|  |
| --- |
|  |
| Architectuurdocument |
| Luc Martens, Remi Arts & Simon van Amstel |
|  |
|  |
|  |

|  |
| --- |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Versie | Wijzigingen | Auteur | Datum | Verzendlijst | Verzoek |
| 1 |  | Simon v. Amstel, Luc Martens & Remi Arts | 26-11-2012 | S. Angelov | g |
| 2 | Feedback verwerkt | Simon v. Amstel, Luc Martens & Remi Arts | 10-12-2012 | S. Angelov | g |
|  |  |  |  |  |  |

Contents

[H1 Inleiding 3](#_Toc342898095)

[H2 Domeinmodel 4](#_Toc342898096)

[Klassendiagram 4](#_Toc342898097)

[Afbakening 5](#_Toc342898098)

[Relaties 5](#_Toc342898099)

[H3 Persistentie 6](#_Toc342898100)

[Eigenschappen van objecten m.b.t. persistentie 6](#_Toc342898101)

[Date opslag na een verandering 6](#_Toc342898102)

[H4 Communicatie 7](#_Toc342898103)

[Eigenschappen van objecten M.B.T. communicatie 7](#_Toc342898104)

[Dataverkeer tijdens een overschrijving 7](#_Toc342898105)

[H5 Realisatie niet-functionele eisen 8](#_Toc342898106)

[Betrouwbaarheid 8](#_Toc342898107)

[Gebruiksvriendelijkheid 8](#_Toc342898108)

[Beveiliging 8](#_Toc342898109)

[Performance 8](#_Toc342898110)

[H6 Componenten 9](#_Toc342898111)

[Componentendiagram 9](#_Toc342898112)

[Koppeling tussen componenten 9](#_Toc342898113)

[Services per component 10](#_Toc342898114)

[Allocatie van objecten 10](#_Toc342898115)

[Remote objecten 11](#_Toc342898116)

[Packagestructuur 11](#_Toc342898117)

[H7 Deployment 12](#_Toc342898118)

[Deploymentdiagram 12](#_Toc342898119)

[H8 Specificatie van interfaces 13](#_Toc342898120)

# H1 Inleiding

We beperken ons tot het overboeken van de ene naar de andere bankrekening van mogelijk verschillende banken (zie ook Bijlage 1). We gaan er vanuit dat de te ontwikkelen programmatuur voor alle banken eender werkt, zodat de vereiste software eenmalig hoeft worden ontwikkeld.

Alle banken zijn bij één overboekcentrale bekend. Elke bankrekening heeft een nummer (maximaal 10 cijfers), een eigenaar (naam en woonplaats) en een saldo (in euro’s). Elke bankrekening heeft een variabel kredietlimiet dat bij opening van de rekening 100 euro’s zal bedragen.

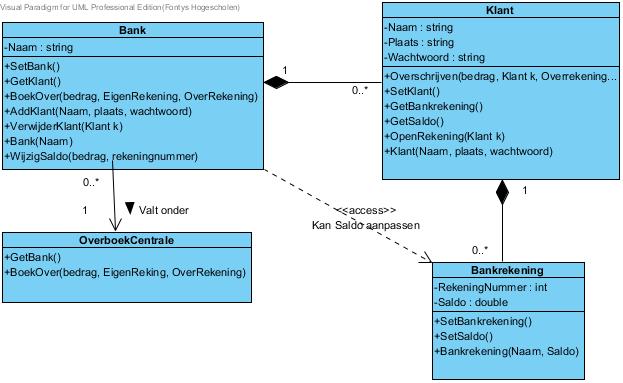
Een eigenaar moet zich via het internet identificeren alvorens hij (of zij) zijn saldo kan inzien en bedragen kan overmaken naar een andere rekening. De inloggegevens bestaan uit de combinatie van een inlogaccountcode en bijbehorend wachtwoord. Alvorens te kunnen inloggen moet er eerst een bankrekening worden geopend. Dat geschiedt door middel van de invoer van de naam en woonplaats in combinatie met de vastlegging van een wachtwoord. Binnen dit bestek gaan we uit van de (weinig realistische) aanname dat de combinatie van naam en woonplaats identificerend voor een klant is.

We gaan verder voor de eenvoud uit van de situatie dat elke bankrekening over precies één inlog-account beschikt. Aan gezamenlijke bankrekeningen (“en/of rekening”) geven we binnen dit bestek geen aandacht. Dus elke bankrekening heeft precies één eigenaar. Wel is het toegestaan dat een klant over meer dan 1 bankrekening beschikt. Een een bankrekening kan op een zeker moment vanuit verschillende computers worden ingezien. Je kunt je hierbij voorstellen dat je je inloggegevens hebt bekend gemaakt aan bijvoorbeeld je levenspartner, die daarmee dezelfde rechten heeft gekregen als jij.

# H2 Domeinmodel

In dit hoofdstuk worden een of meer klassendiagrammen van het domein getoond. Daarnaast wordt een toelichting gegeven en worden gemaakte keuzes verantwoord. Tot slot wordt de afbakening van het project besproken.

## Klassendiagram



Figuur 1: Klassendiagram van het domein.

In figuur 1 is het klassendiagram van het domein afgebeeld. Hierin worden de volgende klassen onderscheidden:

* Bank: Representeert een bank. Een bank kan klanten toevoegen, verwijderen van klanten en het saldo op bankrekeningen aanpassen. Een bank heeft een naam om mee geïdentificeerd te worden.
* Overboek Centrale: Representeert een overboek centrale. De centrale kan banken ophalen en stuurt de overboekingen door.
* Bankrekening: Representeert een bankrekening. De rekening heeft een nummer en een saldo.
* Klant: Representeert een klant. Een klant heeft een naam, plaats en wachtwoord. De klant kan een overschrijving maken en een nieuwe rekening openen.

## Afbakening

Het volledig ontwerp wordt niet in dit document besproken. Alleen de interfaces ten behoeve van interoperabiliteit worden besproken.

## Relaties

* Bank en klant: Een bank heeft meerdere klanten. De bank kan klanten toevoegen en verwijderen.
* Bank en bankrekening: Een bank kan het saldo aanpassen van een bankrekening.
* Klant en bankrekening: Een klant kan één of meerdere bankrekeningen hebben. Hij kan rekeningen aanmaken en verwijderen.
* OverboekCentrale en bank: De overboek centrale heeft meerdere banken. Bij overschrijvingen van één bank naar een andere bank stuurt de centrale de informatie door.

# H3 Persistentie

In dit hoofdstuk wordt de persistentie van de objecten beschreven.

## Eigenschappen van objecten m.b.t. persistentie

* Klant: Naam, plaats, wachtwoord, alle rekeningen.
* Bank: Naam, alle klanten.
* Bankrekening: rekeningnummer, saldo.
* Overboek Centrale: alle banken(ID, naam)

## Date opslag na een verandering

Gegevens van de actoren(Naam, plaats en wachtwoord) worden opgeslagen in de database. Voor deze oplossing is gekozen zodat deze gegevens gecombineerd kunnen worden met andere gegevens zoals adres en telefoonnummer van actoren en die buiten de applicatie om kunnen worden gebruikt en aangepast. Bij een overschrijving word de data na de bevestiging aangepast.

# H4 Communicatie

In dit hoofdstuk wordt de communicatie van en tussen objecten beschreven. Hierbij wordt aangegeven welke attributen van welke objecten moeten worden doorgegeven. Hoe de communicatie wordt gerealiseerd. En wanneer de communicatie plaatsvindt. Hoe en wanneer er communicatie plaats vind kan verschillen per attribuut.

## Eigenschappen van objecten M.B.T. communicatie

* Klant: klantnummer, naam, plaats, wachtwoord
* Bankrekening: rekeningnummer, saldo, klantnummer
* Bank: Id, Naam

## Dataverkeer tijdens een overschrijving

Tijdens een overschrijving wordt er klant en bankrekening gegevens gecommuniceerd. Omdat de communicatie real-time moet geschieden is gekozen RMI. Voor overschrijvingen wordt het push-principe toegepast. Voor RMI wordt de standaardport 1099 gebruikt.

# H5 Realisatie niet-functionele eisen

In dit hoofdstuk wordt de realisatie van niet-functionele eisen besproken. De volgende onderwerpen dienen te worden besproken: Betrouwbaarheid, gebruiksvriendelijkheid, beveiliging en performance.

## Betrouwbaarheid

Het is zeer belangrijk dat de applicatie betrouwbaar is. Tijdens het doen van een overschrijving mogen er geen fouten plaats vinden. Bij fouten zullen ook bijhorende fout-berichten komen. Om fouten tijdens het overschrijven te vermijden zullen altijd tegelijk in de database de gegevens veranderd worden. Wanneer er tegelijk de data aangepast word en er een fout komt, da verliest niemand geld.

## Gebruiksvriendelijkheid

Een bankrekening kan vanuit verschillende computers worden ingezien. Een bankrekening heeft een variabel kredietlimiet. Een klant kan meerdere bankrekeningen hebben. Er is een aparte applicatie voor op telefoons. De klant kan altijd zijn gegevens online aanpassen.

## Beveiliging

Om te kunnen inloggen heeft een gebruiker de accountcode met bijhorend wachtwoord nodig. Als je langer dan 10 minuten geen actie onderneemt wordt de banksessie automatisch beëindigd. De communicatie tussen de systemen worden door RMI afgehandeld, zodat de client geen directe communicatie heeft met de database. Hierdoor is het moeilijker om in te breken op de database en klantgegevens te achterhalen.

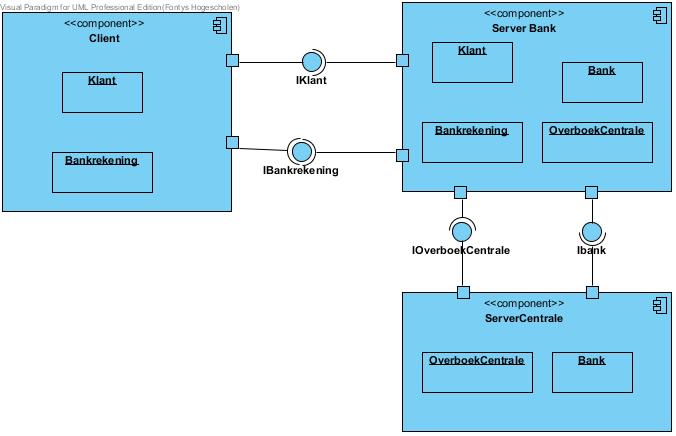
## Performance

Bij deze applicatie is performance niet belangrijk. De applicatie vraagt niet veel van een systeem en zal meestal vloeiend kunnen draaien.

# H6 Componenten

In dit hoofdstuk wordt de opdeling in software componenten besproken aan de hand van een componentendiagram met toelichting. Daarnaast wordt de koppeling en synchronisatie tussen de componenten besproken en de opdeling in software packages.

## Componentendiagram



Figuur 2: Componenten diagram.

Het componenten diagram in figuur 2 definieert 3 componenten met interfaces. De volgende componenten worden onderscheiden:

* Client. Dit component representeert de software applicatie voor de klant.
* ServerBank. Dit component representeert de software applicatie voor de bank.
* ServerCentrale. Dit component representeert de software applicatie voor de Overboek Centrale.

## Koppeling tussen componenten

Voor de onderlinge real-time communicatie wordt Remote Method Invocation(RMI) gebruikt. De koppeling tussen de componenten wordt gerealiseerd middels 4 interfaces, te weten IBank, IKlant, IBankrekening en IOverboekCentrale. Omdat alle componenten informatie nodig hebben van alle andere componenten zijn alle aangeboden interfaces verbonden met alle bijhorende vragende interfaces.

## Services per component

Services worden gerealiseerd middels 4 interfaces, te weten IBank, IKlant, IBankrekening en IOverboekCentrale:

IOverboekCentrale:

* Vraagt klantgegevens.
* Vraagt bankgegevens.
* Verstuurt een nieuw saldo.

IBank:

* Vraagt klantgegevens aan klant.
* Verstuurt klantgegevens.

IKlant:

* Vraagt Bankrekeninggegevens bijbehorende Bankrekening.

IBankrekening:

* Verstuurt saldo naar bijbehorende Klant.

De volgende services worden per component aangeboden:

* Client: IKlant
* ServerBank: IBankRekening, IBank
* ServerCentrale: IOverBoekCentrale

De volgende services worden per component gevraagd:

* Client: IBankrekening
* ServerBank: IKlant, IOverboekCentrale
* ServerCentrale: IBank

## Allocatie van objecten

Allocatie van objecten per component:

* Client: Instanties van Klant
* ServerBank: Instanties van Bank, Bankrekening, OverboekCentrale
* ServerCentrale: Instanties van Bank, Bankrekening

## Remote objecten

Remote objecten per component:

* Client: remote instanties van Bankrekening
* ServerBank: remote instanties van OverboekCentrale
* ServerCentrale: remote instanties van Bank

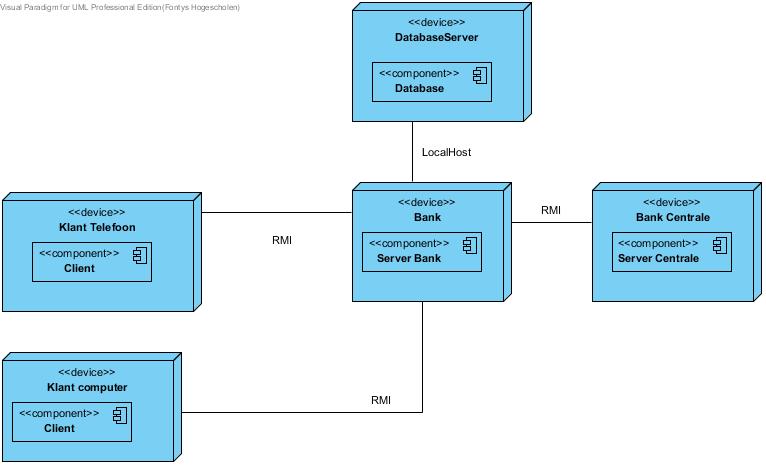
## Packagestructuur

* Package interfaces: IKlant, IBank, IOverboekCentrale, IBankrekening.
* Package centrale: Bank, OverboekCentrale.
* Package bank: Bank, Klant, Bankrekening, OverboekCentrale.
* Package klant: Klant, Bankrekening.

# H7 Deployment

In dit hoofdstuk wordt de toekenning van software componenten aan hardware besproken aan de hand van een deploymentdiagram met toelichting. Daarnaast wordt de koppeling tussen de nodes besproken.

## Deploymentdiagram



Figuur 3: Deployment diagram.

# H8 Specificatie van interfaces

In dit hoofdstuk wordt de specificatie van interfaces besproken. Voor ieder interface wordt per methode gedefinieerd:

* Naam methode
* Naam en type argumenten
* Precondities
* Type returnwaarde
* Beschrijving
* Aanleiding voor excepties

**Interface IBank**

Methode naam: getKlant  
Precondities: de ingevulde gebruikersnaam en wachtwoord zijn correct  
Type returnwaarde: Klant  
Beschrijving: Geeft een Klant object terug  
Aanleiding voor Excepties: -

Methode naam: BoekOver  
Precondities: de gebruiker heeft genoeg saldo en de bankrekening waar het geld heen moet bestaat  
Type returnwaarde: void  
Beschrijving: Bank geeft aan overboekCentrale door dat een saldo van een bankrekening bij een andere bank gewijzigd moet worden.  
Aanleiding voor Excepties: -

Methode naam: AddKlant  
Precondities: de ingevulde gebruikersnaam bestaat nog niet  
Type returnwaarde: void  
Beschrijving: een nieuwe instantie van Klant wordt aangemaakt  
Aanleiding voor Excepties: -

Methode naam: VerwijderKlant  
Precondities:   
Type returnwaarde: void  
Beschrijving: verwijdert een klant uit de database  
Aanleiding voor Excepties: -

Methode naam: WijzigSaldo  
Precondities:   
Type returnwaarde: void  
Beschrijving: wijzigt het saldo van een bankrekening  
Aanleiding voor Excepties: -

**Interface IKlant**

Methode naam: Overschrijven  
Precondities: de gebruiker heeft voldoende saldo op bijbehorende bankrekening  
Type returnwaarde: void  
Beschrijving: Geeft door aan bank dat er een overboeking plaats vindt  
Aanleiding voor Excepties: -

Methode naam: setKlant  
Precondities:   
Type returnwaarde: void  
Beschrijving: maakt een nieuwe instantie van Klant aan  
Aanleiding voor Excepties: -

Methode naam: getBankrekening  
Precondities:.   
Type returnwaarde: Bankrekening  
Beschrijving: haalt de bankrekening op die bij het ingelogde Klant object hoort  
Aanleiding voor Excepties: -

Methode naam: getSaldo  
Precondities:   
Type returnwaarde: int  
Beschrijving: haalt het saldo op  
Aanleiding voor Excepties: -

Methode naam: OpenRekening  
Precondities:   
Type returnwaarde: void  
Beschrijving: Maakt een nieuwe BankRekening aan die bij het huidige Klant object hoort.  
Aanleiding voor Excepties: -

**Interface IBankrekening**

[under construction]

**Interface IOverboekCentrale**

[under construction]