

## Template herkansingsfeedback

Leerlijn: Versie eindopdracht:  
 Naam student: Startmoment:  
 Studentnummer: Poging\*  
 \* Bij meerdere pogingen geef je in de tabellen aan welke feedback bij welke poging hoort.

Heb je jouw eindopdracht ingeleverd en heb je een onvoldoende gehaald? Dan verwerk je een extra hoofdstuk in jouw eindopdracht waarin je aantoont wat je precies hebt verbeterd naar aanleiding van de feedback van de docent.

Dit doe je per beoordelingscriterium, waarin je ook de feedback en originele cijfer benoemt. Gebruik hiervoor onderstaand template. In het geval dat je feedback *niet* verbeterd hebt, geef je dit ook aan.

|  |               |
|--|---------------|
| <b>Criterion: 1.</b> De student schrijft een functioneel ontwerp, met daarin een duidelijk gedefinieerd probleem en de manier waarop de applicatie dit probleem oplost, passende functionele en niet-functionele vereisten voor de beoogde oplossing (minimaal 50), een verzameling visuele inspiraties (minimaal 3 verschillende), use case-tabellen (minimaal 4) en wireframes (minimaal 5 pagina's).  |               |
| <b>Feedbackpoging docent 1:</b><br>Het functionele ontwerp is aangepast en er zijn extra componenten toegevoegd, maar het document is enigszins ongeorganiseerd. De functionele vereisten zijn niet genummerd en er zijn in totaal 36 vereisten. De vereisten zijn niet gecategoriseerd in functioneel/niet-functioneel en er zouden in totaal vijftig vereisten moeten zijn. Hoewel er talloze functionaliteiten worden genoemd, zijn ze prima, maar vul de vereisten aan met niet-functionele vereisten. De use cases zijn niet ontwikkeld volgens de geleerde structuur; bekijk hoofdstuk 10 van HTML&CSS voor de trigger en voorwaarden, maar duidelijke stappen van het belangrijkste successscenario en eventuele alternatieve scenario's ontbreken. Helaas zijn er geen wireframes opgenomen in het document. | [Cijfer]<br>4 |
| <b>Feedbackpoging docent 2:</b>  | [Cijfer]      |
| <b>Uitleg van verbeteringen student poging 1:</b><br>Bijgewerkt op basis van feedback.<br>- Nieuwe functionele en niet-functionele vereisten toegevoegd<br>- Wireframe projectlink en screenshots toegevoegd<br>- Elke vereiste genummerd  |               |
| <b>Toelichting op de verbeteringen bij poging 2 van de student:</b>  |               |

**Criterion. 2.** De student structureert de ontworpen functionaliteiten in een klassendiagram, rekening houdend met de vertaling van het klassendiagram naar een databasemodel (ORM) met behulp van Spring. Het klassendiagram is taal- en platformonafhankelijk en hoeft geen methoden te bevatten.

|   |               |
|---|---------------|
| <b>Feedbackpoging docent 1:</b> De UML-diagrammen in het technisch ontwerp zijn onleesbaar. Het aparte klassendiagrambestand is echter leesbaar, maar ziet er rommelig uit vanwege meerdere componenten. Sommige delen van het diagram zijn reverse-engineered vanuit de code. Het domeinmodel is enigszins uitgewerkt, maar de rol/naam van het systeembestand is niet duidelijk gespecificeerd. In het geval van de aanvraag en vacature wordt een datumveld verwacht, maar dit is niet aanwezig. | [Cijfer]<br>6 |
| <b>Feedbackpoging docent 2:</b>   | [Cijfer]      |
| <b>Uitleg van verbeteringen student poging 1:</b><br>Bijgewerkt op basis van feedback<br>UML-diagram aangepast<br>Diagramzichtbaarheid vergroot<br>Datumvelden toegevoegd   |               |
| <b>Toelichting op de verbeteringen bij poging 2 van de student:</b>   |               |

|   |               |
|---|---------------|
| <b>Criterium. 3.</b> De student legt nauwkeurig en logisch alle verschillende architectuurlagen in het technisch ontwerp vast met behulp van twee sequentiediagrammen, volgens de principes van de Software Development Life Cycle.   |               |
| <b>Feedbackpoging docent 1:</b><br>Mijn excuses voor de verwarring. De student zorgt ervoor dat de architectuurlagen, inclusief de controller-, service- en repositorylagen, correct worden weergegeven in de sequentiediagrammen. De lifelines zijn klasseobjecten en de berichten komen overeen met de oproepen die in de code worden gedaan. De diagrammen worden bijgewerkt om consistent te zijn met de code in termen van de oproepen die tussen objecten en lagen worden gedaan. | [Cijfer]<br>3 |
| <b>Feedbackpoging docent 2:</b>   | [Cijfer]      |
| <b>Uitleg van verbetering studenten poging 1:</b><br>Bijgewerkt op basis van feedback<br>Nieuw ontwerp sequentiediagram en klassediagrammen toegevoegd.   |               |
| <b>Toelichting op de verbeteringen bij poging 2 van de student:</b>   |               |

|   |               |
|---|---------------|
| <b>Criterium. 4.</b> De student ontwikkelt Java-applicaties en past op passende wijze objectgeoriënteerde programmeerstructuren (OOP) toe, zoals overerving, interfaces en abstracte klassen, afhankelijk van de complexiteit van het op te lossen probleem. Bovendien schrijft de student Java-code volgens de principes van Clean Code, met gebruikmaking van Design Patterns en met inachtneming van de SOLID-principes.   |               |
| <b>Feedbackpoging docent 1:</b><br>Bedankt voor het verstrekken van aanvullende details. De student zal het probleem met de centrale uitzonderingscontroller aanpakken en proberen te voorkomen dat er een interne serverfout 500 wordt geretourneerd. Ze zullen ervoor zorgen dat de profiel- en rolcontrollers correct worden geïmplementeerd met relevante functionaliteit. De student zal ook het aantal serviceklassen uitbreiden indien nodig. Validaties in de DTO-klassen worden onderhouden en ze zullen ook validatie van de controllers opnemen. De student zal ervoor zorgen dat de code leesbaar blijft. | [Cijfer]<br>6 |

|  |          |
|--|----------|
| <b>Feedbackpoging docent 2:</b>  | [Cijfer] |
| <b>Uitleg van verbetering studenten poging 1:</b><br>Bijgewerkt op basis van feedback<br>- Nieuwe klassen toegevoegd op verzoek<br>- Leesbaarheid van code verbeterd<br>- Fout beheerd |          |
| <b>Toelichting op de verbeteringen bij poging 2 van de student:</b>  |          |

|  |               |
|--|---------------|
| <b>Criterium. 5.</b> De student maakt gebruik van HTTP-methoden om deze te koppelen aan acties met de gegevens.  |               |
| <b>Feedbackpoging docent 1:</b><br>Bedankt voor het aanwijzen van de beperkingen in de RESTful-paden en -eindpunten. De student zal eraan werken om de paden RESTful te maken en ervoor te zorgen dat ze de juiste conventies volgen. Ze zullen gevallen behandelen waarin een niet-bestaand CV wordt aangevraagd om te downloaden of een ongeldige status wordt opgegeven, en een passend antwoord retourneren in plaats van een 500-fout. De REST-eindpunten worden uitgebreid om het ophalen van een lijst met gebruikers mogelijk te maken, en CRUD-ondersteuning wordt verbeterd door delete-eindpunten toe te voegen en het gebruik van het put-eindpunt uit te breiden naar geschikte locaties. | [Cijfer]<br>6 |
| <b>Feedbackpoging docent 2:</b>  | [Cijfer]      |
| <b>Uitleg van verbeteringen student poging 1:</b><br>Bijgewerkt op basis van feedback  |               |
| <b>Toelichting op de verbeteringen bij poging 2 van de student:</b>  |               |

|  |               |
|--|---------------|
| <b>Criterium. 6.</b> De student past autorisatie en authenticatie toe met Spring Security. Ze gebruiken verschillende componenten van Spring Boot, waaronder maar niet beperkt tot annotaties en applicatieconfiguratie.   |               |
| <b>Feedbackpoging docent 1:</b><br>Goed om te horen dat JWT-authenticatie en -autorisatie correct functioneren in de applicatie. De student heeft autorisatie voornamelijk geïmplementeerd met behulp van de @PreAuthorize-annotatie in de controllers. Het is goed om op te merken dat de onnodige antMatcher voor het /static-pad is geïdentificeerd. De beveiligingsconfiguratie is eenvoudig gehouden, omdat het gebruiksscenario zelf eenvoudig is. | [Cijfer]<br>7 |
| <b>Feedbackpoging docent 2:</b>  | [Cijfer]      |
| <b>Uitleg van verbeteringen student poging 1:</b><br>Bijgewerkt op basis van feedback  |               |

**Toelichting op de verbeteringen bij poging 2 van de student:**

**Criterium. 7.** De student leest en manipuleert data met behulp van SQL, JPA en Hibernate. Ze zorgen ook voor de juiste autorisatie en authenticatie binnen de database.

**Feedbackpoging docent 1:**

*Ik begrijp dat de standaarddatabasenaam is ingesteld op 'les17' (?). Het datamodel is relatief eenvoudig en de relaties tussen entiteiten zijn op de juiste manier vastgesteld. De student haalt gebruikers en rollen op uit de database, wat zorgt voor een goede integratie.*

[Cijfer]

7

**Feedbackpoging docent 2:**

[Cijfer]

**Uitleg van verbeteringen student poging 1:**

*Bijgewerkt op basis van feedback*

- Nieuwe entiteiten toegevoegd en relaties gecreëerd
- Databasestructuur en voorbeelddataset bijgewerkt

**Toelichting op de verbeteringen bij poging 2 van de student:**

**Criterium. 8.** De student bouwt de applicatie met behulp van semantische HTML-elementen, waarbij de nadruk ligt op het belang van een goede structurering. Ze gebruiken zelfgeschreven, schone en goed gestructureerde CSS door modulaire styling toe te passen op specifieke componenten. Deze aanpak zorgt voor onderhoudbaarheid en verbetert de algehele structuur en styling van de applicatie.

**Feedbackpoging docent 1:**

*Je hebt een paar semantische elementen toegevoegd, maar het zijn slechts secties en drie instanties van de hoofdtag. De hoofdtag mag slechts één keer in een applicatie worden gebruikt en ik raad aan om deze om je app.js-bestand te wikkelen. In je applicatie gebruik je Bootstrap, maar dit is niet toegestaan volgens de algemene voorwaarden. In je feedback voor herindiening vermeldde je dat try/catch-problemen zijn opgelost, maar deze worden niet behandeld in deze sectie van feedback. De projectstructuur is niet gewijzigd en het is nog steeds onduidelijk welke pagina's of componenten dat zijn. Er is in dit opzicht geen verbetering aangebracht door het in de duidelijke structuur te plaatsen die werd onderwezen.*

[Cijfer]

4

**Feedbackpoging docent 2:**

[Cijfer]

**Uitleg van verbeteringen student poging 1:**

*Bijgewerkt op basis van feedback*

**Toelichting op de verbeteringen bij poging 2 van de student:**

**Criterium 9.** De student schrijft schone, goed gestructureerde JavaScript-code die op de juiste manier gebruikmaakt van functies, arrays en objecten en die schone codeprincipes toepast.

|  |               |
|--|---------------|
| <b>Feedbackpoging docent 1:</b><br>Je vermeldt hier dat de consolelogs verdwenen zijn, maar je hebt andere belangrijke feedback, zoals het gebruik van klassecomponenten, React-hooks, etc., niet behandeld. In je herindieningsformulier vermeld je dat het is ontwikkeld met behulp van een Java en Spring Boot MVC-ontwerppatroon, maar dit heeft betrekking op een frontend-applicatie en het is onduidelijk hoe het zich verhoudt tot de bovenstaande punten.   | [Cijfer]<br>3 |
| <b>Feedbackpoging docent 2:</b>  | [Cijfer]      |
| <b>Uitleg van verbeteringen student poging 1:</b><br>Bijgewerkt op basis van feedback <ul style="list-style-type: none"> <li>- Er is een aparte implementatie voor database, frontend en backend technologie</li> <li>- Gebruikte objectgerichte concepten (d.w.z. klasse, object, overerving, eigenschappen, methode, etc.)</li> <li>- Sla vergelijkbare bestanden op in één pakket (d.w.z. klasse, afbeeldingen, componenten, etc.)</li> </ul> Structuur verbeterd |               |
| <b>Toelichting op de verbeteringen bij poging 2 van de student:</b>  |               |

|   |               |
|---|---------------|
| <b>Criterium. 10.</b> De student verdeelt zijn toepassing logisch in kleine herbruikbare componenten en zorgt ervoor dat gegevens correct worden doorgegeven via eigenschappen en callback-eigenschappen. |               |
| <b>Feedbackpoging docent 1:</b><br>Er zijn geen verbeteringen doorgevoerd. Er zijn nog steeds geen herbruikbare componenten gevonden in het project.  | [Cijfer]<br>2 |
| <b>Feedbackpoging docent 2:</b>   | [Cijfer]      |
| <b>Uitleg van verbeteringen student poging 1:</b><br>Bijgewerkt op basis van feedback   |               |
| <b>Toelichting op de verbeteringen bij poging 2 van de student:</b>   |               |

|   |               |
|---|---------------|
| <b>Criterium. 11.</b> De student voorziet componenten van de juiste interactie en externe gegevens door alle React life cycle hooks en AJAX correct te gebruiken.   |               |
| <b>Feedbackpoging docent 1:</b><br>Er zijn geen verbeteringen doorgevoerd. In het feedbackformulier voor herindiening vermeldt u alleen "verbeterd", maar u declareert nog steeds async-functies met params en u maakt geen gebruik van de levenscycli. | [Cijfer]<br>2 |
| <b>Feedbackpoging docent 2:</b>   | [Cijfer]      |
| <b>Uitleg van verbeteringen student poging 1:</b><br>Bijgewerkt op basis van feedback   |               |
| <b>Toelichting op de verbeteringen bij poging 2 van de student:</b>   |               |

|   |               |
|---|---------------|
| <b>Criterium. 12.</b> De student implementeert vier belangrijke functionaliteiten correct, waaronder het authenticatieproces van registratie/inloggen, uploaden en downloaden van bestanden en twee andere zelfontworpen, complete use cases.<br><i>Feedback: Er zijn geen wijzigingen aangebracht.</i> |               |
| <b>Feedbackpoging docent 1:</b><br><i>Er zijn geen wijzigingen doorgevoerd.</i>   | [Cijfer]<br>6 |
| <b>Feedbackpoging docent 2:</b>   | [Cijfer]      |
| <b>Uitleg van verbeteringen student poging 1:</b><br><i>Bijgewerkt op basis van feedback</i><br><ul style="list-style-type: none"> <li>- Proofer werkgever en kandidaat registratie toegevoegd</li> <li>- Correct in staat om cv te uploaden en downloaden</li> <li>- Use cases toegevoegd</li> </ul>   |               |
| <b>Toelichting op de verbeteringen bij poging 2 van de student:</b>   |               |

|  |               |
|--|---------------|
| <b>Criterium. 13.</b> De student schrijft een duidelijke installatiehandleiding waarmee derden de applicatie in een andere omgeving kunnen installeren. De handleiding bevat een stappenplan, een lijst met Feedbackpoging docent 2:, testgebruikers, gebruikersrollen, REST-eindpunten en een voorbeeld van JSON.   |               |
| <b>Feedbackpoging docent 1:</b><br><i>U hebt verbeteringen aangebracht in de installatiehandleiding door extra details toe te voegen, zoals een kort gedeelte over Feedbackpoging docent 2:. Dit is duidelijk en u hebt de scripts ook prominenter in het document gemarkeerd, wat netjes is gedaan. Het installatieproces is echter een beetje door elkaar, waardoor het onduidelijk is of de frontend of backend wordt geïnstalleerd. Het zou handig zijn om de installatie-instructies voor de backend en de frontend duidelijk te scheiden. Bovendien is het niet nodig om gebruikers te instrueren om React te installeren nadat npm install is uitgevoerd.</i> | [Cijfer]<br>4 |
| <b>Feedbackpoging docent 2:</b>  | [Cijfer]      |
| <b>Uitleg van verbeteringen student poging 1:</b><br><i>Bijgewerkt op basis van feedback</i><br><ul style="list-style-type: none"> <li>- Duidelijke instructie met screenshot meegeleverd</li> <li>- Gescheiden frontend en backend instructies</li> </ul>   |               |
| <b>Toelichting op de verbeteringen bij poging 2 van de student:</b>  |               |