|  |  |
| --- | --- |
|  | |
| Analyse “Platte Berg”  Analyse-en-ontwerprapport | |
| **Teamnr: TVE1**  Halim Haidari, Eva Burghoorn, Xander Put, Jasper Vermissen, Seppe Meeus | 2de fase IT Factory |
|  |
| Academiejaar 2021-2022  Campus Geel, Kleinhoefstraat 4, BE-2440 Geel |



Inhoudstafel

[Inhoudstafel 2](#_Toc1798114202)

[Inleiding 3](#_Toc1517541431)

[1 Opdrachtbeschrijving 4](#_Toc2134531083)

[1.1 Achtergrondinformatie 5](#_Toc1292121508)

[1.2 Doelstellingen en doelgroepen 5](#_Toc1623845018)

[2 Eisenanalyse 5](#_Toc1524344320)

[2.1 Functionele eisen 6](#_Toc1439715689)

[2.1.1 Use Case Diagram 6](#_Toc1959497296)

[2.1.2 Homepagina 6](#_Toc410773541)

[2.1.2.1 Niet-lid 6](#_Toc672347223)

[2.1.2.2 Lid 7](#_Toc382139524)

[2.1.2.3 Bestuur 9](#_Toc1905802474)

[2.1.3 Primaire Use Case Beschrijvingen met prototypes 10](#_Toc1413080831)

[2.1.3.1 Lidmaatschap aanmaken (Halim) 10](#_Toc1041924898)

[2.1.3.2 Lidmaatschap vernieuwen (Xander) 14](#_Toc1652500218)

[2.1.3.3 Eigen gegevens raadplegen (Eva) 17](#_Toc651786030)

[2.1.3.4 Clubkledij aanvragen (Xander) 20](#_Toc1998822001)

[2.1.3.5 Ritplanning maken (Jasper) 23](#_Toc464111015)

[2.1.3.6 Aanmelden als ritverkenner (Eva) 24](#_Toc905353983)

[2.1.3.7 Ritdetails ingeven (Seppe) 25](#_Toc425108537)

[2.1.3.8 Planning bekijken (Halim) 26](#_Toc530811234)

[2.1.3.9 Aanwezigheden bijhouden (Seppe) 28](#_Toc1284755967)

[2.1.3.10 Puntenpolitiek beheren (Jasper) 29](#_Toc224479806)

[2.1.3.11 Evenementen plannen (Halim) 30](#_Toc1054867650)

[2.1.3.12 Wekelijkse planning doorsturen (Seppe) 33](#_Toc742037970)

[2.1.4 Secundaire Use Case Beschrijvingen met prototypes 34](#_Toc1353609157)

[2.1.4.1 Inloggen 34](#_Toc1937780719)

[2.1.4.2 Ploegen beheren 34](#_Toc606951687)

[2.1.4.3 Gebruikers beheren 35](#_Toc1039374407)

[2.1.4.4 Geslachtstypes beheren 37](#_Toc1017737554)

[2.1.4.5 Kledij beheren 38](#_Toc113712792)

[2.1.4.6 Kledingmaten beheren 39](#_Toc258254484)

[2.1.4.7 Ondergrond beheren 39](#_Toc1803727842)

[2.1.4.8 Prijs lidmaatschap beheren 40](#_Toc781095830)

[2.1.4.9 Kortingen beheren 41](#_Toc1635147478)

[2.2 Niet-functionele eisen 42](#_Toc1916755663)

[3 Prioriteit per functionaliteit 44](#_Toc1392469760)

# Inleiding

In dit verslag wordt eerst de opdracht beschreven en vervolgens de functionele eisenanalyse besproken. Er wordt weergegeven wat de fietsclub “Platte Berg” wil verbeteren en hoe dit zal gerealiseerd worden. Er is met het bestuur van de “Platte Berg” samengezeten om de dit zo goed mogelijk naar hun hand te zetten.

# Opdrachtbeschrijving

In dit hoofdstuk wordt er eerst een beschrijving gegeven van het systeem dat Fietsclub de “Platte Berg” momenteel gebruikt. Daarna wordt de analyse voor het nieuwe systeem toegelicht.

## Achtergrondinformatie

De “Platte Berg - PB” is een fietsclub die fietstochten op de weg en in het bos organiseert voor elk enthousiast lid die een passie heeft voor fietsen.  
Op elke zaterdag en zondag worden er fietsritten georganiseerd voor verschillende niveaus door ritverkenners.

Momenteel wordt er gebruikgemaakt van WhatsApp-groepen om een planning op te maken voor de ritverkenners en papieren worden rondgegeven om aanwezigheden te noteren. Daarnaast worden er Excel-bestanden beheerd voor kleding en kledijbonnen.

De methode die nu gebruikt wordt, brengt wat problemen met zich mee. Door alles manueel vast te leggen worden er regelmatig foutjes gemaakt. De planning opmaken kan erg lang duren en kruisjes worden verkeerd gezet op de aanwezigheidslijsten. Hierdoor worden kledijbonnen verkeerd uitgedeeld. Door de Excel-lijsten handmatig bij te houden, worden verkeerde bestellingen doorgegeven en ontstaan er frustraties tussen de leden en het bestuur.

## Doelstellingen en doelgroepen

Wat er met het nieuwe systeem bereikt wil worden, is een vlottere communicatie zonder conflicten. Punten worden zorgvuldig bijgehouden, want de puntenverdeling wordt geautomatiseerd.   
Dit zorgt ervoor dat de puntentelling voor de kledijbonnen correct verloopt. Hierdoor zullen er geen fouten gemaakt worden en zal het efficiënter zijn.  
De kledij zal online besteld kunnen worden.

De betrokken partijen hebben volgende voordelen aan het systeem:

* *Niet-leden* kunnen zich online inschrijven. Daarnaast kunnen ze de algemene planning zien en twee keer gratis meerijden.
* *Leden* kunnen makkelijk zich herinschrijven. De planning voor de ritten en evenementen kunnen ze makkelijker raadplegen. Je kan online kledij bestellen en je punten bekijken. Ook wordt er wekelijks een ritplanning per email doorgestuurd.
* *De ritverkenners* worden makkelijker en sneller ingedeeld in de planning en kunnen op een efficiëntere manier de aanwezigheden noteren.
* Het bestuur kan de lidmaatschappen, kledijbonnen, aankoop van kledij en andere administratie beter en makkelijker onderhouden.

# Eisenanalyse

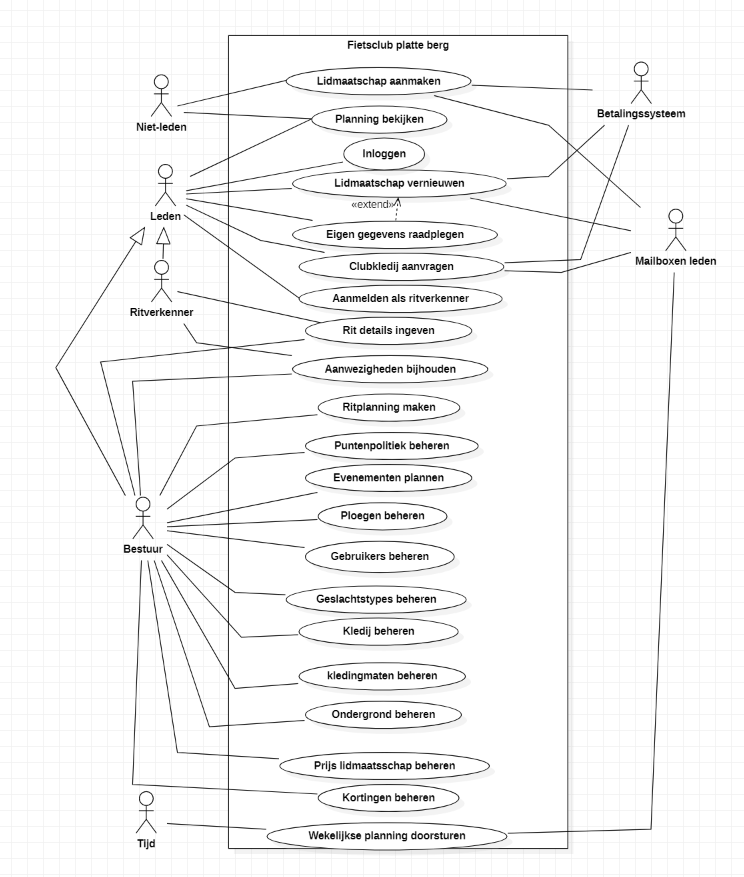
Inleidende informatie voor dit onderdeel van het rapport

## Functionele eisen

In het eerste deel van dit hoofdstuk wordt er eerst besproken wat een use case diagram is. Daarna volgt het use case diagram van het nieuwe systeem. In het tweede deel wordt hier dieper op ingegaan door de individuele use cases te bespreken.

### Use Case Diagram

Het doel van een use case diagram is het illustreren van interacties die gebeuren tussen het systeem en de actoren. Het systeem is in dit geval de software achter de webapplicatie van Fietsclub “Platte Berg”. De actoren, zoals in onderstaande illustratie te zien, zijn de figuren die buiten het kader staan. In het kader staan de use cases, te herkennen aan hun ovale vorm. De lijnen tussen de actoren en use cases zijn de relaties tussen de twee.



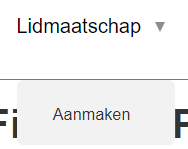
### Homepagina

#### **Niet-lid**

*Functionality:*   
*Als niet-lid, kan ik de homepagina bekijken.*

*Screens:*

*H1 Home Niet-Lid*



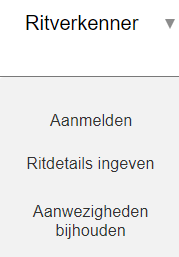
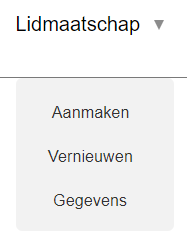
|  |  |
| --- | --- |
| Action | Explanation |
| Home | Naar pagina “H1 Home Niet-Lid” |
| Lidmaatschap aanmaken | Naar pagina “1. Lidmaatschap aanmaken” |
| Planning | Naar pagina “8. Planning bekijken” |
| Inloggen | Naar pagina “13. Inloggen” |

#### **Lid**

*Functionality:*   
*Als lid, kan ik de homepagina van een lid bekijken.*

*Screens*:

*H2 Home Lid*  

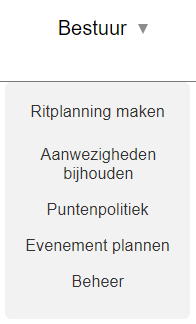
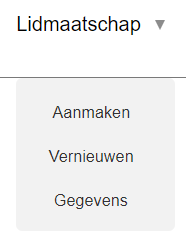
|  |  |
| --- | --- |
| Action | Explanation |
| Home | Naar pagina “H2 Home Lid” |
| Lidmaatschap - Aanmaken | Naar pagina “1. Lidmaatschap aanmaken” |
| Lidmaatschap - Vernieuwen | Naar pagina “2. Lidmaatschap vernieuwen” |
| Lidmaatschap - Gegevens | Naar pagina “3. Eigen gegevens raadplegen” |
| Clubkledij | Naar pagina “4. Clubkledij aanvragen” |
| Ritverkenner - Aanmelden | Naar pagina “6. Aanmelden als ritverkenner” |
| Ritverkenner - Ritdetails Ingeven | Naar pagina “7. Ritdetails ingeven” |
| Ritverkenner – Aanwezigheden bijhouden | Naar pagina “9. Aanwezigheden bijhouden” |
| Planning | Naar pagina “8. Planning bekijken” |

#### **Bestuur**

*Functionality:*   
*Als bestuur, kan ik de homepagina van het bestuur bekijken.*

*Screens*:

*H3 Home Bestuur*  

|  |  |
| --- | --- |
| Action | Explanation |
| Home | Naar pagina “H2 Home Lid” |
| Lidmaatschap - Aanmaken | Naar pagina “1. Lidmaatschap aanmaken” |
| Lidmaatschap - Vernieuwen | Naar pagina “2. Lidmaatschap vernieuwen” |
| Lidmaatschap - Gegevens | Naar pagina “3. Eigen gegevens raadplegen” |
| Clubkledij | Naar pagina “4. Clubkledij aanvragen” |
| Ritverkenner - Aanmelden | Naar pagina “6. Aanmelden als ritverkenner” |
| Ritverkenner - Ritdetails Ingeven | Naar pagina “7. Ritdetails ingeven” |
| Ritverkenner – Aanwezigheden bijhouden | Naar pagina “9. Aanwezigheden bijhouden” |
| Planning | Naar pagina “8. Planning bekijken” |
| Bestuur – Ritplanning maken | Naar pagina “5. Ritplanning maken” |
| Bestuur – Aanwezigheden bijhouden | Naar pagina “9. Aanwezigheden bijhouden” |
| Bestuur - Puntenpolitiek | Naar pagina “10. Puntenpolitiek beheren” |
| Bestuur – Evenement plannen | Naar pagina “11. Evenement plannen” |
| Bestuur - Beheer | Naar pagina “12. Beheer” |

### Primaire Use Case Beschrijvingen met prototypes

In dit onderdeel wordt elke functionaliteit verder uitgewerkt door middel van een use case beschrijving. In deze omschrijving wordt stap voor stap bekeken hoe een bepaalde functionaliteit zich afspeelt. Elke interactie tussen het systeem en de gebruikers wordt hierbij zo nauwkeurig mogelijk beschreven. Ook worden er eventuele precondities en alternatieven besproken.

#### **Lidmaatschap aanmaken (Halim)**

*Functionality:*   
*Als niet-lid, kan ik een lidmaatschap aanmaken.*

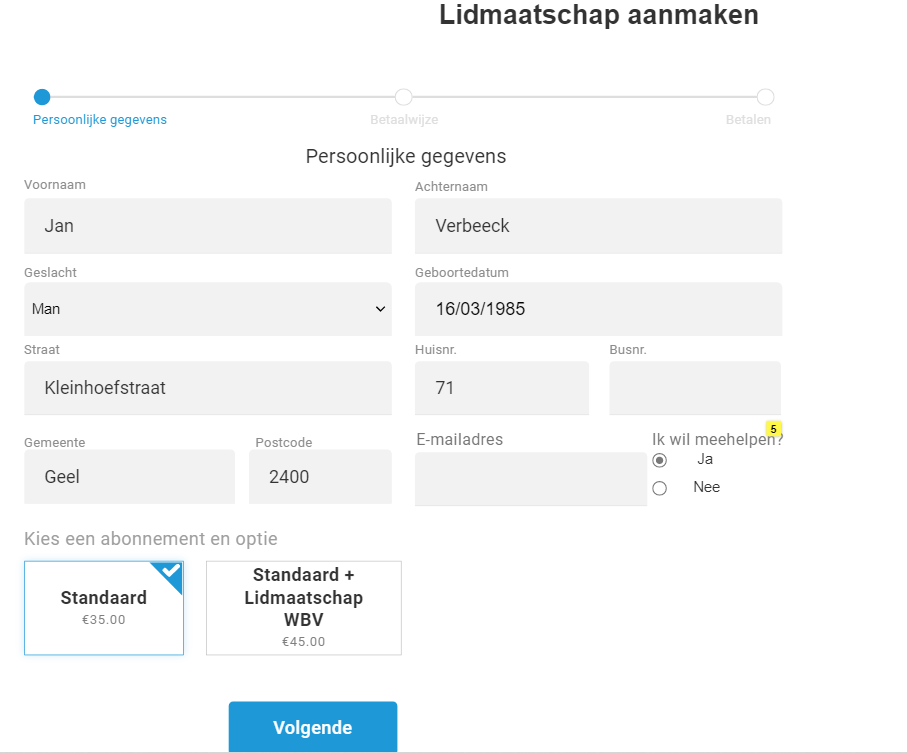
*Normal flow:*   
Systeem toont het aanmeldingsformulier, de optie om ook lid te zijn van WBV en de prijs.   
Actor vult persoonsgegevens in en bevestigt.   
Systeem toont betalingsmogelijkheden.   
Actor kiest betalingssysteem.   
Systeem schakelt over naar het externe betalingssysteem dat de betaling afhandelt. Het betalingssysteem geeft in het systeem aan dat de betaling succesvol was.   
Systeem geeft een melding van succesvolle registratie en stuurt een bevestigingsmail naar de actor met een uitnodigingslink om toegang te krijgen tot zijn/haar persoonlijk portaal.

*Alternatives:*

* Betaling niet succesvol:   
  Systeem geeft melding hiervan en gaat terug naar het begin.
* Al lid:   
  Als de actor al een lid is, verwijst het systeem naar lidmaatschap vernieuwen pagina.

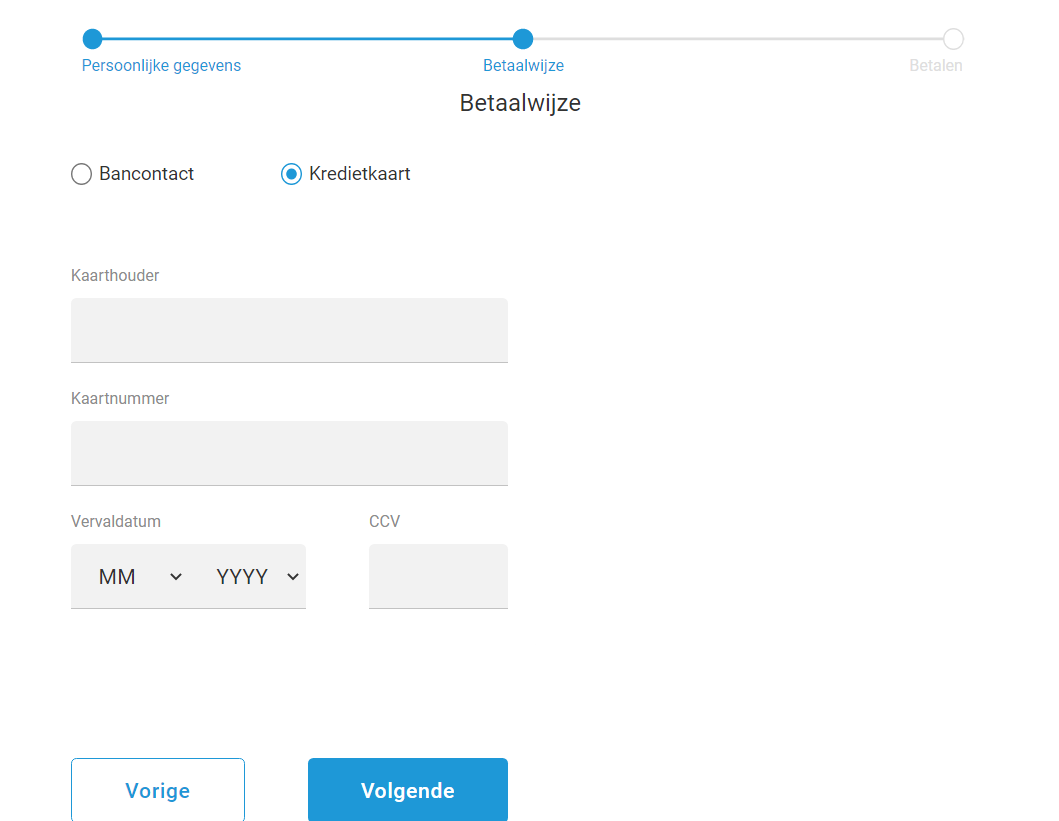
*Screens*:

*1.1 Persoonlijke gegevens*



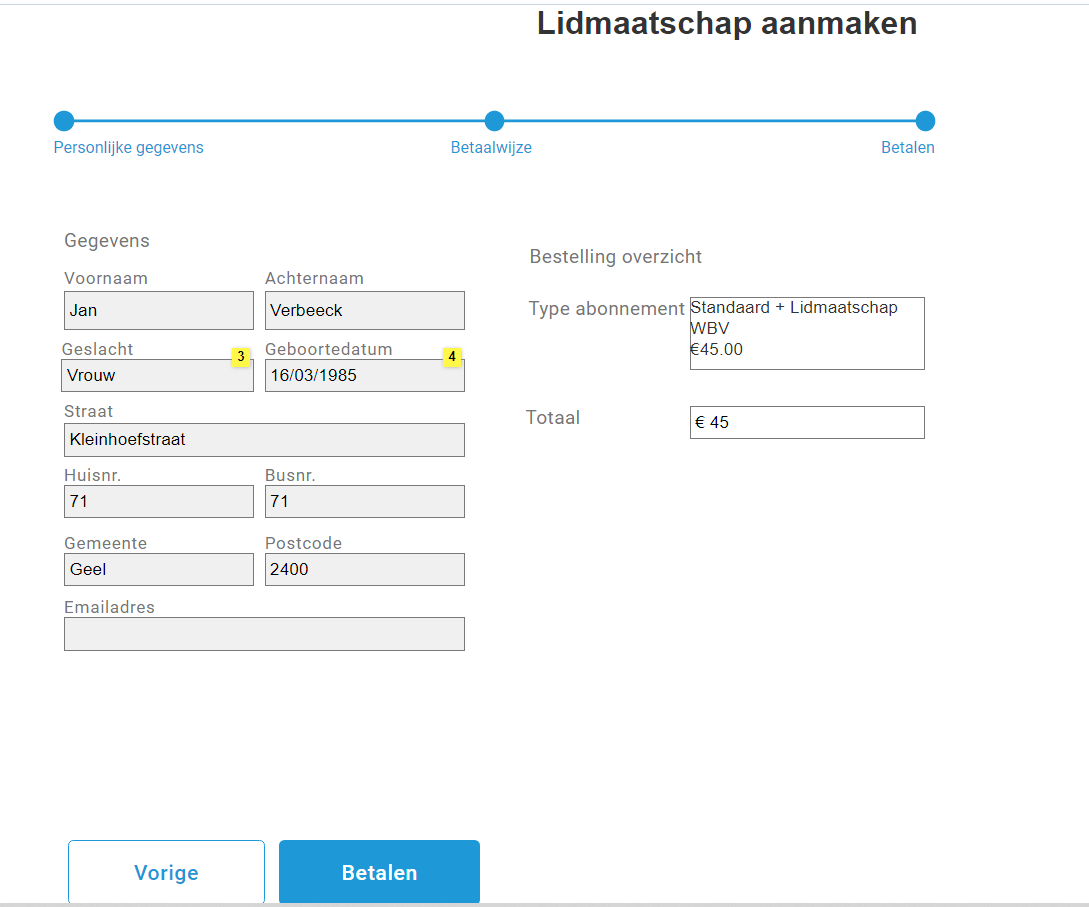
|  |  |
| --- | --- |
| Action | Explanation |
| Volgende | Naar tabblad “1.2 Betaalwijze” |
| 5. Ik wil meehelpen | De gebruiker kan één optie kiezen. |

*1.2 Betaalwijze*



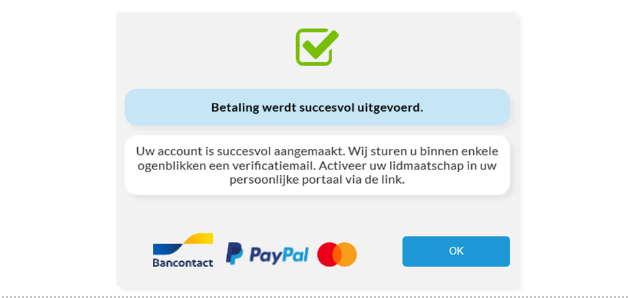
|  |  |
| --- | --- |
| Action | Explanation |
| Volgende | Naar pagina “1.3 Bestelling overzicht” |
| Betaalwijze | Afhankelijk van het type betalingssysteem, worden de overeenkomstige velden weergegeven. |
| Vorige | Naar pagina “1.1 Personlijke gegevens” |

*1.3 Bestelling overzicht*



|  |  |
| --- | --- |
| Action | Explanation |
| Betalen | Naar externe betalingssysteem “Bancontact/Kredietkaart”. Afhankelijk van het gekozen betalingssysteem. |
| Vorige | Naar pagina “1.2 Betaalwijze” |

*1.4 Betaling succesvol*



|  |  |
| --- | --- |
| Action | Explanation |
| Bevestiging | Betaling succesvol werd uitvoert, pop-up weergegeven. |
| OK | Naar homepagina |

#### **Lidmaatschap vernieuwen (Xander)**

*Functionality:*   
*Als lid, kan ik mijn lidmaatschap verlengen.*

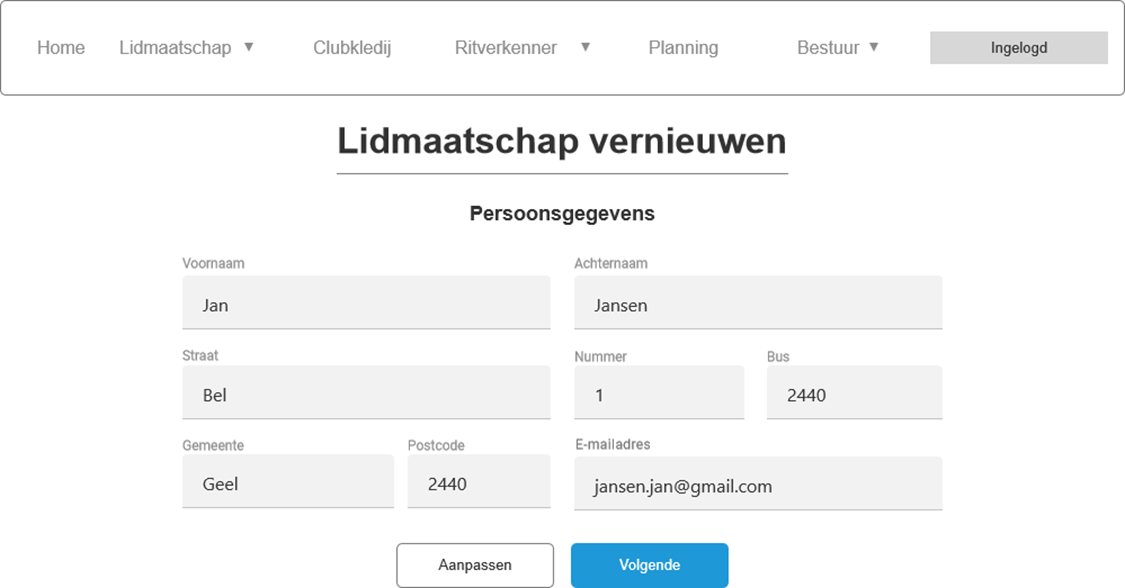
*Preconditions:*   
*Actor is ingelogd.*

*Normal flow:*  
*Systeem geeft persoonsgegevens weer en geeft een optie om deze aan te passen, ook wordt de optie aangeboden om lid te zijn van het WBV.*  
*Actor bevestigt.*  
*Systeem toont prijs en mogelijke betalingsmogelijkheden.*  
*Actor kiest betalingssysteem.*  
*Systeem verwijst door naar gekozen betalingssysteem.*  
*Het betalingssysteem verwijst terug door naar het systeem en bevestigt succesvolle betaling.*  
*Systeem bevestigt inschrijving en stuurt ook een bevestigingsmail naar mailbox van actor.*

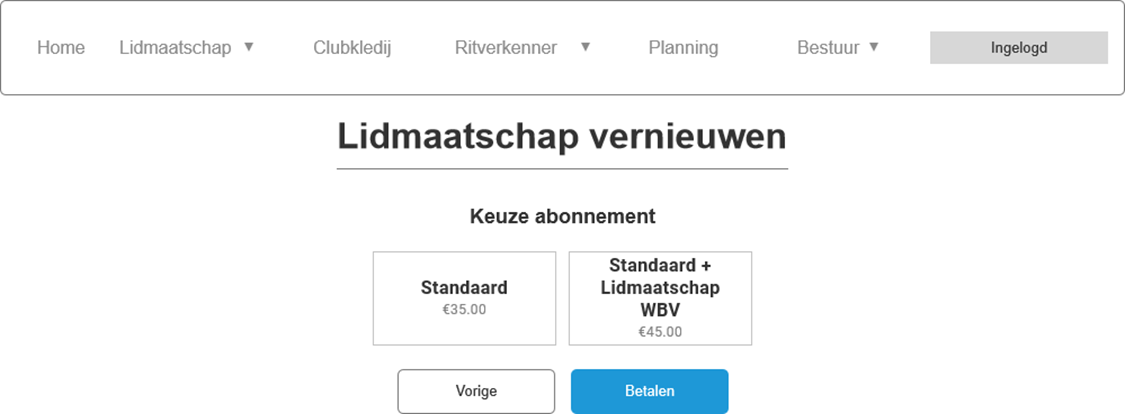
*Alternatives:*

* Betaling niet succesvol:   
  Systeem geeft melding hiervan en gaat terug naar het begin.
* Actor wil gegevens aanpassen:  
  Actor wordt doorverwezen naar use case “Eigen gegevens raadplegen”

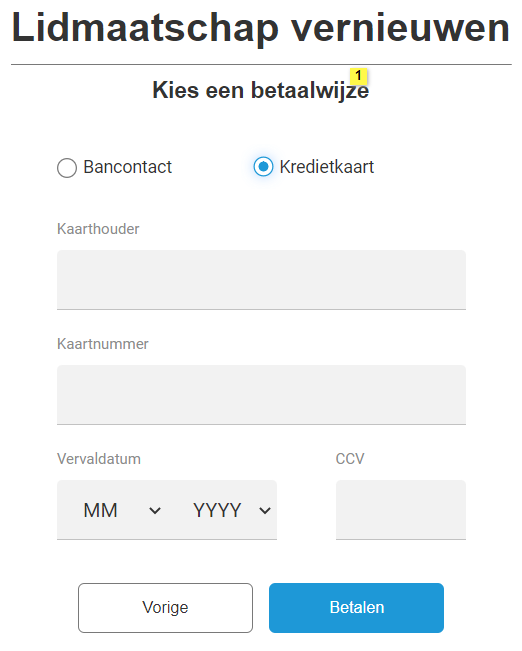
*Screens*:

*2.1 Bevestiging persoonsgegevens*  


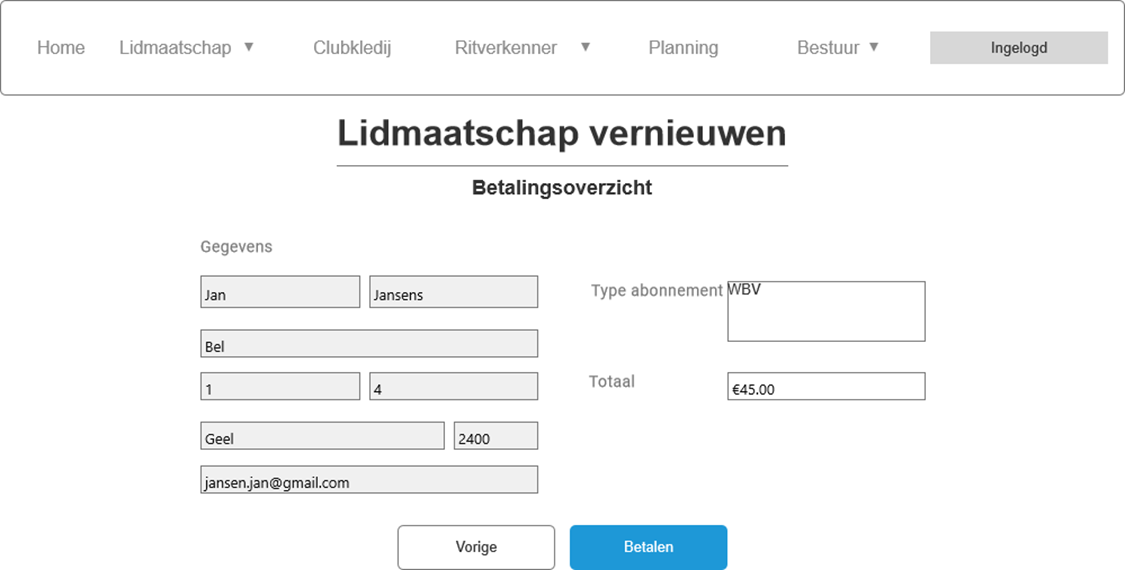
|  |  |
| --- | --- |
| Action | Explanation |
| Aanpassen | Naar pagina “3. Eigen gegevens raadplegen” |
| Volgende | Naar pagina “2.2 Keuze abonnement” |

*2.2 Keuze abonnement*  


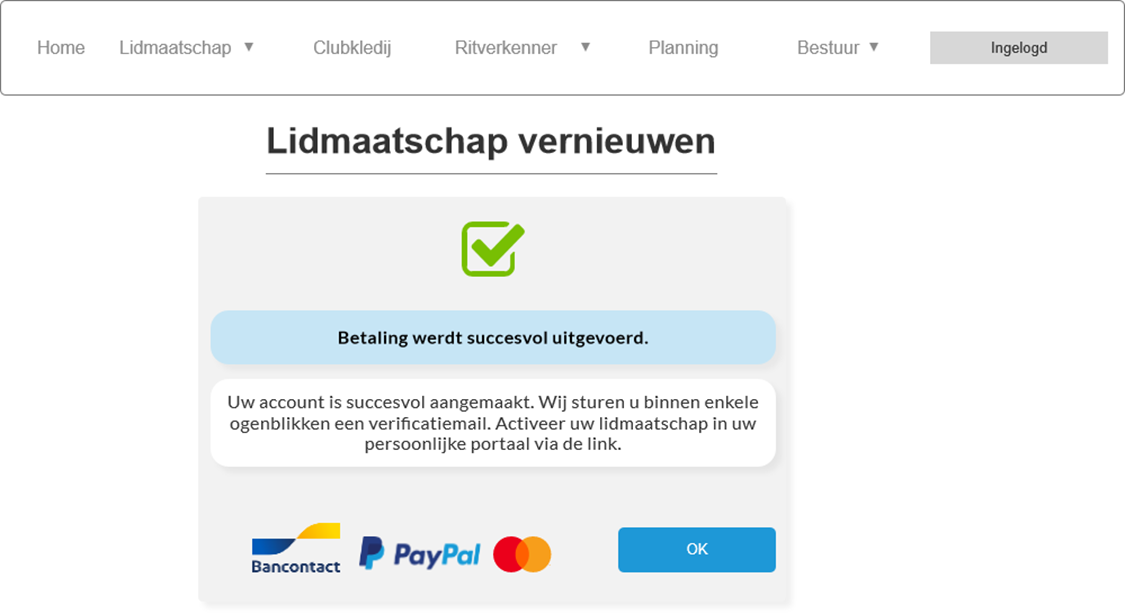
|  |  |
| --- | --- |
| Action | Explanation |
| Vorige | Naar pagina “2.1 Bevestiging persoonsgegevens” |
| Betalen | Naar pagina “2.3 Keuze betaalwijze” |

*2.3 Keuze betaalwijze*  


|  |  |
| --- | --- |
| Action | Explanation |
| Vorige | Naar pagina “2.2 Keuze abonnement” |
| Betalen | Naar pagina “2.4 Betaaloverzicht” |
| 1 | Na het kiezen van een betaalwijze worden de juiste velden hiervoor getoond. |

*2.4 Betaaloverzicht*  


|  |  |
| --- | --- |
| Action | Explanation |
| Vorige | Naar pagina “2.3 Keuze betaalwijze” |
| Volgende | Naar pagina “2.5 Betalingsbevestiging” |

*2.5 Betalingsbevestiging*  


|  |  |
| --- | --- |
| Action | Explanation |
| OK | Naar homepagina |

#### **Eigen gegevens raadplegen (Eva)**

*Functionality:*

Als lid, kan ik mijn eigen gegevens raadplegen.

*Preconditions:*

Actor is ingelogd.

*Normal flow:*  
*Het systeem geeft een lijst met de persoonsgegevens, aantal punten voor kledijbonnen, de aanwezigheden van de ritten en evenementen en de kledij bestellingen van de actor weer.*

*Alternatives:*

* Actor wil persoonsgegevens aanpassen:  
  Systeem geeft velden aan die de actor kan aanpassen. De actor vult aangepaste informatie in en bevestigt dit. Het systeem bevestigt de verandering.

*Screens:*

*3.1 Gegevens raadplegen*

Table

Description automatically generated

|  |  |
| --- | --- |
| Action | Explanation |
| Gegevens aanpassen | Naar pagina “3.2 Gegevens aanpassen” |
| Mijn gereden ritten | Naar pagina “3.3 Gereden ritten” |

*3.2 Gegevens aanpassen*  
Graphical user interface, table

Description automatically generated

|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Slaat aangepaste gegevens op |
| Pas aanpassingen toe | Naar pagina “3.1 Gegevens raadplegen” |

*3.3 Gereden ritten*  
  


|  |  |
| --- | --- |
| Action | Explanation |
| Mijn gereden ritten | Persoonlijk gereden ritten worden weergegeven in een nieuw tabblad |

#### **Clubkledij aanvragen (Xander)**

*Functionality:*   
*Als lid, kan ik clubkledij aanvragen.*

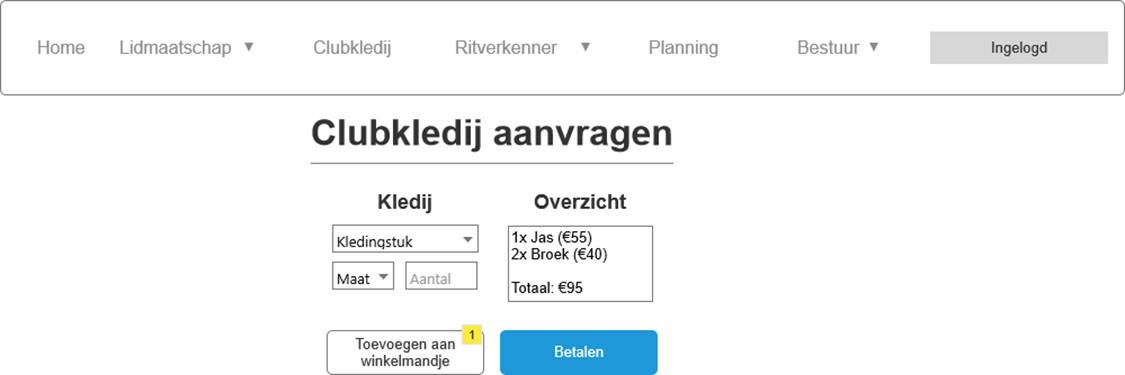
*Preconditions:*   
Actor is ingelogd.

*Normal flow:*  
*Systeem toont lijst van de actieve kledingstukken.*  
*Actor kiest kledingstuk, vult aantal en maat in.*  
*Systeem toont prijs, de mogelijkheid om kledijbonnen in te zetten, en mogelijke betalingsmogelijkheden.*  
*Actor kiest betalingssysteem.*  
*Systeem verwijst door naar gekozen betalingssysteem.*  
*Het betalingssysteem verwijst terug door naar het systeem en bevestigt succesvolle betaling.*  
*Systeem bevestigt aankoop en stuurt een bevestigingsmail naar mailbox van actor.*

*Alternatives:*

* Betaling niet succesvol:   
  Systeem geeft melding hiervan, gaat terug naar het begin en bestelling wordt geannuleerd.

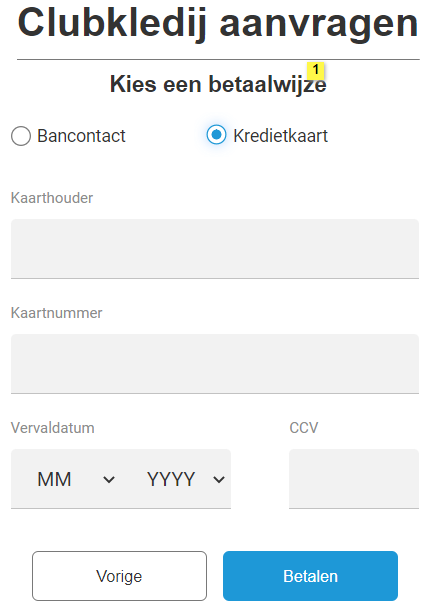
*Screens:*

*4.1 Keuze kledij*  


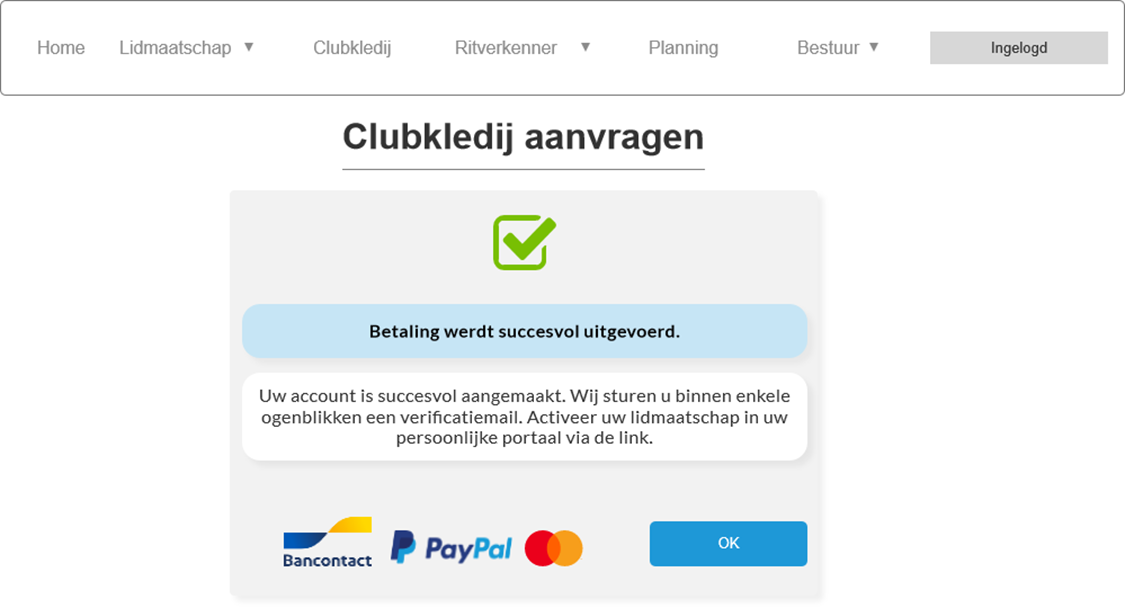
|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Item wordt toegevoegd aan overzicht/winkelmandje en geeft de optie om meer te bestellen |
| Toevoegen aan winkelmandje | Naar pagina “4.1 Keuze kledij” |
| Betalen | Naar pagina “4.2 Kledijbonnen inzetten” |

*4.2 Kledijbonnen inzetten*  


|  |  |
| --- | --- |
| Action | Explanation |
| Vorige | Naar pagina “4.1 Keuze kledij” |
| Volgende | Naar pagina “4.3 Keuze betaalwijze” |

*4.3 Keuze betaalwijze*  


|  |  |
| --- | --- |
| Action | Explanation |
| Vorige | Naar pagina “4.2 Kledijbonnen inzetten” |
| Betalen | Naar pagina “4.4 Betalingsbevestiging” |
| 1 | Na het kiezen van een betaalwijze worden de juiste velden hiervoor getoond. |

*4.4 Betalingsbevestiging*  


|  |  |
| --- | --- |
| Action | Explanation |
| OK | Naar homepagina |

#### **Ritplanning maken (Jasper)**

*Functionality:*

Als bestuur, kan ik een ritplanning maken.

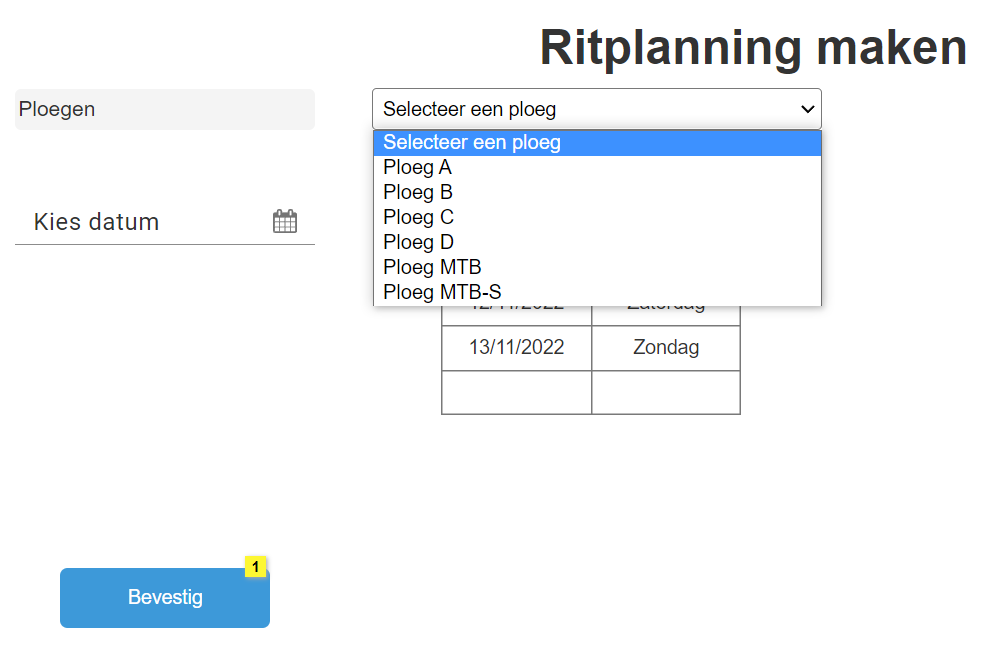
*Precondition:*

Actor is ingelogd.

*Normal flow:*   
Het systeem toont een lijst met de ploegen.  
De actor kiest een ploeg en bevestigt.  
Het systeem toont een kalender.  
De actor selecteert de dagen en tijdstippen waarop deze ploeg zal rijden.  
Het systeem bevestigt.

*Screens:*

*5. Ritplanning maken*



|  |  |
| --- | --- |
| Action | Explanation |
| Kies datum | Als je een datum kiest wordt die toegevoegd in de tabel |
| 1 | Slaagt de gekozen ritten op bij de gekozen ploeg |
| Bevestig | Gaat naar een geüpdatete versie van dezelfde pagina |

#### **Aanmelden als ritverkenner (Eva)**

*Functionality:*

Als lid, kan ik mij aanmelden als ritverkenner.

*Preconditions:*   
Actor is ingelogd.

*Normal flow:*  
Het systeem geeft alle ritten weer die nog geen ritverkenner hebben.   
De actor duidt de ritten aan waarvan hij/zij een ritverkenner wil zijn.  
Systeem bevestigt.

*Screens:*

*6. Aanmelden als ritverkenner*  
Table

Description automatically generated

|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Dient de selectie in die de (nieuwe) ritverkenner maakt |
| Selectie | Gaat naar een geüpdatete versie van dezelfde pagina |

#### **Ritdetails ingeven (Seppe)**

*Functionality:*

Als ritverkenner, kan ik ritdetails ingeven.

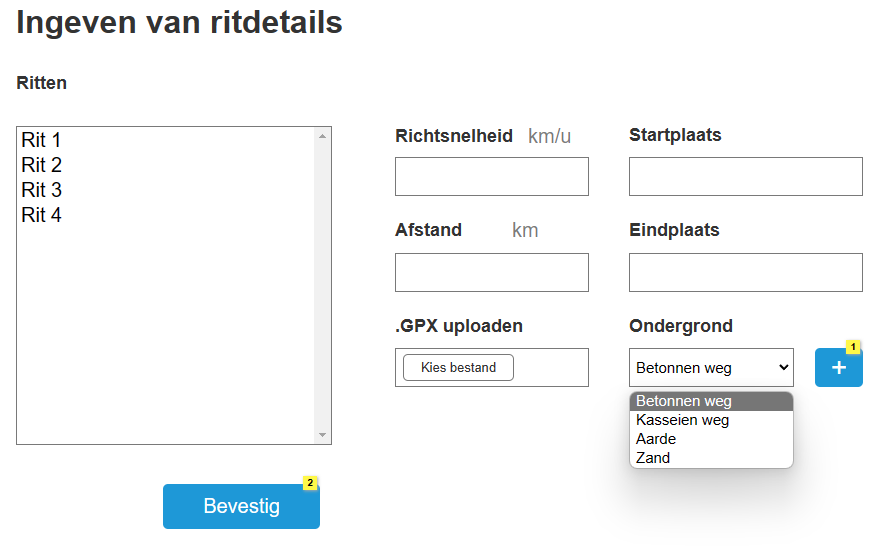
*Precondition:*

Actor is ingelogd.

*Normal flow:*   
Het systeem toont een lijst met opkomende ritten waarvan de actor ritverkenner is.   
De actor kiest een rit, vult de details in en bevestigt.

*Screens:*

*7. Ritdetails ingeven*



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Voegt nieuwe drop down toe om meerdere ondergronden toe te voegen. |
| 2 | Slaagt de gegevens op. |
| Bevestig | Gaat naar een geüpdatete versie van dezelfde pagina |

#### **Planning bekijken (Halim)**

*Functionality:*   
*Als lid en niet-lid kan ik de planning bekijken.*

*Normal flow:*   
Systeem toont een kalenderoverzicht van de geplande ritten, evenementen en tours.   
Actor selecteert een rit, evenement of tour voor meer informatie.

Screens:

*8.1 Planning bekijken - Lijst*



|  |  |
| --- | --- |
| Action | Explanation |
| Planning bekijken -Lijst | Evenementen worden in dynamische gecreëerd lijst weergegeven. |

*8.2 Extra informatie*



|  |  |
| --- | --- |
| Action | Explanation |
| Meer info | Details van het event worden getoond in de hierboven dynamisch gecreëerde tabel. |

#### **Aanwezigheden bijhouden (Seppe)**

*Functionality:*

Als ritverkenner, of als bestuur kan ik aanwezigheden bijhouden.

*Precondition:*

Actor is ingelogd.

*Normal flow:*  
Het systeem toont een lijst van geplande ritten op de huidige datum.  
De actor kiest de rit waarvoor hij aanwezigheden wil bijhouden.  
Het systeem geeft een lijst weer van alle leden.   
De actor duidt de leden aan die aanwezig waren en bevestigt dit.   
Het systeem bevestigt dit en geeft de juiste punten aan de juiste leden.

*Screens*:

*9.1 Beschikbare ritten voor actor*



*9.2 Aanduiden van aanwezige leden*

|  |  |
| --- | --- |
| Action | Explanation |
| 1 | De tabel wordt gefilterd met de naam die wordt ingegeven. |
| 2 | Slaagt de aanwezige leden voor deze rit op. |
| Bevestig | Naar pagina “9.1 Beschikbare ritten voor actor” |

#### **Puntenpolitiek beheren (Jasper)**

*Functionality:*

Als bestuur, kan ik de puntenpolitiek beheren.

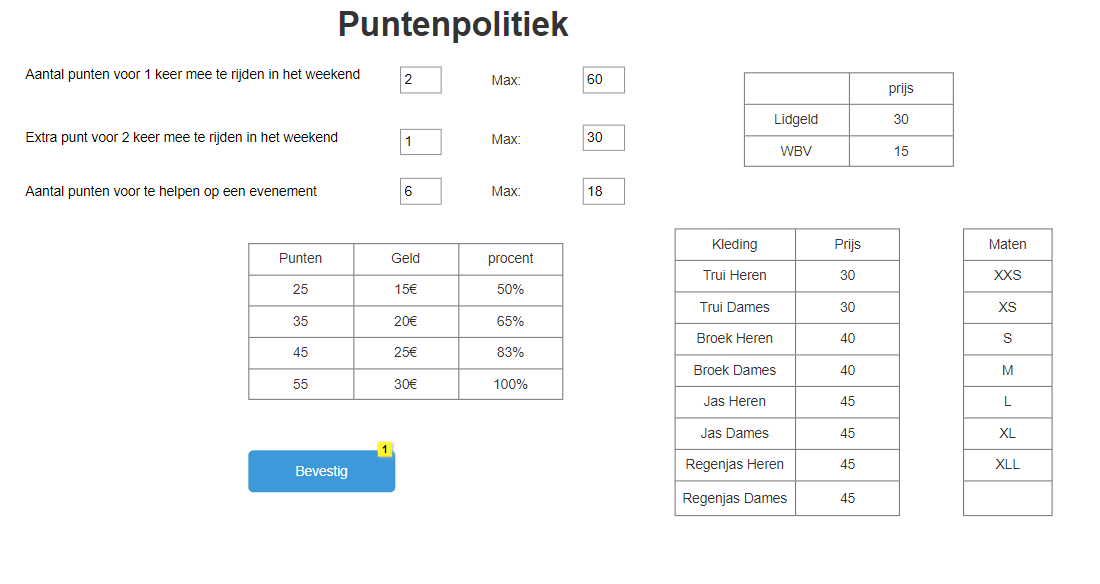
*Precondition:*

Actor is ingelogd.

*Normal flow:*   
Het systeem geeft de momentele puntenpolitiek weer.   
De actor past dit aan tot de nieuwe puntenpolitiek.   
Het systeem past dit toe en geeft de nieuwe puntenpolitiek weer.

*Screens*:

*10. Puntenpolitiek*



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Update de prijzen die veranderd zijn op deze pagina naar de nieuwe prijzen |
| Bevestig | Gaat naar een geüpdatete versie van dezelfde pagina |

#### **Evenementen plannen (Halim)**

*Functionality:*   
Als bestuur, kan ik evenementen plannen.

*Precondition:*   
Actor is ingelogd.

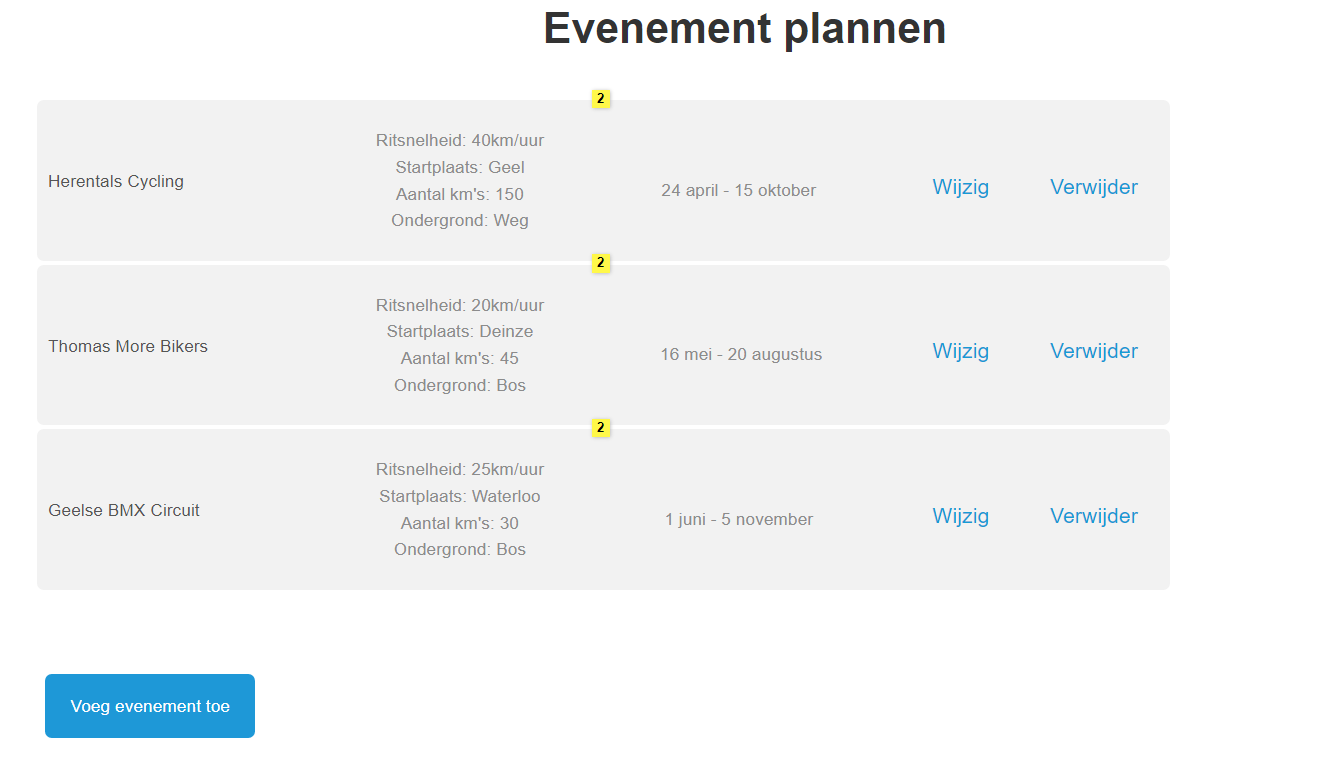
*Normal flow:*   
Het systeem toont een lijst van de evenementen van vorige jaren.   
De actor kiest een evenement uit deze lijst en geeft datum, tijdstip, locatie en terrein (weg, bos) in en bevestigt dit.   
Het systeem bevestigt en bewaart het gekozen evenement.

*Alternatives:*

* Evenement niet in lijst:   
  De actor geeft een nieuwe evenementnaam in.

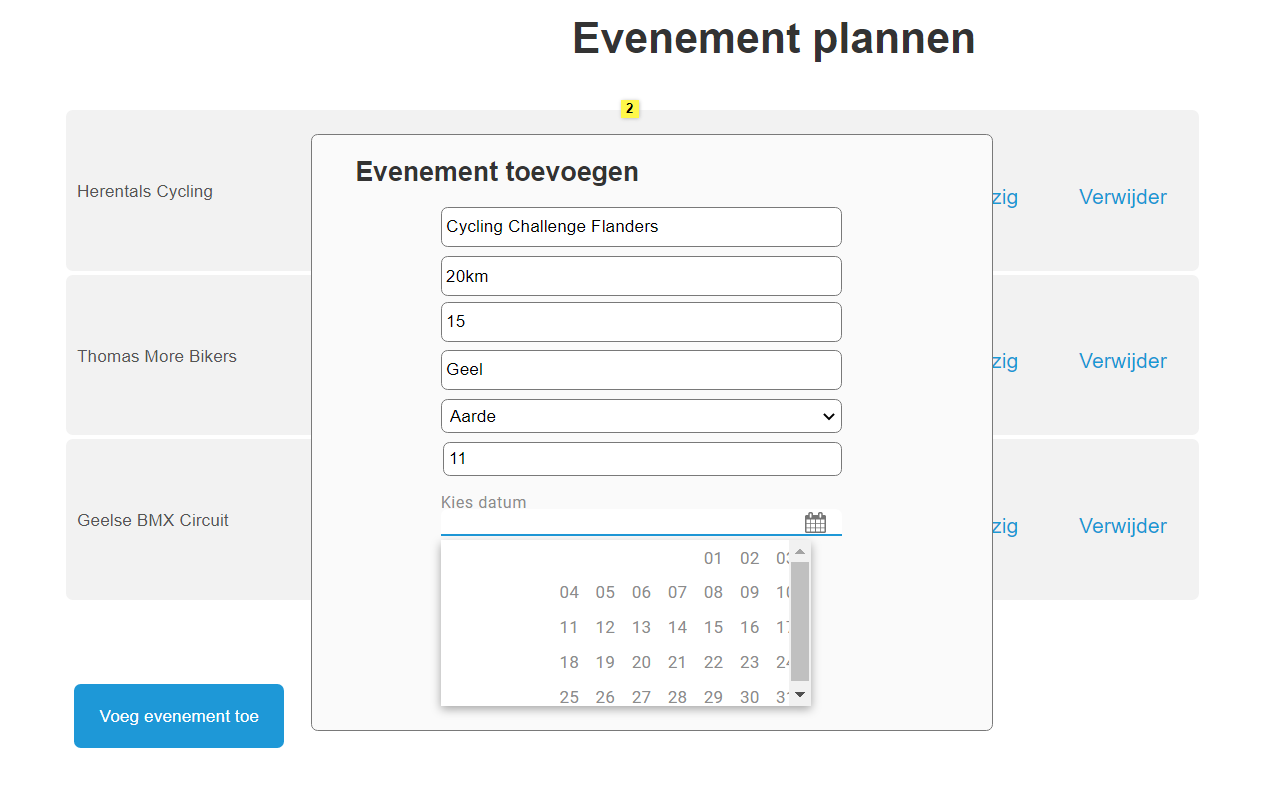
*Screens*:

*11.1 Evenement plannen - Lijst*



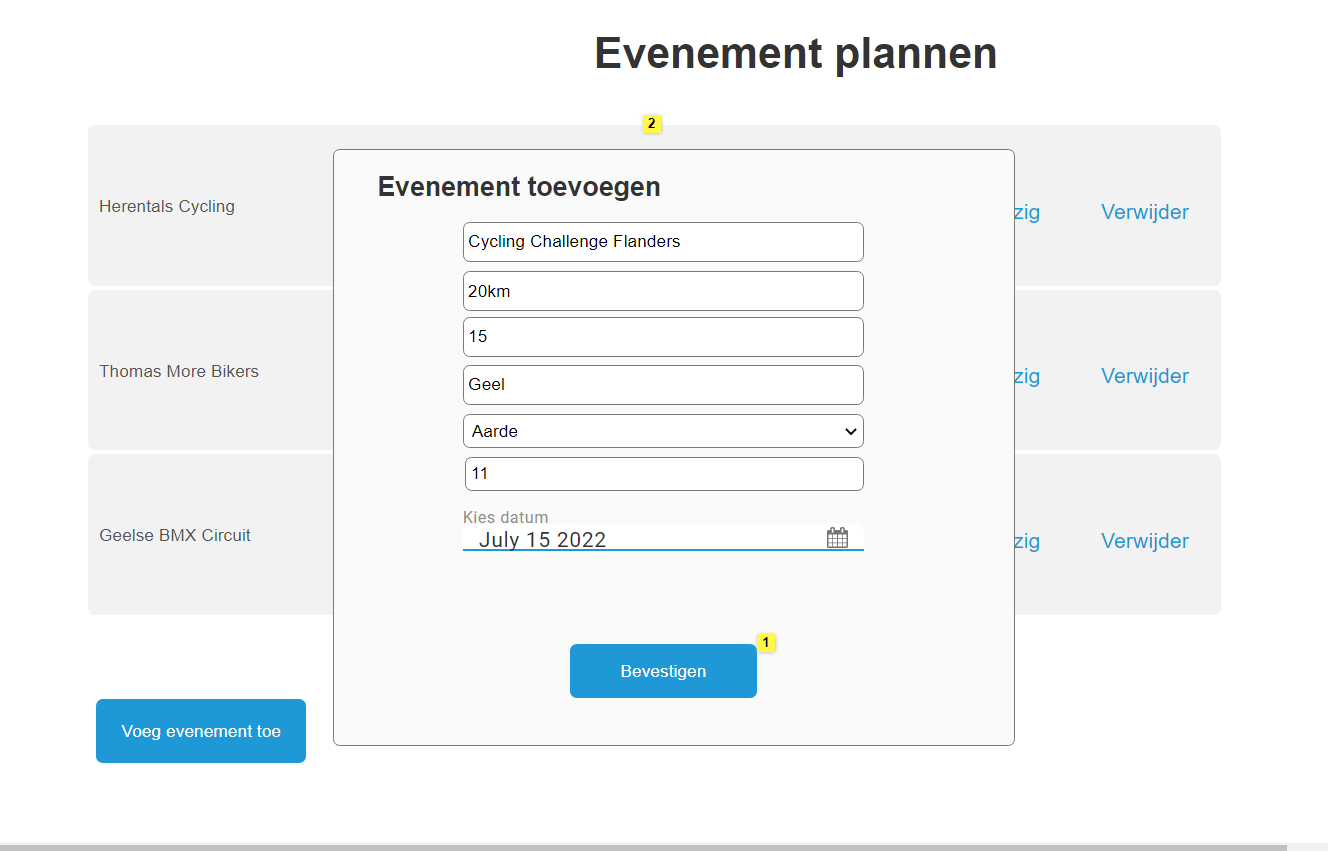
|  |  |
| --- | --- |
| Action | Explanation |
| Voeg evenement toe | Naar pagina “11.2 Evenement toevoegen” als pop-up. |

*11.2 Evenement toevoegen*



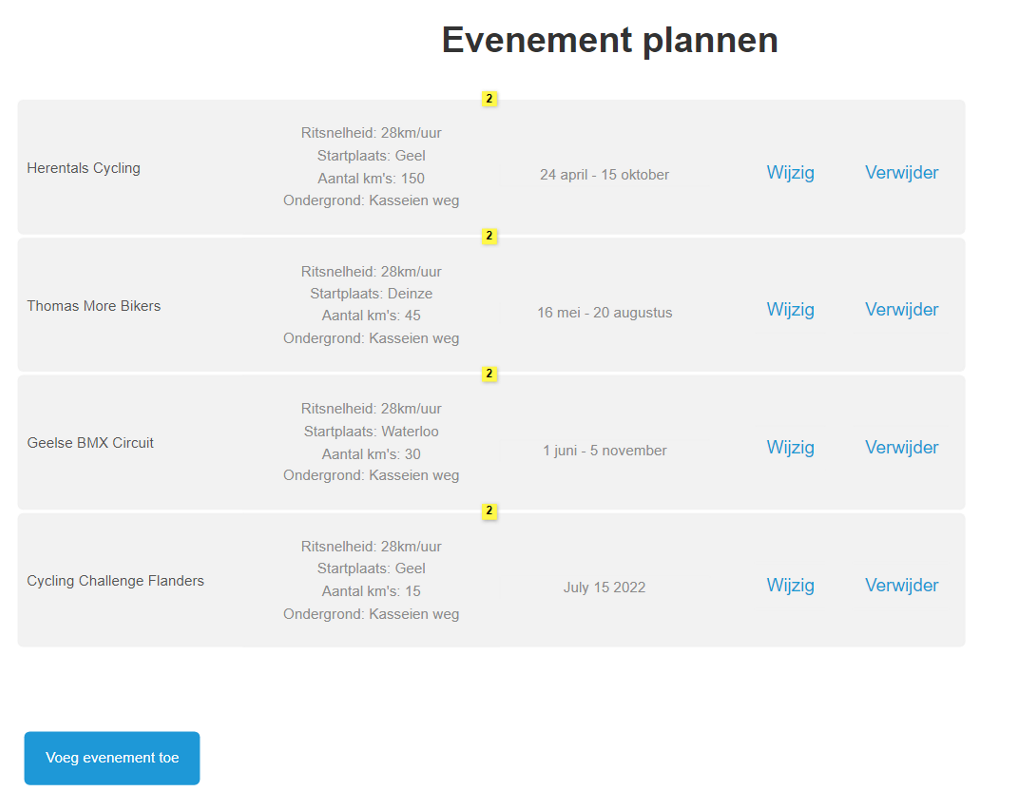
|  |  |
| --- | --- |
| Action | Explanation |
| Kies datum | Dynamische drop down om een datum te selecteren wordt weergegeven. |

*11.3 Evenement bevestigen*



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Nieuw evenement wordt toegevoegd. |
| Bevestigen | Naar pagina “11.1 Evenement plannen - Lijst” |

*11.4 Evenement overzicht*



|  |  |
| --- | --- |
| Action | Explanation |
| 2 | Na het toevoegen van het evenement wordt dit en de bijbehorende informatie weergegeven in het vak. Hiervoor een nieuwe rij dynamisch aangemaakt |
| Wijzig | Naar pagina “11. Wijzig evenement” pop-up. |
| Verwijder | Naar pagina “11. Verwijder evenement” pop-up. Na bevestiging wordt een geüpdatete versie van de pagina weergegeven. |

#### **Wekelijkse planning doorsturen (Seppe)**

*Functionality:*

Als tijd, kan ik wekelijkse planning doorsturen.

*Normal flow:*   
Het systeem stuurt wekelijks een mail van de planning van de ritten, evenementen en ritten waarvoor nog een ritverkenner gezocht wordt naar de mailbox van de leden.

### Secundaire Use Case Beschrijvingen met prototypes

#### **Inloggen**

*Functionality:* Als lid en bestuur kan ik inloggen.

*Normal flow:* Het systeem geeft login velden weer, gebruikersnaam en wachtwoord. De actor geeft de gegevens in en accepteert. Het systeem geeft de homepagina weer voor de juiste actor.

*Alternatives:*

* Gebruikersnaam of/en wachtwoord fout:  
  Het systeem geeft een error weer dat de inloggegevens niet correct zijn.

*Screens:*

*13. Inloggen*



|  |  |
| --- | --- |
| Action | Explanation |
| Login | Naar homepagina voor de juiste actor |

#### **Ploegen beheren**

*Fuctionality*: Als bestuur kan ik de ploegen beheren

*Preconditions*: Bestuur is ingelogd

*Normal flow:* Het systeem geeft de momentele ploegen weer. De actor kiest een ploeg toevoegen. Het systeem vraagt om de naam van de ploeg in te geven. De actor vult de naam in en bevestigt. Het systeem geeft de ploegen weer met de nieuwe aanpassingen.

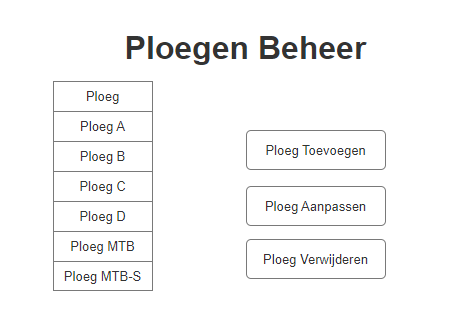
*Alternatives*:

* Aanpassingen: De actor kiest om een ploeg aan te passen. Het systeem geeft deze ploeg weer. De actor past de naam van de ploeg aan.
* Verwijdering: De actor kiest om een ploeg te verwijderen. Het systeem vraagt voor bevestiging. Als de actor bevestigt wordt de ploeg verwijderd.

Let op: als er al connectie zijn voor deze ploeg in het systeem wordt de ploeg niet verwijderd, maar zijn status wel op "inactief" worden gezet.

*Screens*:

*14. Ploegen beheer*



|  |  |
| --- | --- |
| Action | Explanation |
| Ploeg Toevoegen | Je geeft de nieuwe ploeg in |
| Ploeg Aanpassen | Je past een ploeg aan |
| Ploeg Verwijderen | Je verwijdert een ploeg |

#### **Gebruikers beheren**

*Functionality:* Als bestuur kan ik de gebruikers accounts beheren.

*Precondition:* Actor is ingelogd.

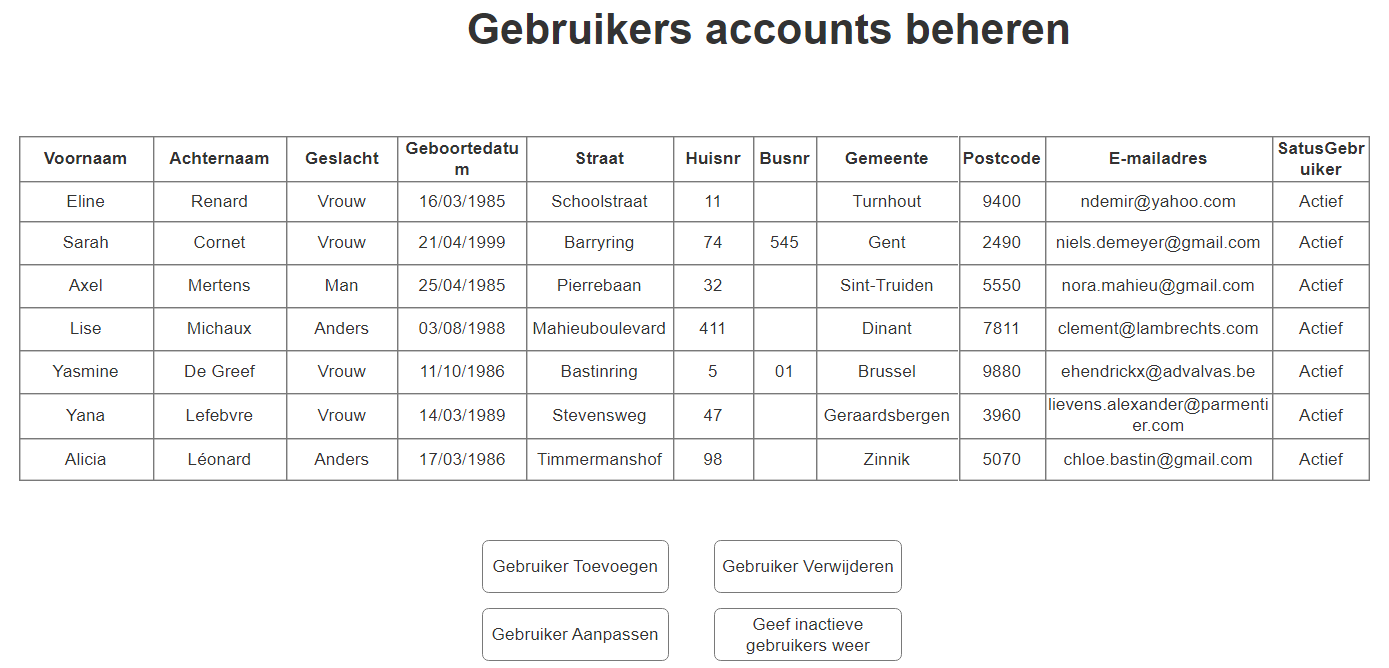
*Normal flow:* Het systeem geeft weer de actieve gebruikers. De actor kiest een gebruiker toevoegen. Het systeem vraagt om de gegevens van de gebruiker in te geven. De actor vult de gegevens in en bevestigt. Het systeem geeft de gebruikers weer met de nieuwe toegevoegde gebruiker.

*Alternatives:*

* *Aanpassingen:* De actor kiest om een gebruiker aan te passen.
* *Verwijderen:* De actor kiest om een gebruikersaccount te verwijderen. Actor bevestigd, het systeem toont het lijst opnieuw weer met het aangepaste lijst weer.
* *Inactieve gebruikers*: De actor kiest “Geef inactieve gebruikers weer”. Het systeem toont alle inactieve gebruikerslijst onder de actieve gebruikers lijst.
* *Opmerking*: de gebruikersaccount is niet effectief verwijderd maar wordt gedeactiveerd.

*Screens:*

*15. Gebruikers accounts beheren:*



|  |  |
| --- | --- |
| Action | Explanation |
| Gebruiker toevoegen | Actor kan een gebruiker toevoegen. |
| Gebruiker aanpassen | Actor kan een bestaand gebruikersaccount aanpassen. |
| Gebruiker verwijderen | Actor kan een bestaand gebruikersaccount verwijderen. |
| Geef inactieve gebruikers weer | Systeem geeft lijst van inactieve gebruikers in nieuwe tabel. |

#### **Geslachtstypes beheren**

*Functionality:* Als bestuur kan ik de geslacht opties aanpassen

*Precondition:* Actor is ingelogd

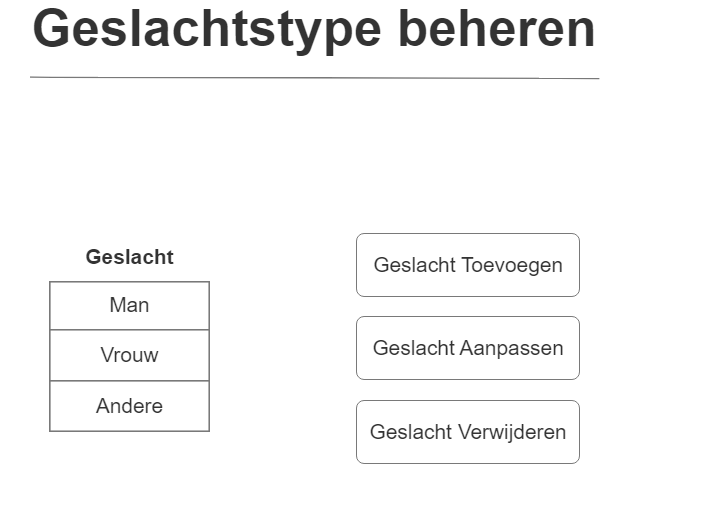
*Normal flow:* Het system geeft een veld weer om een geslacht aan te maken. De actor vult het velt in en accepteert.

*Alternatives:*

* Het geslacht bestaat al:  
  Er wordt een error weergegeven dat het geslacht bestaat

*Screens*:

*16. Geslachtstypes beheren*



|  |  |
| --- | --- |
| Action | Explanation |
| Geslacht toevoegen | Actor kan een geslacht toevoegen. |
| Geslacht aanpassen | Actor kan een bestaand geslacht aanpassen. |
| Geslacht verwijderen | Actor kan een bestaand geslacht verwijderen. |

#### **Kledij beheren**

*Functionality:* Als bestuur kan ik de kledingstukken beheren

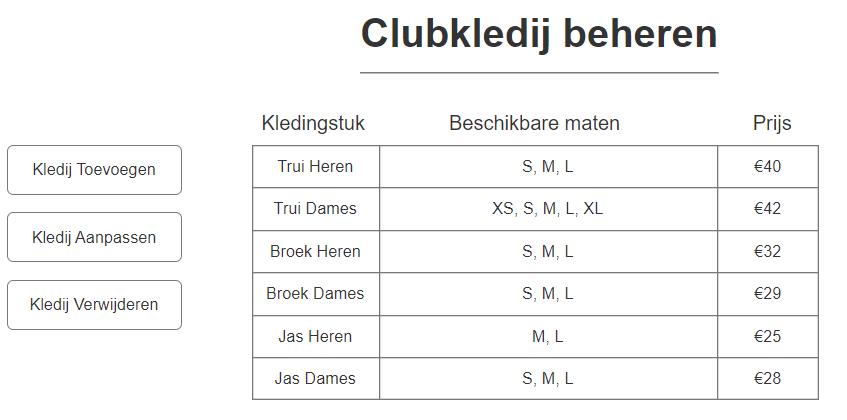
*Precondition:* Actor is ingelogd.

*Normal flow:*   
Het systeem geeft een overzicht weer van de momentele kledingstukken.  
De actor kiest de optie om een kledingstuk toe te voegen en vult het type, de beschikbare maten en de prijs van het kledingstuk in en bevestigt.  
Het systeem bevestigt en toont opnieuw het overzicht met momentele kledingstukken.

*Alternatives:*

* Aanpassen: De actor kiest ervoor om de details van een bestaand kledingstuk aan te passen.
* Verwijderen: De actor kiest ervoor om een kledingstuk te verwijderen. Na bevestiging wordt het kledingstuk verwijderd. Als het kledingstuk gekoppeld is aan bestaande bestellingen wordt het kledingstuk op inactief gezet.

*Screens:*

*17. Kledij beheren*  


|  |  |
| --- | --- |
| Action | Explanation |
| Kledij toevoegen | Actor kan kledingstuk aanmaken |
| Kledij aanpassen | Actor kan een bestaand kledingstuk aanpassen |
| Kledij verwijderen | Actor kan een bestaand kledingstuk verwijderen |

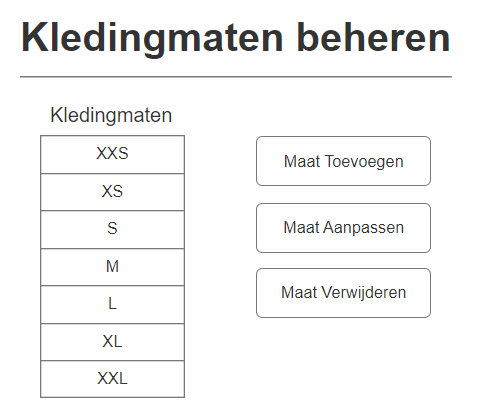
#### **Kledingmaten beheren**

*Functionality:* Als bestuur kan ik de kledingmaten beheren

*Precondition:* Actor is ingelogd.

*Normal flow:* Deze UC is analoog met UC “Ploegen beheren”.

*Screens:*

*18. Kledingmaten beheren*  


|  |  |
| --- | --- |
| Action | Explanation |
| Maat toevoegen | Actor kan een maat toevoegen |
| Maat aanpassen | Actor kan een bestaande maat aanpassen |
| Maat verwijderen | Actor kan een bestaande maat verwijderen |

#### **Ondergrond beheren**

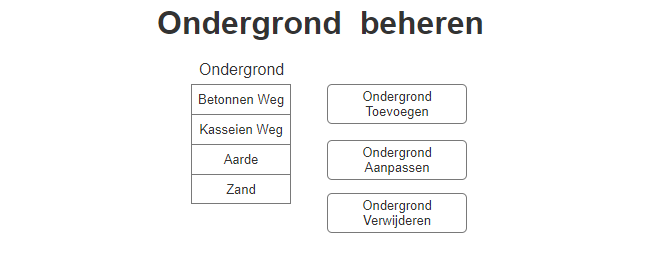
*Functionality:* Als ritverkenner of bestuur kan ik de ondergrond aanpassen.

*Precondition:* Actor is ingelogd.

*Normal flow:* Deze UC is analoog met UC “Ploegen beheren”.

*Screens:*

*19. Ondergrond beheren*



|  |  |
| --- | --- |
| Action | Explanation |
| Ondergrond toevoegen | Actor kan een ondergrond toevoegen |
| Ondergrond aanpassen | Actor kan een bestaande ondergrond aanpassen |
| Ondergrond verwijderen | Actor kan een bestaande ondergrond verwijderen |

#### **Prijs lidmaatschap beheren**

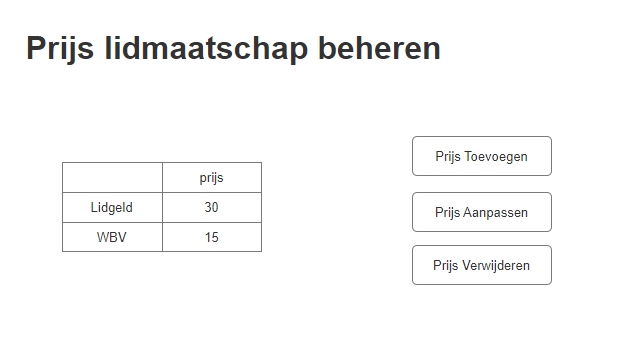
*Fuctionality*: Als het bestuur kan ik de prijzen van de lidmaatschappen beheren

*Preconditions*: Bestuur is ingelogd

*Normal flow:* Deze UC is analoog met UC “Ploegen beheren”.

*Screens*:

*20. Prijs lidmaatschap beheren*



|  |  |
| --- | --- |
| Action | Explanation |
| Prijs Toevoegen | Je geeft de nieuwe prijs in en voor wat de prijs is |
| Prijst Aanpassen | Je past een prijs aan en voor wat de prijs is |
| Prijs Verwijderen | Je verwijdert een prijs en voor wat de prijs is |

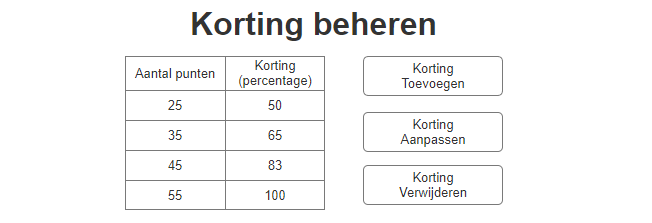
#### **Kortingen beheren**

*Functionality:* Als bestuur kan ik kortingen beheren

*Precondition:* Actor is ingelogd

*Normal flow:* Deze UC is analoog met UC “Ploegen beheren”.

*Screens:*  
*21. Kortingen beheren*



|  |  |
| --- | --- |
| Action | Explanation |
| Korting Toevoegen | Je geeft de nieuwe korting in |
| Korting Aanpassen | Je past een korting aan |
| Korting Verwijderen | Je verwijdert een korting |

## Niet-functionele eisen

* Standaardhoofdingen van de niet-functionele eisen voorzien die van belang zijn voor dit project. Niet-functionele eisen die niet belangrijk zijn voor het project, niet vermelden.
* Doorlopende tekst (volzinnen) gebruiken
* Implementatie zeker beschrijven

+/- 1/2de pagina

Voor de webapplicatie dient als programmeertaal PHP gebruikt te worden, samen met het framework Laravel. Het opslaan van data zal gebeuren in een relationele MySQL database.

Het enige extern systeem dat verbinding maakt met de webapplicatie is de bankingwebsite. De betalingsinfo moet doorgegeven worden aan de hand van de OmniPay API.

Elke pagina moet natuurlijk liefst zo snel mogelijk inladen, wij stellen een maximum laadtijd op van 2 seconden. Het maximumaantal gebruikers die gelijktijdig een functie van de webapplicatie zouden kunnen gebruiken, wordt geschat op 200. Denk hiervoor bijvoorbeeld aan de planning of het gelijktijdig bestellen van een nieuw beschikbaar kledingstuk.

1. Implementatie
2. Hardware / system software
   1. Programming language (PHP)
3. Framework (Laravel)
4. Database
   1. Type (relationeel)
   2. Specific DBMS (MySql)
5. External Interface
   1. Input hardware -> Card Reader -> belfius
   2. Betalingsinfo
   3. Input output format -> OmniPay API
   4. Medium -> internet
6. Performance
   1. Response times:
      1. Planning -> meerdere requests tegelijkertijd.
   2. Scalability
      1. 200 gebruikers -> tegelijkertijd op app
      2. Vertical scaling -> Kubernetes. RAM, CPU
      3. Kubernetes-> creates containers automatically and uses replicas -> which guarantees uptime and speed.
7. Quality Requirements
   1. Software
      1. Security: securing transfer of user credentails. Using a an encryption system. No SSL encryption for payment session, this process is external.
      2. Backup of data needed for data integrity.
   2. Hardware
      1. Access to a smartphone, tablets or a computer.

# Prioriteit per functionaliteit

In dit onderdeel wordt er een MoSCoW prioriteit toegekend aan elke use case.

|  |  |  |  |
| --- | --- | --- | --- |
| Must have | Should have | Could have | Won’t have |
| Lidmaatschap aanmaken | Lidmaatschap vernieuwen | Eigen gegevens raadplegen |  |
| Ritplanning maken | Clubkledij aanvragen | Puntenpolitiek beheren |  |
| Ritdetails ingeven | Aanmelden als ritverkenner | Wekelijkse planning doorsturen |  |
| Aanwezigheden bijhouden | Planning bekijken |  |  |
|  | Evenementen plannen |  |  |