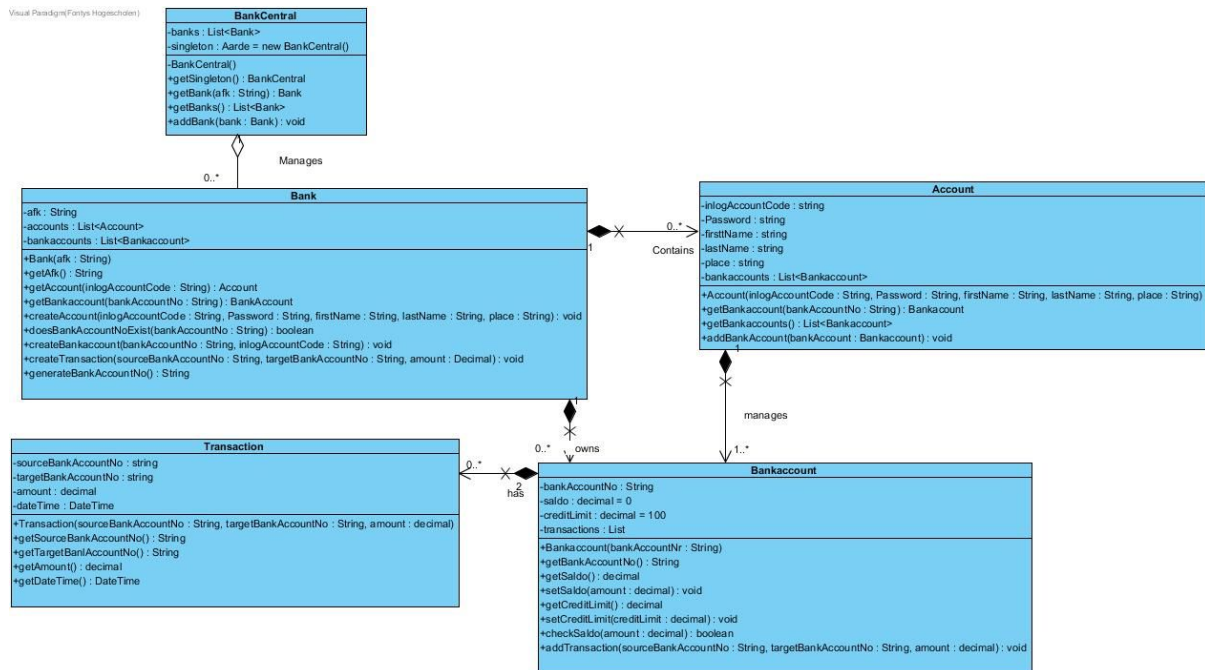


# Documentbeschrijving

## Internetbankieren

Milton van de Sanden & Stefano Verhoeve

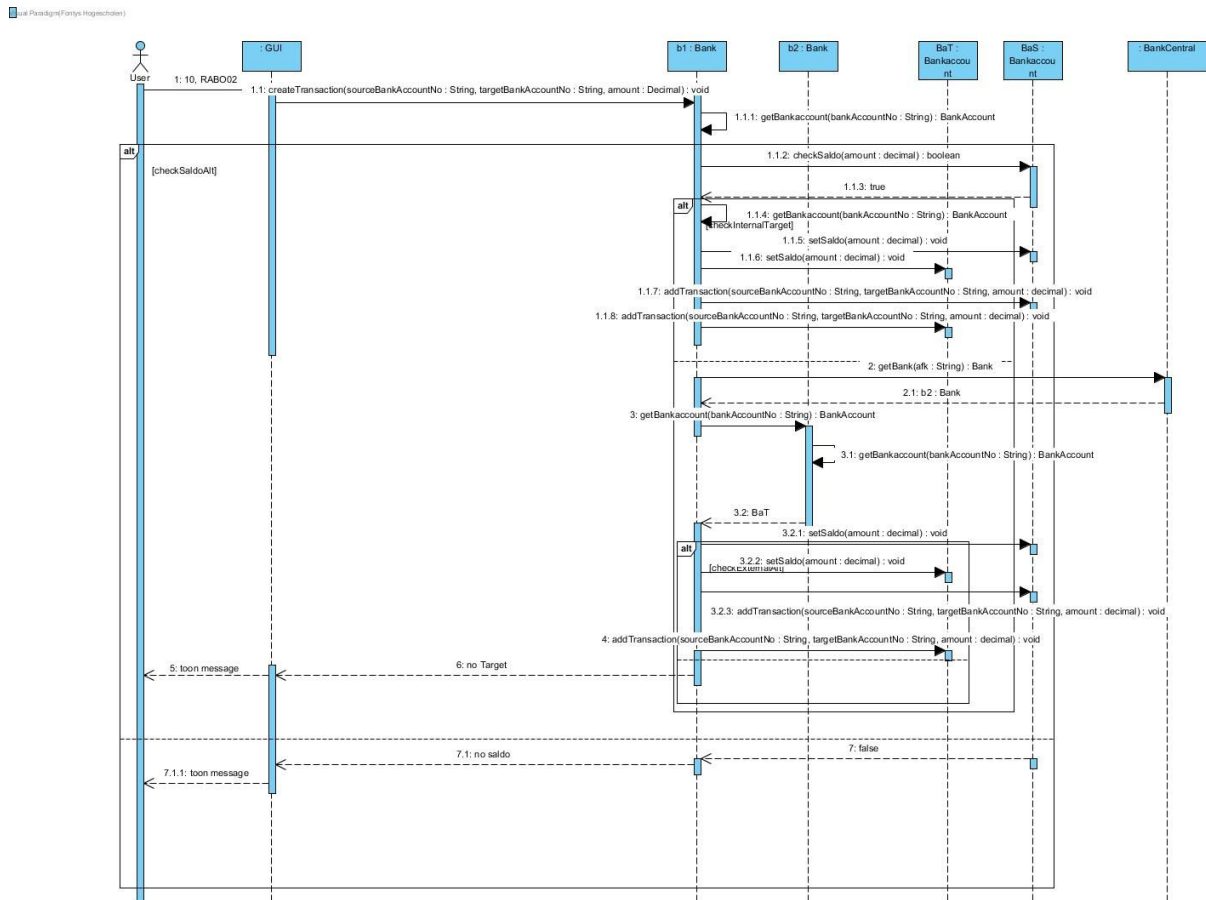
# Klassendiagram - beide



We hebben gekozen voor een overkoepelende klasse, genaamd BankCentral. deze klasse is verantwoordelijk voor toevoegen van een bank aan ene algehele lijst met banken, en kan deze lijst vrijgeven door middel van een getter. BankCentral manages verschillende banken, dit is aangegeven met de relatie. Banken kunnen hoeven niet perse aangesloten te zijn bij een bankcentrale. Bij een bank is het mogelijk om algemene gegevens op te vragen door middel van setter (de afkorting, het account dat bij een inlogaccountcode hoort. Ook is het mogelijk om een account aan te maken, deze vraagt parameters: inlogaccountcode, wachtwoord, firstname, last name en plaats. een bank checkt of een bankaccount bestaat binnen de huidige bank. deze retourneert een boolean. deze methode wordt gebruikt bij het controlleren of er wordt overgeboekt naar een ander account en rekening binnen eigen bank. wanneer een nieuw rekening wordt geopent gebruikt deze klasse de methode generateBankAccountNo(). Bank bestaat uit accounts. in account worden gegevens als Inlogcode, Password, firstname, lastname, place en bankaaccounts bijgehouden. deze zijn private end us alleen opvraagbaar binnen de klasse. een account kan meerdere bankaccounts hebben en daarom wordt er een lijst bijgehouden met bankaccounts. wanneer er een nieuwe bankaccount wordt geopent wordt de methode AddBankAccount() aangeroepen. een bankaccount heeft een bankaccountno een saldo (standaart 0), een kredietlimiet (stadaard 100) en een lijst met transactions. deze transactions kunnen worden weergegeven in de gui.

De klasse bevat verschillende getters om deze gegevens op te vragen. De methode `checkSaldo()` wordt gebruikt om te controleren of er genoeg geld op de rekening staat (niet het kredietlimiet overschrijdt) om een transactie te maken. Wanneer er een correcte overboeking plaatsvindt wordt deze toegevoegd met de methode: `addTransaction()`. De klasse `bankaccount` kan praten met de klasse `bank`. Hiermee wordt gecommuniceert om bijvoorbeeld te controleren of de bank waarnaar een bedrag wordt overgemaakt van dezelfde bank is etc. Als laatste hebben we een klasse `transaction`. Hierin worden het minimum aantal eigenschappen van een transactie vastgelegd. Zaken zoals de bronrekening, en de doelrekening, het bedrag en de datum. Door middel van getters kunnen deze gegevens opgevraagd worden.

# Sequence diagram overboeking - Milton, Stefano



De gebruiker maakt een nieuwe rekening aan, vanuit de GUI wordt op de Bank een methode createTransaction aangeroepen, deze krijgt het rekeningnummer van de bronrekening, het rekeningnummer van de doelrekening, en het over te maken bedrag mee als parameters.

De bronrekening wordt opgehaald en er wordt gecontroleerd of deze genoeg saldo bevat.

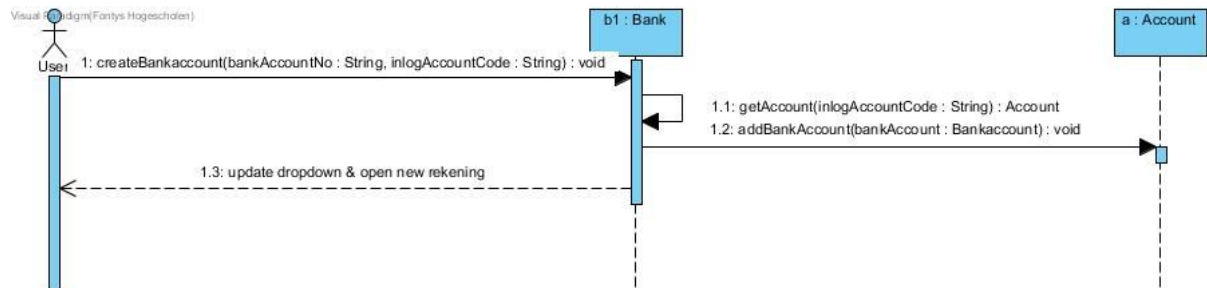
Als de bronrekening genoeg saldo bevat wordt er vervolgens gecontroleerd of de doelrekening van dezelfde bank is.

Als de doelrekening geen onderdeel is van dezelfde bank als de bronrekening wordt er gecommuniceerd via de Centrale Bank.

Nadat de doelrekening is opgehaald, via dezelfde bank of via de centrale bank, worden de saldo's aangepast en een Transaction aangemaakt.

Als de doelrekening niet via de bank of via de centrale bank gevonden kan worden betekend dit dat het opgegeven rekeningnummer voor de doelrekening niet bestaat, indien dit gebeurt wordt de transactie afgebroken.

# Sequence diagram Rekening openen - Stefano, Milton



Het account van de gebruiker wordt opgehaald waarna een nieuwe Bankrekening wordt gegenereerd en gekoppeld aan dit account.