

SoBeNet : Financial Case Study Part 1 : Requirements and Analysis

Bert Lagaisse

Bart De Win

Wouter Joosen

Report CW404, February 2005



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

SoBeNet : Financial Case Study

Part 1 : Requirements and Analysis

Bert Lagaisse

Bart De Win

Wouter Joosen

Report CW404, February 2005

Department of Computer Science, K.U.Leuven

Abstract

This document defines a case study in the world of e-finance. The scope of this case study is retail banking services. In this case study we target on a subset of retail banking services : basic banking services (like current accounts and savings accounts) and more advanced retail banking services concerning private investments. In this document we describe the first part of the case study: the requirements and analysis.

Contents

1	Introduction	3
1.1	Financial products	4
1.1.1	Stocks	4
1.1.2	Shares	5
1.1.3	Options	5
1.2	The trading process	5
1.3	Investment services	6
1.4	Overview of the rest of this document	6
2	Functional requirements	7
2.1	Actors	7
2.2	Other stakeholders	7
2.3	Use cases for basic banking services	9
2.3.1	Use cases for administration	9
2.3.2	Use cases for transactions	13
2.3.3	Use cases for account inspection	14
2.3.4	Use cases needed for the bank system	15
2.4	Use cases for investment services	16
2.4.1	Use cases for custody account administration	17
2.4.2	Use cases for custody account inspection	17
2.4.3	Use cases for stock market catalogue inspection	18
2.4.4	Stock market transactions	19
2.4.5	Use cases for feedback from stock market system	22
2.5	Use cases for transaction processing	23
2.5.1	Use cases for operational management	24
2.5.2	Use cases for processing a transaction	24
2.6	Domain model	25
3	Non functional requirements : quality attributes	29
3.1	Availability	29
3.1.1	General quality scenarios	29
3.1.2	Specific quality scenarios	29
3.2	Performance	30
3.3	Usability	30
3.4	Security	30

4	Security Requirements	31
4.1	Non repudiation and accountability	31
4.2	Access control	32
4.2.1	Role Based Access Control	33
4.2.2	Owner based access control	34
4.3	Data integrity	35
4.4	Authentication	35
4.4.1	Entity authentication	35
4.4.2	Data authentication	35
4.5	Privacy and confidentiality	36
5	Conclusion	37

Chapter 1

Introduction

The goal of this document is to define a case study in the world of e-finance. The scope of this case study is retail banking services. In this case study we target on a subset of retail banking services : basic banking services (like current accounts and savings accounts) and more advanced retail banking services concerning private investments. Other services (loans and insurance) are out of the scope of this case study.

We assume that the reader is familiar with the basic banking services and their possibilities. Each customer of a bank owns a current account to do the daily transactions like cash deposits, cash withdrawals and transfers.

Next to a current account a customer can have a savings account, which is associated with the customer's current account. Savings accounts offer a higher interest rate but have some restrictions on the possible transactions with it. It is not possible to do a cash withdrawal on a savings account and transfers are only possible to the associated current account. For the rest of this introduction we mainly elaborate on the services for private investments.

In this case study we focus on retail banking services hence we limit the world of the stock market transactions to the private investments. The scope of this case study is limited to a subsystem of the administrative software of the bank for the retail banking services mentioned above. This means that the internal workings of other systems involved (like the stock market) are not considered part of this case study. Only communication between these other systems and the bank's system will be included.

The typical users of the system for retail banking services include bank clerks and customers of the bank. Bank clerks use the system from workstations in different branch offices, to handle requests from a customer at the branch office. Customers can use the system directly through a self-banking terminal, through home banking or indirectly as a secondary (supporting) actor at the bank's branch office.

In the rest of this chapter we will explain the different financial products, the stock market, the trading process and the bank's investment services. We conclude with an overview of the rest of the chapters in this document.

1.1 Financial products

There are a lot of financial products to invest in:

- The basic investment products like savings accounts.
- Products with a higher risk like stocks.

In this section we will focus on *stocks*. There are two kinds of stocks: securities and derivatives. In this case study one kind of security and one kind of derivative will be considered:

- *Shares*(as a kind of security)
- *Options* (as a kind of derivative)

First we will explain the main concepts of stocks, shares and options. For the sake of this case study, irrelevant details will be simplified or even omitted.

1.1.1 Stocks

A stock is a financial instrument in which one can invest. It has a certain trading price that fluctuates based on the economical laws of ask and bid. All stocks have some common properties:

Stock Code (ISIN): A standardized identification number to uniquely identify a stock across all markets.

Last Sale: The last trading price.

Currency: The currency in which the stock is notated.

Today's High: The intra-day highest trading price.

Today's Low: The intra-day lowest trading price.

Historical notations: The historical notations of a stock are a list of all important properties of a stock for each trading day at the market. A notation for one trading day contains: the date, opening trading price, highest trading price, lowest trading price and closing trading price

The market. Every stock is notated at a certain market. At his market stocks can be traded in a controlled way. At this market the laws of ask and bid determine the trading price of the stocks. Normally a stock is notated at just one market and can only be traded on this market. Markets are determined by a unique market code. Often markets specialize in one kind of stock ¹. All stocks notated at a market have the same currency. This currency is determined by the market. Each market has a stock catalogue in which is publishes the stocks notated at that market. It contains the different values associated with a stock and the historical notations of the stock.

¹like shares or options

1.1.2 Shares

A share is a participation in the capital of a corporation. Next to the properties mentioned in the generic description of a stock, shares also have historical notations of the traded volume on a certain day.

A share also gives the right on a part of the profits of the corporation one is participating in. Every year the corporation can decide to pay a dividend to the shareholders. This dividend can be a money payment, shares, options or other values.

1.1.3 Options

An option is the right to buy (call option) or sell (put option) a certain underlying security at a price which is determined in advance by the financial institution that issues the option. It is valid for a limited period. Typical properties of an option are:

The kind of option: call or put option

The underlying security: the security for which the option gives you the right to sell or to buy. For example, this security can be a share of IBM.

Period: The period during which the option can be executed.

Execution price: The price to buy or sell the underlying security. This price is granted by the financial institution that issues the option.

Trading price: Next to the execution price, an option also has a trading price. As mention in the properties of a stock, this is the price against which the option is traded.

Quantisation: The amount of securities on which the option applies.

1.2 The trading process

When one wants to buy or sell an amount of securities, an order is sent to the market. All orders are notated in the order book at the market and displayed. This order book has two sides: the selling orders and the buying orders. So first, orders are divided in selling and buying orders. Secondly orders are divided in direct orders and limited orders. A direct order is an order to buy or sell a stock at the market's current best displayed price. A limit order is an order to buy or sell a stock at a customer specified price. Based on these limit orders the best ask price and best bid price for a stock at a certain moment are known. The best bid price is the highest buying price of all limited buying orders. The best ask price is the lowest selling price of all limited selling orders.

The market will try to process the orders by searching matches between the two sides of the order book. So it will try to match a selling order with a buying order. The amount of stocks involved in the orders is not necessarily the same in matching orders. For the amount of stocks in an order that could not be matched, a new order is created and entered in the order book.

When a match is found, the trading transaction will be published. The settlement of the transaction (exchange of the values and payment) is handled by a settlement organisation designated by the stock market.

1.3 Investment services

The investment services of a bank are offered to the customer by means of a *custody account*. This account is used to do the financial transactions involved in trading stocks. In case of buying stocks, the amount to pay is withdrawn from the account, in case of selling stocks it is deposited on the account.

A customer can choose the currency of each of his custody accounts, depending on the market he is frequently trading on. This saves the customer a lot of currency exchange costs. So normally, a custody account in dollar will be used to trade on the stock markets in dollar.

To open a custody account, a customer already needs a current account, with which the custody account will be associated. The possible money transactions with a custody account are limited. The first possible money transactions are the money transfers involved in the settlement of a trading transaction on the stock market. The second possible money transactions are transfers with the associated current account.

When a customer has a custody account, he can request the bank to place an order on a stock market. The custody account contains an overview of all orders that are still pending on the market. When the stock market gives feedback about an order, the pending order will be marked as confirmed (if the order could be processed by the market) or it will be marked as rejected (if the order is rejected by the market). In case the order is confirmed, the market will also give the details about the associated trading transaction.

A custody account also contains a *security depot*. Within a security depot a customer can open a position on a certain stock. Opening a position on a stock means the customer wants to follow up on this stock and wants to invest in it. All trading transactions of the customer on this stock are stored in this security depot, in the open position on the stock. Each open position shows the amount of stocks in the position, profits or losses already made and important properties of the stock like its current trading price and the historical notations.

When a customer is no longer interested in investing in a certain stock, he can close the position. The amount of stocks in the position has to be zero. Closed positions are stored in the *realization depot*. The realization depot contains for each closed position the definitive profit or loss and the trading transactions on the stock.

1.4 Overview of the rest of this document

In the second chapter we will elaborate on the functional requirements of the bank's private investment software. The third chapter will discuss important non-functional requirements (or software qualities). The fourth chapter will further elaborate on the security requirements.

Chapter 2

Functional requirements

In this chapter we discuss the functional requirements. These functional requirements are expressed in the form of actors, stakeholders and use cases. The description of requirements is divided into three sections: basic banking services, investment services and transaction processing. A graphical overview of the functional requirements can be found in the next figure. At the end of this chapter a domain model is added to offer a better overview of the problem domain.

2.1 Actors

The actors of the banking system are the entities that use the functionality of the system. This includes people, organizations and software systems. We distinguish the following primary actors in the system: Bank clerks and customers.

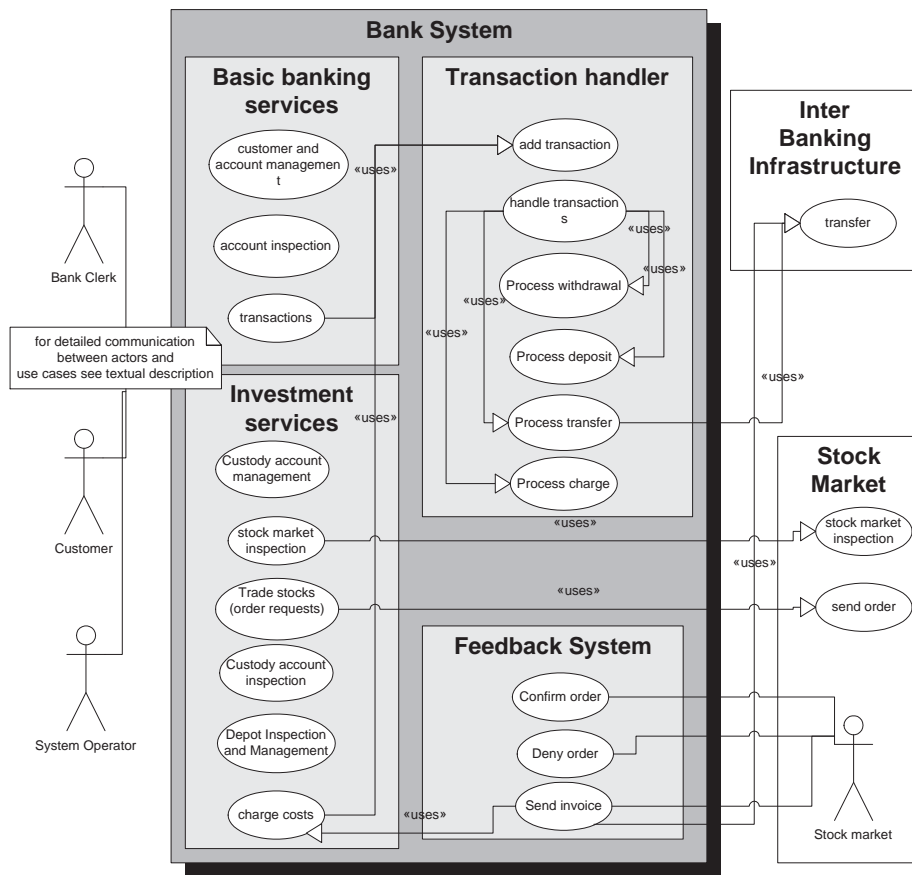
Bank clerks use the system from workstations in different branch offices, to handle requests from a customer at the branch office.

Customers use the system directly through a self-banking terminal, through home banking or indirectly as a secondary (supporting) actor at the bank's branch office.

Other important actors of the system are the following members of the bank's personnel: the office manager and the operational manager. The following systems are also actors within the banking system: the transaction handler, the stock market system and the settlement organization. These will be discussed later.

2.2 Other stakeholders

Inter banking infrastructures Because there are also money transfers between different banks and also between the stock market and the banks, we will assume that every financial institution has an account at some inter banking infrastructures. These organizations are trusted third parties to enable transactions between financial institutions.



2.3 Use cases for basic banking services

This section describes the uses case of the subsystem for the basic banking services. First we summarize the uses cases for administration, transactions, account inspection and inter banking transactions. Second, we describe these uses cases in detail using the usecases.org format.

Use cases for administration (bank clerk only)

- Create new customer
- Show customer information
- Edit a customer
- Open a current account
- Close a current account
- Open a savings account
- Close a savings account
- Block accounts of a customer

Use cases for transactions (bank clerk and customer)

- Withdraw from a current account
- Deposit on an account
- Transfer between two accounts

Use cases for account inspection (bank clerk and customer)

- Show account information.
- Search a transaction.

Use cases needed for the bank system

- Transfer between the bank's own account and another financial institution.

2.3.1 Use cases for administration

2.3.1.1 Create new customer

Primary actor: Bank clerk

Supporting actor: Customer

Stakeholders: Customer, Bank, Bank clerk

Basic flow:

1. The actor selects to create a new customer.
2. The system asks for the required information about the customer (first name, last name, birth date, address, social security number).
3. The actor gives the required information.
4. The system verifies the given information.
5. The system creates a new customer when verification succeeded.
6. The system notices the actor (success or failure).

2.3.1.2 Show customer information

Primary actor: Bank clerk

Basic flow:

1. The actor selects to show a customer's information.
2. The system asks for the social security number.
3. The actor gives the social security number.
4. The system shows the customer information

Alternative scenarios:**1-4a. The customer has to be looked up by name**

1. The actor chooses to search for the customer.
2. The system asks for the name of the customer.
3. The actor gives the name of the customer.
4. The system shows a list of matching customers.
5. The actor selects a customer from the list.
6. The system shows the customer information.

2.3.1.3 Edit customer information

Primary actor: Bank clerk

Supporting actor: Customer

Stakeholders: Customer, Bank, Bank clerk

Preconditions: The actor has selected to show a customer's information.

Basic flow:

1. The actor edits the customer's information.
2. The system verifies the given information.
3. The system creates a new customer when verification succeeded.
4. The system notices the actor (success or failure).

2.3.1.4 Open a current account

Primary actor: Bank clerk

Supporting actor: Customer

Stakeholders: Customer, Bank, Bank clerk

Basic flow:

1. The actor selects to open a new current account.
2. The system asks for the owners of the current account.
3. The actor adds customers as owners.
4. The system verifies the given information.
5. The system generates an account number.
6. The system asks to sign the request for a new current account
7. Each owner and the bank sign the request
8. The system creates a new current account.
9. The system notices the actor (success or failure).

2.3.1.5 Close a current account

Primary actor: Bank clerk

Supporting actor: Customer

Stakeholders: Customer, Bank, Bank clerk

Preconditions: The actor has selected a current account.

Basic flow:

1. The actor selects to close a current account.
2. The system verifies on the existence associated savings accounts.
3. The system verifies if the balance of the current account is zero.
4. The system asks to sign the request for closure.
5. Each owner and the bank sign the request.
6. The system closes the current account.
7. The system notices the actor (success or failure).

2.3.1.6 Open a savings account

Primary actor: Bank clerk

Supporting actor: Customer

Stakeholders: Customer, Bank, Bank clerk

Basic flow:

1. The actor selects to open a new savings account.
2. The system asks for the associated account of the savings account.
3. The actor selects the associated account.
4. The system verifies the given information.
5. The system generates an account number.
6. The system asks to sign the request for a new account
7. Each owner and the bank sign the request
8. The system creates a new savings account.
9. The system notices the actor (success or failure).

2.3.1.7 Close a savings account

Primary actor: Bank clerk

Supporting actor: Customer

Stakeholders: Customer, Bank, Bank clerk

Preconditions: The actor has selected a savings account.

Basic flow:

1. The actor selects to close a savings account.
2. The system verifies if the balance of the account is zero.
3. The system asks to sign the request for closure.
4. Each owner and the bank sign the request.
5. The system closes the savings account.
6. The system notices the actor (success or failure).

2.3.1.8 Block accounts of a customer

Primary actor: Bank clerk

Supporting actor: Office manager

Stakeholders: Customer, Bank, Bank clerk, Office manager

Preconditions: The actor has selected a customer.

Basic flow:

1. The actor selects to block the accounts of the selected customer.
2. The system looks up the accounts of the customer.
3. The system asks to sign the request to block the accounts.
4. The office manager signs the request.
5. The system blocks the accounts.

2.3.2 Use cases for transactions

2.3.2.1 Withdraw from a current account

Primary actor: Bank clerk or customer

Supporting actor: Customer (in case Bank clerk is primary actor)

Stakeholders: Customer, Bank, Bank clerk (in case Bank clerk is primary actor)

Preconditions: The actor has selected a current account.

Basic flow:

1. The actor starts a withdrawal.
2. The system creates a new withdrawal.
3. The system asks for the amount to withdraw.
4. The actor enters the amount to withdraw.
5. The system verifies the withdrawal.
6. The system asks to sign the withdrawal.
7. The customer and the bank sign the withdrawal.
8. The system sends the withdrawal to the transaction handler.
9. The cash money is provided to the customer.

Extensions:

- 9a The bank clerk provides the money to the customer.
- 9b The self banking terminal provides the money to the customer.

2.3.2.2 Deposit on an account

Primary actor: Bank clerk.

Supporting actor: Customer.

Stakeholders: Customer, Bank, Bank clerk.

Preconditions: The actor has selected an account.

Basic flow:

1. The actor starts a deposit.
2. The system creates a new deposit.
3. The system asks for the amount to deposit.
4. The actor enters the amount to deposit.
5. The system asks to sign the deposit.
6. The customer and the bank sign the deposit.
7. The system sends the deposit to the transaction handler.

2.3.2.3 Transfer between two accounts

Primary actor: Bank clerk or customer

Supporting actor: Customer (in case bank clerk is primary actor)

Stakeholders: Customer, Bank, Bank clerk (in case bank clerk is primary actor)

Basic flow:

1. The actor selects an account.
2. The actor starts a transfer.
3. The system creates a new transfer.
4. The system asks for the destination account
5. The actor enters the destination account
6. The system asks for the amount to transfer.
7. The actor enters the amount to deposit.
8. The system verifies the transfer.
9. The system asks to sign the transfer.
10. The customer and the bank sign the transfer.
11. The system sends the transfer to the transaction handler.

2.3.3 Use cases for account inspection

2.3.3.1 Show account information

Primary actor: Bank clerk or customer

Supporting actor: Customer (in case bank clerk is primary actor)

Stakeholders: Customer, Bank, Bank clerk (in case bank clerk is primary actor)

Basic flow:

1. The actor selects to show an account's information.
2. The system asks for the account number.
3. The actor gives the account number.
4. The system shows the account information containing the account number, owners of the account, balance of the account and an overview of pending transactions and completed transactions.

Alternative scenarios:

1a. The account has to be looked up by the customer's id

1. The actor chooses to search for the account.
2. The system asks to select a customer.
3. The actor selects a customer.
4. The system shows a list of accounts of the customer.
5. The actor selects an account from the list.
6. The system shows the account information.

Alternative scenarios:**1a. The customer is operating on the system using home banking**

1. The actor selects to show an account's information.
2. The system shows a list of accounts of the customer.
3. The actor selects an account from the list.
4. The system shows the account information.

2.3.3.2 Search a transaction

Primary actor: Bank clerk or customer

Supporting actor: Customer (in case bank clerk is primary actor)

Stakeholders: Customer, Bank, Bank clerk (in case bank clerk is primary actor)

Preconditions: The actor has selected an account.

Basic flow:

1. The actor selects to search a transaction
2. The system asks for searching variables like the type of transaction, the range of the amount of the transaction and in case of a transfer the destination account.
3. The actor gives some of the searching variables
4. The system shows an overview of pending transactions and completed transactions matching the search variables.

2.3.4 Use cases needed for the bank system**2.3.4.1 Transfer between the bank's own account and another financial institution**

Primary actor: The bank system.

Stakeholder: Inter banking infrastructure providing this service.

Basic flow:

1. The bank system requests a transfer at the inter banking infrastructure providing the destination account, the amount to transfer and a message specifying other information
2. The bank and the inter banking infrastructure sign the transfer.

2.4 Use cases for investment services

Use cases custody account administration

- Open custody account.
- Close custody account.

Use cases for custody account inspection

- Show custody accounts from customer
- Show custody account information

Use cases for custody account transactions These transactions are identical to the transfer between normal accounts and will not be worked out in detail.

- Transfer from account to custody account
- Transfer from custody account to account

Use cases for stock market catalogue inspection

- Show list of markets
- Show list of stocks on market
- Show stock information
- Search stock information based on company name

Use cases for depot management

- Open a position on a stock
- Close a position on a stock

Use cases for stock market transactions

- Place a direct order for a stock
- Place a limited order for a stock
- Execute put option
- Execute call option

Use cases for depot inspection

- Show pending orders
- Show security depot: list of open positions
- Show details of an open position
- Show realization depot : list of closed positions
- Show details of a closed position

Use cases for feedback from stock market system

- Confirm a buying order.
- Confirm a selling order.
- Deny an order.
- Send an invoice for a trading transaction.

2.4.1 Use cases for custody account administration

These are administrative use cases that are restricted to the bank clerks.

2.4.1.1 Open a custody account

Primary actor: Bank clerk

Supporting actor: Customer

Stakeholders: Customer, Bank, Bank clerk

Basic flow:

1. The actor selects to open a new custody account.
2. The system asks for the associated account of the custody account.
3. The actor selects the associated account.
4. The system asks for the currency of the custody account
5. the actor selects the currency
6. The system verifies the given information: all owners should be adults.
7. The system asks to sign the request for a new custody account.
8. Each owner of the associated account and the bank sign the request.
9. The system generates a custody account number.
10. The system creates a new custody account.
11. The system notices the actor (success or failure).

2.4.2 Use cases for custody account inspection**2.4.2.1 Show custody accounts from customer**

Primary actor: Bank clerk or customer

Preconditions: A customer is selected.

Basic flow:

1. The actor selects to get an overview of the custody accounts of the selected customer.
2. The system shows a list of custody accounts of which the customer is owner.

Alternate scenarios

1a The customer is the actor

1. The actor chooses to get an overview of his custody accounts

2.4.2.2 Show custody account information

Primary actor: Bank clerk or customer

Preconditions: The system is showing a list of custody accounts of the involved customer.

Basic flow:

1. The actor selects to get an overview of a custody account.
2. The system shows an overview of the custody account: balance, currency, a list of pending orders, a financial balance of the security depot and a financial balance of the realization depot.

2.4.3 Use cases for stock market catalogue inspection

2.4.3.1 Show list of markets

Primary actor: Bank clerk or customer

Basic flow:

1. The actor selects to get an overview of the markets
2. The bank system asks the stock market catalogue system for a list of the markets
3. The catalogue system gives the bank system the requested list.
4. The bank system shows an overview of the markets containing market code, market name and currency.

2.4.3.2 Show list of stocks on market

Primary actor: Bank clerk or customer

Basic flow:

1. The actor selects to get an overview of the stocks on a certain market.
2. The bank system asks for the market code.
3. The actor gives the market code.
4. The bank system asks the stock market catalogue system for a list of stocks on the given market.
5. The catalogue system gives the bank system the requested list.
6. The bank system shows an overview of the stocks containing stock code (ISIN), company name, last sale, currency, absolute change, relative change and share volume.

2.4.3.3 Show stock information

Primary actor: Bank clerk or customer

Basic flow:

1. The actor selects to get an overview of a stock.
2. The bank system asks for the ISIN code of the stock.
3. The actor gives the ISIN code.
4. The bank system asks the stock market catalogue system for detailed data of the stock.
5. The catalogue system gives the bank system the requested list.
6. The bank system shows an overview of the stock containing stock code (ISIN), company name, last sale, currency, absolute change, relative change, share volume, best bid, best ask, today's high / low and the historical notations containing: the date, opening rate, highest rate, lowest rate, closing rate and share volume.

2.4.4 Stock market transactions**2.4.4.1 Place a direct order for a stock**

Primary actor: Bank clerk or customer

Supporting actors: Involved stock market.

1. The involved custody account has been selected.

Basic flow:

1. The actor selects to place a direct order on a stock.
2. The bank system asks to select between buy or sell.
3. The actor selects between buy or sell.
4. The bank system asks for the ISIN code of the stock.
5. The actor gives the ISIN code.
6. The bank system asks for the amount of stocks involved.

7. The actor gives the amount.
8. The bank system verifies the order.
9. The customer and bank sign the order.
10. The bank system creates a new pending order.
11. The bank system sends the order to the stock market.
12. The stock market gives the order a unique number.
13. The stock market associates the bank as agent for the order.
14. The bank system and the stock market sign the order.
15. The stock market confirms the receipt of the order.
16. The bank system adds the order to the pending orders of the custody account.
17. The bank system notifies the actor.

2.4.4.2 Place a limited order for a stock

Primary actor: Bank clerk or customer

Supporting actors: Involved stock market.

1. The requesting customer has been selected.
2. The involved custody account has been selected.

Basic flow:

1. The actor selects to place a limited order on a stock.
2. The bank system asks to select between buy or sell.
3. The actor selects between buy or sell.
4. The bank system asks for the ISIN code of the stock.
5. The actor gives the ISIN code.
6. The bank system asks for the amount of stocks involved.
7. The actor gives the amount.
8. The bank system asks for the limited price involved.
9. The actor gives the limited price.
10. The bank system verifies the order.
11. The customer and bank sign the order.
12. The bank system creates a new pending order.
13. The bank system sends the order to the stock market.
14. The stock market gives the order a unique number.
15. The stock market associates the bank as agent for the order.
16. The bank system and the stock market sign the order.
17. The stock market confirms the receipt of the order.
18. The bank system adds the order to the pending orders of the custody account.
19. The bank system notifies the actor.

2.4.4.3 Execute put option

Primary actor: Bank clerk or customer

Supporting actors: Involved issuer of the option.

1. The requesting customer has been selected.
2. The involved custody account has been selected.
3. The involved option has been selected.

Basic flow:

1. The actor selects to execute a put option
2. The bank system asks for the amount of options involved.
3. The actor gives the amount.
4. The bank system verifies the execution.
5. The customer and bank sign the execution.
6. The bank system sends the execution to the issuing financial institution.
7. The issuer verifies the execution.
8. The bank system and the issuer sign the execution.
9. The bank system sends an invoice to the issuer with the total amount to pay.
10. The bank system updates the custody account by marking the involved options as executed, updating the balance of the account with the value of the sold stocks and updating the amount of stocks in the associated position.

2.4.4.4 Execute call option

Primary actor: Bank clerk or customer

Supporting actors: Involved issuer of the option.

1. The requesting customer has been selected.
2. The involved custody account has been selected.
3. The involved option has been selected.

Basic flow:

1. The actor selects to execute a put option
2. The bank system asks for the amount of options involved.
3. The actor gives the amount.
4. The bank system verifies the execution.
5. The customer and bank sign the execution.
6. The bank system sends the execution to the issuing financial institution.
7. The issuer verifies the execution.

8. The issuer gives a unique number to the execution.
9. The bank system and the issuer sign the execution.
10. The bank system updates the custody account by marking the involved options as executed. In case no position has been opened for the stock related with the option, a new position is opened. The amount of stocks in the new or already existing position is raised with the amount of bought stocks.

2.4.5 Use cases for feedback from stock market system

2.4.5.1 Confirm a selling order

Primary actor: Stock market system

Basic flow:

1. The stock market system notifies the bank system that a certain selling order has been processed. The stock market gives the unique id of the order, the actual selling price and the unique id of the trading transaction. If the amount of stocks in the order could not be matched completely the stock market also provides the information of the new order that has been entered in the order book with the rest of the amount.
2. The bank system and the stock market sign the confirmation.
3. The bank system sends an invoice to the stock market with the total amount to pay.
4. The bank system updates the custody account by confirming the pending order, specifying the actual selling price, updating the balance of the account and updating the amount of stocks in the associated position.

2.4.5.2 Confirm a buying order

Primary actor: Stock market system

Basic flow:

1. The stock market system notifies the bank system that a certain buying order has been processed. The stock market gives the unique id of the initial order, the actual selling price and the unique id of the trading transaction. If the amount of stocks in the order could not be matched completely the stock market also provides the information of the new order that has been entered in the order book with the rest of the amount.
2. The bank system and the stock market sign the confirmation.
3. The bank system updates the custody account by confirming the pending order, specifying the actual buying price and updating the amount of stocks in the associated position.

2.4.5.3 Deny an order

Primary actor: Stock market system

Basic flow:

1. The stock market system notifies the bank system that a certain order has been rejected.
2. The bank system asks for the unique id of the order and the reason for rejection.
3. The stock market gives the id and the reason for rejection.
4. The bank system and the stock market sign the rejection.
5. The bank system updates the custody account by updating the pending order.

2.4.5.4 Send an invoice for a trading transaction

Primary actor: Stock market system (or settlement organization) or issuer of an option.

Stakeholders: Customer, bank system, inter banking infrastructure.

Preconditions: The stock market has confirmed a buying order or a call option has been executed.

Basic flow:

1. The actor sends an invoice to the bank for a trading transaction in which the bank was a buying agent. This invoice contains the total amount to pay (buying price of the traded stocks and handling costs). The invoice also contains an invoice number, the unique number of the trading transaction and the account number of the actor.
2. The bank system and actor sign the invoice.
3. The bank system creates a charging transaction on the custody account of the customer with the total amount to pay and its own handling costs. This charging transaction is sent to the transaction handler.
4. The bank system requests a transfer from the bank's own account to the account of the actor (the stock market or the settlement organization). The amount on this transfer is what the bank has to pay to the stock market for the trading transaction. This transfer contains a message that refers to the invoice number.

2.5 Use cases for transaction processing

The transaction handler is a system that processes the transaction queue. All transactions (withdrawals, deposits, transfers and cost charges) created by the different users are stored in a transaction queue. Afterwards (mostly at night) this queue is processed in batch by the transaction handler. This system includes the following uses cases.

2.5.1 Use cases for operational management

2.5.1.1 Start handling transaction queue

Primary actor: Operational manager.

Stakeholders: Bank system.

Basic flow:

1. The operational manager starts the transaction handler
2. The transaction handler iterates the transaction queue and executes each transaction (see following use cases for details).

2.5.2 Use cases for processing a transaction

2.5.2.1 Process a deposit

Primary actor: Transaction handler.

Stakeholders: Bank system.

Preconditions: The transaction handler is processing the transaction queue

Basic flow:

1. The transaction handler starts the processing of a deposit transaction.
2. The bank system adds the amount to the account.
3. *The bank system updates the bank's internal accountancy.* (To be investigated)
4. An entry with the transaction information is added to the overview of the completed transactions of the customers account.
5. The transaction is removed from the transaction queue.

2.5.2.2 Process a withdrawal

Primary actor: Transaction handler.

Stakeholders: Bank system.

Preconditions: The transaction handler is processing the transaction queue

Basic flow:

1. The transaction handler starts the processing of a withdrawal transaction.
2. The bank system withdraws the amount from the account.
3. *The bank system updates the bank's internal accountancy.* (To be investigated)
4. An entry with the transaction information is added to the overview of the completed transactions of the customers account.
5. The transaction is removed from the transaction queue.

2.5.2.3 Process a transfer

Primary actor: Transaction handler.

Stakeholders: Bank system, inter banking infrastructure.

Preconditions: The transaction handler is processing the transaction queue

Basic flow:

1. The transaction handler starts the processing of a transfer transaction.
2. The bank system withdraws the transfer amount from the originating account.
3. The bank system requests a transfer from the bank's own account to the account of the bank of the destination account. This transfer contains a message that contains the details of the transaction: the originating account from the customer, the destination account from the other customer and a transfer message.
4. *The bank system updates the bank's internal accountancy.* (To be investigated)
5. An entry with the transaction information is added to the overview of the completed transactions of the customers account.
6. The transaction is removed from the transaction queue.

2.5.2.4 Process a charge

Primary actor: Transaction handler.

Stakeholders: Bank system.

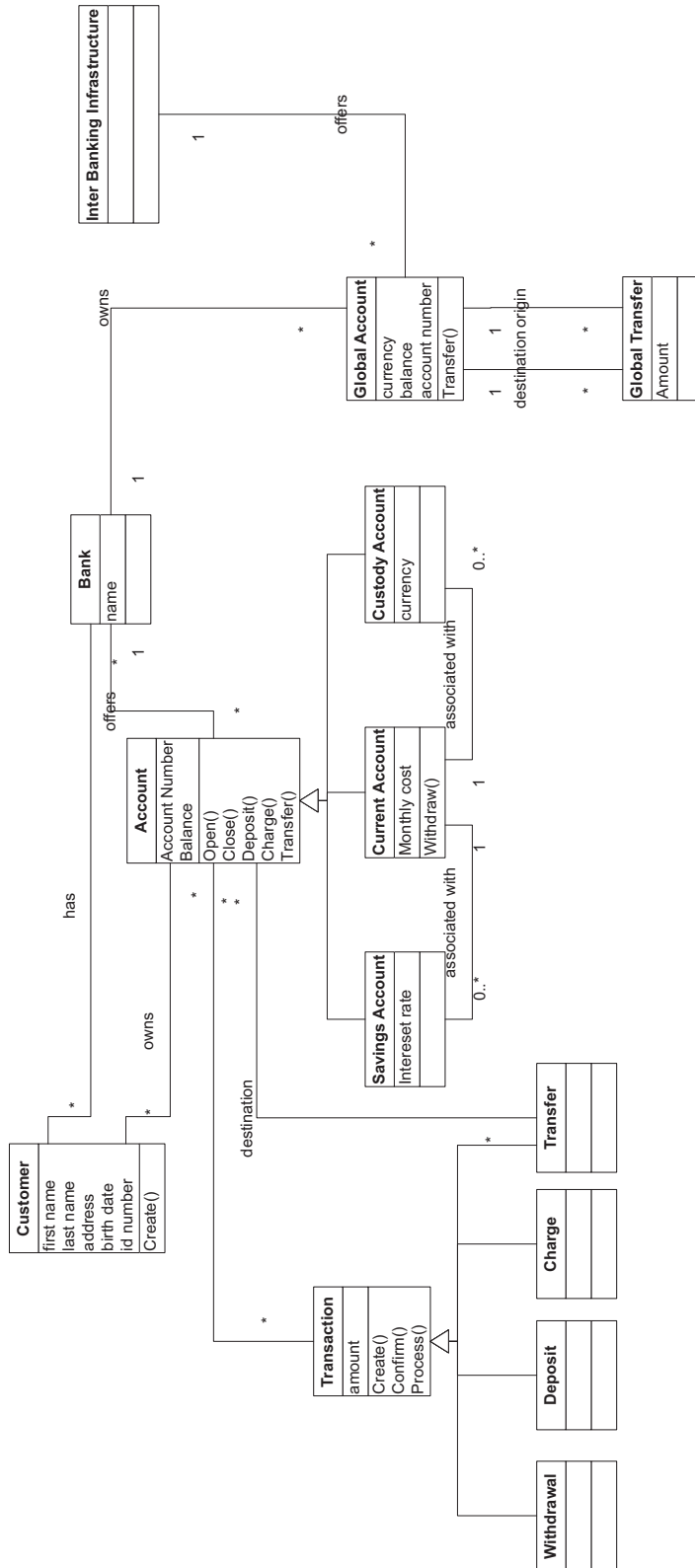
Preconditions: The transaction handler is processing the transaction queue

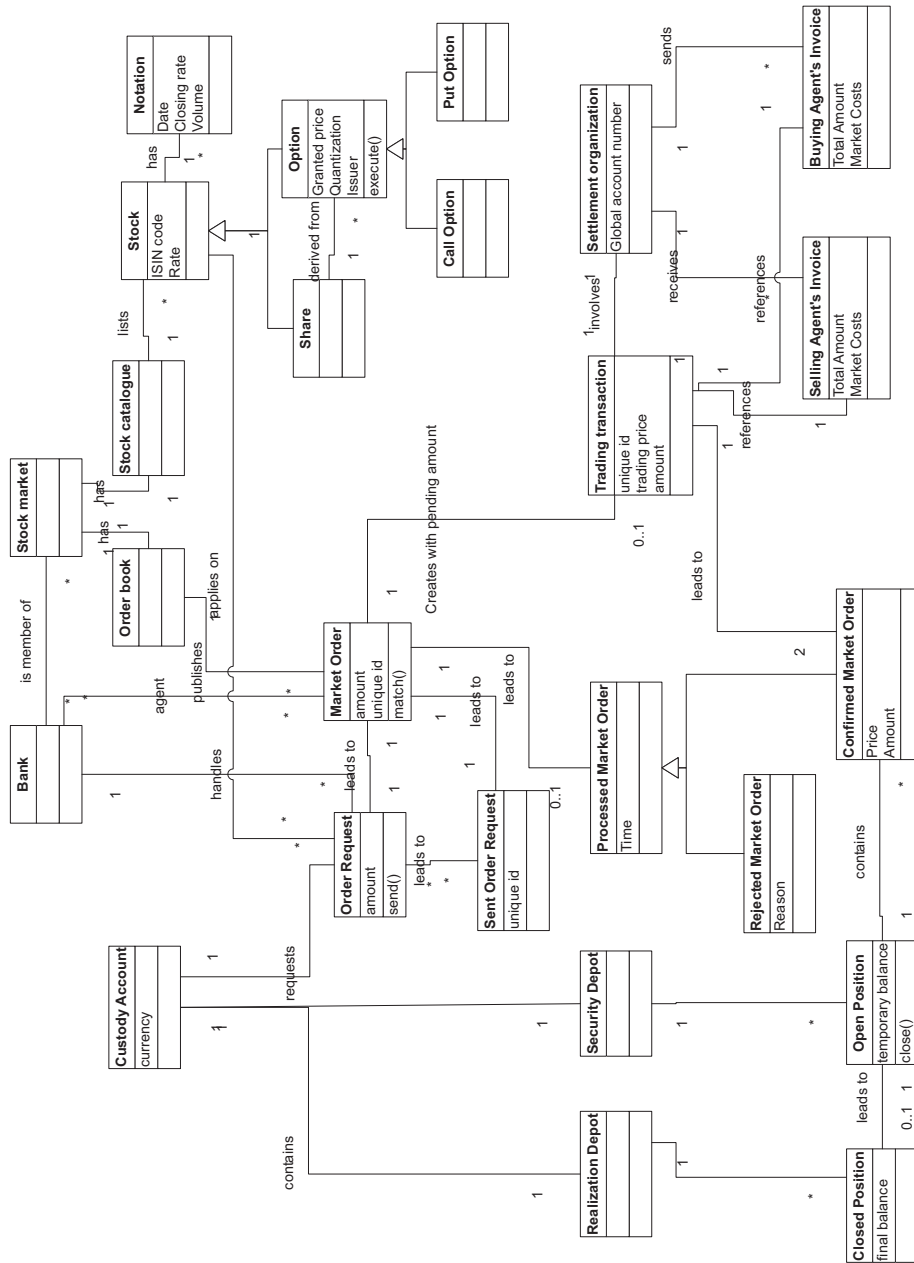
Basic flow:

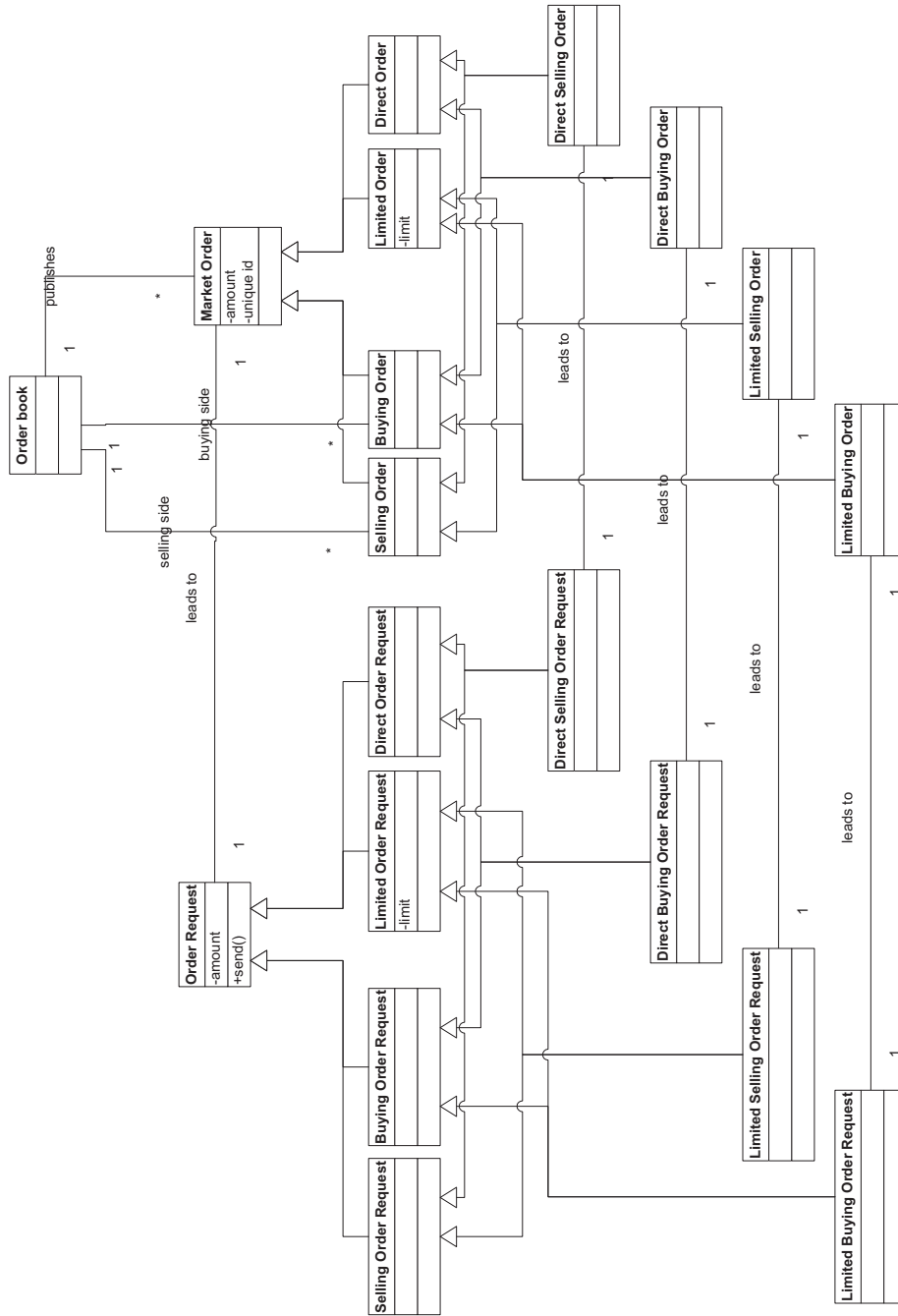
1. The transaction handler starts the processing of a charge transaction.
2. The bank system withdraws the amount from the account.
3. *The bank system updates the bank's internal accountancy.* (To be investigated)
4. An entry with the transaction information is added to the overview of the completed transactions of the customers account.
5. The transaction is removed from the transaction queue.

2.6 Domain model

The following three models describe the problem domain. The first model represents an overview of basic banking services. The second models describes the investment service more in detail. The third model zooms in on the multiple inheritance structure that describes order requests and orders.







Chapter 3

Non functional requirements : quality attributes

We will use quality attribute scenarios to characterize the desired quality attributes of the system.

3.1 Availability

3.1.1 General quality scenarios

1. **Source:** Internal.

Stimulus: Zero downtime of all banking services.

Artifact: Bank system

Environment: Runtime.

Response: Downtime shall be avoided. The bank system is deployed on multiple servers with passive redundancy. The usage of transactional operations and stateless components between transactions grants consistency between redundant servers.

Response measure: virtual zero downtime.

2. **Source:** External.

Stimulus: Failure of external systems.

Artifact: External Communication

Environment: Runtime.

Response: Pre-emptive caching of important external data and gradual degradation of banking services relative to external systems failure.

Response measure: virtual zero downtime with degraded services.

3.1.2 Specific quality scenarios

1. **Source:** Internal.

Stimulus: Bank system application server crash.

Artifact: Bank system application server.

Environment: runtime.

Response: System administrator is informed to investigate cause. The redundant application server takes over. Transactional execution of operations ensures consistency in the data.

Response measure: virtual zero downtime.

2. **Source:** Internal.

Stimulus: Data server unreachable for application server

Artifact: Bank system data server.

Environment: runtime.

Response: The system administrator is informed to investigate cause. The redundant data server takes over. Transactional execution of operations ensures consistency in the data.

Response measure: virtual zero downtime.

3. **Source:** External.

Stimulus: Some stock market is unreachable

Artifact: Investment services.

Environment: runtime.

Response: The system administrator is informed to investigate cause. The bank system goes to degraded mode on investment services based on the unreachable stock market. Inspection of local data is possible. Cached information on stocks is clearly timestamped. Order requests can be made. Sending the orders to the market will be post-poned until the market is reachable.

Response measure:

3.2 Performance

This requirement is taken in account because it interferes with the security requirements. It will be worked out later.

3.3 Usability

An important usability requirement related to security is auto-adaptation of the user interface according to the possible actions allowed for the user. This software architecture document focuses on the functional (business) layer of software for retail banking services. The usability requirement requirement is about the presentation layer, so we do not design this in detail. Though, it is an interesting problem and we will shortly sketch a possible strategy to achieve this requirement.

3.4 Security

Security requirements are discussed in a separate chapter after this one.

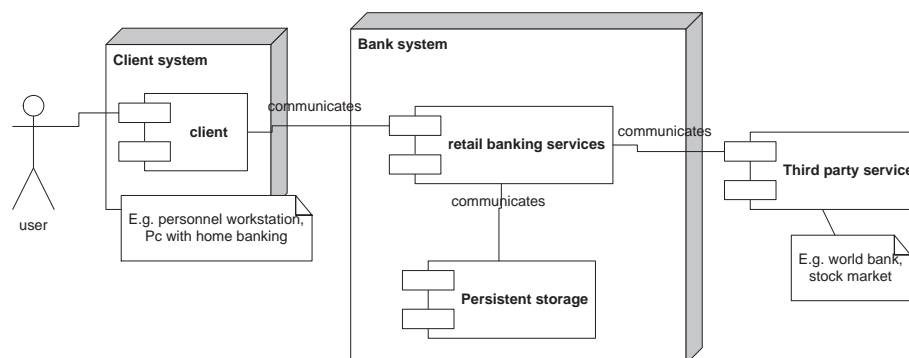
Chapter 4

Security Requirements

Security requirements describe the systems ability to resist unauthorized usage while still providing its services to legitimate users. The security requirements are divided into five categories.

- Non repudiation and accountability: protection against falsely denial of participation in communication or certain actions.
- Access control: protection against non-authorized access to functionality provided by the system.
- Integrity: protection against non-authorized creation, altering or removal of information.
- Authentication: protection against masquerading.
- Confidentiality: protection against non-authorized access to information.

In the next figure we sketch a high level system view as a basis to illustrate the different security requirements.



4.1 Non repudiation and accountability

Accountability is the requirement that a certain operation or event cannot be denied by one or more of the parties to it. Accountability ensures that a person

can be held responsible for his actions. Non repudiation also delivers cryptographic proof of those actions. The legal value of cryptographic proof depends on a lot of constraints that we will discuss shortly when we elaborate on the security architecture.

In general, the requirement of accountability with non repudiation applies to operations such as access to a certain retail banking service as well as access, deletion, modification and creation of data in the persistent storage. Within the banking system this requirement applies to a lot of scenarios and involves different parties that have to commit undeniably.

A first part are the requests of the customers for a certain service of the bank that only involve the customer and the bank. Both the customer and the bank need to confirm undeniably the request for this service. The customer needs to confirm that he requested the service and the bank needs to confirm it received this request. For the customer, his confirmation also includes that the service has been provided to him. Typical scenarios where this applies include:

- The customer requests a certain service like a current account, savings account or custody account.
- The customer requests to do an internal financial transaction at the bank like a transfer between his own account and another account at that bank, a withdrawal of his own account or a deposit on his own account.

A second part are the requests of the customers for a certain service that involves a third party like another bank or a stock market.

The customer needs to confirm that he requested the service and the bank needs to confirm it received this request. The bank now also needs the proof that the third party has been involved, in case a dispute about the completeness of the transaction arises. So there has to be non repudiation of origin from the customer towards the bank and from the bank towards the third party. There has to be non repudiation of receipt from the bank towards the customer and from the third party towards the bank.

- The customer requests to perform a money transfer to an account on another bank. The requesting bank needs a proof of the inter banking infrastructure that the requesting bank has received a request to transfer the money to the destination bank, in favor of the customer's destination account (non repudiation of receipt). The requesting bank will also need to confirm undeniably toward the global banking organization that it requested the transfer (non repudiation of origin).
- The customer requests to do a market order. He needs to confirm this, and the bank needs to confirm it received this request. The customer and the bank need the proof that the order has been sent to the market. The market also needs the proof that the bank has requested the order on the market.

4.2 Access control

The access control requirements for the different operations in the banking system include requirements first based on the role of the user and secondly based

on owners of the object the operation applies on. We discuss the specific requirements for the system concerning role based access control and owner based access control separately in the respective sections.

Two important principles in the access control policy are separation of duty and the principle of least privilege. Separation of duty, as a security principle, has as its primary objective the prevention of fraud and errors. This objective is achieved by disseminating the tasks and associated privileges for a specific business process among multiple users. For example, creating an account should be confirmed by the bank clerk and the customer. The principle of least privilege states that a user should have only the authority it needs to accomplish his task. For example, the system administrator should not be given access to the account information of the customers.

4.2.1 Role Based Access Control

Within the banking system we distinguish the following roles of importance within the scope of this case study:

- Bank clerk
- Customer
- Operational manager
- Stock market

Bank clerk The role of bank clerk applies to the personnel of the bank. They operate on the banking system from the branch offices. A first set of operations a user can do within this role are the inspection operations:

- Inspecting customer information
- Inspecting information about all accounts of all customers. This includes the balance, the transaction log, pending market orders, realization depot, etc. . .
- Inspecting all stock market information

A next set of operations a bank clerk can do are the operations that provide a service towards the customer, and with which the customer has to agree.

- basic banking administrative operations (creating accounts, editing customer information, ...)
- basic banking financial transactions (money transfers , order requests for the stock market, ...).
- custody account administrative operations
- custody account transactions
- depot management
- stock market transactions

Next to the fact that the user needs to be in the role of bank clerk to do these operations, also proof is needed that the customer has requested this operation. Otherwise a bank clerk should not be able to do this operation. In case of a financial transaction the requesting customer also needs to own the account the financial transaction applies on (see owner based access control).

A bank clerk can also block the accounts of a customer, but only if there is proof that the office manager has requested this operation.

Operational management is not allowed for bank clerks.

Customer The role of customer applies to the customers of the bank. As a user of the banking system, they are operating on the system through home banking. Customers can not do administrative operations. They can do financial transactions on their own accounts (see owner based access control). Inspection operations are also only allowed on their own accounts. They can inspect all stock market information.

Operational manager The operational manager is responsible for the operational management of the banking system. He can add or remove users and assign roles to users. It is not possible for him to do customer management, account management or financial transactions.

Stock market The stock markets are external systems for stock trading known by the bank. Only these external systems are allowed to the operations for feedback about orders. An extra requirement is that the stock market that wants to give feedback about a certain order also has to be the stock market that is handling that order. So also here applies owner based access control.

4.2.2 Owner based access control

Although role based access control may seem to offer the main classification of the access control requirements, also owner based access control is of crucial importance. As mentioned above, for the role of customer and stock market, owner based access control applies to all operations on their own assets.

This means that a customer can only inspect information about his own accounts and can only do financial transactions with his own accounts. For the stock market owner based access control applies this in a dual way. The stock market can only provide feedback about an order to a bank if the stock market is handling that order and if that bank placed that order. So the stock market has to own the order, and the involved bank has to own the order request for it.

Normally, a customer and his accounts are bound to a certain branch office of the bank. Some operations as creating a new account or closing an account can only be done at that specific office. This is also a form of owner based access control. In this case study we assume that the bank wants to evolve to one virtual office accessible through home banking and self-banking terminals. Today, banks also decrease the number of branch offices. To avoid administrative overload for migrating customers and accounts in case of closure of a branch office, this form of access control for account management will not be applied in this case study.

4.3 Data integrity

Data integrity is a requirement which addresses the unauthorized alteration of data. To assure data integrity, one must have the ability to detect data manipulation by unauthorized parties. Data manipulation includes such things as insertion, deletion, and substitution.

Although Access control also protects against data manipulation, it is mainly applied at service level. The data integrity requirement also includes the ability to detect manipulation without using the service, but by directly attacking the data. This data integrity is a requirement that applies to two kinds of data: the information involved in communication between a user and the bank system and the data in the persistent storage of the bank system.

4.4 Authentication

Authentication is a requirement related to identification in a certain communication (or transaction) between two entities. Normally, this requirement applies to both entities and the information involved in the transaction. Two parties entering into a transaction should identify each other. Information delivered over the channel used for the transaction should be authenticated as to origin, data content, time sent, etc. So the authentication requirements for this case study can be subdivided into two major classes: entity authentication and data authentication.

4.4.1 Entity authentication

As mentioned above, when a transaction between a user and the bank system takes place, both entities should identify each other. For usability reasons, we want to achieve a single sign on system. Whenever a user wants to use the system, he should authenticate himself at the start of his session. This initial authentication process should be sufficient for the rest of his session. In the other direction, the bank system will also identify itself so the user is certain that he is communicating with the bank system. This requirement applies to all possible communication in the system between a (remote) client and the main bank system. These remote clients could be: customers using home banking software, customers at self-banking terminals, personnel at workstations in the branch offices and the stock market systems.

4.4.2 Data authentication

The information involved in a transaction between a user and the bank system should also be authenticated to its origin. Data origin authentication grants that the information is coming from whom it claims to be coming from. This requirement directly includes data integrity of the information involved in a transaction.

4.5 Privacy and confidentiality

This requirement states that information should be kept secret from all but those who are authorized to see it. This is partially covered in the requirement about access control. But this requirement also includes that the information involved in a transaction between the user and the bank should be kept secret while it is on the channel used for the communication. As a matter of fact, all information can be transported over a public untrusted channel: Home banking can use the Internet, branch offices can be connected through a leased line, stock markets also could use the Internet. Self-banking terminals can use the telephone network.

So the policy about confidentiality states that all information transported over a public network should be kept confidential to the parties not involved in the transaction.

Chapter 5

Conclusion

In this document we have described the requirements and analysis of a case study in the world of e-finance. Functional requirements and non functional requirements of the following retail banking services have been described : basic banking services and more advanced retail banking services concerning private investments. We also gave a detailed description of the problem domain. A next document will describe the software architecture.