TECHNOLOGY

2nd Semester Project

Vengeful Games Adventures

Linda Augustina Carolus Fuchs
Simon Jensen
Juliána Paľková
Dimitar Pilyakov
Ákos Skutovics

UCN – COMPUTER SCIENCES AP DEGREE
DMAI0917 – PROJECT GROUP 1

This page is intentionally left blank.

University College of Northern Denmark

Computer Sciences Academy Profession Degree Programme

Class:

DMAI0917 – Project group 1

Title:

2nd Semester Project

Vengeful Games Adventures

Project participants:

Linda Augustina Carolus Fuchs

Simon Jensen

Juliána Paľková

Dimitar Pilyakov

Ákos Skutovics

Supervisor:

Dimitrios Kondylis

External Collaborator:

Thomas Edvard Mikkelborg Calix

(owner of Vengeful Games)

Abstract

The paper evaluates the planning, creation and implementation of stock management and bookkeeping software. The client company is displeased with the current software and website they are using because they do not fit his requirements and prove to be inefficient and slow. The improvements encompass a replacement for the current bookkeeping- and order tracking system with connection to a database for data storage and statistical analysis. The text describes, in detail, the process of outlining to creating the software in terms of business planning, system development and programming. The development method used was the unified process, an iterative and use-case based approach. The results of these changes are a more efficient and agile working environment. This software is not only an improvement over the prior one, but it is good framework for future upgrades.

Submission date: 04-06-2018

Revision date: 24-06-2018

Number of characters including white spaces, excluding cover, Table of Contents, tables and Appendices: 89.054 (of total allowed 96.000).

Linda A.C. Fuchs Simon Jensen Juliána Paľková Dimitar Pilyakov Ákos Skutovics

This report has been written as part of the 2nd semester project for the Computer Sciences course at UCN, University College Nordjylland. The main objective of this project is to "effectively and professionally create and implement IT systems with interfaces towards users and databases as well as master fundamental elements of the computer science profession" (University College Nordjylland, 2014, p. 4)*, which includes "new and further development of small database systems (…) the selection and use of technology (…) and a relevant business aspects"* (University College Nordjylland, 2014, p. 5) *within the timeframe that has been given by the institution.*

Preface

We would like to thank **Dimitrios Kondylis** for his continuing support during this project. His feedback was invaluable and led to the successful completion of this project.

And we would like to thank **Thomas Edvard Mikkelborg Calix**, owner of **Vengeful Games** for allowing us take a closer look at his company and use it as a basis for our 2nd semester project.

Linda, Simon, Juliana, Dimitar and Ákos.
04-06-2018

This page is intentionally left blank.

Table of Contents

This page is intentionally left blank.

Introduction

Starting up a new business can be an exciting and daunting new adventure, but it also has a high risk of failure. In fact, only 43% of all Danish start-ups survive the first 10 years (Center for VækstAnalyse, 2017, p. 18). A recent study into start-up failure shows that there are a lot of different reasons why start-up companies fail (CB Insights, 2018) which is why this project aims to assist in avoiding some of these points and give the business a helping hand by analysing the business model, giving advice and introducing a new software system which will make stock management significantly less time consuming.

The subject for this project is board game store 'Vengeful Games' (Calix, 2018), a one man business owned and run by Thomas Edvard Mikkelborg Calix. This company uses an '**Internet Pureplay**' business model, also called a '**Clicks-Only**' model (Chaffey, 2015, p. 66), where the store mainly has an online representation and the customer simply orders from a large range of board game products from their website and the orders are automatically placed in the company's order list. This self-service ensures the owner has as little as possible involvement in setting up orders, freeing up his time for other managerial tasks.

Problem Area

The self-service mentioned above is great for eliminating order setup time but has two massive drawbacks when it comes to processing them. One: The customer can create (for example) 10 different individual orders, all containing a single product. This will result in the person who handles the order, having to manually go into each single order and click every single item and changing item availability/ product order status, which is very time consuming. The second drawback is when a customer creates one large order which contains items that are currently unavailable. The person handling this order will have to come back to this order multiple time to change the different items' status every time they become available for shipping, plus they have to keep an eye on the order to ensure they don't forget this order has an item in backorder.

These issues can occur in any combination and can cause unneeded stress and frustration for the store owner. A Burnout and a resulting lack of passion (nr 19 and 14 respectively on the above mentioned top 20 list of reasons of failure) could be a very likely outcome. Especially once the business starts growing and the company gets more customers, and thus more orders, big and small (nr7, not scalable).

Problem Statement

Can the efficiency and usability be increased in a bookkeeping system according to Vengeful Games' needs? If yes, to what extent can the change be done before the requested deadline?

How, and in what way, can statistics of items and customer be implemented to make a significant contribution to the future business development of Vengeful Games for the next 5 years?

In what way do we ensure that all the data inserted by a user is valid and operations are executed simultaneously?

For the full Problem Statement, please refer to Appendix C.

Empirical Data Collection

To ensure that the findings of this report live up to a high standard that can be expected from a university, all information which has been gathered and used for this report, has been checked for authenticity and reliability. By using only data that has been gathered through **empirical research** this high quality standard can be ensured.

Structure of the report

The report is split into 6 different parts (not including the introduction) as listed below. The Project plan on the next page gives a more detailed explanation of the individual chapters of each part of the report. Words in **bold** (throughout the entire report) are explained in further detail in the Glossary in Appendix A, and a complete reference list for sources used can be found in Appendix B.

Introduction with a *project plan* and *risk analysis*.
Part 1 elaborates on the *feasibility study* which analyses the business.
Part 2 focusses on the *methodology* used to research and construct this project
Part 3 details the different phases in the *Unified Process*.
Part 4 holds the discussion and conclusion of the project
Part 5 reflects on the *group process* and gives feedback on the members and the project.
Part 6, the final part of the report, contains a collection of relevant *appendices*.

Project Plan

The **Project Plan** gives a detailed description of the entire scope of the project, in chronological order. Starting with a detailed explanation of the setup of this report, some logistical information regarding organising the data, a **Risk Analysis** about the project and lastly a **Phase Plan**.

Detailed explanation of report setup

This report follows the main structure of a system development according to the **Unified Process (UP)**. Chapters are combined into 6 distinct parts in the report, which take the reader through the complete development process step by step. Below follows a detailed explanation of the contents of each of these 6 steps.

Part 1: Feasibility Study

The first part of the report will focus mainly on the business side of the project. This will include an analysis of the company which describes its **Organisational Form**, **-Structure** and **-Culture**, the **Mission** & **Vision** and an analysis of its **Digital Business** and supply chain management. Once all this is defined it's possible to make a **Stakeholder Analysis**, define the influence of '**Porter's 5 Forces**' and make a well grounded **SWOT Analysis**. Ultimately leading to the **Business Case** which includes a **Payback Projection**, a **Cost-Benefit Analysis**, and Conclusion and Recommendations about the feasibility of the project

Part 2: Methodology

The second part features an explanation of the **Methodology** used in this project. If includes a brief explanation of methodology, a comparison of the different types that are available to use, which type was picked and a justification for this choice.

Part 3: Unified Process

The **Unified Process (UP)** part of this report contains a detailed explanation of the different stages of the unified process as well as all relevant artefacts which are needed to fully develop the system for Vengeful Games. It is split up into four sections, each elaborating on the 4 different development sections of the Unified Process; **Inception**, **Elaboration**, **Construction** and **Transition**

3.1 Inception

The inception chapter focuses on getting the proper **Requirements** for the system. This part includes the **Activity Diagram** and accompanying **Employee-Task-Goal-Table**. From these the **Use-Cases** are identified and a proper **Risk Analysis** can be made, after which a **System Vision** is presented which includes a **Persona Profile** and **Mock-Ups** for the system.

3.2 Elaboration

The Elaboration chapter includes the **Fully Dressed Format** of the Use Cases that have been implemented and the **Test Cases** that go with these use cases. From these Use-Cases a **Domain Model** has been set up which has been transformed into a **Relational Model** for the use with a Database. This part also describes the Database and SQL scripts used. To conclude the Analysis part of the project;

System Sequence Diagrams have been set up for the relevant interactions with the user and the system, as well as **Operation Contracts** that go with them. **Sequence Diagrams** then show the interaction within the system when a method is called. Last; a **Design Class Diagram** which shows the complete system with the **3-Layer Architecture** in place.

3.3: Construction

In contrast to the previous chapters in the Unified Process, the Construction chapter is quite short and defines the **Code Standards** and the architecture that has been used in the program. Furthermore it explains which Use-Cases have been implemented first, why these have been implemented and the choices that were made while programming.

3.4: Transition

The Transition chapter is brief, it explains the steps taken to ensure proper project delivery, including a **Software Manual**

Part 4: Discussion & Conclusion

Part 4 is a **Discussion** on the development process and **Conclusion** which evaluates the project.

Part 5: Group Evaluation

The group evaluation part contains an explanation of the group contract, the group process as a whole and the feedback from each group member on each other.

Part 6: Appendices

The final part of the report contains a collection of relevant appendices. The first is a Glossary, which contains a detailed explanation of various terms used through the report. Followed by a reference list, citing the sources used during the construction of the report. The appendices that follow after that are in chronological order in which they are mentioned in the report.

Logistical Information

Below is some logistical information in regards to the project. The SVN location where the program can be found, the Database location and the revision number relevant for the version of the project that this report reflects on.

SVN & SQL-Database link

SVN: 'https://kraka.ucn.dk/svn/dmai0917_1sem_projekt_1/2nd_semester_project**'**
SVN-Version: 'Revision Nr. 327**'**

SQL-Database: 'kraka.ucn.dk/dmai0917_1067303:1433**'**
username: 'dmai0917_1067303**'**
password: 'Password1!**'**

Project Risk Analysis

To properly assess all the risks involved with this project, and to ensure that the project does not suffer any unnecessary delays, a risk analysis has been set up according to the steps stated by Dave Chaffey in his book on Digital Business and E-commerce (Chaffey, 2015, p. 511). This analysis displays the risks on the left side of Table 1, the probability of this occurring on a scale of 1-5, the severity of this risk on a scale of 1-5 and the total threat level of this risk, which is found by multiplying the numbers for probability and severity. Finally, a short solution is presented in the 'Handling' column, which gives a brief guideline on how to prevent this risk, or how to handle it should it occur.

Table 1: Project Risk Analysis

Risk	Probability	Severity	Total	Handling
Scope of the project too big	5	4	20	Setting goals and deadlines
Feature Creep	4	4	16	Setting goals and deadlines
Trouble connecting to new database	4	4	16	Ensuring good test environment
Integrating unknown technology	5	3	15	Ask for assistance
Issues with Communication with Client	4	3	12	Working according to contract.
Clients make requirement changes	3	3	9	Working in iterations
Iteration length	2	4	8	Ensure 1-wk Iterations according to school guidelines.
Extended Holiday	4	2	8	Proper project planning and risk assessment, working in iterations.
Team members drop out of project	2	4	8	Proper project planning and risk assessment, working in iterations.
Communication Issues Within Team	2	3	6	Working according to contract. Pair Programming
Code standardisation	2	2	4	Code Clean-up, Set up guidelines, Pair Programming. Code Documentation
SVN syncing problems	1	4	4	Fix manually with code-/ SVN revision history
Bad coordination because of team members working during nights	3	2	6	Proper project planning and risk assessment, working in iterations.

Phase Plan

As mentioned earlier, the methodology chosen for this project is the Unified Process (more about this choice and a justification in Chapter 8). In order to manage a project in the Unified Process it is recommended to use a project management program which can keep track of the different iterations in the project, this not only to ensure the project stays on track, but also to have a reference, a historical overview, of the progress on the project. That way it is possible to reflect and evaluate better on the group process later on.

For this project, the free web based project management application "Trello" (Trello, 2011) was used. Trello features the use of different boards which were ideal for use for the different parts of the project, and the use of lists and cards within said lists. Figure 1 displays an example of how the System Development board was organised. Every part in the project has been divided into weekly iterations, starting with a to-do-list containing the cards for the week in question which are then assigned to the different group members. Each card also has a deadline and where needed a deadline or requirements tag. The Unified Process, which is an iterative approach is thus assured as long as this plan is followed.

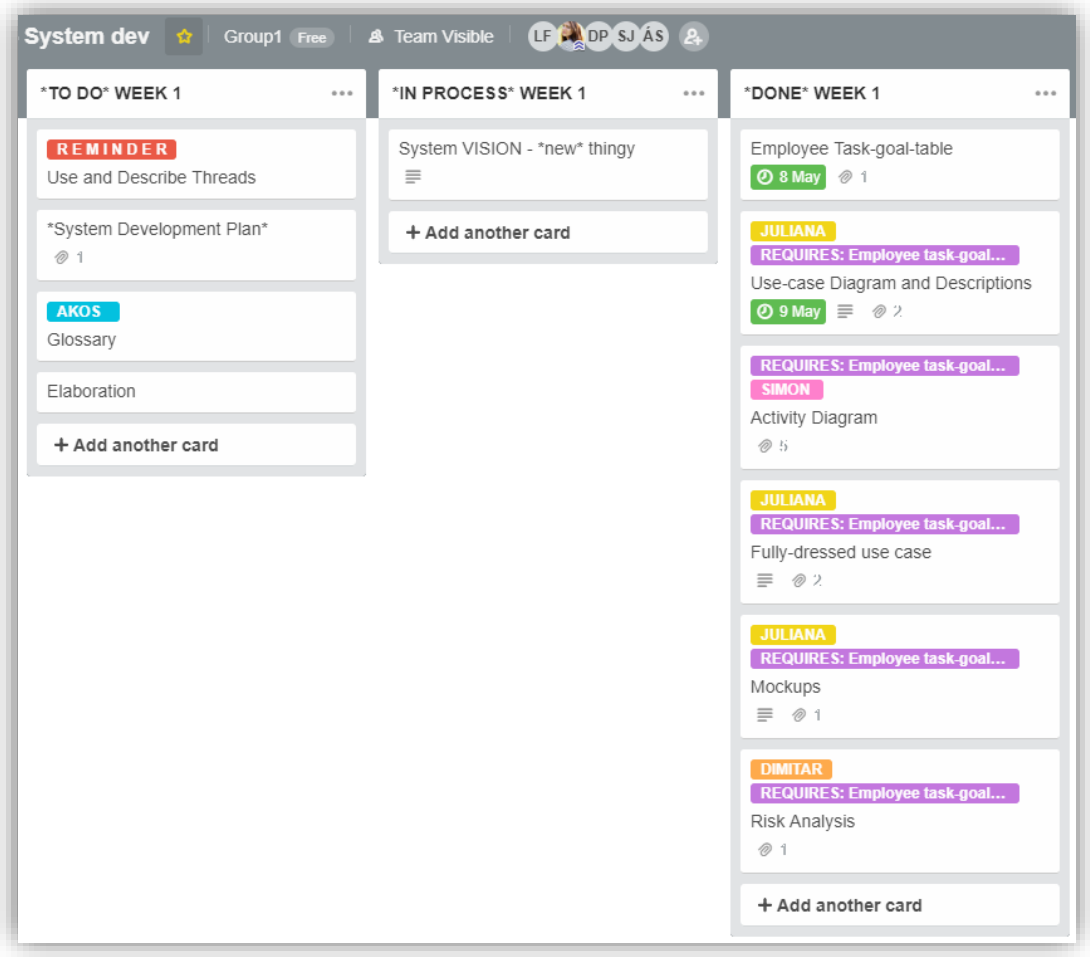


Figure 1: Trello board example

The full phase plan can be found on the next page in Table 2. For each part of the project, it displays the goals that have been set, how many iterations have been planned to implement this part of the project, the key milestone dates, and which artefacts should be included.

Table 2: Phase Plan

Phase/ Milestone	*Feasibility Study*	*Inception*	*Elaboration*	*Construction & Transition*	*Discussion & Conclusion, Group Evaluation*
Goal	To determine if the project is feasible and has a positive impact on the business.	Establishing initial requirements and basic framework for the program.	Elaborate on the most critical Use-Cases and start coding.	Propper code implementation of critical use cases, adding extra features. And create a user manual for the customer.	Evaluate the project, discuss possible improvements for next time. Draw conclusions and evaluate the team.
Number of iterations	2	1	1	2	1
Deadline	Sunday 6th of May	Sunday 13th of May	Sunday 20th of May	Sunday 3rd of June	Sunday 3rd of June
Artefacts	• Introduction • Organisational form, structure and Culture analysis. • Mission, vision & strategy. • Digital business & SCM. • Stakeholder Analysis. • Porter's 5 forces. • SWOT Analysis. • Business Case. • Methodology.	• Activity Diagram • Employee-Task-Goal-Table. • Use-Cases. • System Risk Analysis. • System Vision. • Persona Profile. • Mock-ups.	• Fully Dressed Use-Cases. • Test Cases. • Domain Model. • Relational Model. • Database & SQL Scripts. • SSD & Contracts. • Interaction Diagram. • Design Class Diagram.	• Code Standards. • Architecture. • Use-Case Implementation. • User Manual.	• Discussion. • Conclusion. • Group Evaluation. • Group Contract.

This page is intentionally left blank.

PART I

Feasibility Study

This page is intentionally left blank.

1. Organisational Form, Structure and Culture

This section describes the organisational form and structure of 'Vengeful Games', which in itself proposes quite an interesting challenge. The company is a one-man business, so labelling the company as any one type of form or structure is quite difficult at this point in time due to the lack of other employees which help define said structure. It is however possible to make a prediction based on the assumption of expansion in the future and the type of business that Vengeful Games is in.

1.1. Form

To properly describe this one-man start-up company, the organisational forms as described by Mintzberg's, will be taken into evaluation. Regardless of describing a one-man company or even if normal growth is taken into consideration, the company would still be structured as a 'simple structure' which defines itself as a company whose "strategic apex dominates the supervision of the work done by the operating core. Numbers of technical and support staff are minimal and there is little in the way of formalised rules and procedures." as according to (Brooks, 2009, p. 213). Any of the other structures from Mintzberg's organisational forms theory only apply to either very large organisations or highly complex businesses which require fast reactions, all of which do not apply in this situation. But what does this mean for Vengeful Games? The company will mainly consist of the strategic apex (directors, senior executives etc) and an operating core, if any.

1.2. Structure

As far as the organisational structure goes, the company has three choices, a one man business without a proper structure. Or, should the company grow, a multifunctional structure where the company is divided according to the functional tasks of the employees (e.g. financing, acquisition, distribution, etc) or a multidivisional structure where the company is divided by type (e.g. board games, kickstarter products, miniature games, etc). The other structures are really only applicable to larger organisations (Brooks, 2009). In this case it would be wiser to go for a multidivisional structure. This structure has the benefit that the employees won't become too specialised and are thus able to switch freely inside the different branches of the company as they can do the same job in a different field (for example switching from processing board game orders to processing kickstarter orders) (Brooks, 2009, pp. 194-197). Finances and management can stay at the head of the structure or outsourced to a specialised company. A suggested structure in Figure 2 below.

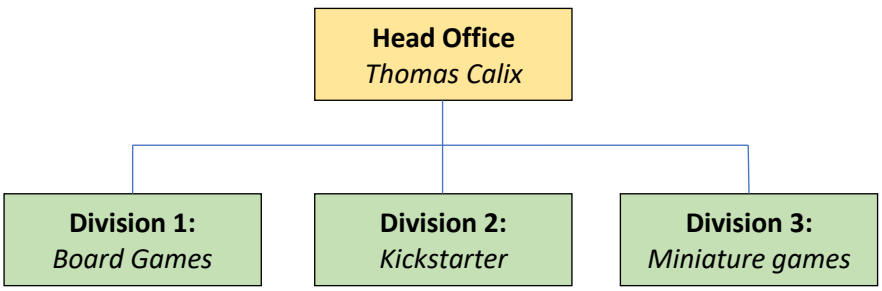


Figure 2: Organisational Structure - Future Vision

Another thing to take into consideration is whether or not the owner of Vengeful games should expand the business or not. His primary reason for having the company is to explore the idea of owning and maintaining an 'clicks-only' board game store in Denmark and to see where it could lead. The company serves the purpose of an entertaining and interesting hobby, more than a professional enterprise with a goal of growth and high profits.

Keeping this in mind, it might not be a good idea to expand as this would mean the company, which is currently registered as a one-man business (or a '***enkeltmandsvirksomhed***' in Danish terms), will have to switch to an **ApS** (or an '***anpartsselskaber***' in Danish) and will require for him to have complete transparency in his financial books (JUF.dk ApS, 2018). With his current structure he is able to see into his competitors books to see their financial state and adjust his company profit margins on products in such a way that he is able to easily out-compete his competitors as much as possible. His competitors are not able to do this to him in return as his books are private in the current structure.

1.3. Culture

This section will briefly go into the desired organisational culture for the company, should the owner decide to go for the expansion option. Taking into account that the board gaming industry is a multi-national industry, with many products coming from Germany, England and even the United States and Asia, the company will have to be open to working with many different nationalities and inevitably with different cultures. Figure 3 shows an overview of the 10 organisational cultural characteristics according to (Larson & Gray, 2011, pp. 76-80), and where on the slider (left or right) the company should be. The figure gives a schematic overview of the ideal characteristics for an internet store which works with an international supply chain where for example the Conflict Tolerance (nr 8) is high due to working with people from many different cultures. For a more detailed description of all these 10 characteristics, please refer to Appendix D.

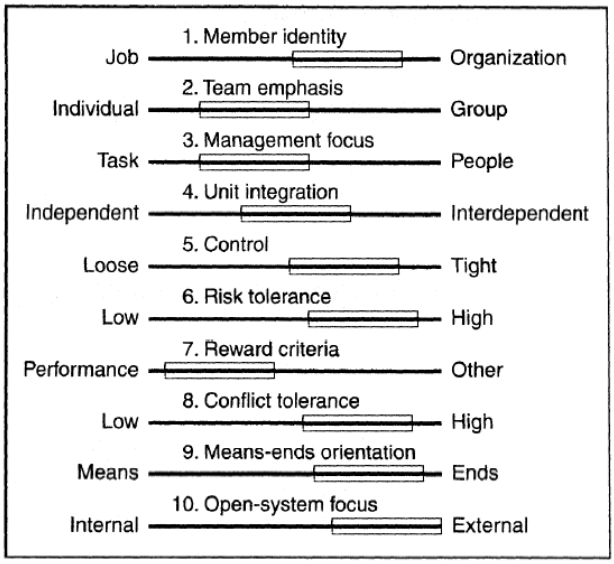


Figure 3: Organisational Characteristics

2. Mission & Vision, and Strategy

The mission and vision reflect what the company wishes to achieve and how they wish to do that. In Vengeful Games' case, the mission and vision were not defined yet. In order to properly be able to advise Vengeful Games and to get a better view of how and where a new system would fit into the company, the following mission and vision were set up for, and approved by the CEO.

2.1. Mission

As described by (Bloisi, Cook, & Hunsaker, 2006, p. 76) the mission is: "The fundamental purpose of an enterprise that defines the nature of its business and provides strategic direction to unify the use of human and other resources". With this in mind, the following mission statement has been created, and approved by the CEO:

"To expand one's imagination

offering the games with the best price

on the Danish market."

2.2. Vision

As described by (Bloisi, Cook, & Hunsaker, 2006, p. 77) the vision is: "A desired future image of the organisation and its processes and products that integrates current realities and expected future conditions within a specific time frame". With this in mind, the following vision statement has been created, and approved by the CEO:

"Our vision is to take people

through every gaming 'adventure'

that has ever been created."

2.3. Business Strategy goals

Strategic goals mentioned below are the company's way to achieve its future development. Setting up the goals is undoubtedly one of the most important aspects of entrepreneurship.

The most significant of all the strategic goals is to keep the lowest price on the Danish market. To achieve that, the company encourages its customers to document the products with the lower prices they have found at the competition.

Moreover, the current maintenance of the company requires more time than necessary. Therefore, the company is hiring special IT-consultants to create completely new IT system and thus, to avoid the time redundancy whereas more important tasks could be done.

Last but not least, in the future prospects, the expansion of the company would not probably be very essential. The company has not been started primarily for the profit purposes, but mostly, to have it 'just for fun', or rather, to have a hobby you can to turn to.

3. Digital Business & Supply Chain Management

This chapter of the report will go further into detail about the Digital Business side of the company, is supply chain and how it is managed and how to find out which products are the most profitable and why this is important to know.

Digital business involves every aspect of the business that is managed through digital technologies and media, for example optimising and improving competitiveness and internal processes with online and traditional channels to market and supply (Chaffey, 2015, p. 15). Vengeful Games is in its essence an online game store, which means the E-Commerce side of the digital business will be the main focus of this section. Dave Chaffey (2015, p. 13) describes e-commerce as "all electronically mediated information exchanges between an organisation and its external stakeholders".

3.1. E-Commerce

This section describes the E-Commerce environment around Vengeful Games and analyses the approach to be taken for the business to flourish in the area. In this day and age it is impossible, and inadvisable, for companies to isolate themselves entirely from the internet if they seek any meaningful success. This, and the low entry cost of starting an internet Pureplay business, encourages many entrepreneurs to create internet start-ups as a way to follow their passion with relative low risks, such as Vengeful Games did, compared to starting a traditional '**Brick-And-Mortar**' business. This makes the E-business area very competitive, and in turn makes a good E-commerce strategy paramount.

Vengeful games currently uses an array of different programs to manage the store. This ranges from the Content Management System which is integrated into the hosting package from the webstore hosting company (Hostedshop), the use of additional software to buy and create packing labels (Pakkelabel.dk), and another program which creates the invoices(Billy) and crawls the data on his own website for the most recent information about the ordered goods(StoreBuddy). While this combination of software packages fulfils the company's needs, it also comes with some drawbacks. For everyone piece of software and support, the company will have to purchase a license, which can be easily avoided by using one over encompassing system. The scope of this project will however only focus on the most pressing need, which is to be able to record orders and combine them in any which way possible, as mentioned in the Introduction.

At this time, Vengeful Games is comfortable with their position in the market but wishes to make the administrative tasks less time consuming and focus more on the products and customer relationship. To fulfil the demands of the company, meetings between Vengeful Games and the system developers have been planned to keep a clear goal and direction of the project, so as to minimize errors, miscommunication and updated requirements. To accomplish this, it is recommended to follow a **Prescriptive Strategy**, to keep a clear sequence of events and go through the strategy with more clear bullet points rather following the more dynamic approach of the **Emergent Strategy** model.

This is described in the Business Case section in Chapter 7, Business Case.

3.2. Supply Chain Management

This chapter focuses on the Supply Chain Management (or SCM) which, according to (Chaffey, 2015, p. 252) can be described as follows: "The coordination of all supply activities of an organisation from its suppliers and partners to its customers". Below in Figure 4 is a visual representation of a very simple supply chain. Note that there can be many different intermediaries, and supplier and customer (depicted by (*x) in the figure). This all depends on the type of good that the chain deals with. More detail on this in how this effects Vengeful Games and the development of the program later in this chapter.

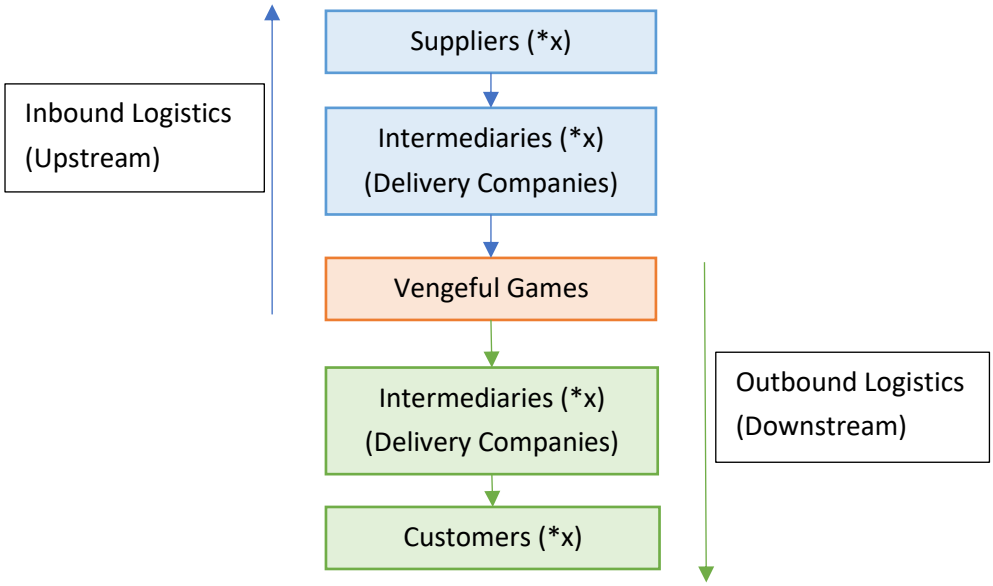


Figure 4: Supply Chain

3.2.1. Upstream Supply Chain

The upstream supply chain involves more than just the buying of goods, it involves all activities involved with the buy-side of e-commerce. In this section, the suppliers have been addressed first, followed by an overview of the intermediaries between the suppliers and the business.

Buying from suppliers

As mentioned earlier, the board gaming industry is a large multi-national market, which means the suppliers are also located all across the globe. The three most important locations however, apart from suppliers in Denmark, are suppliers and retailer in Germany, the United Kingdom and the United states. Often, the board games are directly purchased from the production company itself, other times from other retailers when they are for example on a sale that makes it worthwhile purchasing them.

Upstream Intermediaries

Inbound Logistics is the management of material resources which are entering an organisation from its suppliers and other partner (Chaffey, 2015, p. 263). In Vengeful Games' situation, this is first and foremost done by a group of different delivery companies, which take care of the delivery of the resources purchased from various suppliers and partners both nationally and internationally. Unfortunately, Vengeful Games is not in the position to select its own inbound intermediaries, even

though in some cases it would greatly increase the quality of the product. The only thing Vengeful Games can do, and already is doing, is advise the supplier about which intermediary to use, as some delivery companies don't handle the deliveries with enough care and shipment boxes get (badly) damaged. When this happens to a board game it instantly ruins the product and cannot be sold anymore. Below in Table 3 a brief overview of the companies used for inbound logistics.

Table 3: Inbound Intermediaries

Country	Preferred Intermediary	Back-up Intermediary	Reason
Denmark	GLS, "PostNord"	UPS	
Germany	GLS DPD	UPS	
United Kingdom	Royal Mail → PostNord	UPS	
United States	Fedex, USPS	UPS	
Other	PostNord		

Other inbound logistics that need to be managed are for example monetary resources, and payment authorizations coming in from intermediaries like the bank or PayPal.

3.2.2. The position of the company

The company itself has a large amount of influence over the supply chain itself. They can chose different suppliers or intermediaries, or even advertise differently to attract different kinds of customers. One of the more important decisions the company has to make is if they want to '**Push**' the products of their choosing onto the market, or have the market '**Pull**' the products for them. For Vengeful Games, the company primarily relies on a market pull. One of Vengeful Games' **Unique Selling Points** is that customers can pre-order a game which is either currently not in stock or not release onto the market by the suppliers yet, a feature which is currently not offered by its competitors. This gives Vengeful Games the *unique opportunity* of attracting a specific type of customer; *the game collector*.

3.2.3. Downstream Supply Chain

The downstream supply chain involves more than just the selling of goods, it involves all activities involved with the sell-side of e-commerce. In this section, the customers have been addressed first, followed by an overview of the intermediaries between the customer and the business.

Downstream Intermediaries

When following resources from the business in the direction of the customer, the first part of the downstream supply chain which has to be managed is the intermediaries. This management of resources supplied from an organisation to its customers and intermediaries is also called the **Outbound Logistics,** (Chaffey, 2015, p. 263). In this case, by far the largest part of intermediaries are the delivery companies taking care of shipping orders from the company to the customers. As with the upstream, there are a number of different delivery companies supplying the orders, the reasons for choosing the a company is almost the same as the upstream: 1: delivery company operation location, 2: delicate handling of orders, but there is also a 3rd reason: it has to be affordable (and thus

attractive) for the customer. Below in Table 4 is the overview of the different delivery companies used for shipping orders.

Table 4: Outbound Intermediaries

Country	Preferred Intermediary	Back-up Intermediary	Reason
Denmark	GLS	OK/ DAO 365	during holidays
Europe	GLS	OK	during holidays
Norway	Postnord/ Bring		Cheapest
United States	Postnord		Cheapest
Other	Postnord		Cheapest

Other outbound logistics that need to be managed are for example monetary resources, and refunds going to intermediaries like the bank or PayPal.

Selling to customers

The final stage of the downstream supply chain is the customers. At this moment in time a full customer analysis has not been made, nor has there been issued a customer satisfaction survey. There are a few points however which Vengeful Games itself advertises to the customers which are aimed at increasing satisfaction and attractiveness.

The first is the goal of delivering a high quality product. This is also the primary reason why, as mentioned in the upstream- and downstream intermediaries, the choice is made for slightly more expensive delivery companies instead of the cheapest option. High quality products results into more satisfied customers.

Vengeful Games also offers services aimed at customer satisfaction, like the ability to request refunds, or purchase gift cards. Other services which are unique to Vengeful Games are for example pre-purchases and reservations.

The last part to be discussed here is the delicate issue of customer complaints. Vengeful Games is approachable through various different forms of social media as well as direct contact through phone or e-mail. Complaints are logged and if possible handled in a way which is most beneficial to the customer.

3.3. ABC Products and logistical analysis

In 1896, Italian economist Vilfredo Pareto stated his famed '80/20' law. Later known as simply Pareto's principle states that 20% is responsible for 80% of the outcome. This principle can be applied in many different situations and still holds true today. If applied in business terms for example, 20% of the sales are responsible for 80% of the profit, and that 80% of the sales come from 20% of the clients/ customers (Kondylis, Logistics and SCM, 2018).

Using this 80/20 principle it is possible to set up a so called 'ABC-analysis' which analyses the products purchased and ranks them according to the amount of profit they contribute to the company. Once this is clear it is also possible to optimise the logistics by calculating the most profitable way of ordering

and storing products. In Vengeful Games' case; product storage takes up no extra costs as this is done in a small secure warehouse directly as the owner's address. However, should the company continue to grow, it will be good to know these optimization values.

As of yet it is impossible to make the above mentioned analysis however, mainly due to the fact that statistical analysis is one of the features that have yet to be implemented into the system.

4. Stakeholder Analysis

This chapter focusses on the stakeholder analysis. A stakeholder analysis will paint a clear picture of all the stakeholders involved in the project and helps the development team to understand the various parties in the project (Boddy & Buchanan, 1992, pp. 55-60). Stakeholders are people, businesses and organization with an interest in the project. There are many reasons why one has a stake in the project, Table 5 on the next page displays a clear overview of the different stakeholders that have been identified and what their role is in the project, their goals and interests, past reactions and expected behaviour, how they are defined and what can be done to motivate them.

Table 5: Stakeholder Analysis Table

Stakeholder	Goals and Interests	Past Reactions	Behavior Expected	Power, Legitimacy and Urgency	Ideas for Action
Thomas Edvard Mikkelborg Calix	Run the business as efficiently as possible while still having fun	Hard working but with an annoyance that the administration takes too much work	Supporting and welcoming of changes in the IT system	Definitive Stakeholder	Involve in development process
Thomas' Family	See the business flourish, while also having family-time with Thomas	Annoyance that administration takes up a lot of time	Happy with changes	Definitive Stakeholder	Keep informed
Customers	Buying products	Happy with good delivery and quality products	Could complain if deliveries take longer	Dangerous Stakeholder	Inform if needed
Suppliers	Selling products	*Unknown*	None	Dangerous Stakeholder	No action needed
Intermediaries	Up and downstream connections	*Unknown*	None	Dangerous Stakeholder	No action needed

Stakeholder Map

The stakeholder map represents the company in the centre and the stakeholders that have an interest in this company around it. The more relevant a stakeholder is, the closer the centre they are. This is shown in Figure 5. Additionally, the internal stakeholders are on the left, and the external stakeholders on the right. The owner and his family are of equal distance to the company, the customer are closer to the company than the suppliers and intermediaries. The customers won't buy if they are not happy.

Figure 5: Stakeholder map

Relationship Map

Figure 6 gives an overview of how the stakeholders are related and how they can affect each other. A new management system will the most biggest effect on the top half of this map, the internal stakeholders. As mentioned before in this report, the new program aims to greatly reduce the workload of processing orders and increase the ability to analyse the products and sales. In the bottom half of the map, only the Customer can notice any effect of this change. A new system takes some time to learn, which means that immediately after deployment there might be a slight slowdown in order processing times. However the manual provided should aid the transition and there are no expected long term effects for these stakeholders.

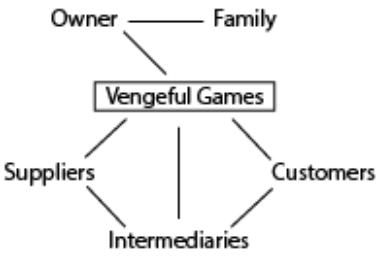


Figure 6: Relationship Map

5. Porter's 5 forces

Porter's Five Forces is a framework for analysing business competition. The theory it is based on dictates that these five forces are used to measure the level of competition for a given company and how that competition may positively or negatively affect its profit (Tang, 2014).

Supplier Power

In general products can be purchased from multiple suppliers in wholesale and encourages competition between supplier, which makes it easy to find new and thereby pressure suppliers to lower prices and/or offer better services. Some products are only bought from the manufacturers themselves, which means that there is only one supplier for certain product, in some cases very valuable product, which gives the supplier an enormous amount of power over the distribution of the product and thereby over retailers like Vengeful Games.

Threat of New Entry

As is the case with Vengeful Games, it is possible to enter this market as a hobbyist and not have to rely on the business as a source of income, which means that the threshold of entry is very low and new competitors can potentially appear relatively quickly and without much warning.

Buyer Power

This market is niche, which makes the customer base small. This means that the company must ensure loyal customers and potentially cast a geographically wider net. Additionally, Vengeful Games is a retailer, which makes the transition costs for customers very low, therefore it is more difficult retain

customers. Replenishing the customer pool is quite difficult for Vengeful Games, as it, in addition to being in a niche market, is a small company and therefore have limited finances for marketing.

Threat of Substitution

The threat of substitution is high, as the company does not provide any unique service or manufacture any products, which means it is easy for customers to find other retailers with the same capabilities.

Competitive Rivalry

There are relatively few board game retailers in this market, but the competition is still tough, as the playing field is very even for all companies in this market. Companies often must buy their products from the same suppliers and compete for the same customers. This means that, most of the strategic decisions should be made with customers service in mind, as well as lowering costs as much as possible.

6. SWOT Analysis

A SWOT analysis identifies the Strengths, Weaknesses, Opportunities and Threats of a company which allows for the development of a strong business strategy. There are internal and external factors that influence the company. The internal attributes are strengths and weaknesses, which can be changed. Opportunities and threats are external to the company, the company cannot easily change them (Bloisi, Cook, & Hunsaker, 2006, pp. 86-87). Figure 7 gives a clean visual representation of the identified strengths, weaknesses, opportunities and threats.

Strengths

One of the core strengths of the company is that it is easily managed. Since the CEO is also the only employee, any changes to the company can immediately be implemented and **Change Management** is a negligible aspect of the company management. The new IT system will also be a great help in managing the orders to company receives.

Having a one-man company requires no internal communication. This strength is a direct result of the previous one. By not having additional employees to worry about, it is possible to focus primarily on the tasks of fulfilling orders and stakeholder relations, instead of handling employee-related business. As a reference: the pros and cons of expansion have discussed in Chapter 1.

Another direct advantage of being a one-man business, which is specific to Denmark, is that the company's financial books are not open to being viewed by the public (and thus its competitors).

The way Vengeful Games operates has the added benefit of making it easy to keep the stakeholders satisfied. Vengeful games offers not only the best prices on the Danish market (which has even resulted in some customers from abroad), but also offers separate order shipment, which is especially attractive as Vengeful Games is the only company which offers this.

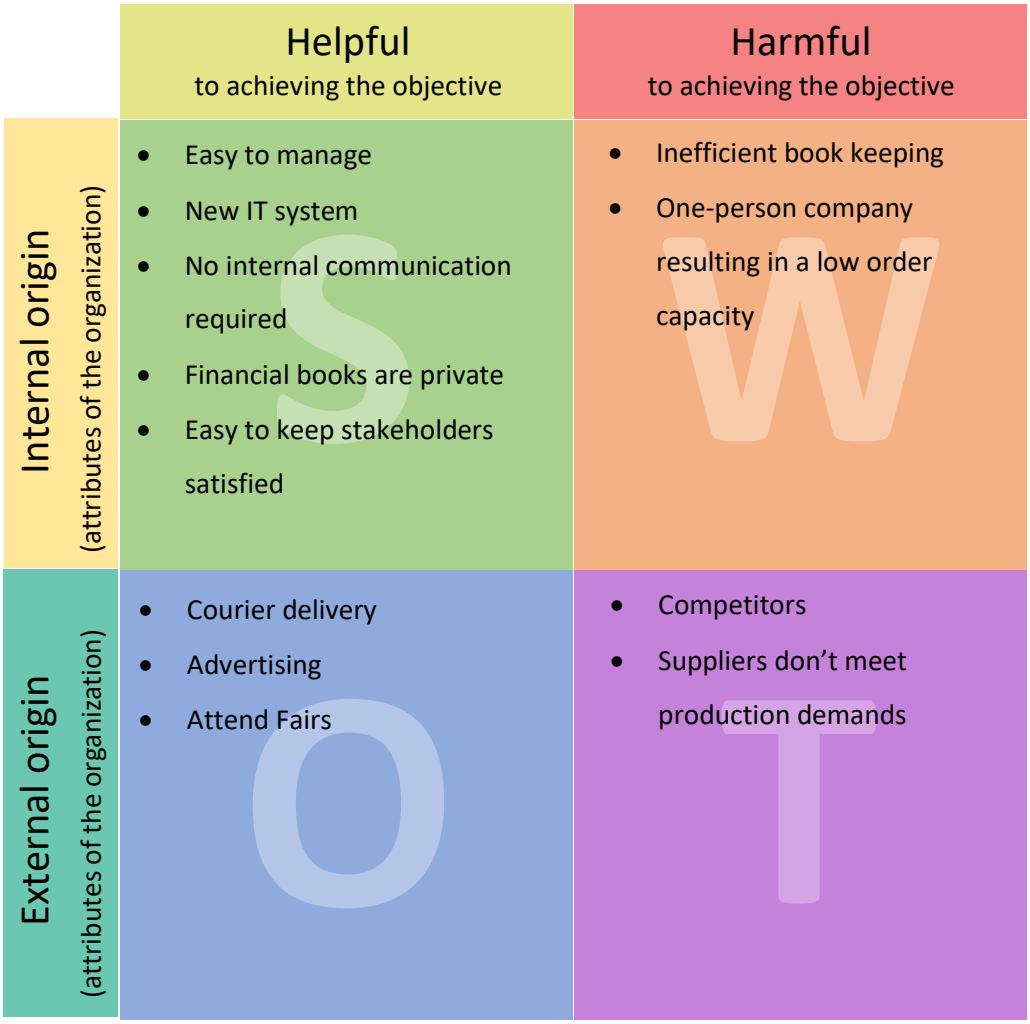


Figure 7: SWOT Analysis

Weaknesses

One of the problems the company faces is inefficient book keeping, currently this is due to the software being used. This project aims to change this weakness into a strength by fully supporting the separate order shipping mentioned above.

Being a one-man company, and in part due to the inefficient software system, has the unfortunate downside of having a 'cap' on the amount of orders which are realistically possible to be handled over a period of time. This cap is raised with the new system, however two people can still do more work than one, so hiring additional staff is a way to resolve this weakness entirely even though it would also mean that the company changes from being a hobby business to being a fully fledge business for profit.

A lot of these weaknesses are compensated for by the strengths of the company.

Opportunities

Private courier delivery, from supplier to the company, could increase profit as it guarantees better upstream control and less product damage and less time consumed replacing damaged products. It can, however also end up costing more resources for the company to utilise courier services.

There are also two things readily available which the company can take advantage of; one is online and in store advertising to draw in more customers. The other is attending fairs which focus on fun and play.

Threats

One of the biggest external threats are of course the competitors. As mentioned before, it is easy to start a new Internet Pureplay (or Click-Only) store and be a board game retailer. The need to stand out from the crowd is high and while Vengeful Games currently has a unique selling point, other competitors might pick this up as well, which could end very badly (possible bankruptcy) should the company not act upon this appropriately.

Another threat comes from the Suppliers, specifically when suppliers are unable to meet the demands from customers and provide a large amount of a particular board game. Leading to angry customers, whom often take out their frustrations to retailers, like Vengeful Games

7. Business Case

This section describes the business case of Vengeful Games in relation to implementing a new IT-system by examining the available options, conducting a cost-benefit analysis and then determining the impact and risks of the implementation.

Introduction & background

The goal of this report is to clarify purpose of implementing a new IT-system for Vengeful Games and the potential pitfalls thereof. The current system is a web-based PHP system, which lacks some critical feature demands of the company, as it is very inefficient for the company to use. This justifies an exploration of possible redesigns, which can facilitate growth and efficiency.

Summary

The current system does not lack any fundamental features, but has significant problems regarding inventory control, bookkeeping and statistics.

The purpose of this project is to address these issues by integrating a new IT-system containing improvements of existing inventory and bookkeeping features, as well as implementing statistics, including graphs of significant metrics within the company.

Improvements could be made to the existing system, but as it is implemented using PHP, which has limitations in working with SQL (The PHP Group, n.d.), the group has concluded that the best course of action is to build an entirely new system.

Description of problems & opportunities

Inventory control: The current inventory system does not allow for an easy overview of items and orders, which makes it difficult to track stock and shipping.

Bookkeeping: The current system does not have an intuitive search feature and does not facilitate grouping of orders, which means it can be difficult to get a clear perspective of the state of a single

customer with multiple orders. The current system also lacks significant statistics regarding sales trends and contribution margins of products.

Usability: The issues raised above makes the system very frustrating and tiresome to work with for the owner of Vengeful Games, which can take a significant toll on him and his family. Therefore, the new system must, other than implement the requested features, be user-friendly and intuitive.

Available options & consideration

The current system is still very much usable and could be for a long time, but the cost would keep increasing in terms of manhours, human error and frustrations.

Employees could be hired to deal with the logistical issues, but this a significant new expense for a one-person company, in addition, this only postpones the problems to a, perhaps, critical time in the company, which should be avoided when possible. The company could also keep products in stock regardless of the contribution margin of any product in particular. This would eliminate the need to track sales and purchase statistics, but this is not a sustainable solution, as the company would be oblivious to any non-profitable products.

The recommended approach is to implement an entirely new IT-system, as this will address the problems at a more fundamental level. Additionally, this allows the new and improved features to be integrated more smoothly into the system, as there will not be any problems regarding compatibility, which would undoubtedly be the case when improving the current system.

Financial implications

The cost-benefit analysis aims to get to the core of the pros and cons involved in the recommended solution before committing fully to the project. This is done to determine, if the project should be carried out before committing too many resources to the project. Table 6 below shows that there are more benefits than costs and that the benefits have a positive impact on many aspects of the company.

Table 6: Cost-Benefit Analysis

Cost		**Benefit**	
Tangible	*Intangible*	*Tangible*	*Intangible*
Learning period (Time)	Frustration in the learning period (Happiness)	Less time consuming administrative work (Time)	Less frustrations about missing or unintuitive features (Happiness)
Software maintenance (Money)		Less reliance on 3rd party software and services (Money)	Better overview and actionability (Knowledge)
			Less uncertainty of stock and orders (Peace of mind)

A cost-benefit analysis is a good starting point, but it is not enough to justify the large investment of implementing a new IT system, as it does not quantify the investment or the savings. For this, a payback projection has been made to determine, the investment required to complete the project and the projected savings per year for the company as a whole. This is done to analyse the balance between the investment and the savings and estimate the time frame for the project to become profitable. As can be seen in Table 7 below, the project is projected to have a positive return on investment within 5 years.

Table 7: 5-year Payback Projection

Item	Year 1	Year 2	Year 3	Year 4	Year 5
Software Purchase	100,000				
Software Maintenance	25,000	25,000	25,000		
IT Training	20,000				
Cumulative total costs	145,000	170,000	195,000	195,000	195,000
Administrative time saved	20,000	20,000	20,000	20,000	20,000
Obsolete 3rd party licences	20,000	20,000	20,000	20,000	20,000
Cumulative savings	40,000	80,000	120,000	160,000	200,000

Impart & risks

The implementation of a new IT-system has a tremendous impact on any company, as almost everything is connected to IT-systems in the modern world. This is especially true for companies like Vengeful Games, as this is an internet pureplay retailer. Therefore, the frustrations of an inadequate IT-system are amplified as everything is done through the IT-system. This, of course, also increases the severity of the risks associated with the implementation of an entirely new system, which means missing functionalities, poor usability and bugs can have an enormous negative impact on the company. This making testing and debugging a very important part of the development of the system.

Conclusions & recommendations

The conclusion of this analysis, is that it is advisable to go ahead with this project, as the many frustrations and lack of clarity are a significant burden on the company and will continue to be so in the future. This can also become more of a problem if the company expands. Therefore, the recommendation is to implement a new IT-system, to avoid any limitations and compatibility issues of the current system.

PART II

Methodology

This page is intentionally left blank.

8. Methodology

"A software development methodology is a way of managing a software development project." (Young, David C.; Computer Sciences Corporation, 2013, p. 1). Choosing the right methodology is one of the most important and crucial decisions to consider before a new system starts being developed. This decision also differs from system to system, from project to project, according to its complexity, size or other factors.

In this chapter, some of the system methodologies are continuously introduced, followed by a brief description and comparison. Below, there is also an explained decision which methodology was used for development of the system for Vengeful Games.

8.1. The types of methodologies

A waterfall process (methodology) is considered as an older and traditional method of software development where the emphasis is given on gathering all the requirements of the system before its actual implementation and testing. This process is usually used for creating systems where all the requirements are straightforwardly defined from the beginning of the lifecycle process.

A Unified Process (UP) is a software development process that describes an approach to building, deploying, and possibly maintaining software (Larman, 2004, p. 18). This process uses an iterative and evolutionary approach to get better understanding of what user really needs in form of getting an early feedback and adapting the changes to the system in a shorter timeframe.

Agile modelling is a software methodology where requirements are meant to be changed very often. "The purpose of modelling and models is primarily to support understanding and communication, not documentation." (Larman, 2004, p. 30)This means that creating many detailed UML diagrams is not necessary, but the diagrams should rather be helpful to the developer in exploring alternatives to a good design.

8.2. Brief comparison

The waterfall uses clear structure, steps are predefined and followed in order. Each project has requirements gathering phase, system analysis and design, implementation and then, testing, followed by deployment of a complete product (Lucidchart Content Team, 2017). This could be seen as a huge advantage in terms of saving some time, but on the other hand, straightforwardly work on the system without getting any feedback may be dangerous, nowadays. People often do not even know what they want, until they see what developer delivers, which might not always be what they were looking for. "Success/failure studies show that the waterfall is strongly associated with the highest failure rates for software projects." (Larman, 2004, p. 18).

To contrast, research shows that iterative and evolutionary development is associated with higher success and productivity rates. It is precisely because of the quick development steps, feedback and its adaptation to the system. Unified process includes working in the iterations, so complexity can be managed more easily when the systems are divided into smaller parts.

The waterfall and iterative approach are, however, very similar. Unified process, basically, takes parts of the waterfall method and makes it iterative, allowing for changes. This is the reason why it is very common, while switching from waterfall into the iterative approach, some developers tend to fall into the waterfall, defining all the requirements at the first place and then implement it.

In comparison to the two methodologies mentioned before, the agile methods are more flexible. The Agile Manifesto says: “Individuals and interactions, over processes and tools. Working software, over comprehensive documentation. Customer collaboration, over contract negotiation. Responding to change, over following a plan.” (Larman, 2004, p. 29).

Agile methodology uses an adaptive approach that responds to changes favourably. All members of the team are usually involved in all aspects of planning, implementation, and testing. Therefore, the agile projects are developed in smaller teams, where daily face-to-face meetings are possible to manage.

From the other point of view, since the agile approach puts so much focus on the developing the software, it lacks documentation. That brings other disadvantage, when let’s say, a new developer joins a team but because of the lack of documentation, his adaptation to the project is a lot harder in comparison to Unified Process that is well documented.

8.3. Methodology used in Vengeful Games software

Since the inseparable part of a software creation for Vengeful Games is the feedback of the user and its adaptation, Unified Process has been used as a system methodology. Not only to see early progress, but also to learn within an iteration so improvement in the development process can be done.

8.4. Unified Process (UP)

As mentioned earlier in this chapter, Unified Process is a software development methodology that uses iterations. It is best described in Figure 8 on the next page. There are nine iterations within four phases (Inception, Elaboration, Construction and Transition). Each of the iteration should last from two to six weeks. It should not last less because it is hardly possible to get requirements and implement them within one-week iteration. At the same time, it should not last more than six weeks, because customer needs to see progress as soon as possible. It should be noted however that due to time restrictions and project demands, each iteration lasted only one week.

On the left side of the diagram, there are six disciplines that are used in UP, namely, Business Modelling which largest part is done in the Inception and Elaboration phases. Requirements are collected and processed in the Elaboration. The Analysis & Design are made in the Elaboration phase but continuously revised in the Construction as the Implementation of code is done. Tests should be made while implementing the code or even at earlier stage. In the end, the Deployment of final product is done in the Transition phase.

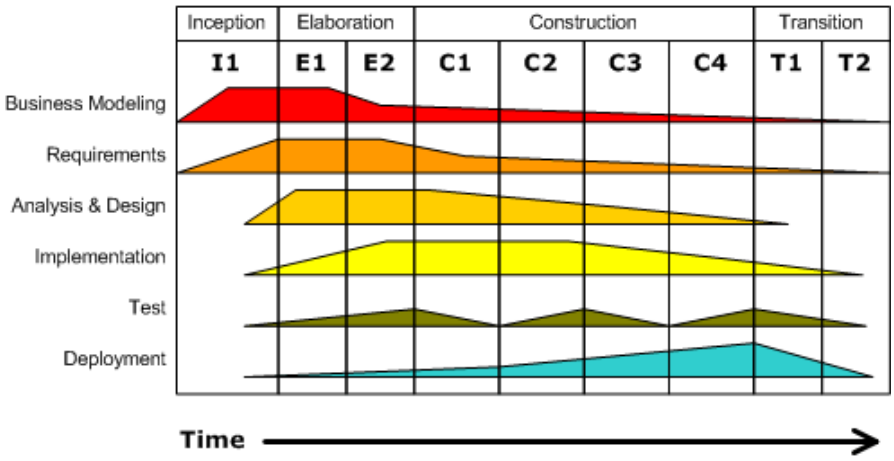


Figure 8: Phases of the Unified Process

Requiring the team to focus on the most risky and critical part of the system in the beginning of each iteration, Unified Process is risk-driven. UP is also use case-driven, thanks to the use cases describing the functionalities of the system. Use cases are responsible for planning and controlling the progress of the development. "The architecture expresses the common vision of the system." (Kondylis, Recap of System Development (2nd Sem.), 2018, p. 6). UP is architecture-centric, using a different kind of models, for example, use case model, domain model, design models. Because of the Elaboration, Construction and Transition being divided into series of time-boxed iterations, Unified Process is iterative. Each of these iterations should result in the system getting larger, gaining more functionalities and improving the old ones. This is the reason why, one of the characteristics of Unified Process is, being incremental.

From this point onwards, the report is split into the four phases of UP, specifically, Inception, consisting of few requirements of the system. The rest of the requirements were gathered during the Elaboration phase, where the design and analysis were made. In the Construction phase, there is a part about code standards, architecture and implementation. Last, but not least, the Transition part of the system includes the manual for the Vengeful Games (Kondylis, Recap of System Development (2nd Sem.), 2018, pp. 3-8).

This page is intentionally left blank. .

PART III

Unified Process

This page is intentionally left blank.

9. Inception

This chapter covers the inception phase of the unified process and covers the first iteration into defining the project. This includes the Activity Diagram, Use Case Diagram and descriptions and the System Vision.

9.1. Activity Diagram

This section analyses the tasks of the owner in further detail by examining a critical task's workflow in an activity diagram, to determine the steps involved and identify the required components.

9.1.1. Create Order

The activity diagram below in Figure 9 describes the workflow for creating an order in the system in order to get an idea of which steps are crucial for an employee to create an order.

It quickly becomes apparent that the create order use case involves a lot more than just the 'create' step from a **CRUD** use case. This use case is complex because it involves many different parts of the system, such as creating the order, searching for items in the database or adding them to the database if needed and automatically setting an ordered status to the order. The order has to also be associated with a customer. Once all this is done a discount will be applied and the total is calculated.

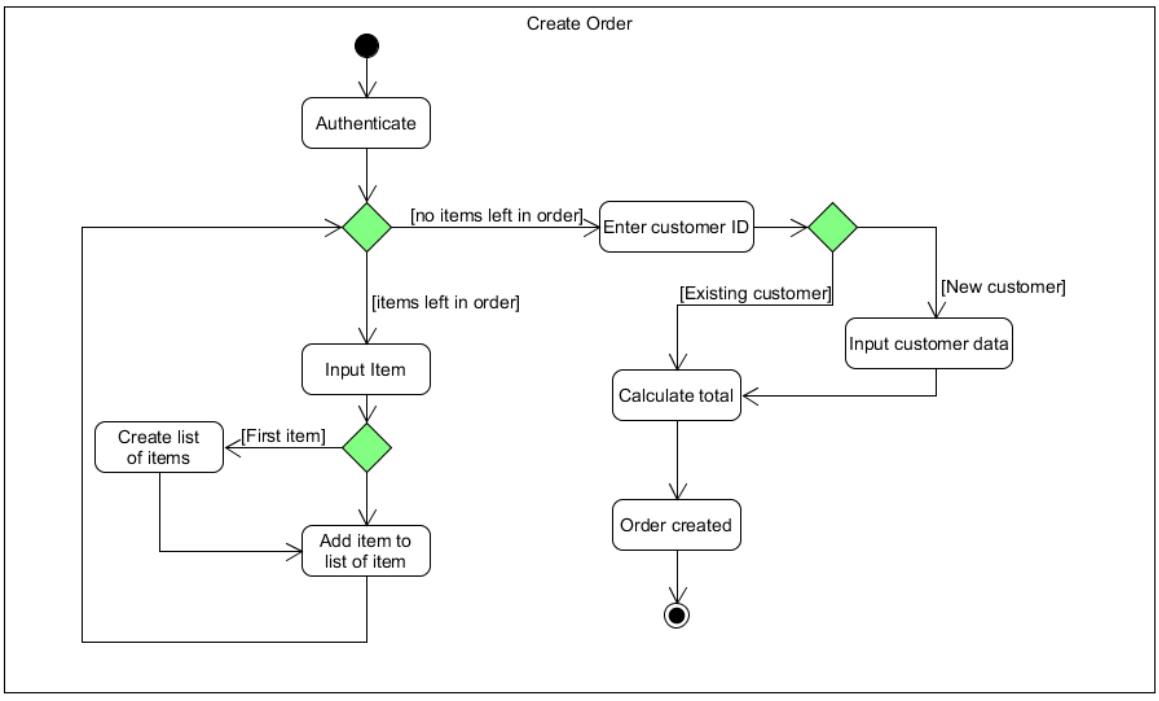


Figure 9: Use-Case - Create Order

9.1.2. Update Order Status

The activity diagram below in Figure 10 describes the workflow for the Update Order Status use case.

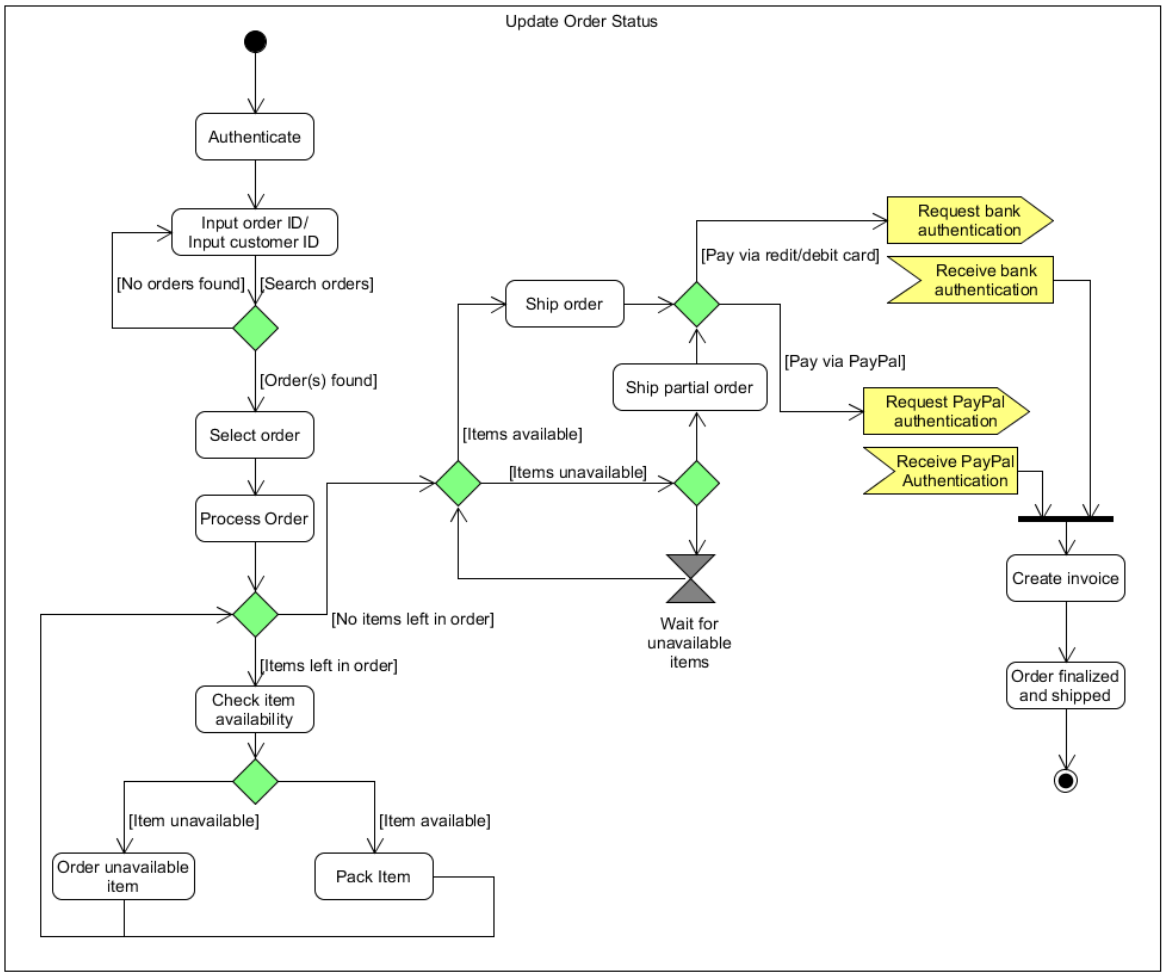


Figure 10: Use-Case - Update Order Status

This is initiated by the employee by authenticating themselves into the system and searching for an order or customer and selecting an order. First the availability of the items in the order must be checked, to determine whether the order should be shipped. If some items are out of stock, a decision must be made on whether a partial order should be sent or put on standby until all items can be shipped. When shipping, a payment method must be chosen and authenticated with the appropriate external systems (for example bank transfer or PayPal). Finally, an invoice is created, and the order is finalized.

9.2. Employee-Task-Goal-Table

This section focusses on the tasks the employee has to do and which steps they go through to achieve the final goal of said task. This is organised in Table 8 below.

Table 8: Employee-Task-Goal-Table

Actor	Task	Goal	Steps in task
Vengeful Games Employee	Register Customer	Customer Registered	Authenticate Input customer data Enrol to the database of regular customer
	Register Item	Item Registered	Authenticate Input item data Enrol to the database of regular item
	Create Order	Order Created	Authenticate Input customer id Input item id Add item to order
	Update Order Status	Order Status Updated	Authenticate Get order id Set order status
	Group Orders	Orders Grouped	Authenticate Get customer id Edit orders Add orders to grouped orders
	Get Customer Statistics	Identify Valuable Customers	Authenticate Get customer list Sort by desired parameter
	Get Item Statistics	Identify Valuable	Authenticate Get item list Sort by desired parameter

9.3. Use-Cases

This paragraph focusses on identifying the use cases and describing them. A use-case can be defined as “a collection of related success and failure scenarios that describe an actor using a system to support a goal” (Larman, 2004, p. 63)

9.3.1. Use-Case Diagram

The diagram below in Figure 11 is the use-case diagram and shows both, the functionality of the system and the communication of an actor with the system.

Due to the better and simpler overview of all the use-cases of the system, some of them are summarized in the use-case named CRUD.

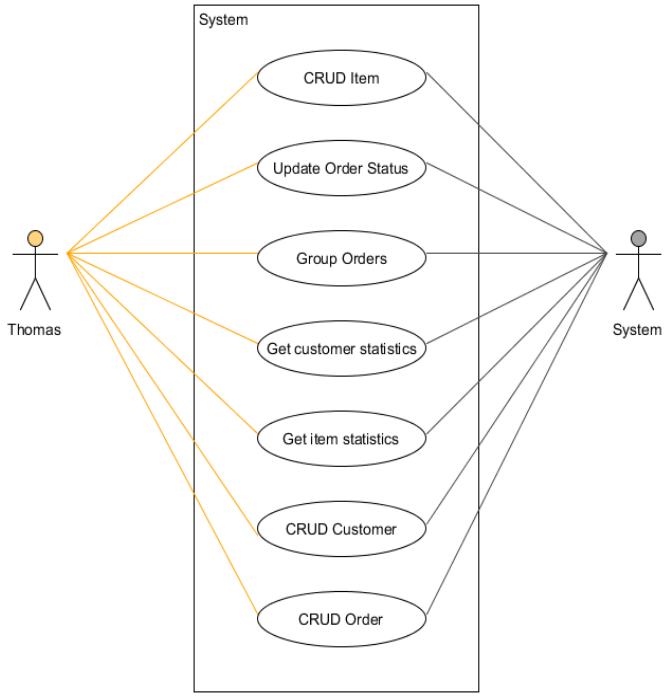


Figure 11: Use-Case Diagram

9.3.2. Use-Case Descriptions

In this paragraph, the use-cases will be continuously described in a brief format and prioritized according to the **Criticality**, **Complexity** and the **Coverage**. The system risk analysis in Table 9 on the next page assigns each use case a score between 1-5 for both all three factors, The higher the score the higher the priority in implementing the use-case. Each iteration strives to implement the next one in the list.

The use-cases with the highest priority are analysed and implemented in the first iterations, since it is the most complex and critical part of the system that needs to be covered fully.

Table 9: System Risk Analysis

Use-Case	Criticality	Complexity	Coverage	Total
Create Item – CRUD Item	5	3	1	9
Update Order Status	5	5	4	**14**
Group orders	4	5	4	13
Get customer statistics	2	3	2	7
Get item statistics	2	3	2	7
Create Customer – CRUD Customer	5	3	1	9
Create Order – CRUD Order	5	5	5	**15**

Create Item – CRUD Item

After the actor authenticates and enters the system, he inputs the data of the item that is about to be registered. Inputting all the data correctly, enrolment of the item to the database is done.

Update Order Status

The actor authenticates in the system and searches for the order using either the id of the order or in the menu according to the last status of that order. After the results are found, he sets new status of the order. This is repeated couple of times since the order is placed, till it is delivered to its final destination. For the fully-dressed form, please refer to Chapter 10.1.3.

Group orders

This use-case can only be done by Vengeful Games, so a customer himself is not able to group his orders. After authentication of the employee (Thomas), he searches for the customer's id and gets all the information about the customer. He chooses the option of editing the orders and adds orders to the grouped orders list of the desired customer.

Get customer statistics

The actor authenticates himself in the system and chooses to see the statistics of the customers. There is a customer list displayed on the screen sorted by a specific default parameter. From this point on, the actor chooses different kind of sorting of all the customer to get various information.

Get item statistics

The actor authenticates himself in the system and chooses to see the statistics of the items. There is an item list displayed on the screen sorted by a specific default parameter. From this point on, the actor chooses different kind of sorting for all the items to get various information.

Create Customer – CRUD Customer

After the actor authenticates and enters the system, he inputs the information of the customer that is about to be registered. Inputting all the data correctly, enrolment of the customer to the database is done.

Create Order – CRUD Order

This use-case is the most critical and complex one that needs to be fully covered. It also is the main feature of the system and thus, needs to be described properly, with all the basic and alternate flows. To see the fully-dressed description of this use-case, please refer to Chapter 10.1.1.

9.4. System Vision

Below are a few paragraphs which give a brief representation of the entire scope of the system. Starting with a short introduction with the purpose and scope, followed by 'where' the business and the product currently stands, and a summary or the system stakeholders.

9.4.1. Introduction

Purpose

The purpose of this document is to collect, analyse, and define high-level needs and features of the Bookkeeping and Inventory Control System (BICS). It focuses on the capabilities needed by the stakeholders, and the target users, and why these needs exist. The details of how the Bookkeeping and Inventory Control System fulfils these needs are detailed in the use-case and supplementary specifications.

Scope

This Vision Document applies to the Vengeful Gaming Inventory System (VGIS), which will be developed by the DMAI0917 class - Group 1 development team. The team will develop a new inventory system to ease the customer needs. The (VGIS) monitors, controls the wide variety of games such as board games, card games, dice games, family games, star wars and accessories. The system supports local access through a keyboard.

9.4.2. Positioning

Business Opportunity

Having a proper bookkeeping system is an inseparable part of being a retailer. The 'Vengeful Games Inventory' system includes creating an order, registering new customers and items, updating an order status and much more.

In addition to having all these necessary features, the Vengeful Games Inventory is much simpler for the user in form of not having three different systems, but rather have just one complete system for all the bookkeeping.

Problem Statement

The problem statement below in Table 10 is a brief version of the full problem statement found in Appendix C.

Table 10: Problem Statement

The Problem	The increasing administrative demands
Affects	Small businesses
A successful solution would be	A user friendly and efficiency driven UI

Product Position Statement

Below in Table 11 is the Product Position Statement which displays a more detailed overview of what the product should look like and what it should be able to do.

Table 11: Product Position Statement

For	Online board game retailer
Who	Needs to manage his day-to-day tasks more efficiently
The Bookkeeping and Inventory Control System (BICS)	Is a software product
That	Provides user-friendly administration support for web-based companies and encourages an efficient workflow for both managers and employees
Unlike	Current freemium solution, largely based on self-contained addon modules, which results in significant compatibility and collaboration difficulties between these modules
Our product	Aims to deliver a cohesive and comprehensive solution for small web-based businesses, so as to minimize compatibility issues and thereby reduce inefficiencies and frustrations at the workplace

9.4.3. Stakeholder Summary

Contrary to the stakeholder analysis in Chapter 4, Table 12 gives a short overview of the different stakeholders which have an interest in the final product of this project, the software system.

Table 12: Stakeholder Summary

Name	Description	Responsibilities
Thomas Edvard Mikkelborg Calix	Company owner, interested in obtaining a better alternative to the currently inadequate software being used.	Ensuring that his business succeeds by meeting his customers' needs
Dimitri Kondylis	Supervisor that provides guidance and criticism on the project's state	Aiding the development of the project and assure that it is fulfilling the necessary requirements
Project Group	development team employed by the [person] to produce adequate software for personal use.	Planning, coding, testing and iterating over the IT system in order to ensure the best possible quality of the product

Main Product Features

Below a list of the main features of the software system.

- Secure Log in
- Search the orders
- Select the order from order table
- Update order table
- Create new order
- Select customer from customer table
- Register new customer
- Add items to the order
- Remove items from the order
- Cancel new order
- Update order
- Change the status of the order
- Change the sorting in the order table

9.4.4. Constraints

In this section, non-functional requirements of the Vengeful Games Inventory are mentioned and shortly described below.

Usability

The system is user-friendly and very easy to use. It does not include unnecessary features according to request made by the user.

Reliability

The program should be reliable and ensure data persistence after every database transaction. Any unexpected errors or program crashes should not affect the data in the database. To achieve this the data can be committed to the database after every successful order creation.

Performance

Emphasis has been put on increasing performance of the system, as increase efficiency is one of the main priorities of the system

Supportability

To ensure the program is as maintainable and adaptable as possible, certain design choices have be take this into account. The architecture will have to support a flexible system and the code standards will have to support **High Cohesion** and **Low Coupling.**

Miscellaneous

Due to demands from the educational institute for which this project is written, it is only possible to write the program in Java and SQL, otherwise the language of choice would have been PHP which integrates fully with the already existing system. This constraint means that the program is a stand-alone program, this does however have the added benefit that additional features can be added later on, even to such an extent that the user will only need 1 software system instead of the many they use now. Which limits the amount of licences they need and thus saves on expenditures.

9.4.5. Mock-ups

Mock-ups of the system have been created to get better overview of the main features of the system and to get customer feedback. The main menu consists of the News, Orders, Customers, Items and the Statistics. After clicking any of these, a new "sub-menu" is displayed, as can be seen in Figure 12 when 'orders' has been clicked, it then displays all 'received orders'.

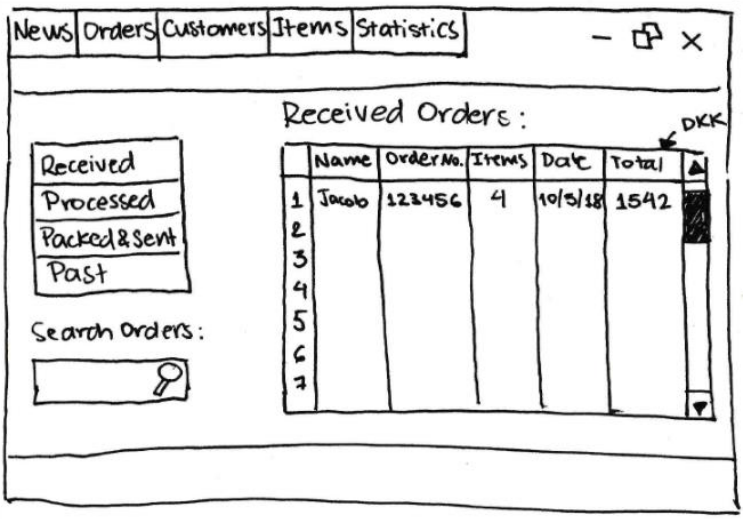


Figure 12: 'Received Orders' Mock-up

On the left side of the menu is a small "submenu" showing when clicked every order that has been received (new orders), or processed, packed & sent or all orders that have been already delivered (past).

Right under, there is a text field for searching, where the user can search for the order ID.

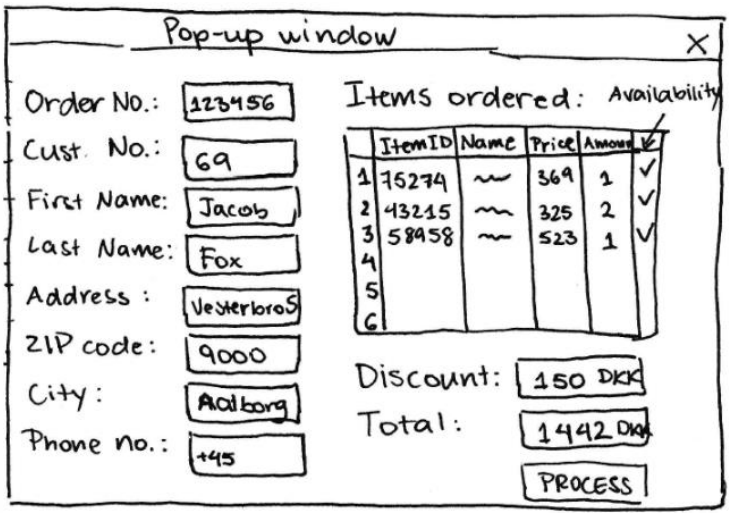


Figure 13: 'Process Order' Mock-up

On the right side, there is a table with all new orders that need to be processed. They are ordered according to a date of the order. They need to be processed as the first ones.

The second mock-up on the right side, Figure 13, shows the window that is popped-up when the user clicks and chooses to process the order. In the window there is all the information filled in, on the left side, and on the right side, there is a table of all the items of that specific order. At the end, in the right

corner, there is a discount calculated and the total amount to pay. When the 'process' button is clicked, the order is moved to 'Processed' orders and from that point, can be found it that menu.

The last mock-up in Figure 14 below is about 'Statistics' menu. This feature has been requested by the user of the system, Thomas. It is just for him to see how it is going with the regulars and which products are famous, and so on.

In this picture, there is small "sub-menu" on the left side, from which it is possible to choose either 'Customers' or 'Items' statistics. Right underneath, there is also the option to sort the statistics by different categories and again, on the right side, there is a table with the statistic information.

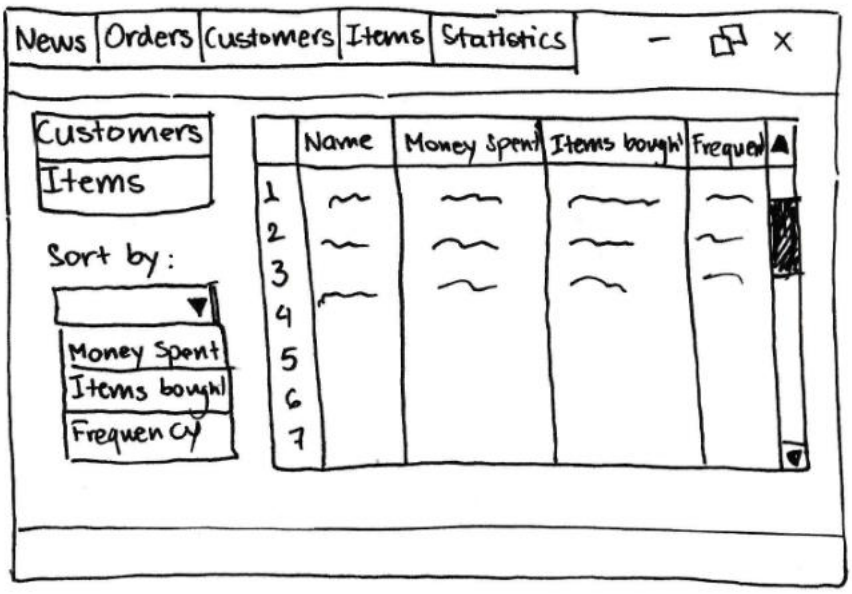


Figure 14: 'Statistics' Mock-up

10. Elaboration

This chapter elaborates on the most critical use cases, Create order and Update order and displays the fully dressed form of both and elaborates on the test cases. It also depicts the domain model, which is used to create the relational diagram for the database and goes into the Database and SQL scripts. Final part consists of Sequence diagrams and their operation contracts, interaction diagrams and the design class diagram.

10.1. Use-Cases Fully Dressed form and Test Cases

Below the two most critical use cases, Create Order and Update Order Status in fully dressed forms in Table 13 and Table 17 respectively, with their test cases. These test cases mainly test on a level of integration testing, however unit tests were also performed and a whole system test will be done

10.1.1. Use-Case 1: Create Order - Fully Dressed

Table 13: Fully Dressed Form: Create Order

Use Case Name	**Create Order**	
Scope	Create Order part of IT System	
Primary Actor	Thomas – Employee	
Secondary Actor	System	
Preconditions	The user (Thomas) has authenticated. The new order has been started.	
Postconditions	The order has been created.	
Frequency	5 per day	
Flow of events	**Actor**	**System**
Main success scenario	1. Inputs an item ID into the system.	2. Retrieves list of matching items.
	3. Selects item from the list.	4. Displays all the info about the item.
	5. Clicks 'add to order'.	6. Adds item to the order.
	Steps 1 – 6 are repeated until all the items are inserted into the order.	
	7. Enters customer ID.	8. Searches for the customer information.
		9. Ties the customer to the order.
		10. Calculates total with the discounts.
Alternate Flows	2. List of items is not retrieved. 1. Nothing is displayed. 2. Employee inputs an item ID again.	
	4. System cannot display item information. 1. System displays an error message. 2. Employee selects the item again.	
	6. The item is not in stock. 1. System displays a warning message. 2. System prompts the user to order additional items.	
	7. Customer not found. 7a. Customer ID is incorrect. 1. System displays an error message. 2. Employee enters the customer ID again. 3. [cont. main success scenario] 7b. Customer does not exit. 1. System displays an error message. 2. Employee creates a new customer.	
	8. Discount cannot be applied. 1. Employee enters a discount manually.	

10.1.2. Use-Case 1: Create Order – Test Cases

Below the test cases for Use-Case 1: Create Order. First the Scenario's identified in the Scenario Matrix, Then the Test Case validity table which displays what sort of data makes for a valid test, and last the Test Cases with Test data table, which shows examples of valid and invalid data.

The test cases are the sets of scenarios generated from the alternate scenarios of fully-dressed use-cases. At first, a scenario matrix, as can be seen in Table 14 below, was generated identifying each combination of main and alternate flows.

Table 14: Scenario Matric: Create Order

Scenario Name	Starting Flow	Alternative
Scenario 1 – successful order	Basic Flow	
Scenario 2 – list of items not retrieved	Basic Flow	A2
Scenario 3 – item info not displayed	Basic Flow	A4
Scenario 4 – item is not in stock	Basic Flow	A6
Scenario 5 – incorrect customer ID	Basic Flow	A7 – a
Scenario 6 – customer does not exist	Basic Flow	A7 – b
Scenario 7 – discount cannot be applied	Basic Flow	A8

Secondly, the test cases were identified below. The headers of the columns are steps of main success scenario of the fully-dressed use-case. Each test case has its ID and a scenario/condition. In Table 15 there are no actual data values but instead, a valid (V), an invalid (I) or not applicable (N/A) values used. In Table 16 validation was replaced for the actual data values the system has been tested with. These data values are identified for each of the test cases.

Test Case ID	Scenario/ Condition	Input ID	Get list of items	Select item	Display info about item	Add item to order	Enter customer ID	Search customer info	Tie customer to order	Calculate total + discount	Expected results
CO1	Scenario 1 – Successful order	V	V	V	V	V	V	V	V	V	Successful Order
CO2	Scenario 2 – List of items not retrieved	I	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Error message displayed on screen
CO3	Scenario 3 – Item info not displayed	V	V	N/A	I	N/A	N/A	N/A	N/A	N/A	Error message.
CO4	Scenario 4 – Item is not in stock	V	V	V	V	I	V or N/A (Order can continue)	V or N/A	V or N/A	V or N/A	Warning message displayed
CO5	Scenario 5 – Incorrect customer ID	V	V	V	V	V	I	N/A	N/A	N/A	Error message displayed; back to customer info screen
CO6	Scenario 6 – Customer does not exist	V	V	V	V	V	V	N/A	N/A	N/A	Error message displayed; register customer screen.
CO7	Scenario 7 – Discount cannot be applied	V	V	V	V	V	V	V	V	I	Error message.

Table 15: Test Case Identification Table: Create Order

Test Case ID	Scenario/ Condition	Input ID	Get list of items	Select item	Display info about item	Add item to order	Enter customer ID	Search customer info	Tie customer to order	Calculate total + discount	Expected results
CO1	Scenario 1 – Successful order	123	123456 781234 888123	123456	Pie Town, Good game, 242,25DKK	added	888888	Greg Bach Hasseris 12 9000 Aalborg	Customer Added	242,25DKK	Successful Order
CO2	Scenario 2 – List of items not retrieved	123	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Error message displayed on screen
CO3	Scenario 3 – Item info not displayed	123	123456 781234 888123	123456	Not displayed	N/A	N/A	N/A	N/A	N/A	Error message.
CO4	Scenario 4 – Item is not in stock	123	123456 781234 888123	123456	Pie Town, Good game, 242,25DKK	Not added	V or N/A (Order can continue)	V or N/A	V or N/A	V or N/A	Warning message displayed
CO5	Scenario 5 – Incorrect customer ID	123	123456 781234 888123	123456	Pie Town, Good game, 242,25DKK	V	888887	N/A	N/A	N/A	Error message displayed; back to customer info screen.
CO6	Scenario 6 – Customer does not exist	123	123456 781234 888123	123456	Pie Town, Good game, 242,25DKK	V	888855	N/A	N/A	N/A	Error message displayed; register customer screen.
CO7	Scenario 7 – Discount cannot be applied	123	123456 781234 888123	123456	Pie Town, Good game, 242,25DKK	V	888888	Greg Bach Hasseris 12 9000 Aalborg	Customer Added	285DKK	Error message.

Table 16: Test Case Data Table: Create Order

10.1.3. Use-Case 2: Update Order Status- Fully Dressed

Table 17: Fully Dressed Form: Update Order Status

Use Case Name	**Update Order Status**	
Scope	Update Order Status part of IT System	
Primary Actor	Thomas – Employee	
Secondary Actor	System	
Preconditions	The user (Thomas) has authenticated. The order has been created.	
Postconditions	The order has been shipped.	
Frequency	Approx. 10 times per day	
Flow of events	**Actor**	**System**
Main success scenario	1. Inputs either item ID or customer ID into the system.	2. Searches the orders.
	3. Selects the order.	
	4. Chooses to process the order.	
		5. Checks availability of the item.
	6. Packs the item.	
	Steps 5 - 6 are repeated until all the items are checked and packed.	
	7. Chooses to ship the order.	8. System asks for the payment method.
	9. Chooses the payments method.	
	10. Inserts the payment information.	11. Registers the information.
		12. Creates an invoice.
Alternate Flows	2. The order cannot be found. 1. System displays an error message. 2. The employee inputs the id again.	
	5a. Items are currently not in the stock. 1. System asks the user whether he wants to order missing items. 2a. Employee decides to order missing items. 1. System asks whether to send partial order. 2aa. Employee decides to send partial order. 1. Ships the order. 2. [cont. happy day scenario] 2ab. Employee decides not to send partial order. 1. Waits until all the items are available. 2. Ships the order. 3. [cont. happy day scenario] 2b. Employee decides not to order missing items. 1. Cancels the order.	

	5b. Not all the items have been released. 1. Employee sends a partial order. 2. System marks items in the order as not released. 3. Not released items are sent after the release date.
	10. Payment information is incorrect. 1. System displays an error message. 2. Employee inputs the information again.
	12. An invoice cannot be created. 1. Employee tries again.

10.1.4. Use-Case 2: Update Order Status– Test Cases

Below the test cases for Use-Case 2:Update Order Status. First the Scenario's identified in the Scenario Matrix, Then the Test Case validity table which displays what sort of data makes for a valid test, and last the Test Cases with Test data table, which shows examples of valid and invalid data.

The test cases are the sets of scenarios generated from the alternate scenarios of fully-dressed use-cases. At first, a scenario matrix, as can be seen in Table 18 below, was generated identifying each combination of main and alternate flows.

Table 18: Scenario Matrix: Update Order Status

Scenario Name	Starting Flow	Alternative
Scenario 1 – successful status update	Basic Flow	
Scenario 2 – order not found	Basic Flow	A2
Scenario 3 – item not in stock	Basic Flow	A5 – a
Scenario 4 – item not released	Basic Flow	A5 – b
Scenario 5 – incorrect payment info	Basic Flow	A10
Scenario 6 – not created invoice	Basic Flow	A12

Secondly, the test cases were identified below. The headers of the columns are steps of main success scenario of the fully-dressed use-case. Each test case has its ID and a scenario/condition. In Table 19 there are no actual data values but instead, a valid (V), an invalid (I) or not applicable (N/A) values used. In Table 20 validation was replaced for the actual data values the system has been tested with. These data values are identified for each of the test cases.

Table 19: Test Case Identification Table: Update Order Status

Test Case ID	Scenario/ Condition	Input ID	Search orders	Select order	Process order	Check availability	Pack item	Ship order	Choose payment method	Insert payment info	Register info	Create invoice	Expected results
UOS1	Scenario 1 – Successful status update	V	V	V	V	V	V	V	V	V	V	V	Order shipped.
UOS2	Scenario 2 – Order not found	V	I	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Error message; back to search orders.
UOS3	Scenario 3 – Item not in stock	V	V	V	V	I	N/A	N/A	N/A	N/A	N/A	N/A	Warning message; option to continue order.
UOS4	Scenario 4 – Item not released	V	V	V	V	I	N/A	N/A	N/A	N/A	N/A	N/A	Warning message; mark as not released, option to continue order.
UOS5	Scenario 5 – Incorrect payment info	V	V	V	V	V	V	V	V	I	N/A	N/A	Error message; back to payment screen.
UOS6	Scenario 6 –Not created invoice	V	V	V	V	V	V	V	V	V	V	I	Nothing happens; user requests the invoice again.

Table 20: Test Case Data Table: Update Order Status

Test Case ID	Scenario/ Condition	Input ID	Search orders	Select order	Process order	Check availability	Pack item	Ship order	Choose payment method	Insert payment info	Register info	Create invoice	Expected results
UOS1	Scenario 1 – Successful status update	157	15789 45157 15788	45157	Processing	Available	Packed	Shipped	Credit Card	4444 4444 4444 4444 09/18 456	Registered	Invoice	Order shipped.
UOS2	Scenario 2 – Order not found	157	-	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Error message; back to search orders.
UOS3	Scenario 3 – Item not in stock	157	15789 45157 15788	45157	Processing	Not available	N/A	N/A	N/A	N/A	N/A	N/A	Warning message; option to continue order.
UOS4	Scenario 4 – Item not released	157	15789 45157 15788	45157	Processing	Not available	N/A	N/A	N/A	N/A	N/A	N/A	Warning message; mark as not released, option to continue order.
UOS5	Scenario 5 – Incorrect payment info	157	15789 45157 15788	45157	Processing	Available	Packed	Shipped	Credit Card	4444 4444 4444 4444 02/18 466	N/A	N/A	Error message; back to payment screen.
UOS6	Scenario 6 –Not created invoice	157	15789 45157 15788	45157	Processing	Available	Packed	Shipped	Credit Card	4444 4444 4444 4444 09/18 456	Registered	nothing	Nothing happens; user requests the invoice again.

10.2. Domain Model

Figure 15 below is the domain model for the system. It shows how the different classes are related, what their function is, and their attributes. It also illustrates the inheritance between the super- and subclasses, namely *Order*, *Entity* and *Item*.

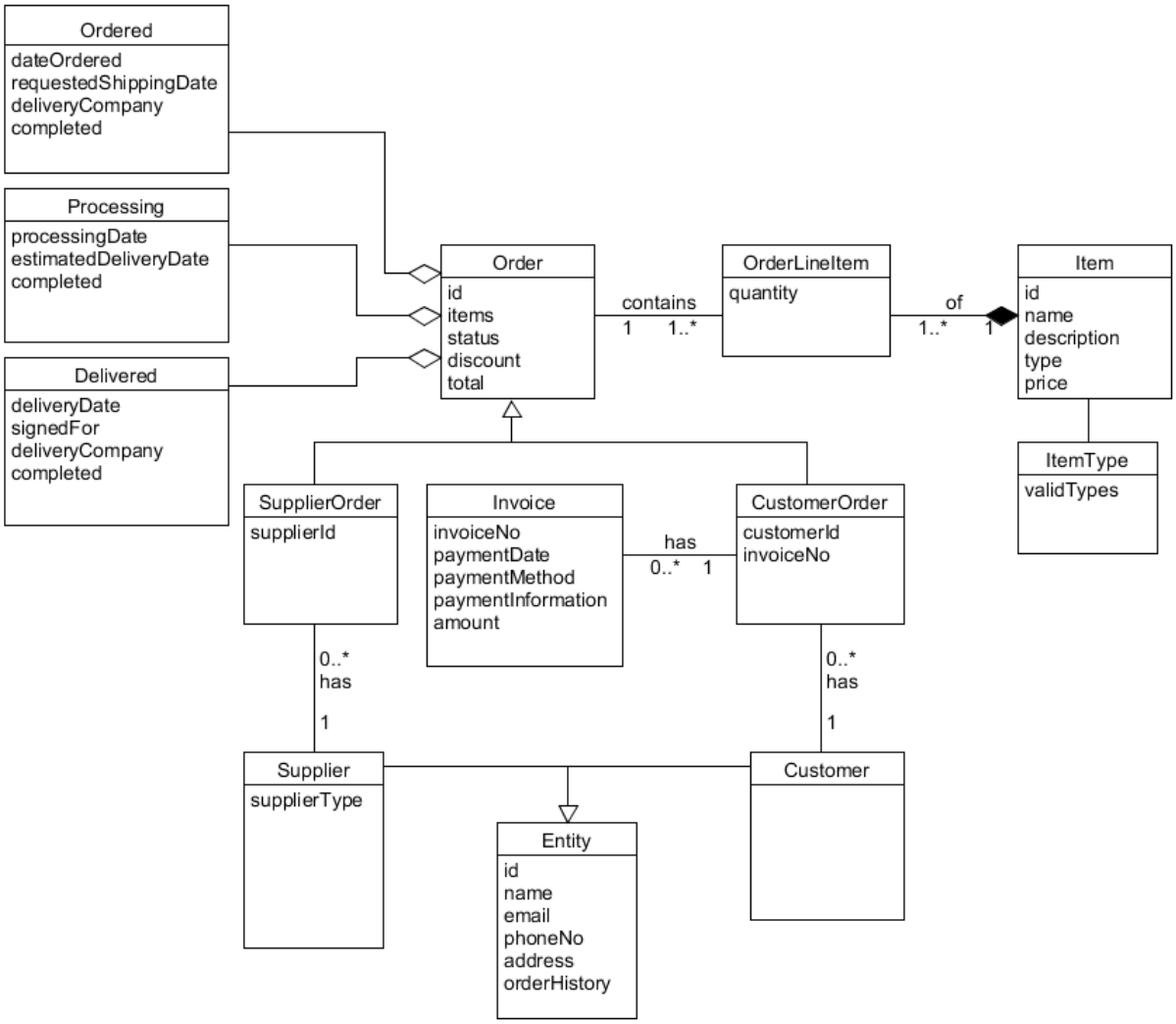


Figure 15: Domain Model

In Figure 15, the *Order* class is a superclass of *SupplierOrder* and *CustomerOrder*, both of which use the same information as in the *Order* itself. *Entity* is also a superclass of *Supplier* and *Customer*; however, *Customer* inherits its attributes from *Entity* as all the information they store is the same. Our *Order* class also has a ***State Pattern***, which signifies the *status* (*Ordered*, *Processing*, *Delivered*) of the order as it is being processed. *OrderLineItem* is an *instance* or a copy of our generic *Item* class, it is created for the specific order.

10.3. Relational Model

The diagram on the next page, Figure 16, illustrates the relational model for the system. It is a representation of how the tables in the database will store information about the system and how that information is connected.

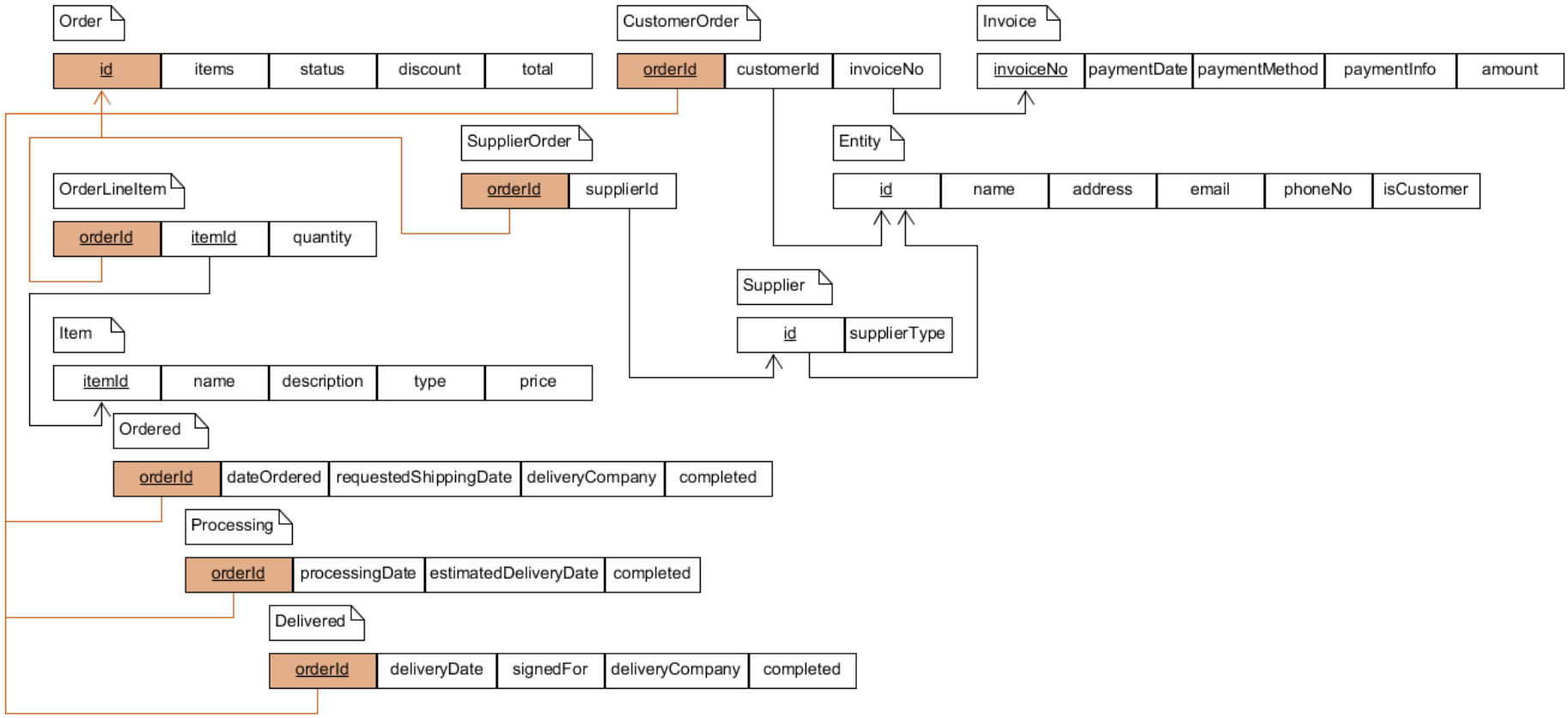


Figure 16: Relational Model

This relational model is essentially a mirror of the domain model mapped to a relational database. This means a good domain model can reduce the amount of time and effort needed to construct a reasonable relational model. As a result, the relational model constructed was already in 3rd normal form (3NF) before tweaking it. This reduced the amount of redundancy and makes for a more efficient data structure. In addition, it is very useful to examine the relations between tables, especially when encountering a many-to-many relation, which need to be eliminated if at all possible.

10.4. Database & SQL Script

This section describes the database and the use of scripts in the system, as well as a database clean up scripts for ensuring the database is always in a valid state before and after tests are being run. A relational database allows for relations between tables and specific tuples, which reduces redundancy and allows the tables to be cross referenced and joined. The database provided for this project is a MS SQL version 12.0.2000.8 database, were as the current version is 14.0.1000.169 (Microsoft, n.d.). This means some newer features and statement cannot be used and as such when researching MS SQL functionalities, one must be careful to take this into account. E.g. In newer versions the "IF EXISTS" statement can be used, but this is not available in the MS SQL database used for the project.

The full script can be found in the "DbCleanup.java" file, which contains dropping and creating the tables and inserting data into them. The first part of this clean up script is dropping the existing tables shown in Figure 17. This ensures that if a table with a specified "OBJECT_ID" exists the table is dropped. This is done to clean the database of all relevant tables to permit the creation of new table with set predictable data.

```
IF OBJECT_ID('dbo.SupplierOrder', 'U') IS NOT NULL
 DROP TABLE dbo.SupplierOrder;
IF OBJECT_ID('dbo.Supplier', 'U') IS NOT NULL
 DROP TABLE dbo.Supplier;
IF OBJECT_ID('dbo.CustomerOrder', 'U') IS NOT NULL
 DROP TABLE dbo.CustomerOrder;
IF OBJECT_ID('dbo.Invoice', 'U') IS NOT NULL
 DROP TABLE dbo.Invoice;
IF OBJECT_ID('dbo.Entity', 'U') IS NOT NULL
 DROP TABLE dbo.Entity;
IF OBJECT_ID('dbo.Delivered', 'U') IS NOT NULL
 DROP TABLE dbo.Delivered;
IF OBJECT_ID('dbo.Processing', 'U') IS NOT NULL
 DROP TABLE dbo.Processing;
IF OBJECT_ID('dbo.OrderOrdered', 'U') IS NOT NULL
 DROP TABLE dbo.OrderOrdered;
IF OBJECT_ID('dbo.OrderLineItem', 'U') IS NOT NULL
 DROP TABLE dbo.OrderLineItem;
IF OBJECT_ID('dbo.Orders', 'U') IS NOT NULL
 DROP TABLE dbo.Orders;
IF OBJECT_ID('dbo.Item', 'U') IS NOT NULL
 DROP TABLE dbo.Item;
```

Figure 17: SQL Clean-up Scripts

Next, the tables are recreated (as seen in Figure 18) with all constraints and data types. This sets up the **integrity constraints**, which ensures the data entered is only asserted as valid if the constraints are met to obtain consistency. Most of the foreign keys are set to cascade on deletion and update, so that, if an order is deleted no OrderLineItem, CustomerOrder, Invoice or Status will be deleted, as opposed to being left without an Order to reference.

```
 --OrderLineItem Table
create table OrderLineItem(orderReferenceId bigint not null,
itemReferenceId bigint not null,
quantity int not null,
constraint pkOIId primary key (orderReferenceId, itemReferenceId),
constraint fkOLIOrderId foreign key (orderReferenceId) references Orders(orderId) on delete cascade on update cascade,
constraint fkOLIItemId foreign key (itemReferenceId) references Item(itemId) on delete cascade on update cascade);

 --OrderOrdered Table
create table OrderOrdered(orderIdStatus1 bigint not null,
dateOrdered nvarchar(25) not null,
requestedDeliveryDate nvarchar(25) not null,
deliveryCompany nvarchar(25),
completed bit not null,
primary key (orderIdStatus1),
constraint fkOOOrderId foreign key (orderIdStatus1) references Orders(orderId) on delete cascade on update cascade);
```

Figure 18: SQL Table creation

After this, data is inserted, in order to make sure there is some valid data to run tests on. In Figure 19 the customerOrder insertion is shown. This inserts data into six different tables in a specific order, as multiple or them have foreign keys to refer to the entries in the other tables.

```
14  --insert Customer Order
15  insert into Orders values('Ordered', 20, 150000);
16  insert into OrderLineItem values(100000000000000, 100000000000000, 3);
17  insert into OrderOrdered values(100000000000000, '16/05/2018', '18/05/2018', 'DHL', 0);
18  insert into Invoice values('2018-05-17T08:00:00', 'Mastercard', '5151 5151 5151 5151', 120000);
19  insert into CustomerOrder values(100000000000000, 1000000, 100000000000000);
20
21  insert into Orders values('Processing', 10, 100000);
22  insert into OrderLineItem values(100000000000001, 100000000000000, 2);
23  insert into OrderOrdered values(100000000000001, '16/05/2018', '18/05/2018', 'DHL', 1);
24  insert into Processing values(100000000000001, '20/05/2018', '22/05/2018', 0);
25  insert into Invoice values('05/17/2018', 'Visa', '4892 2012 2012 2012', 90000);
26  insert into CustomerOrder values(100000000000001, 1000000, 100000000000001);
```

Figure 19: SQL Object Insertion

10.5. System Sequence Diagram & Operation Contracts

The following diagrams, called system sequence diagrams, show the interaction of the user with the system while not showing inner part of the system, just its response to the actor's actions. Each diagram is followed by a list of Operation Contract which accompany the steps in the System Sequence Diagram

10.5.1. Use Case 1: Create Order

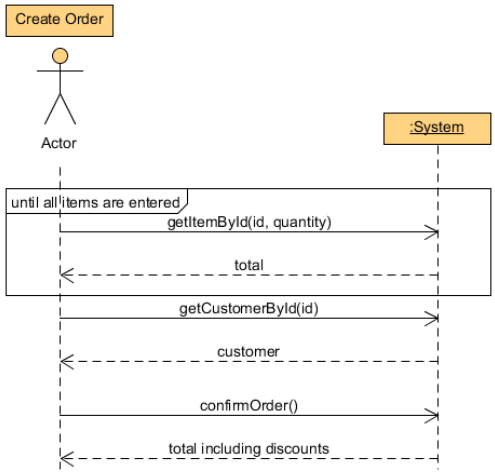


Figure 20: System Sequence Diagram: Create Order

Figure 20 on the left side displays the system sequence diagram for the use-case Create Order.

The new order starts by entering the item id and the quantity into the order and system returning the total amount. This Thomas enters a customer id in order to tie the customer to the order. System returns the customer information.

The actor confirms the order and system returns the total amount due to pay including discounts.

Below in Table 21 a list of Operation Contracts for every step in the System Sequence Diagram.

Table 21: List of Operation Contracts: Create Order

Operation	inputItemId(id)
Use-cases	Create Order
Precondition	An Order instance (object) *order* must exist
Post conditions	*orderLineItem* was associated with the Order *order.discount* was updated *order.total* was updated

Operation	enterCustomerId(id)
Use-cases	Create Order
Precondition	An Order instance (object) *order* must exist
Post conditions	*customer* was selected

Operation	confirmOrder()
Use-cases	Create Order
Precondition	An Order instance (object) *order* must exist
Post conditions	*order* is validated and saved

10.5.2. Use Case 2: Update Order Status

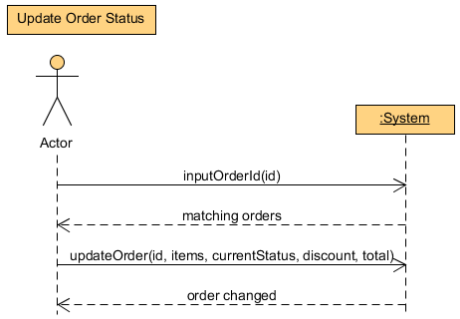


Figure 21: System Sequence Diagram: Update Order Status

Figure 21 on the left side displays the system sequence diagram for the use-case Update Order Status.

Update starts inputting the id of the order after which a system searches for matching orders and returns them. The actor then inputs the information about the order and the system returns a confirmation.

Table 22 contains a list of the Operation Contracts for every step in the System Sequence Diagram.

Table 22: List of Operation Contract: Use Case Update Order Status

Operation	inputOrderId(id)
Use-cases	Update Order Status
Precondition	An Order instance (object) *order* must exist
Post conditions	List of Order instances (objects) *orderList* is created List of Order instances (objects) *orderList* is displayed

Operation	selectOrder()
Use-cases	Update Order Status
Precondition	An Order instance (object) *order* must exist
Post conditions	An Order instance (object) is stored in *selectedOrder*

Operation	confirmProcess()
Use-cases	Update Order Status
Precondition	An Order instance (object) *selectedOrder* must exist
Post conditions	*selectedOrder.status* is updated An Invoice instance (object) *invoice* is created *invoice* is associated with *selectedOrder*

Operation	confirmShip()
Use-cases	Update Order Status
Precondition	An Order instance (object) *selectedOrder* must exist selectedOrder.status has a value of Ordered
Post conditions	*selectedOrder.status* is updated

10.6. Interaction Diagram

There are two types of Interaction diagrams. The Sequence Diagram and the Communication diagram. Both are valid forms of UML notations in the Unified Process, however why someone would choose one over the other depends on the project.

The sequence diagram has the added benefit of being able to display the call-flow sequence a lot better than a communication diagram. Communication Diagrams on the other hand have the benefit of being much more space efficient and easier to adjust during the conceptual phase. Allowing for easy erase and adding of objects into the diagram, which lends itself more to be used in an agile approach. The latter was also why in this project the choice was made to use the Sequence Diagram. In addition to that, the ability to easily follow the call-flow from the sequence diagram makes it easier to document what is going on. As according to (Larman, 2004, p. 229).

Below in Figure 22 is the sequence diagram for Inserting an order into the database. (A larger version can be found on the next page in Figure 23)

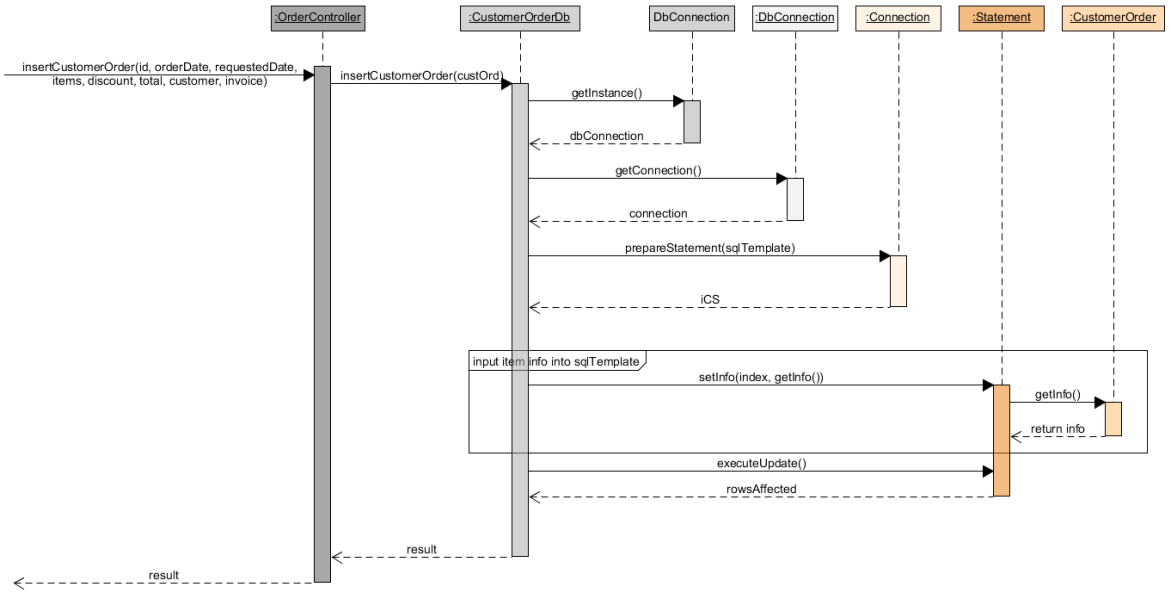


Figure 22: Sequence Diagram: Insert Order (Small)

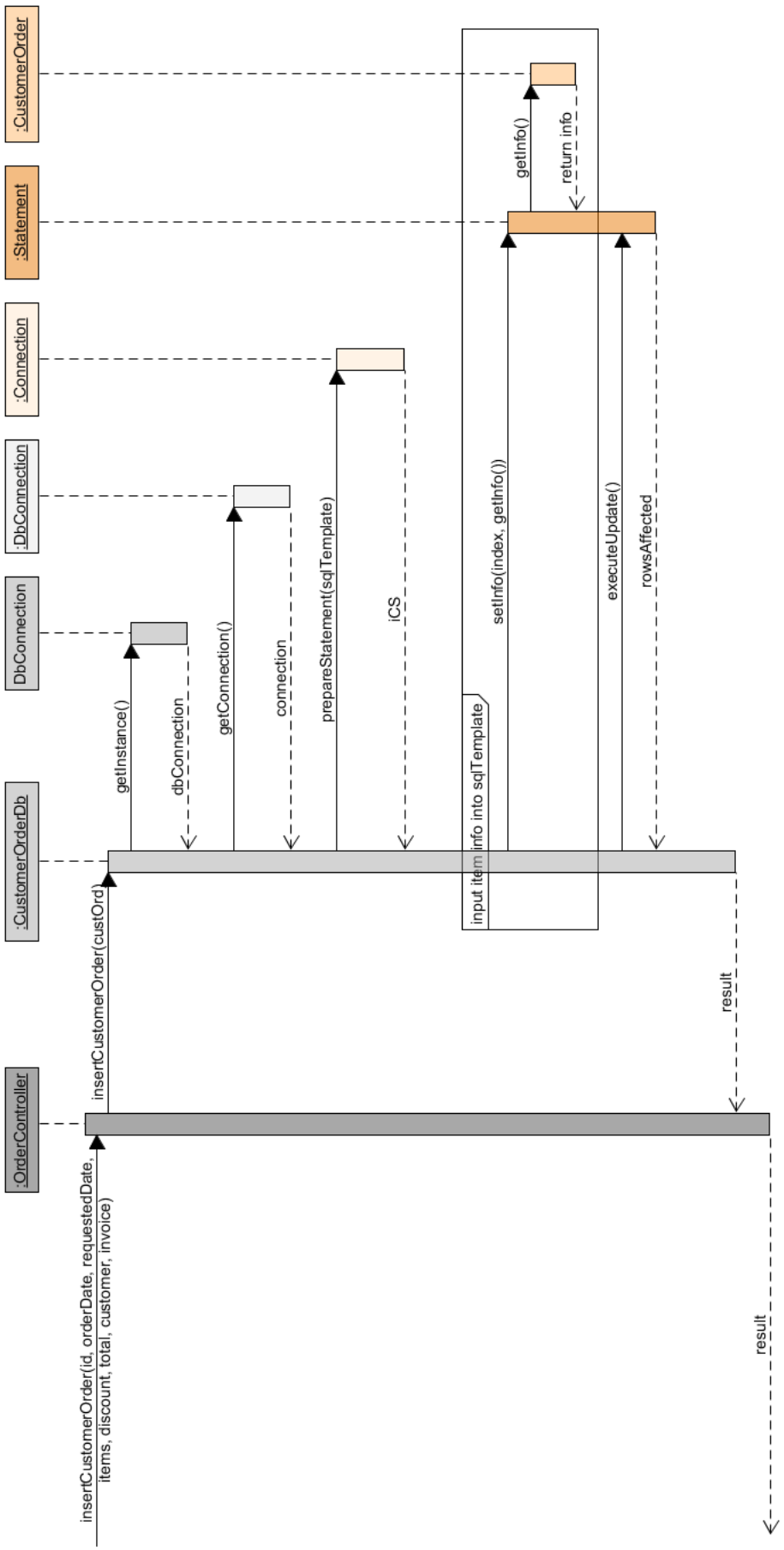


Figure 23: Sequence Diagram: Insert Order (Large)

10.7. Design Class Diagram

The Design Class Diagram shown below in Figure 24 visualizes the classes, their attributes and methods as well as the relation between them. This is to clarify the architecture that has been used in the system and identify potential problems.

Due to the inclusion of the database into the system, design class diagram becomes more complex and, therefore, constructing clear representation is vastly more important.

The graphical user interface classes, getters and setters are not included in this diagram as they would obscure the critical parts of the system.

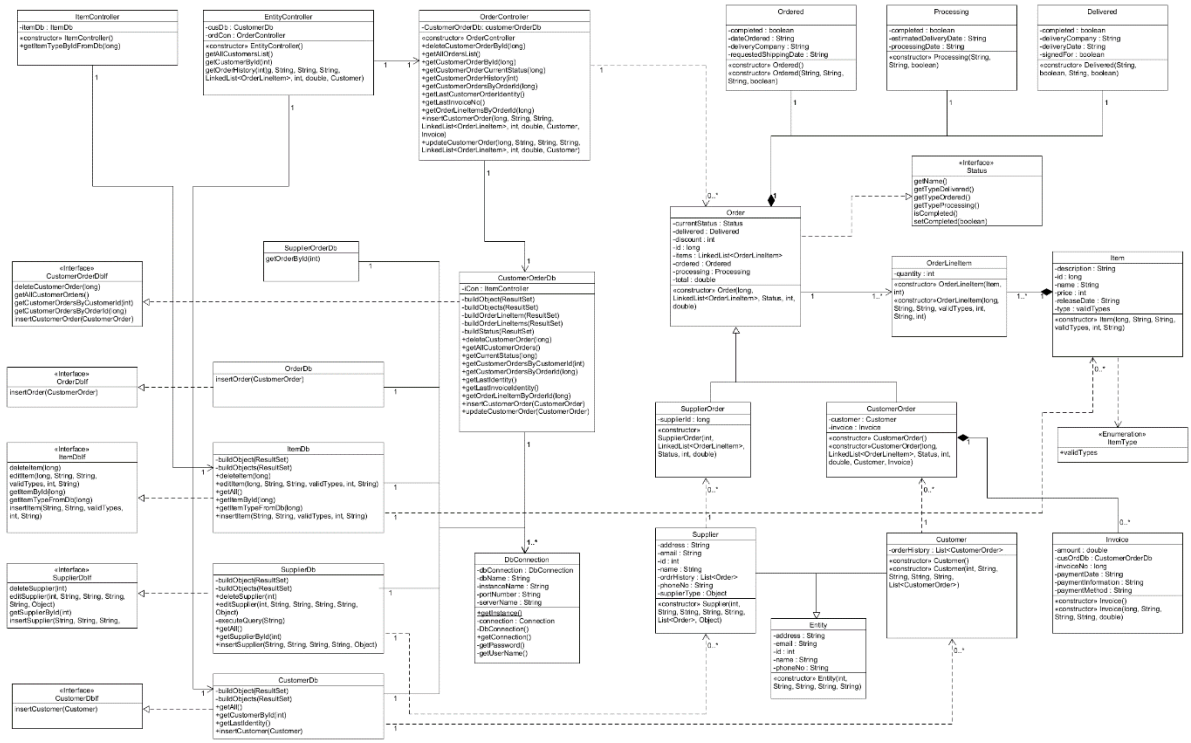


Figure 24: Design Class Diagram

Please refer to Appendix E for a large version of the Design Class Diagram.

11. Construction

This phase in the Unified Process focusses on the implementation less critical parts of the system. This chapter also focusses on all other features, standards and design choices that went into developing the system.

11.1. Code Standards & Architecture

This paragraph focusses on the code standards used when constructing the program, the architecture upon which the program is built and the design choices that had to be made during construction.

11.1.1. Code Standards

The program was written in accordance to Java's code standards (Sun Microsystems Inc., 1997). It utilizes spaces for tabs to allow the maintenance of indentation across any **IDEs** and a maximum line length of 80 characters to maintain the readability in smaller terminal or IDE windows. This is exemplified in Figure 25 below. There was no reason to choose tabs over spaces as Eclipse easily allows the use of both. Because of this it is strictly advantageous to choose spaces as they allow compatibility across different platforms.

```
1396    /*¤¶
1397     * Generates an popup based on the given message and title. By default it¤¶
1398     * does not block interactivity with the program and is disposed after 3¤¶
1399     * seconds.¤¶
1400     */¤¶
1401    private void createPopup(String message, String title) {¤¶
1402        ¤¶
1403        JOptionPane pane = new JOptionPane(message);¤¶
1404        final JDialog popup = pane.createDialog(title);¤¶
1405        ¤¶
1406        // Popup fades after 3 seconds¤¶
1407        Timer timer = new Timer(3000, event -> popup.dispose());¤¶
1408        // Doesn't block interactivity with the program¤¶
1409        popup.setModalityType(ModalityType.MODELESS);¤¶
1410        popup.setAlwaysOnTop(true);¤¶
1411        popup.setVisible(true);¤¶
1412        timer.setRepeats(false);¤¶
1413        timer.start();¤¶
1414    }¤¶
```

Figure 25: Code Standards

There are also ample comments present to ensure ease of understanding for future iterations or implementations of the code. The program follows a very descriptive method and variable naming convention that is convenient for pinpointing where they are being used and for what purposes. This one was chosen over a shorter, less-descriptive naming convention which would have to make organizing code and iterating over it simpler. A variable naming convention where the names would be abbreviated or even shortened to their first letters would be faster to type, but more difficult to decipher for a third party that has not worked on the program.

Examples Figure 26 and Figure 27 below.

```
deleteCustomerOrderById(long) : boolean
getAllOrdersList() : List<CustomerOrder>
getCustomerOrderByOrderId(long) : CustomerOrder
getCustomerOrderCurrentStatus(long) : Status
getCustomerOrderHistory(int) : List<CustomerOrder>
getLastCustomerOrderIdentity() : long
getLastInvoiceNo() : long
getOrderLineItemsByOrderId(long) : LinkedList<OrderLineItem>
insertCustomerOrder(long, String, String, LinkedList<OrderLineItem>, int, double, Customer, Invoice) : boolean
updateCustomerOrder(long, String, String, String, LinkedList<OrderLineItem>, int, double, Customer) : boolean
```

Figure 26: Code Standards: Method Names

```
txtCreateOrderOrderDate
txtCreateOrderOrderDiscount
txtCreateOrderOrderId
txtCreateOrderOrderRequestedDate
txtCreateOrderOrderTotal
txtCreateOrderSelectedCustomerId
```

Figure 27: Code Standards: Variables

11.1.2. '3-Layer Architecture'

When constructing a software system it is critical to think about the architecture of the system before the actual coding starts. Using the proper architecture ensure the system will have high cohesion and low coupling. In layman's terms this means the system needs to be designed in such a way that every class is specialized to handle one, and one thing only. Programming this way ensures that the classes are easily adaptable to any new changes in the future. Should one part change, only that part and it's immediate connected classes needs to be changed, not any of the other ones. (Barnes & Kölling, 2017, pp. 259, 260)

There are a few choices in terms of architecture, but the choice was made to work with a three-layer architecture, where the classes are separated into three different layers, see the Figure 28 below which also works well with the use of databases.

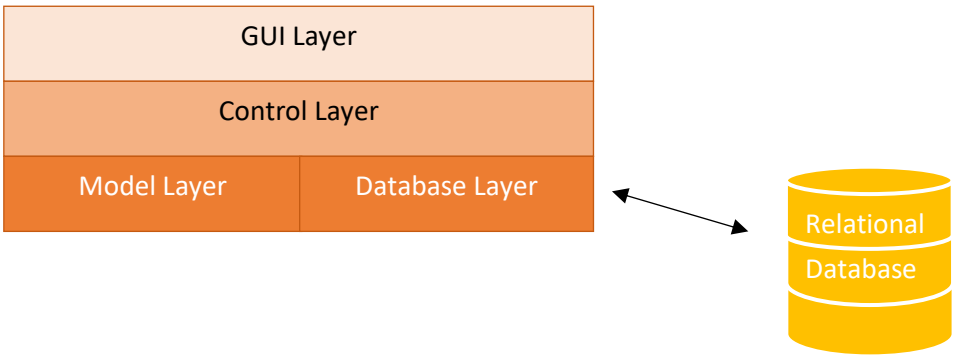


Figure 28: 3-layer architecture

Since the GUI layer (graphical user interface) should be easily replaceable by the new ones, it is the thinnest layer of the system. The control layer holds all the logic of the system and communicates with the GUI, the model and the database layers to access the information. Model layer, as known before, is now split into two parts, model and DB (the database) layer. The control layer also takes an input

from the user (through GUI layer) and inserts it to the database layer from where then the data can be inserted into the database. All the model classes are stored in the model layer and for each of these classes there is a database class created in the database layer. Through the database layer the control layer accesses the database.

11.1.3. Design Choices

Ensuring unique identity in the database

During the construction phase a discussing emerged about, how uniqueness should be ensured to identify orders and other elements. Initially, an approach to generate the identities through Java was discuss as opposed to having MS SQL generate them, because it would allow a more pleasing separation between customer and supplier orders. Ultimately it was decided against, as it would result in a lot of overhead, as the database would have to be checked for matching identities and this would dramatically impact the performance of the system. In the end, it was decided to use the identity function in MS SQL, as this is a more efficient and less time-consuming solution.

Stubs

To focus on the critical parts of the system, it was decided to implement stub for some functionalities. This was also used to have temporary methods. As the system was developed fewer and fewer stubs were present as they were replaced by functioning methods.

Lambda

Lambda were used in the action listeners in the GUI, as they greatly increased the clarity of the methods instead of having a larger code block. (Oracle, 2017)

10 heuristics

During the project, the 10 heuristics (Campbell, 2015) were considered and as many as possible were follow, but due to time constraints not all of these could be fulfilled. User control and freedom was included by allowing the user to go back and forth as they please. Consistency and standards was established by making sure the interface around the current panel was the same in every window. Error prevention was achieved by displaying error messages and preventing non-valid information to be entered. Efficiency of use was considered by improving the performance of the system through SQL query optimization and minimizing the number of queries to the database. Helping users with errors is greatly aided by the informative error messages, which can be relayed to the developers for bug fixing. Documentation is included with the system to enable the customer to assess the inner workings of the system.

ACID

When designing the transactions to communicate with the database, ACID (Atomicity, Consistency, Isolation, Durability) was used to ensure the validity in the event of errors. Errors in a transaction would cause the entire transaction to be rolled back, as a partial update would likely cause the data to be invalid and/or make future queries and updates return wrong or invalid information.

12. Transition

The transition phase of the Unified process is short and simple, it focusses on the final rounding off of the project and highlights the transition from developer to client. This chapter features a short description of the user manual that comes with the software system.

12.1. Manual

To make the transition as smooth as possible, a manual was written on how to operate the software system. This is a thin brochure that describes the product, Vengeful Games Inventory. It includes all the instruction, from how to start the program to the actual order creation (which is the main use case of the system). At the end of the brochure, there is a contact to the developers, in case of any problems.

Please refer to Appendix F for the full manual.

PART IV

Discussion and Conclusion

This page is intentionally left blank

13. Discussion and Conclusion

The discussion and conclusion focusses on evaluating the results of project. In order to achieve this, some, the conclusion has been divided into three sections as according to the report writing guide by (Nordbjerg & Nielsen, 2011):

13.1. Assessment:

Below are the three problem statements, they will be addressed individually, followed by a conclusion on whether the goal has been achieved.

Problem Statement Q1:

Can the efficiency and usability be increased in a bookkeeping system according to Vengeful Games' needs? If yes, to what extent can the change be done before the requested deadline?

A simpler UI and concise program functionality will increase daily performance and therefore maximize possible company profit. However, the project planning was not accurate enough to ensure that all the necessary features could be implemented before the requested deadline, but the critical use cases have been developed and implemented

Problem Statement Q2:

How, and in what way, can statistics of items and customer be implemented to make a significant contribution to the future business development of Vengeful Games for the next 5 years?

Statistics will provide an overview of the company's function over a long period of time; this information can be used to evaluate where the company may be generating less profit or even losing money. In the next 5 years this could be a great benefit to it in terms of inventory and sale optimization.

Problem Statement Q3:

Within the timeframe, in what way to ensure that all the data inserted by a user are valid and operations executed simultaneously?

The program uses SQL transactions to ensure that changes to data in the database are executed in one action or rolled back if they encounter an error. Data cannot be modified or read while it is being modified as part of a transaction.

The conclusion is that the most critical parts of the program have been implemented and are fully functional, but they do not cover all of the client's requirements.

13.2. Perspective:

Improvements include further implementation of threads, mutual exclusion and subsequently deadlock prevention to increase the program's performance when it will be faced with a large mass of data processing. Statistics could also be included in future iterations as a feature for analysis.

13.3. Reflection:

During this project the group was familiarized with the concept of iterative design, front-end development as well as database management. The iterative design that was followed divided development into 1-week iterations that focused on a single use case implementation in order of most critical to least critical to program functionality. The front-end development was focused on creating an easy-to-use and easy-to-understand graphical user interface based on the 10 heuristics of a good graphical use interface. In terms of database management, many optimizations and stability features are implemented to ensure it is fast and does not allow data corruption.

PART V

Group Evaluation

This page is intentionally left blank.

14. Group Process

This chapter focusses on evaluating the group process during the 2nd semester project at the Computer Science course at UCN Aalborg. It features the group contract and an evaluation of the process which takes into account some key aspects of working in a group.

14.1. Group Contract

As soon as the group was formed in the first week of the 2nd semester, a group contract was set up in order to ensure the full cooperation of all team members involved. It is a straight to the point document with what is expected and possible consequences. The full document is included in Appendix G. All group members agreed and signed the group contract which has now become in integrated part of this document. Their signatures have moved to the title page of this report. By signing this title page, all participants agree that these are also valid for the group contract in the appendix.

14.2. Evaluation of Group Process

This section evaluates the process of the group to establish how well it collaborated on the project.

Collaboration

We used the limited time we had as adequately as we could, but the total sum of time we had could not be effectively utilized. This is because we did not yet have all the knowledge of what we were expected to do, therefore anything done "prematurely" would require later revision and possibly changes. We feel like we were not always contributing to the best of all our individual abilities. We communicated efficiently and rarely encountered problems of misunderstanding. We always asked to ensure that if anyone had an opinion to voice they would do so that it could be acknowledged. The group feels that Ákos has shown great development in terms of his assertiveness in the group, it's a great improvement from the start of the project, however we feel that he has more he can improve on. Due to this being Simon's first time managing a group, we agree that he could've shown more leadership and could have put more effort into organization. Our group collaborated towards a common goal.

Coordination

It was agreed upon that the group would meet every weekday (unless otherwise specified) and that we would work whenever we had the ability to do so. It was unanimously understood that the product itself was not as important as the development process, report and ultimately the knowledge and experience we gained during it.

Information

The group spent time querying our client for input both on how he would like the system to function and on what his opinions is on the iteration of the system so far. We also did research from the provided course materials, as well as individual research from outside sources. These sources include various books, articles and websites which are sourced in our report.

Coalition

Group morale remained high throughout most of the project, there was high enthusiasm, light-heartedness and humour present. During the times of difficulty or prolonged hitches nobody reacted with irritation, stress or discontent.

Control

Our work is being held to a high professional standard, we adhered to our group contract and enforced the rules we all agreed upon. We strived to create a high-quality product that would be usable, efficient, simple and dynamic. Each group member can work on their own as well as with other members of the group.

PART VI

Appendices

This page is intentionally left blank.

Appendix A. Glossary

0-9

3-Layer Architecture: An architectural design choice which separates the software code into three different levels, the Graphical User Interface layer(which takes care of user interaction), The Control Layer(which takes care of the logic) and the Model layer(which handles the data).

A

Activity Diagram: An activity diagram is a visual representation of a sequence of actions performed by an actor. (Larman, 2004, pp. 477,478)

Anpartsselskaber: A company owned by a single or more owners, who are not personally accountable for the company's finances and must have a management, including articles of association and a founding document. Translated from (Erhvervsstyrelsen, 2018)

ApS: *see Anpartsselskaber.*

B

Brick-and-Mortar: A traditional store in an real building, 'built by bricks and mortar'.

Brief Form: *See Use-Cases – Brief Form.*

Business Case: A document which defines what is to be done, why, and what the timescale and costs involved are. (Cadle & Yeates, 2008, pp. 31-40) (*See also: Cost-Benefit Analysis, and Payback Projection)*.

C

Casual Form: *see Use-Cases – Casual Form.*

Change Management: "Change Management (CM) refers to any approach to transitioning individuals, teams, and organizations using methods intended to re-direct the use of resources, business process, budget allocations, or other modes of operation that significantly reshape a company or organization." (Duffy, 2018)

Clicks-Only: *See Internet Pureplay.*

Code Standards: "Code conventions improve the readability of the software, allowing engineers to understand new code more quickly and thoroughly" (Oracle, 1999)

Communication Diagram: *See Interaction Diagram – Communication Diagram.*

Complexity: *See Use Cases - Complexity*

Conclusion: After weighing out the positives and negatives in the *Discussion*, a conclusion can be drawn about the success of the project, answering the questions stated in the problem statement.

Construction: *See Unified Process – Construction.*

Cost-Benefit Analysis: "An estimation of strengths and weaknesses of solutions, used to determine the validity of an investment beforehand" (Bragg, 2017).

Coverage: *See Use Cases – Coverage.*

Criticality: *See Use Cases – Criticality.*

CRUD: A Use-Case which focusses on a combination of four primary data manipulation tasks; Create, Read, Update and Delete.

D

Design Class Diagram: "A static view of the class definitions which illustrates the attributes and methods of the classes" within the system. (Larman, 2004, p. 10)

Digital Business: Every aspect of the business that is managed through digital technologies and media, for example optimising and improving competitiveness and internal processes with online and traditional channels to market and supply (Chaffey, 2015, p. 15).

Discussion: Addresses the positive and negative aspects of the project and discusses any improvements for the future.

Domain Model: Visual representation of conceptual classes or real-situation objects in a domain. (Larman, 2004, p. 135)

E

E-Commerce: "all electronically mediated information exchanges between an organisation and its external stakeholders" (Chaffey, 2015, p. 13)

Elaboration: *See Unified Process – Elaboration.*

Emergent Strategy: "A dynamic approach to business strategy, which is developed iteratively and therefore, strategic analysis, strategic development and strategy implementation are developed in parallel". (Chaffey, 2015, p. 191)

Empirical Research: research performed using empirical data, gained by and/ or verifiable by observation or experience, rather than theory or pure logic. (Oxford Dictionaries, 2018)

Employee-task-goal-table: "Description of how the most important tasks are carried out - in steps." (Toftegaard, System Development Repetition Module 01, 2018)

Enkelmandsvirksomhed: A company owned by a single owner, who is personally accountable for the company (including any debt), which may or may not have any employees. Translated from (Erhvervsstyrelsen, 2018)

F

Feasibility Study: A study on the feasibility of the project. It covers a full business analysis including stakeholder analysis, SWOT analysis and a business case.

Fully Dressed Form: *See Use-Cases – Fully Dressed Form.*

I

IDE: *See Interactive Development Environment.*

Inbound Logistics: The management of material resources entering an organisation from its suppliers and other partners. (Chaffey, 2015, p. 263)

Inception: *See Unified Process – Inception.*

Integrity Constraints: a set of constraints designed to maintain the integrity of the data inside a database. Examples are for example specifying a data type for each data item, or specifying that a record in one file must be related to records in other files, (Elmasri & Shamkant, 2017, p. 51)

Interaction Diagram: "A generalization of two more specialized UML diagram types, Communication and Sequence diagram. It illustrates how objects interact via messages". (Larman, 2004, pp. 221,222)

Interaction Diagram – Communication Diagram: "Illustrate object interactions in a graph or network format, in which objects can be placed anywhere on the diagram". (Larman, 2004, p. 223)

Interaction Diagram – Sequence Diagram: "Illustrate [object] interactions in a kind of fence format, in which each new object is added to the right" (Larman, 2004, p. 222), creating a 'fence' type structure which includes method calls.

Interactive Development Environment: "A system for supporting the process of writing software" (TheFreeDictionary, n.d.)

Internet Pureplay: An organisation with principally an online presence (Chaffey, 2015, p. 66)

M

Methodology: Guidelines to follow for completing activities – for example Waterfall, UP, Scrum, Kanban. (Kondylis, Unified Process and good practices from Scrum, 2018, p. 2)

Mission: "The fundamental purpose of an enterprise that defines the nature of its business and provides strategic direction to unify the use of human and other resources". (Bloisi, Cook, & Hunsaker, 2006, p. 76)

Mock-up: An initial visual representation for the system with the intent of gaining feedback from the customer.

O

Organisational Culture: The culture of an organisation is defined by its habits and the written an unwritten rules of the organisation.

Organisational Form: A visual representation of the division of people and power in an organisation, according to the division by Mintzberg.

Organisational Structure: A schematic visualisation of how the organisation is organised. Often following standardised structures.

Operation Contract: A more precise description of system behaviour which uses a pre-and post-condition form to describe detailed changes to objects in a domain model, as the result of a system operation. (Larman, 2004, p. 181)

Outbound Logistics: The management of resources supplied from an organisation to its customers and intermediaries. (Chaffey, 2015, p. 263)

P

Payback Projection: Also referred to as Payback Period, refers to the required amount of time for an investment to break-even and subsequently become profitable. (Bragg, 2017)

Persona Profile: is a fictional, yet realistic, description of a typical user of the product. (Harley, 2015)

Phase Plan: A schematic overview of the different phases of the project, including the iteration length, deadlines and relevant artefacts which are vital for the completion of each phase.

Porter's 5 Forces: A classical model from 1980 by Michael Porter, which depicts five main competitive forces that affect a company. (Chaffey, 2015, p. 199)

Prescriptive Strategy: "An approach to business strategy, which rely heavily on foresight to determine a set sequence of strategic analysis, strategic development and strategy implementation beforehand". (Chaffey, 2015, p. 191)

Problem Statement: "A problem statement is a clear concise description of the issue(s) that need(s) to be addressed by a problem solving team." (Ceptara: Chris Lindstrom, 2009)

Project Plan: a detailed description of the entire scope of the project, in chronological order, including required artefacts for process clarification.

Push and Pull Model: The degree to which the company pushes the product onto the market through for example advertising, or the customer pulls a product to the market due to high demand.

R

Relational Model: Formal model of relational database. Represents the database as a collection of relations. Used for building functional databases. (Elmasri & Shamkant, 2017, p. 180)

Relational Database: A database is a relational database when the tables inside the database contain objects which have a common relation in the sense that they are of the same type. Every new type will be a separate table in the database. (Elmasri & Shamkant, 2017, p. 54)

Requirements: "Capabilities and condition to which the system – and more broadly, the project- must conform" (Larman, 2004, p. 54). They can be separated into functional and none-functional requirements.

Risk Analysis: A process which identifies, analyses and assesses the possible risks involved in a project and assigns a risk probability and severity rating to them in order to create a total risk value. The higher the value to more important it is to address the issue.

S

Sequence Diagram: *see Interaction Diagram – Sequence Diagram.*

Stakeholder Analysis: A systematic plan which avoids unexpected difficulties by identifying critical stakeholder groups, their interests and spots any potential trouble. It can also help in identifying potential allies and coalitions and help manage relations with them. (Boddy & Buchanan, 1992, pp. 55,56)

State Pattern: "State design pattern is used when an Object changes its behaviour based on its internal state." (Kumar, n.d.)

SCM: *See Supply Chain Management*

Supply Chain Management: "The coordination of all supply activities of an organisation from its suppliers and partners to its customers." (Chaffey, 2015, p. 252)

SWOT Analysis: Strength, Weakness, Opportunity and Threats analysis. Depicting an overview and description of the various internal (SW) and external (OT) factors that may affect a business. (Bloisi, Cook, & Hunsaker, 2006, pp. 86,87)

System Sequence Diagram: "A picture that shows, for one particular scenario of a use case, the events that external actors generate, their order, and inter-system events. All system are treated as a black box; the emphasis of the diagram is events that cross the system boundary from actors to systems." (Larman, 2004, p. 176)

System Vision: "A short executive overview document for quickly learning the project's big ideas". (Larman, 2004, p. 58)

T

Test Cases: "A test case is a set of test inputs, execution conditions, and expected results." (Software Testing Class, n.d.)

Transition: *See Unified Process – Transition.*

U

Unified Process: A Unified Process (UP) is a software development process that describes an approach to building, deploying and possibly maintaining software. As according to: (Larman, 2004, p. 18)

Unified Process – Construction: "System design, programming and testing. Building the remaining system in short iterations" (Toftegaard, Planning by UP, 2017, p. 13) inspired by (University of Chicago, n.d., p. 17). Defined by "an Iterative implementation of the remaining lower risk and easier elements, and preparation for deployment" (Larman, 2004, p. 33).

Unified Process – Elaboration: "Develop an understanding of the problem domain and the system architecture, risk significant portions may be coded/tested, about 80% of major requirements identified". (Toftegaard, Planning by UP, 2017, p. 13) inspired by (University of Chicago, n.d., p. 16).

Unified Process – Inception: "Establish the business case for system, define risks, scope, obtain 10% of the requirements, estimate next phase effort." (Toftegaard, Planning by UP, 2017, p. 13) inspired by (University of Chicago, n.d., p. 15).

Unified Process – Transition:" Deploy the system in its operating environment. Deliver releases for feedback and deployment" (Toftegaard, Planning by UP, 2017, p. 13) inspired by (University of Chicago, n.d., p. 18).

Unique Selling Point: A part in the strategy of the company which is unique to this company, when compared to its competitors, with the aim at attracting customers.

UP: *see Unified Process.*

Use-Cases: "a collection of related success and failure scenarios that describe an actor using a system to support a goal" (Larman, 2004, p. 63)

Use-Cases – Brief Form: "Brief one -paragraph summary. Usually of the main success scenario" (Larman, 2004, p. 66).

Use Cases – Casual Form: "Informal paragraph format. Multiple paragraphs that cover various scenarios." (Larman, 2004, p. 66)

Use Cases – Complexity: The degree of complexity, in other words, intricate and complicated, the use case presents.

Use Cases – Coverage: The degree of coverage depicts how large the use case is. In other words, how much of the software system is described by this one use case.

Use Cases – Criticality: The criticality of a use case refers to the importance of this use case in the software system.

Use Cases – Fully Dressed Form: "All steps and variations of the use case are written in detail, which includes supporting sections, such as preconditions and success guarantees". (Larman, 2004, p. 67)

V

Vision: "A desired future image of the organisation and its processes and products that integrates current realities and expected future conditions within a specific time frame". (Barnes & Kölling, 2017, p. 77)

This page is intentionally left blank.

Appendix B. List of References

Barnes, D., & Kölling, M. (2017). *Objects First with Java, A Practical Introduction Using BlueJ* (6th ed.). Harlow: Pearson Education Limited.

Bloisi, W., Cook, C. W., & Hunsaker, P. (2006). *Management and Organisational Behaviour* (2nd ed.). New York: McGraw-Hill Education.

Boddy, D., & Buchanan, D. (1992). *Take the Lead, Interpersonal Skills for Project Managers.* Upper Saddle River: Prentice Hall.

Bragg, S. (2017, July 29th). *Payback Method | Payback period formula*. Retrieved May 29th, 2018, from Accounting Tools: https://www.accountingtools.com/articles/2017/5/17/payback-method-payback-period-formula

Brooks, I. (2009). *Organisational Behaviour: Individuals, Groups and Organisations.* (4th ed.). Harlow: Pearson Education Limited.

Cadle, J., & Yeates, D. (2008). *Project Management for Information Systems* (5th ed.). Harlow: Pearson Education Limited.

Calix, T. E. (2018). Retrieved from Vengeful Games: http://www.vengefulgames.dk/

Campbell, K. (2015, August 30th). *Heuristics 101: Understanding and Applying Design Principles*. Retrieved May 6th, 2018, from LinedIn SlideShare: https://www.slideshare.net/KamariaCampbell/heuristics-101-understanding-and-applying-design-principles

CB Insights. (2018, February 2nd). *CB Insights: The Top 20 Reasons Startups Fail*. Retrieved May 25th, 2018, from CB Insights: https://www.cbinsights.com/research/startup-failure-reasons-top/

Center for VækstAnalyse. (2017, March). *Iværksætterbarometer 2017.* Retrieved May 26th, 2018, from http://vaekstanalyse.dk/file/640342/ivaerksaetterbarometer_2017.pdf

Ceptara: Chris Lindstrom. (2009, March 18th). *How To Write A Problem Statement*. Retrieved June 3rd, 2018, from Ceptara: http://www.ceptara.com/blog/how-to-write-problem-statement

Chaffey, D. (2015). *Digital Business and E-Commerce Management, Strategy, Implementation and Practice* (6th ed.). Harlow: Pearson Education Limited.

Duffy, J. (2018). *Organisational Management.* New York: Library Press.

Elmasri, R., & Shamkant, N. B. (2017). *Fundamentals of Database Systems* (7th ed.). Harlow: Pearson Education Limited.

Erhvervsstyrelsen. (2018, February 2nd). *Anpartsselskab (ApS)*. Retrieved June 3rd, 2018, from Virk: Startværk: https://startvaekst.virk.dk/bliv-klar-til-start/vaelg-virksomhedsform/anpartsselskab-aps

Erhvervsstyrelsen. (2018, February 2nd). *Enkeltmandsvirksomhed*. Retrieved June 3rd, 2018, from Virk: Startvækst: https://startvaekst.virk.dk/bliv-klar-til-start/vaelg-virksomhedsform/enkeltmandsvirksomhed

Harley, A. (2015, February 16th). *Personas Make Users Memorable for Product Team Members*. Retrieved May 6th, 2018, from Nielsen Norman Group: https://www.nngroup.com/articles/persona/

JUF.dk ApS. (2018). *juf.dk*. Retrieved May 6th, 2018, from https://www.juf.dk/selskabsformer

Kondylis, D. (2018). Logistics and SCM. Aalborg. Retrieved May 30th, 2018, from ucn.instructure.com

Kondylis, D. (2018). Recap of System Development (2nd Sem.). Aalborg. Retrieved May 29th, 2018, from ucn.infrastructure.com

Kondylis, D. (2018). Unified Process and good practices from Scrum. Retrieved May 30th, 2018

Kumar, S. (n.d.). *State Design Pattern*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/state-design-pattern/

Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development* (3rd ed.). Upper Saddle River: Addison Wesley Professional.

Larson, E., & Gray, C. (2011). *Project Management, the managerial process* (5th ed.). New York: McGraw-Hill/Irwin.

Lucidchart Content Team. (2017, November 2nd). *The Pros and Cons of Waterfall Methodology*. Retrieved May 29th, 2018, from Ludichart: https://www.lucidchart.com/blog/pros-and-cons-of-waterfall-methodology

Microsoft. (n.d.). *SQL Server Downloads | Microsoft*. Retrieved June 3rd, 2018, from Microsoft: https://www.microsoft.com/en-us/sql-server/sql-server-downloads

Nordbjerg, F., & Nielsen, R. (2011, August). Report writing guide. Aalborg, Denmark. Retrieved May 17th, 2018

Oracle. (1999). *Code Conventions for the Java Programming Language*. Retrieved May 26th, 2018, from Oracle: http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-139411.html#16712

Oracle. (2017). *Oracle: The Java™ Tutorials*. Retrieved May 24th, 2018, from Oracle: https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html

Oxford Dictionaries. (2018). *English Oxford Dictionary*. Retrieved May 27th, 2018, from https://en.oxforddictionaries.com/

Software Testing Class. (n.d.). *System Testing: What? Why? & How?* Retrieved from Software Testing Class: http://www.softwaretestingclass.com/system-testing-what-why-how/

Sun Microsystems Inc. (1997, September 12). *Java Code Conventions.* Retrieved May 22nd, 2018, from Oracle: http://www.oracle.com/technetwork/java/codeconventions-150003.pdf

Tang, D. (2014, October 21st). *Introduction to Strategy Development and Strategy Execution*. Retrieved June 1st, 2018, from flevy.com: http://flevy.com/blog/introduction-to-strategy-development-and-strategy-execution/

The PHP Group. (n.d.). *Limitations*. Retrieved May 7th, 2018, from PHP: Hyperscript Preprocessor: https://secure.php.net/manual/en/sdodasrel.limitations.php

TheFreeDictionary. (n.d.). Retrieved from the Free Dictionary: https://www.thefreedictionary.com/

Toftegaard, L. L. (2017). Planning by UP. Retrieved May 29th, 2018, from https://ucn.instructure.com/courses/12783/files/folder/Module08?preview=503850

Toftegaard, L. L. (2018). System Development Repetition Module 01. Retrieved June 3rd, 2018

Trello. (2011). (Atlasian) Retrieved February 2018, from Trello: https://trello.com

University College Nordjylland. (2014, September). *Curriculum for the Academy Profession Degree Programme in Computer Science. National Section.* Retrieved May 21st, 2018, from UCN.dk: https://www.ucn.dk/

University of Chicago. (n.d.). Lecture 4, Process and Method: An introduction to the Rational Unified process. Retrieved May 29th, 2018, from people.cs.uchicago.edu/~matei/CSPP523/lect4.ppt

Young, David C.; Computer Sciences Corporation. (2013). *Software Development Methodologies.* Retrieved May 18th, 2018, from https://www.asc.edu/sites/default/files/org_sections/HPC/documents/sw_devel_methods.pdf

This page is intentionally left blank.

Appendix C. Full Problem Statement

Problem Statement – Vengeful Games.dk.	
Student names	Linda Fuchs, Simon Jensen, Juliana Palkova, Dimitar Pilyakov, Akos Skutovics
Title (initial)	Vengeful Games Bookkeeping system
	Vengeful Games is an online store, which sells board games and board game accessories. Currently Vengeful Games has an inadequate bookkeeping system and have several feature requests along with some existing features.
Problem / Problem area	Vengeful Games is having problems with the administrative side of the business. This includes the handling of customers with multiple orders and accessing the accumulated price of items in an order/orders. These issues along with other minor issues result in a lot of frustration while running the business. As a result, it takes the owner several hours every day to run the administrative side of the business, which is not the reason the business was started to begin with. In addition to this, the customer would like to be able to view statistics and graphs related to sales and contribution margins, as this can help get a better sense of, which items and customers are more worth cultivating than others. The main problems of Vengeful Games' system can be represented as the following four points: 1. The customer wants to be able to more easily find multiple orders associated with one customer. 2. Cannot easily access the accumulated price of the items in the order/orders. 3. Cannot easily manipulate the data about multiple orders by the same customer. 4. Lacking clarity of how the business is going and the value of different areas of the business.
Problem statement	Can the efficiency and usability be increased in a bookkeeping system according to Vengeful Games' needs? If yes, to what extent can the change be done before the requested deadline? How, and in what way, can statistics of items and customer be implemented to make a significant contribution to the future business development of Vengeful Games for the next 5 years? In what way do we ensure that all the data inserted by a user is valid and operations are executed simultaneously?
Method / procedure	This project is to be carried out using the UP development methodology, which means that UML diagrams are to be made to help structure the project and system. This will be done using "UMLet".

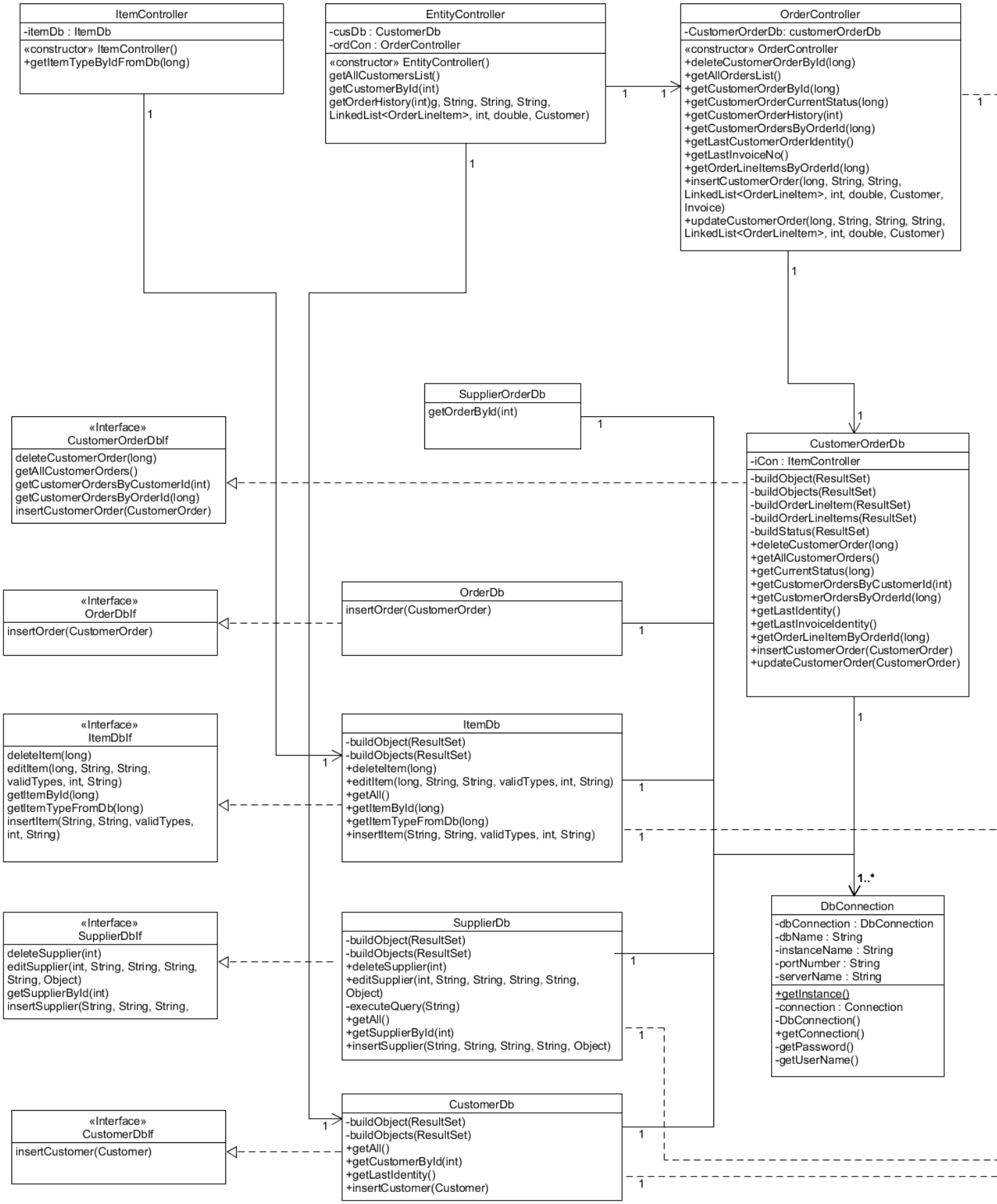
	The system is to be written in Java with “Eclipse” as IDE and a GUI is to be made in swing using the Eclipse plugin “WindowBuilder”. For version control SVN will be used. To construct a database to persistently store items, orders and customers, MS SQL is to be used. “Microsoft SQL Server Management Studio 17” will be used to manage the database. To ensure the quality of this system, scope and feature creep will be in focus and tests are to be conducted. In addition, options for statistics and graph algorithms will be examined and implemented in a reasonable fashion according to the wishes of the customer.

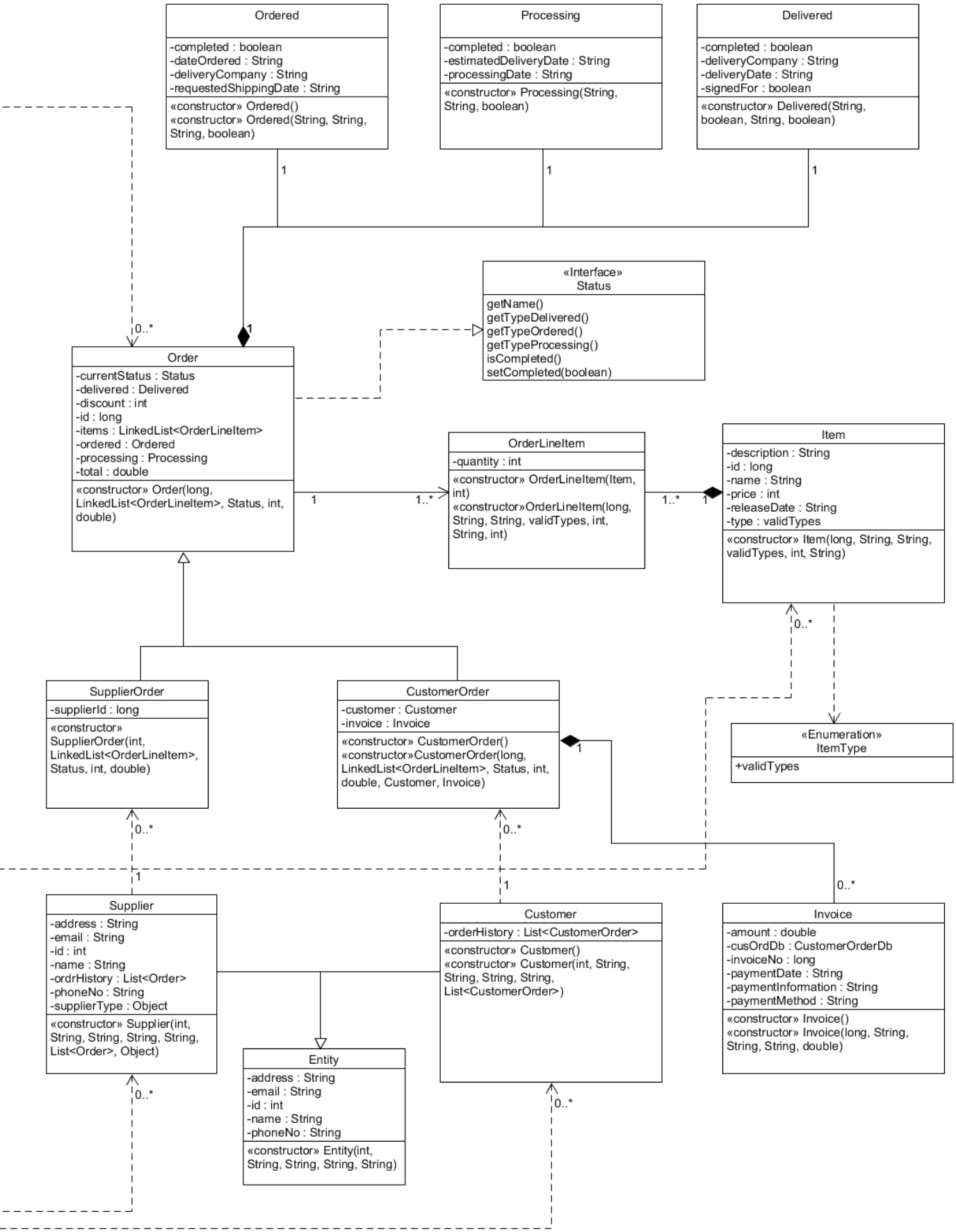
Appendix D. Ten Primary Cultural Characteristics

According to (Larson & Gray, 2011, pp. 74-75) "research suggest that there are 10 primary characteristics which, in aggregate, capture the essence of culture:

1. **Member identity**: the degree to which employees identify with the organization as a while rather that with their type of job or field of professional expertise.
2. **Team emphasis**: the degree to which work activities are organized around groups rather than individuals.
3. **Management focus**: the degree to which management decisions take into account the effect of outcomes on people within the organization.
4. **Unit integration**: the degree to which units within the organization are encouraged to operate in a coordinated or interdependent manner.
5. **Control**: the degree to which rules, policies and direct supervision are used to oversee and control employee behaviour.
6. **Risk tolerance**: the degree to which employees are encouraged to be aggressive, innovative and risk seeking.
7. **Reward criteria**: the degree to which rewards such as promotion and salary increases are allocated according to employee performance rather than seniority, favouritism, or other non-performance factors.
8. **Conflict** tolerance: the degree to which employees are encouraged to air conflicts and criticisms openly.
9. **Means versus end orientation**: the degree to which management focuses on outcomes rather than on techniques and processes used to achieve those results.
10. **Open-system focus**: the degree to which the organization monitors and responds to changes in the external environment."

Appendix E. Design Class Diagram – Full




Ordered
-completed : boolean
-dateOrdered : String
-deliveryCompany : String
-requestedShippingDate : String
«constructor» Ordered()
«constructor» Ordered(String, String, String, boolean)
Processing
-completed : boolean
-estimatedDeliveryDate : String
-processingDate : String
«constructor» Processing(String, String, boolean)
Delivered
-completed : boolean
-deliveryCompany : String
-deliveryDate : String
-signedFor : boolean
«constructor» Delivered(String, boolean, String, boolean)
«Interface»
Status
getName()
getTypeDelivered()
getTypeOrdered()
getTypeProcessing()
isCompleted()
setCompleted(boolean)
Order
-currentStatus : Status
-delivered : Delivered
-discount : int
-id : long
-items : LinkedList<OrderLineItem>
-ordered : Ordered
-processing : Processing
-total : double
«constructor» Order(long, LinkedList<OrderLineItem>, Status, int, double)
OrderLineItem
-quantity : int
«constructor» OrderLineItem(Item, int)
«constructor»OrderLineItem(long, String, String, validTypes, int, String, int)
Item
-description : String
-id : long
-name : String
-price : int
-releaseDate : String
-type : validTypes
«constructor» Item(long, String, String, validTypes, int, String)
SupplierOrder
-supplierId : long
«constructor» SupplierOrder(int, LinkedList<OrderLineItem>, Status, int, double)
CustomerOrder
-customer : Customer
-invoice : Invoice
«constructor» CustomerOrder()
«constructor»CustomerOrder(long, LinkedList<OrderLineItem>, Status, int, double, Customer, Invoice)
«Enumeration»
ItemType
+validTypes
Supplier
-address : String
-email : String
-id : int
-name : String
-ordrHistory : List<Order>
-phoneNo : String
-supplierType : Object
«constructor» Supplier(int, String, String, String, String, List<Order>, Object)
Entity
-address : String
-email : String
-id : int
-name : String
-phoneNo : String
«constructor» Entity(int, String, String, String, String)
Customer
-orderHistory : List<CustomerOrder>
«constructor» Customer()
«constructor» Customer(int, String, String, String, String, List<CustomerOrder>)
Invoice
-amount : double
-cusOrdDb : CustomerOrderDb
-invoiceNo : long
-paymentDate : String
-paymentInformation : String
-paymentMethod : String
«constructor» Invoice()
«constructor» Invoice(long, String, String, String, double)

USER MANUAL

VENGEFUL GAMES INVENTORY SYSTEM

GROUP 1

Linda A. C. Fuchs
Simon Jensen
Juliána Paľková
Dimitar Pilyakov
Ákos Skutovics

USER MANUAL

VENGEFUL GAMES INVENTORY SYSTEM

TABLE OF CONTENTS

Introduction

Congratulations on your new purchase, Vengeful Games Inventory System. This piece of software will help you and your company develop a more satisfying workflow. This user manual will go through the features and workflows of the inventory system step-by-step to ensure you get the most bang for your buck. This user manual also includes customer service contact details, should you have any further questions or run in to unexpected difficulties, you are more than welcome to contact the developers.

Start using the Vengeful Games Inventory System

To start the system simply double-click the .jar file and the system will start and connect to your database server after a few seconds. Once the system is open, the order screen is displayed as show below.

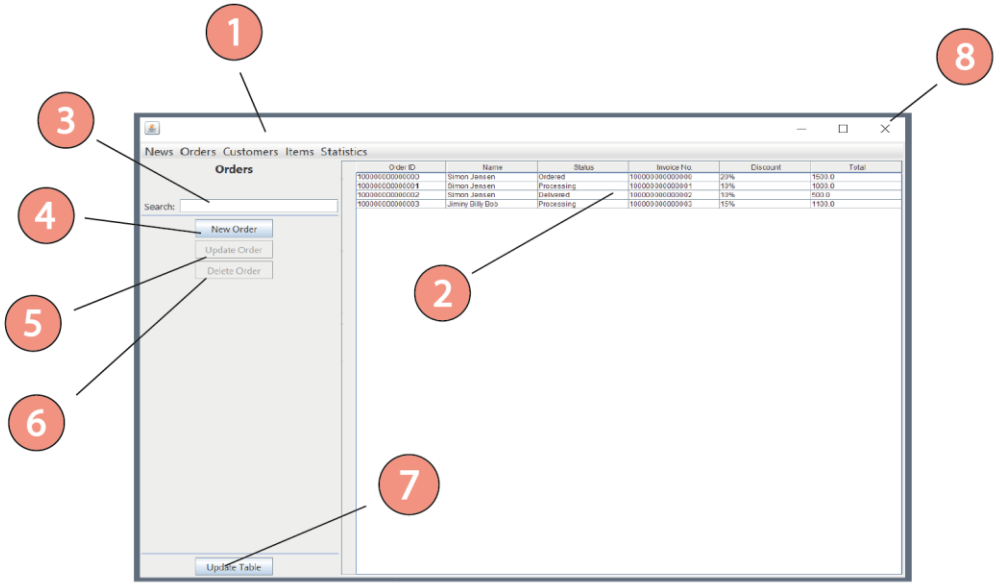


Figure no. 1

This screen, Figure no. 1, shows a list of orders in order to get you started as smoothly as possible.

Explanatory Notes:

1. Main Menu Bar
2. Table of all Orders
3. Searching Field
4. New Order Button
5. Update/ Edit Order Button
6. Delete Order Button
7. Update Table Button
8. Exit the Program

1 Main Menu Bar

Main Menu Bar is placed on the top of the window. In the menu, there are several submenus (News, Orders, Customers, Items and Statistics). If you click on one of the submenus there are displayed different menu items as shown in the figure no. 2.

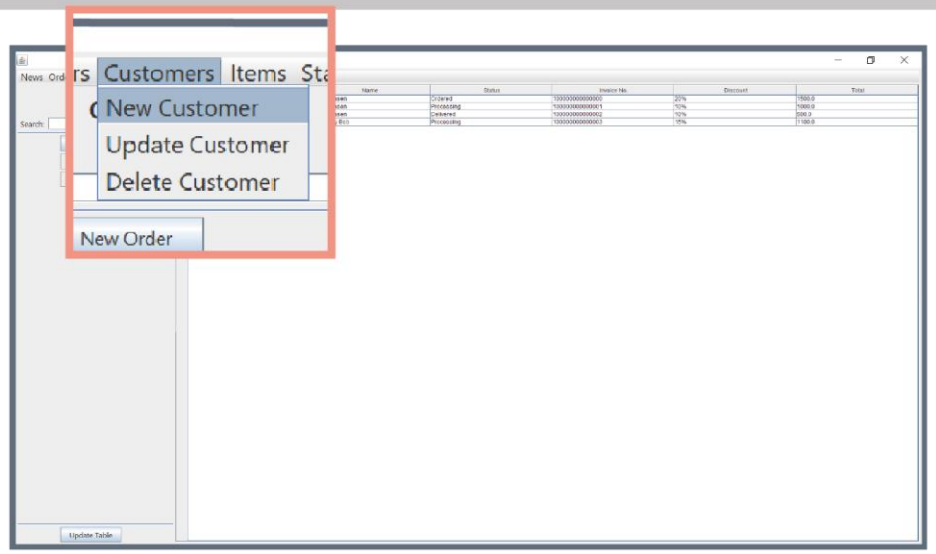


Figure no. 2

2 Table of all Orders

When you open the program, there is a table of all orders displayed as shown in the Figure no. 3. Every order is described by the various additional information (Order ID, Name, Status, Invoice No. Discount, Total).

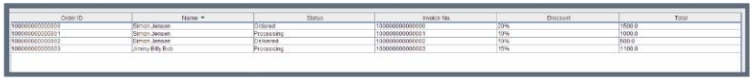


Figure no. 3

When clicking on the column title (Order ID, etc.) as shown on the left side, in Figure no. 4 and 5, it is possible to sort the Orders ascendingly or descendingly in aphabet or by ID.

Figures no. 4 and 5

3 Searching Field

To search the Orders, simply type any keyword. While typing, there are results displayed in the table, shown in Figure no. 6. It is possible to search by all attributes, namely, Order ID, Customer's Name, Status of the Order, Invoice No., Discount or by Total amount paid for an Order.

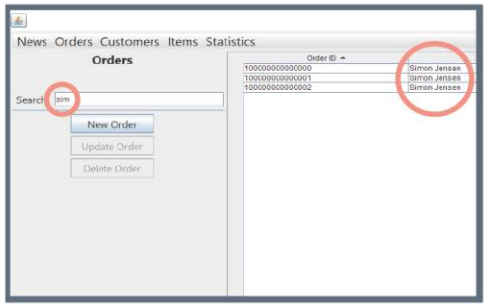


Figure no. 6

Selecting an Order

To select an Order, click on the order you want to modify or display. When the order is selected, two buttons (Update Order & Delete Order) are enabled, as shown in the Figure no. 7.

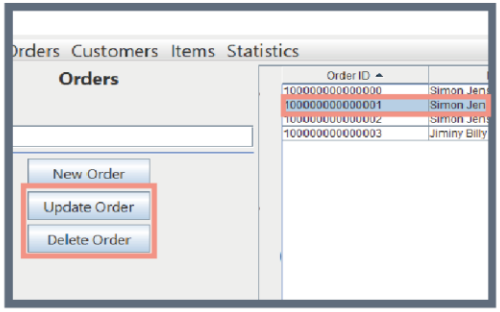


Figure no. 7

4 New Order Button

When the 'New Order' Button is clicked, new Menu, Figure no. 8, pops up consisting of different panels (from left - Order Information panel, Customer table panel, Customer Information panel - for registering new customer).

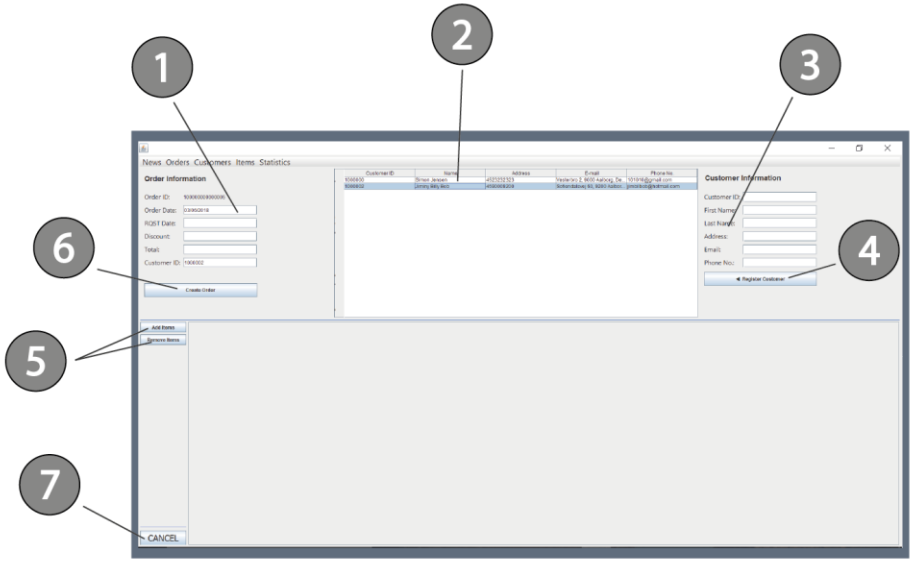


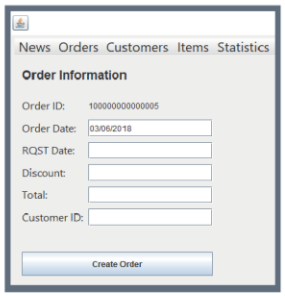
Figure no. 8

Explanatory Notes:

1. Order Info Panel
2. Customer Table
3. New Customer Panel
4. Register Customer Button
5. Add & Remove Item Buttons
6. Create Order Button
7. Cancel Order Button

1 Order Info Panel

Figure no. 9



The Order Information Panel, Figure no. 9, consists of five different textfields that need to be filled in before creating a new order. Order Date is filled in automatically by the system, which inputs the current date of particular order. 'RQST Date' stands for Requested Date which is a date when customer wants to have the order delivered. Discount and Total amount to pay are calculated according to requested parameters of the user. Customer ID is filled in automatically when selecting a customer from the Table of Customers.

2 Customer Table

The Customer Table, Figure no. 10 on the next page, is table of all registered customers in the system. From this table it is possible to modify data of a regular customers or select and view the customer. When creating a new order for a customer that is already in the system, the customer can be simply selected in the Customer Table and his ID is automatically tied with that order as shown in the picture

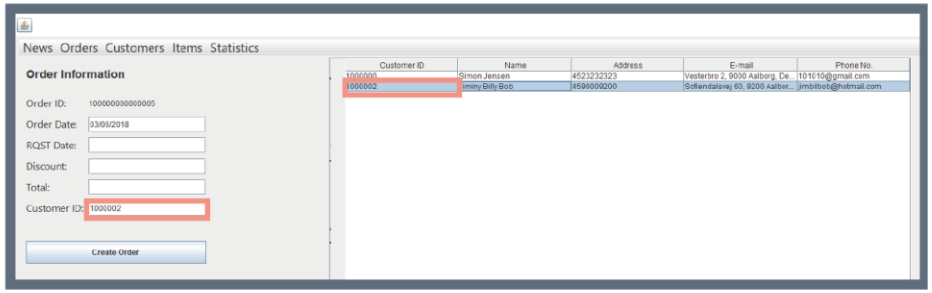


Figure no. 10

3 New Customer Panel

When a new order comes from not registered customer, in this panel you can register a new customer inserting all necessary information about the customer into the textfields as it is shown in the Figure no. 11.

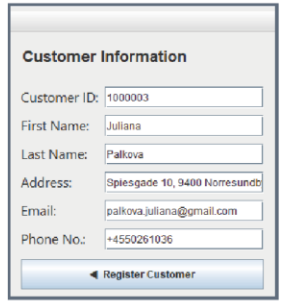


Figure no. 11

10 Order Info Panel

11 New Customer Panel

12 Register Cust. Button

Customer Table

Add, Remove Item Buttons

13 Create Order Button

14 Update/Edit Order Button

15 Delete Order Button

Cancel Order Button

Update Table Button

4 Register Customer Button

The Register Customer Button is clicked after all the necessary information about new customer is filled in, in the all textfields. Then the new customer appears in the Customer Table and can be selected for the next order he places.

5 Add & Remove Item Buttons

Add Item Button is displayed next to the Item Table (not implemented yet), from which you can choose an item which needs to be added into the order.

Remove Item Button is displayed underneath the Add Item Button. When you click this button, it removes an item from a particular order, if order has not been created yet.

6 Create Order Button

Create Order Button should be clicked when all desired items are in the order and customer is tied to the order. After the button is clicked, all information are stored in the table of orders.

7 Cancel Order Button

When the Cancel Button is clicked, the order is not saved in the database. You are returned to the Main Menu page, when all the orders are displayed.

10 Order Info Panel

11 New Customer Panel

12 Register Cust. Button

Customer Table

Add, Remove Item Buttons

13 Create Order Button

14 Update/Edit Order Button

15 Delete Order Button

Cancel Order Button

Update Table Button

5 Update/ Edit Order Button

When the 'Update Order' Button is clicked, new Menu, pops up consisting of Order Information panel. This panel has various textfields that are already filled in as you can see in the Figure no. 12 There is also a combobox where you can change the satus of the order by choosing one of the options.

When the Status of the order is being changed to Processing, as displayed in the Figure no. 12, two of the textfields change (Order Date to Processing Date and Requested Date to Estimated Delivery Date), as in the Figure no. 13.

When the Status of the order is being changed to Delivered, Figure no. 14, the two textfields are changed (Processing Date to Delivery Date and Estimated Delivery Date is not enabled anymore).

All the times, after the changes are made, the order needs to be updated, so the Save Changes Button needs to be clicked.

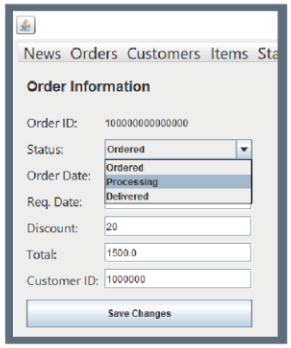


Figure no. 12

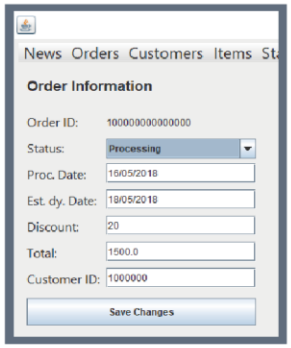


Figure no. 13

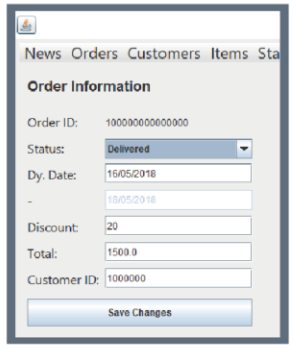


Figure no. 14

6 Delete Order Button

If the Delete Order Button is clicked, the order is deleted from the database. **(Note: SHOULD NOT BE USED UNLESS REALLY NECESSARY)**

7 Update Table Button

When the Update Table Button is clicked, the Order Table gets updated.

8 Exit the Program

To quickly exit the program, go to the top right corner and click the cross in the navigation.

Conclusion

In case of any questions do not hesitate to contact any member of the development team. Hope you will have a great experience and the software, you have just purchased, will help you in your future business.
Sincerely

Developers

CONTACT

Linda Augustina Carolus Fuchs	1067625@ucn.dk
Simon Jensen	1067303@ucn.dk
Juliana Palkova	1067676@ucn.dk
Dimitar Pilyakov	1067632@ucn.dk
Akos Skutovics	1067638@ucn.dk

Appendix G. Group Contract

Group Contract

DMAI0917 - Group 1

February 23, 2018

---

§1 Collaboration with Supervisor

It is mandatory to attend all Supervisor meeting at the specified time.

§1.1
In the event a member of the group is unable to attend a meeting, this member is to report this to the rest of the group as soon as possible.
It is only acceptable to be absent from a Supervisor meeting, if the reason is of a serious nature and cannot be rescheduled.

§1.2
All members of the group are obligated to have examined the subject of the meeting, so that every member of the group is able to participate actively in the meeting.

§2 Group meetings

It is mandatory for all group members to attend all scheduled group meetings. Unlike the meetings described in §1, these meetings have a more flexible starting time of +/- 15 minutes, unless otherwise specified.

§2.1
In the event a member of the group is unable to attend a meeting, this member is to report this to the rest of the group as soon as possible.

§3 Group Agreements

All group members are obligated to stay updated on the Facebook Messenger group chat daily.

§3.1
As a general rule an agreement in the Facebook Messenger group chat is considered accepted, unless a group member explicitly raised objections in the comments of the given post.

§3.2

During group meetings and group collaboration, Facebook Messenger chat should not be used for anything other than communicating with group members.

§4 Classes and Lectures

All group members are required to attend and/or keep up with all classes and Lectures, in order to ensure full participation in the project.

§5 Deadlines

Agreements regarding deadlines must be met. The Quality of the assignment must be satisfactory in respect to the agreement. However, specifications are subject to unforeseen workloads, these complications must be brought to the attention of the group as soon as they occur to the member responsible for the assignment, in order to re-evaluate the given deadline.

§6 Other

§6.1

All the terms, conditions and requirements set by University College of Northern Denmark must be respected.

§6.2

In case a group member does not follow the group contract, the offending member owes the rest of the group a snack for the next day.

§6.3

If the group assess that a group members violation of the terms of the group contract as serious, the given violations will be brought to the attention of the Supervisor, where any potential sanctions will be measured.