Backend beoordelingscriteria V3.4

Totaal: 70/100

**De student vertaalt de casus (of een eigen idee voor een applicatie) naar een opsomming van tenminste 25 relevante functionele en niet-functionele eisen. 10**

Feedback

Je eisen zijn goed. Soms zijn je "omdat" clausules wat arbitrair in betekenis. Probeer termen als "om de klus te klaren" te vermijden.

**De student structureert in het technisch ontwerp de functionaliteiten in een overzichtelijke en logisch gestructureerde klassendiagram, waarbij hij kardinaliteiten bij de associaties aangeeft tussen verschillende klassen. Het klassendiagram is taal- en platformonafhankelijk en bevat geen methodes of architecturale lagen. 7.**

Feedback

Je Gender enumeratie staat 2 keer in je diagram. Je User-Authority relatie in je diagram komt niet overeen met je code. Je klassen hebben in je code nog een Id veld. Je klassendiagram heeft geen tekstuele toelichting nodig.

**De student legt op logische en correcte wijze alle verschillende architecturale lagen (controller, service en repository) vast in twee duidelijke en complete sequentiediagrammen. Deze sequentiediagrammen bevatten klasse- en methodenamen die overeenkomen met die in het ingeleverde project. 7**

Feedback

Door je kleurpatroon is de scheidslijn van je alt frame bijna onzichtbaar. De database hoeft niet in je sequentiediagram. Het tweede alt-blok in je tweede diagram (usernameexists) is niet juist.

**De student implementeert 4 belangrijke kern- functionaliteiten op correcte en kwalitatieve wijze. 7**

Feedback

Deleten van een Character gaat niet soepel. Waarom kan ik "getcharacterasc" niet zonder naam aanroepen? Je hebt wel voldoende functionaliteit, maar veel functies zijn nog oppervlakkig uitgewerkt.

**De student maakt een kwalitatieve Java-applicatie waarbij hij logica implementeert door het gebruik van attributen, methoden, overerving, interfaces en abstracte klassen. 6**

Feedback

Je basis java kennis is voldoende. Verder laat je, behalve je Gender enum, nog weinig verdiepende kennis zien.

**De student voert minimaal 2 geslaagde integratie-tests uit door de application context van zijn applicatie te testen met Spring Boot test en WebMvc. Tevens test de student twee of drie klassen uit de service laag met een line coverage van 100% door middel van minimaal 10 unit-tests, gebruikmakend van de drie A’s. 6**

Feedback

Je laat goed zien dat je kunt testen. Je gebruikt soms "asserNotEquals" waar je dat niet moet doen. De test slaagt wel, maar geeft dus eigenlijk een "false positive". Het verplaatsen van logica uit de service naar de controller is natuurlijk ook geen oplossing, probeer dat nog eens aan het werk te krijgen.

**De student past exception handling op effectieve wijze toe bij het beheren van mogelijke errors binnen de applicatie. 9**

Feedback

Je vangt de meest voorkomende errors af. Voor je validatie is er nog ruimte voor verbetering, maar dat geef je zelf ook al aan.

**De student beheert zijn code met correct gebruik van Git om met versiebeheer de voortgang van het project vast te leggen. De student werkt op verschillende branches, maakt kleine commits met compacte en zinvolle commit-messages, maakt pull requests en merged regelmatig. 9**

Feedback

Je hebt voldoende commits. Je commit messages zijn duidelijk. Je hebt gebruik gemaakt van branches, maar dit mag je nog wat meer op een feature basis doen. Je branches zijn gemerged.

**De student maakt logisch gebruik van DTO’s waarmee validatie wordt toegepast op data van en naar de database. Tevens stelt hij relevante query’s op om testdata uit de database op te halen. 5**

Feedback

Je gebruikt DTO's op de juiste plekken. Je gebruikt validatieregels en valideert deze ook. Je DTO's bevatten entity modellen, waardoor je het nut van de DTO breekt. Je hebt voldoende test data en je maakt mooit gebruikt van eigen jpa methodes.

**De student past, door correct gebruik van Spring Security, authenticatie toe op zowel de gebruiker die in de database is opgeslagen als op het gecodeerde wachtwoord. Daarnaast past hij autorisatie toe op de endpoints, door adequaat gebruik te maken van een JWT. 9**

Feedback

Je hebt spring security goed toegepast, het kan alleen wat overzichtelijker en logischer opgeschreven worden. Je password wordt encrypted opgeslagen en wordt niet meegegeven in je DTO. Een gebruiker kan de gegevens van andere gebruikers opvragen, maar dit geef je ook al zelf aan als verbeter punt.

**De student implementeert een kwalitatieve webservice volgens de RESTful richtlijnen, waarbij hij HTTP-methodes gebruikt om de vertaalslag te maken naar acties met de data. 5**

Feedback

Je paden zijn niet altijd helemaal restful, je gebruikt vaak woordjes in je pad die geen nut hebben, zoals "post" in een post-request, "disassociate" in een delete-request en "upload/download" in je image requests. Je geeft niet altijd de juiste HTTP status terug, met name voor DELETE.

**De student past de principes van Clean Code en SOLID correct toe. 6**

Feedback

Je naamgeving in je modellen is soms onnodig lang, zoals "characterAliasName" ipv "aliesName" in de Character klasse. Je methode namen zijn beschrijvend. In je FileService gebruikt je in feite 3 keer dezelfde methode voor een andere repository, dat kan meer DRY. Op andere plekken heb je wel helper methodes gebruikt. Voor elk zoek criterium heb je een eigen methode gedefinieers, waar dat ook in 1 methode zou kunnen (aangezien het hetzelfde pad is).

**De student schrijft op basis van de Software Development Life Cycle een duidelijke installatiehandleiding waarmee de applicatie door een collega-developer kan worden geïnstalleerd. Deze is voorzien van een stappenplan, een lijst van benodigdheden, testgebruikers, userrollen en rest-endpoints. 8**

Feedback

Je beschrijft het configureren van de database in zowel IntelliJ als PgAdmin. De configuratie in je application.properties is niet helemaal duidelijk genoeg. In je postmand export had je beter gebruik kunnen maken van mappen in plaats van lege OPTIONS requests.

**De student schrijft een volledig, goed gestructureerd en duidelijk verantwoordingsdocument waarin hij een beargumenteerd overzicht geeft van minimaal 5 toegepaste technieken. 6**

Feedback

Je hebt 5 technische keuzes verantwoord, maar je verantwoording is niet altijd heel sterk of uitgebreid. Je beschrijft een ManyToMany in plaats van een koppeltabel, maar er is natuurlijk altijd een koppeltabel. De eis van 7 entiteiten, was een eis van de ideefase, niet van de eindopdracht. Het gebruik van "api/v1" is wel een algemene best practice in het bedrijfsleven. Je doorgroeimogelijkheden zijn realistisch en duidelijk. Het is fijn om te lezen dat je een goede ervaring bij Novi hebt gehad en dat je jezelf een goede leerstrategie hebt kunnen aanmeten.