INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Recomendação de Preços Baseada em Estratégias

INÊS MACHADO CARVALHO
Junho de 2021

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Strategy Based Price Recommendation

Inês Machado Carvalho

Dissertation to obtain the master's degree in
Informatics Engineering, Specialization in
Computer Systems

Supervisor: Nuno Ferreira
Co-Supervisor: Bruno da Silva
Company Supervisor: Luís Guimarães

Porto, June 2021

Abstract

The retail area is the one that suffers the most from the economic crisis, constantly facing challenges related to this volatile market and having to be increasingly vigilant to maintain margins. The large increase in retail and the different channels available allows its consumers to compare prices between different retailers, thus negatively affecting the retailer's results, causing the retailer to think differently about how to do price optimization.

In an attempt to overcome those challenges, retailers turn to a new reality, technology, in order to rely on tools that allow them to reposition themselves in relation to the consumers' perception of their price and value, and thus gaining an advantage over its competitors. These tools are centered on price strategies based on mathematical and statistical methods that allow a consistent pricing approach.

However, the current tools on the market do not fulfill all the current needs of retailers. In this sense, the solution presented throughout the thesis focuses on the analysis, design and development of a highly configurable and scalable solution capable of supporting a strategic and unified approach to price and markdown initiatives. So that it can be integrated with Oracle Retail solutions or other necessary systems. Being the main objective, closing the existing gap by providing retailers with a pricing solution capable of strategizing and enabling retailers to maximize profit.

To achieve these goals, a research and study was carried out in order to better understand the retail area and the commercial and technical challenges that retailers face in the pricing area. An assessment was also made of possible pricing approaches and existing strategies that could be put into practice.

Allied to this, a study was also carried out on Oracle Retail solutions and technologies to which the solution must be integrated, a study of existing solutions on the market, as well as the best pricing practices in the retail sector. That said, and based on the study carried out, a set of functionalities for a minimum viable product was defined and the requirements were identified, which served as support for the design of the solution.

Based on the proposed architectural design, a solution capable of managing strategies and recommending prices based on predefined targets was developed. All analysis, design and development is described throughout the thesis.

To test and validate the developed solution, it was exposed to several tests, which allowed us to collect useful information and possible improvements to be made in future work, as well as verifying its correct functioning. In general, after the evaluation, it is possible to conclude that the solution met all the objectives defined at the beginning of the thesis.

Keywords: Retail, Pricing Strategies, Oracle Retail, ADF, Computer Applications

Resumo

O mercado dos retalhistas é o que mais sofre com a crise económica, estando constantemente a enfrentar desafios relacionados com este mercado volátil e tendo que estar cada vez mais atentos para conseguir manter as margens. O grande aumento no retalho e dos diferentes canais disponíveis permite que os seus consumidores possam compara preços entre diferentes retalhistas, afetando assim negativamente os resultados do retalhista, fazendo com que este pense de forma diferente sobre como fazer a otimização de preço.

Na tentativa de superar estes desafios os retalhistas voltam-se para uma nova realidade, a tecnologia, com o objetivo de se apoiarem em ferramentas que lhes permita reposicionarem-se em relação à perceção dos consumidores sobre seu preço e valor, e ganhando assim, vantagem sobre os seus concorrentes. Ferramentas estas que se baseiam em estratégias de preço com base em métodos matemáticos e estatísticos que permitem uma abordagem de preços consistente.

No entanto, as atuais ferramentas no mercado não preenchem todas as necessidades atuais dos retalhistas. Neste sentido, a solução apresentada ao longo da tese centra-se na análise, designe e desenvolvimento de uma solução altamente configurável e escalável capaz de apoiar uma abordagem estratégica e unificada para iniciativas de preço e *markdowns*. De maneira que possa ser integrada com as soluções da Oracle Retail ou outros sistemas necessários. Com o principal objetivo de fechar a lacuna existente ao fornecer aos retalhistas uma solução de atribuição de preços capaz de criar estratégias e que permita aos retalhistas uma maximização de lucro.

De forma a alcançar estes objetivos, foi feita uma pesquisa e estudo de forma a melhor entender a área do retalho e os desafios comerciais e técnicos que os retalhistas enfrentam na área de atribuição de preços. Foi feita ainda uma avaliação a possíveis abordagens de preço e a estratégias existentes que possam ser postas em prática. Aliado a isto, foi ainda feito um estudo das soluções e tecnologias Oracle Retail a qual a solução deve ser integrada, um estudo de soluções existentes no mercado, bem como as melhores práticas de preços do setor de retalho. Posto isto, e com base no estudo efetuado foi definido um conjunto de funcionalidade para um produto mínimo viável e identificaram-se os requisitos, o que serviu de suporte para o designe da solução.

Com base na conceção arquitetural proposta, procedeu-se ao desenvolvimento da solução capaz de gerir estratégias e recomendar preços com base em targets predefinidos. Toda a análise, conceção e desenvolvimento é descrito ao longo do teste. Por forma a testar e validar a plataforma desenvolvida, está foi exposta a vários testes, que permitiu recolher informação útil e possíveis melhorias a fazer num trabalho futuro, permitiu ainda verificar o seu correto funcionamento. De forma geral, após a avaliação é possível concluir que a solução deu resposta a todos os objetivos definidos no início da tese.

Palavras-chave: Retalho, Estratégias de Preço, Oracle Retail, ADF, Aplicações informáticas

Acknowledgment

The development of this Solution/Thesis involved the collaboration of several people who supported me and who contributed to the success of this dissertation, which I must thank:

To my Advisor Professor Nuno Ferreira and Co-advisor Professor Bruno da Silva, for their guidance and all their availability during the different stages of the project. We had many meetings throughout the development of the thesis that kept me afloat and on the right path, thus managing to keep the defined deadlines. Thank you for all constructive criticism and suggestions. Therefore, I thank and owe a lot of the work and success of this dissertation.

To Luís Guimarães, for his availability, patience, and support, as well as all the knowledge transmitted over the several months of work. For the opportunity and confidence placed in the development of the dissertation. All the support given was tireless, being always available to guide me and clarify any doubts. Thank you for having provided me with all possible means to feel integrated and for all the moments of good mood.

To Retail Consult for having given me the opportunity to carry out the dissertation project in their company. And for providing all the necessary means for my learning and for the development of the thesis, providing my growth.

Finally, a special thank you to my family, especially my mother, my sister and Tiago for all the support, encouragement, and motivation along this path. For all your patience during the most difficult times. Your love and support allowed me to get to where I am today.

A special thanks to my father who, despite not being with me anymore, I believe he would be proud of this entire journey.

"Your life is yours alone. Rise up and live it."

-Terry Goodkind

Contents

List of Figures

List of Tables

Acronyms and Symbols

List of Acronyms

ADF	Application Development Framework
CI	Consciousness Index
CPQ	Index Configure Price Quote
ERP	Enterprise Resource Planning
IBM	International Business Machines Corporation
ISEP	*Instituto Superior de Engenharia do Porto*
IT	Information Technology
JET	JavaScript Extension Toolkit
KVI	Key Value Items
MOM	Merchandising Operations Management
MRP	Material Requirements Planning
MVC	Model–view–controller
MVVM	Model–view–viewmodel
OEM	Original Equipment Manufacturers
ORMS	Oracle Retail Merchandising System
ORPM	Oracle Retail Price Management
PO	Purchased Order
PL/SQL	Procedural Language/Structured Query Language
QEF	Quantitative Evaluation Framework
RC	Ratio of Consciousness
RRP	Recommended Retail Price
RTV	Returns to Vender
SKU	Stock Keeping Unit

SQL	Structured Query Language
UPC	Universal Product Code
VAT	Value Added Tax
VFP	Vendor-Financed Promotion
VFD	Vendor-Financed Discount
XED	Cross Price Elasticity of Demand

List of Symbols

€	Euro
λ	Lambda
%	Percentage

1 Introduction

The Strategy Based Price Recommendation Solution is a project carried out at Retail Consult in the retail software area, mainly focused on pricing strategies, for suggesting regular selling prices. For this, it takes into account factors such as promotional and/or clearance pricing, marketing initiatives, and other internal and external pricing related drivers.

This chapter intends to introduce the theme addressed in the dissertation. It starts by contextualizing and framing the problem that it is proposed to solve, then the objectives are described from which it is possible to guide the elaboration of the project. Finally, the structure of the dissertation is presented.

1.1 Context

Pricing is an important commercial problem faced by retailers, with the growth of the retail sector and its constant change, there is an increasing need to be aware of the new trends that the market imposes.

Retail Consult (RC) is an Oracle partner specializing in digital solutions for the retail industry, with a strong focus on Oracle Retail solutions and related technologies - these solutions address a wide range of retail business needs. Through a partnership with Oracle, it provides software solutions, systems integration and process reengineering to its customers, in order to enable a constant evolution of the business.

Retail Consult has several clients in different segments of retail such as food, fashion, telecommunications, among others that support commerce, logistics and management for both small companies and global retailers such as Selfridges, Desigual and Sonae. These technologies are part of the Oracle Retail Management software solution that allows you to define, maintain and review price changes, sales and promotions (Mckibbon, 2009).

Over the years, this business area has undergone a great evolution, from increasingly economic prices to easy access to retail by the consumer, and now with the impact of the pandemic there

is a great economic uncertainty. While retailers are struggling to maintain margins and market share, we are seeing an increase in the number of retailers that are looking to use pricing strategies as one of the key tools to reposition themselves, looking to gain an advantage over competitors. This makes it necessary to employ pricing strategies to generate incremental revenue benefits, designing new aspects that allow retail analytical optimization to prescribe unified price optimization, promotions, and discount recommendations.

The project carried out is directly related to the various modules of Oracle Retail. As Retail Consult is a company specialized in technology solutions for retail, it has noticed the lack of software to monitor the changing consumption habits. Thus, the work carried out, will focus on the analysis, evaluation, and resolution of these areas for the realization of an automated, unifying pricing solution. Through the creation of a highly configurable and scalable prototype that can be integrated with both Oracle Retail base modules and other necessary systems, such a solution has the potential to be a key differentiator.

1.2 Problem

Over the years, retailers, business consultants and systems' integrators have adapted to the specifics of the market in which they operate through strategies and business expansion and responding in an ever-faster way to the constant changes of the consumer. From the extensive number of existing features of Oracle's retail solution, there is a highly configurable, strategy-based pricing solution that suggests and assists with pricing decisions called Oracle Retail Price Management.

"Pricing is one of the most important strategic areas retailers use to gain market advantage. The challenge retailers face in making the right pricing decisions often stems from inaccurate, fragmented or sometimes too much information to find the right actions" (Oracle and its affiliates, 2020).

Of all the retail components, the price is the one that most quickly affects the competitiveness, the number of sales and the profit of the companies. "Oracle Retail Price Management is designed for retailers who require complete visibility and control in order to streamline pricing decisions across all selling channels throughout the item lifecycle. Oracle's solution provides retailers with a rules-based pricing strategy and execution engine to ensure that target margin or a competitive position is achieved in line with corporate objectives" (Oracle and its affiliates, 2020).

However, there is currently a gap between current retailers' needs and what most software vendors have to offer in terms of pricing software. Despite all the features that Oracle Retail Price Management provides for executing and maintaining price events there is still a need and an opportunity to enhance the capabilities by adding intelligence layers and automation. Retailers are experiencing a new reality that is creating new challenges the proposed solution aims to solve these challenges.

1.2.1 The Retail Business

Retail is the main direct sales channel to the final consumer, in essence, it is the process of selling goods and services directly to consumers in small quantities for their final use. Retailing is a modality that occurs for items such as food, clothing, electronics, hygiene products, household items, utensils, among others. We can consider that any product or service that is sold directly to end consumers is defined as retail. Thus, retailing involves more than selling tangible products. The purchase of a service such as haircut or dry cleaning is also a retail transaction (Cox and Brittain, 2004).

Merchants or sellers are called retailers. The entities involved in retail include manufacturers, wholesalers, retailers and, finally, the final consumers themselves.

The following example aims to help understanding the retail concept: Teresa wanted to buy a chocolate. She went to the grocery store and bought one for herself. In this case Teresa is the final consumer who went to a place (grocery store) and bought a chocolate (quantity one) for her.

Every day, millions of consumers buy consumer goods and services. The place where products or services are sold or distributed can be made in many establishments. Some are: malls, hypermarkets, construction services, pharmacies, dealerships, clothing, furniture and decoration stores, gas stations, electronics stores, bookstores, and several other locations that are part of the retail trade. But there are still other forms of retail sale, such as those that take place over the internet (e-commerce) or over the phone, for example. The retailing industry is one of the largest and most diversified industries in the world, and several retailers rank among the biggest corporations in the world (Dekimpe, 2020).

Retail works through mark-up or profit margin, which consists of increasing the final price to cover costs and make a profit. Retailers buy the goods at a cost price, and then add the cost of labor, equipment and distribution together with the desired profit margin and sell at a higher price.

Being considered a very complex and highly competitive area, this sector is dependent on several factors from economic aspects, consumer preferences and sales and purchase trends.

The supply chain is the steps taken by a company and its suppliers to deliver goods or services to the consumer. These steps range from supplying materials to transforming them into finished products. This chain involves several different entities, resources and activities necessary for the production and delivery of the final product/service.

In the image below we can see a reference model for the retail supply chain (Ayers and Odegaard, 2017).

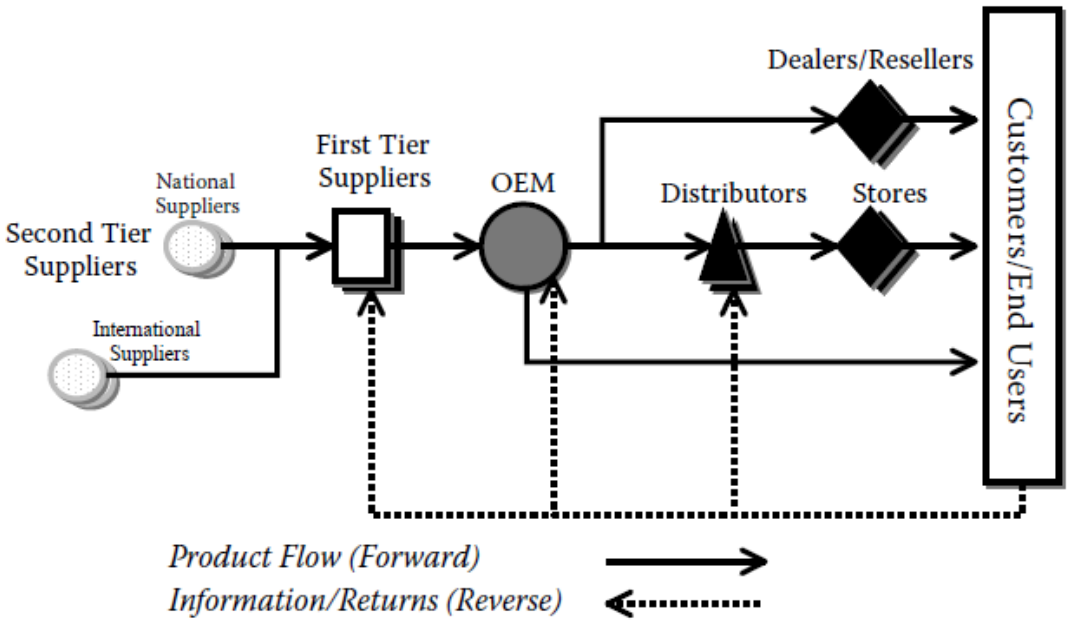

Figure 1 - Retail Supply Chain Model

The simplified retail supply chain model shown in the figure include the following:

1. Customers or end-users
2. Retailers
3. Distributors
4. Original equipment manufacturers (OEMs)
5. First-tier suppliers
6. Second-tier suppliers
7. Service providers

Retail supply chains are more than stores and technically begin with the original sourcing of raw materials for product manufacture. There are exceptions to this supply chain, but mostly the importance of the supply chain is to increase the operations performance to minimize the waste or loss to the business and add value to the product. An efficient supply chain provides a diversity of product mix at affordable prices to the customers.

1.2.2 Challenges of a Retailer

Consumers have been changing the way they see the world of shopping and as a consequence the retail world is undergoing these changes and facing increasingly challenges.

“The way that consumers make purchasing decisions has dramatically altered: they stand in stores, using their smartphones to compare prices and product reviews; family and friends instantly weigh in on shopping decisions via social media; and when they’re ready to buy, an

ever growing list of online retailers deliver products directly to them, sometimes on the same day" (MacKenzie, Meyer and Noble, 2013).

This means that there is an increasing need to improve efficiency and reduce resource and operational costs in order to meet the needs of consumers and keep them interested, maintain a viable price and ensure that they are not behind the competition. In the U.S., 59% of the customers will walk away after several bad experiences, 17% after just one bad experience and 32% of all customers would stop doing business with a brand they loved after one bad experience (PwC, 2018).

"With inflation accelerating in major markets, governments making shifts in monetary and fiscal policies, and most of the emerging markets experiencing significant currency depreciation, the global economy will slow down soon. For retailers, this change will mean slower consumer spending growth, higher consumer prices, and disrupted global supply chains" (Kirasirova and Biryukov, 2019).

Shopping in a physical store is not as rewarding as it used to be, over the past decade, consumers have changed the way they buy and look for manufactured goods. Nowadays, customers prefer to stay in the shelter of their homes where they can choose the exact product and still compare prices. So, with the last changes and in today's economy with the appearance of the pandemic, retailers are at a turning point and there is a growing diversification of challenges. Since the beginning of COVID-19, more and more retailers have closed their businesses. Thus, there is an increasing need to have an in-depth knowledge of the needs of customers and their consumption as they become more demanding.

Progressively more retailers are turning to technologies to obtain better results and overcome the biggest and main challenges of retail in order to provide better customer experiences. According to Retail's Consult internal documentation, some of the main obstacles and challenges that retail domain addresses today are identified and described below.

Consumers are spending less

Year after year consumers have changed their preferences for buying goods for services. Spending more on vacation, activity outside the home, experience and entertainment and in return they spend less on manufactured goods. The arrival of the digital age and the use of communication technologies, such as smartphones, social networks and online sites, changed the way consumers make purchasing decisions and, consequently influenced the tourism industry (Chen *et al.*, 2015). In contrast, with the appearance of COVID-19, according to a study (Ernst & Young, 2020) by the EY Future Consumer Index carried out in Portugal, the Portuguese have spent more on delivery services, canned goods and fresh food. Between the need to spend more and pay more for basic needs, which has also been increasing, and the purchase of services there is a lesser tendency to spend on other items. This is causing a decrease in spending and a greater difficulty for retailers working in non-market areas related to basic needs which, in return, are obliged to increase prices to support higher costs.

Investment in private labels and cheaper brands

There are more private labels and cheaper brands on the market than ever before, that offer the customer a low-cost brand alternative (Kumar and Steenkamp, 2007). Due to the economic difficulty in which we find ourselves, more and more consumers exchange national brands for private labels or cheaper brands, being willing to exchange quality for price. During the economic crisis due to the high availability of these products and the low price at which they are available, consumers saw this as an advantage in times of difficulty and as a good way to save in times of great uncertainty. A study (Gala and Patil, 2013) has shown that customers see private label as a local product and most prefer private label products due to suggestions from people close to them or for a better offer. This appearance of different brands brought greater competition for national brands, with a need to encourage innovation in order to remain competitive, providing certain challenges such as branding, advertising, stock and promotions and price management in relation to other brands.

Switching stores and brands

With the growth of e-commerce and the digital world, consumers have the possibility to compare products in relation to price benefit. And with the increasing focus on price as the main influencer in the purchase, the choice of store and brand they buy tends to vary considerably, making consumers increasingly disloyal to their stores. According to a study done in twenty-one grocery stores for three years in Missouri, 57.2 percent of shopping trips where switches (Popkowski Leszczyc and Timmermans, 1997). In addition, they are constantly changing from offline to online stores where they can interact as they wish, being able to see the product in the store for later purchase online. This constant exchange makes retaining loyal consumers a constant challenge.

Promotion sensitivity

Promotional sensitivity defines the limits to which the consumer is willing to pay for a product or service and prefers. Promotions have become increasingly relevant as most consumers tend to reduce spending and save with fear and uncertainty about their financial stability or with the aim of saving for other purposes. This demand to save makes the consumer look for financial advantages like promotions (Pacheco and Rahman, 2015). This digital age provides an easy way to inform the customer about promotions and therefore increasing the number of consumers who are sensitive to promotions.

Contraction of supply chains

Today's supply chains are increasingly complex and there is a need for better risk and cost management. Supply chains need global resilience, and the lack of flexibility inhibits the ability to meet customer demands. COVID-19 interrupted supply chains worldwide and before that the economic crisis was already affecting supply chains. From higher costs of storage and transportation, high costs of IT systems, new commercial agreements and employee health and safety, all these operations, which for different reasons become more expensive, have an influence on the final price and finally on the margin.

This project consists of the evaluation of potential price approaches and the implementation of a solution that allows to fill this gap focusing on monitoring the consumption habits and prices of competitors in order to react and obtain a more predictable and profitable outcome. Thus, the solution must support a strategic and unified approach to price and markdown initiatives that is scalable and allows multiple configurations.

1.3 Goals and Objectives

Within the context referred to in the previous section, the goals of this dissertation are twofold. First, is to analyze, design and build the technical structure of a pricing solution and, second, to define resources for the implementation of that pricing solution. This solution must be fully integrated with the Oracle Retail solution stack and be configurable and scalable in a way that will potentially enable it to interact with any other required external systems.

To achieve the intended goals, the following objectives are defined:

1. Research based on pricing challenges and understanding of the retail domain and Oracle retail platform;
2. Study and evaluate possible pricing approaches and interact with stakeholders to determine what are the best approaches to solve the problem addressed;
3. Define a minimum set of specifications for a minimum viable product;
4. Define a set of requirements for the solution in terms of priority that are enforceable during the internship period and the solution roadmap;
5. Develop the solution prototype that must be configurable and scalable.

1.4 Research Method

To carry out the dissertation, the Design Science Research (DSR) (Kuechler and Vaishnavi, 2008) research method was used, with the objective of solving problems that do not yet have a solution or that the existing solution is not the most effective. This has as its main focus the study of the problem and the identification of opportunities that can count as a contribution to the studied problem.

There are several variants of the DSR, the variant shown in the figure below (Kuechler and Vaishnavi, 2008), proposed by Kuechler and Vaishnavi, was used to carry out the thesis. This proposal is divided into five steps, Awareness of Problem, Suggestion, Development, Evaluation, and Conclusion. In addition, the proposal presented follows the agile model because, according to Kuechler and Vaishnavi, it has a cyclical approach where the phases presented must be repeated for refinement and improvements until the solution presented is satisfactory.

The study of the topic addressed, and the implementation of the solution were followed by the 5 phases described below (Kuechler and Vaishnavi, 2008):

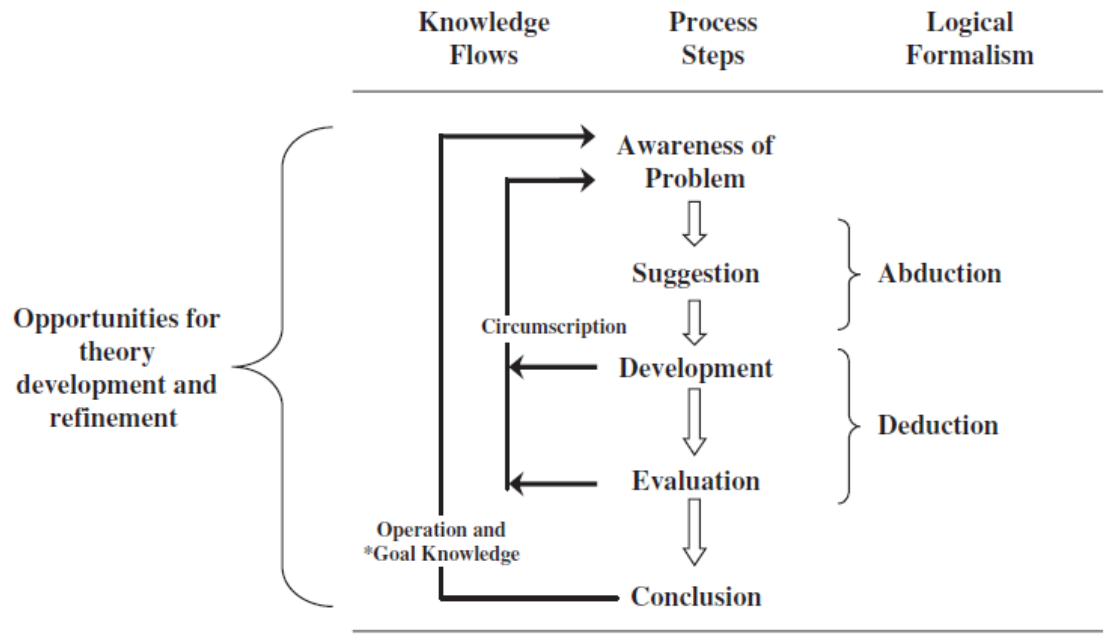

Figure 2 - Design Science Research Cycle

1) Awareness of Problem

This phase consists of highlighting the problematic situation of the business, understanding the relevance of the problem, and analyzing the current solutions in order to identify opportunities and thus substantiate the value of the solution. For this, a literature review was made on the topic addressed, some interviews and meetings with interested people and finally an analysis of the data.

2) Suggestion

In this phase, the criteria, and specific requirements that the solution must meet for the defined problem are defined, it is also the phase where suggestions for developing the solution must be presented. In addition, it is necessary to have knowledge of methods, technologies and theories that can help in defining the objectives. For this, the premises and requirements for the construction were explained, models and diagrams were built, and different design hypotheses were studied, where each one is accompanied by reasons that justify or exclude the attempt.

3) Development

In this phase, development is started based on the requirements and architecture defined in the previous phase, where processes, methods and technologies are built and applied to create the solution that solves the problem identified in the first phase. This phase was developed in the context of an internship at Retail Consult, by a team responsible for developing the prototype.

4) Evaluation

In this phase, an evaluation of the solution is made based on the defined objectives and requirements, for that a comparison is made and the results were evidenced in relation to the metrics and objectives initially defined. For the evaluation, the Quantitative Evaluation Framework (QEF) method was used, a quantitative method that aims to validate the quality of the solution (Escudeiro and Bidarra, 2008).

5) Conclusion

Throughout this process it is normal that in some phases, mainly in the development and evaluation, certain questions arise and that it is necessary to reformulate. For this, the cycle, composed of the phases described in this chapter, is started again to correct, and develop what was not validated in previous cycles. In this phase, all learning throughout the work is also concluded and synthesized and the results obtained with the studied problem are justified.

1.5 Structure

This report is divided into 6 sections, also including the acknowledgments, the executive summary, the abbreviations, and the glossary at the beginning of the document and the bibliography and appendices at the end.

The first chapter of the document, Introduction, presents the context and description of the respective work, as well as the problem to be investigated and solved and its main objectives and expected results. The structure of the dissertation is also presented.

The second section, State of the Art, presents the bibliographic review in which several topics are approached with special relevance to the work presented and which appear as a contribution to the definition of requirements. Thus, it begins with an approach to research methods, then the company and the retail world are presented. Then an approach to price management is made, from retail price to price strategies. Finally, different Oracle Retail technologies and solutions are addressed as well as agile development methodologies.

The third section, Analysis and Design, presents the domain model, solution requirements and restrictions. It also presents an analysis and comparison of various architecture alternatives of the solution to be implemented, as well as possible technologies to be used. Finally, an in-depth approach is made to the chosen design proposal, presenting the architecture and description of implementation.

The fourth section, Development and Testing, the solution development phase is exposed.

The fifth section, Solution Assessment, presents the results obtained taking into account the achievement of the proposed objectives, and the discussion of those results.

The sixth and final section, Conclusions and Future Work, presents the conclusion of the work done, and discusses future work.

2 State of The Art

This chapter intends to introduce the retail area and present the current scientific and conceptual approaches of the study area explored in this project and also the current technology and tools.

For the review, the Systematic Literature Review (SLR) (Systematic reviews and meta-analyses: a step-by-step guide, 2013) methodology was used, which consists of researching, synthesizing and evaluating the bibliographic collection to respond to a specific query of a given study with well-defined criteria.

Bearing in mind that the main focus of this dissertation is related to the retail area, first of all the general concepts of the retail domain and its entire life cycle are introduced and then the main difficulties for retailers in recent years are presented. Then, price management is covered where the main concepts of price, promotion, markdowns and price in the retail area are explained and finally existing price strategies are discussed.

Afterwards the two main Oracle solutions that will be involved in the project and will be the main basis of the solution are presented, Oracle Retail Merchandising System and Oracle Retail Price Management, where some of its features and characteristics are introduced. Finally, two possible frameworks to be used for software development are discussed, with a small comparison between the two.

The following chapter presents the state of the art and is intended to support the project.

2.1 Pricing Contextualization and Strategies

For a better notion of an area as complex as price, this subchapter gives a brief introduction to price and how it plays a decisive role in retail success. In addition, a brief contextualization is made of some existing strategies that may be relevant to a best practice and benefit the retailer.

2.1.1 Retail Pricing

The price is one of the most important marketing variables to consider, it consists of the value, in cash, paid in exchange for a service or good. There are different price definitions, but in the retail setting the price of an item or service represents the company's attempt to capture part of the value of the profit it makes. This is the only variable that allows improving a company's profit since small differences in the price of a product or service can have a big impact on consumer satisfaction, which determines sales performance and consumer loyalty.

Pricing is a complicated process that depends on different factors, from production costs, distribution, goals, competition, and brand positioning, all to generate profit and achieve the company's goals without suffering losses, but also taking into account interest, satisfaction and customer budget.

Therefore, pricing does not consist of mere calculations, it is also necessary to assess the market, positioning and the behavior of end consumers. For this, it is necessary to know all the factors that can influence the price and the possible pricing strategy to achieve the desired objectives. These strategies define pricing practices that enable retailers to achieve better results and increase customer loyalty. But retailers must look beyond prices to diagnose flaws in their pricing strategy. A comprehensive pricing strategy is made up of several layers that form the strategic price pyramid, creating a basis for pricing that maximizes profits over time.

The strategic price pyramid below consists of the following layers (Hogan and Nagle, 2005):

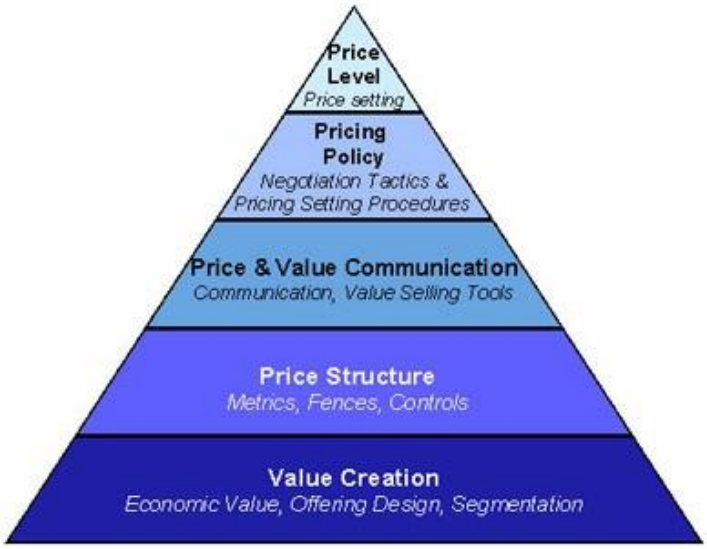

Figure 3 - Strategic Price Pyramid

- Value Creation - At the base of the pyramid is the definition and creation of significant values for the different customers for which they are willing to pay for

- Price Structure - Pricing should be defined for the different costumer segment instead of a product by creating a structure associated with the value;
- Price and Value communication - Communication strategy that discloses the value and price of the product based on the values for the different segments and that in a simplified way demonstrates the advantages and benefits of the same;
- Pricing Policy - A specific pricing and promotion policy must be followed using the price proactively to influence the customer;
- Price Level - Define the appropriate price according to the segmentation and the defined value, this price must be regularly analyzed and changed.

For the development of a price structure it is necessary to understand how goods or services create value for a consumer (Hogan and Nagle, 2005). The pricing strategy is complex, so it is important to know the business and the objectives well to be able to align the price with the different factors.

2.1.2 Price Strategies

The following is a set of possible strategies, which can solve some of the problems raised by the challenges presented in the chapter 1.2.2 and which are relevant in today's market environment. Thus, allowing retailers to use pricing strategies by applying statistical and mathematical models.

2.1.2.1 Identifying Key Value Items

Key Value Items (KVI) are best-selling and researched items that boost the brand image, whose prices consumers tend to decorate. For example, Levis t-shirts that everyone wears are considered a KVI because they boost the brand. Typically, retailers use KVIs to better manage item pricing decisions.

The items that will belong to the list of KVIs are identified using these three types of methods (BenMark *et al.*, 2017):

- **Transaction and basket data** - analyze and rank an item based on the number of sales;
- **Shopper price-perception data** - identify the categories and items that most drive value perception;
- **Merchant judgment** - Experienced merchants can review and add strategic items with high degree of competitive intensity.

There are four types of KVIs:

- **Perceived value drivers** - Commonly used and memorable items which are normally popular with consumers;
- **Drivers of assortment perception -** Items in a list showing a perspective on what they should buy which shapes customers' purchasing decisions;
- **Traffic drivers -** High-demand products that lead to mora shopping trips, for example, toilet paper and diapers;
- **Basket drivers** - Low-velocity items which lead to buying other products, for example, the purchase of *francesinha* sauce will likely drive the purchase of cheese, loaf bread and meat.

KVIs are identified based on financial performance, sales, and competitive analysis. To optimize the perception of value, a retailer must set the price of KVIs more accurately in relation to the competition (Kilroy, MacKenzie and Manacek, 2015). Without the right approach to identifying and maintaining a list of KVIs, retailers are at risk of focusing on the wrong items.

2.1.2.2 Targeting Pricing

The definition of a price directly affects the results and sales of a company, when pricing an item, the most common method would be to add production costs with the desired margin, but currently this method is increasingly contestable. This is because, this type of pricing does not take into account demand and price-cost-volume ratios and in addition it ignores competing prices (Al-Hussari, 2006).

The target price is a method that has been used more and more and that allows to previously analyze the prices and the target market. Basically, this method allows the retailer to set different prices for different market segments. An example of this method is air travel that is usually never presented to customers at the same price and a seat for the same trip can be sold at different prices.

This method consists of identifying and dividing the different customers, products, and geographic areas into segments, thus being able to define prices for each of these segments and making the price strategy clearer and more profitable. In addition to cost margins and competition, the essential is the value of the product and its correct communication to the customer, for this it is essential to know the different customer segments.

There are three different types of targeting:

- **Customer Groups** - is one of the most used segmentations and consists of identifying the target audience. This group can be created based on ages, gender, values, consumption, education, ethnicity, attitudes, among others. Basically, it consists of dividing a group of people based on demographic behavior or classification;
- **Item Categories** - comprises dividing items according to their characteristics, for example grouping by functionality, color, shape or physical or digital service;
- **Geographic Areas** - consists of segmenting and categorizing customers based on their geographic location, such as city, ZIP code or country. An example of this segmentation is a clothing company targeting customers who live in very cold areas to sell polar clothing.

This implies defining different prices using buyer analysis, market trends, purchase factors and demand for the various targets established on lists of KVIs, Lists of zones and other possible lists of targets. Based on the different data for each group, it is possible to calculate the ideal price (Mulvenna, 2020).

The price segmentation can be created from the study of the market and the analysis and study of the data of expenses and visits by the customers. Basically, the essential thing is to

understand the market and after implementing this method it will be possible to specify prices for each of the defined segments.

2.1.2.3 Scheduling Promotions

With the retail market growing and becoming increasingly virtual, there is great competitiveness on the part of the retailer, both in the physical store and in the digital store, and the launch of new promotions is one of the best ways to create brand awareness and encourage the purchase of certain items. Although the definition of the promotion seems like a simple step, planning the right promotion to implement is a very complicated process. A promotion cannot be defined and programmed in any way, it is necessary to be aware of the type of discount, which products will be affected, when this promotion will be stipulated, if there are no other promotions or advertisements that are superimposing and even if the demand will not be affected for other related items. All these needs make planning a promotion a challenging problem.

An example of planning based on data and computation is the Google patent (Cohen et al., 2020) that applies systems and methods that generate a schedule of promotion prices using the model of demand for multiple products and considering all the limitations mentioned above and more.

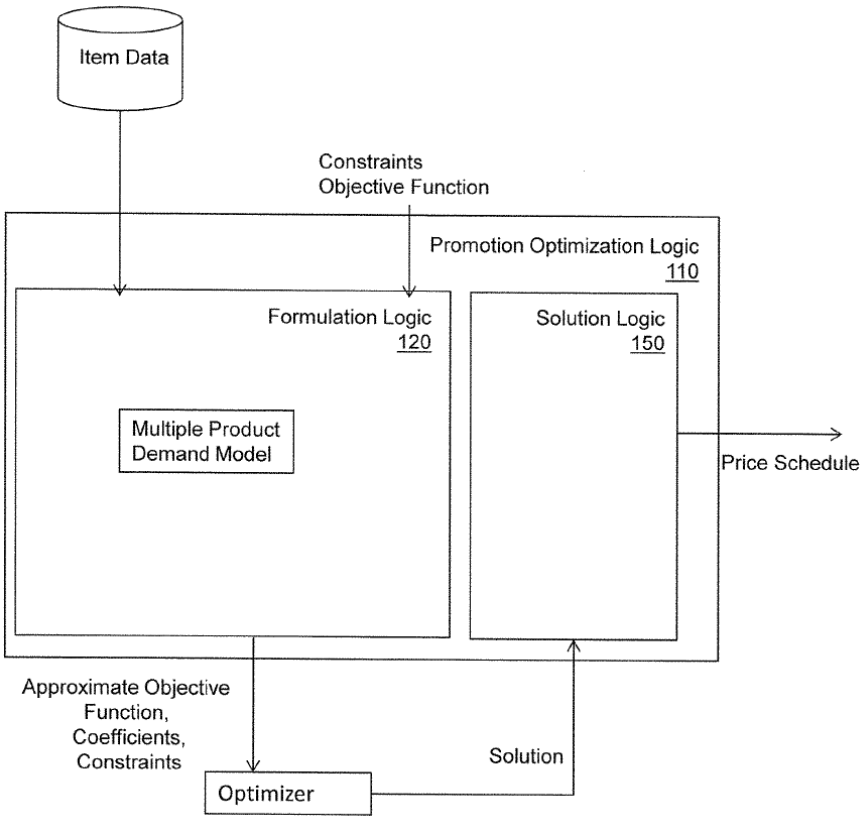

Figure 4 - An Embodiment of a System Associated with Promotion Price Scheduling

In the figure above (Cohen *et al.*, 2020) it is possible to see one of these systems that generates a promotion price schedule for a group of items based on a promotion optimization logic that controls a processor to perform various operations. The module used by the retailer will communicate with the system to request a table for items sold by the retailer based on the business rules and restrictions defined. For the insertion of item data, the system uses the retailer's price system. The promotion optimization logic comprises the formulation logic that formulates an approximate objective function based on the item data and the objective function that creates a price schedule for each item based on the target solution determined by the optimizer. In turn, the formulation logic includes the demand model for various products, which consists of a set of instructions that calculates the demand for an item based on previous prices and current prices for other items. In addition, the promotion optimization logic does the calculations that will be used by the optimizer to determine the price optimization schedule. Based on the target solution determined by the optimizer, the solution logic creates a price list for each item. And finally, the schedule promotion is communicated to the system used by the retailer (Cohen *et al.*, 2020).

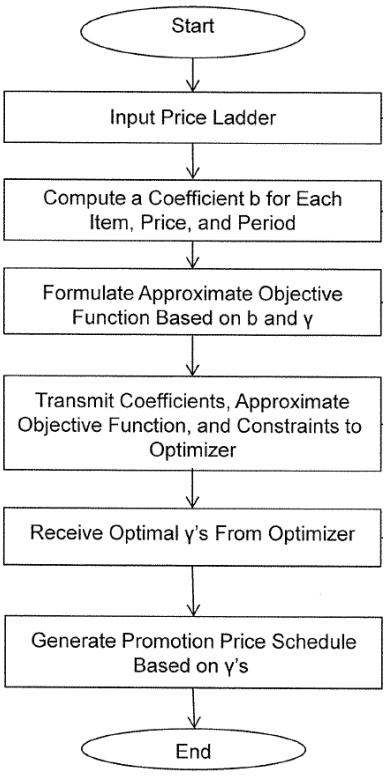

Figure 5 - Embodiment of a Method Associated with Promotion Price Scheduling

The figure above (Cohen *et al.*, 2020) depicts the method implemented in the previously described system. The first step of the method consists of inserting a price scale for each item that indicates a set of prices to be considered for the promotion. The item coefficients are then

calculated for each period in the price schedule and for each price on the previously defined scale. Then, based on the objective function, an approximate function is formulated that includes the products of the respective item coefficients and decision variables. In the next step, the method fills the data structures with the coefficients, the approximate function and the defined restrictions and sends these structures to the optimizer. The optimizer then sends the determined data structure that encodes the values of the item's decision variables. Finally, the method generates the promotion price schedule for each item based on the item's decision variables (Cohen *et al.*, 2020).

Based on solutions like the one presented above, it is possible to better define and schedule a promotion without the worry of overlapping promotions, violating implemented rules and restrictions.

2.1.2.4 Performing What-If Analysis

The What-If methodology is a technique that consists of the analysis and identification of risks and failures, capable of reviewing possible directions and providing different conditions. This technique must be done as a previous study of risk analysis and can be applied to different areas.

With the implementation of this type of analysis in the solution, it is possible to create and evaluate in an automated way different types of hypothetical forecasting scenarios that can identify and prevent losses in terms of price strategies, being able to forecast all the different scenarios. Thus, it is possible to make the best decisions based on the prediction of all the problems that each stage may face taking into account different causes involved and considering risks from different sources.

An example given by Golgarelli et al. (Golfarelli, Rizzi and Proli, 2006) of an What-If analysis was, “How would my profits change if I run a three for two promotion for one week on some products on sale?”, where the analysis would inspect all the possible scenarios and measure how the different variables would have an impact on the profit, thus allowing to decide the best scenario based on the objectives.

The What-If analysis allows to support all processes that involve the definition of prices and the possible impacts on decision making. These strategies can be realized through the creation of new scenarios by the user and by scenarios previously implemented in the solution.

2.1.2.5 Analyzing Item and Cross-Item Price Elasticity

Price elasticity is the concept used when measuring consumer sensitivity to a product. Basically, it measures a consumer's response or reaction to a price change for a certain product. For example, in a scenario of a situation of elasticity, if a well-known brand if juice is being sold at an affordable price, a consumer will probably take several to take advantage of the price, but with an upward price change, the reaction of the consumer to the price variation is negative, promoting a decrease in product demand.

In some cases, the amount of demand goes up or down a lot depending on the reaction to the price whereas for other products there is almost no reaction and there is practically no change in the level of demand. In cases where there is a big reaction on the part of consumers, it is called elasticity of demand.

$$Elasticity\ of\ Demand(XED) = \frac{\%\text{change in Quantity Demanded}}{\%\text{change in Price}}$$

The elasticity can be calculated by dividing the ratio of the percentage change in the quantity demanded and the ratio of the percentage change in the price (Mulugeta *et al.*, 2013).

Another concept of existing elasticity is the cross elasticity that measures the variation in the amount of demand in the face of the consumer reaction to another product that has suffered a price variation. For example, if there is a price change in the Coca-Cola brand drink, demand for the Pepsi drink may increase, thus there is a cross elasticity between the two products. In some cases, increasing or decreasing the price may be intended to encourage the sale of another complementary product, for example, decreasing the price of a bicycle will encourage the demand for this same product, but also for other complementary products, such as the helmet.

$$Cross\ Price\ Elasticity\ of\ Demand(XED) = \frac{\%\text{ in Demanded of Good X}}{\%\text{ in Price of Good Y}}$$

The cross elasticity can be calculated by dividing the percentage change in the quantity of demand for product X and the percentage change in the price of product Y. If the Cross Price Elasticity of Demand (XED) is greater than one, it is considered elastic; if it is less than one it is considered inelastic (Mulugeta *et al.*, 2013). From this calculation it is possible to identify the extent to which a product complements or replaces another.

In the image below (Gbadebo-Smith, 2018), it is possible to identify the differences between elasticity and inelasticity. In the graph on the left it is possible to see that with a small price change the demand is disproportionately greater, thus being an elastic demand, while in the graph on the right, with a high price change it is possible to observe a lower demand, thus being an inelastic demand.

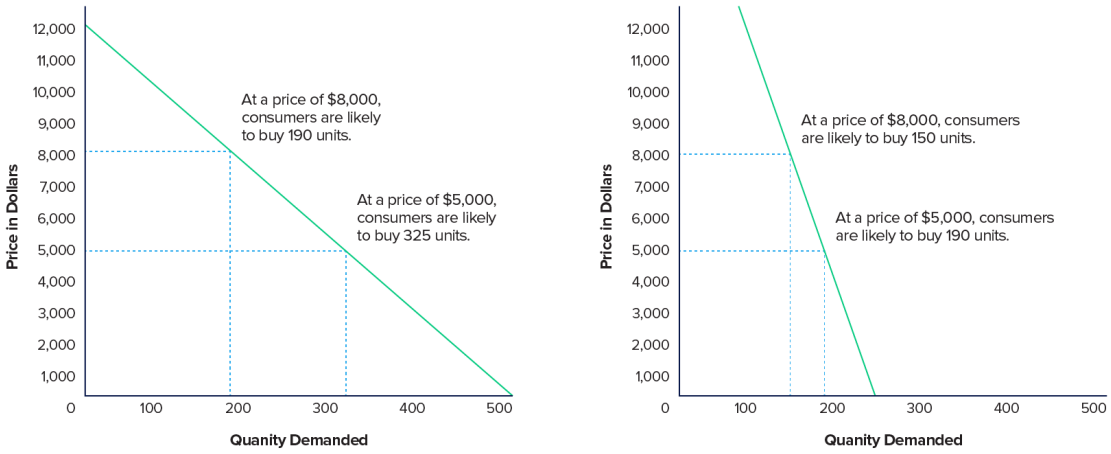

Figure 6 - Elastic vs. Inelastic Demand Curves
adapted from (Gbadebo-Smith, 2018)

From the analysis of the cross elasticity, it is possible to carry out specific strategies to create price relationships between different products, thus being able to have advantages in making pricing in a unified strategy for several products.

2.1.2.6 Defining and Executing Private Label Strategies

Private label consists of products or services that belong to the retailer or retail chain and that represents the brand or company and is normally responsible for carrying the brand identity. Most supermarkets today have private label products, that are characterized by good quality, but relatively lower prices. Private labels have advantages such as the possibility of having a higher profit and greater loyalty to the brand by consumers.

With better price management, retailers can have full control over costs and price, but for that it is also necessary to be aware of competitors' trends and prices.

According to the “private Label strategy” Book (Kumar and Steenkamp, 2007) there are more and more private labels that are known for their low prices and quality, where the difference in price between manufactured brands and private labels benefits the latter more. Therefore, having a price strategy that is sensitive to the price of the private label and the effect of the price of the manufacturer and the private label on each other is essential.

Based on the analysis of competitors private labels and with the proper comparison between the retailer's own brand, it is possible to create pricing strategies and determine the fair price for the private label product that offer advantages in competition with the manufacturers' brands.

2.2 Oracle Retail Solutions and Technologies

According to Oracle's definition, Enterprise Resource Planning (ERP) systems are all software used by organizations to manage business activities, from accounting, purchasing, projects, finance, supply-chain, risk and performance management, among many other processes. ERP systems unite a multitude of business processes and allow data to flow between them from a single data structure.

ERP systems are essential in the retail area to streamline all the processes that involve this complex business, allowing to obtain a strategic vision. Retail ERP systems provide different features such as:

- Inventory management
- Financial and accounting management
- Customer management
- Management of the supply chain
- Store management
- Warehouse management
- Omnichannel Management

In addition to the processes mentioned above, there are other features, which can be adopted according to the company's needs. These systems bring several benefits to the company such as greater efficiency. There are several ERP systems for flaps available on the market, the ones in the figure below (*The market of ERP systems in a year added 10%. In top five one change.*, 2019) being the main market leaders according to Gartner Analyst Data:

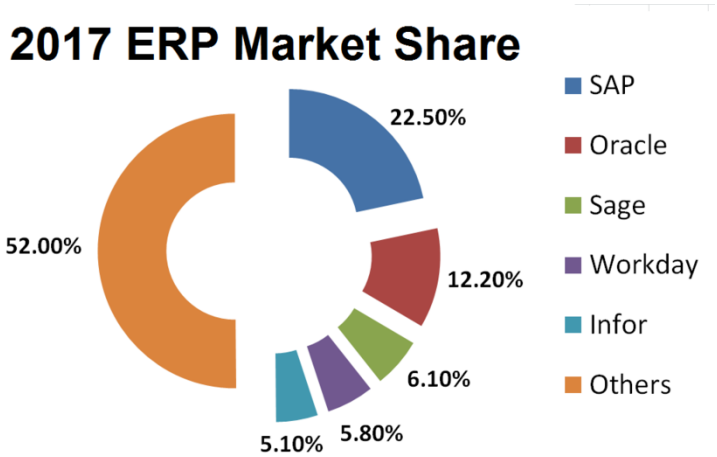

Figure 7 - ERP Market Leaders

Oracle Merchandising Operation Management (MOM) has several ERP businesses resources, the most important and relevant for this project can be classified into 3 categories: Merchandise Operation Management, Supply chain management and integration store operation. This chapter will focus on two Merchandise Operation Management applications, Oracle Retail

Merchandising System (ORMS) and Oracle Retail Price Management (ORPM), and then some of its competitors are presented.

2.2.1 Oracle Retail Merchandising System

The Oracle Retail Merchandising System, ORMS or simply RMS, is an integral module of Oracle Retail responsible for the execution of essential merchandising activities, which allows better management, control and execution of the crucial activities of the retail business.

RMS is the market-leading integrated merchandising solution for retailers that records and controls virtually all data in the retail enterprise and ensures data integrity across all integrated systems. This solution enables many of the best retailers in the world to better manage, control, and perform crucial day-to-day merchandising activities with ease by providing key functions such as item maintenance, inventory management, and replenishment. “This functionality provides easy access to the information that is crucial to the day-to-day merchandising activities within a retail organization, providing the ability to focus on key decisions that help achieve sales and profit targets” (Andrew, 2016).

RMS supports retailers by providing tools that allow you to maximize your business performance and improve operational efficiency. In addition, it has visibility of all the company's inventory and costs. “RMS streamlines business practices and unifies business systems across retail channels to better serve customers. Because RMS has been developed as a Web-based, scalable product, it fully supports the large volumes found in retail, leaving more time for retailers to concentrate on the bottom line” (Andrew, 2016).

Below is high level ORMS life cycle (Mishra, 2016):

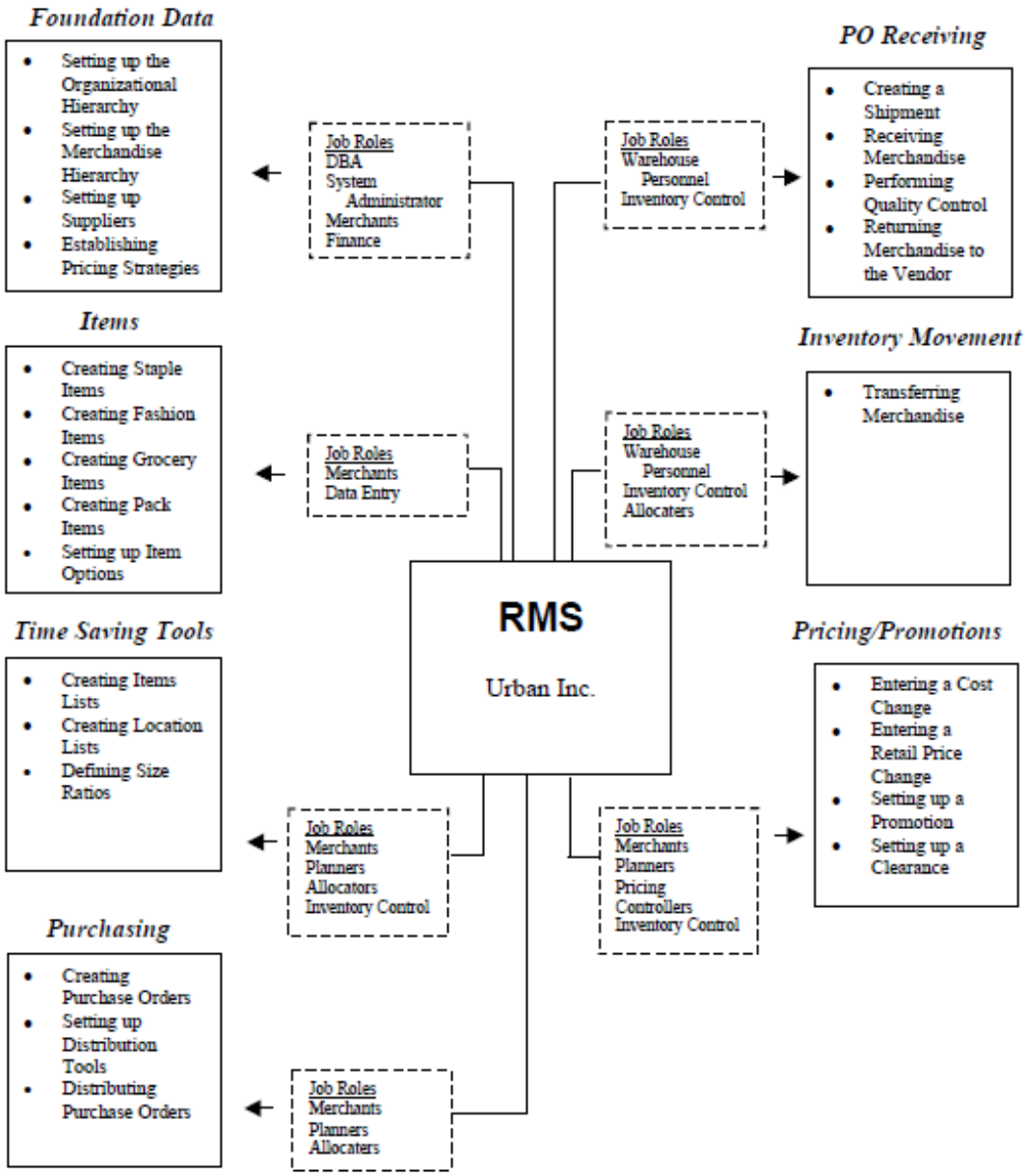

Figure 8 - RMS Life Cycle

The RMS has a set of functionalities with the objective of helping in tasks different from those according to the needs of the retail business. The features are as follows:

- Foundation Data - Merchandise Hierarchy, Organizational Hierarchy, Supplier and Partner Management
- Item Maintenance
- Purchase Orders
- Contracts
- Deals
- Cost Management
- Multiple Sets of Books

- Inventory Control
- Replenishment
- Franchise Management
- Stock Ledger
- Investment Buy

Next, each of these specific functional areas of the RMS identified above will be detailed.

2.2.1.1 Foundation Data

The foundation data is the basis for creating all the functionalities on which the fundamental business elements of RMS will work. Foundation data can be entered from specific systems through conversion processes (Andrew, 2016). For the use and control of the RMS this must be the first step. There are three types of Foundation Data: Organizational Hierarchy, Merchandise Hierarchy and Supplier and Partner Management.

Organizational Hierarchy

The Organizational Hierarchy allows defining the different relationships to create the operational reference structure of an organization/company. Depending on the company's needs, the preferred organizational structure can be created for the case. This structure will make it possible to assign the necessary responsibilities to each person at each level and also provide support for reports for the various levels of the organization. In the figure below we can see the different levels of the organizational hierarchy which will be described below (Mishra, 2016).

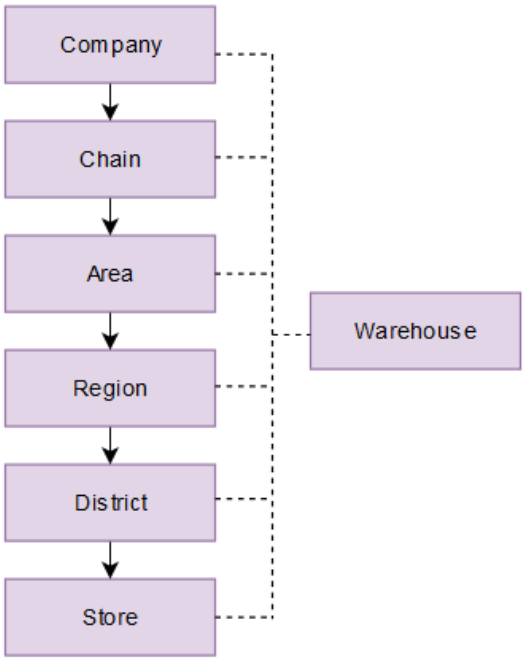

Figure 9 - RMS Organizational Hierarchy

At the top of the hierarchy chain, it is the Company. In RMS we can only define one Company per instance, and this is the only level that can be shared between organizational and merchandise hierarchies.

The next level is the Chain, this level usually serves to distinguish branding or trade names within the Company and is based on the needs of the Company. The Chain can be used to group different store formats, concepts and locations within the Company.

The Area is the level following the Chain, it can only belong to one Chain and is used to define the geographic group within the Company according to your needs.

The Region level is used to group locations in a geographic area. A Region can only belong to one Area and is defined according to the needs of the Company.

The District is the level following the Region and normally serves to group geographically locations in a Region and may belong to only one Region.

The last level of the hierarchy is the Store. The Store is the level where business transactions take place. There are two types of stores that can be defined, the stock maintenance location which is usually the physical stores or the non-stock location representing an e-commerce or other non-physical locations where sales are processed. A Store can only belong to one district. The Store has several attributes and features available.

The Warehouse defines a physical storage or distribution facility in an organization. This is the only one that can vary in level and can be associated with any level of the hierarchy, because the warehouse can send goods throughout the organizational hierarchy of the Company. A Warehouse has all of the store's transaction capabilities except sales. These can be configured with physical locations, or virtual. The physical warehouse consists of the warehouse and they must have at least one associated virtual warehouse where they will manage the stock. All transactions such as purchase orders and transfers take place in the virtual warehouses.

Merchandise Hierarchy

The Merchandise Hierarchy consists of the different levels of hierarchy relationships that are used to organize the merchandise and that support the Company's product management structure. This allows you to group items by division, group, department, class and subclass and improves the analysis and operations involved. In the figure below we can see the different levels of the merchandise hierarchy that will be described next (Oracle, 2019).

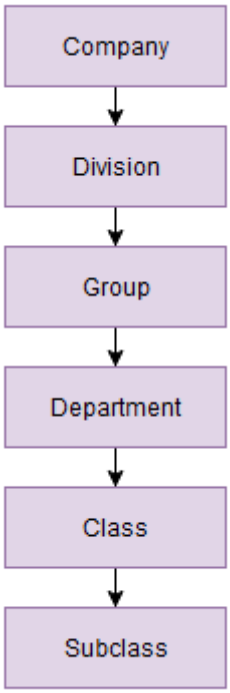

Figure 10 - RMS Merchandise Hierarchy

At the top of the structure of the Merchandise Hierarchy, as in the Organizational Hierarchy, this is the Company, which can also be defined only one per instance of the RMS.

The next level is the Division and is usually used to describe the type of merchandise, such as televisions or vacuum cleaners. A Company can have several divisions.

Then the server Group level to differentiate or segment the merchandise within a specific division, for example in the case of televisions, we can have HD TVs or 4K TVs. There can be several groups in a division, but a group can only belong to one division.

The Department level, which is where products/merchandise are best defined (type of merchandise, own stock, consignment, or concession), and where the budget, markup type and profit calculations are assigned. A department can belong to only one group, but there can be several departments in a group.

The Class further defines the type of merchandise. And finally, the lowest level is the Subclass, which is the level to which the items are attached.

Below is an example of Oracle's merchandise hierarchy (Oracle Learning, 2019):

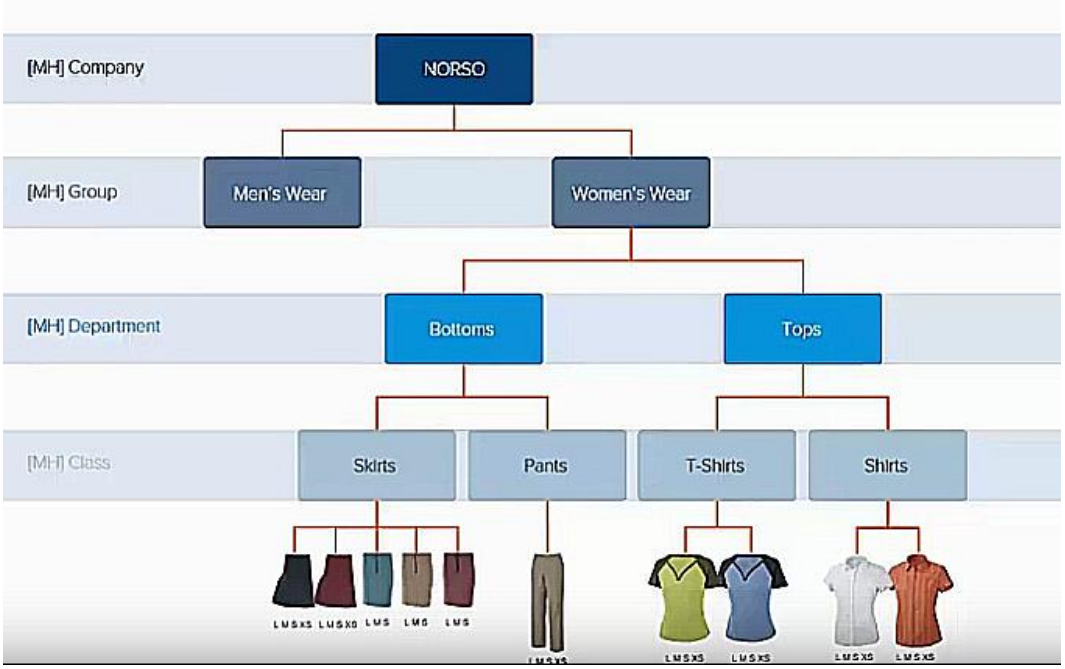

Figure 11 - Example of a Merchandise Hierarchy

Supplier and partner management

In the RMS, a supplier is considered the organization that supplies the stores or warehouses with the goods that are sold, while a partner is the entity associated with the company in the various areas of the procurement process (Oracle, 2019). Suppliers can have a variety of information, such as financial agreements, inventory management parameters, types of EDI transactions and invoice matching attributes. Partners, such as factories, banks or manufacturers, may be involved in the transport of goods, trade, provision of credit and provision of services. Suppliers can do several transactions and functionalities available.

2.2.1.2 Item Maintenance

The RMS groups the items into three levels and is responsible for creating and maintaining them. One of these levels is the transaction level, which is where all price inventories and sales transactions occur. When configuring an item, we can use the desired number of levels up to 3, that is, we can have items with only one level and others with 3 levels.

Normally when using the 3 levels the names of the levels are the following, as shown in the image below (Mishra, 2016):

1. **Product** - definition of the product itself;
2. **Stock Keeping Unit** (SKU) - level where goods are traded and is usually also the transaction level;
3. **Universal Product Code** (UPC) - this is usually where the reference item associated with the SKU is stored, such as the bar code assignment.

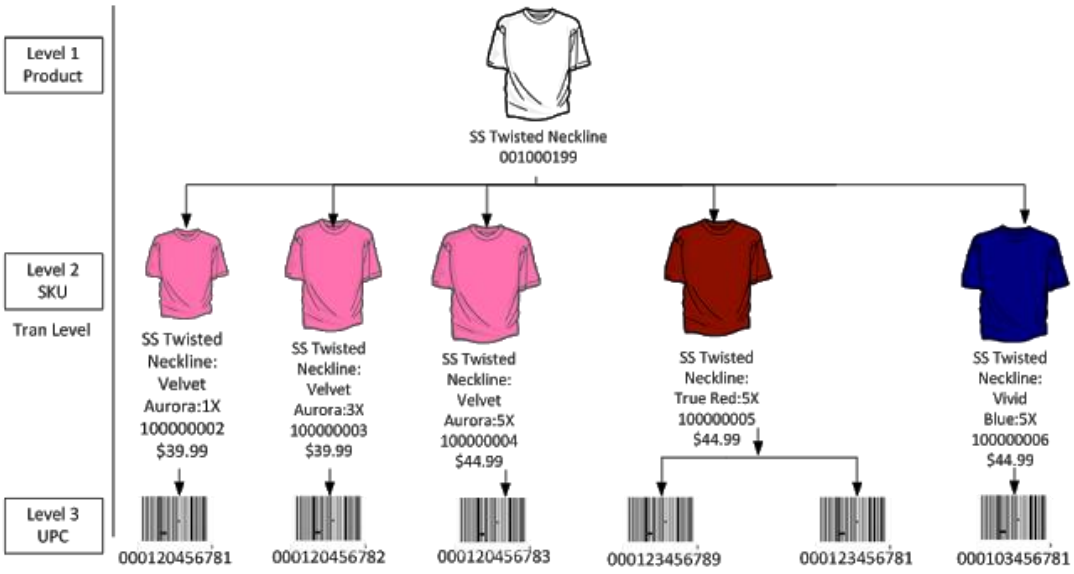

Figure 12 - Example of an Item with 3 Levels

In RMS an item can be defined with differentiators or diff, that is basically a characteristic that distinguishes the different items. These differences can be colors, sizes, flavor or others. This differentiation is normally used in level two items. There are four different Diffs: Diff Type, Diff ID, Diff Group and Diff Range.

The Diff type is used to identify and characterize the differentiator, such as color or flavor.

The Diff ID defines the specific differentiators for each type. For example, in the case of color, we would have orange, green, pink to differentiate colors and in the case of flavor, we could have chocolate, strawberry and caramel for the type of differences in flavor.

The Diff Group are the different sets we can have, using the different ID diffs and type diffs, usually for specific business processes. For example, to create different groups of ice cream flavors like Fresh Flavors (Mint, Lime), Fruit Flavors (Strawberry, Orange, Mango), Sugar Free Flavors (Avocado, Cream, Yogurt). Diff groups are assigned to level 1 items.

Diff ranges serve to differentiate diff ids within a group. For example, a specific range of the most ordered flavors of ice cream. Or in the case of a t-shirt a diff range can be a specific range of sizes.

The RMS allows the creation of different types of items such as: regular items, pack items, deposit items, concession and consignment items, transformable items and catch weight items.

The item maintenance module is responsible for creating and maintaining all items and all of their information.

2.2.1.3 Purchase Order

Purchased Order also known as POs consist of maintaining purchases where the objective is to replenish stores and warehouses or respond to orders from the final consumer. This provides orders for product suppliers in specific quantities for specific locations (Andrew, 2016).

POs can be created in different ways:

- **Manually** - Manually made the RMS directly providing the necessary information that includes the supplier, terms, dates, items, quantities, costs and locations;
- **Automatically by replenishment or contract** - Orders already pre-programmed normally through contracts or orders programmed by stores;
- **External sources** - usually consist of orders created by suppliers that manage part of the stock, a store made by a supplier, Vendor Managed Inventory or management applications. These orders can be sent directly to ORMS.

2.2.1.4 Contracts

In RMS, a contract is a binding agreement with a supplier to purchase a volume of items at a specified cost. The RMS allows the creation, sending and approving of contracts according to the supplier's availability and commitments (Andrew, 2016)**.** Contracts make it possible to calculate costs and persevering deadlines for orders. Contracts can be sent manually or from an external source.

There are 4 different types of contracts: Plan/Availability, Plan/No Availability, No Plan/No Availability, No Plan/Availability. The Plan defines quantities, shipping dates. Availability determines the quantity according to the supplier's availability.

2.2.1.5 Deals

The deals manager allows the creation and maintenance of deals, that is, discounts or rebates negotiated with certain suppliers or manufacturers. After the deal is completed, it needs to be defined in the RMS to apply discounts on purchase orders or retrieve discounts from the supplier. If a supplier is associated with a negotiation, only the items provided by that supplier are applicable to the negotiation.

The different types of businesses supported by RMS are - off-invoice, specific order, refund, refund, vendor-financed promotion (VFP), vendor-financed discount (VFD) and fixed deals.

2.2.1.6 Cost Management

The RMS cost manager allows to create, accept, edit and view price changes for different items. The cost change is an adjustment, upwards or downwards, of the supplier's cost according to the defined cost change reasons (Andrew, 2016). An item's cost record is updated only after the change is duly approved.

There are different types of calculated costs: *Base Costs*, *Net Costs*, *Net Net Costs* and *Dead Net Costs*. According to Oracle, the *Base Cost* is the unit cost that complies with the main supplier. The *Net Cost* is the calculation of the *Base Cost* minus any trade done that has a defined net *Cost Cost*. *Net Net Cost* is the calculation of the *Net Cost* minus any transaction made that has

a cost defined as the *Net Net Cost*, normally used for reimbursements. *Dead Net Cost* is *Net Net Cost* minus any trades that have a cost defined as net *Dead Cost*, generally used for reimbursement of reimbursement. These costs are used for revenue calculations, margin calculations and reports.

The RMS cost management can also be used to create cost zone groups for specific import, transportation or other fixed costs directly related to the geographical area.

2.2.1.7 Inventory Control

Inventory control is considered the main functionality of ORMS. This functionality allows the user to get a sense of what their stock is doing, and which stock is ideal and concrete according to the needs of the store or warehouse. Stock monitoring can be done in two ways, perpetually or financially. Perpetual tracking consists of the number of units of an item in a store or warehouse while the financial stock is the stock value.

Inventory control is done using different functions, some of which are described below (Andrew, 2016):

- **Transfers** - consists of the movement of stock to another location, the RMS allows to create movements, and to monitor this movement. Transactions between companies, book transfers, transfers linked to orders, and other transfers can be made through RMS. A transfer can be created for unavailable stock or available stock. There are several types of transfers, the most basic being manual requests;
- **Returns to Vender** - or RTV, are transactions used to return one or more items to a supplier. This function is normally used by retailers when they want to return damaged, defective, or excess goods. To create an RTV, it's necessary to provide item, quantity and cost data;
- **Inventory adjustments** - are used to increase or decrease inventory to calculate events that occur outside the normal course of business. The RMS allows for two types of stock adjustments, the stock on hand inventory, which adds or removes goods from the available inventory, or unavailable inventory adjustments, which pass the status to unavailable and doesn't remove goods from the stock itself or affect the value of the goods;
- **Purchase order receipts** - record the growth to on-hand when goods are received based on receipt of an Advance Shipping Notice or created manually. The receipt updates many values in the RMS, including the order receipt quantity, available inventory, and weighted average cost;
- **Stock count** - is the stock counting process in a specific place to be compared with the stock levels of the system for discrepancies. There are two types of stock count. Unit stock count is usually performed more often and aims to adjust available quantities at one location, and it also creates an inventory adjustment transaction for the stock ledger. Unit and value stock counts have the same goal but instead of creating a transaction it adjusts the stock ledger book to the stock count results (Andrew, 2016).

2.2.1.8 Replenishment

Replenishment consists of ordering products to replenish a supplier or warehouse's depleted stock. The RMS allows retailers to have detailed stock management in different periods and estimates demand, sales, and stock limits for automatic orders for transfers and purchases allowing to maintain the ideal stock.

Items can be configured for automatic replenishment with the previous configuration of automated replenishment parameters. These parameters include (Andrew, 2016):

- Review cycle and order control
- Due order processing
- Investment buy attributes
- Scaling constraints
- Rounding attributes
- Supplier minimums
- Truck splitting constraints

Replenishment orders can be generated by different replenishment methods/algorithms. There are several different algorithms (Andrew, 2016):

- **Constant** - the idea of this method is to keep the stock at a constant level, that is, when the stock reaches a certain level, the method will place the order for replenishment and can always maintain the same stock level desired;
- **Min/Max** - relatively similar to the constant method, this method performs the replenishment request to refuel to the maximum level when the stock level reaches the defined minimum. This method always keeps the stock between the minimum and maximum level;
- **Floating Point** - relatively similar to the previous methods, this method places the order for replenishment when the level is at the low maximum calculated by the system through the sales history;
- **Time Supply** - it's an algorithm that makes a forecast based on the number of days the stock is kept in a store or warehouse. There are two variants of this algorithm, the time supply seasonal, and the time supply issues;
- **Dynamic** - it is the most complex algorithm. This algorithm aims to have enough stock to make sales between replenishments. This is possible from a series of dynamic calculations based on current sales, order quantities, trends, number of orders and other factors. This method has two variants, Dynamic Seasonal and Dynamic Issues;
- **Store Orders** - this method allows the store to place the order for replenishment according to its needs by calculating the recommended quantity. This recommendation is based on store orders and delivery availability.

2.2.1.9 Stock Ledger

The RMS stock ledger records all merchandising-related financial transactions, from sales, price changes, stock adjustments. All of these transactions are recorded in the stock ledger and at the end of the day, week or month these data are summarized and later used for financial reports or for stock and profit calculations.

The Stock Ledger supports the accounting and costing methods in the system, allowing you to practice both retail and analytical accounting. This feature allows you to work with several books or multi-ledger features, it also allows you to keep separate currencies and charts of accounts for each entity. For example, a set of books can be created for each country of operation.

2.2.2 Oracle Retail Price Management

Oracle Retail Price Management, or RPM, is a configurable system that streamlines retailers in making price-related decisions. It provides optimization and price automation features that allow retailers to obtain more predictable and profitable results by executing pricing strategies. This system is involved in all phases of a product, from regular price to promotions. In addition, it offers pricing strategies that allow retailers to adjust strategies while setting prices according to objectives resulting in better margins and productivity (Andrew, 2016).

The following images (Andrew, 2016) shows the position of RPM in relation to other modules:

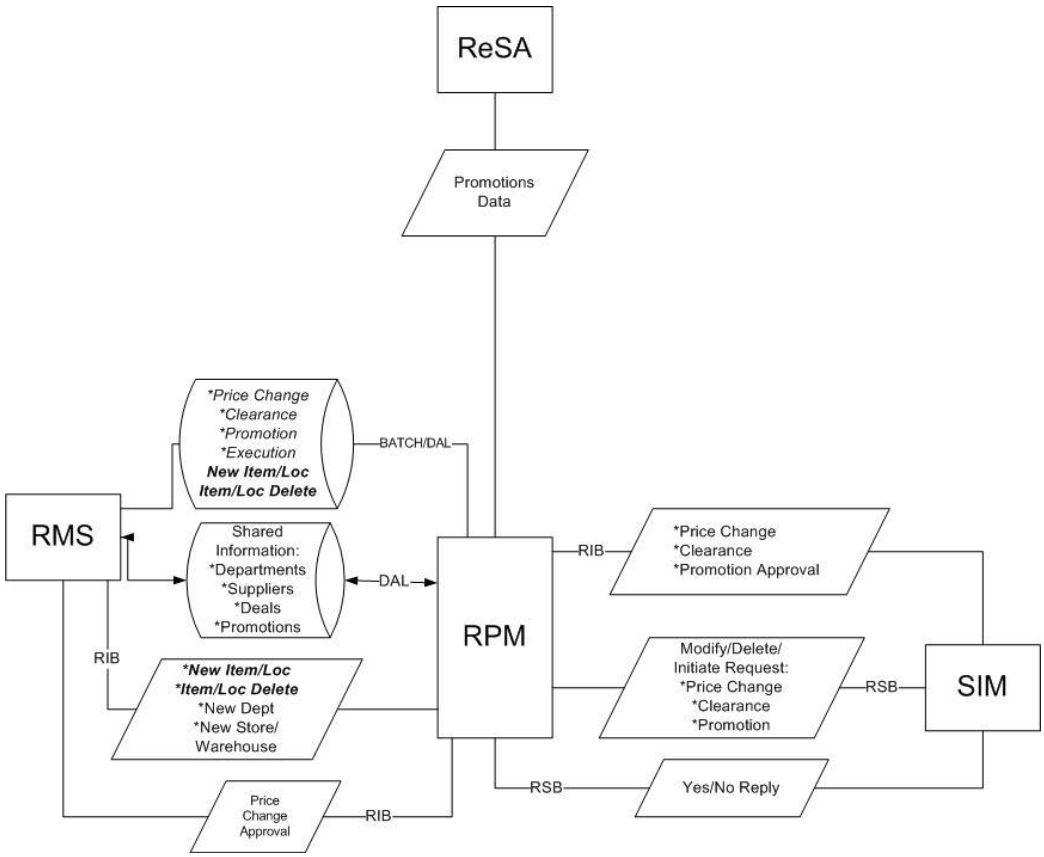

Figure 13 - RPM Relation to the Other Oracle Models

The main capabilities of RPM are:

- Zones Groups

- Initial Price
- Regular Price Changes
- Promotions
- Clearance
- Strategies
- Approval and conflict validation
- Price Change Communication

Before we look at each of the features, let's see what Foundation Data is needed for the RPM to work properly. As in the RMS, for the correct operation of the Price Management system, it is necessary to provide the basis for the management of pricing events. This includes rounding rules, aggregation levels, price zone groups, and system default options, among other important data.

Below is an overview of each functional area:

- **Aggregation Levels -** Creation of a new hierarchy in RMS with the proper definition: the department, the class and subclass of that department. The RPM will assign a default value for each parameter and allow the user to define parameters according to the department level that vary for pricing strategies and price spreadsheets (Andrew, 2016). In addition, it will be necessary to associate an Initial Marking Percentage to determine the initial price of an item with the help of the zone groups associated with the departments;
- **Link Codes and Market Basket Codes -** The RPM allows the assignment of link codes to identical items that will share the same price from merchandise hierarchies, item levels or diffs. The same is true for locations that can be associated with the link code by group zones. Market Basket Codes are codes defined for an item or zone that allows you to group them according to similar price characteristics. They are used in competitive strategies to obtain a price above or below the competitor's price (Mishra, 2017);
- **Zone Structures -** group structures consist of defining a combination of locations for better price management. Zone groups are division categories for the highest levels of groupings. There are 3 types of zone groups: regular zone, release zone and promotion zone;
- **Price Guides -** allows the creation of uniform pricing strategies in order to maintain a consistent set of price ranges. The user can also define what type of rounding is desired. Price guides can be used when defining price strategies, groups of primary zones, promotions, price changes, releases (Andrew, 2016);
- **Calendar -** The calendar aims to define review periods, that is, the time we have to act on certain pricing strategy events (Young, 2016b);
- **Candidate Rules -** are rules that determine whether the item / location that is extracted from the system should be marked for review. These can be inactive or active and can be inclusive or exclusive, the inclusive ones identifying the items that should be flagged in the review for complying with the rule and the exclusive ones that will not be included in the review (Young, 2016b).

2.2.2.1 Price Changes

The RPM has specific pricing events called a price change. These price changes are used to permanently change the sale price of an item or location. To create these changes the retailer has to provide certain information (Andrew, 2016):

- The item to be changed;
- Where is the change being made;
- How the item's price is changing;
- Why is this change taking place.

Price changes can be created from different types of price events supported by RPM (Andrew, 2016):

- **Regular Price Changes** - Price changes made by the retailer;
- **Vendor Funded Markdown** - Price changes to supplier-subsidized promotions;
- **Link Code Price Change** - Price changes for identical items previously grouped in order to set equal prices;
- **Multi-Unit Price Change** - Price change based on the number of units purchased by customers, for example, pants of 5 euros when sold in sets of 3 are 4 euros each.

A settlement is used to clear the final stock of the season or to maintain margin. Settlement, also called Clearance markdowns, are considered permanent price changes. When creating a settlement, it will be necessary to inform which item and location affected and the discount or price defined for the markdown. These events can be generated with a reset date that allows the price to return to normal on the defined date. This is used for seasonal items such as swimsuits, in late summer it makes sense to spend the stock but in early summer the price should return to normal. There are two types of settlements. Regular, which means that the settlement was at the initiative of the retailer or Vendor Funded which means that the supplier is subsidizing part or all of the discount.

2.2.2.2 Promotions

Promotions or temporary markdown consists of temporarily reducing the price of an item. These are configured to apply a regular price, a settlement price, or both, and when the promotion ends, the price returns to the original or to another defined price. These promotions as well as sales can also be financed.

Promotions are organized in three levels as shown in the image below (Mishra, 2017).

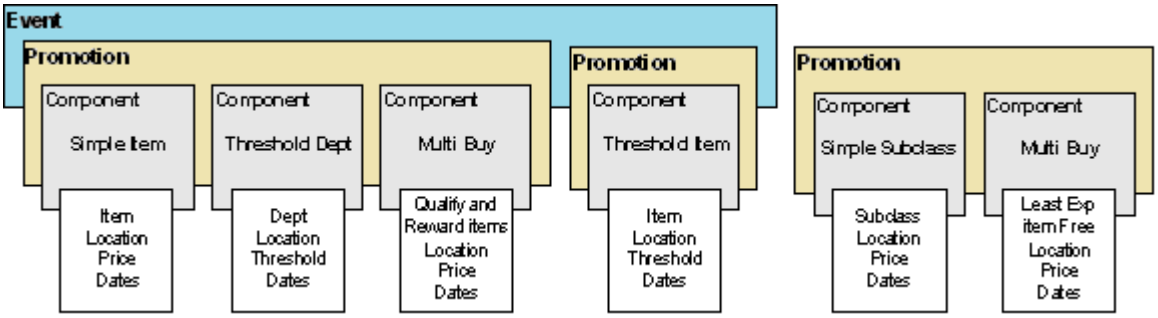

Figure 14 - Promotions Organization and Hierarchy

The first so-called events are used to link promotions with similar themes such as promotions for Valentine's Day. Promotions are not required to have an event. The next level is the promotion header, which better describes and defines promotions. And finally, the promotion component that details the discount, the items involved and the locations / stores. There are two types of promotion, Simple and Complex (Mishra, 2017), a promotion can have more than one of the components specified below:

- **Simple** - discounts taken from the purchase price, either a percentage or specific price;
- **Complex** - the discount is determined based on the purchase of items from a particular shopping list:
    - Threshold - the discount is determined according to the amount purchased or amount spent. For example, when you buy 2 scarves you get a 5% discount and when you buy 5 you get a 10% discount;
    - Multi-Buy - the discount is based on the purchase of one or more items from a defined list. Any item on the list can be combined to obtain the discount. There are several possible types of Multi Buy. Buy items from different discount lists to receive the fixed discount price, for example, buy a plate, dessert and coffee for 10 euros. Buy items from different lists to receive a fixed discount on one or more specific items, for example, buy speakers and an amplifier and receive a free disc. Or have a discount on a value purchase within a price range, for example, spend 30 euros and get a 5% discount.

2.2.2.3 Conflict Check

The RPM allows the verification of conflicts of promotions, price changes. Basically, this functionality aims to make checks every time a change is made that implies the price, such as verifying that an item / location does not have more than one assigned price, in addition it verifies that no price change operation or promotion puts the negative price and that there is no overlap of promotions. These checks can be done in the background, if problems are found in price changes, sales, or promotions this feature uses a notification to alert the user or an error will appear on the price change screen.

In case of conflict the user can check the details on the conflict screen and check its cause to determine the best way to solve the problem.

At the bottom of each time a price change event is sent, there may be a conflict, the system with this functionality allows monitoring prices and verifying that they are in accordance with the rules existing in the system or established and added by the user, they can there are different rules to protect the margin or ensure that the price is no more than a certain amount. For an event to be approved, it must first pass the conflict check without any error.

This check can be done in the background asynchronously. The conflict check process is used in the following areas (Mishra, 2017):

- Price Changes;
- Clearance;
- Promotions;
- Worksheet.

Below are some conflict checking rules (Mishra, 2017):

- There can be more than one price for one item/location on the same day;
- Prices cannot be negative;
- Regular retail cannot be less than clearance retail;
- Clearance changes cannot exist for the same item in the same day;
- Clearance markdowns must be less than the previous markdown;
- Fixed price clearance cannot overlap with a fixed price promotion;
- Clearance retails must be less than or equal to the regular retail;
- Only one fixed price promotion per day is allowed on an item location;
- Promotional retail on top of clearances should be less than clearance retail.

2.2.2.4 Price Strategies

Pricing strategies allow the user to define rules by which the RPM can be guided to suggest certain price changes. These strategies can be defined at any level of the merchandise hierarchy to identify the items affected by the strategy and can be used for regular or settlement price changes. An item can be associated with multiple strategies as long as there is no overlap of the review period and that maintains a margin strategy (Young, 2016b).

Other factors that are relevant to the creation of pricing strategies include aggregating price guides, specifying a schedule to be executed in relation to the pricing strategy, and attaching warehouses for inventory visibility.

There are five different pricing strategies that are described below (Young, 2016a).

1. Area Differentials and Competitor Strategy

This feature allows you to create prices for a given zone in relation to a primary zone and to price other zones outside the primary zone using a defined differential. Area differentials are based on meeting certain financial objectives and can be used when creating a price change to ensure consistent prices.

This functionality allows the retailer to focus on establishing primary retail, which drives prices generated by the system for other secondary areas.

2. Clearance Strategy

A clearance pricing strategy is based on meeting certain financial objectives and allows you to define the method used for markdown items at the specified hierarchy level and whether the reduction percentage is applied to regular retail or clearance retail. Each price reduction specified has an associated percentage. The standard clearance strategy allows you to set up a strategy to generate subsequent discounts after an initial clearance event.

3. Material Requirements Planning Strategy

The Material Requirements Planning (MRP) strategy, modeled by the margin strategy, is used to calculate the new Retail Sale item based on the selected MRP. This strategy is created by the user based on the business need. This strategy proposes new sales in the new Material Planning approved.

4. Competitive Strategy

The competitive pricing strategy is based on the positioning of a retailer's price in relation to its competitors. For this type of strategy, competitive price information maintained in the RMS is used. This strategy allows the customer to define which competitor's retailers to consult and how to make comparisons with those retailers to define the pricing strategy for items based on the prices of their main competitor.

Competitors' price information allows you to compare the proposed prices with a competitor's available prices. If the proposed prices vary from the target percentages set for competitors, the retailer will be informed so that he can adjust the price.

5. Margin Strategy

A margin pricing strategy aims that the retailer defines its strategy based on margin targets and can thus fulfill certain financial objectives. This is possible from the creation of a markup target value above cost that will later be used by the system to propose new retail items.

The margin maintenance strategy allows to define the pricing strategy based on the imminent cost changes. The proposed changes depend on the retail margin or cost. Reference competitors may be designated for comparison purposes to maintain the margin strategy.

2.3 Price Optimization Software Competitors

With this new digital age and the price being the main factor in the purchase and sale of products, some retailers have already started looking for new price optimization solutions that would allow them to keep pace with the market in order to retain customers, maintain margins and still gain a competitive advantage.

The Price Optimization software consists of a price management system that allows you to execute price strategies using price optimization tools. These systems develop a series of price

recommendations at the product item level based on financial, sales and even competitor goals, with a focus on achieving the company's objectives. (Schmidt, 2017).

Of the existing solutions, Profimetrics Price Optimizer, SAP Configure Price Quote Solution and DemandTec Price Optimization, are possibly the biggest competitors in the market for the project to be developed. Each of the solutions will be presented below, as well as their characteristics.

2.3.1 Profimetrics Pricing Lifecycle Optimisation

Profimetrics S.A. is a company, belonging to the Itim Retail group based in the United Kingdom, which develops different software solutions to optimize business processes for retailers. This company has five different patchwork solutions, one of which is Pricing Lifecycle Optimization.

Pricing Lifecycle Optimization is a price optimization solution that aims to monitor the different phases of the product's life cycle and protect the price margin, without compromising the integrity of the stipulated prices allowing to plan and manage price strategies, promotions, and markdowns.

Some of the features of this software are (Pricing Lifecycle Optimisation: Pricing, Promotions & Markdowns, 2018):

- Data migration, pricing strategies and tactics optimized to obtain the best price, across all channels;
- Responds strategically to price changes from competitors;
- Simulates the impact of sales, margins, and prices before making a decision;
- Ensures price consistency;
- Plan, predict and measure strategic promotions, maximizing marketing and promotional investments;
- Create and monitor promotions, at different levels of the company;
- Predict the best result, negotiating, in the best way, with suppliers;
- To drain the inventory in a profitable way, reaching the goals foreseen;
- Recommend the right discount for each product;
- Simulate alternative scenarios to markdowns;
- Integrate with the strategy defined for promotions, integrating the general plan of the company.

According to a study carried out by Itim, one of its customers after six years was able to increase the total gross margins by 6% after the implementation of processes and price optimization strategies provided by the Profimetrics platform.

2.3.2 SAP Configure Price Quote Solution (SAP CPQ)

SAP is a company management software company with different collaborative and multi-company business solutions and a leader in enterprise applications.

Among the different solutions of SAP Customer Experience is the SAP Sales cloud, a solution adapted to sales and marketing, which allows to provide a personalized service based on the management of the relationship with customers. SAP CPQ, SAP Configure, Price and Quote, is part of this solution and allows you to optimize and create quotes and offers of complex products with optimized prices to your customers.

Some of its main features are (SAP CPQ : Configure Price Quote Solutions, 2020):

- Margin protection and expansion;
- Increase sales effectiveness;
- Increase margins;
- Get deals done faster;
- Launch new products faster;
- Increase cross-selling;
- React faster to market changes;
- Enable channel partners;
- Consolidate configurators;
- Onboard salespeople faster;
- Support omnichannel selling;
- Create customized, compelling proposals automatically;
- Automate quoting collaboration.

According to SAP, using the SAP CQP solution, Dell had an increase of over 200 purchase orders per week and Verizon Connect had a 300% increase in quotes and sales improvement (Configure, Price, Quote: Performance, simplified, 2018).

2.3.3 DemandTec Price Optimization

DemandTec is a solution from one of the largest technology companies in the world, IBM, International Business Machines Corporation. This solution allows you to provide prices, promotions and discounts optimized to grow in the retail area and still manage business with suppliers. This solution features different modules that address retail price challenges. One of the existing modules in this solution is price optimization, which focuses on price optimization, executing its pricing strategy to return the best price and still sustain healthy margins (Price Management and Optimization, 2020).

Some of its key features are (Price Management and Optimization, 2020):

- Efficiently execute rules-based pricing through price management;

- Deliver optimal prices through the industry's latest AI modeling techniques;
- Accurately recommend prices based on granular, current price elasticity and demand trends;
- Utilize the most productized science features including strategy recommendations and KVI recommendations;
- Visualize key metrics, prioritize based on notifications and alerts;
- Align to price strategies and understand the cost of rules and constraints to continually refine and improve pricing strategies.

This solution is in the TOP 5 in 17 categories in the 2021 Software LeaderBoard as a price provider for retailers.

2.3.4 Competitor Analysis

In order to better understand the different characteristics of each competitor, a systematization of the different tools presented by each of them was carried out. This analysis does not include the Oracle solution because oracle recently repositioned itself on the market and removed the RPM modules for price strategies and recommendations. The analysis can be seen in the table below and the proper description to each feature also below.

Table 1 - Competitive Matrix of Competitors

		Competitors		
		Profimetric's Price Optimizer	**SAP CPQ**	**DemandTec Price Optimization**
FEATURES	Plan and measure strategic promotions	✓	✓	✓
	Apply price	X	X	X
	Promotion optimization and recommendation	✓	X	X
	Reacting to competitive prices	✓	X	X
	Price forecast	✓	X	✓
	Price analysis	✓	✓	✓
	Elasticity-based prices	X	X	✓

		Competitors		
		Profimetric's Price Optimizer	SAP CPQ	DemandTec Price Optimization
	Pricing Strategies according to specified objectives	X	✓	X
	Profit Analysis	✓	✓	✓
	Reacting to External Events (Price, costs, inventory)	✓	X	✓
	Margin-based strategies	✓	X	X
	Inventory strategies	X	X	X
	Notification System	X	X	✓

The features identified in the table above are described below:

- **Plan and measure strategic promotions** - Feature that allows the user to set goals and priorities for a strategic plan allowing a clear and achievable target;
- **Apply price** - Feature that allows the recommended price to be automatically applied by the solution to the final price software;
- **Promotion optimization and recommendation** - Feature that, based on the history of promotions and based on the results obtained, proposes recommendations or optimizations for the right promotion for each product;
- **Reacting to competitive prices** - Feature that reacts to the price of a competitor's item and recommends a price for an identical item according to the retailer's target;
- **Price forecast** - Feature that predicts the price of an item or service by evaluating various factors, such as demand, trends, competitors' prices, etc.;
- **Price analysis** - Feature that allows to examine the price or price recommendations of different items in distinct locations;
- **Elasticity-based prices** - Feature that recommends prices for an item based on how sensitive the quantity demanded of it is to its price;
- **Pricing Strategies according to specified objectives** - Feature that allows for price recommendations based on previously created goals and objectives;
- **Profit Analysis** - Analytical feature that allows the user to evaluate retail and business profitability;
- **Reacting to External Events** (Price, costs, inventory) - Feature that reacts when events like changes in cost, low inventory or competitors promotions occurs;
- **Margin-based strategies** - Strategy that recommends a price based on a target margin set by the user;

- **Inventory strategies** - Strategy thar recommends a price based on the quantity of items in the inventory, for example reacting when there are no sales of certain items;
- **Notification System** - Feature that delivers messages or notifications based on certain events.

DemandTec's Price Optimization solution allows you to manage and forecast prices and costs. In addition, it also performs price analysis and monitoring of competitors' prices. But it does not perform any kind of price calculation from the data obtained.

SAP CPQ does not allow market data analysis or competency price tracking. Furthermore, it does not provide any kind of price forecast, but on the other hand it allows for good price management and the proper calculation of prices and costs.

Like DemandTec, Prometric's Price Optimizer allows you to manage prices, enabling you to forecast prices and track competition prices. But on the other hand, it does not perform any kind of price calculation.

Although there are already several price optimization software on the market, there is still a large percentage of retailers that use manual pricing processes. There are also reports from some retailers that they stopped using some of the existing solutions due to a weak or poor implementation of pricing strategies.

The solution proposed in this thesis aims to present new pricing strategies and still more efficiently execute some of the techniques existing on the market. In addition, the solution is being implemented at Retail Consult, which is one of the largest companies in the Oracle Retail area and which, in addition to already having a vague list of possible customers for this new solution, also has extensive knowledge in retail and much price expertise. This puts Retail Consult in a unique position from the point of view of implementers in this area and from the point of view of knowledge of the business.

2.4 Oracle Retail Development Frameworks

In this section, an approach and study of different oracle technologies is made, where two possible frameworks to be used in development are introduced and compared.

2.4.1 Oracle Application Development Framework

Oracle Application Development Framework known as ADF, is an end-to-end development framework based on the Java Enterprise platform, implementing the best practices of the Java EE standard and open-source technologies. This focuses on the business, not on low-level programming and on the consolidation and evolution of previous frameworks.

The framework provides integrated infrastructure solutions for the different layers of the application and covers the complete development cycle from conception to deployment,

containing some interesting features such as: drag and drop, bindings, User interface design, and other development resources, thus allowing an easy way to develop them and minimizing the need to write code.

Application Development Framework (ADF) is based on the Model-View-Controller (MVC) design pattern which promotes weak coupling or increased cohesion in the application classes. Thus, the use of MVC, can facilitate the development and maintenance of the application. Oracle ADF provides a solution for each of the MVC layers and supports easy integration of the various layers, along with integrated security for a custom solution.

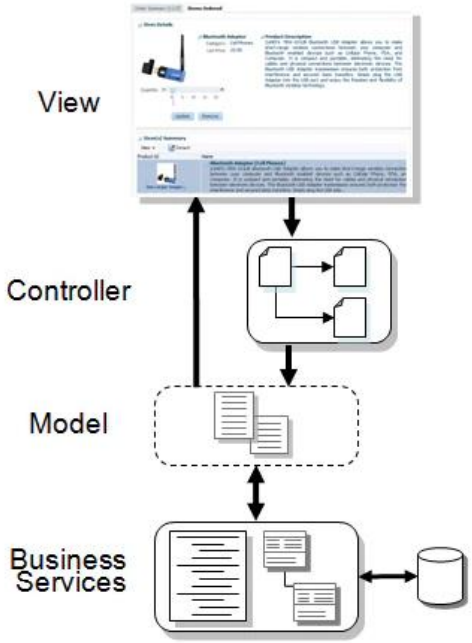

Figure 15 - The Model-View-Controller (MVC) Architecture Design Pattern

The MVC standard, as shown above (Oracle and/or its affiliates, 2015), is divided into three essential layers:

- **Model or Business Object Model** - It is the layer responsible for the business logic that contains the data structure. It has the responsibility to manage and control the way the data behaves and the access and manipulation and validation of the data. It receives the information from the control layer, validates it and translates it into relevant information to be displayed by View;
- **View or client-side** - This is the visualization layer responsible for the user interface. This is responsible for presenting information visibly to the end user. The view displays a representation of the data where only features linked to appearance are applied using HTML and / or XML. This layer of interaction with the user is responsible for transmitting information or questions to the controller and returning the responses obtained to the user;

- **Controller** - It is the control layer, responsible for serving as a means of communication/connection between the model and the view, which manages the application flow and acts as an intermediary between the requests sent by View with the responses provided by Model. This processes the data provided by the user and distributes it to the remaining layers, also controlling which model will be used and which view will be presented to the user.

Separating applications into these three layers simplifies component maintenance and reuse between applications. The independence of each layer from the others results in a loosely coupled Service Oriented Architecture (SOA).

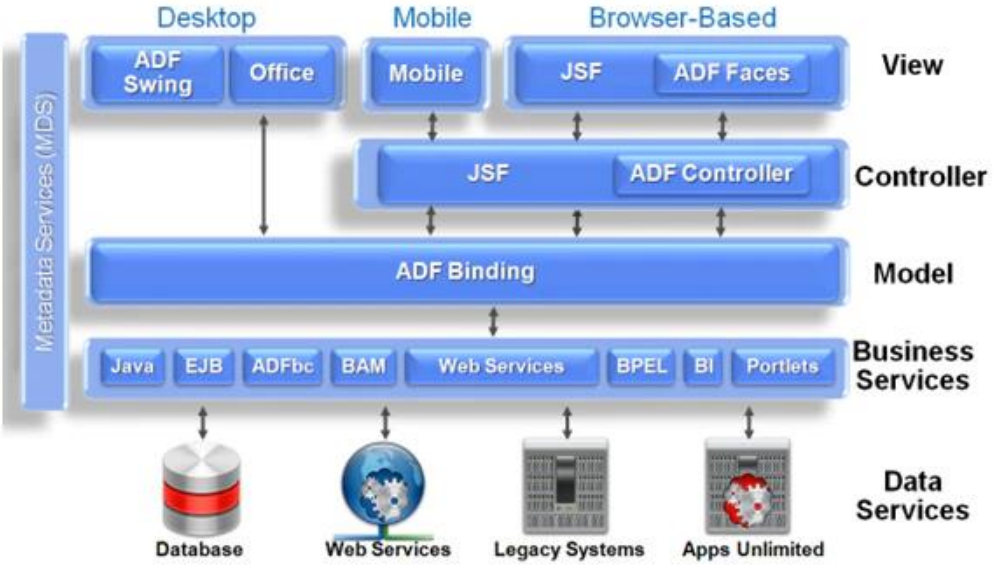

Figure 16 - The Oracle ADF Architecture

Oracle ADF implements MVC, as shown above (Eralp, 2013), and further separates the business services model layer to enable service-oriented application development.

The ADF is divided into 4 layers:

- **ADF Business Components** - are objects that allow you to program the business logic as one or more services using pre-built reusable components with common design standards allowing for better performance and faster product delivery. This layer provides the main functionality of the application, this includes all kinds of calculations, business rules, and the very important functionality of storing data (Vesterli, 2017);
- **ADF Binding / Model** - This layer works on top of the Business Components layer and allows the View and Controller layers to work with different business

service implementations from a Service Oriented interface. This Layer connects the business services to the objects that use them in the other layers. “The model layer consists of two components, data controls and data bindings, which use metadata files to define the interface” (O’Brien and Shmeltzer, 2011). This offers better productivity by providing the option of drag and drop and a greater reusability by their Discover & Share services;

- **ADF Controller** - In the control layer, the control of the application's job flows is carried out as well as the manipulation of the user's input, thus offering better navigation. For example, when you click a Search button on a page, the controller determines what action to perform (do a search) and where to navigate to (the results page) (O’Brien and Shmeltzer, 2011);
- **ADF Faces Components** - consists of the application's user interface layer. “For Web based interface Oracle ADF offers a rich set of over a 150 Ajax enabled JavaServer Faces components that simplified the creation of dynamic and appealing user interfaces” (O’Brien and Shmeltzer, 2011).

Oracle ADF implements best practices from the Java EE standard and is more productive as it allows for a declarative and visual development approach that allows you to focus on the business and application logic without having to enter low-level programming.

2.4.2 Oracle JavaScript Extension Toolkit

The Oracle JavaScript Extension Toolkit also known as JET is a toolkit for JavaScript WEB development with a corporate focus.

It has support for popular open-source technologies and instead of creating a new WEB tool, Oracle JET is based on well-established free libraries and frameworks like Hammer, Knockout, Cordova, jQuery, jQueryUI and Require.JS and supports corporate features like advanced navigation page, accessibility, and internationalization. The complexity of WEB application development is facilitated and accelerated with the integration of several free components in a single toolkit. In addition to allowing to generate hybrid mobile Web applications, it allows you to create and serve tasks for Android, iOS and Web applications.

According to Geertjan Wielenga, Product Manager, the goal of Oracle JET is to provide a stable foundation for intermediate and advanced JavaScript developers in order to efficiently visualize data in the cloud.

JET incorporates these libraries and simplifies the development of JavaScript, HTML and CSS applications. Allowing the use of any development environment such as Visual Studio Code, Netbeans, Atom, among others.

Oracle JET also uses Grunt, Yeoman, npm and Bower, popular tools in front-end development. In addition to performing complete project life cycle management, it provides built-in support,

intelligent resource management and is rich in different User Interface (UI) sets to help aid in data visualization.

JET is based on the Model-View-ViewModel architecture design pattern also known as MVVM. The framework uses the knockout library to implement data binding between the View layer and the model layer based in the MVVM pattern. This architecture has certain benefits: the application can have even more decoupled components, each part of the code remains granular and allows for less data transactions between the server side and the client side making the application more flexible and faster while using less resources.

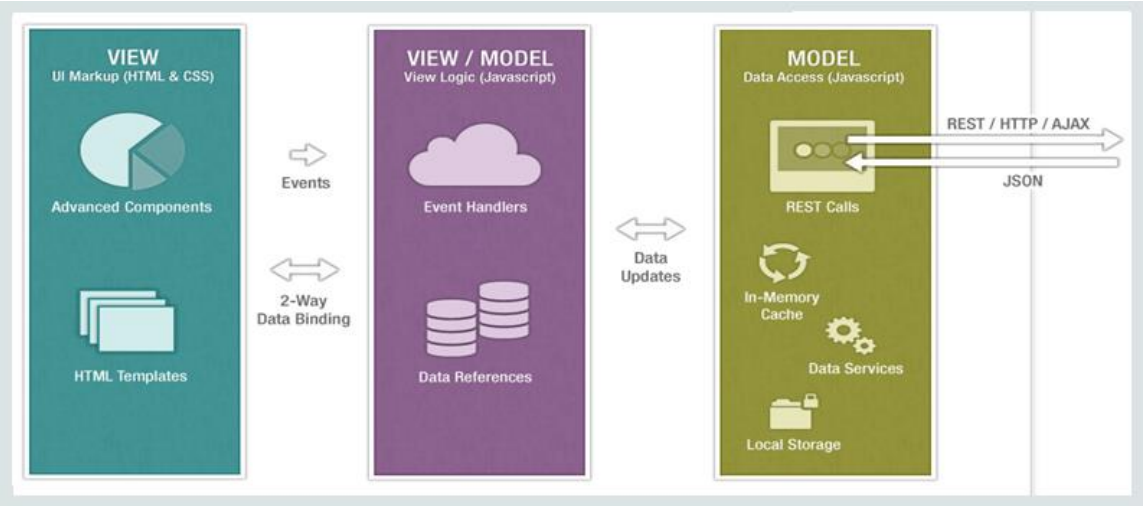

Figure 17 - The Model-View-ViewModel (MVVM) Architectural Design Pattern

The MVVM standard, as shown above (Gordon, 2020), is divided into three essential layers:

- **Model** - It has a similar function to that presented in the previous topic regarding the MVC architecture, that is, it has the responsibility to manage and control the behavior of the data and the access and manipulation of that data;
- **View/Model** - The ViewModel handles the communication between the View and the Model by passing all the necessary data from the View to the Model in a form that the Model can digest (Kouraklis, 2016). It provides View layer with presentation logic, that is, exposes data flows and commands to View to activate certain events and to maintain the state of the view. It's also responsible for passing commands from the view to the model keeping this layer updated according to the user's actions. These changes are made automatically through the existence of Data Binding between the View and ViewModel layers;
- **View** - it's the user interaction layer, it has all the interface formatting and the specific validations of the interface. This is considered an active layer because it has Data Bindings with the ViewModel layer so that they are in sync. Here the data obtained from the interface is processed to be made available to the other layers.

This architecture, shown below (Egan, 2020), offers certain benefits because the logical and presentation layer is loosely coupled, which means better maintenance, testability and extensibility.

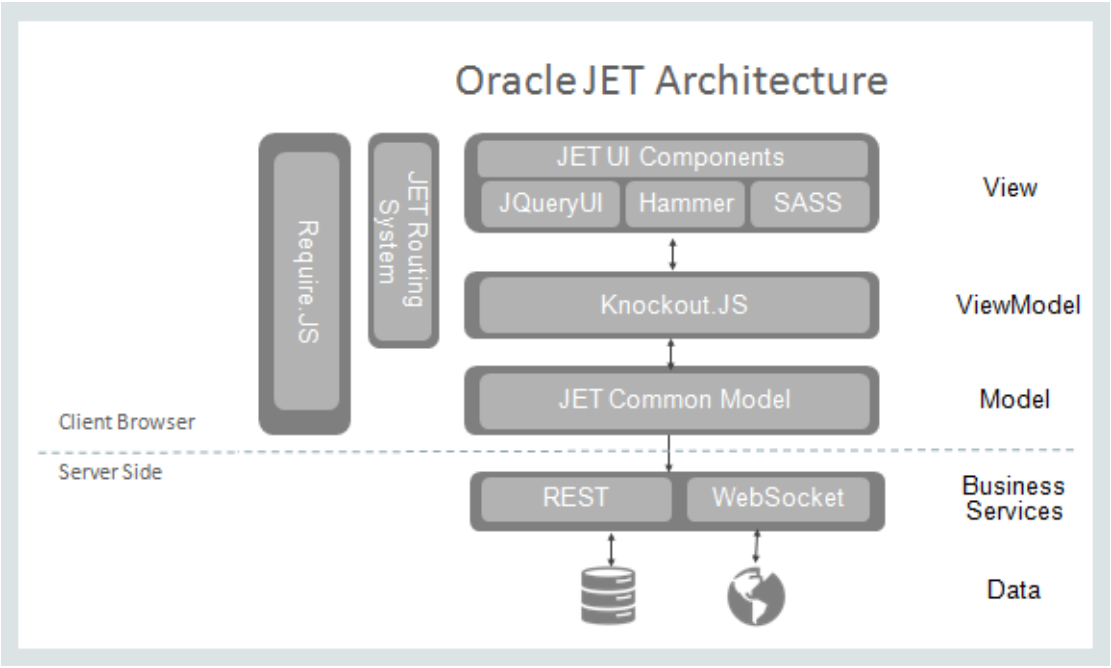

Figure 18 - Oracle JET Architecture

“Oracle JET Common Model and the application programming interface collection implement the model layer, that includes the following JavaScript objects: Model, collection, events and KnockoutUtils” (Egan, 2020).

The view layer is implemented by a collection of components implemented as HTML and the viewModel layer is implemented by Knockouts Js library for Data Binding implementation.

2.4.3 JET VS ADF

ADF and JET target completely different audiences. In the image below (Shmeltzer, 2015) we can see the main differences between the two frameworks. These differences will help decide the best and most appropriate framework for this project.

Oracle JET vs Oracle ADF – The Basics

	Oracle ADF	Oracle JET
Who is it for	Java Developer	JavaScript Client Developer
Where it runs	Server	Client
Development Experience	Declarative	Code centric
Development Tools	JDeveloper / OEPE	Any
Architecture	MVC	MVVM
Data Services	ADF BC, EJB/POJO, SOAP, REST	REST, JSON
Customization & Backward Compatibility	MDS and XML based	Develop Your Own
Based on	Java EE	Popular open source libraries
Integrated Solutions	SOA, BI, WebCenter	Application Builder Cloud Service
What Oracle Builds with it	SaaS applications	PaaS products

Figure 19 - Main Differences Between ADF and JET

The basic and most important characteristics to take into account is that ADF is a server side framework and also both front and back end framework and JET is client side focused and UI layer only (Shmeltzer, 2019). For this project the main focus will be the back end having barely any contact with the UI making ADF the most adequate framework.

Another relevant feature is that the target audience of Oracle ADF is Java developers while in Oracle JET the target is Javascript developers. Retail Consult mostly uses ADF for all their projects having experienced developers in this framework and in the MVC architecture.

While the two differences above were the main ones to be considered, there were other aspects that were taken in consideration in the image above.

Learning the whole ADF stack may prove to be a challenge but considering the background of Retail Consult, the skills of the other members who will join the team and the needs of the project, ADF is the best choice.

2.5 Database Analysis and Comparison

Among the different technologies involved in creating the software one of the most important is the database. There are different database models, the most common being the relational model and the non-relational model. In the image below adapted from (Rainergewalt, 2020), we can see the difference between both models.

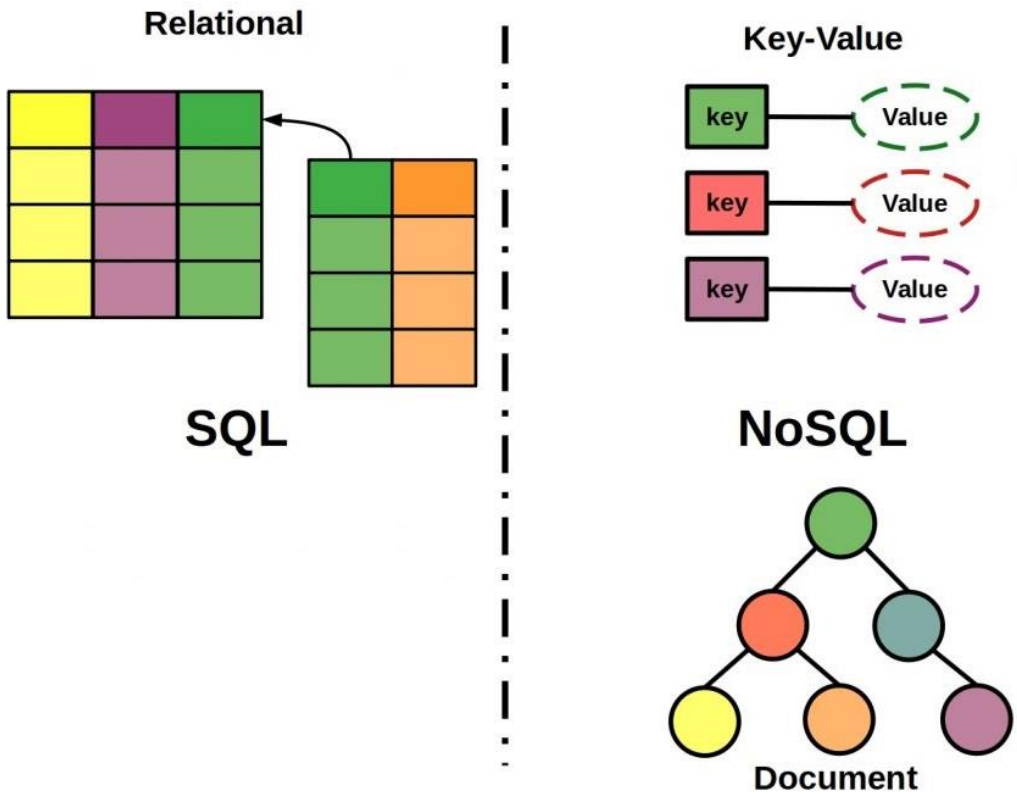

Figure 20 - Difference Between Relational and Non-Relational Model

In the case of relational databases, the data is stored and organized in tables, each consisting of columns and rows, where the column defines an attribute of the entity and each row includes the data of the entity in question. For example, if we have a product table, the different lines would be the different existing products and the columns their attributes, such as product identification, product name, price.

In one table, normally one of the attributes is defined as the primary key, to identify a unique row, which can be used in other tables as a foreign key to relate the two tables.

This model allows greater security as it is possible to assign restrictions and permissions to different users, it is flexible regarding data accessibility and it avoids duplication of data.

Some examples of the best ranked relational databases (DB-Engines Ranking, 2021) are Oracle Database, MySQL, Microsoft SQL Server, PostgreSQL and IBM Db2.

Non-relational databases, also known as NoSQL, are all databases that do not follow the relational model. Basically, these allow you to save the data in a single structure without first defining the data structure. Unlike the relational database, where it is necessary to define the structure and relationships in advance, in this model you can have data with different properties in the same structure without defining it.

There are different types of NoSQL databases which are distinguished by the way of storing the data. The two most well-known are the Document stores where data is stored in documents (JSON, XML) or collections and the Key-value Store which are a simpler database where the structure consists of a dictionary or map and each item contains keys and values.

Some examples of the best ranked NoSQL databases (DB-Engines Ranking, 2021) are MongoDB, Redis, Cassandra, Neo4j, and HBase.

For the solution to be as compatible as possible with all types of products that may exist on the market and since all the major ERPs in the retail world operate on a relational database, we will align the software with the existing architecture and implement a relational database as it is the most obvious and most flexible answer.

The Oracle database, which allows customers to simplify relational database environments and reduce management workloads and is known for being robust and complex and for providing security in different layers, was the Database management system chosen. The Oracle database is one of the most widely used SQL databases for providing high scalability, performance and reliability (Cost-optimized and High-Performance Database, 2021). In addition, as previously mentioned, the project is being developed in a company that is directly linked to Oracle, so the primary objective is to have the solution running on Oracle Retail's RPM and RMS modules, modules that use the database Oracle data. For data manipulation, PL / SQL (procedural Language / Structured Query Language) will be used.

2.6 Critical Analysis

The state of the art has made it possible to obtain a better knowledge of the retail area and of all the processes involving this complex area. In addition, it was possible to better understand the problems of flaps and the need for a retailer and also the difficulties that the retailer has in terms of pricing.

ERPs are a system highly sought after by retailers to manage business activities, taking into account that the project is being carried out at Retail Consult, which is a direct partner of Oracle Retail, the solution to be implemented will be based on RMS solutions and RPM.

As was seen in chapter 2.4, there are already some price optimization software on the market, but the solution proposed in this thesis aims to introduce new pricing mechanisms and to execute even more efficiently some of the existing techniques on the market. In addition, the solution is being implemented at Retail Consult, which has extensive knowledge in retail and a lot of expertise in pricing. This puts Retail Consult in a unique position from the point of view of the implementers in this area and from the point of view of business knowledge.

After the study and comparison of two Oracle frameworks, JET and ADF, it was also possible to conclude that the most suitable framework for the realization of the solution is the ADF for

being the one that most applies to the software needs, for being server side but also because the other elements of Retail Consult involve in the project usually ADF.

This study provided the basic knowledge for the realization of the proposed solution based on the research and study of existing approaches.

3 Analysis and Design

In order to start the development of the project, it is necessary to have theoretical bases for each of the technologies with which one will work. It starts by doing an analysis and survey of requirements, modeling the work systems, and addressing the technologies that are used.

The analysis and survey of requirements is an important step to make a good representation about the system that is intended to be created, besides allowing to intervention in the system without causing problems to the other existing applications. For this purpose, the FURPS+ requirements classification system was used, an improvement on the model initially developed by Grady and Caswell at Hewlett-Packard (Eeles, 2005a).

According to the FURPS+ model, requirements are defined within the following categories: Functionality, Usability, Reliability, Performance, Supportability and the "+" that represents other possible nonfunctional requirements.

There are some requirements gathering techniques that are complementary and can be used within each requirement:

- **Interviews** - traditional technique that is simple to use and that produces good results in obtaining data;
- **Questionnaire** - technique used when there is a large number of people from whom it is intended to extract the same information;
- **Workshop** - in which a team of analysts and a selection of stakeholders are present, defining the purpose for which the system will be used, obtaining more defined requirements;
- **Brainstorming** - used in workshops, presenting the problem to a specific group in order to obtain other perspectives and new solutions;
- **Group dynamics** - informal discussion with the purpose of obtaining in-depth qualitative information.

3.1 System Actors

During the analysis of the system, it was possible to identify the three different actors, the system administrator, the commercial assistant, and the commercial director. Briefly speaking, this is what each of them will be able to do in the solution and their main function:

3.1.1 System Administrator

The main function of the system administrator is to plan, organize and monitor the solution system and assist in the management of data and information, this role being played by someone with extensive knowledge of the organization and its objectives and basic knowledge in IT. In this case, is responsible for defining all system administration settings, for managing the security of the application and for managing its maintenance.

3.1.2 Commercial Assistant

The commercial assistant is responsible for supporting a commercial sector, mainly providing assistance in the pricing processes, being an actor with high knowledge on how to guarantee profits and on customers. In this case, this actor will manage and maintain all existing strategies in the best way that meets the company's objectives. In addition, he is responsible for analyzing all the data provided after the execution of a strategy and for submitting the suggested price or a price recommendation for approval from a commercial director.

3.1.3 Commercial Director

The commercial director, whose main function is to supervise the analysis and decision of prices, is responsible for making the final decision and for approving all the tasks performed by the commercial assistant. Like the commercial assistant, he reviews all the conclusive data of one or more strategies potentially at an aggregate level and can also submit a price recommendation. He is the only one that has the power to approve a price recommendation.

3.2 Domain Model

The domain model is a set of diagrams, which consists of the representation of classes, relationship between classes and objects that illustrate the problem domain. This diagram is of high importance in the analysis and design phase.

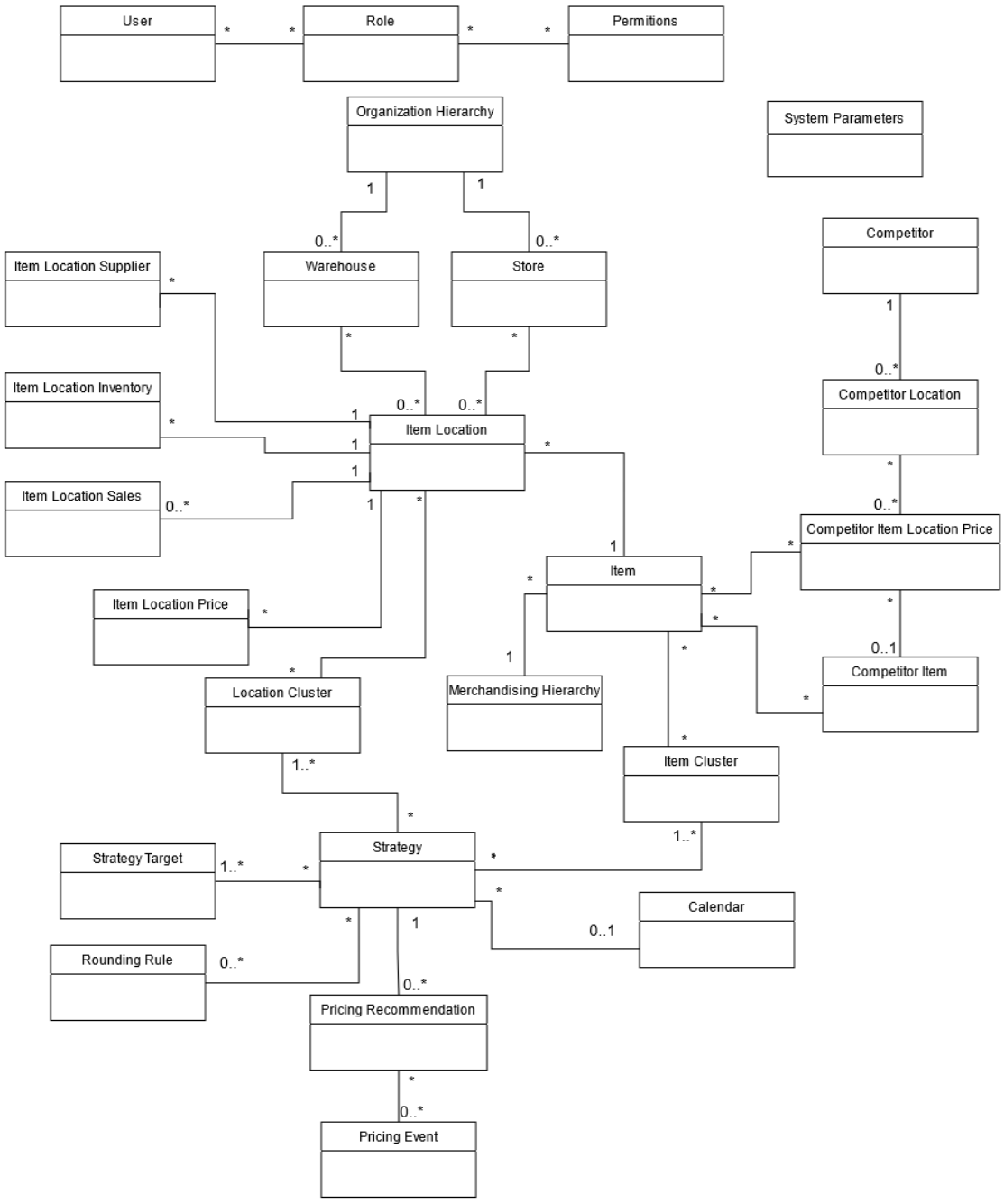

Figure 21 - Domain Model of the Solution

The domain model shown in the figure above, presents the concepts and objects of the domain of interest. It should be noted that this is not yet a final version and that it may change.

3.3 Functional Requirements

This chapter presents all the functional requirements identified intrinsic to the solution to be developed. This was followed by the method presented in an article (Santos, Ferreira and Machado, 2021) that reveals different technical approaches such as interviews, questionnaires and workshops to determine the minimum requirements according to the stakeholder expectations, while the remaining features of the roadmap will be refined later. The functional requirements were defined according to the FURPS+ model and fall under the Functionality category of the same.

Figure 22 - Use Cases Diagram of the Solution

In the figure above the use-case diagram shows the functional requirements that were identified according to the different areas and the existing actors and which will be detailed below.

3.3.1 System Settings

The system settings area is the section where you can customize the solution settings. This can be accessed by users with the System Administrator role.

The settings available in the system settings area are:

3.3.1.1 UC1 – Define / Alter System settings

This requirement defines that the solution must allow the definition and alteration of different system configurations. These settings include configurations such as system language settings, currency settings, notification settings, active hours / work hours definitions, location definitions and other definitions relevant to the system.

3.3.1.2 UC2 - Define / Alter security settings

This requirement defines the possibility of creating and changing different security settings. For example, the creation and maintenance of different users with different roles and permissions that define the possibility of editing, viewing, submitting and approving according to the requirements of their function, and limiting access to information based on the hierarchy of permissions. It also allows to change authentication settings.

3.3.1.3 UC3 - Define / Alter maintenance settings

This requirement defines the possibility of managing the system's maintenance settings with the main focus on preventive and corrective maintenance. These settings allow to manage backup and migration settings, update settings and configuration, export backups, setting a time to perform maintenance of the system and other settings that involve maintenance.

3.3.2 Core Data Maintenance Area

The Core Data Maintenance Area is the section in which it is possible to configure foundation data as hierarchies, groups such as competitors. This can be accessed by users with the Commercial Assistance role.

The features available in the Core Data Maintenance Area are:

3.3.2.1 UC4 - Define relationships between merchandising hierarchy branches

This requirement allows to define different relationships between different levels of the merchandising hierarchy, such as division, group, department, class and subclass, and also create relationships between the three levels of an item, Style, SKU, Barcode.

3.3.2.2 UC5 - Define relationships between organization hierarchy branches

This requirement allows defining the different relationships between the 6 levels of the organization's hierarchy, chain, area, Region, district, channel, warehouses and stores, thus maintaining the desired operational structure.

3.3.2.3 UC6 - Define relationships between items

This requirement allows defining relationships with stores or items of different stores or locations that have items in common and that when a price analysis is made will be analyzed and compared with the items in question to be able to determine the best price.

3.3.3 Strategy Management Area

The Strategy Management Area is the section in which it is possible to configure and maintain the necessary data and configurations for the correct functioning of the strategies. This can be accessed by users with the Commercial Assistance role.

The Strategy Management Area has the following features:

3.3.3.1 UC7 - Maintain margin strategy

This strategy consists of defining or advising a price based on the desired profit margin and helping through recommendations, for example, being advised when a cost change is planned that could affect the margin, that is, it is not just a reactive process, but one of anticipation. This allows to set the price of a product taking into account the production price and the proposed target margin. For that, it is necessary to maintain and guarantee some specifications, such as the desired profit margin targets for the different products / items.

3.3.3.2 UC8 - Maintain competitor's strategy

This strategy consists of recommendations based on competitors' price for a product from previously selected competitors and comparing it in order to determine the best price or to provide information in the price decision. This allows to adjust the price if there are changes, above or below, the price of competitors, according to the definitions set for this strategy. For that, it is necessary to maintain and ensure some specifications, such as the level of percentage change in the price of a competitor, if we want to be notified with the price increase or decrease and what items will be compared.

3.3.3.3 UC9 - Maintain Relationship strategy

This strategy consists of defining or advising a price change based on a relationship, this relationship can consist of relations or links between clusters of items and locations. For this, it is necessary to maintain and guarantee some specifications, such as identifying the group or list that are part of a strategy and the goal for this group.

3.3.4 Revision and Approval Area

The Revision and Approval Area is the section in which it is possible to review all the price recommendations generated by the strategies and to submit and approve a recommendation. This can be accessed by users with the Commercial Assistance and Commercial Director role.

The features available in the Revision and Approval Area are:

3.3.4.1 UC10 - Perform what if analysis

This requirement consists of performing what if analysis that provides a preview of the different alternatives and the possible outcome in the execution of a price change. Thus, allowing a general review of the recommendation.

3.3.4.2 UC11 - Submit price recommendations

This requirement consists of the possibility to submit a price recommendation to be executed. When the submission is made, waits for the needed approval.

3.3.4.3 UC12 - Review and approve price recommendations

This requirement consists of the final review of the price recommendation submitted and according to the final decision, approval, or rejection of the same recommendation.

3.4 Non-functional Requirements

In accordance with the FURPS+ model, the non-functional requirements were identified and defined according to the customer's needs and described below.

3.4.1 Usability

Usability is based on being concerned with features such as accessibility, aesthetics, consistency and problem in the user interface (Eeles, 2005a).

The solution must be simple to use and have a reduced number of steps necessary to carry out an action to make the use as intuitive as possible and easy to use. It must be accessible so that the user has greater ease of work and is able to navigate without fail, with the least number of errors possible. This provides the user of the application with a quick learning of all its functioning and a better experience.

Combining ease of handling with speed of execution and the quality of the data provided should be one of the focal points in the development of the application that will provide a good user experience. The usability requirements identified are:

- **NFR1** - Simple and user-oriented user interface;
- **NFR2** - The solution must be intuitive and easy to use.

3.4.2 Reliability

Reliability is primarily concerned with features such as availability, performance, robustness and consistent calculations by the system, depending on the quality of the inputs that are provided, and the ability to recover from errors and failures, it’s basically the solution’s ability to run flawlessly (Eeles, 2005a).

The solution must be available on the working hours on weekdays, it must be useful and provide quality information and operations. It must be able to operate with a low failure rate, in the event of a failure or error, the “functional” errors should be clearly marked and presented to users so that they can resolve them, the errors that are not possible to anticipate should be presented to the user (an error occurred, please contact your system manager) and they must have an adequate followed up in terms of support and maintenance of the solution, such as logs. The software must contain exactly those features, offering users all the operations they expect from it. The reliability requirements identified are:

- **NFR3** - Must be available on the working hours on weekdays;
- **NFR4** - In case of errors, a clear warning must be presented to the user;
- **NFR5** - Present the functionalities and operations as expected.

3.4.3 Performance

Performance is concerned with features such as transfer rates, throughput, response time, recovery time, startup time and shutdown time (Eeles, 2005a).

The solution must carry out the operations requested by the user as soon as possible and with immediate feedback, avoiding delays in data availability, impairing the user experience, as well as the efficiency of the solution, for this any action that takes more than 10 seconds must be made in the background. As a rule, we will try to ensure that everything that takes longer than that adopts this strategy of running in the background with alerts for users when completed. In other situations where the delay time is not predictable, when an action begins that may or may not take a while, we give the user the option of switching to background (Processing warning message with a button as an option to switch this processing to the background).

All high-performance computing processing necessary for the correct functioning must be done from a daily batch process, in an 8-hour window for data processing and must be done in the background. The Performance requirements identified are:

- **NFR6** - User interaction must have feedback within 100 milliseconds;
- **NFR7** - Actions that are predicted to take more than 10 seconds must run on the background;
- **NFR8** - Present the user the possibility to switch actions processing to the background;
- **NFR9** - Daily Batch for data processing must run on the background in an 8-hour window.

3.4.4 Supportability

Supportability covers requirements associated with characteristics such as testability, adaptability, maintenance, compatibility, configurability, instability, scalability and localization (Eeles, 2005b).

The software must be modular, created in a way that allows the insertion of new requirements and possible alteration and modification of functionalities in an easy way, in addition to being able to add new ones. In addition, the solution must be available for different web browsers. Its development should focus mainly on a process of continuous improvement, working to ensure that its content is developed according to accepted standards for accessibility through a process of continuous testing to ascertain that there are no errors. Regarding the localization characteristic, it must have the possibility to extend the solution (through configuration) to new languages, translations and internationalization.

The solutions must have logging, where all the errors are registered, and audit, where all the records of any changes to the data are registered, for support and maintenance purposes.

The maintenance of the solution must be done in order to improve its performance, seeking exhaustively to correct any and all errors. The entire maintenance process is directly related to the documentation performed throughout the application implementation process. More broadly, the application must offer the possibility to add new features without losing its identity and the server must be able to grow.

If there is a need to carry out an update, it must be done outside working hours so as not to interrupt possible uses, or, in case that's not possible, it must be done the quickest way possible and a warning must be displayed temporary unavailability of the service to the user.

The supportability requirements identified are:

- **NFR10** - The solution must be modular;
- **NFR11** - The solution must support different browsers;
- **NFR12** - The solution must allow different configurations regarding languages and internationalization;
- **NFR13** - There must be logging and auditing log;
- **NFR14** - Ability to perform maintenance and updates.

3.4.5 Security

Security covers requirements associated with the need to provide a solution or service to protect access to certain data, resources or information (Eeles, 2005a).

Security in software development is undoubtedly one of the key points for its success. The solution to be developed will contain a login system agnostic to what exists, depending on the ERP, each retailer will have a different authentication tool. In a first stage, for the sake of time, we will ignore and simply guarantee that the user exists in a local table. When the project is more advanced, security will have a flexible abstraction layer for each ERP.

Each user may have different user roles and depending on their role, their access, edit, view, approve and send permissions will be defined, thus being able to determine that the user who is doing a certain action has the necessary permissions to do so. there will also be a Data level

security, which refers to permissions in terms of data (e.g., certain users have their access restricted to certain branches of a marketing structure).

The security requirements identified are:

- **NFR15** - Restricted access to authenticated users;
- **NFR16** - Access and permissions depending on the defined roles.

3.4.6 Regulatory

Regulatory covers requirements associated with the rules and regulations imposed by the government in which a company is inserted, specifying codes of practice to carry out pricing. In this case, an initial phase will only be followed by Portuguese price laws, for example, the selling price cannot be less than the cost price, the balances can be realized in any period as long as it does not exceed 124 days per year, the anti-dumping law must be followed, among others possible regulations existing in the Portuguese law decree. The regulatory requirements identified are:

- **NFR17** - The solution must comply with the pricing laws imposed by the law decree.

3.5 Competitor Analysis

After analyzing the requirements, a new analysis was carried out in order to compare the competitors previously identified in the section 2.3.4 and the proposed solution.

Table 2 - Updated competitive Matrix of Competitors

			Competitors		
		Proposed Solution	**Profimetric's Price Optimizer**	**SAP CPQ**	**DemandTec Price Optimization**
FEATURES	Plan and measure strategic promotions	X	✓	✓	✓
	Apply price	✓	X	X	X
	Promotion optimization and recommendation	X	✓	X	X

			Competitors		
		Proposed Solution	Profimetric's Price Optimizer	SAP CPQ	DemandTec Price Optimization
	Reacting to competitive prices	✓	✓	X	X
	Price forecast	X	✓	X	✓
	Price analysis	✓	✓	✓	✓
	Elasticity-based prices	✓	X	X	✓
	Pricing Strategies according to specified objectives	✓	X	✓	X
	Profit Analysis	✓	✓	✓	✓
	Reacting to External Events (Price, costs, inventory)	✓	✓	X	✓
	Margin-based strategies	✓	✓	X	X
	Inventory strategies	✓	X	X	X
	Notification System	✓	X	X	✓

In the table above, it is possible to see that the solution proposed in this thesis presents practically all the features described in the table, some of them original to the solution and others already existing on the market. Regarding the features already on the market, the objective is to make them more efficient, based on Retail Consult's vast knowledge in the retail area, its pricing expertise and its business point of view. The new features in the proposal aim to meet customer requirements and respond to the gap between what retailers want and existing solutions offer. Although it was necessary to make certain decisions regarding which requirements would be mandatory in terms of implementation and which would be included in the thesis, and which would be considered "nice to have". It was necessary to scale the amount of work due to the fact that it was carried out exclusively by one person, without the participation of any other developers.

3.6 Design Alternatives

This chapter presents the different architecture alternatives for the solution that were identified according to the objectives and requirements identified.

During the analysis and design phase of the solution architecture, different alternatives were evaluated, which will be further investigated. It was decided to use the deployment diagram to present the different architectural alternatives where the different components of the system are presented and how they interact with each other.

Eight different components have been identified:

- **GUI ADF** - The platform or graphical interface that is presented to the client/user, which is located on the Web Server node;
- **ADF Faces** - Consists of the application's user interface layer of ADF, which is located on the Application Server node;
- **ADF Controller** - This ADF component is responsible for the control of the application's job flows as well as the manipulation of the user's input, which is located on the Application Server node;
- **ADF Binding** - This ADF Layer connects the business services to the objects that use them in the other layers, and it's located in the Application Server node;
- **ADF Business Service** - This ADF layers manages transactions and provides the access to data from other sources, and it's located in the Application Server node;
- **Price Recommendation Engine** - It is the component responsible for providing services and mechanisms to read/persist data, and for calculation processes;
- **Abstraction Layer** - is the abstraction layer component that allows us to read and write from any ERP independently and regardless of the structure of the ERP database, by configuration, this layer makes the necessary connection;
- **Data Base Server (ERP)** - is the database server component of the ERP in question, in this case an external component that we are not responsible for creating or managing.

Of the three architectures presented, the two major differences between them are the placement of the two components, Price Intelligent Engine and Abstraction Layer, in the different nodes.

To choose the best alternative to adopt, the multicriteria decision method Analytic Hierarchy Process was also applied, as seen in section 3.7.

3.6.1 Alternative 1 - Flexibility Focus Approach

The diagram below represents the architecture of alternative 1, a flexible and dynamic architecture that allows you to fine tune the relationships of the solution presented to any ERP that allows price management. In this case, the abstraction layer component would be in the application server module and would have to work from Webservices, messages or files. This solution is considered flexible and dynamic because the abstraction layer is included in the application server. That is, to use the solution with another ERP, it would only be necessary to change the abstraction layer relationships. But one of the problems with this architecture is that the ERP would have to make the data available in real time, we are talking about requesting and reading millions of records through a synchronous communication pattern, making this architecture in terms of performance, feeble. One could try to persist the data to calculate and send, but it would not be in real-time and it would also require a duplicate persistence of millions of records, between the ERP database and that of our solution.

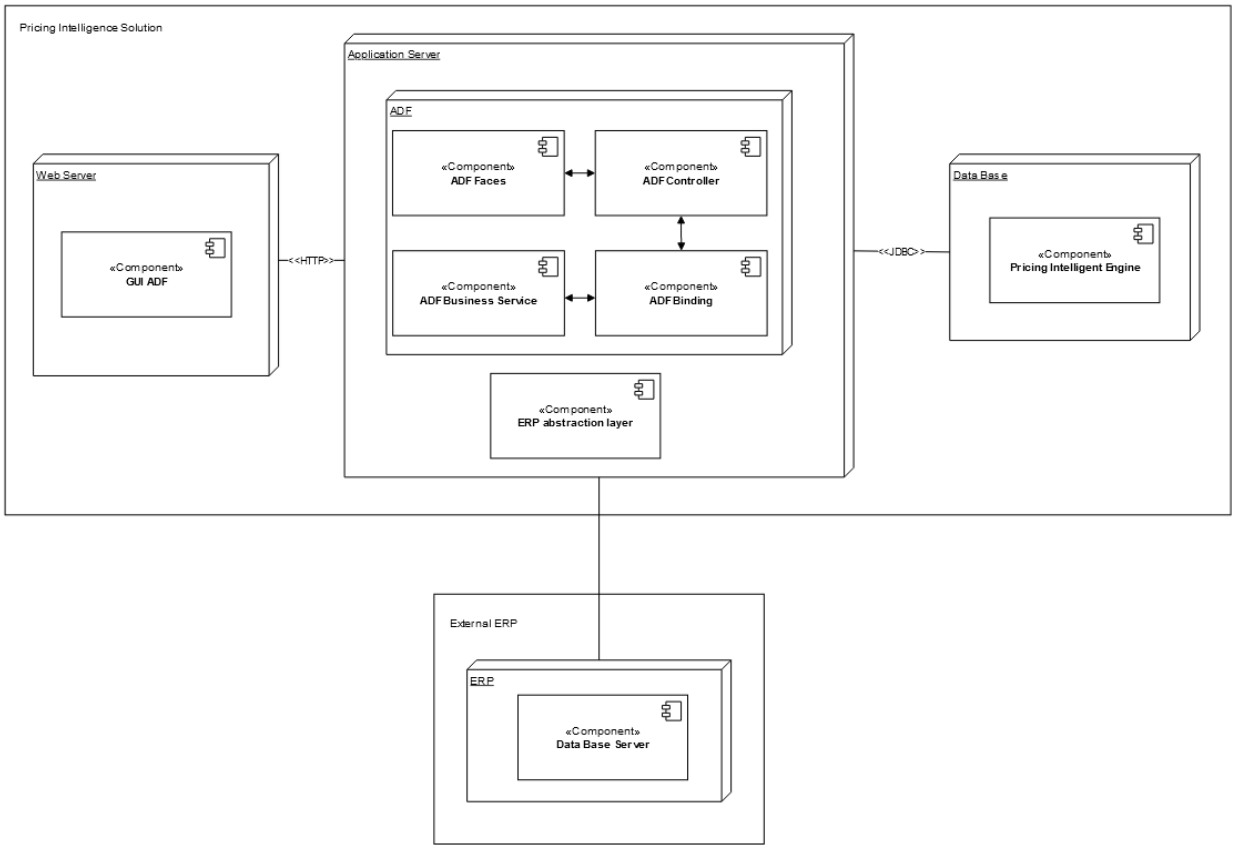

Figure 23 - Alternative 1 Deployment Diagram

3.6.2 Alternative 2 - Cost Focus Approach

The diagram below represents the architecture of alternative 2, a cheap and flexible architecture because in the end we would be pushing the responsibility of the ERP in question to the client. This architecture would have advantages, as we are passing on part of the ownership and charge to the solution's client, which would make the solution cheaper, however, we would be dependent on the ERP in question to receive the data in the necessary way. In addition, the Pricing Intelligent Engine, is in the application server, which raises performance problems since it is typically easier to scale from a performance perspective in the database than in the Application Layer.

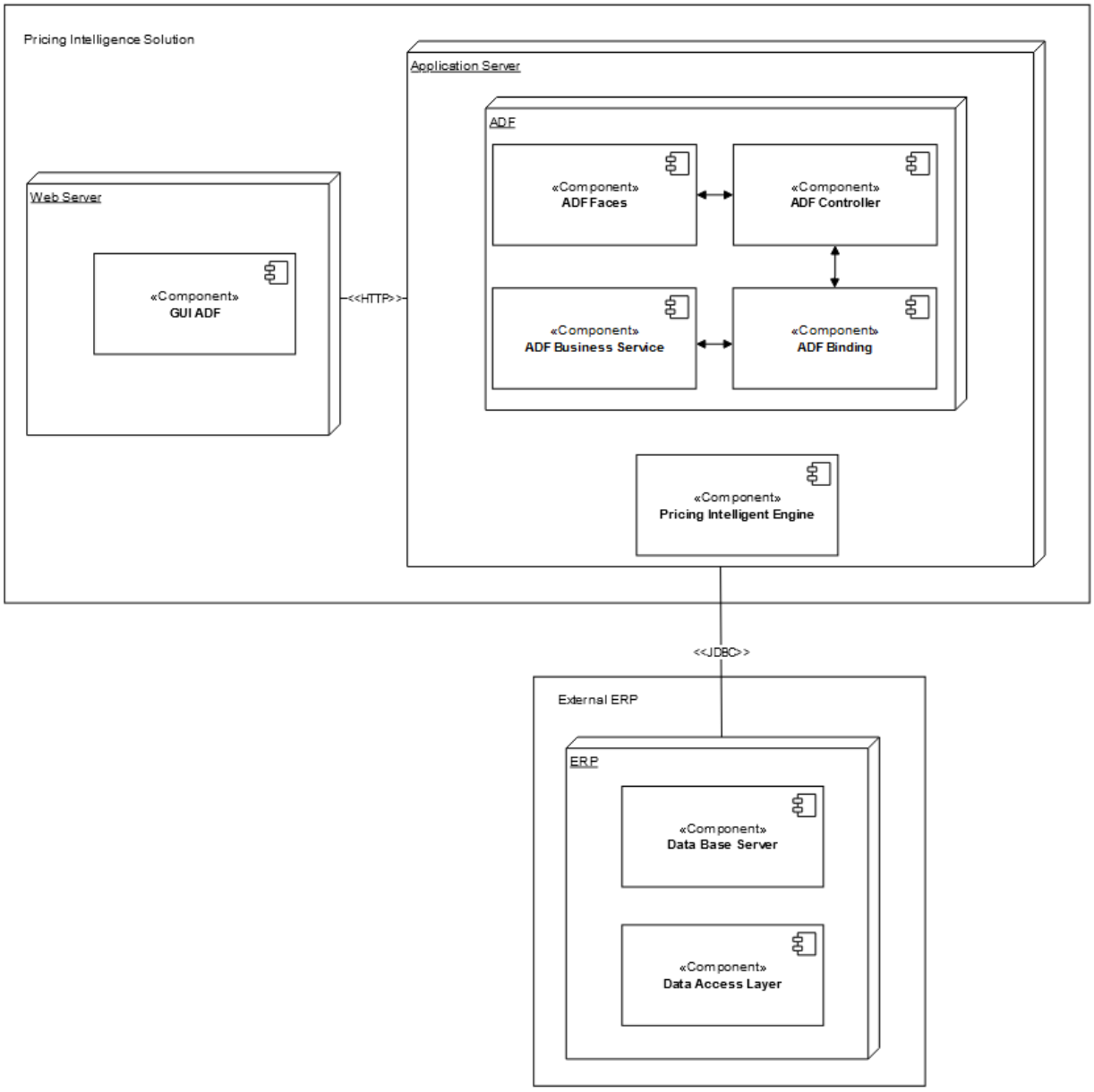

Figure 24 - Alternative 2 Deployment Diagram

3.6.3 Alternative 3 - Balanced Approach

The diagram below represents the architecture of alternative 3, a scalable and flexible architecture. In this case, both the Price Intelligent Engine and the Abstraction layer would be in the Node of the Database, which made it easier to scale because the engine was in the database and not in the application layer. In addition, it allows to have the data in real time. The disadvantage is that we are assuming, for simplicity in connecting our solution to the ERP, that they will both work with Oracle databases.

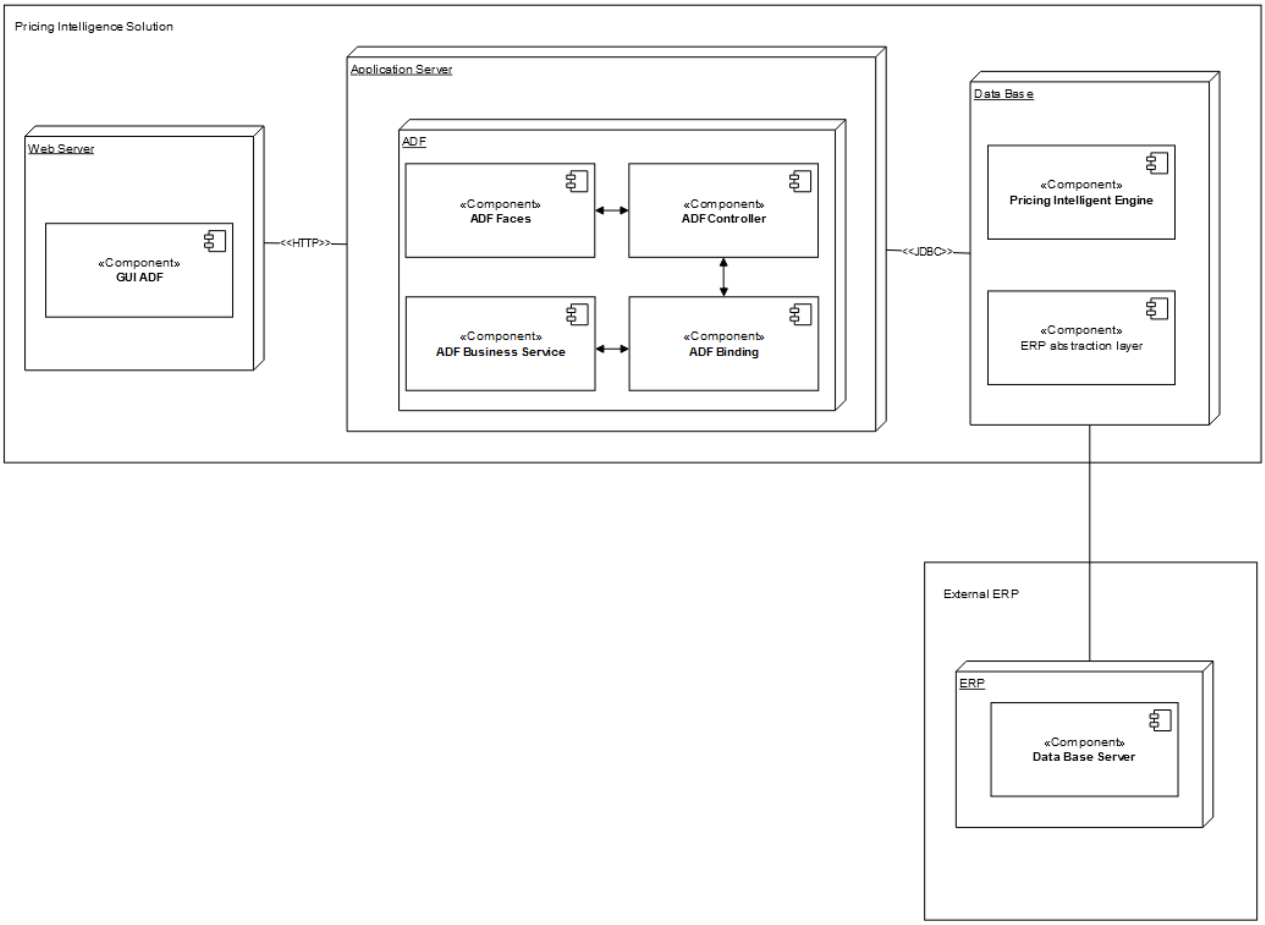

Figure 25 - Alternative 3 Deployment Diagram

3.7 Comparison of Alternatives and Decisions

The Analytic Hierarchy Process (AHP) is a method used for multicriteria decision making, planning and conflict resolution that has as main focus the removal of conscience and, considering different factors simultaneously and making the necessary compensations allows to reach a realistic conclusion (Saaty, 1987).

This method's main goal is to support the evaluation of the alternatives and help in the process of the choice of the best option, therefor, facilitating the work of the evaluators and the difficulties in choosing the best option.

The advantages of this hierarchical decomposition are clear. By structuring the problem this way, it is possible to better understand the decision to be achieved, the criteria to be used and the projects to be evaluated. So, we can list the alternatives ordered by their overall priority and get the best alternative, the one with the best overall priority.

a) Construction of the hierarchical decision tree

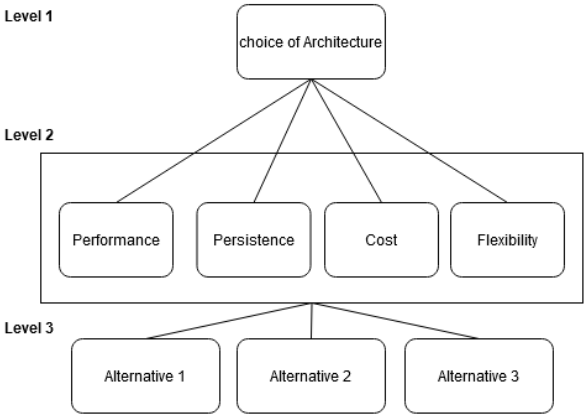

Figure 26 - AHP Hierarchical Decision Tree

According to the logic of the AHP method, the hierarchical decision tree presented in the figure above was built. The first level defines the objective, that is, the problem to be solved. In this case the objective is to choose the most suitable architecture for the realization of the proposed solution.

At level two, requirements are defined to be taken into account when evaluating each of the proposed alternatives. In this case, 4 criteria were selected.

- Performance;
- Persistence;
- Cost;
- Flexibility.

At the last level, the different alternatives proposed are defined. In this case, three alternatives were the three possible architectural designs.

b) Comparison of alternatives and criteria

Table 3 - Fundamental scale - Levels of Importance of Comparisons
adapted from: (Satty, 1991)

Intensity of Importance	Definition
1	Equal Importance
3	Moderate Importance
5	Strong Importance
7	Very Strong
9	Extreme Importance
2, 4, 6, 8	For Compromise Between the Above Values

Then the comparison was made between the different existing criteria and from the comparison matrix available above, which follows the levels of importance of comparisons of the fundamental scale (Satty, 1991). The scale used in AHP for the comparison matrix is a scale from 1 to 9.

Considering the four criteria defined in the hierarchical tree, the following comparison matrix was developed between the different criteria:

Table 4 - Pairwise Comparison Matrix

	Performance	Persistence	Cost	Flexibility
Performance	1	1	3	3
Persistence	1	1	3	3
Cost	1/3	1/3	1	1/2
Flexibility	1/3	1/3	2	1
Sum	**2 2/3**	**2 2/3**	**9**	**7 1/2**

c) Relative priority of each criterion

Then it is necessary to determine the relative priority of each criteria by dividing each element of the comparison matrix by the total of the respective column, for this each value of the matrix is divided by the total of the respective column. Finally, the arithmetic mean of the values of each line of the normalized matrix is made.

In the table below it is possible to see the normalized matrix with the respective priority vectors:

Table 5 - Normalized Comparison Matrix and Estimated Weights

	Performance	Persistence	Cost	Flexibility	Relative Priority
Performance	0,3750	0,3750	0,3333	0,4000	**0,371**
Persistence	0,3750	0,3750	0,3333	0,4000	**0,371**
Cost	0,1250	0,1250	0,1111	0,0667	**0,107**
Flexibility	0,1250	0,1250	0,2222	0,1333	**0,151**
Sum	**1,000**	**1,000**	**1,000**	**1,000**	**1,000**

According to the results obtained, it can be concluded that the Performance and Persistence criterions are considered the most important followed by Flexibility and Cost.

d) Assess the consistency of relative priorities

In the next step, the objective is to calculate the ratio of Consciousness (RC), which consists of measuring how consistent the judgments are. For the judgements to be consistent the ratio of consciousness needs to be less than 0,1, because in order to obtain a coherent explanation, it must be in accordance with an order of magnitude of the inconsistency priority, that is, an error in the measure of consciousness (Saaty, 1990).

But for that it is necessary to first calculate or obtain the value λ_{max} using the following equation:

$$Ax = \lambda_{max}\, x$$

Calculation of Ax:

Table 6 - Ax Calculation

$$\begin{vmatrix} 1 & 1 & 3 & 3 \\ 1 & 1 & 3 & 3 \\ 1/3 & 1/3 & 1 & 1/2 \\ 1/3 & 1/3 & 2 & 1 \end{vmatrix} \cdot \begin{vmatrix} 0{,}371 \\ 0{,}371 \\ 0{,}107 \\ 0{,}151 \end{vmatrix} = \begin{vmatrix} 1{,}517 \\ 1{,}517 \\ 0{,}430 \\ 0{,}613 \end{vmatrix}$$

Calculation of λ_{max}:

Table 7 - λ_max Calculation

$$\begin{vmatrix} 1{,}517 \\ 1{,}517 \\ 0{,}430 \\ 0{,}613 \end{vmatrix} = \lambda_{max} \cdot \begin{vmatrix} 0{,}371 \\ 0{,}371 \\ 0{,}107 \\ 0{,}151 \end{vmatrix} \quad <=> \lambda_{max} = 4{,}061$$

After calculating the λ_{max} it is possible to obtain the Consciousness Index (CI), necessary to calculate the consciousness ratio (RC):

$$IC = \frac{\lambda_{max} - n}{n - 1} = \frac{4{,}061 - 4}{4 - 3} = 0{,}0204$$

Finally, we can calculate the ratio of Consciousness (RC), which consists of the division between the ratio of conscience and the random index according to the number of criteria.

According to the table of relative priorities, visible in the figure below, the IR value for 4 criteria is 0.90.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Figure 27 - Relative Priorities Table

Formula and calculation of the ratio of Consciousness (RC):

$$RC = \frac{IC}{IR} = \frac{0{,}0204}{0{,}90} = 0{,}0227$$

The awareness index (CC) is less than 0.1 so it is possible to conclude that the values of the relative priorities are consistent.

e) Construction of the parity comparison matrix for each criterion, considering each of the selected alternatives

In the fifth phase, the comparison matrices between the three alternatives are constructed for each of the different criteria.

After performing all the comparison tables, it is possible to synthesize all the information and obtain the table below with the relative priorities for each of the alternatives and the respective criteria.

Table 8 - Parity Comparison Matrix

	Performance	Persistence	Cost	Flexibility
1	0,080	0,064	0,143	0,164
2	0,219	0,290	0,714	0,297
3	0,701	0,646	0,143	0,539

f) Obtain composite priority for alternatives

Finally, in the last phase, the priorities for each alternative are calculated. This calculation is made by multiplying the priority matrix and the criteria weights, as shown below:

Table 9 - Priority for Each Alternative

0,080	0,064	0,143	0,164	.	0,371	=	0,094
0,219	0,290	0,714	0,297		0,371		0,310
0,701	0,646	0,143	0,539		0,107		0,597
					0,151		

g) Choice of alternative

According to the AHP method, from the results obtained it can be concluded that the most suitable alternative for the proposed objective, according to the defined criteria and their respective importance, is the alternative 3.

In this case for the choice of the alternative it was also taken into account that most of the data will be obtained from a set of results in the database. To do the processing and all the calculations involved in the strategies outside the database layer would imply loading the data into some data structure, making the calculations based on those structures and then returning it all back. This method would have several problems, first in terms of reading and writing time, there would be downgrade in performance times, and also, because we would have to do math in java and java is not strong in this type of processing, in addition all this processing can be done easily at the database level. PL/SQL is executed on Oracle servers without the overhead of multiple readings or writing on networks or communication between processes, thus allowing a low latency. In addition, high-performance batch processing should happen close to the data for better performance. That is why having the price engine in the database layer and using the procedural language PL/SQL is more efficient and faster, making perfect sense to choose alternative 3.

In this case, taking into account the realization time for the project, it was decided to build a first version entirely built with the data model of oracle retail MOM. And the abstract layer in this first phase will be built as a Data access layer that is prepared for MOM and that in the long term will be changed to work with different ERPs. The final Architecture is presented below.

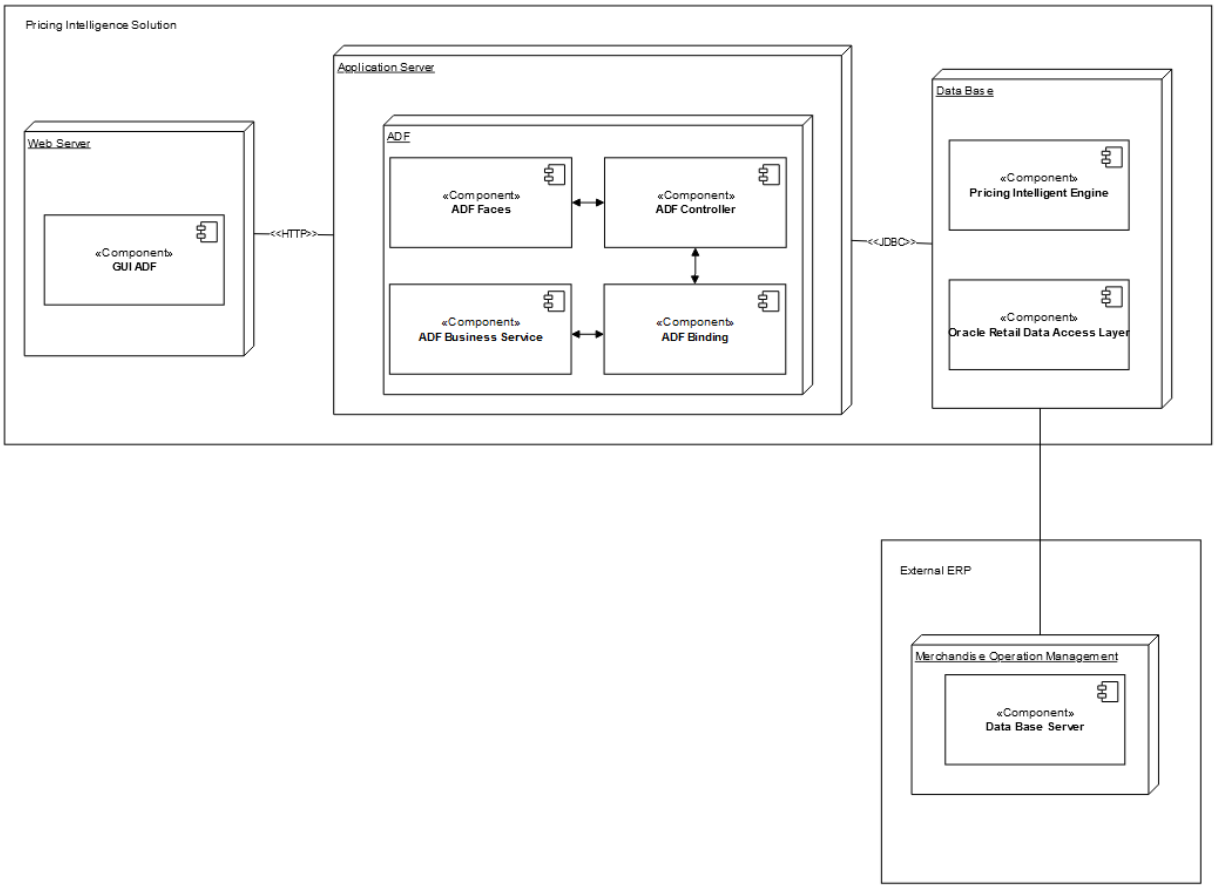

Figure 28 - Solution Final Architecture

4 Development and Testing

4.1 Database

In this chapter, the implementation of the database and the relational model implemented in the solution will be presented. As mentioned in chapter 2.5, the database used is of the relational type and was developed using the multi-model database management system Oracle Database.

4.1.1 Database Preparation

For the creation and management of the database it was first installed the integrating Oracle Express Edition, a web-based software built on the Oracle Database platform. After proper installation and configuration, the PL/SQL Developer configuration was started, a specific IDE for development with the use of Oracle database.

For the configuration it was necessary to edit the tnsnames.ora file, a configuration file that allows you to connect the database when entering the required data. In this case the settings were made to be connected to two databases. The first setting is relative to the new database created for the solution, called Strategy Based Price Recommendation. The second configuration is relative to the MOM1601_RC database, which is the Oracle Merchandising Operation Management database and contains all the base data to be used by the solution (e.g., stores, items, average cost, current selling price, etc.). In figure below it is possible to see the .tnsnames.ora file after the required modifications.

tnsnames.ora

C: > instantclient_12_1 > Network > ADMIN > tnsnames.ora

```
#ORM1601trainning
#RMS/[redacted]
MOM1601_RC=
 (DESCRIPTION =
  (ADDRESS=(PROTOCOL=TCP)(HOST=[redacted])(PORT=1521))
  (CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=MOM))
  )

#PRICING_INTELLIGENCE
#PRICING_INTELLIGENCE/[redacted]
PRICING_INTELLIGENCE =
 (DESCRIPTION =
  (ADDRESS=(PROTOCOL=TCP)(HOST=[redacted])(PORT=1521))
  (CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=xe))
  )

```

Figure 29 - .tnsnames.ora File with the Database Configuration

In PL/SQL Developer the database we create will have the name we define, to facilitate identification. When starting PL/SQL Developer It should be something like this:

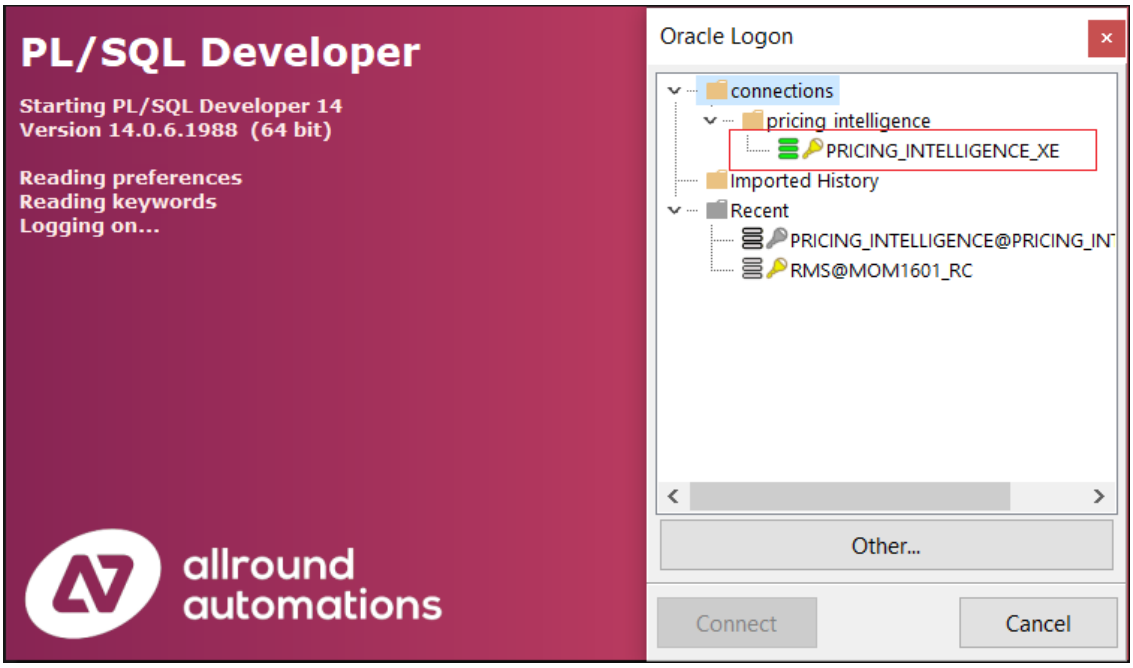

Figure 30 - PL/SQL Developer after the Configuration

After configuring both databases a DB link was set up in the Pricing Intelligence database so that it has access to MOM's database objects. After the right DB Link is setup it's possible to

access all the data in the RMS database and create the desired views. The “Abstraction Layer” is based on these views created through this DB Link, and that the connection of the solution to any other ERP only depends on the adaptation of this layer. This is an important step since we will namely need the data of the items and prices that are defined in the RMS database to be able to carry out and create the strategies. Below is an image of the configuration of the DB Link.

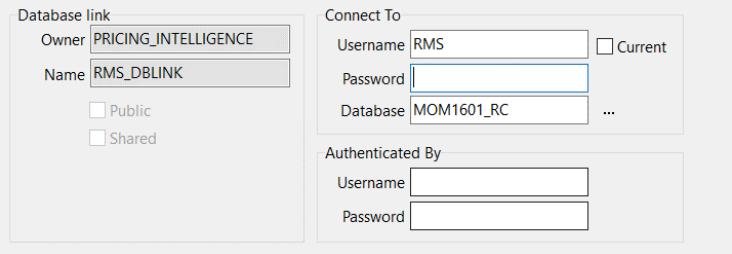

Figure 31 - RMS DB Link

4.1.2 Relational Model

In this section, the relational model of the solution will be displayed and described. The presented model contemplates all attributes that must be present in the database tables and aims to describe the system data and the relationships between entities. In the below figures it is possible to see the tables and relationships between them, the attributes and their primary and foreign keys.

4.1.2.1 Settings Relational Model

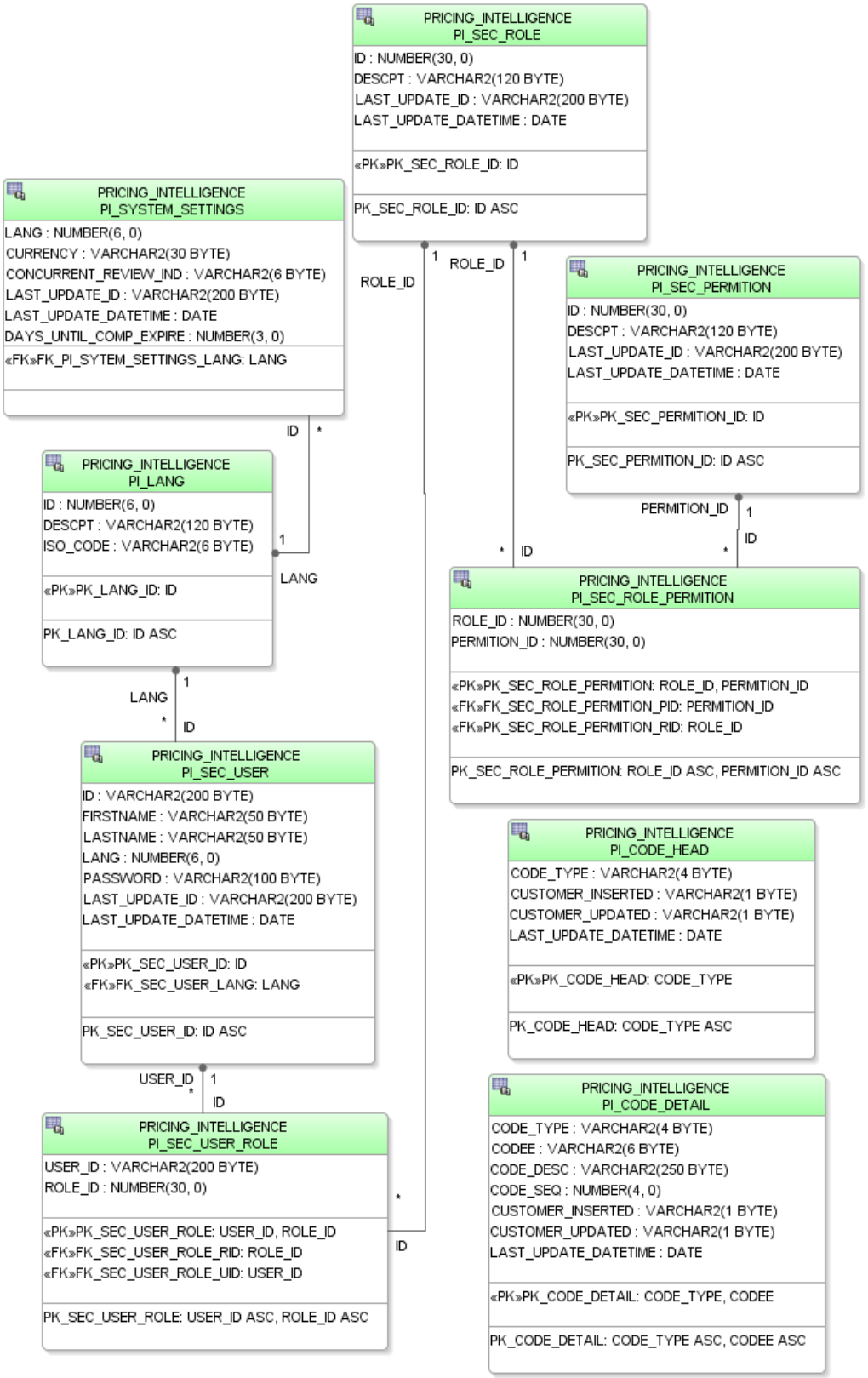

Figure 32 - Settings Relational Model

- PI_SEC_LANG - Table that contains all the languages used in the system;
- PI_SEC_USER - Table that contains the system users and the associated security/permissions;
- PI_SEC_ROLE - table that holds the distinct roles available in the solution;
- PI_SEC_USER_ROLE - table that holds the roles that each user has access to;
- PI_SEC_PERMISSION - table that holds the distinct permissions supported by the solution;
- PI_SEC_ROLE_PERMISSION - table that holds the permissions that each role has;
- PI_SEC_SYSTEM_SETTINGS - this table holds general system settings;
- PI_CODE_HEAD - This table holds different set of codes. This table is being used, for example, to avoid hardcoded, and to typify dropdown options;
- PI_CODE_DETAIL - table that contains the codes, detail and description for each code type defined in the CODE_HEAD TABLE.

4.1.2.2 Data Maintenance Relational Model

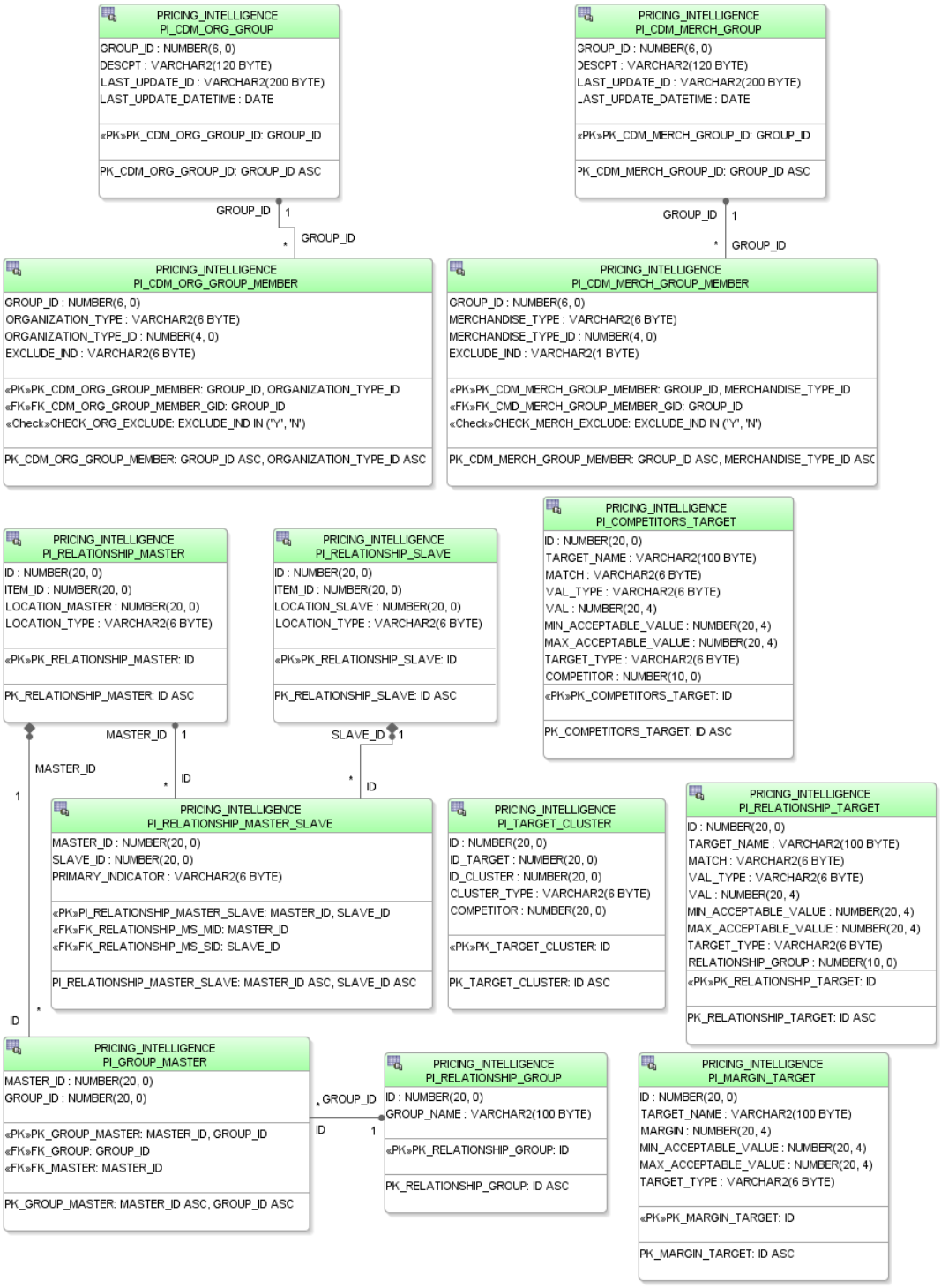

Figure 33 - Data Maintenance Relational Model

- PI_CDM_MERCH_GROUP - This table holds different defined merchandising groups, composed of branches of the merchandising hierarchy;
- PI_CDM_MERCH_GROUP_MEMBER - This tables hold the members that belong to one merchandising group (PI_CDM_MERCH_GROUP table);
- PI_CDM_ORG_GROUP - This table holds different defined organization groups, composed of branches of the organization hierarchy;
- PI_CDM_ORG_GROUP_MEMBER - This tables hold the members that belong to one organization group (PI_CDM_ORG_GROUP table);
- PI_MARGIN_TARGET - This table holds the targets for margin strategies;
- PI_COMPETITORS_TARGET - This table holds the targets for competitor's strategies;
- PI_RELATIONSHIP_TARGET - This table holds the targets for relationship strategies;
- PI_TARGET_CLUSTERS - This holds the clusters that belong to a target;
- PI_RELATIONSHIP_MASTER - This table holds the master items/locations of relationship strategies;
- PI_RELATIONSHIP_SLAVE - This table holds the slave items/locations of relationship strategy;
- PI_RELATIONSHIP_MASTER_SLAVE - This table holds the relation between master and slave records for relationship strategies;
- PI_RELATIONSHIP_GROUP – This table holds the groups of relationships for relationship strategies;
- PI_GROUP_MASTER – This table holds the relations between a group and master records for relationship strategies.

4.1.2.3 Strategy Management and Approval Relational Model

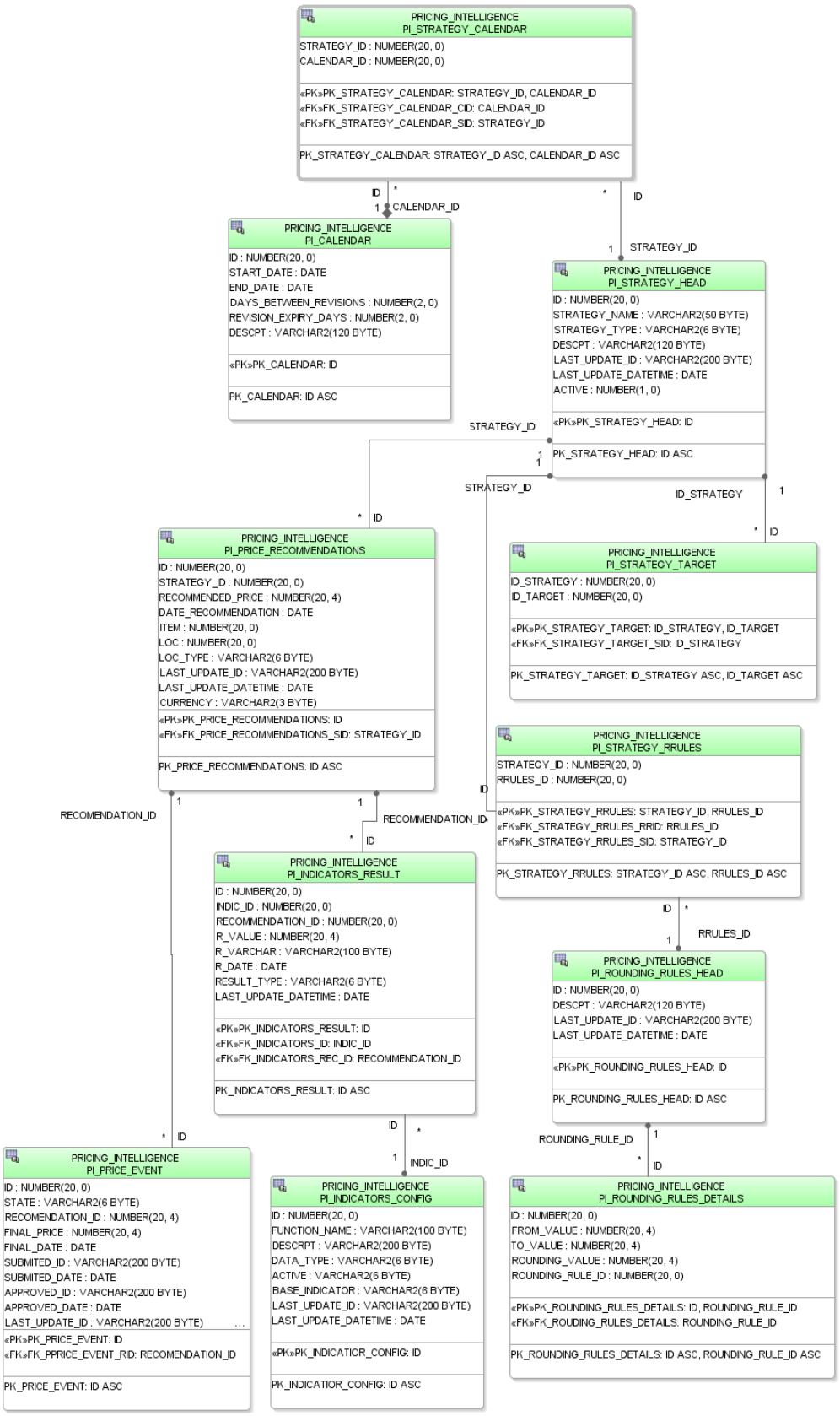

Figure 34 - Strategy Management and Approval Relational Model

- PI_CALENDAR - table that holds the calendar with important dates for the strategy price suggestion process;
- PI_ROUNDING_RULES_HEAD - This table holds the existing rounding rules;
- PI_ROUDING_RULES_DETAILS - This holds the details of each rounding rule defined in the PI_ROUNDING_RULES_HEAD table;
- PI_STRATEGY_HEAD - This table holds the main/common data of the strategy;
- PI_STRATEGY_CALENDAR - This table connects a calendar to a strategy;
- PI_STRATEGY_RRULES - This table connects rounding rules to a strategy;
- PI_STRATEGY_TARGET - This table connects the targets to a strategy;
- PI_PRICE_RECOMENDATIONS - This table holds the price recommendations generated by the strategies;
- PI_PRICE_EVENT - This table hold all price events(request/approvals) and the user related to the actions taken in the solution;
- PI_INDICATORS_CONFIG - definition of configurable indicators that complement the recommendations (like other relevant data / calculations);
- PI_INDICATORS_RESULT - Result for the calculation of each indicator.

4.2 Strategies

As defined in the analysis and design phase, three strategies were implemented in this solution:

- Margin strategy;
- Relationship strategy;
- Competitor strategy.

All strategies created/defined by users through the graphical interface will have recommended selling prices calculated by a nightly batch process meeting the NFR9 requirement and are based on the calendars associated with each strategy at the time of batch execution. Furthermore, it is taken into account the law decree, and all the pricing laws imposed are met, complying with the NFR17 requirement. Below there is a detailed description of each strategy and its algorithm.

4.2.1 Margin Strategy

The main objective of the margin strategy is to recommend prices based on previously defined markup targets. These targets define the targeted margin, the minimum value and the maximum acceptable margin value. So, when using the margin strategy for a given item, it is first necessary to check what type of margin should be calculate, cost markup or retail markup.

The formula for the Cost Markup:

$$Cost\ Markup = \frac{Selling\ Price - Average\ Cost}{Average\ Cost}$$

The Formula for the Retail Markup:

$$Retail\ Markup = \frac{Selling\ price - Average\ Cost}{Regular\ Unit\ price}$$

Depending on the type of markup, the current margin for that item is calculated, and then it is checked whether the margin is within the acceptable range. If the margin is outside the acceptable range, that is, below the minimum value or above the maximum value defined by the target, then it is necessary to calculate the Recommended Retail Price based on the target margin. Therefore, depending on the type of markup, the RRP is calculated as follows.

The formula for RRP with Cost markup:

$$RRP = \big(Average\ Cost * (1 + Targetted\ Margin)\big) * (1 + VAT)$$

The formula for RRP with Retail markup:

$$RRP = \frac{Average\ Cost}{1 - Tagetted\ Margin} * (1 + VAT)$$

After calculating the RRP, the rounding rule associated with the strategy is implemented, thus rounding the RRP according to the defined rules. This only applies if a there is a rule associated with the strategy in question. Finally, the recommended value is inserted in the database with the remaining following details: Strategy Id, Recommendation Date, Item, Location, Location Type and Currency. As shown in the image below, which represents the flow of the algorithm step by step, it is possible to see the price recommendation process based on the margin strategy.

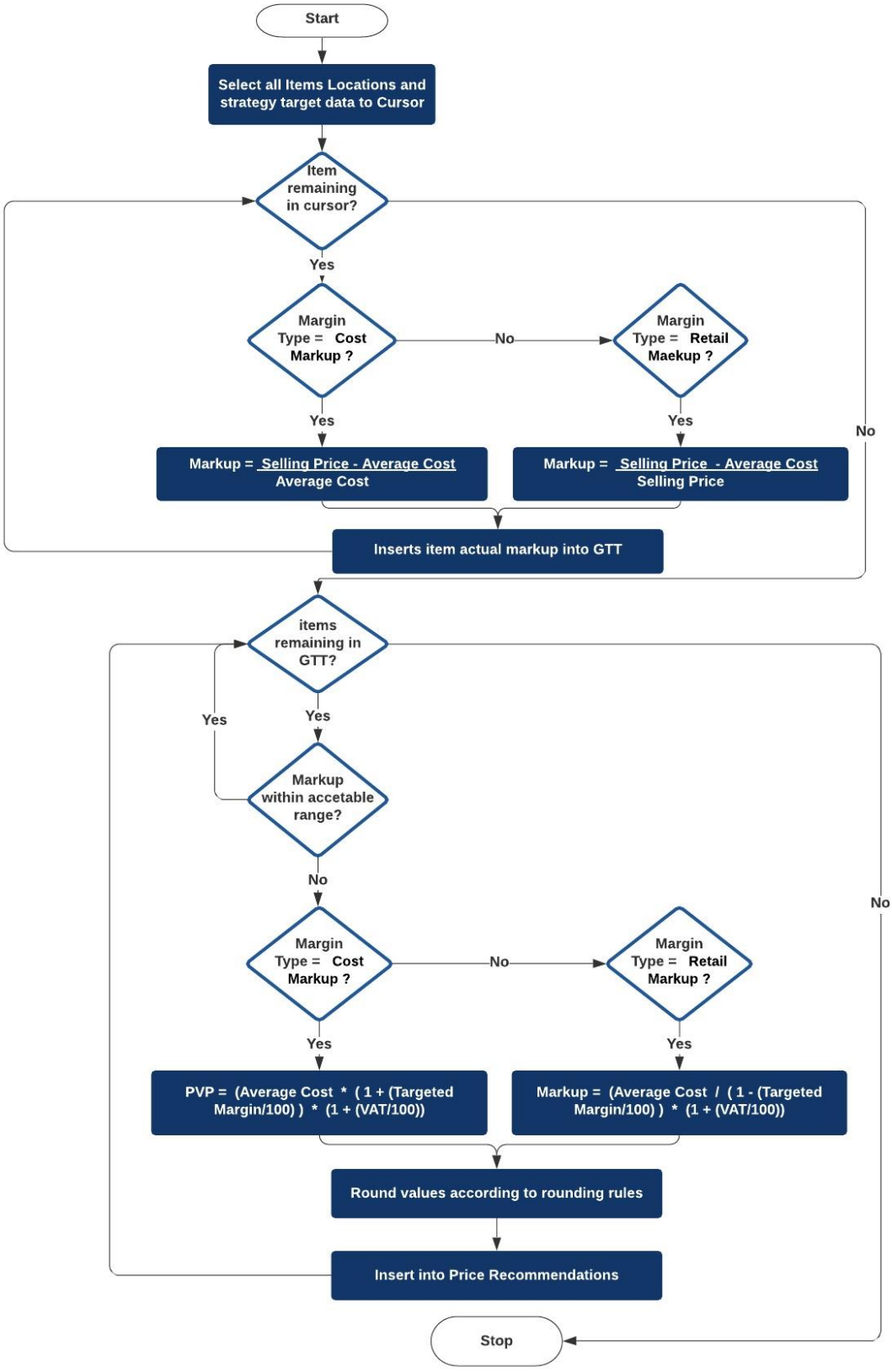

Figure 35 - Margin Strategy Flowchart

4.2.2 Competitor Strategy

The main objective of the Competitor's strategy is to recommend prices based on the price of a competitors' items. The target of the strategy defines whether the final price should be equal or not to the competitors' or whether it should be calculated based on a value (percentage or amount) above or below the competitor, and also which competitor we want to compare the price to.

So, if the target is set to march the competitor's price the function to calculate the Recommended Retail Price is as follows:

$$RRP = Competitors\ Retail\ Price$$

If the target is not to match but to calculate the Recommended Retail Price based on a value, the functions to calculate the RRP depends on whether we are dealing with a percentage or a value.

The formula for RRP based on a percentage value:

$$RRP = Competitor\ Price + \frac{Competitor\ Price * \pm Percentage}{100}$$

The formula for RRP based on a value:

$$RRP = Competitor\ Price + Value$$

After calculating the RRP, the rounding rule associated with the strategy is implemented, thus rounding the RRP according to the defined rules. This only applies if a there is a rule associated with the strategy in question. Finally, the recommended value is inserted in the database with the remaining following details: Strategy Id, Recommendation Date, Item, Location, Location Type and Currency. As shown in the image below, which represents the flow of the algorithm step by step, it is possible to see the price recommendation process based on the Competitors strategy.

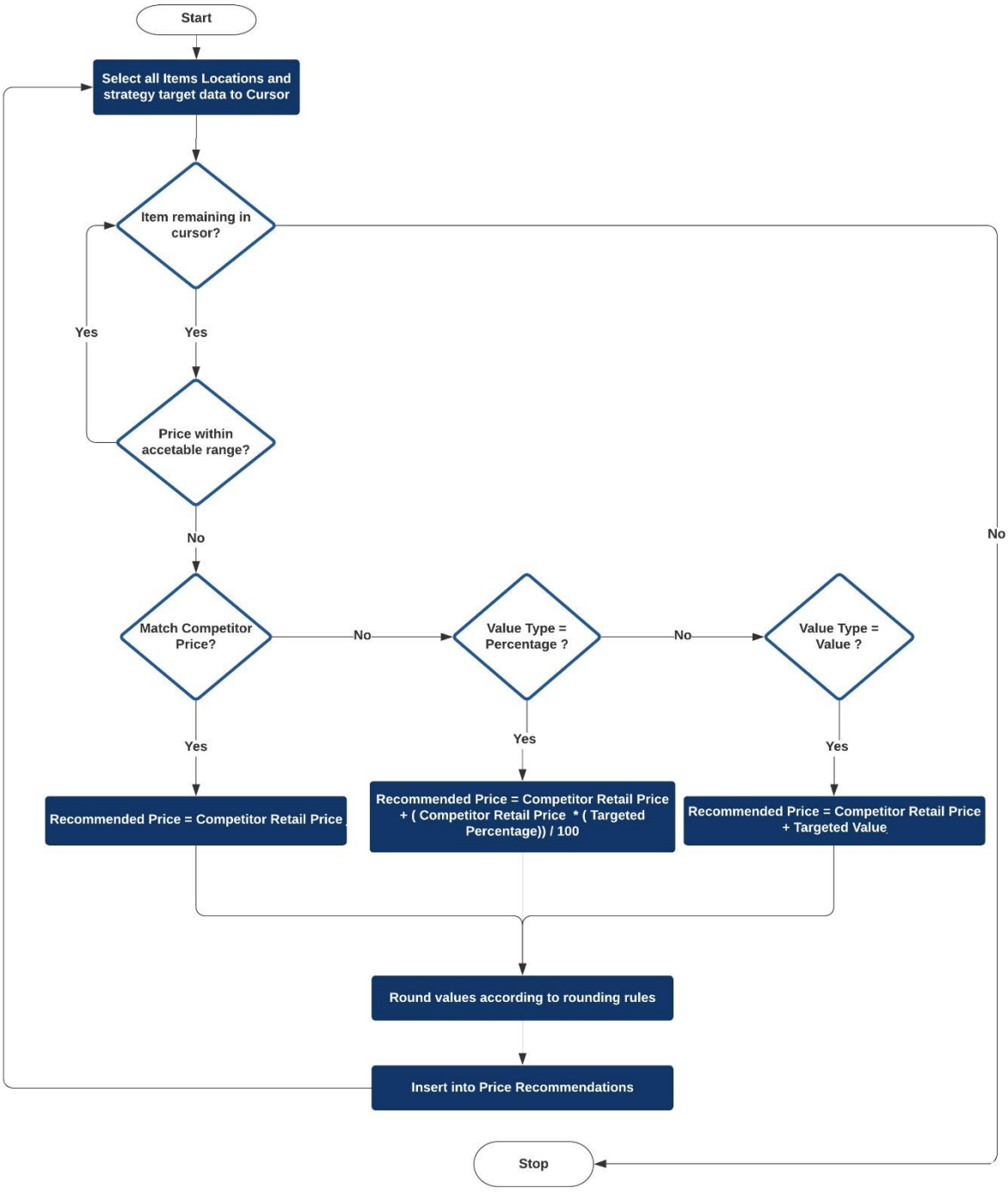

Figure 36 - Competitor Strategy Flowchart

4.2.3 Relationship Strategy

The main objective of the Relationship Strategy is to recommend prices based on relations between items in different locations. The relationships between these items are previously created, where a master item can be associated to different slave items, and the slave RRP will be calculated based on the master price. The target of the strategy defines whether the final price should be equal or not to the master's price or whether it should be calculated based on a value (percentage or amount) above or below the masters, and also which master we want to compare the price to. The logic of this strategy is very similar to the Competitor strategy but the principle is the opposite. The big difference is that the competitor strategy has external drivers (the supposed competitors), and the relationship strategy has internal drivers (the retailer's own items).

So, if the target is set to march the master's price the function to calculate the Recommended Retail Price is as follows:

$$RRP = Master\ Retail\ Price$$

If the target is not to match but to calculate the Recommended Retail Price based on a value, the functions to calculate the RRP depends on whether we are dealing with a percentage or a value.

The formula for RRP based on a percentage value:

$$RRP = Master\ Price + \frac{Master\ Price * \pm Percentage}{100}$$

The formula for RRP based on a value:

$$RRP = Master\ Price + Value$$

After calculating the RRP, the rounding rule associated with the strategy is implemented, thus rounding the RRP according to the defined rules. This only applies if a there is a rule associated with the strategy in question. Finally, the recommended value is inserted in the database with the remaining following details: Strategy Id, Recommendation Date, Item, Location, Location Type and Currency. As shown in the image below, which represents the flow of the algorithm step by step, it is possible to see the price recommendation process based on the Relationship Strategy.

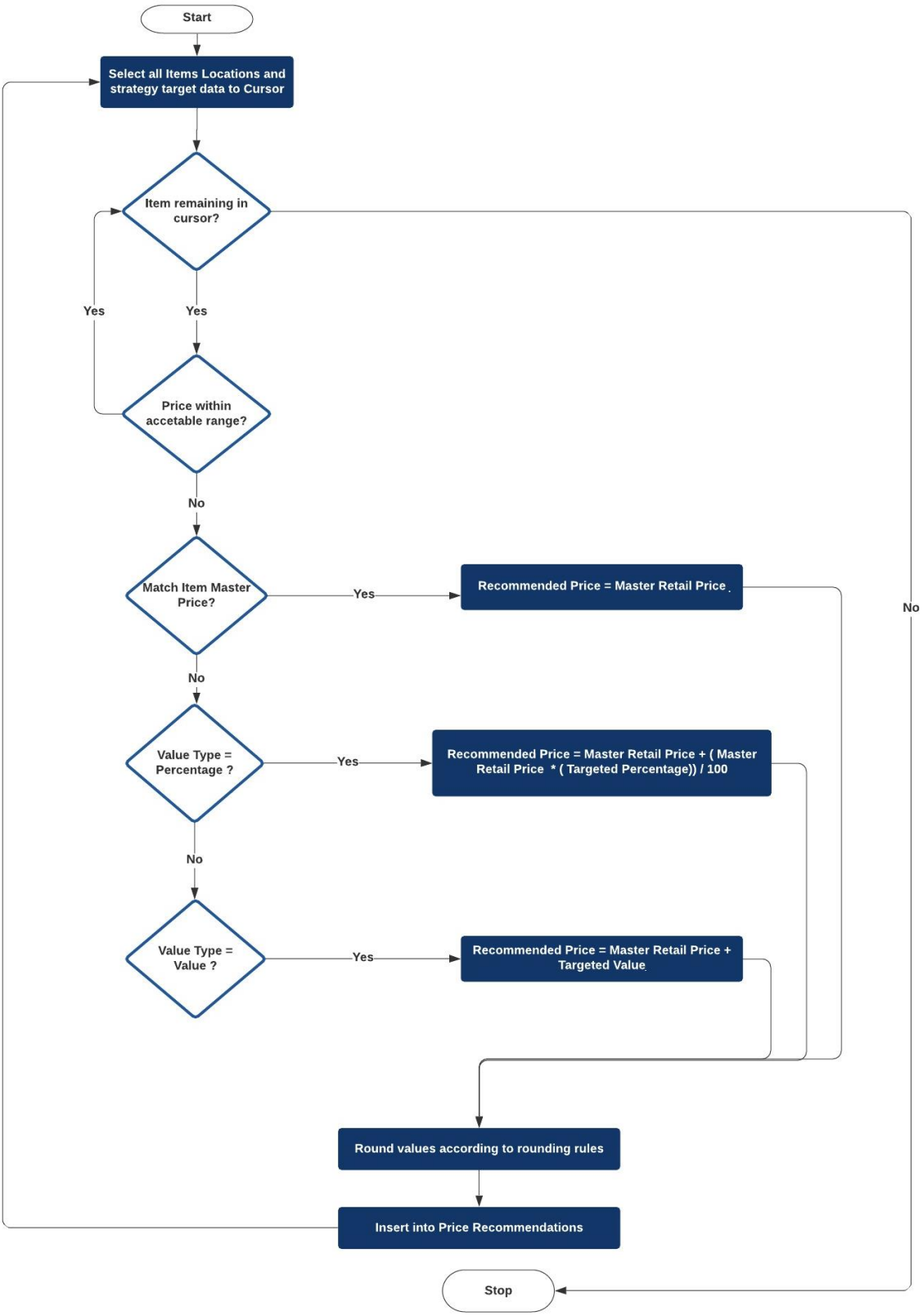

Figure 37 - Relationship Strategy Flowchart

4.3 User Interface

This chapter presents the interface created based on screenshots taken and a brief explanation of the tasks that are available. All the interfaces were made to present the functionalities and operations as expected and with a good structure that allows users to access contents in an intuitive way aiming for a user-oriented interface which meets the requirement NFR1, NFR2 and NFR5.

4.3.1 Login

The login page, which meets the requirement UC2, NFR15 and NFR16, is shown in the figure below. This page is the first to be presented to the user in case the user is not logged in. For authentication in the solution, only users with the appropriate credentials can login. These credentials must be previously created by the System Administrator.

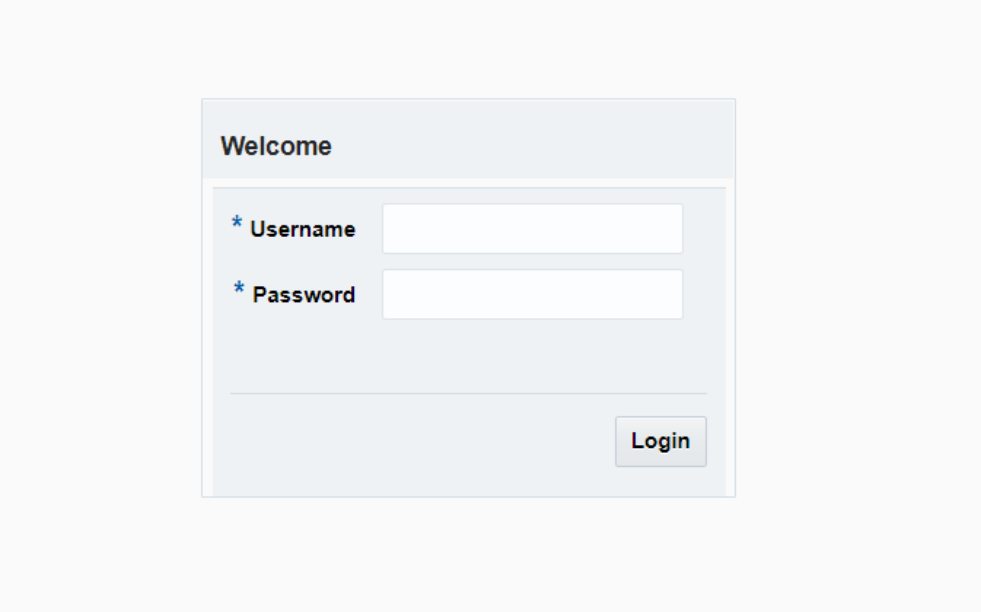

Figure 38 - Login Page

In case of invalid data or error, the appropriate warning will be presented to the user, for example, mandatory fields to be filled, wrong username or password or invalid symbols which meets the requirement NFR4 and NFR6.

If the login is successful, the user is redirected to the solution's homepage where he will have access to different actions according to his assigned roles and corresponding permissions.

4.3.2 Main Page

In the figure below it is possible to see the main page of the solution. With the proper login, this page will be displayed and will always be available to the user.

Figure 39 - Main Page

In the upper right corner, the user has access to their settings and can also log out, on the left side of the page is the menu and, in the center, will be where all tabbed pages appear when selected through the menu. The user can also have more than one tab open and move between them freely.

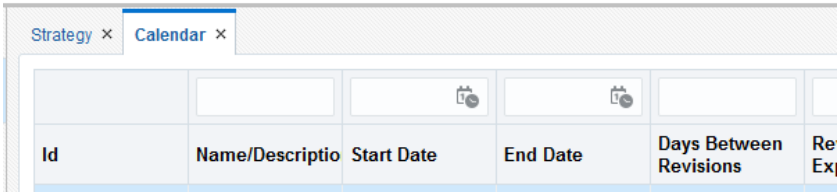

Figure 40 - Tabbed Pages

The following figure shows the main menu that allows to select the desired option and that is used for navigation in the solution. From this menu it is possible to have access to all available pages based on the user's permissions. If a user does not have access to a page or section, it appears in a grayed out tone that indicates it is disabled.

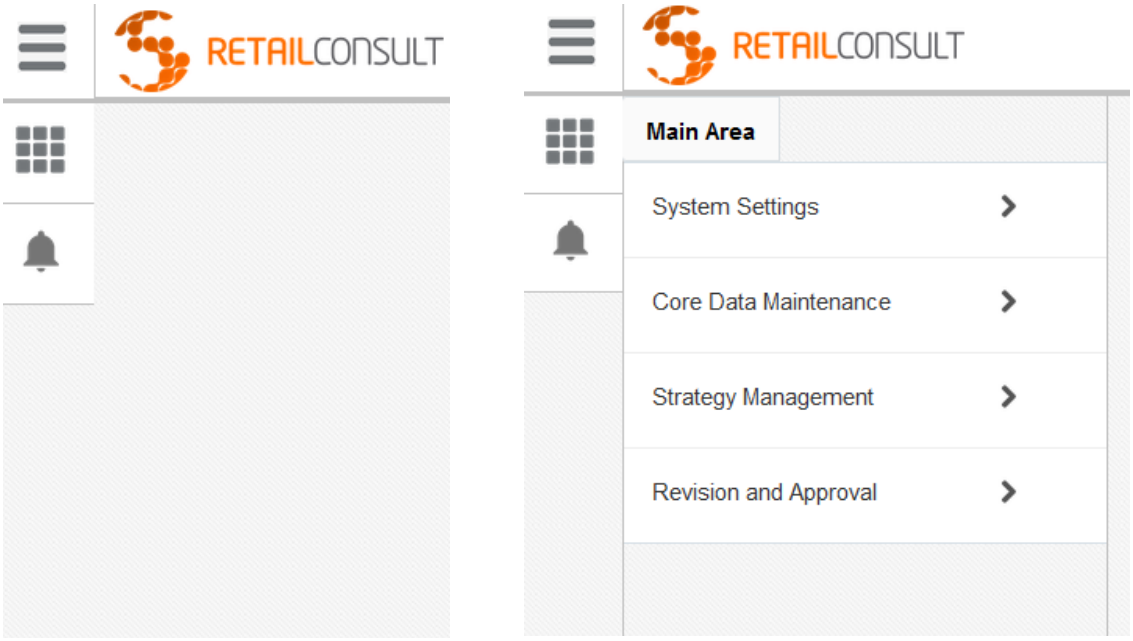

Figure 41 - Minimizable Menu

This menu is developed according to the tree navigation pattern, which presents the menu according to the hierarchical relationship between the options. There is also the possibility of minimizing and maximizing the menu according to the need for space on the page.

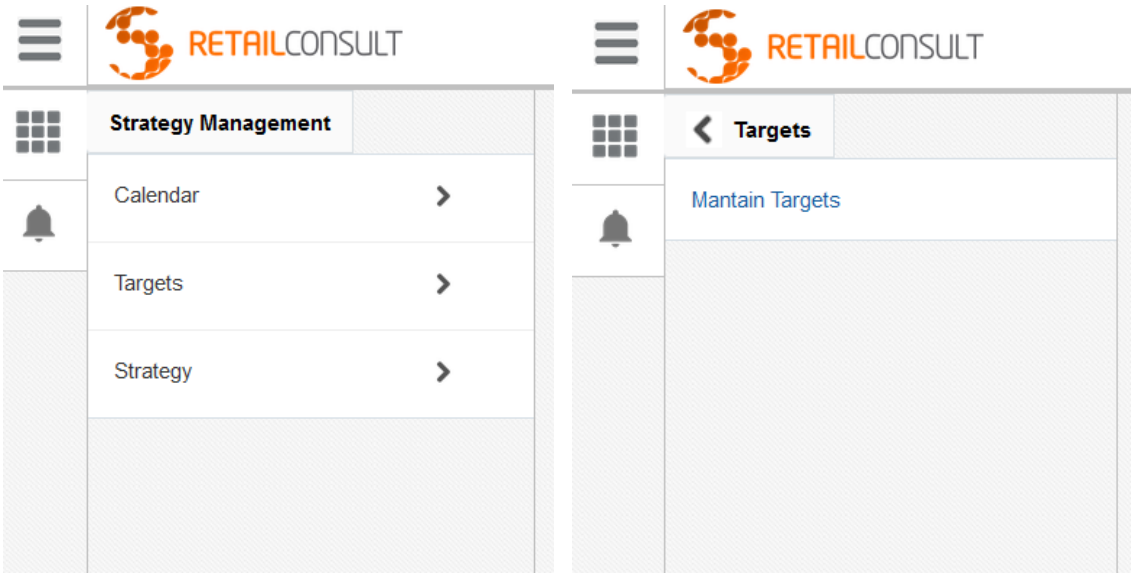

Figure 42 - Different levels of the Tree Menu

4.3.3 System Settings

The General Settings and Security Configuration page that meet the requirement UC1, UC2 AND UC3 allow to adjust settings that will affect how the solution works, such as language, country currency, and so on. In addition, it also allows for the creation of users, roles, and permissions.

For reasons of time and because there were other requirements with more importance/relevance to the overall solution, the interface, that is, the page where these settings can be changed, has not been developed. At this moment, the data tables relating to the configurations and the price recommendation engine are finished and the data from these tables is being used; all that remains is to finish the pages that allow for this data to be maintained via a graphical interface.

4.3.4 Maintain Core Data Maintenance

4.3.4.1 Define Relationships Between Items

On the Items Relationship page, that meets the functional requirement UC6, it is possible to create different relationship between item and locations, where one will be the master item and the other the slaves. The price recommendations for the slave items will then be based upon the master's price according to strategy specific settings. On the home page, the table is presented with the data of the created relations masters, where it is possible to see the Item Id, the Master location and the location type.

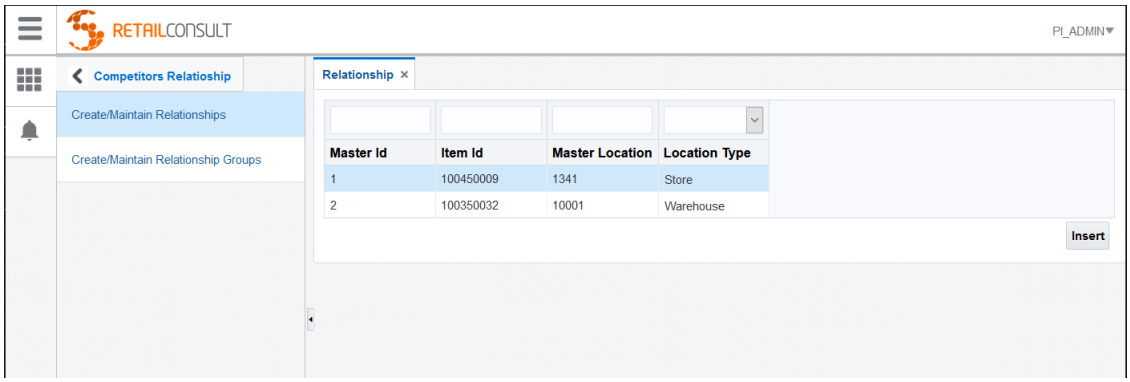

Figure 43 - Items Relationship Page

When selecting the insert button, a popup like the image below is displayed where it is possible to insert or delete relationships. In this popup, the user can download an excel file where we can see all the masters and slaves and choose to insert or delete. After altering the excel, the user should upload the file again and all changes will be made according to the data in excel - this is a commonly used mass data maintenance pattern used in the Oracle Retail solutions.

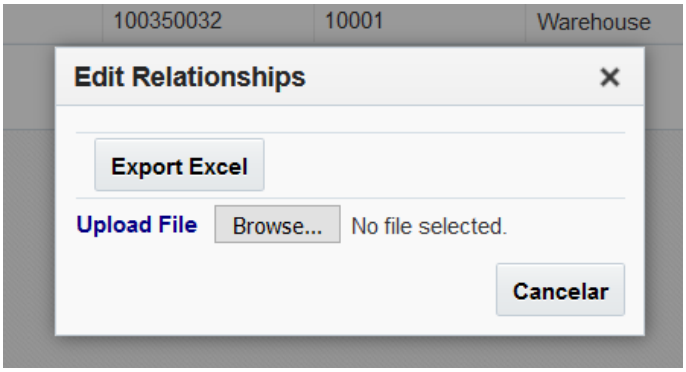

Figure 44 - Add New Relationship Popup

In the image below is a screenshot of the excel file, where in the first column the user can select Create or Delete. The create option will create a new relationship, and the delete option will delete the relationship. If the field is empty, the line in question will not change.

The other columns in the file are related to master and slave data, such as Id, location and Location type (Warehouse or Store).

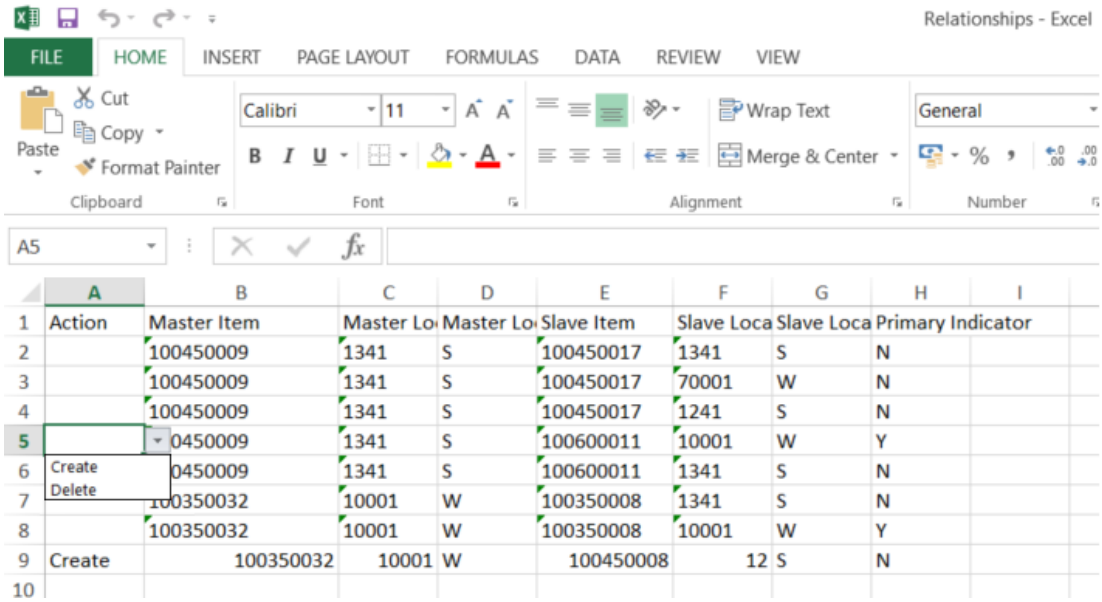

Figure 45 - Excel File to Edit Relationships

As targets can have more than one relation and more than one slave, to make the association of masters to a target easier, another page was created for the creation of relation groups. On this page we can see the data of all groups that already exist, such as the group name and if the group is associated with a target.

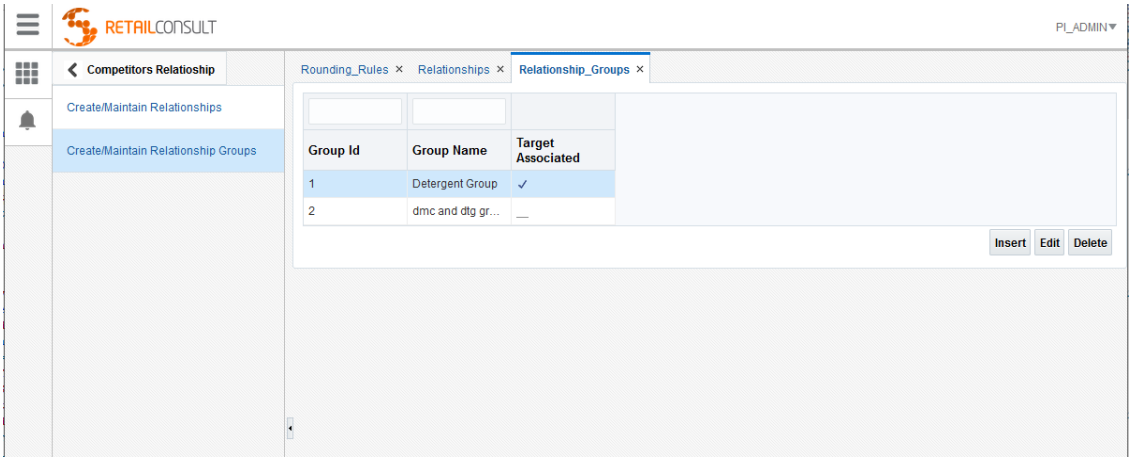

Figure 46 - Relationship Group Page

When selecting the insert button, a popup like the image below is displayed where it is possible to create a group by giving the group a name and by associating different masters. To associate the masters to a group it presented a shuttle where in the left panel appear all the masters available and by moving them to the right the master will be associated to the group. By selecting the "OK" button, the group is then saved, and the popup closed.

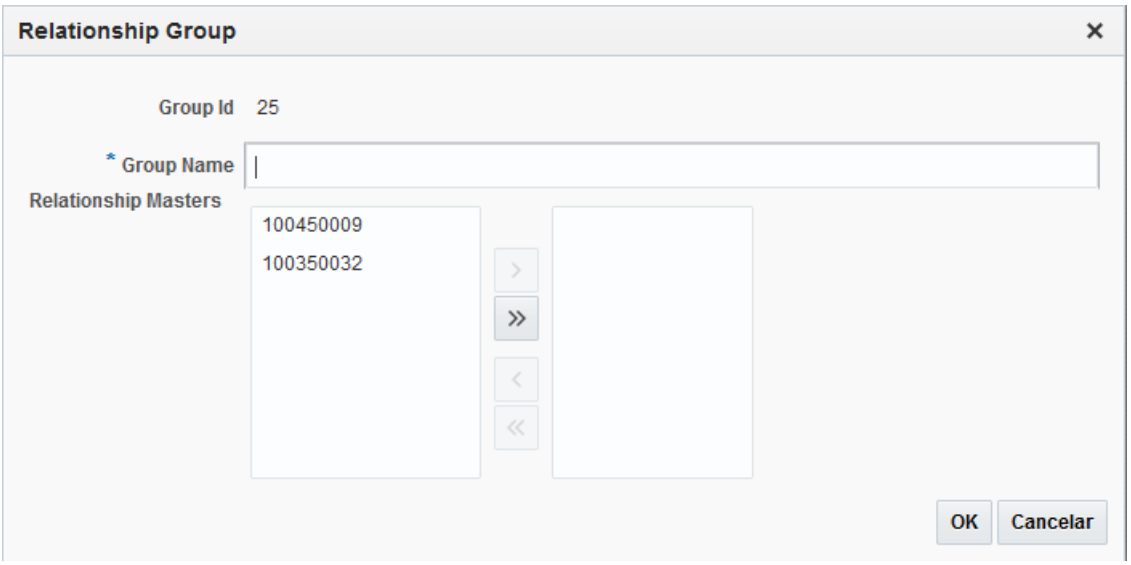

Figure 47 - Add New Relationship Group Popup

When selecting the edit button a similar popup appears, the only difference being that the information is already filled, and the user can change the information as he/she wishes. The delete button shows a popup to verify if the user really wishes to delete that group.

4.3.4.2 Create Merchandising Hierarchy Groups

On the merchandising hierarchy page, that meets the functional requirement UC4, it is possible to create groups with different merchandising hierarchies. This feature is fully functional from a data model perspective, as part of the strategy setup and price engine, but the main page to

create the groups was not developed due to time constraints. Initially the requirements were designed for a team of about five people, but for reasons outside of our control, the solution ended up being made by only one person, which led to having to put some aside requirements considered as non-critical for a first release.

4.3.4.3 Create Organization Hierarchy Groups

On the merchandising hierarchy page, that meets the functional requirement UC5, it is possible to create different relationship between different merchandising hierarchies. Like the merchandising group page this functionality is running in the background but the main page to create the group is also not available for the same reasons presented above.

4.3.5 Strategy Management

In this section are presented all the pages that are related to the management and maintenance of strategies, and that meet the functional requirements UC7, UC8 and UC9.

4.3.5.1 Maintain Strategy Targets

The next page, called Maintain Strategy targets, aims to manage all created targets and meets the functional requirements UC7, UC8 AND UC9 because the definition of the targets for the different strategies is an important step in their creation and maintenance. On this page, it is possible to create three types of targets according to the intended strategy: margin target, competitor's' target or Relationship target. On the home page, a table with all the existing targets is presented. This table, in addition to showing the main information related to the targets, also informs whether a target is linked to strategies or not.

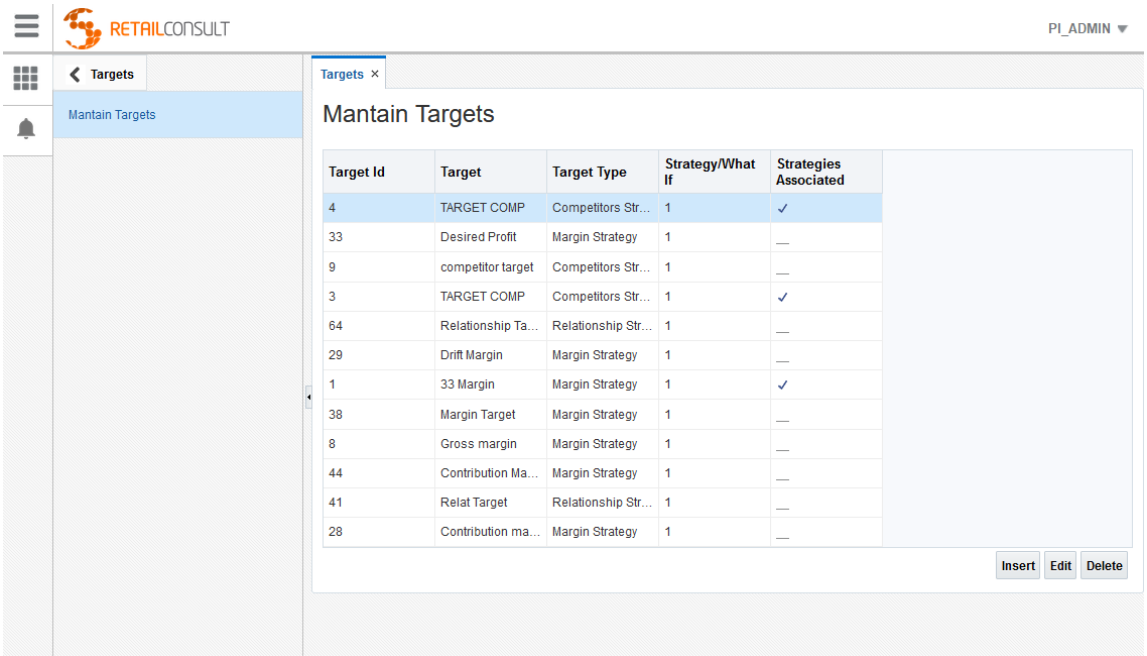

Figure 48 - Maintain target Page

When selecting the insert option, a popup with a train is activated. A train represents a progression of related activities that guides an end user to the completion of a task. In this case the train will have tree stops that guide the user in the process of creating a new target. In the first stop of the train, the user is asked to select the type of target to be created, having three options: Margin Target, Competitor Target or Relationship Target.

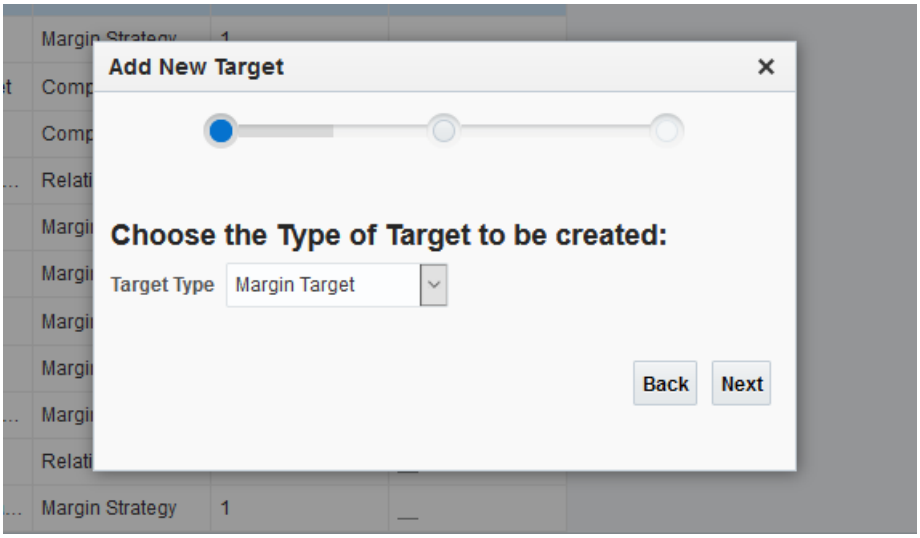

Figure 49 - Add New Target Popup

The next train stops that are presented to the user depend on this first selection, for example when selecting the margin target at the next stop the form that will be presented to the user is the form related to the margin target.

UC7 - Maintain Margin Strategy

In case the user selected a Margin target the form presented will be the one shown below and that complies with the requirement UC7. Where the user is asked to choose the target name, indicate the margin, indicate the minimum and maximum acceptable value for that target and the type of target (recommendation target or what if analysis).

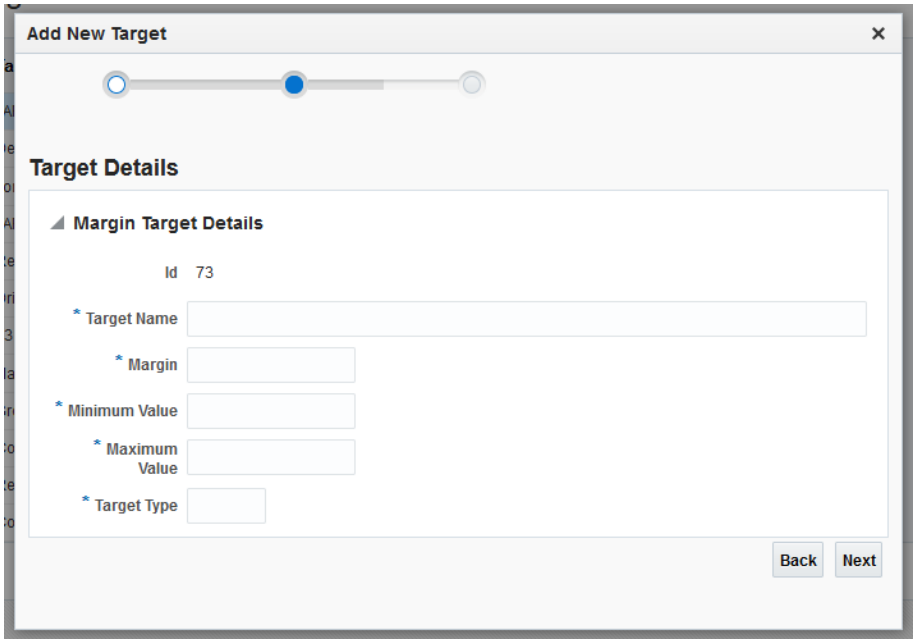

Figure 50 - Add New Margin Target Popup

UC8 - Maintain Competitor Strategy

In case the user selected a Competitor target the form presented will be the one shown below and that complies with the requirement UC7. Where the user is asked to choose the target name, indicate if it's supposed to match the competitor value or not, indicate the type of value (percentage or value), indicate the value, indicate the minimum and maximum acceptable value for that target, choose the type of target (recommendation target or what if analysis) and choose the competitor.

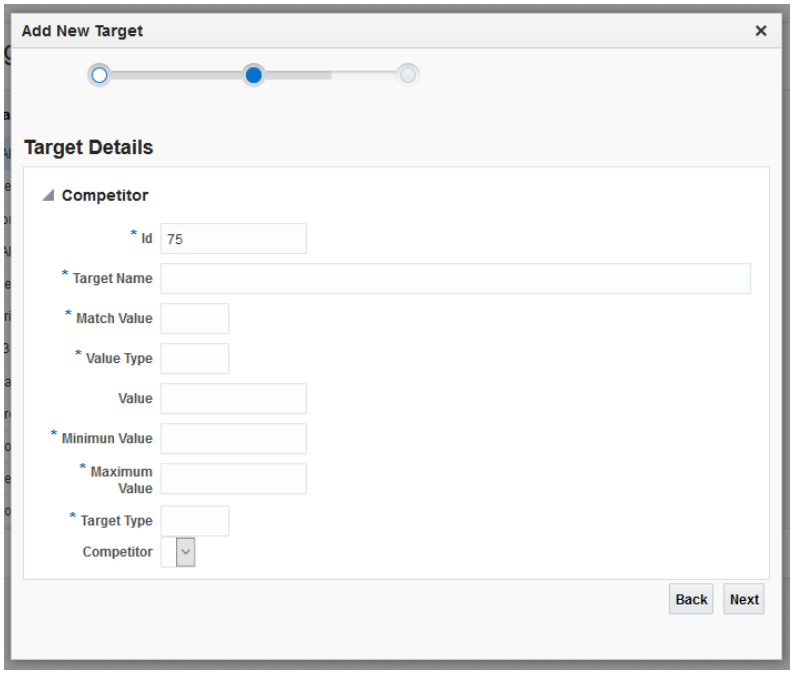

Figure 51 - Add New Competitor Target Popup

UC9 - Maintain Relationship Strategy

In case the user selected a relationship target the form presented will be the one shown below and that complies with the requirement UC9. Where the user is asked to choose the target name, indicate if it is supposed to match the related item value or not, indicate the type of value (percentage or value), indicate the value, indicate the minimum and maximum acceptable value for that target and choose the type of target (recommendation target or what if analysis).

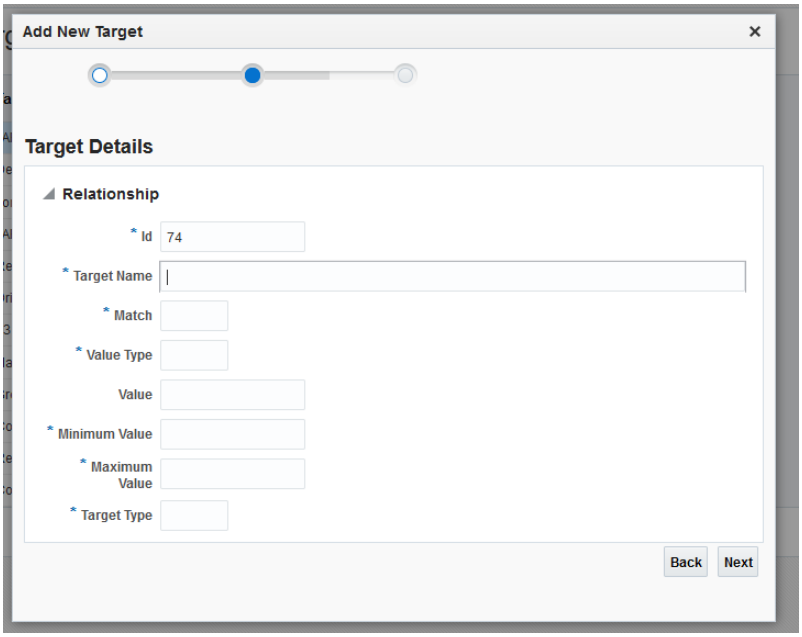

Figure 52 - Add New Relationship Target Popup

And finally, the last stop on the train, which is the same for all types of targets, features two different shuttles. A shuttle for the organizational clusters and a shuttle for the merchandising clusters that were previously created. Here the user can select the organizational and merchandising groups he wants to associate with the targets by moving them to the right column.

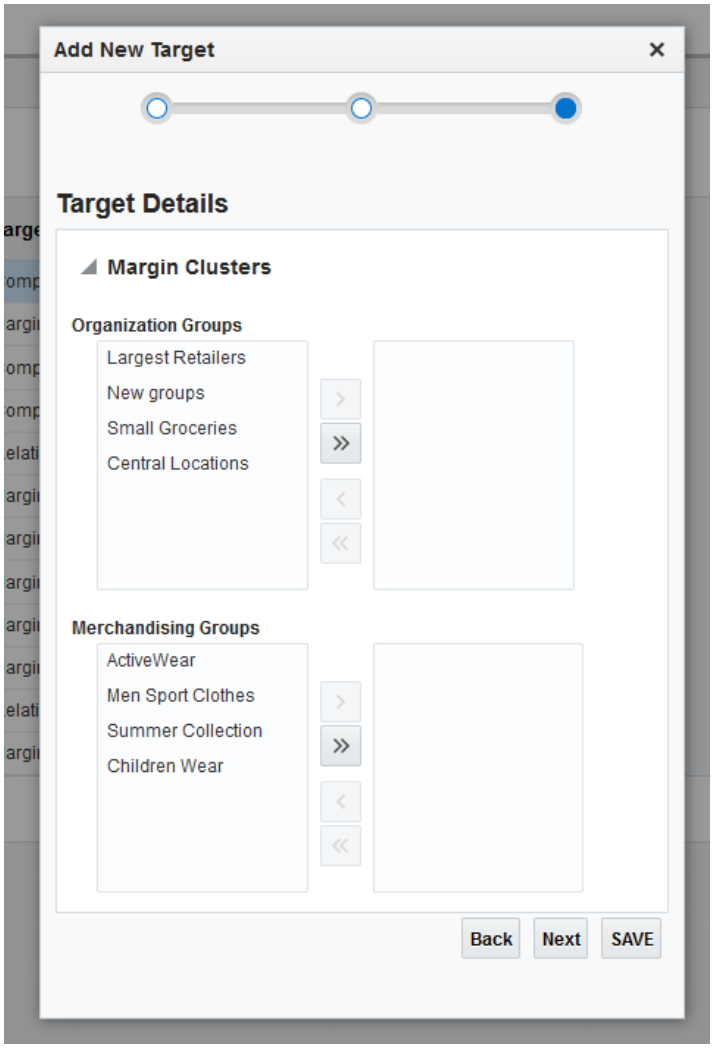

Figure 53 - Add New Target Clusters Popup

When selecting the edit button to edit a target, the user is automatically taken to the second stop of the train as it is not necessary to select the target option. In this option the user can edit the target form and edit the associated clusters.

By selecting a target to eliminate, a popup is presented with a table of the strategies associated with that target that will be affected and which asks the user for confirmation to eliminate the selected target.

4.3.5.2 Strategy Calendar

On this page the user has the possibility to manage all the calendars that will be associated with the strategies. When opening the main calendar page, a table is presented with the data of the existing calendars and a column that identifies whether or not a calendar is associated with strategies.

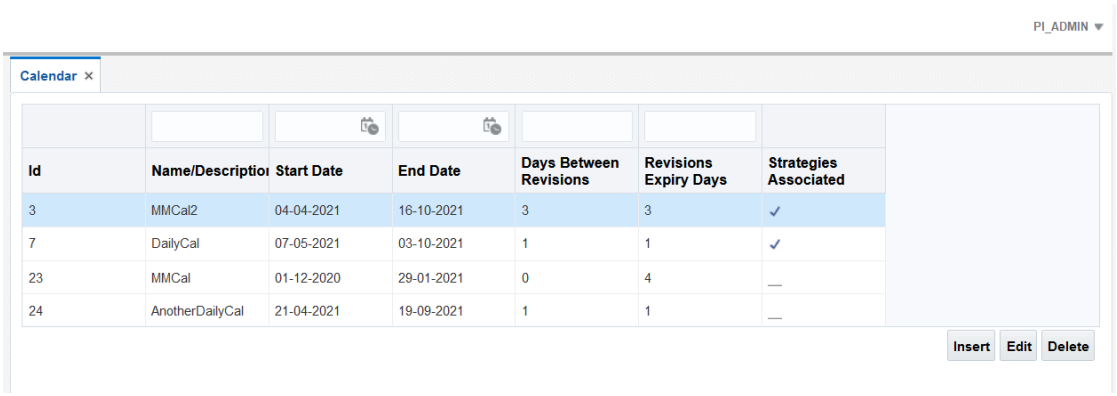

Figure 54 - Maintain Calendar Page

At the bottom of the page, 3 buttons are available.

The insert button that displays a pop-up where you can create a new Calendar by entering the following data:

- **Name / Description** - for better identification of the calendar;
- **Start date** - date on which the strategy is to be started;
- **End date** - end date of the strategy, this being deactivated;
- **Days between revisions** - how often should the strategy be calculated;
- **Revisions expiry dates** - the number of days that a recommendation is valid after it has been calculated.

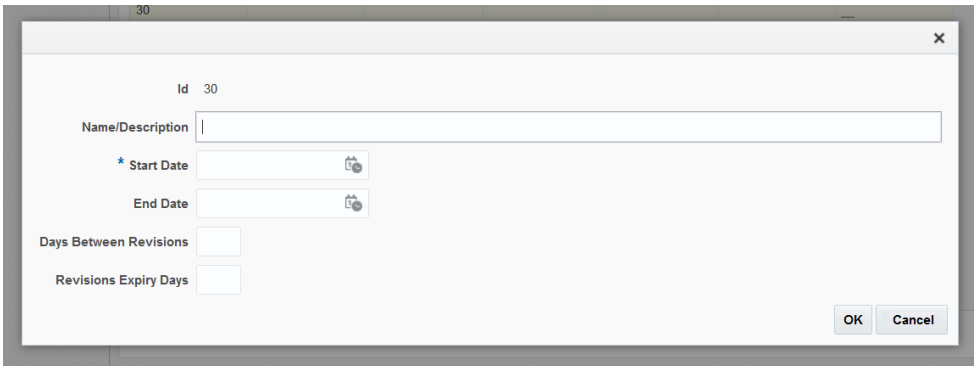

Figure 55 - Add New Calendar Popup

The edit button displays the same popup as the insert button, but in this case the data is filled with the data from the calendar previously selected in the table. The user has the possibility to edit all data except the calendar id. A small table is also presented with the associated strategies so that the user can be aware of which strategies will be affected in the event of a change.

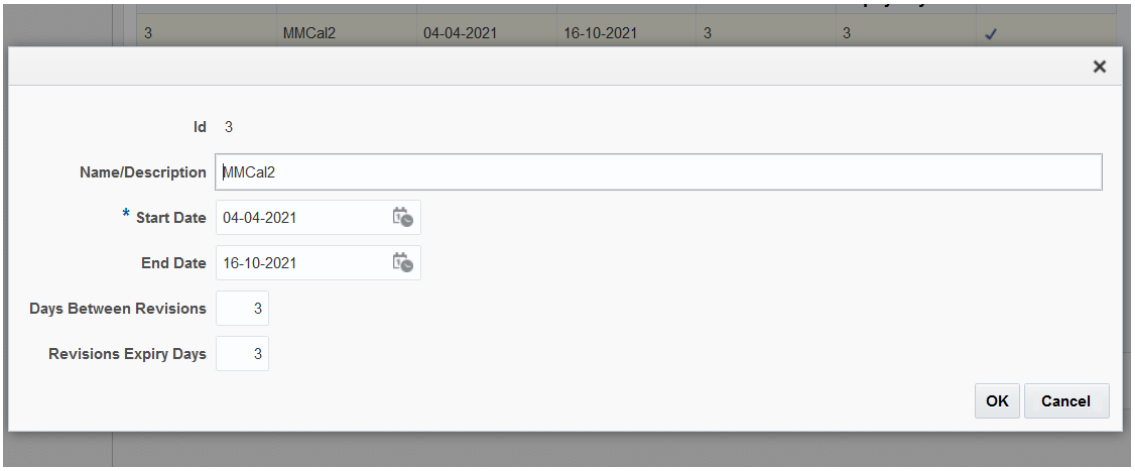

Figure 56 - Edit Calendar Popup

When selecting a row from the table and clicking the delete button, a popup is activated that asks the user for confirmation to delete the calendar. The table with the strategies to which the calendar is associated is also presented.

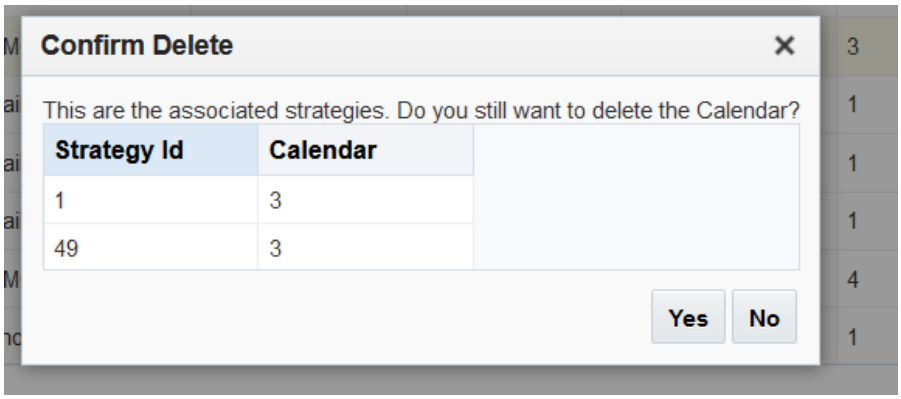

Figure 57 - Eliminate Calendar Popup

4.3.5.3 Strategy Rounding Rules

On the Rounding Rules page, it is possible to create different rounding rules that can then be associated with strategies. On the home page, a table is presented with the data of the existing rounding rules and a column that identifies whether the rounding rules are associated with strategies.

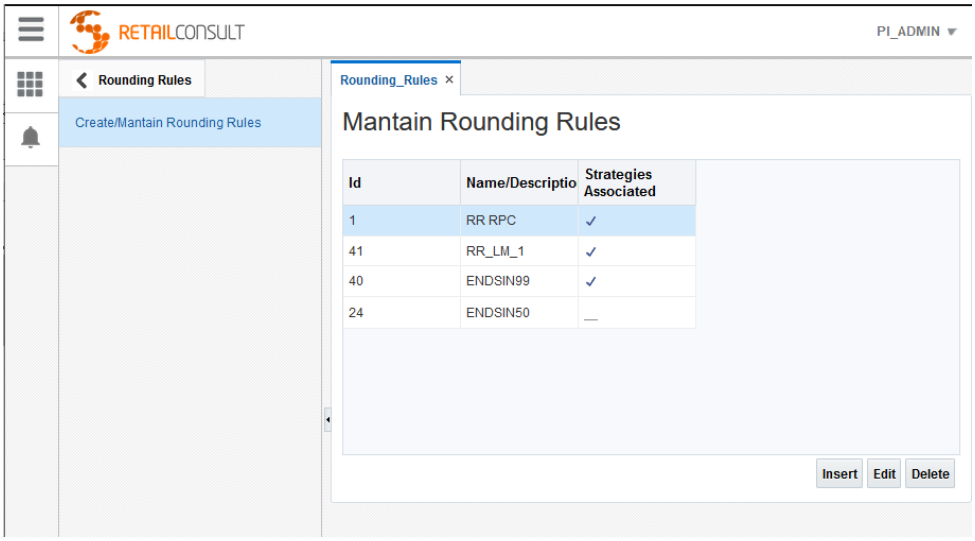

Figure 58 - Maintain Rounding Rule Page

When selecting the insert button, a popup like the image below is displayed where it is possible to insert a new rounding rule. In this popup, a form is presented asking for the name/description of the rounding rule to be inserted.

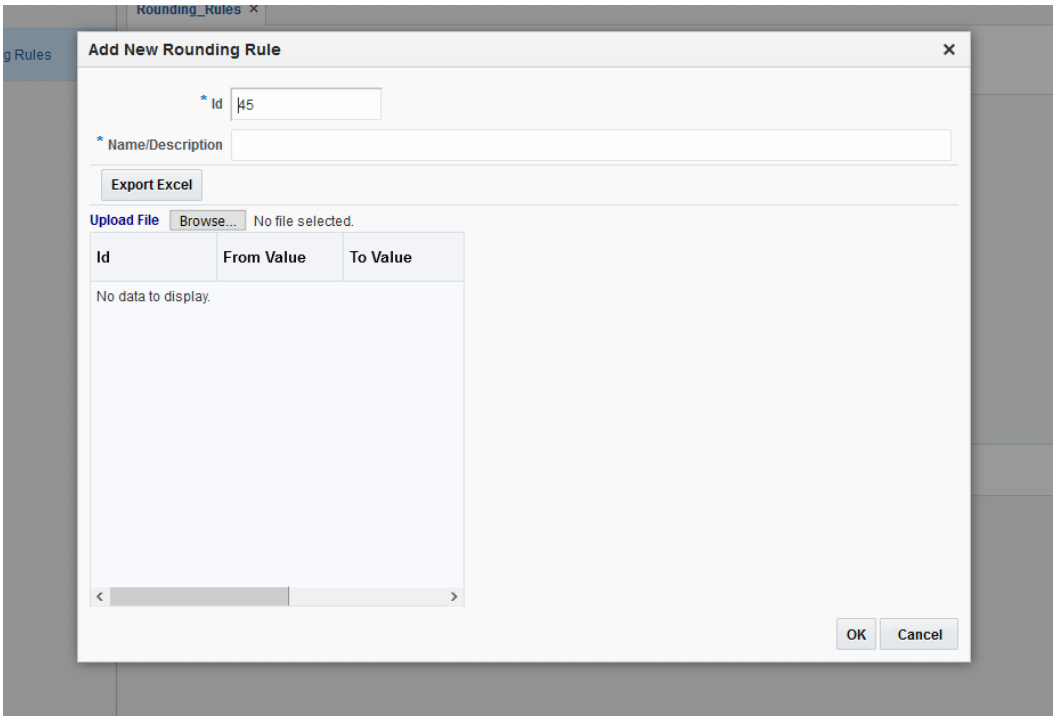

Figure 59 - Add New Rounding Rule Popup

To facilitate the insertion of rounding rules data, since there may be many rounding rules, it was decided to insert the rules through an excel file. In the same popup there is a button called export excel which allows us to download the excel table that we must fill out. In the figure below it is possible to see the excel that must be filled out. This file as 3 fields: from value, to

value and rounding value. So, when a strategy is being calculated and the recommended price is between the “from value” and the “to value” the final implemented recommended price will be rounded to the rounding value.

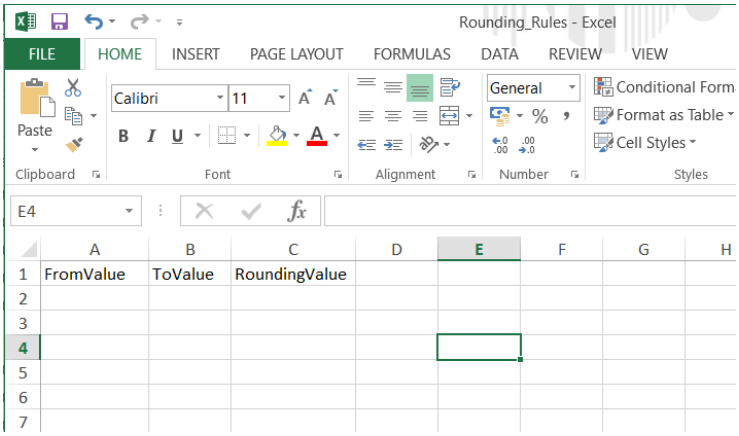

Figure 60 - Exported Excel

After filling the Excel file, in the solution we select the option “Upload File – browse” where we select the modified file. After selecting and inserting the file, the table will be updated and will present the data entered in the Excel. It only remains to select the ok option to finalize and save the created rounding rule.

When selecting a row in the table and clicking on the edit button, a popup will be displayed just like inserting with the data of the selected rounding rule.

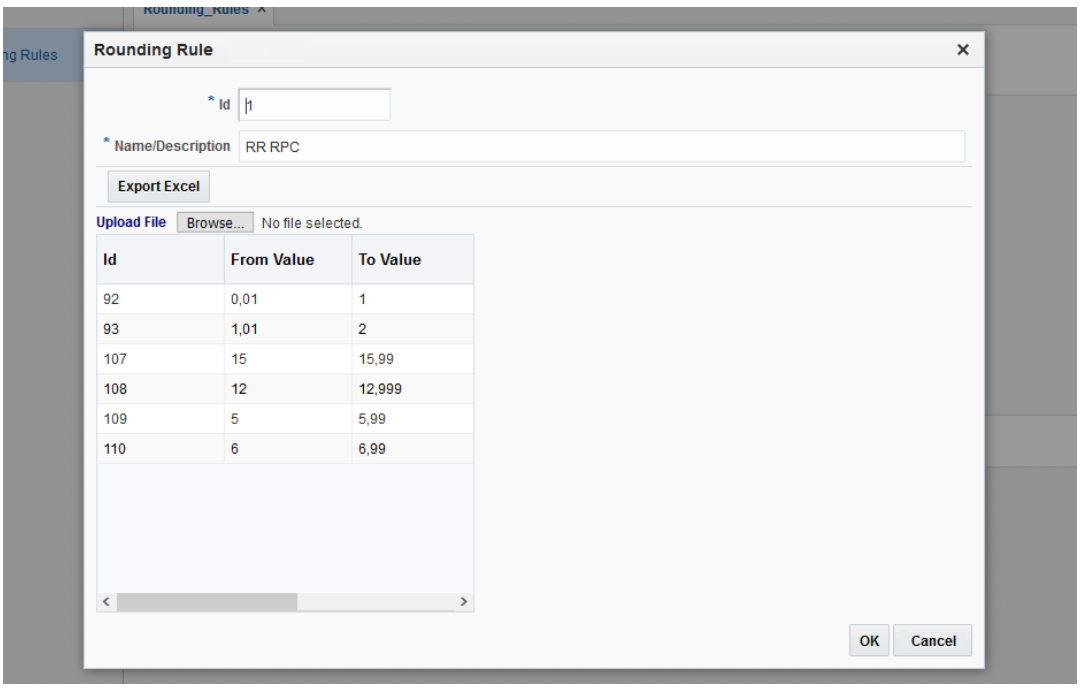

Figure 61 - Edit Rounding Rule Popup

As we can see in the image below, in this case the table is already filled and shows the rounding data previously created. This time, when exporting the excel, has shown below, this will be filled with the same data from the table allowing the user to edit that data or simply delete and add new one.

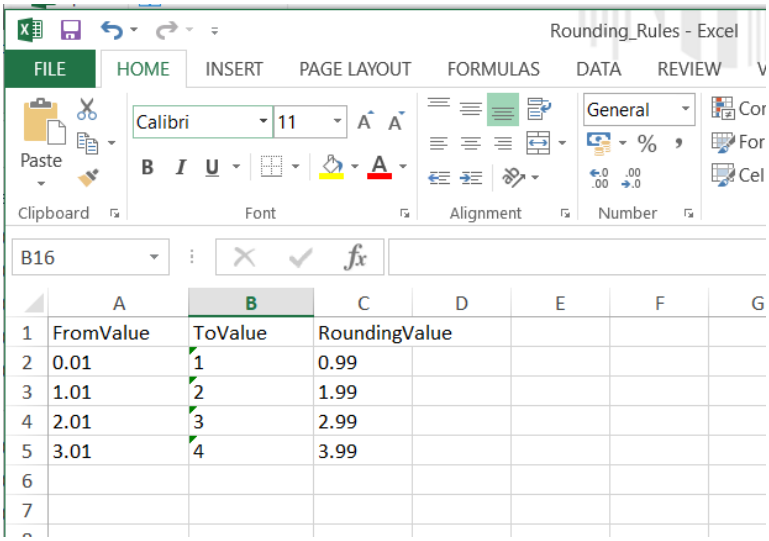

Figure 62 - Exported Excel to Edit

It is possible to edit or insert more data in the table, and after inserting the modified excel data in the solution the table will update accordingly to the new data. To finish it is only necessary to select the “OK” option and the data will be updated.

When selecting a row in the table and clicking the delete button, a popup is activated that asks for confirmation from the user to delete the rounding rule. In this option, the table with the strategies to which the rounding rules are associated is presented for the user to be aware of the strategies that are affecting them. By selecting the ok option, the rounding rule is eliminated.

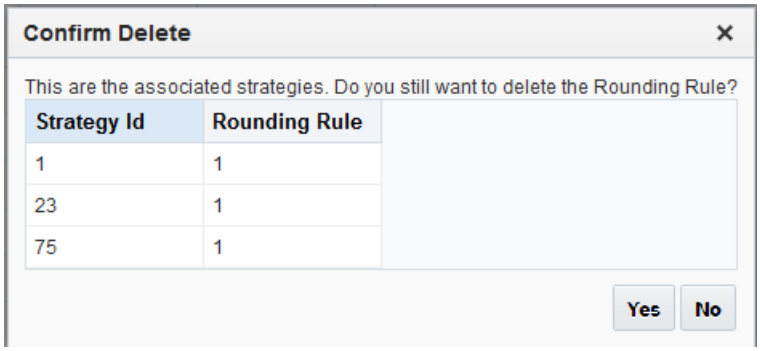

Figure 63 - Delete Rounding Rule Popup

4.3.5.4 Maintain Strategy

On the management page of the strategies that meets the requirements UC7, UC8 and UC9, the objective is to manage and maintain the different, where the user can create a new strategy based or edit an existing strategy. Here is presented a table with the data of all the existing

strategies, from margin, competitor or relationship. In addition to the strategy data, this table has an editable column with a checkbox that indicates whether a strategy is active or not, it is possible to activate or deactivate a strategy as desired, but when deactivating a strategy, we are also deactivating the price recommendations it has generated. We do not allow deleting strategies so that calculated recommendations are never lost - inactivating is a more elegant solution, which allows you to hide a strategy and its recommendations while keeping its data available for analytical purposes or for future use.

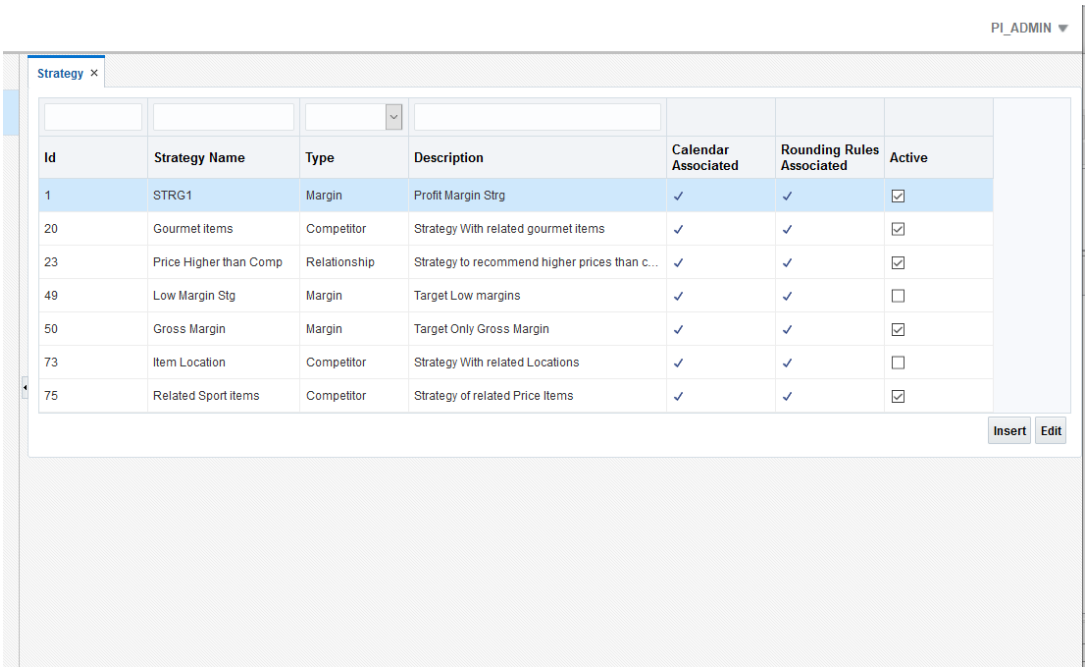

Figure 64 - Maintain Strategy Page

When selecting the option to insert the user is presented with a popup with a train, like the one on the image below. The first stop of the train presents a form that must be filled with the following data:

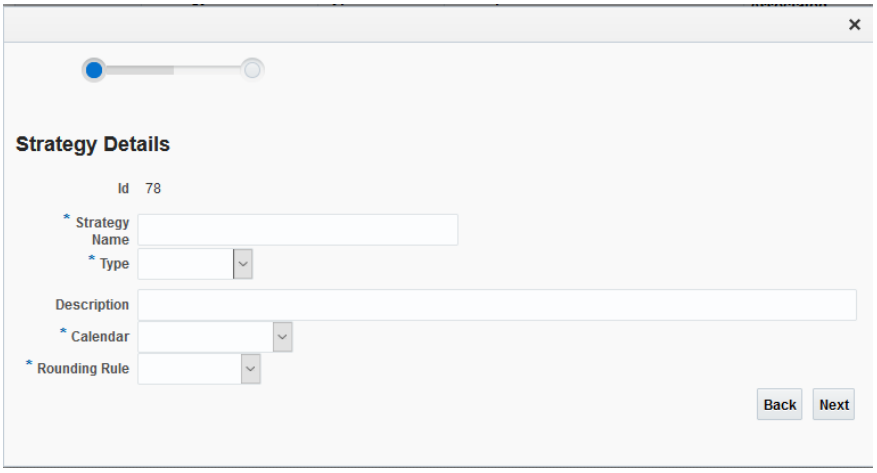

Figure 65 - Add New Strategy Popup

- **Strategy Name** - which identifies the strategy.
- **Type** - the type of strategy must be selected in a dropdown with the following options available: margin strategy, competitor strategy, relationship strategy.
- **Description** - a description that explains the objective of the strategy.
- **Calendar** - the calendar to which the strategy is associated. The calendar can be selected from a dropdown that displays the id and name of all existing calendars.
- **Rounding Rule** - rounding rule to which the strategy is associated. The rounding rule can be selected from a dropdown that shows the id and name of all existing rounding rules.

After filling in the data, select the next button that takes us to the next train stop. A shuttle is shown on this page, as shown in the figure below, which allows selecting multiple values from a list allowing the user to move items between the two lists. The list on the left shows all the existing targets for that type of strategy. You can select items from that list and move them to the list on the right that represents the list of selected items. Finally, the save option is selected and the strategy is created with the data provided.

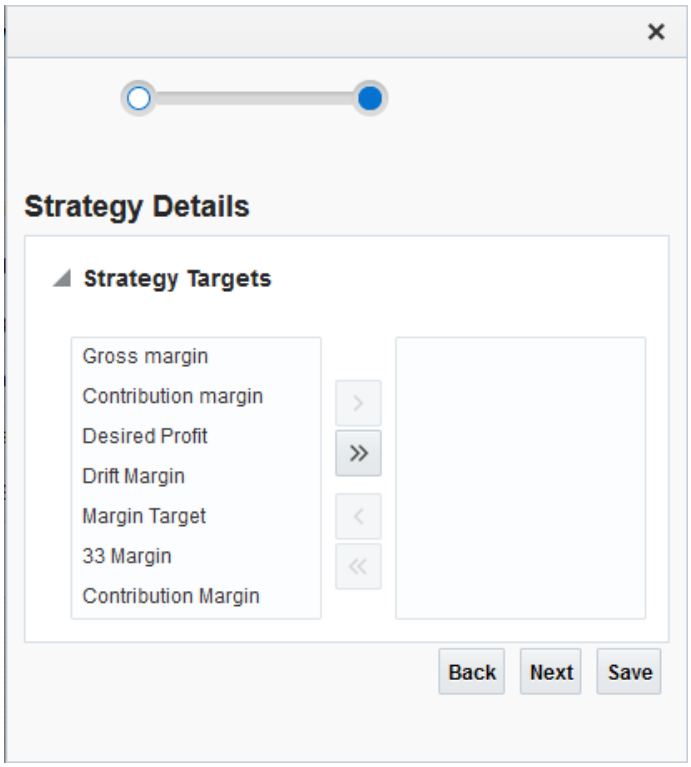

Figure 66 - Add New Strategy Targets Popup

In the initial table, when selecting a strategy and clicking on the edit option, the same popup as the insert option is displayed, but in this case the data is already filled with the original data.

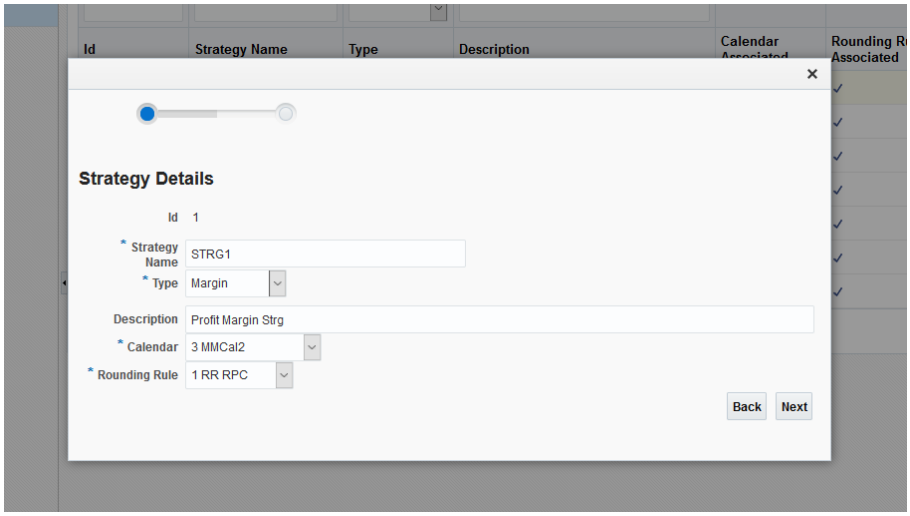

Figure 67 - Edit Strategy Popup

It is possible to edit this data and select or deselect targets on the shuttle when moving them between lists.

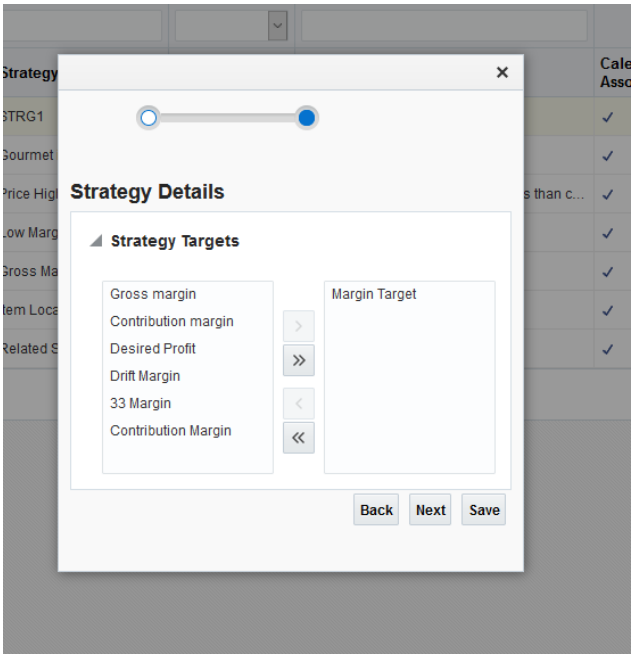

Figure 68 - Edit Strategy Targets Popup

4.3.6 Price Recommendations

4.3.6.1 Perform What-If Analysis

This feature has been left out entirely for reasons of time. But the idea would be to work relatively similarly to existing strategies. The idea of the What-If analysis is basically to allow the

simulation of conditions and analysis of how they would be reflected in the strategy's price recommendations, but without affecting the recommendations calculated by the real targets.

4.3.6.2 Submit, Review and Approve Price Recommendations.

The page shown in the image below meets the functional requirements UC11 AND UC12 and aims to review generated recommendations, submit a price recommendation and the final approval of a submission.

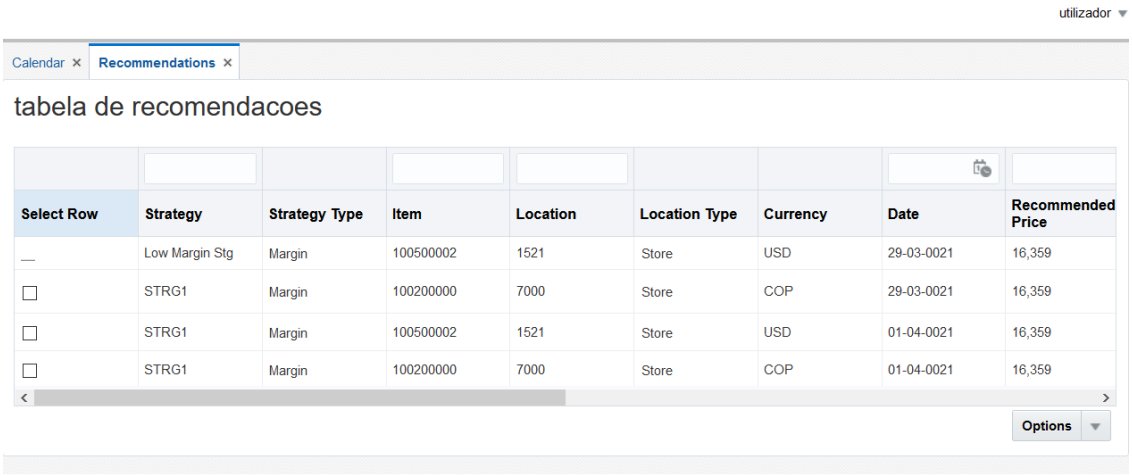

Figure 69 - Price Recommendation Page

On the recommendation page a table is presented with all the price recommendations generated based on the strategies previously created and the defined targets. The table also has the ability to apply filters allowing a better search which is particularly relevant in this table, which will have many records. The table provides a series of information related to each of the price recommendations that are described below:

- **Select Row** - Allows you to select one or more recommendations and perform an action such as submitting, approving or rejecting a recommendation. In this case, only the selected rows will be affected;
- **Strategy** - Presents the strategy to which the recommendation belongs;
- **Strategy Type** - Presents the it's a recommendation based on a margin, competitors or relationship strategy;
- **Item** - Displays the item for which a price is being recommended;
- **Location** - Displays the item location for which a price is being recommended;
- **Location Type** - Display if the location its of type Store or Warehouse;
- **Currency** - Displays the currency of the recommendation;
- **Date** - Shows the date the recommendation was generated;
- **Recommended Price** - Displays the price calculated by the strategy and which is being recommended;
- **Current Margin** - Presents the current margin;
- **Recommendation Margin** - Presents the margin for the price being recommended;

- **Sales** – Shows the number of sales for that item location in a given configurable time period;
- **Final price Margin** - Presents the final Price Margin based on the value inserted in the Final Price field. This field is updated each time the final price field is changed;
- **Final Price** - Final price to be submitted. This field is editable when the recommendation is new and has not yet been submitted;
- **Final Date** - The date the price recommendation should be implemented. When the recommendation state is 'new' the field is editable and the user can define the final data;
- **State** - Shows the state of the recommendation. New – when the recommendation is new and didn't suffer any event. Awaiting Approval – when a recommendation was submitted and is awaiting approval or rejection. Approved – the recommendation was already approved. Rejected – The recommendation submitted was rejected.

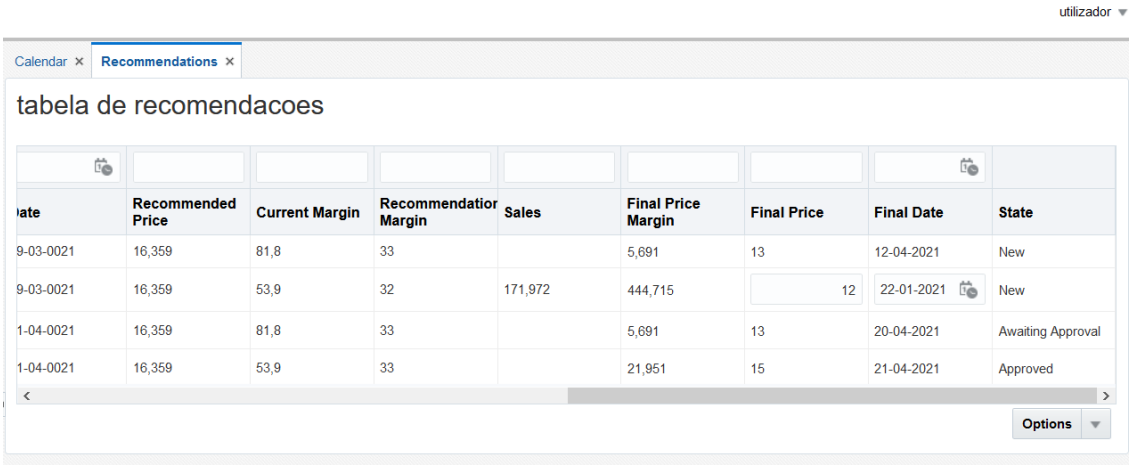

Figure 70 - Price Recommendation Page

In case a strategy is deactivated, the recommendation is also deactivated, which is the case of the first row, in which the Select Row column is unavailable.

To perform an action the user must select the lines he wants to be affected and choose one of the available options, as can be seen in the figure below. This options that are presented depend on the roles/permissions of each user.

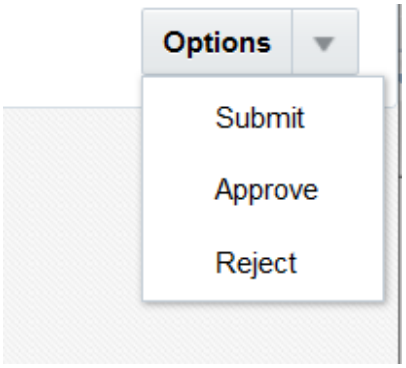

Figure 71 - Different Menu Options

After the action has been completed, a popup is displayed that informs the user that the task has been successfully completed and the table is updated.

4.4 Tests

Software testing is an essential phase in developing a solution so that the application is able to perform its proper functions and to ensure that it is delivered to the end customer correctly.

There are different types of software tests, in this case we focused on usability, integration and unit tests. So, to guarantee a better quality in terms of user experience, usability tests were carried out, and to guarantee the good performance of the solution, in terms of functional tests, unit tests and integration tests were carried out.

4.4.1 Unit tests

Unit tests are the isolated testing of individual modules of a solution in order to verify that the code is working correctly regardless of its interaction with other modules. This means testing the small functional units of the project individually by running all possible paths (Brar and Kaur, 2015). Unit tests test a part of the system and if the resulting behavior is in agreement with the expected result the test passes with a valid result; if not, it fails indicating that there is a problem in the code in the module that was tested. This type of test is relatively simple and quick.

In the developed solution, tests were developed in the application module to test the Java code and in the Pricing Intelligent Engine module to test the PL/SQL functions.

For the development of test cases in the application module, the JUnit framework was used. These unit tests focused on testing functionality such as calculations, the user input and database interaction functions such as finding an item or entering a target.

In the figure below it is possible to see an example of a unit test that checks if the competitor target with ID 4 is in the database or not.

```
45      @Test
46      public void findCompetitorsTarget(){
47          ViewObject view = fixture1.getApplicationModule().findViewObject("PiCompetitorsTargetView1");
48          int id = 4;
49          Row row[] = view.findByKey(new Key(new Object[] {id}), 1);
50          Integer count = row.length;
51          //assertTrue fails when second parameter evaluates to "false"
52          assertTrue("Competitors Target Not Found", count.compareTo(0) > 0);
53      }
```

Figure 72 - Unit Test

When running the test set it is possible to observe all successful tests:

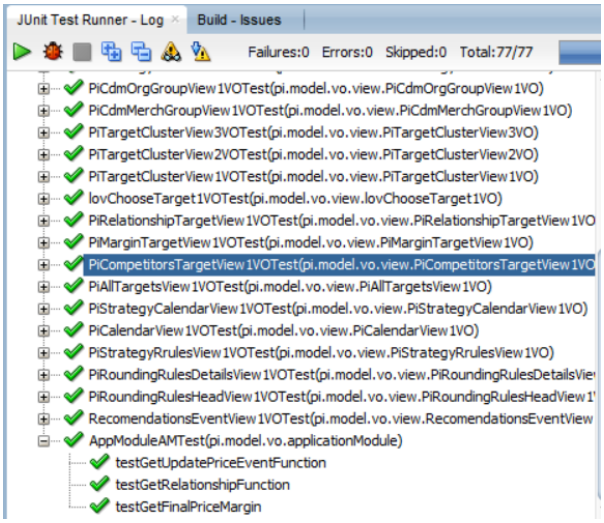

Figure 73 - Result of the Unit Tests

For the development of test cases in the Engine module, we used the SQL Developer unit test feature, which provides a framework for testing PL/SQL objects, such as functions and procedures. These unit tests focused on testing different PL/SQL functions used in calculating markups, pricing recommendations, and strategies.

In the image below is a simple example of a unit test made to test the margin strategy function.

```
1   declare
2     -- Boolean parameters are translated from/to integers:
3     -- 0/1/null <--> false/true/null
4     result boolean;
5   begin
6     -- Call the function
7     result := PI_MARGIN_STRATEGY(I_STRATEGY_ID => :I_STRATEGY_ID);
8     -- Convert false/true/null to 0/1/null
9     :result := sys.diutil.bool_to_int(result);
10  end;
```

Figure 74 - PL/SQL Unit Test

4.4.2 Integration Tests

Integration testing aims to verify that the different modules are working correctly when tested together, testing components formed by the union of two or more units and checking their interaction with each other (Brar and Kaur, 2015). Basically, it consists of testing modules that are integrated and analyzing the behavior of the solution, the interaction of its different modules and verifying whether the data is processed correctly. These tests include modules such as interfaces, servers, database, etc.

The idea is to understand if there is any failure/instability and expose possible defects in the interaction between these modules.

To carry out the integration tests, the Selenium tool was used, which allows the execution of automated tests. In this case, JUnit and Selenium WebDriver were used, which send native events to the controlling browser.

In the figure below, you can see a Junit test method newTargetStgAssociation that aims to test the creation of a target and associate it with a new strategy. The test starts by navigating to the solution page and through various events, such as clicks, keyboard and other events, it accesses the create targets page and creates a new target with the data provided. Then accesses the strategy page and creates a new strategy to which it associates the target that was previously created.

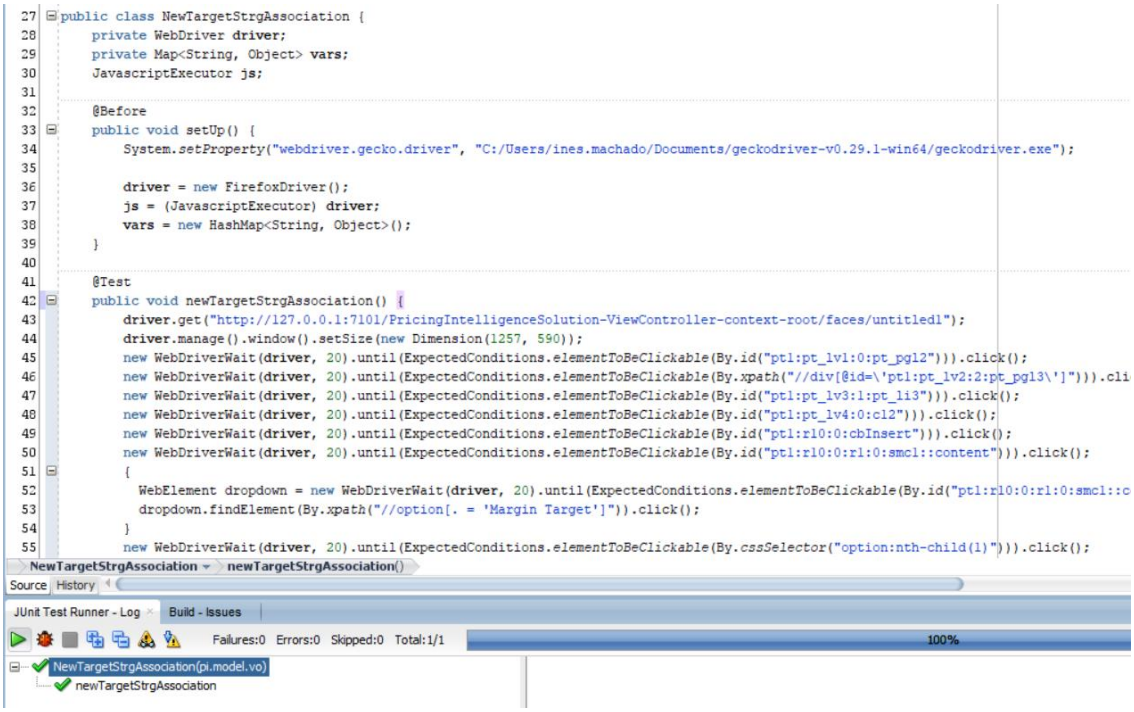

Figure 75 - Integration Test

When running the test case the main browser window will show the real web pages and the actions performed by the test, as you can see in the figure below.

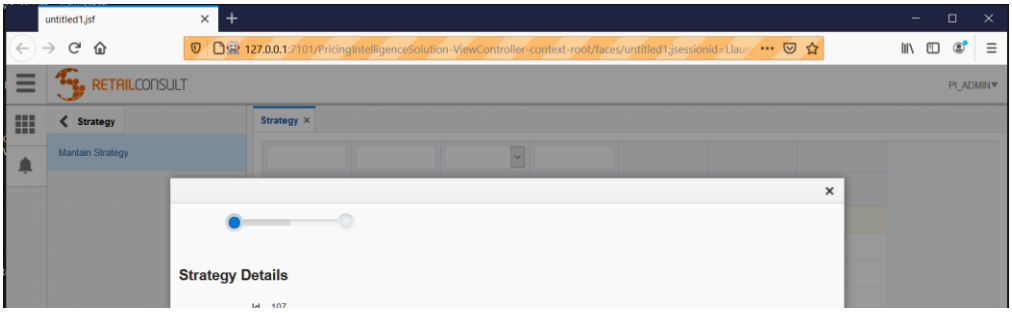

Figure 76 - Selenium Running

4.4.3 Usability Tests

The usability test aims to verify the ease of handling and understanding the solution developed is in terms of user experience. This type of testing is usually performed with real users who are made to use the solution for the first time in order to expose usability defects. The goal is to understand how intuitive, attractive and understandable the solution is for the end user. Furthermore, they can also be used to evaluate the experience in terms of graphical interface such as fonts, images, colors, and other characteristics that can affect the user experience. These tests allow us to design products with the user's role in mind as the knowledge we gain about our users' experience supports all aspects of design and development (Barnum, 2020). It is important to mention that this is a complex and highly configurable solution, and as such the approach defined for its implementation will always go through a formation/training phase for its users.

To test the app, we asked two different groups of people to try it out for the first time. The aim of this tests where to verify if the NFRs are being met and find possible problems. The first group was formed by some members of Retail Consult already used to working with retail applications such as RMS and RPM, the second group was formed by external people who had no knowledge of the area or project. The feedback we received was insightful, and as development progressed, some of these issues were fixed. One of the issues identified and resolved was the option to download the Excel file. Upon downloading, the file was then saved in the downloads folder and simply alerted the user that the file was available in the downloads, which caused confusion on the part of many users. Most users reported either that they had not realized that the download had already been done or that it made no sense not to have the option to select the desired folder. To solve the problem, the JFileChooser resource was used, which provides a GUI to navigate the file system and thus choose the directory where the user wants to save the file.

4.4.4 Metrics and Coverage Degree

To guarantee the quality of the code of the solution and its coverage, the SonarQube tool was used. SonarQube allows all developers to write cleaner and safer code, as it analyses reviews the code and indicates possible bugs, vulnerabilities code smells and possible changes so that

the code can be improved (Code Quality and Code Security | SonarQube, 2021). It also allows us to get a sense of the current status of code coverage, the code complexity and also get tips and guidance for better quality and security. This code quality examination tool is open source and is available for several programming languages and can be integrated in different DevOps or continuous integration tools. In Retail Consult SonarQube is integrated with GitLab and will be triggered whenever there is a GitLab commit or merge. The static code analysis will then be available on a dashboard.

The SonarQube instance used to test the developments is centrally managed and, as such, it is not possible to change the user interface language. For that, all figures presented related to the SonarQube usage are in Portuguese.

To implement and activate SonarQube, in the setup phase, it is necessary to configure the .gitlabci.yml file, presented in the image below, that allows for continuous integration.

```
stages:
  - Code Review

Static Code Analysis without tests:
  stage: Code Review
  image: ${RC_SONAR_SCANNER_IMAGE}
  only:
    changes:
      - src/**/*
  except:
    variables:
      - $STATIC_ANALYSIS_DISABLED
  allow_failure: true
  before_script:
    - export CI_SONAR_projectKey=$(eval echo $(eval echo ${CI_PROJECT_NAMESPACE,,} | tr "/
" .):${CI_PROJECT_NAME,,}:${CI_COMMIT_REF_NAME^^} | tr " " _)
    - export CI_SONAR_projectVersion=$(date +%Y%m%d-%H:%M.%S)
  script:
    - sonar-scanner -Dsonar.projectKey=$CI_SONAR_projectKey
                    -Dsonar.projectBaseDir=$CI_PROJECT_DIR
                    -Dsonar.projectVersion=$CI_SONAR_projectVersion
                    -Dsonar.gitlab.project_id=$CI_PROJECT_PATH
                    -Dsonar.gitlab.commit_sha=$CI_COMMIT_SHA
                    -Dsonar.gitlab.ref_name=$CI_COMMIT_REF_NAME
                    -Dsonar.java.binaries='./src/ViewController/src/pi/view'

```

Figure 77 - .gitlabci.yml File

SonarQube offers a number of code quality metrics, including:

- Complexity;
- Code Duplication;
- Issues;
- Quality;
- Test Coverage;
- Size;

- Maintainability Index;
- Reliability Index;
- Security Index;

Each programming language has its analysis rules, in the case of java code the rules and metrics used for the analysis were as follows:

Sonar way BUILT-IN

Condições

Conditions on New Code

Métrica	Operador	Value
Cobertura	é menor que	80,0%
Linhas duplicadas (%)	é maior que	3,0%
Índice de manutenibilidade	é pior que	A
Índice de confiabilidade	é pior que	A
Security Hotspots Reviewed	é menor que	100%
Índice de segurança	é pior que	A

Figure 78 - SonarQube Rules and Metrics

After the correct configuration, when a commit is performed the pipeline will be executed and the code will be analyzed according to the rules for the programming language, providing then a dashboard with all the information.

In the case of the project presented, all possible bugs and vulnerabilities detected by SonarQube were resolved. From the tests performed, it was possible to obtain 81.3 percent coverage. In the figure below it is presented all the information provided by SonarQube.

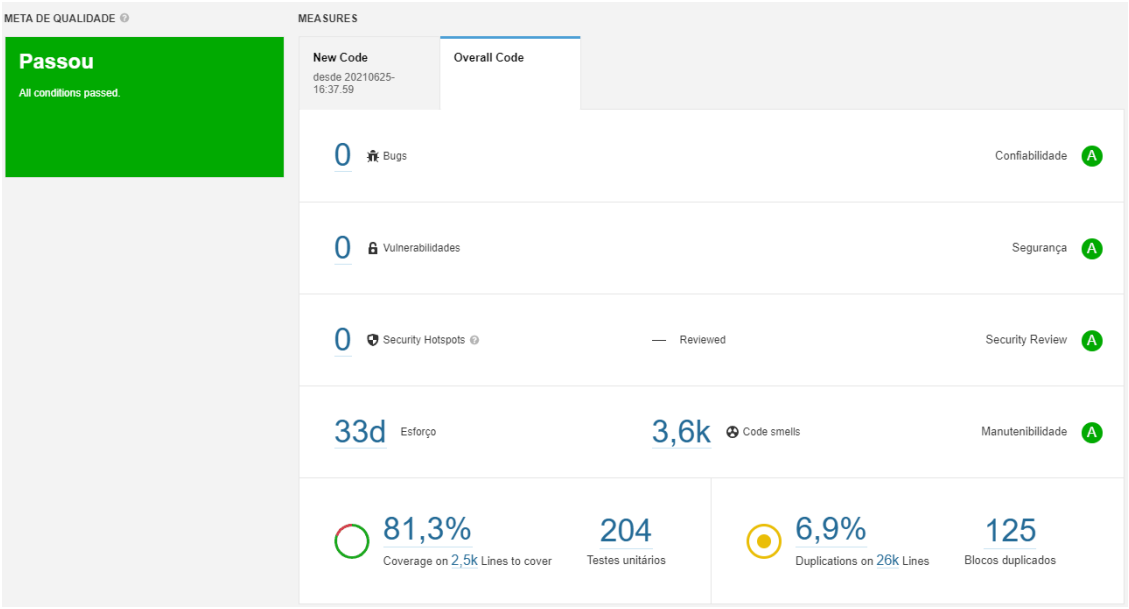

Figure 79 - SonarQube Dashboard

Testing and using the SonarQube tool allowed us to identify different problems and failures in the solution and to correct them, thus making the solution more efficient and with better quality.

5 Solution Evaluation

This chapter describes the entire process for determining the quality of the proposed solution and the method that will be used to evaluate the same solution. The main objective is to evaluate the solution designed and implemented by adopting evaluation processes and to properly justify the evaluation made for a given problem using a rigorous and well-founded methodology.

For this, the hypotheses to be tested and the indicators and sources are identified and then a brief description of the evaluation methodology that will be used to determine the quality of the solution is made.

In the case of the evaluation methodology, it was decided to use the Quantitative Evaluation Framework (QEF). This model is described and explained in the following sections, as well as the dimensions, factors, requirements, and evaluation metrics for the right usage of the framework.

Finally, a summary of the final evaluation of the solution is made, with an analysis of the results obtained and a comparison between the initial objectives and the expected results.

5.1 Quantitative Evaluation Framework

The Quantitative Evaluation Framework (QEF) consists of a quantitative evaluation model whose main objective is to evaluate and guarantee the quality of solutions in a three-dimensional space where each of these dimensions aggregates factors (Escudeiro and Bidarra, 2008).

This method assumes that we have a holistic view of quality, a view in which quality is relevant from various points of view in an integrated manner and that therefore quality is the set of these various views that may have different perspectives, complementary to the system.

The tool consists of assessing the quality of the system, in the Cartesian referential system, where the QEF quality scenario is defined. The quality scenario will be the ideal system, and throughout the project the real system will be evaluated to verify the extent to which the various elements of the ideal system are present in the system, at any given time.

The measure of the distance between the two vectors, the distance between the system considered ideal and the real system (at a given moment), is a measure of quality. The greater this distance, the further away the real system is from the ideal system and therefore the quality will be lower. The quality is greater the closer the two vectors meet.

To create the quality scenario, it is necessary to define the different dimensions and factors, where each factor has its respective associated weight. The sum of the weights of the factors belonging to a dimension must always be equal to 1. In addition, an evaluation metric must be implemented for a defined requirement.

5.1.1 Dimensions and Factors

The quality scenario or quality benchmark where we will measure the length of each of the characteristics is organized in a hierarchy with three levels. The dimensions, the factors, which are elements that make up the dimensions, and the requirements that are elements that contribute to fulfill a certain factor. That is, to determine that a certain factor is being met, it is necessary to analyze all the requirements that are defined in that factor and that will have an impact. And to be able to understand the quality of the system from the point of view of a dimension, it is necessary to know the extent to which the requirements that contribute to each factor are verified or not.

Below are defined the dimensions, factors, and requirements of the quality scenario:

5.1.1.1 Functionality Dimension

The functionality dimension aims to evaluate the operational aspects in relation to the existing functionalities in order to verify if they are implemented as defined.

This dimension adds three factors: Functional (referring use cases), User interaction and Content Quality.

5.1.1.2 Efficiency Dimension

The efficiency dimension aims to evaluate the performance of the solution based on different factors in a competent manner and with minimum effort, both by the system and the user.

This dimension adds four factors: Navigation, security, quality of service and technical aspects.

5.1.1.3 Adaptability Dimension

The adaptability dimension aims to understand the ease of adaptation of the solution in the face of a need to change functionality, add new requirements or functionality and perform maintenance without prejudice.

This dimension is divided into five factors: maintenance, management, technical aspects, data management and Usability.

The full QEF diagram is available in the appendix A section.

5.2 Evaluation of the Solution

In order to validate the hypothesis to be tested and correctly fill in the QEF, the first tests were scheduled where some employees were asked to use some of the features and execute a preset

sequence of orders in the solution. After having the first contact and using the solution, they were asked to answer some questions regarding the usability of the solution and possible suggestions. All suggestions made by employees during the testing phase were taken into account and are being reflected in the next development. After completing the tests, the QEF was filled out, taking into account all the answers obtained by the users.

Overall, after performing the tests and due analysis and final filling of the QEF table, the result obtained was of 92% quality rate. The result is available in the updated QEF diagram that is available in the appendix A section. Considering that the project is still at an early stage and that not all functionalities are available, this level of quality rate is very positive.

It is intended that when the missing requirements are completed there is a second phase of testing in order to evaluate the changes made according to the recommendations previously made by the employees and to test the new features available. Since the second phase of testing has not yet been carried out, this will not be the final evaluation of the solution. The second test phase is planned to occur three months, after closing the document, and as such will not be presented. After this second test group, the QEF will be revised again and the hypothesis test will be applied, to compare the means of the two test groups.

6 Conclusion

This chapter presents the conclusion regarding the project carried out and the results achieved. Starting with the achieved objectives where the results are evaluated in relation to the objectives that were initially defined. Then, the limitations that appeared during development were identified. Finally, the future work and a final assessment of the overall project are presented.

6.1 Achieved Goals

The main objective of the project was to close the gap between what retailers need to make better profits and what existing technologies offer. Namely, the lack of solutions that allow obtaining price recommendations based on pre-defined strategies by the user. The developed solution allows retailers to define different strategies for a set of items in a certain location with the possibility of obtaining price recommendations based on defined targets.

The realization of this project enabled the acquisition of new knowledge in the retail area, essential for the development of the solution. An essential study was also made of the different technologies used, namely, both Oracle Retail solutions used, RPM and RMS, and the main development framework, ADF.

In the following paragraphs, the objectives stipulated at the beginning of the project are presented with a brief analysis and description of how they were achieved

1. **Research based on pricing challenges and understanding of the retail domain and Oracle retail platform:**
   In this initial phase, an intensive study of the retail area and of Oracle Retail's RPM and RMS technologies was carried out. The study was based on reading different articles, books and manuals from Oracle Retail, in addition, the author of the thesis made different trainings in the scope of retail and Oracle Retail provided by Retail Consult. The Retail Consult advisor was also available several times to explain certain areas of

Oracle Retail solution and underlying business processes. After the study of the domain, the study and analysis of different technologies that were used in the development were also carried out. Given that Retail Consult's solutions are mostly based on Oracle ADF, it was clear right from the start that this would be one of the technologies to adopt. Practical trainings were also carried out, namely, PL/SQL Basics, ADF basics and DevOps training.

2. **Study and evaluate possible pricing approaches and interact with stakeholders to determine what are the best approaches to solve the problem addressed:**
   This phase started with a study and evaluation of potential pricing approaches (margin, key value item, price relationships, item and cross-item price elasticity, competitor price etc.) and reading articles related to these approaches to pricing. In addition, several meetings were held with the Project Manager, especially for the discussion of this topic, where it was possible to conclude which would be the approaches to be used in this first phase and how they would be implemented.

3. **Define a minimum set of specifications for a minimum viable product:**
   This phase consisted of the study of retailers' needs, the analysis of customer requirements and a study and analysis of the competitor's products where it was possible to identify existing gaps and failures. Furthermore, this objective was achieved and the required specifications were created through several meetings with the Project Manager, this phase being one of the most important for development.

4. **Define a set of requirements for the solution in terms of priority that are enforceable during the internship period and the solution roadmap:**
   This phase was completed in conjunction with the Project Manager, who played a key role in pursuing this objective. To define the requirements, several meetings were held, where we started by defining the authors of the system, then the domain model was made and finally the different requirements (functional and non-functional) were defined based on the study carried out and based on the set of previously defined specifications.
   For all requirements, a priority was defined, that is, the weight of importance of each requirement. It was at this stage that decisions were taken regarding which requirements would be mandatory in terms of implementation, and which would be considered "nice to have". It was necessary to scale the amount of work to the fact that it was carried out exclusively by the author without the participation of any other developers. This was one of the most important stages and which served as the basis for the development phase.

5. **Develop the solution prototype that must be configurable and scalable:**
   Based on the requirements raised in the first phase and due priority, a development plan was made that is available in the Appendices. The project started with the design and construction of the database data model. Then, dummy data were inserted to carry

out tests where the correct functioning of the database was verified. After that, the phase of design and build of the APP framework was started, establishing baseline for UI/UX components. And then the ERP Abstraction Layer for reading base data required. After this initial phase, the requirements were developed over the course of the thesis according to the defined plan. The development of the solution interface was done through the Oracle ADF framework. During development, several tests were performed and the SonarQube tool was also used to obtain the decree of coverage.

The use of the DSR method, Design Science Research (Kuechler and Vaishnavi, 2008), during the thesis made a great contribution to the document and elaboration of the solution, as it allowed for greater control over the design and a better resolution of problems in certain contexts. DSR made it easier to achieve the goals by making the entire process more rigorous and relevant.

The goals of this dissertation where to analyze, design and build the technical structure and define resources for the implementation of a pricing solution that can be fully integrated with the Oracle Retail solution stack and that is configurable and scalable in a way that will potentially enable it to interact with any other required systems. The development of a prototype, as a proof of concept of the operation of the solution, allows to prove that this objective was fulfilled. Therefore, it can be considered that all the objectives of this project were fulfilled.

6.2 Limitations

Throughout the project, several difficulties were encountered, but despite everything it is important to mention that these did not negatively influence the achievement of the objectives but had some influence on the development of the project and a potential delay.

During the initial phase of the project, the definition of the feature set for a minimum viable product was done with the idea that the project would be executed by a team of about 5 people. All through the research phase and finally reaching the development stage, Retail Consult had an abnormal workload, which made it impossible for other potential team members to enter. Thus, the entire development and testing process was carried out by the author of the thesis, accompanied by the company supervisor.

Given this new knowledge, some requirements were put aside and will be developed outside the thesis time as part of the product roadmap mentioned in the following section.

6.3 Future Work

After testing and the results obtained from the developed solution, it was possible to conclude that although there are already many features there are some that still need to be developed. Namely the features that were put aside due to time constraints, some details that could be improved and features that throughout development were thought of as a good addition for

better functioning. Besides, since this solution is a minimum viable product, it would be interesting to put a possible Retail Consult client test and experiment the solution to identify other possible limitations. And, as proof of the good work done, this solution is planned to be presented to a large client in Latin America for potential implementation.

That said, the following points are a small sample of the product roadmap defined based on the developed solution and potential enhancements that have been gathered:

- Development of the graphical interface component for changing settings;
- GUI component development to define relationships between merchandising hierarchy branches according to the UC4 requirement;
- Development of graphical interface components to define relationships between organization's hierarchy branches according to the UC5 requirement;
- Develop the What-If target analysis requirement according to the UC10 requirement;
- Present the possibility to switch actions processing to the background according to the UC5 requirement NFR8;
- Enhance system authentication and access permissions depending on the defined roles according to the NFR16;
- Possibility to add facilitators for creating "repeating" promotions (e.g., that take place every X day), possibly limited also by a start and end time to support the “happy hour” pricing concept (e.g. from 9am to 11am);
- Possibility to add facilitators for loading price events, useful for versions of RPM or other price execution systems that don't have the capabilities to do mass upload;
- "Hierarchy" of strategies, to allow for price recommendations to be aggregated based on multiple concurrent strategies;

References

Al-Hussari, H. (2006) The influence of contextual factors on cost system design and pricing decisions: a study of UK companies in the food processing and other industries. PhD Thesis. University of Huddersfield. Available at: eprints.hud.ac.uk/5969/1/430294.pdf (Accessed: 24 February 2021).

Andrew, M. (2016) 'Oracle Retail Merchandising Implementation Guides'. Available at: https://docs.oracle.com/cd/E12440_01/rpm/pdf/150/html/merch_impg/RMS.htm (Accessed: 28 December 2020).

Ayers, J. B. and Odegaard, M. A. (2017) Retail Supply Chain Management. CRC Press. Available at: https://books.google.pt/books?id=TXJQDwAAQBAJ (Accessed: 21 December 2020).

Barnum, C. M. (2020) Usability Testing Essentials: Ready, Set ...Test! Morgan Kaufmann. Available at: https://books.google.pt/books?id=L6_SDwAAQBAJ (Accessed: 4 June 2021).

BenMark, G. et al. (2017) 'How retailers can drive profitable growth through dynamic pricing', McKinsey & Company. Recuperado de https://www. mckinsey. com/industries/retail/our-insights/howretailers-can-drive-profitable-growth-throughdynamic-pricing. Available at: https://www.mckinsey.com.br/~/media/McKinsey/Industries/Retail/Our%20Insights/How%20retailers%20can%20drive%20profitable%20growth%20through%20dynamic%20pricing/How-retailers-can-drive-profitable-growth-through-dynamic-pricing.pdf (Accessed: 3 January 2021).

Brar, H. K. and Kaur, P. J. (2015) 'Differentiating Integration Testing and unit testing', in 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom). 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 796–798. Available at: https://ieeexplore.ieee.org/document/7100358 (Accessed: 4 June 2021).

Chen, C.-H. et al. (2015) 'Exploring Electronic Word-of-Mouth (eWOM) in The Consumer Purchase Decision-Making Process: The Case of Online Holidays – Evidence from United Kingdom (UK) Consumers', Journal of Travel & Tourism Marketing, 32(8). Available at: https://doi.org/10.1080/10548408.2014.956165 (Accessed: 23 February 2021).

Code Quality and Code Security | SonarQube (2021). Available at: https://www.sonarqube.org/ (Accessed: 19 May 2021).

Cohen, M. et al. (2020) 'Computerized promotion price scheduling utilizing multiple product demand model'. Available at: https://patents.google.com/patent/US10740782B2/en (Accessed: 11 November 2020).

Configure, Price, Quote: Performance, simplified (2018) SAP. Available at: https://www.sap.com/documents/2018/11/26e199d8-2b7d-0010-87a3-c30de2ffd8ff.html (Accessed: 16 January 2021).

Cost-optimized and High-Performance Database (2021). Available at: https://www.oracle.com/database/ (Accessed: 27 January 2021).

Cox, R. and Brittain, P. (2004) Retailing: An Introduction. Pearson Education. Available at: https://books.google.pt/books?id=JMgasWRAZNEC (Accessed: 21 December 2020).

DB-Engines Ranking (2021) DB-Engines. Available at: https://db-engines.com/en/ranking (Accessed: 16 February 2021).

Dekimpe, M. G. (2020) 'Retailing and retailing research in the age of big data analytics', International Journal of Research in Marketing, 37(1). Available at: http://www.sciencedirect.com/science/article/pii/S016781161930062X (Accessed: 7 December 2020).

Eeles, P. (2005a) 'Capturing architectural requirements', IBM Rational developer works. Available at: https://www.researchgate.net/profile/Peter_Eeles2/publication/329760910_Capturing_Architectural_Requirements/links/5c197761458515a4c7e8bae1/Capturing-Architectural-Requirements.pdf (Accessed: 29 January 2021).

Eeles, P. (2005b) 'Non-functional requirements', IBM Software Group. Available at: https://pdfs.semanticscholar.org/f3bb/91080c4573f6f78f30bc5b48bda3ef252bf2.pdf (Accessed: 29 January 2021).

Egan, R. G. (2020) 'Developing Applications with Oracle JET'. Available at: https://bit.ly/2TXDAFa (Accessed: 20 December 2020).

Eralp, A. (2013) Introduction to Oracle ADF. Available at: https://blogs.oracle.com/ardaeralp/introduction-to-oracle-adf (Accessed: 30 January 2021).

Ernst & Young (2020) 'EY Future Consumer Index', EY Building a Better Working World. Available at: https://www.google.com/search?client=firefox-b-d&q=EY+Future+Consumer+Index+carried+out+in+Portugal (Accessed: 16 February 2021).

Escudeiro, P. and Bidarra, J. (2008) 'Quantitative evaluation framework (QEF)', Conselho Editorial. Available at: https://www.researchgate.net/profile/Jose_Bidarra2/publication/257579051_Quantitative_Evaluation_Framework_QEF/links/541afafc0cf203f155ae6da5/Quantitative-Evaluation-Framework-QEF.pdf#page=17 (Accessed: 31 January 2021).

Gala, D. and Patil, R. D. (2013) 'Consumer attitude towards private labels in comparison to national brands', International Journal of Business and Management Invention, 2(5). Available at: https://www.academia.edu/download/40681217/C251218.pdf (Accessed: 23 February 2021).

Gbadebo-Smith, O. (2018) 'Price Elasticity 2.0: From Theory to The Real World', Toptal Finance Blog. Available at: https://www.toptal.com/finance/pricing-consultants/price-elasticity-of-demand (Accessed: 16 February 2021).

Golfarelli, M., Rizzi, S. and Proli, A. (2006) 'Designing what-if analysis: Towards a methodology', in. DOLAP: Proceedings of the ACM International Workshop on Data Warehousing and OLAP. Available at: https://www.researchgate.net/publication/220933881_Designing_what-if_analysis_Towards_a_methodology (Accessed: 16 February 2021).

Gordon, R. (2020) The Oracle JET Architecture, Oracle Help Center. February2020. Available at: https://docs.oracle.com/en/middleware/developer-tools/jet/8.1/develop/oracle-jet-architecture.html#GUID-293CB342-196F-4FC3-AE69-D1226A025FBB (Accessed: 20 December 2020).

Hogan, J. and Nagle, T. (2005) 'What is strategic pricing?', Strategic Pricing Group Insight. Available at: www.michelbeauregard.com/Monitor_What_Is_Strategic_Pricing.pdf (Accessed: 4 January 2021).

Kilroy, T., MacKenzie, I. and Manacek, A. (2015) 'Pricing in retail: Setting strategy'. Available at: https://www.mckinsey.com/industries/retail/our-insights/pricing-in-retail-setting-strategy (Accessed: 5 January 2021).

Kirasirova, E. and Biryukov, V. (2019) 'Global Powers of Retailing 2019', Deloitte. Available at: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiq1p_i363xAhVOJhoKHcelCNkQFjAAegQIBRAD&url=https%3A%2F%2Fwww2.deloitte.com%2Fcontent%2Fdam%2FDeloitte%2Fglobal%2FDocuments%2FConsumer-Business%2Fcons-global-powers-retailing-2019.pdf&usg=AOvVaw3nATVZnFqG4lT5GdHbTN8K (Accessed: 22 December 2020).

Kouraklis, J. (2016) 'MVVM as Design Pattern', in. doi: 10.1007/978-1-4842-2214-0_1.

Kuechler, B. and Vaishnavi, V. (2008) 'On theory development in design science research: anatomy of a research project', European Journal of Information Systems, 17(5). Available at: https://www.tandfonline.com/doi/pdf/10.1057/ejis.2008.40 (Accessed: 19 February 2021).

Kumar, N. and Steenkamp, J. E. m (2007) Private Label Strategy: How to Meet the Store Brand Challenge. Harvard Business Review Press. Available at: https://books.google.pt/books?id=QnT_BgAAQBAJ (Accessed: 16 February 2021).

MacKenzie, I., Meyer, C. and Noble, S. (2013) 'How retailers can keep up with consumers'. Available at: https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers (Accessed: 22 December 2020).

Mckibbon, S. (2009) 'Oracle Price Management Operations Guide', An Oracle White Paper. Available at: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj-hOD44K3xAhWJxYUKHYMHAtcQFjABegQIBBAD&url=https%3A%2F%2Fdocs.oracle.com%2Fcd%2FB31319_01%2Frpm%2Fpdf%2FIndia%2F12010%2Frpm-12010in-og.pdf&usg=AOvVaw1y9WtDp1lLXaJSJsx-FNiJ (Accessed: 17 December 2020).

Mishra, N. (2016) 'Oracle Retail Merchandising System (RMS) - Deep Dive Tutorial', Oracle Retail. Available at: https://oretail.blogspot.com/2016/07/oracle-retail-merchandising-system-rms.html (Accessed: 29 December 2020).

Mishra, N. (2017) 'Oracle Retail Price Management (RPM) – Deep Dive Tutorial', Oracle Retail. Available at: https://oretail.blogspot.com/2017/10/oracle-retail-price-management-rpm-deep.html (Accessed: 30 December 2020).

Mulugeta, D. et al. (2013) Price-and Cross-Price Elasticity Estimation using SAS. Available at: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwif_Kqk4a3xAhVPhRoKHc2rAJcQFjAAegQIBBAD&url=https%3A%2F%2Fsupport.sas.com%2Fresources%2Fpapers%2Fproceedings13%2F425-2013.pdf&usg=AOvVaw1vW88mqGHn3sMkR9R_5Rvu (Accessed: 29 January 2021).

Mulvenna, A. (2020) '9 Must-see Types of Profitable Retail Pricing Strategies and Tactics', Competera - Pricing software for Online Retail. Available at: https://competera.net/resources/articles/what-are-the-major-pricing-strategies (Accessed: 16 February 2021).

O'Brien, S. and Shmeltzer, S. (2011) 'Oracle Application Development Framework Overview', An Oracle White Paper. Available at: https://docs.oracle.com/cd/E12440_01/rpm/pdf/160/rpm-160-og.pdf (Accessed: 17 December 2020).

Oracle (2019) 'Retail Merchandising Foundation Cloud Service', Oracle. Available at: https://docs.oracle.com/cd/F11448_01/doc.160030/f11690/toc.htm (Accessed: 29 December 2020).

Oracle and its affiliates (2020) 'Oracle Retail Price Management Datasheet'. Available at: https://www.oracle.com/a/ocom/docs/industries/retail/retail-price-management-ds.pdf (Accessed: 17 December 2020).

Oracle and/or its affiliates (2015) Oracle® Fusion Middleware Developing Fusion Web Applications with Oracle Application Development Framework. Available at: https://docs.oracle.com/middleware/1212/adf/ADFFD/intro.htm#ADFFD101 (Accessed: 30 January 2021).

Oracle Learning (2019) Merchandising: Merchandise Hierarchy Overview. Available at: https://www.youtube.com/watch?v=km0SebgNVHY (Accessed: 23 February 2021).

Pacheco, B. G. and Rahman, A. (2015) 'Effects of sales promotion type and promotion depth on consumer perceptions: the moderating role of retailer reputation', The International Review of Retail, Distribution and Consumer Research, 25(1). Available at: https://www.tandfonline.com/doi/pdf/10.1080/09593969.2014.918047 (Accessed: 23 February 2021).

Popkowski Leszczyc, P. and Timmermans, H. J. P. (1997) 'Store-Switching Behavior', Marketing Letters, 8. Available at: https://www.researchgate.net/publication/226787185_Store-Switching_Behavior (Accessed: 23 February 2021).

Price Management and Optimization (2020) DemandTec. Available at: https://www.demandtec.com/solutions/price-management-and-optimization/ (Accessed: 16 January 2021).

Pricing Lifecycle Optimisation: Pricing, Promotions & Markdowns (2018) Profimetrics. Available at: https://www.profimetrics.com/pricing-lifecycle-optimisation?lang=pt-pt (Accessed: 16 January 2021).

PwC (2018) 'Experience is everything: Here's how to get it right', PwC. Available at: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ve

d=2ahUKEwjd4tyG463xAhVpyoUKHT2WAbwQFjAAegQIBBAD&url=https%3A%2F%2Fwww.pwc.de%2Fde%2Fconsulting%2Fpwc-consumer-intelligence-series-customer-experience.pdf&usg=AOvVaw3MQjfK7HggLhovzVQ6zmkR (Accessed: 22 December 2020).

Rainergewalt (2020) 'Important Database Fundamentals - When To Use NoSQL Vs SQL – Fly Spaceships With Your Mind', 20 December. Available at: https://starship-knowledge.com/when-to-use-nosql-vs-sql (Accessed: 30 January 2021).

Saaty, R. W. (1987) 'The analytic hierarchy process—what it is and how it is used', Mathematical Modelling, 9(3). Available at: https://www.sciencedirect.com/science/article/pii/0270025587904738 (Accessed: 6 February 2021).

Saaty, T. L. (1990) 'How to make a decision: The analytic hierarchy process', European Journal of Operational Research, 48(1), pp. 9–26. doi: 10.1016/0377-2217(90)90057-I.

Santos, N. A., Ferreira, N. and Machado, R. J. (2021) AMPLA: An Agile Process for Modeling Logical Architectures, Balancing Agile and Disciplined Engineering and Management Approaches for IT Services and Software Products. Available at: www.igi-global.com/chapter/ampla/259171 (Accessed: 16 February 2021).

SAP CPQ : Configure Price Quote Solutions (2020) SAP. Available at: https://www.sap.com/products/cpq.html (Accessed: 16 January 2021).

Schmidt, D. E. (2017) 'How to Achieve Effective Pricing: Leveraging the Power of Price Optimization', Analytics Journal. Available at: https://www.researchgate.net/profile/Marco_Vriens3/publication/313578365_Assessing_the_Impact_of_a_Brand_Crisis_using_Big_Data_The_case_of_the_VW_Diesel_Emission_Crisis/links/58bcb9a845851591c5e3fd2b/Assessing-the-Impact-of-a-Brand-Crisis-using-Big-Data-The-case-of-the-VW-Diesel-Emission-Crisis.pdf#page=86 (Accessed: 16 January 2021).

Shmeltzer, S. (2015) Oracle ADF and Oracle JET - What's the Story? Available at: https://blogs.oracle.com/jdeveloperpm/oracle-adf-and-oracle-jet-whats-the-story (Accessed: 7 December 2020).

Shmeltzer, S. (2019) Oracle ADF or Oracle JET - An Update. Available at: https://blogs.oracle.com/shay/oracle-adf-or-oracle-jet-an-update (Accessed: 7 December 2020).

Systematic reviews and meta-analyses: a step-by-step guide (2013) Centre for Cognitive Ageing and Cognitive Epidemiology. Available at: https://www.ccace.ed.ac.uk/research/software-resources/systematic-reviews-and-meta-analyses (Accessed: 20 February 2021).

The market of ERP systems in a year added 10%. In top five one change. (2019) TAdviser.ru. Available at: https://tadviser.com/index.php/Article:ERP_systems_(world_market) (Accessed: 30 January 2021).

Vesterli, S. (2017) Oracle ADF Survival Guide: Mastering the Application Development Framework. Available at: http://www.books24x7.com/marc.asp?bookid=138108 (Accessed: 17 December 2020).

Young, N. (2016a) ‘Oracle Price Management - Operations Guide’, Oracle. Available at: https://docs.oracle.com/cd/E12440_01/rpm/pdf/160/html/operations_guide/toc.htm (Accessed: 31 December 2020).

Young, N. (2016b) ‘Oracle Retail Price Management User Guide’, Oracle. Available at: https://docs.oracle.com/cd/E12440_01/rpm/pdf/160/html/user_guide/toc.htm (Accessed: 30 December 2020).

Appendix

Appendix A – QEF

Table 10 - Quantitative Evaluation Framework

q	D	Q_j	Dimension	Q_i	W_{ij} (Factor Weight / in Dim // [0,1]	Factor	rw_{jk} (requirement weight k in Factor /) {2, 4, 6, 8, 10}	Requirement	wf_k % requirement fulfillment k) [0,100]
92%	0.22	86	Functionality	79	0.48	Functional (Refering Use Cases)	2	UC1 – Define/ Alter settings	50
							2	UC2 – Define / Alter security settings	50
							2	UC3 – Define / alter maintenance settings	50
							4	UC4 – Define relationships between merchandising hierarchy branches	50
							4	UC5 – Define relationships between organization hierarchy branches	50
							6	UC6 – Define relationships with competitors	100
							8	UC7 – Maintain margin strategy	100
							8	UC8 – Maintain competitor's strategy	100
							8	UC9 – Maintain Relationship strategy	100
							10	UC10 – Perform what if analysis	20
							8	UC11 – Submit price recommendations	100
							10	UC12 – Review and approve price recommendations	100
				89	0.36	User Interaction	10	FUI01 – Solution is intuitive	100
							10	FUI02 – Reduce number of steps necessary to carry out an action	100
							8	FUI03 – All the solution present a same control experience	100
							8	FUI04 – All the solution present a same navigation experience	100
							4	FUI05 – Permissions and user type specific options are guaranteed	60
							8	FUI06 – User interaction have immediate feedback	100
							8	FUI07 – Actions that take more than 10 seconds must run on the background	100
							4	FUI08 – Present the possibility to switch actions processing to the background	0
							4	FUI09 – Access and permissions depending on the defined rolls	60
				100	0.16	Content Quality	6	FCQ02 – The texts are well written and all the setences make perfect sense	100
							6	FCQ03 – All the messages are easy to understand and human personified	100
							6	FCQ04 – All the contents are related to the solution	100
							10	FCQ05 – Modular solution that presents a set of interfaces, adding the possiblity of introducing new features	100
		100	Adaptability	100	0.50	Reliability	6	AR01 – Must be available on the working hours on weekdays	100
							8	AR02 – In case of errors, a clear warning is presented to the user	100
							8	AR03 – Present the functionalities and operations as expected	100
				100	0.33	Usabilidade	8	AU01 –Simple and user-oriented user interface	100
							8	AU02 – The solution must be intuitive and easy to use	100
				100	0.17	Manutenance	6	AM01 – Ability to perform maintenance and updates	100
		83	Efficiency	50	0.17	Security	2	ES01 – Autentication System and access and permissions depending on the defined rolls	50
				90	0.83	Navigation	8	EN01 – The solution has a good structure and allows users to access contents in a intuitive way to the main functions	100
							8	EN02 – The user interface is quick and fast responsible, with progress information	80
							8	EN03 – The solution runtime does not have errors, and unexpectable erros should be well treated	100
							4	EN04 – Login and Access security	50
							8	EN05 –The solution support Different Browsers	100

Appendix B – SWOT

The creation of the SWOT Analysis is attributed to Kenneth Andrews and Roland Christensen, it consists of a model for assessing the competitive position of an organization in the market. For the methodology he uses, there are those who also call it the Harvard model.

In the SWOT analysis, what is done is nothing more than making a diagnosis about the project's strengths and weaknesses, not only looking at our own navel, but also relating them to the project's surroundings.

Thus, the SWOT analysis is performed taking into account two environments: the internal environment, where the strengths and weaknesses are diagnosed, and the external environment, where through its analysis the opportunities and threats in relation to the organization are determined.

Table 11 - SWOT Analysis

	Positive Factors	Negative Factors
Internal Factors	**Strengths** • Product with unique characteristics • Retail Consult already has loyal customers than can be interested in the new solution • Proper market research and planning	**Weakness** • Laws and regulatory parameters constantly changing • Small team and not full time
External Factors	**Opportunities** • Larger retail market • greater demand for technologies that help and facilitate pricing	**Threats** • Different economic laws across all markets • strong competitors

Appendix C – QFD

For companies to remain competitive, organizations must determine what is driving the customer's perception of the value or quality of a product or service. For that, Quality Function Deployment (QFD) is used, which consists of a set of tools used to effectively define customer requirements and convert them into engineering specifications that produce the products that will meet those requirements. This methodology uses a defined set of matrices to facilitate this progression.

To ensure that the solution presented in the dissertation meets the client's requirements and expectations, this technique was used. Based on several meetings held with the project manager, it was possible to conclude the client's needs and prioritize each one. For prioritization, a scale from 1 to 9 was used.

Based on these requirements, the engineering characteristics were identified, and the functional requirements were defined. Then the relationship between the customer's criteria and the identified requirements is made. For this, the scale below was used:

Strong relationship	● = 9
Medium relationship	○ = 3
Weak relationship	▽ = 1

All this analysis was done with the construction of the House of Quality diagram shown in the figure below.

Analyzing the diagram after calculating the relative weights for each requirement, it is possible to conclude that the most important requirements are: the requirements that involve maintaining strategies (7,8,9), the requirement to perform what-if analyzes (10) and the requirements to submit, review and approve a price recommendation (11.12).

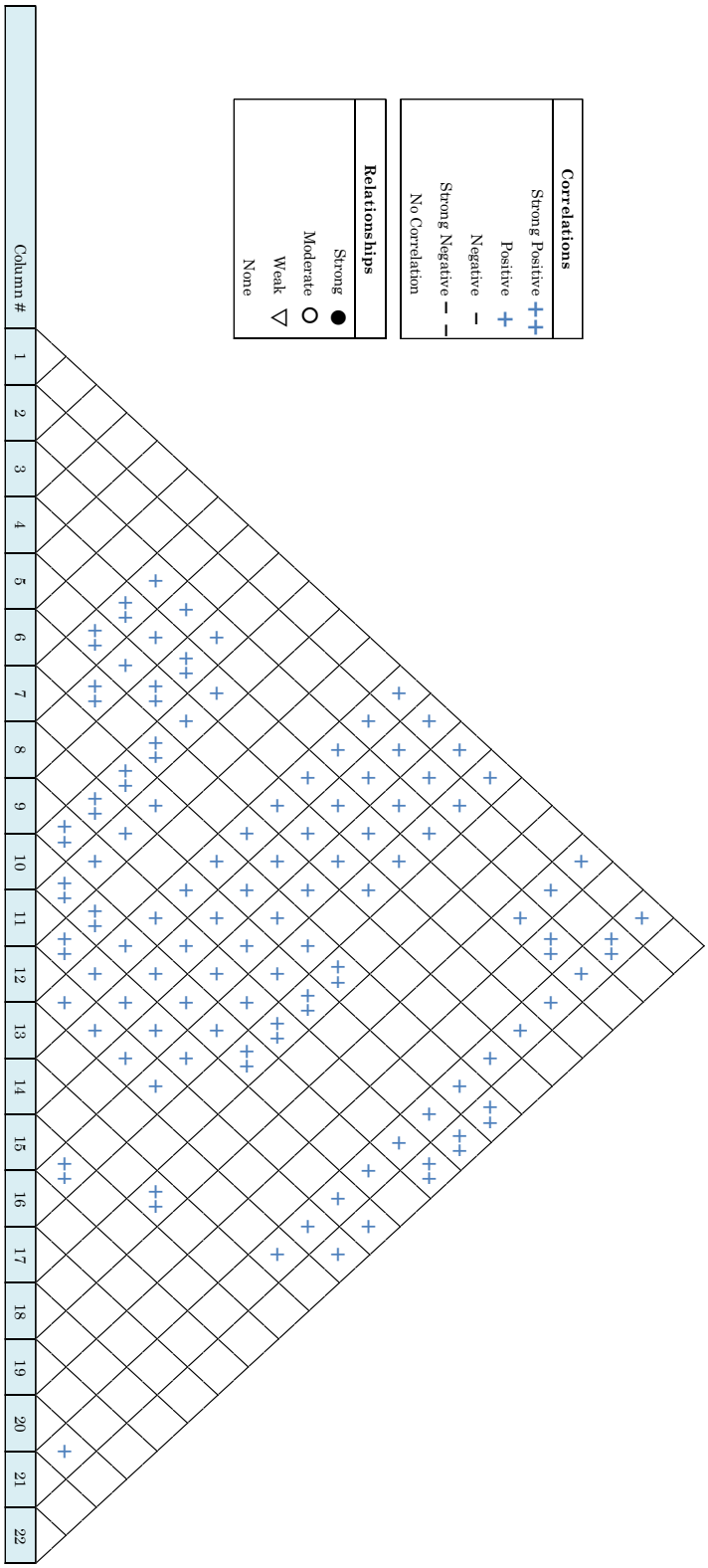

Figure 80 - Quality Function Deployment - Relationship between requirements

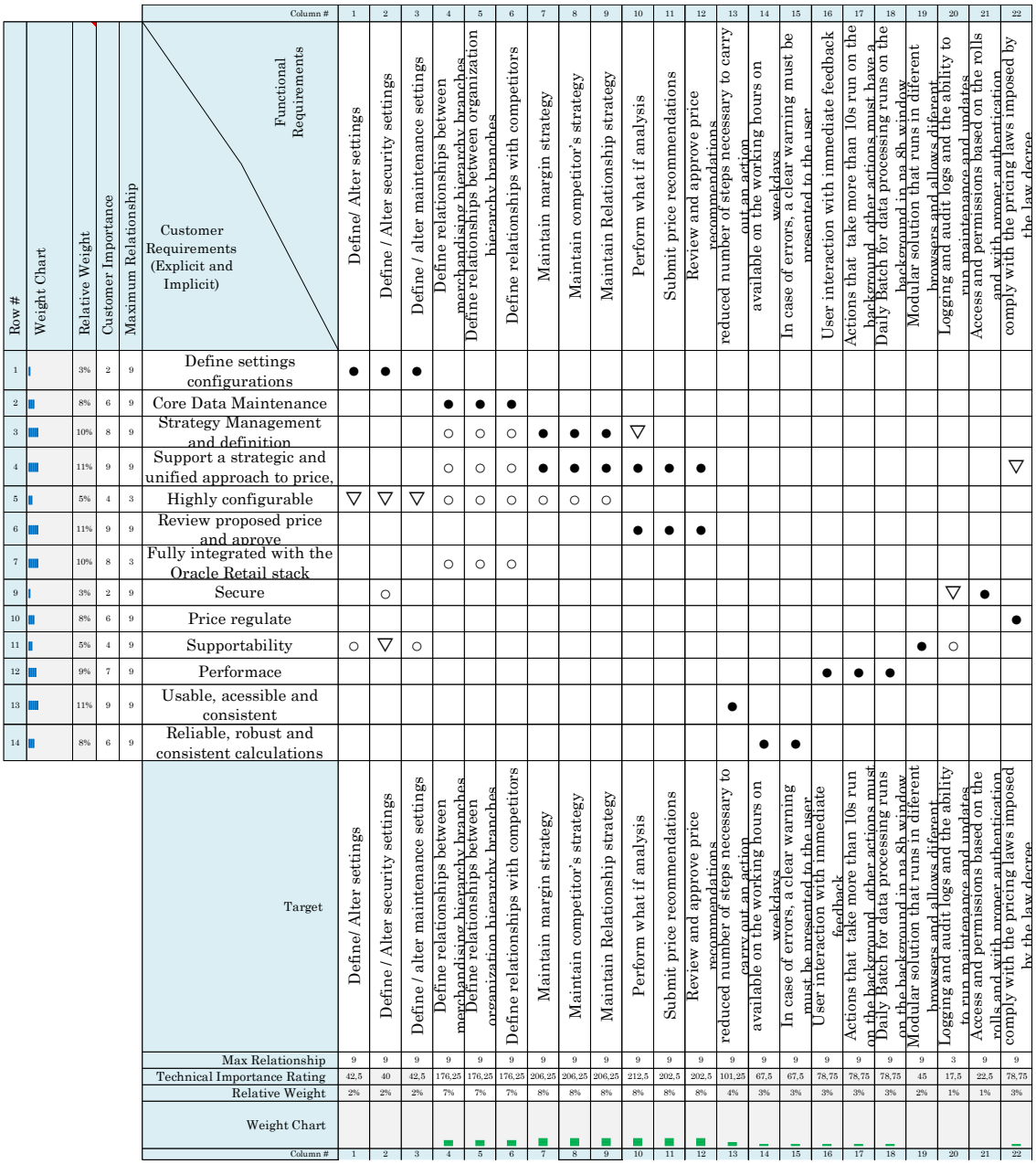

Figure 81 - Quality Function Deployment Diagram

Appendix D – Development Plan

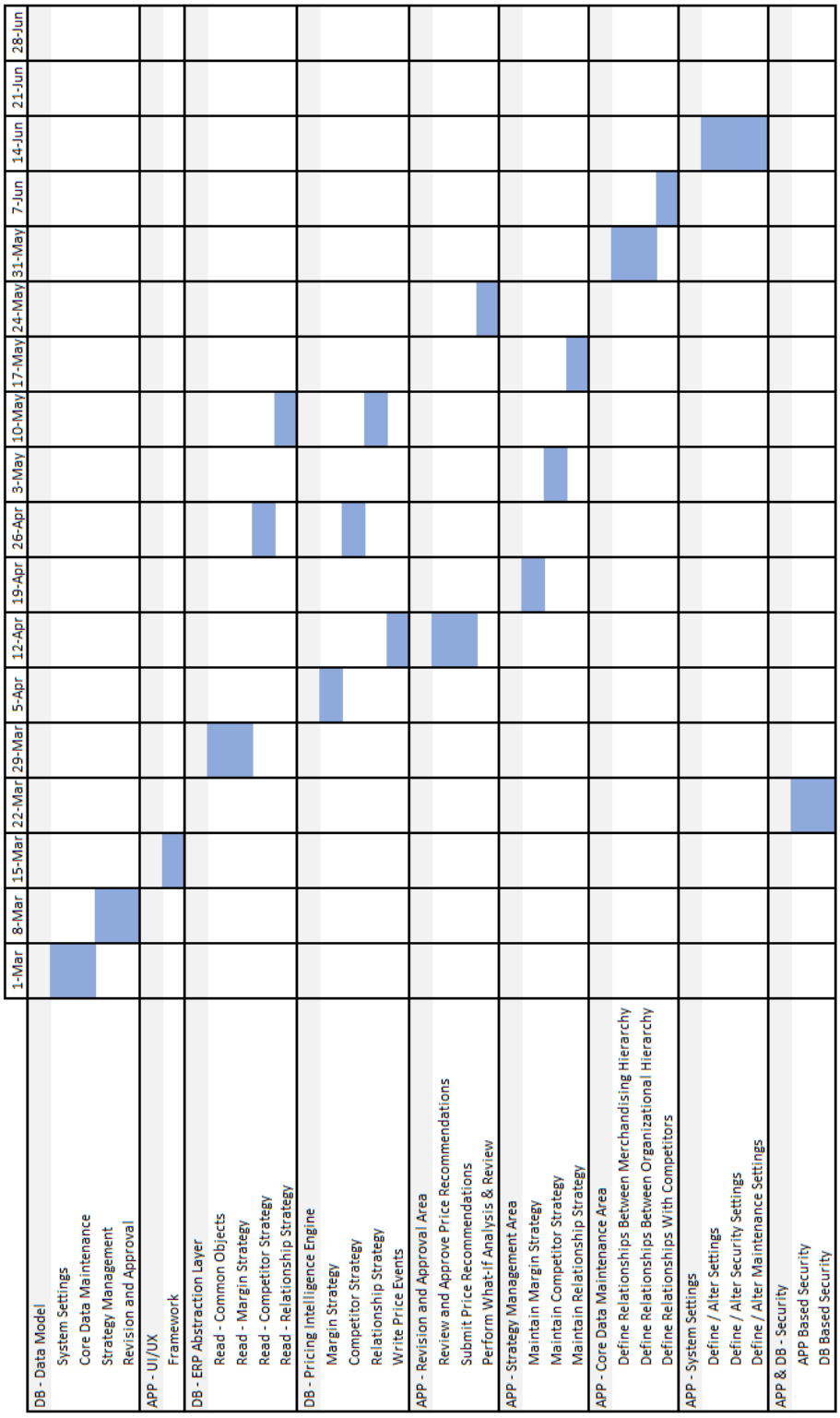

Figure 82 - Development Plan