



Opdracht database en testen – FoP2

Vooraf

Schrijf een applicatie volgens het gelaagde model (databaselaag, transactiel laag, interfacelaag). Start vanaf het gegeven project dat terug te vinden is op Toledo.

Voorzie de applicatie van Junittesten.

Zorg dat je op een correcte manier aan exceptionhandling doet.

De applicatie wordt gebouwd met de netbeans-omgeving: Netbeans 8.2. Je zal om dit project uit te werken een database moeten aanspreken. Het SQL-script en ERD-schema vind je bijgevoegd bij de opdracht.

Deze opdracht wordt individueel uitgewerkt.

Deadlines

De opdracht moet gefaseerd ingediend worden.

Deadline 15 oktober 2017 om 23u55 – Use-cases

Dien van alle functionaliteit de use-cases in. De use-cases beschrijven wie welke acties kan doen en welke gevolgen deze acties hebben. Zo zou in een banksysteem volgende use-case kunnen uitgeschreven worden voor de actie “storten”.

Use Case naam: storten

Omschrijving: Klant 1, Pietje Precies, stort een positief bedrag op de rekening van klant 2, Bob Debouwer. Het bedrag gaat van zijn rekening af, en wordt toegevoegd aan het saldo van de rekening van klant 2.

Betrokken actoren: Pietje Precies, Bob Debouwer

Pre-condities:

- Klant 1 en klant 2 hebben een rekening.
- Het saldo van de rekening van klant 1 is groter dan het bedrag dat gestort moet worden.
- Het te storten bedrag is positief.

Initiërende gebeurtenis: klant 1 geeft zijn eigen rekening in, vult het rekeningnummer van klant 2 in en geeft het bedrag dat klant 1 wil storten door.

Post-condities:

- Het saldo van de rekening van klant 1 is het oude saldo – het gestorte bedrag
- Het saldo van de rekening van klant 2 is het oude saldo + het gestorte bedrag

Het document met alle use-cases is van type pdf, en krijgt de naam:

naam1_voornaam1_naam2_voornaam2_use-case.pdf (dus bv. Deleu_Katrien_Cools_Kasper_use-case.pdf)

Deadline 25 maart 2018 om 23u55 – Definitieve versie

Dien de definitieve versie van je applicatie in met alle gevraagde functionaliteit en de testklasse voor RitDB en RitTrans, aangevuld met een PDF-document waarin kort wordt uitgelegd welke handelingen wij als evaluatoren moeten ondernemen om de applicatie te kunnen opstarten, welke extra functionaliteiten je aan de applicatie toegevoegd hebt en een opsomming van tekortkomingen/fouten in je applicatie. Mocht je de database aangepast hebben, dan argumenteer je waarom je dit deed.

Dit alles:

- de map met oplossing (het project heeft de naam naam_voornaam_VIVESbike)
- een export van je databank (create statements en populate statements)
- het PDF-bestand

wordt gezipd tot één file, en krijgt de naam:

naam1_voornaam1_naam2_voornaam2_VIVESbike.zip (dus bv. Deleu_Katrien_Cools_Kasper_VIVESbike.zip).

Puntenverdeling

- Wie **administratief** niet in orde is, verliest punten op het eindresultaat van de opdracht. Er wordt onder andere rekening gehouden met volgende zaken:
 - naam bestand
 - naam project
 - pdf in degelijk Nederlands
 - scripts voorzien
- **Use-cases** op 10%. Let op je taal!
- **DB-klassen** op 20%
- **Transactieklassen** op 30%
- **Grafische userinterface** op 15%. Wie meer schermen maakt of meer functionaliteit voorziet, kan extra punten krijgen, maar je kan maximum 15/15 behalen voor dit onderdeel.
- **Testklassen** op 15%. Je kan extra punten verdienen als je voor de andere klassen ook goede testklassen schrijft, maar je kan maximum 15/15 behalen voor dit onderdeel.
- **Mondeling** op 10%. Wie geen mondeling aflegt, krijgt ook geen punten voor de onderdelen DB-klassen, Transklassen, grafische userinterface en testklassen. Wanneer tijdens het mondeling examen dat de student z'n project niet kan toelichten, kan extra punten verliezen op de andere onderdelen.

Gegeven

- script voor het aanmaken van de database. De database maak je best aan met USBWebserver.
- Basisproject met volgende gegeven klassen
 - Bags Fiets, Lid en Rit
 - Connectieklassen ConnectionManager en DBProp
 - Datatypeklasse Status, Standplaats, Geslacht, Rijksregisternummer
 - Exceptionklassen DBException en ApplicationException
 - Interfaces (die niet gewijzigd mogen worden!) voor de databaselaag
 - Concrete klassen voor de databaselaag, die aangevuld moeten worden. Vergeet niet om daar waar nodig de header aan te vullen met "throws" en de juiste exceptionklassen.
 - Interfaces (die niet gewijzigd mogen worden!) voor de transactielaag
 - Concrete klassen voor de transactielaag, die aangevuld moeten worden. Vergeet niet om daar waar nodig de header aan te vullen met "throws" en de juiste exceptionklassen.
 - Eenvoudige UI-klassen die aangevuld moeten worden (starten nu niet correct op)

Gevraagd

VIVESbike

Vives wil naar analogie van de blue-bike, ook fietsen verhuren (standplaats station) bij de verschillende vestigingen van VIVES.

Om een fiets te kunnen huren moet een persoon lid zijn. Hij betaalt hiervoor per jaar 3€ lidgeld en krijgt hiervoor een VIVESbike-kaart, en kan dan om het even wanneer één fiets huren (één persoon kan nooit meerdere fietsen tegelijk huren). Een persoon blijft lid totdat hij zich uitschrijft door z'n VIVESbike-kaart terug in te dienen. We noteren dan z'n uitschrijfdatum, maar verwijderen de persoon niet. Naast de standaardgegevens, houden we van een lid opmerkingen bij. Hier kan dan bijvoorbeeld bijgehouden worden of er dikwijls problemen zijn met deze persoon (betalingen, beschadigingen, diefstal, ...). Aangezien de inschrijfdatum bijgehouden wordt, kan er vanuit het systeem ieder jaar automatisch een factuur vertrekken om de 3€ lidgeld aan te rekenen. Het factureren valt buiten de scope van de oefening. De gegevens van een lid kunnen gewijzigd worden, zolang deze ingeschreven is. Het gaat dan om naam, voornaam, geslacht, telefoonnummer, e-mail en opmerkingen. De startdatum kan ook gewijzigd worden, maar moet altijd voor de start van de eerste rit blijven, anders zou de persoon een fiets gehuurd hebben zonder lid te zijn. De einddatum wordt ingevuld wanneer het lid zich uitschrijft, en kan niet meer gewijzigd worden. Een lid kan zich maar uitschrijven wanneer hij/zij geen fiets aan het huren is.

De fietsen die gehuurd kunnen worden staan steeds aan het station. Wie een fiets afhaalt aan het station betaalt 1€ en mag die 24 uur gebruiken. Wie de fiets langer wil gebruiken, mag dat, maar die betaalt voor elke 24 uur opnieuw 1€. Bij het afhalen aan het station kan de huurder met z'n VIVESbike-kaart de fietssleutel ophalen aan de sleutelautomaat. Op dit moment start zijn huurtijd. De fiets moet telkens teruggebracht worden naar hetzelfde station. Op het moment dat de huurder z'n sleutel terug binnenbrengt, eindigt de huurtijd. De prijs wordt automatisch berekend en bijgehouden. Uiteraard kan een fiets niet aan twee leden tegelijk verhuurd worden.

Een fiets kan nooit van standplaats wijzigen. Er kan ten allen tijde een opmerking toegevoegd worden aan een fiets. Een fiets kan zich in 3 verschillende toestanden bevinden: actief, in herstel of uit omloop.

- **ACTIEF:** Een fiets kan maar gehuurd worden wanneer die zich in de toestand actief bevindt. Wanneer een fiets wordt toegevoegd is hij per definitie actief.
- **IN HERSTEL:** Een fiets kan na een rit beschadigd zijn. De huurder kan dit na de rit melden aan de sleutelautomaat. Wanneer zo'n melding wordt gemaakt, wordt de fiets in toestand 'herstel' gebracht. Er kan aanvullend een opmerking toegevoegd worden aan het dossier van de huurder of aan de gehuurde fiets. Wanneer de fiets hersteld is, wordt die weer actief gemaakt.
- **UIT OMLOOP:** Er kan beslist worden dat een fiets niet meer gebruikt zal worden en dus uit omloop wordt gehaald. Dergelijke fiets wordt nooit meer in omloop gebracht.

Volgende zaken worden dus bijgehouden in de database:

- Lid (per jaar 3€, niets mee doen)
 - rijksregisternummer
 - naam
 - voornaam
 - geslacht
 - telefoonnummer
 - e-mail
 - start lidmaatschap
 - einde lidmaatschap
 - opmerkingen (bv. dikwijls schade aan fiets, diefstal van fiets, verlies van sleutel, slechte betalen, ...)
- Fiets
 - registratienummer
 - standplaats (station Roeselare, Kortrijk, Torhout, Tielt, Oostende, Brugge)
 - status (actief, herstel, uit_omloop)
 - opmerkingen (bv. waarom de fiets uit omloop gehaald werd)
- Rit
 - id
 - starttijd
 - eindtijd
 - prijs
 - huurder
 - gehuurde fiets

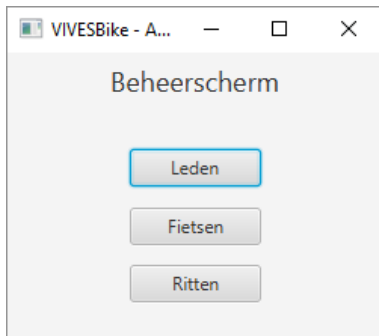
Code

In de applicatie moet volgende mogelijk zijn tot op transactieniveau:

- Toevoegen lid, wijzigen lid, uitschrijven lid
- Toevoegen fiets, wijzigen status van fiets en opmerkingen toevoegen aan fiets
- Toevoegen rit en afsluiten rit

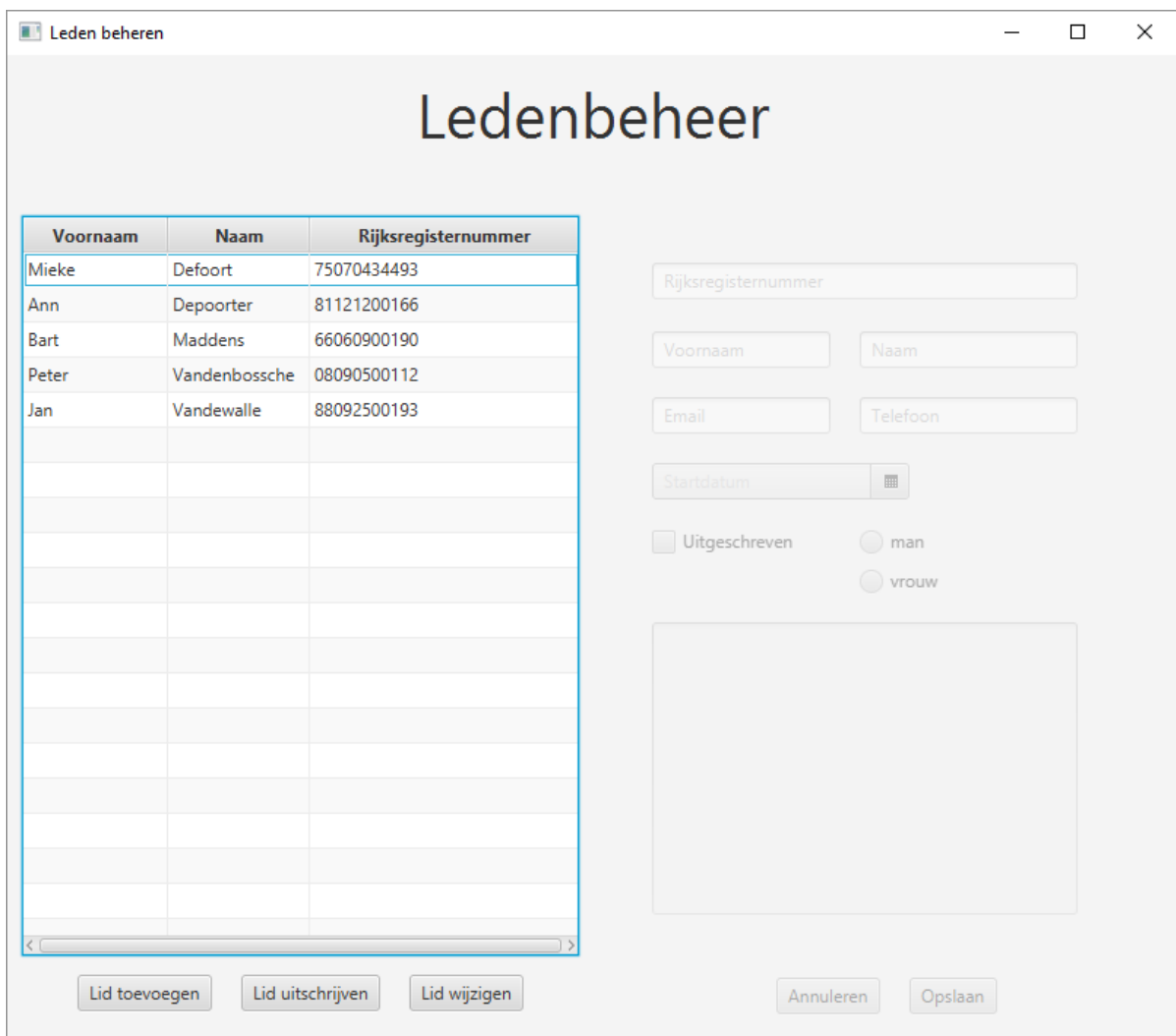
Grafische userinterface

In de UI moeten minstens volgende zaken uitgewerkt zijn¹:



Figuur 1 - Startscherm

Bij het opstarten van het programma kan de gebruiker kiezen tussen ‘Leden’, ‘Fietsen’ en ‘Ritten’. Enkel de knop ‘Leden’ moet werken en start scherm Figuur 2 - Overzicht leden op.



Figuur 2 - Overzicht leden

¹ De grafische userinterface is minimalistisch en niet altijd even vlot in gebruik. De nadruk ligt bij de userinterface eerder bij het functioneren van de knoppen en het opvangen en melden van fouten.

Wanneer een lid wordt geselecteerd wordt alle extra info getoond in het linker deel van het scherm (Figuur 3 - Een lid selecteren).

Leden beheer

Ledenbeheer

Voornaam	Naam	Rijksregisternummer
Mieke	Defoort	75070434493
Ann	Depoorter	81121200166
Bart	Maddens	66060900190
Peter	Vandenbossche	08090500112
Jan	Vandewalle	88092500193

81121200166

Ann Depoorter

ann.depoorter@vive 092000034

23/12/2016

☐ Uitgeschreven ☐ man ☒ vrouw

Annuleren Opslaan

Lid toevoegen Lid uitschrijven Lid wijzigen

Figuur 3 - Een lid selecteren

Wanneer de gebruiker een lid wil toevoegen (en dus op 'Lid toevoegen' klikt) worden alle tekstvelden links leeggemaakt en worden de correct knoppen beschikbaar (Figuur 4 - Op de knop 'Lid toevoegen' klikken).

[illegible]

Figuur 4 - Op de knop 'Lid toevoegen' klikken

Wanneer de gebruiker bij het toevoegen van een nieuw lid vergeet om een verplicht veld in te vullen en op opslaan klikt, verschijnt een foutmelding (Figuur 5 - Een lid toevoegen met onvoldoende informatie).

[illegible]

Figuur 5 - Een lid toevoegen met onvoldoende informatie

Wanneer de gebruiker bv. een lid probeert uit te schrijven, en hiervoor op de knop 'Lid uitschrijven' klikt, wordt het geselecteerde lid uitgeschreven. Wanneer blijkt dat er geen lid geselecteerd werd, verschijnt een foutboodschap (Figuur 6 - Foutmelding wanneer op knop 'Lid uitschrijven' wordt geklikt, maar er geen lid geselecteerd is.).

[illegible]

Figuur 6 - Foutmelding wanneer op knop 'Lid uitschrijven' wordt geklikt, maar er geen lid geselecteerd is.

Om een lid te wijzigen klikt de gebruiker op 'Lid wijzigen'. Wanneer er geen lid geselecteerd is, verschijnt een foutmelding. Is er wel een lid geselecteerd, dan wordt de overeenkomstige data beschikbaar in de tekstvelden en kan het lid aangepast worden (Figuur 7 - De naam van lid Ann wijzigen naar Maaïke).

The screenshot shows a web application window titled 'Leden beheer'. On the left is a table with member data. The row for 'Ann Depoorter' is highlighted. On the right is a form to edit the selected member's details. The form fields contain the following data:

Voornaam	Naam	Rijksregisternummer
Mieke	Defoort	75070434493
Ann	Depoorter	81121200166
Bart	Maddens	66060900190
Peter	Vandenbossche	08090500112
Jan	Vandewalle	88092500193

Form fields on the right:

- Rijksregisternummer: 81121200166
- Voornaam: Maaïke
- Naam: Depoorter
- E-mail: ann.depoorter@vive
- Telefoon: 092000034
- Geboortedatum: 23/12/2016
- Uitgeschreven: ☐
- Gender: ☐ man, ☒ vrouw
- Profile picture: Empty box

Buttons at the bottom:

- Lid toevoegen
- Lid uitschrijven
- Lid wijzigen
- Annuleren
- Opslaan

Figuur 7 - De naam van lid Ann wijzigen naar Maaïke

Hoe ga je te werk ?

Om deze opdracht tot een goed einde te brengen is het belangrijk dat je de oefeningen op de Java FX-technologie reeds hebt gemaakt.

Bestudeer ook grondig de cursus 'Databases in Java' zodat je een goede notie hebt van hoe het meerlagenmodel werkt. Het voorbeeld van de bank zou je hierbij verder op weg moeten helpen.

Zorg dat je de pesisitentielaag en de transactielaag goed test! Het staat als een paal boven water dat wanneer deze twee lagen stabiel zijn, het ontwerpen van een grafische interface veel gemakkelijker maakt: als je in deze fase nog fouten tegenkomt, moet je die bij de deze laatste klassen zoeken.

Het correct toepassen van de het lagenmodel, het correct gebruik van exceptions en de inputcontrole zijn van veel groter belang dan mooie kleuren en fonttypes. Zorg dus in eerste instantie dat de lagen goed werken en er op de gepaste momenten exceptions worden gegooit en opgevangen. De inputcontrole kan maar goed werken wanneer die zaken goed zitten. Alles wat je hierna nog kunt toevoegen om de grafische userinterface gebruiksvriendelijk te maken, zijn leuke extra's.

Veel succes!!