1st Semester Project

Vestbjerg Byggecenter

Linda Augustina Carolus Fuchs
Simon Jensen
Juliána Paľková
Dimitar Pilyakov

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:

1st Semester Project – Vestbjerg Byggecenter

Project participants:

Linda Augustina Carolus Fuchs

Simon Jensen

Juliána Paľková

Dimitar Pilyakov

Supervisor:

Henrik Munk Hvarregaard

Lars Landberg Toftegaard

Gianna Belle

Submission date:

13-12-2017

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

This page is intentionally left blank.

Preface

We would like to thank **Henrik Munk Hvarregaard**, **Lars Landberg Toftegaard** and **Gianna Belle** for their continuing support during this project. Their feedback was invaluable and led to the successful completion of this project.

Linda, Simon, Juliana and Dimitar.
13-12-2017

This page is intentionally left blank.

Abstract

This report documents the UCN Computer Science 1st semester project by group DMAI0917 group 1. The project concerns the creation of a store system for a DIY centre and lumber yard store, Vestbjerg Byggecenter A/S. This includes an analysis of the organisational structure, stakeholders and SWOT and a business case as a conclusion.

The report also details the system development process to clarify the requirement and design considerations and related the decision-making process. This flows into the documentation of the implementation of the system, including the code standards, to make the code more easily readable.

The final part of the report reflects on the group process and timetable evaluation.

Table of Contents

This page is intentionally left blank.

Introduction

This report was written as a part of 1st semester project by students of UCN, University College Nordjylland. The main objective of the project is to create new IT system for Vestbjerg Byggecenter A/S, selling “do it yourself” materials.

However, one of the essential purposes is to prove what knowledge members of the group have already gained in connection to business, system development and programming.

In the beginning of the report overall plan of working process was created for entire project and subsequently, the report was divided into three parts, business, system development and group evaluation.

In the business part, organisational structure, its evaluation and problems were introduced, followed by mission and vision of the company. The stakeholders and their strategic goals needed to be established and after determination of strengths, weaknesses, opportunities and threats the business case was created.

In the second part of the report, called System development, many aspects were considered. Preliminary study consisting of activity diagram, employee-task-goal table and mock-ups were done after which the functional, non-functional and information requirements of the system were easily determined. Organized by their functionality, use-case diagram, description of use-cases, domain model and system vision were constructed. After creating system sequence diagram and its operation contracts, which are considered as a part of use-case analysis, the design part was done by describing the architecture, creating interaction diagrams and design class diagram. Last, but not least, the implementation of use-cases could be accomplished followed by description of code standard and unit testing.

The very last part of the report deals with group evaluation, describing different categories that each member could assign points to according to what he thought the group work was like.

Project Plan

The project plan will go further into the separate tasks that have to be completed in order to successfully complete this project. All tasks mentioned in this plan have also been summarized into a timetable in Table 1 on the next page.

In order to make the project more manageable a decision was made to first analyse the available information. This quickly determined that the project has to be split into several different parts. By also working together in pair it makes the process quicker and more effectively by controlling other´s work as it is being written. Regular appointments ensure that the project stays on track and the group members stay focussed on the issue at hand.

Part I of the project envelops the business aspect and has to be defined before the system development process can start. This consists of an analysis of the company and defining the type of structure the company uses. Stating its mission and vision, the stakeholders and evaluating the problems that could arise. After all these aspects have been taken into consideration a business case can be set up.

Part II, which is the System Development part, starts with the preliminary study and covers the information gathering for the requirements of the system. The found requirements are then processed, defined and coded in Java using Unified Process (UP). UP is an iterative process and provides a structure for how to do object oriented analysis and design (OOA/D). UP is flexible and lends from other agile methods. (Larman, C. p19). Implementation has been included to provide insights about code standards and unit testing.

Part III describes the group process during the project. This includes the group contract, group evaluation and time evaluation.

Table 1: Timetable

Tasks	Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed
Project plan	🟦	🟦											
Part I – Business	🟩	🟩	🟩										
1. Organisational structure	🟩	🟩											
Organisational type	🟩												
Mission/ vision		🟩											
Evaluation and problems		🟩											
Organisational culture		🟩											
2. Stakeholder analysis	🟩	🟩											
Strategic goals		🟩											
3. SWOT analysis		🟩	🟩										
4. Business case		🟩	🟩										
Part II – System development			🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧
5. Preliminary study			🟧	🟧									
Activity diagram			🟧										
Employee-task-goal table				🟧									
Mock-up				🟧									
6. Requirements				🟧	🟧								
Use-case diagram				🟧									
Use-case descriptions				🟧									
Domain model				🟧	🟧								
System vision					🟧								
7. Use-case analysis					🟧								
SSD and OC					🟧								
8. Design						🟧	🟧	🟧					
Architecture						🟧							
Interaction diagram						🟧	🟧	🟧					
Design class diagram								🟧					
9. Implementation								🟧	🟧	🟧	🟧	🟧	🟧
Code implementation									🟧	🟧	🟧	🟧	🟧
Code standard									🟧				
Unit testing											🟧	🟧	🟧
Part III – Group evaluation													🟪

This page is intentionally left blank.

PART I

Business

This page is intentionally left blank.

1. Organisational Structure

This chapter focusses on the organisational structure of Vestbjerg Byggecenter as a whole, by examining the organisational-type, mission and vision. After which, this was analysed and evaluated, in order to determine potential problems surrounding to company. In addition, this was used to clarify the organisational culture and the leadership style of the prominent persons in the company.

In order to provide a solid structure to the business analysis, Organisational Behaviour (4th) by I. Brooks (2009) was used as a guideline in many of the chapters in this section.

1.1. Organisational Type

This section describes the organisational structure of Vestbjerg Byggecenter, as well as the advantages and disadvantages of this structure.

Vestbjerg Byggecenter consists of three parts, Administration, DIY Centre and Timber merchant departments. Anders Olesen is the head of the administration department, Thomas Olesen is head of the DIY Centre and Casper Olesen is head of the timber department. This results in a multifunctional organisation (I. Brooks, pp194-195) as seen in Figure 1 below.

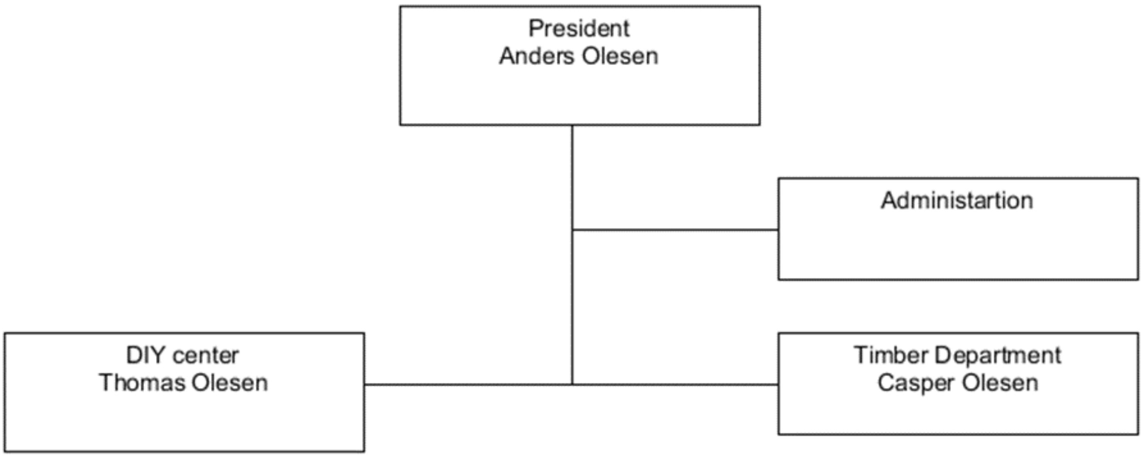


Figure 1: Organisational Chart

The multifunctional structure has clear advantages, as it facilitates a clear overview of the company for the leaders of each functional branch of the company. Additionally, it makes it easier for the departments to focus on specializing in their area of expertise, as well as a clearer view of the causes of problems. This results in a better ability to correctly assess the viability of the individual departments, so changes can be made accordingly. A counter point to be made against the multifunctional structure is that, the departments can accidentally duplicate the work done by other departments. This means the communication between department heads is of great importance.

Due to the small size of the company it is hard to pinpoint whether it is organic or mechanistic in nature. While the company shows typical mechanistic traits, like a hierarchical structure of control and specialisation in different tasks, it is also easy for any employee to talk to anyone at the top, due to the small size and effort put into staff evaluation by the President, Anders Olesen.

1.2. Mission & Vision

Mission

The mission of the company is to create a pleasant shopping experience for its customers, as well as a good work environment for the employees. The goal is to maintain a sense of community, while remaining competitive.

Vision

The company has plans to expand the company's customer base and the range of products they carry in the coming years, while maintaining a high standard of customer service. This is to ensure the continued competitiveness and profitability of the company.

1.3. Evaluation and Problems

This section further evaluates the organisational structure of Vestbjerg Byggecenter and the problems associated with it.

For Vestbjerg Byggecenter, it makes sense to split the company into divisions depending on their function, because the DIY and timber departments are fundamentally different, in that, the DIY centre caters to private customers, who need materials and tools for personal use and the timber departments caters to professional craftsmen, who need materials and tools for commercial use. A separate administrative department facilitates this, as it centralizes the administrative work, so that the leaders can easily access the bookkeeping, wages and debit-credit operations. Problems can arise with this structure, as the internal functions of the departments are not clear to the administration, when it is separate from them.

1.4. Organisational Culture

This section explores the organisational culture of Vestbjerg Byggecenter and analysis the leadership style of Anders, Thomas and Casper.

The general work environment is informal and relaxed. Everyone can come up and talk to Anders, who likes to make the rounds in the company and know what is going on in the company. Thomas and Casper are very concerned with the work environment and want to keep the employees happy. This also places a large amount of responsibility on the employees, as well as a high degree of freedom, especially in the administrative department, as Anders finds this side of the business boring and outside of his area of expertise. A large amount of trust is placed in the employees and they are expected to work hard and be responsible. If this is not followed to a satisfying degree, Anders is gives categorical feedback and if this does not help, he is not afraid to fire the person, but this rarely happens, and he as satisfied with his employees. All of this results in a "Power culture", as it relies heavily on trust and unity around the purpose of the company.

In general, the company leader very democratic in their leadership style, as they place a large amount of trust in their employees' capabilities and expertise. Anders and Thomas consult the employees to a large extend, when faces with decisions, where Casper leans a bit more towards selling the employees on his solution, because of his knowledge of the trade.

2. Stakeholder Analysis

This chapter focusses on the stakeholder analysis. A stakeholder analysis will paint a clear picture of all the stakeholders involved in the company and what their personal goals are within said company. Take the Lead (1994) by Boddy & Buchannan (pp 55-60) was used as a base reference while making this analysis.

Stakeholders are people, businesses and organisation with an interest in the company. There are many reasons why one has a stake in the company, Table 2 below shows a clear overview of the different stakeholders that have been identified and what their goals, predations, behaviour and so one are.

Table 2: Stakeholder Analysis

Stakeholder	Their goals	Past reactions	Behaviour expected	Power, legitimacy and urgency	Ideas for action
Anders Olesen	Run a successful company.	Adapting the company.	Embracing and supporting changes in the IT system.	Definitive Stakeholder	- Regular updates. - IT training.
Thomas Olesen	To keep the employees of the company happy and motivated.	Encourages change and growth.	Embracing and supporting changes in the IT system.	Definitive Stakeholder	- Ask for advice. - Involve in decision making regarding employees. - IT training.
Casper Olesen	To help employees solve their tasks and stay happy & motivated.	Encourages change and growth provided quality doesn't diminish.	Embracing and supporting changes in the IT system.	Definitive Stakeholder	- Ask for advice. - Involve in decision making regarding employees. - IT training.
Other Managers	Do their job.	*unknown*	Happy with change.	Definitive Stakeholder	- Keep informed. - IT training.
Salesman	Advertise the company.	*unknown*	Happy with change.	Definitive Stakeholder	- Keep informed. - IT training
Other Employees	Have a more user-friendly system.	*unknown*	Embracing and supporting changes in the IT system.	Dangerous Stakeholder	- Ask them for needs and wants in the system. - IT training.
Shareholders	Financial gain.	Cautious about change.	none	Dormant Stakeholder	- Keep informed.
Customers	- Buying goods. - Getting service.	*unknown*	none	Dangerous Stakeholders	- Inform when needed.

It is important to note at this point that the users of the new IT system are casual PC users and do not possess a high level of computer skills, which means the system must maintain a high degree of usability.

Stakeholder Map

The stakeholder map represents the project (in the centre) and the stakeholders that have an interest in this project. The more relevant a stakeholder is, the closer to the centre they are. This is shown in Figure 2 below.

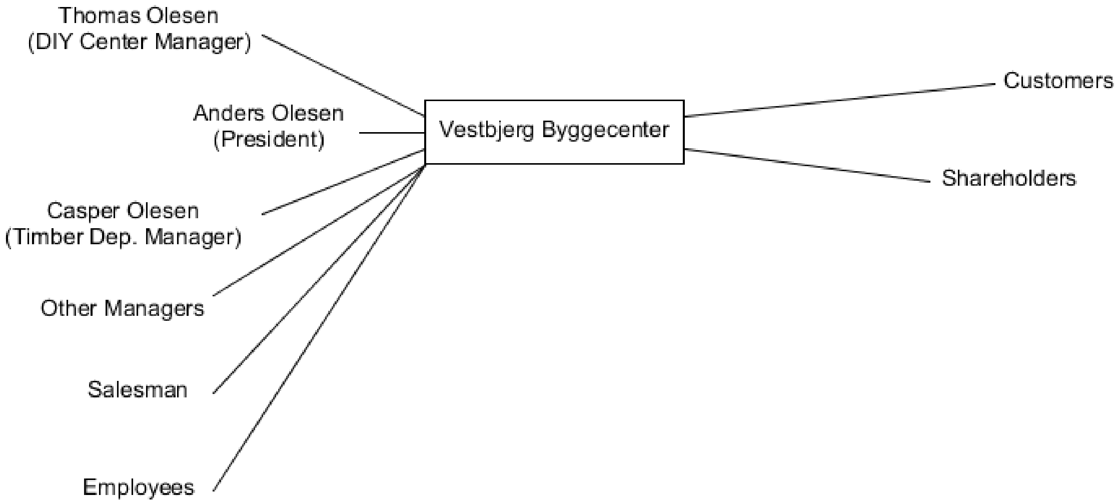


Figure 2: Stakeholder Map

Furthermore, in this particular stakeholder map, the internal stakeholders are on the left, and the external stakeholders on the right.

2.1. Strategic Goals

The strategic goal explains in a SMART (Specific, Measurable, Assignable, Realistic, Time-related) way what the plans are for the future and what the strategy is to get there.

The company has stated a desire to expand in the coming years. To increase both their amount of customers and the variety of products they sell. Implementing a new IT system, which is more functional and user-friendly than the current one, would improve the employees' performance and make it easier for them to do their day-to-day tasks, thus increasing company efficiency, decreasing managerial responsibilities and eventually increase their profitability and competitiveness.

3. SWOT Analysis

SWOT analysis is a grouped list of business' strengths, weaknesses, opportunities and threats. This analysis helps people to develop solid business strategy. Difference between internal and external factors is ability to change them. Internal origins are strengths and weaknesses, which can be changed. Opportunities and threats are external to the company, so there isn't possibility to change them.

*"An assessment of internal resources and competences(**S**trengths and **W**eaknesses) in relation to conditions in an organisation's external environment (**O**pportunities and **T**hreats)."* (W. Bloisi p87)

Table 3 below represents a breakdown of the company's main Strengths, Weaknesses, Opportunities and Threats. The 4 categories have their respective initials in the background. We have separated the categories by what may be helpful or harmful to the company and by the sources: internal or external.

Table 3: SWOT Analysis

	Helpful to achieving the objective	Harmful to achieving the objective
Internal origin (attributes of the organization)	- Many regular customers. - Diversity in products. - Leasing out items. - Cooperation with other companies. - Relevant education of leaders. - Knowledge at the top. - Good relationship with employees. - Feedback on employee´s work.	- Discounts up to 20%. - Inefficient ordering of goods. - No location codes. - Handling online orders manually. - Keeping eye on lent tools manually. - Not adapting to technological advances.
External origin (attributes of the organization)	- Rural location. - New IT system. - Expanding the selection of products.	- Other competitors – Stark, Bauhaus, Jem & Fix. - Cost of materials increasing. - Small yearly income.

Strengths

One of the strengths of the business is regular customers. Thanks to having approximately 1000 regular customers, the company has a fixed income. Having a wide range of items helps the company to fully satisfy its clients. Leasing out about 30 different items like carpet cleaners, grinding machines, and other larger tools distinguishes the company from others. Another strength is cooperation with different companies, which gives the ability to buy goods in bulks or having a voice in case that any new regulation appears. Relevant education of the leaders helps the company to properly lead itself. Last, but not least is the president's experience and knowledge about trade. He strives to have a good relationship with all his employees, motivate them and give them feedback on their work. As a result, employee related problems are quickly and easily solvable.

Weaknesses

Discounts for regular clients are up to 20%, which is one of the weaknesses of the company, because it loses a lot of money. Another weakness is an inefficient goods ordering system. The note which needs to be brought when customer wants to pick up items is fraud sensitive or can just easily get lost. There is a problem with localization of items, because the company doesn´t have any specified code for each department, so the staff has no idea where the item is.

Another weakness is the handling of online orders, as this is done manually and costs a lot of time. The company misses a system that would be integrated with XL- byg, which is the reason why the employees have to spend a lot of time doing it manually. There currently is no leasing system, at the moment this is done manually by the employee and they have to manually keep track of which tools and machines are currently lent out. This could be done much faster and more accurate using a computer system that simply keeps track of them all. On top of this, not having a modern IT system means they are unable to keep up with and adapt to technological advances.

Opportunities

A big opportunity is a new IT system, which will improve the company. Because of it, manual activities can be automated. Another very huge opportunity is that thanks to the rural location of the company, the customers living outside the city centre having to take care of their farms will visit the store more. One more opportunity is to expand the current selection of products.

Threats

The most important threat is competitors. Stores like Bauhaus or Stark have more accessible location for residents of Aalborg. These competitors have also adapted to technological advances better than Vestbjerg Byggecenter, making them exceedingly more efficient in comparison. Another threat is the increase of material costs, specifically timber, which will further increase the costs for the company. The next problem of the company is in relation to this: its small yearly profit. This will eventually lead to more costs than profit.

4. Business Case

This chapter describes the business case of Vestbjerg Byggecenter in relation to implementing a new IT-system by examining the available options, conducting a cost-benefit analysis and the impacts and risks of the implementation.

Introduction & background

This document aims to clarify the purpose of introducing a new IT-system into the company and the potential pitfalls thereof. The current IT-system is an old UNIX system, which cannot handle the information and demands of the company, as it has evolved and expanded. This justifies almost any improvement and/or reinvention of the system, to facilitate further growth.

Summary

The current system has several major problems regarding, inventory control, bookkeeping, management, website compatibility and usability.

This project aims to solve most of these problems by integrating a new IT-system into more aspects of the company and improving the functionality of the inventory control and bookkeeping.

Improvements could be made to the existing system, but as it is outdated, it would not reach the required functionality, integration or usability, without near insurmountable hurdles and even then, the compatibility new other current and future system would be limited.

Descriptions of problems and opportunities

Inventory control: The current inventory system does not include locations of the items, which makes it difficult to keep track if items, which makes the inventory management a larger and more frustrating task. Additionally, this does not encourage an organised warehouse from a systematic standpoint.

Bookkeeping: In the current system there is a lot of manual bookkeeping, partly because the system does not communicate with the online ordering.

Management: The new system should give a clearer view of the employees' performance and data, in a way that is helpful to the management, when collecting statistics of each employee, the departments and the company in general.

Usability: The leaders, management and employees are very tired of the current system, because it is outdated and lacks a lot of functionalities, which means that the new system must be user-friendly and comprehensive.

Options available and considered

The current IT-system is still useable and could be for a long time, but it would be increasingly costly to keep using in term of manhours, human error and frustrations.

More employees could be hired to combat the current and future logistical issues of the company, but this results in more expenses for salaries, which is not a feasible long-term solution to a systemic problem, especially considering the company's current economy, which does not allow for large additional expenses.

The company could also withdraw from the XL-Byg collaboration in order to eliminate the troublesome internet ordering system, as this would allow the management to downsize the administration department, but this is again a short-term solution to a persistent problem. Additionally, this would decrease both cost and expenses, but as more and more people shop online the profit to be made would undoubtedly outweigh the costs in the long-term.

The recommended approach is to implement an entirely new IT-system, as this would address the root of the problems instead of treating the symptoms there of. This allows for more streamlined administrative tasks as it can be integrating with more current systems and tailored to a more IT heavy time. Additionally, a new system can be tailored to the company instead of adapting outdated software to accommodate new requirements.

Create a cost-benefit analysis

The cost-benefit analysis clarifies the pros and cons to the recommended solution before committing to it to help determine, if the project is worth starting before spending too many recourses on it. Table 4 below shows the cost-benefit analysis for the recommended solution.

Table 4: Cost- Benefit Analysis

Cost		Benefit	
Tangible	**Intangible**	**Tangible**	**Intangible**
Learning period (time)	Frustration about new knowledge (Happiness)	Less manual administrative work (time)	Eliminate frustration about outdated system (Happiness)
New hardware (server to handle online orders(money))	Possible firing of unnecessary staff (Happiness)	Items in stock can be located more easily (time)	Scaling for the expansion of the company (peace of mind)
Hardware maintenance (money)		Downsize staff due to efficiency (money)	Clearer overview of departments (knowledge)
Software maintenance (money)			

As part of the cost-benefit analysis, a payback projection is very helpful to determine, how large of an investment is needed to see the project through to completion and how long it would ideally take from the project to pay for itself. Table 5 on the next page shows the payback projection for the coming five years.

Table 5: Payback Projection

Item	Year 1	Year 2	Year 3	Year 4	Year 5
Server rental	3,500	3,500	3,500	3,500	3,500
Hardware purchase	100,000				
Hardware maintenance	2,000	2,000	2,000	2,000	2,000
Software purchase	20,000				
Software maintenance	500	500	500	500	500
IT Training	30,000				
Cumulative total costs	156,000	162,000	168,000	174,000	180,000
Staff savings per year	300,000	300,000	300,000	300,000	300,000
Cumulative savings	300,000	600,000	900,000	1,200,000	1,500,000
Cumulative savings less costs	+144,000	+438,000	+732,000	+1,026,000	+1,320,000

Impact and risks

The implementation of a new IT-system has an enormous impact on a company, as almost everything goes through IT-systems these days. Therefore, the implementation of a new IT-system has a lot of risks associated with it, as missing functionalities, poor usability and/or bugs can have devastating consequences for the company.

Conclusions and recommendations

The conclusion of this, is that the project is worth doing, as the current setup is causing a lot of unnecessary manual work and a lot of frustration within the company. The recommendation is to implement a completely new IT-system, as this is one of the main problems with the business and is contributing to a lot of the other problems by introducing additional difficulties to the workflow. According to employee Henrik Munk Hvarregaard “Yes, this project is worth doing.”.

This page is intentionally left blank.

PART II

System Development

This page is intentionally left blank.

5. Preliminary Study

This chapter will go into the details of the preliminary study about the new IT system for Vestbjerg Byggecenter. In order to establish the proper demands for the system an activity diagram has been created. This diagram has then been used to set up an employee-task-goal table, both of which have helped considerably in establishing the use cases for the new system.

In order to provide a solid structure to the System Development Process, Applying UML and Patterns (2004) by Larman, C. was used as a guideline in many of the chapters in this section.

5.1. Activity Diagram

In order to analyse what would be included in the system, it was important to establish the current employee workflows and adjust these with the wishes and desires for the new IT system. Following are these workflows in an several activity diagrams for the most important parts of the employee's day to day work schedule.

Create Sale

The activity diagram below in Figure 3 gives an overview of the tasks the employee has to perform in the new system in order to successfully sell an item to the customer.

In this activity diagram the workflow of the employee is displayed while he or she sells a product to a customer. The employee authenticates him/herself and starts by inputting the item barcodes. Once no more items are left on the belt the order gets closed, discounts get added and payment is made.

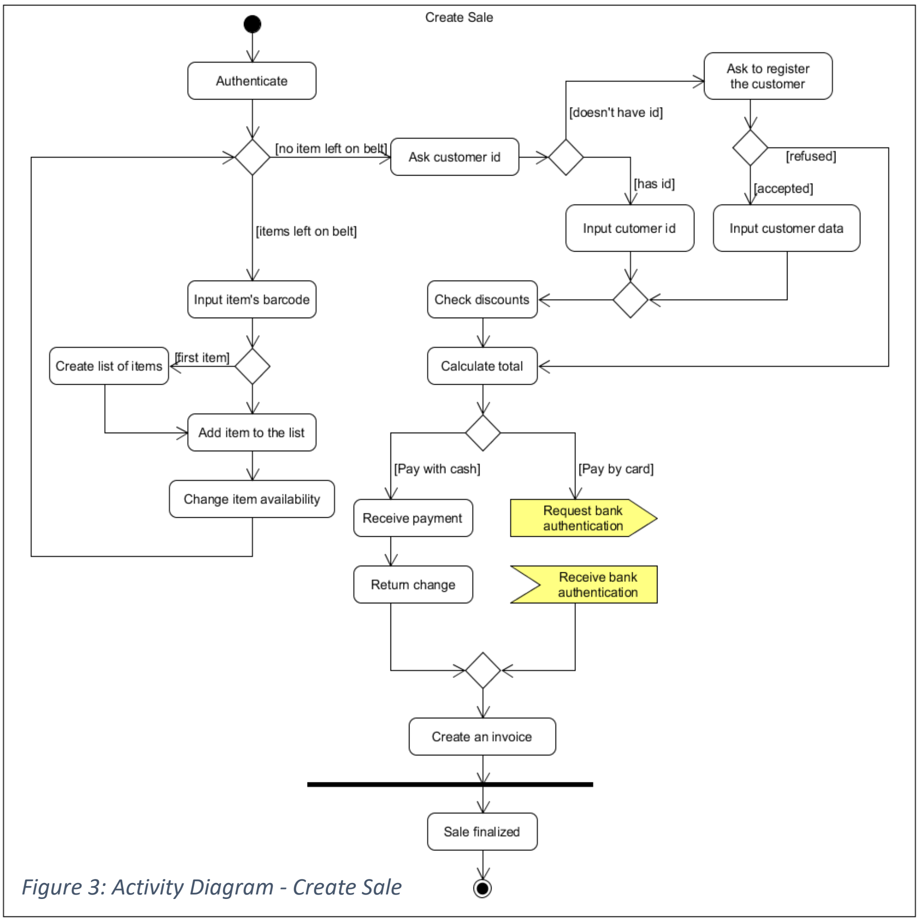


Figure 3: Activity Diagram - Create Sale

Create Loan

Whenever a customer wants to loan an item, for example a sander or a drill, the employee has to register the loaned item in the system and link it to a customer's information. Lending is not possible without adding (or selecting) customer information. The following three diagrams show the workflow for creating a loan (Figure 4), handing out the loaned item (Figure 5) and receiving the loaned item back from the customer (Figure 6).

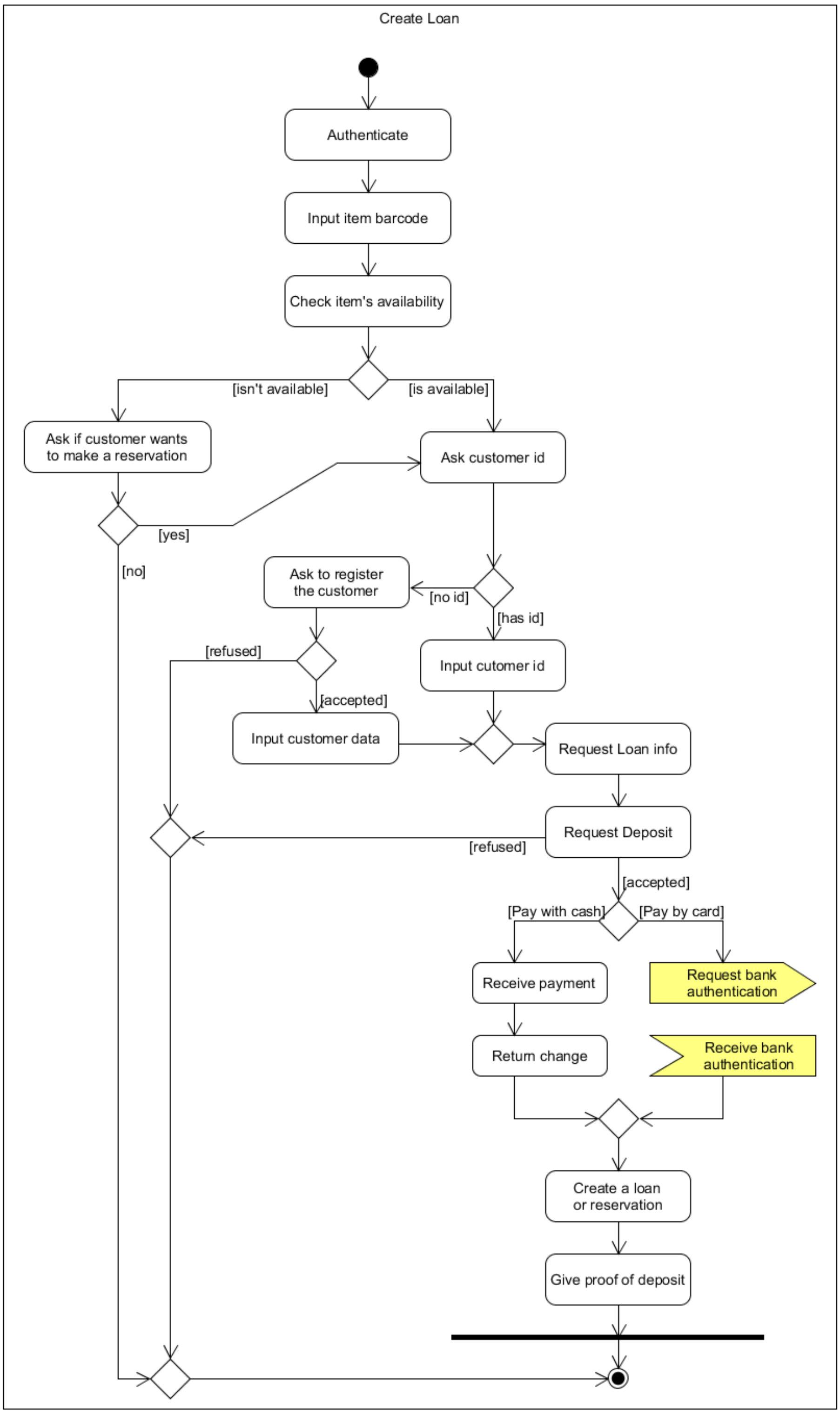


Figure 4: Activity Diagram - Create Loan

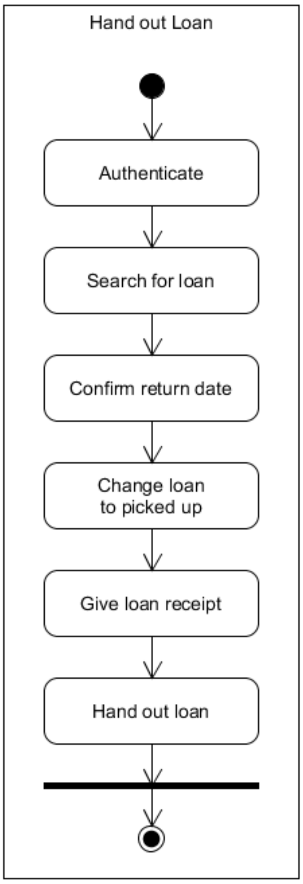


Figure 5: Activity Diagram - Hand out Loan

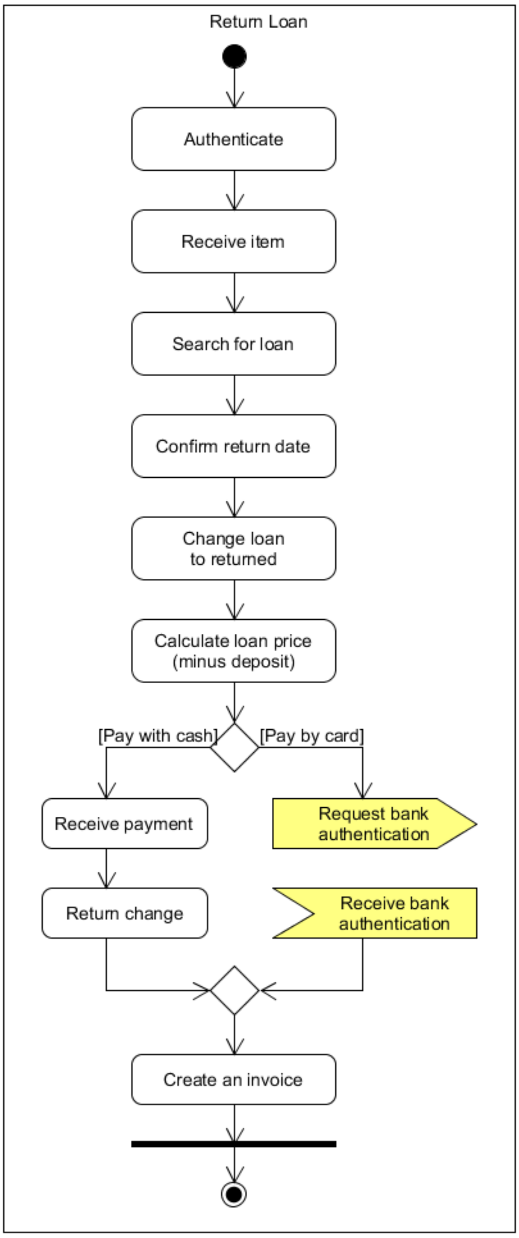


Figure 6: Activity Diagram - Return Loan

When a loaned item is returned the employee checks if there are any additional charges on top of the normal lending fee and proceeds to request payment. Afterwards an invoice is created and handed to the customer.

Create Order

'Create order' handles orders done by the webshop on XL Byg and orders done in person in the store. In case of an order from XL Byg webshop, the customer will have already filled in their personal information and have paid for their order before it gets sent to the company, so these two processes can be omitted. Orders done in the physical shop however will have to include this. Figure 7 and Figure 8 on the next page display the activity diagrams from online order and in store order respectively.

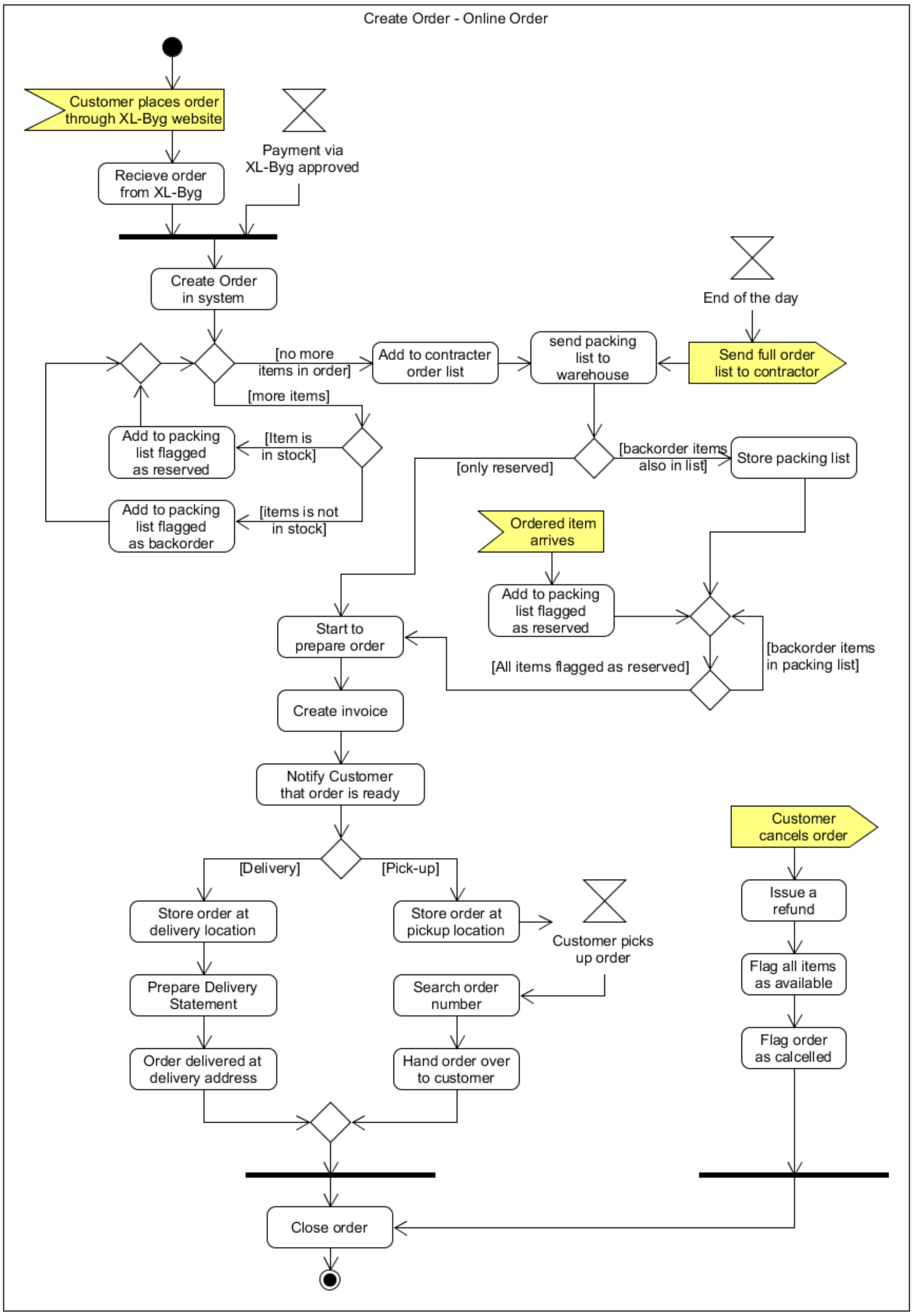

Create Order - Online Order
Customer places order through XL-Byg website
Payment via XL-Byg approved
Recieve order from XL-Byg
Create Order in system
End of the day
[no more items in order]
Add to contracter order list
send packing list to warehouse
Send full order list to contractor
[more items]
Add to packing list flagged as reserved
[Item is in stock]
[backorder items also in list]
Store packing list
[only reserved]
Add to packing list flagged as backorder
[items is not in stock]
Ordered item arrives
Add to packing list flagged as reserved
Start to prepare order
[All items flagged as reserved]
[backorder items in packing list]
Create invoice
Notify Customer that order is ready
Customer cancels order
[Delivery]
[Pick-up]
Store order at delivery location
Store order at pickup location
Customer picks up order
Issue a refund
Prepare Delivery Statement
Search order number
Flag all items as available
Order delivered at delivery address
Hand order over to customer
Flag order as calcelled
Close order

Figure 7: Activity Diagram - Create Order: Online Order

The two diagrams look fairly similar once the packing list has been sent to the warehouse. As mentioned before the only exception being that the payment method is omitted from the online order system. Additionally, the physical in store order does not include a refund option when the order gets cancelled, as payment is done after delivery/ pickup. It is still possible to get a refund for the order afterwards, this is not included in this diagram as the order is considered closed after payment. A refund would be a separate workflow.

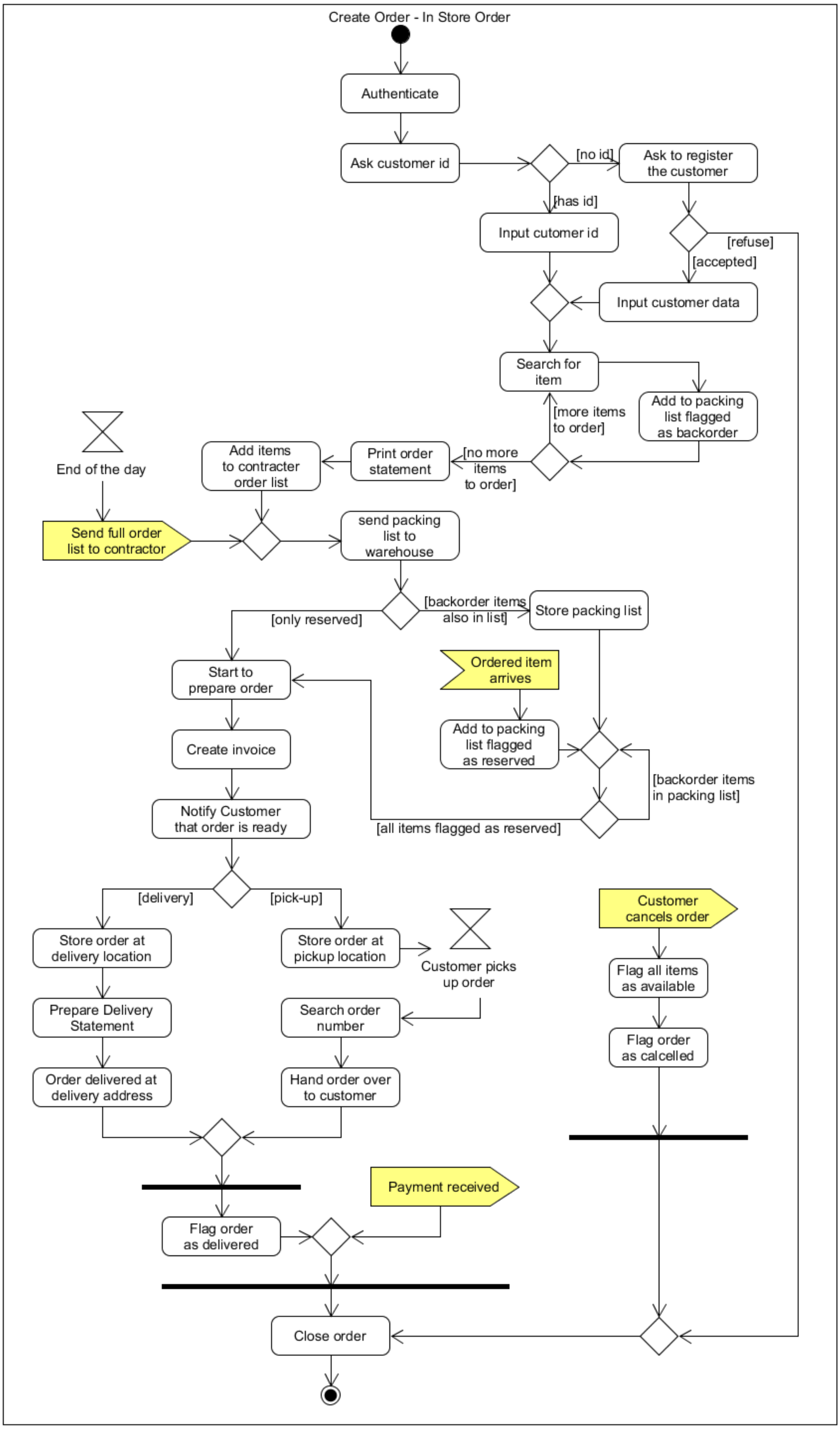

Create Order - In Store Order
Authenticate
Ask customer id
[no id]
Ask to register the customer
[has id]
Input cutomer id
[refuse]
[accepted]
Input customer data
Search for item
Add to packing list flagged as backorder
[more items to order]
[no more items to order]
Print order statement
Add items to contracter order list
End of the day
Send full order list to contractor
send packing list to warehouse
[backorder items also in list]
Store packing list
[only reserved]
Ordered item arrives
Add to packing list flagged as reserved
[backorder items in packing list]
[all items flagged as reserved]
Start to prepare order
Create invoice
Notify Customer that order is ready
[delivery]
[pick-up]
Store order at delivery location
Prepare Delivery Statement
Order delivered at delivery address
Store order at pickup location
Customer picks up order
Search order number
Hand order over to customer
Customer cancels order
Flag all items as available
Flag order as calcelled
Payment received
Flag order as delivered
Close order

Figure 8: Activity Diagram - Create Order: In Store Order

5.2. Employee-Task-Goal Table

Once the complete overview of the tasks had been established it was possible to create an Employee-task-goal table. This table, as seen in Table 6 below, depicts the individual tasks the actor, in this case employee or manager, has to perform in order to reach the goal.

Table 6: Employee-Task-Goal table

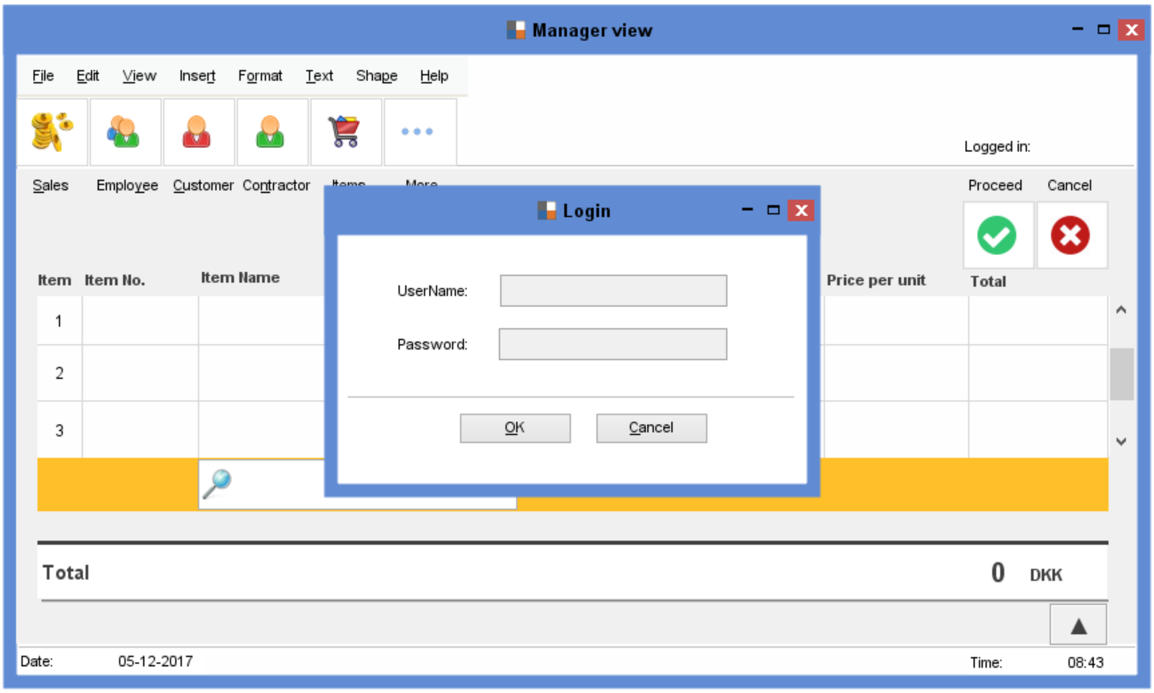
Actor	Task	Goal	Step in task
Employee AND Manager	Register a customer	Customer registered	Authenticate Input customer data Enrol to the database of regular customer Print confirmation
	Sell goods	Goods sold	Authenticate Input item barcode Add item to the list Change item availability If customer is regular: Input customer id Calculate total & tax. Authorise payment Create an invoice
	Register an item	Item registered	Authenticate Input item data Add item to the item database Print confirmation
	Loan item	Item lent	Authenticate Input item barcode Input customer id If item is available and customer with this id exists Change availability of the item Create a loan Print confirmation
	Receive lent item	Lent item received	Authenticate Input customer id Input item barcode Set item location Set availability of the item
	Create Order	Order created	Authenticate input customer id input item barcodes add item to order
	Get statistic about item	Statistic about item got	Authenticate Input items barcode Receive item statistics.
	Get statistic about customer	Statistic about customer got	Authenticate Input customers id Receive customer statistics
	Get information about contractor	Information about contractor got	Authenticate Input contactor info Receive contractor statistics.

Manager only	Register employee	Employee registered	Authenticate Input employee data Receive confirmation
	Set price of item	Item price is set	Authenticate Input item barcode input price Receive confirmation
	Set discount	Discount set	Authenticate Input or select discount category Input new discount Receive confirmation
	Register new contractor	Contractor registered	Authenticate Input contractor data Receive confirmation

5.3. Mock-up

After the initial goals and tasks were identified it was possible to make a first mock-up of the system. By creating a mock-up this early on in the process it is possible for the company to decide which direction they wish to go to in, in terms of the general appearance and functionality of the system. The mock-ups have also helped in identifying both functional and non-functional requirements.

The mock-ups that have been chosen display the manager view of the system. The manager is able to input information about employees, customers, items, modify them and can make a new sale. Because the regular employee cannot have access to other colleague’s information nor update them, the only difference between manager and employee view is “employee” option, which is not accessible from employee view. Figure 9 below displays the login screen that every user sees when they try to authenticate.



Total 0 DKK

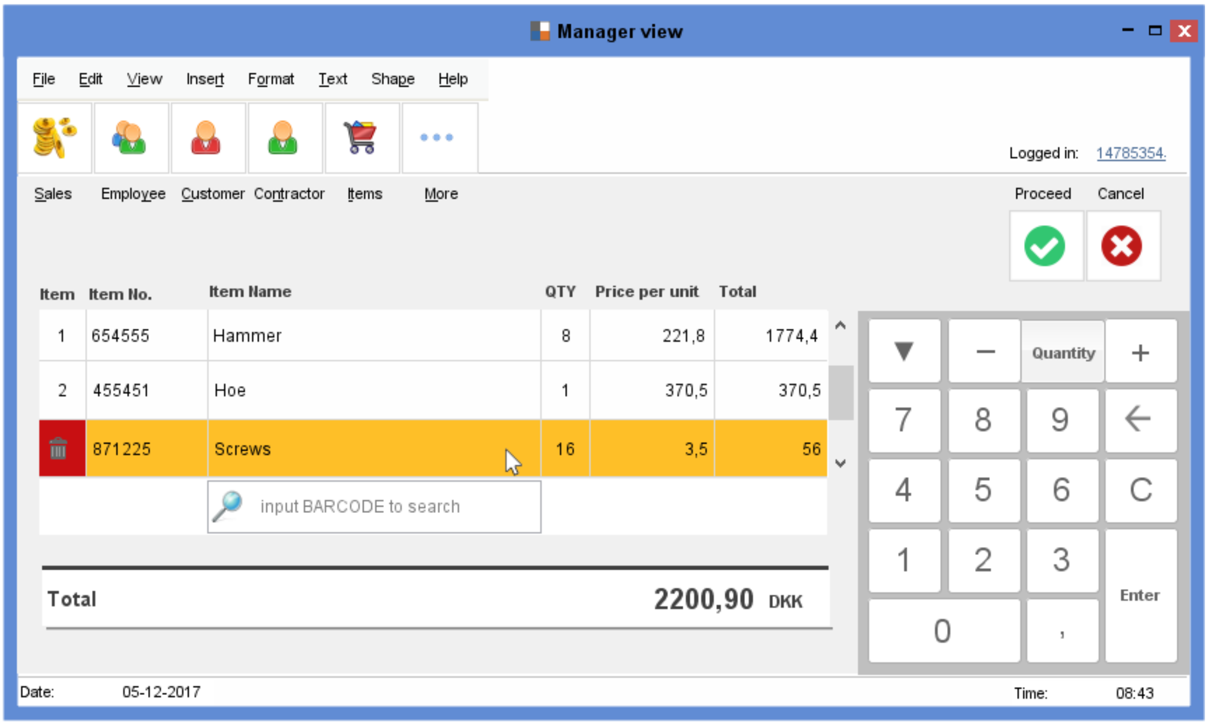
Date: 05-12-2017

Time: 08:43

Figure 9: Mock-up - Login

Create new sale

While creating a new sale, the barcode needs to be either scanned or manually input into the sale. In the case when the barcode is input manually, the user needs to open the calculator and type the barcode. From the calculator it is also possible to set quantities of products. After the barcode has been input, the product is displayed as in the picture Figure 10 below. The same applies to the following products until they are all scanned. In the case that a user wants to remove the item from the list of scanned items, he can click on the item and a small "bin" icon is displayed on the left side of the product row which will delete the product from the product row. Below the list of all items, the total amount to pay is displayed and updated every time a new item is scanned. The user then clicks "proceed" to continue the sale.



Logged in: 14785354.

Item	Item No.	Item Name	QTY	Price per unit	Total
1	654555	Hammer	8	221,8	1774,4
2	455451	Hoe	1	370,5	370,5
🗑	871225	Screws	16	3,5	56

input BARCODE to search

Total 2200,90 DKK

Date: 05-12-2017

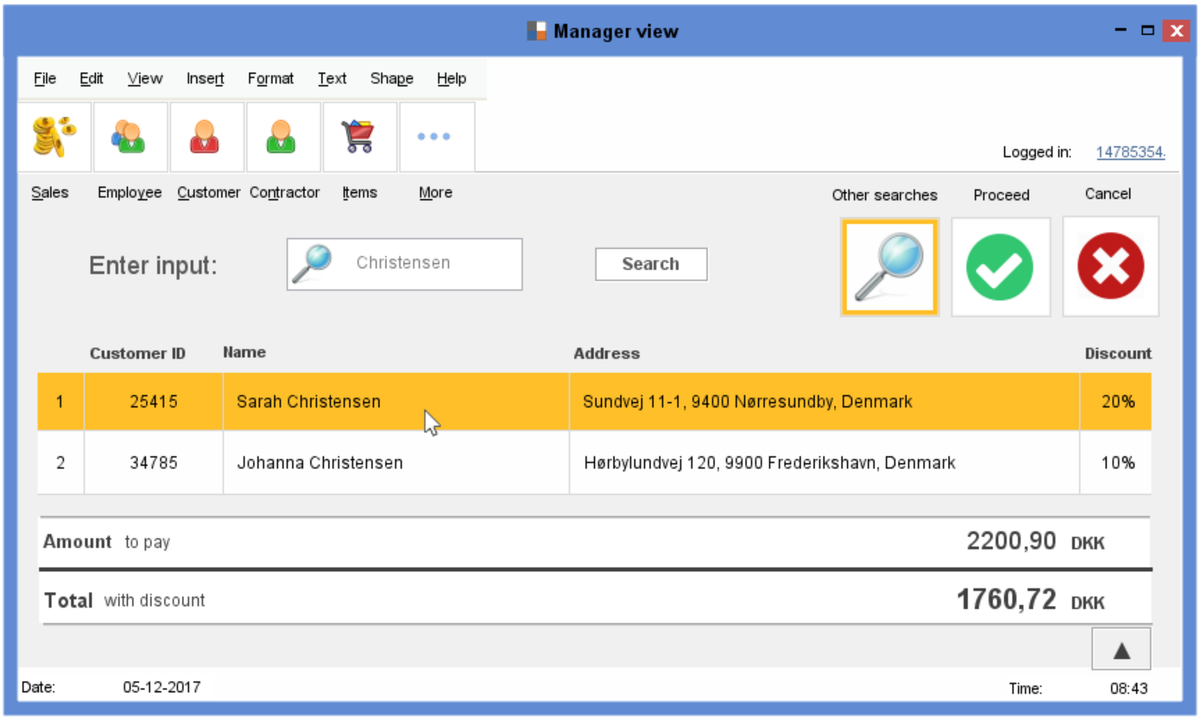
Time: 08:43

Figure 10: Mock-up - Sale

Apply customer's discount

After scanning all the items, the user gets to the point where the discount can be applied. By default the search will be set to search for customer ID. Searching by ID will displayed the calculator where he can enter the id number. When searching only one result (customer) will be displayed because the ID is unique. After clicking on the customer, the discount is automatically applied to the sale.

However it is possible to perform other searches in case the customer cannot remember their ID, this is done by selecting the 'other searches' button on the right side of the screen. Different searching options are displayed afterwards, because it is possible to search by name, address or phone number. In the example on the next page in Figure 11 the "searching by name" option has been chosen. After typing the name of the customer, all the customers with the same name are displayed.



Logged in: 14785354

Enter input: Christensen

	Customer ID	Name	Address	Discount
1	25415	Sarah Christensen	Sundvej 11-1, 9400 Nørresundby, Denmark	20%
2	34785	Johanna Christensen	Hørbylundvej 120, 9900 Frederikshavn, Denmark	10%

Amount to pay 2200,90 DKK

Total with discount 1760,72 DKK

Date: 05-12-2017

Time: 08:43

Figure 11: Mock-up - Search Customer by name

Then, just by choosing the right one, the discount is applied the same way as before. Prices before and after applying the discount are shown right below the results that were found after searching. In the case that a user changes his mind and wants to search by customer ID instead, he just unclicks the “Other Searches” option on the right side of the picture and goes back to search customer menu. To continue in sale, “proceed” needs to be clicked on.

6. Requirements

This chapter will further explain the use cases that were found in the preliminary study. The use-cases will be laid out in a use-case diagram which depict how and which actors can use the new system. Following are the brief- casual- and fully-dressed use-case descriptions. Out of these the domain model was made and finally the system vision.

6.1. Use Case Diagram

The use case diagram below in Figure 12 shows the interaction between different actors and the system and which actor has access to which part of said system.

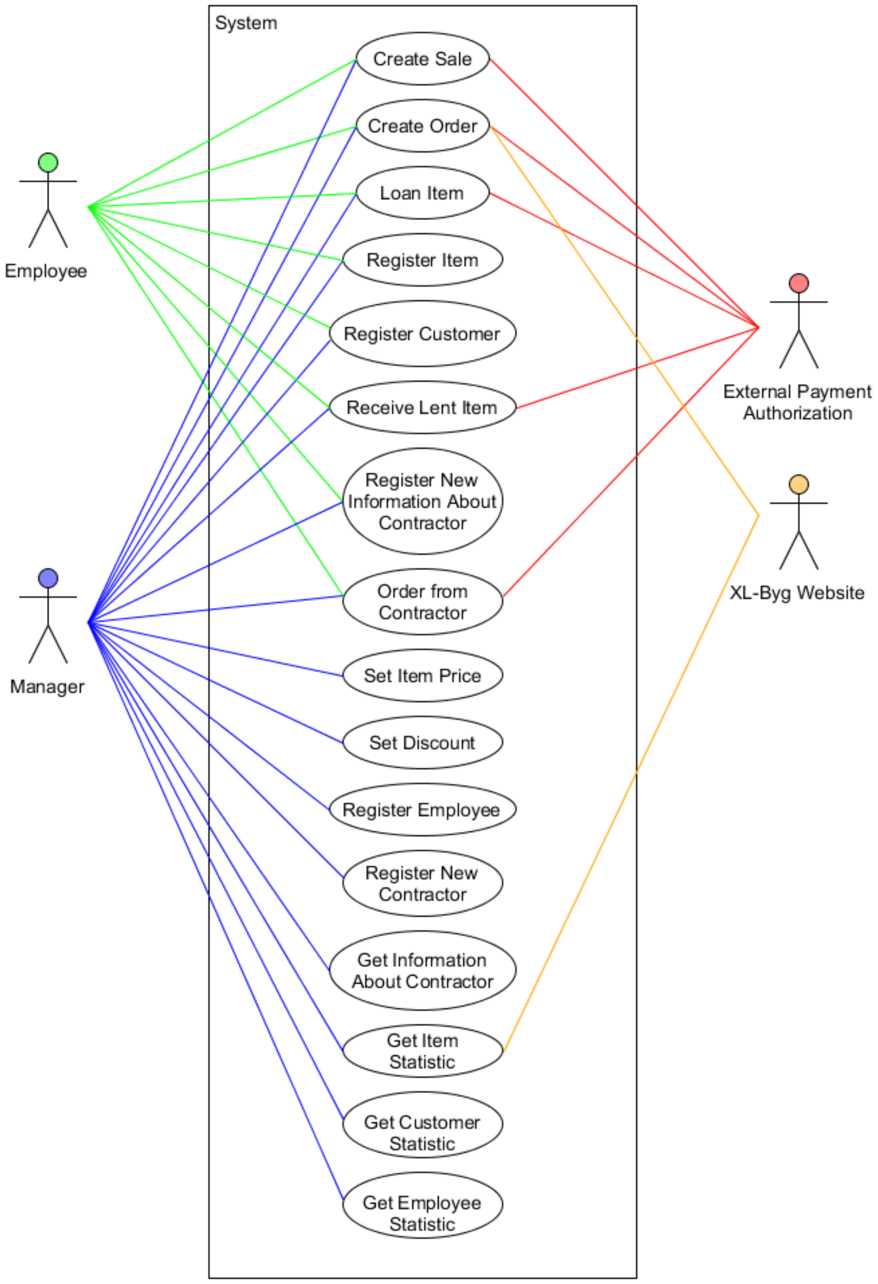


Figure 12: Use-case Diagram

Description use case Diagram

The system has to be able to handle the sales, loans and orders for the customers to be able to buy, loan and order items from the store. It must also handle the inventory system as well as the administration of customers, employees and contractors.

There are two main actors, managers and employees, who interact directly with the system. Managers have access to all use cases, as they are in charge of the administration of the company and needs to be able to see the statistics of the company, in order to make informed decisions. Employees only have the ability to make sales, loan and receive lent items, create orders and register items and customers, as well as new information about existing contractors.

Additionally, the system must communicate with two other systems, as secondary actors, the external payment system and XL-Byg website. The external payment system (PBS) handles payments with debit and credit cards, as the bank information has to be verified for the order to go through. The XL-Byg website needs to be able to communicate with the system directly, as the old system did not, thereby creating a large amount of manual administrative work.

6.2. Use Case Brief Descriptions

This paragraph lists all the use cases in the system. Most have been described in either a brief- or casual form, however some have been described in more detail in Chapter 6.3, in fully dressed form. These are the most complex use cases for the system, and the ones that will be implemented first.

Create Sale

The create sale use case is the primary function of this system. Because of this, this use case will be the first one that will be implemented into the system and is therefore described in fully dressed form. Please refer to Chapter 6.3: Use Case Fully Dressed Description.

Create Order

The use case Create Order handles orders from both online order placements through XL Byg's web shop, as well as in person order placements done in the store itself. Below both cases are described in casual format, including possible alternate scenario's.

Online order

The order placed on XL Byg's website gets automatically sent to the system. It creates an order list which inputs every separate item and flags them either as reserved (in stock and thus reserved for the customer) or as ordered (not in stock, ordered from Contractor). All items flagged as ordered will be added to a contractors order list and will be sent out to the contractor at the end of the day. When all items are in stock a message will be sent to the warehouse to start packing the order. Once order is packed and ready to ship, the system will send an automatic e-mail informing the customer that the order is ready to be picked up / will be delivered soon. Up until this point it is possible for the customer to cancel the order and get a full refund, however once the order has started shipping, this shipping charge cannot be refunded. Once the order has been delivered or picked up the order will be flagged as closed.

Online order alternate scenario's

* The system does not receive the orders from XL Byg automatically.
* The system does not create an order list automatically.
* Items do not get added to the packaging list as reserved or ordered.
* Items do not get added to the contractor ordered list.
* Contractor order doesn't get sent out automatically at the end of the day.
* Items do not get flagged as reserved when they arrive in stock
* Packaging list does not get flagged for ready to start packing once all items are in stock.
* Automatic e-mail to customer does not get sent once the order is ready to be shipped/ picked up.
* System sends multiple e-mails to the customer that the order is ready to be shipped/ picked up.
* The system doesn't change the state of the order (ready, delivered, picked up, closed, cancelled).
* The order doesn't get cancelled when a customer cancels.
    * Resulting from this: The order gets delivered anyway, resulting in additional charges for the company

In store order

A customer approaches an employee with the intent to order an item that is out of stock in the store. The employee authenticates and asks for the customer id so he can attach the ordered item to the customer. Then he searches for the items to order and adds them to the packing list flagged as 'backorder'. After approving the list the list gets printed as proof for the customer. All items get added to the contractor order list which gets sent out at the end of the day. The packing list will be sent to the warehouse and checked if any items on the list are marked as 'backorder'. As long as an item is in backorder the order will be kept in storage with every delivered item marked as reserved. Once all items on the list are marked as reserved the order will be packed. Invoice will be created and a notification is sent to the customer. Up until this point it is possible for the customer to cancel the order, however once the order has started shipping the shipping charge cannot be refunded. The customer can have the order delivered or pick it up themselves. In both cases the payment is done when the customer receives the order. After receiving payment the order is considered closed.

In store order alternate scenario's

* Items do not get added to the packaging list as reserved or ordered.
* Items do not get added to the contractor ordered list.
* Contractor order doesn't get sent out automatically at the end of the day.
* Items do not get flagged as reserved when they arrive in stock
* Packaging list does not get flagged for ready to start packing once all items are in stock.
* Automatic e-mail to customer does not get sent once the order is ready to be shipped/ picked up.
* System sends multiple e-mails to the customer that the order is ready to be shipped/ picked up.
* The system doesn't change the state of the order (ready, delivered, picked up, closed, cancelled).
* The order doesn't get cancelled when a customer cancels.
    * Resulting from this: The order gets delivered anyway, resulting in additional charges for the company

Loan Item

Manager authenticates, input item barcode and checks availability. After input in form of customer´s id, availability of item changes and loan is created. Loan is then added to list of loans.

Register Item

Manager or employee authenticates and inputs all data needed for registering item. After item is registered it is added to the system of all items.

Register Customer

Manager or employee authenticates and is then able to register new customer. He chooses register customer button and he writes all customer´s data needed for making new registration. After the input is done, new customer is registered in the system.

Receive Lent Item

Manager or employee authenticates, chooses "return item" option and writes all required input. Item is returned, and its availability is changed.

Register New Information About Contractor

Manager or employee authenticates and chooses the option that navigates him to all the contractors company has. He then chooses which contractor he wants to register information about. Information is registered.

Order from Contractor

Online and in store orders add orders to order lists from contractors. The system automatically sends out the orders at the end of the day without any help or approval from management. If needed management can cancel an order. Depending on the terms stated in the contract management can pay all orders at once at any given time interval (for instance at the end of the week). The external payment system requests payment authentication from bank. After approval orders are flagged as paid.

Set Item Price

Manager authenticates, navigates to the item he wants to change, inputs new price, confirms it.

Set Discount

Manager authenticates and set discount by choosing Discount option. Then he chooses discount witch he'd like to change. After that he set discount and save it.

Register Employee

Manager authenticates and chooses Employee option. Then he input all requested data about the employee and confirm registration.

Register New Contractor

Manager authenticates and chooses "contractor" button. Then, he chooses "register new contractor" option and writes all information needed for registering. After that, new contractor is registered and information saved in database.

Get Information About Contractor

Manager authenticates and chooses Contactor option. After that he writes contactor's identifying data and search all information about the contractor.

Get Item Statistic

Manager authenticates and chooses Item option. Then he chooses Statistic and statistics about all items are shown.

Get Customer Statistic

Manager authenticates and chooses Customer option. Then he chooses Statistic and statistics about all employees are shown.

Get Employee Statistic

Manager authenticates and chooses Employee option. Then he chooses Statistic and statistics about all employees are shown.

6.3. Use Case Fully Dressed Description

Below follow the use cases described in fully dressed form, including alternative scenarios.

Create Sale

Create sale is a large use case that not only handles the flow of items from company to customer, but it touches on many more complex tasks such as finding the sales item in the database, connecting the sales item with the employee that sold it, creating an order list for the customer, and so on. The complete description is found in Table 7 below.

Table 7: Fully Dressed Use-case - Create Sale

Use Case Name	Create Sale – Physical sale	
Scope	Sales part of IT system	
Primary Actor	Employee	
Secondary Actor	Customer	
Preconditions	Employee authenticated, Customer has an item they want to purchase	
Postconditions	Successful sale	
Frequency of Occurrence	50 per hour	
Flow of events	**Actor**	**System**
Main Success Scenario	1. Employee enters the item barcode into the system	2. Registers the barcode
		3. Searches for the item's price, information
		4. Calculates total amount due by the customer. (Total amount calculated from a set of price rules)

	Steps 1 - 4 are repeated until all the items are entered into the system	
		5. Presents total with discounts calculated
	6. Employee tells Customer the total and asks for payment	
	7. Customer pays	8. Registers the payment
		9. Logs completed sale
		10. Sends payment request to external banking system
		11. Receives approval from banking system
		12. Creates invoice
		13. Prints receipt
	14. Employee hands Customer receipt	
	15. Customer leaves with receipt & purchased items	
Alternative Scenarios	*a. Manager requests an override for the sale 1. Manager enters manager-authorized mode in the System 2. Manager or employee performs one manager-authorized mode action (e.g. void a sale, resume a sale on a different register, remove an item from the sale line, etc.) 3. System reverts to Cashier-authorized mode	
	*b. System fails 1. Employee restarts system 2. Employee authenticates 3. Employee requests recovery of previous state 4. System attempts to reconstruct previous state 4a. System fails to reconstruct previous state 1. System displays an error message 2. System logs the error 3. System loads a clean state 4. Employee starts a new sale	
	1a. Customer/Manager want to resume a suspended sale 1. Employee requests for the system to resume the suspended sale 2. Employee enters suspended sale ID 2a. Sale not found 1. System displays an error message 2. Cashier starts a new sale 3. Cashier continues sale	
	2-7a. Customer asks Employee to remove (void) one or more items 1. Employee enters Item ID in order to remove it from the current sale 2. System removes the item and updates the total	
	2-7b. Customer asks Employee to cancel sale 1. Employee cancels sale in System	
	2-7c. Employee suspends the sale 1. System records the sale state, so that it is available on any other register 2. System provides the necessary sale ID to the Employee 3. System provides a sales receipt, in case the sale ID fails	

	3a.	System fails to find item information 1. System signals error and rejects the entry 2. Employee responds: 2a. There is a human-readable item ID (a numeric UPC) 1. Employee manually enters the ID 2. System displays the price 2a. Invalid item ID: System signals error. Cashier tries an alternative method 2b. There is no human-readable item ID, but there is a price tag on the item 1. Employee asks Manager to perform override 2. Manager performs override 3. Employee indicated manual price entry 4. Employee manually enters price 2c. Employee performs Find Item Help to obtain the valid Item ID and price 2d. Employee asks another Employee for a valid ID or price 1. Employee asks Manager to perform override 2. Manager performs override 3. Employee indicates manual price or ID entry 4. Employee manually enters price or ID
	3b.	There are multiple of the same Item category and tracking unique Items is not important 1. Employee enters Item category identifier 2. Employee enters item quantity
	3c.	Item requires manual category and price entry 1. Employee manually enters Item category 2. Employee manually enters item price
	3d.	System supplies a regular price, when a modified one is needed 1. Employee asks Manager to perform override 2. Manager performs override operation 3. Employee manually enters modified price 4. System presents new price
	5a.	Customer says they are eligible for a discount, not in the system 1. Employee verifies that the Customer is eligible 2. Employee asks their Manager to perform an override 3. Manager performs override 4. Employee enters the discount
	7a.	Customer pays by cash 1. Employee enters the amount of the given cash 1a. Customer does not have enough cash 1. Employee asks Customer to use an alternative payment method 1a. Customer tells Employee to cancel the sale 1. Employee cancels sale in the System 1b. Customer requests to use an alternative payment method 1. Employee selects the ‘change payment’ option 2. System changes the payment method

	2. System presents the balance due and opens the cash drawer 3. Employee enters cash given 4. Employee returns balance in cash to Customer 5. System records the cash payment
	7b. Customer pays by card 1. Customer enters their credit account information 2. System displays the payment information for verification 3. Employee verifies the payment 3a. Employee cancels the payment 1. System reverts to “Item entry” mode 4. System sends payment information to external banking system 4a. System detects failure to collaborate with the external banking system 1. System signals error to Employee 2. Employee asks Customer to use an alternative payment method 3. … 5. System receives payment confirmation 5a. System receives payment denial 1. System signals payment denial to the Employee 2. Employee asks customer for an alternative payment method 3. … 5b. Timeout while waiting for response 1. System signals timeout to the Employee 1a. Employee tries again 1b. Employee asks customer for an alternative payment method 1. … 6. System signals payment approval to Employee 7. System presents PIN input mechanism 8. Employee asks Customer to enter their PIN 9. Customer enters their PIN 9a. Customer enters wrong PIN 1. Customer enters PIN again 1a. Correct PIN 1b. Wrong PIN 1. Repeat until Customer’s card gets declined 2. ... 10. System records payment
	7c. Customer intended to pay by cash, but does not have enough 1. Employee asks the Customer to use an alternative payment method 1a. Customer tells Employee to cancel the sale 1. Employee cancels sale in the System
	7d. Employee cancels payment step 1. System reverts back to “Item entry” mode
	7e. Customer presents coupon(s) 1. Employee enters the coupon(s) 1a. Entered coupon(s) not valid for current purchase 1. System signals the error to the Employee 2. Employee notifies the customer

	2. System reduces the price for the coupon(s) 3. System records the coupon(s)
	12a. Invoice fails to create 1. Call the Manager 2. Report the problem to the Manager
	13a. Printer is out of paper 1. System signals the Employee about the problem 2. Employee replaces paper 3. Employee requests another receipt

6.4. Domain Model

"In the UP, the term "Domain Model" means a representation of real-situation conceptual classes, not of software objects."(Larman, C. p134).

The domain model of this system shows how the classes in the system are connected. It displays the inheritance between person and the subclasses thereof, the different items that have to be added, and the copy, sale and loan classes. This has been visualised in Figure 13 below.

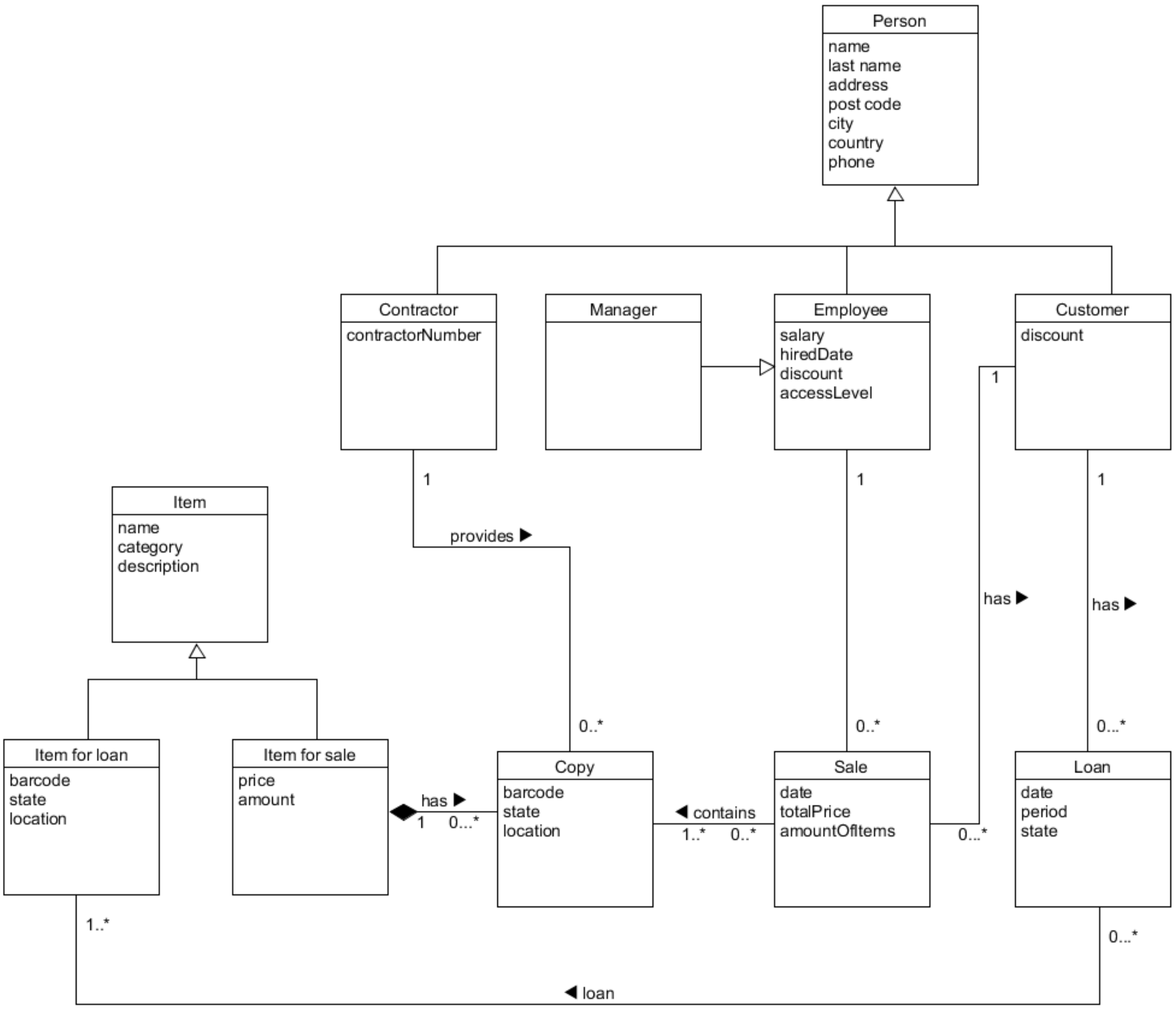


Figure 13: Domain Model

The domain model in Figure 13 shows the relation of the different classes in the system. The person class is a superclass of Contractor, Employee and Customer as they all use the same information as stated in the person class. The Manager class inherits from the employee class, so can do everything the employee can do, plus has additional manager access. This same pattern can be seen in the Items superclass, which inherits to Items for loan and Items for sale.

6.5. System Vision

The system vision includes a description about the purpose of the project and the scope of it. It covers the problem statement and a list of stakeholders and users. Finishing up with a paragraph about the technology involved and a list of features the system should have.

Purpose

The purpose of the system is to help Vestbjerg Byggecenter A/S run a more efficient business, by empowering the management and employees to work with a more comprehensive system. This includes better compatibility with a more up to date system, better stock control and overall better bookkeeping and statistics capabilities. The company has in some sense grown past the current system, which does not fulfil the requirements for a company of this scale.

Scope

In the **short** term, the system is to provide VestBjerg Byggecenter A/S with a solution for sales and stock control, as this is the most critical use case of the system.

In the **long** term, the system should provide Vestbjerg Byggecenter A/S with a comprehensive IT-solution including internet sales.

Stakeholders and users

A stakeholder of the system is anyone who has an interest in the system. This includes among others; users, who interact with the system, but also others who have an interest in seeing the company grow. stakeholders are an inseparable part to be considered while creating this new system. The users described the problems with their current system and request some features that new system needs to have, these specifications will be taken into account when the system is created. The focus of the system is to assist the managers and employees and satisfy their requirements. For a more detailed overview of the stakeholders see Chapter 2: Stakeholder Analysis

Problem statement

The problem of the current system is a lack of features and compatibility with other systems, which is a direct consequence of being outdated. This affects the stakeholders, especially the users, by decreasing efficiency and it does not provide the desired usability. To address the numerus problems with the current system, an entirely new system is to be implemented, as the current system is too outdated to modify within requirements to any meaningful extent.

Technology

A database for handling the approximately 28,000 item numbers, which can expand up to 100,000. As this database is to be used by multiple people in different locations, a central server should be implemented. It is therefore also necessary to establish a network, to connect the server and clients used by the employees.

List of Features

The current system has several features missing which are vital for the new system, this includes:

- Stock control: The current system does not include the location of an item, which can make it difficult to find it in the inventory.
- Leasing of tools: The current system does not handle the leasing of items.
- Tags and prices: The current system is not capable of printing tags and prices.
- Specifically for complete kitchens and bathrooms: every individual item in the kitchen or bathroom set should have the option to show which full set it is part of.

7. Use Case Analysis

"A system sequence diagram (SSD) is a fast and easily created artifact that illustrates input and output events related to the systems under discussion."(Larman, C. p173). This chapter will depict the system sequence diagram for the Create Sale use-case, as well as the accompanying operation contract.

This chapter shows the System Sequence diagram plus accompanying operation contracts for the use-cases mentioned in Chapter 6.3: Use Case Fully Dressed Description. Due to the sheer size of the new system, the choice was made to focus on one aspect first, the "create sale use-case". This use-case encompasses everything from business line item sale (when a customer takes a hammer from a shelf and buys it), to connecting the sale with the employee, customer, product etc. More on this in the coming paragraphs.

7.1. System Sequence Diagram

Figure 14 on the right displays the system sequence diagram for the use-case Create Sale.

The diagram depicts the actions the employee has to perform in order to get a response from the system. The sale is started by inputting the item barcode, most likely by scanning the item barcode. The system will create a collection which holds all items scanned and returns the total amount the customer has to pay. After the employee is done scanning all the items the system returns a total. The employee can now add a customer ID which applies the highest discount possible for this specific customer. The new total is returned and the system requests a payment method. In the happy day scenario the customer will pay by bank card. Once the employee has selected a payment type the banking system is initiated and takes over for the last part of this use case. Once a successful transaction request has been received from the bank the system will create an invoice and return this to the employee.

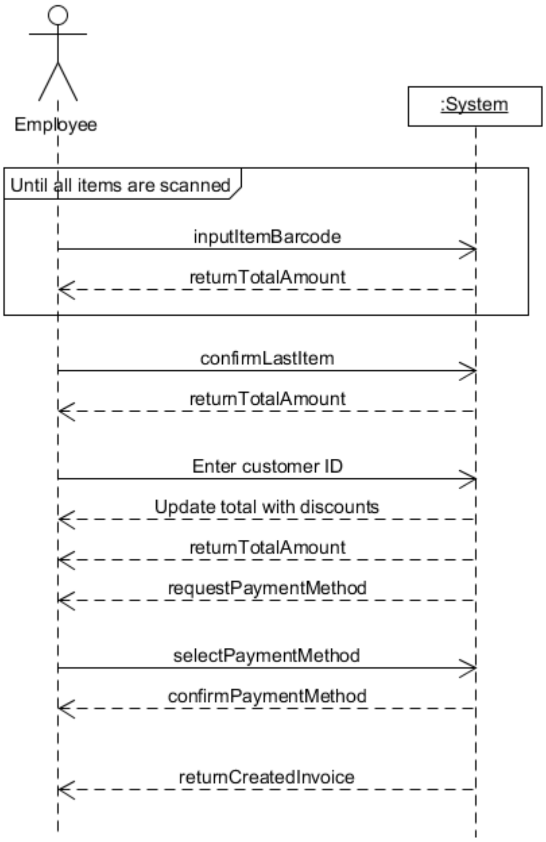


Figure 14: System Sequence Diagram of Use Case - Create Sale

Operation contracts

Operation	**createNewSale()**
Cross References	Use case: Create Sale
Preconditions	User has been authenticated
Postconditions	New Sale instance ***s*** was created (instance creation). ***s*** was associated with an Employee (association formed). Attributes of *s* were initialized.

Operation	**enterItem(itemID: ItemID, quantity: integer)**
Cross References	Use case: Create Sale
Preconditions	Sale is in progress.
Postconditions	- A SalesLineItem instance ***sli*** was created (instance creation). - ***sli*** was associated with the current Sale (association formed). - sli.quantity became quantity (attribute modification). - ***sli*** was associated with ItemDescription, based on itemID match (association formed).

Operation	**endSale()**
Cross References	Use case: Create Sale
Preconditions	Sale is in progress.
Postconditions	- Sale.complete became true (attribute modification).

Operation	**addCustomer(customerID: customerID)**
Cross References	Use case: Create Sale
Preconditions	Sale is in progress.
Postconditions	- A Customer instance ***c*** was created (instance creation). - c.ID became customerID (attribute modification). - ***c*** was associated with a Discount, based on customerID match (association formed).

Operation	**MakePayment(amount: Money)**
Cross References	Use case: Create Sale
Preconditions	Sale is in progress.
Postconditions	- A Payment instance ***p*** was created (instance creation). - p.amountTotal became amount (attribute modification). - ***p*** was associated with the current Sale (association formed). - The current Sale was associated with the Department (association formed); (to log it for statistical analysis).

8. Design

This chapter will go into the design of the system. It will detail the architecture of the system, including a 3-layer structure, will detail the interaction diagram and the design class diagram. By working this way it was possible to assure high quality standard in the programming of the system. Every aspect of the system has been written with this design in mind.

8.1. Architecture

The system’s code follows a three-layer architecture. That means that the classes of code are separated in 3 main layers, namely: the user interface, or “UI” layer, the control layer and the model layer. The UI layer holds all the code for displaying menu options and taking user input. This is also where the input is checked for errors. Next, the control layer acts as a bridge between the UI layer and model layer, it takes the user input and delivers it to the model layer. The control layer uses singleton classes for containers, this ensures that only a single container can be instantiated at one time (Barnes, D p554). In the model layer, the input is processed within the system and, if necessary, output is returned through the control layer again.

To develop this, an iterative development approach is being employed. This means that the development process involves planning, developing, programming and testing one major function of the program at a time. This way the features of the system get added iteratively and thus making it easier to expand in the future.

8.2. Interaction Diagram

An interaction diagram can be displayed in two separate ways; the communication diagram or sequence diagram. In this report the sequence diagram has been used as this diagram has the advantage of being able to display the interaction in greater detail (Larman pp 222-223).

This sequence diagram in Figure 15 on the next page illustrates the process of adding a SaleItem to a Sale. This is done by entering a barcode, which is passed from the SaleUI to the SaleCtr and on to the Inventory class. The Inventory searches through the items in order to find an item with a matching barcode. When found the SaleItem is returned to the SaleCtr, which checks it and passes it to the SaleContainer. The SaleContainer searches for the correct Sale and passes the item to the Sale, which inserts it into the array of items connected to that sale.

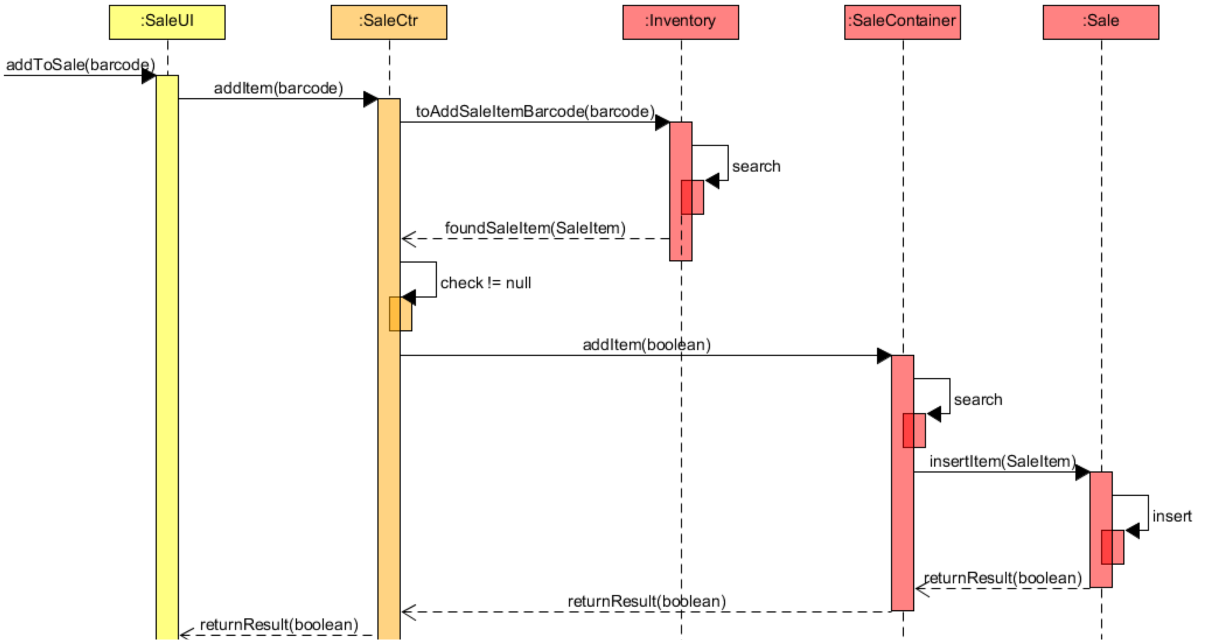


Figure 15: Sequence Diagram - addItem

8.3. Design Class diagram

This Design Class Diagram visualises the architecture, classes, attributes and methods of the system. As shown in Figure 16 on the next page the system is created with a 3-layer structure, as this offers a better understanding and clearer purpose of the different parts of the system. Additionally, it eases the process of adding or modifying features I the system, as the coupling should be lower and offer higher cohesion. This also helps the future process of changing the user interface to graphical rather than text based. Note that the getters and setters are not shows in the diagrams as they will obscure the purpose of the diagram.

Figure 16 is a small version of the full design class diagram. For a full and readable version please refer to .

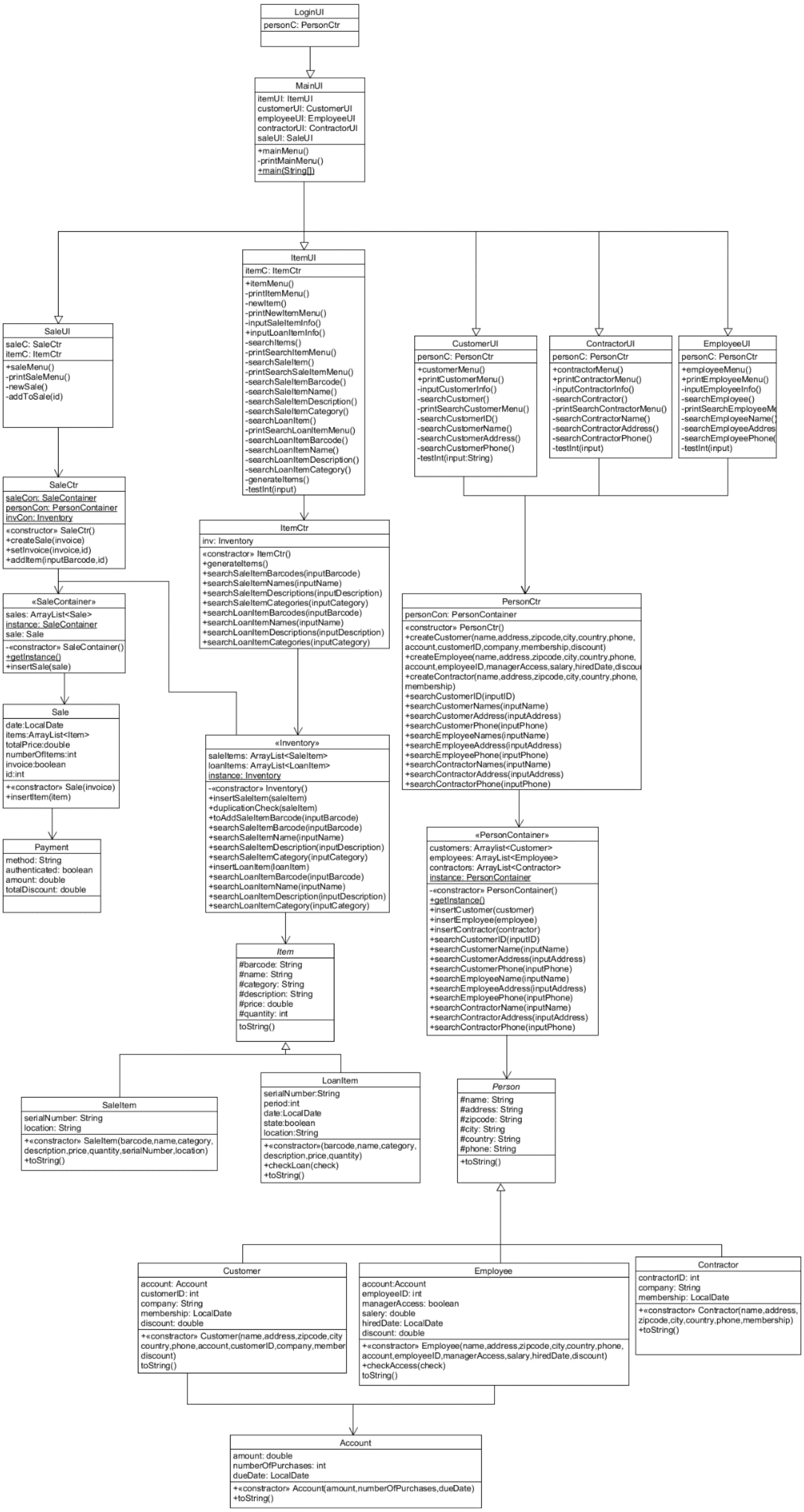

LoginUI
personC: PersonCtr
MainUI
itemUI: ItemUI
customerUI: CustomerUI
employeeUI: EmployeeUI
contractorUI: ContractorUI
saleUI: SaleUI
+mainMenu()
-printMainMenu()
+main(String[])
SaleUI
saleC: SaleCtr
itemC: ItemCtr
+saleMenu()
-printSaleMenu()
-newSale()
-addToSale(id)
ItemUI
itemC: ItemCtr
+itemMenu()
-printItemMenu()
-newItem()
-printNewItemMenu()
-inputSaleItemInfo()
+inputLoanItemInfo()
-searchItems()
-printSearchItemMenu()
-searchSaleItem()
-printSearchSaleItemMenu()
-searchSaleItemBarcode()
-searchSaleItemName()
-searchSaleItemDescription()
-searchSaleItemCategory()
-searchLoanItem()
-printSearchLoanItemMenu()
-searchLoanItemBarcode()
-searchLoanItemName()
-searchLoanItemDescription()
-searchLoanItemCategory()
-generateItems()
-testInt(input)
CustomerUI
personC: PersonCtr
+customerMenu()
+printCustomerMenu()
-inputCustomerInfo()
-searchCustomer()
-printSearchCustomerMenu()
-searchCustomerID()
-searchCustomerName()
-searchCustomerAddress()
-searchCustomerPhone()
-testInt(input:String)
ContractorUI
personC: PersonCtr
+contractorMenu()
+printContractorMenu()
-inputContractorInfo()
-searchContractor()
-printSearchContractorMenu()
-searchContractorName()
-searchContractorAddress()
-searchContractorPhone()
-testInt(input)
EmployeeUI
personC: PersonCtr
+employeeMenu()
+printEmployeeMenu()
-inputEmployeeInfo()
-searchEmployee()
-printSearchEmployeeMe
-searchEmployeeName()
-searchEmployeeAddres
-searchEmployeePhone(
-testInt(input)
SaleCtr
saleCon: SaleContainer
personCon: PersonContainer
invCon: Inventory
«constructor» SaleCtr()
+createSale(invoice)
+setInvoice(invoice,id)
+addItem(inputBarcode,id)
ItemCtr
inv: Inventory
«constractor» ItemCtr()
+generateItems()
+searchSaleItemBarcodes(inputBarcode)
+searchSaleItemNames(inputName)
+searchSaleItemDescriptions(inputDescription)
+searchSaleItemCategories(inputCategory)
+searchLoanItemBarcodes(inputBarcode)
+searchLoanItemNames(inputName)
+searchLoanItemDescriptions(inputDescription)
+searchLoanItemCategories(inputCategory)
PersonCtr
personCon: PersonContainer
«constructor» PersonCtr()
+createCustomer(name,address,zipcode,city,country,phone,
account,customerID,company,membership,discount)
+createEmployee(name,address,zipcode,city,country,phone,
account,employeeID,managerAccess,salary,hiredDate,discou
+createContractor(name,address,zipcode,city,country,phone,
membership)
+searchCustomerID(inputID)
+searchCustomerNames(inputName)
+searchCustomerAddress(inputAddress)
+searchCustomerPhone(inputPhone)
+searchEmployeeNames(inputName)
+searchEmployeeAddress(inputAddress)
+searchEmployeePhone(inputPhone)
+searchContractorNames(inputName)
+searchContractorAddress(inputAddress)
+searchContractorPhone(inputPhone)
«SaleContainer»
sales: ArrayList<Sale>
instance: SaleContainer
sale: Sale
-«constractor» SaleContainer()
+getInstance()
+insertSale(sale)
Sale
date:LocalDate
items:ArrayList<Item>
totalPrice:double
numberOfItems:int
invoice:boolean
id:int
+«constractor» Sale(invoice)
+insertItem(item)
Payment
method: String
authenticated: boolean
amount: double
totalDiscount: double
«Inventory»
saleItems: ArrayList<SaleItem>
loanItems: ArrayList<LoanItem>
instance: Inventory
-«constractor» Inventory()
+insertSaleItem(saleItem)
+duplicationCheck(saleItem)
+toAddSaleItemBarcode(inputBarcode)
+searchSaleItemBarcode(inputBarcode)
+searchSaleItemName(inputName)
+searchSaleItemDescription(inputDescription)
+searchSaleItemCategory(inputCategory)
+insertLoanItem(loanItem)
+searchLoanItemBarcode(inputBarcode)
+searchLoanItemName(inputName)
+searchLoanItemDescription(inputDescription)
+searchLoanItemCategory(inputCategory)
«PersonContainer»
customers: Arraylist<Customer>
employees: ArrayList<Employee>
contractors: ArrayList<Contractor>
instance: PersonContainer
-«constractor» PersonContainer()
+getInstance()
+insertCustomer(customer)
+insertEmployee(employee)
+insertContractor(contractor)
+searchCustomerID(inputID)
+searchCustomerName(inputName)
+searchCustomerAddress(inputAddress)
+searchCustomerPhone(inputPhone)
+searchEmployeeName(inputName)
+searchEmployeeAddress(inputAddress)
+searchEmployeePhone(inputPhone)
+searchContractorName(inputName)
+searchContractorAddress(inputAddress)
+searchContractorPhone(inputPhone)
Item
#barcode: String
#name: String
#category: String
#description: String
#price: double
#quantity: int
toString()
SaleItem
serialNumber: String
location: String
+«constractor» SaleItem(barcode,name,category,
description,price,quantity,serialNumber,location)
+toString()
LoanItem
serialNumber:String
period:int
date:LocalDate
state:boolean
location:String
+«constractor»(barcode,name,category,
description,price,quantity)
+checkLoan(check)
+toString()
Person
#name: String
#address: String
#zipcode: String
#city: String
#country: String
#phone: String
+toString()
Customer
account: Account
customerID: int
company: String
membership: LocalDate
discount: double
+«constractor» Customer(name,address,zipcode,city
country,phone,account,customerID,company,member
discount)
toString()
Employee
account:Account
employeeID: int
managerAccess: boolean
salery: double
hiredDate: LocalDate
discount: double
+«constractor» Employee(name,address,zipcode,city,country,phone,
account,employeeID,managerAccess,salary,hiredDate,discount)
+checkAccess(check)
toString()
Contractor
contractorID: int
company: String
membership: LocalDate
+«constractor» Contractor(name,address,
zipcode,city,country,phone,membership)
+toString()
Account
amount: double
numberOfPurchases: int
dueDate: LocalDate
+«constractor» Account(amount,numberOfPurchases,dueDate)
+toString()

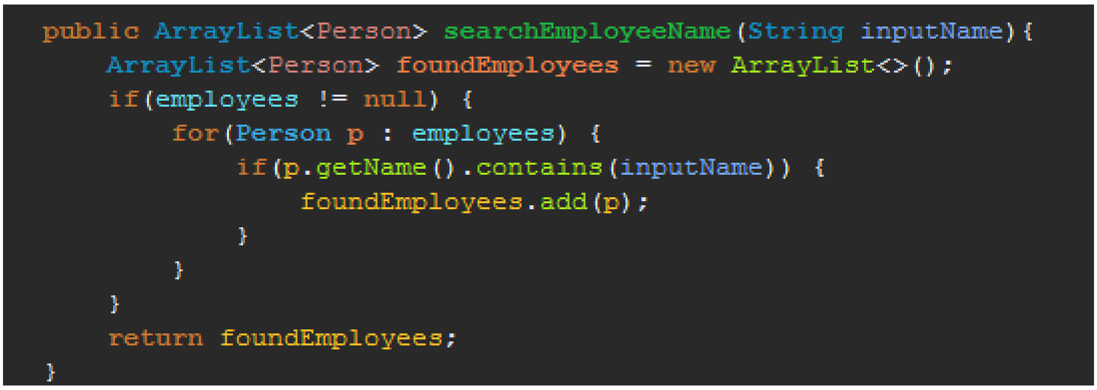
Figure 16: Design Class Diagram

9. Implementation

The implementation details the standards that were used while coding the system, which include which naming conventions were used, how the program is organised and the finer details of coding standards, such as line breaks and spaces. Lastly this chapter will discuss the unit testing that was used in order to ensure the system works as intended.

9.1. Code Standards

Choosing the design components of the system or the program is very essential because of different aspects. Therefore, one of the inseparable and most important parts of the design is choosing the right code standard resulting in code being easily readable or manageable by other people not only the original author. The code follows the general Java code convention (Sun Microsystem 1997). Some of the code standards are shown and described in Figure 17 below.



```
public ArrayList<Person> searchEmployeeName(String inputName){
    ArrayList<Person> foundEmployees = new ArrayList<>();
    if(employees != null) {
        for(Person p : employees) {
            if(p.getName().contains(inputName)) {
                foundEmployees.add(p);
            }
        }
    }
    return foundEmployees;
}
```

Figure 17: Code Standards

Initialization

Local variables were initialized mainly according to Java conventions, where they are declared (i.e. foundEmployees initializes new ArrayList of type Person).

Naming

According to Java code standard, descriptive names of classes, variables and methods were used in the created code. In the attached figure above, there is a method called ‘searchEmployeeName’ that searches only through all the employees’ names. All the classes in the code start with a capital letter and their names are singular nouns (i.e. class ‘Employee’ or ‘Sale’). All the methods and variable names start with lowercase letters but together with class, capital letters were used to start each word after the first one to increase the readability (i.e. variable ‘inputName’ or method ‘searchEmployeeName(…)’).

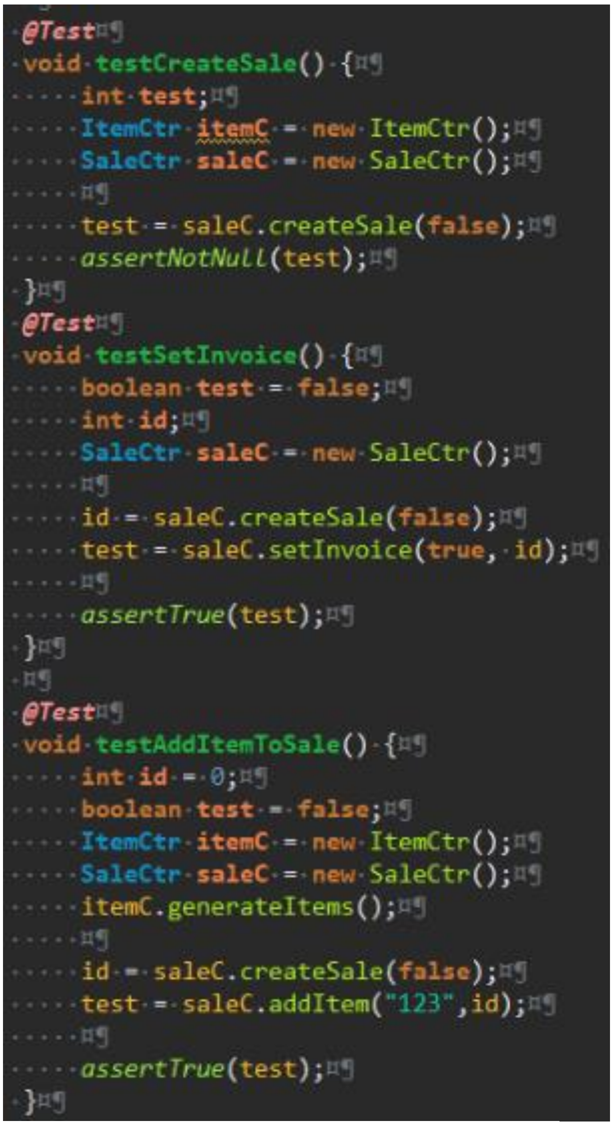
Layout

The code was constructed to be read very clearly, therefore, one level of indentation is four spaces, and all the statements within a block are indented one level. All the blocks or statements open with the braces at the end of the line that contains the keyword or expression defining the block. For the closing brace is then used separate line and the brace is aligned under the keyword that defines the block (i.e. if-statement or for-each loop). As well as spaces around the operators were used, blank lines were used between all methods and sometimes also inside of the methods to get simple and quick overview of what is happening inside of the block.

9.2. Unit Testing

Testing in Java is done with JUnit tests. These tests act as trials for certain parts of the program so as to ensure that it functions like it should. Throughout the course of programming, the code is changed a great deal. Because of this, JUnit tests will produce different results and as such, acts like a notification indicating the code not behaving correctly. JUnit tests are simply a way to automate the manual testing procedures.

Figure 18 below shows snippets from some test methods. These methods ensure that the main functions of our code produce the results that we want them to so that we may make changes freely and be certain that the functions remain the same.



```
 @Test
 void testCreateSale() {
     int test;
     ItemCtr itemC = new ItemCtr();
     SaleCtr saleC = new SaleCtr();
     
     test = saleC.createSale(false);
     assertNotNull(test);
 }
 @Test
 void testSetInvoice() {
     boolean test = false;
     int id;
     SaleCtr saleC = new SaleCtr();
     
     id = saleC.createSale(false);
     test = saleC.setInvoice(true, id);
     
     assertTrue(test);
 }
 
 @Test
 void testAddItemToSale() {
     int id = 0;
     boolean test = false;
     ItemCtr itemC = new ItemCtr();
     SaleCtr saleC = new SaleCtr();
     itemC.generateItems();
     
     id = saleC.createSale(false);
     test = saleC.addItem("123",id);
     
     assertTrue(test);
 }
```

Figure 18: JUnit Test

This page is intentionally left blank.

PART III

Group Evaluation

This page is intentionally left blank.

10. Group Process

This chapter will reflect on the group process throughout the project. It will cover a group contract that all participants agreed upon. A group evaluation which gives each member the opportunity to evaluate the group process. Finally there is a time evaluation which shows how the group worked and if deadlines were met.

10.1. Group Contract

At the start of the group collaboration a group contract was set up. Because this contract is in a4 size it has been added as an appendix at the end of the report. By signing the title page on the report all group members also agree with the group contract. See Appendix C for the contract.

10.2. Group Evaluation

This section focuses on the evaluation of the group process by reflecting on five different sections being: Team Work, Communication, Punctuality, Motivation and Final Result of the project. This was done in order to improve future project collaboration and efficiency. Table 8 below does not reflect an individual person's participation in the group, instead it focuses on this particular group member's view on said topic. Each member assigns zero to five points to every section and at the end average is counted.

Table 8: Group Evaluation

Name	Team Work	Communication	Punctuality	Motivation	Final Result
Linda	5★	4★	4★	5★	4★
Simon	5★	4★	4★	5★	4★
Juliana	5★	3★	4★	5★	4★
Димитьр	5★	4★	4★	5★	4.5★
		Average:	87.5★/100★		

10.3. Time Evaluation

During the entire project the original timetable in Table 1 was continuously updated and marked whenever a new section of the report or program was finished. The original timetable colours were set to green and are in some cases still visible in the background, and the actual path taken has been coloured in orange. See Table 9 on the next page for a complete overview of the group progress during this project. As is visible, the group has worked very effectively, even finishing early on some parts of the project.

Table 9: Timetable - Evaluation

Tasks	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Mon**	**Tue**	**Wed**
Project plan	🟧	🟧											
Part I – Business	🟧	🟧	🟧										
1. Organisational structure	🟧	🟧											
Organisational type	🟧												
Mission/ vision		🟧											
Evaluation and problems		🟧											
Organisational culture		🟧											
2. Stakeholder analysis	🟧	🟧											
Strategic goals		🟧											
3. SWOT analysis		🟧	🟧										
4. Business case		🟧	🟧										
Part II – System development			🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧
5. Preliminary study			🟧	🟧									
Activity diagram			🟧										
Employee-task-goal table				🟧									
Mock-up				🟩	🟧								
6. Requirements				🟧	🟩								
Use-case diagram				🟧									
Use-case descriptions				🟧									
Domain model				🟧	🟩								
System vision				🟧	🟩								
7. Use-case analysis					🟧								
SSD and OC					🟧								
8. Design					🟧	🟧	🟧	🟧					
Architecture					🟧	🟩							
Interaction diagram					🟧	🟧	🟧	🟩					🟧
Design class diagram							🟧	🟩					🟧
9. Implementation							🟧	🟧	🟧	🟧	🟧	🟧	🟧
Code implementation								🟧	🟧	🟧	🟧	🟧	🟧
Code standard									🟧				🟧
Unit testing											🟩	🟧	🟧
Part III – Group evaluation												🟧	🟧

This page is intentionally left blank.

This page is intentionally left blank.

Part IV

Appendixes

This page is intentionally left blank.

Appendix A. Table of References

Books

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

Bloisi, W., Cook, W. and Hunsaker, L. (2006). *Management and Organisational Behaviour*. 2nd ed. McGraw-Hill Higher Education, pp 86-87.

Boddy, D. and Buchanen, D. (1994). *Take the Lead, Interpersonal skills for project managers*. Prentice Hall, pp 55-60.

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

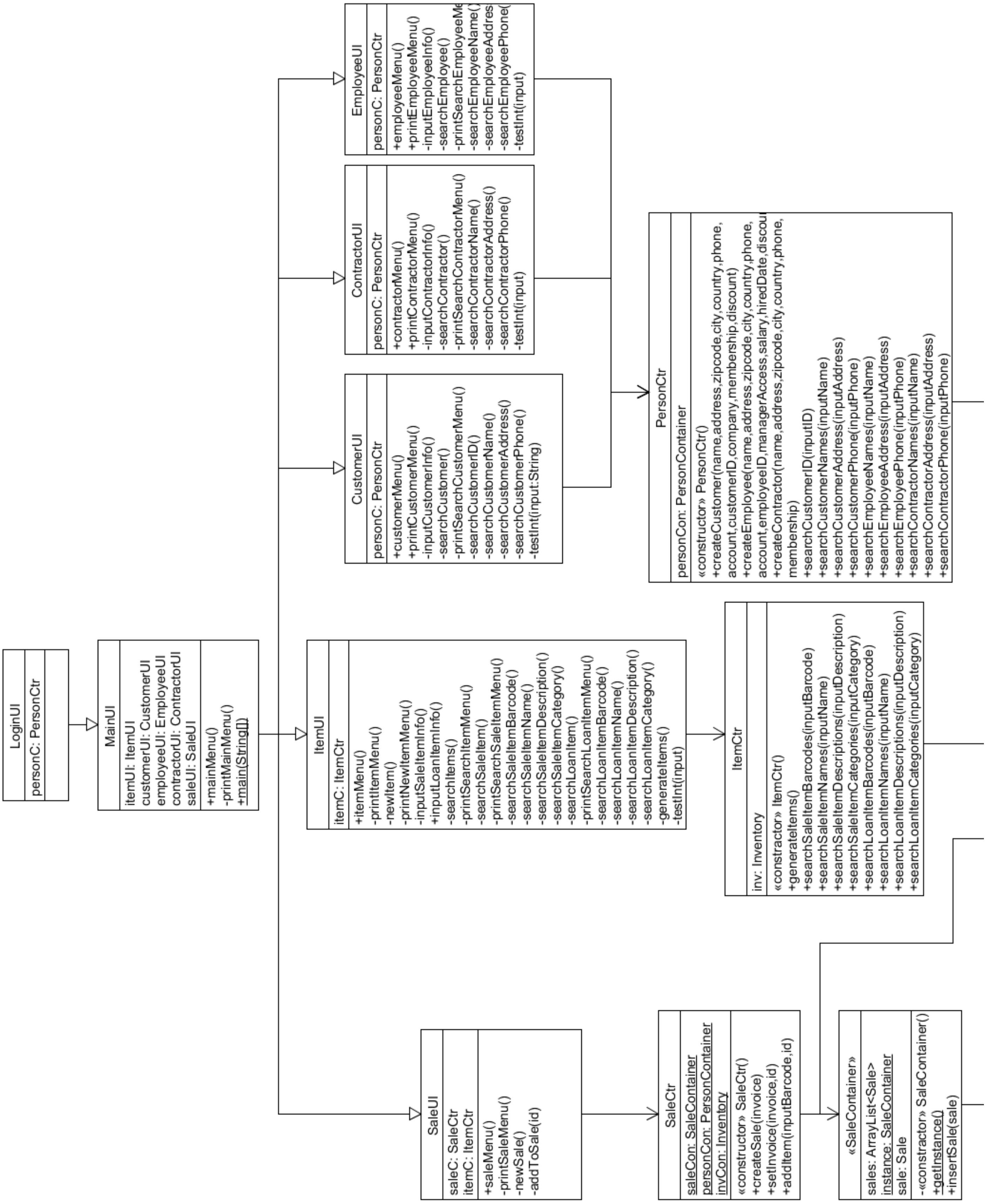
Barnes, D. and Kölling, M. (2017). *Objects First with Java, A Practical Introduction Using BlueJ*. 6th ed. Harlow: Pearson Education Limited.

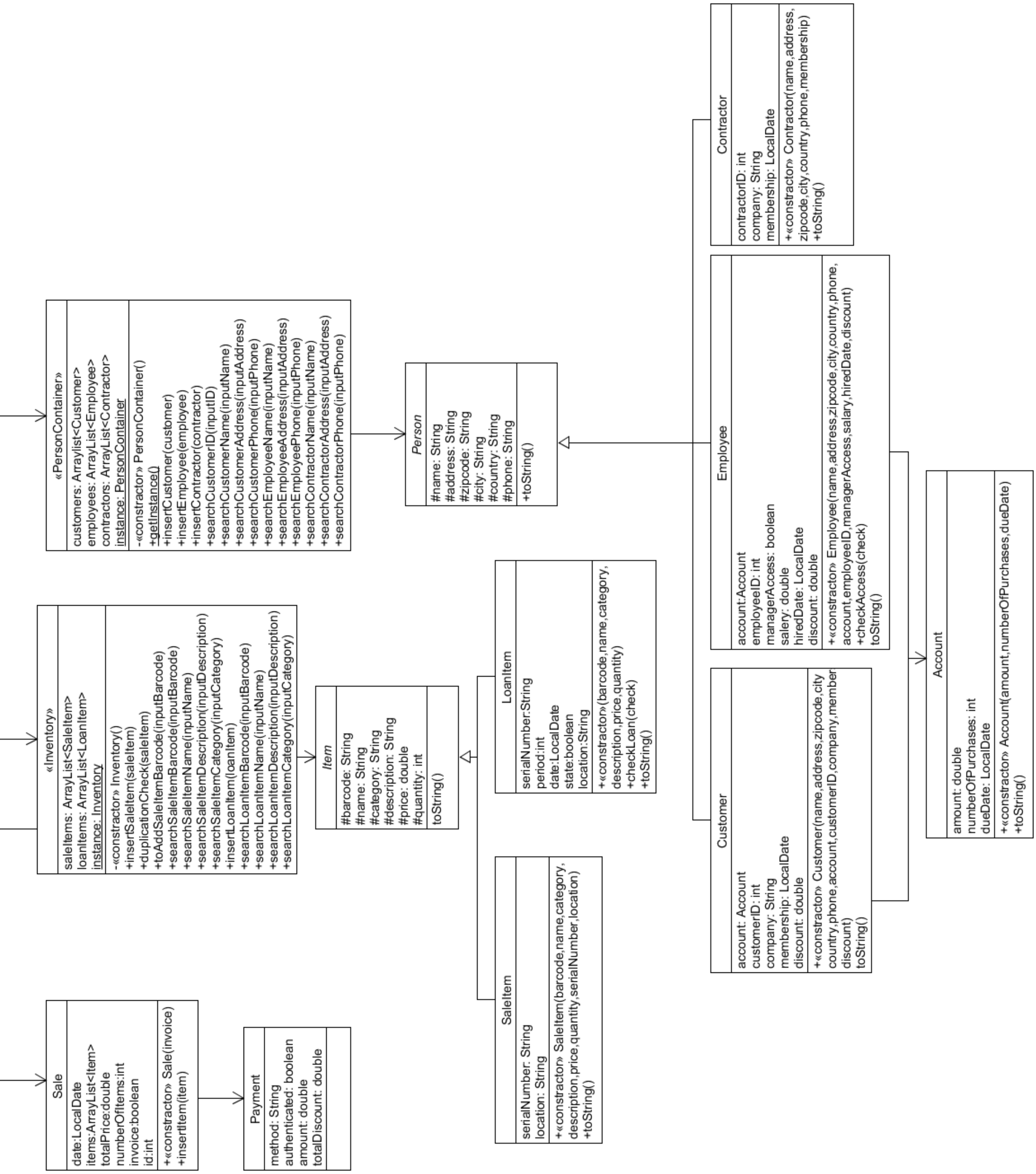
(1997). *Java Code Conventions*.[pdf] Mountain View, CA: Sun Microsystems. Available at http://www.oracle.com/ [accessed 13 12. 2017].

Tables

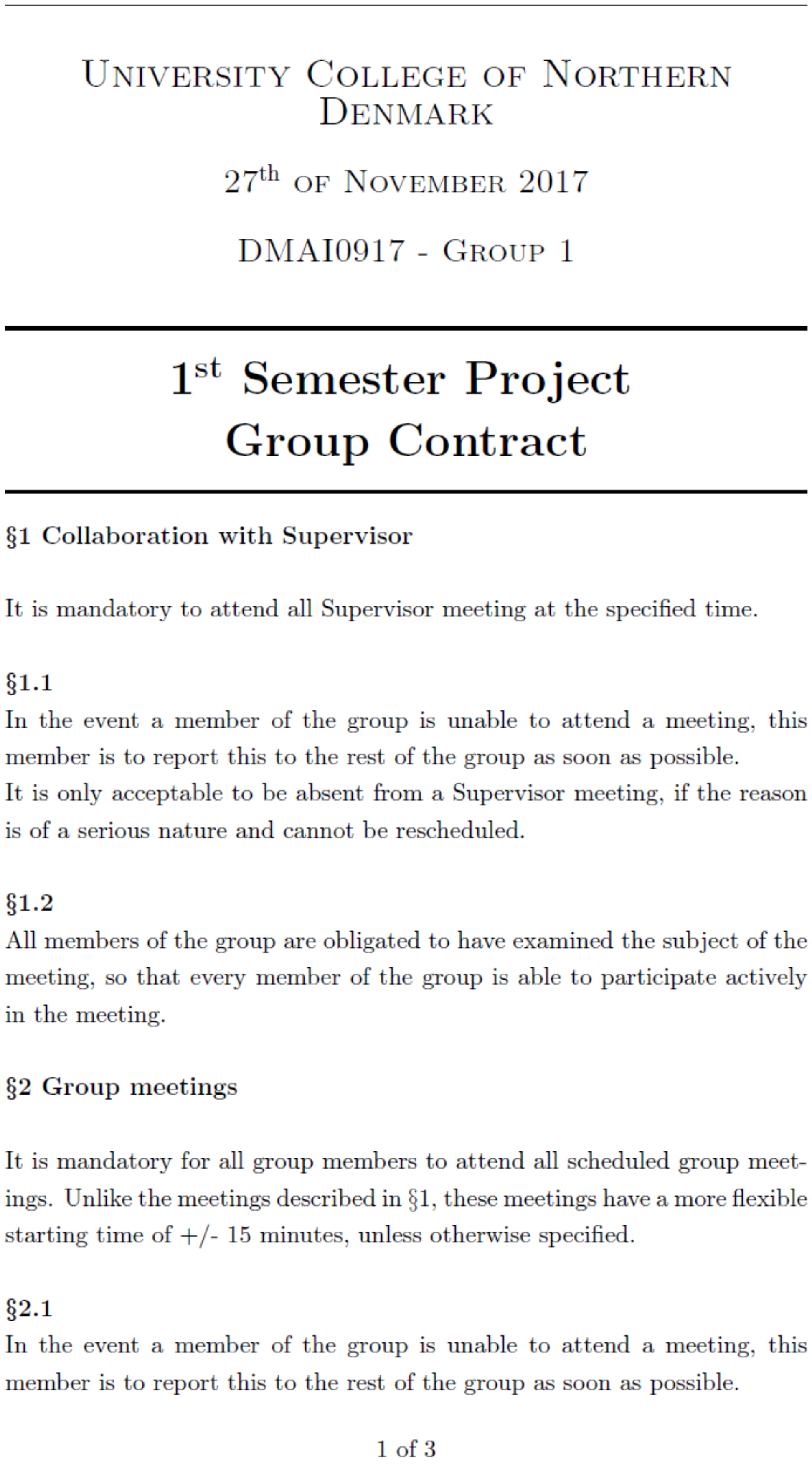
Figures

Appendix B. Design Class Diagram




«PersonContainer»
customers: Arraylist<Customer>
employees: ArrayList<Employee>
contractors: ArrayList<Contractor>
instance: PersonContainer
-«constractor» PersonContainer()
+getInstance()
+insertCustomer(customer)
+insertEmployee(employee)
+insertContractor(contractor)
+searchCustomerID(inputID)
+searchCustomerName(inputName)
+searchCustomerAddress(inputAddress)
+searchCustomerPhone(inputPhone)
+searchEmployeeName(inputName)
+searchEmployeeAddress(inputAddress)
+searchEmployeePhone(inputPhone)
+searchContractorName(inputName)
+searchContractorAddress(inputAddress)
+searchContractorPhone(inputPhone)
Person
#name: String
#address: String
#zipcode: String
#city: String
#country: String
#phone: String
+toString()
Contractor
contractorID: int
company: String
membership: LocalDate
+«constractor» Contractor(name,address,
zipcode,city,country,phone,membership)
+toString()
Employee
account:Account
employeeID: int
managerAccess: boolean
salery: double
hiredDate: LocalDate
discount: double
+«constractor» Employee(name,address,zipcode,city,country,phone,
account,employeeID,managerAccess,salary,hiredDate,discount)
+checkAccess(check)
toString()
Customer
account: Account
customerID: int
company: String
membership: LocalDate
discount: double
+«constractor» Customer(name,address,zipcode,city
country,phone,account,customerID,company,member
discount)
toString()
Account
amount: double
numberOfPurchases: int
dueDate: LocalDate
+«constractor» Account(amount,numberOfPurchases,dueDate)
+toString()
«Inventory»
saleItems: ArrayList<SaleItem>
loanItems: ArrayList<LoanItem>
instance: Inventory
-«constractor» Inventory()
+insertSaleItem(saleItem)
+duplicationCheck(saleItem)
+toAddSaleItemBarcode(inputBarcode)
+searchSaleItemBarcode(inputBarcode)
+searchSaleItemName(inputName)
+searchSaleItemDescription(inputDescription)
+searchSaleItemCategory(inputCategory)
+insertLoanItem(loanItem)
+searchLoanItemBarcode(inputBarcode)
+searchLoanItemName(inputName)
+searchLoanItemDescription(inputDescription)
+searchLoanItemCategory(inputCategory)
Item
#barcode: String
#name: String
#category: String
#description: String
#price: double
#quantity: int
toString()
LoanItem
serialNumber:String
period:int
date:LocalDate
state:boolean
location:String
+«constractor»(barcode,name,category,
description,price,quantity)
+checkLoan(check)
+toString()
SaleItem
serialNumber: String
location: String
+«constractor» SaleItem(barcode,name,category,
description,price,quantity,serialNumber,location)
+toString()
Sale
date:LocalDate
items:ArrayList<Item>
totalPrice:double
numberOfItems:int
invoice:boolean
id:int
+«constractor» Sale(invoice)
+insertItem(item)
Payment
method: String
authenticated: boolean
amount: double
totalDiscount: double

Appendix C. Group Contract



University College of Northern Denmark

27th of November 2017

DMAI0917 - Group 1

1st Semester Project Group Contract

§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.

1 of 3

§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 group chat should not be used for anything other than communicating with group members.

§4 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.

§5 Other

§5.1

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

§5.2

In case a group member does not follow the group contract, the offending member owes the rest of the group cake.

§5.3

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.