1st Year Project Report

TEVOS DRUGSTORE

Project Participants:

- **Erik Petra** <1081490@ucn.dk>
- **Krisztián Henrik Papp** <1081477@ucn.dk>
- **Marek Strúcka** <1081474@ucn.dk>
- **Rémi Teissier** <1084954@ucn.dk>
- **Tiago Fernandes** <1081471@ucn.dk>

Supervisors:

- **Gianna Belle** <gibe@ucn.dk>
- **Jesper Strandgård Mortensen** <jesm@ucn.dk>
- **Dimitrios Kondylis** <dko@ucn.dk>
- **Simon Kongshøj** <siko@ucn.dk>

Repository path:
https://github.com/dmai0919-group3/2nd-semester-project

Normal pages / characters: 39.2 pages / 93 973 characters*

*A normal page is defined as 2400 characters incl. whitespaces and footnotes. Frontpage, title page, table of contents, literature list and appendices are not included.

MAY 29, 2020

University College of Northern Denmark
AP Degree in Computer Science
dmai0919 – Group 3

PROFESSIONSHØJSKOLEN

Table of Contents

Abstract

This report holds a detailed solution to the first-year project concerning the Tevos drugstore company. The company has not updated its warehouse order management system for a long time, and it has become insufficient for today's needs of the company. Our solution to the Tevos' business strategy as well as the creation, the design, and the implementation of a completely new company's warehouse system is documented below. An iterative approach together with the Unified Process was used, resulting in a quicker and more responsive system that can ease the workflow of the whole company not only nowadays but will also serve well in the future expansion.

Preface

We would like to thank all our teachers that they helped us with giving feedback on the project, answering our questions, and with their supportive attitude towards our project. The biggest thank goes to our supervisor, Dimitrios Kondylis, for his never-ending support and valuable information he provided us with throughout the work on the project.

We would also like to thank Pavel Herel, the owner of Tevos drugstore, and Igor Strúcka, the sales representative of the company, for trusting us with the project, providing us with the information about the company and spending their time answering our questions.

Introduction

In today's world, there are such competitive markets in some fields that when a company in this field is not innovating its processes and business model, its sales may rapidly fall in the blink of an eye. The lesson to take is simple: innovate or die. As written in a Forbes article (Cole, 2019), the leaders that fear the change, are arrogant or unwilling to take the risk, may quickly fall behind for the sake of those who decided to innovate, to implement changes with advanced technologies used.

This report focuses on a company called Tevos, a big player in the drugstore business in Slovakia. Tevos drugstore was created by Pavel Herel in 1992 and can still be characterized as a small enterprise. It started as a small store in his hometown, where mostly his relatives and friends kept coming. Pavel soon opened his second store in a different part of the town, and then another in a town nearby. Tevos has grown huge since then, now has a large warehouse and more than 25 stores in most of the west and central Slovakia towns. Its position on the Slovak drugstore market is well deserved and indisputable. However, it is not as strong as it used to be since this is a notable example of the company that forgot to innovate.

Their problem

The warehouse works as it is for a very long time, and so does the IT system. Since the emphasis was put on opening as many new stores as possible, the innovation of warehouse management and

ordering system was postponed until the very last moment. Now, with so many stores opened and such a huge database, the company has reached a point of no return, and they are in dire need of a new system. The process of ordering supplies by the stores needs to be thoroughly refined and automatized. Also, statistics are especially vital nowadays, so we should aim to provide the managers with a tool to see those.

Problem statement questions

Three main questions had arisen when we familiarized ourselves with the problem. These, also written in the *Full problem statement*, are:

- Are we able to design a brand-new system for the company that would make the process of ordering more effective?
- Are we able to create such databases for the drugstore, that the stock-taking and other information could be easily retrieved?
- Are we able to ensure the smoothness, faultlessness, and user-friendliness of the new system?

For the complete problem statement, please check Appendix E.

The focus of the project

The project focuses mostly on delivering the new user-friendly IT- system for the company. The system the company uses nowadays has a huge disadvantage: the stores' employees cannot see the central database, so they are unable to make orders by themselves. The critical task to solve is connecting the stores to a central database so that the orders would be made by the stores' employees that would see the updated central database every time.

Structure of the report

There were seven different parts - we divided the report into - distinguished, taking a chronological approach in describing the workflow. The report started with an abstract, a preface, and an introduction. After this, a reader can expect to see a detailed report setup, some helpful information about the project and a phase plan, followed by the following parts:

Part I. ***Preliminary study***
Part II. ***Methodology, Architecture and Technology***
Part III. ***Inception***
Part IV. ***Elaboration***
Part V. ***Construction***
Part VI. ***Transition***
Part VII. ***Conclusion***

The report is then wrapped up with an Appendices Part, where we included the Literature List & References, our Customer Interview, Group Contract, Design Class Diagram – Full, Full problem statement and a User Manual for the finished software.

Detailed report setup

The list mentioned above of 8 various parts follows the phases of a system development according to a Unified Process. The report is divided into logical and chronological parts, which describe the solutions to the various parts of the project:

Part 1: Preliminary Study, where the emphasis is put into the Business analysis of the Tevos company. Starting with the customer interview, followed by descriptions of various business indicators – structure and culture of the company, its SWOT and Porter's five analysis, stakeholder analysis, finished with comments on business strategy of the company and its E-commerce.

Part 2: Methodology, Architecture and Technology, where a discussion about various methodologies used in system development is held, followed by the description of the Unified process we opted for and a three-layer architecture. Finally, the technologies to be used in this project are explained to the reader.

Part 3: Inception, with all the requirements and features of the system listed. Activity diagram, Employee-Task-Goal table, mockups, and initial use cases are found. This phase also contains a Business case, a proposal to the client where our observations, recommendations, and conclusion on the feasibility of the project are presented.

Part 4: Elaboration, where work on the two most important fully dressed use cases together with all the needed artifacts and testing is documented. The domain model and the relational model are also in this part.

Part 5: Construction, where the whole programming of the project is described- code standards, code snippets, tests, architecture, minor use cases and the problems we met while implementing the system.

Part 6: Transition, although we were not concerned with this final part of the UP, we have at least produced a user manual for the future users of the system.

Part 7: Conclusion, a look back on the initial problem statement and conclusion can be found in this part, together with evaluating the workflow in the group, the problems met while working on the project and the group contract can be found in this part as well.

Part 8: Appendices, including a list of references, a glossary, a user manual, a complete design class diagram and

Helpful information

SVN Repository Path and Revision Number:

<REDACTED>

Git Repository Path[1]:

https://github.com/dmai0919-group3/2nd-semester-project

Final executable download link:

https://github.com/dmai0919-group3/2nd-semester-project/releases

MS SQL Database Hostname: <REDACTED>, **Database name:** <REDACTED>,
Username: <REDACTED> and **Password:** <REDACTED>

Starting the software:

1. Download the *v1-0-0.zip* from GitHub and extract it on your computer
2. Write the database credentials from above into the 'config.properties' file
3. Start the software with the `java -jar dmai0919_2Sem_3.jar` command
4. Log in to the system with one of the following demo users:

Role	Username	Password
Warehouse	Warehouse	Password
Store	store	password

Figure 1: Demo users for the system

The clickable links in the report are marked in *italic*, the links which are pointing to an external website are *underlined and italic* and every quote/reference have a highlighted background.

[1] As there were some technical problems with the SVN server (kraka.ucn.dk) provided by UCN at the end of the project (May 26th), on that date we changed to a GitHub repository for the last few days of the project and therefore the final version of the project can be found on GitHub instead of SVN.

Phase plan

In order to be able to manage the system development smoothly, different tools were created to help the developers. Since we decided to use the Unified Process as our project methodology, we had to divide and conquer our project iteratively. For the iterations, a Phase Plan table shown below was created. Here we could see UP phases of the project together with their primary goals, the number of iterations, the deadline, and the artifacts to be produced during each phase. Since we were sure that this project would take longer to implement and it was not going to get to the transition phase by the end of May, we wrote the deadlines accordingly. We found this table extremely helpful throughout the project.

Phase / Milestone	Inception	Elaboration	Construction	Transition
Goal	To collect the requirements, get an understanding of the project, agree on its feasibility	To determine the most complex use cases, design, implement and test them, outline the architecture	The complete system is designed, developed, and tested	A user manual is created
Iterations	1	2	3	1
Deadline	April 27, 2020	May 18, 2020	June 2020	July 2020
Artifacts	Business case Activity diagram, ETG table Use cases, mockups Risks, plans, estimates	Fully dressed use cases + artifacts + tests + implementation Domain and relational model + SQL scripts	Use cases revised, implemented, tested Documentation	The system is ready to be deployed User manual

Figure 2: Phase Plan table

Based on this table, we created a Gantt chart, which helps us visualize the timeline of the project as well as the deadlines and goals we set for ourselves. Large UP phases were split into smaller sub-parts of each phase, the day limit was assigned to each task and milestones were created at the end of each bigger task (mostly each UP phase). We used free software, called GanttProject (https://www.ganttproject.biz/) and made the following chart:

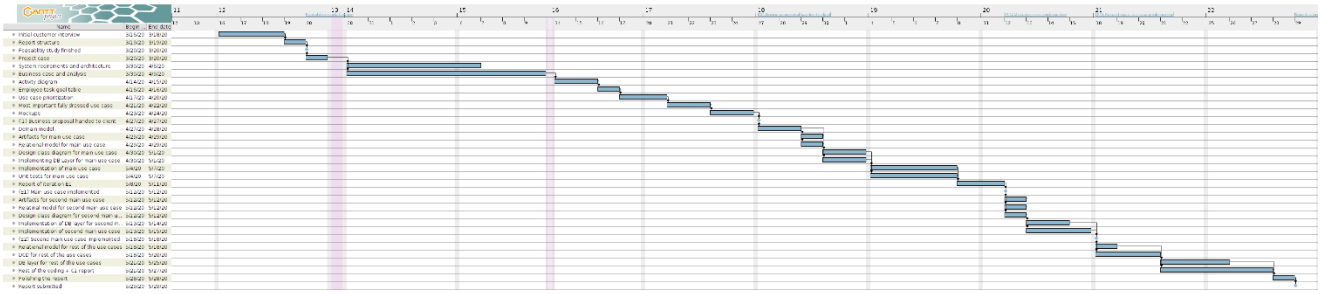


Figure 3: Gantt Chart showing the timeline of the project

Another tool we used was a task board that was provided to us by Microsoft, called Planner. With this task board, the developers and supervisors both can find the tasks assigned to them, with a brief description and a deadline for each of them. Every group member could also create a new task, assign it to various people, and set its deadline. The task board example can also be seen below (Figure 2).

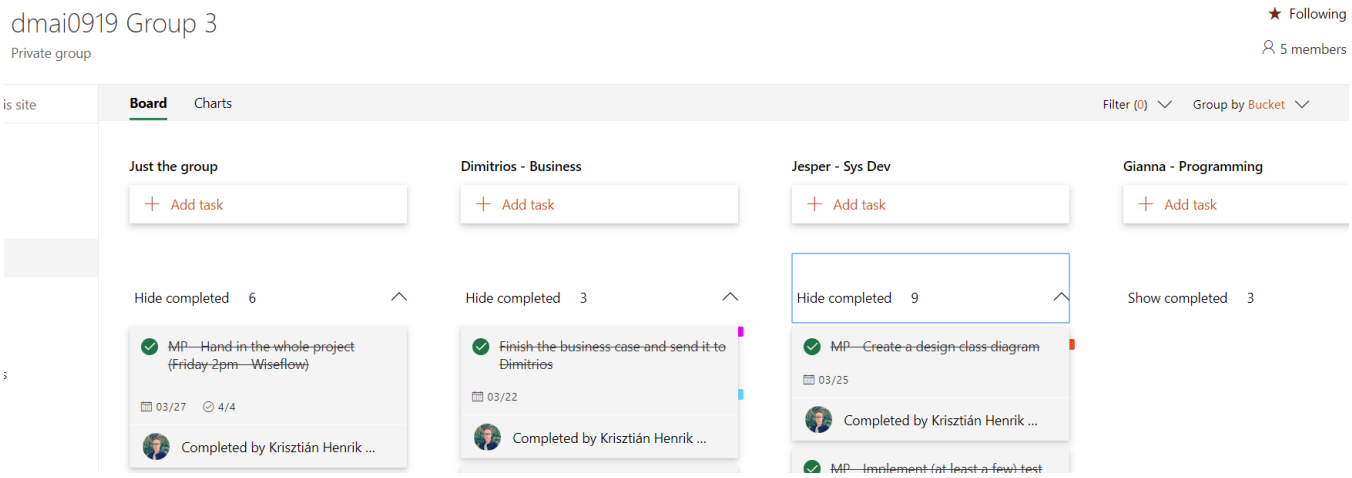


Figure 4: Microsoft Planner – Task Board

Part I.
Preliminary study

The preliminary study part focuses mostly on analyzing the company, its business model, and the problems within the company. At the very beginning, an interview was conducted. With all the information gathered, a business analysis of the company with plenty of business artifacts was created.

1. Customer Interview

The first step was interviewing at least one employee or, at best, also to interview the owner of the Tevos drugstore himself. We first discussed the issues we wanted to resolve with the company's representatives. We found possible blind spots in our vision of mostly the business part of the project, and we finally wrote down the questions we wanted to ask during the interview. Some of those can be found in *Appendix B*.

We managed to interview the sales representative for around an hour, and we also managed to talk to the owner himself, but just for a brief period. The notes we took from the interviews can be found in *Appendix B* as well.

After the interview, we felt like we had an overview of the company and about the new system we were to develop. We started with the preliminary study, which is documented below. While doing so, there were new questions raised, and some crucial points had to be resolved, so we tried to reach out to the company representatives once again. The sales representative was willing to spend some time with us in order to ensure a great result, so we interviewed him again for an additional 30 minutes, which helped us a lot.

2. Organizational Form, Structure and Culture

This section describes how the company works daily. Since all these elements were formed somehow naturally as the time passed, without major intervention from the owner, an owner itself was initially struggling to outline those. The report to these quite challenging observations is below:

2.1. Organizational Form

The company itself has only 12 people present at the warehouse every day, 5 of which are warehouse workers, and 3 are warehouse drivers, meaning they have no executive power. Therefore, the company, although it has many stores across Slovakia, is a small enterprise because there are only one owner and three different officers as the executives for the whole company. The owner of a company is an autocratic leader who makes most of the decisions on his own.

If looked at the company in terms of their ability to see and adapt to changes, Tevos Drugstore could be considered a "reactor" organization.

"…reactor: act only when environmental change 'forces' them to do so. They are not 'proactive' organizations." (Miles, Snow, Meyer, & Coleman, Jr.)

More examples of why this company is a reactor will be obvious later.

2.2. Organizational Structure

> The purpose of structure is to organize and distribute work among the members of an organization so that their activities are best harnessed to meet the organization's goals." (Brooks, 2018)

This company has a simple structure that is typical for these kinds of small enterprises. The owner is the autocratic leader, and three officers being there to manage the whole company according to his will strictly - centralization can be observed here. The IT & administration manager works mostly on his own, directly reporting to the owner, the sales representative is the one to whom all the stores' salespeople report to and communicate with. The warehouse manager oversees the whole warehouse and warehouse workers and reports to the warehouse director. Only these three officers are in contact with the owner, so the owner's span of control – 3 – makes it easy for him to run such a well-established business. The only specialization identified is between the workers and drivers; even though both work in the warehouse, their roles are not interchangeable. The organization is mechanistic, with a high level of formalization and specialization, a stable environment and with a lot of rules and routines to follow and obey.

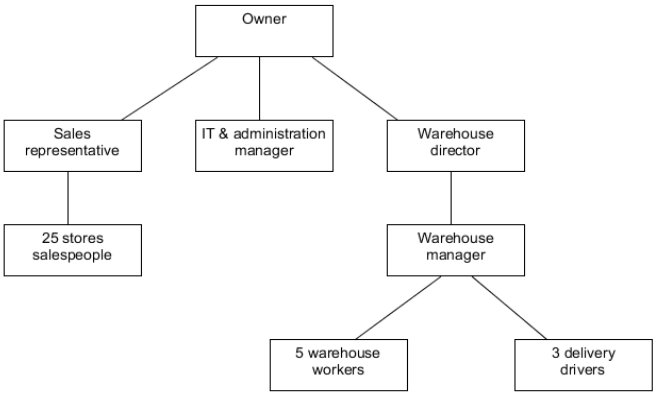


Figure 5: Organizational structure

2.3. Organizational Culture

> "Organizational culture is often seen as an 'intangible glue' which binds people together. As such, it is argued, culture reduces potential conflict in organizations which have a strongly held value and a belief system." (Brooks, 2018)

The culture within the organization may be described as informal and task-oriented. However, we may see a power culture in there, since it is a small company, and the competence of the owner is essential.

Since most of the employees work in the warehouse for many years, they know each other well, and they developed friendly relationships among themselves. If we were to define the stage of the warehouse workers group according to a model from the book Organizational Psychology (Bass & Ryterband, 1979), the workers are forming a group in stage 4. They already know, trust each other, and now they independently work together as a team instead of competing among themselves, with their daily goal being the completion of all the orders.

According to Gray's book on Project Management (Larson & Gray, 2010), ten primary characteristics contribute to the culture within the company. Assessing by those, the management's focus is mostly on tasks, and there is an emphasis on a group since the workers have to finish the job, it does not matter which individual will contribute the most to it. The risk and conflict tolerance are low within the

company, and the workers are tightly controlled. The performance is not a reward criterion at all in this company.

2.4. Stakeholder analysis

"A stakeholder in an organization is (by definition) any group or individual who can affect or is affected by the achievement of the organization's objectives." (Freeman, Harrison, Wicks, Parman, & Colle, 2010)

We differentiated seven groups of stakeholders in this project, not including the project team itself. These stakeholders will contribute the most to the system, the decisions are made by the owner and the warehouse director, followed by the sales representative as the spokesperson for all the stores people.

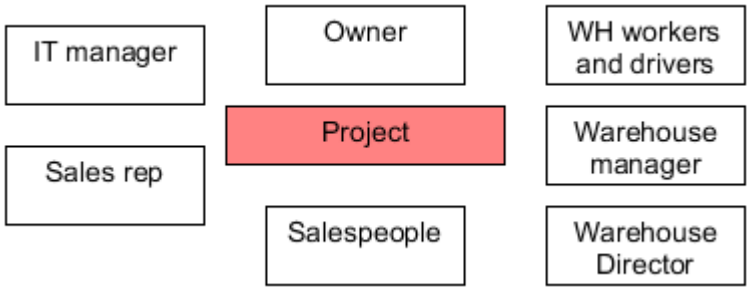


Figure 6: Stakeholders of the company

After finding the most important stakeholders, we drew a stakeholder table:

Stakeholder	Contribution	Interest	Influence/Attitude	Importance
Owner	Money, decisions, feedback	A better system for communication between stores and warehouse	High/ neutral	High
Warehouse director	Decisions, feedback, knowledge	A system that will unburden them of some work	High/ positive	High
Warehouse manager	Feedback, knowledge		Medium/ positive	High
Sales rep	Feedback	A more user-friendly system	High/ positive	Medium
IT manager			Medium/ positive	Medium
WH workers and drivers			Low/ neutral	Medium
Salespeople			Low/ negative	Low

Figure 7: Company stakeholder Table

This table was created after the interview was conducted. The feeling was that the owner was forced by his employees to undertake all this finally. However, since he is mostly an autocratic leader, he must have given many thoughts to it to approve it- and this was also planned in strategic goals - so for now, his attitude stays neutral, but it is crucial to satisfy his expectations. The warehouse director and manager are welcoming to this change that would unburden them. The attitude of salespeople in the stores is negative since this might be another thing they will have to learn. It must be explained to them how the system can make the orders easier and how to use the system that must be as user-friendly as possible.

2.5. Change management

Since the motivation of employees in the company is low and they are more than happy to follow the well-defined rules, there are no relevant examples of any changes and their management in this company. Using knowledge of the company's culture and hierarchy, it can be said that when the change is about to happen, it does not follow the proper description of Kotter's 8 step model (Kotter, 2019), talking about the steps to implement the changes successfully.

It is assumed that future change would follow a top-to-bottom approach since the decisions must be made or approved by one of the executives. The management decides the changes, and those are then implemented to the workflow, with an active learning approach being present for the next few days to see whether the change has helped the company. The upcoming IT system will undoubtedly be the most significant change in the last years, so it must be approached carefully. As written in the book, Organizational Behavior:

> "Change can be a source of frustration, fear and anxiety or a challenge and renewed source of motivation: perhaps even both. Organizational change, and the manner in which it is managed, and, if appropriate, communicated, can have a lasting impact on the motivation of those involved." (Brooks, 2018)

This project and the changes that would be implemented within its scope are expected to raise the motivation of the workers and ease the company's culture and everyday workflow.

2.6. Supply Chain Management and Logistics

> "Supply chain management (SCM) involves the coordination of all supply activities of an organization from its suppliers and delivery of products to its customers." (Chaffey, 2011)

We distinguish between the inbound (to the store) and outbound (to the customer) logistics, which are both handled differently. If those two works, then the company is on the right way to flourish in the future.

2.6.1. Supply Chain Management (SCM)

The supply chain of the Tevos drugstore consists of large suppliers without intermediaries. This is a huge advantage since the prices could be lowered when they have a direct supplier. Since the

drugstore's focus is on the well-known products and low prices, this – getting abundant supplies from suppliers – is contributing to the other factor involved in having the lowest prices: the economies of scale, since the per-unit cost is low.

It may be said without hesitation that the SCM of Tevos is cost-driven, with the economies of scale, no intermediaries, standard product types in their portfolio, and lowest prices.

2.6.2. Inbound logistics

The inbound logistics was touched even above, and it depends on the large suppliers and their trucks. Since Tevos is also a big player on the market, the suppliers keep fulfilling their obligations and supply the warehouse regularly.

2.6.3. Outbound logistics

The outbound logistics is handled entirely by the warehouse and its drivers. The warehouse is in the town of Prievidza that has an excellent location. 15 out of 27 stores are within an hour of driving.

2.6.4. The warehouse layout and the supply routes

The warehouse workers structured the warehouse layout themselves, so it is well designed, and everything has its place. On the other hand, the workers know the locations by heart; the products are not located at a strictly defined place every time. This is a disadvantage for the newcomers since they must learn the layout of the warehouse in the first days.

The supply routes were designed by the drivers to be as efficient as possible. Also, when a store is closed, or a new one is opened, the routes are refined. They use only trucks since there is no way to use a different kind of transport in these parts of Slovakia. The truck has a maximum load of 3 tons, and they usually load supply for three stores into one truck. It is required for a store to order a maximum of one ton of supply, the warehouse manager oversees this, and if the limit is reached, other stores should lower their orders to keep the whole truck's load under 3 tons. The weight limit warning is also a feature to be implemented in the software.

2.6.5. Waste disposal

The waste disposal is also managed by warehouse drivers in their working time on the company's oldest truck, and this could not be more efficient. It can be said that the logistics part works well in the company.

2.7. ABC analysis, Pareto law

The ABC analysis is a handy tool to keep track of the products and their sales together with their business importance for the company. When combined with a Pareto 80/20 law (Pareto, 1980), which in general states, that 20% of the effort is responsible for 80% of the results, it can be translated into anything; in this case, the Pareto law would be understood as "20% of the products are responsible for 80% of the sales." Knowing this, an ABC analysis of these products will show their contribution margin

– the accurate display of the most profitable products. With this analysis in hand, the company should, in theory, try its best to ensure as smooth and cheap logistics for these products as possible.

For the Tevos drugstore, the ABC analysis is an unknown term. Since their statistical information is inferior and needs to be implemented in a better way in the new system, performing the ABC analysis is impossible nowadays. The managers just assume which are the top 20% of products from their experience, but they do not have that confirmed. The lack of an ABC analysis is a huge issue that may contribute to low sales of the company, and the project team should aim to implement such statistics, that performing the ABC analysis would then be possible for the company.

2.8. SWOT analysis

A Strength, Weakness, Opportunity, and Threat (or SWOT) analysis is a crucial artifact when talking about business. Every company must know its SWOTs. The analysis of the Tevos drugstore is summed up in the table below:

Strengths	Weaknesses
• Lowest prices • Well set up in major towns • Loyal customer base	• One warehouse for the whole company • Messy IT system • Unmotivated employees • Weak advertisement
Opportunities	**Threats**
• Expansion to the eastern part of Slovakia • Opening a second warehouse • Salary bonuses, team building events • Better advertisement	• More powerful competitors • Insufficiency of warehouse storage area • Government increasing the minimum wage • Shopping malls and supermarkets • Pandemics and border closure

Figure 8: SWOT Analysis

Strengths

The significant strengths of the Tevos drugstore are the prices and the location in large towns. However, converting from Slovak crown into Euro has diminished the people's feeling of prices, since nowadays the difference between the prices is less visible: 270 Sk and 300 Sk was perceived a lot differently than 9.10 € and 9.99 €. The customer base is loyal but consists mostly of older people that are used to shop in Tevos.

Weaknesses

Having only one warehouse is a massive drawback for the company, halting some major expansion. The messy IT system makes the already unmotivated employees working with it angry, which prolongs

the time spent on tasks that could be finished in a shorter period. The level of advertising is the smallest when compared to competitors; therefore, the drugstore cannot attract new customers to its stores.

Opportunities

Concerning opportunities, if the owner was open to suggestions, there are huge. Firstly, the 2nd warehouse would have to be opened, located more to the east of Slovakia. Then the expansion to the eastern part of Slovakia could be rapid since the competition there is also the lowest. If there was no will to go for such a risky move, various things could be improved in the current state as well, e.g., doing better marketing by mostly improving online advertising. Also, improving the environment in the warehouse by doing team-building events or having salary bonuses would help the company.

Threats

Transnational companies are showing up on the Slovak drugstore market, and since they are more powerful and have more money, they are a considerable threat to Tevos. Since most of Tevos' stores are in ordinary buildings, people are less willing to invest their time to go and shop in specialized shops. They are more likely to buy drugstore equipment in either a store located in a shopping mall or a supermarket.

The Slovak government does not help the small entrepreneurs either, since it increases the minimum wage every while, meaning the wages of all the salespeople in Tevos' stores must be increased too, and the amount of taxes to deduct is higher. This is fatal for small entrepreneurs since they would like to raise the wages for people, but when they add up the taxes to it, they simply cannot do that. Moreover, now, in the time of coronavirus pandemics, since most of the suppliers have their warehouses outside Slovakia, it is difficult for Tevos to get their requested goods since the borders are closed. If this continues for a more extended period, it will have a massive impact on Tevos' economics.

2.9. Porter's five forces

Porter's five forces (Porter, 1979) are another great tool when analyzing the company and its position on the market. This tool helps us to find the competition and helps us to predict its impact on our business. There are five steps in this model:

Buyer power: Every person needs drugstore products daily, which is suitable for the company. On the other hand, as mentioned earlier, people become less willing to shop in specialized shops and go to the supermarket or a shopping mall to buy everything they need in one place. With this, the buyer power is diminishing each year for a specialized store as the drugstore is. The fewer people will come to their stores, the easier it would be for the drugstore to keep its prices.

Supplier power: The Tevos drugstore has different suppliers, but all of them are substantial transnational companies, and each supply comes in several trucks. Although Tevos is a significant and well-established player in the drugstore business only in Slovakia, it has no power to dictate the prices for the transnational companies that are supplying its warehouse. Switching the suppliers is impossible

since the drugstore sells the most commercially known brands and products which are produced only by the actual supplier and no one else.

The threat of new entry: It is also impossible nowadays to come to a Slovak drugstore market and open another drugstore company. Despite that, it may be possible to open some local drugstores in Slovak towns, which could diminish the sales in the Tevos' store, which may result in closing down that store completely.

The threat of substitution: This was mentioned above in the "buyer power" section, as the people are more likely to visit one huge supermarket instead of five specialized shops, the Tevos drugstore has suffered from this a lot in the past few years and is expected to suffer even more if no change would be done.

Competitive rivalry: There are two leading players in the drugstore market: TETA drugstores and DM drugstores. They are far more significant than Tevos is, but Tevos has had a stable customer base for a long time. Nowadays, also supermarkets have appeared and are now a part of the market since they offer many drugstore products for even lower prices than Tevos has due to the economies of scale they use to get supplies of these products. The supermarkets and the people's will to visit a supermarket mentioned earlier are the main threats for the future existence and expansion plans of Tevos.

To sum it up, today's problem of more significant players destroying smaller businesses in every market is the biggest threat to the business model of Tevos drugstore as well. Since the situation does not look like it is about to change, the Tevos must change its business model to survive on the market.

3. Mission, Vision and Slogan of the company

The company's mission, vision and slogan statements should reflect the course the company was taking from the very beginning (mission, slogan) and is willing to continue fulfilling (vision). For Tevos, these were easily identified, since they were created a long time ago, and these three quotes are something that the company follows ever since.

3.1. Slogan

The Tevos drugstore started as a family business, and it was the slow but ongoing expansion that has transformed the initially small stores to self-service ones. Even though the outlook of the store has changed, the first slogan behind the Tevos has stayed the same:

"The prices are falling, even in your town!"

3.2. Mission

Tevos was the definition of low prices ever since. Since it started in smaller towns, the customers knew the cashiers and vice versa, so they knew where to go if they wanted to get the best products and not to get scammed. Since its customer base is still mostly loyal older people, this is applicable even today.:

"Our mission is to handle all the customers as if they were our relatives and provide them the lowest possible prices and quality service."

3.3. Vision

The vision of the company has evolved with time, and it has finally grown this huge:

"Our vision is to provide the cheapest drugstore for everyone in every major Slovak city at least once."

This also defines the strategy they would like to take for the future- to expand even more to new large Slovak towns. The standard of lowest prices is to be kept as well.

4. E-business and E-commerce

"The E-business is aimed at enhancing the competitiveness of an organization by deploying innovative information and communications technology throughout an organization and beyond, through links to partners and customers." (Chaffey, 2011)

The company's E-marketing is at a deficient level without any E-marketing planning. The company can be undoubtedly characterized as a **"brick and mortar"** enterprise (Chaffey, 2011). They did not manage to catch up with the potential of internet marketing at the start, and it goes with them until now. The executives thought that it is not necessary to have a Facebook site or an Instagram profile, nor to pay for internet advertising. It was partly true since there are not so many older people in Slovakia that use the internet daily. However, the problem is that the company has failed to reach out to the younger generation via the internet.

Nowadays, the company has only a Facebook site that is vaguely administered and has a limited reach. They have no paid online advertisement at all, nor they have any other social network profile. Compared to other competitors in the business, this is the area where they are behind. It is predicted that the internet will be here for some time; it is never too late to set up an internet marketing for such company with so huge E-advertising potential.

5. Business Strategy

"Business strategy is the direction and scope of an organization over the long-term: which achieves advantage for the organization through its configuration of resources within a changing environment to meet the needs of markets and to fulfill stakeholder expectations." (Johnson, Scholes, & Whittington, 2005)

To sum it up, the business strategy for the drugstore is to keep up with the trends that were set previously: to continue with the expansion by opening new stores, to offer a wider variety of products, store them in the warehouse and try to sell them to the customers continuously.

The company had well-defined expansion plans, namely:

- List of towns where to open a new store and in what year, up to 2024
- List of new products that will be distributed by the suppliers Tevos want to obtain and sell in its stores
- If the revenues are rising by at least 5% each year until 2022, look for a new bigger warehouse nearby
- If the revenues are rising even more rapidly, do market research in eastern Slovakia and consider setting up a second warehouse and opening five stores at once there
- Renew the current IT-system and the hardware by the end of 2020
- Think about an E-shop if the drugstore products' online sales keep increasing

Since our team spoke to the sales rep and the owner when the COVID 19 pandemic was peaking in Europe, the expansion plans have been halted, and they will most likely have to be redefined. As the owner said, overcoming the actual situation without losing stores and employees is the priority. He also understood the value of statistical information about his stores and warehouse, so he urges the team to implement this feature in the system.

6. Conclusion on preliminary study

This preliminary study of Tevos' company describes its business model and the overall structure and the workflow within the company. As it was observed, various aspects of the company are not working as they should, whereas the other parts work correctly. The aim here is to provide such solutions and ideas so that the whole company would function even better after this project is finished.

Part II.
Methodology, Architecture and Technology

In this shorter part, the focus is put on the choice of adequate methodology the group of developers will use throughout the project. Starting with a description of the most known ones, the group eventually finds the right one for this project. The group's outlook on the technologies that will be used and the architecture that will be applied in the IT- system is discussed in this part.

7. Types of methodologies

Over the past years, different approaches to build projects from scratch were developed and worshipped. It is crucial to define a system development strategy for your project at the very beginning because this determines various things – the time spent working on the project, the approach to coding, testing, error handling or refactoring. Among the most common methodologies, there are the following:

Waterfall approach: The waterfall approach is the old and traditional method of software development with a clear structure. The crucial part here is gathering all the requirements at the very beginning because the basis of this approach lies in moving forward to the next phase all the time, never coming back to the earlier one. This approach was valid back in the days when systems were not so complicated, and the requirements were straightforward, but with most of today 's systems need of maintenance, this approach is not applicable.

Extreme programming: XP methodology has its basis in agile programming, putting even more focus on the unstable environment. It aims to lower the costs wherever possible at early phases, but this may turn out badly, and the expenses might be high in the later phases if not executed correctly. It is heavily reliant on the project team.

Unified process: The UP methodology uses an iterative, use-case driven approach to system development. The requirements are defined at an early stage, and right after that, the main use cases are designed, implemented, and tested in the Elaboration phase. This ensured a high-quality system from early on. Afterwards, less important use-cases are implemented in the next iterations of a Construction phase. The advantage lies in the proper maintenance of a system since the processes are well documented." (TatvaSoft, 2015)

8. The development process in this report

The group of developers decided to use the **Unified Process** as a development process. We find it to be the best tool to control the work on the project, since we can review and test each iteration, which, if planned well, will not disrupt the schedule. The preliminary investigation has been done; the project was now moved to the Inception phase where the system is designed, the requirements will be written down, and the artefacts will be produced, which will later advance into two Elaboration phases with the most important use cases coded and tested; finalizing it with plenty Construction phases where the rest of the system will be designed, implemented and tested. By the end of the construction phase, the project should be ready for a transition and use.

After this Preliminary study part, the report is divided into the UP Phases as seen in Figure 5, starting with Inception, where mostly the requirements were written down, elaborating on them in the Elaboration phase, where the primary use cases were designed entirely together with the domain and relational model. Followed by the Construction and the Transition phase, the Unified Process comes to the finish line.

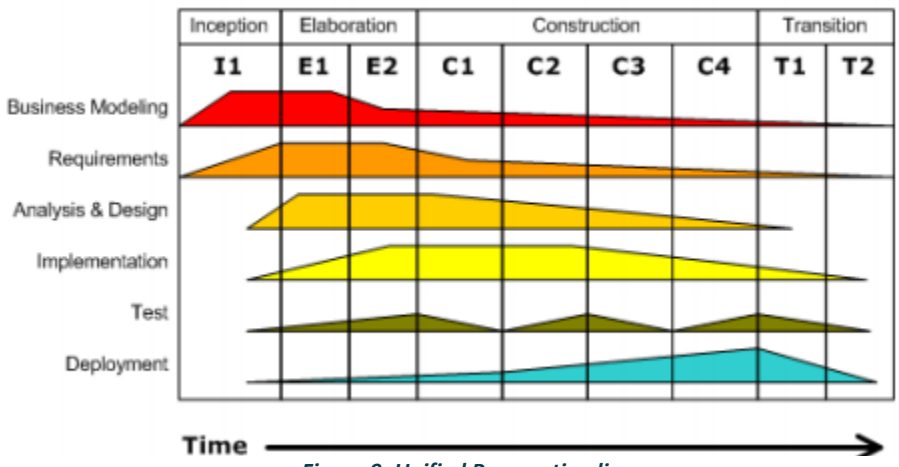


Figure 9: Unified Process timeline

9. Technology used

For the database, we will use Microsoft SQL Server, which will be hosted at an external cloud provider to offer 99.95% uptime and backup features. This, along with Java backend and GUI in Java Swing, will supply stable software and a smooth development process.

10. Software architecture

For software architecture, we opted for three-layered architecture since the development team is most comfortable using it. This process is described in more detail later in this report, during the Construction phase (Part V.).

11. Conclusion

The project team has, after a lively discussion, chosen the Unified Process as the system development process for this project. The whole report from now on will be structured accordingly. The three-layered architecture will be implemented in this project, with the UI layer being a graphical one.

Part III. Inception

The very first phase of a Unified Process is called an Inception Phase. In this phase, the very first iteration is conducted, where initial system development artifacts such as an activity diagram, Employee-Task-Goal table, or a use case diagram with the description and prioritization of use cases are produced. A business case is created in this phase, as well.

1. Business Case

1.1. Business problem

Since the COVID-19 pandemics is a significant threat for all businesses these days in the whole world, it is even more difficult for our team to outline the future, and improvements for Tevos Drugstore, not knowing for how long will the situation is going to stay like this and what consequences will it have.

The drugstore's main problem lies in the outdated and slow ordering system and the hardware they use. Lack of E-business and E-commerce, together with an insufficient warehousing area and competitive market, are also among the main threats and weaknesses.

1.2. Business opportunity

The most significant opportunities that were found lie in the E-business side and the IT system itself.

The team has shown its arguments and initial calculations to the owner, Pavel Herel, and he decided to commit to this IT-system project with a total capital investment of DKK 2 million.

1.2.1. IT-system

The IT system is expected to save a huge amount of time for the warehouse manager and distribute the process of ordering among the stores people. It would also minimize errors since everyone will be able to access a real-time updated database and order just as much as there is in stock. This can also lead to savings for the owner in terms of paid working hours and overtime since there is a high chance of the employees finishing their tasks earlier.

To supply a quick IT system, not only a good programming job has to be done but also a renewal of old hardware in both the warehouse and the stores is needed. The company has a long-time partner in a local PC- shop which we will discuss the hardware requirements with, and they will provide the computers with a loyalty discount for the drugstore.

1.2.2. E-sales, E-business, E-marketing

E-sales are not handled at all nowadays. It is a great idea to connect the database with an online intermediary such as *Heureka* here in Slovakia and allow them to sell products of this drugstore online. In the beginning, this may be done only locally, with a customer coming to the warehouse (or a store) to pick up his goods but can later be expanded well.

The other, more costly idea would be to create a so-called Services-oriented relationship-building web site. On this site, the products would not be sold online, but it would stimulate offline purchases and

build relationships with the customers. This site would provide customers with valuable detailed information about the products or ongoing discounts. Implementation of an E-newsletter on this site could rapidly increase the customers' awareness of Tevos. It would move the company from "Bricks and mortar to clicks and mortar" (Chaffey, 2011) enterprise, which is a must nowadays.

For the E-marketing, the suggestion is to use the SOSTAC™ framework (Smith, 2000) developed by Paul Smith (1999), where all the stages involved in marketing strategy, from its development to its implementation are summarized. This framework is an effective and recommended tool when planning E-marketing.

1.2.3. Expansion

Another huge opportunity – that comes with the highest cost – lies in the expansion. The company should undoubtedly continue to open new stores in Slovak towns, but there is more to it that can be done.

Namely, the most expensive and risky idea is to expand to the market of eastern Slovakia. This would come with huge investment into a brand-new warehouse (we suggest a town of Poprad or nearby, mostly because of its infrastructure and central-to-east location) together with hiring new employees and opening at least five stores at once in this part. The rough estimate calculated for this is a budget of DKK 7 000 000. However, since the lowest competition and the lowest income families are in eastern Slovakia, the cheapest drugstore could thrive in this region, and there is a high chance of return of investment.

Among the less expensive expansion opportunities, there is either a new, larger warehouse located nearby the current one where all the goods would fit, and there would also be some space left for future expansion of assortment.

We have also touched the idea of having a second warehouse in the western part of Slovakia, but since the logistics are now at a great level and there are stores in most major towns in these regions, we found this inefficient.

1.2.4. Motivation of workers

The motivation of the workers in the company is another huge issue. The workers do not agree to the every-month-rising overtimes, nor do they like their wages and the amount of work they do every day.

According to the equity theory (Adams, 1963), it is nearly impossible to determine what changes and rewards would satisfy the employees, since everyone has their criteria to meet.

Hiring at least one more warehouse worker and one driver would unburden the workers from the overtimes. This measure would not be as expensive as it looks, because the overtime would not be paid afterward. Also, providing them with rewards - at least a discount on the warehouse stock - could satisfy them in general.

1.2.5. Bullet-point summary

If we were to describe all the measurements briefly, we would suggest the company would undertake; it would look like this:

- Move to a larger warehouse
- Finish the transition to a new IT-system by the end of 2020
- Think of integration to a Heureka shop
- Continuously try to attract new customers
- Open new stores
- Keep a stable customer base; impose special discounts or customer cards
- When the pandemic season is over, and the company is doing OK, think about a bigger goal: entering a market of eastern Slovakia
- Despite the lack of money, try to mimic the competitors' advertising at least
- Think about hiring new employees instead of working overtime

1.3. Strategic alignment

The strategic alignment for the Tevos drugstore was derived from the SWOT analysis that was concluded before. The main goals in this plan include the new real-time updated database, more efficient workers, and the system that will serve the company for upcoming years no matter the level of expansion:

Plan	Goals/objectives	Relationship to the Project
2020 strategic plan for Tevos drugstore system	Real-time updated database	This project will provide all the stores with real-time information about the products and their stocks in the database
2020 strategic plan for Tevos drugstore system	Engage the workers and make them work more efficiently	This project will provide a valuable tool for the workers that will cut the time they spend working with the system, and they will have time to focus on other tasks
2020 strategic plan for Tevos drugstore system	A future proof system for expansion	If the company decides to expand in the future, it would not be difficult to add other warehouses, providers, products and stores to the system

Figure 10: Plan-Goal Table

1.4. Risks, Project Constraints

With every project, there are some issues, risks and constraints that can arise while working on it. That is why we tried to define the most probable ones that would disrupt our workflow into a table:

Risk	Probability	Impact	Priority	Solution
Data transfer issues	2	3	**6**	Try to understand the old system, take care when manipulating with it
Complex UX	2	2	**4**	Show mockups to the people who will work with the system the most, apply feedback
Stores people worried about the system	4	3	**12**	Create a user manual, be present when the transition happens, help them, answer questions
Requirement is imprecise	1	3	**3**	Keep the client updated, discuss with him after every iteration
Impact of COVID-19 on the project team, working remotely	5	2	**10**	Well-defined regular meetings, everyone has his task and deadline for it

Figure 11: Risk-Solution table

What the team is worried about the most are the stores' employees that have no IT background and would have to learn how to manage the new system. It must be ensured that these people will get their lectures on the new system so that they are well-prepared to manage the system on their own after the transition.

The other thing that may cause huge issues is the effect of COVID-19 pandemics on the project team. Since the project team cannot meet in person, everything is managed via online communication tools such as MS Teams, Messenger, or Skype. It will be more challenging to manage and finish everything only by talking to each other from separate places. However, the team is confident that this new experience will not strike badly on the overall quality of the project, and everything will be delivered functioning and on time.

The GUI that will be created has to be clear to navigate in and include as little straightforward steps as possible so that the occurrence of errors will be diminished.

The requirement was imprecise at first, but since now we understand what we are supposed to create in detail, we feel like this is not as big of an issue anymore.

1.5. Costs and benefits

1.5.1. Costs

The costs of the project will involve three essential things:

Item	Details	Price
Software	Five people will work for 100 days, 5 hours a day, 500 DKK / hour	1 250 000 DKK
Hardware	27 computers for the stores, 5 for the managers = 32 in total The new computers will cost 7000 DKK. There will be a need for a new computer for the database server, which will cost 10 000 DKK.	234 000 DKK
Education	The education of the employees on how to use the new system. Two years of support with the possibility of extension (up to 10 years)	50 000 DKK
==========		**TOTAL COST: 1 534 000 DKK**

Figure 12: Cost table

The estimated costs are therefore totaling at **1 534 000 DKK**, which seems reasonable compared to the budget of DKK 2 million. The project team will perform its best to deliver a well-performing solution within the agreed period.

1.5.2. Benefits

The benefits of the project are huge, some of them were previously mentioned, to sum them up:

- A better, quick, and responsive IT system with a real-time database information
- The orders will be done quicker with fewer errors occurring during the process
- The hardware would also be renewed, resulting in a smoother and less stressful work with the computer for the employees
- Better statistics information for the owner and the managers. With the new system, the statistical information would be so extensive, that the determination of the most bought products according to Pareto Law and the ABC analysis of the products would be possible

1.6. Competitor Comparison

The two more significant players in the drugstore market are the DM drugstore and the TETA drugstore. These are far bigger and have a higher amount of capital investment they can work with. These differences can be seen in numerous examples: namely TV advertising and the location of stores. This all is diminishing the reach of Tevos drugstore to the ordinary people since their shops are based in more remote parts of the town, and the level of advertising is deficient, as we mentioned before.

Despite that, we feel like Tevos could compete with these (and other) players for the market share, it just needs to do better marketing and advertising so that more people will get to know what Tevos drugstore is and why it is worth going in there. Therefore, Tevos drugstore should try its best to achieve the following in the upcoming year:

- creating an e-shop and setting up an e-shop oriented department in their company to handle e-shop orders. In Slovakia, many e-shops now supply products found in drug stores
- Think of ways of advertising the drugstore more. If there is no money for a TV spot, try to reach out to the public through radio spots or internet advertising
- Reach a milestone of 20 000 Facebook site likes, create at least five posts every week
- Redesign the website (*www.tevos.sk*), update the information, post the latest information about the discounts, create at least an online newsletter
- Motivate the store's employees to increase the sales, reward them if successful

1.7. Business Case Conclusion

To sum it up, the Tevos drugstore is in dire need of renovating various aspects of its business. Most importantly, the implementation of a new IT system to handle the orders is necessary; then, countless opportunities come with the E-sales, and the warehouse itself described in this business case. The system itself and the hardware can be both obtained within the budget of 2 million DKK, which was initially set by the owner. We recommend that the company’s management undertake this project.

2. Employee-Task-Goal Table

After the activity diagram showing the crucial tasks the employees would perform was created, it was decided to summarize all the tasks into an Employee-Task-Goal table, which helps to see the goal and the steps included in each task. This way, the whole task can be divided into smaller steps to get a better idea of what is needed to be thought of in terms of design and code.

Employee	Task	Goal	Steps in task
Warehouse manager	Read statistics	To see the statistics and change things accordingly	• Open statistics menu & • choose the statistic.
	Accept order	To see the stores' orders, be able to modify and accept them	• Open list of new orders • See the order and its weight • Edit the order
	Create a warehouse order	To refill the stock	• Select products and amounts that arrived at the warehouse along with the provider. • Save the warehouse order
	Overview stock	See products in the warehouse	• Open stock • Generate products list • Print
	Overview store stock	To see the number of products in the store	• Select the store from the list
	Manage stores	To add/edit/remove stores and their basic information	• Select the store from the list • Make necessary changes
	Manage products	To add/edit/remove products and information about them	• Select the product • Change the product

Store person	Create order	To create an order with information about the store and the warehouse	• Select product and amount • Confirm order
	Confirm order delivery	To confirm that the order has been delivered	• Check if all items are present in order, if some are missing, mark it in the system
	Report stock	To report stock amount to warehouse manager	• Count all stock in store • Submit them into the system

Figure 13: Employee-Task-Goal table

3. Use Cases

After finishing the *Employee - Task - Goal table*, the initial ideas for use cases started to show up. Since a use case is a very complex artifact and it provides a basis for the system design and later its architecture, it is crucially important to determine them and their priority well, so that the most complex ones are created as a fully dressed ones.

3.1. Use-Case Diagram

The phase continued with the artifact called a use case diagram, which shows the actors that will interact with the system and the tasks they will be able to perform. Since it was decided that neither warehouse workers nor warehouse drivers will interact with the system, warehouse manager and a store person were identified as our actors. All the tasks the actors will do were there written down:

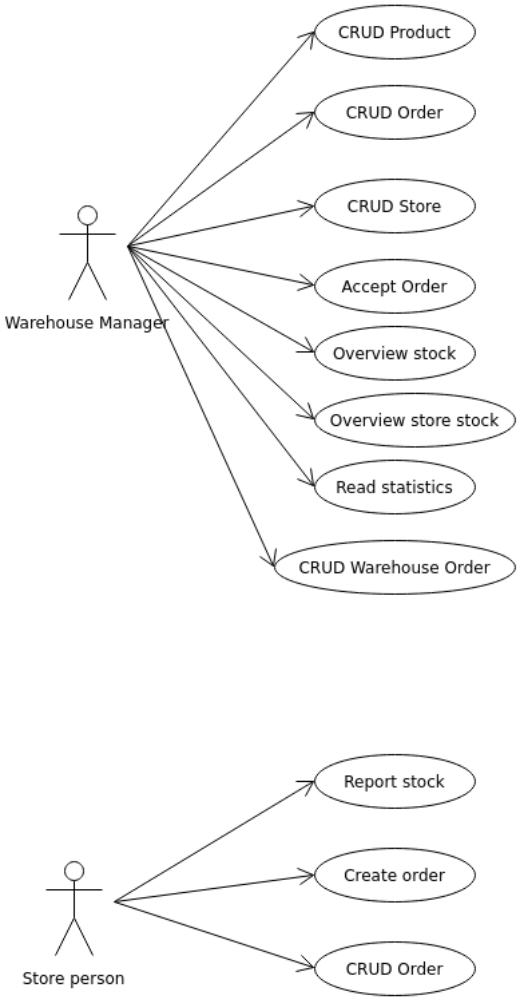


Figure 14: Use-case diagram

3.2. Use-Case Prioritization

It was decided to omit the CRUDs and make a prioritization table for only the non-CRUD use cases. Their architectural importance and business value were discussed; these two values were then multiplied to get their priority. The use case with the highest priority is “Create order” which will be designed as the fully dressed one and implemented in the very first iteration. The second highest is “Accept order”, which will be designed and implemented afterward.

Use case	Architectural importance	Business value	Priority
Create order	10	8	80
Accept order	6	10	60
Read statistics	6	9	54
Confirm order delivery	6	8	48
Overview stock	6	7	42
List orders	3	8	24
Report stock	2	7	14
Overview store stock	2	5	10

Figure 15: Use-Case priority table

3.3. Brief Use-Case Description

After the most critical use cases were discovered, there were other use cases left to be described. Since those were not as architecturally important, it was decided to just briefly describe them in words, so that the idea behind each use case is known:

List orders: The manager will be able to see the list of the orders and see their status. He will also be able to filter the orders according to their status.

Confirm order delivery: The store person will be able to load the order he made for the store and confirm its delivery to the store. There will be an option to report missing items in the order.

Overview stock: The manager will be able to see the stock of the warehouse in real-time.

Overview store stock: The manager will be able to see all stock reports of the specific store he chooses.

Read statistics: The manager will be able to look up statistics about the sales, stock, orders, and products.

Report stock: The store person will be able to report the current stock in the store. This serves for warehouse managers and statistics to have access to this information. This does not serve as a stock-taking. Stock-taking is handled by each store individually.

CRUD Product: The manager will be able to create a new product that will come into the warehouse, modify and delete it. This product will be saved into the database of products.

CRUD Store: The manager will be able to create a new store that will be opened, modify its information, and delete it when closed. This store will be saved in the database.

CRUD Order: The manager will be able to create a new order, read, modify, and reject it. Orders cannot be deleted, only rejected. This order is saved in the database.

CRUD Warehouse order: The manager will be able to create a new warehouse order that will come as supply into the warehouse, modify and reject it. Warehouse orders cannot be deleted, only rejected. This warehouse order, together with items and revisions in it, will be saved into the database.

4. Activity Diagram

The activity diagram was created to show the workflow when performing the most critical tasks in the company, creating and accepting an order. It helps to visualize how the system would function with a detailed step-by-step procedure.

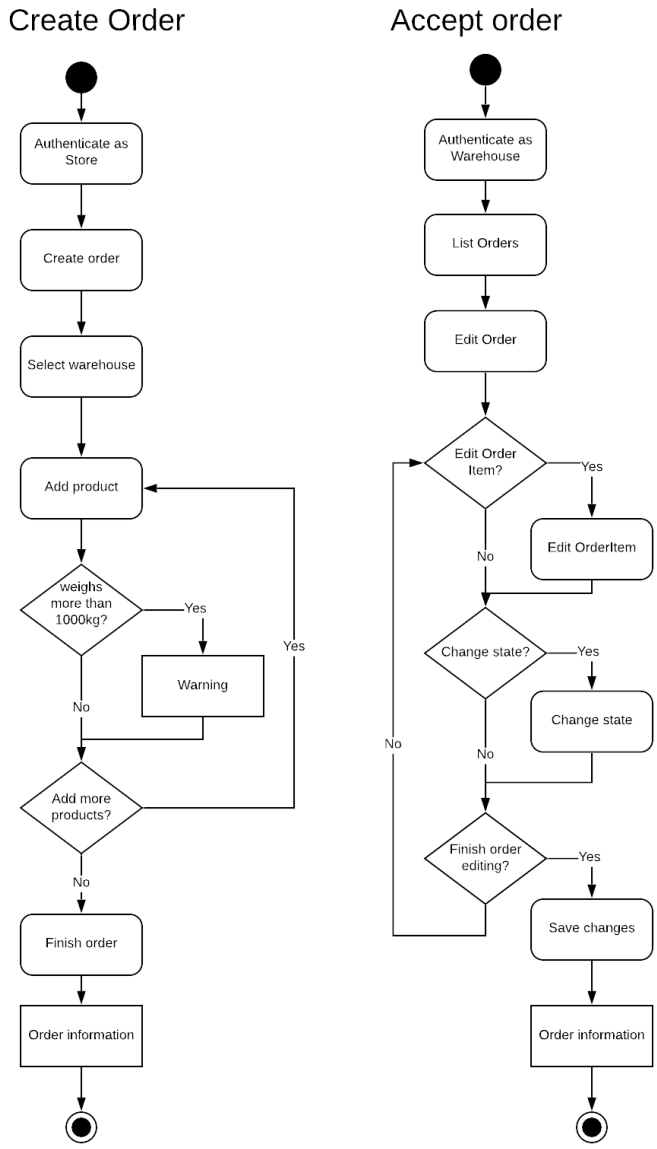


Figure 16: Activity Diagram

5. System Vision

A system vision is a summary of what was discovered during the preliminary study phase, which gives a vision of system scope, features, and goals to be achieved.

5.1. Scope

The scope of this project is to create warehouse management and ordering system that would be easy to work with and would satisfy the needs of a company. The real-time system updates would be dealt with properly since there will be many events when more users would want to access the system at the same time.

When the system is finished, it would save much time to the users and provide the managers with statistical information about the company.

5.2. Stakeholders

The stakeholders that are interested in the successful completion of this project are the company owner and employees, the project group, and its supervisor, Dimitrios Kondylis.

5.3. Main Product Features

To show the understanding of the crucial concepts of the system, it must fulfill, the product features table was created, where the highest priority features of the system were listed. The ones worth mentioning are the real-time database info, sufficient database, or different login features, not to forget the CRUD for Order and an Item itself, all being the highest priority.

Feature	Description	Priority
F1	The system must show real-time info from the database	high
F2	The system can CRUD info about an Order	high
F3	The system can CRUD info about a Product	high
F4	The system should have a sufficient database	high
F5	The system must be quick, responsive, error- preventative	high
F6	The system should have distinctive features based on login	high
F7	The system must handle all the calculations well	high
F8	The system must inform the warehouse manager if the order exceeds 1 ton	medium

Figure 17: Feature-Priority table

5.4. System requirements

Requirements in the Unified Process are categorized according to the famous FURPS+ model, where each letter stands for a word that characterizes a category of the requirement. In this project, the requirements identified and named according to this model are:

Functional: Functional requirements are written all over the report in the form of use cases, the highest priority ones are creating order by a store person and having a real-time warehouse database

Usability: The system has to be easy to use for the store people who might not be extremely skilled in working with computers.

Reliability: the system must be error prone as much as possible, everything must be added and altered in the database correctly in each process to ensure having a similar number of stocks and money in every calculation

Performance: The system should respond quickly to inputs from the user.

Supportability: The system should be scalable for more stores and warehouses that are there to come and easily maintainable. High cohesion and low coupling should be ensured in each part of the code.

5.5. Mock-Ups

We created some of the main screens of the software as mockups for customer feedback and to help us visualize the actual software before we start coding it. We used the popular tool Adobe XD (Adobe Inc., n.d.)

First, the user of the system must log in using a simple login screen, which is shown in Figure 8. We decided it would make the system more user friendly and flexible if we did not ask if the user logging in is working in a store or a warehouse. To achieve this, the user type and the main menu they see after login is decided based on the user's role.

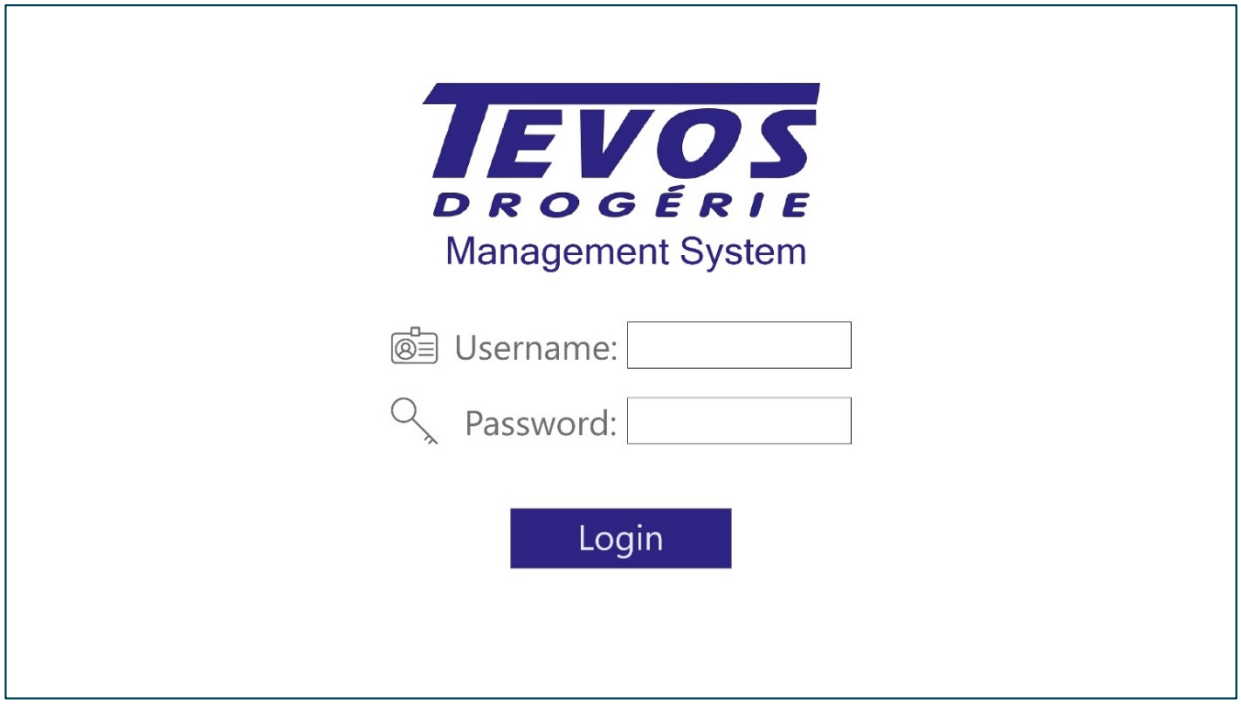


Figure 18: 'Login screen' Mock-up

After logging in, based on the user's role, they see one of the two dashboard screens shown in Figure 9 and Figure 10. From here, they can make new orders by clicking on the 'Orders' button or hovering over it and clicking 'New order' from a drop-down menu (store manager) and view/process already existing orders via clicking on the said button (warehouse manager).

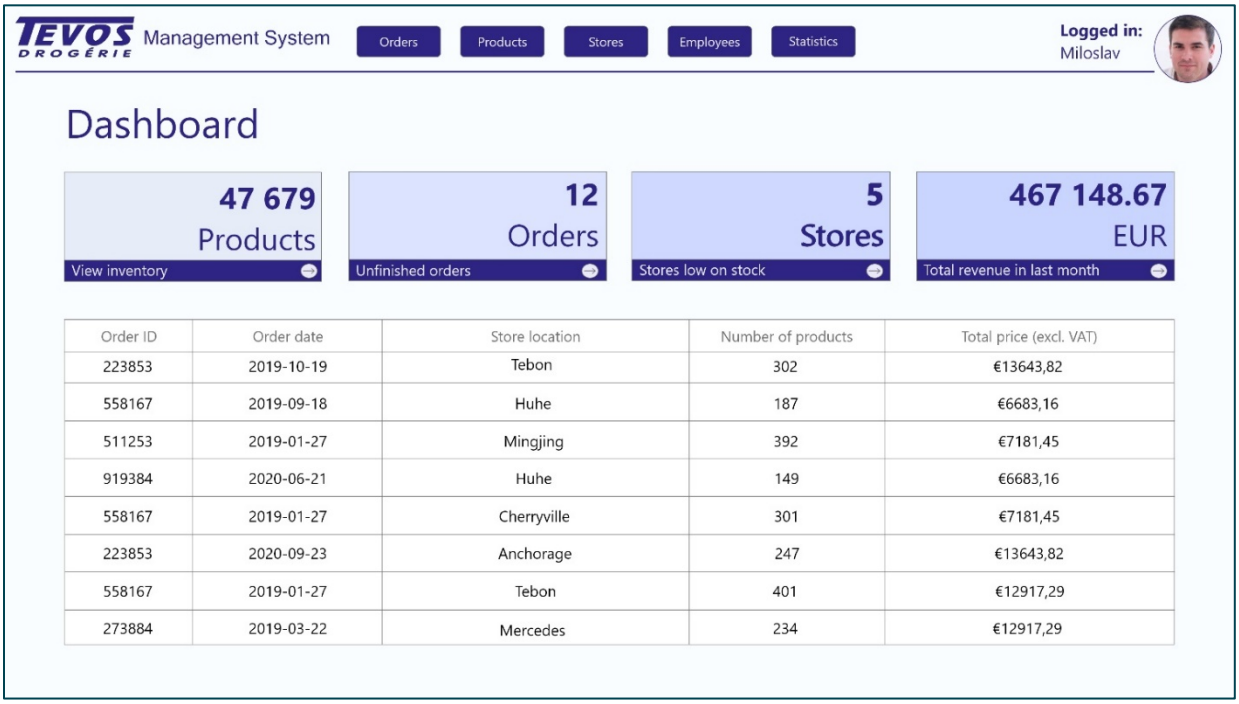


Figure 19: Warehouse manager's Dashboard

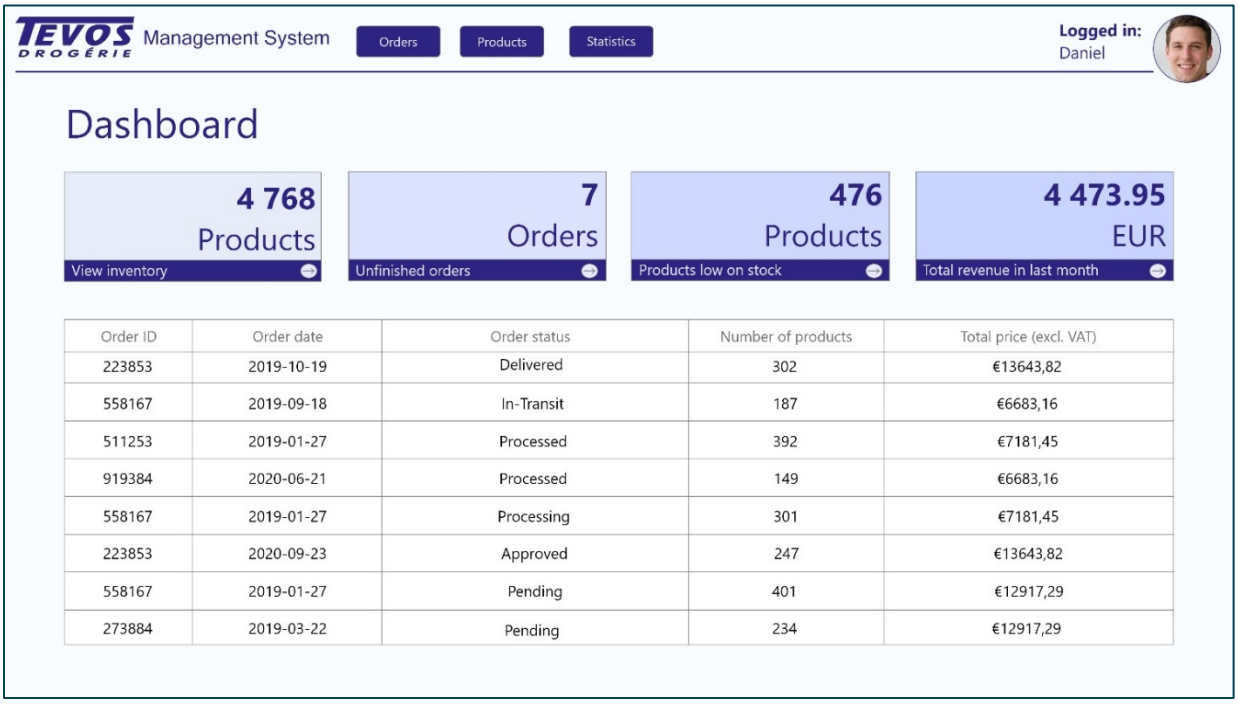


Figure 20: Store manager' Dashboard

Part IV. Elaboration

In this part of the report, the two most important use cases are described. Those being “Create order” and “Accept order”, the complete fully dressed use cases together with their artifacts such as system sequence diagram or operation contract are listed in this part. The tests performed on these use cases are documented as well. Designing of the domain- and relational models, together with creating SQL scripts, are explained at the end of this part.

1. Fully Dressed Use-Case 1: Create order

Use case	**Create order**	
Actor	Store person	
Pre-condition	A store person is logged in and can connect to the warehouse database and create an order accordingly	
Post-condition	An order is created, saved in the database	
Frequency of occurrence	Ten times a day (all the stores together)	
Flow of Events	**Actor**	**System**
Main success scenario	1. The user selects options to create order and chooses the warehouse	2. List of available products shown
	3. The user selects the product and enters the amount.	4. Product is added to the order
	Steps 3-4 are repeated until all the products are added	
	5. After selecting products, the user submits the order	6. Order is created in the database and information about the order are shown
Alternate flows	2. List of products is not shown 1. Products to choose from are not displayed 2. The user creates an order again	
	3. The amount inserted is more than there is in stock 1. Error message displayed 2. User must add the product again with a valid amount	
	5. The weight of an order is more than 1000 Kg 1. Warning displayed	
Special requirements	User cannot choose more products than there are in the stock	
	The order has its weight limit, and the user should be notified when it is exceeded, but the order can still be created	

The very first fully dressed use case was “Create order”: A store person creating a supply order for the store he is in, which would be handled by the warehouse. Its fully dressed form is documented in the table above.

1.1. System Sequence Diagram and Operation Contract

The very first artifact that was produced straight after the fully dressed form was an SSD. According to our UML textbook, an SSD shows the external actors that interact directly with the system, the system (as a black box), and the systems that the actors generate (Larman, 2004). A system sequence diagram for Create order use case is below:

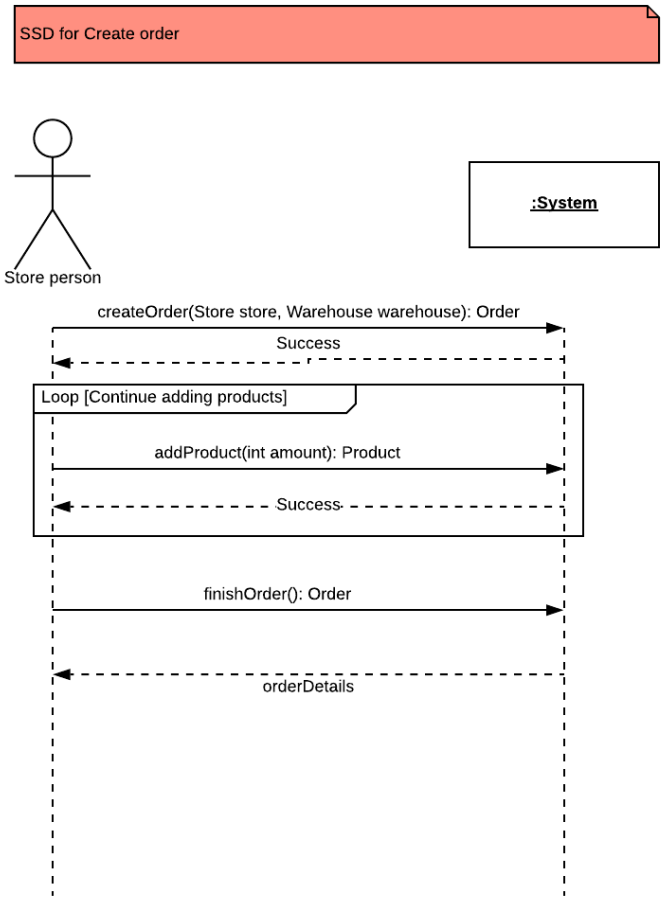


Figure 21: System Sequence Diagram for Create Order

The operation contract shows every single method from the SSD, its cross-reference, and pre- and post-conditions. This diagram was created to clarify what was meant by which method in the SSD and what will each method result in.

In the SSD, we can see that the order is creating when two parameters – a store that creates the order and a warehouse that will supply the order – are sent. Then there is a loop in which multiple products may be added to the order. After all the products were added to the order, the Order is finished, and the order details are returned.

Then an operation contract for this use case was created. The operation contract essentially provides us with a more precise and detailed system behavior. It does so by using pre- and post-conditions, and cross-references to describe the system operation. The operation contract for Create order use case can be seen below.

Operation contracts for Create order
Operation: createOrder(Store store, Warehouse warehouse) Cross-reference: use case Create order Pre-condition: User is logged in Post-condition: Order o was created, associated with a store and a warehouse
Operation: addProduct(int amount) Cross-reference: use case Create order Pre-condition: Order exists and the product is found Post condition: Product p was added to the Order o
Operation: finishOrder() Cross-reference: use case Create order Pre-condition: Order exists and has products Post condition: Order o was finished, database was updated, order info is printed

Figure 22: Operation Contracts for Create Order

1.2. Interaction Diagram

The term interaction diagram is a generalization for both the communication diagram and the sequence diagram. Communication diagrams are in general less complicated and there are more of them created, resulting in easy-to-read diagrams. In contrast, in the sequence diagram, we can combine more method calls and objects in the same diagram, creating a more complex but only one diagram for the use case. After all, as it is just a matter of preference, here it was decided to use a sequence diagram:

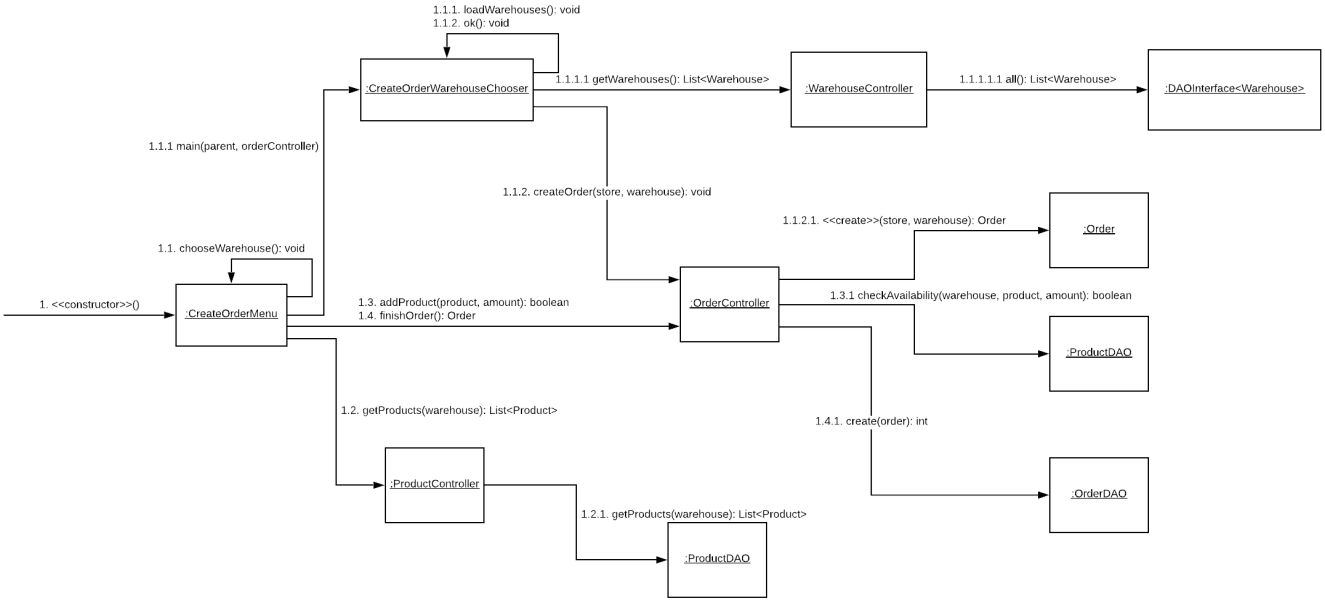


Figure 23: Sequence Diagram for Create Order

1.3. Tests for Create Order

Testing is an essential part of the development process. The functionality of the most crucial use cases must be tested in order to ensure proper implementation. Successful tests will also permit to continue with the implementation of the 2nd use case. At first, a scenario matrix table was created based on the scenarios identified in the fully dressed form:

ID	Scenario	Initial flow	Alternative flow
S1	Successful order creation	Main success scenario	n/a
S2	Product list not shown	Main success scenario	A2
S3	The amount is too high	Main success scenario	A3
S4	Weight limit reached	Main success scenario	A5

Figure 24: Test Scenario table for Create Order

We then proceeded to the test identification table, which shows the "steps" taken for the use case to be completed and their validation: **V** for valid, **I** for invalid, and **n/a** if not applicable. The table is shown below:

Test case ID	Scenario	Warehouse added to the Order	Order created	Show product list	Product added with the valid amount	Order weight less than 1000 Kg	Order finished and created	Expected result
T1	S1	V	V	V	V	V	V	Order successfully created
T2	S2	V	V	I	n/a	n/a	n/a	Error message
T3	S3	V	V	V	I	n/a	n/a	Error message
T4	S4	V	V	V	V	I	V	Warning message

Figure 25: Test Identification table for Create Order

After the identification table was created, we proceeded to perform manual tests of the listed tests. The actual input values (which were used later while testing the software) were recorded into the Test case data table below, and the result was described:

Test case ID	Scenario	Warehouse added to the Order	Order created	Show product list	Product added with the valid amount	Order weight less than 1000 Kg	Order finished and created	Expected result
T1	S1	Warehouse 1	Order 1	"Shampoo", "Washing gel 3Kg"	"Shampoo" x 5 "Washing gel 3Kg" x 3	12 Kg	Order 1	Order successfully created
T2	S2	Warehouse 1	Order 2	n/a	n/a	n/a	n/a	Error message
T3	S3	Warehouse 1	Order 3	"Shampoo"	"Shampoo" x 999999	n/a	n/a	Error message
T4	S4	Warehouse 1	Order 4	"washing gel 3Kg"	"Washing gel 3Kg" x 500	1500 Kg	Order 4, warning displayed	Warning message

Figure 26: Test Data table for Create Order

2. Fully Dressed Use-Case 2: Accept order

The second fully dressed use case was “Accept order”: A warehouse manager opens the menu to see all the orders, then checks them and eventually refines them (or let them in the same state) and accepts them, so that they are ready to be prepared by the warehouse workers and supplied to the store by the drivers. Its fully dressed form is documented in the table below:

Use case	**Accept order**	
Actor	Warehouse manager	
Pre-condition	A store created order and order is not yet accepted	
Post-condition	An order is accepted, warehouse information updated	
Frequency of occurrence	Ten times a day	
Flow of Events	**Actor**	**System**
Main success scenario	1. User opens ‘accept order’ menu	2. Information about the order is shown, including products, price and store info
	3. User alters order items if needed	4. System updates the order
	5. User sets the state of the order	6. Order is updated
	Steps 3-5 are repeated until the order is in the desired state	
Alternate flows	2. The order was not retrieved from the database 1. User is notified, asked to select the order again	
	4. The order item is wrongly altered 1. The system notifies the user, asks him to alter the item again	
Special requirements	The order has its weight limit, and the user should be notified when it is exceeded, but the order can still be accepted.	
	This process will happen multiple times during the order lifecycle until the state is set to complete.	
	If the order state is set to “PENDING”, stock info is updated	

2.1. System Sequence Diagram and Operation Contract

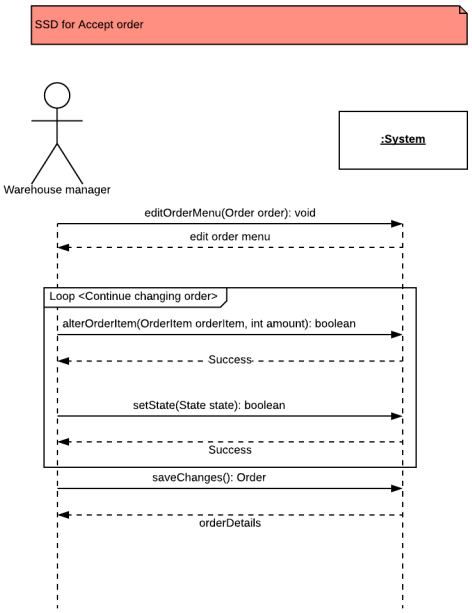


Figure 27: System Sequence Diagram for Accept Order

The operation contract was once again created with both the pre- and post-conditions to provide a clear idea about the use case and the flow of events in it.

Both the SSD and the operation contract were produced for this use case as well.

In this SSD, we can see that at first, the menu where the warehouse manager can alter the Order is opened, passing in the order as a parameter. Then the manager can change the ordered items as well as their amount. After this process is finished and the manager is satisfied with the order, he changes the state of the order, and the changes are then saved, order details returned.

Operation contracts for Accept order
Operation: editOrder(Order order) Cross-reference: use case Accept order Pre-condition: Warehouse manager is logged in Post-condition: edit order menu is shown
Operation: alterOrderItem(OrderItem orderItem, int amount) Cross-reference: use case Accept order Pre-condition: Order exists and the order item to be edited is found in the order Post condition: order item oi was edited in the Order o
Operation: setState(State state) Cross-reference: use case Accept order Pre-condition: Order exists Post condition: Order state was changed
Operation: saveChanges(Order order) Cross-reference: use case Accept order Pre-condition: Order exists and changes were made Post condition: Order was changed, order revision was created, database was updated and order info is printed

Figure 28: Operation Contract for Accept Order

2.2. Interaction Diagram

Same as with the first use case, also in the second one, it was decided to create one big sequence diagram showing all the necessary interactions with different layers:

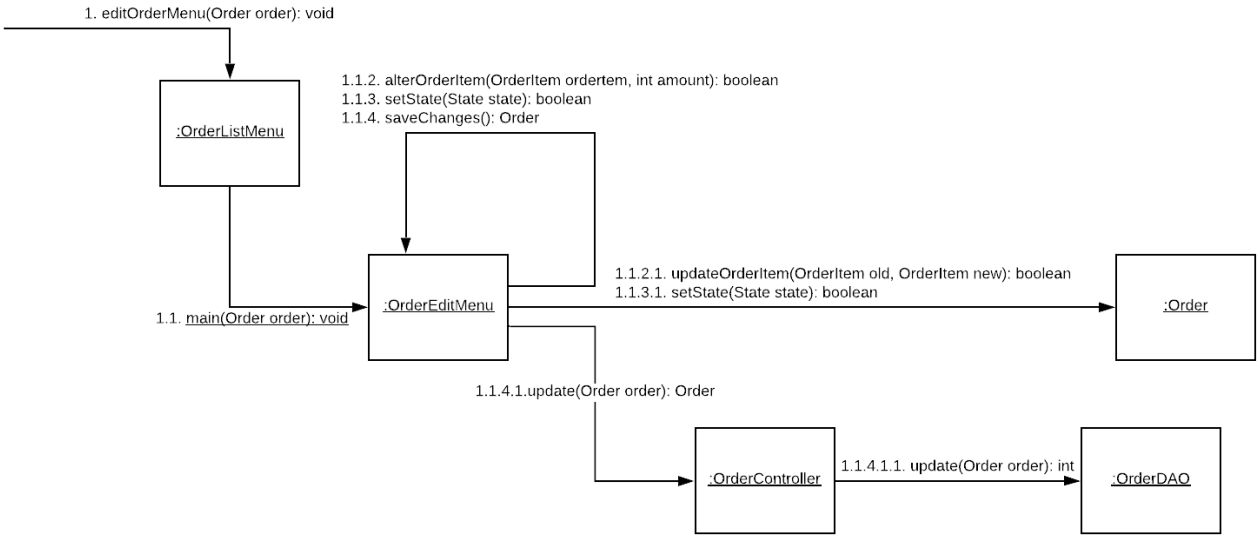


Figure 29: Interaction Diagram for Accept Order

2.3. Tests for Accept Order

The same approach to testing was used for the 2nd main use case. First of all, the scenario matrix was created:

ID	Scenario	Initial flow	Alternative flow
S1	Order accepted successfully	Main success scenario	n/a
S2	Order not retrieved	Main success scenario	A2
S3	Order Item altered wrongly	Main success scenario	A3

Figure 30: Test Scenario table for Accept Order

The test identification table was created afterward:

Test case ID	Scenario	Retrieve order	Alter order	Save altered order	Expected result
T1	S1	V	V	V	Order successfully created
T2	S2	I	n/a	n/a	Error message
T3	S3	V	I	n/a	Error message

Figure 31: Test Identification table for Accept Order

This was followed by manually testing the use case; all the test data and results were recorded in the Test case data table:

Test case ID	Scenario	Retrieve order	Alter order	Save altered order	Expected result
T1	S1	Order 1	"Shampoo", x 10	Altered order 1	Order successfully created
T2	S2	n/a	n/a	n/a	Error message
T3	S3	Order 3	"Shampoo", x 1000	n/a	Error message

Figure 32: Test Data table for Accept Order

2.4. Domain Model for the most important use cases

Since the most important use cases use exactly the same model classes, it was decided to put only one Domain model of the Elaboration phase into the report. It can be seen below, including every class up to the end of the 2nd iteration of the Elaboration phase – "accept order" Fully Dressed Use-Case:

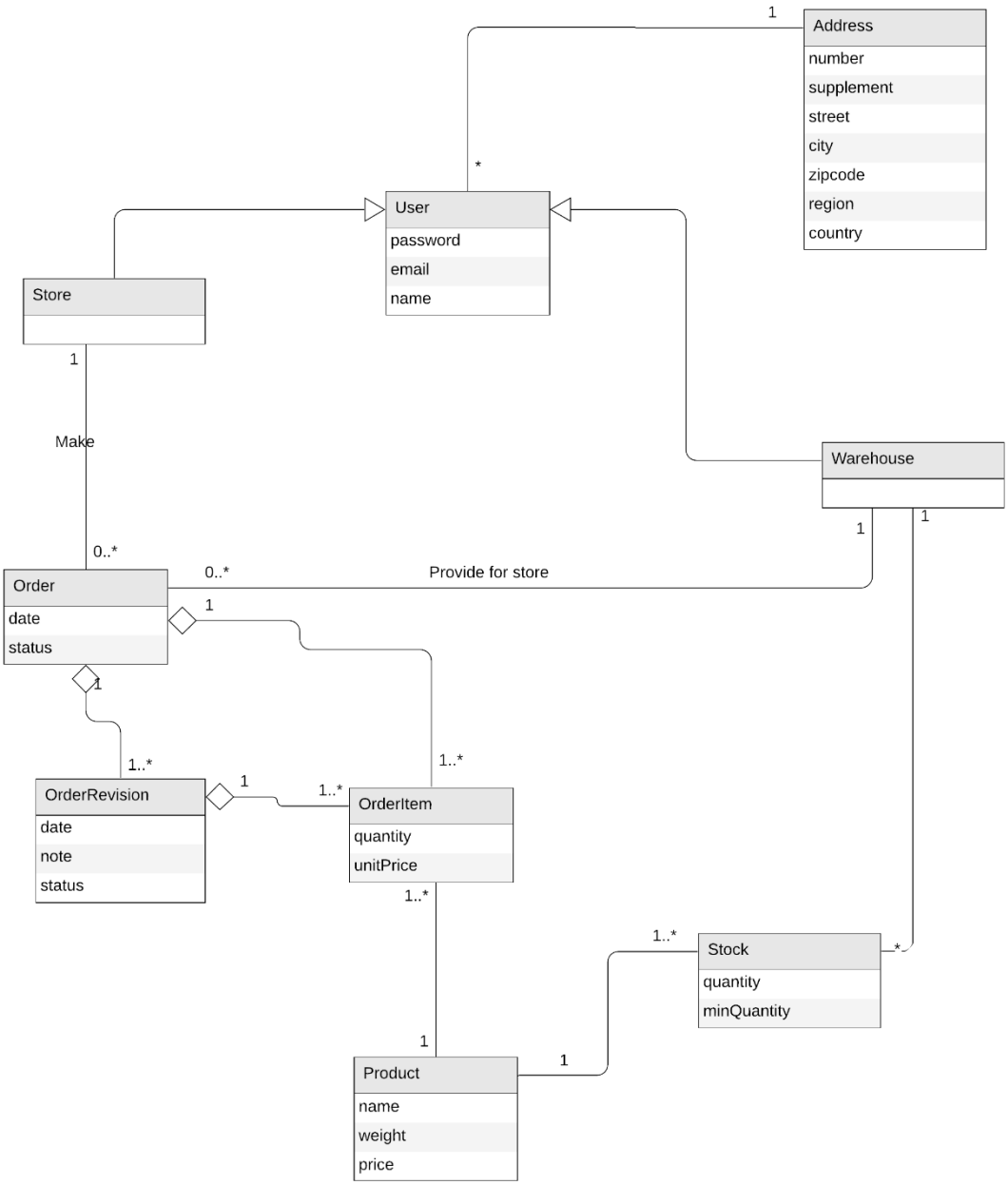


Figure 33: Domain Model for Create Order

3. Domain Model for the whole system

The domain model is the most critical artifact in the whole initial analysis. Having identified the conceptual classes first, below is the domain model for the system. It was created after the fully dressed forms of use cases were finished and before the other artifacts were created, even though it is placed underneath them in the report to ensure better readability. It shows the relations between the classes, their functions, and their attributes.

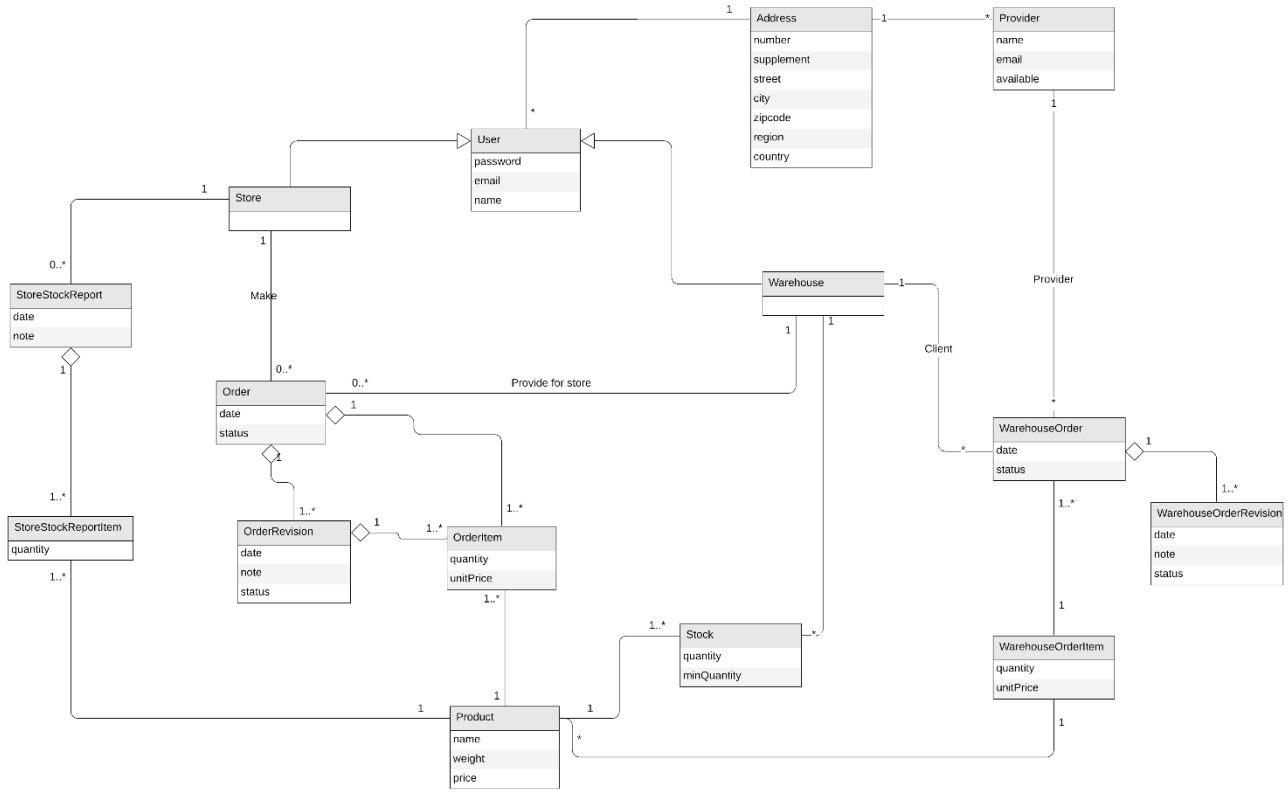


Figure 34: Domain model

In this model, the inheritance is shown in a way that "Store" and "Warehouse" are subclasses of the "User" class. All the attributes are inherited from the "User" class to the two kinds of users our system has. The only difference between them is the name of the class and the things they can do while logged in to the system.

The *status* field for the "Order", "OrderRevision", and "WarehouseOrderRevision" classes is an Enumerator, which is not shown on this diagram. We decided to store the history of status changes for both kinds of orders, and that is what these classes help us to do so.

Since orders have status, they do not fit the collaboration pattern. The program is designed to be monitored a few times in a day; it does not need instant notification. It does not match the State Pattern either. The states are defined by a single variable. It does not define the behavior of the entity itself but the way it will be handled by others.

The product has its weight attribute in order to ensure calculations of the weight of the whole order that was one of the features to be implemented; if the order exceeds a weight of 1 ton, there will be a notification shown on the screen.

Since there may be multiple providers and they can change quite often, the “available” boolean attribute will ensure this information is kept in the database.

The stock will be decremented after the order status is set to “Approved”. And if approved order is rejected, the stock will be incremented. The stock will also change if any items change in the Order. OrderRevision will keep track of these changes along with status changes. This will give users nice overview about the lifecycle of the Order.

4. Relational Model

Having finished the Domain Model (*Figure 34*), we created a relational model for the system based on the Domain Model and mapped to a database (Microsoft SQL Server 2019 in our case). Since much time was invested in the creation of a good domain model, the development of a relational model was easier afterwards, because no significant obstacles arose.

We did our best to split possible multivalued attributes (Address) and minimized null values successfully. A result is a relational model in third normal form:

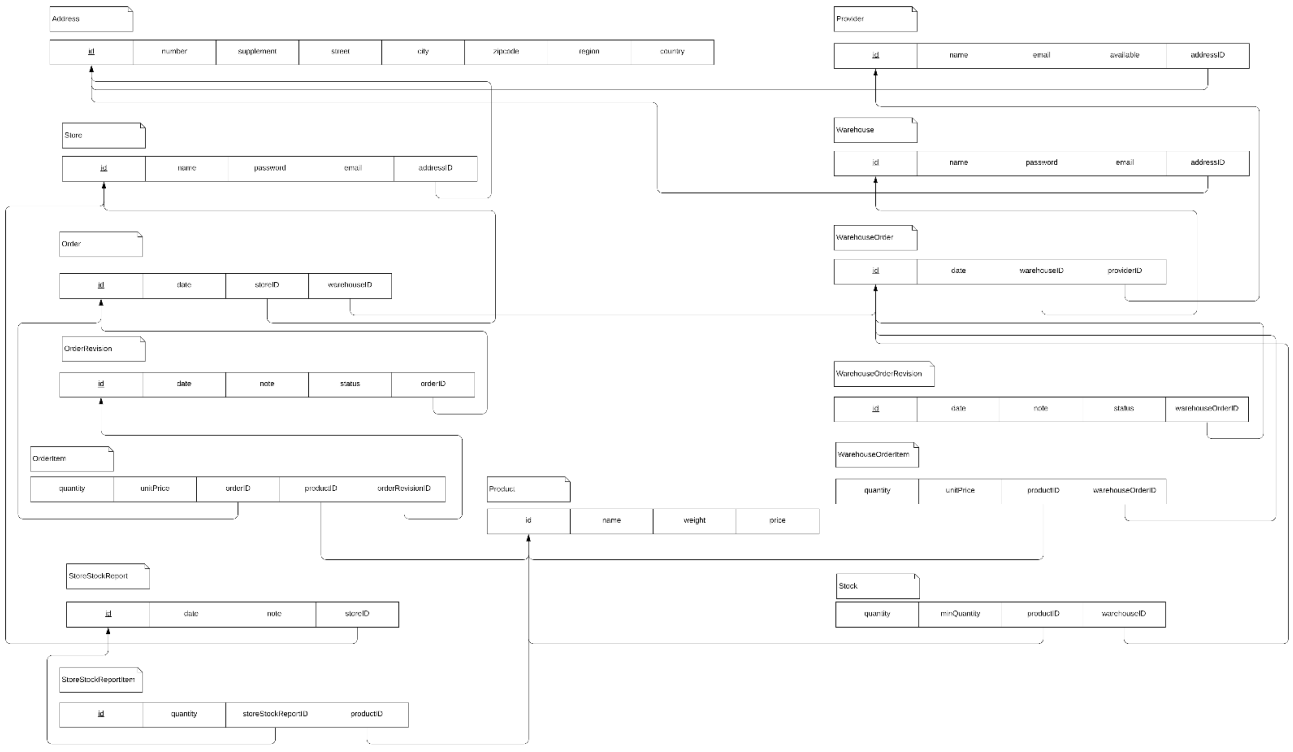


Figure 35: Relational Model

A more easily readable version of this relational model can be found in *Appendix E*

5. Database & SQL Script

Since relations between tables and specific tuples may be created using relational databases, this is a good practice to reduce redundancies. The tables may even be cross-referenced and joined

For this project, we used a Microsoft SQL Server 2019 (Express edition) database system, which is provided to us by UCN (University College of Northern Denmark).

```
-- Create Product table with constraints
CREATE TABLE [Product]
(
 [id]     int NOT NULL ,
 [name]   varchar(120) NOT NULL ,
 [weight] real NOT NULL ,
 [price]  money NOT NULL ,


 CONSTRAINT [PK_Product] PRIMARY KEY CLUSTERED ([id] ASC)
);
GO

-- Create Stock table with constraints
CREATE TABLE [Stock]
(
 [warehouseID] int NOT NULL ,
 [productID]   int NOT NULL ,
 [quantity]    int NOT NULL ,
 [minQuantity] int NOT NULL ,


 CONSTRAINT [PK_Stock] PRIMARY KEY CLUSTERED ([warehouseID] ASC, [productID] ASC),
 CONSTRAINT [FK_108] FOREIGN KEY ([warehouseID])  REFERENCES [Warehouse]([id]) ON DELETE CASCADE ON UPDATE CASCADE,
 CONSTRAINT [FK_111] FOREIGN KEY ([productID])  REFERENCES [Product]([id]) ON DELETE CASCADE ON UPDATE CASCADE
);
GO
```

Figure 36: Part of the 'Table Creation' SQL Script

As can be seen in *Figure 36*, we wrote an SQL script that creates all the tables and sets up the table constraints (primary keys and foreign keys), which ensures data consistency. Most of the foreign keys are set to cascade on update and deletion so that they would not contain any rows with invalid references.

Next, we created a short and simple script (*Figure 34*), which can be used to 'reset' the database. This script drops data in the database and drops every table, so it must be used with caution.

```
DECLARE @Sql NVARCHAR(500) DECLARE @Cursor CURSOR

SET @Cursor = CURSOR FAST_FORWARD FOR
    SELECT DISTINCT sql = 'ALTER TABLE [' + tc2.TABLE_SCHEMA + '].[' +  tc2.TABLE_NAME + '] DROP [' + rc1.CONSTRAINT_NAME + '];'
    FROM INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS rc1
        LEFT JOIN INFORMATION_SCHEMA.TABLE_CONSTRAINTS tc2 ON tc2.CONSTRAINT_NAME =rc1.CONSTRAINT_NAME

OPEN @Cursor FETCH NEXT FROM @Cursor INTO @Sql

WHILE (@@FETCH_STATUS = 0)
    BEGIN
        Exec sp_executesql @Sql
        FETCH NEXT FROM @Cursor INTO @Sql
    END

CLOSE @Cursor DEALLOCATE @Cursor
GO

EXEC sp_MSforeachtable 'DROP TABLE ?'
GO
```

Figure 37: The 'Database Reset' SQL Script

After this, there had to be some sample data inserted into the database to make running tests possible. In *Figure 35*, a part of this script can be seen. This script inserts a few rows into several different tables so the database can be tested. It is essential that this script is not used in 'production' by the customer, but only during the development for testing purposes.

```
insert into Store (name, email, password, addressID)
values ('store', 'store@mail.com', 'password', 1);
insert into Warehouse (name, email, password, addressID)
values ('warehouse', 'warehouse@mail.com', 'password', 2);
GO;

insert into Product (name, weight, price) VALUES ('Deodorant', 0.12, 1.20);
insert into Product (name, weight, price) VALUES ('Shampoo', 0.2, 2.50);
insert into Product (name, weight, price) VALUES ('Washing gel', 1, 3);
insert into Product (name, weight, price) values ('Washing powder', 5, 2);
insert into Product (name, weight, price) values ('Dishwaser tablets', 0.5, 5);
GO;


insert into Stock (warehouseID, productID, quantity, minQuantity) VALUES (1, 1, 500, 100);
insert into Stock (warehouseID, productID, quantity, minQuantity) VALUES (1, 2, 100, 150);
insert into Stock (warehouseID, productID, quantity, minQuantity) VALUES (1, 3, 50, 49);
insert into Stock (warehouseID, productID, quantity, minQuantity) VALUES (1, 4, 300, 200);
insert into Stock (warehouseID, productID, quantity, minQuantity) VALUES (1, 5, 200, 150);
GO;

insert into [Order] (storeID, warehouseID, status, date) values (1, 1, 'PENDING', '2012-06-18T10:34:09');
insert into [Order] (storeID, warehouseID, status, date) values (1, 1, 'PENDING', '2012-06-18T10:34:09');
insert into OrderItem (orderID, quantity, unitPrice, productID, orderRevisionID) VALUES (1, 5, 2, 4, NULL);
insert into OrderItem (orderID, quantity, unitPrice, productID, orderRevisionID) VALUES (2, 4, 5, 5, NULL);
insert into OrderRevision (orderID, status, date, note)
VALUES (1, 'PENDING', '2012-06-18T10:34:09', 'Bring it fast');
insert into OrderRevision (orderID, status, date, note)
VALUES (2, 'PENDING', '2012-06-18T10:34:09', 'I do not care when you bring it');
insert into OrderItem (orderID, quantity, unitPrice, productID, orderRevisionID) VALUES (NULL, 5, 2, 4, 1);
insert into OrderItem (orderID, quantity, unitPrice, productID, orderRevisionID) VALUES (null, 4, 5, 5, 2);
GO;
```

Figure 38: Part of the 'Data insertion' SQL Script

6. Design Class Diagram

6.1. Design Class Diagram for Create Order

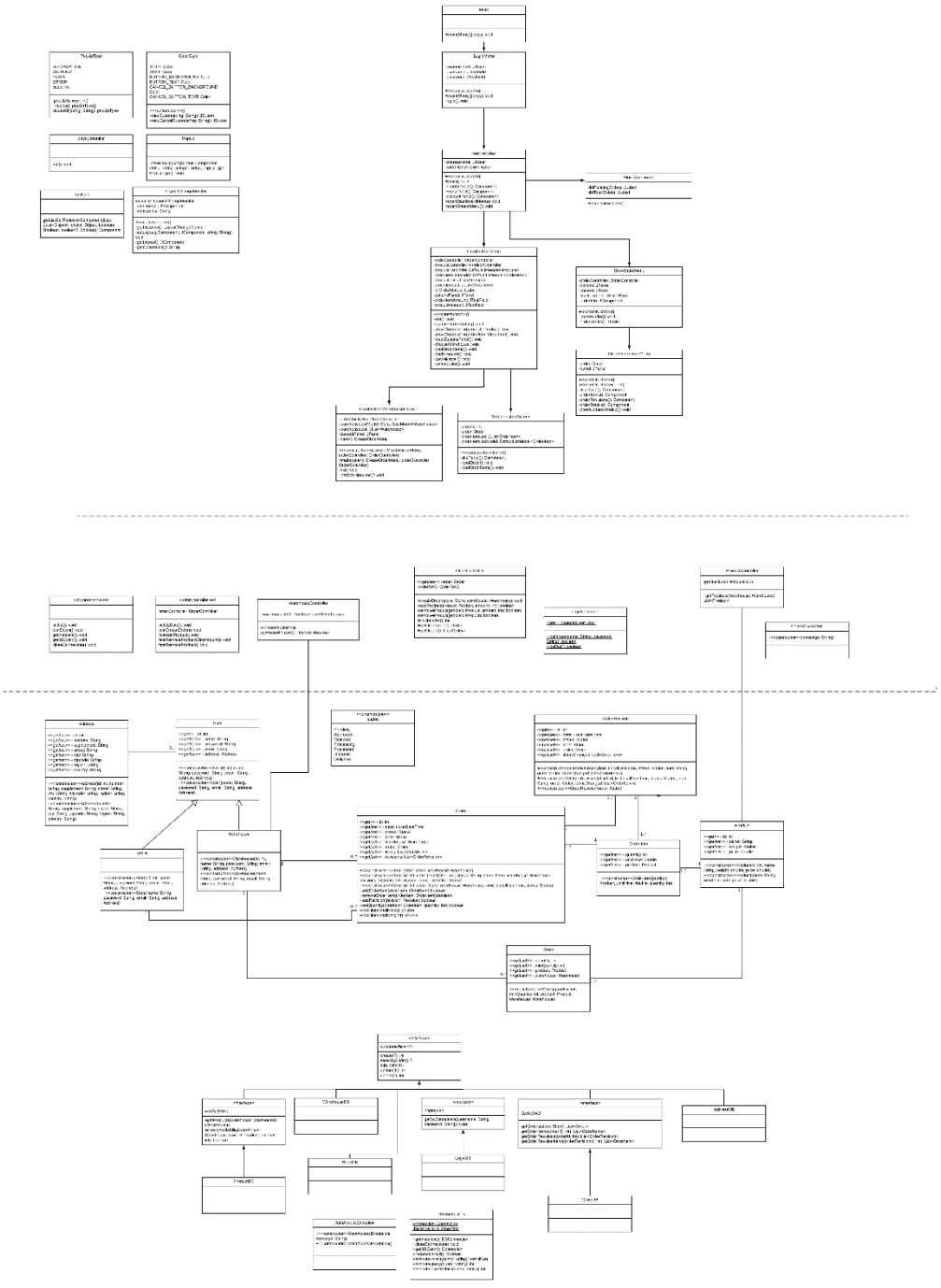


Figure 39: Design Class Diagram for Create Order

6.2. Design Class Diagram for Accept Order

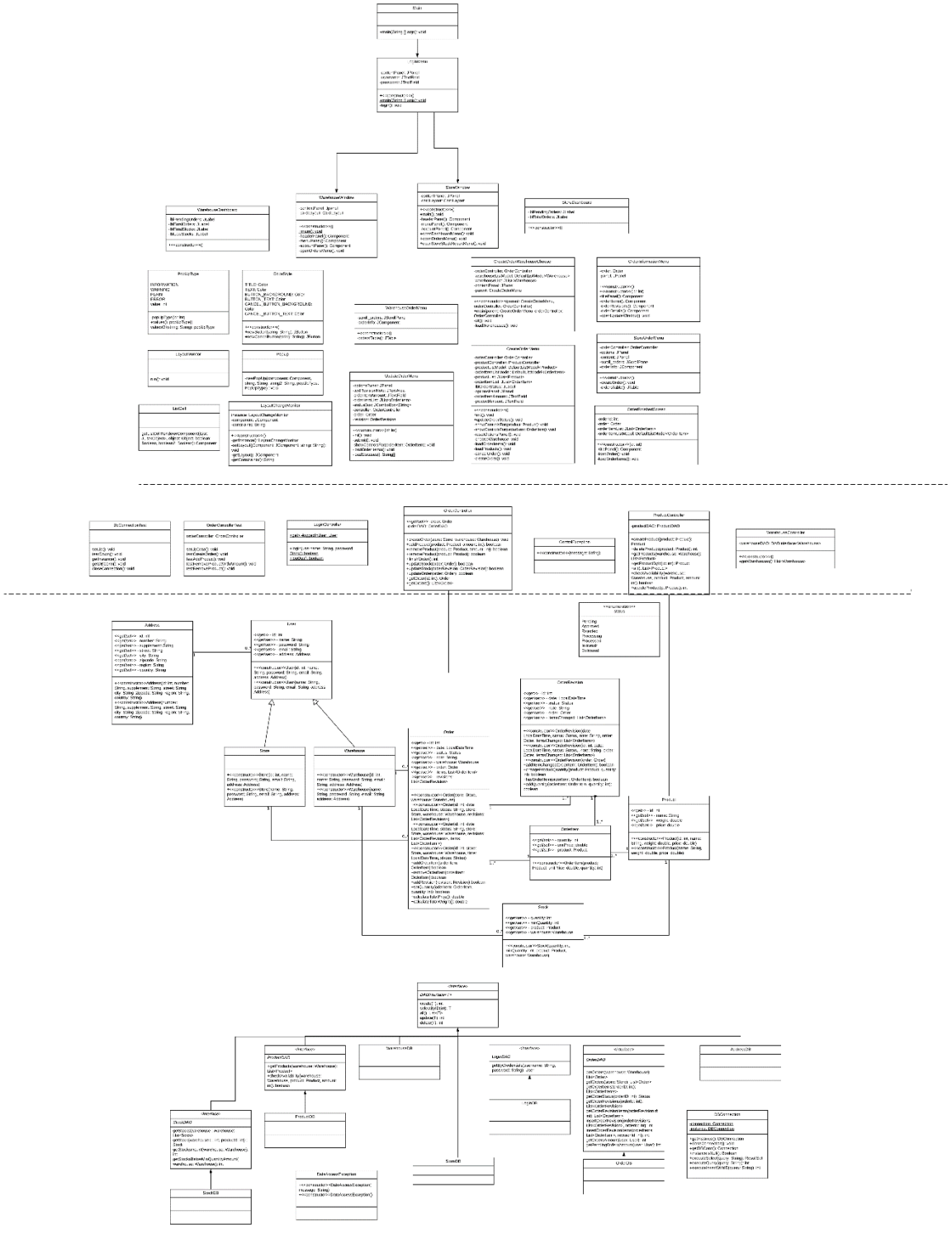
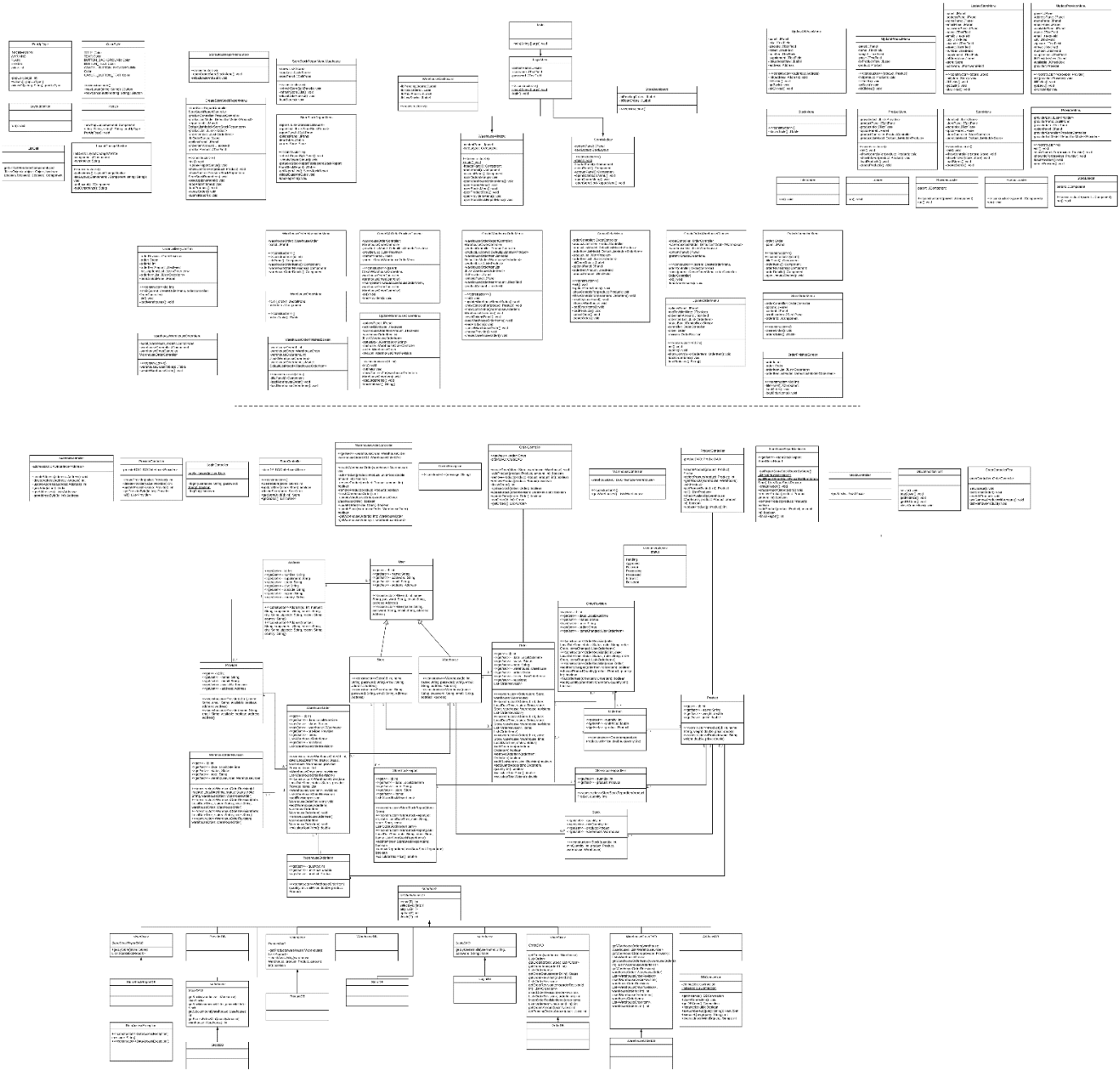


Figure 40: Design Class Diagram for Accept Order

6.3. Design Class Diagram for the whole system

The Design Class Diagram of the whole system in Figure 39 shows classes with their attributes, methods and constructors as well as the relations between them. The whole design class diagram and the classes inside could not have been connected with lines due to the robustness of the system. This diagram shows the state of the system at the end of the project (after the Construction phase).



Please refer to *Appendix D* for a larger version of this diagram.

Part V. Construction

1. Code Standards & Architecture

1.1. Code Standards

The code conventions are crucial when a team is developing the same project. In order to maintain some programming practices, we wrote the following rules for this project, which are based on the *Java Code Conventions*:

1. **Variables, methods**
    a. Class variables should be private; usage of getters and setters is important in order to access and modify them easily
    b. Variables and method names should follow the Java standard, which says that:
        i. All regular variables (both public and private) should be in camelCase
        ii. Constant variables (with final modifier) should be in UPPERCASE
        iii. Generic type parameter names should be UPPERCASE single letters
        iv. Methods should always be verbs in camelCase
    c. Each method should be responsible for only one task
2. **Classes**
    a. Class names should be nouns in PascalCase
    b. Class names should be short but meaningful
    c. Each class is to have only one purpose. If it fulfills more purposes, it should be split into more classes
3. **Version Control**
    a. Everything should be regularly committed and uploaded to the SVN server provided by UCN
    b. Every commit should only contain changes to a couple of files and should be only about one change
    c. Each commit must have a descriptive commit message
4. **Documentation**
    a. Each class and all its methods should be documented inside the .java files, according to the Javadoc conventions
    b. The documentation should contain an overview of what the given class/method does and how to use it

1.2. '3-Layer Architecture'

"Three-tier architecture is a client-server software architecture pattern in which the user interface (presentation), functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms." (Eckerson, 1995)

Having a Graphical User Interface (GUI) layer that is separated by the logic with a Controller layer is a great way to structure the system, protect the layers so that they communicate only one level up/down and avoid errors. This approach also ensures that the "high cohesion, low coupling" design standard is maintained, and that the system is very easily scalable and alterable. Database connection in this project will be built based on the DAO pattern (explained below) using a publicly available JDBC connector library.

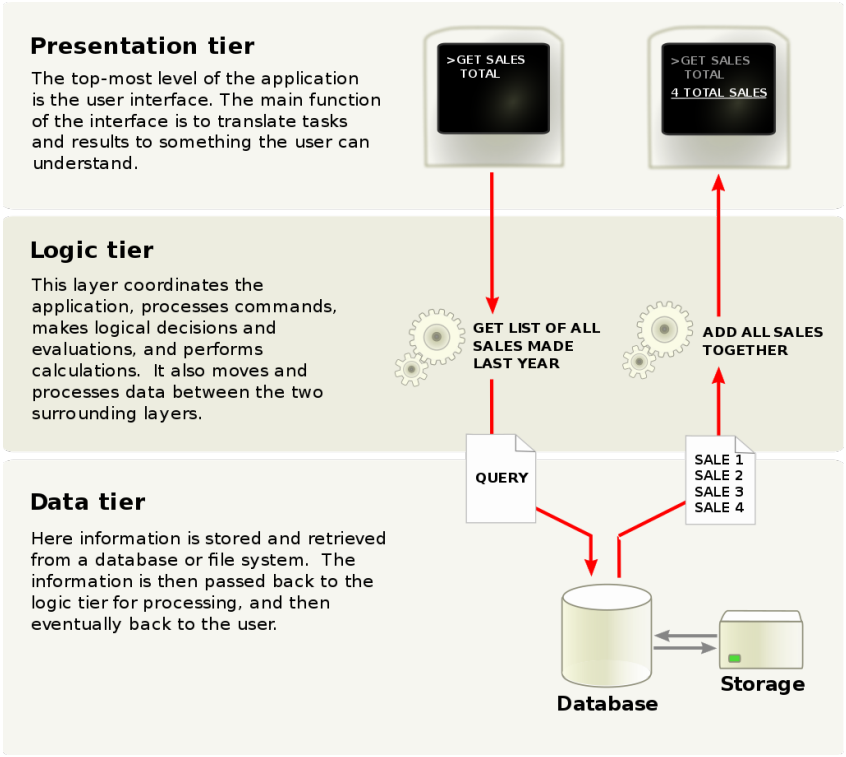


Figure 41:Overview of a Three-tier application (User:Bartledan, 2009)

1.3. Design Choices

1.3.1. Method stubs

Method stubs are a handy tool when coding a huge project; at first, just a method signature is created, and its return type is defined (void, boolean, int, etc.) just to ensure that the method will not be forgotten. The code functions even without the logic implemented within the method as it simply returns either null or -1 value, indicating that the method is not yet implemented. As the use cases progressed, method stubs were replaced with actual methods with logic.

1.3.2. Lambdas

Lambdas are primarily used to shorten the code and make it more readable and clearer. The greatest use of lambdas in this project occurred in the GUI layer, where methods with action listeners were shortened significantly. Here are a few examples:

```
JButton btnCreateOrder = ColorStyle.newButton("Create Order");
options.add(btnCreateOrder);
btnCreateOrder.addActionListener(actionEvent -> createOrder());
```

Create order button in StoreOrderMenu.

1.3.3. Threads

In our GUI layer, threads are frequently used to load large amounts of data from the database. In classes where this happens, we created inner classes that extend the Thread class. We initialize them at the end of the GUI class constructor and start them. This allows us to display the GUI before data is loaded, so users will not have to wait after pressing the button. A prime example of this is the WarehouseOrderMenu class, which shows orders from all Stores for specific a Warehouse.

Here is the constructor of WarehouseOrderMenu class:

```
/*
 * Create the panel.
 */
public WarehouseOrderMenu() {
    // Set layout
    GridLayout gridLayout = new GridLayout(1, 2);
    gridLayout.setHgap(10);
    setLayout(gridLayout);

    // List of orders
    scrollOrders = new JScrollPane();
    add(scrollOrders);

    orderInfo = new OrderInformationMenu();
    add(orderInfo);

    new Loader().start();
}
```

Figure 42: Constructor for the WarehouseOrderMenu class

This constructor is simple and contains only layout, scroll pane for orders and order info pane for displaying chosen order information. This will execute and show to user very quickly, but without any data. Here the inner class Loader is started.

This is what Loader class looks like:

```
 39      * Create table and fetch data
 40      */
 41     private JTable ordersTable() {
137
138     private class Loader extends Thread {
139         @Override
140         public void run() {
141             JTable table = ordersTable();
142             EventQueue.invokeLater(() -> scrollOrders.setViewportView(table));
143         }
144     }
```

Figure 43: The inner Loader class inside the WarehouseOrderMenu

The ordersTable() method retrieves all the orders related to Warehouse and creates table from them. That table is then returned. In Loader we call this method to load table into memory and then using

EventQueue.invokeLater we make sure it gets displayed on GUI. This is only one of many examples where we use multithreading to load data, while making sure the GUI is responsive. This approach is used in multiple GUI classes.

At first, we didn't use Threads and EventQueue while creating our GUI classes and when we created multiple Orders, the GUI became unresponsive and Orders card would take longer than one second to load and this was unacceptable.

1.3.4. Streams

Usage of streams was recommended by our programming teacher to display our knowledge about them. There is only one usage of Stream in our program, as we didn't find any more specific usages for it. It's in ProviderController class, method getAvailableProviders(). This Stream replaced a need for new SQL query which would retrieve available providers, and this would require new Provider DAO interface.

```
public List<Provider> getAvailableProviders() throws DataAccessException {
    List<Provider> all = providerDAO.all();
    Stream<Provider> providers = all.stream();
    return providers.filter(Provider::isAvailable).collect(Collectors.toList());
}
```

Figure 44: ProviderController.getAvailableProviders() method

1.3.5. Monitors

In this project we implemented only one monitor, but it has a crucial role in our approach how we handle GUI changes. It's LayoutChangeMonitor class. This class contains two synchronized methods and is a Singleton class (Larman, 2004). It also contains two fields representing JComponent, its name and getter for name, since JComponent is returned in getLayout method.

Here are the synchronized methods:

```
    public synchronized void setLayout(JComponent component, String constrains) {
        this.component = component;
        this.constrains = constrains;
        notifyAll();
    }

    public synchronized JComponent getLayout() {
        try {
            wait();
        } catch (InterruptedException e) {
            return this.component;
        }
        return this.component;
    }
```

Figure 45: The synchronized methods in the LayoutChangeMonitor class

Method setLayout takes parameters and saves their value into fields and calls notifyAll, which interrupts wait method called in getLayout method and then getLayout will return the actual component.

1.3.6. GUI

After the first semester project GUI experience, where we had separate window for each action, we decided to make everything in one window, with exception of dialogs. We achieved this by using only three JFrames, which only one can run at the time per instance of application. These three are LoginMenu, StoreWindow and WarehouseWindow. LoginMenu is the first window that appears and after submitting valid credential it will open the window based on user type.

Both StoreWindow and WarehouseWindow contain an inner class LayoutMonitor, which extends Thread and contains infinite loop. This is the code of the LayoutMonitor:

```
    private class LayoutMonitor extends Thread {
        @Override
        public void run() {
            LayoutChangeMonitor monitor = LayoutChangeMonitor.getInstance();
            while (true) {
                JComponent component = monitor.getLayout();
                String componentName = monitor.getConstrains();
                contentPanel.add(component, componentName);
                cardLayout.show(contentPanel, componentName);
            }
        }
    }
}
```

Figure 46: LayoutMonitor inner class inside StoreWindow and WarehouseWindow

LayoutMonitor will call the getLayout method on LayoutChangeMonitor. This method as mentioned earlier contains wait method call, which waits for any LayoutChangeMonitor notify method call. This Thread is started once inside the constructor of Window and runs all the time.

The combination of LayoutChangeMonitor and LayoutMonitor inner classes ensures that any GUI can be displayed from any place inside the code.

```
    // Start layout monitor
    new LayoutMonitor().start();
}
```

Figure 47: new LayoutMonitor().start()

```
private void createOrder() {
    CreateOrderMenu component = new CreateOrderMenu();
    LayoutChangeMonitor.getInstance().setLayout(component, "create_order_menu");
}
```

Figure 48: Method createOrder() in class StoreOrderMenu

```
public void createStore() {
    LayoutChangeMonitor.getInstance().setLayout(new UpdateStoreMenu(null), "update_store");
}
```

Figure 49: Method createStore in class StoreMenu

1.3.7. DAO

All the classes interacting with database (DB classes) are implementing DAOInterface. This one ensures that basic interaction with the database is possible. In case we need a special interaction, which lead to the creation of a new function (like in StoreStockReportDB), a specific DAO interface is created (like StoreStockReportDAO). This ensures us a good track of all interactions implemented. (like StoreStockReportDAO). This ensures us a good track of all interactions implemented.

```
public interface StoreStockReportDAO extends DAOInterface<StoreStockReport> {
    List<StoreStockReport> getByStore(Store store) throws DataAccessException;
}
```

Figure 51: StoreStockReportDAO extending the original DAOInterface

```
public class StoreStockReportDB implements StoreStockReportDAO {
    DBConnection db = DBConnection.getInstance();
```

Figure 52: StoreStockReportDB implementing its own DAO Interface

1.4. Plans for next two iterations of construction phase

1.4.1. Optimizing tables of results

Orders table in GUI is the main one that needs a bit more optimization. Currently it's already optimized to a point, where the data loads on another thread and GUI loads before the data is loaded.

Optimization of the table would lie in implementing filters, sorting options and most importantly a pager to load and show only a limited amount of data. This would drastically improve the performance and load on database servers.

1.4.2. Transactions

Implementing transactions was planned for other construction phases, so we have a working system at first. ACID, or Atomicity, Consistency, Isolation and Durability is the basic set of properties that database transactions must follow in order for the data to be consistent and to guarantee data validity even in the event of a critical error (such as power failures, etc.). Transactions are very important for our system to ensure data consistency.

We failed to identify them higher in our priority list and thus we ended up not implementing them. This was the biggest mistake we made during this project.

1.4.3. Statistics

We are aware that the statistics that are currently implemented in the system are just a placeholder for more complex ones, that would be displayed on the dashboard. Those would be derived from business case and consulted with the company.

1.4.4. Better UI

Our current UI works and is usable, but it will need some polishing in the future.

Part VI. Transition

1. Handing the finished software to the customer

In the Unified Process, the Transition phase is the shortest and simplest one as its main objectives are finetuning of the project and "deploying" the system for real-world usage. This point was not fully reached in the report, since two more Construction phases would need to be executed in order to implement all the desired functionalities with the best accuracy possible.

2. User Manual

To ease the change from manual work to a new IT system, we wrote a simple user manual. It includes every detail from how to start the program and log in to creating or accepting an order (which is the main and most important use case of the system).

See *Appendix D* for the full user manual.

Part VII. Conclusion

The conclusion part should be mostly focused on evaluating the project, looking back at the successes and failures of the project, summarizing what has been done and what to expect in the future. Below is the summary and the group's perspective of the project, then there is a polemic about the initial problem statement questions finished by our outlook on the future.

1. Problem statement

At the very beginning, a problem statement was written, involving three main questions to be solved with this project. The group's opinion, whether it was successful or not can be found below:

Q1: Are we able to design a brand-new system for the company that would make the process of ordering more effective?

A1: We strongly believe that this was the crucial thing to achieve in this project. We focused mostly on delivering a system, which is as user-friendly as possible with a simplistic, but sufficient design for Tevos company. We are also convinced that we succeeded in doing so.

Q2: Are we able to create such databases for the drugstore, that the stock-taking and other information could be easily retrieved?

A2: The time we spent designing the domain model firstly, and then the relational model proved to be worth it, the final result is impressive; orders can be easily made by the store, the warehouse manager can alter them according to the needs and capacities of the warehouse and everything is then successfully recorded in particular databases.

Q3: Are we able to ensure the smoothness, faultlessness, and user-friendliness of the new system?

A3: We are satisfied with the result in terms of user-friendliness and smoothness. However, the database was not populated with much information, nor was the processes so frequent and massive as they could be in the future; in the end, this is the part of the project we identified as a possible blind spot and a thing to be careful about in the future. We also identified some candidate functionalities to implement before the actual deployment of the system would happen that are not part of the current system. Despite that, we believe that there will not be any significant troubles with the smoothness and effectiveness of the system.

2. Future vision

The implemented system is expected to serve the company well for the upcoming years since its scalability was already thought of and later implemented into the system; new stores and a possible second warehouse would be easily implemented, the same goes for various new products that will come to the warehouse.

The statistics are valuable tool nowadays and will improve processes in the company and increase the revenue. We definitely see a potential to implement more of them in the future.

As mentioned above, the system was not tested for extremely huge and massive processes. However, we firmly believe that with this implementation and some minor tweaks like paging and filtering of the lists, the system is going to be sufficient for the company for years to come.

3. Group's reflection

The group is satisfied with the result, although it could have been better – namely, the time constraint was tight and since the group did not manage the time well, there was a rush in the last days. We could blame this on the COVID-19 crisis, but it ultimately comes to self-discipline of individual members. The call attendance wasn't always 100% and it was frustrating for other members during the call. This must serve as a lesson for the future work of the group members. Otherwise, the group members have familiarized themselves with the Unified Process and various concepts the UP contains, improved their knowledge of system design, coding in Java, and database management as well. The most important features of the project – the two main use cases – were implemented well, but in the end, the group lacked a few weeks that would provide them time to implement all the minor features the group has outlined, but weren't really clear in use-case diagram with exception of statistics. What we didn't reach were iterations 2 and 3 of construction phase, in which we planned to work on fine-tuning the UI, making huge lists like orders filterable, and lastly implementing database transactions so ensure consistency of the data. We are aware, that these features are important for the system, but due to our prioritization and time scheduling alongside some missed milestones, we had to move other milestones further.

4. Group processes

This part evaluates the work within the group. A group contract can be seen here as well. Group work, group processes, and handling of the situations that appeared during the development are described below.

4.1. Group Contract

We based our group contract on two sample group contracts we found online (College of Natural Sciences, University of Texas Austin) and (The Griffin Tate Group, 2002). All group members agreed and will stick to this contract the best they can during the project.

The entire group contract can be read in *Appendix B*, but the main points are as follows:

1. Each member agrees to show up to every scheduled meeting on time.
2. If a member is late, he may join, but he must catch up on what he missed.
3. Members agree to treat one another with respect.
4. Only agree to do work that we are qualified and capable of doing.
5. Keep other team members informed.

4.2. Evaluation of Group Work

4.2.1. Initial research of information

Since the company we chose to create a project is located in Slovakia and Marek's father is one of the managers in the company, it was mostly him and Erik that interviewed the manager and the owner – the fact that the managers do not speak English very well was the reason that only two members performed the interview directly, the other group members read the notes and prepared questions for the second interview.

All the information for the report and project were found in the study materials on Canvas and in the books related to each course. If there was something extra to resolve, the internet was used to lookup information, which is then credited appropriately in Appendix A (Literature List & References).

4.2.2. Collaboration

The collaboration was at a very decent level for the duration of the project. There were, of course, some days when not everyone participated in the meetings, but this was expected since the project lasted for a long time. The group was formed just before this project, so this was the first experience, and the members had to get to know each other at the start. Despite not having as much time as the group would wish for, we tried our best to utilize our time as well as we could.

There were no significant disputes or problems within the group; the minor ones were always solved by discussion and explanation of possible solutions.

Every group member has had a part-time job as well as personal problems at some point, so we cannot blame someone for not participating fully on the project for the whole duration of it.

It was especially tough for Remi, who came for an Erasmus exchange semester and could not familiarize himself with all the concepts and study style at UCN because of the ongoing COVID-19 pandemic. However, he did a great job helping the group as much as he could and was a valuable member of the group.

4.2.3. Leadership

According to the Insights personality tests, there was no "red" leader in this group. During the feasibility study phase, Marek has shown himself to be the one in charge, mostly because he was the one that knew the company at the start. As the time progressed, Krisztián proved himself to be the tech-nerd who managed all the data and documents for the group in an organized manner and always came with a tool to ease our work, e.g., Microsoft Planner or SharePoint.

Erik was the natural leader during the development process since he was the most experienced of us all and had a clear system vision idea from the beginning.

Appendices

Appendix A. Literature List & References

Adams, J. S. (1963). Towards an understanding of inequity. *The Journal of Abnormal and Social Psychology, 67*(5), 422-436.

Adobe Inc. (n.d.). *UI/UX design and collaboration tool | Adobe XD*. Retrieved April 22, 2020, from https://www.adobe.com/products/xd.html

Bass, B. M., & Ryterband, E. C. (1979). *Organizational Psychology* (2nd ed.). London: Allyn & Bacon.

Brooks, I. (2018). *Organisational Behaviour: Individuals, Groups and Organisation* (5th ed.). Pearson.

Chaffey, D. (2011). *E-Business and E-Commerce Management - Strategy, Implementation and Practice* (4th ed.). Prentice Hall.

Cole, B. M. (2019, January 10). *Innovate Or Die: How A Lack Of Innovation Can Cause Business Failure*. Retrieved from Forbes: https://www.forbes.com/sites/biancamillercole/2019/01/10/innovate-or-die-how-a-lack-of-innovation-can-cause-business-failure/#2cc55d622fcb

College of Natural Sciences, University of Texas Austin. (n.d.). *Sample Group Contract.* Retrieved March 5, 2020, from https://cns.utexas.edu/images/CNS/TIDES/teaching-portal/Examplegroupcontract.pdf

Eckerson, W. W. (1995). Three Tier Client/Server Architecture: Achieving Scalability, Performance and Efficiency in Client Server Applications. *Open Information Systems, X*(1), 20.

Freeman, R. E., Harrison, J. S., Wicks, A. C., Parman, B. L., & Colle, S. D. (2010). *Stakeholder Theory: The State of the Art.* Cambridge: Cambridge University Press.

Johnson, G., Scholes, K., & Whittington, R. (2005). *Exploring Corporate Strategy: Text & Cases* (7th ed.). Oxford: Financial TImes Press.

Kotter. (2019, April). *8 Steps to accelerate change in your organization.* Retrieved April 14, 2020, from https://www.kotterinc.com/wp-content/uploads/2019/04/8-Steps-eBook-Kotter-2018.pdf

Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development.* Prentice Hall.

Larson, E. W., & Gray, C. (2010). *Project Management - The Managerial Process* (5th ed.). McGraw Hill.

Miles, R. E., Snow, C. C., Meyer, A. D., & Coleman, Jr., H. J. (n.d.). Organizational Strategy, Structure, and Process. *The Academy of Management Review, 3*(3). Retrieved April 30, 2020, from https://www.jstor.org/stable/257544

Pareto, V. (1980). *Manual of Political Economy.* (A. S. Schwier, A. N. Page, Eds., & A. S. Schwier, Trans.)

Porter, M. E. (1979, May). How Competitive Forces Shape Strategy. *Harvard Business Review, 57*(2), 137-145.

Smith, P. (2000). *Marketing Communications: An Integrated Approach* (2nd ed.). Kogan Page.

TatvaSoft. (2015, April 15). *Top 12 Software Development Methodologies & its Advantages / Disadvantages.* Retrieved April 30, 2020, from TatvaSoft: https://www.tatvasoft.com/blog/top-12-software-development-methodologies-and-its-advantages-disadvantages/

The Griffin Tate Group. (2002). *Sample Team Contract.* Retrieved April 22, 2020, from http://www.pmtraining.com.tw/member_pmp/Team%20Contract%202.0.pdf

User:Bartledan, W. (2009, Februrary 17). *File:Overview of a three-tier application vectorVersion.svg.* Retrieved from Wikipedia: https://commons.wikimedia.org/wiki/File:Overview_of_a_three-tier_application_vectorVersion.svg

Appendix B. Customer Interview

Interview questions

- Define the usual day and the workflow within the company
- Identify all the employees and their function within the company
- What do you see as major drawbacks of the current system?
- What is your vision of the future system?
- What is the business strategy for the upcoming years?
- What about the competition in the market, how do you fight it?
- Is there anything you provide to the people that other competitors lack?
- Could you identify any weaknesses of your company?
- What about the marketing, do you have Facebook/Instagram/ website? How often do you post there?
- How do you manage the warehouse layout?
- How do you manage the supply logistics for the stores? Are there any specially calculated routes? Who decides about them?

Interview notes

- The officer calls 10 drugstores around the country daily and asks them what they want to get to their store, he writes it into the outdated system and a paper invoice is printed, given to 5 warehouse workers who get the order ready, load the order into the car and the driver then delivers it to the store. The store's salesperson then checks whether everything is ok and if so, she calls the officer in the warehouse that it is fine.
- The problem with the system is that the salespeople in stores does not see the updated central database and they are unable to make an order themselves.
- There is 1 owner, Sales representative and 27 salespeople in stores, IT & admin manager, Warehouse director, warehouse manager, 5 workers and 3 drivers.
- The competition is rough with 2 greater players in the market. The fact that large supermarkets have drugstore equipment in their portfolios does not help the situation.
- They still manage to get the lowest prices of them all, having a stable customer base.
- Last year 4 new stores were opened and 1 closed, totaling to 27 stores as of now and one warehouse.

- For their future plans they aim mostly to opening new stores, but they are aware, that a new larger warehouse would be needed. They moved to this one 10 years ago from a smaller one and its capacity is getting insufficient. The plans are now halted due to the coronavirus pandemics and will have to be redefined.
- They have an outdated website, they have a Facebook page with 5000 likes, no Instagram at all, they do not deliver paper newsletters, nor they have TV advertisements.
- The warehouse layout is managed by the warehouse manager and warehouse workers, the delivery routes were designed by the drivers in cooperation with owner and managers and they function well, since they refine them every time a store is opened/closed.
- The culture is informal, since all the employees work there for a long time, but there are no stimuli from the owner, no team buildings, no extra bonuses etc. Also, the workers must work extra hours almost every day, since there is too much to do each day.

Appendix C. Group Contract

This contract is based on a Sample Group Contract (College of Natural Sciences, University of Texas Austin) and a Sample Team Contract (The Griffin Tate Group, 2002) and it is created for the 2nd Semester project of the Computer Science AP program at University College of Northern Denmark in the dmai0919 class's 3rd group.

The success of the group will depend on the cooperation and professionalism of its members. Employers will expect you to know how to work effectively in groups: how to determine what needs to be done, how to find information, how to assess information, how to share the workload, and how to resolve interpersonal conflicts that might arise. The effective collaboration includes, but is not limited to:

- Participating fully (in spirit and actuality)
- Participating professionally (i.e., civil discourse; abiding by the rules of academic honesty)
- Meeting responsibilities (i.e., completing assigned tasks on time and to the best of your ability)
- Taking the consequences of not abiding by the group's rules
- Giving group members proper credit where due
- Not taking credit where it is not due

After reading this document, each member needs to sign the document at the end.

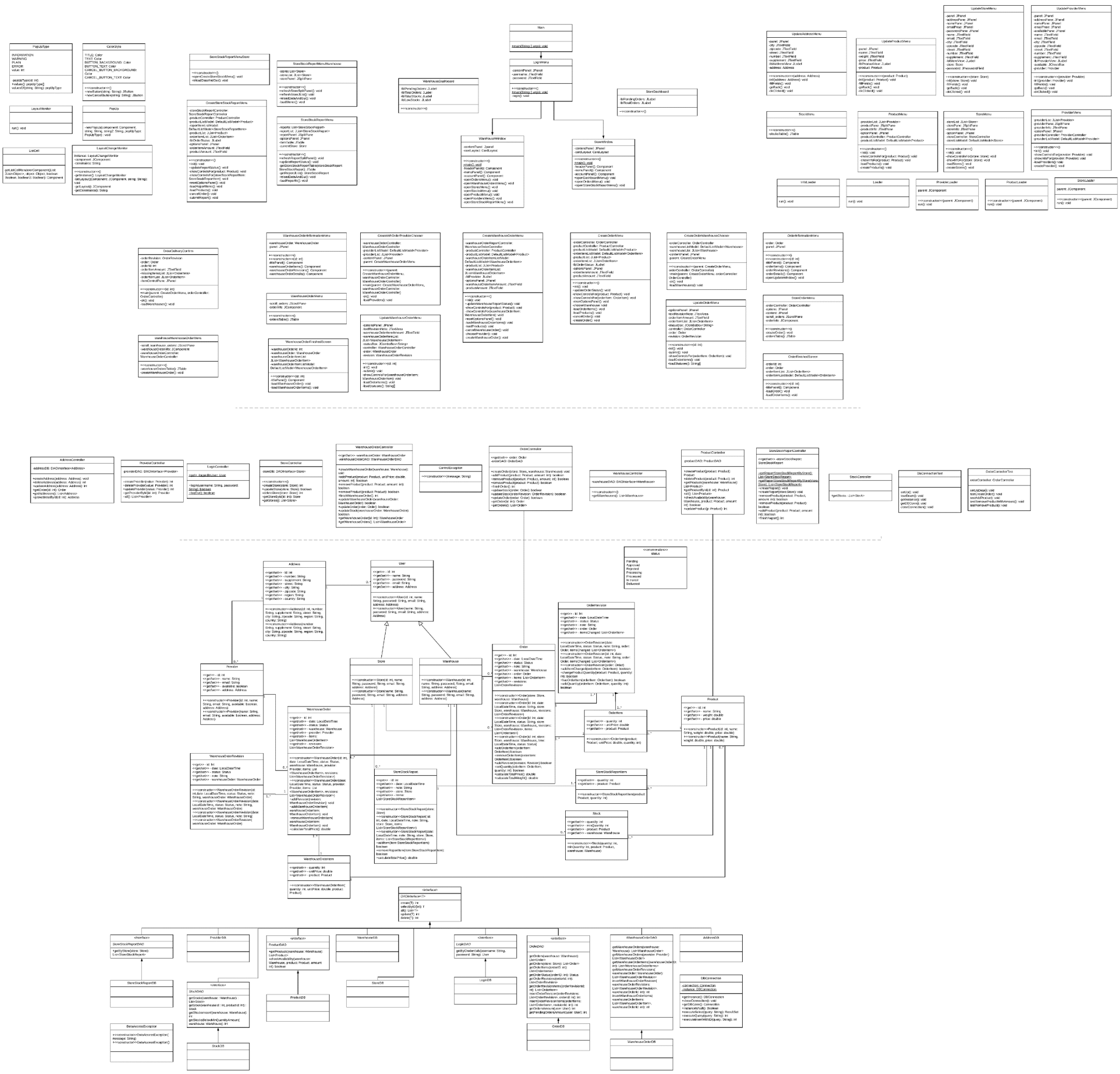
1. Each member agrees to show up to every scheduled meeting on time.
2. If a member is late, he may join, but he must catch up on what he missed.
3. Members agree to treat one another with respect.
4. Only agree to do work that we are qualified and capable of doing.
5. Keep other team members informed.
6. Focus on what is best for the project.
7. Ask all group members if they can support a decision before the decision is made

Date and signature of all the participants:

Aalborg, March 5, 2020

Erik Petra	Krisztián Henrik Papp	Marek Strúcka	Rémi Teissier	Tiago Fernandes

Appendix D. Design Class Diagram – Full



Appendix E. Relational Model

Appendix F. Full problem statement

Problem	Warehouse management and ordering system
Group members	Erik Petra, Krisztián Henrik Papp, Marek Strúcka, Rémi Teissier, Tiago Fernandes
Brief problem description	Tevos drugstore was opened in 1992 and has established itself well, with 25+ stores in Slovakia it is one of the greatest players. Their problem lies in insufficient and outdated warehouse management and ordering system.
Problem	The main problem Tevos drugstore faces is the communication between the central warehouse and the stores when ordering a supply for the stores. There are also problems retrieving the complete statistics for the managers. The orders are now made by calls between the stores' employees and the warehouse, which translates into many mistakes throughout the process completion.
Problem statement	Are we able to design a brand-new system for the company that would make the process of ordering more effective? Are we able to create such databases for the drugstore, that the stock-taking and other information could be easily retrieved? Are we able to ensure the smoothness, faultlessness, and user-friendliness of the new system?
Tools, methodology, processes	The group has decided to use the Unified Process as the methodology, meaning that an iterative approach with a focus on the most important use cases will be used, with later completing the whole project. Initial mockups and designs will be made in UML. As an IDE, Eclipse and SVN will be used for version control. The code will be in Java language, the GUI will be made using Java Swing, and the databases will be in MS SQL.

Appendix G. User Manual

Introduction

This user manual is intended to guide the user of the program through the features and functionalities of the program. Clarifying the steps for each task can improve the experience with the program for new users and this should be used as a resource when a problem occurs, or the user have difficult when using the system.

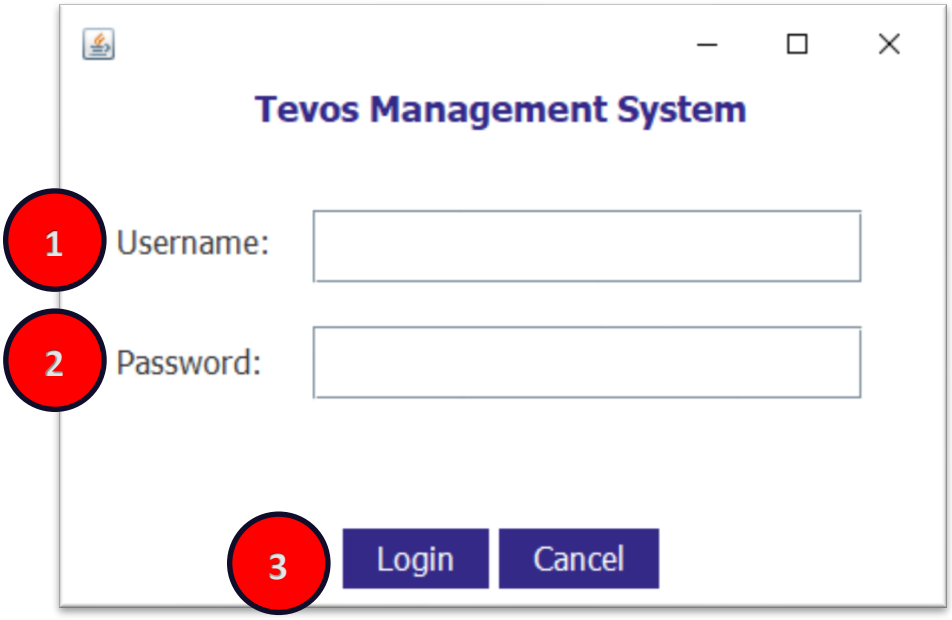
The system was built to be simple and user friendly so in every feature a clear step guide is easy to figure it out and there is a clear flow throughout the steps.

Below its possible to consult the functionality's made available for the user and how to use them.

Login

For security measures and system structures an account system is implemented and it's not possible to access further functionalities without a registered account.

Each account as assigned a specific location, being it a Warehouse or a Store. A user is only associated with one of those 2. When a login step is made, the system will only display allowed areas of the program to that specific user. In the case of the account used to login is associated with a Store, then it will be showing the window to manage a Store. The same applies for an account associated with a Warehouse.



This window presents a simple layout and the process to logging into the system can be done following the steps:

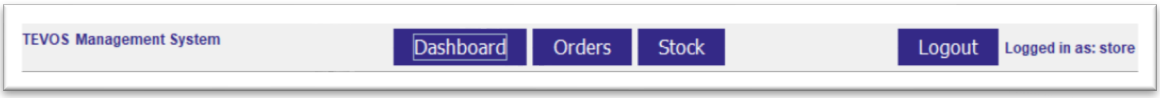
1. Insert account name
2. Insert password for the account
3. Click the Login button to log into the system. It's also possible to cancel this process if the user changes their mind, for that just need to click the button "Cancel"

Possible problems:

"The entered username or password doesn't match our records." – Means that the account information inserted doesn't match any registered account in the system. To solve this, confirm if the user information if it is correct.

Store

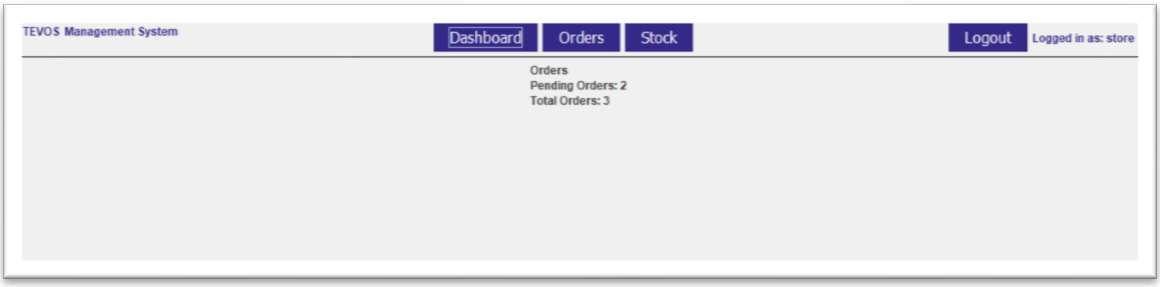
In the store window, which only stores' accounts can access, it's made available all the functionality required to a store to manage itself.



In the menu bar, it's possible to navigate between each different tab. On the right corner it's located the user logged in and the option to logout from the system.

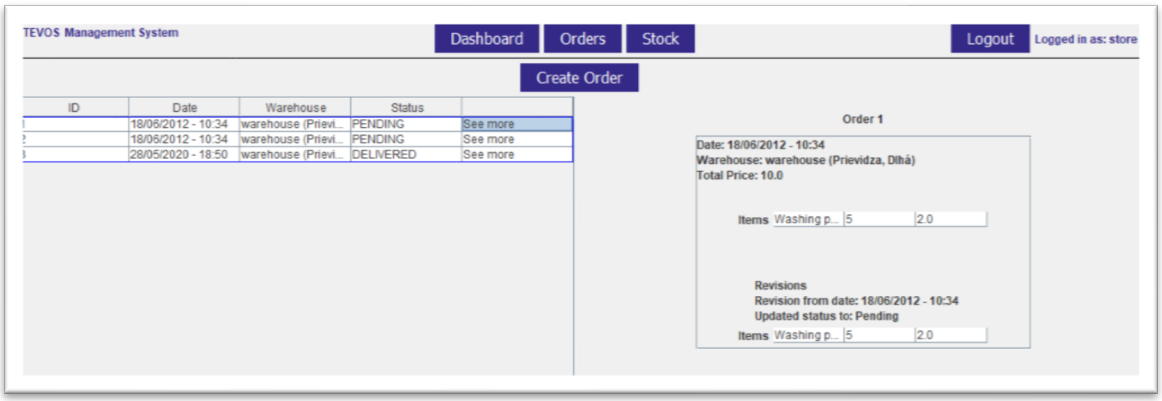
Dashboard

The dashboard serves the purpose of give feedback to the user about several data from the store such as orders status and important information. The store can take this information to make decisions or keep track of the store flow of orders and items.



Orders

The tab orders serve to manage orders from stores to warehouses and place new orders.

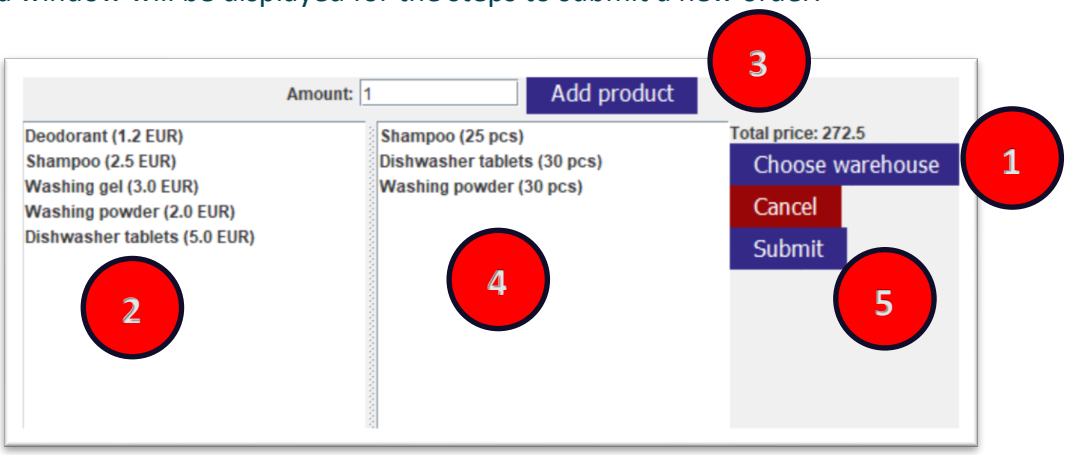


On the left panel is listed the orders registered in the system. At first sign, its possible to see important information about the orders such as data of creation, addressed warehouse and the latest status on the order.

When the user clicks on “See more” for a single row order the right panel is showed on the right with an extension of the details regarding the order. This information includes order details, items included in the order and changes made by warehouse at the moment of the newest status.

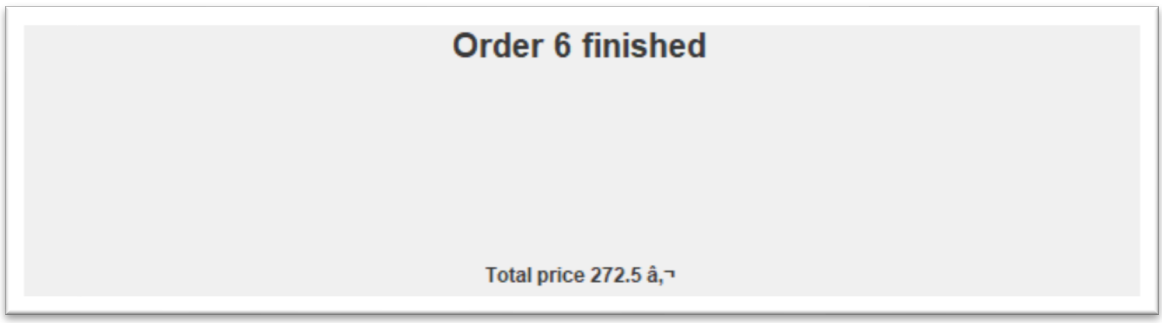
CREATE ORDER

When located in the orders menu, a button appears below the menu bar with the name “Create Order”. By pressing it a window will be displayed for the steps to submit a new order.



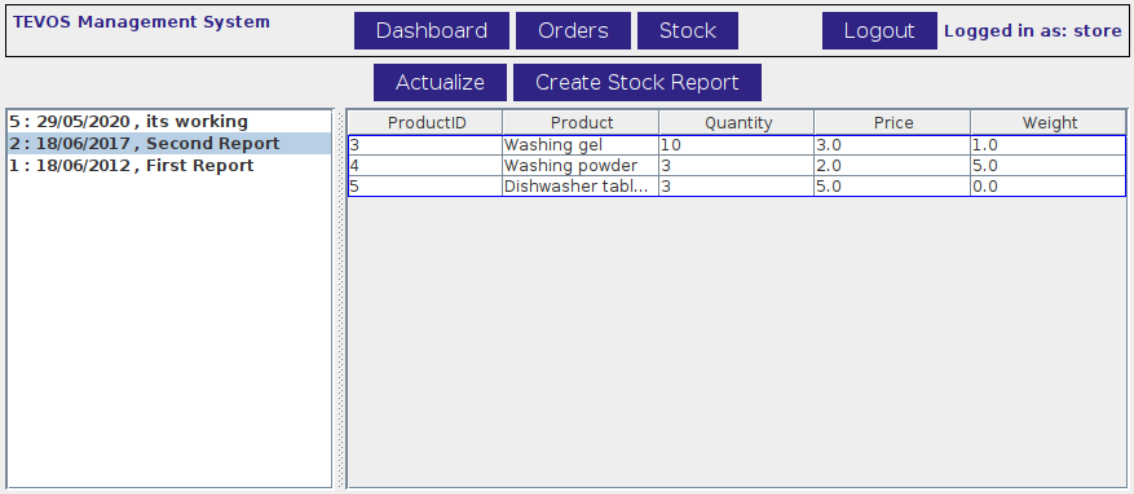
Steps to create an order:

1. A warehouse need to be chosen first so the Store can get available items to add to the order.
2. Select a product that is needed to be included in the order.
3. Store must submit the quantity necessary to fulfil their necessities and press the button “Add product” to be added to the order.
4. Products included in the order are listen in this panel. Store must confirm if the products desired are included in it. (Steps 2,3 and 4 have to be repeated until the order reach the desired products for the store)
5. To finish the order drawing the user must press “Submit” to register it in the system. The “Cancel” button is available to cancel this process at any time.



Stock

On the stock tab is possible to have access to stock reports for stores inventory which are made to keep track of items on stores and have important information to make the all the system more efficient.



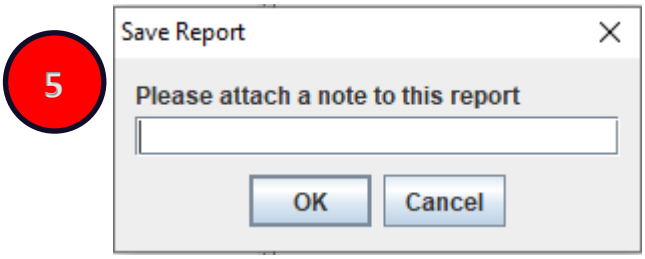
On the left panel it's showed the previous stock reports with registered date. When one of the reports is selected the right panel comes up and its possible to have a deep information about the inventory of the store, organized by products.

Create Stock Report

Stores will often create stock report and provide that information to warehouses. To have access to this process the user should click on “Create Stock Report” to be submitted to the window.

Steps to submit a store stock report:

1. A list of products available at the store are showed and user must go thought one at a time.
2. The quantity in inventory of the selected product need to the updated and after that the user must press “Add product” to add it to the report.
3. Here is possible to view the draft of the report with the added products and their quantity. (Steps 1, 2 and 3 need to repeated for each product in inventory).
4. After the report in complete, user must press “Submit” button to registered it on to the system. A “Cancel” button is also available to undo de draft of the report.

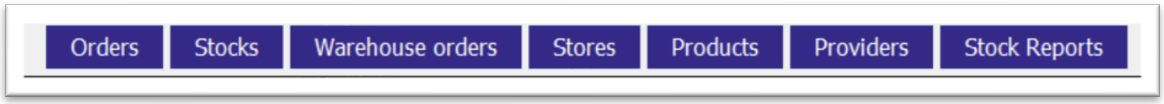


5. A note can be added to the report to complement the information.

After these steps are completed the report is uploaded to the system and is available for the Store and for the Warehouses.

Warehouse

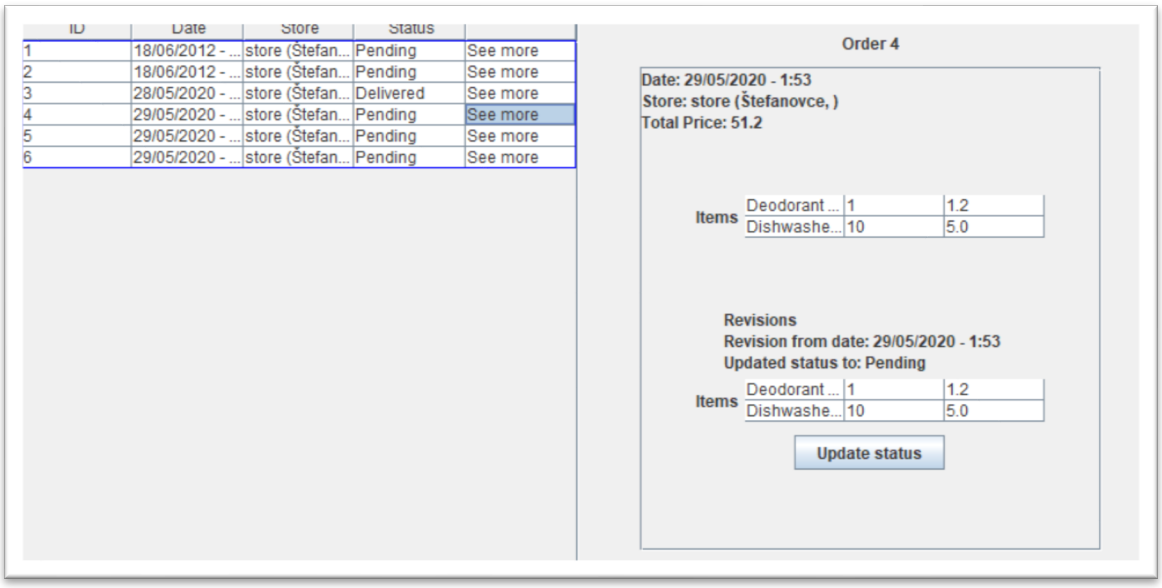
In the warehouse window that is available to warehouse accounts and only those it's made available all the functionality required to manage a warehouse and the relations to the stores.



The same as the store window, in the warehouse window the navigation is made by the menu where it's possible to switch between all the features.

Order

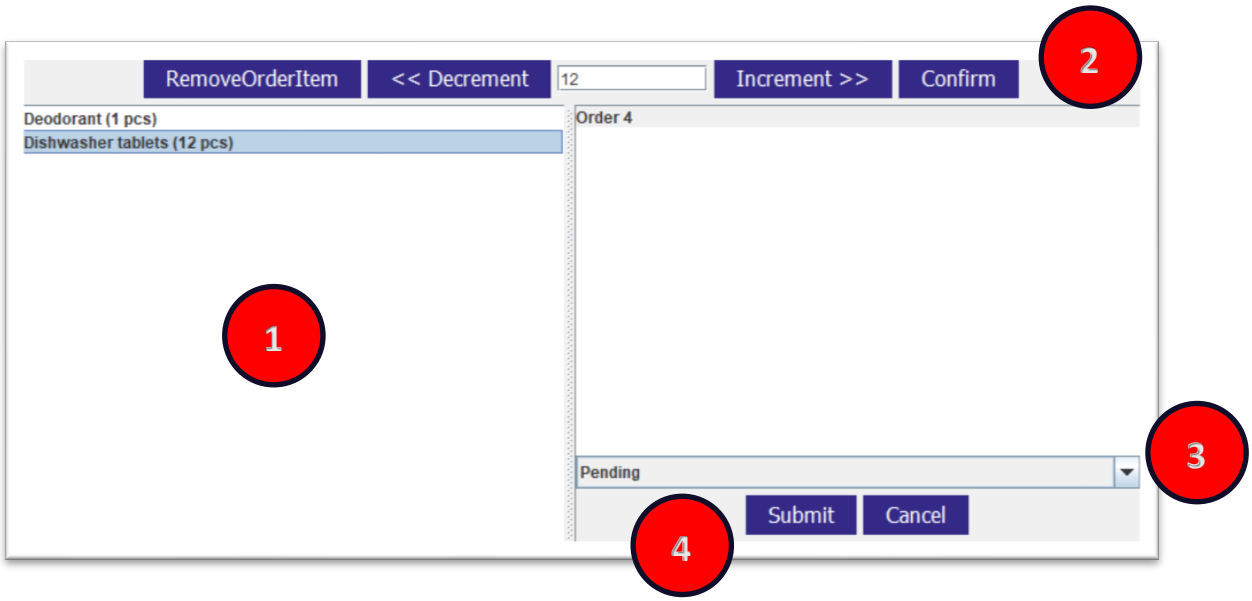
Tab orders redirect the user to a window where he can access current orders from stores.



On the left panel, the user can find a list of orders that need attention because of their status. If the users selects an order, the right panel will be displayed and extended information is available about the order.

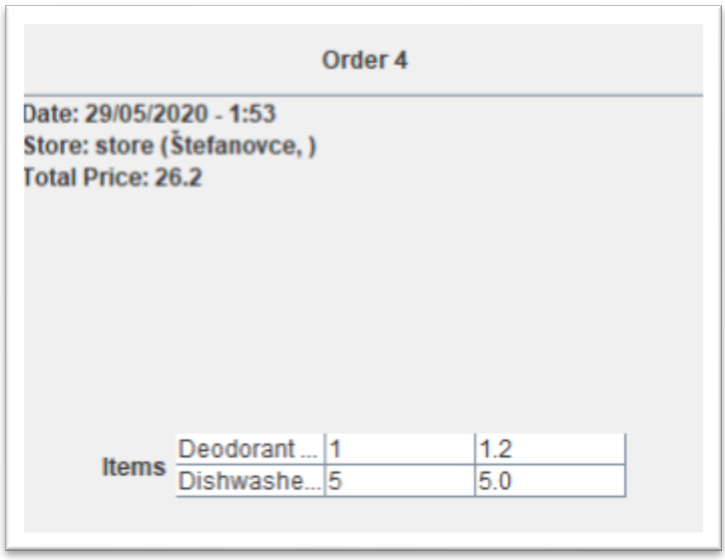
Update Status

A warehouse user as the option to update a order with the current status of progress. At the end of the right panel the user can find a “Update status” button and it will redirect to the window where is possible to manage the order.



Steps to update status of order:

1. List of products contained in the order.
2. When a product is selected, the user can choose to update or even deleted from the other if that is necessary. After finish the button “Confirm” must be clicked. If any changes are made to the order, the recent version of it is displayed in the left panel
3. A new status can be chosen for the order, alternating between predefined values (PENDING, APPROVED, REJECTED, PROCESSING, PROCESSED, IN-TRANSIT, DELIVERED)
4. When the order is updated it can be submitted to the system and the information is +updated to the store as well.



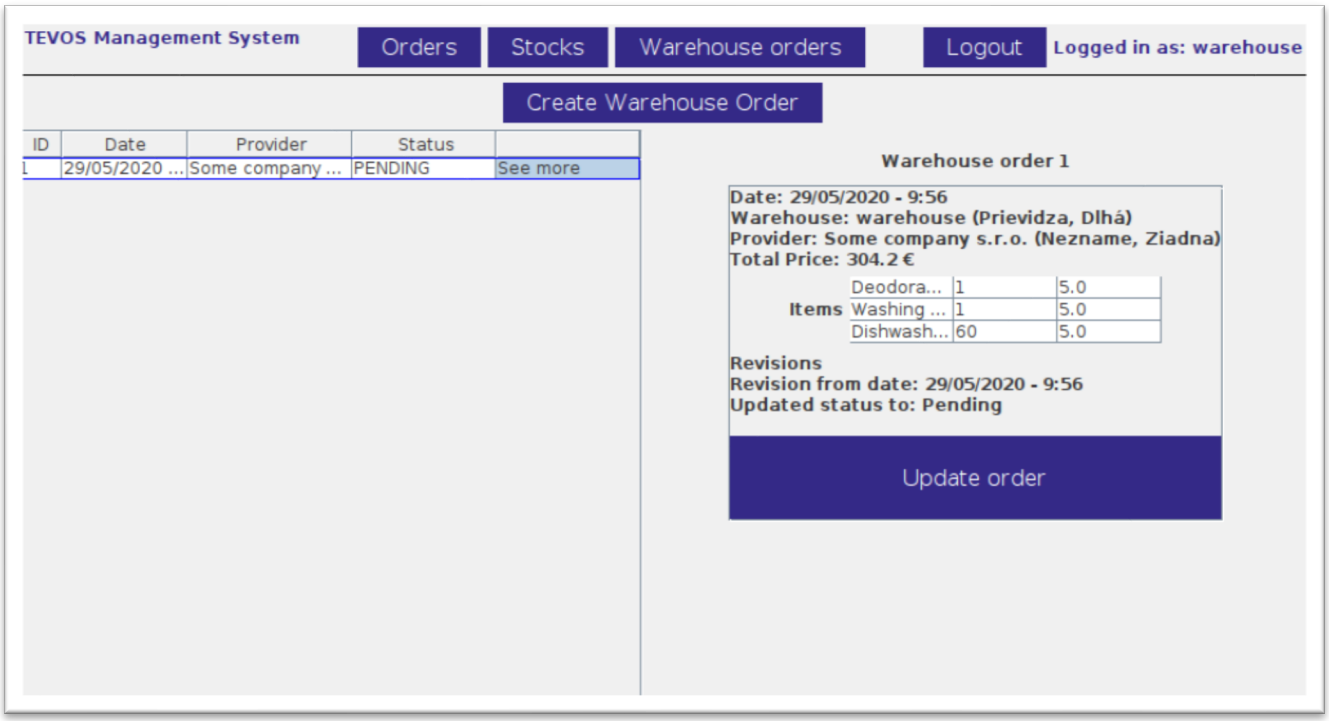
Stock

The stock tab is made available to the store with information in real time about the current status of each product in inventory. The user will find a window with all products listed and information regarding each product such as quantity in stock, minimal required quantity set for the product and the unit price.

ProductID	Product	Quantity	Quantity Min	Price
1	Deodorant	499	100	1.2 â,¬
2	Shampoo	100	150	2.5 â,¬
3	Washing gel	50	49	3.0 â,¬
4	Washing powder	300	200	2.0 â,¬
5	Dishwasher tablets	115	150	5.0 â,¬

Warehouse Order

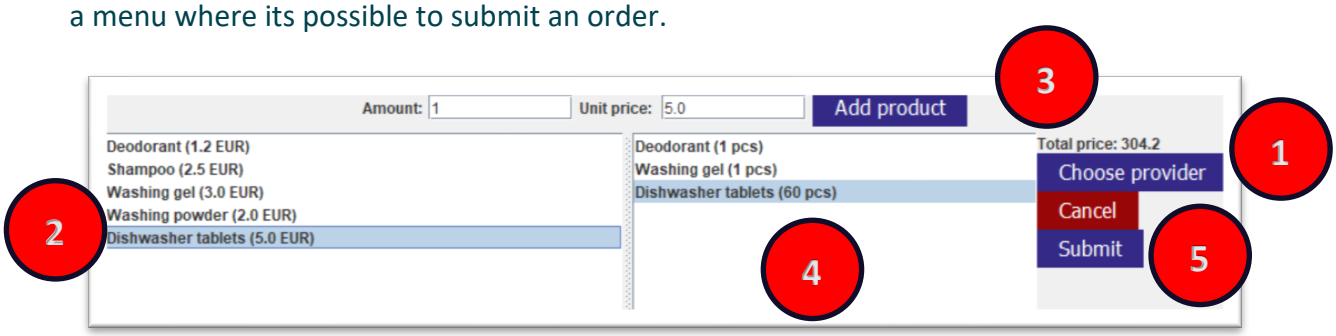
To register products in the system and add inventory the warehouse needs to create an order for a provider. This process is done in warehouse order tab.



As in previous windows, at the left we find listed the orders registered in the system and information regarding them such as date, the destination provider and current status. If an order is selected, the user can find extended information on the right panel.

Create Order

Below the menu bar, the user finds a “Create Warehouse Order” button where it will direct the user to a menu where its possible to submit an order.

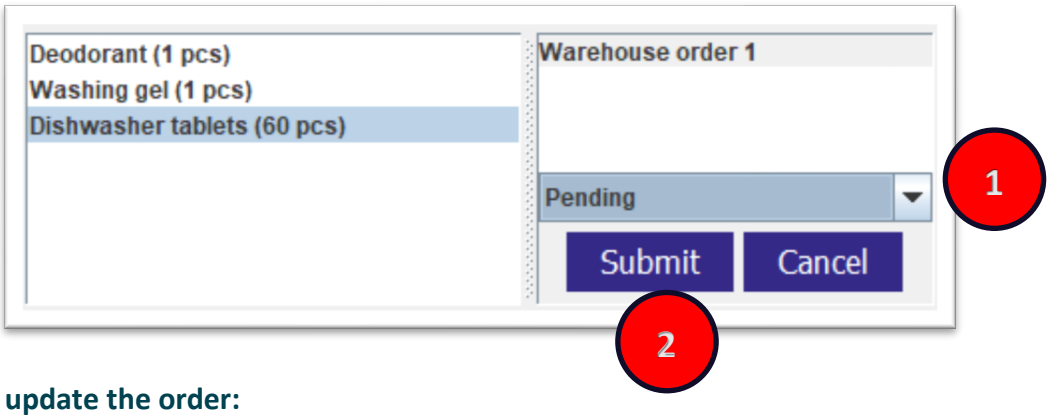


Steps to submit an order to a provider:

1. It’s necessary for the user to choose first a provider registered in the system so the available products can be displayed.
2. In here it’s listened all the products associated with the provider. The user should select a product.
3. Amount and unit price need to be inserted so the information can be stored in the system. To add the product to the order, draw the button “Add product” must be clicked.
4. User can have a look of the current draft of the order with the selected products. (Steps 2, 3 and 4 must be repeated until all products necessary are selected)
5. To submit the order the user can click on “Submit” and the order will be registered in the system. A “Cancel” button is available to undo the draft and cancel the process.

Update Order

Its possible to update the status of an order form a provider to help warehouses keep track of them. For that user uses the “Update order” button in the order info.

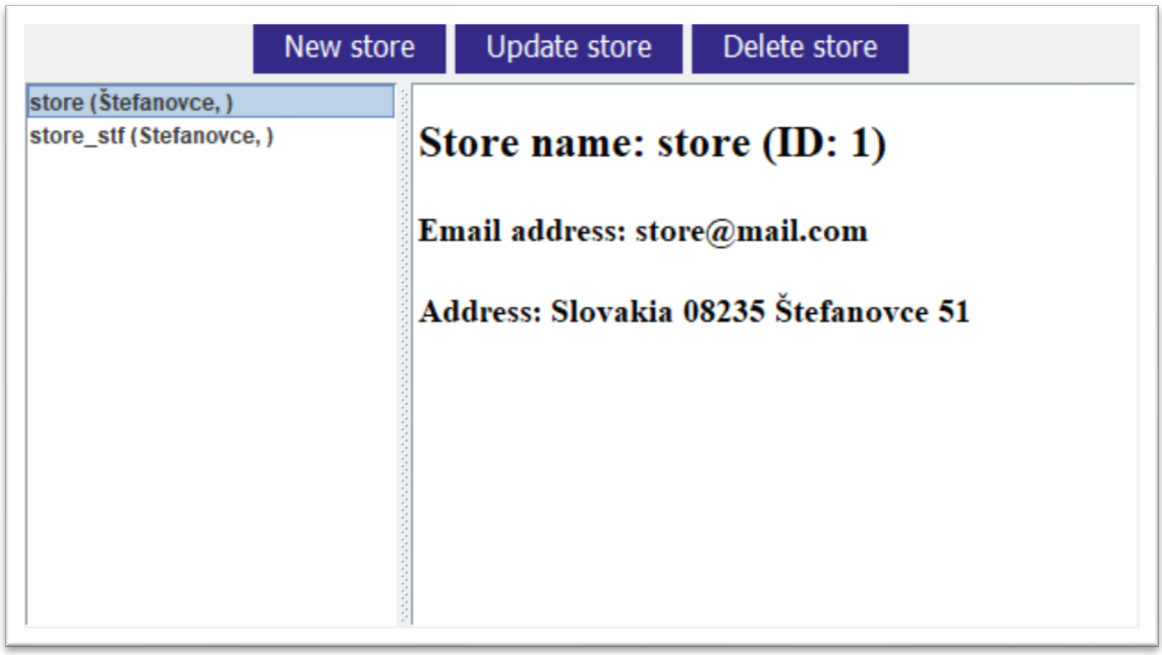


Steps to update the order:

1. A current status of the order can be update. User must choose a status from the predefined values (PENDING, APPROVED, REJECTED, PROCESSING, PROCESSED, IN-TRANSIT, DELIVERED)
2. After the update is done, user can register it to the system by clicking in “Submit” button. A “Cancel” option is also available to undo the update.

Store

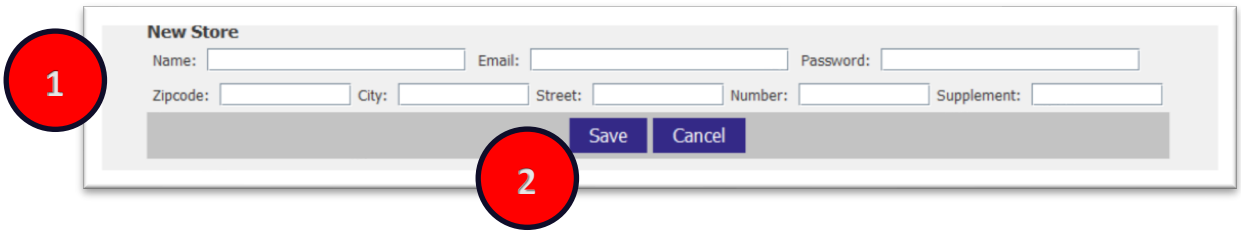
Warehouses accounts can manage stores registered in the system. To do that, user needs to access store tab.



As the same structure in other windows, a list of stores is displayed in the left panel and by selecting a store extended information is showed in the right panel. Warehouses have the option to manage stores, such as creating stores, updating and deleting from the system.

Create Store

The process to create a store is simple. These options is accessible by the button “New Store” on top of the store tab. When clicked, user is redirected to a menu where he can create a store.



Steps to create a store:

1. Information about the new store need to be written to the form made available to the user.
2. To complete the registration of the store user must submit it by clicking “Save” button. A “Cancel” button is also available to abort this process.

Update Store

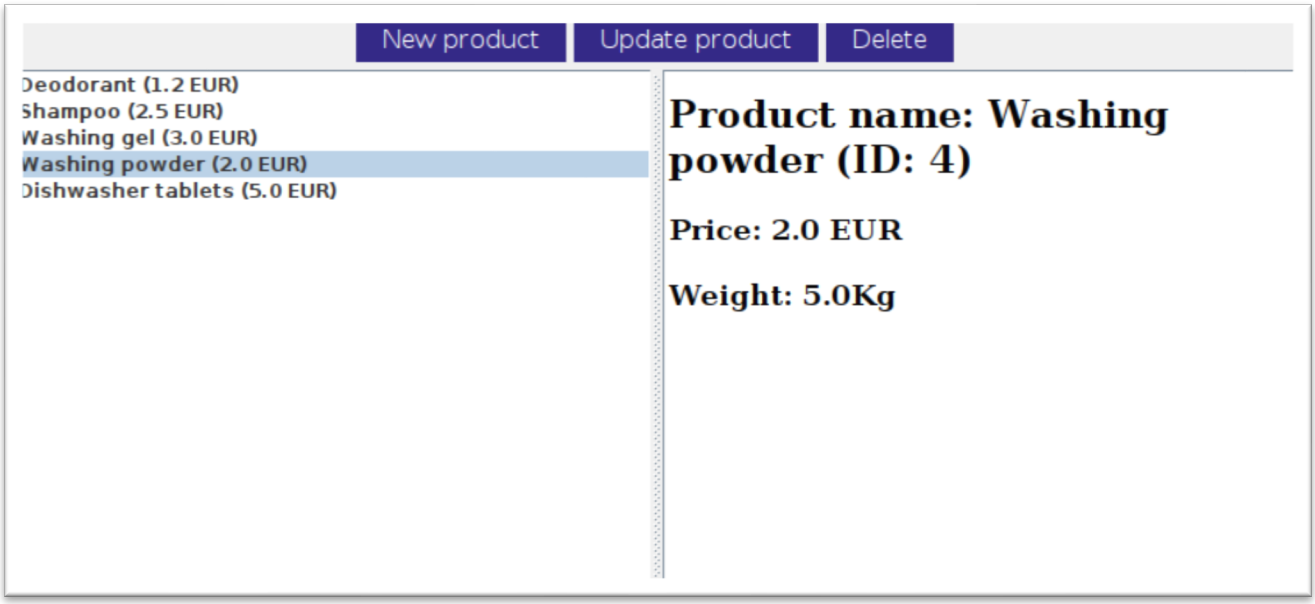
Warehouse accounts can, at any time, update information about stores if needed. A window with a form for stores will be displayed and user can manipulate the data. This follows the same steps as creation of order. A store must be selected first so the button “Update store” is showed.

Delete Store

In a case a store needs to be deleted from the system, user needs to follow a simple flow. It needs to select a store and the button “Delete store” will be available. By pressing it the store will be deleted from the system.

Products

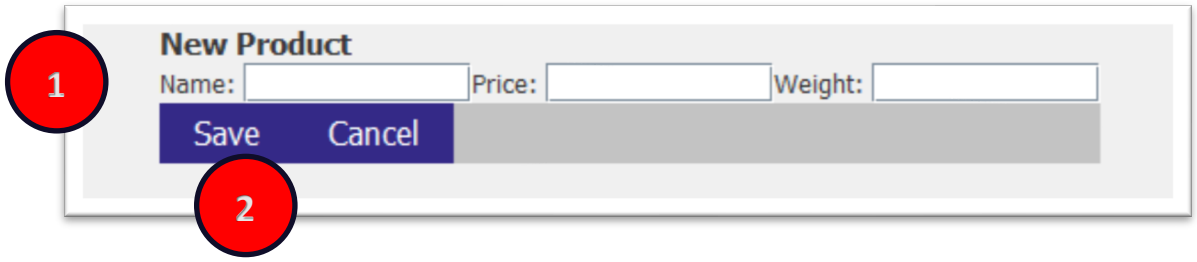
Only warehouse accounts can manipulate products in the system by adding them, deleting or updating. All these features are available thought products tab.



Products registered in the system are displayed in the left panel and by selecting one extended information is showed in the right panel.

Create Product

To create new products in the system, user need to click on “New product” button and it will redirect him to a window where this process can be done. The process to create a product is also simple.



Steps to create product:

1. Fill the fields in the form with product information
2. “Save” button is used to register the product in the system. “Cancel” button is available to abort the process.

Update Products

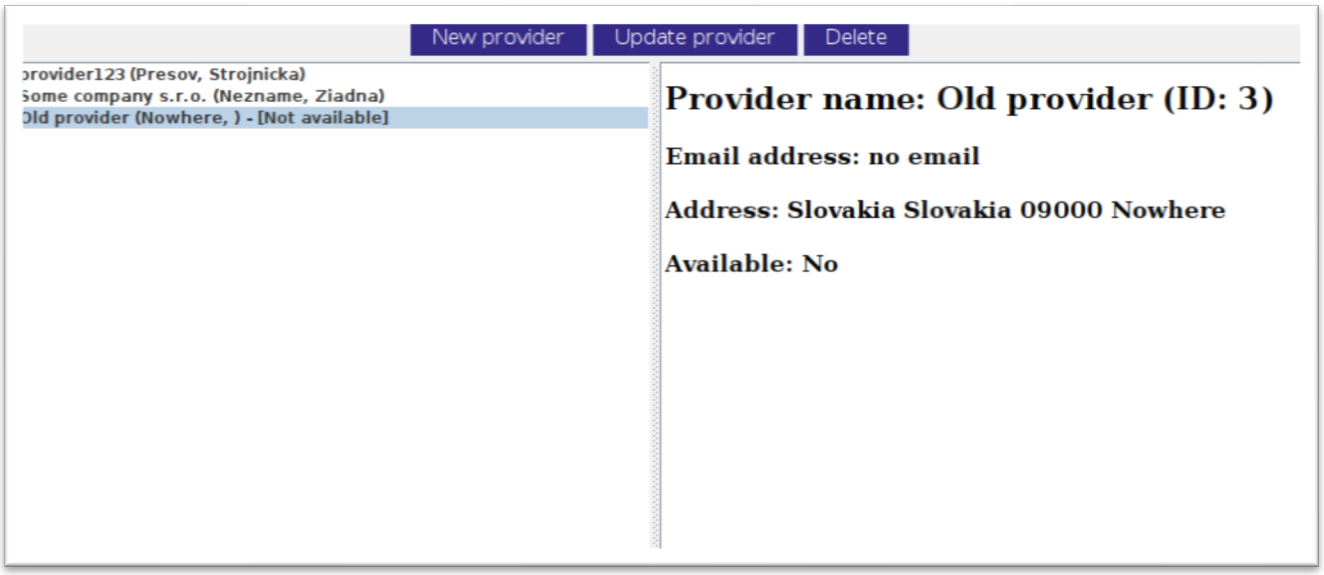
To update a product, user must select one from the list and then click on “Update product”. The products to manipulate the product flows the same steps as create order.

Delete Products

By selecting one product, it’s made available to the user the button “Delete product”. By pressing it, the user can delete a product from the system.

Providers

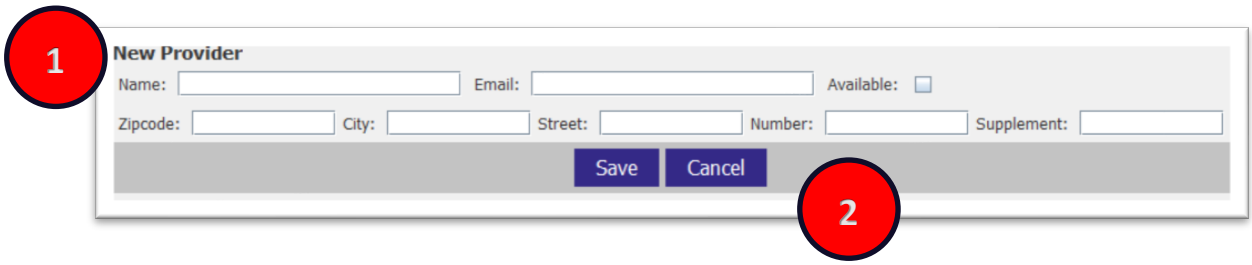
External providers can be registered into the system to help warehouses handle products and orders to full fill demand. These providers can be found at the providers tab.



A list of already registered providers in the system is made available to the user and its possible to extend the information about one provider.

Create Provider

As the same in others tabs, the process to create a provider follows the same structure as products and stores.



Steps to create a provider:

1. Fill the form with provider information
2. "Save" button to submit it to the system. "Cancel" button to undo the process.

Update Provider

To update the information of a provider, user must follow the same steps as in other tabs. First, a selection of a provider needs to be done and then the "Update provider" window is available to be opened. To manipulate the provider the same steps as a creation of an provider need to be followed.

Delete Provider

To delete a provider, the process is simple. Select a existing provider and the button “Delete provider” will be available. By pressing it the provider selected will be deleted from the system.

Stock Reports

Warehouses can access stock reports made by stores. These can be done by accessing the stock reports tab. To access a single stock report, user must select a store and then choose a report to get information about it.