Object Oriented Systems – Assignment 1

Contents

[2 Task 1](#_Toc446880187)

[2.1 Functionality required 1](#_Toc446880188)

[2.2 Overall Key Requirements 1](#_Toc446880189)

[2.3 Server/Client requirements 1](#_Toc446880190)

[Approach 2](#_Toc446880191)

[3 The Protocol 2](#_Toc446880192)

[3.1 Authentication Process 2](#_Toc446880193)

[4 How transfers work 3](#_Toc446880194)

[4.1 How Local Transfers Work 3](#_Toc446880195)

[4.1.1 What is a local transfer 3](#_Toc446880196)

[4.2 How remote transfers work 3](#_Toc446880197)

[4.2.1 What is a remote transfer 3](#_Toc446880198)

[4.2.2 Behind the scenes 3](#_Toc446880199)

[5 Implementation issues 4](#_Toc446880200)

[6 Why I decided to make a protocol for my assignment 4](#_Toc446880201)

[7 What would I change? 4](#_Toc446880202)

[8 The overdraft system 4](#_Toc446880203)

[8.1 Entering negative balance 4](#_Toc446880204)

[8.2 Daily overdraft check 4](#_Toc446880205)

[8.3 Interest rate check 5](#_Toc446880206)

[9 Youtube videos 5](#_Toc446880207)

# Task

The task is to create a banking system using Java. The banking system should allow one or more bank accounts to be created for the same customer. These bank accounts can be different types such as a current account or a savings account. The bank accounts should be uniquely identified using an account number that consists of eight digits 0-9 and should be formatted as “XXXXXXXX” and a sort code that consists of six digits 0-9 and should be formatted as “XX-XX-XX”.

## Functionality required

* The bank accounts should be able to be deposited into and withdrawn from.
* The user should be able to check the balances of the bank accounts.
* Overdrafts should be enabled for certain types of bank accounts.
* Interest rates should be set for certain types of bank accounts.

The bank system should be able to do the following account operations:

* Ability to browse bank accounts using a menu system
* Deposit and withdraw money from bank accounts
* Set overdraft limits on current accounts
* Apply a default bank charge when a current account exceeds the overdraft limit
* Apply an interest rate on a savings account
* Generate a statement of account transactions that can be saved to a file

## Overall Key Requirements

* The bank system should work on a client and server basis where the user will use their client to talk to the bank server to do any of the functionality in the [Functionality Required](#_Functionality_required) and any functionality stated elsewhere in this document.
* The bank system should not be limited to one server.
* The bank system should have the ability to connect to other banks that use this banking system so that the connector may perform transfers.

## Server/Client requirements

* Every server client must run on its own thread
* Every non-server client must not use a thread due to fear of threading issues with the socket

# Approach

The approach is to create a banking protocol called the BTP(Bank Transmission Protocol) that will allow my banking server and banking client to easily communicate between each other and other banks using the protocol. The BTP(Bank Transmission Protocol) needs to completely run in the background with very little need for interaction with the programmer. The BTP(Bank Transmission Protocol) will have an event handler that will contain events such as “customerLogin” and “getBalance” and other methods like this. These methods will get called by the server automatically when the server receives commands relative to these events. Should there be a problem with these requests maybe perhaps the programmer wishes to deny a customer login request as the credentials were incorrect the programmer simply throws an exception. This exception gets fed back to the client through the BTP(Bank transmission protocol).

The banking server also uses SQLite which is a decent database library to allow you to make a database file and query it just like any other database. Although SQLite is not as powerful as some of the other database engines that exist SQLite is easy to use and simple but also powerful enough to get the job done.

# The Protocol

## Authentication Process

The authentication process is the first thing that happens when connecting to a BTP(Bank Transmission Protocol) Server. The client will pass a decimal value as displayed in the table below. This decimal value will correspond to the type of login

Below is a table of all the three possible authentication types.

|  |  |  |
| --- | --- | --- |
| **Value** | **Type** | **Access** |
| Decimal 0 | Customer Login | All bank accounts for the customer in question |
| Decimal 1 | Employee Login | All bank accounts in the entire database |
| Decimal 2 | Transfer Login | Transferring money into a bank account |

The authentication type that is selected will determine how the client must authenticate its self.

# How transfers work

## How Local Transfers Work

### What is a local transfer

In the BTP(Bank Transmission Protocol) a local transfer is a transfer to the bank that the customer transferring the money is a part of. Let’s say you have two people let’s call them Mary and Jane. Now Mary and Jane are both part of Lloyds bank. If Mary wishes to send money to Jane’s bank account this would result in a local transfer as they are both part of Lloyds bank. The money does not need to leave the bank therefore it is a local transfer

## How remote transfers work

### What is a remote transfer

Remote transfers are a bit more complicated than a local transfer. A remote transfer is where the person you are sending money to is a part of a different bank therefore their bank account is based on another server.

### Behind the scenes

When the programmer sets up there bank server they add any banks they wish to communicate with to their trusted bank list. This list contains information such as the banks sort code and IP address/hostname and port. The list also contains a very vital part of the security and that is an authorization code. This code is set by the programmer who then sends this same authorization code to the bank they have trusted. The programmer for this other bank server can then do the same thing as the other programmer did and will use the same authorization code created by him.

Both banks now agree with each other and any of their customers can send money between each other.

Once Mary whose bank account is registered at Lloyds bank attempts to send money to Sarah whose bank account is at HSBC an interesting thing happens. Mary’s bank client will contact Lloyds Bank’s server to initiate a transfer. Now Lloyds Bank will notice the sort code is not of its own bank and will search its trusted bank’s list. Once it finds HSBC’s sort code it will then send its own sort code and authorization key to a HSBC server. This is the transfer authorization type. This will invoke the “transferLogin” event on HSBC’s sever. Since Lloyds Bank is in the trusted banks list of HSBC and they both share the same authorization code HSBC should allow authorize Lloyds Bank’s transfer client.

Once authorized Lloyds bank can now send funds to Sarah’s bank account. Should something go wrong HSBC will simply throw an exception and it will be thrown back to Lloyds Bank’s server which will then throw it back to Mary. If everything went right then the funds are withdrawn from Mary’s bank account and HSBC will deposit the funds it was notified about into Sarah’s bank account. A transaction is then made for safe keeping.

# Implementation issues

It took me three weeks to design the entire protocol. Unfortunately although I knew the sort of design pattern I wished to follow the class diagram had to be updated quite regularly. Mistakes were made I had to modify my table in my database recently as I got the columns wrong and was missing one. Upon creating the table I was not thinking that transactions would have had to be logged for remote transfers therefore the foreign key that I had in place would have not worked as expected.

# Why I decided to make a protocol for my assignment

I decided to make a protocol for my assignment because a typical bank system would have taken me 1-2 hours and I would have probably got a grade of 50. I didn’t want this project to be easy and boring so I thought it would be much more fun to design my own protocol and event handler and spend a lot of time planning and designing the system. The reason for using a database is because in my personal opinion it is much easier than parsing a text file. Data can get complicated and you want a structured system and this is why database engines are very important.

# What would I change?

If I had more time I would have wrote my own stream classes that would send raw data rather than sending integers as strings and decoding them on the other end. I believe for efficiency this would be important although internet speeds are getting faster and it wouldn’t really be a problem I believe that it is better practice. The reason I didn’t do this originally is because of all the assignments being quite close together it was not worth the risk even though something like this wouldn’t be too hard to implement, going back and changing it all would be time consuming.

# The overdraft system

## Entering negative balance

Upon entering negative balance the system will withdraw funds from the bank’s bank account this is very important otherwise you are essentially generating fake money. Every bank has its own bank account whose balance should be very high as should the bank’s bank account run out of money the system will not work as expected. Using this setup it means that money cannot be created but only moved. This was an important design decision as if real banks just generated money from nowhere I can imagine the British sterling being worth very little.

## Daily overdraft check

Every 24 hours the system will loop through every bank account in the system and the ones that are overdrawn and have breached their overdraft limit will be charged a percentage based on their bank account type.

## Interest rate check

Every 48 hours the system will loop through every bank account in the system and the accounts that are savings accounts will be given money based on a percentage specified in their bank account type.

# Youtube videos

I filmed myself programming the entire project and the YouTube playlist Is here: <https://www.youtube.com/watch?v=iKX2iqSFJKE&list=PLDS_sVpO_inaI4wH0rKZRFNtDPc_MWFvT>

As of 27/03/2016 not all of the videos have yet been uploaded and will be uploaded shortly.