KVV RAUW

TEAM: ACO1

Analyse- en ontwerprapport

Quinte Belmans
Maarten Ceulemans
Kyan Collier
Ferre Kerkhofs
Siebe Sampermans

2025 - 2026

Inhoudsopgave

1. Inleiding

Dit analyse- en ontwerprapport vormt de basis voor de ontwikkeling van de webapplicatie voor KVV Rauw. Het doel van deze applicatie is om de administratieve processen binnen de club te vereenvoudigen, centraliseren en digitaliseren. Zowel spelers, ouders, trainers als beheerders moeten via het systeem op een efficiënte manier hun taken kunnen uitvoeren, zoals inschrijvingen beheren, kledij bestellen, activiteiten raadplegen, afwezigheden melden en betalingen registreren.

In dit document worden eerst de functionele en niet-functionele eisen beschreven die nodig zijn om tot een volledige en gebruiksvriendelijke oplossing te komen. Vervolgens wordt het datamodel uitgewerkt dat de structuur van de onderliggende gegevens beschrijft. Tot slot worden de verschillende functionaliteiten geprioriteerd volgens de MoSCoW-methode, zodat duidelijk is welke onderdelen essentieel zijn voor de eerste oplevering en welke later kunnen worden uitgebreid.

Dit rapport dient als leidraad voor het ontwikkelteam en zorgt voor een helder en gedeeld begrip van de scope, verwachtingen en technische fundamenten van het project.

2. Eisenanalyse

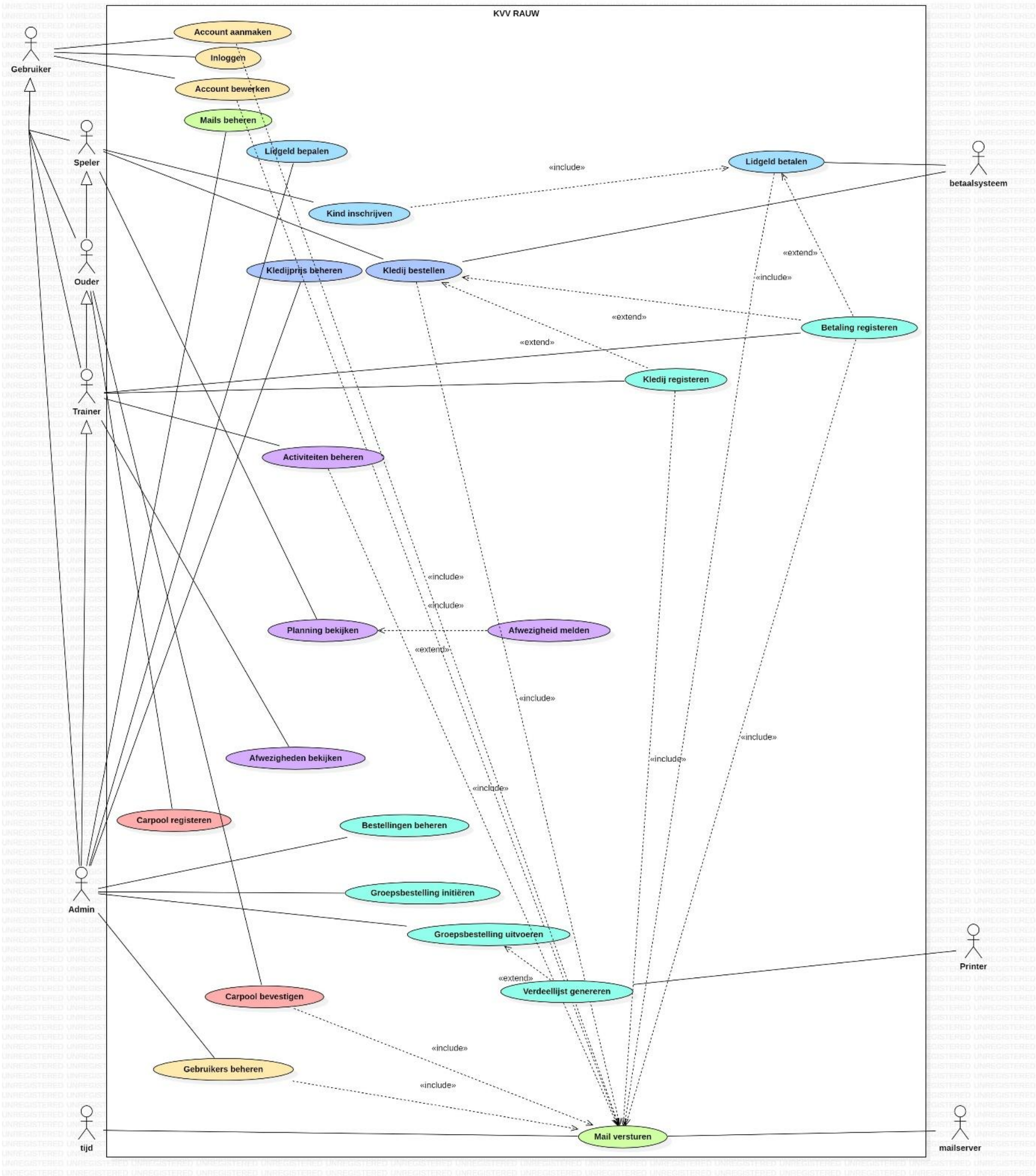
In dit hoofdstuk beschrijven we wat de webapplicatie precies moet kunnen en aan welke voorwaarden ze moet voldoen. We bekijken eerst de functionele eisen: dit zijn de taken die gebruikers in het systeem moeten kunnen uitvoeren, zoals inloggen, inschrijven, bestellingen maken of activiteiten beheren. Deze worden voorgesteld met use cases, zodat duidelijk is hoe elke functie werkt.

Daarnaast bekijken we de niet-functionele eisen. Dit zijn zaken zoals gebruiksvriendelijkheid, snelheid en vormgeving, die belangrijk zijn om de applicatie prettig en betrouwbaar te maken.

De eisenanalyse geeft dus een duidelijk overzicht van alles wat nodig is om de applicatie goed te kunnen ontwikkelen.

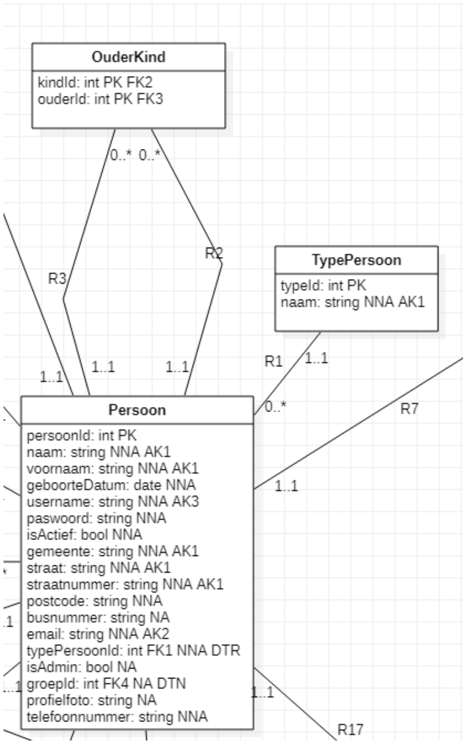
2.1. Functionele eisen

2.1.1. Use Case Diagram



2.1.2. Use Cases

2.1.2.1. Use case Account aanmaken – Kyan Collier



Functionaliteit:
Als speler kan ik een account aanmaken

Voorwaarde:
Actor heeft nog geen account.

Normaal verloop:
Het systeem geeft keuze tussen speler en ouder. Actor kiest “speler”. Het systeem vraagt om je gegevens en wachtwoord in te vullen. Actor vult de gegevens in. Het systeem toont een lijst van geregistreerde ouders. Actor selecteert een ouder en voltooit de registratie. Switch naar use case mail versturen.

Alternatieven:

- Actor kiest “ouder”: Als de actor kiest om een ouderaccount aan te maken, vraagt het systeem de persoonsgegevens van de ouder op. Het systeem toont de lijst van geregistreerde kinderen. De actor kan dan 0 of meerdere kinderen koppelen. Het systeem voltooit de registratie van de ouder.
- Geen geselecteerde ouder: de actor vult alle gegevens in, maar kiest geen ouder. Het systeem registreert de speler zonder ouderkoppeling.
- Meerdere ouders selecteren: de actor kan, indien hij reeds een ouder geselecteerd heeft, een extra ouder of voogd toevoegen.

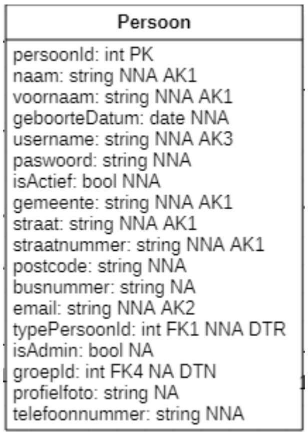
2.1.2.2. Use case inloggen – Siebe Sampermans

Functionaliteit:
Als speler kan ik inloggen.

Voorwaarde:
De gebruiker moet een geldig account hebben.

Normaal Verloop:
Het systeem begint en toont het inlogscherm. De gebruiker geeft zijn gebruikersnaam en wachtwoord in. Het systeem controleert de ingevoerde gegevens. Bij geldige gegevens logt de actor in. Het systeem toont nu het dashboard van het systeem.

Alternatieven:
- <u>Ongeldige gegevens:</u> de ingevoerde gegevens komen niet overeen met een account. Het systeem toont een foutmelding: “Ongeldige gegevens”.
- <u>Wachtwoord vergeten:</u> de actor is zijn wachtwoord vergeten. Hij kiest voor een wachtwoordherstel.
- <u>Account geblokkeerd:</u> de ingevoerde gegevens komen overeen met een account dat niet meer actief is. Het systeem toont een foutmelding: “Account geblokkeerd”.

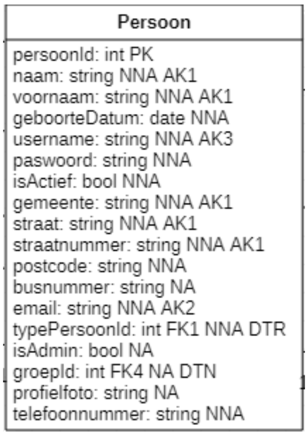


2.1.2.3. Use case Account bewerken – Siebe Sampermans

Functionaliteit:
Als speler kan ik mijn account bewerken

Voorwaarde:
De actor moet ingelogd zijn.

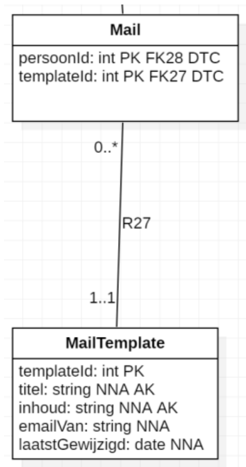
Normaal Verloop:
De actor opent het menu “account bewerken”. Het systeem toont de huidige accountgegevens. De actor wijzigt de gewenste gegevens (wachtwoord, e-mailadres, meldingen). De actor bevestigt de wijzigingen. Het systeem valideert en slaat de nieuwe gegevens op. Het systeem toont een bevestiging van de nieuwe gegevens.



Alternatieven:
- <u>Geen wijzigingen:</u> de actor wijzigt niets aan zijn huidige gegevens en gaat terug naar het startmenu.
- <u>Wachtwoord gewijzigd:</u> het systeem stuurt een bevestigingsmail voor verificatie.
- <u>Ongeldige invoer:</u> het systeem toont een foutmelding en vraagt om correctie.

2.1.2.4. Use case Mails beheren – Maarten Ceulemans

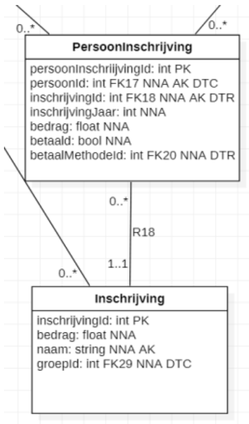
Functionaliteit: Als admin kan ik mails beheren
Vereisten: de actor moet ingelogd zijn
Normaal verloop: Systeem toont mails, actor selecteert mail, systeem toont inhoud van mail
Alternatieven:
- <u>Mail verwijderen:</u> Actor kiest voor verwijderen, systeem vraagt om bevestiging, actor bevestigt, systeem toont bevestiging van verwijdering
- <u>Nieuwe mail aanmaken:</u> Actor kiest voor het aanmaken van een mail, systeem toont invulveld voor mail, actor schrijft e-mail en verstuurt e-mail, systeem vraagt bevestiging, actor bevestigt, systeem toont bevestiging van versturing. Switch naar use case “mail versturen”
- <u>E-mailtemplates beheren:</u> Systeem toont lijst van templates, actor kiest template, systeem toont inhoud van template, actor bewerkt template en slaat template op, systeem vraagt bevestiging, actor bevestigt, systeem toont bevestiging van aanpassing



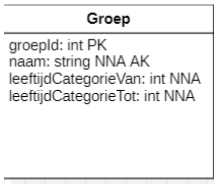
2.1.2.5. Use case Lidgeld bepalen – Maarten Ceulemans

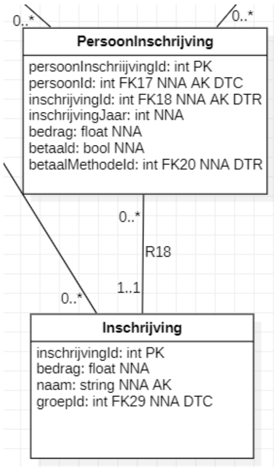
Functionaliteit: Als admin kan ik lidgeld bepalen
Vereisten: de actor moet ingelogd zijn
Normaal verloop: Systeem toont lijst met lidgeld, actor kiest lidgeld, systeem toont bedrag van lidgeld
Alternatieven:

- Lidgeld verwijderen: Actor kiest voor verwijderen, systeem vraagt om bevestiging, actor bevestigt, systeem toont bevestiging van verwijdering van lidgeld
- Nieuw lidgeld aanmaken: Actor kiest voor het aanmaken van een lidgeld, systeem toont invulveld voor lidgeld, actor vult bedrag van lidgeld in, systeem vraagt bevestiging, actor bevestigt, systeem toont bevestiging van aanmaak van lidgeld
- Lidgeld beheren: Systeem toont lijst van lidgelden, actor kiest lidgeld, systeem toont lidgeld, actor bewerkt lidgeld en slaat lidgeld op, systeem vraagt bevestiging, actor bevestigt, systeem toont bevestiging van aanpassing van lidgeld



2.1.2.6. Use case Speler Inschrijven – Ferre Kerkhofs





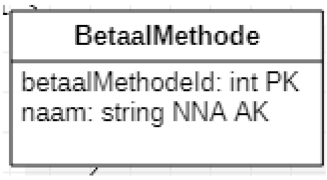
Functionaliteit:
Als gebruiker kan ik een speler inschrijven.

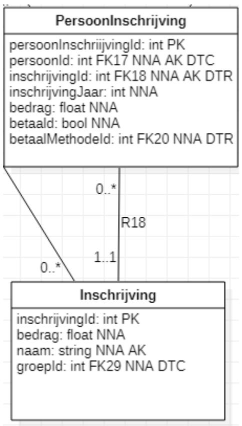
Precondities:
De actor is ingelogd.

Normale flow:
Het systeem toont het inschrijvingsformulier met verplichte velden (naam, geboortedatum, adres, medische info).
De actor vult de gegevens in, laat een U-groep kiezen op basis van leeftijd en bevestigt de inschrijving.
Het systeem controleert de ingevoerde gegevens.
Het systeem slaat de inschrijving op in de ledenadministratie.
Het systeem toont een bevestiging en schakelt naar use case “Mail Versturen”.

Alternatieven:
Onvolledige gegevens: als er velden ontbreken, vraagt het systeem om deze aan te vullen.
Dubbele inschrijving: als het kind al geregistreerd is, toont het systeem een melding.
Zelf groep kiezen: De actor kiest zelf een groep en wordt hieraan toegewezen

2.1.2.7. Use case Lidgeld Betalen – Ferre Kerkhofs





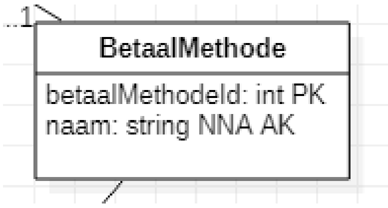
Functionaliteit:
Als gebruiker kan ik het lidgeld betalen

Precondities:
De gebruiker is ingelogd.

Normale flow:
Het systeem toont het openstaande lidgeld en beschikbare betaalopties.
De actor kiest een betaalmethode en bevestigt.
Het systeem leidt door naar het externe betaalsysteem.
Het betaalsysteem verwerkt de betaling en stuurt een bevestiging terug.
Het systeem schakelt over naar use case “Mail Versturen” en markeert het lidgeld als betaald.

Alternatieven:
Betaling geannuleerd**:** als de actor de betaling stopt, blijft de status “onbetaald”.
Cashbetaling : de actor kiest voor cashbetaling op de club

2.1.2.8. Use case Betaling Registreren – Ferre Kerkhofs



Functionaliteit:
Als trainer kan ik een cashbetaling registreren.

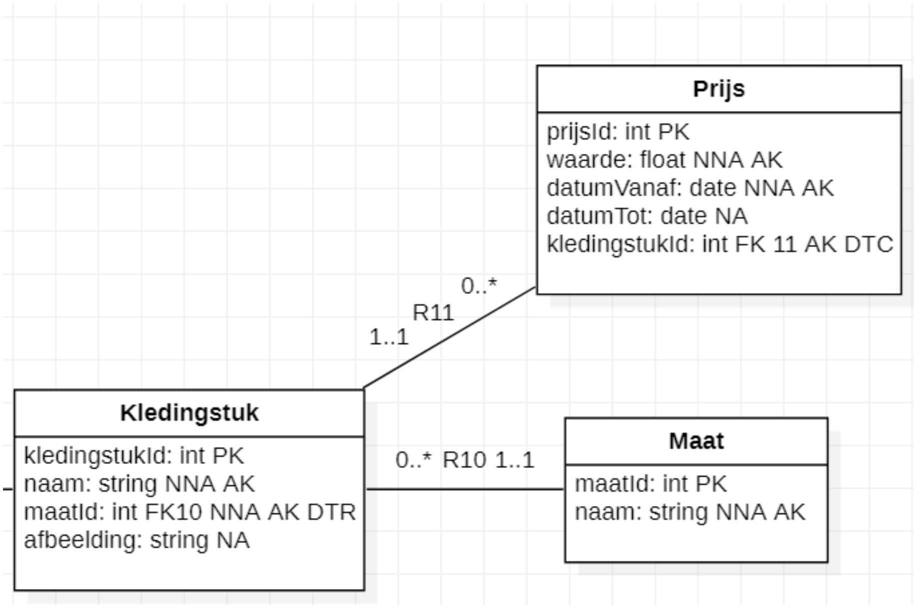
Precondities:
De trainer is ingelogd.

Normale flow:
Het systeem toont een overzicht van leden met openstaande bedragen.
De actor kiest het juiste lid en selecteert “betaald”.
Het systeem registreert de cashbetaling.
Het systeem toont een bevestiging van de registratie en schakelt over naar use case “Mail Versturen”.

2.1.2.9. Use case Prijs kledij beheren – Maarten Ceulemans

Functionaliteit: Als admin kan ik prijs kledij beheren
Vereisten: de actor moet ingelogd zijn
Normaal verloop: Systeem toont lijst met kleren, actor kiest kledij, systeem toont kledij en prijs van kledij
Alternatieven:

- Prijs kledij verwijderen: Actor kiest voor verwijderen van prijs van kledij, systeem vraagt om bevestiging, actor bevestigt, systeem toont bevestiging van verwijdering van prijs van kledij
- Nieuwe prijs kledij aanmaken: Actor kiest voor het aanmaken van een nieuwe prijs van kledij, systeem toont invulveld voor bedrag, actor vult bedrag van kledij in, systeem vraagt bevestiging, actor bevestigt, systeem toont bevestiging van aanmaak van prijs van kledij
- Prijs kledij beheren: Systeem toont lijst van kledij, actor kiest kledij, systeem toont kledij en prijs van kledij, actor bewerkt prijs van kledij en slaat prijs op, systeem vraagt bevestiging, actor bevestigt, systeem toont bevestiging van aanpassing



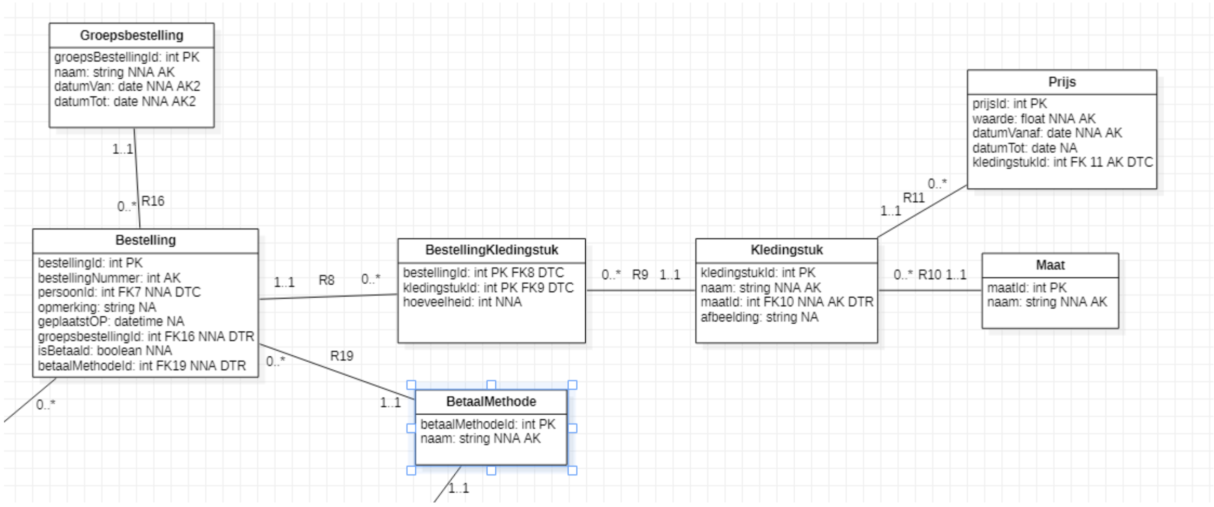
2.1.2.10. *Use case Kledij bestellen – Quinte Belmans*

Functionaliteit: Als gebruiker kan ik kledij bestellen

Voorwaarde: de actor is ingelogd.

Normaal verloop: het systeem toont een overzicht van alle beschikbare kledij, inclusief prijzen en maten. De gebruiker selecteert het gewenste kledingstuk en kiest de juiste maat. Vervolgens toont het systeem de winkelmand. De gebruiker kiest het juiste bestelmoment. Daarna vraagt het systeem of de gebruiker online wil betalen of contant op de voetbalclub. De gebruiker kiest voor online betaling met kaart. Het systeem toont een bevestiging van de bestelling en schakelt vervolgens over naar de use case “Mail versturen”.

Alternatieven:
- Levering voetbalclub: de gebruiker kiest voor **levering op de voetbalclub** en kiest daarna zijn voorkeur voor de betalingswijze. Vervolgens schakelt het over naar de use case “Mail versturen”.
- Betaling contant: de gebruiker kiest voor **contante betaling op de voetbalclub** in plaats van online betaling. De gebruiker betaalt contant op de voetbalclub en schakelt daarna over naar de use case “Kledij registreren”.



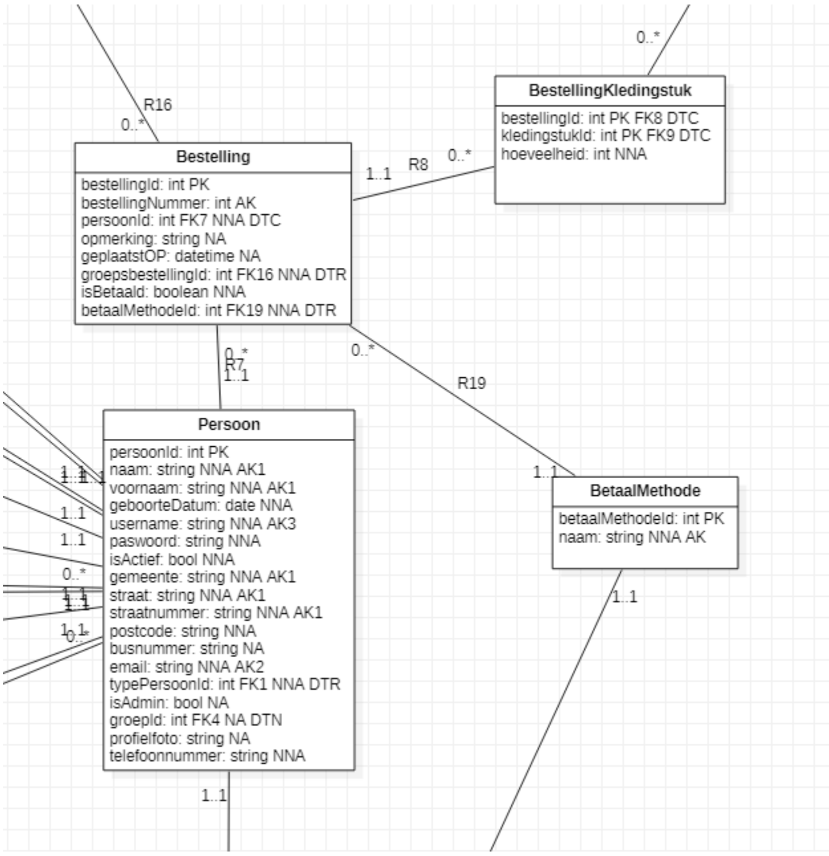
2.1.2.11. *Use case Kledij registreren – Quinte Belmans*

Functionaliteit: Als trainer kan ik kledij registreren

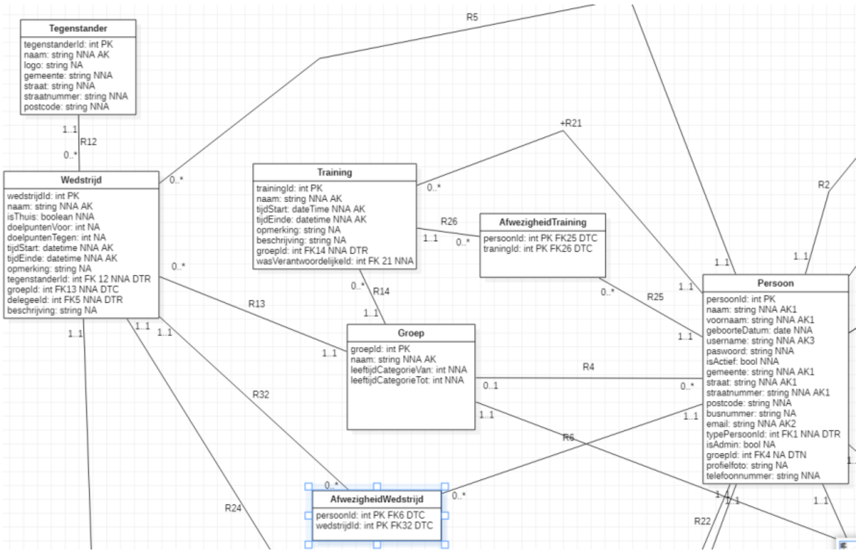
Voorwaarde: Trainer is ingelogd

Normaal verloop: het systeem geeft een melding dat een ouder heeft gekozen voor contante betaling op de club. De trainer krijgt een scherm te zien met de openstaande betalingen van de leden die gekozen hebben voor contante betaling. De trainer kan in dit scherm aanduiden dat de betaling ontvangen is. Zodra dit geregistreerd is, schakelt het systeem over naar de use case “Mail versturen”.

Alternatieven:
- Ouderbetaling niet gebeurd: de ouder heeft gekozen voor contante betaling, maar is nog niet aanwezig om te betalen. Het systeem houdt de bestelling in **“**in afwachting van betaling**”**-status. Zodra de betaling later wordt uitgevoerd, registreert de trainer deze alsnog in het systeem. Vervolgens wordt overgeschakeld naar de use case “Mail versturen”.



2.1.2.12. *Use case Activiteiten beheren – Kyan Collier*



Functionaliteit: Als admin kan ik de activiteiten beheren.

Voorwaarde: Admin is ingelogd.

Normaal verloop: het systeem toont alle geplande activiteiten (wedstrijden en trainingen). Actor kiest om een activiteit toe te voegen. Actor kiest het type van de activiteit (wedstrijd of training). Het systeem toont dan alle benodigde gegevens voor deze activiteit. Actor vult deze gegevens in en slaat de activiteit op. Het systeem verwerkt de gegevens en laadt opnieuw alle activiteiten zien, inclusief de nieuwe activiteit.

Alternatieven:

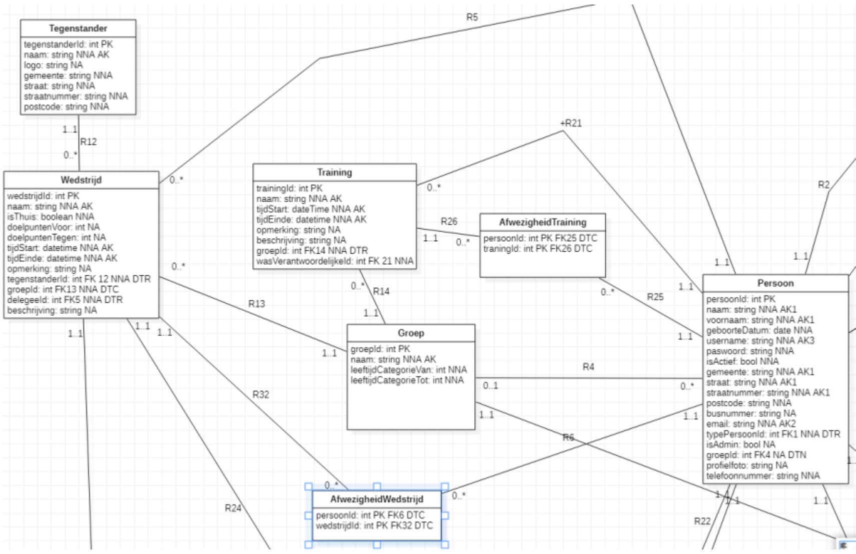
- Bewerken: De actor kiest om een activiteit in de agenda te bewerken. Het systeem toont de gegevens van die activiteit. De actor kan deze gegevens van deze activiteit veranderen.
- Verwijderen: de actor kiest om een activiteit uit de agenda te verwijderen. Het systeem vraagt om bevestiging. De actor bevestigt. Na bevestiging wordt de activiteit uit de agenda verwijderd.

2.1.2.13. Use case Planning bekijken – Kyan Collier

Functionaliteit: Als speler kan ik de planning bekijken.

Voorwaarde: speler is ingelogd.

Normaal verloop: Het systeem toont een overzicht van de geplande wedstrijden en activiteiten. Actor klikt op een bepaalde activiteit. Het systeem toont de gegevens over die activiteit (datum en tijd, type (training/wedstrijd), team, …) en meldt dat je aangemeld bent voor die activiteit.



Alternatieven:

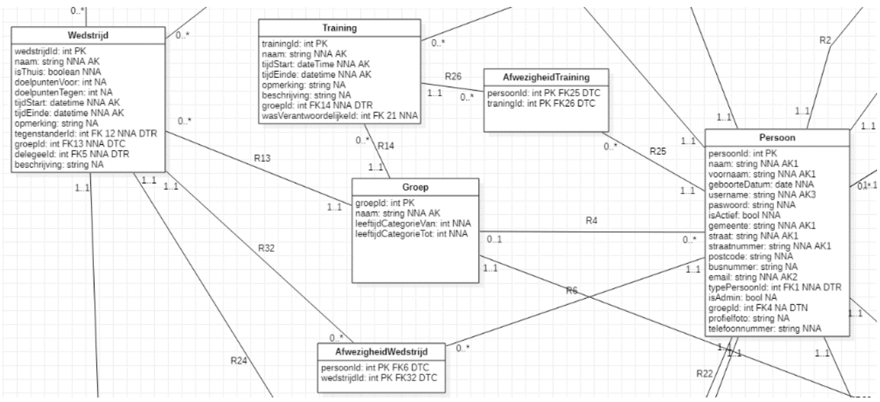
- geen geplande activiteiten: Als er geen geplande activiteiten zijn, toont het systeem dat er momenteel geen toekomstige activiteiten zijn ingepland.
- afwezigheid melden: Als de gebruiker niet aanwezig zal zijn op de activiteit, schakelt het systeem over op use case "Afwezigheid melden".

2.1.2.14. Use case Afwezigheid melden – Kyan Collier

Functionaliteit: Als speler kan ik afwezigheid melden

Voorwaarde: speler is ingelogd.

Normaal verloop: het systeem toont de activiteit met het vakje "aanwezig" dat je kan afvinken. De actor klinkt op het vinkje om zich afwezig te melden. Het systeem toont de actor een bevestiging en zet de speler op afwezig voor die activiteit.



Alternatieven:

- Terug aanmelden: Indien de speler zich bedenkt, kan hij zich terug aanmelden voor de activiteit door het vakje terug aan te vinken.

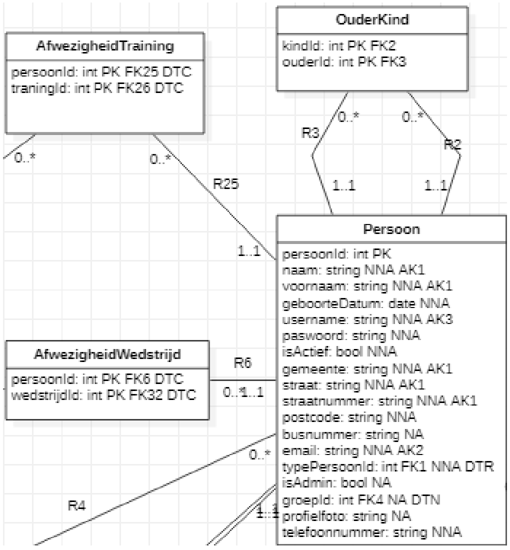
2.1.2.15. Use case Afwezigheden bekijken – Siebe Sampermans

Functionaliteit:
Als trainer kan ik afwezigheden bekijken.
Voorwaarde:
De trainer moet ingelogd zijn en er moet een training of wedstrijd ingepland zijn.
Er zijn afwezigheden geregistreerd in het systeem.

Normaal Verloop:
De trainer klikt op het menu "Afwezigheden". De trainer krijgt een overzicht van alle aanwezigheden en afwezigheden. De actor kan filters toepassen op datum, team, spelers. Het systeem toont het gefilterde overzicht. De actor kan details openen van specifieke afwezigheden.

Alternatieven:

- <u>Geen afwezigheden:</u> het systeem toont "Er zijn geen afwezigheden geregistreerd.".
- <u>Aanpassen van afwezigheden:</u> de trainer heeft maar een aantal personen nodig voor de selectie en zet de overige zelf op afwezig met als detail "Niet in de selectie".



2.1.2.16. Use case Carpool registreren – Quinte Belmans

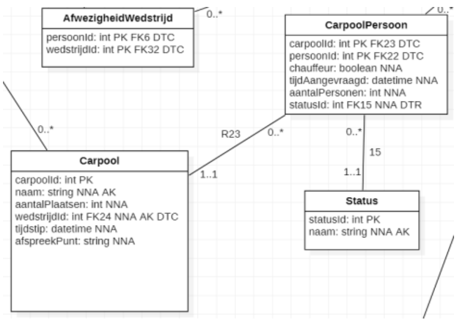
Functionaliteit: Als <u>ouder</u> kan ik carpool registreren

Voorwaarde: ingelogd zijn

Normale flow: het systeem toont een lijst met alle uitmatchen. De ouder selecteert een gewenste uitmatch waarvoor hij wil meerijden. Het systeem toont een formulier. De ouder vult het formulier in en bevestigt de aanmelding. Het systeem toont een bevestiging. Switch naar use case 'mail versturen'

Alternatieven:

- <u>Rijden:</u> Het systeem toont andere invulvelden. De ouder vult het formulier in en bevestigt de aanmelding. Het systeem toont een bevestiging. Switch naar use case 'mail versturen'



2.1.2.17. *Use case Carpool bevestigen – Quinte Belmans*

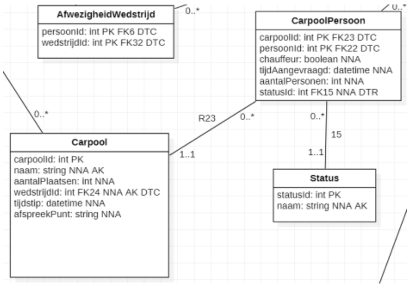
Functionaliteit: Als ouder kan ik carpool bevestigen

Voorwaarde: ingelogd zijn

Normale flow: Het systeem toont een lijst met de geregistreerde matchen waarvoor de ouder gaat rijden. De ouder selecteert een openstaande aanvraag en accepteert een aanvraag. Systeem toont een bevestiging. Switch naar use case 'mail versturen'

Alternatieven:

- Aanvraag weigeren: Een ouder weigert een aanvraag. Systeem toont knop 'toch accepteren'.
- Contactgegevens: Ouder klikt op informatieknop. Systeem toont telefoonnummer van aanvrager.
- Aanvraag toch accepteren: Ouder klikt op toch accepteren. Systeem toont bevestiging. Switch naar use case 'mail versturen'.



2.1.2.18. *Use case Gebruikers beheren – Siebe Sampermans*

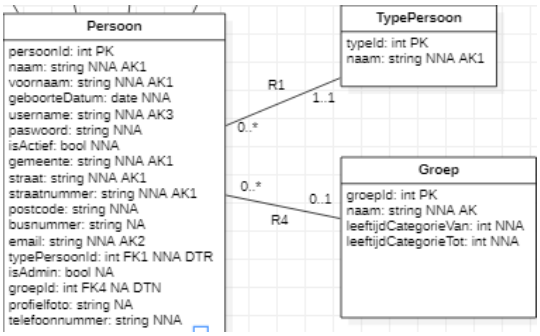
Functionaliteit:
Als admin kan ik gebruikers beheren.

Voorwaarde:
De admin moet ingelogd zijn.

Normaal Verloop:
De admin opent het menu "gebruikers beheren". Het systeem toont een overzicht van alle bestaande gebruikers met de bijbehorende gegevens. De admin kiest een actie: gebruiker toevoegen, verwijderen of bewerken. De admin vult de nodige gegevens in of past deze aan. Het systeem valideert de invoer en slaat de wijziging op. Het systeem toont een bevestiging dat de bewerking is voltooid.

Alternatieven:

- Geen wijzigingen: De admin wijzigt niets aan de huidige gegevens van een gebruiker en gaat terug naar het startmenu.
- Verwijdering annuleren: de admin drukte op verwijderen van een gebruiker, maar wil dit niet meer doen. Het systeem toont na het drukken op verwijderen "Bent u zeker dat u deze gebruiker wilt verwijderen" en geeft nog een optie om te annuleren. Er wordt geannuleerd en het account blijft behouden.



- Ongeldige invoer: de gegevens zijn verkeerd ingevuld (e-mail is ongeldig). Het systeem geeft een foutmelding en vraagt om correctie.

2.1.2.19. *Use case Groepsbestelling initiëren – Siebe Sampermans*

Functionaliteit:
Als admin kan ik groepsbestellingen initiëren.

Voorwaarde:
Als de admin is ingelogd

Normaal Verloop:
Het systeem toont een overzicht van alle groepsbestellingen en hun status (gesloten, bezig). De admin klikt op “nieuwe groepsbestelling” en gaat naar het volgende scherm. Het systeem vraagt om een start- en einddatum voor de groepsbestelling en de actor vult die in. De groepsbestelling wordt geïnitieerd.

Alternatieven:
- Gesloten groepsbestelling selecteren: Switch naar use case ‘groepsbestelling uitvoeren’
- Bezige groepsbestelling selecteren: Switch naar use case ‘bestellingen beheren’

2.1.2.20. *Use case Bestellingen beheren – Siebe Sampermans*

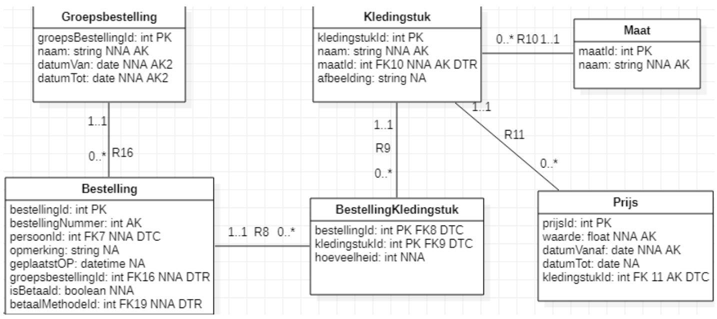
Functionaliteit:
Als admin kan ik bestellingen beheren

Voorwaarde:
Als er bestellingen zijn binnengekomen.
Als de admin is ingelogd.

Normaal Verloop:
Het systeem toont een overzicht van alle personen die een bestelling hebben geplaatst en wat ze allemaal besteld hebben. De admin verwijdert een van de bestellingen, omdat deze niet meer nodig is. Het systeem toont de bestellingen nog eens met daarbij degene die verwijderd is in de zijbalk, met hoeveel deze terugbetaald moet worden.
Alternatieven:
- Extra bestelling: iemand wou nog iets extra bestellen, maar kan dit zelf niet doen. De admin past het aantal aan. Het systeem toont in de zijbalk “te betalen” van die persoon.

2.1.2.21. *Use case Groepsbestelling Uitvoeren – Siebe Sampermans*

Functionaliteit:
Als admin kan ik groepsbestellingen uitvoeren.



Voorwaarde:
Als de groepsbestelling gesloten is.
Als de admin is ingelogd.

Normaal Verloop:
Het systeem laat de bestellijst zien. De admin klikt op bestellen en de admin gaat verder naar het betalen. Het systeem geeft nu een bestelbon voor de leverancier en een bestellijst voor de admin weer. De admin kan deze allebei afdrukken.

Alternatieven:
- Verdeellijst genereren: switch naar use case ‘verdeellijst genereren’.

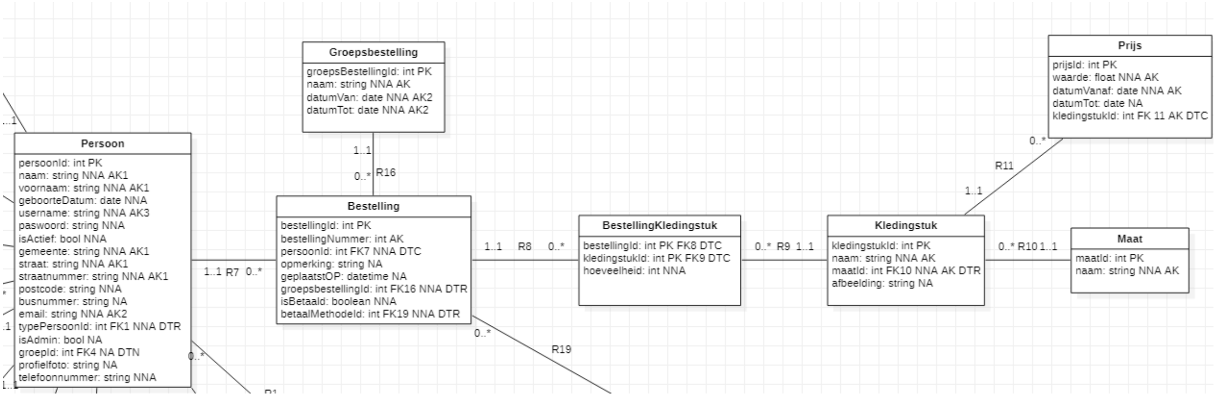
2.1.2.22. Use case verdeellijst genereren – Kyan Collier

Functionaliteit:
Als admin kan ik een verdeellijst genereren.

Voorwaarde:
De bestelling is uitgevoerd.
De admin is ingelogd.

Normaal Verloop:
Het systeem toont alle bestelde kledingstukken per persoon voor de geselecteerde groepsbestelling. De actor selecteert “lijst afdrukken”. Het systeem toont een voorbeeld van de geprinte versie en vraagt om te bevestigen. De actor bevestigt en print de lijst af.

Alternatieven:
- Lijst exporteren: de actor kiest “lijst exporteren” in plaats van “lijst afdrukken”. De verdeellijst wordt dan opgeslagen als PDF.



2.2. Niet-functionele eisen

Voor deze webapplicatie moeten we natuurlijk ook rekening houden met hoe het systeem de functionele eisen kan voldoen. Daarom is het belangrijk dat we ook rekening houden met de volgende niet-functionele eisen.

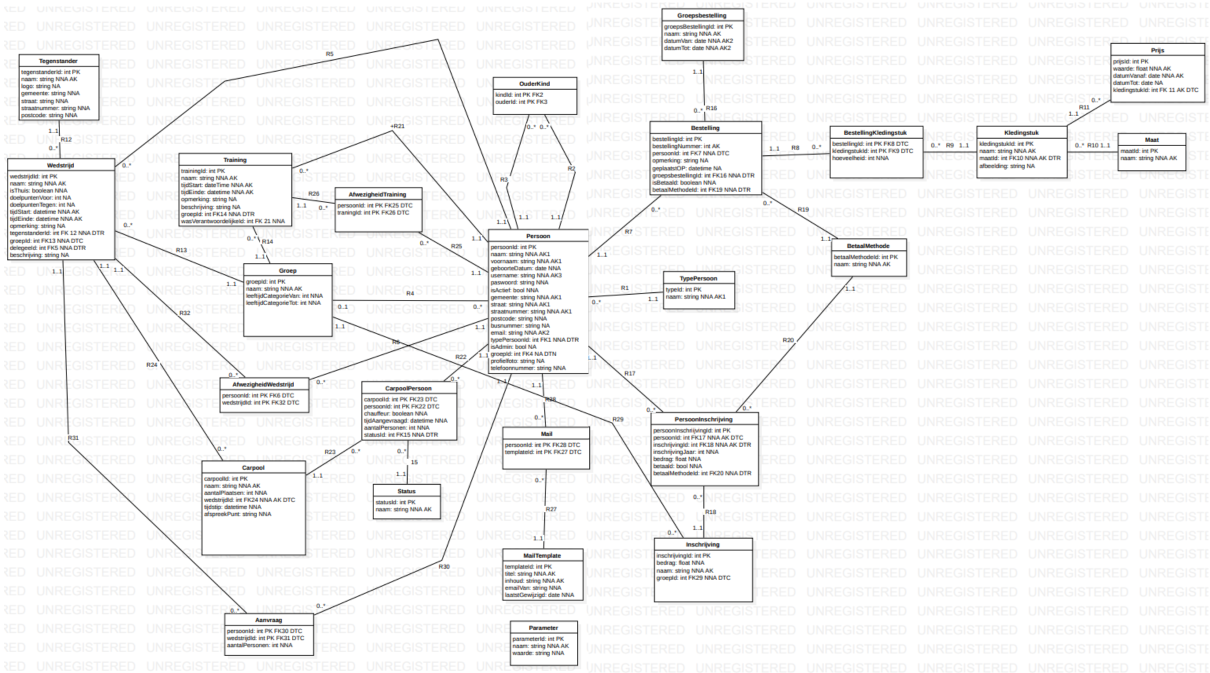
Ten eerste is *usability* zeer belangrijk in onze applicatie. Het doel van dit project is immers om het leven van de gebruikers makkelijker te maken. Daarom is het belangrijk dat de applicatie gebruiksvriendelijk is, zodat het doel echt bereikt kan worden. Daarom zorgen we ervoor dat de navigatie binnen het systeem vanzelfsprekend is en dat je niet lang moet zoeken om te vinden wat je nodig hebt binnen de app.

Naast *usability* is *performance efficiency* ook cruciaal. We moeten er immers voor zorgen dat de verschillende pagina's niet te lang laden. We willen immers dat de gebruiker de applicatie snel kan gebruiken om kleine zaken te regelen (zoals planning, carpool,…), waardoor snelheid zeer belangrijk is. Dit proberen we te bereiken door een database te creëren die efficiënt werkt. Het systeem moet piekperiodes aankunnen om aan deze eis te voldoen.

Daarnaast moesten we ook rekening houden met *functional suitability*. Hier hebben we reeds hard aan gewerkt door prototypes te maken en deze te tonen aan de klant. Deze geeft dan wat opmerkingen waarmee we aan de slag gaan om de applicatie te verbeteren. Zo zorgen we ervoor dat de functies compleet zijn en ook nuttig. Zo zijn er geen functies in onze applicatie die de gebruikers totaal niet nodig hebben.

Ten slotte heeft de klant ook gevraagd om de kleuren van KVV Rauw te gebruiken in de applicatie. Daarom gebruiken we rood als accentkleur binnen de applicatie, terwijl de overige UI-elementen hoofdzakelijk in zwart-wit worden vormgegeven.

3. Data Model



R1: Een persoon heeft één type en een type kan toegekend worden aan meerdere personen

R2 + R3: Een kind kan meerdere ouders hebben en een ouder kan meerdere kinderen hebben.

R4: Een persoon kan toegekend worden aan één groep en een groep kan toegekend worden aan meerdere personen.

R5: Een persoon kan als delegee worden aangeduid voor meerdere wedstrijden en een wedstrijd heeft één delegee.

R6 + R32: Een persoon kan afwezig zijn bij meerdere wedstrijden en een wedstrijd kan meerdere afwezige personen hebben.

R7: Een persoon kan meerdere bestellingen plaatsen en een bestelling behoort toe aan één persoon.

R8 + R9: Een bestelling kan meerdere kledingstukken bevatten en een kledingstuk kan aan meerdere bestelling toebehoren.

R10: Een kledingstuk heeft één maat en een maat kan aan meerdere kledingstukken toebehoren.

R11: Een kledingstuk kan meerdere prijsrecords bevatten en een prijsrecord behoort toe aan één kledingstuk.

R12: Een tegenstander kan aan meerdere wedstrijden toebehoren en een wedstrijd heeft één tegenstander.

R13: Een groep kan meerdere wedstrijden spelen en een wedstrijd behoort toe aan één groep.
R14: Een groep kan aan meerdere trainingen toebehoren en een training behoort toe aan één groep.

R15: Een persoon heeft per carpool één status en een status kan toebehoren aan meerdere personen.

R16: Een bestelling behoort toe aan één groepsbestelling en een groepsbestelling bestaat uit meerdere bestellingen.

R17 + R18: Een persoon kan meerdere inschrijvingen hebben en een inschrijving kan aan meerdere personen toebehoren.

R19: Een bestelling maakt gebruik van één betaalmethode en een betaalmethode kan gebruikt worden door meerdere bestellingen

R20: Een inschrijving van een persoon maakt gebruik van één betaalmethode en een betaalmethode kan gebruikt worden door meerdere persoonsinschrijvingen.

R21: Een persoon kan als wasverantwoordelijke worden aangeduid voor meerdere trainingen en een training heeft één wasverantwoordelijke.

R22 + R23: Een carpool kan meerdere personen bevatten en een persoon kan aan meerdere carpools toebehoren.

R24: Een wedstrijd kan meerdere carpools bevatten en een carpool behoort toe aan één wedstrijd.

R25 + R26: Een persoon kan afwezig zijn bij meerdere trainingen en een training kan meerdere afwezige personen hebben.

R27 + R28: Een persoon kan mails ontvangen van meerdere templates en een template kan verstuurd worden naar meerdere personen.

R29: Een groep kan toebehoren aan meerdere inschrijvingen en een inschrijving is verbonden met één groep.

R30 + R31: Bij een wedstrijd kan er door meerdere personen een carpool aangevraagd worden en een persoon kan een carpool aanvragen bij meerdere wedstrijden.

4. Prioriteit per functionaliteit

Must have:

- Account aanmaken
- Inloggen
- Kind inschrijven
- Kledij bestellen
- Carpool registreren
- Carpool bevestigen
- Mail versturen

Should have:

- Betaling registreren
- Lidgeld bepalen
- Planning bekijken
- Activiteiten beheren
- Afwezigheid melden
- Kledij registreren

Could have:

- Groepsbestelling initiëren
- Groepsbestelling uitvoeren
- Lidgeld betalen
- Kledijstuk beheren
- Bestellingen beheren
- Verdeellijst genereren

Would like:

- Account bewerken
- Mails beheren
- Gebruikers beheren

5. Conclusie

In dit analyse- en ontwerprapport is een compleet overzicht gegeven van de functionele en niet-functionele eisen voor de webapplicatie van KVV Rauw. Aan de hand van gesprekken met de klant en interne analyse zijn alle gebruikersrollen, gebruikssituaties, gegevensstructuren en processen grondig vastgelegd. Het datamodel geeft bovendien een helder overzicht van de interacties tussen de diverse entiteiten, wat cruciaal is voor een juiste en effectieve uitvoering.

De MoSCoW-prioritering verduidelijkt welke functies cruciaal zijn voor de eerste oplevering en welke later kunnen worden verder uitgebreid. Dit stelt het ontwikkelingsteam in staat om geleidelijk en doelgericht te werken. Dit document biedt een stevig fundament voor de fase van implementatie en ontwikkeling, en garandeert dat alle betrokkenen een gemeenschappelijk en duidelijk inzicht hebben in de verwachtingen van de projectscope.

6. Glossary

Actor
Een rol die een gebruiker vervult binnen het systeem, zoals speler, ouder, trainer of admin.

Data Model
Een schematische voorstelling van alle gegevens en hun onderlinge relaties binnen de applicatie.

Functionele eisen
Eisen die beschrijven wat een systeem moet kunnen of welke acties gebruikers moeten kunnen uitvoeren.

Niet-functionele eisen
Eisen die bepalen hoe goed het systeem moet werken (zoals snelheid, gebruiksvriendelijkheid, veiligheid).

MoSCoW-methode
Een methode om prioriteiten toe te wijzen aan functionaliteiten: Must have, Should have, Could have, Would like.

Use Case
Een beschrijving van hoe een gebruiker een bepaalde taak uitvoert binnen het systeem.

7. Bibliografie

Binnen dit analyse- en ontwerprapport zijn geen externe informatiebronnen toegepast. Alle beschreven functionaliteiten, modellen en eisen zijn ontwikkeld op basis van interne analyse, gesprekken met de opdrachtgever en de eigen expertise van het projectteam.