



Vrije Universiteit Brussel

Faculty Science

User Interface Design

Requirements, Analysis, Modelling,
Design and Evaluation Report

Rik Vanmechelen

Roeland Matthijssens

Thierry Janssens

year 2011-2012



Contents

1 Abstract	4
2 People using the system	5
2.1 Shop-clerk	5
2.2 Shop-manager	5
2.3 Stock-manager	5
2.4 Super-admin	5
2.5 Clients	5
2.6 Suppliers	6
3 User classes	7
3.1 Shop-clerk	7
3.2 Shop-manager	7
3.3 Stock-manager	8
3.4 Clients	8
3.5 Suppliers	8
4 Functional requirements	10
4.1 General	10
4.2 Sales module	10
4.3 Stock module	10
4.4 Account module	11
4.5 Overview module	11
5 Associated usability requirements	12
5.1 General	12
5.2 Stock module	12
5.3 Sales module	14
5.4 Account module	16
5.5 Overview Module	18
6 User Tasks	21
6.1 The first step: Logging in	21
6.2 Stock manager: Task Scenarios	21
6.2.1 Get items in stock from a certain supplier	21
6.2.2 Create new stock item	22
6.2.3 Process delivery	25
6.3 Store-clerk: Task Scenarios	25
6.3.1 Process a "normal" sale, including mixed payment	26
6.3.2 Add money to user account, Starting from the Sales module	26
6.3.3 Create account	29
6.3.4 Edit account details	30

6.4	Shop Manager: Task Scenarios	30
6.4.1	Get overview of all products sold this month	31
7	User Object Modeling	34
7.1	Stock Manager	34
7.1.1	Stock	34
7.1.2	Orders	34
7.1.3	Deliveries	34
7.2	Shop Clerk	36
7.2.1	Sales	36
7.2.2	History	36
7.2.3	Account	36
7.3	Shop Manager	36
8	Style Guide	39
8.1	Standards for switching modules	39
8.2	Standard Window layout	39
8.3	Sales window	39
8.4	Appearance	39
8.5	Navigation	41
8.6	Standards for color	41
9	Design Report	42
9.1	General tasks	42
9.1.1	Enabled task Set	42
9.1.2	Login	42
9.2	Stock Module	42
9.2.1	Enabled task Set	42
9.2.2	List All	43
9.2.3	Item	43
9.2.4	New Delivery	44
9.2.5	New Item	44
9.2.6	Orders	44
9.3	Sales Module	44
9.3.1	Enabled task Set	44
9.4	account Module	45
9.4.1	Enabled task Set	45
9.5	Overview Module	46
9.5.1	Enabled task Set	46
10	Prototype	47
10.1	Login	47
10.2	Stock	47
10.3	Sales	47
10.4	Overview	47

1 Abstract

This document contains all the written deliverables for the User Interface Design assignment. The document is structured after the assignment. It will cover the first seven deliverables. The purpose of the assignment is to go through all the phases that are involved in creating a good User Interface. This is done by designing a User Interface for a store management tool.

The tool consists of four modules as described in the assignment. The sales module has to facilitate the user in selling products to customers. The products are managed in the stock module in which the user can manage several attributes of the stock. Customers can have internal accounts. This means there has to be a account system that is integrated into the sales module so customers can pay with their personal account. Lastly there is the overview module in which the user can get several statistics about products, sales, etc...

This document will first describe the requirements. That chapter will describe the user classes and the Usability requirements. The second chapter will elaborate on the user tasks, the user object models and the style guide. The last chapter will talk about the actual design and show the final prototype.

2 People using the system

This section will give an overview of all the user that will come in contact with the system, either direct or indirect.

2.1 Shop-clerk

He is responsible for the actual sales in store. He helps the clients by checking their products and accepting their payments. He also helps the clients with creating accounts and depositing money on set accounts.

2.2 Shop-manager

He has a supervisory role, this includes managing the finances and being able to check the stock for it's integrity. Managing finances means that he is the responsible for paying the bills, to suppliers as well as the shop personal. He is the one that is responsible for setting and updating the buy and sale-price. The shop manager has access to an overview of all the sales.

2.3 Stock-manager

He manages the in- and output of stock. He accepts the deliveries and ensures the correctness of the stock.

2.4 Super-admin

He will be the top level admin of the system. He has the ability to delete accounts and update the system. In a small environment he will be responsible for the training of the users of the system.

2.5 Clients

Clients do not typically interact directly with the system. They are usually served by the shop-clerk. Their actions will include buying products, creating an account and depositing/retrieving money on set account.

2.6 Suppliers

Suppliers do not interact with the system directly. They will deliver their goods and the stock-manager will process the delivery in the system.

3 User classes

3.1 Shop-clerk

- Type of user: Direct
- motivation/goals: See section 2.1.
- Application Familiarity: Advanced beginner.
- Frequency of use: Daily.
- Task knowledge: High.
- Use: Mandetory.
- Computer experience: Low.
- Number of users: 5.
- Training: Half a day.
- Tasks performed: See later.

3.2 Shop-manager

- Type of user: Direct
- motivation/goals: See section2.2
- Application Familiarity: Competent performer
- Frequency of use: Occasionally
- Task knowledge: High
- Use: Mandatory
- Computer experience: Medium, PC in general use
- Number of users: 1 to 2
- Training: a day
- Tasks performed: See later.

3.3 Stock-manager

- Type of user: Direct
- motivation/goals: See section 2.3.
- Application Familiarity: Advanced beginner.
- Frequency of use: Occasionally.
- Task knowledge: High.
- Use: Mandatory.
- Computer experience: Low.
- Number of users: 1 to 2
- Training: Half a day.
- Tasks performed: See later.

3.4 Clients

- Type of user: Indirect
- motivation/goals: see section 2.5
- Application Familiarity: N/A
- Frequency of use: Occasionally
- Task knowledge: N/A
- Use: voluntary
- Computer experience: N/A
- Number of users: 1000
- Training: N/A
- Tasks performed: See later

3.5 Suppliers

- Type of user: Indirect
- motivation/goals: see section 2.5
- Application Familiarity: N/A
- Frequency of use: Occasionally

- Task knowledge: N/A
- Use: voluntary
- Computer experience: N/A
- Number of users: 5
- Training: N/A
- Tasks performed: See later

4 Functional requirements

Here we will describe the functional requirements per Module

4.1 General

- Searching the data on the current page
- Logging of user actions

4.2 Sales module

- Selling an item using the barcode scanner
- Selling an item by typing in the product name
- Selling an item by typing in the barcode
- Process cash payment
- Process payment by user account
- Process payment using a mix of the previous two
- Adding and subtracting money from a user account

4.3 Stock module

- Creating a new stock item
- Changing a stock item information
- Bulk stock update
- Process a delivery of a supplier
- Create a new order for the supplier
- Get list of the previous orders
- Get a list of all the products
- Filter the list of all the products

4.4 Account module

- Create a new account
- Change the details of an account
- Add and subtract money from an account
- Show a buy history of the account
- Get a list of all the accounts
- Filter the list of all the accounts
- Sort the list of accountso (filtered or not)

4.5 Overview module

- Show statistics for users
- Show statistics for clients
- Show statistics for items
- Filter ranges for statistics
- Sort list of statistics
- Show graphs for any list

5 Associated usability requirements

5.1 General

[1]. Searching is intuitive and easy

Motivation Searching is used so often to speed up the process of finding what the user is looking for. Therefor the search field should be accesible on every page. It should not be hidden in a submenu. But is located directly on the main page of every module. This will speed up the search process, and it will be used more often when it is visible at all times.

User class Stock-manager, Shop-manager, Store-clerk

Measuring concept User Friendliness

Measuring method Task scenario

Judging criteria Best case: $\leq 10\text{s}$: It takes the user less then ten seconds to search for an item, and receive the result.
Longer waiting times are unacceptable.

5.2 Stock module

[2]. Creating a new stock item should not allow errors

Motivation Creating a new stock item is essential to the store. This should not allow corrupted or incomplete data, since this could impact the whole sales process. (i.e. When a item with a wrong price is sold it creates problems for accounting)

User class Stock-manager

Measuring method Task scenario

Judging criteria Less than 1/1000 stock items are allowed to contain errors

Prevention technique Faulty data prevention

[3]. Updating a stock item should be easy and safe

Motivation Stock items will be updated constantly. It should be easy and fast to do, but still make sure there are no errors in the data.

User class Stock-manager

Measuring concept User Friendliness

Measuring method Users will be (automatically) monitored. We will measure the time it takes to successfully update a stock item and keep track of the amount of errors.

Judging criteria Best case: < 20s : It takes the user on average less than 20 seconds to update an item.

Worst case would be more than 40 seconds.

[4]. Stock data should have useful filtering

Motivation It is nice to be able to get an overview of several products. It should be easy to filter them and get the desired result.

User class Stock-manager

Measuring concept User Friendliness

Measuring method Users will be (automatically) monitored. We will measure the amount of times every filter option is used and in which combinations they are used.

Judging criteria Best case: 80% of the time a user uses the list view, he also uses at least one filter.

Worst case is when it is less than 30% of the time.

[5]. Allow for easy 'bulk' updating

Motivation By using the earlier explained filters and sorting, it should be easy to make minor bulk updates such as changing the amount in stock.

User class Stock-manager

Measuring concept User Friendliness

Measuring method Users will be (automatically) monitored. We will measure the time it takes a user to update the stock, given some criteria.

Judging criteria Best case: < 40s : The user performs the update in less than 40 seconds.

Worst case is if it takes more than 2 minutes to get the task done.

[6]. Easy processing of deliveries

Motivation It is essential that the user can easily process a new delivery without making errors.

User class Stock-manager

Measuring concept User Friendliness

Measuring method Users will be (automatically) monitored. We will measure the time it takes a user to process a delivery. We will also keep track of the amount of errors the user makes. (A mock delivery consists of a list of 10 products, 8 with and 2 without bar-codes, and the amounts)

Judging criteria Best case: The user doesn't make any errors and processes the delivery in less than 1.5 minutes.

Worst case would be when it takes longer than 5 minutes or the user makes several errors.

5.3 Sales module

The sales list contains the items that one customer is going to buy.

[7]. allow for easy adding of sales items to the sales list

Motivation Adding a sales item is the most basic action of the system and should be possible using the simplest interactions with the system. This includes multiple ways to recognize the items (barcode scanning, list selection or simply typing the barcode or name of the product if the scanner won't read the barcode).

User class Store-clerk

Measuring concept Quality of task performance

Measuring method Time needed to add an item

Judging criteria The system will record the amount of time it takes to add an item to the sales list.

Best case: < 2s

Acceptable level: < 6s

Unacceptable level: > 6s

[8]. Allow for the easy removal of an item in the sales list

Motivation In case you added an unwanted item, you should be able to remove that item with only a couple of steps. These steps include: selecting the item, selecting the quantity to remove and approving the removal.

User class Store-clerk

Measuring concept Quality of task performance

Measuring method Time needed to remove an item

Judging criteria The system will record the amount of time it takes to remove an item from the sales list.

Best case: < 4s

Acceptable level: < 7s
Unacceptable level: > 7s

[9]. Easy choice of payment

Motivation When all items are scanned the user can choose between two ways of payment(Cash or Internal Account).

User class Store-clerk

Measuring concept Quality of task performance

Measuring method Actions needed to perform the task

Judging criteria The system will record the amount of actions it takes to choose the payment type

Best case: 1 action

Acceptable level: 2 actions

Unacceptable level: > 2 actions

[10]. Easy using mixed payment methods

Motivation Give the customer the possibility to pay using different methods. This should only require the amount of steps equal to the sum of the steps of each method used plus one step for each additional method.

User class Store-clerk

Measuring concept Quality of task performance

Measuring method Actions needed to perform the task

Judging criteria The system will record the amount of actions it takes to perform the payment

Best case: 2 action

Acceptable level: 3 actions

Unacceptable level: > 3 actions

[11]. Easy deposit money on user account

Motivation In case the customer would have the need to consult his account (to add money, remove money , ...), this should be fast and easy.

User class Store-clerk

Measuring concept Quality of task performance

Measuring method Time needed to deposit money

Judging criteria The system will record the amount of time it takes to deposit money on an account.

Best case: < 30s

Acceptable level: < 50s

Unacceptable level: > 50s

5.4 Account module

[12]. Creating a new account should be easy

Motivation Create the account of a new customer is an important part of the business.
It should be easy and fast.

User class Shop-manager

Measuring concept User Friendliness

Measuring method Users will be (automatically) monitored. We will measure the time it takes to successfully update a stock item and keep track of the amount of errors.

Judging criteria Best case: < 30s : It takes the user on average less than 30 seconds to create an account.

Worst case would be more than 60 seconds.

[13]. Creating a new account should not allow errors

Motivation Erroneous information shouldn't be filled in the form used for the registration.
Faulty data in the account will have an impact on other parts of the system.

User class Shop-manager

Measuring method Task scenario

Judging criteria less than 1/1000 accounts are allowed to contain errors

Prevention technique Faulty data prevention

[14]. Adding money to an account during sales should be easy and fast

Motivation Adding money is possible by simply filling in a number and confirm the deposit.

User class Shop-manager

Measuring concept Quality of task performance

Measuring method Time needed to add money during sales

Judging criteria The system will record the amount of time it takes to deposit money on an account.

Best case: < 30s

Acceptable level: < 50s

Unacceptable level: > 50s

[15]. Removing money from an account should be easy and fast

Motivation Removing money is possible by simple filling in a number and confirm the withdrawal.

User class Shop-manager

Measuring concept Quality of task performance

Measuring method Time needed to extract money

Judging criteria The system will record the amount of time it takes to extract money on an account.

Best case: < 30s

Acceptable level: < 50s

Unacceptable level: > 50s

[16]. Finding a user should be fast

Motivation A user account should be easily accessible.

User class Shop-manager

Measuring concept Quality of task performance

Measuring method Actions needed to perform the task

Judging criteria The system will record the amount of actions it takes to find the account

Best case: 2 action

Acceptable level: 5 actions

Unacceptable level: > 5 actions

[17]. Updating an account should be painless and easy

Motivation Updating the account should only require editing some values and confirming those changes.

User class Shop-manager

Measuring concept Quality of task performance

Measuring method Time needed to edit an account successfully.

Judging criteria The system will record the amount of time it takes to update the account.

Best case: < 30s

Acceptable level: < 50s

Unacceptable level: > 50s

[18]. Updating an account should not allow faulty data

Motivation When updating the account, it should not be possible to fill in faulty data.

When the user enters incorrect/inconsistent information the system should provide a visual queue to indicate this. This will minimize the amount of incorrect submissions, and the amount of frustration for the user.

User class Shop-manager

Measuring concept Quality of task performance

Measuring method Amount of errors received while updating an account.

Judging criteria The system will record the amount errors it takes to update the account.

Best case: = 0

Acceptable level: = < 1

Unacceptable level: > 1

5.5 Overview Module

[19]. Have an overview of all products and their sales numbers

Motivation Get a list of all the products (ordered alphabetically by default) and their respective sales numbers.

User class Shop-manager

Measuring concept User Friendliness

Measuring method Monitor the amount of actions it takes to find the overview page.

Judging criteria The system will record the amount of actions it takes to get to the overview page..

Best case: < 5

Acceptable level: < 10

Unacceptable level: > 10

[20]. Filter the previous mention overview using a range in time

Motivation Filter the sales number of the previous functionality with a limiting time factor (for example taking into account only the last week).

User class Shop-manager

Measuring concept User Friendliness

Measuring method Users will be (automatically) monitored. We will measure the amount of times every filter option is used and in which combinations they are used.

Judging criteria Best case: 80% of the time a user uses the list view, he also uses at least one filter.

Worst case is when it is less than 30% of the time.

[21]. Get a graphical overview of the sales of one product over time

Motivation Get a graphical (for example a statistic) of the sales of a product (where one axis can be time or any other factor you deem important).

User class Shop-manager

Measuring concept Quality of task performance

Measuring method Users will be (automatically) monitored. We will measure the amount of time it takes for the user to find the statistics of a specified item

Judging criteria The system will record the amount of time it takes to get the asked information of an item within a time interval.

Best case: < 20s

Acceptable level: < 30s

Unacceptable level: > 30s

[22]. Combine multiple products in the graph of the previous item

Motivation Combine the graphs of multiple products to get a clear comparison between them.

User class Shop-manager

Measuring concept User Friendliness

Measuring method Users will be (automatically) monitored. We will measure the amount of time it takes to compare different items sale numbers with each other

Judging criteria The system will record the amount of time it takes to get the asked information of a set of items.

Best case: < 20s/item

Acceptable level: < 30s/item

Unacceptable level: > 30s/item

[23]. Get an overview of sales numbers of one Employee

Motivation Get the sales number of an employee.

User class Shop-manager

Measuring concept User Friendliness

Measuring method Actions needed to perform the task

Judging criteria The system will record the amount of actions it takes to find the employee number

Best case: 3 action

Acceptable level: 5 actions

Unacceptable level: > 5 actions

[24]. filter sales numbers of one employee using time range

Motivation Get the sales numbers of an employee within a specific range of time.

User class Shop-manager

Measuring concept User Friendliness

Measuring method Users will be (automatically) monitored. We will measure the amount of times the filter option is used

Judging criteria Best case: 80% of the time a user uses the filter.

Worst case is when it is less than 30% of the time.

[25]. Sort based on different criteria

Motivation Sort the overview of all products based on different criteria (name, price, ...).

User class Shop-manager

Measuring concept User Friendliness

Measuring method Users will be (automatically) monitored. We will measure the amount of times the sort function is used.

Judging criteria Best case: 80% of the time a user uses the sorted list view, .

Worst case is when it is less than 30% of the time.

6 User Tasks

In this section, we give an overview of the specific tasks the users of the system will be able to perform. The tasks are modelled in a graphical way, by means of the Concurrent Task Trees (or CTT's). Since CTT's tend to become big, they are split up into smaller parts. Most CTT's will be accompanied by a small task Scenario to give a better view of what is going on.

6.1 The first step: Logging in

When a store employee first opens the application, he is asked to log in. This way it is easy to monitor the employees when they are using the system. A user has two ways to log in into the system. The first way is by using his name and password, the second way is by just scanning his employee card. When the user is logged in he can perform a set of different tasks, these will be explained in the coming sections. From anywhere in the application the user is able to quit/log out of the system. All this is shown in figure 6.1

6.2 Stock manager: Task Scenarios

First we will give the complete CTT (figure 6.2 of the Stock manager tasks. Then we will give some scenarios (and the corresponding CTT) to give a better understanding of the tasks.

The stock module allows the user to create and manage stock items as well as orders and deliveries. The stock manager can easily process a delivery by scanning all the products in the delivery.

From almost anywhere in the stock module the user can go back to the list of all products, the deliveries and create new products. This gives the user a good overview of where he is, and where he can go to.

From within a product it is possible to edit the attributes of the product, or order more with the supplier. See figure 6.3.

6.2.1 Get items in stock from a certain supplier

See figure 6.4

Type - Typical

Situation - The stock manager wants to know which items are almost out of stock from a given supplier

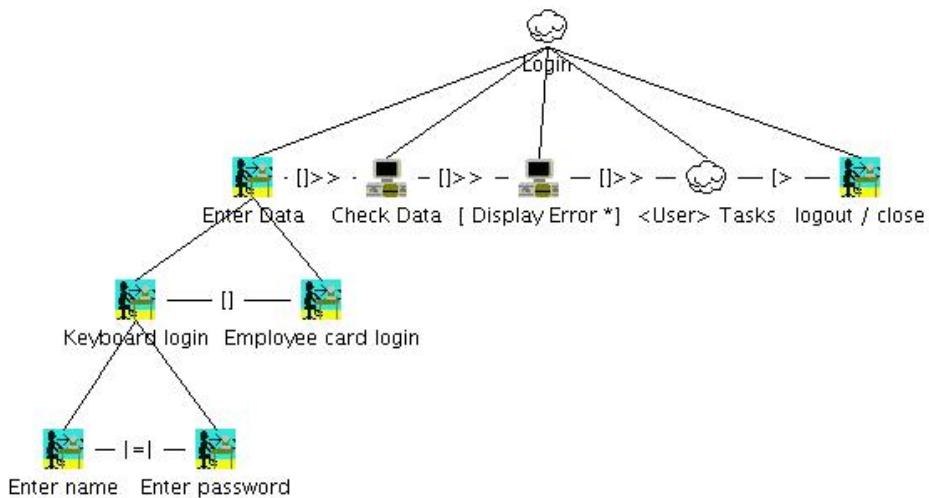


Figure 6.1: CTT to login

Script

Stock M.: Go to the stock module.

System: Show stock module.

Stock M.: Select "list of all products".

System: Gives a list of all the products.

Stock M.: Select "filter".

System: Give the filter options

Stock M.: Add filters (amount in stock < 10 and supplier is Colruyt).

System: Give the filtered list.

6.2.2 Create new stock item

See figure 6.5

Type - Typical

Situation - A new kind of product was delivered and it must be added to the system

Script

Stock M.: Go to the stock module.

System: Show stock module.

Stock M.: Select "Create product".

System: Show the selected form.

Stock M.: Fill all the needed data in for this product.

Stock M.: Confirm addition.

System: Save the new type of item in the database.

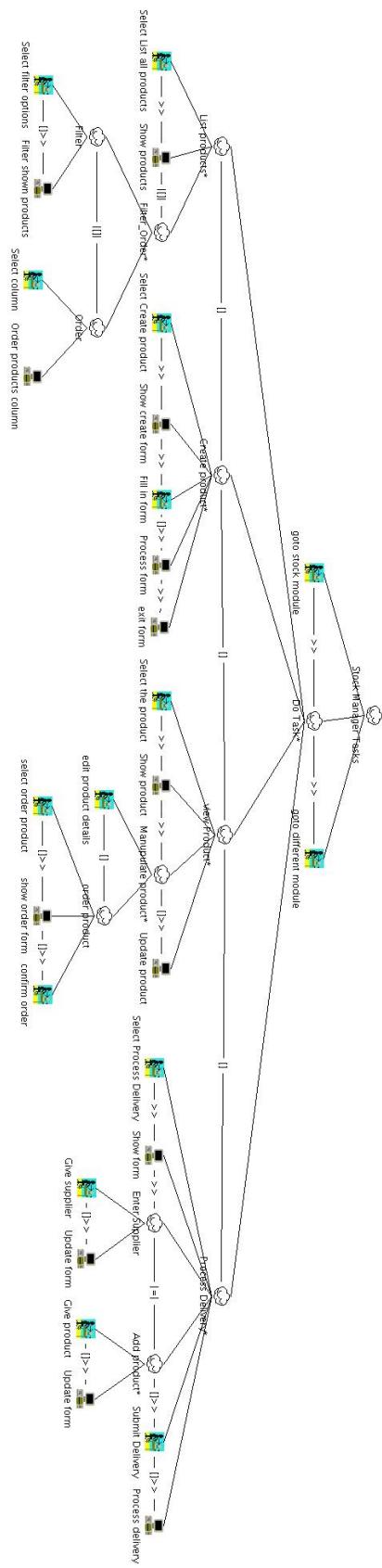


Figure 6.2: Complete stock-manager CTT

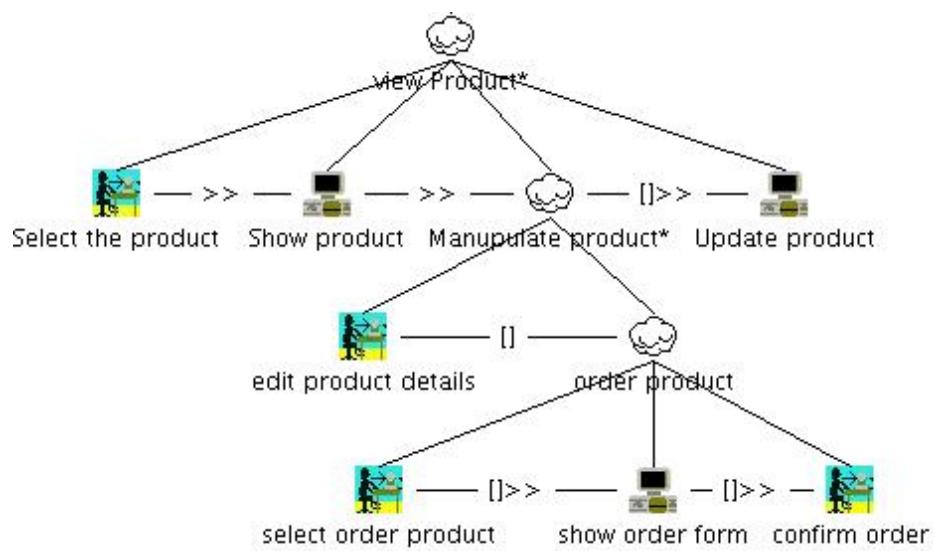


Figure 6.3: CTT: tasks for manipulating a single product

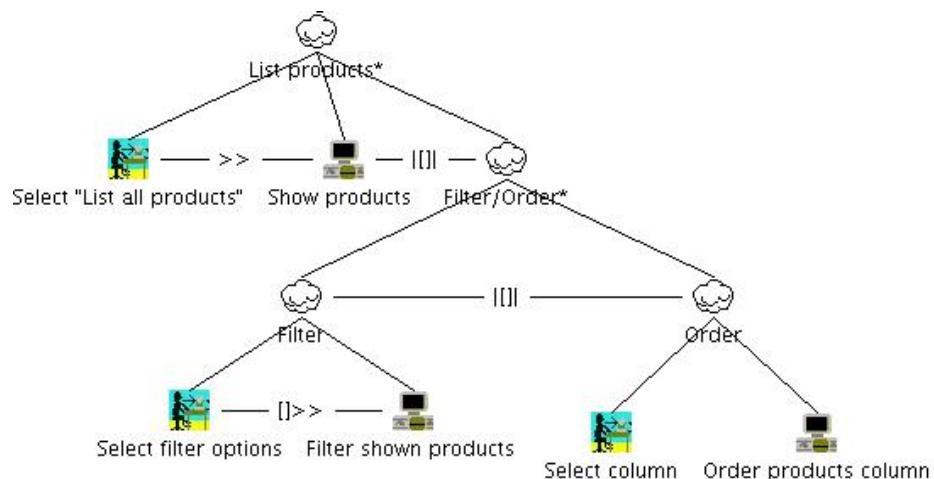


Figure 6.4: CTT to show a list of stock items

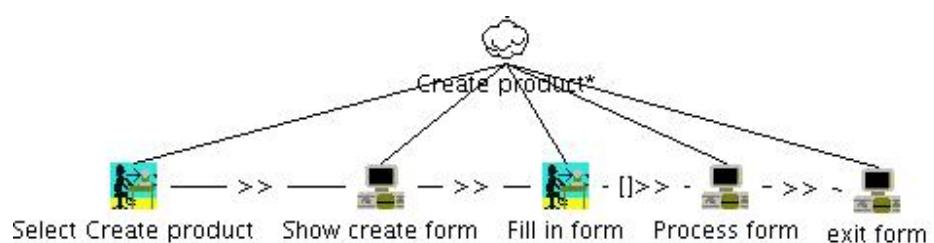


Figure 6.5: CTT to create new product

6.2.3 Process delivery

See figure 6.6

Type - Typical

Situation - A supplier delivered some goods and the stock manager has to enter them into the system.

Script

Stock M.: Go to the stock module.

System: Show stock module.

Stock M.: Select "Process Delivery".

System: Show the selected form.

Stock M.: Enter the Supplier

System: Update the form with supplier info.

Stock M.: Give a product (by scanning it using a barcode scanner) and other nessesaray info such as the amount that has been delivered.

System: Update the form with the product info.

Stock M.: Submit the delivery

System: The delivery will be processed and the items will be added to the stock.

6.3 Store-clerk: Task Scenarios

In figure 6.7 you can see the complete CTT of the store-clerk. Figure 6.9 and 6.8 show the accounts part and the sales part again. The following scenarios will include smaller parts of these CTT's.

The store-clerk will spend most of his time in the sales module. This tool allows him to sell products to the customers as well as depositing money to user accounts. This second part is also possible from within the account module. It is available in the sales module to accommodate a seamless integration of both modules.

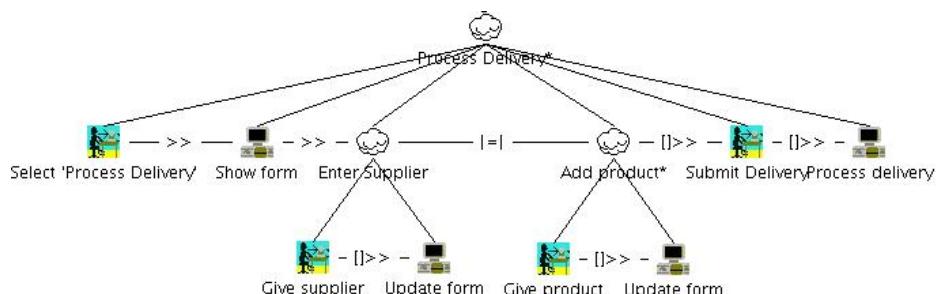


Figure 6.6: CTT to process a delivery

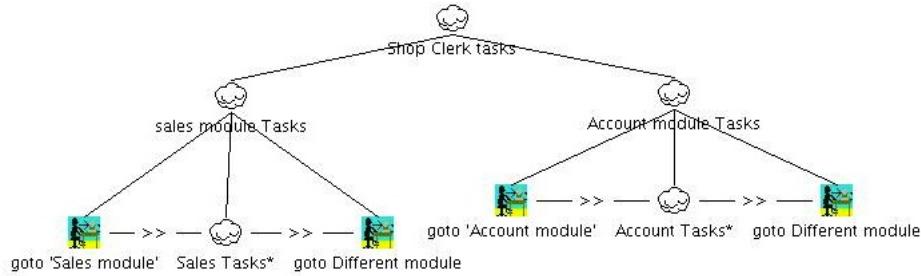


Figure 6.7: Top level Store-clerk CTT

6.3.1 Process a "normal" sale, including mixed payment

See figure 6.10

Type - Typical

Situation - A customer comes to the shop clerk to buy different products, and the shop clerk uses the scanner to process those in the system. The custumor wants to pay using both cash, and his internal account.

Script

Customer: Gives a product to the shop clerk.

Clerk: Takes product and scans it with the scanner

System: System searches for the product and add its to the sales list

Clerk: Asks for next product, otherwise asks the customer for the way he wants to pay.

Customer: Customer wants to use mixed payment, he wants to pay 50% cash and 50% with his account and gives his card

Clerk: Scans the card, the same way you would scan a product

System: Searches for the clients account

System: Shows the account of the customer

Clerk: Clerk first clicks the cash button, receives the amount of the customer

System: Tells the clerk how much he has to return to the customer

Clerk: Returns the money to the customer and tells the system he received the money

System: Updates the amount that still has to be payed (processes the payment)

Clerk: Clicks the pay with account button

Clerk: Confirms the payment

System: Processes the payment

Clerk: Gives proof of payment to customer.

6.3.2 Add money to user account, Starting from the Sales module

See figure 6.11

Type - Typical

Situation - The customers wants to add money to his account, but the clerk is still in the sales module.

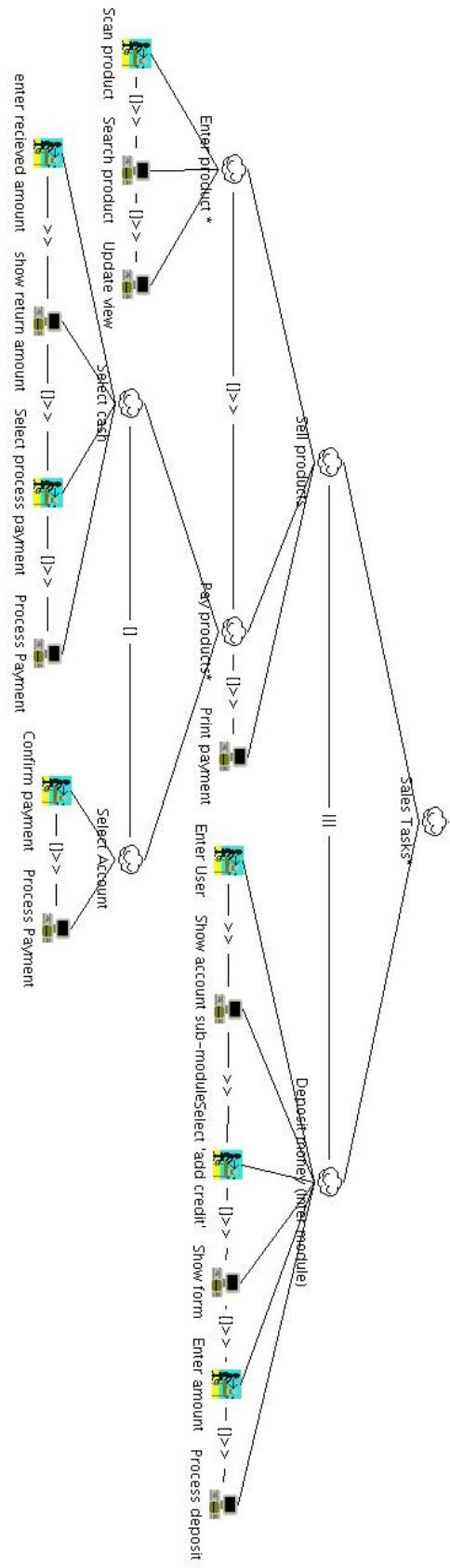


Figure 6.8: CTT to performs sales

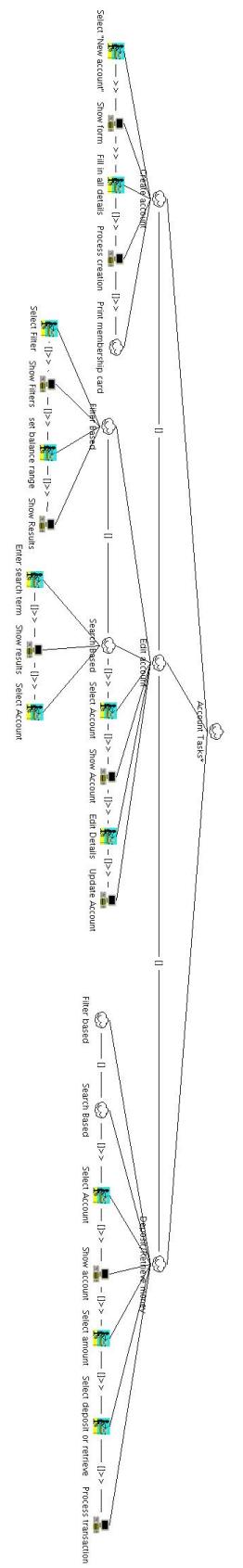


Figure 6.9: CTT to manipulate accounts

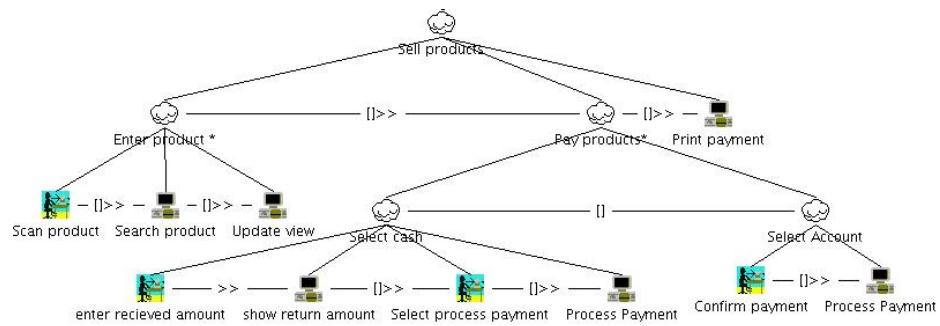


Figure 6.10: CTT for selling products

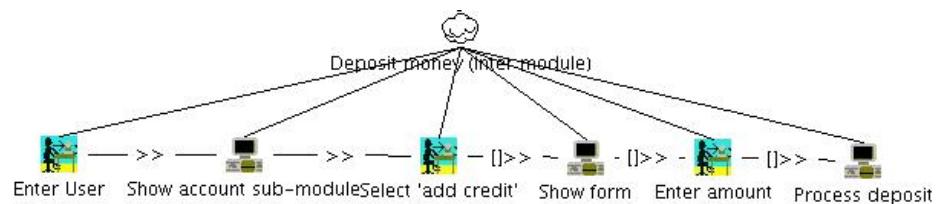


Figure 6.11: CTT for depositing money

Script

Clerk: Enters the user name (or scans the card)

System: shows the user

Clerk: clicks "add credit"

System: Show a form to enter an amount

Clerk: Enters the amount that has to be added to the account

System: Processes the deposit

6.3.3 Create account

See figure 6.12

Type - Typical

Situation - A customer wants to create an account in your shop.

Script

Clerk: Asks all needed information from the customer.

Clerk: Goes to the account module.

System: Show the account module.

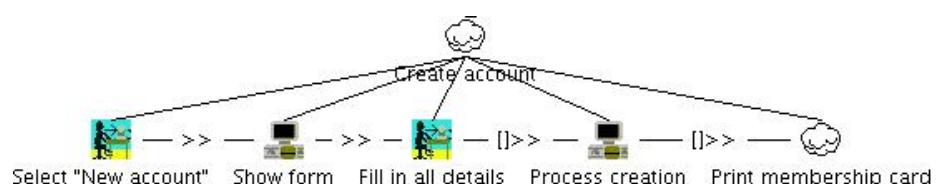


Figure 6.12: CTT for creating a new account

Clerk: Select "create new account".

System: Show subscription form.

Clerk: Fill in name.

Clerk: Fill in surname.

Clerk: Fill in gender.

Clerk: Fill in address.

Clerk: Fill in mail and/or telephone number.

Clerk: Confirm subscription.

System: If form filled in correctly print out membership card, otherwise give error.

Clerk: Give membership card to Customer.

6.3.4 Edit account details

See figure 6.13

Type - Typical

Situation - A customer wants to change the telephone number of his account.

Script

Clerk: Start the account module.

System: Show the account module.

Clerk: Searches for the customer by entering his name.

System: Give list of results.

Clerk: Selects the correct account.

System: Show the account.

Clerk: Change the telephone number.

Clerk: Saves the changes.

System: Saves changes in the database.

6.4 Shop Manager: Task Scenarios

The shop manager can for most parts do what the Store-clerk can do. He has one extra module that is aimed to fulfil his managing needs. The overview module give the manager the ability to get a overview of what is going on in the store. The module has graphs to show product sales over time, as well as which employee sells the most, and which clients are the most valuable (those who buy the most products).

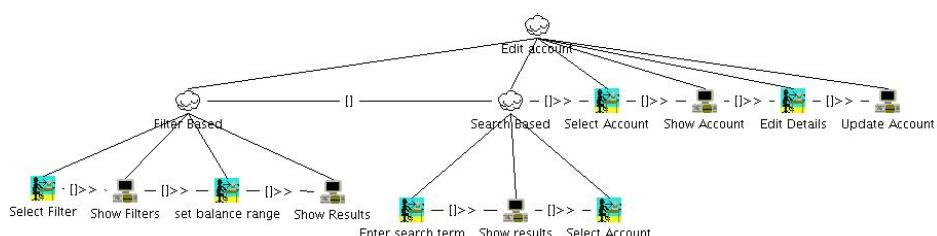


Figure 6.13: CTT for editing an account

Figure 6.14 shows the complete overview of the for the store manager (minus the parts he has in common with the store-clerk).

Only one scenario will be given to show how to use the overview, since all the different overviews are basically identical. The CTT's for the other overviews are given in figure 6.16 and figure 6.17 for completeness.

6.4.1 Get overview of all products sold this month

See figure 6.15

Type - Typical

Situation - The shop manager wants to have a list of all the products that have been sold during the last month.

Script

Shop M.: Start the overview module.

System: Show the overview module.

Shop M.: Select "all products".

System: Show all products of the shop.

Shop M.: Select "filter".

System: Show filter settings.

Shop M.: Add first filter on amount sold.

Shop M.: Add second filter on time range of last month.

System: show products sold in last month.

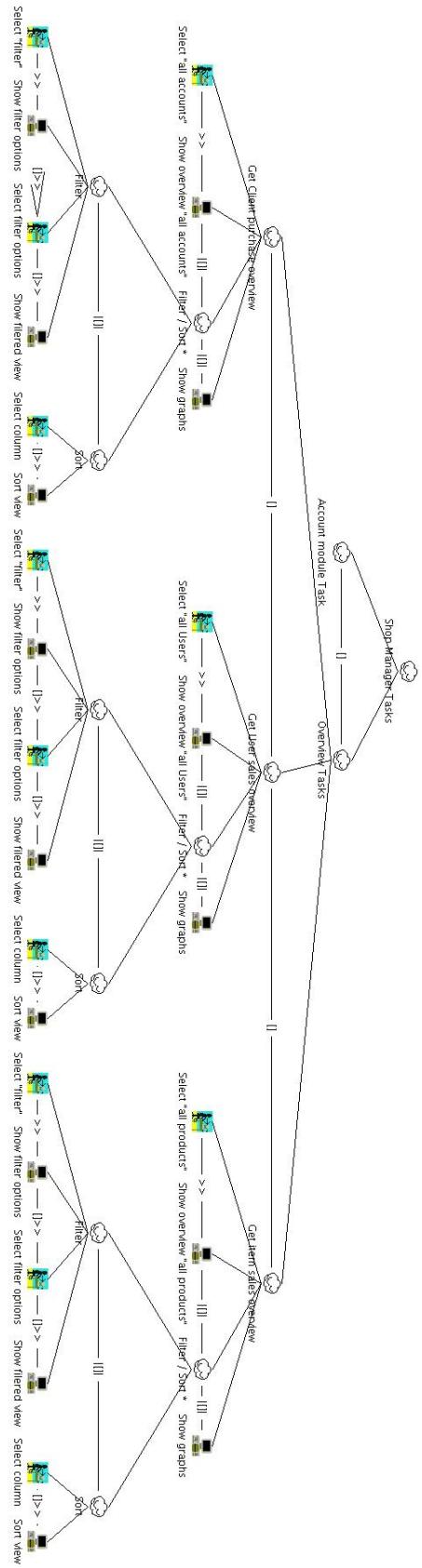


Figure 6.14: CTT for the manager tasks

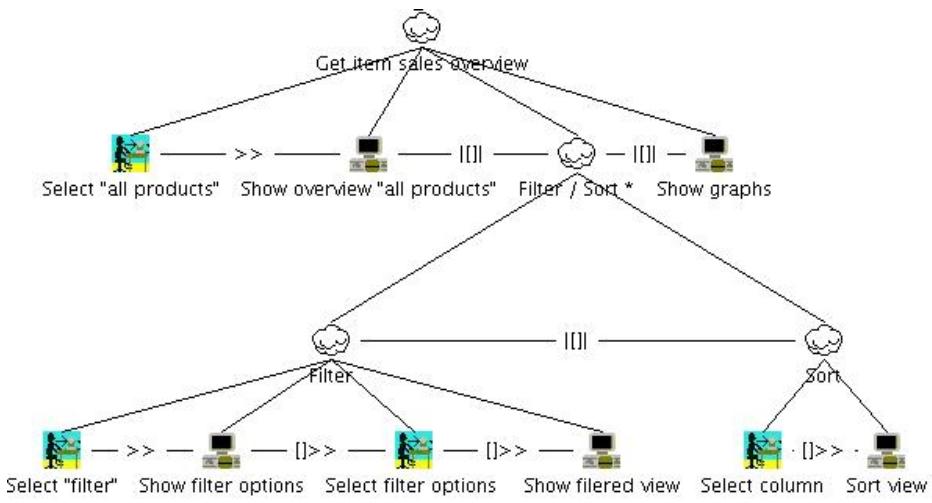


Figure 6.15: CTT to show a sales overview

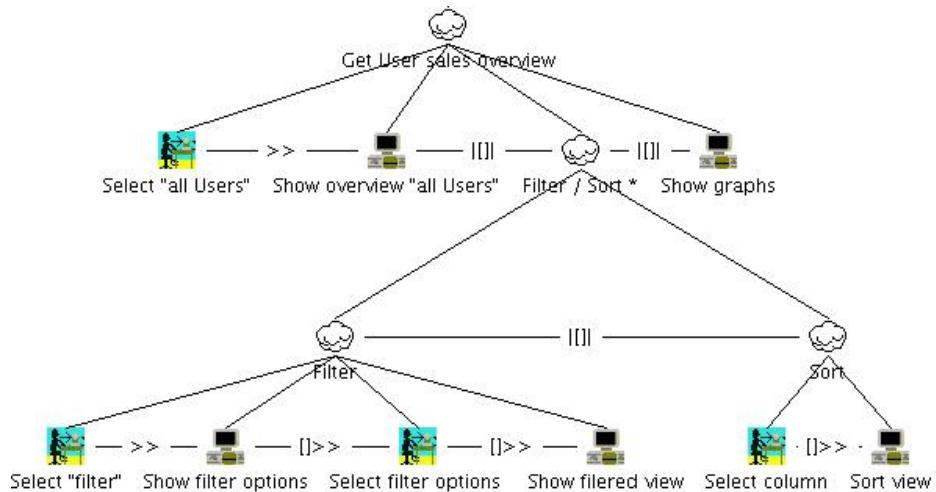


Figure 6.16: CTT to show the amount of sales per employee

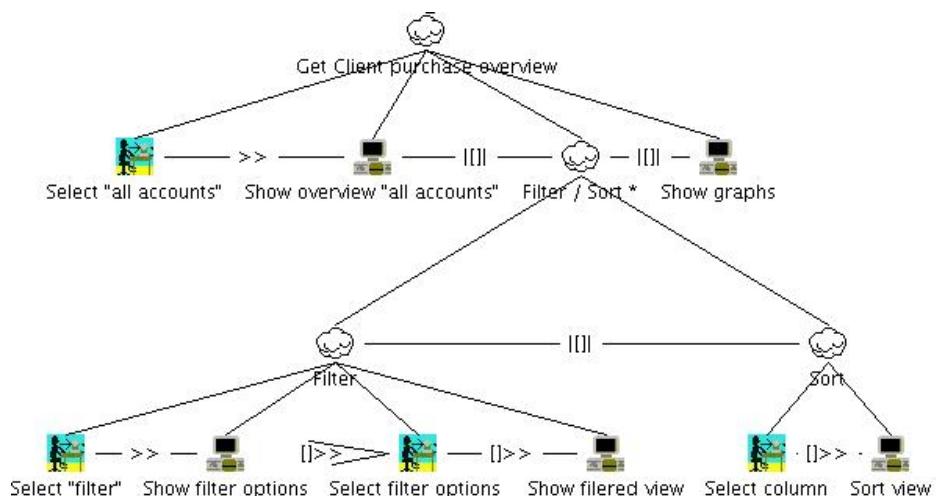


Figure 6.17: CTT show the amount of purchases of the clients

7 User Object Modeling

In the following chapter we will discuss the user objects. We have modeled them using ORM diagrams. This gives us a nice conceptual view of the functionalities that each user of the system has.

Because these ORM diagrams can quickly become big and unclear, we decided to split them up as usual. However some of the objects modeled in one of the diagrams are also used in others. Instead of giving the full definition over and over again, we indicated which objects are defined elsewhere. These objects are indicated with a shadow under them.

7.1 Stock Manager

The Object diagram for the Stock Manager can be seen in figure 7.1

The Stock Manager is able to edit and view the amount of items that are currently in stock. He also needs to create new items for the shop when this is necessary.

7.1.1 Stock

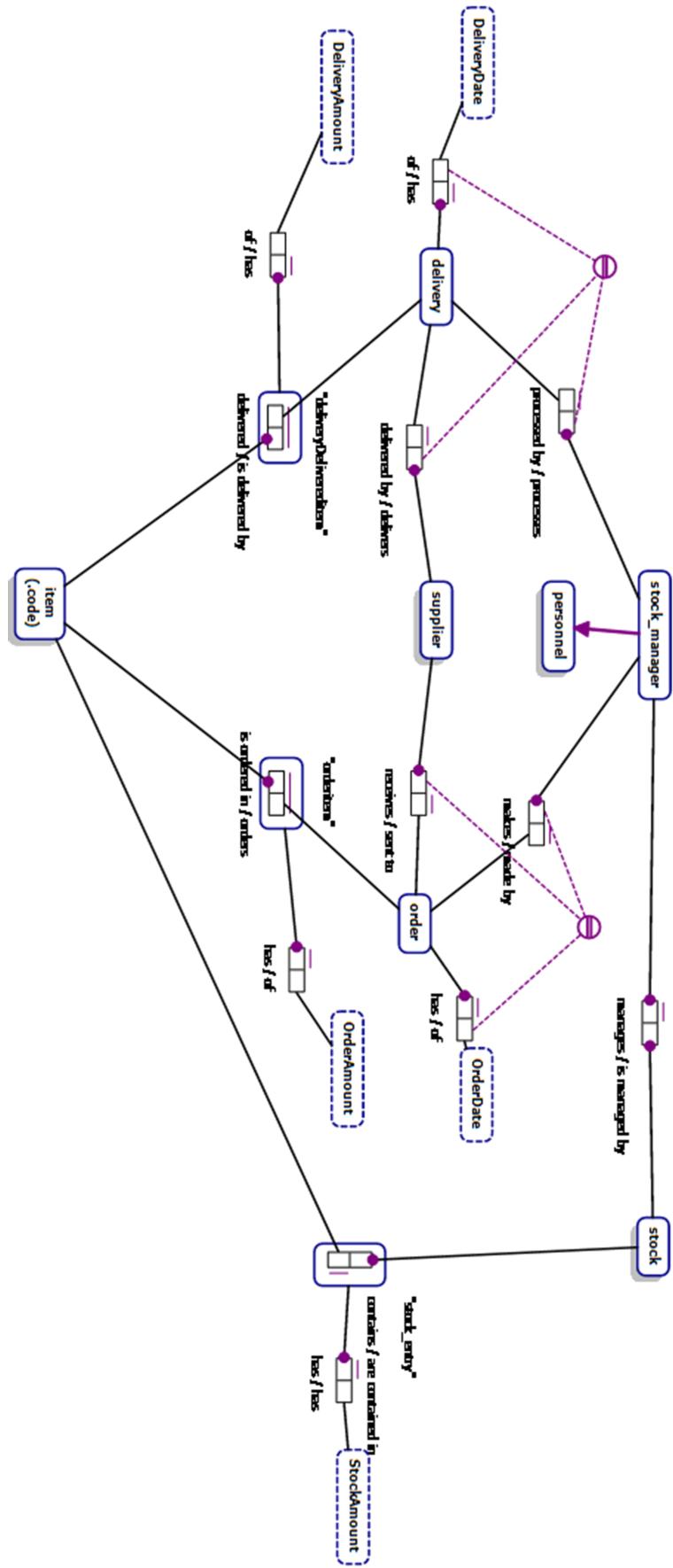
The Stock-Object is unique for a shop, and represents the actual real-world stock of the shop. All items that are being sold in the shop have a unique entry in the Stock. This entry contains all the information the stock manager needs to manage the orders and process the deliveries. This includes the prices of the products, and how many of the items are currently in stock.

7.1.2 Orders

When the Stock Manager notices that one of the items is running out of stock, he will create a new order. This order contains again of entries with all the information needed to place a real-world order with one of the suppliers. When he has entered all the items and amounts he is buying, he can choose a supplier. The supplier will then receive the order, and make the delivery.

7.1.3 Deliveries

The Supplier is an external user who will deliver items to the shop. These deliveries are entered into the system by the stock manager. He will be able to scan the items, in order to update the current amount that is in stock. Or he can create a new item, when it is not



yet present. These deliveries are also logged by the system, and will receive a unique delivery number.

7.2 Shop Clerk

The Object diagram for the Shop Clerk can be seen in figure (7.2)

The Shop Clerk helps the customers in the shop. This means he will need to access the users' accounts. He is also creates entries in the sales log by selling items to the clients.

7.2.1 Sales

A sale objects contains multiple sale-entries. Each of them has an amount and a reference to the item that was sold. The sale is the connection between the client and the shop-clerk, and both are needed. This link is used to log the information, and handle complaints from the clients as well.

7.2.2 History

The History object keeps track of all the items that are bought by the clients. It has a date, so that the overview can be filtered to have a better idea of which clients are shopping on what times.

7.2.3 Account

Each registered client has an internal account. This includes the users of the system. Each account has a balance which can be changed by putting money on the account, or extracting money from it.

7.3 Shop Manager

The Object diagram for the Shop Manager can be seen in figure 7.3

The Shop clerk manages the shop. The interface provide tools to help him with this. He gets an overview page on which he can view all interesting statistics of the system. He can get information of the users, customers and items of the shop. This will allow him to make informed decisions to manage the shop. (e.g. give a raise to hard working personnel, give presents to loyal customers, buy more or less of certain season items during certain periods, ...)

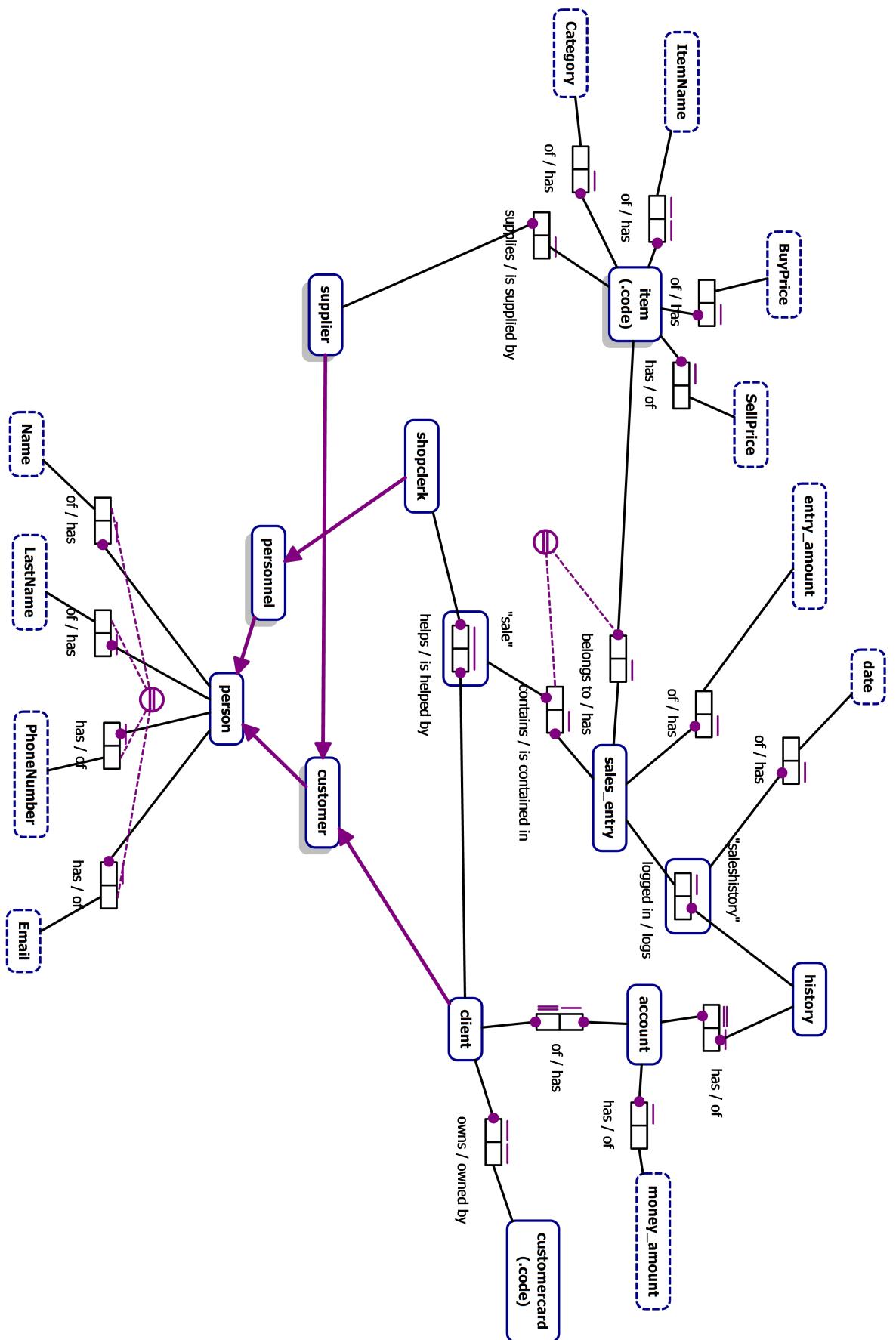
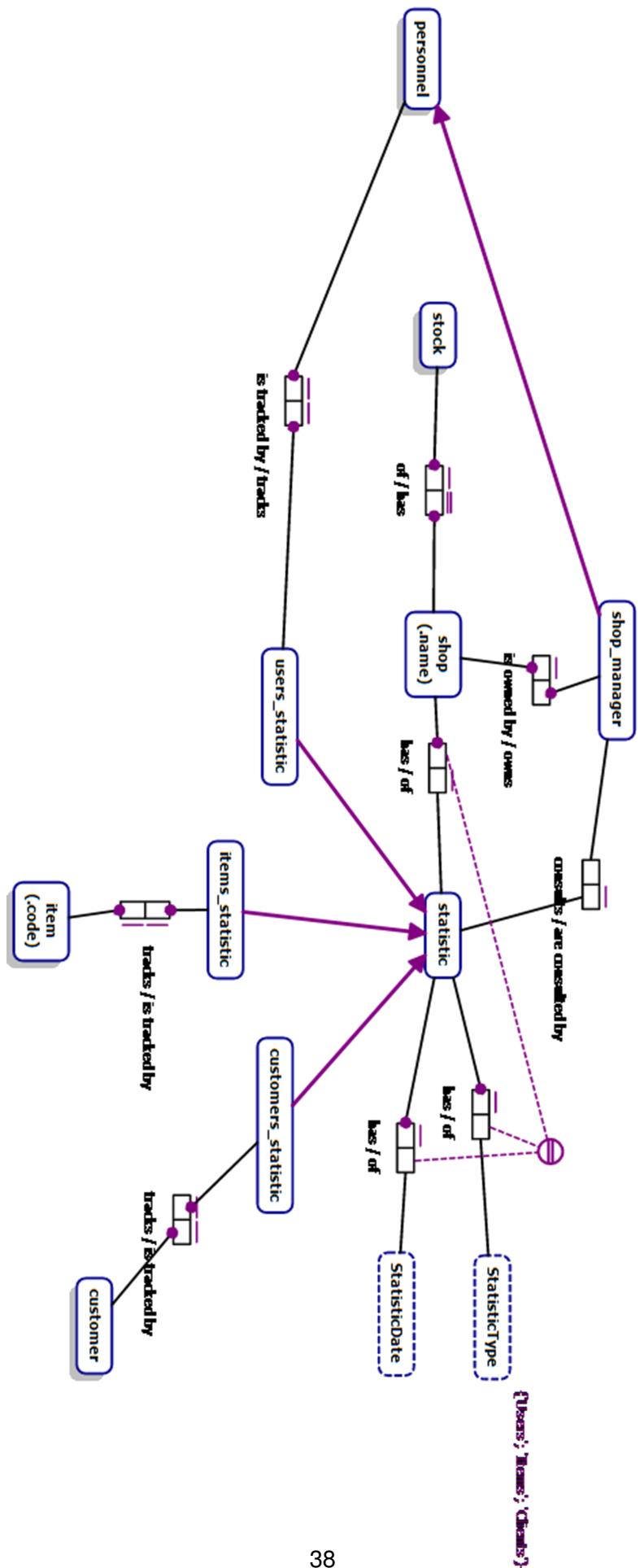


Figure 7.2: Shop-clerk ORM diagram



8 Style Guide

This part of the document will elaborate on the design standards that the application has to comply with.

8.1 Standards for switching modules

The application exists of different modules. It is essential that the user can switch between the modules he has access to. Therefore each module will be seen as a tab. This can be seen in image 8.1. The rightmost tab is used to log out. This button will be distinguished from the other tabs by giving it a different color. The active module will use yet another color to give the user feedback on which module he is currently using.

8.2 Standard Window layout

All windows, with the exception of the sales window, will use the same layout. Next to the module navigation on top they have a secondary, inter-module navigation, navigation. Image 8.2 shows this.

8.3 Sales window

Since the sales module doesn't have any sub menu's, the secondary navigation is omitted. In stead the complete main pane is used. On the left side the store-clerk will scan products. When he scan an account card the account will be shown in top right. This gives easy access to the account information and makes it easy to deposit money on an account. The bottom right is used to put the payment buttons. A simple preview of this can be found in image 8.3

8.4 Appearance

The Application will be modelled using html5 and css3. This means the application can be packaged with a browsers that allow application mode, such as Google Chrome. Since the rendering of html5 and css is consistent within browser families, the application will look the same on every webkit based browser (e.g. Google Chrome, Safari, ..). This means the application will look identical across different operating systems. Since we used html it will be very easy to port the current interface to a web application.

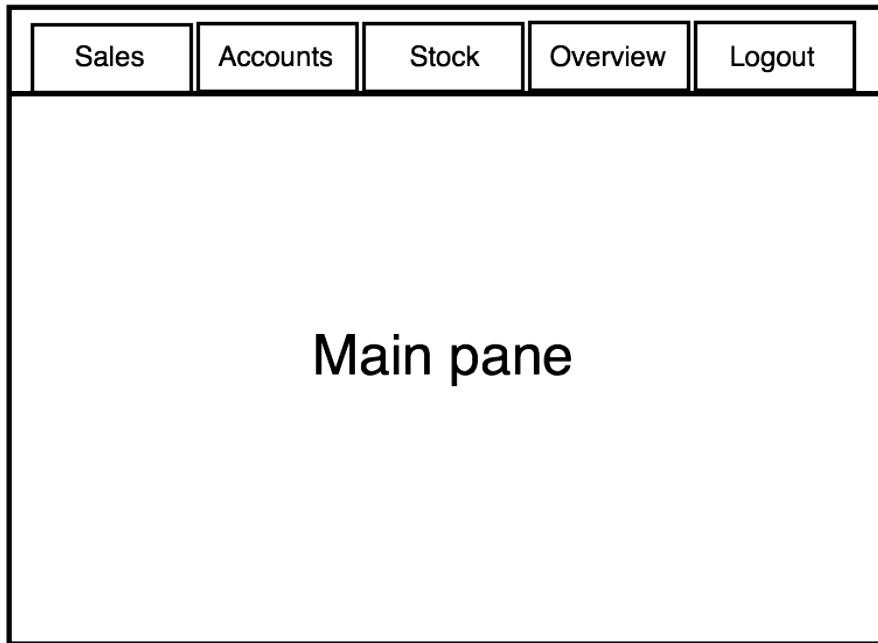


Figure 8.1: The general view,tabbed modules on top

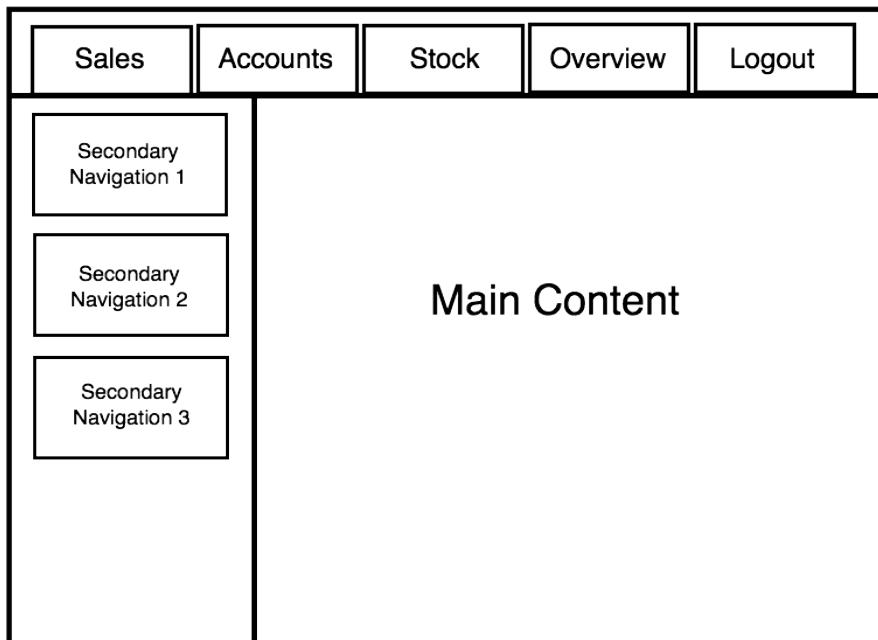


Figure 8.2: A standard window

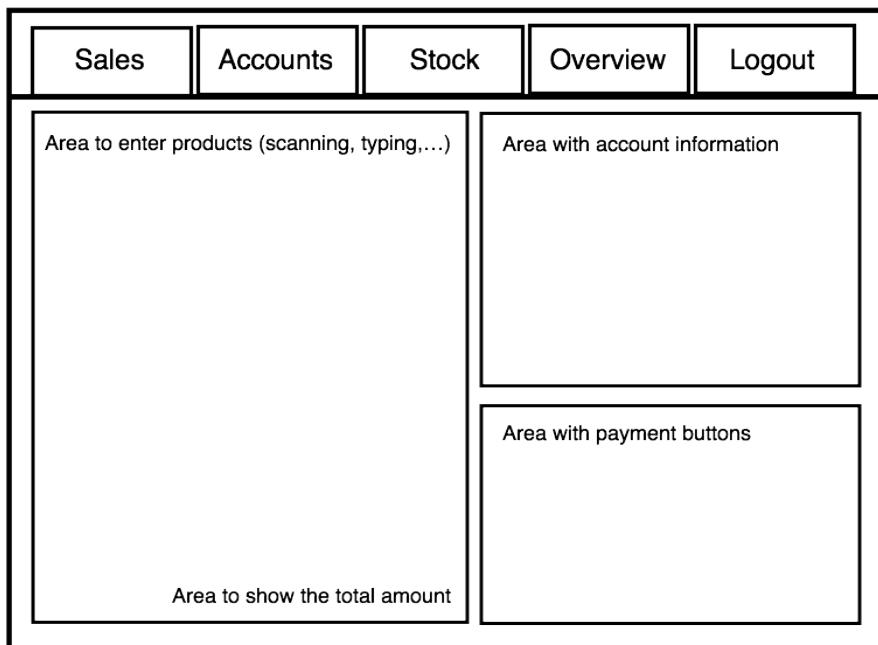


Figure 8.3: A standard window

8.5 Navigation

In the previous mock-ups (images 8.2, 8.1, 8.3) you see that the buttons are rather big. This keeps the interface clean and makes it easy to use the interface using a touch screen. All the navigation in this application will be usable on touch screens.

8.6 Standards for color

All colors are based on a color scheme, image 8.4, that was created for this application. The orange color is mainly used to indicate current location (i.e. The current module tab will be coloured orange). The blue tints are used to give slightly emphasise some text (i.e. the total amount that has to be payed will be a blue tint. Also the logout button will use this color). Input fields have two possible colors. The lighter color means the input field is usable, the darker input fields are input fields that are disabled.



Figure 8.4: The color scheme

9 Design Report

9.1 General tasks

When the User starts the application he first has to log in. Since the application is modular there is an distinction between the different modules. Every module will be accessed through the button on top of the application. The current module will be highlighted.

9.1.1 Enabled task Set

This task set is based on the the CTT in figure 6.1.

```
{  
    {Enter name, Employee card login},  
    {enter password},  
    {check Data},  
    {Display Error},  
    {User Tasks, logout / close}  
}
```

9.1.2 Login

Section 10.1 shows the implementation of the login screen. When a user clicks on his name to login the background wil be grayed out and a pop-up will be given (see image 10.2). When the login is successfully the system will load the "home" module for that user in the background while a positive feedback pop-up is shown. This pop-up stays for 5 seconds and then automatically disappears. The user can also click anywhere to remove the overlay, as is shown by the text on image 10.3.

9.2 Stock Module

Section ?? shows the design of the stock module.

The stock module shows buttons to navigate between all functionality of the module. The main page also lists an overview of the items that are in stock, with their amounts.??

9.2.1 Enabled task Set

It is Bases on the Stock manager CTT in figure 6.2

```
{  
    {goto stock module},
```

```

{goto different module},
{Select list all products, select create product,
    select the product, select process delivery},
{Show products, Select filter options, select column}
{Filter show products},
{order products column},
{show create form},
{fill in form},
{process form},
{exit form},
{show product},
{edit product details, select order product},
{Show order form},
{confirm order},
{update product},
{show form},
{give supplier, give product}
{update form},
{submit delivery},
{process delivery}
}

}

```

9.2.2 List All

This page enables the stock manager to edit the amount in stock for the items. The stock manager is aided with filters to make bulk editing easier. As seen on figure 10.4

When the user clicks in any of the input fields he can change the amount that is in stock.

When he presses the change button all items in the current list will be updated.

When he wants to do bulk updates he will open the filter menu, and chooses how to filter the lists. The new list is automatically generated when the filters change, and the new list can be updated in the same way as the initial list.

Clicking on any item in the lists will bring the user to that specific items' page.

9.2.3 Item

An example of the item page can be seen on figure 10.5. This page shows all the information of the item. It shows the statistics of how often it is sold in different months. Clicking in any of the information fields will allow the stock manager to update that information. This is particularly useful when the bar-code of certain items changes.

Clicking on the Change stock button allows the user to edit the stock of that item. (This is not the default way of editing stock, but is allowed nonetheless)

The list at the bottom of the page gives an history of the prices it had based on the delivery it was in.

9.2.4 New Delivery

The new delivery page can be seen on figure 10.6. This user will click the New Delivery button on the left of the screen to access this page from the stock module. He will then take the items that where delivered by the supplier, and scans them. If the scanner is broken, or the bar-code is unreadable, the user can type the name of the item in the field (which will autocomplete the name).

The items scanned (or entered) this way will be grouped by name. Scanning an item multiple times will increase the number. Independently of how many other items are scanned in between.

To make it easier for the user, the stock manager can also manually edit the amount that is to be added to the stock. This prevents the user from having to scan an item alot when he knows the delivered amount beforehand.

9.2.5 New Item

When at any point in the stock module the user encounters an item that is nog yet in the system he can click the New Item button. This will open a popup-form (seen on figure 10.7) in which the user can type all the required information. And click Create to submit. The window will disappear, and the page will show again as it was before. This method of entering new items will minimize the memory usage of the user, since the task they were doing before will still be in the background, and can be picked up seamlessly.

9.2.6 Orders

The order page can be seen on figure 10.8 it shows a list of orders. The status of the order is listed with the entries. This makes it easy for the user to follow the deliveries when needed.

Clicking on one of the orders will open the listed orders will open that orders overview page.seen on figure 10.9

9.3 Sales Module

9.3.1 Enabled task Set

Bases on the store-clerk CTT in the figure 6.8

```
{  
{scan product, enter user},  
{search product},  
{update view},  
{enter received amount. confirm payment (account)},  
{show return amount},  
{select process payment},  
{process payment},  
{show account},
```

```

{select add credit},
{show form},
{enter amount},
{process deposit}
}

```

This module consists of one page (seen in figure 10.10) on which the shop clerk can help the customers with their sales. When the customer arrives with his goods, the user asks for his customer card and scans it.

The input field is smart enough to know when a customer card is scanned, or an item. This means that the customer can search for his card, while the user is already processing the items. This may speed up the sales process.

When the card is scanned the information of the customer is shown on the page (figure 10.11). In this pane the user can quickly change the balance of the customers account. This will speed up the process if customers don't want to buy something, but only come to pay back some debts they have on their account. The user doesn't have to leave the sales window then.

When items are scanned they are listed in the left pane. Like the stock module similar items are grouped, and it doesn't matter in which order the scans happen.

When the sale is done the user can click one of the buttons on the left to start the payment. Popup windows will ask for an amount, and this amount is then subtracted from the amount that is still to be paid (figures 10.12, 10.13, 10.14)

9.4 account Module

9.4.1 Enabled task Set

Bases on the store-clerk CTT in the figure 6.9

```

{
{select new account, select filter, enter seach term}
{show form},
{fill in all details},
{process creation},
{print membership card},
{show filters},
{set balance range},
{show result},
{select account},
{show account},
{edit details},
{update account},
{select amount},
{select deposit or retrive},
{Process transaction}
}

```

9.5 Overview Module

Bases on the store-manager CTT in figure 6.14 All the overviews are basicly the same, the only thing that differs is the date you start of with.

9.5.1 Enabled task Set

```
{  
    {select all accounts, select all users, select all products},  
    {show overview of all accounts, select filter, select column, show graphs},  
    {show overview of all users, select filter, select column, show graphs},  
    {show overview of all products, select filter, select column, show graphs},  
    {show filter options},  
    {select filter options},  
    {show filtered view},  
    {sort view}  
}
```

This module allows the shop manager to see the statistics of his shop. The interface of the page is shown in figure 10.15

This is also the default page when entering the overview module.

There are 2 other overview pages, shown in figure 10.16 and figure 10.17. These two pages show the overviews for customers ans items.

10 Prototype

This chapters provides the screen-shots of the final prototype. More information on these screen-shots can be found in the previous chapter.

10.1 Login

10.2 Stock

10.3 Sales

10.4 Overview



Figure 10.1: The login window

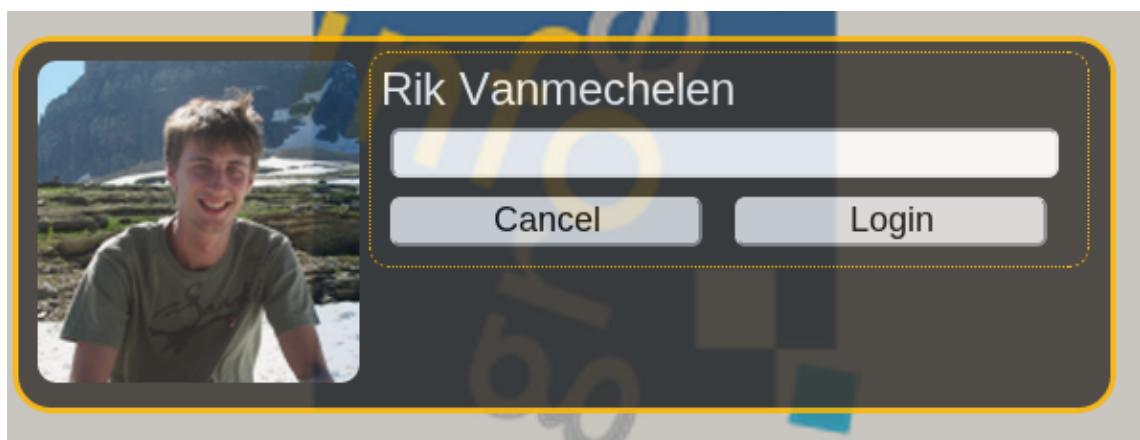


Figure 10.2: The login dialog

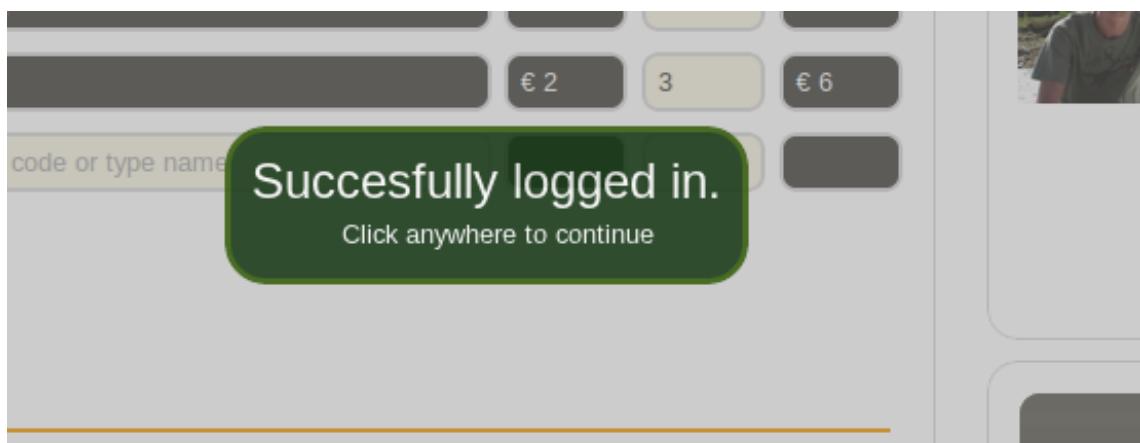


Figure 10.3: Succesfull login feedback

HIDE FILTERS

Buy price

Sell price

Amount in stock

Supplier
Category

Figure 10.4: Stock filter

Royco: Tomato soup

Statistics

Highcharts.com

Name	Royco: Tomato soup
Code	9780735711020
Supplier	Colruyt
Buy Price	€2
Sell Price	€4

In Stock: 30

[Change Stock](#) [Order](#)

Date	Supplier Name	Amount	Unit Cost	Total Cost	Order Nr
11/12/13 14:15	Colruyt	30	€2	€4	51964
11/12/13 14:15	Colruyt	30	€2	€4	51965

Figure 10.5: Item

New Delivery

Supplier

Remove	Item	Unit sell price	Unit buy price	Amount delivered
X	Royco: Tomaten soep	€ 24	€ 4	6
X	Coca Cola: can 33cl	€ 6	€ 2	3
<input type="text" value="Scan product or type code or type name"/>				

TOTALS

€ 6	9
Submit	

Figure 10.6: New Delivery

Create New Stock Item

Item Name	Chocolade
Item code	3563278
Supplier	Colruyt
Unit Sell Price	€ 4
Unit Buy Price	€ 2
Cancel	Create

Figure 10.7: New Item

Orders

SHOW FILTERS

Date	Supplier Name	Total Cost	Status	Order Nr
11/12/13 14:15	Colruyt	€122	Ordered	51964
11/10/09 12:13	Coca Cola Company	€250	Delivered	51962

Figure 10.8: Orders

Order Nr : 51964

Supplier : Colruyt

Item	Unit sell price	Unit buy price	Amount delivered
Royco: Tomaten soep	€ 24	€ 4	6
Coca Cola: can 33cl	€ 6	€ 2	3

Figure 10.9: Order

Sales	Accounts	Stock	Overview	Logout Rik
Remove  Royco: Tomaten soep  Coca Cola: can 33cl Scan product or type code or type name	Item unit price amount total price € 4 6 € 24 € 2 3 € 6 Scan product or type code or type name	TOTAL AMOUNT TO BE PAYED € 30	 Name: Rik Lastname: Vanmechelen Email: rik.vanmechelen@gmail.com Phone: 011/12 13 14	€100.99 Add Extract Pay cash Pay with Account

Figure 10.10: sales

	Name: Rik Lastname: Vanmechelen Email: rik.vanmechelen@gmail.com Phone: 011/12 13 14
€100.99 Add Extract	

Figure 10.11: client field sales



Figure 10.12: pay with cash

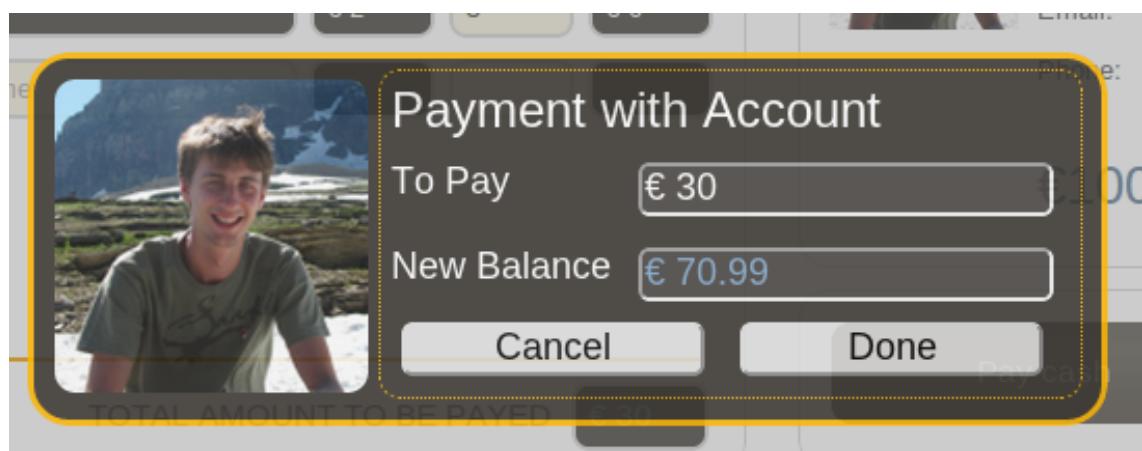


Figure 10.13: pay with account



Figure 10.14: left to pay

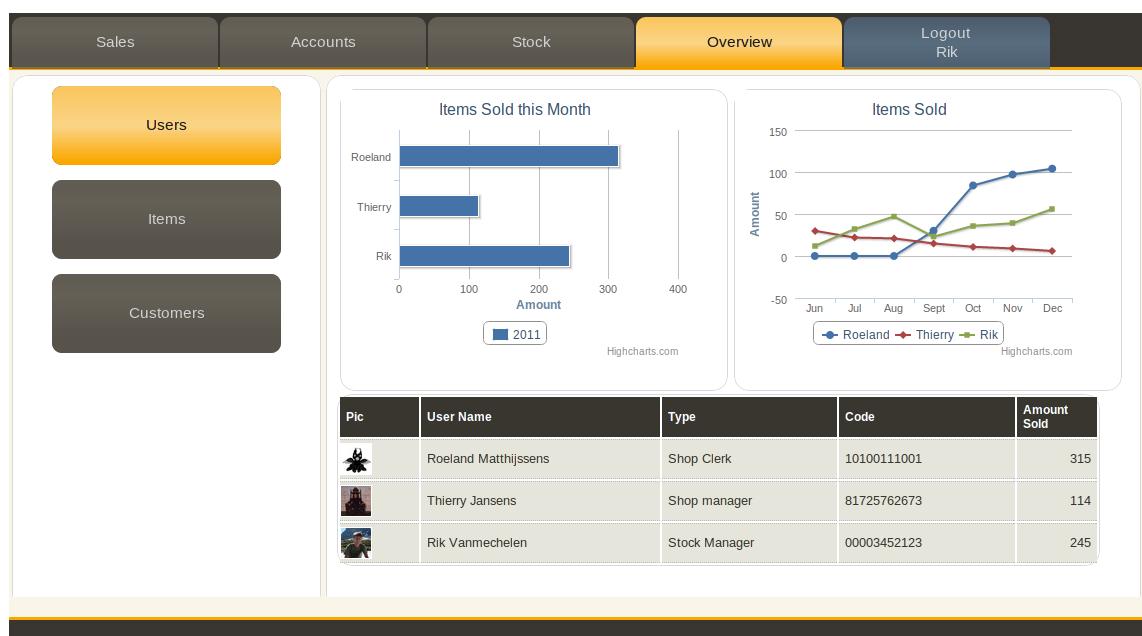


Figure 10.15: Overview



Figure 10.16: Customer Overview



Figure 10.17: Item Overview