

# Opdracht Spotitube

## 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 en samen feedback verwerken van een ander twee- of drietal, 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

Spotify en Youtube hebben de handen ineen geslagen en werken gezamenlijk aan een app (Spotitube) waarmee een klant inzicht kan krijgen zijn afspeellijsten met daarin streams en video's. 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 eisen voor de Spotitube-applicatie zijn vastgelegd in een [Software Requirements Specification \(SRS\)](#), het ontwerp in een [Software Design Document \(SDD\)](#).

## 3. Opdracht

### 3.1. 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

#### 3.1.1. Verwerk feedback

Inventariseer voor een van de andere twee-/drietalen:

- minimaal 2 vragen
- minimaal 2 verbeterpunten
- minimaal 2 complimenten

Verwerk de vragen en verbeterpunten die je zelf gekregen hebt en neem laat zien waar de verschillen in welke diagrammen of toelichtingen zitten.

### 3.2. Realiseer de applicatie op basis van deze casusbeschrijving en het gegeven ontwerp

Realiseer de Spotitube-applicatie. Demonstreer de tussenresultaten van elke stap aan je docent:

1. Data access layer
  - Implementeer het Table Data Gateway pattern (DAO)
  - Gebruik JDBC en MySQL
2. Domain layer
  - Implementeer het domain model pattern

- Implementeer het service layer pattern
  - Maak 2 implementaties van de service layer
    - Een REST/JSON service die testbaar is in een REST client zoals Postman
    - Een POJO service
  - Spreek de data access layer uit stap 1 aan.
3. Presentation layer
- Gebruik JSP en Servlet
  - Implementeer het MVC en het Page Controller pattern
  - Spreek de domain layer uit stap 2 aan.
4. Interactie tussen lagen
- Verlaag de afhankelijkheid tussen de lagen door dependency injection toe te passen
  - Toon aan dat de afhankelijkheid verlaagd is door een of meerdere unittests, kies zelf geschikte klassen waarvoor je deze unittests maakt.

### 3.3. Maak een opleverdocument

Maak een opleverdocument met daarin minimaal:

- de UML diagrammen inclusief motivatie/toelichting
- een tabel met functionele eisen en niet-functionele eisen waaruit duidelijk wordt waar deze eisen in de broncode geïmplementeerd zijn. In de broncode staat cross-references naar deze eisen zodat duidelijk wordt welke eis waar in de broncode is geïmplementeerd.

Het opleverdocument moet voldoen aan de ICA stijlkaart.

## 4. 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
<p>Het packagediagram voldoet aan de UML specificatie en is voorzien van een geschreven motivatie. In de motivatie is opgenomen:</p> <ul style="list-style-type: none"> <li>• welke requirement je ermee raakt en/of oplost</li> <li>• welke design patterns of andere ontwerpprincipes je hebt toegepast</li> <li>• wat een alternatieve oplossing zou zijn geweest</li> <li>• waarom de huidige oplossing het beste is</li> </ul>		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"> <li>• welke requirement je ermee raakt en/of oplost</li> <li>• welke design patterns of andere ontwerpprincipes je hebt toegepast</li> <li>• wat een alternatieve oplossing zou zijn geweest</li> <li>• waarom de huidige oplossing het beste is</li> </ul>		10%
<p>De applicatie bevat minimaal een (1) unit-test die aantoont dat lagen volgens interfaces van elkaar gescheiden zijn en daarmee los testbaar zijn.</p>		5%
<p>Er is een opleverdocument met daarin minimaal:</p> <ul style="list-style-type: none"> <li>• de UML diagrammen inclusief motivatie/toelichting</li> <li>• een tabel met functionele eisen en niet-functionele eisen waaruit duidelijk wordt waar deze eisen in de broncode geïmplementeerd zijn. In de broncode staat cross-references naar deze eisen zodat duidelijk wordt welke eis waar in de broncode is geïmplementeerd.</li> </ul> <p>Het opleverdocument moet voldoen aan de ICA stijlkaart.</p>		5%

<p>De implementatie voldoet aan de door de HAN gegeven en de student aangevulde ontwerpen en (functionele en niet-functionele) eisen uit de casusbeschrijving. Hiervoor worden in elk geval de volgende patterns geïmplementeerd:</p> <ul style="list-style-type: none"> <li>• DIP principe geïmplementeerd met CDI</li> <li>• Table Data Gateway a.k.a. Data Access Object (DAO), werkend met JDBC en MySQL</li> <li>• Service Layer, werkend met REST/JSON en POJO's</li> <li>• Domain Model</li> <li>• PageController, werkend met JSP en Servlet</li> <li>• Model View Controller, werkend met JSP en Servlet</li> </ul>		70%
--	--	-----