

Opdracht Vodagone REST

1. Inleiding

Deze opdracht bestaat uit verschillende delen:

- uitbreiding ontwerp met package- en deploymentdiagram
- realisatie van de functionaliteit

Je levert deze opdracht individueel in en demonstreert zelf de werking aan de docent. Tijdens de ontwerpfase mag je samenwerken aan de UML-diagrammen, tijdens de bouwfase schrijf je alle code zelf maar mag je met collega-studenten overleggen en/of elkaar hulp bieden. Je bent en blijft zelf verantwoordelijk voor wat je inlevert en demonstreert.

2. Casus: Spotitube

Vodafone en Ziggo hebben de handen [ineen geslagen](#) en werken gezamenlijk aan een app (Vodagone) waarmee een klant inzicht kan krijgen in de abonnementen die bij de bedrijven zijn afgesloten. Ze willen eerst een deel van de back-end ontwikkelen en deze testen via een eenvoudige webapplicatie alvorens over te gaan tot de ontwikkeling van de app. De broncode van de back-end moet voldoen aan de clean code principes/regels zodat deze daarna makkelijk uit te breiden is als de webapplicatie een succes blijkt.

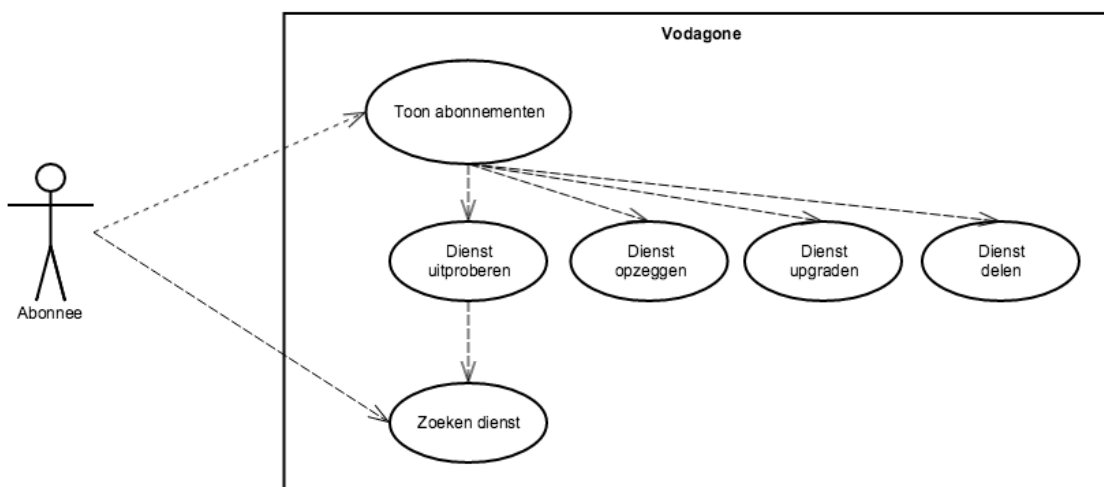
De applicatie moet naast MySQL5.1 ook minimaal 1 andere relationele database kunnen ondersteunen (het daadwerkelijke bewijs daarvan kan je leveren tijdens de onderzoeksopdracht en valt buiten de scope van deze opdracht).

De applicatie maakt gebruik van de volgende APIs en frameworks:

- JAX-RS v2.0 (REST, JSON)
- CDI (Context & Dependency injection)
- JDBC API, JDBC driver v5.1.34 voor MySQL
- De applicatie wordt ontwikkeld met Java versie 8 en moet draaien in Apache TomEE Plus.

De front-end en back-end moeten Restful (JSON over HTTP) kunnen communiceren volgens de [REST API REST API specificatie](#).

3. Functionele eisen



In de bovenstaande figuur staan de architectureel significante use cases, ofwel de use cases die als functionaliteit gekozen zijn om aan te tonen dat aan de niet-functionele eisen voldaan kan worden.

Een abonnee (waarvan een e-mailadres en naam zijn vastgelegd) start de applicatie en de applicatie toont alle abonnementen voor de abonnee, dat kan een abonnement van Ziggo of van Vodafone zijn. In het abonnementenoverzicht wordt de totaalprijs per maand berekend van alle abonnementen.

Van elk abonnement kan de einddatum verlengd (voor een bestaand abonnement) of vastgesteld (nieuw abonnement) worden voor een maand, half jaar of jaar, de startdatum kan niet worden aangepast. Hoe langer het abonnement loopt, hoe goedkoper de dienst.

De abonnee heeft de mogelijkheid om vanaf het abonnementsoverzicht een of meerdere nieuwe diensten aan te vragen, zoals bijvoorbeeld Eredivisie Live of HBO. Een aanvraag start altijd met een gratis proefabonnement, dit abonnement duurt altijd een maand, gaat in per de eerste van de maand en wordt stilzwijgend omgezet in een maandabonnement tenzij het opgezegd wordt. Opzegging van een abonnement wordt uitgevoerd per ingang van de einddatum en kan niet ongedaan gemaakt worden. De status van een abonnement is dus *proef, actief of opgezegd*.

Om een dienst uit te proberen kan in de gezamenlijke catalogus van Ziggo en Vodafone gezocht worden (zet de onderstaande voorbeelden zelf in een aparte tabel, hier hoeft je geen beheer-schermen voor te maken).

Alle diensten hebben een naam, prijs per maand, prijs per half jaar en prijs per jaar, enkele diensten zijn deelbaar met 1 of meerdere andere abonnementshouders of zijn te verdubbelen of *upgraden* (in snelheid of aantal minuten/SMS/GB). Een dienst is nooit deelbaar en verdubbelaar tegelijkertijd en een deling of verdubbeling kan niet ongedaan gemaakt worden.

Abonnementsoorten en -prijzen

| Aanbieder | Naam van de dienst | Prijs per maand | Prijs per half jaar | Prijs per jaar | Deelbaar | Verdubbeling mogelijk |
|-----------|---|-----------------|---------------------|----------------|-------------------------------|----------------------------------|
| Vodafone | Mobiele telefonie 100 <ul style="list-style-type: none">100 minuten, SMS of GB. | €5,- | €25,- | €45,- | Nee | Nee |
| Vodafone | Mobiele telefonie 250 <ul style="list-style-type: none">250 minuten, SMS of GB. | €10,- | €50,- | €90,- | Nee | Ja, tegen 50% per periode extra. |
| Vodafone | Glasvezel-internet (download 500 Mbps) | €40,- | €200,- | €360,- | Nee | Ja, tegen 50% per periode extra. |
| Ziggo | Kabel-internet (download 300 Mbps) | €30,- | €150,- | €270,- | Nee | Nee |
| Ziggo | Eredivisie Live 1,2,3,4 en 5 | €10,- | €50,- | €90,- | Ja, met max 2 andere abonnees | Nee |
| Ziggo | HBO Plus | €15,- | €75,- | €135,- | Ja, met max 2 andere abonnees | Nee |

3.1. Use-case Descriptions

In deze sectie wordt elke use case in detail beschreven in het fully dressed formaat.

3.1.1. Toon abonnementen

| | | |
|------------------------------------|--|---|
| Primary actor | Abonnee | |
| Brief Description | Een abonnee start de applicatie en de applicatie toont alle abonnementen voor de abonnee, dat kan een abonnement van Ziggo of van Vodafone zijn. In het abonnementsoverzicht wordt de totaalprijs per maand berekend van alle abonnementen. In het overzicht worden ook abonnementen opgenomen worden die gedeeld worden met anderen, als het abonnement is gestart/afgesloten door een andere abonnee betaalt de huidige abonnee geen extra kosten. | |
| Preconditions | De abonnee is ingelogd. De abonnee heeft minimaal 1 bestaand abonnement. | |
| Postconditions (Success Guarantee) | De abonnee heeft een overzicht van zijn abonnementen en de totaalprijs per maand. | |
| Main Succes Scenario | Actor Action | System Responsibility |
| | 1. De abonnee start de applicatie | 2. Het systeem toont de abonnementen en de totaalprijs per maand. De abonnee kan vanaf hier een van de volgende use cases starten: <ul style="list-style-type: none">Dienst uitproberenDienst opzeggenDienst upgradenZoeken dienst |

3.1.2. Dienst uitproberen

| | | |
|---|---|--|
| Primary actor | Abonnee | |
| Brief Description | De abonnee heeft de mogelijkheid om vanaf het abonnementenoverzicht een of meerdere nieuwe diensten aan te vragen, zoals bijvoorbeeld Eredivisie Live of HBO. Een aanvraag start altijd met een gratis proefabonnement, dit abonnement duurt altijd een maand, gaat in per de eerste van de maand en wordt stilzwijgend omgezet in een maandabonnement tenzij het opgezegd wordt. | |
| Preconditions | De use case Toon abonnementen is succesvol uitgevoerd. | |
| Postconditions (Success Guarantee) | De abonnee heeft een nieuw abonnement afgesloten met als status "PROEF". | |
| Main Success Scenario | Actor Action | System Responsibility |
| | 1. De abonnee geeft aan een nieuw abonnement te willen aanvragen. | 2. Het systeem geeft de abonnee de mogelijkheid om te zoeken naar een dienst. De use case Zoeken Dienst start. |
| | 3. De abonnee kiest een specifieke dienst om te willen uitproberen. | 4. Het systeem voegt de dienst toe aan de lijst van abonnementen voor een looptijd van een maand met als status "PROEF". |

3.1.3. Dienst upgraden

| | | |
|---|---|---|
| Primary actor | Abonnee | |
| Brief Description | De abonnee kiest een dienst om te upgraden. | |
| Preconditions | De use case Toon abonnementen is succesvol uitgevoerd. | |
| Postconditions (Success Guarantee) | De gekozen dienst is geupgradet doordat de status van de klasse Abonnement op true staat | |
| Main Success Scenario | Actor Action | System Responsibility |
| | 1. De abonnee selecteert een dienst die geupgradet moet worden. Alleen diensten die daadwerkelijk upgradebaar zijn kunnen gekozen worden. | Het systeem upgradet de gekozen dienst door het aantal minuten/SMS/GB te verdubbelen en de prijs per maand aan te passen. De status van het abonnement verandert naar "verdubbeld". |

3.1.4. Dienst opzeggen

| | | |
|---|--|------------------------------|
| Primary actor | Abonnee | |
| Brief Description | De abonnee kiest een dienst om op te zeggen. | |
| Preconditions | De use case Toon abonnementen is succesvol uitgevoerd. | |
| Postconditions (Success Guarantee) | De gekozen dienst is opgezegd doordat de status van de klasse Abonnement op "OPGEZEGD" staat | |
| Main Success Scenario | Actor Action | System Responsibility |

| | | |
|--|---|---|
| | 1. De abonnee selecteert een dienst die opgezegd moet worden. | 2. Het systeem zet de status van het betreffende abonnement op "OPGEZEED". De maandprijs wordt 0 na het verlopen van de abonnementsperiode. |
|--|---|---|

3.1.5. Dienst delen

| | | |
|---|---|---|
| Primary actor | Abonnee | |
| Brief Description | De abonnee kiest een dienst om op te delen. | |
| Preconditions | De use case Toon abonnementen is succesvol uitgevoerd. | |
| Postconditions (Success Guarantee) | De gekozen dienst is gedeeld met minimaal 1 andere abonnee. | |
| Main Success Scenario | Actor Action | System Responsibility |
| | 1. De abonnee selecteert een dienst die hij wil delen. Alleen diensten die daadwerkelijk deelbaar zijn kunnen gekozen worden. | 2. Het systeem geeft de abonnee de mogelijkheid een naam van een andere abonnee in te voeren. |
| | 3. De abonnee voert de naam van andere abonnee in. | 4. Het systeem registreert de deling en vraagt of de dienst met nog meer abonnee's moet worden gedeeld. |
| | 5. De abonnee geeft aan dat het delen is afgerond. | 6. Het systeem toont het abonnementenoverzicht, de use case Toon abonnementen start. |
| Extensions (Alternative Flows) | 5a. De abonnee geeft aan nog een abonnee te willen toevoegen als "deler". | De use case gaat verder met stap 2. |

3.1.6. Zoeken dienst

| | | |
|---|--|--|
| Primary actor | Abonnee | |
| Brief Description | De abonnee zoekt naar een dienst om hier meer informatie over te krijgen of om de dienst toe te voegen aan het abonnement. | |
| Preconditions | De abonnee is ingelogd. | |
| Postconditions (Success Guarantee) | De abonnee heeft een lijst van diensten die voldoen aan zijn criteria. | |
| Main Success Scenario | Actor Action | System Responsibility |
| | 1. De abonnee voert de naam in van de dienst. | 2. Het systeem toont de diensten die voldoen aan de ingevoerde naam. |
| Extensions (Alternative Flows) | | 2a. Er is geen dienst beschikbaar die voldoet aan de ingevoerde naam. Het systeem toont dit middels een melding en geeft de abonnee een nieuwe kans om te zoeken of om terug te gaan naar het abonnementenoverzicht. |

4. Opdracht

4.1. Maak een package diagram en een deployment diagram

Realiseer op basis van het huidige ontwerp en de niet-functionele eisen volgende diagrammen:

- een UML packagediagram dat de packagestructuur van de broncode illustreert, inclusief een toelichting (voor details, zie bullets bij *motivatie*) die uitlegt welke ontwerpkeuzes gemaakt zijn
- een UML deploymentdiagram dat de runtimeomgeving, protocollen, componenten en overige infrastructuur zoals servers, besturingssystemen en databases laat zien, inclusief een toelichting (voor details, zie bullets bij *motivatie*) die uitlegt welke ontwerpkeuzes gemaakt zijn

Licht elk diagram toe met een geschreven motivatie waarin je opneemt:

- welke requirement je ermee raakt en/of oplost
- welke design patterns of andere ontwerpprincipes je hebt toegepast
- wat een alternatieve oplossing zou zijn geweest
- waarom de huidige oplossing het beste is

4.2. Realiseer de applicatie op basis van deze casusbeschrijving

Realiseer de Spotitube-applicatie waarbij je gebruik maakt van de volgende layers:

1. Data access layer
 - Implementeer de patterns Data Mapper, Identity Map en Separated Interface (ook wel bekend als het ISP uit SOLID) door middel van JDBC en MySQL.
2. Domain layer
 - Implementeer het Service Layer pattern
 - Implementeer het Remote Facade pattern en Data Transfer Object pattern door middel van REST-services en JSON.
3. Interactie tussen lagen
 - Verlaag de afhankelijkheid tussen de lagen door dependency injection toe te passen
 - Toon aan dat de afhankelijkheid verlaagd is door meerdere unit-tests, kies zelf geschikte klassen waarvoor je deze unit-tests maakt en maak gebruik van test-doubles zoals mocks en/of stubs.

De presentation layer is gegeven middels de client-applicatie en hoeft je niet zelf te realiseren maar je moet deze wel gebruiken om jouw back-end aan te spreken.

4.3. Maak een opleverdocument

Maak een opleverdocument met daarin minimaal:

- de UML diagrammen inclusief motivatie/toelichting
- een verzameling ontwerpkeuzes die je illustreert met UML diagrammen naar keuze

Het opleverdocument moet voldoen aan de ICA stijlkaart.

5. Beoordelingschecklist

| Beoordelingscriterium | Oordeel (Cijfer 1-10. Voor alle onderdelen minimaal een 5 nodig voor het berekenen van een gemiddeld cijfer. Alle onderdelen zijn dus verplicht maar er is onderlinge compensatie mogelijk) | Weging |
|---|---|--------|
| Het packagediagram voldoet aan de UML specificatie en is voorzien van een geschreven motivatie. In de motivatie is opgenomen: <ul style="list-style-type: none">• welke requirement je ermee raakt en/of oplost• welke design patterns of andere ontwerpprincipes je hebt toegepast• wat een alternatieve oplossing zou zijn geweest• waarom de huidige oplossing het beste is | | 10% |

| | | |
|--|--|-----|
| <p>Het deploymentdiagram voldoet aan de UML specificatie en is voorzien van een geschreven motivatie. In de motivatie is opgenomen:</p> <ul style="list-style-type: none"> • welke requirement je ermee raakt en/of oplost • welke design patterns of andere ontwerpprincipes je hebt toegepast • wat een alternatieve oplossing zou zijn geweest • waarom de huidige oplossing het beste is | | 10% |
| <p>De applicatie bevat meerdere unit-tests die aantoont dat lagen volgens interfaces van elkaar gescheiden zijn en daarmee los testbaar zijn. Je maakt hierbij gebruik van test-doubles (mocks en/of stubs).</p> | | 15% |
| <p>Er is een opleverdocument met daarin minimaal:</p> <ul style="list-style-type: none"> • de package- en deploymentdiagrammen inclusief motivatie/toelichting • een verzameling ontwerpkeuzes die je illustreert met UML diagrammen naar keuze <p>Het opleverdocument moet voldoen aan de ICA stijlkaart.</p> | | 5% |
| <p>De applicatie implementeert de REST API zoals deze gedefinieerd is en verwacht wordt door de client. In een demo toon je aan dat de client via de REST API kan communiceren met jouw applicatie. In je applicatie gebruik je in elk geval de volgende patterns/principes:</p> <ul style="list-style-type: none"> • DIP principe geïmplementeerd met CDI • Data Mapper, Identity Map en Separated Interface (ISP), werkend met JDBC en MySQL • Service Layer • Remote Facade, werkend met REST/JSON en Data Transfer Objects in de vorm van JSON | | 50% |
| <p>Je applicatie- en testcode voldoet aan de clean code principes/regels.</p> | | 10% |