

Faculteit Bedrijf en Organisatie

Valentin Vaerwyckweg 1

9000 GENT

ACADEMIEJAAR 2021-2022

OPLEIDING TOEGEPASTE INFORMATICA

PROJECTWERK

Opleidingsonderdeel 'Software Development Project II'

Fluvius Dashboard

GROEP: 01

STUDENTEN:

Rayme Emin

Christof Remeysen

Henri De Bel

BEGELEIDER: *Guy Dekoning*

Inhoud

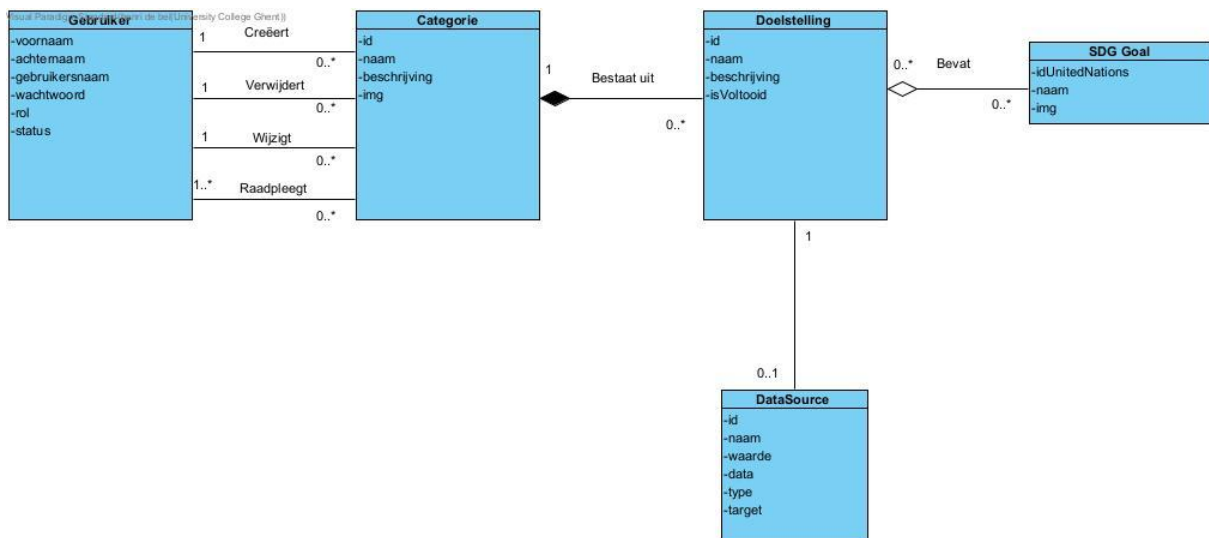
Domeinmodel	5
Use-Cases	5
Toevoegen Categorie	5
Primaire actoren	5
Stakeholders	6
Precondities	6
Postcondities	6
Normaal verloop	6
Alternatieve verlopen	6
Domeinregels	6
Op te klaren punten	6
Wijzigen Doelstelling	7
Primaire actoren	7
Stakeholders	7
Precondities	7
Postcondities	7
Normaal verloop	7
Alternatieve verlopen	7
Domeinregels	8
Op te klaren punten	8
Registreren Gebruiker	8
Primaire Actoren	8
Stakeholders	8
Precondities	8
Postcondities	8
Normaal verloop	9
Alternatieve verlopen	9
Domeinregels	9
Op te klaren punten	9
Aanpassen Rol	9
Primaire Actoren	9

Stakeholders	10
Precondities	10
Postcondities	10
Normaal verloop	10
Alternatieve verlopen	10
Domeinregels	10
Op te klaren punten	10
Verwijder datasource	10
Primaire actoren	10
Stakeholders	11
Precondities	11
Postcondities	11
Normaal verloop	11
Alternatieve verlopen	11
Domeinregels	11
Op te klaren punten	11
Operation contracts	12
Toevoegen Categorie	12
Wijzigen Doelstelling	12
Registreren Gebruiker	12
Aanpassen Rol	12
Verwijder datasource	13
Sequence Diagrams	13
Toevoegen Categorie	13
Wijzigen Doelstelling	14
Registreren Gebruiker	15
Aanpassen Rol	15
Verwijder datasource	16
Non Functional Requirements	17
Toevoegen Categorie	17
Wijzigen Categorie	19
Registreren Gebruiker	21

Aanpassen Rol	22
Verwijder categorie	23
Testplan	25
Toevoegen Categorie	25
Resources	25
Features	25
Methoden	26
Assumpties	28
Risico's	28
Test matrix	28
Wijzigen Doelstelling	29
Resources	29
Features	29
Methoden	29
Assumpties	31
Risico's	31
Test matrix	32
Registreren Gebruiker	32
Resources	32
Features	32
Methoden	33
Assumpties	35
Risico's	35
Test matrix	35
Aanpassen Rol	36
Resources	36
Features	36
Methoden	36
Assumpties	38
Risico's	38
Test matrix	38
Verwijder datasource	39

Resources	39
Features	39
Methoden	39
Assumpties	40
Risico's	40
Retrospectief	42
Henri De Bel	42
Rayme Emin	42
Christof Remeysen	44
Persoonlijke ervaringen	44
Technische ervaringen	45
Slotwoord	45

1. Domeinmodel



2. Use-Cases

a. Toevoegen Categorie

Primaire actoren

MVO-coördinator.

Stakeholders

Directeurs, Managers en Externe stakeholders.

Precondities

- De primaire actor is aangemeld.

Postcondities

- De categorie is toegevoegd.

Normaal verloop

1. De coördinator wenst een categorie toe te voegen.
2. Het systeem vraagt: naam, beschrijving, afbeelding.
3. De coördinator vult in: naam, beschrijving, afbeelding.
4. Het systeem valideert volgens DR_CATEGORIE.
5. Het systeem registreert de categorie.
6. Het systeem toont een gepaste melding.

Alternatieve verlopen

- 6.A. Het systeem detecteert dat validatie faalt.
 - A.1. Het systeem toont een gepaste melding.
 - A.2. Ga naar stap 3 van het normaal verloop.

Domeinregels

DR_CATEGORIE

- De naam is uniek in het systeem.
- De naam bestaat uit minstens 2 en hoogstens 50 karakters.
- De beschrijving bestaat uit minstens 5 en hoogstens 250 karakters.

Op te klaren punten

- De eindklant dient de categorieën aan te leveren.

- De eindklant dient te bevestigen of een doelstelling verplicht is, alsook of deze tot meerdere categorieën mag behoren.

b. Wijzigen Doelstelling

Primaire actoren

MVO-coördinator.

Stakeholders

Directeurs, Managers en Externe stakeholders.

Precondities

- De coördinator is aangemeld.
- De primaire actor heeft een coördinator rol.
- De applicatie bevat minstens één categorie met doelstelling.

Postcondities

- De doelstelling is gewijzigd.

Normaal verloop

1. De coördinator wenst een doelstelling te wijzigen
2. Het systeem toont de gegevens van de doelstelling.
3. De coördinator wijzigt: naam, beschrijving, status, SDGs entiteit(en).
4. De coördinator raadpleegt datasource.
5. Het systeem valideert volgens DR_DOELSTELLING.
6. Het systeem registreert de wijzigingen.
7. Het systeem toont een gepaste melding.

Alternatieve verlopen

- 4.A.1. De coördinator voegt datasource toe.
- 4.B.1. De coördinator verwijdert datasource.
- 4.C.1. De coördinator wijzigt datasource.

5. A. Het systeem detecteert dat validatie faalt.
- A.1. Het systeem toont een gepaste melding.
 - A.2. Ga naar stap 3 van het normaal verloop.

Domeinregels

DR_DOELSTELLING

- Alle velden zijn verplicht in te vullen, SDGs optioneel.
- De naam is uniek in het systeem.
- De naam bestaat uit hoogstens 50 karakters.
- De beschrijving bestaat uit hoogstens 250 karakters.
- De input bestaat uit alfanumerieke karakters. Speciale tekens zijn toegelaten.
- Een doelstelling kan slechts tot één categorie behoren.
- Een doelstelling kan slechts één datasource bevatten.

Op te klaren punten

- De eindklant dient de doelstellingen aan te leveren.
- De eindklant dient te specificeren hoe hij drempelwaarden ziet. We hebben voorlopig een statusveld voorzien dat aangeeft of het doel is bereikt.
- De eindklant dient te bevestigen of een datasource verplicht is, alsook of deze tot meerdere doelstellingen mag behoren.
- De eindklant dient te bevestigen of een doelstelling meerdere data sources kan bevatten.

c. Registreren Gebruiker

Primaire Actoren

Gebruiker

Stakeholders

Directeurs, Managers en Externe stakeholders.

Precondities

/

Postcondities

- Het systeem heeft de gebruiker geregistreerd

Normaal verloop

1. De gebruiker wenst zich te registreren
2. Het systeem vraagt gebruikersnaam, voornaam, familienaam, wachtwoord
3. De gebruiker voert de gegevens in
4. Het systeem valideert volgens DR_REGISTREREN
5. Het systeem registreert de succesvolle registratie volgens DR_REGISTRATIE
6. Het systeem toont gepaste melding

Alternatieve verlopen

4A. Het systeem detecteert dat DR_REGISTREREN niet gevolgd werd

A.1. Het systeem toont een gepaste melding

A.2. Ga naar stap 3 van het normaal verloop

Domeinregels

DR_REGISTREREN

- Wachtwoord moet bestaan uit minstens 5 en hoogstens 20 karakters
- Gebruikersnaam moet bestaan uit minstens 2 en hoogstens 20 karakters
- Voornaam moet bestaan uit minstens 2 en hoogstens 30 karakters
- Familienaam moet bestaan uit minstens 2 en hoogstens 30 karakters
- De gebruikersnaam is nog niet in gebruik

DR_REGISTRATIE

- Succesvolle poging: tijdstip, gebruikersnaam, voornaam, familienaam, default rol stakeholder

Op te klaren punten

- Een Admin mag een rol aanpassen

d. Aanpassen Rol

Primaire Actoren

Admin

Stakeholders

Directeurs, Managers en Externe stakeholders.

Precondities

- Er bestaat een gebruiker
- De primaire actor is een Admin

Postcondities

- Het systeem heeft de rol van een gebruiker aangepast

Normaal verloop

1. De admin selecteert een gebruiker
2. Het systeem toont alle mogelijke rollen
3. De admin selecteert een gewenste rol
4. Het systeem registreert de succesvolle rol verandering
5. Het systeem toont een gepaste melding

Alternatieve verlopen

/

Domeinregels

DR_GEBRUIKERSROLLEN

- Mogelijke rollen: ADMIN, MANAGER, MVO COÖRDINATOR, STAKEHOLDER

DR_AANPASSINGROL

- Primaire actor moet een admin zijn
- Er moet een andere gebruiker dan de Primaire actor beschikbaar zijn
- Admin kan niet de rol van een andere Admin wijzigen

Op te klaren punten

/

e. Verwijder datasource

Primaire actoren

MVO-coördinator.

Stakeholders

Managers.

Precondities

- De coördinator is aangemeld.
- De primaire actor heeft een coördinator rol.
- De applicatie bevat minstens één datasource.

Postcondities

- De datasource is verwijderd.

Normaal verloop

1. De coördinator wenst een datasource te verwijderen.
2. Het systeem vraagt om de verwijdering actie te bevestigen.
3. De coördinator bevestigt de verwijdering actie.
4. Het systeem valideert de verwijdering volgens DR_DATASOURCE.
5. Het systeem registreert de verwijdering.
6. Het systeem toont een gepaste melding.

Alternatieve verlopen

2A. De coördinator weigert

Use case stopt zonder bereiken postconditie.

4.A. validatie faalt DR_DATASOURCE

A.1. Het systeem toont een gepaste melding.

Use case stopt zonder bereiken postconditie.

Domeinregels

DR_DATASOURCE

- Datasource kan niet verwijderd worden als die aan minstens 1 MVO doelstelling gekoppeld is.

Op te klaren punten

/

3. Operation contracts

a. Toevoegen Categorie

Contract: categorie aanmaken	
Operatie	pasRolAan(userId, rol)
Cross Reference	UC 2: toevoegen categorie
Precondities	> De huidige gebruiker is MVO-coördinator
Postcondities	> De categorie is toegevoegd

b. Wijzigen Doelstelling

Contract: wijzig doelstelling	
Operatie	wijzigDoelstelling(naam, beschrijving, status, SDGs)
Cross Reference	UC 14: wijzig doelstelling
Precondities	> Een MVO doelstelling is geselecteerd > De huidige gebruiker is MVO-coördinator
Postcondities	> De geselecteerde MVO doelstelling is gewijzigd

c. Registreren Gebruiker

Contract: aanpassen rol	
Operatie	registreer(gebruikersnaam, voornaam, familienaam, wachtwoord)
Cross Reference	UC 5: registreren gebruiker
Precondities	/
Postcondities	> De gebruiker is gerigistreerd

d. Aanpassen Rol

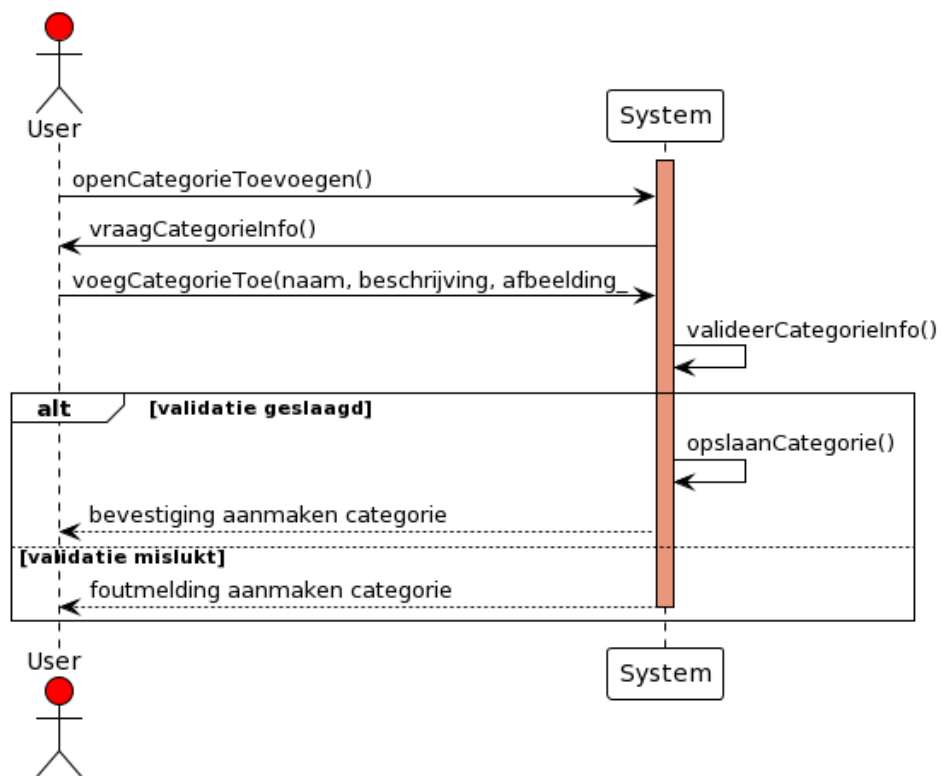
Contract: aanpassen rol	
Operatie	pasRolAan(userId, rol)
Cross Reference	UC 7: aanpassen rol
Precondities	> Een gebruiker is geselecteerd > De huidige gebruiker is Admin
Postcondities	> De Rol van de geselecteerde gebruiker is aangepast

e. Verwijder datasource

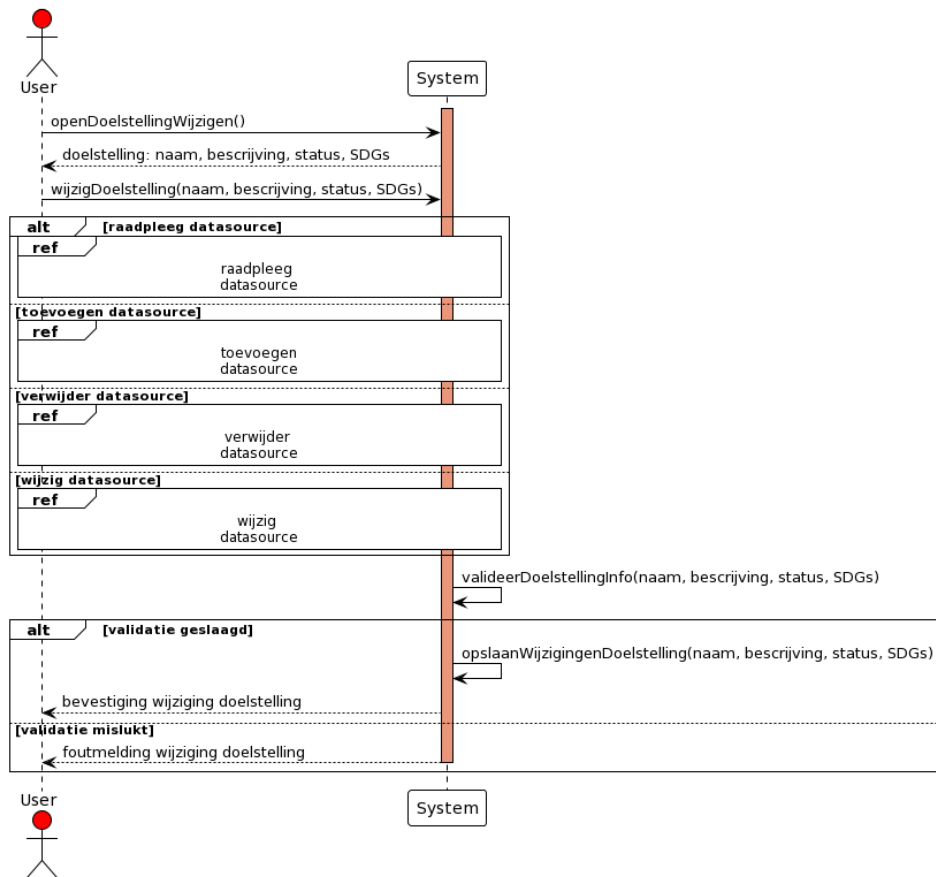
Contract: verwijder datasource	
Operatie	verwijderDatasource(datasourceId)
Cross Reference	UC 11: verwijderen datasource
Precondities	> Een datasource is geselecteerd > De huidige gebruiker is MVO-coördinator
Postcondities	> De geselecteerde datasource is verwijderd

4. Sequence Diagrams

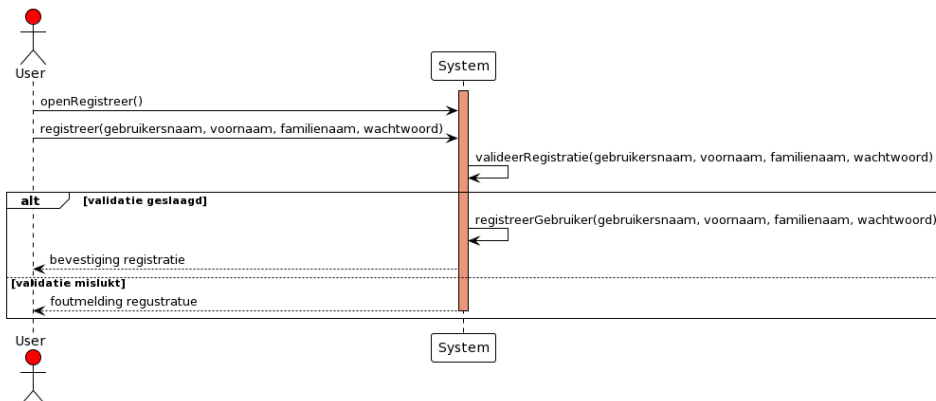
a. Toevoegen Categorie



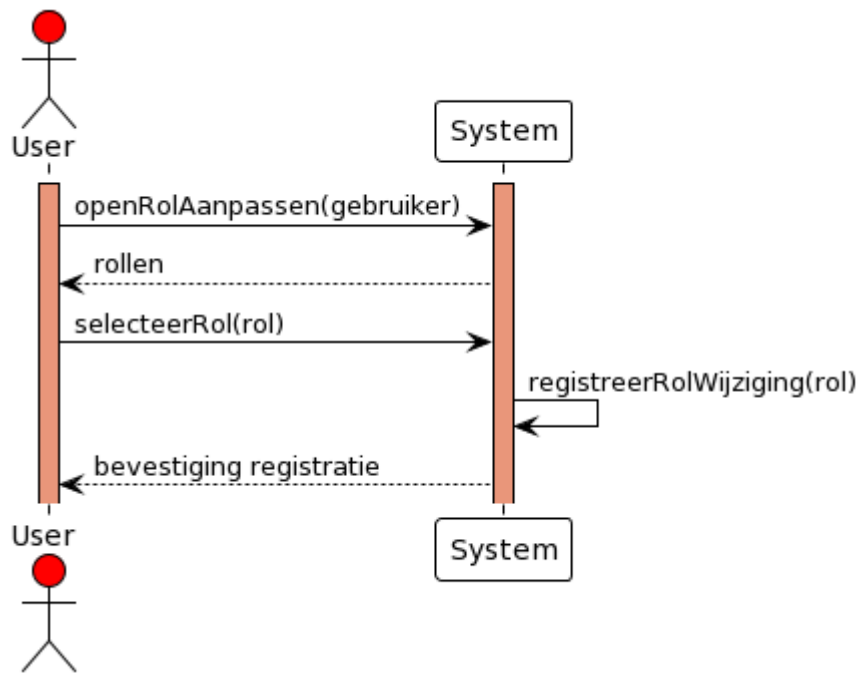
b. Wijzigen Doelstelling



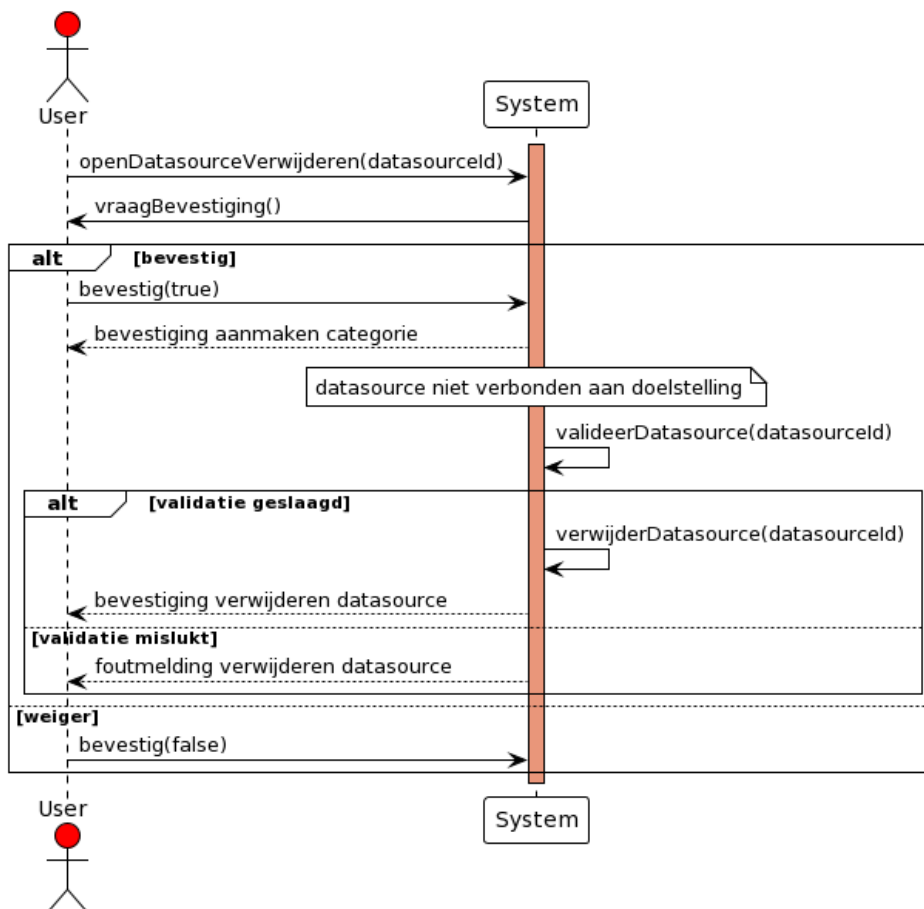
c. Registreren Gebruiker



d. Aanpassen Rol



e. Verwijder datasource



5. Non Functional Requirements

a. Toevoegen Categorie

NFR	Functionele geschiktheid
Indicator	Functionele correctheid
Meetvoorschrift	4 MVO coördinatoren voegen elks op een geldige manier 5 categorieën toe. Ze vullen deze gegevens in: naam, beschrijving, afbeelding, doelstelling(en).
Norm	1) De back-end van de applicatie slaat de gegevens van een nieuwe categorie in alle gevallen binnen de 3 seconden correct op. 2) De front-end van de applicatie toont de nieuwe toegevoegde categorie. De state van de applicatie klopt in 100% van de gevallen.

NFR	Prestatie-efficiëntie
Indicator	Capaciteit
Meetvoorschrift	4 MVO coördinatoren voegen elks op een geldige manier 5 categorieën toe. Ze vullen deze gegevens in: naam, beschrijving, afbeelding, doelstelling(en). Terwijl ze dit doen, bezoeken 5000 interne en externe gebruikers de applicatie gelijktijdig.
Norm	De back-end van de applicatie slaat de gegevens van een nieuwe categorie in alle gevallen binnen de 5 seconden correct op.

NFR	Bruikbaarheid
-----	---------------

Indicator	Leerbaarheid
Meetvoorschrift	4 MVO coördinatoren voegen elks voor de eerste keer 5 categorieën via het overzicht toe.
Norm	Alle startende MVO coördinatoren slagen er binnen de 30 minuten in om minstens 3 nieuwe categorieën aan te maken.

NFR	Bruikbaarheid
Indicator	Voorkomen gebruikersfouten
Meetvoorschrift	1 MVO coördinator maakt een nieuwe categorie aan, waarbij hij 3 keer een naam tracht in te geven waarvoor er reeds een categorie bestaat. Daarna tracht hij de categorie met een te lange beschrijving en zonder afbeelding toe te voegen.
Norm	Het systeem toont meteen een waarschuwing bij elke foutieve invoer en actie.

NFR	Betrouwbaarheid
Indicator	Foutbestendigheid
Meetvoorschrift	1 MVO coördinator geeft een naam voor een categorie in die reeds in gebruik is. Vervolgens probeert hij de categorie met een te lange beschrijving en zonder afbeelding toe te voegen.
Norm	Het systeem valt niet uit door de foutief ingevulde gegevens en gebruikersacties.

NFR	Betrouwbaarheid
Indicator	Maturiteit
Meetvoorschrift	4 MVO coördinatoren voegen elks op een geldige manier 5 categorieën toe. Ze vullen deze gegevens in: naam, beschrijving, afbeelding, doelstelling(en).

Norm	De back-end van de applicatie handelt de verzoeken in 100% van de gevallen na enkele seconden consistent af.
-------------	--

NFR	Betrouwbaarheid
Indicator	Beschikbaarheid
Meetvoorschrift	4 MVO coördinatoren voegen elks op een geldige manier 5 categorieën toe. Ze vullen deze gegevens in: naam, beschrijving, afbeelding, doelstelling(en). Terwijl ze dit doen, bezoeken 5000 interne en externe gebruikers de applicatie gelijktijdig.
Norm	De front- en back-end van de applicatie handelen de verzoeken in 95% van de gevallen zonder uitval en trage user experience af.

b. Wijzigen Categorie

NFR	Functionele geschiktheid
Indicator	Functionele correctheid
Meetvoorschrift	4 MVO coördinatoren wijzigen elks 1 categorie. Ze passen deze gegevens aan: naam, beschrijving, afbeelding, doelstelling(en).
Norm	<p>1) De back-end van de applicatie slaat de gegevens van een gewijzigde categorie in alle gevallen binnen de 3 seconden correct op.</p> <p>2) De front-end van de applicatie toont de gegevens van de gewijzigde categorie op een juiste manier. De state van de applicatie klopt in 100% van de gevallen.</p>

NFR	Prestatie-efficiëntie
------------	------------------------------

Indicator	Capaciteit
Meetvoorschrift	4 MVO coördinatoren wijzigen elks op een geldige manier 1 categorie. Ze passen deze gegevens aan: naam, beschrijving, afbeelding, doelstelling(en). Terwijl ze dit doen, bezoeken 5000 interne en externe gebruikers de applicatie gelijktijdig.
Norm	De back-end van de applicatie slaat de gegevens van een gewijzigde categorie in alle gevallen binnen de 5 seconden correct op.

NFR	Bruikbaarheid
Indicator	Leerbaarheid
Meetvoorschrift	4 MVO coördinatoren wijzigen elks voor de eerste keer 1 categorie via het overzicht.
Norm	Alle startende MVO coördinatoren slagen er binnen de 10 minuten in om 1 categorie te wijzigen.

NFR	Bruikbaarheid
Indicator	Voorkomen gebruikersfouten
Meetvoorschrift	1 MVO coördinator wijzigt een categorie, waarbij hij 3 keer de naam tracht aan te passen in één waarvoor er reeds een categorie bestaat. Daarna tracht hij de categorie met een te lange beschrijving te updaten.
Norm	Het systeem toont meteen een waarschuwing bij elke foutieve invoer en actie.

NFR	Betrouwbaarheid
Indicator	Foutbestendigheid

Meetvoorschrift	1 MVO coördinator past een categorienaam aan in één die reeds in gebruik is. Vervolgens probeert hij de categorie met een te lange beschrijving te updaten.
Norm	Het systeem valt niet uit door de foutief ingevulde gegevens en gebruikersacties.

NFR	Betrouwbaarheid
Indicator	Maturiteit
Meetvoorschrift	4 MVO coördinatoren wijzigen elks 1 categorie. Ze passen deze gegevens aan: naam, beschrijving, afbeelding, doelstelling(en).
Norm	De back-end van de applicatie handelt de verzoeken in 100% van de gevallen na enkele seconden consistent af.

NFR	Betrouwbaarheid
Indicator	Beschikbaarheid
Meetvoorschrift	4 MVO coördinatoren wijzigen elks 1 categorie. Ze passen deze gegevens aan: naam, beschrijving, afbeelding, doelstelling(en). Terwijl ze dit doen, bezoeken 5000 interne en externe gebruikers de applicatie gelijktijdig.
Norm	De front- en back-end van de applicatie handelen de verzoeken in 95% van de gevallen zonder uitval en trage user experience af.

c. Registreren Gebruiker

Indicator	Voorkomen gebruikersfouten
------------------	----------------------------

Meetvoorschrift	Een gebruiker probeert zich te registreren met een correcte gebruikersnaam en wachtwoord, maar voornaam en familienaam werd niet ingevuld.
Norm	Het systeem toont meteen een foutboodschap bij elke foutieve of lege invoer.

NFR	Prestatie-efficiëntie
Indicator	Foutbestendigheid
Meetvoorschrift	10 gebruikers proberen elk zich te registreren 5 gebruiken telkens een correcte gebruikersnaam, voornaam, familienaam en wachtwoord de andere 5 telkens een correcte voornaam, familienaam en wachtwoord, maar de gebruikernaam is al in gebruik.
Norm	Het systeem voert voor elke gebruiker met correcte gegevens de registratie correct uit en deze registratie zal worden geregistreerd. De gebruiker krijgt een alert te zien dat zijn taak succesvol werd uitgevoerd. Voor de 5 gebruikers met een foutief gebruikersnaam wordt nooit een registratie effectief uitgevoerd maar telkens een passende alert getoond zonder dat de applicatie uitvalt.

NFR	Functionele geschiktheid
Indicator	Functionele correctheid
Meetvoorschrift	10 gebruikers proberen elk met correcte gegevens zich te registreren.
Norm	alle 10 de succesvolle registratiepogingen werden door het systeem geregistreerd

d. Aanpassen Rol

NFR	Bruikbaarheid
Indicator	Voorkomen gebruikersfouten

Meetvoorschrift	Een admin probeert zijn eigen rol te veranderen naar MVO_Coordinator, maar hij is als enige als admin geregistreerd in het systeem. Er moet altijd minimaal 1 gebruiker aanwezig zijn als admin in het systeem.
Norm	Het systeem toont meteen een foutboodschap dat er minimaal een admin moet aanwezig zijn in het systeem.

NFR	Prestatie-efficiëntie
Indicator	Foutbestendigheid
Meetvoorschrift	Een MVO_Coordinator probeert zijn eigen rol te veranderen naar admin. Alleen een admin kan een andere gebruiker een adminrol geven.
Norm	De MVO_coordinator krijgt een alert te zien dat zijn taak onsuccesvol werd uitgevoerd.

NFR	Betrouwbaarheid
Indicator	Beschikbaarheid
Meetvoorschrift	Gebruiker A heeft een stakeholder. Gebruiker B heeft een Manager rol. Gebruiker C heeft een MVO_coordinator rol.
Norm	Gebruikers A en B hebben de optie om een rol van een gebruiker te wijzigen niet gebruiker B wel.

e. Verwijder categorie

NFR	Functionele geschiktheid
Indicator	Functionele correctheid
Meetvoorschrift	4 MVO coördinatoren verwijderen elks 1 categorie.
Norm	<p>1) De back-end van de applicatie verwijdert de gegevens van een categorie correct binnen de 3 seconden, inclusief bijbehorende doelstellingen en datasources.</p> <p>2) De front-end van de applicatie toont de gegevens van een verwijderde categorie niet meer, maar wel deze van de andere categorieën. De state van de applicatie klopt in 100% van de gevallen.</p>

NFR	Prestatie-efficiëntie
Indicator	Capaciteit
Meetvoorschrift	4 MVO coördinatoren verwijderen elks 1 categorie. Terwijl ze dit doen, bezoeken 5000 interne en externe gebruikers de applicatie gelijktijdig.
Norm	De back-end van de applicatie verwijdert in alle gevallen op aangeven de gegevens van een categorie binnen de 5 seconden, inclusief bijbehorende doelstellingen en datasources.

NFR	Bruikbaarheid
Indicator	Leerbaarheid
Meetvoorschrift	4 MVO coördinatoren verwijderen elks voor de eerste keer 1 categorie via het overzicht.
Norm	Alle startende MVO coördinatoren slagen er binnen de 10 minuten in om 1 categorie te verwijderen.

NFR	Betrouwbaarheid
Indicator	Maturiteit
Meetvoorschrift	4 MVO coördinatoren verwijderen elks 1 categorie.
Norm	De back-end van de applicatie handelt de verzoeken in 100% van de gevallen na enkele seconden consistent af.

NFR	Betrouwbaarheid
Indicator	Beschikbaarheid

Meetvoorschrift	4 MVO coördinatoren verwijderen elks 1 categorie. Terwijl ze dit doen, bezoeken 5000 interne en externe gebruikers de applicatie gelijktijdig.
Norm	De front- en back-end van de applicatie handelen de verzoeken in 95% van de gevallen zonder uitval en trage user experience af.

6. Testplan

a. Toevoegen Categorie

Resources

De onderstaande tabel geeft aan wie de testen voor deze use case zal uitvoeren.

Tester	% Werkuren
Rayme Emin	33% (2u), Java, React
Henri De Bel	33% (2u), React
Christof Remeysen	33% (2u), React en Java

Features

De ontwikkelde features voor deze use cases staan hieronder beschreven.

Feature	Beschrijving
Voeg een categorie toe knop	De overzichtspagina heeft een knop voor het creëren van een categorie die via de REST API wordt opgeslagen in de databank.
Categorie creatie pagina	Een pop-up geeft velden weer voor het invullen van : naam, beschrijving en afbeelding.
Routing	De routing zorgt ervoor dat je van de overzichtspagina naar de bijbehorende categorie creatie pagina kan navigeren en omgekeerd.
REST API (React) JPA (Java)	De REST API/JPA ontvangt en behandelt verzoeken van de front-end. De REST API/JPA bevat seeds met mock-data;
CDN platform	Het CDN platform bevat de gebruikte afbeeldingen (React).
Resources folder	Deze folder bevat de afbeeldingen van een categorie (Java).
MySQL databank	Deze bevat de data van een categorie, inclusief afbeeldingen (Java).
CSS stijlen	De CSS stijlen zorgen voor de opmaak van een scherm: <ul style="list-style-type: none"> • React: Chakra, CSS. • Java: JavaFX, CSS.

Alerts	Er worden alerts voorzien om de gebruiker te informeren over het slagen/falen van de uitgevoerde taak.
Foutmeldingen	Er worden foutmeldingen voorzien bij de inputvelden volgens de domeinregels.

Methoden

Deze sectie geeft aan hoe de hierboven vernoemde features worden getest.

	Type test	Beschrijving
Voeg een categorie toe knop	Unit en component, Functioneel	<p>De unit en component testen verlopen automatisch. Ze worden enkel voor Java (JUnit) voorzien.</p> <p>De functionele testen verlopen exploratief, zowel voor Java als React. De testers, waarvan er minstens twee een verschillende rol hebben, klikken op de Voeg een categorie toe knop en controleren of de categorie creatie pop-up wordt weergegeven. Voor React gebeurt dit semi automatisch met Cypress.</p>
Categorie creatie pagina	Unit en component, Functioneel, Load of performance	<p>De unit en component testen verlopen automatisch. Ze worden enkel voor Java (JUnit) voorzien.</p> <p>De functionele testen verlopen exploratief, zowel voor Java als React. De tester opent de Categorie creatie pagina en vult vervolgens alle verplichte velden in. Vervolgens de tester kan kiezen om een categorie toe te voegen door op een voeg toe knop te drukken of te annuleren. Voor React gebeurt dit semi automatisch met Cypress.</p>
Routing	Functioneel Performance	<p>De functionele testen verlopen exploratief, zowel voor Java als React. De tester controleert of de juiste pagina's bij navigatie openen.</p> <p>Voor Java, verifieert de tester de compilatie duur en snelheid van laden op exploratieve wijze. Voor React, zet de tester hiervoor Google Lighthouse en React Profiler in.</p>

REST API	Integratie Performance	De integratietest verloopt exploratief. Ze gebeurt enkel voor React. De tester verifieert op het zicht of de data goed en snel genoeg vanuit de Nest.js API wordt opgeslagen en geladen.
CDN platform	Integratie Performance	De integratietest verloopt exploratief. Ze gebeurt enkel voor React. De tester verifieert op het zicht of afbeeldingen goed en snel genoeg vanuit Cloudinary worden ingelezen en opgeslagen.
Resources folder	Functioneel Performance	De tester verifieert exploratief of afbeeldingen goed en snel genoeg vanuit de resources folder op de overzichtspagina laden.
MySQL databank	Integratie Performance	De integratietest verloopt exploratief. Ze gebeurt enkel voor Java. De tester verifieert op het zicht of de data en afbeeldingen goed en snel genoeg in de MySQL databank worden opgeslagen.
CSS stijlen	Functioneel Usability	De tester verifieert exploratief of de CSS stijlen goed inladen, zowel voor Java als React. Daarbij let hij op gebruiksvriendelijke aspecten, zoals goed aanklikbare knoppen.
Alerts	Unit en component Functioneel	De unit en component testen verlopen automatisch. Ze worden enkel voor Java (JUnit) voorzien. De tester verifieert exploratief of de juiste alerts getriggerd worden bij het succesvol of niet succesvol toevoegen van een categorie.
Foutmeldingen	Unit en component Functioneel	De unit en component testen verlopen automatisch. Ze worden enkel voor Java (JUnit) voorzien. De tester verifieert exploratief of de juiste foutmeldingen getriggerd worden bij een foutieve invoer volgens de domeinregels.

Verwacht resultaat

De categorie creatie pagina opent correct. In te vullen velden zoals naam, beschrijving en afbeelding worden correct getoond. Bij het toevoegen van een nieuwe categorie indien alle gegevens werden correct ingevuld zal de nieuwe categorie op de overzichtspagina verschijnen. Bij een foutieve invoer krijgt de tester een foutmelding te zien dat zijn gegevens niet correct zijn. Bij het onsuccesvol of succesvol toevoegen van een categorie krijgt de gebruiker een gepaste alert te zien.

Beoordeling resultaat

[In te vullen nadat de test is uitgevoerd]

De foutmeldingen werden niet correct getoond. Dit werd in de code aangepast.

Remediëring

[In te vullen nadat de re-test is uitgevoerd]

De resultaten zijn voor het React-gedeelte zoals verwacht.

Assumpties

De tester maakt de volgende assumpties bij aanvang:

- De applicatie is in één taal gebouwd: het Nederlands.
- Elke rol kan testen. Authenticatie wordt pas in een volgende use case voorzien.
- De “mock” data is aanwezig in de persistentie laag om de exit criteria te kunnen bereiken.

Risico's

De onderstaande tabel geeft risico's mee voor het verdere verloop van het project.

Legende:

- **Klein:** de gevolgen zijn klein wanneer fouten op de productie-omgeving optreden. De gebruiker voert taken met weinig hinder in de applicatie uit. De aanbieder ondervindt geen schade: financieel, reputatie, etc.
- **Gemiddeld:** de gevolgen zijn gemiddeld wanneer fouten op de productie-omgeving optreden. De gebruiker ervaart enige moeite om taken in de applicatie uit te voeren. De aanbieder ondervindt enige schade: financieel, reputatie, etc.
- **Groot:** de gevolgen zijn groot wanneer fouten op de productie-omgeving optreden. De gebruiker kan de taken in de applicatie niet uitvoeren. De aanbieder ondervindt substantiële schade: financieel, reputatie, etc.

Test matrix

Testen op domeinregels voor use-case 2:

	DOMEINREGEL TEST			
FEATURES	Categorie unieke naam	Categorie naam, beschrijving, afbeelding zijn verplicht	Categorie naam ,beschrijving, afbeelding mogen niet leeg zijn	De naam bestaat minimaal uit 2 hoogstens 50 karakters. De beschrijving bestaat uit minimaal 5 hoogstens 250 karakters.
Categorie creatie pagina	Word getest	Getest	Getest	Getest

De testen slagen.

b. Wijzigen Doelstelling

Resources

De onderstaande tabel geeft aan wie de testen voor deze use case zal uitvoeren.

Tester	% Werkuren
Rayme Emin	33% (30 minuten), Java
Henri De Bel	33% (30 minuten), Java en React
Christof Remeysen	33% (30 minuten), React

Features

De ontwikkelde features voor deze use case staan hieronder beschreven.

Feature	Beschrijving
Detailpagina	Deze geeft de details van de doelstelling weer: naam, beschrijving, status, SDGs entiteiten, datasource.
Call-to-action	Deze nodigt de gebruiker uit om een doelstelling via het formulier te wijzigen.
Formulier	Dit biedt de functionaliteit aan om een doel te wijzigen. Deze velden moeten worden ingevuld om het formulier geldig in te dienen: naam, beschrijving, status, SDGs entiteiten. Het formulier bevat een annuleer-functie om terug naar de detailpagina te keren.
Validatieboodschap	Deze geeft feedback inzake wijziging van de doelstelling.
REST API (Nest.js)	Deze bevat een HTTP PUT methode om een doelstelling te wijzigen (React).
MySQL databank	Deze luistert naar een DAO methode om een doelstelling te wijzigen (Java), welke via de doelstelling repository vanuit de Domeincontroller in de GUI wordt aangesproken.
CSS stijlen	Deze zorgen voor de opmaak van een scherm: <ul style="list-style-type: none">• React: Chakra.• Java: JavaFX CSS.

Methoden

Deze sectie geeft aan hoe de hierboven vernoemde features worden getest.

Feature	Type test	Beschrijving
---------	-----------	--------------

Detailpagina	Unit en component Functioneel	<p>De unit en component testen verlopen automatisch. Ze worden enkel voor Java (JUnit) voorzien.</p> <p>De functionele testen verlopen exploratief, zowel voor Java als React. De tester opent de detailpagina en controleert deze. Voor React gebeurt dit semiautomatisch met Cypress.</p>
Call-to-action	Functioneel	De functionele testen verlopen exploratief, zowel voor Java als React. De tester klikt op de call-to-action en controleert of het formulier goed verschijnt. Voor React gebeurt dit semiautomatisch met Cypress.
Formulier	Functioneel	De functionele testen verlopen exploratief, zowel voor Java als React. Eerst tracht de tester het formulier ongeldig in te dienen en keert hij terug. Hierna wijzigt hij elk veld op een geldige manier. Voor React gebeurt dit semiautomatisch met Cypress.
Validatieboodschap	Unit en component Functioneel	<p>De unit en component testen verlopen automatisch. Ze worden enkel voor Java (JUnit) voorzien.</p> <p>De functionele testen verlopen exploratief, zowel voor Java als React. De tester checkt of de boodschappen correct verschijnen wanneer het formulier ongeldig en geldig wordt ingediend. Voor React gebeurt dit semiautomatisch met Cypress.</p>
REST API (Nest.js)	Integratie Performance	De integratietest verloopt exploratief. Ze gebeurt enkel voor React. De tester verifieert op het zicht of de data enkel na geldige indiening van het formulier wordt opgeslagen. Daarnaast checkt hij of de Nest.js API snel genoeg op het PUT verzoek reageert.
MySQL databank	Integratie Performance	De integratietest verloopt exploratief. Ze gebeurt enkel voor Java. De tester verifieert op het zicht of de data enkel na geldige indiening van het formulier wordt opgeslagen. Daarnaast checkt hij of de MySQL databank snel genoeg op het DAO verzoek reageert.

CSS stijlen	Functioneel Usability	De tester verifieert exploratief of de CSS stijlen goed inladen, zowel voor Java als React. Daarbij let hij op gebruiksvriendelijke aspecten, zoals goed aanklikbare knoppen en leesbaarheid van de formulervelden.
-------------	-----------------------	---

Assumpties

De tester maakt de volgende assumpties bij aanvang:

- De applicatie is in één taal gebouwd: het Nederlands.
- De “mock” data is aanwezig in de persistentie laag om de exit criteria te kunnen bereiken.
- Elke rol kan testen. Authenticatie wordt pas in volgende use cases (UC5 en UC6) voorzien.

Risico's

De onderstaande tabel geeft risico's mee voor het verdere verloop van het project.

Legende:

- Klein: de gevolgen zijn klein wanneer fouten op de productie-omgeving optreden. De gebruiker voert taken met weinig hinder in de applicatie uit. De aanbieder ondervindt geen schade: financieel, reputatie, etc.
- Gemiddeld: de gevolgen zijn gemiddeld wanneer fouten op de productie-omgeving optreden. De gebruiker ervaart enige moeite om taken in de applicatie uit te voeren. De aanbieder ondervindt enige schade: financieel, reputatie, etc.
- Groot: de gevolgen zijn groot wanneer fouten op de productie-omgeving optreden. De gebruiker kan de taken in de applicatie niet uitvoeren. De aanbieder ondervindt substantiële schade: financieel, reputatie, etc.

#	Risico	Impact	Mitigatieplan
Afwezigheid van authenticatie (vertrouwelijkheid)	Elke gebruiker kan een doelstelling wijzigen, ook al heeft hij de juiste rol niet.	Groot	We aanvaarden het risico voor staging. Zodra de applicatie live kan, testen we authenticatie en de zichtbaarheid van features voor rollen in detail.
Capaciteit en mock-data (performantie)	Het aantal gelijktijdige verzoeken voor de REST API en databank en de hoeveelheid mock-data weerspiegelen geen realistische niveaus inzake laadtijden en performantie.	Gemiddeld	De testers voeren manuele laadtesten uit. Vlak voor go live, wordt een integratietest op de productie-databank en -API uitgevoerd.

Dataformaat (consistentie)	De data bevindt zich in een API voor React en in een databank voor Java, welke aanvragen in bepaalde formaten moeten blijven verwerken.	Groot	De testers voeren exploratieve en contract testen voor de API en DAO verzoeken uit.
----------------------------	---	-------	---

Test matrix

Testen op domeinregels voor use case 14 (DR_DOELSTELLING):

DOMEINREGEL	FEATURE			
	Detailpagina met accordeon	Formulier met validatie		
Unieke naam	Getest	Getest	Getest	Getest
Naam <= 50 tekens	Niet getest	Getest	Getest	Getest
Beschrijving <= 250 tekens	Niet getest	Getest	Getest	Getest
Alfanumerieke tekens	Getest	Getest	Getest	Getest
1 categorie	Niet getest	Getest	Getest	Getest
Maximum 1 datasource	Niet getest	Niet getest	Niet getest	Niet getest
Alle velden verplicht	Niet getest	Getest	Getest	Getest

Legende matrix:

- Getest: de tester dient de domeinregel tijdens het testen voor de feature te verifiëren.
- Niet getest: de tester dient de domeinregel niet tijdens het testen voor de feature te verifiëren.

De test is geslaagd. De domeinregel "<= 1 datasource" wordt in een volgend testplan opgenomen.

c. Registreren Gebruiker

Resources

De onderstaande tabel geeft aan wie de testen voor deze use case zal uitvoeren.

Tester	% Werkuren
Rayme Emin	33% (2u), Java, React
Henri De Bel	33% (2u), React
Christof Remeysen	33% (2u), React en Java

Features

De ontwikkelde features voor deze use cases staan hieronder beschreven.

Feature	Beschrijving
Registratie pop-up pagina	Een pop-up geeft velden weer voor het invullen van : naam, wachtwoord familienaam en voornaam.
Routing	De routing zorgt ervoor dat je van de registratiepagina naar de info pagina kan navigeren en omgekeerd.
REST API (React) JPA (Java)	De REST API/JPA ontvangt en behandelt verzoeken van de front-end. De REST API/JPA bevat seeds met mock-data;
MySQL databank	Deze bevat de data van een gebruiker (Java) (React).
CSS stijlen	De CSS stijlen zorgen voor de opmaak van een scherm: <ul style="list-style-type: none">• React: Chakra, CSS.• Java: JavaFX, CSS.
Alerts	Er worden alerts voorzien om de gebruiker te informeren over het slagen/falen van de uitgevoerde taak.
Foutmeldingen	Er worden foutmeldingen voorzien bij de inputvelden volgens de domeinregels.

Methoden

Deze sectie geeft aan hoe de hierboven vernoemde features worden getest.

Feature	Type test	Beschrijving
Registratie pop-up pagina	Unit en component Functioneel	<p>De unit en component testen verlopen automatisch. Ze worden enkel voor Java (JUnit) voorzien.</p> <p>De functionele testen verlopen exploratief, zowel voor Java als React. De tester opent de registratie pop-up pagina en vult vervolgens alle verplichte velden in. Vervolgens de tester kan kiezen om zich te registreren of te annuleren. Voor React gebeurt dit semiautomatisch met Cypress.</p>
Routing	Functioneel Performance	<p>De functionele testen verlopen exploratief, zowel voor Java als React. De tester controleert of de juiste pagina's bij navigatie openen.</p> <p>Voor Java, verifieert de tester de compilatieduur en snelheid van laden op exploratieve wijze. Voor React, zet de tester hiervoor Google Lighthouse en React Profiler in.</p>

REST API	Integratie Performance	De integratietest verloopt exploratief. Ze gebeurt enkel voor React. De REST API (Nest.js) zorgt voor het opslaan van een nieuwe registratie van een gebruiker in de databank (React). De tester verifieert op het zicht of de registratie verzoeken goed en snel genoeg door de Nest.js API worden uitgevoerd.
MySQL databank	Integratie Performance	De integratietest verloopt exploratief. Ze gebeurt enkel voor Java. De tester verifieert op het zicht of de data goed en snel genoeg in de MySQL databank worden opgeslagen.
CSS stijlen	Functioneel Usability	De tester verifieert exploratief of de CSS stijlen goed inladen, zowel voor Java als React. Daarbij let hij op gebruiksvriendelijke aspecten, zoals goed aanklikbare knoppen, responsive design.
Alerts	Unit en component Functioneel	De unit en component testen verlopen automatisch. Ze worden enkel voor Java (JUnit) voorzien. De tester verifieert exploratief of de juiste alerts getriggerd worden bij het succesvol of niet succesvol registreren van een tester.
Foutmeldingen	Unit en component Functioneel	De unit en component testen verlopen automatisch. Ze worden enkel voor Java (JUnit) voorzien. De tester verifieert exploratief of de juiste foutmeldingen getriggerd worden bij een foutieve invoer volgens de domeinregels.

Verwacht resultaat

In te vullen velden zoals gebruikersnaam, wachtwoord, familienaam en voornaam worden correct getoond. Bij een succesvolle registratie zal de tester automatisch aangemeld worden en indien gewenst kan die de dashboard raadplegen. Bij een foutieve invoer krijgt de tester een foutmelding te zien dat zijn gegevens niet correct zijn. Bij een onsuccesvolle of succesvolle registratie krijgt de tester een gepaste alert te zien.

Beoordeling resultaat

[In te vullen nadat de test is uitgevoerd]

De foutmeldingen werden niet correct getoond. Dit werd in de code aangepast.

Remediëring

[In te vullen nadat de re-test is uitgevoerd]

De resultaten zijn voor het React-gedeelte zoals verwacht.

Assumpties

De tester maakt de volgende assumpties bij aanvang:

- De applicatie is in één taal gebouwd: het Nederlands.
- Authenticatie is voorzien. De tester krijgt een default gebruikersrol.

Risico's

De onderstaande tabel geeft risico's mee voor het verdere verloop van het project.

Legende:

- **Klein:** de gevolgen zijn klein wanneer fouten op de productie-omgeving optreden. De gebruiker voert taken met weinig hinder in de applicatie uit. De aanbieder ondervindt geen schade: financieel, reputatie, etc.
- **Gemiddeld:** de gevolgen zijn gemiddeld wanneer fouten op de productie-omgeving optreden. De gebruiker ervaart enige moeite om taken in de applicatie uit te voeren. De aanbieder ondervindt enige schade: financieel, reputatie, etc.
- **Groot:** de gevolgen zijn groot wanneer fouten op de productie-omgeving optreden. De gebruiker kan de taken in de applicatie niet uitvoeren. De aanbieder ondervindt substantiële schade: financieel, reputatie, etc.

Risico	Beschrijving	Impact	Mitigatieplan
API-data (consistentie)	De data bevindt zich in een API voor React, welke aanvragen in bepaalde formaten moet blijven verwerken.	Groot	De testers voeren exploratieve en contract testen voor de API calls uit.

Test matrix

Testen op domeinregels voor use-case 5:

	DOMEINREGEL TEST			
FEATURES	Gebruikersnaam is uniek en moet bestaan uit minstens 2 en hoogstens 20 karakters	Wachtwoord moet bestaan uit minstens 5 en hoogstens 20 karakters	Voornaam moet bestaan uit minstens 2 en hoogstens 30 karakters	Familienaam moet bestaan uit minstens 2 en hoogstens 30 karakters
Registratie pop-up pagina	Word getest	Getest	Getest	Getest

d. Aanpassen Rol

Resources

De onderstaande tabel geeft aan wie de testen voor deze use case zal uitvoeren.

Tester	% Werkuren
Rayme Emin	33% (2u), Java, React
Henri De Bel	33% (2u), React
Christof Remeysen	33% (2u), React en Java

Features

De ontwikkelde features voor deze use cases staan hieronder beschreven.

Feature	Beschrijving
Aanpassen gebruikersrol pagina	Een admin kan een bestaande gebruiker kiezen uit een lijst met alle gebruikers en zijn rol aanpassen.
Routing	De routing zorgt ervoor dat je van de dashboard naar de aanpassen gebruikersrol pagina kan navigeren en omgekeerd.
REST API (React) JPA (Java)	De REST API/JPA ontvangt en behandelt verzoeken van de front-end. De REST API/JPA bevat seeds met mock-data;
MySQL databank	Deze bevat de data van een gebruiker (Java) (React).
CSS stijlen	De CSS stijlen zorgen voor de opmaak van een scherm: <ul style="list-style-type: none">• React: Chakra, CSS.• Java: JavaFX, CSS.
Alerts	Er worden alerts voorzien om de gebruiker te informeren over het slagen/falen van de uitgevoerde taak.
Foutmeldingen	Er worden foutmeldingen voorzien bij de inputvelden volgens de domeinregels.

Methoden

Deze sectie geeft aan hoe de hierboven vernoemde features worden getest.

Feature	Type test	Beschrijving
---------	-----------	--------------

Aanpassen gebruikersrol pagina	Unit en component Functioneel	<p>De unit en component testen verlopen automatisch. Ze worden enkel voor Java (JUnit) voorzien.</p> <p>De functionele testen verlopen exploratief, zowel voor Java als React. De tester opent de aanpassen gebruikersrol pagina en selecteert een gebruiker. Vervolgens kan de tester de rol van de gekozen gebruiker wijzigen. Voor React gebeurt dit semi automatisch met Cypress.</p>
Routing	Functioneel Performance	<p>De functionele testen verlopen exploratief, zowel voor Java als React. De tester controleert of de juiste pagina's bij navigatie openen.</p> <p>Voor Java, verifieert de tester de compilatie duur en snelheid van laden op exploratieve wijze. Voor React, zet de tester hiervoor Google Lighthouse en React Profiler in.</p>
REST API (React) JPA (Java)	Integratie Performance	<p>De integratietest verloopt exploratief. De REST API (Nest.js) stuurt een verzoek naar de databank voor het ophalen van een lijst met bestaande gebruikers. (React) (JPA voor Java). De tester verifieert op het zicht of de aanpassen gebruikersrol verzoeken goed en snel genoeg door de Nest.js API worden uitgevoerd.</p>
MySQL databank	Integratie Performance	<p>De integratietest verloopt exploratief. De tester verifieert op het zicht of de data goed en snel vanuit de databank wordt opgehaald.</p>
CSS stijlen	Functioneel Usability	<p>De tester verifieert exploratief of de CSS stijlen goed inladen, zowel voor Java als React. Daarbij let hij op gebruiksvriendelijke aspecten, zoals goed aanklikbare knoppen, responsive design.</p>
Alerts	Unit en component Functioneel	<p>De tester verifieert exploratief of de juiste alerts getriggerd worden bij het succesvol of niet succesvol wijzigen van een gebruikersrol.</p>

Verwacht resultaat

Bij het onsuccesvol of succesvol wijziging van een rol zal de tester een gepaste alert zien.

Beoordeling resultaat

[In te vullen nadat de test is uitgevoerd]

/

Remediëring

[In te vullen nadat de re-test is uitgevoerd]

/

Assumpties

De tester maakt de volgende assumpties bij aanvang:

- De applicatie is in één taal gebouwd: het Nederlands.
- Authenticatie is voorzien.
- De databank werd gevuld met mock-data van bestaande gebruikers.

Risico's

De onderstaande tabel geeft risico's mee voor het verdere verloop van het project.

Legende:

- **Klein:** de gevolgen zijn klein wanneer fouten op de productie-omgeving optreden. De gebruiker voert taken met weinig hinder in de applicatie uit. De aanbieder ondervindt geen schade: financieel, reputatie, etc.
- **Gemiddeld:** de gevolgen zijn gemiddeld wanneer fouten op de productie-omgeving optreden. De gebruiker ervaart enige moeite om taken in de applicatie uit te voeren. De aanbieder ondervindt enige schade: financieel, reputatie, etc.
- **Groot:** de gevolgen zijn groot wanneer fouten op de productie-omgeving optreden. De gebruiker kan de taken in de applicatie niet uitvoeren. De aanbieder ondervindt substantiële schade: financieel, reputatie, etc.

Risico	Beschrijving	Impact	Mitigatieplan
API-data (consistentie)	De mock-data bevindt zich in een API voor React, welke aanvragen in bepaalde formaten moet blijven verwerken. Als een functionaliteit aangepast wordt kan de mock-data	Groot	De testers voeren exploratieve en contract testen voor de API calls uit.
Mock-data (performantie)	De mock-data weerspiegelt geen realistische niveaus inzake laadtijden en performantie.	Gemiddeld	De testers voeren manuele laad testen uit. Vlak voor go live, wordt een integratietest op de productie-databank uitgevoerd.

Test matrix

Testen op domeinregels voor use-case 7:

	DOMEINREGEL TEST		
FEATURES	Tester heeft een Admin rol	Minimaal 1 andere niet admin gebruiker	Een admin kan niet de rol van een andere admin wijzigen
Aanpassen gebruikersrol pagina	Wordt getest	Wordt getest	Wordt getest

e. Verwijder datasource

Resources

De onderstaande tabel geeft aan wie de testen voor deze use case zal uitvoeren.

Tester	% Werkuren
Rayme Emin	33% (30 minuten), Java
Henri De Bel	33% (30 minuten), React
Christof Remeysen	33% (30 minuten), Java en React

Features

De ontwikkelde features voor deze use case staan hieronder beschreven.

Feature	Beschrijving
Detailpagina	Deze geeft de details van de datasource weer: naam.
Call-to-action	Deze nodigt de gebruiker uit om een datasource te verwijderen.
Confirmatie pop-up	Deze vraagt aan de gebruiker of hij zeker is dat de datasource mag worden verwijderd.
REST API (Nest.js)	Deze bevat de data van een datasource (React).
MySQL databank	Deze bevat de data van een datasource (Java).
CSS stijlen	Deze zorgen voor de opmaak van een scherm: <ul style="list-style-type: none"> • React: Chakra. • Java: JavaFX CSS.

Methoden

Deze sectie geeft aan hoe de hierboven vernoemde features worden getest.

Feature	Type test	Beschrijving
Detailpagina	Unit en component Functioneel	De unit en component testen verlopen automatisch. Ze worden enkel voor Java (JUnit) voorzien. De functionele testen verlopen exploratief, zowel voor Java als React. De tester opent de detailpagina en controleert deze. Voor React gebeurt dit semi automatisch met Cypress.
Call-to-action	Functioneel	De functionele testen verlopen exploratief, zowel voor Java als React. De tester klikt op de call-to-action en controleert of de confirmatie pop-up goed verschijnt. Voor React gebeurt dit semi automatisch met Cypress.
Confirmatie pop-up	Functioneel	De functionele testen verlopen exploratief, zowel voor Java als React. De tester klikt afwisselend op de annuleer- en bevestig-functie. Voor React gebeurt dit semi automatisch met Cypress.
REST API (Nest.js)	Integratie Performance	De integratietest verloopt exploratief. Ze gebeurt enkel voor React. De tester verifieert op het zicht en via Postman of de data goed en snel genoeg in de Nest.js API wordt verwijderd.
MySQL databank	Integratie Performance	De integratietest verloopt exploratief. Ze gebeurt enkel voor Java. De tester verifieert op het zicht of de data goed en snel genoeg in de MySQL databank wordt verwijderd.
CSS stijlen	Functioneel Usability	De tester verifieert exploratief of de CSS stijlen goed inladen, zowel voor Java als React. Daarbij let hij op gebruiksvriendelijke aspecten, zoals goed aanklikbare knoppen en leesbaarheid van tekst.

Assumpties

De tester maakt de volgende assumpties bij aanvang:

- De applicatie is in één taal gebouwd: het Nederlands.
- De “mock” data is aanwezig in de persistentie laag om de exit criteria te kunnen bereiken.
- Elke rol kan testen. Authenticatie wordt pas in volgende use cases (UC5 en UC6) voorzien.

Risico's

De onderstaande tabel geeft risico's mee voor het verdere verloop van het project.

Legende:

- Klein: de gevolgen zijn klein wanneer fouten op de productie-omgeving optreden. De gebruiker voert taken met weinig hinder in de applicatie uit. De aanbieder ondervindt geen schade: financieel, reputatie, etc.
- Gemiddeld: de gevolgen zijn gemiddeld wanneer fouten op de productie-omgeving optreden. De gebruiker ervaart enige moeite om taken in de applicatie uit te voeren. De aanbieder ondervindt enige schade: financieel, reputatie, etc.
- Groot: de gevolgen zijn groot wanneer fouten op de productie-omgeving optreden. De gebruiker kan de taken in de applicatie niet uitvoeren. De aanbieder ondervindt substantiële schade: financieel, reputatie, etc.

#	Risico	Impact	Mitigatieplan
Afwezigheid van authenticatie (vertrouwelijkheid)	Elke gebruiker kan een datasource verwijderen, ook al heeft hij de juiste rol niet.	Groot	We aanvaarden het risico voor staging. Zodra de applicatie live kan, testen we authenticatie en de zichtbaarheid van features voor rollen in detail.
Capaciteit en mock-data (performantie)	Het aantal gelijktijdige verzoeken voor de REST API en databank en de hoeveelheid mock-data weerspiegelen geen realistische niveaus inzake laadtijden en performantie.	Gemiddeld	De testers voeren manuele laadt testen uit. Vlak voor go live, wordt een integratietest op de productie-databank en -API uitgevoerd.
Dataformaat (consistentie)	De data bevindt zich in een API voor React en in een databank voor Java, welke aanvragen in bepaalde formaten moeten blijven verwerken.	Groot	De testers voeren exploratieve en contract testen voor de API en DAO verzoeken uit.

Test matrix

Testen op domeinregels voor use case 11 (DR_DATASOURCE):

	FEATURE		
DOMEINREGEL	Detailpagina met accordeon	REST API (Nest.js)	MySQL databank
Unieke naam	Getest	Getest	Getest

Legende matrix:

- Getest: de tester dient de domeinregel tijdens het testen voor de feature te verifiëren.
- Niet getest: de tester dient de domeinregel niet tijdens het testen voor de feature te verifiëren.

De test is geslaagd.

7. Retrospectief

Henri De Bel

Voor dit tweede vakoverschrijdende groepswork leerde ik heel wat omdat we veel vrijheid hadden in de manier waarop we bepaalde problemen oplossen. Deze vrijheid zorgt er ook voor dat verschillende meningen en oplossingen telkens anders benaderd werden door de groepsleden wat de kwaliteit van het eindproduct ten goede kwam. Voor mij persoonlijk was het moeilijkste onderdeel de overstap maken van analyse en ontwerp naar het programmeren, aangezien ik vaak het gevoel had dat het werk sneller verliep door onmiddellijk te programmeren. Achteraf gezien denk ik toch dat het een goede zaak is dat we verplicht zijn de analyse en ontwerpfase te doorlopen omdat het ons verplicht om meer na te denken over de keuzes die we maakten bij de implementatie van de applicatie.

De onderlinge samenwerking verliep in zijn geheel goed ondanks enkele minpunten. Eerst en vooral hebben we problemen gehad met onze stiptheid. Als verzachtend omstandigheid is het wel zo dat we elk parttime studenten zijn en we slechts drie actieve groepsleden hadden. Een tweede minpunt komt voort uit het pluspunt dat we elk heel zelfstandig zijn waardoor er soms niet genoeg communicatie was en er vermijdbare fouten in het project slopen. Desondanks deze werkpunten slaagden we er naar mijn mening toch in om een goed afgewerkt product af te leveren waarin iedereen zijn bijdrage leverde.

Als ik reflecteer op de bijdragen van de groepsleden dan is het zo dat in het begin vooral ikzelf de leiding nam wat goed was aangezien het compenseerde voor het wegvallen van ons vierde groepslid, dankzij deze inspanning bleven we toch op schema. Naar het einde van het project toe werd mijn inbreng wel minder wat correct is aangezien ik al een groot deel had gedaan. Christof was degene die ervoor zorgde dat er toch altijd communicatie was en hij heeft doorheen het project telkens actief meegewerkt en ondanks zijn mindere ervaring met het programmeren in React compenseerde hij dit door bij analyse en ontwerp meer inbreng te geven en bij het programmeren constructieve vragen te stellen.

Wat betreft mijn eigen bijdrage tot het project heb ik over het algemeen iets minder gefocust op analyse en ontwerp omdat ik hier minder vertrouwd mee was, gelukkig werd dit opgevangen door mijn teamgenoten en kon ik zo inpijken waar nodig. Op vlak van communicatie denk ik dat ik soms wat actiever had mogen zijn maar zoals eerder al vermeld werd dit gelukkig opgevangen door de zelfstandigheid van de groepsleden. Ik denk dat mijn grootste bijdrage bij het programmeren lag waar ik doorheen het project het overgrote deel van mijn tijd aan besteed heb.

Tenslotte kan ik zeggen dat dit project een leerrijke ervaring was omdat we ervoeren hoe we incrementeel een softwareproduct in een team kunnen uitwerken. Het voordeel van deze

praktijkgerichte aanpak is dat we hiermee het meeste bijleren voor wie later op professioneel niveau wil softwareprojecten wil uitwerken.

Rayme Emin

	Wat heb ik bijgeleerd?	Aandachtspunten:
Analyse	<p>Communicatie: het was mooi om te zien hoe belangrijk een goede analyse/ontwerp van een project is aangezien het als startpunt dient voor de code. Het dient ook als communicatiemiddel tussen de teamleden. Als er iets werd aangepast of toegevoegd in de code wat nog niet in het domeinmodel stond werd het gelijk ook aangepast in het domeinmodel wat andere teamleden konden zien en gebruiken tijdens het coderen.</p> <p>Uitbreiding skills: ik heb de use cases kort en krachtig leren schrijven zonder triviale informatie te vermelden.</p> <p>Begeleiding: Mijnheer Dekoning is een begeleider die vol is met enthousiasme en deelt zijn kennis graag met de studenten. Dit werd zeker geapprecieerd.</p>	<p>Communicatie: de verwachtingen voor het analyse gedeelte waren niet goed gecommuniceerd voor dit vak i.v.m. Projecten 1.</p> <p>Planning: afspraken tussen studenten en lector werden niet altijd gerespecteerd. Het heeft voor verwarring gezorgd tussen de teamleden.</p> <p>Verwachtingen: de verwachtingen voor het analyse gedeelte werden last-minute aangepast. We hebben meer werk gedaan dan nodig. Naar mijn mening werd onze tijd en moeite niet altijd gewaardeerd.</p>

TECH: JAVA/ REACT	<p>Aanwezigheid: teamleden deelden graag gemiste feedback wat heeft gezorgd voor het succesvol afronden van het Java gedeelte.</p> <p>Uitbreiding skills (JAVA): ik heb geleerd hoe ik het best de observable pattern kan toepassen door de Observables, Listeners en Bindings van JavaFX te gebruiken.</p> <p>Uitbreiding skills (REACT): mijn teamgenoten hebben meer ervaring als mij met React/Typescript aangezien ze al Front-end developers zijn. Ik heb dankzij hun mijn kennis verder kunnen uitbreiden.</p> <p>Vrijheid: de vrijheid die we voor dit project hebben ontvangen heeft voor eindeloze leuke mogelijkheden gezorgd om een oplossing te formuleren.</p> <p>Teamwork: het is een leuke uitdaging om met andere mensen te werken. Ze brengen nieuwe ideeën tot leven en alternatieve oplossingen voor een probleem.</p>	<p>Aanwezigheid: wegens een operatie en als gevolg andere doktersafspraken heb ik jammer genoeg twee belangrijke feedback momenten gemist.</p> <p>Relevatie (JAVA): De verwachtingen voor Projecten 2 waren gelijkaardig aan de verwachtingen voor Projecten 1. Het zou interessanter zijn als het een Spring project was geweest.</p> <p>Relevatie (REACT): De verwachtingen voor Projecten 2 waren gelijkaardig aan de verwachtingen voor Web Services en Front-end Web development. Het zou interessanter geweest zijn als het project niet in twee delen werd opgesplitst wat de mogelijkheid zou bieden om een ingewikkelder project op te leveren.</p> <p>Verwachtingen: aangezien we met 3 teamleden waren i.p.v. 5 we zouden minder werk moeten krijgen dan andere groepen. Jammer genoeg werd er niet duidelijk gemaakt wat precies zou wegvallen. Dit heeft voor onzekerheid en stress gezorgd.</p>
--	---	---

Christof Remeysen

Persoonlijke ervaringen

Sterktes	Zwaktes
----------	---------

<p>Communicatie: ik heb voldoende overlegmomenten met Henri en Rayme ingepland, waarin ik me constructief heb opgesteld. Dat brengt dat iedereen mee was inzake taakverdeling.</p> <p>Leergierig: ik heb een grote wil om kennis te vergaren. Dat is een troef in een sector als die van de informatica die snel evolueert.</p> <p>Analytisch inzicht: ik zie verbanden redelijk snel, wat het voordeel oplevert om voldoende kritisch naar het werk te kijken.</p> <p>Doorzetter: ik ben bereid tot hard werken om een doel te bereiken, ondanks tegenwind. Het feit dat ik nog geen React-kennis had, heeft me niet doen stoppen om bij te dragen aan de groep. Ik wil Henri bedanken dat hij de grootste brok van het werk voor het React-gedeelte heeft gerealiseerd, alsook Rayme om kennis met mij te delen en het Java-stuk prima te verdelen.</p> <p>Zelfstandigheid: ik kan autonoom zaken afwerken, waarmee ik anderen ontlast.</p>	<p>Planning: ik ben niet zo sterk in haalbaarheid afwegen. Aansluitend is mijn agenda vaak overladen voor het werk. Grenzen stellen zou mijn persoonlijke balans verbeteren, zodat ik niet tot zeer laat moet werken.</p> <p>Stiptheid: ik haal deadlines te last-minute. Een betere planning zou helpen, alsook een minder perfectionistische blik op het design.</p> <p>Onzekerheid: ik ga niet zo goed om met onzekerheid. Toen het project begon, had ik nog geen ervaring met React. In het begin kwam ik daardoor wat zenuwachtig over. Dat was onnodig. Zoals de docent aangaf, moet ik in onverwachte situaties beter uitgaan van de concepten die ik al wel zeer goed beheers.</p> <p>Zelfstandigheid: ik moet soms sneller hulp vragen of uitreiken naar anderen, waarmee ik tijd zou besparen en dubbel werk vermijdt.</p>
--	---

Technische ervaringen

Sterktes	Zwaktes
<p>Grafische aanleg:</p> <p>ik kan interfaces mooi vormgeven met eender welke technologie. Zelfs in minder gebruiksvriendelijke tools als JavaFX, lukt me dat sterk. Dat zorgt voor een optimale gebruikerservaring.</p> <p>Programmeren in Java:</p> <p>ik heb doorheen de jaren op de HoGent keurig in Java en JavaFX kunnen programmeren. Hoe meer ik het doe, hoe beter ik erin wordt. Er zit nog marge op.</p>	<p>Programmeren in React:</p> <p>ik ga de komende weken blijven investeren om mezelf het framework aan te leren. Dit kan in mijn huidige job van pas komen.</p> <p>Programmeerervaring:</p> <p>ik programmeer weinig in mijn huidige job. Dat maakt dat mijn praktische ervaring relatief beperkt is. Ik ga ervoor meer tijd trachten vrij te maken om erin te groeien.</p>

Slotwoord

Als we naar onze resultaten voor Analyse, React en Java kijken, dan mogen we best trots zijn op wat we de afgelopen weken als werkstudenten op een zelfstandige manier hebben gepresteerd. In de eerste plaats wil ik mijn groepsleden en docenten bedanken voor de kennisdeling en constructieve dialogen.

Daardoor zitten de applicaties onderliggend technisch mooi in elkaar. Voor het React-gedeelte, hebben we een innovatieve koers gevaren door TypeScript en Nest.js in te zetten. React was mij in tegenstelling tot mijn groepsgenoten vreemd, maar ik vond het best leuk om iets nieuws te leren. Voor het Java-gedeelte, hebben we bepaalde patronen zoals Observers en Interfaces handig aangewend, alsook bepaalde datastructuren. Het grafische aspect zou voor beide nog wel wat kunnen worden geoptimaliseerd, maar de grootste stappen zijn gezet.

Om tot het huidige resultaat te geraken, was samenwerken natuurlijk belangrijk. Ieder van ons heeft intensief aan het project meegewerkt, zich flexibel in tijd opgesteld en zijn sterktes goed in functie van de groep benut. Tot slot, wil ik daarom Henri en Rayme graag bedanken om deze leerrijke ervaring samen met mij aan te gaan en eindproducten met potentieel af te leveren.

