where most have seen their grocery bills increase, those not severely affected often cannot fathom the burden this poses for low-income households. The reason is simple. Low-income families spend a much more significant chunk of their disposable income on food than the median household. On average, middle-class families spend 12% of their disposable income on food, whilst with low-income households, this number can be as high as 27% (Soll 2022). This

discrepancy can be explained by a significant shift in spending patterns in the 20th century (Chao and Utgoff 2006). During the 1900s, life's necessities, including food, were relatively much more expensive compared to today, leaving little room for expenditure on additional goods and services (Ibid.). On average, families spent 54% of their income on food and clothing in 1901, which dropped to a mere 14.2% in 2020 (Bureau of Labor Statistics 2022). This sharp decline led to a massive reshuffling of middle-class budgets, carrying over large chunks to transportation, insurance, and discretionary spendings such as eating out, vacations, and entertainment (Soll 2022).

For now, the data from McKinsey's latest European Consumer Survey carried out this October indicate that households are gradually reversing this process. As 97% of people experience their groceries becoming increasingly expensive (Bazzoni et al. 2022), "consumer pessimism is at an all-time high in Europe, with rising prices the number-one worry" (World Economic Forum 2022). Both are not entirely out of the ordinary, as Portugal's earlier mentioned 18,9% food inflation equals Europe's average, which is the fastest rise on record (Alderman 2022). More than 80% of consumers indicate to have altered their grocery shopping behaviour, and value for money and promotions seems fundamental to those changes (Bazzoni et al. 2022). As such, consumers' strategies include looking diligently for items on sale and, most of all, trading down – that is, they buy the store brand (Vardhini 2020). Every category except fruit and vegetables, meat and fish, and alcohol has seen more than half of European consumers switching to those cheaper store brands (Bazzoni et al. 2022). Even so, almost one-third have entirely switched to alternative stores, often in the form of discount retailers (World Economic Forum 2022). Regarding discretionary expenditure, cinemas in Germany, France, Italy, Spain and the UK saw a drop in revenue to 59% below pre-pandemic levels in September and October (Palmer 2022). Although not to this extent, numerous industries have experienced similar

trends, as two-thirds of consumers indicate to cut out jewellery, apparel, and consumer electronics expenditure in the coming season (Bazzoni et al. 2022).

3.1.4 The Effect of Food Inflation on Low-Income Households

Nonetheless, where middle-class families grew into this position from which concessions could be made, the 20th-century wealth redistribution mainly left low-income families behind. As they kept devoting most of their income to food and shelter (Soll 2022), these households still possess little to no discretionary spending to cut back on to free up funds for food (Genetian et al. 2021). Therefore, as a last resort, over a third of European consumers, including half of the millennials, withdrew money from savings to cover their expenses in the last three months (Bazzoni et al. 2022). Even prior to the recent turmoil, in 2020, 9.8% of the Portuguese population had already experienced food insecurity (Aguiar et al. 2022). In layman's terms, this can be defined as being without reliable access to sufficient, affordable, and nutritious food (Bahn and Bahn 2021). This uncertainty has a toll on people beyond 'merely' insufficient funds. A study at Sacred Heart University, Connecticut, found that food insecurity causes people to consume less protein and vegetables and rely more on sugary and fatty foods (Solis-Moreira 2022).

One explanation is that as inflation gradually pushes more expensive healthy produce out of reach, highly processed foods become an attractive alternative since they are more affordable and have a longer shelf life (Drexel University 2022). As such, health professionals fear prolonged inflation given that these products increase the risk of diet-related diseases in the long run (Fu 2022). Furthermore, in association with the consumption of these unhealthy products, a United States national study found that the mental strain caused by food insecurity is linked to a 257% higher risk of anxiety and a 253% higher risk of depression amongst low-income households (Fang et al. 2021).

Including Portugal, the global number of those facing acute food insecurity has soared from 135 million to 345 million since the pandemic's start (World Food Program 2022). Moreover, given the risk of food prices rising even further through climate change and the continuation of war, the number of people facing such physical and mental hardship is prone to increase (IGD 2022). Therefore, a tool that allows these consumers to save money on grocery expenditures and prevent or alleviate food insecurity could be relevant. However, such drastic changes are challenging to implement within rigid food corporations, especially considering that, in this case, some seem to be at the core of the problem (Soll 2022). Accordingly, the following section will explore the feasibility and necessity of a solution from the consumer's perspective as food prices continue to rise in Portugal (Donn 2022).

3.2 Empirical Research

Developers create mobile applications to provide users with an improved and more engaging way of accessing information, services, and products (Kim et al. 2013). Thus, placing users at the center of an application's development processes is a crucial principle of mobile application design (Uddin 2022). Such user-centricity involves understanding users' needs, preferences, and behaviours and designing products and services that meet those needs (Ibid.). Therefore, companies must strive to uncover and comprehend these data for the application to provide perceptible value (Ibid.).

Customer surveys are helpful in this process, as they provide insights into areas the app development team may need to explore. Thus, to understand customers' perception of Portugal's grocery industry, a survey was conducted as part of this project. This survey investigated various aspects of a potential solution to combating rising grocery prices through a mobile application. These include participants' perceptions of price increases on grocery items and the current usage of technology to optimize their shopping routine. As the overarching problem was now

clearly defined, the survey's secondary focus was on the needs and expectations of users arising from this problem. These included participants' willingness to incorporate new tools into their shopping routine, preferred features they would like to see in the said tool, and willingness to pay for such services. These findings provided a lens through which the application's business model was conceived and allowed to pinpoint different use cases with varying demographic- & socioeconomic backgrounds. The latter enabled a deliberate and user-oriented application development process.

3.2.1 Research Instrument

The survey was undertaken from the 9th of November until the 9th of December 2022 and included 31 questions designed to gain insight into customers' needs and areas of difficulty. The survey was published in both English and Portuguese to improve its reach in Portugal. Although the sample size of 74 responses does not provide statistically significant results, the findings are an essential indicator for ensuring user-centricity in app development. The following demographic statements about the survey participants represent the consecutive findings.

The survey participants were predominantly female (71%), and the majority were aged between 25 and 34 (48%). A vast share of participants was from Portugal (39%), Germany (36%), and other European and non-European countries. Most participants were full-time employed (47%), while the other more significant portion were students (37%).

3.2.2 The Perception of Inflation and Its Effect on the Reduction of Real Net Income

Question four to six explored participant's perception of the current economic situation. The results showed that 62% of participants felt that their budget for covering basic needs was adequate, but 86% reported feeling an increase in prices and the impact of inflation on their spending potential. In the Portuguese sample, 92% reported a decrease in disposable income

due to inflation, with 36% specifically citing rising grocery prices as a major concern. These findings suggest that solutions to reduce grocery spending could be beneficial for consumers.

3.2.3 Users' Willingness to Adapt New Technologies for Reduced Grocery Spending

Question 10-13 and their associated follow-up questions explored participants' shopping habits and their pre-existing use of technology-based tools to improve their shopping experience. The results showed that 49% of respondents compared prices before making a purchase, with a majority of these being Portuguese participants (85%). These results suggest a potential need for technology-based solutions in the market, as all respondents cited the lack of convenient solutions as the reason for not comparing prices.

Furthermore, 79% of respondents indicated that they would be receptive to such solutions, with the majority expressing their interest in an application that helps them save money. However, 21% were unwilling to use such an application, citing concerns about engagement, time and effort investment, and loyalty to a single supermarket. All in all, these findings suggest that the Portuguese market may be particularly receptive to the proposed solution, as there is a lack of existing solutions and a potential competitive advantage in addressing this void.

3.2.4 Assessing User's desired Features and their Willingness-To-Pay for it

The survey's final section explored participants' preferences for additional features in a proposed mobile application and their willingness to pay for such a product. The results showed that 32% of international respondents were interested in a virtual piggy bank, while 36% of Portuguese respondents preferred the ability to turn loyalty points into coupons at supermarkets. These findings suggest that forming strategic partnerships with retailers could be beneficial for the success of the proposed application.

Further, although 59% of participants showed a willingness to pay for the application, the remaining 41% cited a need for more trust in the promised savings. In addition, reluctance

to pay could also stem from the paradoxical implications of paying for a solution whose sole purpose is to help consumers save money. To address these concerns, the offering of a trial period and incorporating advertising revenue instead of charging users directly may need to be considered. Together, these findings provided important insights into potential revenue models and user adoption barriers.

4 SOLUTION PROPOSAL

4.1 Opportunity Identification (Individual Part: Pepijn van Es)

The literature demonstrated that the conflict in Ukraine had had a profound effect on global supplies of wheat and corn, causing the price of food to skyrocket (Caldara et al. 2022; Hanselman 2022). This has created a new market segment of people who, in response, have adopted similar shopping behaviours (Bazzoni et al. 2022). These entailed reducing discretionary spending and finding the best value for money for essential products. With the World Economic Forum and the International Monetary Fund predicting price spikes to continue into the first quarter of next year (Gourinchas 2022; Ward-Glenton 2022), this shopping pattern will likely remain prevalent. Therefore, it is fair to assume that this identified market will continue to grow at least until early next year.

Current economic sentiment backs this assumption, as 83% of consumers do not trust that the economy will recover within six to twelve months (Bazzoni et al. 2022). Hence, an opportunity presents itself with initiatives to help the growing market save money by identifying the best value-for-money groceries. Even so, for many middle-class households, this is the first time they need to adopt such shopping behaviours (Soll 2022). Thus, on the one hand, a potential solution can assist these consumers with effectively adjusting to the new economic

climate. On the other, it could help low-income consumers with finetuning the shopping tactics they used all along (Leibtag and Kaufman 2003).

Additionally, although 85-90% of sales still occur in offline stores (Badenhop and Frasquet 2021), many supermarkets have developed beyond their brick-and-mortar business models to gain shares in the growing e-grocery segment to improve their customer service (Ibid.). Consequently, now operating as multi-channel retailers, supermarkets have been posting product information online. The disclosure of such information is crucial in developing a solution since product data needed to support consumers' decisions is now readily available. However, product data are scattered between countless websites and, thus, seemingly, have yet to be centralized and presented practically for consumer use. Empirical data backs this assumption as 71% of consumers believe that currently, no tool allows them to quickly find the cheapest items, even though 79% state they would adopt such a solution.

Furthermore, technological media have tremendously increased the bandwidth between consumers and such online information by making access simple, fast, and affordable (Ford 2012). As a result, two-thirds of shoppers already utilize this transparency by using their mobile devices to research goods and compare prices (Freer 2022). On the one hand, this shows that information search has become a conscious and directed effort that facilitates decision-making (Chen et al. 2022). On the other hand, from a psychological standpoint, these consumers already associate their mobile devices with the contextual cue – a trigger to specific behaviour – of going grocery shopping (Freer 2022). This connection could significantly reduce the barrier for individuals to incorporate a solution that utilizes their mobile device as a platform into their daily lives (Charm et al. 2020).

4.2 Solution Introduction (Individual Part: Stefanie Neumann)

On that note, the proposed solution, named SuperPoupa, will be introduced in the following section. In a nutshell, SuperPoupa helps customers to find their groceries at the lowest price as fast as possible across Portugal's various supermarket chains. As branding is losing its significance (Bazzoni et al. 2022), the app plays into the fact that many consumers are 'trading down' and are simply looking for the best price-quality ratio (Ibid.). The main difference between SuperPoupa and competitive applications is that instead of searching for a particular product of a particular brand, the product search engine asks consumers only to choose a product category (e.g., Chicken) and add a corresponding subcategory (e.g., Drumsticks) to the list. Having repeated this two-step approach as often as necessary, the algorithm running in the back-end of the app finds the cheapest items within each product subcategory and returns the cheapest possible grocery list. Depending on the user's product choices, the resulting list may include items from multiple supermarkets, such as Pingo Doce and Intermarché. In the future, users will have the option to select the number of supermarkets they wish to visit to maximize their savings.

4.2.1 Business Model

For developing SuperPoupa's business model, the Business Model Canvas by Osterwalder and Pigneur (2011) served as the conceptual framework. Due to the nature of the proposed solution, emphasis was placed on examining the success factors for mobile applications business models. Before delving into the specifics of the building blocks of the Business Model Canva, a review of the "Application Economy" will enhance the comprehension of the unique challenges that the business environment for mobile applications entails. The comprehension of this mobile application market's unique features is crucial as, due to the novelty of the market, frequent changes place businesses in precarious circumstances from which strategic decision-

making is complex (Camponovo and Pigneur 2003). Simultaneously, the market for mobile applications is expanding as increasing profits, and successful ventures draw in more developers, thus intensifying the competition in this sector (Vannieuwenborg et al. 2012). Hence, establishing a competitive advantage is essential for any new application entering the market to gain traction. To construct the optimal business plan for SuperPoupa, the following sections will serve as a foundation for linking pertinent literature discoveries with best practices.

4.2.2 Challenges of the Business Environment

The proliferation of mobile devices has driven the expansion of the mobile app market, characterized by a high volume of app downloads and steady growth (Wac et al. 2011; Fong 2017). The so-called "Application Economy" has become a lucrative business in the digital domain, with an anticipated compound annual growth rate of 9.27% between 2022 and 2026 (Statista 2022). The app stores of major mobile platforms, such as Google's Play Store and Apple's App Store, together boasting roughly 5.7 million applications, function as two-sided marketplaces that facilitate the distribution of software programs and provide convenience for app creators and users (Cuadrado and Dueñas 2012).

However, these marketplaces face the challenge of achieving a critical mass of users and developers to maintain equilibrium and prevent market failure. Network effects, which refer to the increasing value of a platform as more users and developers join, play a crucial role in the strategic decisions of app developers (Heitkoetter et al. 2012). While most app developers struggle to yield a profit, only a few have generated significant revenue from their apps (Tian et al. 2015). Additionally, much research has yet to be done directly addressing successful business model features for mobile applications (Brockmann et al. 2015). Therefore, this chapter aims to bridge the need for knowledge of what applications' architecture helps them prevail and applied these findings to defining SuperPoupa's business model.

4.2.3 Business Model Canvas

Even though business model concepts are widely used to conceptualise corporate strategies, a unified definition of such a concept is absent (Camponovo and Pigneur 2003). However, the Business Model Canvas (2011) developed by Osterwalder and Pigneur has been particularly noteworthy among the multitude of models and frameworks available due to its practicality and the minimal involvement of interviewees (Vannieuwenborg et al. 2012). In this framework, the business model splits into four main elements:

- 1) The product innovation component, including the value proposition
- 2) The customer relationship, dealing with target customers, channels and relationships
- 3) The infrastructure, including critical resources required to deliver the value proposition
- 4) The financials which reflect on revenues and costs of a business.

4.2.3.1 Product Innovation

Value Proposition

A company's value proposition outlines how a value is delivered to its customers via certain services or products, distinguishing it from rival offerings through distinctive characteristics (Payne 2017). The value proposition is thereby addressing which customer needs and problems it is trying to solve with the businesses' bundle of products and services (Osterwalder and Pigneur 2011). Product- or service-bundles can create superior value, and thus a competitive advantage, by differentiating from competitive offerings through either quantitative (price or service time) or qualitative (brand or design) features (Ibid.). For example, in the digital domain of software products, value usually derives from adding convenience and time-, as well as cost-savings to end-users (Pucci 2018).

SuperPoupa's value proposition can be characterized by the amount of time saved and the amount of money saved. The list-building feature enables users to quickly compile a shopping list by avoiding searching for particular items and comparing prices before including them in the list. Simply specifying categories and subcategories saves users time when entering items into their lists. In addition, the potential to obtain the lowest-priced list of products for given categories leads to financial savings. By combining these two aspects, this value proposition accurately addresses the users need for both time-efficient and cost-effective solutions.

4.2.3.2 Customer Relationship

Customer Segments

Identifying the right target customers and understanding their needs and preferences is crucial to the success of mobile applications (Osterwalder and Pigneur 2011). However, the broad availability of apps and diverse use cases make it challenging to identify potential customer segments. Therefore, app companies must appeal to a diverse app-user demographic and meet their demands while distinguishing themselves from the vast number of other mobile apps (Tang 2016).

As shown by the literature analysis and the business opportunity, many households have been severely impacted by the growing grocery sector costs due to the cost-of-living crisis. Whereas previously, only lower-class households had to pay closer attention to their grocery spending, now also middle-class households need to adapt their shopping behaviour due to the changing economic climate (Soll 2022). Consequently, SuperPoupa's main customer groups divide into two distinct segments. First and primarily, consumers, most of which middle-class households, that, for the first time in their life, must concede to the new economic climate by

adopting new shopping behaviours with which the application can help. On the other, the application assists lower-class households with finetuning the shopping tactics they already used to prevent facing food insecurity.

Even though SuperPoupa primarily targets households that aim to reduce or maintain their grocery spending in times of rising prices, additional user groups can also be seen as potential targets for the application. Due to the competitive nature of app store marketplaces, app developers must make their applications available to a wide range of users to remain competitive in the rankings of their respective categories. As a result, they cannot only limit their applications to a specific user group. SuperPoupa must thus open up to the mass market to capitalize on the importance of high rankings as a key factor for revenue success.

Customer Channels

The customer channels of a company describe how the Value Proposition can be delivered to the previously identified target customers (Osterwalder and Pigneur 2011). In the context of mobile applications, distribution channels must be able to effectively deliver the software to the end-users' devices. Distribution channels for mobile apps include app stores, direct downloads, mobile networks, and third-party websites (Beimborn and Palitza 2013). However, app stores have become the leading distribution channel, as they have greatly simplified the distribution of mobile applications (Cramer et al. 2010). App stores provide a convenient linkage between developers and users through their marketplaces by allowing the application providers to access a large potential market with minimal marketing and advertising expenditures while allowing app users to obtain the software with minimum effort (Deloitte 2018). App stores as such work as two-sided marketplaces, where the ecosystem's orchestrator entices content providers and consumers to commit to the mobile ecosystem to achieve network effects (Hyrnsalmi et al. 2012). This interdependence of market participants is called the cross-side

network effect. It occurs when participants on one side of a market benefit from more participants on the other (Collin et al. 2015). However, rating systems can also create same-side network effects, as reviews for an application attract new users who rely on the social proof of other market participants (Ibid.).

App developers who rely on app stores as their primary distribution channel however are faced with the challenge of achieving a high ranking within their application's category, which is determined by user ratings, download volume, and revenue generated (Brockmann et al. 2015). These top list rankings are relevant for applications as they significantly influence the turnover rates and revenue generated, thus determining the success of the application. This circumstances traces back to platforms matching-algorithms which is based on these top ratings (Bresnahan et al. 2014), making it difficult for app developers to gain popularity among the nearly 1 million apps competing for a user's attention. To overcome this challenge, the proposed solution, SuperPoupa, should increase its exposure in app stores through various marketing activities, such as paid advertising, media outreach, and strategic partnerships with retailers. These partnerships can benefit both sides, as the application gains increased visibility and retailers benefit from the additional exposure of their products. Developing a strategic marketing plan to maximize the potential of the app store distribution channel and take advantage of the momentum created by the cost-of-living crisis is crucial for the success of SuperPoupa.

Customer Relationships

Designing appropriate relationships to customers impacts greatly the overall experience with a company's services or products (Osterwalder and Pigneur 2011). The formation of customer relationships is dependent upon the customer's specific needs and expectations in relation to their willingness to pay for services (Ibid.).

Due to app stores' role as intermediaries between users and developers the customer relationship to users is defined through these platforms to a great extent: as users execute the search for an application, the purchase of it and the download to the end-device all via these marketplaces app developers just come into action once users open the application.

SuperPoupa's engagement with its users consequently starts when they leave the sphere of the app stores and open the downloaded application for the first time. Users will then be intuitively guided through the initial set-up steps of the application to become familiar with the interface and features. Thus, designing the application with an intuitive orientation for users therefore is an important objective in terms of user-centricity, which also largely shapes the resulting customer relationship of SuperPoupa to its users. As the application seeks to address a large user base, automating customer relationship is a necessary option to handle growing user volumes. For the automation to be successful, users must be able to use the application without needing personal assistance. This gives important implications for the app development, as creating an intuitive user experience is paramount for automating the customer relationships.

However, SuperPoupa also prospectively seeks to integrate its customers into the value-creation process by allowing users to update price data or insert product information. This form of customer engagement not only enhances the accuracy of displayed prices and improves the product's user-centricity, but also facilitates the creation of a supportive community within the application, thereby reducing customer support costs.

4.2.3.3 Infrastructure

Key Resources

A company's essential resources are the tangible or intangible assets necessary to make its business model work (Brockmann et al. 2015). For mobile application this includes the resources required to develop, implement, and operate a software product (Vannieuwenborg et al. 2012). For example, for SuperPoupa, a mobile application that allows users to compare grocery prices between supermarkets, the most important resource is the intellectual property related to the app and the personnel with the necessary skills and expertise to develop and deploy it (Thomke and Reinertsen 2012).

Initially, the MVP for SuperPoupa was developed using a combination of traditional programming in Python and low-code development in OutSystems. This required advanced programming knowledge to build the Python algorithm for price comparison and the back-end architecture. In the future, human expertise in both traditional and low-code development will remain the key resource for the operation of the business. In addition to human resources, the availability of price data from supermarkets is also a critical resource for SuperPoupa to provide the best-money-value for grocery items. The current technological approach allows for independence from strategic partnerships with grocery retailers. However, such collaborations' potential benefits should be considered to maximize value for both parties in the long run (Vannieuwenborg et al. 2012).

Key Activities

The key activities undertaken by a company are the tasks necessary for its successful operation (Osterwalder and Pigneur 2011). For mobile applications, these activities include the maintenance and development of the app itself, as well as customer support to ensure user satisfaction (Vannieuwenborg et al. 2012).

In the case of SuperPoupa the key activities include the development of the app and its associated services to provide the desired functionality. This includes tasks such as customer research, UI/UX development, algorithm enhancement, frequent supermarket data procurement, and marketing activities. These activities are essential for establishing SuperPoupa as a viable business opportunity and ensuring its sustained commercial success.

Key partnerships

The central partnerships formed by a business often compromise a network of important suppliers relevant for the business model to work (Osterwalder and Pigneur 2011). Partnerships can thereby be formed as a requirement for gaining resources, but also to reduce risk as an alliance or optimize the business model (Ibid.).

SuperPoupa's key partnerships are with grocery retailers. These partnerships are essential for the company's business model, as it relies on the ability to extract price data from the retailers' product assortment. In the past, supermarkets have increasingly recognized the benefits of establishing an e-commerce presence in addition to their physical stores, allowing them to improve the customer experience and extend their reach (McLean et al. 2020). SuperPoupa can benefit from this trend by using web scraping, a process of automated extracting data from websites, to acquire price information from retailers' websites.

Although SuperPoupa does not need to form official partnerships with retailers to access their pricing data, strategic partnerships with these retailers can still be beneficial. For example, voluntarily providing data could increase the accuracy of the displayed price information and reduce costs by eliminating the need for web-scraping. Additionally, such partnerships could improve the visibility of both parties, increasing exposure to potential customers.

4.2.3.4 Financials

Revenue Streams

Generating revenues are the most vital part of a business, as without sufficient financial streams the survival of a company cannot be ensured (Osterwalder and Pigneur 2011). Companies must quantify the value of their product- or service-bundle and assess customer's willingness-to-pay in order to establish the most profitable strategy for their business.

SuperPoupa's revenue streams are based on a mixed approach that combines in-app purchases and advertisements to generate revenue while maintaining its mission of helping users save money. This approach allows the company to serve a broad user base without charging upfront fees, which could create adoption barriers for potential users and reduce the savings generated through the app. In addition to in-app purchases and advertisements, SuperPoupa can also incorporate premium features within the application that users can pay to access. These features can provide added value to users and generate additional revenue for the company.

The choice of revenue streams for SuperPoupa is influenced by the "Price Conundrum" of in-app selling, which refers to the difficulty of generating revenue from mobile applications due to consumers' expectation that online information is accessible (Ford 2012). To overcome this challenge, SuperPoupa must carefully consider its pricing mechanisms to generate revenue, taking into account its primary objectives and the potential risks and opportunities associated with different pricing strategies (Osterwalder and Pigneur 2011; Tang 2016; Tani et al. 2022). In the case of SuperPoupa, the company's main objective is to attract a large user base and support them in adapting to inflationary trends in the grocery sector, which makes an upfront payment for the app an unsuitable option. Instead, the company can use in-app purchases and advertisements to generate revenue without charging users for the app itself.

Cost Structure

All the previously described activities that are necessary to deliver value to the customers incur costs stemming from the business's activities (Osterwalder and Pigneur 2011). Companies thereby either can be driven by cost or by value: depending on their value proposition companies rather focus on minimizing costs or delivering premium value, which in turn incurs higher costs (Ibid.).

SuperPoupa's cost structure is designed to be cost-oriented due to the competitive nature of app stores. It must keep costs low until the app generates enough revenue to cover business expenses and ensure the business's survival. The main costs associated with the development of SuperPoupa are personnel wages, IT equipment and infrastructure, and marketing costs to promote the app. In the initial stage of the business, SuperPoupa may only make ends meet once the app is widely adopted and generates income from advertising and the sale of premium features. However, the potential for low-code development and agile product development can help mitigate costs and offer a working MVP that can generate early revenue.

The distribution of SuperPoupa will also result in decreasing marginal costs due to the scalability of mobile applications. Once the user base expands, the costs per unit will decrease, and the company's competitive position will improve. This is an essential advantage for SuperPoupa, as it can help the company to generate revenue and support its growth and success. In addition to the costs associated with the app's development, SuperPoupa must also consider the fees app stores charges (refer to Appendix) for access to their distribution channels. These fees can add to the costs of the business and must be accounted for in the company's financial planning.

Commented [SN1]: Add Appendix Nr

4.2.4 Competitive Analysis (Individual Part: Stefanie Neumann)

As described in the preceding sections, the competition between developers to reach the top of the app store listings is fierce. Not only do applications compete with applications that offer similar values to users, but they also compete with applications that operate in the same field but offer different features. As of SuperPoupa this entails all the applications operating in the "Shopping Category" segment in the mobile app stores. This category includes online shopping apps, price comparison apps, coupon apps, and retail-specific applications. As SuperPoupa is an application that enables users to create shopping lists, compares prices of various retailers based on the inserted list and optimizes shopping trips, various direct and indirect competitors can be identified.

Direct competitors include all applications that tackle the main objective of generating customer savings in grocery items. However, indirect competitors may also threaten the success of SuperPoupa: mobile applications that cover other grocery-related activities may reduce the number of potential users interested in SuperPoupa, as they may be chosen instead. Examples of those can be applications for tracking expenses or creating shopping lists. The following section will shed light on SuperPoupa's noteworthy competitors and will elaborate on how to differentiate from those accordingly.

4.2.4.1 Direct Competitors

As direct competitors of SuperPoupa, mobile applications with the primary objective of comparing grocery prices and thereby allowing users to optimize their spending can be considered. As a result of supermarkets' evolution into omnichannel retailers, numerous price-comparison mobile applications have been developed. Not only do they differ in the markets in which they operate, but also in the product bundles they offer. For SuperPoupa's development,

three major competitors have been identified that will be presented in the following section (refer to [Appendix X](#) for an overview of the Competitive Analysis).

Commented [SN2]: Add Appendix

Frugl | The Size Competitor

Frugl is an Australian retail intelligence ecosystem that offers a mobile app for price comparison in the grocery sector. The app allows users to compare prices, adapt their search to dietary requirements, and use a scanning feature for faster product discovery. Frugl is active in the Australian, American, Canadian, and British markets and generates revenue through the sale of consumer behaviour data. Despite its significant presence in these markets, Frugl poses no immediate threat to SuperPoupa due to the geographical distance between the two companies and their different approaches to grocery list building. However, Frugl can serve as a best-practice example for SuperPoupa in market expansion. By examining Frugl's triumphant entry into multiple markets and its strong rankings in the "Shopping Segments" of app stores, SuperPoupa can develop strategies to penetrate the European markets accordingly.

Super Save | The Portuguese Competitor

Super Save is a price comparison app that was established in Portugal. It uses data from six of Portugal's largest supermarket chains and offers basic grocery price comparison, list creation, and product price history. The app is free for users and generates revenue through advertisements from grocery stores. Unlike other price comparison apps that rely on scraping to obtain price information, Super Save obtains its data directly from supermarket chains, which allows it to offer more accurate and up-to-date information. This collaboration with supermarket chains also provides them with an additional advertising channel for their products.

Super Save poses formidable barriers to market entry for SuperPoupa, due to its existing user base and network of supermarket partnerships. However, SuperPoupa's optimization algo-

rhythm and intuitive user experience can give it a competitive edge over Super Save. By promoting these unique features and highlighting the benefits of using SuperPoupa, the company can win significant market share in Portugal. In addition, Super Save can serve as a best-practice example for establishing a presence in the Portuguese market. SuperPoupa can learn from its successes and challenges to develop a successful strategy for growth in this market.

Smhaggle / The Innovative Competitor

Smhaggle is a price comparison application founded in 2007 in Germany. The developer's objective was to enable customers to maximize their savings and create a community of like-minded money-savers who would support one another. As a result, it should not come as a surprise that Smhaggle's features surpass those of previously mentioned apps, as they go beyond merely comparing prices to constructing an application that fulfils a community's needs. Through the receipt uploading feature, Smhaggle integrates users into the value-creation process by updating the most recent price changes. Furthermore, users are incentivized to upload their receipts in exchange for a monetary reward. Strategic partnerships support these cash backs with the platform's retailers. In addition, the behavioural data extracted from customers' receipts are sold and account for a portion of the company's revenue.

If expanded to Portugal, Smhaggle can potentially threaten SuperPoupa because its features distinguish it from other price-comparison applications. For SuperPoupa, these innovative features that, on the one hand, generate revenue and, on the other, foster superior user loyalty through a variety of innovative features, can serve as best-practices for the app's development.

4.2.4.2 Competitive Advantage SuperPoupa

SuperPoupa's competitive analysis reveals that the grocery market is viable for mobile applications with numerous opportunities for success. Despite numerous direct and indirect com-

petitors, opportunities to improve existing solutions and gain a competitive edge still exist. SuperPoupa's competitive advantage lies in its ability to satisfy users' needs for cost savings effectively. This is achieved by combining the simplicity of a grocery list with the effectiveness of price comparison apps, which is further supported by an optimization algorithm. In addition, SuperPoupa's innovative search approach, which employs a subcategory-selection process, is a key differentiator from other competitors. This approach reduces the burden of making product decisions and eliminates the need for broad or non-relevant search results.

In conclusion, SuperPoupa's focus on price optimization and budget management sets it apart from its competitors. SuperPoupa can differentiate itself and carve out a niche in the crowded grocery shopping app market by offering a unique value proposition and targeting a specific user need. Additionally, SuperPoupa can offer users a more personalized and efficient shopping experience by incorporating features such as price histories and store-specific promotions. SuperPoupa needs to keep track of direct and indirect competitors to remain informed of the latest developments in the grocery application market. However, the superior value offered by SuperPoupa in terms of cost and time efficiency provides a positive outlook regarding its competitive advantage.

5 SOLUTION DEVELOPMENT (GROUP PART)

Considering the time constraints of the work project, the goal was to develop SuperPoupa in its most rudimentary yet functional form, otherwise known as MVP. This product has enough features to attract early-adopters and allows developers to collect feedback and validate learnings about customer interaction with minimum effort (Münch et al. 2013). Further, it allows to optimize the solution's core function and enriches it with additional functionalities without spoiling resources down the line (Lenarduzzi and Taibi 2016). This section will lay out

the MVP's entire development process split into two prime sections: back-end and front-end development. Each stage and chapters within will highlight its importance, what best practices were applied and how this was done, any limitations that were encountered, and any other relevant findings. All findings are supported with relevant graphics so the reader can visualize the progress made between each stage.

To develop SuperPoupa, investigating all possible solutions to technical problems and architectural design questions has a crucial initial step. It required taking into account all variables and constraints to meet the development objectives while creating a sustainable and scalable home for SuperPoupa. Best practices of agile development and software engineering were implemented wherever possible to construct the necessary self-build technical components, including a data model foundational to the external database with scraped supermarket data, as well as an Application Programming Interface (API) containing SuperPoupa's "Cheapest Path" algorithm to communicate between the front-end and back-end. The back-end was conceptualized prior to the front-end and thus is presented first.

5.1 Back-end Development

5.1.1 Building a modular back-end using Python (Individual Part: Antonius Struk)

Adherence to best practices and the creation of a structure that would enable agility, iteration, scalability, and sustainability for future development necessitated the construction of a back-end independent of OutSystems, despite the comprehensive capabilities of OutSystems' Service Studio Software to create both front and back-end. The development of the algorithm in Python function by function, step by step, follows the stated goals, as it offers a modular development process. This enabled to quickly test, debug and modify the code instead of wait-

ing for front-end development to create screens to display the outcome of the evolving algorithm. Furthermore, it allowed to quickly pivot and adapt with greater flexibility if the front-end and back-end needed to be developed in tandem. In general, modular development is a recommended practice in software development, as it enables changes to be made without affecting other parts of the software, thereby enhancing development reliability, scalability, and maintainability, which helps to develop at a faster speed (Kleppmann 2017).

Given the algorithm's importance for the project's potential success, it was desirable to develop it as fast as possible to adjust the entire project if its failure became a reality. Fortunately, in September, swift development made it possible to create an algorithm with a tremendous internal assurance that it could find the least expensive route across supermarkets. It is reasonable to propose that this early verification could have been accomplished with something other than OutSystems. The developer's expertise and familiarity with OutSystems were at a rudimentary level, compared with an intermediate python knowledge and an extensive familiarity with additional tools and documentation, which made the challenge of creating the back-end in a short time interval appear challenging but achievable. On the other hand, formulating it in OutSystems implicated a greater risk of failing for reasons mentioned before. Moreover, Python is becoming an increasingly popular option for application development (Stack Overflow 2022). Python is one of the fastest-growing programming languages and also has evidence of significant adoption on GitHub, ranking second among the most popular programming languages in 2021 (Gehring 2022; Krebs and Martinez 2022). The thriving Python community is enhancing every facet of the language. Outstanding documentation helps to grasp new programming concepts faster, and many open-source libraries are being produced to cover various topics, including artificial intelligence, machine learning, and website building (Krebs and Martinez 2022). All in all, Python was the best option for us.

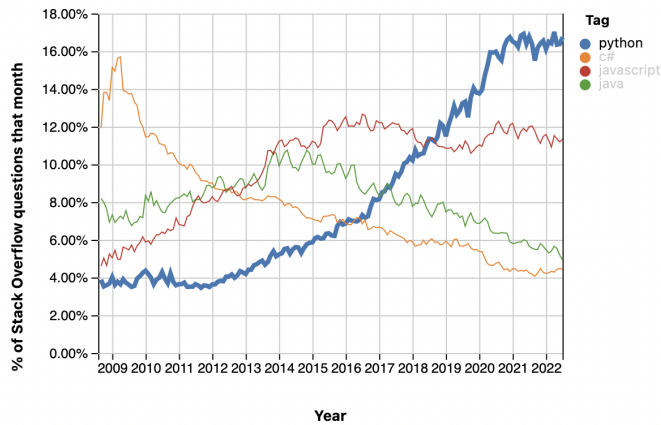


Figure 1 - Percentage of Programming Language Community Questions on Stack Overflow

Sourced from: Krebs and Martinez 2022

Consequently, the modular approach also lowered complexity of development due to subdividing the entire system into more manageable sections that could be separated and, in parallel, solved by using more familiar techniques. It, therefore, enabled clear roles and individual development tasks that could be fulfilled and evaluated without waiting for the progress of front-end components or vice versa. This quickens development speed, enhances individual concentration, and facilitates clearer debugging and testing on each side.

Additionally, the back-end's modular development allows for the potential reuse or alteration of constructed components. This can become beneficial if the concept is later extended into a legitimate business while taking advantage of improved technologies or tools with greater competitive advantages than the status quo solution. Reusing assets such as external databases, algorithms, and APIs will allow greater adaptability and thereby potentially increasing value for both the development of SuperPoupa and the use of it.

Lastly, establishing an external database, coding an algorithm, constructing an API, and scraping data necessitated a more elevated degree of learning due to the limited range of development capabilities within the OutSystems environment. However, the skills and understanding acquired in building the architecture can be applied to other mobile app development projects.

5.1.2 The Architecture Model of SuperPoupa

Reflecting on the choice to develop the core functionality as an algorithm in python provided the clarity needed to complete the architectural design of SuperPoupa. However, to ensure successful operation, an Application Programming Interface (API) was required that facilitates communication and connectivity between the OutSystems front-end and external back-end.

An API is a set of programming instructions and standards for accessing a web-based software application or tool. A REST API is a type of API that follows the Representational State Transfer (REST) software architectural style, which provides a way to access and manipulate web resources through a fixed set of operations. In other words, a REST API is an API that uses HTTP requests to GET, PUT, POST and DELETE data (Gillis 2022). In SuperPoupa's case, the users' grocery list data is exchanged in the JavaScript Object Notation (JSON) format through the API. Simplified: an API connection between OutSystems and the algorithm is a necessary medium for communication and connection based on the architectural design.

An API hosted on a web server has been constructed that ensures that it is publicly accessible and capable of responding to requests sent by OutSystems. Moreover, the algorithm was built into the API code, which runs its operation and queries the external database whenever a POST request is issued on the OutSystems side. Thus, a unique piece of software was created that works as the puzzle between having a list of grocery items a user wants to buy and then seamlessly receiving the cheapest grocery path after pressing the "Submit" button.

After the user presses the "Submit" button, OutSystems will communicate with the API by issuing a POST request containing unique category and subcategory identifiers in JSON format. The algorithm uses these unique data inputs to identify the "cheapest path". The returned result is provided to OutSystems in the same JSON format, which is then read. Its content displays the products purchased from each supermarket to obtain the lowest price possible for groceries. It is important to note that the POST method is the only GET, PUT, POST, and DELETE REST APIs that can receive but also post data (Fielding and Reschke 2014). The user grocery list information comes from the OutSystems side, so OutSystems "posts" the data to the API and will receive the cheapest path in return (Commented Code in Appendix 7, line: 65-75, 375). As depicted in Figure 2 the SuperPoupa architecture had to be conceptualized initially to ensure that the design was achievable. The potentiality of failure was not an option with the limited timeframe.

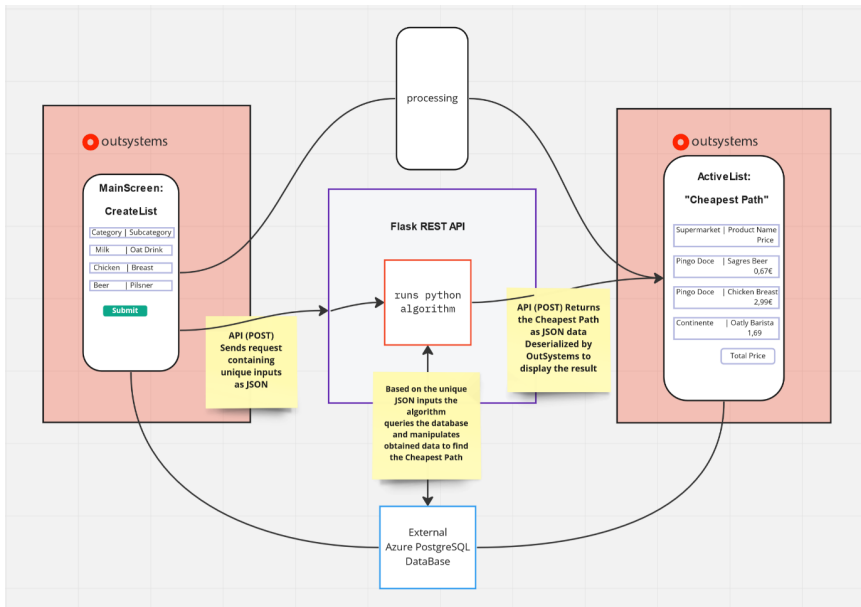


Figure 2 - High Level SuperPoupa System Architecture

It was decided to use an external Cloud Azure PostgreSQL database accessed by the algorithm and integrated into OutSystems. This way, only the required data is communicated via the API instead of the entire data set that the algorithm might require. In addition, the connection of the same database to OutSystems and the algorithm will ensure data reliability and dependable performance. For example, any alterations in the data will be reflected in both software platforms, and only the most necessary data needs to be exchanged between both systems.

5.1.3 From Architecture to Algorithm and Data Model

The initial concept was that the MVP would enable users to pay less quickly to colloquially “save” money for their groceries by computing the most cost-effective route for a given grocery list. This entails calculating the lowest price for each product in a defined primary category (e.g., beer, chicken or milk) and a more specific subcategory (e.g., pilsner beer, chicken

breast or light-skimmed milk). This subcategory specification allows for a larger selection and the comparison of prices for similar products within the same subcategory (e.g. all pilsner beer, all chicken breasts and all light-skimmed milk products).

Comparing the same product across different supermarkets to identify money-saving opportunities has limitations. It needs to consider the potential of opting for a cheaper alternative product of the same product family. The approach considers that money-savers may be more likely to choose the cheaper Sagres pilsner beer instead of the more expensive Super Bock pilsner beer because of both identities in the end as a pilsner beer with (near) identical ingredients.

It is essential to have the main category specified by a subcategory since a user looking for, e.g., pilsner beer may be willing to select from various pilsner beers. However, such flexibility would extend to something other than dark beer, which, while still a type of beer, has a distinct taste from pilsner beer. Therefore, it is beneficial to refine what the user is searching for by having them select a subcategory of the main category to compare the prices of products within that same subcategory (e.g., all pilsner beers or all chicken breast products across supermarkets), as this can ensure a greater degree of certainty regarding the type of product the user is looking for.

The algorithm and its result are called the “cheapest path” because there is a strong likelihood that the lowest price for the products in the same subcategory can vary across supermarkets. This would indicate that a user must stop at more than one supermarket to save the most money. This means that the calculated result most likely contains a grocery list with products from various supermarkets for being the cheapest list possible. For example, from an initial grocery list with 12 products, four could be cheapest at Supermarket A, four at Supermarket B, and four at Supermarket C. However, depending on the prices between supermarkets, multiple

cheapest paths could be chosen with different ratios of products to buy at each supermarket if there are multiple cheapest prices for the same category (more in the algorithm section).

All in all, some parts of the mentioned algorithmic logic evolved gradually. However, the general approach to designing an algorithm that finds the cheapest path based on comparing prices of products of the same subcategory between supermarkets was straightforward. This will be a good use case for the first MVP that will attract recurring users. Therefore, it was needed to conceptualize a data model which would frame and incorporate all the data, relations and integrity constraints needed to successfully develop the end-user price comparison application solution which was aimed for.

5.1.4 The Data Model

A model is often a simplified representation of reality. A data model is an approach to representing a context through structuring data by visualising the relationships and integrity constraints between the entities and their attributes of that specific context (IBM Cloud Education 2021). The development of an accurate data model was fundamental to building the technical solution because it helped to redefine newly abstract grocery shopping through grocery categorising and pricing data and their interacting relationships. Before constructing the data model, database normalisation was identified as a best practice and implemented. In addition, four dimensions were identified which require the design of a- dependable and scalable interacting relation: Category, Subcategory, Supermarkets and Products.

Database normalisation is a technique employed to organise a database into a series of logically related tables to reduce data repetition and improve data accuracy (Elmasri and Navathe 2016). This entails breaking down sizable, unstructured tables into smaller, related ones linked together through primary and foreign keys. Normalisation also involves implementing a series of rules and standards to guarantee that data is stored effectively and consistently

throughout the database (Elmasri and Navathe 2016; Learn Microsoft 2022). Furthermore, the relations between the data in a relational database enable to use the power of Structured Query Language (SQL) to query and combine data as needed, depending on the desired outcome. This can be extremely useful for businesses, as it allows for comprehensive data analysis, data control, and data flexibility (IBM Cloud Education 2021). In addition, normalised tables can be more efficient to query due to their ability only to select and combine pertinent data, leading to improved database performance (Schwartz et al. 2008).

Conversely, a bad example would involve having only one table for all variables, relations and data types, thus making this table the only one in the database and likely causing various issues regarding performance, data redundancy, and data control. The normalised data structure facilitates maintenance and updates with changes needing to be applied in one location if the proper deletion rules are established. Therefore, normalised data stored in relational databases is, for many successful companies the go-to method for structured data organisation and access (Chamberlin 1976; Bourgeois et al. 2019).

Codd (1970, 1971a, 1971b) pioneered the notion of normalization by introducing the first, second, and third normal forms (1NF, 2NF, 3NF). Currently, six normal forms exist, with 1NF being the least normalized and 5NF being the most normalized. While a 6NF degree exists, it is considered to be of academic rather than practical significance. Attaining a higher degree of normalization requires fulfilling the former prerequisites (Kumar and Azad 2017). Meeting the 3NF rules is usually sufficient to ensure the absence of deletion, insertion, and updating inconsistencies (Date 1999). Nevertheless, Elmasri, Ramez, and Navathe (2019) have questioned the practical value of forms higher than 3NF. Higher normalization can lead to reduced system performance due to complex queries and joins needed to assemble and aggregate data, which can take longer to execute. Furthermore, it is often challenging to normalize data to a

higher degree, as the relationships and dependencies between data must be precisely identified. Therefore, 3NF is the most sensible normalization from a human and machine processing perspective, as it offers versatility in data representation and processing while minimizing data duplication and incoherencies (Date 1999; Mathkour 2009). Accordingly, all tables have been normalized and engineered (Figure 3) to a 3NF degree to balance practicality and versatility in data representation and processing with constructability, maintainability, and data integrity. The rules for a 3NF table are as follows (Demba 2013; Codd 1971a):

1. All non-key attributes must be fully dependent on the primary key.
2. All attributes must be directly dependent on the primary key.
3. There should not be any transitive dependencies of non-key attributes.

The utilization of best practices and theories within the software engineering industry supports a strategy of constructing an application correctly from its inception. One such best practice is the implementation of an Entity-Relationship Model (ERM) framework. Peter Chen first released this in 1976 (Chen 1976). A visualization in the style of an ERM, depicting the data model, was utilized during the conceptualization process of the data model and database.

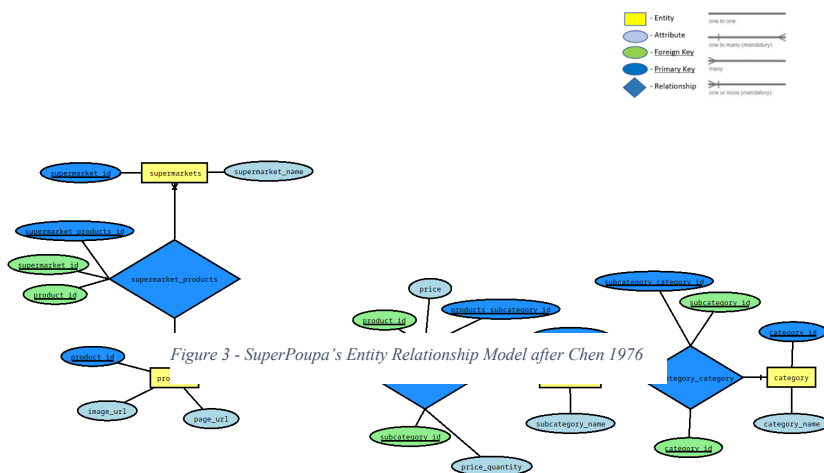


Table 1 presents a comprehensive overview of all tables, their primary keys, foreign keys, and the relationship and integrity constraints that enable an understanding of the ERM and the database model.

	supermarkets	supermarkets_products	products	products_subcategory	subcategory	subcategory_category	category
Primary Key:	supermarket_id	supermarkets_products_id	product_id	products_subcategory_id	subcategory_id	subcategory_category_id	category_id
Foreign Keys:	none	supermarket_id, product_id	none	product_id, subcategory_id	none	subcategory_id, category_id	none
Relations:	none	one supermarket can have many products one product can have many supermarkets	none	one product can belong to only one subcategory one subcategory can have many products	none	one subcategory can belong to only one category one category can have many subcategories	none
Integrity Constraints:	Primary Key cannot be Null	each product in a supermarket must have a price and price_quantity each product_id can only match a supermarket_id once. Primary Key cannot be Null	Primary Key cannot be Null	each product can only belong to one subcategory Primary Key cannot be Null	Primary Key cannot be Null	each subcategory can only belong to one category Primary Key cannot be Null	Primary Key cannot be Null
Multiplicity Constraints	none	Any value recorded in supermarkets_products (table) supermarket_id (or product_id) field must correspond to a valid supermarket_id (product_id) primary key value of the supermarkets table (products table)	none	Any value recorded in products_subcategory (table) subcategory_id (or product_id) field must correspond to a valid subcategory_id (product_id) primary key value of the subcategory table (products table)	none	Any value recorded in subcategory_category (table) category_id (or subcategory_id) field must correspond to a valid category_id (subcategory_id) primary key value of the category table (subcategory table)	none

Table 1 - Data Model Characteristics

The Entity Relationship Diagram (ERD) in Figure 4 was generated using the database administration tool DBeaver. It contains the same model and logic as the ERM, but provides a visualization on a higher, less granular level. It is the visualization that is available on DBeaver tool.

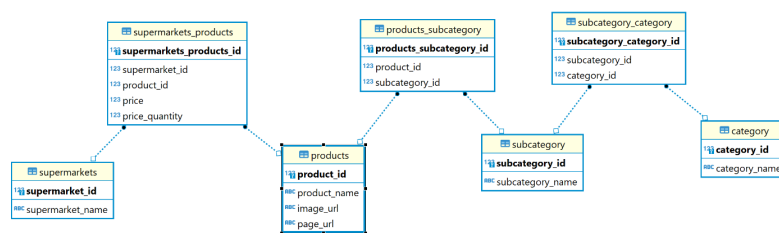


Figure 4 – SuperPoupa's Entity Relationship Diagram in Data Base Management Tool DBeaver

A data model was manually designed on paper to create and construct the algorithm and Excel CSV files were generated to populate the tables with artificial data to form the initial dataset in a relational database structure. Subsequently, the same model logic and the artificial "dummy data" were implemented into OutSystems to support the front-end development. These files provided the data groundwork for initial coding and experimentation in JupyterLab using Python and Panda's library. This process was commenced by writing the "load_data()" function (Appendix). This function was eventually altered to connect to a Post-gresSQL database in the cloud instead of CSV files. The original load_data() function (using CSVs) and the updated version (connecting to PostgreSQL on Microsoft Azure) are both included in the appendix. The initial load_data() function was the first significant milestone in the development of the algorithm. Next, SQL joins are used to join the normalized tables and their data as needed (Appendix , Line: . In the case of SuperPoupa, tables need to be joined to create new tables that possess all the data necessary for the algorithm to function.

5.1.5 The “cheapest path” Algorithm Development

This section presents a detailed exposition of the underlying logic of the "cheapest path" algorithm. A mathematical description of the algorithm is provided to elucidate this, accompanied by visualizations of the core concepts utilized in its development. A thoroughly commented version of the code is included in the appendix, ensuring a comprehensive understanding of the code and the work that went into it. This thesis aims to equip readers who may need to become more familiar with programming with a complete understanding of algorithmic logic.

What is the cheapest path, and how is the algorithm designed to find it? Here, in figure 5, circle A represents the complete set of product and price combinations of all grocery subcategories across all supermarkets of the dataset utilized by the app. In contrast, circle B represents the particular product and price combinations of the grocery subcategories selected by the user using SuperPoupa. The algorithm can determine the most cost-efficient path for the user by analysing the intersection between these two circles.



Figure 5. Sets of Product Price Combinations

The visualization in the Figure illustrates that the user is creating a subset of potential products to purchase by selecting the category (e.g., beer) and subcategory (e.g., pilsner) for each item on their list. The more (less) products the user has on his list, the more (less) categories (i.e., beer, chicken, milk) and subcategories (i.e., pilsner, breast, semi-skimmed milk) he has to select, the bigger (smaller) circle B becomes. This is due to the principles of combinatorics, as each subcategory of a category contains a certain number of products that are part of the same subcategory, thereby creating the potential for combinations of the products within the different selected subcategories (e.g., all pilsner beer products times all chicken breast products times all light-skimmed milk products). Consequently, Circle B provides the data that the

algorithm will use as input to determine the most cost-effective "cheapest" path for the user. The challenge lies in finding the combination of products within subset B that will be the cheapest.

Intuitively, the process of finding the cheapest combination of products across all subcategories would require the calculation of a vast number of possible combinations due to the exponential growth of the number of products as the number of categories that must be considered increases. This process can be illustrated by an example in which a user has nine grocery items to purchase, and it is assumed that each subcategory has an average of ten products (i.e., ten different pilsner beers, ten different chicken breast products). Mathematically, this can be expressed using the formula: $10^9 = 1.000.000.000$. Thus, the number of possible combinations is 1 billion, and calculating the cheapest option for each combination by comparing the total sum in price would be prohibitively time-consuming, if not impossible to compute:

All possible product combinations $(a, b) = a^b$

$a \rightarrow$ Number of products belonging to each selected subcategory (1)

$b \rightarrow$ Number of subcategories selected by user

In this example where $a = \text{ten}$ and $b = \text{nine}$, it means that from nine different subcategories, ten different products in each subcategory could be chosen, which would result in the following:

$$\text{Combinations } (a, b) = 10^5 = 100.000$$

$$\text{Combinations } (a, b) = 10^9 = 1.000.000.000$$

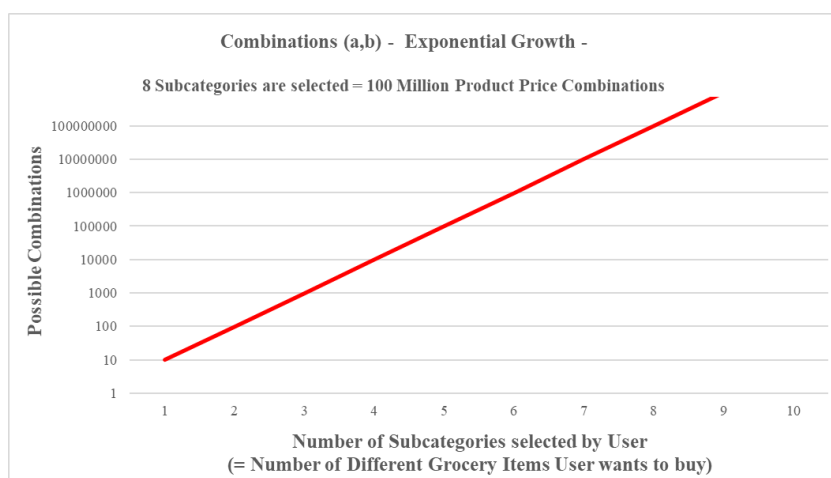


Figure 6 - Exponential Growth of Combinations

It is infeasible to calculate all possible combinations of prices when accurate price data is taken into account, as the exponential growth of potential combinations would be too computationally expensive to be practicable. Moreover, even if such a task were feasible, it would not be scalable, considering the assumption that multiple users would be using the application to seek their cheapest paths. Consequently, to provide an optimal solution while still maintaining good performance, an alternative had to be found to calculate the cheapest path without having to calculate all the possible combinations. To this end, a greedy algorithm-approach was adopted. This approach entails searching for the optimal solution out of multiple locally optimal solutions.

Commented [AVB3]: @Antonius Struk im Text gibt es keine Bezugnahme zu dieser Figure

A greedy algorithm is an algorithmic method that aims to discover a global optimum by selecting the locally optimal option at each stage. It is an iterative problem-solving strategy, which divides the problem into smaller subproblems and solves each in succession. Greedy algorithms are utilised in a variety of problem-solving activities, such as optimisation issues (i.e., shortest path), graph theory (i.e., travelling salespeople problem), and machine learning (i.e., to create decision trees). In general, greedy algorithms are utilised to locate solutions to optimisation and decision-making issues rapidly. They are efficient but only sometimes arrive at the optimal result). Therefore, they often need to be utilised with other algorithms to find the optimal answer (Cormen 2009; Dasgupta 2011; Geng et al. 2011).

A possible solution to finding the cheapest path without calculating every possible combination is to select the locally optimal solution at each stage. This approach involves querying an external database to obtain the user-selected subcategories and related products, which are then organised in ascending order of price per quantity and stored in individual data frames. The minimum price identifies the optimal local option at each stage for each subcategory, marked in green (Tables 2 to 5).

In order to provide a simplified example, the price per quantity of four items is utilised to ascertain the most economical way of purchasing them. The price of each item varies by its amount or weight and thus must be considered when making a comparison. The individual data frames for each product are presented in the following tables.

Sorted Prices (Ascending) of Products per Unique Subcategory					
Supermarket Id	Supermarket	Category	Subcategory	Price/L/Kg	Product Id
1	Pingo Doce	Coffee	Beans	8.99 €	11
3	Continente	Coffee	Beans	8.99 €	11
2	Intermarché	Coffee	Beans	10.00 €	12
1	Pingo Doce	Coffee	Beans	11.50 €	13
3	Continente	Coffee	Beans	13.00 €	14

Table 2 - Sorted Price Variation for Coffee

Sorted Prices (Ascending) of Products per Unique Subcategory					
Supermarket Id	Supermarket	Category	Subcategory	Price/L/Kg	Product Id
1	Pingo Doce	Milk	Light-Skimmed	1.39 €	15
3	Continente	Milk	Light-Skimmed	1.49 €	16
1	Pingo Doce	Milk	Light-Skimmed	1.59 €	17
3	Continente	Milk	Light-Skimmed	1.69 €	18
1	Pingo Doce	Milk	Light-Skimmed	1.79 €	19

Table 3 - Sorted Price Variation for Milk

Sorted Prices (Ascending) of Products per Unique Subcategory					
Supermarket Id	Supermarket	Category	Subcategory	Price/L/Kg	Product Id
3	Continente	Chicken	Breast	10.00 €	1
3	Continente	Chicken	Breast	11.00 €	2
1	Pingo Doce	Chicken	Breast	12.00 €	3
1	Pingo Doce	Chicken	Breast	12.50 €	4
2	Intermarché	Chicken	Breast	14.00 €	5

Table 5 - Sorted Price Variation for Chicken

Sorted Prices (Ascending) of Products per Unique Subcategory					
Supermarket Id	Supermarket	Category	Subcategory	Price/L/Kg	Product Id
1	Pingo Doce	Beer	Pilsner	1.50 €	6
2	Intermarché	Beer	Pilsner	1.50 €	7
3	Continente	Beer	Pilsner	1.79 €	8
2	Intermarché	Beer	Pilsner	1.89 €	9
3	Continente	Beer	Pilsner	1.99 €	10

Table 4 - Sorted Price Variation for Beer

Upon examining the tables, it is evident that multiple products belonging to the same subcategory may share the lowest price. This means that the algorithm considers these prices to be a minimum. For instance, Coffee Beans and Pilsner Beer have the lowest prices at different supermarkets. In contrast, Chicken Breast and Light-Skimmed Milk only have one minimum pricing. Table 6 displays the minimal prices for each subcategory, and it can be seen that Pilsner Beer and Coffee Beans each have the two lowest prices at disparate supermarkets.

All Cheapest (Minimum) Products per Unique Subcategory across Supermarkets					
Supermarket Id	Supermarket	Category	Subcategory	Price/L/Kg	Product Id
3	Continente	Chicken	Breast	10.00 €	1
1	Pingo Doce	Beer	Pilsner	1.50 €	6
2	Intermarché	Beer	Pilsner	1.50 €	7
1	Pingo Doce	Milk	Light-Skimmed	1.39 €	15
1	Pingo Doce	Coffee	Beans	8.99 €	11
3	Continente	Coffee	Beans	8.99 €	11

Table 7 6 - Cheapest (Minimum) Products per Unique Subcategory Across Supermarkets

All Cheapest (Minimum) Products per Unique Subcategory across Supermarkets				
	Chicken Breast	Pilsner Beer	Light-Skimmed Milk	Coffee Beans
Number of Cheapest Products	1	2	1	2
Unique Supermarket Id's Count	1	2	1	2
Supermarket Id	3 (=Continente)	1 (=Pingo Doce) & 2 (=Intermarché)	1 (=Pingo Doce)	1 (=Pingo Doce) & 3 (=Continente)

Table 8 7 - Minimal Price Comparison for Chicken, Beer, Milk, and Coffee

The number of possible cheapest paths can be determined by examining the number of minimum prices per subcategory and their respective supermarket identifiers. Table 7 can be mathematically expressed as follows:

Number equations

Corresponding Supermarket Id's of Number of Cheapest Products:

$$= [[3], [1, 2], [1], [1, 3]]$$

$$S_i = \text{unique subcategory}$$

$$M_i = \text{number of cheapest products of subcategory } S_i$$

Possible Cheapest Path Combinations (PCPC)

$$= (S_1 * M_1) \times (S_2 * M_2) \times (S_3 * M_3) \times \dots$$

$$PCPC = \prod S_i \times M_i$$

$$\text{For this example case: } (1 * 1) \times (1 * 2) \times (1 * 1) \times (1 * 2) = 4 \text{ PCPC}$$

By substituting the lowest prices for each subcategory with their respective supermarket identifier, all possible PCPCs and the involved supermarkets can be identified. Iteration over all unique combinations of the supermarket identifiers is then conducted, and the following path combinations from equation X are yielded: [3, 1, 1, 1]; [3, 1, 1, 3]; [3, 2, 1, 1]; [3, 2, 1, 3]. Table 8 demonstrates the four PCPCs generated from the given user example.

All Possible Cheapest Paths				
	Chicken Breast at Supermarket	Pilsner Beer at Supermarket	Light-Skimmed Milk at Supermarket	Coffee Beans at Supermarket
Cheapest Path 1	Continente (Id=3)	Pingo Doce (Id=1)	Pingo Doce (Id=1)	Pingo Doce (Id=1)
Cheapest Path 2	Continente (Id=3)	Pingo Doce (Id=1)	Pingo Doce (Id=1)	Continente (Id=3)
Cheapest Path 3	Continente (Id=3)	Intermarché (Id=2)	Pingo Doce (Id=1)	Pingo Doce (Id=1)
Cheapest Path 4	Continente (Id=3)	Intermarché (Id=2)	Pingo Doce (Id=1)	Continente (Id=3)

Table 8 - All Possible Cheapest Paths

It has been established that all combinations of the lowest prices achieve the lowest total price for the given grocery list. Four options have been identified which are equally cost-effective. An efficient shortcut is selecting the locally optimal solution to obtain the cheapest route. This is because filtering for the least expensive route is more efficient than calculating all possible combinations. As such, it can quickly become unmanageable in number (as demonstrated in Figure X) yet would have yielded the same result had it been computationally viable.

Commented [AVB4]: Referring to what?

The primary difference between the four paths lies in the ratio of products acquired at each Supermarket (i.e., Pingo Doce, Continente or Intermarché). Therefore, the next important question is, which cheapest path out of the four is optimal for the customer?

An algorithmic rule was implemented in OutSystems wherein the cheapest path, consisting of the least number of supermarkets, is chosen for a display to the user. This rule was based on the judgement that it is desirable for the user to acquire the cheapest products with minimal effort and time spent (i.e., by commuting to supermarkets), thus obtaining the best user value.

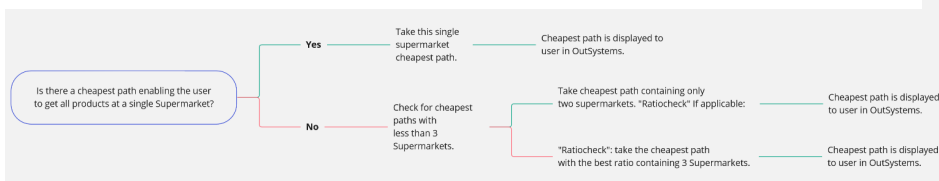


Figure 7 - Algorithmic Decision Tree for Choosing Supermarkets

The cheapest path must consider all three supermarkets, so a "ratio check" is conducted. As the technical limitation of providing the user with the option to only get the lowest prices for one or two chosen supermarkets by default exists, the algorithm considers all three supermarkets. The path that allows most products to be bought at one supermarket is selected to mitigate this limitation. For instance, if two cheapest paths exist for a grocery list of 15 products - Cheapest Path 1 requires 13 products to be bought at Continente, one at Pingo Doce and one at Intermarché - and Cheapest Path 2 requires five products each at Continente, Pingo Doce and Intermarché, the algorithm will always choose Cheapest Path 1. This allows the user the greatest flexibility to purchase most of their items from one place before assessing if the prices of the remaining items from the other supermarkets are worth the commute and time or if it is acceptable to buy them at Continente too.

The user can optimize their route if they are commuting to purchase items by following a hierarchical principle: going to the single-product supermarkets first (1 product at Pingo Doce, 1 product at Intermarché). This is more convenient if they travel by foot or bike as they only

need to buy one item at each store. Even if the user travels by car, this logic makes sense if they buy frozen or heat-sensitive items, as such items should be bought last (13 products at Continnente). Thus, it is most efficient and effective to select a path where most products can be found at one supermarket, in the best case, all of them.

By examining the example of Chicken Breast, Pilsner Beer, and so on, path 1 (Table ...) has been chosen to display the cheapest route. This is beneficial due to the hierarchical principle of selecting the route requiring the smallest number of supermarket visits. The cheapest route will be output in a JSON format and transmitted via API to the OutSystems platform. A logic process will be carried out to reveal the items that should be purchased at each supermarket to minimize the overall cost for the user.

5.1.6 Flask REST API (Antonius Struk)

A concise definition of Application Programming Interface (API) was provided in the section (Architecture Model) with the conclusion that an API is crucial for facilitating communication between the front-end and back-end architecture. This section further elaborates on the process and challenges of implementing an API in such a system.

An initial assessment of the available API frameworks was conducted to identify the most suitable option for this project. Due to the time constraints, it was necessary to quickly acquire the technical knowledge required to work with and build APIs, which needed to be covered in the curriculums of Business Analytics or Impact & Entrepreneurship. Since Python is and stays the chosen language, the evaluation identified Django, Flask, and FastAPI as the three most viable and popular options.

A decision had to be made between the full-stack Django framework and the micro-framework Flask, considering the needed complexity and performance of the web application. Django offers an extensive feature set but can be cumbersome to customize and has a large

codebase which can be challenging to debug and maintain. Furthermore, handling the same number of requests requires more computing power and memory. On the other hand, Flask was designed with scalability and simplicity and is particularly suitable for creating smaller web applications. Furthermore, it provides a lightweight framework while still giving the developer a great degree of freedom and flexibility (Vincent 2017; Krebs and Martinez 2022; Welcome to Flask 2022; Django Project 2022).

Given the considerations outlined above, Flask was the most advantageous option available. This was affirmed by the rapid implementation of a test API to assess the communication between OutSystems and the API itself, facilitated by the straightforward micro-framework provided by Flask. Furthermore, it allowed to use only a few lines of code to build an API. The appendix (Figure X) in this document contains an illustration of the code for the test API, which was employed to facilitate the exchange of the classic "Hello World" message between OutSystems, and the API being studied.

The objective of this project was to make progress as quickly as possible. Thus, the decision was made to employ the services of NGROK (<https://ngrok.com/>) to tunnel the local server of the private computer to a public server, providing a public address that OutSystems or Reqbin could easily access. In addition, Reqbin (<https://reqbin.com/>) was also utilized as an online API testing tool to test the API using JSON inputs easily. However, despite Ngrok being a viable solution to test the API and verify its functionality quickly, this approach is unsuitable for hosting a web app in the long run, as it relies upon a single computer or server and has no scalability to handle a large number of users or requests. Additionally, the web app would be vulnerable to security threats and other risks. Therefore, it is necessary to host the API on a cloud platform such as Amazon Web Services, Microsoft Azure, or Google Cloud Platform, which provide managed services and support for deploying and managing web

apps, as well as scalability, reliability, and cost-efficiencies to ensure a professional deployment of the mobile application (Amazon Web Services 2022; Microsoft Azure 2022). This will be the case until the public defence of this paper.

Integrating the algorithm into the API was a substantial challenge, necessitating extensive experimentation to address the various errors resulting from combining modular pieces of code. Assistance from StackOverflow and Reqbin enabled a JSON input of the user's grocery list to be received and a JSON with the cheapest path to be returned. This project was worthy of celebration, as it marked the milestone of end-to-end communication between the front-end and back-end. This demonstrated that the technical architecture was working, and the finish line was reached. Figure X in the appendix shows the uncommented API/Algorithm code, while Figure XI contains the same code with comments. A screenshot of a Post Request Jupyter is also included.

5.1.7 External Database – Implementing Azure PostgreSQL (Group Part)

Implementing an external database into the architecture model necessitated separate access to the data stored by the external API/Algorithm. This allows only the users' grocery list data (category and subcategories IDs) to be sent as JSON inputs to the algorithm, as opposed to the entire data set related to the grocery list. Sending a large amount of data would decrease performance and scalability significantly, making it an impractical choice. Additionally, the data is kept up to date by having a single database connected to both the front- and back end. This is a critical factor, as two different databases would be vulnerable to not being synchronized, resulting in data inaccuracies providing inaccurate results to the user.

PostgreSQL, a free and open-source relational database management system (RDBMS), was selected for this project. This RDBMS is utilized as a primary database by notable companies such as Instagram, Reddit and Skype (Wikipedia 2022). Hosting this RDBMS on Azure

Microsoft Cloud Services presented numerous benefits for a mobile business, including increased flexibility and scalability, cost savings, improved security, increased availability, and reduced administration. These advantages are attributable to the database being hosted in a secure data centre, allowing for more efficient scaling and easier administration while decreasing hardware and software costs. Additionally, the external cloud database can offer increased security and availability due to its redundant hosting (Gibson et al. 2012; Salesforce 2022).

5.1.8 Data Acquisition of Portuguese Supermarket Data (Antonius Struk)

Initially, dummy data (Appendix, Table) was the quickest and most convenient to focus on the development of the algorithm. However, it was evident that real supermarket data for real users would be necessary to generate a meaningful impact and an actual product for daily life. Consequently, consideration was given to how actual supermarket data could be obtained. Two potential strategies for obtaining the desired data were identified: engaging in business-to-business negotiations with supermarkets to access their data; and automatically extracting ("scraping") price and product data from the supermarkets' online stores.

Price monitoring, market research, and sentiment analysis are three of the most widely employed applications of web scraping (Khder 2021). In SuperPoupa's case, the accuracy of grocery prices is of utmost importance. Additionally, grocery items' costs can fluctuate daily (Aparicio et al. 2021). Manual tracking of prices is not a practical option when such frequent variations occur, especially when dealing with thousands of products from different supermarkets. Web scraping solves this problem by enabling the timely extraction of price and product data from supermarkets, which can then be stored in a database.

However, the best option for SuperPoupa will be to pursue cooperation with supermarkets, as this will render technology that must be timely engineered for constant data scraping and integration unnecessary. Instead, data acquisition efforts could be focused on integrating

data into a Relational Database Management System (RDBMS). Despite attempts from the Business Development Team to establish a line of communication, success has yet to be achieved. With a working MVP to showcase or a user base to demonstrate, the ability to negotiate is allowed. To gain further insight into this issue, a conversation was initiated with the Founder of Super Save (Google Play 2022), which provides a tool to compare grocery prices and can be seen as a competitor with a different User Experience (UX) and Unique Selling Proposition (USP). The founder indicated that cooperation was established with significant supermarkets (e.g., Aldi, Continente, Pingo) to acquire updated data daily without needing to scrape it, as was done initially. As a result, it was decided to scrape data from the respective online shops to engage in data negotiations with supermarkets at a point where a good user base and a functioning product could be demonstrated.

Analysis of Portuguese grocery retailers revealed that Pingo Doce, Continente and Intermarché offer full online-shop availability, including ordering groceries. However, Aldi and Lidl did not provide this service and were excluded from scraping attempts. It is believed that Aldi and Lidl, as discount retailers, could benefit from a solution like SuperPoupa's due to their competitive pricing relative to supermarkets such as Continente, Pingo Doce, and Intermarché. Given Aldi's and Lidl's unavailability of an online supermarket, the focus is currently directed towards Pingo Doce, Continente and Intermarché. The prevalence and accessibility of 518 Continentes, 460 Pingo Doces, and approximately 200 Intermarchés throughout Portugal (Wikipedia 2022; Jerónimo Martins 2022) provide good options for potential users.

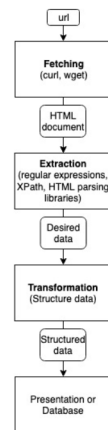


Figure 8 - Web Scraping Process (Persson 2019)

There are many different techniques to scrape desired data off the internet into a single database or spreadsheet for further use (i.e., HTTP Programming, HTML Parsing, DOM parsing, Web Scraping Software, and Customized Python "Scrapers") (Khder 2021). However, the process itself stays technically the same: The web scraping process consists of three stages: Fetching, Extraction, and Transformation. During the Fetching stage, the target website is accessed using the HTTP protocol, a common Internet request and response system also used by web browsers. Libraries such as curl and wget are then used to send a GET request to the URL and receive the HTML page. Following this, the Extraction stage uses tools such as regular expressions, HTML parsing libraries, and XPath queries to identify and extract the relevant data. Lastly, the Transformation stage converts the data into a structured format for presentation or storage (Persson 2019; Khder 2021).

Utilizing ParseHub Web Scraping Software Tool (<https://www.parsehub.com/>) was deemed the most efficient approach to obtaining data from the chosen supermarkets due to the

Commented [AVB5]: @Antonius Struk also not referred to in the text

unique structure of each website, which could result in coding challenges that might require an unpredictable amount of time to solve. Despite this method posing its difficulties, video calls with ParseHub employees facilitated the process, and screenshots of the scraping designs are provided in the appendix.

Scraping of product information from three major Portuguese supermarkets was conducted to obtain the essential features: Page URL, Product Name, Price, Price Quantity per Kilogram or Liter, and the product image URL. In particular, the Page URL was beneficial for product categorization, as it indicated the product's category, which is integral to SuperPoupa's unique selling point. Other features available at some supermarkets included Brand of product and Units per product. A total of 56,829 unique products were collected from Intermarché (12,096), Pingo Doce (13,876) and Continente (30,857). However, subcategorization was done manually, limiting the number of products used in the dataset. An advanced machine-learning model would need to be constructed to make the process scalable to any data.

5.1.9 Limitations and Missing Technology (Group Part)

The current MVP is limited in two major ways. Firstly, the scraping and integration of product data are not automated nor done daily or weekly, resulting in price data not being up to date, as well as the lack of integrating weekly specials and discounts. The data could be obtained from supermarkets, as Super Save does. To provide users with accurate data, creating automated scrapers for daily updates on product prices will be beneficial. Having the **technology** stack for automated scraping together with a machine learning model for sub-and categorization is promising to create monetizable value not only for a smart grocery app. The concept's applicability demonstrated in this study may be transferrable to various physical stores that sell products, such as hardware stores, where the easy filtering and price comparison of products according to the customer's needs and budget provides monetizable value (delete, move?).

The second major limitation is the ability to automate the sub- and categorization of the entire product catalogue across supermarkets. This can be achieved using a product classifier based on an automated machine learning model. Currently, only a subset of the scraped dataset has been subcategorized manually. An accurate automated classifier would provide the desired solution as initially intended. There is great potential in providing a solution that allows consumers to search for the most economical, ecologically conscious or cost/taste-effective product within a subcategory by clicking a few buttons. However, automated scrapers and a machine learning model that can classify products must be employed if the necessary data can be obtained directly.

Given the limited time frame, it was not feasible to create automated web scrapers as the data would have still required meticulous manual cleaning and categorization. Furthermore, constructing a machine learning model that accurately sorts products into sub- and categories would be worth a work project. Investment in hiring Data Engineers will be essential to create the necessary technology,

The automatic classification of product data using modern technology is a currently discussed subject in research, given the challenge of vast amounts of products that outlets such as Walmart or eBay have to process and classify regularly. Various research approaches have tackled this, such as natural language processing techniques for parsing text titles, machine learning classifiers for selecting product categories, and a method for automatically categorizing a product in an electronic marketplace (Baeza-Yates and Ribeiro-Neto 2011; Gupta et al. 2016; Zhong et al. 2020). Hafez et al. (2021) conclude that traditional machine learning algorithms like Fuzzy K-Nearest Neighbors (FKNN) perform better than deep neural networks and a score-based ranking method. However, the score-based approach is a close second. Applying the latest research, testing and ultimately building a model which can automatically categorize

and subcategorize products accurately will be essential for the potential future of SuperPoupa. Technology and know-how are available, but data engineers and scientists are costly.

5.1.9.1 Feature Engineering (Pepjin van Es)

Parsehub's scraper tool allowed to extract product information on 56.786 items rather quickly and accurately from Pingo Doce, Intermarché, and Continente. However, the data was still raw and thus unusable, meaning it had to be cleaned and transformed. This process, otherwise known as feature engineering, refers to the manipulation – addition, deletion, combination, and mutation – of data to enhance a machine-learning model (Domino 2022). The algorithm then uses the resulting dataset containing these artificial features to produce the most accurate models and valuable insights (Patel 2021). As such, this process (figure 9) provides a more profound and practical understanding of the data concerning the business problem (Ibid.).

Before manipulating the data, it was necessary to zoom out and consider the following: First, what is the problem that consumers can be helped at solving by using the data? Second, what features ought to be engineered approaches for this to happen as effectively as possible? Last, what medium allows consumers access to the data, and how?

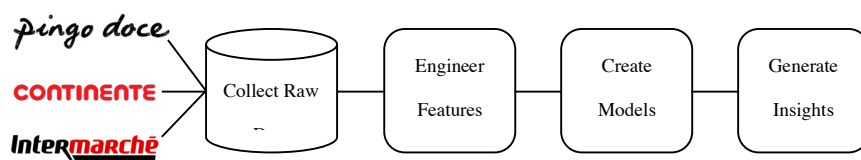


Figure 9 - Feature Engineering Process

Looking at the problem, many Portuguese consumers are adjusting to new grocery shopping habits, of which value-for-money deals have become integral (Bazzoni et al. 2022). However, empirical data shows that there is yet to be a practical solution on the market that allows them to. Therefore, the collected data should assist consumers in finding the best value-for-

money deals at either of the three mentioned stores as effectively as possible. Considering how labour-intensive feature engineering the entire dataset is, together with the timeframe of the work project, it would have been unfeasible to include all scraped data into the resulting dataset to be used by the algorithm. Therefore, having analysed the data it has been decided on the MVP's twenty-five categories (table 9) primarily food related. The assumption was made that this was the product range within these categories that was substantial enough to provide value to the end user yet manageable in terms of cleaning.

Product Categories				
Banana	Potato	Paprika	Toilet Paper	Butter
Apple	Beer	Courgette	Tomato	Olive Oil
Milk	Carrot	Broccoli	Kiwi	Chicken
Peanut Butter	Rice	Mushroom	Onion	Beef
Egg	Cucumber	Water	Cauliflower	Lemon

Table 9 - Chosen Categories

5.1.9.2 Grouping

That said, considering each supermarket mainly had unique product categories, the overlap between them was negligible. Therefore, the first step was to create the categories through an engineering method called feature grouping, which, as the name states, creates a more straightforward model by grouping variables (Phoenix 2022). For instance, having picked a predetermined category, beer, for demonstration purposes, a comparison was drawn across the three supermarkets. As such, it could be assessed which of their original product categories and the products within would fit under this each newly constructed product category. In some cases, categories had to be split because of some products' unsuitability within the intended category.

Supermarket	Original Categories	Items	Action	Category	Items
Pingo Doce	National Beer	52	Merge	Beer	189
Intermarché	Beer with Alcohol	86	Split and Merge		

Intermarché	Beer without Alcohol	12	Merge		
Continente	Artesenal Beer	80	Split and Merge		
Continente	Beer without Alcohol	14	Merge		
Continente	Traditional Beer	76	Split and Merge		
Continente	Beermarket	9	Split and Merge		

Table 10 - Grouping Supermarkets' Original Categories

5.1.9.3 Splitting

In this step, the consideration of the medium through which the users access the data – the product search tool developed through OutSystems – became essential. At this point, “Beer” contained a range of items that had to be individually assigned to new subcategories through a feature-splitting process. The usage of subcategories as the highest level of customization rested on the assumption that, as branding loses significance, consumers would opt for the cheapest item within a subcategory. However, this had to be done thoughtfully because, considering the mechanics of the application’s search engine, these subcategories dictate how users create their shopping lists. With that in mind, each category had to balance detail in that they allowed users to find exactly the products they needed and ease of use so that their level of detail did not hinder the ease with which users could create shopping lists.

Category	Items	Subcategory	Items
Beer	189	Lager	106
		Stout	15
		Indian Pale Ale	7
		Dark Beer	11
		White Beer	4
		Lager without Gluten	2
		Bohemian Beer	10
		Abadia Beer	5
		Radler	4
		Malt Beer	3
		Lager without Alcohol	15
		Dark Beer without Alcohol	4
		Radler without Alcohol	3

Table 11 - Splitting Constructed Categories

5.1.9.4 Standardization

Considering products' price-quality ratio became a central theme in consumer choices, a feature that accurately displayed this information was crucial both from a consumer and development perspective. This feature would tell the user what they would be paying for their product and, to some extent, would thereby justify the choices made by the algorithm. At the same time, the algorithm needed a measurement on which to base these choices, in other words, to identify which product is truly the cheapest. Unfortunately, when the data was scraped, some items displayed price per unit, others per kilo, and some contained a mix of these and more. standardized into a price per kilo and litre.

Product Name	Price per Quantity	Uniform Feature
Veal Steak	0,2 kg ~ 1 un 10,49 a/kg	10,49
Veal Round Steak	0,25 kg ~ 1 un 10,49 a/kg	10,49
Veal Round Beefsteak	0,6 kg ~ 1 un 11,98 a/kg	11,98
Veal Steak for Stews	1 kg ~ 1 un 6,99 a/kg	6,99
Veal Chops	0,3 kg ~ 1 un 10,49 a/kg	10,49
Tenderloin of Veal	1 kg ~ 1 un 19,99 a/kg	19,99
Bone-in flap	5,47 a/kg	5,47
Bone-in flap	5,99 a/kg	5,99

un = Unit; a = Per; kg = Kilogram

Table Table 12 - Standardizing price features

5.1.9.5 Categorical Encoding

Finally, now that the price data was standardized, the categorical data linked to it – including the product category, subcategory, and product name – had to be encoded into numerical values. This had to be done because one cannot pass nominal data onto models since machine learning algorithms require that input and output variables are numbers (Verma 2021).

Therefore, for the algorithm to provide the user with the correct permutations, each categorical item was given a unique identification number. However, the categorical data had to be preserved as they were still relevant from a user perspective. Evidently, using only numerical data in an application would negatively affect the user experience.

Category	Identification Number	Subcategory	Identification Number
Apples	2	Fuji	4
		Granny smith	5
		Reinata Parda	6
		Starking	7
		Golden	8
		Royal Gala	9
		Bravo Esmolfe	10

Table 13 - Encoding Nominal Data

5.2 Front-End Development (Group Part)

In alignment with the project objectives, identified user needs and technical requirements, as well as the back-end architecture and data model, the next step was to translate these elements into an end product by building a user-centric front-end. Front-end development aims to design and construct the user interfaces of applications and webpages in a clear, simple, and fast-to-use manner (Codesido 2009). The front-end of an app combines all design elements with the functionalities of the app and allows the user to interact with it. It is the layer of design that is visible to the user and, therefore, a critical factor in creating a positive user experience.

5.2.1 Technical Landscape for Building Mobile Apps

Traditionally, the development of an application's front-end traditionally materializes through programming languages such as Java, a popular choice for Android OS, and Objective-C or Swift, frequently used for iOS (Fireart 2022). However, the methods used to construct applications have been limited to those with the skills to write computer code. Luckily, this has

seen a change since so-called low-code and no-code (LCNC) solutions gained widespread popularity by lowering the barriers to who can develop an application. For SuperPoupa, use of LCNC technologies was made by working with the OutSystems development platform. Hereby, a functioning app could be built in less than 100 days by a team of non-developers.

5.2.1.1 LCNC Technologies

LCNC is a software development approach that allows developers and non-developers, otherwise known as citizen developers, to create software applications with little to no traditional programming. This technology lets users quickly build and deploy sophisticated applications by offering graphical user interfaces (GUI) instead of requiring classical coding interfaces. Next to the GUI, LCNC tools also consist of a library of pre-built components, screen templates and tools that allow users to drag and drop components to build screens and complex functionalities. The first versions of this technology can be dated back to Apple's HyperCard in 1998 or Microsoft's Visual Basic in 1991 (Tozzi 2021; IvyPanda 2022). Since then, various tools have been released for enterprise and personal use.

Albeit simplifying the development process, low-code solutions still require some knowledge about traditional programming compared to purely no-code solutions (Tariq 2021). However, with more and more powerful solutions being offered, boundaries between these two are blurred, which is why they are often referred to as Low-Code/No-Code Solutions (Hurlburt 2021). Their widespread adoption of this technology enables organizations to greatly accelerate the development process, quickly building, deploying, and maintaining applications while reducing deployment and maintenance expenses (Elshan, Dickhaut, and Ebel 2023). A 2022 Forrester report states that large organizations can benefit from a return on investment of 506% within six months by implementing OutSystems (Forrester 2022).

5.2.1.2 The OutSystems Development Platform

OutSystems is a low-code development platform that enables developers to create, manage and deploy enterprise web- and mobile applications using a graphical user interface. OutSystems offers five main components: The platform server, Integration Studio, Service Center, LifeTime, and Service Studio. The OutSystems Platform Server is a multi-tier server architecture that provides the infrastructure for developing and running the applications and services built within OutSystems (OutSystems n.d. a). Integration Studio extends OutSystems' functionalities by offering to connect with various external tools and software solutions. This was used this to integrate the external database. Service Center is a platform server management application that helps monitor and administrates an app's server. For this project, Service Center was used to monitor the communication protocols between OutSystems and the Flask API. While Service Center manages a single environment with its corresponding modules, roles, and settings, LifeTime is not environment specific. LifeTime helps manage several environments, security settings, and infrastructure on a higher level (OutSystems 2020). Lastly, Service Studio is the visual development environment in which the development and deployment of OutSystems Apps happen. Service Studio offers a powerful and easy-to-use GUI with all the features needed to build the front-end of SuperPoupa, including complex workflows.

5.2.1.3 Traditional vs. LCNC Front-End Development

Low-code front-end development and traditional front-end development possess the same objective of creating a seamless and user-centric experience. However, the methodology utilized in each case varies significantly. Low-code systems typically include an already existing infrastructure. In Service Studio's case, this infrastructure comprises the Processes, Interface, Logic, and Data tabs within the GUI. Additionally, Service Studio provides the capability to implement modular blocks that can be effortlessly inserted and employed across multiple

screens. Moreover, the visual elements, including screens, blocks, and widgets that, in sum, make up the interfaces, are organized into hierarchical widget trees. This allows developers to find the respective elements when needed quickly. Within the four primary tabs, elements are already organized into corresponding folders in which new elements can be added. This improves the speed at which components can be found and relationships understood. Furthermore, design aspects such as global styles for buttons, colours, backgrounds, or fonts are also already centrally organized by OutSystems enabling users to make changes that affect the entire application without having to adjust single elements. In traditional front-end development, the developing team must set up all these structures and relations from scratch. The bigger an application, the more complex the code becomes and the more critical a proper structure is. This not only takes time but is much more prone to errors.

Service Studio facilitates accelerated development, testing, and debugging – the process of identifying and rectifying errors in the application. To prevent bugs from happening in the first place, OutSystems constantly runs error checks in the background while using building components and logics in the GUI. These errors can include parsing errors, data type errors, missing values, etc. All these are errors that, in some way or another, disturb the general logic of the application. Therefore, if such errors occur, OutSystems notes them within the error console and does not allow the app to be published unless the errors have been removed. This dramatically reduces the risk of publishing a faulty app. With the 1-click publish process of OutSystems, apps can be continuously and rapidly tested even after minor changes have taken effect. In the testing environment, all features of the app that are functional on a web or mobile browser can be tested. This helps to find design- or layout discrepancies and functional errors quickly. If a flaw in the execution of the features, otherwise known as a bug, has been encountered, users can pin down the problem using the debugger. This tool provides an interactive,

step-by-step debugging console that allows developers to trace, analyse, and debug an application's action flows. By allowing developers to check the values of variables and view the results of all operations, the root of bugs can be found at an increased speed.

5.2.1.4 Limitations of the Front-End Development with OutSystems

With all this being said, OutSystems may not be suitable for all users or situations. Despite the presence of a well-developed framework, Service Studio remains challenging to comprehend due to its intricate nature, necessitating a significant amount of time to become familiar with the system. OutSystems is aimed at enterprises and IT professionals rather than non-developers, meaning that a comprehensive knowledge of development terms and concepts such as variables, loops, attributes, events, actions, relational databases, CSS styles and HTML structures is necessary to use the low-code platform.

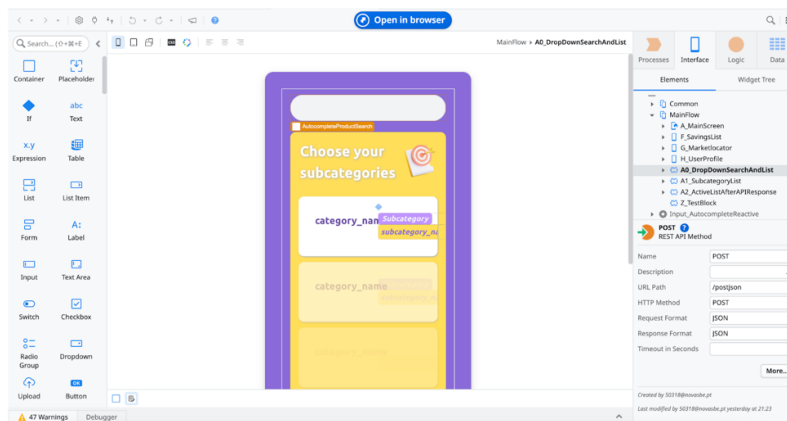


Figure 10 - Service Studio Graphical User Interface

Furthermore, many learning resources and feature documentation requires some understanding of development. This strategy by OutSystems also reflects on their pricing structure. While popular no-code apps such as Bubble (Bubble n.d.) or Flutterflow (FlutterFlow n.d.) start at 25\$/month and 30\$/month, respectively, for a paid plan, OutSystems' cheapest paid plan

starts at over 1,500\$/month (OutSystems n.d. b). These considerations must be made by individuals or smaller companies when thinking about developing with OutSystems. Luckily, Nova SBE allowed to build SuperPoupa using the university environment within OutSystems.

5.2.2 User Experience (UX) Design (Pepijn Van Es)

Following best practices in line with the MVP's general philosophy, instead of diving headfirst into developing the front-end using OutSystems, the groundwork was laid for the application's user interface using what is known as wireframes. Wireframe modelling allows developers to demonstrate to end-users and, most of all, team members how the application will look and feel using rudimentary shapes and lines (LucidChart 2019). Nonetheless, it is one of the design processes' most crucial steps. In essence, it creates the blueprint for the user interface (UI) structure, layout, and functions to be developed in high-fidelity in later stages (Hamm 2018). It does so by considering for each screen what elements will be displayed in the user

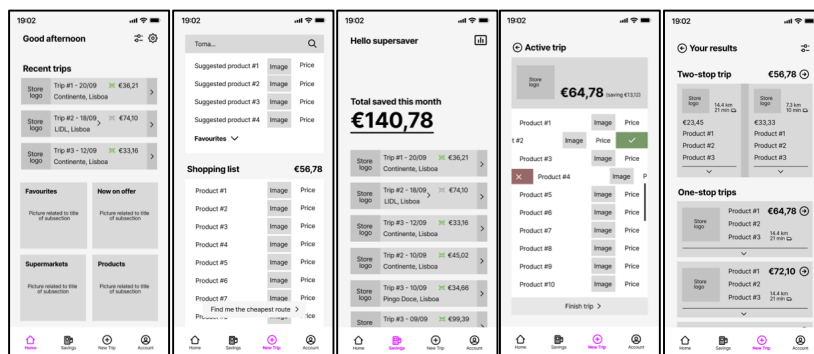


Figure 11 - UX wireframes made in Figma

interface, where those elements are placed on the screen, and how the user will interact with them (Lucid-Chart 2019). The right user experience paired with a good user interface is key, as this interaction, apart from the technical function, will mainly dictate the value customers identify in a product (Georgieva 2022). The platform used to design the UX wireframes was Figma,

a web-based prototype design tool. Figma is easy to use and allows for effective collaboration between designers. As the software is web browser-based, anyone with permission can see the designer at work in real-time (Danielyan 2021). This meant that designing the wireframes and translating them into high-fidelity structures in OutSystems could be done simultaneously. Also, by identifying design flaws early in the process, having to reverse-engineer complex structures down the line could be avoided. On the other hand, finishing the UX wireframes was a pivotal moment in the development process. The wireframes designed can be seen in figure 11.

5.2.3 Main Feature Development of SuperPoupa (Adriano Villa Bascón)

Using the Figma wireframes as guidance for the development within OutSystems, the decision was made to work in an agile fashion that comprised two-week cycles of goal setting and iterating. It was concluded that this is the right approach to stay flexible and adjust goals when-ever needed. Goals were mainly tied to features, blocks, or screens that were planned. Each of those was based on user stories and narratives which have been agreed on beforehand.

5.2.3.1 Working Style and Frameworks

User Narratives

In web development, user narratives, also known as user stories, are simple descriptions of a feature told from the user's perspective. They are used to inform the development of a website or web application and to extract what features potential users would benefit from an application. More extended versions (Figure 12) involve details about the potential user persona and can imply many potential features. From them, shorter, more precise user narratives can be distilled. Shorter versions usually focus on one specific benefit of a feature (Figure 13).



Figure 12 - Longer Versions of SuperPoupa's User Narratives

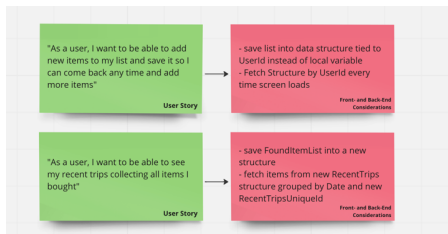


Figure 13 - Shorter Versions of SuperPoupa's User Narratives

Specific implications for the back-end and front-end development can then be drawn from it. Furthermore, user narratives help prioritise features by turning abstract technical descriptions of features into comprehensible and relatable descriptions.

For SuperPoupa, first versions were more extended versions of user narratives about various potential users' perspectives. From there, shorter versions have been extracted that could directly be translated into to-dos. Then, with the overall goal of the app and the user narratives in mind, the features planned have been sorted into two dimensions: *Impact on overall functionality and user experience* as well as *workload*, defined by the amount of time and level of difficulty to build. The two dimensions were further separated into four quadrants and assigned the features, accordingly, as shown in Figure 14. This helped to break features down into levels of priority.

1. High priority - implement these features as soon as possible. These features are characterised by their high contribution to the app's overall goal and low barrier of implementation.
2. Medium priority - implement when higher priority features are done or on hold. These features are still essential for the app's overall goal, but an MVP could work without them.
3. Low priority - implement these features only if resources allow. These features can add value to the user experience or app performance but are not essential to them.

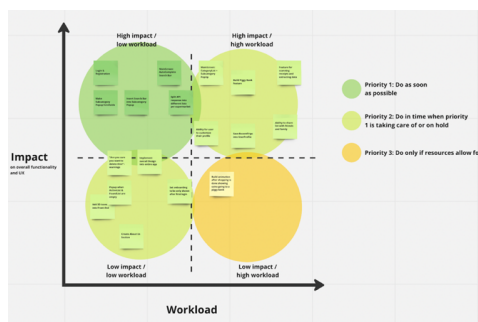


Figure 14 - Prioritization Quadrant for SuperPoupa's Features

Thanks to user narratives' simplistic nature, the value of features can be better communicated to various individuals within an organization. Based on the same user story, developers can build features, designers can build user experiences, marketers can layout campaigns, writers can create slogans, and managers can make relevant decisions. Basing the development process on user narratives was, therefore, a crucial project component given that this project is both technical and entrepreneurial.

Two-week Iteration Cycles

Execution of the plan was conducted in intervals of two weeks. Initially, the vision of the product contained a multitude of features and screens. Reorganizing the features and screens into the prioritization quadrant enabled adaptation to the potentiality of the limited

timeframe. Subsequently, the primary functionalities that were deemed of the greatest value were emphasized. This included registration, a log-in function, and the main screen where the app's primary function, the set-up of the grocery list, would take place. The iterative nature of the agile technique enabled high levels of adaptability and the capacity to swiftly alter objectives when necessary. Combined with the speed with which OutSystems facilitates the process of constructing, publishing, and assessing applications, it was possible to continually refine ideas and adjust objectives if required. Development of SuperPoupa's Features Analysis of the survey results revealed that an application should be developed which possesses features that facilitate the user's adoption of it. Subsequently, user narratives and a prioritization quadrant were utilized to identify the key features that must be incorporated into the minimal viable product (MVP) of SuperPoupa.

5.2.3.2 Priority-One Features

Registration and Login

The initial two features featured were the registration and login screens. OutSystems offers a login screen with every new project, containing the mandatory workflows. Thereby, the registration was built upon a template with some modifications. This enabled a significant saving of time and effort, which could be utilized on other important, non-existent features. SuperPoupa's main feature could be used without the need for signing in; however, later features necessitated the use of a distinct user identifier. These features included the capacity to store a list without it being erased upon closure of the app, a user profile displaying a username and email address, and the capability for a user to view their past trips saved in their profile.

Dropdown Autocomplete Bar

A central element of the grocery list set-up is the auto-complete dropdown search bar that anticipates what a user is looking for based on the characters that have been typed in. From

a user perspective, it is crucial to quickly find the desired categories to add to the list without scrolling through a long list of categories. This ability is not only convenient but essential for quickly setting up a grocery list, which plays into SuperPoupa's main value proposition. Luckily, OutSystems offers a widget called "Input Autocomplete". This helped quickly establish the feature. As can be seen in figure 15, the dropdown autocomplete search bar was, thereby, the first feature that was built. This element fetches its list of options (items) from the integrated database's 'categories' entity. Therefore, users only need to start typing the main category (e.g. 'Tom') and then be shown all the corresponding options (e.g., 'Toma-to paste', 'Canned tomatoes, etc.). Once a dropdown item is clicked, it is added to a list that appears below. This can

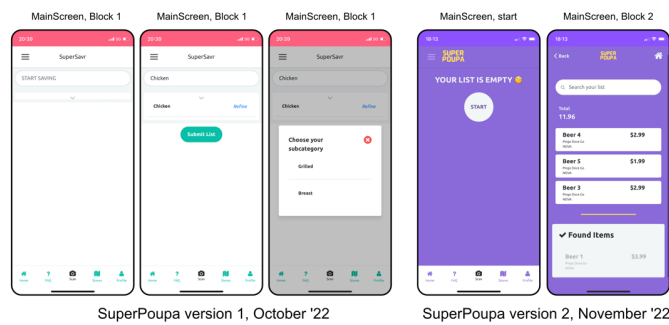


Figure 15 - SuperPoupa's First and Second Version

be repeated as many times as needed. Furthermore, main categories can be added to the list more than once, as each category needs to be specified with a subcategory. This allows users to add two or more subcategories within the same main category, e.g., "Gouda Cheese" and "Brie Cheese".

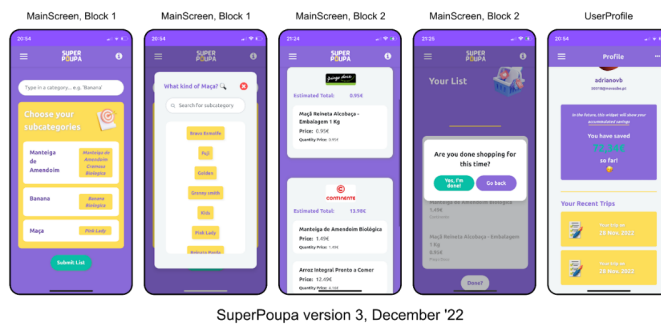
Interactive List with Subcategory Picker

The second element of the list-building block is the interactive list widget below the dropdown autocomplete search bar in which selected items are collected. As mentioned above,

this list consists of the items the user has chosen from the dropdown. Once clicked, a new list item, or row, is added to the list. This new list item first contains only the chosen main category (e.g., “Cheese”). It then requires further specification by adding a subcategory using the subcategory picker. Having to choose a subcategory is a necessity for the algorithm to work. It narrows down the range of entries (or products) that need to be searched upon within the database. However, it also helps users be more specific about what they want to buy. Without further specification, the result for, e.g., the cheapest cheese could stretch from slices of gouda to cream cheese. In chocolate, the results could stretch from chocolate bars to hot chocolate powder. Therefore, it was clear that although the goal is to make the list-building effort as minimal as possible, no value can be generated if the user cannot further specify their needs. A third step to further specify might need to be considered at later stages. The argument remains, however, that consumers who are sensitive to price may benefit more from the time and money saving offered by SuperPoupa, compared to the time that would be required to create a thorough list.

Once the subcategory button is clicked, a pop-up window will appear on the screen. This pop-up holds all the subcategories a user can choose within the clicked category. The user can then select one subcategory, which will trigger the pop-up to close and the subcategory picker to be replaced by the chosen subcategory. This needs to be repeated for many category-list items on the screen. The list can be submitted only when a subcategory has replaced all subcategory pickers. If the user wishes to change a subcategory, the chosen subcategory must be clicked again. This will reopen the pop-up, and another subcategory can be chosen to replace the first one. Does the user try to submit before all categories are assigned a subcategory, an error message will be shown. Therefore, it is crucial that the user is only able to submit the list once all the information needed by the algorithm is present.

Furthermore, by swiping from right to left on category list items, entries can be deleted from the list. Unless the user submits the finished list, that list will stay intact for as long as it is not deleted or submitted. From a user experience point of view, not resetting the list is crucial as new ideas on what to buy can occur at any time. Therefore, the user can keep adding new items to the list until he or she is ready to submit the list and go grocery shopping. This list is saved into a data entity in OutSystems. Whenever the user is logged in, this entity is fetched, filtered by the corresponding UserId, and shown on the main screen. The development of this feature necessitated the restriction of full use of the app to only registered users. Figure 16 shows SuperPoupa's latest version.



SuperPoupa version 3, December '22

Figure 16 - SuperPoupa's Current Version. From Left to Right: Dropdown Search Bar and Category List, Subcategory Pop-Up, API List Split by Supermarket, Finish Shopping Pop-Up, Profile with Recent Trips

API Reponse, ActiveList & FoundList

This feature is what SuperPoupa is about: getting a price-optimized list of products and where to get them based on a list of categories that is fast and easy to assemble. Because the functionality for this feature was built outside of OutSystems, the approach to assembling this feature was different. Until then, all application features were built in OutSystems using actions, otherwise known as workflows or action flows. These include the workflows needed to log in,

The upper section of Block 2 consists of the final grocery list containing the specific products separated into supermarkets where they can be found at a given price. Each active list also displays a total sum for the products in that list. This sum adjusts to items being deleted. An item can either be deleted by swiping it from right to left or marked as found by swiping left to right. If an item is marked as found, it gets removed from the active list and displayed in a list below called *FoundList*. This list gathers all found items regardless of the supermarket they were located. If an item has been moved down by accident, the same left-to-right swipe



Figure 19 - Process Flow Between Block 1, Flask API, and Block 2

action will move it back to its original list. Once no items are left in any of the active lists, a button appears below the FoundList. This button allows the user to finish the trip and save the items in the FoundList into a new data entity called *RecentTrips*.

5.2.3.3 Priority-Two Features

Recent Trips

Having constructed the features within the priority-one quadrant, features in the priority-two quadrant were addressed. This feature had a substantial workload with a moderate impact. Nevertheless, at the point when the decision was made to develop this feature, sufficient experience with OutSystems was attained in order to create it quickly and effortlessly. Recent Trips are defined as the FoundList that a user submits after finishing a grocery run. The items submitted are filtered by the UserId as well as a unique identifier per trip. Those trips are fetched and displayed in a list whenever a user visits the user profile. Each row consists of a finished trip with the corresponding date on which the trip was made. Once a row is clicked, a pop-up window opens up. This pop-up contains all the items, their prices, as well as the total sum and the date from the trip that has been selected. This can be repeated for all trips in the user's history.

5.2.3.4 Feature Pipeline and Future Outlook

It was previously noted that the pipeline of features for SuperPoupa was extensive. Not all features within the prioritization quadrant were achievable given the scope of the thesis. Nevertheless, features were gathered which were not presently feasible but are highly promising for future development.

5.2.3.5 Choosing Between the Cheapest Path and One-Stop Trips

Section 4.2.2 of User Experience shows the development of a wireframe illustrating a screen that gives users the option to split their grocery list into multiple supermarkets or select

one store and receive the cheapest route within it. The latter is referred to as the "One-Stop Trips".

First, users can opt for the cheapest list possible, the current default setting. This list is purely optimized by price and what was earlier referred to as the “cheapest path”. It will, therefore, most likely result in several lists, one for each supermarket, that offer the cheapest products for the desired categories. If, by coincidence, all products in the desired categories have the lowest price within one store, the cheapest list equals the cheapest One-Stop Trip.

Second, they can opt for the cheapest One-Stop Trip. This list will show all products within the desired categories within one supermarket at the lowest price. However, as a result, this is not necessarily the overall cheapest path.

Third, users can get the cheapest list for the closest One-Stop Trip. As such, further options for One-Stop Trips will be offered ranked by the proximity to the user’s location. This, too, can be the cheapest One-Stop Trip or the cheapest path while being the closest store, as neither is mutually exclusive.

Local Stores & Store Selection

By now, the lists are constructed by optimizing price and location. To be able to offer more lists and a more individualized experience beyond this, a feature for store selection will be implemented. Users can share their location or set an address as their desired location. From there, they can leave all stores selected by default or pick stores they want to exclude from the results. This would allow users to exclude stores they dislike and do not have access to because there might not be parking spots, but also, e.g., exclude all non-organic stores. Of course, this could impede the availability of the cheapest path; however, it could still be the most cost-effective list for a user's specifications, which could be even more beneficial for specific shoppers.

User Preferences for Groceries

On top of store preferences, another possibility for individualizing the experience will be introduced. With the User Preferences feature, users can filter all products considered by the algorithm by personal preferences such as “Organic”, “Plant-Based”, “Gluten-Free”, and “Nutri Score Rating”. This might seem like a nice-to-have but could potentially attract new groups of users who do not depend on saving on groceries but, when given the possibility, will still look for the cheapest option within their preferences. Users would select this when performing the onboarding. From there on, they have the same interface to build a categories/subcategories list, but the algorithm will already know which products to include or not include.

Favorite Items

Further personalization can be achieved by enabling users to mark categories, subcategories, and specific products as favourites. The benefit of this feature is twofold. First, users will find their favourite items easier within the search. Has a user, e.g., marked “Cherry tomatoes” as a favourite, that item will be shown on top of all others when typing “tomatoes” into the main search or “cherry” into the subcategory search for tomatoes. This speeds up the list-building process even further. Second, by gathering this information, personalized recommendations can later on be made to users. A future home screen version could then, e.g., display when favoured products or products within favoured categories are on sale. A history of what users prefer and favourite also helps further improve the product for various target groups. Historical data on prices would also allow users to understand the price histories of their favourite items.

Coupons

The survey results suggest that the option to exchange points gathered by using the app for coupons to redeem at supermarkets may be an important factor in appealing to Portuguese

users. Implementing this feature would allow for a more concrete demonstration of the advantages that SuperPoupa provides, thus making the app's value more tangible. However, given the complexity of the technical implementation and the need for collaboration with supermarkets, this feature is likely to remain a distant prospect for the foreseeable future.

Re-Submit Recent Trips

Another option to simplify the usage of SuperPoupa is to offer users the option to re-submit the exact categorical specification of a recently completed trip. This could help users who e.g., do weekly shopping, in which most of the items are the same, to have an even faster and easier experience using SuperPoupa.

Shared lists

Introducing a collaborative list-sharing feature will be a crucial element of SuperPoupa. This technology will enable families, housemates, friends, colleagues, and other groups to organize and carry out assignments. Numerous grocery list applications have already implemented this functionality (Brown 2020), and its presence is likely to be a significant factor in how users choose among the available apps.

Create various lists for different occasions

This proposed feature would enable users to efficiently manage their shopping lists by allowing them to save and work on multiple lists. This is beneficial for users who need to purchase items for multiple purposes, such as personal and business-related items, and for those who need to make multiple trips to the store for different occasions, such as weekly grocery trips and dinner parties.

Piggy Bank & Savings Account

Some users reported a reluctance towards paying for SuperPoupa. Therefore, generated savings must somehow be made more tangible to users. A *Piggy Bank* feature will allow users to

collect the savings they have made into their profile. This works by adding savings to an accumulated savings account whenever a user submits a FoundList. This will allow users to track how much money they have saved over time using SuperPoupa. This feature is crucial for emphasizing the value of SuperPoupa. However, a challenge that must be confronted when attempting to quantify the savings achieved through this feature is determining the benchmark that will be used for comparison. This benchmark will be essential for determining the true magnitude of the savings customers can expect.

Thus far, the savings quantified by SuperPoupa have only been represented digitally on the application's user interface and are purely hypothetical. Whether or not these savings are actually generated is beyond the reach of SuperPoupa. An add-on to the Piggy Bank feature could address the issue by enabling the transfer of money from bank accounts, credit cards, or PayPal accounts into SuperPoupa. By doing so, users would be further aided in saving money. Existing fintech applications such as Revolut feature dedicated vaults to which users can add money in order to save up for specific goals (Revolut, n.d.). From the user's point of view, this feature has the potential to create meaningful cost reductions. From the business perspective, however, this may represent a vast financial cash flow opportunity if a sufficient number of users can generate significant savings saved within SuperPoupa, always considering that users would retain the ability to withdraw savings whenever they wanted.

Creating Lists that is Optimized by Budget

This feature's development takes place later than the aforementioned ones due to its technical complexity. The idea here is that users can add a budget for a list they can spend. Then, the algorithm, once it has enough information about their shopping habits, will automatically create a grocery list for them, considering the budget restrictions and balancing dietary needs. The algorithm, of course, would need to know the users so well as to consider the history

of purchases and make predictions based on it. This is why it likely will be one of the most complicated to deploy features.

Nevertheless, if deployed and working correctly, this feature would be the most advantageous feature for consumers with a highly restricted budget. While the current list-building algorithm optimizes individual prices, lists can still cost more than a consumer is willing or even able to spend. However, if a user can set a fixed budget of, e.g., 50€ for the following grocery list, personal finances become easier to plan.

5.2.4 User-Feedback (Pepijn Van Es)

The following section examines whether consumers experienced the application as has been intended throughout the user-centric development. Especially in the case of an MVP, this process is of utmost importance. First, it evaluates the quality of the work done so far. However, most importantly, at this stage, customer feedback lays the foundation for adaptations and other design decisions that will be made throughout the further development of the application (Kleshchuck 2022). The mechanism – in this case, an online customer survey – also clarified any remaining questions in the literature concerning consumer perception and behaviour. The following questions acted as guided the setup of the research instrument.

- To what extent does the application alleviate consumers' grocery shopping problems?
- How satisfied are consumers with the application's look, feel, and ease of use?
- In what areas is the application lacking performance?

As such, it was possible to collect, organise, analyse, and interpret numerical statistics to generalise over a larger population, keeping in mind the representativeness of the retrieved sample (Sheard 2018). The target population were individuals residing in Lisbon and neighbouring areas, age and sex disregarded. The questionnaire has been distributed through WhatsApp, LinkedIn, and Instagram, with data collected from 20 November until 10 December. As a result,

a sample of 27 valid surveys was retrieved, with a reported average age of 26, 59.3% male, with an approximate non-response rate of 97%.

5.2.4.1 Research Instrument

To ensure customers completed the survey, the research instrument (figure X) followed several best practices (Shad 2020; Wellington 2022; Kleshchuck 2022). First, the number of questions was kept at a bare minimum. Only questions that would provide valuable insights which could be translated into actionable ideas were included. Also, the questionnaire used consistent rating scales and avoided loaded or double-barrelled questions (Ibid.). It consisted of 17 questions, of which seven were based on a 5-point Likert scale. Additionally, before embarking on the trial run and reporting their experiences, consumers were instructed on what to expect and how to use the platform most effectively, bearing in mind the early stages in which it finds itself.

Participants were asked what problems they faced when shopping for groceries and how the application helped them overcome these obstacles to assess the Customer Effort Score (CES). In addition, the Customer Satisfaction Score (CSAT) was explored by investigating the degree to which consumers were satisfied with the look and feel, ease of use, installation, and onboarding experience of the application. Also, consumers could elaborate on how the application could be improved regarding the relevant topic in the case of a low score. Finally, the Net Promoter Score (NPS) was revealed by exploring the willingness of consumers to recommend the application to friends and family (Shad 2020).

Beforehand, all respondents were introduced to the underlying meaning of the research and informed of the data processing procedures. Then, all were questioned for their consent to do so, provided that answering would be done anonymously. The data were anonymised and analysed upon collection using IBM SPSS Statistics (IBM 2020).

5.2.4.2 Customer Effort

In line with the empirical findings preceding the conceptualisation stage, across the board, participants again indicated their struggle with rising grocery prices. They thus hoped that SuperPoupa could ease this pain. Almost 63% of users agreed or fully agreed that this was indeed the case. However, for the remaining users, there were several re-marks made, which prevented users from finding the cheapest product. A reoccurring theme was that once users visited the store indicated on their device, they found local prices to differ from those displayed in the application. This was most likely because product data that was used for the trial was only scraped once. Although the trial's instructions indicated the likelihood of this happening, it became evident that their data must be scraped regularly, given the volatile nature of product pricing.

Another crucial overlooked point was that supermarkets' entire product catalogue was scraped; the algorithm used all these products to find the cheapest permutation. As a result, when users went to the store, occasionally, they could not find products because online product catalogues far exceeded the physical products available in individual – often smaller – stores (Boice 2021). This means that for the application to work optimally, it must consider individual store inventories before sending users on their way. Otherwise, it defies the whole purpose of the application since searching for inexistent items merely adds effort to grocery shopping.

5.2.4.3 Customer Satisfaction

Looking at customer satisfaction, 92,6% of users were satisfied with the look and feel of the application, and thus, there were few remarks. Several participants pointed, however, to the striking resemblance between SuperPoupa and Getir – a grocery delivery app (Getir 2022) – in terms of colour scheme and general design. Given the platform's limited reach, this should not be harmful. However, further down the line, when the application has expanded its user

base, and with that renown, this could become a problem, especially considering that both applications operate in the same space. Furthermore, in case of any party perceives SuperPoupa to be guilty of violating design plagiarism, they could pursue legal action, which puts in motion a cascade of unfavourable events (Verma 2022). Therefore, in hindsight, it is fair to say that an alteration of the application's design must be considered.

Now, although 70,3% of participants were satisfied with the ease of use of the application, there were remarks users made that are worth mentioning. First, several users' applications gave a 404 error, which, in the case of SuperPoupa, meant that the server running in the back could not find the requested products (Ionos 2022). The API's initial hosting configuration necessitated a connection to the internet for the host-computer, which resulted in this error. Whenever was not the case, users encountered a 404 error when trying to reach the API via the submit-button. Also, some participants stressed that sometimes they do not want to choose a subcategory yet are satisfied with any product within a category. As such, the search engine and algorithm should be set up in a way that does not mandates the selection of subcategories.

Generally, 85,2% of people were happy with the installation and onboarding experience. However, one user pointed out the inconvenience of having two devices within hand reach when using a QR code instead of a clickable link to access the application. Nonetheless, this should be no issue since the application will eventually be posted in Apple's App and Google's Play Store. Last, when asked for general points of annoyance regarding the application, it became clear that, after receiving their final grocery list, many users still needed to understand what to do with it. This could suggest two things. First, although the onboarding made sense, its content needed to provide more instructions on how to use the app effectively. Second, although the look and feel of the application were satisfactory, the interface design was not sufficiently intuitive. This needs to be further addressed for future iterations of it.

5.2.4.4 Additional Metrics

Regardless of its flaws, 77,8% of users indicated that they would recommend SuperPoupa to friends and family. However, this number most likely represents the user's opinion of a scenario in which SuperPoupa is fully functional and more refined. That said, 59,3% of users indicated a willingness to pay for the application, most of which would prefer paying a usage-based fee. Furthermore, in an informal discussion with participants, it became clear that some users want to store their accumulated savings on the application in actual money. This narrative came up once more when users were questioned about future features they would like to add to the application, as 56% would soon like to see a piggy bank where they can find their accumulated savings. Some even appeared to be willing to dedicate a part of these savings as payment instead of paying for the platform itself, which aligns with earlier findings. Also, almost 80% of users seemed keen on being able to exchange usage points – the form of which is to be decided upon – for coupons at supermarkets.

Users commit to the idea of connecting the application to their funds instead of having the app estimate how much they would have saved when using it. This implies that, on the one hand, they trust the platform to the extent that they would store money in it. On the other, they see a real-life application for the tools that SuperPoupa provide. It must be noted that assumptive reasoning is applied in the absence of data on these matters. However, should these statements be accurate, they would support positive outcomes in further development.

6 EVALUATION OF SUPERPOUPA'S IMPACT POTENTIAL (GROUP PART)

SuperPoupa aims to provide easily accessible and quickly implementable mobile solutions that are effective in reducing costs for consumers and combatting the effects of rising food prices. While its reach is broad, SuperPoupa is particularly well-suited for vulnerable households who are struggling to maintain their accustomed level of consumption. With the savings that SuperPoupa can enable, these consumers can gain greater autonomy over their grocery expenditures and new financial potential to address other areas affected by the cost-living-crisis. The cost-living-crisis has had a ripple effect on various aspects of life, with households facing budget constraints being unable to put their surplus into savings accounts. SuperPoupa's presence in this economic landscape is marked by its ability to provide the generated savings to cover other household expenses, hence for many users, saving is a necessity to stay afloat in the face of the rising costs of living.

SuperPoupa's influence is not solely limited to optimizing households' budgets; research has indicated that inflation can have a far-reaching effect on people's mental and physical health (Pappas 2022; Drexel University 2022). Specifically, the uncertainty caused by inflation can contribute to depression and other mental health issues, whereas opting for cheaper, yet less healthy, grocery items can have serious physical health implications. Therefore, it is important to consider the potential effect of the solution on these levels to gain a full understanding of its wide-reaching impact. The provision of resources and freedom of choice to customers can have beneficial impacts on households' physical and mental health, as well as restoring their sense of human dignity. An assessment of the true impact of SuperPoupa can only be conducted after its implementation in a real-world context, and the establishment of a clear method for measuring how it affects consumer spending and their lives. Despite lacking empirical evidence, the likely causal link between the adoption of SuperPoupa and financial relief for consumers can be preliminarily identified at this point.

7 CONCLUSION (GROUP CONTRIBUTION)

This thesis aims to examine the impact of current inflation rates on consumer grocery spending and assess the potential of a new mobile application, SuperPoupa, to alleviate its effects. The literature review and surveys show that consumers are adversely affected by rising food prices and are willing to adopt solutions to mitigate the financial strain. The findings suggest that the current climate will likely encourage even more users to seek assistance through these solutions. Furthermore, the surveys reveal a positive attitude towards adopting such solutions, particularly SuperPoupa, due to its simplicity and potential for significant time and money savings compared to other solutions. The approach taken by SuperPoupa offers an entrepreneurial opportunity to provide real value to consumers in Portugal in mitigating the effects of the cost-of-living crisis in the most convenient manner currently available.

In the future, introducing new features could provide further value propositions to a larger target audience looking for ways to reduce their grocery expenditures while prioritizing sustainability, health, and ethical considerations. In the long term, SuperPoupa has the potential to reduce the financial burden associated with rising food prices and address unhealthy and unsustainable dietary behaviours caused by a lack of resources and information asymmetry. However, due to the time constraints of this project, a comprehensive examination of the potential cost-saving outcomes achievable through the use of SuperPoupa could not be conducted. From here on, it is necessary to refine the solution and ensure its adoption by a sufficient number of users in order to make meaningful conclusions. The concept and impact of SuperPoupa are to be further validated. Once supportive evidence for achieving SuperPoupa's objectives has been established, it will be essential to assess its financial viability, as this will inform decisions concerning the revenue model. In conclusion, SuperPoupa has the potential to address a current

issue affecting consumers in Portugal, although further research and widespread implementation of SuperPoupa will be necessary to validate this claim.

8 REFERENCES

- Al Jazeera. 2022. "Wheat, corn prices surge deepening consumer pain." <https://www.aljazeera.com/economy/2022/3/3/wheat-corn-prices-surge-as-consumer-pain-mounts>. Accessed December 11, 2022.
- Alderman, L. 2022. "Bread Prices Skyrocket as Inflation Grips Europe - The New York Times." <https://www.nytimes.com/2022/10/19/business/europe-food-prices-inflation.html>. Accessed December 11, 2022.
- Atske, Sara. 2019. "4. The Internet Will Continue to Make Life Better." Pew Research Center: Internet, Science & Tech (blog). October 28, 2019. <https://www.pewresearch.org/internet/2019/10/28/4-the-internet-will-continue-to-make-life-better/>.
- Badenhop, Anna, and Marta Frascuet. 2021. "Online Grocery Shopping at Multichannel Supermarkets: The Impact of Retailer Brand Equity." *Journal of Food Products Marketing* 27 (2): 89–104. <https://doi.org/10.1080/10454446.2021.1894296>.
- Bahn, R., and Bahn, S. 2021. "Food Insecurity - an overview | ScienceDirect Topics." <https://www.sciencedirect.com/topics/social-sciences/food-insecurity>. Accessed December 11, 2022.
- Bazzoni, E., Mijer, M., Jacob, M., Land, S., Moulton, J., Welcherling, S. 2022. "Flipp's New App Turns Store Flyers Into Shopping Lists, Helps Compare Prices | TechCrunch." https://techcrunch.com/2015/02/18/flipp-turns-flyers-into-shoppinglists/?guc-counter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlMmNvbS8&guce_referrer_sig=AQAAACvgAOSNTCZYTT02Hz6b7N4OL4aBDVFSWTjD5ZhagOMXik7zVPo8wF3Kfani8gqPFRGgVFuIA26X7AUIU5BnxZ_G16RvQrvYbHRYT-PLvhlKhQMixoBOJfJ58pnZRXVe8hDLPP5nwxrZBNjzEsok9H4u24-lgEbNSnx5qOwtRTxiB7. Accessed September 9, 2022.
- Bazzoni, E., Mijer, M., Jacob, M., Land, S., Moulton, J., Welcherling, S., 2022. "European consumer pessimism intensifies in the face of rising prices" [WWW Document]. URL <https://www.mckinsey.com/capabilities/growth-marketing-and-sales/our-insights/survey-european-consumer-sentiment-during-the-coronavirus-crisis>.
- Beimborn, Daniel, and Maximilian Palitz. 2013. "Enterprise App for Mobile Applications – Development of a Benefits Framework.", 11. <https://core.ac.uk/download/pdf/301360241.pdf>
- Bivens, J. 2022. "Corporate profits have contributed disproportionately to inflation. How should policymakers respond?" <https://www.epi.org/blog/corporate-profits-have-contributed-disproportionately-to-inflation-how-should-policymakers-respond/>. Accessed November 14, 2022.
- Boice, M., 2021. Top Reasons Consumers Shop Online - Why Online Shopping is Popular [WWW Document]. Jungle Scout. URL <https://www.junglescout.com/blog/reasons-consumers-shop-online/> (accessed 12.9.22).
- Bresnahan, Timothy, Joe Orsini, and Pai-Ling Yin. 2014. "Platform Choice By Mobile App Developers" 37.
- Brockmann, Tobias, Stefan Stieglitz, and Arne Cvetkovic. 2015. "Prevalent Business Models

- for the Apple App ." *Wirtschaftsinformatik Proceedings* 2015, 17
- Brown, S., 2020. "Best shared grocery list apps to save you another trip to the store." CNET. February 23, 2020. <https://www.cnet.com/tech/services-and-software/best-shared-grocery-list-apps-to-save-you-another-trip-to-the-store/>
- Bubble.io. (n.d.). Pricing. <https://bubble.io/pricing>. Accessed on December 11, 2022.
- Buliř, Aleř. 2001. "Income Inequality: Does Inflation Matter?" *IMF Staff Papers* 48 (1): 139–59.
- Bureau of Labor Statistics. 2022. "Spending on food: 1901 versus 2002-03: The Economics Daily: U.S. Bureau of Labor Statistics." <https://www.bls.gov/opub/ted/2006/may/wk4/art05.htm>. Accessed November 14, 2022.
- Caldara, D., Conlisk, S., Iacoviello, M., and Penn, M. 2022. "The Effect of the War in Ukraine on Global Activity and Inflation."
- Caldara, D., Conlisk, S., Iacoviello, M., Penn, M., 2022. "The Effect of the War in Ukraine on Global Activity and Inflation."
- Camponovo, Giovanni, and Yves Pigneur. 2003. "Business Model Analysis Applied To Mobile Business," 12.
- CBPP. 2021. "Tracking the COVID-19 Economy's Effects on Food, Housing, and Employment Hardships." *Cent. Budg. Policy Priorities*. <https://www.cbpp.org/research/poverty-and-inequality/tracking-the-covid-19-economys-effects-on-food-housing-and>. Accessed November 14, 2022.
- Chao, E. and Utgoff, K. 2006. "100 Years of U.S. Consumer Spending: Data for the Nation, New York City, and Boston."
- Charm, T., Dhar, R., Haas, S., Liu, J., Novemsky, N., Teichner, W., 2022. "Understanding and shaping consumer behavior in the next normal." [WWW Document]. URL <http://ceros.mckinsey.com/coronavirus-promo-video-desktop>.
- Codesido, I., 2009. "What is front-end development?" *The Guardian*. September 28, 2009. <https://www.theguardian.com/help/insideguardian/2009/sep/28/blogpost>
- Collin, Jari, Kari Hiekkänen, Janne J Korhonen, Marco Halén, Timo Itälä, and Mika Helenius. 2015. "IT Leadership in Transition - the Impact of Digitalization on Finnish Organizations." *Aalto University Publication Series SCIENCE + TECHNOLOGY* 7: 121.
- Comastri, Thomas. 2015. "The Application Economy." *World Economic Forum*. 2015. <https://www.weforum.org/agenda/2015/01/the-application-economy/>.
- COVID-19 Pandemic: Prevalence and Associated Sociodemographic Characteristics." *Port. J. Public Health* 40: 35–42. <https://doi.org/10.1159/000522319>
- Cramer, Henriette, Mattias Rost, Nicolas Belloni, Frank Bentley, and Didier Chincholle. 2010. "Research in the Large. Using App , Markets, and Other Wide Distribution Channels in Ubicomp Research." In *Proceedings of the 12th ACM International Conference Adjunct Papers on Ubiquitous Computing - Ubicomp '10*, 511. Copenhagen, Denmark: ACM Press. <https://doi.org/10.1145/1864431.1864501>.
- Cuadrado, Félix, and Juan Dueñas. "Mobile Application Stores: Success Factors, Existing

- Approaches, and Future Developments." IEEE Communications Magazine 50, no. 11 (November 2012): 160–67. <https://doi.org/10.1109/MCOM.2012.6353696>.
- Deloitte. 2018. "The App Economy in the United States - A Review of the Mobile App Market and Its Contribution to the United States Economy."
- Dhar, R. 2022. "How Does Inflation Change Consumer Behavior? | Yale Insights." <https://insights.som.yale.edu/insights/how-does-inflation-change-consumer-behavior>. Accessed November 14, 2022.
- Domino, 2022. "What is Feature Engineering?" Domino Data Science Dictionary [WWW Document]. URL <https://www.dominodatalab.com/data-science-dictionary/feature-engineering>.
- Donn, N. 2022. "Food emergency threatens Portugal - Portugal Resident." <https://www.portugalresident.com/food-emergency-threatens-portugal/>. Accessed November 14, 2022.
- Drexel University. 2022. "Inflation puts healthy food out of reach for millions of Americans." Cent. Hunger Free Communities. <https://drexel.edu/hunger-free-center/news-events/media-coverage/2022/September/i-felt-like-i-failed/>. Accessed December 11, 2022.
- Dua, A., Mahajan, D., Oyer, L., and Ramaswamy, S. 2020. "US small-business recovery after the COVID-19 crisis | McKinsey." <https://www.mckinsey.com/industries/public-and-social-sector/our-insights/us-small-business-recovery-after-the-covid-19-crisis>. Accessed November 14, 2022.
- Elshan, E., Dickhaut E., Ebel P., 2003. „An Investigation of Why Low Code Platforms Provide Answers and New Challenges”.
- Euronews. 2022a. "Eurozone facing recession at the end of the year, Brussels says." Euronews. <https://www.euronews.com/my-europe/2022/11/11/europes-record-inflation-will-peak-at-year-end-but-remain-high-in-2023-says-brussels>. Accessed November 14, 2022.
- Euronews. 2022b. "Portugal drought: Fruit growers' harvest gloom after months of dry weather | Euronews." <https://www.euronews.com/2022/09/29/>
- European Central Bank. 2022. "Monetary Policy in a Cost-of-Living Crisis," September. <https://www.ecb.europa.eu/press/key/date/2022/html/ecb.sp220930~9dac17b1fe.en.html>.
- Ezell-Thomas, J. "Meatpacking Profiteers Testifying Today Saw Nearly \$13B in Profits After Racking Up \$384M In Price-Fixing Fines and Settlements." Accountable US. Accessed November 14, 2022. <https://accountable.us/meatpacking-profiteers-testifying-today-saw-nearly-13b-in-profits-after-racking-up-384m-in-price-fixing-fines-and-settlements/>.
- Fang, D., Thomsen, M. R., and Nayga, R. M. 2021. "The association between food insecurity and mental health during the COVID-19 pandemic." BMC Public Health 21: 607. <https://doi.org/10.1186/s12889-021-10631-0>
- Fireart, 2022. "Top Most Popular Programming Languages for Mobile App Development" fireart.studio. August 14, 2022. <https://fireart.studio/blog/top-most-popular-programming-languages-for-mobile-app-development/>

- FlutterFlow. (n.d.). Pricing. Retrieved from <https://flutterflow.io/pricing.html>. Accessed on December 11, 2022.
- Flynn, Jack. 2022. "40 Fascinating Mobile App Industry Statistics [2022]: The Success Of Mobile Apps In The U.S. – Zippia." 2022. <https://www.zippia.com/advice/mobile-app-industry-statistics/>.
- Fong, Adrian. 2017. "The Role of App Intermediaries in Protecting Data Privacy." *International Journal of Law and Information Technology* 25 (2): 85–114. <https://doi.org/10.1093/ijlit/eax002>.
- Ford, Caroline Morgan. 2012. "Smartphone Apps on the Mobile Web: An Exploratory Case Study of Business Models." Georgia State University.
- Forrester, 2022. "The Total Economic Impact™ of OutSystems - Cost Savings And Business Benefits Enabled By OutSystems". April, 2022. Retrieved from <https://www.outsystems.com/1/low-code-roi-tei/>
- Freer, A., 2022. "Shoppers are increasingly using their mobile phones and apps in store" [WWW Document]. *Bus. Apps*. URL <https://www.businessofapps.com/news/shoppers-are-increasingly-using-their-mobile-phones-and-apps-in-store/>.
- Frugl. 2022. "Save on Groceries with the Frugl App - Frugl Grocery." Accessed December 12, 2022. <https://www.frugl.com.au/>
- Fu, J. 2022. "'I felt like I failed': inflation puts healthy food out of reach for millions of Americans | US news | The Guardian." <https://www.theguardian.com/environment/2022/sep/30/inflation-healthy-food-eating-america>. Accessed December 11, 2022.
- Gajewski, Jean-François, and Li Li. 2015. "Can Internet-Based Disclosure Reduce Information Asymmetry?" *Advances in Accounting* 31 (1): 115–24. <https://doi.org/10.1016/j.adiaac.2015.03.013>.
- Gennetian, L., Conwell, J., and Daniels, B. "How Do Low-Income Families Spend Their Money?" *Econofact*. Accessed November 14, 2022. <https://econofact.org/how-do-low-income-families-spend-their-money>.
- Georgieva, E., 2022. "What Is UX Design and Why Does It Matter?" [WWW Document]. *Resolute Softw*. URL <https://www.resolutesoftware.com/news/what-is-ux-design/>.
- Getir, 2022. *Getir: compras em minutos* [WWW Document]. URL <https://getir.com/pt/> (accessed 12.9.22).
- Gourinchas, P.-O. "War Dims Global Economic Outlook as Inflation Accelerates." IMF. Accessed November 14, 2022. <https://www.imf.org/en/Blogs/Articles/2022/04/19/blog-weo-war-dims-global-economic-outlook-as-inflation-accelerates>.
- Gourinchas, P.-O., 2022. "War Dims Global Economic Outlook as Inflation Accelerates" [WWW Document]. IMF. URL <https://www.imf.org/en/Blogs/Articles/2022/04/19/blog-weo-war-dims-global-economic-outlook-as-inflation-accelerates>.
- Green, M. 2022. "Forty Percent of the World Food Program's Wheat Supplies Come from

- Ukraine | Wilson Center." <https://www.wilsoncenter.org/blog-post/forty-percent-world-food-programs-wheat-supplies-come-ukraine>. Accessed December 11, 2022.
- Grennes, Thomas. 2022. "The New World and the Ukraine-Russia Breadbasket." *Regulation* 45: 42.
- Hamm, M.J., 2018. *Wireframing essentials: an introduction to user experience design; learn the fundamentals of designing the user experience for applications and websites*. Packt Publishing, Birmingham Mumbai.
- Hanselman, H. "The coronavirus effect on global economic sentiment." McKinsey. Accessed November 14, 2022. <https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/the-coronavirus-effect-on-global-economic-sentiment>.
- Hanselman, H., 2022. "The coronavirus effect on global economic sentiment" [WWW Document]. URL <https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/the-coronavirus-effect-on-global-economic-sentiment>.
- Heitkoetter, Henning, Kay Hildebrand, and Claus Usener. 2012. "Mobile Platforms as Two Sided Markets." *AMCIS 2012 Proceedings*, 10. <https://aisel.aisnet.org/amcis2012/proceedings/AdoptionDiffusionIT/11>.
- Hurlburt, G., "Low-Code, N-Code, What's Under the Hood?" *IT Professional* 23, 6 (01 Nov. Dec. 2021): 4-7. 10.1109/MITP.2021.3123415
- Hyrynsalmi, Sami, Tuomas Makila, Antero Jarvi, Arho Suominen, Marko Seppanen, and Timo Knuutila. 2012. "App , Marketplace, Play! An Analysis of Multi-Homing in Mobile Software Ecosystems," 14.
- Idu, Andrei, Tommy van de Zande, and Slinger Jansen. 2009. "Multi-Homing in the Apple Ecosystem: Why and How Developers Target Multiple Apple App Utrecht University, 7.
- IGD. 2022. "Food inflation rate to peak in early 2023, then slow." <https://www.igd.com/home/article-viewer/t/food-inflation-rate-to-peak-in-early-2023-then-slow/i/30254>. Accessed December 11, 2022.
- Ionos, 2022. What "Error 404" means and how to fix it [WWW Document]. IONOS Digit. Guide. URL <https://www.ionos.com/digitalguide/websites/website-creation/what-does-the-404-not-found-error-mean/> (accessed 12.9.22).
- ITU. 2016. "The Race for Scale: Market Power, Regulation and the App Economy." IvyPanda, "The History and Evolution of the Visual Basic Programming Language." March 2, 2022. <https://ivypanda.com/essays/the-history-and-evolution-of-the-visual-basic-programming-language/>.
- Jensen, J. B. "Economic consequences of the COVID-19 pandemic." 2020.
- Jia, Changjiang, Yan Cai, Yuen Tak Yu, and T.H. Tse. 2016. "5W+1H Pattern: A Perspective of Systematic Mapping Studies and a Case Study on Cloud Software Testing." *Journal of Systems and Software* 116 (June): 206–19. <https://doi.org/10.1016/j.jss.2015.01.058>.
- Kaushal, V., 2022. "Why is customer retention the ultimate growth strategy?" Hippo [WWW

- Document]. JungleWorks. URL <https://jungleworks.com/why-is-customer-retention-the-ultimate-growth-strategy/>.
- Kim, Eunice, Jhih-Syuan Lin, and Yongjun Sung. 2013. "To App or Not to App: Engaging Consumers via Branded Mobile Apps." *Journal of Interactive Advertising* 13 (1): 53–65. <https://doi.org/10.1080/15252019.2013.782780>.
- Kirby, P. "Why has Russia invaded Ukraine and what does Putin want?" BBC News. Accessed November 14, 2022. <https://www.bbc.com/news/world-europe-56720589>.
- Kleshchuck, S., 2022. "Getting the Most from Customer Feedback of your MVP" [WWW Document]. URL <https://enkonix.com/blog/mvp-feedback>.
- Kleshchuck, S., 2022. Getting the Most from Customer Feedback of your MVP | Enkonix [WWW Document]. URL <https://enkonix.com/blog/mvp-feedback> (accessed 11.18.22).
- Leibtag, E.S., Kaufman, P.R., 2003. "Exploring Food Purchase Behavior of Low-Income Households: How Do They Economize?" 8.
- Lenarduzzi, V., Taibi, D., 2016. "MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product." <https://doi.org/10.1109/SEAA.2016.56>.
- Louis, Tristan. 2013. "How Much Do Average Apps Make?" *Forbes*. 2013. <https://www.forbes.com/sites/tristanlouis/2013/08/10/how-much-do-average-apps-make/>.
- LucidChart, 2019. "What Is a Wireframe? Why You Should Start Using This UX Design Tool." Lucidchart Blog [WWW Document]. URL <https://www.lucid-chart.com/blog/what-is-a-wireframe>.
- McDaniel, Stephen W., C. P. Rao, and Ralph W. Jackson. 1986. "Inflation-Induced Adaptive Behavior." *Psychology and Marketing* 3 (2): 113–22. <https://doi.org/10.1002/mar.4220030207>.
- McLean, Graeme, Kofi Osei-Frimpong, Khalid Al-Nabhani, and Hannah Marriott. 2020. "Examining Consumer Attitudes towards Retailers' m-Commerce Mobile Applications – An Initial Adoption vs. Continuous Use Perspective." *Journal of Business Research* 106 (January): 139–57. <https://doi.org/10.1016/j.jbusres.2019.08.032>.
- Müller, Roland M., Björn Kijl, and Josef K. J. Martens. 2011. "A Comparison of Inter-Organizational Business Models of Mobile App : There Is More than Open vs. Closed." *Journal of Theoretical and Applied Electronic Commerce Research* 6 (2): 13–14. <https://doi.org/10.4067/S0718-18762011000200007>.
- Münch, J., Fagerholm, F., Johnson, P., Pirttilähti, J., Torkkel, J., Järvinen, J., 2013. "Creating Minimum Viable Products in Industry-Academia Collaborations." https://doi.org/10.1007/978-3-642-44930-7_9.
- Nelson, R. 2022. "Russia, Ukraine, and the Impact on the World's Food Supply." GLG. <https://glginsights.com/articles/russia-ukraine-and-the-impact-on-the-worlds-food-supply/>. Accessed December 11, 2022.
- Osterwalder, Alexander, and Yves Pigneur. 2011. *Business Model Generation: Ein Handbuch für Visionäre, Spielveränderer und Herausforderer*. Campus Verlag.

- OutSystems, 2020. "Managing the Applications Lifecycle: Manage Your OutSystems Infrastructure." success.outsystems.com. https://success.outsystems.com/Documentation/11/Managing_the_Applications_Lifecycle/Manage_Your_OutSystems_Infrastructure
- OutSystems. (n.d. a) "Platform Services." OutSystems, www.outsystems.com/evaluation-guide/platform-services/#1. Accessed December 10, 2022.
- OutSystems. (n.d. b). Pricing and Editions., <https://www.outsystems.com/pricing-and-editions/>. Accessed December 11, 2022.
- Palmer, A. "Three ways companies change their products to hide inflation." The Conversation. Accessed November 14, 2022. <http://theconversation.com/three-ways-companies-change-their-products-to-hide-inflation-189924>.
- Palmer, A. 2022. "European consumers cut back on discretionary spending." *Financ. Times*.
- Pappas, Stephanie. 2022. "Inflation Could Hit Your Mental Health as Much as Your Wallet, Psychologists Say." *Livescience.Com*. February 21, 2022. <https://www.livescience.com/inflation-mental-health-impact>.
- Parker, K., Minkin, R., and Bennett, J. "Economic Fallout From COVID-19 Continues To Hit Lower-Income Americans the Hardest." *Pew Res. Center's Soc. Demogr. Trends Proj.* Accessed November 14, 2022. <https://www.pewresearch.org/social-trends/2020/09/24/economic-fallout-from-covid-19-continues-to-hit-lower-income-americans-the-hardest/>.
- Parker, K., Minkin, R., Bennett, J., 2020. "Economic Fallout From COVID-19 Continues To Hit Lower-Income Americans the Hardest." *Pew Res. Center's Soc. Demogr. Trends Proj.* URL <https://www.pewresearch.org/social-trends/2020/09/24/economic-fallout-from-covid-19-continues-to-hit-lower-income-americans-the-hardest/>.
- Patel, H., 2021. "What is Feature Engineering — Importance, Tools and Techniques for Machine Learning" [WWW Document]. *Medium*. URL <https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10>.
- Patrick, Ruth, and Katie Pybus. 2022. "Cost of Living Crisis: We Cannot Ignore the Human Cost of Living in Poverty." *BMJ* 377 (April): o925. <https://doi.org/10.1136/bmj.o925>.
- Payne, Adrian, Pennie Frow, and Andreas Eggert. 2017. "The Customer Value Proposition: Evolution, Development, and Application in Marketing." *Journal of the Academy of Marketing Science* 45 (4): 467–89. <https://doi.org/10.1007/s11747-017-0523-z>.
- Peterson, E Wesley F. 2022. "The Coming Global Food Crisis." *Cornhusker Economics*. Lincoln: University of Nebraska.
- Phoenix, J., 2022. "11 Feature Engineering Tactics For Your Machine Learning Project" *Just Understanding Data* [WWW Document]. URL <https://understandingdata.com/11-feature-engineering-tactics-for-an-ml-project/>.
- Pucci, Raymond. 2018. "Mobile Apps Drive Brand Value Proposition." *PaymentsJournal*

- (blog). April 30, 2018. <https://www.paymentsjournal.com/mobile-apps-drive-brand-value-proposition/>.
- Qualtrics. 2022. "Customer Segmentation Analysis: Definition & Methods Qualtrics. Accessed December 13, 2022. <https://www.qualtrics.com/experience-management/brand/customer-segmentation/>.
- Rao, Srikumar, and John U. Farley. 1987. "Effects of Environmental Perceptions and Cognitive Complexity on Search and Information Processing." *Psychology and Marketing* 4.
- Revolut. "Meet your financial goals with vaults." Revolut. <https://www.revolut.com/en-PT/meet-your-financial-goals-with-vaults/>. Accessed on December 11, 2022.
- Richter, F., 2022. "Chart: Americans Suffer Pay Cut as Inflation Outpaces Wage Growth." Statista [WWW Document]. URL <https://www.statista.com/chart/27610/inflation-and-wage-growth-in-the-united-states/>.
- Schweitzer, J., Khattar, R., 2022. "Wages and Employment Do Not Have To Decline To Bring Down Inflation." Center for American Progress [WWW Document]. URL <https://www.americanprogress.org/article/wages-and-employment-do-not-have-to-decline-to-bring-down-inflation/>.
- Schweizer, E., 2022. "How Profit Inflation Made Your Groceries So Damn Expensive" [WWW Document]. URL <https://www.forbes.com/sites/erolschweizer/2022/09/12/how-profit-inflation-made-your-groceries-so-damn-expensive/>.
- Shad, A.A., 2020. "How to Get Minimum Viable Customer Feedback." Growth Mentor. URL <https://www.growthmentor.com/blog/how-to-get-minimum-viable-customer-feedback/>.
- Shad, A.A., 2020. How to Get Minimum Viable Customer Feedback. Growth Mentor. URL <https://www.growthmentor.com/blog/how-to-get-minimum-viable-customer-feedback/> (accessed 11.18.22).
- Shang, Y., Li, H., Zhang, R., 2021. "Effects of Pandemic Outbreak on Economies: Evidence From Business History Context." *Front. Public Health* 9.
- Sharma, B., 2022. "6 Benefits of Developing a Mobile App for Your Business" [WWW Document]. URL <https://techgenies.com/6-benefits-of-developing-a-mobile-app-for-your-business/>.
- Smith, K.-A., Bailie, K., 2022. "Here's Why Food Prices Are Still Rising." *Forbes Advisor* [WWW Document]. URL <https://www.forbes.com/advisor/personal-finance/why-are-food-prices-still-rising/>.
- Solis-Moreira, J. 2022. "Fat and broke: People eat more processed, unhealthy food during a recession - Study Finds." <https://studyfinds.org/more-fat-sugar-during-recession/>. Accessed December 11, 2022.
- Soll, D. 2022. "How food prices hit low-income households hardest - Chicago Sun-Times." <https://chicago.suntimes.com/2022/7/18/23268537/food-prices-inflation-low-income-households-consumer-spending-david-soll-the-conversation-op-ed>. Accessed November 14, 2022.
- Soll, D., 2022. "How food prices hit low-income households hardest." *Chicago Sun-Times*

- [WWW Document]. URL <https://chicago.suntimes.com/2022/7/18/23268537/food-prices-inflation-low-income-households-consumer-spending-david-soll-the-conversation-op-ed>.
- Soll, D., 2022. "How food prices hit low-income households hardest." Chicago Sun-Times [WWW Document]. URL <https://chicago.suntimes.com/2022/7/18/23268537/food-prices-inflation-low-income-households-consumer-spending-david-soll-the-conversation-op-ed>.
- Statista. 2022. "App - Worldwide | Statista Market Forecast." Statista. 2022. <https://www.statista.com/outlook/dmo/app/worldwide>.
- Stawarz, K., Rodríguez, M.D., Cox, A.L., Blandford, A., 2016. "Understanding the use of contextual cues: design implications for medication adherence technologies that support remembering." *Digit. Health* 2, 2055207616678707. <https://doi.org/10.1177/2055207616678707>
- Tang, Ailie K. Y. 2016. "Mobile App Monetization: App Business Models in the Digital Era." *International Journal of Innovation, Management and Technology*, 224–27. <https://doi.org/10.18178/ijimt.2016.7.5.677>.
- Tariq, H., 2021. "Low-Code Versus No-Code And The Future Of Application Development". *Forbes*. May 7, 2021. <https://www.forbes.com/sites/forbescommunicationscouncil/2021/05/07/low-code-versus-no-code-and-the-future-of-application-development/>
- Tetlow, Dr Gemma. 2022. "Tackling Rising Inflation and Slowing Growth," 18. <https://www.instituteforgovernment.org.uk/publications/rising-inflation-slowing-growth>.
- Tian, Yuan, Meiyappan Nagappan, David Lo, and Ahmed E. Hassan. 2015. "What Are the Characteristics of High-Rated Apps? A Case Study on Free Android Applications." In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 301–10. Bremen, Germany: IEEE. <https://doi.org/10.1109/ICSM.2015.7332476>.
- Tozzi, C., 2021. "The Evolution of Low-Code/No-Code Development" *ITPro Today*. November 30, 2021. <https://www.itprotoday.com/no-codelow-code/evolution-low-codenocode-development>.
- Trading Economics, 2022. "Portugal Inflation Rate - October 2022 Data - 1961-2021 Historical - November Forecast" [WWW Document]. URL <https://tradingeconomics.com/portugal/inflation-cpi>.
- Uhl, J., 2022. "75% Of Middle Class Households Say Their Wages Aren't Keeping Up With Inflation." *Hate It Here Never Want Leave*. URL <https://neverleave.substack.com/p/75-of-middle-class-households-say>.
- United Nations, 2022. "Fragile economic recovery from COVID-19 pandemic upended by war in Ukraine" [WWW Document]. U. N. URL <https://www.un.org/en/desa/fragile-economic-recovery-covid-19-pandemic-upended-war-ukraine>.
- Vannieuwenborg, Frederic, Laurent Mainil, Sofie Verbrugge, Mario Pickavet, and Didier

- Colle. 2012. "Business Models for the Mobile Application Market from a Developer's Viewpoint." In 2012 16th International Conference on Intelligence in Next Generation Networks, 171–78. Berlin, Germany: IEEE.
<https://doi.org/10.1109/ICIN.2012.6376022>.
- Vardhini, P., 2020. "All you wanted to know about down trading." The Hindu BusinessLine [WWW Document]. URL <https://www.thehindubusinessline.com/opinion/columns/slate/all-you-wanted-to-know-about-down-trading/article32204231.ece>.
- Verma, D., 2022. Are you guilty of web design plagiarism? webdew.com [WWW Document]. URL <https://www.webdew.com/blog/web-design-plagiarism> (accessed 12.9.22).
- Verma, J., 2021. "A Complete Guide to Categorical Data Encoding" [WWW Document]. URL <https://analyticsindiamag.com/a-complete-guide-to-categorical-data-encoding/>.
- Vos, Rob, Joseph Glauber, David Laborde, and Manuel Hernandez. 2022. "COVID-19 and Rising Global Food Prices: What's Really Happening?"
<https://doi.org/10.13140/RG.2.2.31433.62563>.
- Wac, Katarzyna, Selim Ickin, Jin-Hyuk Hong, Lucjan Janowski, Markus Fiedler, and Anind K. Dey. 2011. "Studying the Experience of Mobile Applications Used in Different Contexts of Daily Life." In Proceedings of the First ACM SIGCOMM Workshop on Measurements up the Stack - W-MUST '11, 7. Toronto, Ontario, Canada: ACM Press.
<https://doi.org/10.1145/2018602.2018605>.
- Wallace, A., 2022. "Extreme weather could push food inflation even higher." CNN Business [WWW Document]. URL <https://edition.cnn.com/2022/09/14/economy/heat-inflation-economy-drought/index.html>.
- Ward-Glenton, H. 2022. "IMF cuts global growth forecast for 2023, warns 'worst is yet to come'." <https://www.cnbc.com/2022/10/11/imf-cuts-global-growth-forecast-for-2023-warns-worst-is-yet-to-come.html>. Accessed December 11, 2022.
- Ward-Glenton, H., 2022. "IMF cuts global growth forecast for 2023, warns 'worst is yet to come'" [WWW Document]. URL <https://www.cnbc.com/2022/10/11/imf-cuts-global-growth-forecast-for-2023-warns-worst-is-yet-to-come.html>.
- Wellington, E., 2022. "Customer Feedback: Why It's Important + 7 Ways to Collect It" Help Scout [WWW Document]. URL <https://www.helpscout.com/blog/customer-feedback/>.
- Wellington, E., 2022. Customer Feedback: Why It's Important + 7 Ways to Collect It - Help Scout [WWW Document]. URL <https://www.helpscout.com/blog/customer-feed>.
- World Economic Forum. 2022. "9 charts that show how inflation and the Ukraine war are impacting European consumers." World Econ. Forum. <https://www.weforum.org/agenda/2022/06/how-inflation-and-the-conflict-in-ukraine-are-impacting-european-consumers/>. Accessed December 11, 2022.
- World Economic Forum. 2022. "Economic Inequality Has Deepened during the Pandemic. That Doesn't Mean It Can't Be Fixed." Accessed December 14, 2022.
<https://www.weforum.org/agenda/2022/04/economic-inequality-wealth-gap-pandemic/>.
- World Food Programme, 2022. "A global food crisis." World Food Programme [WWW Doc

ument]. URL <https://www.wfp.org/global-hunger-crisis>.

WorldBank, 2022. "Russian Invasion of Ukraine Impedes Post-Pandemic Economic Recovery in Emerging Europe and Central Asia" [WWW Document]. World Bank. <https://doi.org/10/04/russian-invasion-of-ukraine-impedes-post-pandemic-economic-recovery-in-emerging-europe-and-central-asia>.

Zhu, Kevin. 2002. "Information Transparency in Electronic Marketplaces: Why Data Transparency May Hinder the Adoption of B2B Exchanges." *Electronic Markets* 12 (2): 92–99. <https://doi.org/10.1080/10196780252844535>.

References Toni

"12 Benefits of Cloud Computing and Its Advantages." n.d. Salesforce.Com. Accessed December 11, 2022. <https://www.salesforce.com/products/platform/best-practices/benefits-of-cloud-computing/>.

"A Quote by Frank Lloyd Wright." n.d. Accessed December 10, 2022. <https://www.goodreads.com/quotes/5522-an-idea-is-salvation-by-imagination>.

"Amazon Web Services AWS – Server Hosting & Cloud Services." n.d. Accessed December 10, 2022. <https://aws.amazon.com/de/>.

"An Investigation into Mobile Application Development Processes: Challenges and Best Practices - -." n.d. Accessed December 10, 2022. <https://bia.unibz.it/esploro/outputs/journalArticle/An-Investigation-into-Mobile-Application-Development-Processes-Challenges-and-Best-Practices/991005773410501241>.

Aparicio, Diego, Zachary Metzman, and Roberto Rigobon. 2021. "The Pricing Strategies of Online Grocery Retailers." Working Paper. Working Paper Series. National Bureau of Economic Research. <https://doi.org/10.3386/w28639>.

"Applied Sciences | Free Full-Text | Classification of Retail Products: From Probabilistic Ranking to Neural Networks." n.d. Accessed December 10, 2022. <https://www.mdpi.com/2076-3417/11/9/4117/htm>.

Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. 2011. *Modern Information Retrieval: The Concepts and Technology Behind Search*. 2nd ed. New York: Addison-Wesley Professional.

Bourgeois, David, Joseph Mortati, Shouhong Wang, and James Smith. n.d. "Information Systems for Business and Beyond (2019)," 327.

Chamberlin, Donald D. 1976. "Relational Data-Base Management Systems." *ACM Computing Surveys* 8 (1): 43–66. <https://doi.org/10.1145/356662.356665>.

Chen, Peter Pin-Shan. 1976. "The Entity-Relationship Model—toward a Unified View of Data." *ACM Transactions on Database Systems* 1 (1): 9–36. <https://doi.org/10.1145/320434.320440>.

"Cloud-Computing-Dienste | Microsoft Azure." n.d. Accessed December 10, 2022. <https://azure.microsoft.com/de-de/>.

- Codd, E. 1971. "Further Normalization of the Data Base Relational Model." *Research Report / RJ / IBM / San Jose, California*. <https://www.semanticscholar.org/paper/Further-Normalization-of-the-Data-Base-Relational-Codd/7be40103ca2c4114e07bb327bf5f902d5b081808>.
- Codd, E F. 1970. "A Relational Model of Data for Large Shared Data Banks" 13 (6): 11.
- Codd, E. F. 1971. "Normalized Data Base Structure: A Brief Tutorial." In *Proceedings of the 1971 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control - SIGFIDET '71*, 1. San Diego, California: ACM Press. <https://doi.org/10.1145/1734714.1734716>.
- Cormen, Thomas H., ed. 2009. *Introduction to Algorithms*. 3rd ed. Cambridge, Mass: MIT Press.
- Coronel, Carlos, Steven Morris, and Peter Rob. 2013. *Database Systems: Design, Implementation, and Management*. 10e [edition]. Boston, MA: Course Technology/Cengage Learning.
- Dasgupta, Sanjoy, Christos Papadimitriou, and Umesh Vazirani. 2011. *Algorithms*. 1st ed. Science Engineering & Math.
- "Database Normalization Description - Office." n.d. Accessed December 11, 2022. <https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description>.
- Date, C. J. 1999. *An Introduction to Database Systems (7th Ed.)*. USA: Addison-Wesley Longman Publishing Co., Inc.
- Demba, Moussa. 2013. "Algorithm for Relational Database Normalization Up to 3NF." *International Journal of Database Management Systems* 5 (3): 39–51. <https://doi.org/10.5121/ijdms.2013.5303>.
- "Django." n.d. Django Project. Accessed December 10, 2022. <https://www.djangoproject.com/>.
- "Eileen Gray: A Design Pioneer." n.d. Studio Nicholson. Accessed December 10, 2022. <https://www.studionicholson.com/blogs/features/eileen-gray-a-design-pioneer>.
- Elmasri, Ramez, and Sham Navathe. 2016. *Fundamentals of Database Systems*. Seventh edition. Hoboken, NJ: Pearson.
- Fielding, Roy T., and Julian Reschke. 2014. "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content." Request for Comments RFC 7231. Internet Engineering Task Force. <https://doi.org/10.17487/RFC7231>.
- "Flask vs Django (2020) - Will Vincent." n.d. Accessed December 10, 2022. <https://wsvincent.com/flask-vs-django/>.
- Gehring, Wolfgang. n.d. "The Top Programming Languages." The State of the Octoverse. Accessed December 9, 2022. <https://octoverse.github.com/2022/top-programming-languages>.
- Genero, Marcela, Geert Poels, and Mario Piattini. 2008. "Defining and Validating Metrics for Assessing the Understandability of Entity–Relationship Diagrams." *Data & Knowledge Engineering* 64 (3): 534–57. <https://doi.org/10.1016/j.datak.2007.09.011>.
- Geng, Xiutang, Zhihua Chen, Wei Yang, Deqian Shi, and Kai Zhao. 2011. "Solving the Traveling Salesman Problem Based on an Adaptive Simulated Annealing Algorithm with Greedy Search." *Applied Soft Computing* 11 (4): 3680–89. <https://doi.org/10.1016/j.asoc.2011.01.039>.

Gibson, Joel, Robin Rondeau, Darren Eveleigh, and Qing Tan. 2012. "Benefits and Challenges of Three Cloud Computing Service Models." In *2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, 198–205. <https://doi.org/10.1109/CASoN.2012.6412402>.

Gillis, Alexander. n.d. "What Is REST API (RESTful API)?" App Architecture. Accessed December 9, 2022. <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>.

Group, PostgreSQL Global Development. 2022. "PostgreSQL." PostgreSQL. December 10, 2022. <https://www.postgresql.org/>.

Gupta, Vivek, Harish Karnick, Ashendra Bansal, and Pradhuman Jhala. 2016. "Product Classification in E-Commerce Using Distributional Semantics." In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 536–46. Osaka, Japan: The COLING 2016 Organizing Committee. <https://aclanthology.org/C16-1052>.

Hafez, Manar Mohamed, Ana Fernández Vilas, Rebeca P. Díaz Redondo, and Héctor Olivera Pazó. 2021. "Classification of Retail Products: From Probabilistic Ranking to Neural Networks." *Applied Sciences* 11 (9): 4117. <https://doi.org/10.3390/app11094117>.

IBM Cloud Education. 2021. "What Is Data Modeling?" January 6, 2021. <https://www.ibm.com/cloud/learn/data-modeling>.

K. Flora, Harleen, Xiaofeng Wang, and Swati V.Chande. 2014. "An Investigation into Mobile Application Development Processes: Challenges and Best Practices." *International Journal of Modern Education and Computer Science* 6 (6): 1–9. <https://doi.org/10.5815/ijmecs.2014.06.01>.

Khder, Moaiad. 2021. "Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application." *International Journal of Advances in Soft Computing and Its Applications* 13 (3): 145–68. <https://doi.org/10.15849/IJASCA.211128.11>.

Kleppmann, Martin. 2017. *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media.

Krebs, Bruno, and Juan Cruz Martinez. 2022. "Developing RESTful APIs with Python and Flask." Auth0 - Blog. September 15, 2022. <https://auth0.com/blog/developing-restful-apis-with-python-and-flask/>.

Kumar, Kunal, and S. K. Azad. 2017. "Database Normalization Design Pattern." In *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)*, 318–22. <https://doi.org/10.1109/UPCON.2017.8251067>.

"List of Supermarket Chains in Portugal." 2022. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=List_of_supermarket_chains_in_Portugal&oldid=1107315040.

Mathkour, Hassan. 2009. "Formalization and Verification of Relational Database Normal Forms Using the Gamma Framework." *Journal of King Saud University - Computer and Information Sciences* 21: 1–11. [https://doi.org/10.1016/S1319-1578\(09\)80001-3](https://doi.org/10.1016/S1319-1578(09)80001-3).

"Ngrok - Online in One Line." n.d. Accessed December 10, 2022. <https://ngrok.com/>.

"Online API Testing Tool | Test Your API Online." n.d. ReqBin. Accessed December 10, 2022. <https://reqbin.com>.

- “ParseHub | Free Web Scraping - The Most Powerful Web Scraper.” n.d. Accessed December 10, 2022. <https://www.parsehub.com/>.
- Persson, Emil. 2019. *Evaluating Tools and Techniques for Web Scraping*. <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-271206>.
- “Pingo Doce | Food Distribution.” n.d. *Jerónimo Martins* (blog). Accessed December 10, 2022. <https://www.jeronimomartins.com/en/about-us/what-we-do/food-distribution/pingo-doce/>.
- “PostgreSQL.” 2022. In *Wikipedia*. <https://en.wikipedia.org/w/index.php?title=PostgreSQL&oldid=1122798483>.
- “Relational-Databases.” 2022. July 11, 2022. <https://www.ibm.com/cloud/learn/relational-databases>.
- Schwartz, Baron, Peter Zaitsev, Vadim Tkachenko, Jeremy Zawodny, Arjen Lentz, and Derek J. Balling. 2008. *High Performance Mysql, 2nd Edition*. Second. O’Reilly.
- “Six Advantages of Cloud Computing - Overview of Amazon Web Services.” n.d. Accessed December 10, 2022. <https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html>.
- “Stack Overflow Developer Survey 2022.” n.d. Stack Overflow. Accessed December 9, 2022. https://survey.stackoverflow.co/2022/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2022.
- “Super Save Preços mais baixos – Apps bei Google Play.” n.d. Accessed December 10, 2022. <https://play.google.com/store/apps/details?id=com.savecook.save.wallet&hl=de&gl=US>.
- “Welcome to Flask — Flask Documentation (2.2.x).” n.d. Accessed December 10, 2022. <https://flask.palletsprojects.com/en/2.2.x/>.
- Zhong, Changhong, Lifen Jiang, Yan Liang, Huazhi Sun, and Chunmei Ma. 2020. “Temporal Multiple-Convolutional Network for Commodity Classification of Online Retail Platform Data.” In *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*, 236–41. ICMLC 2020. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3383972.3383989>.

9 APPENDICES

9.1 Customer Survey

Question 1-3

Socio-demographics

Q1 |What is your sex? Male ☐ Female ☐ Non-binary ☐

Q2 |How old are you? 18-24 ☐ 25-34 ☐ 35-44 ☐ 45-54 ☐ 55-64 ☐ 65+ ☐

Q3 |Which country are you from?

Open

Question 4-9

Inflation & Welfare

Q4 |What is the best way to describe your employment status in the last three months? Full-time ☐ Part-time ☐ Unempl. ☐ Non-Work. ☐ Student ☐ Retired ☐ Other ☐

Q5 |Please give your best guess about your net-income per month >400€ ☐ 400€-799€ ☐ 800€-1199€ ☐ 1200€-1799€ ☐ 1800€-2499€ ☐ 2500€-3499€ ☐ <350€ ☐

Q6 |How do you feel about your monthly net-budget covering your basic needs? Tight budget ☐ Decent budget ☐ Ample budget ☐

Q7 |Do you feel your monthly budget shrunk in the past due to rising prices (inflation)? Yes ☐ No ☐ Not certain ☐

Q8 |Please rank the following expenditures to where you felt the most affected by increasing prices.

Living expenses (rent, gas, electric bills e.g.)	1
Grocery expenses (food products, sanitary products, e.g.)	2
Commuting expenses (gas, public transport, e.g.)	3
Other	4

Q9 |Please specify other areas you felt an increase in prices

Open

Question 10-13

Grocery Shopping Behavior

Q10 |How often do you shop groceries on average in a week? Daily ☐ 3-4 times per week ☐ Once per week ☐

Q11 |How do you evaluate the following parts of your shopping trips

	Plan shopping	Comparing prices	Actual shopping	Tracking expens.
Not doing this activity	☐	☐	☐	☐
Dislike	☐	☐	☐	☐
Neither like nor dislike	☐	☐	☐	☐
Like	☐	☐	☐	☐

Q12 |When you go grocery shopping do you compare prices before (offerings, discounts, etc.)? Yes ☐ No ☐ Not certain ☐

Q12.1 |Which mediums do you use to check current prices & offers of supermarkets? Retailers Website ☐ Advertising brochures ☐ Price comparison apps ☐ Other ☐

Q12.1.1 |Which apps are you currently using?

Q12.1.2 |Please indicate which other sources you use to compare prices & offers of supermarkets

Q12.2 |Please indicate why you are not using tools to compare prices of your grocery products

I don't know beforehand which products I am going to buy	☐	The potential savings are not in relation to the work of comparing prices	☐	There are no convenient solutions for price comparison on the market yet	☐	Other	☐
--	--------------------------	---	--------------------------	--	--------------------------	-------	--------------------------

Q13 |What do you care most about when shopping for grocery products? I care...

Only about price	☐	Mostly about price, a bit about quality	☐	About both price & quality	☐	Mostly about quality, a bit about price	☐	Only about quality	☐
------------------	--------------------------	---	--------------------------	----------------------------	--------------------------	---	--------------------------	--------------------	--------------------------

Q13.1 |If you care both about quality and price, which of the following statement fits best to you?

I am on a tight budget, saving each cent matters	☐	I have to track my spendings, but I choose quality if I can	☐	I can afford quality, paying more for it is fine	☐
--	--------------------------	---	--------------------------	--	--------------------------

Q13.1.1 |For which products would you rather opt for quality than for the cheaper price?

Open

Question 14-16

Features

Q 14 (Would you consider integrating an application into your shopping activities if savings could be generated (E.g. app for price comparison from different stores that finds for your cheapest products)?)

Yes ☐No ☐Not certain ☐

Q14.1 (Please indicate why you would not use an app even though you could generate savings)

Open

Q14.2 (How many savings have to be generated at least so you are likely to use an app? (SRP = Suggested Retail Price))

For any amount I would make use of an app ☐ At least 5% of the SRP ☐ At least 10% of the SRP ☐ More than 15% of the SRP ☐

Q15 (Imagine there is an app that is using real price data to search for the cheapest products of your shopping list. Which other features would you like the app to have?)

- ☐ A piggybank where I can find my accumulated savings
- ☐ A community page where I can interact with other users
- ☐ A way to exchange usage points for coupons at supermarkets
- ☐ A way to tailor the application to my dietary needs
- ☐ A way for me to share shopping lists with friends
- ☐ Other

Q16 (Anything else you can think of which would improve a smart grocery app?)

Open

Question 17-18

Willingness-to-pay

Q17 (If the app would be generating savings as described before would you be willing to pay for it?)

Yes, a one-time installation fee ☐Yes, a monthly subscription fee ☐Yes, a usage-based fee ☐No I would not pay for it ☐

Q17.1 (How much would you pay once for the before described app?)

Less than 10€ ☐Between 10€ and 25€ ☐More than 25€ ☐

Q17.2 (How much would you pay monthly for the before described app?)

Less than 5€ ☐Between 5€ and 10€ ☐More than 10€ ☐

Q17.3 (How would you pay for the usage of the before described app?)

Less than 50ct. per search ☐Between 50ct.- 1€ per search ☐More than 1€ per search ☐

Q17.4 (Please indicate why you would not pay for the service like the price comparison app offers?)

Open

Q18 (What other pricing strategy could you imagine for the before-described app?)

Open

9.2 Customer Survey Results English

Survey Results | SuperPoupa EN

Survey Results: English SuperPoupa Questionnaire



Inflation & Welfare

Employment Status	Choice Count
Full-time	17
Part-time	7
Unemployed	1
Housewife	0
Student	22
Retired	0
Other	0

Income Range	Choice Count
less than 400€	2
400€-799€	4
800-1199€	17
1200€-1799€	15
1800€-2499€	6
2500€-3499€	3
3500€-4999€	0
More than 5000€	0

A horizontal bar chart with three bars representing different budget statements. The x-axis is labeled 'Choice Count' and ranges from 0 to 30 with major ticks at 0, 10, 20, and 30. The bars are red. The first bar, 'My budget is tight, I have to save ...', has a value of 12. The second bar, 'My budget is decent, I can afford ...', has a value of 30. The third bar, 'My budget is ample, I don't really ...', has a value of 5.

Budget Statement	Choice Count
My budget is tight, I have to save ...	12
My budget is decent, I can afford ...	30
My budget is ample, I don't really ...	5

Response	Choice Count
Yes	40
No	2
Not sure	5

A bar chart comparing average monthly expenses in PLN across three groups of respondents. The Y-axis represents the amount in PLN, ranging from 0 to 30. The X-axis shows three groups: 1 (do not use public transport), 2 (use public transport), and 3 (use a car). For each group, three bars represent different expense categories: Living expenses (rent, gas, electric bills etc.) in red, Grocery expenses (food products, sanitary products, e.g.) in blue, and Commuting expenses (gas, public transport, e.g.) in green. The values for each bar are labeled on top.

Group	Living expenses (rent, gas, electric bills etc.)	Grocery expenses (food products, sanitary products, e.g.)	Commuting expenses (gas, public transport, e.g.)
1 (do not use public transport)	15	22	5
2 (use public transport)	22	15	5
3 (use a car)	5	29	4

[illegible]

Survey Results | SuperPoupa EN

Question 10-13
Grocery Shopping Behavior

Q10 How often do you shop groceries on average in a week?



Q11 How do you evaluate the following parts of your shopping trips



Q12 (When you go grocery shopping do you compare prices before (offerings, discounts, etc.)?



Q12.1 (Which mediums do you use to check current prices & offers of supermarkets?



Q12.1.1 (Which apps are you currently using?



Q12.1.2 (Please indicate which other sources you use to compare prices & offers of supermarkets

No data available

Q12.2 (Please indicate why you are not using tools to compare prices of your grocery products

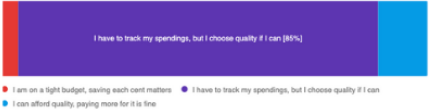
I'm already shopping at the cheapest stores and buy in bulk when I see sth on offer that I buy regularly
I don't have the time to go to different stores - always going to the same one
I don't have many shops close to my flat, so changing where I shop is quite difficult.

Q13 (What do you care most about when shopping for grocery products? I care...



Survey Results | SuperPoupa EN

Q13.1 If you care both about quality and price, which of the following statement fits best to you?

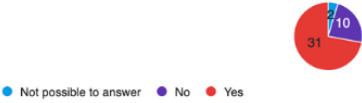


Q13.1.1 For which products would you rather opt for quality than for the cheaper price?



Question 14-16
Features

Q 14 (Would you consider integrating an application into your shopping activities if savings could be generated (E.g. app for price comparison from different stores that finds for your cheapest products)?



Q14.1 Please indicate why you would not use an app even though you could generate savings



Q14.2 How many savings have to be generated at least so you are likely to use an app? (SRP = Suggested Retail Price)



Q15 Imagine there is an app that is using real price data to search for the cheapest products of your shopping list. Which other features would you like the app to have?



Q16 (Anything else you can think of which would improve a smart grocery app?)

- An included map to see the distance of the stores
- Ordering directly through the app
- Comparing the prices of the closest markets + considering informal shops prices and local ones (not specifically each one but an average price on each of them)
- recipes that I can choose and automatically get the grocery list for
- Mo a simple blog with tips on cooking, improving diet, interesting facts
- Offer the option of choosing delivery, suggest recipes with what I'm going to buy or already have, suggest ingredients, walking distance to each place, carbon footprint or sustainable score for the products.
- Daily or weekly discounts/offers from supermarkets
- I use bring as a shopping list, therefore an integration a shopping list would be awesome

Question 17-18

Willingness-to-pay

Q17 (If the app would be generating savings as described before would you be willing to pay for it?)



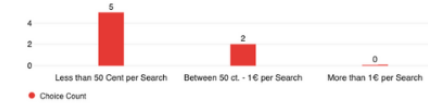
Q17.1 (How much would you pay once for the before described app?)



Q17.2 (How much would you pay monthly for the before described app?)



Q17.3 (How would you pay for the usage of the before described app?)



Survey Results | SuperPoupa EN

Q17.4 Please indicate why you would not pay for the service like the price comparison app offers?

- I don't really need it
- Cause you could do the research for free
- It's easier to just try out a few supermarkets. The cheapest are always Lidl & Aldi.
- Cause I don't like to go shopping at several places. It takes too much time
- From my point of view, the main reason why I don't compare much prices beforehand across different supermarkets in Portugal is because it is time-consuming (time = \$). So I wouldn't pay for a price comparison app because paying for the service means spend my savings in the shopping list
- Because it want to makes me save money
- Not enough value
- Because my basic need (time saving) is not being met. If this app applied to online grocery services (gotilias, getir etc) I would consider it tho
- Not a fan of subscription based payments and this feels like it's going to be one
- Scepticism on the calculation of savings and if it actually works
- Not sure about what I'll finally get
- I am trying to save money - worst case I just go to the supermarket myself and compare

Q18 What other pricing strategy could you imagine for the before-described app?

- Add based or sponsoring by participating supermarkets
- Have supermarkets and brands pay
- No pricing at all for end consumers (for the basic functions) info.
- The only pricing option I see suitable:
- A monthly subscription to unlock more features (e.g. integration with food tracking apps, recipes, no ads, etc) but the basic use should be for free
- Käse den Auge hamud!
- Why per seach? Would make more sense to make it related the amount you save? Like 10% auf what you saved using the app.
- One time purchase would be something I'd take a look at - depending on the price
- one week free trial, that I can see if I can save money
- Partnership with supermarkets
- Pay a percentage of the saving
- Percentage of what I buy -> so that I can see how much % in saving I still have nevertheless
- I would give 5% of the estimated money I've saved every month using the app

9.3 Customer Survey Results Portuguese

Survey Results | SuperPoupa PT

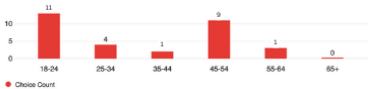
Survey Results: Portuguese SuperPoupa Questionnaire

Question 1-3
Socio-demographics

Q1 (What is your sex?)



Q2 (How old are you?)



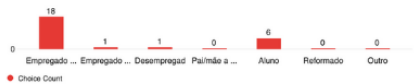
Q3 (Which country are you from?)

portugal

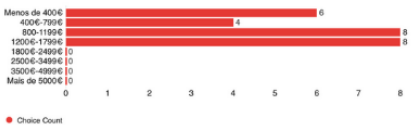
Survey Results | SuperPoupa PT

Question 4-9
Inflation & Wellfare

Q4 (What is the best way to describe your employment status in the last three months?)



Q5 (Please give your best guess about your net-income per month)



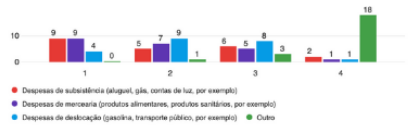
Q6 (How do you feel about your monthly net-budget covering your basic needs?)



Q7 (Do you feel your monthly budget shrunk in the past due to rising prices (inflation)?)



Q8 (Please rank the following expenditures to where you felt the most affected by increasing prices.



Q9 (Please specify other areas you felt an increase in prices

Aumento de preços foi global em todos os itens que estão no mercado foi sentida uma subida de preço
Medicação e consultas
Vestúlio, Alojamento Turístico.
Aquecimento para a casa, as peles aumentaram quase três vezes mais
Em tudo
2
Literatura, cultura, restauração

Survey Results | SuperPoupa PT

Question 10-13

Grocery Shopping Behavior

Q10 How often do you shop groceries on average in a week?



Q11 How do you evaluate the following parts of your shopping trips



Q12 (When you go grocery shopping do you compare prices before (offerings, discounts, etc.)?)



Q12.1 (Which mediums do you use to check current prices & offers of supermarkets?)



Q12.1.1 (Which apps are you currently using?)

No data available

Q12.1.2 (Please indicate which other sources you use to compare prices & offers of supermarkets)

No data available

Q12.2 (Please indicate why you are not using tools to compare prices of your grocery products)

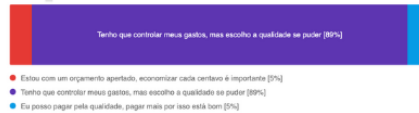


Q13 (What do you care most about when shopping for grocery products? I care...



Survey Results | SuperPoupa PT

Q13.1 *If you care both about quality and price, which of the following statement fits best to you?*

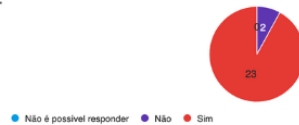


Q13.1.1 [For which products would you rather opt for quality than for the cheaper price?



Question 14-16
Features

Q 14 [Would you consider integrating an application into your shopping activities if savings could be generated (I.e. app for price comparison from different stores that finds for your cheapest products)?]



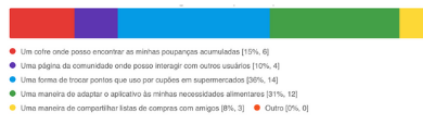
Q14.1 | Please indicate why you would not use an app even though you could generate savings



Q14.2 How many savings have to be generated at least so you are likely to use an app? (SRP = Suggested Retail Price)



Q15 Imagine there is an app that is using real price data to search for the cheapest products of your shopping list. Which other features would you like the app to have?



Q16 (Anything else you can think of which would improve a smart grocery app?

Sugestão de produtos mais baratos que sejam uma alternativa a produtos mais caros

Question 17-18

Willingness-to-pay

Q17) If the app would be generating savings as described before would you be willing to pay for it?



Q17.1 (How much would you pay once for the before described app?



Q17.2 (How much would you pay monthly for the before described app?



Q17.3 (How would you pay for the usage of the before described app?



Survey Results | SuperPoupa PT

Q17.4 Please indicate why you would not pay for the service like the price comparison app offers?

- No passado existiu uma app Kuantokusta supermercado gratuita apenas está temporariamente desativada funciona da forma que estão a pensar. Real time. Comparação de preços em várias plataformas simultaneo
- Porque acho desnecessário
- Posso fazê-lo na mesma sem ter de pagar
- Não sentiria necessidade de usar a aplicação
- I don't use paid apps
- Não pagaria por uma app de comparar preços se o posso fazer de outras maneiras grátis
- Por motivos financeiros
- Porque também implica paciência a sua utilização
- Pois existe no mercado aplicações grátis

Q18 What other pricing strategy could you imagine for the before-described app?

- Comprar por pacote ou em grupo
- Não sei
- Parcerias com supermercados
- Comparação de preços em relação aos supermercados mais próximos da minha localização.
- Não sei

9.4 Competitive Analysis

Categories Apps		Frugl	Super Save	Smhaggle
i	Company Overview	Frugl is an international retail intelligence ecosystem that acquires publicly available data from retailers. Their main product is the frugl price comparison application.	SuperSave is a Portuguese price comparison application for groceries including major retailers and offering different services in complementing apps	Smhaggle is a German-based price comparison application that includes it's community to keep price accurate (scanning option) and uses incentives such as cashbacks.
	Main competitive advantage	 Size Frugl is a competitor with a respectable outreach and user-base operating internationally.	 First-Mover SuperSave is a competitor as it was the first (successful) mobile application in Portugal offering price comparison of groceries.	 Innovator Smhaggle is a competitor as the application makes use of various innovative features to leverage best customer experience.
Key Info	Founding date	2014	2016	2007
	HQ based in	Australia	Portugal	Germany
	Countries active			
	Active Partners (e.g. Retailer)	500	6	8
Multihomeing	 Apple App Store  Rating  Downloads	  3.1 500k+	  4.5 30k+	  3.6 70k+
	 Google Play Store  Rating  Downloads	  3.5 10k+	  4.5 50k+	  4 100k+
Business Info	Revenue Streams	Collects behavioural data which is sold to Retailers, e.g.	- Commission from Retailers - Paid advertisement	Strategic partnership with retailers
	Differentiating Features	- Includes dietary preferences - Scanning of barcode of prod. - Sharing possibility of lists	- Includes price history - Has complementing Apps - SaveCook - SaveList	- Selling of behavioral data - Cash-back Coupons for additional discounts - Scanning of receipts for accurate price data
⚠	Best-Practice Use-Cases	Market-Penetration	Portuguese Market	Product-Innovation
	Implication for SuperPoupa	Copy strategy to reach top-ratings	Strong differentiation necessary as same market	Use product features as inspiration for own features

9.5 Customer Feedback Survey

Question 1-3

Socio-demographics

Do you give me consent to use your data for further analysis?	Yes ☐	No ☐
What gender do you most identify with?	Male ☐	Female ☐
How old are you?	Open	Prefer not to say ☐

Question 4-6

Customer Effort Score

<i>Please state your opinion on the following statements.</i>	Fully Disagree				Fully Agree
What problem(s) are you trying to solve with this application?			Open		
The application helped me solve this problem	1. ☐	2. ☐	3. ☐	4. ☐	5. ☐
If you scored the previous question below three, could you please elaborate?			Open		

Question 7-12

Customer Satisfaction Score

<i>Please state your opinion on the following statements.</i>	Fully Disagree				Fully Agree
I am satisfied with the look and feel of the application	1. ☐	2. ☐	3. ☐	4. ☐	5. ☐
If you scored the previous question below three, could you please elaborate?			Open		
I am satisfied with the ease of use of the application	1. ☐	2. ☐	3. ☐	4. ☐	5. ☐
If you scored the previous question below three, could you please elaborate?			Open		
I am satisfied with the installation and onboarding of the application	1. ☐	2. ☐	3. ☐	4. ☐	5. ☐
If you scored the previous question below three, could you please elaborate?			Open		

Question 13

General Information

What confused/annoyed you about the application?	Open
--	---------------------------

Question 14-17

Willingness to Pay

Would you be willing to pay for the application?	
Yes, a one-time installation fee	☐
Yes, a monthly subscription fee	☐
Yes, a usage-based fee	☐
No, I would not be willing to pay	☐

Question 18

Customer Satisfaction Score

<i>Please state your opinion on the following statements.</i>	Fully Disagree				Fully Agree
I am satisfied with overall quality of the application	1. ☐	2. ☐	3. ☐	4. ☐	5. ☐

Question 19

Customer Effort Score

What features would you like to add to the application?	
Yes, a one-time installation fee	☐
Yes, a monthly subscription fee	☐
Yes, a usage-based fee	☐
No, I would not be willing to pay	☐

Question 20

Customer Satisfaction Score

<i>Please state your opinion on the following statements.</i>	Fully Disagree				Fully Agree
I will recommend this application to a friend, colleague, or family member	1. ☐	2. ☐	3. ☐	4. ☐	5. ☐

9.6 User Feedback Results

Table 1

Frequencies: Effort Score

<i>The application helped me the problem identified in the previous question.</i>	Frequency	%	Σ
Fully Disagree	1	3.7	3.7
Disagree	2	7.4	11.1
Neutral	7	25.9	37
Agree	13	48.1	85.1
Fully Agree	4	14.8	100
Total	27	100	

% = relative percentage from complete data

Σ = accumulative percentage

Table 2

Frequencies: Customer Satisfaction Score

<i>I am satisfied with the look and feel of the application.</i>	Frequency	%	Σ
Fully Disagree	0	0	0
Disagree	0	0	0
Neutral	2	7.4	7.4
Agree	18	66.7	74.1
Fully Agree	7	25.9	100
Total	27	100	

% = relative percentage from complete data

Σ = accumulative percentage

Table 3

Frequencies: Customer Satisfaction Score

<i>I am satisfied with the ease of use of the application.</i>	Frequency	%	Σ
Fully Disagree	0	0	0
Disagree	5	18.5	18.5
Neutral	3	11.1	29.6
Agree	8	29.6	59.2
Fully Agree	11	40.7	100
Total	27	100	

% = relative percentage from complete data

Σ = accumulative percentage

Table 4

Frequencies: Customer Satisfaction Score

<i>I am satisfied with the installation and onboarding experience of the application.</i>	Frequency	%	Σ
Fully Disagree	1	3.7	3.7
Disagree	1	3.7	7.4
Neutral	2	7.4	14.8
Agree	14	51.9	66.7
Fully Agree	9	33.3	100
Total	27	100	

% = relative percentage from complete data

Σ = accumulative percentage

Table 5

Frequencies: Willingness to Pay.

<i>Would you be willing to pay for this application?</i>	Frequency	%	Σ
Yes, a one-time installation fee	2	7.4	7.4
Yes, a monthly subscription fee	1	3.7	11.1
Yes, a usage-based fee	14	51.9	63
No, I would not be willing to pay	10	37	100
Total	27	100	

% = relative percentage from complete data

Σ = accumulative percentage

Table 6

Frequencies: Future Functionalities.

<i>What future features would you like to add to the application?</i>	Frequency	%
A piggybank where I can find my accumulated savings	15	55.6
A community page where I can interact with other users	2	7.4
A way to exchange usage points for coupons at supermarkets	21	77.8
A way to tailor the application to my dietary needs	10	37
A way for me to share shopping lists with friends	6	22.2
None of the above	0	0
Total	54	100

% = percentage of complete data

9.7 Initial load_data() function loading CSV files

```
def load_data():  
    path = r"D:\Users\anton\Desktop\Nova_SBE\Master_Thesis\Data\DataBase"  
  
    df_cat = pd.read_csv(f"{path}\Category.csv ")  
    df_sup = pd.read_csv(f"{path}\Supermarkets.csv ", index_col = 0)  
    df_prod = pd.read_csv(f"{path}\Products.csv ", index_col = 0)  
    df_prod_cat = pd.read_csv(f"{path}\ProductsCategory.csv ", index_col = 0)  
    df_sup_prod = pd.read_csv(f"{path}\SupermarketsProducts.csv ", index_col = 0)  
  
    # we merge the prod_cat composite table based on product ids  
  
    df_cat_ids = pd.merge(df_sup_prod, df_prod_cat, left_on = "ProductId", right_on = "ProductsId")  
  
    # we merge the category names based on category ids  
  
    df_final = pd.merge(df_cat_ids, df_cat, left_on = "CategoryId", right_on = "Id ")  
  
    return df_cat, df_sup, df_prod, df_prod_cat, df_sup_prod, df_final
```

9.8 Commented Code | API & Algorithm in Python

```

1  ### Code Explanation
2
3  # General Summary: This code is a Python Flask app which provides a postJsonHandler function to handle
4  a POST request. The code reads in data from a PostgreSQL database and then performs various calculations
5  on the data to determine the cheapest path between products from different categories. The code then returns
6  a json response with the result of the calculations.
7
8  # More detailed:
9  # 1. A POST request is made to the route '/postjson' and the JSON content of the request is saved as the
10 variable 'user_categories_as_json'. From the JSON content, the 'CategoriesList' and 'subcategory_id' are
11 saved as the variable 'user_chosen_subcategories'. The code then sets up a connection to a Postgres database
12 using the credentials provided and a cursor object is created.
13
14 # 2. A function 'load_data' is defined which reads the data from the Postgres database and stores it in data-
15 frames.
16 # A function 'product_subcategories_for_algorithm' is defined which takes the user's chosen categories
17 # and the dataframe 'df_final' as input and returns a subset of the dataframe with the chosen categories.
18 # A function 'product_subset_of_each_subcategory' is defined which takes a single product category
19 # and the subset dataframe as input and returns the lowest prices for that product.
20
21 # 3. The function 'get_cheapest_paths' is defined which takes the user's chosen categories as input
22 # and returns the cheapest paths with the lowest prices for those categories.
23 # It calls the functions 'load_data' and 'product_subcategories_for_algorithm'
24 # and then loops through the categories and creates a list of the cheapest paths.
25 # The cheapest paths are evaluated and the one with the most supermarkets is chosen
26 # The result of the function is saved as the variable 'result' and then a JSON response is created and returned.
27 # The returned JSON is received and decoded by OutSystems.
28
29
30 # Import Libraries
31
32 # The following lines of code import various libraries needed for the program to work.
33 # pandas is a library for data manipulation and analysis,
34 # numpy is a library for scientific computing, Math is a library for mathematical functions,
35 # itertools is a library that provides functions for working with iterable objects,
36 # psycopg2 is a library for working with PostgreSQL databases,
37 # sys and os are libraries that provide system and operating system related functions,
38 # psql is a library for working with SQL databases, Flask is a library for building web applications,
39 # and Flask-restful is a library for working with APIs.
40
41 import pandas as pd
42 import numpy as np
43 import math
44 import itertools
45 import psycopg2
46 import sys, os
47 import pandas.io.sql as psql
48 from flask import Flask
49 from flask_restful import Api, Resource
50 from flask import request

```

```

51 import json
52
53 # App initialization
54 # This line creates an instance of the Flask class and assigns it to the variable app.
55 # The variable __name__ is set to __main__ when the program starts, so this will be the first step in running
56 the program.
57 # This line creates an instance of the Api class and assigns it to the variable api.
58 # This will allow us to create APIs for our application.
59 app = Flask(__name__)
60 api = Api(app)
61
62 # Route configuration
63 # This line creates a route for the application.
64 # This route will be used to receive a post request from the client and process it.
65 @app.route('/postjson', methods = ['POST'])
66 def postJsonHandler():
67
68 # This line checks to see if the request contains valid JSON data.
69 # If it does, the data is stored in the user_categories_as_json variable.
70     request.is_json
71     user_categories_as_json = request.get_json()
72
73 # This line extracts the user chosen category from the user_categories_as_json variable
74 # and stores it in the user_chosen_subcategories variable.
75     user_chosen_subcategories = [json["CategoriesList"]["subcategory_id"] for json in user_categories_as_json]
76
77
78 # These lines store the database credentials in variables.
79 # These credentials will be used to connect to the PostgreSQL database.
80     PGHOST= "funghiseverpostgresql.postgres.database.azure.com"
81     PGDATABASE= "postgres"
82     PGUSER= "adriano" #Adriano's credentials were used, but Antonius set it up and coded everything.
83     PGPASSWORD= "os12345!"
84     PGPORT= "5432"
85
86 # This line connects to the PostgreSQL database using the credentials stored in the variables.
87     connection = psycopg2.connect(database=PGDATABASE, user=PGUSER, password=PGPASSWORD,
88 host=PGHOST, port=PGPORT)
89     print("Connected!")
90
91 # This line creates a cursor object which will be used to execute queries on the database.
92     cursor = connection.cursor()
93
94 ####
95
96 # This line of code defines the load_data() function.
97     def load_data():

```

```

98
99 # This line of code uses Pandas to read the tables from the PostgreSQL database and stores them in a data-
100 frame.
101 df_category = pd.read_sql("SELECT * FROM category;", connection)
102 df_products = pd.read_sql("SELECT * FROM products;", connection)
103 df_products_subcategory = pd.read_sql("SELECT * FROM products_subcategory;", connection)
104 df_subcategory_category = pd.read_sql("SELECT * FROM subcategory_category;", connection)
105 df_supermarkets = pd.read_sql("SELECT * FROM supermarkets;", connection)
106 df_supermarkets_prod = pd.read_sql("SELECT * FROM supermarkets_products;", connection)
107 df_subcategory = pd.read_sql("SELECT * FROM subcategory;", connection)
108
109 # This line of code uses Pandas merge to combine the supermarkets_products and products_subcategory
110 tables
111 # Stores it in a dataframe.
112 df_category_ids = pd.merge(df_supermarkets_prod, df_products_subcategory, \
113                             left_on = "product_id", right_on = "product_id")
114
115 # This line of code uses Pandas merge to combine the df_category_ids and subcategory_category tables
116 # Store it in a dataframe.
117 df_final = pd.merge(df_category_ids, df_subcategory_category, \
118                     left_on = "subcategory_id", right_on = "subcategory_id")
119
120 # This line of code returns all of the dataframes stored in the function back to the main program.
121 return df_category, df_products, df_products_subcategory, df_subcategory_category, df_supermarkets, \
122 df_supermarkets_prod, df_category_ids, df_subcategory, df_final
123
124 ###
125
126 # This line defines a function named "ratiocheck" which takes a single parameter named "path".
127 def ratiocheck(path):
128
129 # This line creates a dataframe named "df"
130 # This is the result of grouping the data in the parameter "path" by "supermarket_id"
131 # It's then counting the number of records for each "supermarket_id".
132 # The grouped data is then sorted in descending order by the count of records for each "supermarket_id".
133 df = path.groupby("supermarket_id", as_index = False).count()["index"].sort_values("index", ascending
134 = False)\
135 .reset_index()
136
137 # This line creates a new column in the dataframe "df" called "shiftcalculation"
138 # Which is populated with the number of records for each "supermarket_id" shifted one row down.
139 # This allows us to compare the number of records for a "supermarket_id" with its adjacent "supermarket_id".
140 df["shiftcalculation"] = df["index"].shift(-1)
141
142 # This line sets the value of the last row in the "shiftcalculation" column to the number of records for the last
143 # "supermarket_id". This is necessary to ensure that the last comparison is between the last "supermarket_id"
144 # and the one before it.

```

```

145     df["shiftcalculation"].iloc[df.shape[0]-1] = df["index"].iloc[df.shape[0]-1]
146
147 # This line creates a new column in the dataframe "df" called "dif"
148 # Which is populated with the difference between the number of records for each "supermarket_id"
149 # And the number of records for the adjacent "supermarket_id".
150     df["dif"] = df["index"] - df["shiftcalculation"]
151
152 # This line returns the sum of the values in the "dif" column
153 # Which is the total difference between the number of records for each "supermarket_id"
154 # And The number of records for the adjacent "supermarket_id".
155     return sum(df["dif"].to_numpy())
156
157 ###
158
159 # This function takes two arguments - a list of product categories, and a dataframe of final product data
160 # Returns a dataframe of possible products in supermarkets.
161     def product_subcategories_for_algorithm(chosen_products, df_final):
162         """Accepts a list of product categories as input and returns a dataframe of possible products in supermar-
163         kets"""
164
165         # The dataframe returned only includes the columns "supermarket_id", "product_id", "price_quantity" and
166         "subcategory_id".
167         # The dataframe is filtered to only include the chosen product categories which have been given as an argu-
168         ment.
169         df_final_subdataframe = df_final[df_final["subcategory_id"].isin(chosen_products)]\
170         [["supermarket_id", "product_id", "price_quantity", "subcategory_id"]]
171
172         # Returns a dataframe of possible products in supermarkets.
173         return df_final_subdataframe
174
175     ###
176
177 # Defines a function that takes two parameters: a product category (product) and a dataframe (final_sub)
178     def product_subset_of_each_subcategory(product, final_sub):
179         """Accepts a product category as input and sorts values ascending to its price. Returns the lowest prices."""
180
181         # Checks the length of the dataframe is greater than 0
182         if len(final_sub) > 0:
183
184             # Creates a new dataframe (df_sub_prod) that is sorted by "price_quantity" in ascending order, and resets the
185             index
186             df_sub_prod = final_sub[final_sub["subcategory_id"] == product].sort_values(by = "price_quantity")\
187             .reset_index()
188
189             # Creates a new dataframe (df_sub_prod)
190             # that filters the dataframe to include only rows with the minimum value of "price_quantity"
191             df_sub_prod = df_sub_prod[df_sub_prod["price_quantity"] == df_sub_prod["price_quantity"].min()]

```

```

192
193 # Returns the filtered dataframe
194     return df_sub_prod
195
196 ###
197
198 # This line of code defines the function "get_cheapest_paths" which takes one argument, user_chosen_sub-
199 categories.
200     def get_cheapest_paths(user_chosen_subcategories):
201
202 # This line of code calls the function "load_data" which loads data from the database and assigns it to varia-
203 bles.
204     df_category, df_products, df_products_subcategory, df_subcategory_category, df_supermarkets, \
205     df_supermarkets_prod, df_category_ids, df_subcat, df_final = load_data()
206
207 # This line of code calls the function "product_subcategories_for_algorithm"
208 # which takes two arguments, user_chosen_subcategories and df_final and returns a dataframe.
209     final_sub = product_subcategories_for_algorithm(user_chosen_subcategories, df_final)
210
211 # This line of code calls the function "product_subset_of_each_subcategory" which takes two arguments,
212 prod and final_sub
213 # and returns a dataframe. This line of code also uses a list comprehension to check if the dataframe is None
214 # and if it has more than 0 rows and if so, it stores it in the list cheapest_path.
215     cheapest_path = [product_subset_of_each_subcategory(prod,final_sub) for prod in user_chosen_sub-
216 categories\
217                     if product_subset_of_each_subcategory(prod,final_sub) is not None and \
218                     product_subset_of_each_subcategory(prod,final_sub).shape[0] > 0]
219
220 # This line of code uses a list comprehension to get the number of rows in each item of the cheapest_path list
221 # And stores it in the list number_of_cheapest_products_per_subcategory.
222     number_of_cheapest_products_per_subcategory = [item.shape[0] for item in cheapest_path]
223
224 # This line of code uses the numpy product function to calculate the product of all the elements
225 # in the list number_of_cheapest_products_per_subcategory and stores it in the variable amount_of_cheap-
226 est_paths.
227     amount_of_cheapest_paths = np.prod(number_of_cheapest_products_per_subcategory)
228
229 # This line of code uses a list comprehension
230 # to create a list of empty arrays with the length of the list number_of_cheapest_products_per_subcategory
231 # and stores it in the list all_cheapest_paths.
232     all_cheapest_paths = [np.empty(len(number_of_cheapest_products_per_subcategory))\
233                          for s_paths in range(int(amount_of_cheapest_paths))]
234
235 # This line of code uses a list comprehension
236 # to get the indices from the list number_of_cheapest_products_per_subcategory where the element is equal
237 to 1
238 # and stores it in the list single_entry_indices.

```

```

239     single_entry_indices = [index for index, el in enumerate(number_of_cheapest_products_per_subcate-
240 gory) if el == 1]
241
242 # This line of code uses a list comprehension
243 # to get the indices from the list number_of_cheapest_products_per_subcategory where the element is greater
244 than 1
245 # and stores it in the list multiple_entry_indices.
246     multiple_entry_indices = [index for index, el in enumerate(number_of_cheapest_products_per_subcat-
247 egory) if el > 1]
248
249 # This line of code uses a nested loop
250 # to fill the list all_cheapest_paths with the first element of the corresponding item in the list cheapest_path.
251     for cheap_path in all_cheapest_paths:
252         for index in single_entry_indices:
253             cheap_path[index] = cheapest_path[index]["supermarket_id"].iloc[0]
254
255 # This line of code checks if the length of the list multiple_entry_indices is equal to 1.
256     if len(multiple_entry_indices) == 1:
257
258 # This line of code converts the array in the corresponding item of the list cheapest_path to a numpy array
259 # and stores it in the variable comb.
260         comb = cheapest_path[multiple_entry_indices[0]]["supermarket_id"].to_numpy() # comb = combi-
261 naitaiton
262
263 # This line of code assigns the first element of the list multiple_entry_indices to the variable index.
264         index = multiple_entry_indices[0]
265
266 # This line of code uses a loop to fill the list all_cheapest_paths with the items in the array comb.
267         for ind, sup_id in enumerate(comb):
268             all_cheapest_paths[ind][index] = sup_id #sup_id = supermarket_id
269
270 # This line of code checks if the length of the list multiple_entry_indices is greater than 1.
271         elif len(multiple_entry_indices) > 1:
272
273 # This line of code uses a list comprehension
274 # to convert the arrays in the corresponding items in the list cheapest_path to numpy arrays
275 # and store them in the list all_supermarket_ids.
276         all_supermarket_ids = [cheapest_path[index]["supermarket_id"].to_numpy() for index in multi-
277 ple_entry_indices]
278
279 # This line of code uses a list comprehension
280 # to create a list of tuples of all possible combinations of elements in the list all_supermarket_ids
281 # and store it in the variable comb.
282         comb = [l for l in itertools.product(*all_supermarket_ids)]
283
284 # This line of code uses a nested loop to fill the list all_cheapest_paths with the items in the list comb.
285         for index, combination in enumerate(comb):

```

```

286         for ind, value in enumerate(multiple_entry_indices):
287             all_cheapest_paths[index][value] = comb[index][ind]
288
289 # This line of code creates an empty list and stores it in the variable all_cheapest_paths.
290 all_cheapest_paths = []
291
292 # This line of code uses a nested loop to create a list of dataframes
293 # which correspond to the items in the list all_cheapest_paths.
294 for index, cheap_path in enumerate(all_cheapest_paths):
295     frames = []
296     for ind, sup_id in enumerate(cheap_path):
297         frames.append(cheap_path[ind][cheap_path[ind]["supermarket_id"] == sup_id])
298
299 # This line of code uses an if statement to check if the list frames is not empty
300 # and if it is not, it concatenates the dataframes in the list frames and stores it in the list all_cheapest_paths
301 if len(frames) > 0:
302     all_cheapest_paths.append(pd.concat(frames))
303
304 # This line of code checks if the length of the list all_cheapest_paths is equal to 1
305 # and if it is, it returns the first element of the list all_cheapest_paths.
306 if len(all_cheapest_paths) == 1:
307     return all_cheapest_paths[0]
308
309 # This line of code uses a loop to create two lists, pathlist and pathlistII,
310 # which store the length of unique supermarkets in each dataframe of the list all_cheapest_paths.
311 else:
312     pathlist = []
313     pathlistII = []
314     for index, path in enumerate(all_cheapest_paths):
315         pathlist.append(len(path["supermarket_id"].unique()))
316         pathlistII.append([len(path["supermarket_id"].unique()), index])
317
318 # This line of code checks if the list pathlist is not empty
319 # and if it is not, it stores the minimum value in the variable minpathlist.
320 if len(pathlist) > 0:
321     minpathlist = min(pathlist)
322
323 # This line of code uses a loop to create a list of dataframes
324 # which correspond to the minimum number of supermarkets in the list pathlist
325 # and stores it in the list final_pathlist.
326 final_pathlist = []
327 for pathindex in pathlistII:
328     if pathindex[0] == min(pathlist):
329         final_pathlist.append(all_cheapest_paths[pathindex[1]])
330
331 # This line of code checks if the list final_pathlist is equal to 1
332 # and if it is, it returns the first element in the list final_pathlist.

```

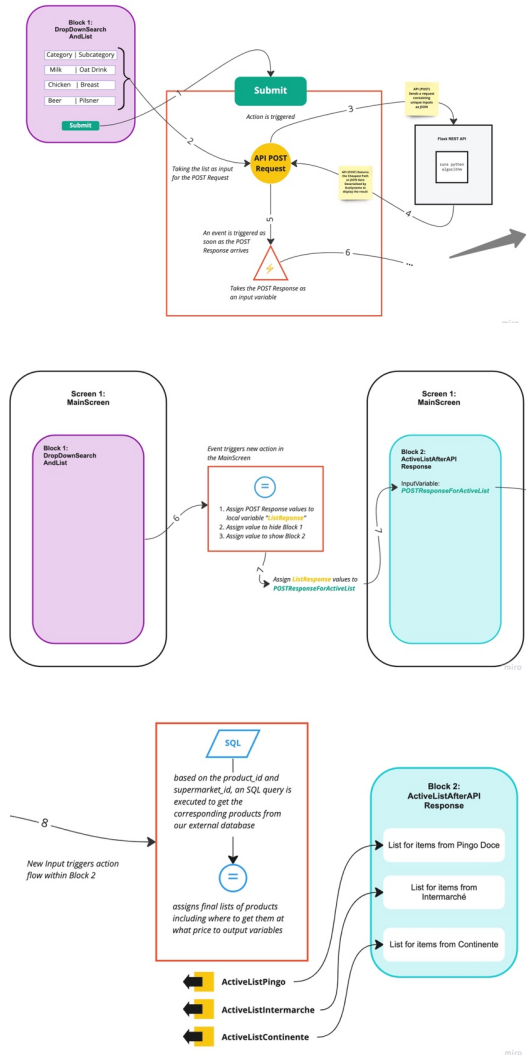
```

333     if len(final_pathlist) == 1:
334         return final_pathlist[0]
335
336     # This line of code uses a loop to create two lists, difference_list and difference_list_index,
337     # which store the result of the function ratiocheck for each dataframe in the list final_pathlist.
338     else:
339         difference_list = []
340         difference_list_index = []
341         for index, finalpaths in enumerate(final_pathlist):
342             difference_list.append(ratiocheck(finalpaths))
343             difference_list_index.append([ratiocheck(finalpaths), index])
344
345     # This line of code checks if the list difference_list is not empty
346     # and if it is not, it stores the maximum value in the variable maxdiff.
347     if len(difference_list) > 0:
348         maxdiff = max(difference_list)
349
350     # This line of code uses a loop to find the index in the list difference_list_index
351     # where the value is equal to the maximum value and returns the corresponding dataframe in the list
352     all_cheapest_paths.
353     for l in difference_list_index:
354         if l[0] == maxdiff:
355             return all_cheapest_paths[l[1]]
356
357     # This line of code calls the function get_cheapest_paths with the argument user_chosen_subcategories
358     # and stores the result in the variable result.
359     result = get_cheapest_paths(user_chosen_subcategories)
360
361     # This line of code creates an empty dictionary
362     # and stores it in the variable the_optimal_cheapest_path_as_json_response_back_to_outsystems.
363     the_optimal_cheapest_path_as_json_response_back_to_outsystems = {}
364
365     # This line of code checks if the result is not None
366     # and if it is not, it creates a list of the columns in the dataframe and stores it in the dictionary
367     # the_optimal_cheapest_path_as_json_response_back_to_outsystems.
368     if result is not None:
369         for column in result.columns:
370             the_optimal_cheapest_path_as_json_response_back_to_outsystems[column] = result[column].tolist()
371
372     # This line of code returns the dictionary the_optimal_cheapest_path_as_json_response_back_to_outsystems.
373     return the_optimal_cheapest_path_as_json_response_back_to_outsystems
374
375     # This line checks if the code is being run as the main program
376     # and if it is, it will run the app with a debug mode and without reloader.
377     if __name__ == "__main__":
378

```

```
app.run(debug=True, use_reloader=False)
```

9.9 Process Flow Between Block 1, Flask API, and Block 2



384 **9.10 Pricing Strategies of Different Applications**

Pricing Strategy	Description	Advantages	Disadvantages
1) Paid apps (Example: <u>ProCreate</u> ProCreate is a digital illustration application that is purchased in advance by users and is widely regarded by professionals in the field as providing superior tools and services in this segment.)	One-time installation fee is paid upfront. Users can afterwards access (usually) all features of an application.	• Safer cash-model • Less competing apps in the paid segment of the app stores • Effective marketing tools available such as “limited free offer”	• Payment is a barrier for users to obtain new applications • Without a reputation or proofed service hard to establish on the market
2) In-app purchase (Example: <u>Tinder</u> Tinder is a prominent online-dating platform that utilizes matching algorithms to link users. The premium membership offers additional features to enhance the matching process, while the free membership provides access to the basic features.)	Application is provided to end-users for free. Add-ons, special features and premium products are sold to a smaller fraction of the user base (also freemium) while the basic features remain free of cost.	• Lower / no barriers for users to try-out app • Reach of larger user base Network-effects come into action more quickly	• Only small user-base is generating revenues Low engagement from users as no investment was made and app can get quickly discarded
3) Subscriptions (Example: <u>Netflix</u> Netflix is a Subscription Video on Demand (SVOD) application in which users pay a monthly fee to gain access to a large library of media content, available to them at any time and from any location.	User must obtain a subscription, which is typically paid for on a recurring basis (monthly or yearly) to access the full range of features. If the subscription is not renewed, the features will no longer be available.	• Steady stream of revenues • Lower entry-barrier of users as with paid apps (trial-period) • High Customer Loyalty	• Higher churn rates due to flexible cancellation policies • Constant need to remain relevant and improve product to keep user base

4) In-app advertising

(Example: Spotify)

Spotify is a digital music streaming platform that offers a free version with advertisements interspersed between the streamed content, as well as a premium version that eliminates advertisements.

Application is generally for free for the advertised version. Third parties integrate their advertisement which generates revenues for the application providers. Often premium versions allow the exclusion of advertised content.

- Nearly user-independent income
- Different ways to integrate advertisement
- Lower / no barriers for users to try-out app
- Reach of larger user base

- Negative effect on User-Experience
- Users are needed anyway to justify paid advertisement
- Depending on displayed advertisement danger of losing customers to the advertisers

Appendix 6

App store	Fixed Fees	Commission
Apple App Store	Annually 299\$ Apple Enterprise Program	- 30% for apps and in-app products - Except 15% for subscriptions after 12 months
Google Play Store	One-time 25\$ Developer Account	- 30% for apps and in-app products - Except 15% for subscriptions after 12 months