Samen eten app

Systeemdocumentatie

**

Team sites van Google met alle informatie over het team en proces:  
<https://sites.google.com/view/blog-team21/homepage?authuser=0>

Project Agile Development 2019, 1e jaar BIM  
Team: IB105, Team 21  
Versie: 1  
Datum: 26.02.2020

Studenten: 500717442, Robert Kieft   
 500823372, Ike Iyere 500760605, Animesh Ghisai500762427, Jamie Kneebone

**inhoudsopgave**

[**Individuele Verantwoording.**](#_1fob9te) **6**

[**Inleiding**](#_3znysh7) **7**

[**Product Visie en Idee.**](#_2et92p0) **7**

[**Use Case Model**](#_tyjcwt) **8**

[**Actoren**](#_3dy6vkm) **8**

[**Doelen van de Actoren**](#_1t3h5sf) **8**

[**Use Case Diagram**](#_4d34og8) **9**

[**Use Cases**](#_2s8eyo1) **12**

[**XX-UC-001 “Beheert profielen & registratie”**](#_17dp8vu) **13**

[**2.4.2 XX-UC-002 “Matchmaking”**](#_m0gautchvli9) **14**

[**2.4.3 XX-UC-003 “Match naar Diner”**](#_qr5ts0v7wu8c) **15**

[**Domain model.**](#_26in1rg) **17**

[**(Conceptueel) UML Domein Model**](#_lnxbz9) **17**

[**Entiteiten:**](#_35nkun2) **17**

[**Relaties:**](#_1ksv4uv) **17**

[**Modules**](#_44sinio) **18**

[**Mendix Domein Model module\_naam1.**](#_2jxsxqh) **18**

[**Mendix Domein Model module\_naam2.**](#_z337ya) **18**

[**Pages.**](#_3j2qqm3) **20**

[**PG-naam1 Homepage.**](#_1y810tw) **20**

[**Gebruikersinteractie.**](#_4i7ojhp) **21**

[**Vormgeving en styling**](#_2xcytpi) **21**

[**Security en Authorisatie**](#_1ci93xb) **21**

[**PG-naam2: Lijst gebruikers.**](#_3whwml4) **22**

[**Vormgeving en styling**](#_mqo3uc986czt) **24**

[**Security en Authorisatie**](#_9jij0l9p7qov) **24**

[**PG3 - Advertenties kokende studenten**](#_3e1r98fx1g8) **24**

[**Gebruikersinteractie.**](#_8xqz3v4h61jc) **25**

[**Vormgeving en styling**](#_ur90wtaf3h5f) **26**

[**Security en Authorisatie**](#_d0gsyzpf03) **26**

[**PG4: Toevoegen van advertentie**](#_qnyi32mjkf9g) **27**

[**Gebruikersinteractie.**](#_7z29tnl50pnt) **28**

[**Vormgeving en styling**](#_wr7v3h6nbmn) **28**

[**Security en Authorisatie**](#_5d7q97ebszvs) **28**

[**PG-naam5: ProfielGebruikerUitbreiden**](#_l0oysqh3oj4e) **29**

[**Vormgeving en styling**](#_o6uoc3wfrrnk) **31**

[**Security en Authorisatie**](#_9e3y3og4qrtm) **31**

[**PG-naam5: Matchgeschiedenis**](#_usmop7mcd23g) **33**

[**Gebruikersinteractie.**](#_m08xaaul39tm) **34**

[**Vormgeving en styling**](#_2dt5wzpgrvtt) **34**

[**Security en Authorisatie**](#_yo5lbztj5sld) **34**

[**PG-naam7: Chatscherm**](#_byoc9j1nzjdt) **35**

[**Gebruikersinteractie.**](#_7sxi4iv405up) **35**

[**Vormgeving en styling**](#_um7exwz1ow51) **35**

[**Security en Authorisatie**](#_p7o35h3kztph) **35**

[**Microflows**](#_2bn6wsx) **36**

[**MF-naam1.**](#_qsh70q) **36**

[**Invoer en uitvoer**](#_trsi8t4mw02o) **36**

[**Implementatie toelichting.**](#_3as4poj) **36**

[**Foutsituaties en exceptions.**](#_1pxezwc) **36**

[**Security en Autorisatie**](#_49x2ik5) **36**

[**MF-naam2.**](#_2p2csry) **37**

[**Speciale programma code en modules.**](#_147n2zr) **38**

[**Module xxx.**](#_3o7alnk) **38**

[**Interfaces met externe systemen.**](#_23ckvvd) **39**

[**Interface xxx.**](#_ihv636) **39**

[**Test resultaten**](#_32hioqz) **40**

[**Resultaten regressietest d.d. xx-xxx-2018.**](#_1hmsyys) **40**

[**Resultaten regressietest d.d. xx-xxx-2018.**](#_41mghml) **40**

[**Referenties**](#_2grqrue) **42**

# Individuele Verantwoording.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Use Case | Mendix PageDesign | Mendix MicroFlow |
| Student  Ike Iyere | *UC2\_MatchMaking* | *Page\_Afspraakgeschiedenis +*  *Page\_HomePagina* | *ACT\_SaveAdvertentie* |
| Student  Robert Kieft | *UC\_1\_BeheertProfielen&Registratie UC\_2\_Matchmaking* | *AdvertentiesKokendeStudenten + Afspraak bevestigen +*  *LijstVanAlleGebruikers +*  *ProfielGebruikerWijzigen + ToevoegenVanEenAdvertentie* | *ACT\_CreateAfspraak* |
| Student  Jamie Kneebone | *UC1\_BeheertProfielen&Registratie* | *Account\_Overview*  *Account\_New*  *Account\_Edit* | *ACT\_CreateAfspraak* |
| Student Animesh Ghisai | *UC3\_MatchNaarDiner* | *Page\_ChatPagina* | *ACT\_CreateBericht* |

Team sites van Google met alle informatie over het team en proces:  
<https://sites.google.com/view/blog-team21/homepage?authuser=0>

# Inleiding

Dit is de systeemdocumentatie van de samen eten app.

De systeemdocumentatie is gemaakt n.a.v. het PAD project.

## Product Visie en Idee.

Het product is de samen eten app, deze app moet studenten stimuleren om samen te eten met andere studenten. De samen eten app zal een aantal doelstellingen realiseren.

De doelstellingen van de app zijn het volgende:

1. Sociale communicatie tussen (internationale) studenten verbeteren.
2. Studenten stimuleren om samen te eten met anderen.
3. Voedselverspilling in studentenhuizen tegengaan.
4. De app moet een oplossing bieden zodra iemand niet weet wat hij/zij gaat eten.

Om deze app te realiseren hebben wij een plan van aanpak voor het ontwerp, bouw & realisatie van de app. Wij werken in een team van 4 mensen en werken aan de hand van de scrum methode.

Het team verdeelt de taken en werkt in sprints. Om een verschil te maken en de doelstellingen van de product owner te realiseren zal het team incrementeel de taken toevoegen en uitvoeren in de app.Aan het eind van elke Sprint is er een audit waarin het team feedback krijgt op de kwaliteit van het werk tot dat moment

Het team communiceert vaak de met de opdrachtgever zodat het team onduidelijkheden gelijk kan oplossen of vermijden. Op deze manier kunnen de doelstellingen van de product owner gerealiseerd worden.Aan het eind van elke Sprint is er een audit waarin het team feedback krijgt op de kwaliteit van het werk tot dat moment

# Use Case Model

## Actoren

Hieronder volgt een overzicht van de actoren in de applicatie. Actoren zijn gebruikersprofielen, -rollen of externe systemen die een interactie kunnen hebben met de applicatie. Van elke actor wordt een omschrijving gegeven die de rol en belangen van de actor binnen de organisatie typeert.

|  |  |
| --- | --- |
| Actor | Omschrijving, typering |
| Internationaal student | Een internationaal student aan een universiteit in een grote stad (Amsterdam), uitwonend in studentenhuis |
| NL student uitwonend | Nederlands student, uitwonend in (studenten)huis |
| NL student thuiswonend | Nederlands student, kunnen studenten- leven in studentenstad ontdekken, thuiswonend |
| Admin | Beheert de data en de app |

## Doelen van de Actoren

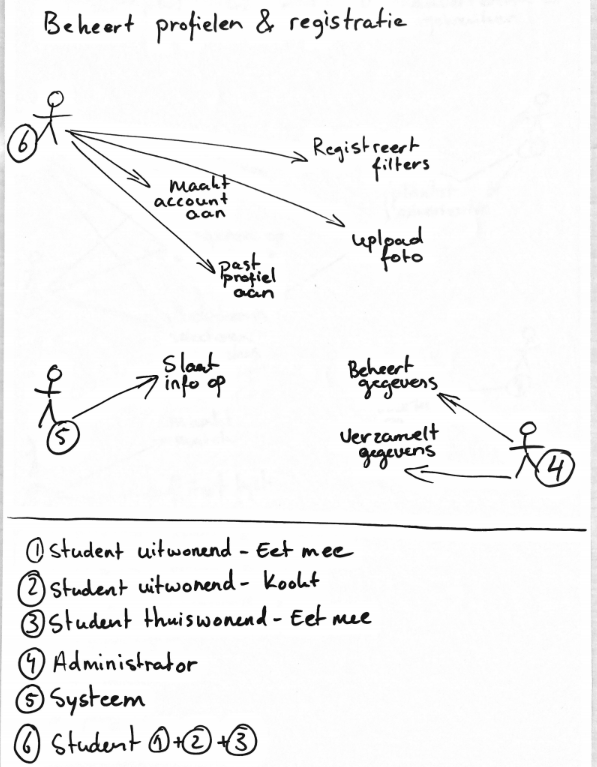
De functionele doelen van de actoren worden hier samengevat en toegelicht.

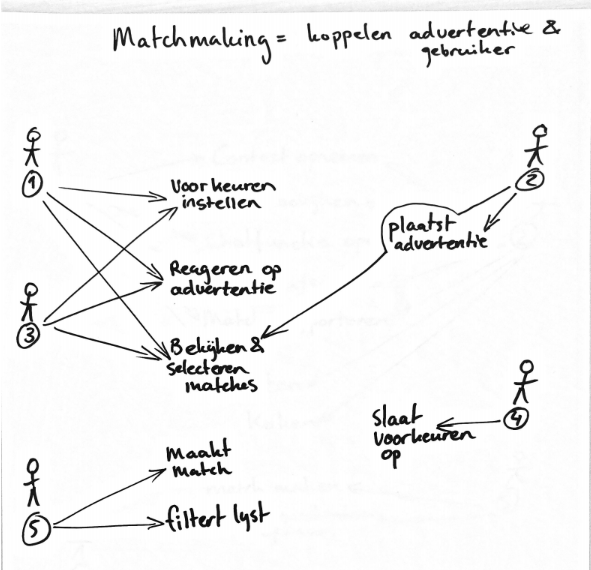
|  |  |
| --- | --- |
| Actor | Functioneel doelen |
| Internationaal student | -Nieuwe mensen leren kennen van eigen of andere culturen, door middel van het organiseren of aanschuiven bij via de app georganiseerde etentjes  -Geld besparen door eet-kosten te verdelen  -Nederlandse normen en waarden leren |
| NL student uitwonend | -Geld besparen door eet-kosten te verdelen  -Gebruikt de app om nieuwe mensen te ontmoeten  -Gebruikt de app om niet alleen te hoeven eten |
| NL student thuiswonend | -Gebruikt de app om af en toe van thuissituatie te kunnen ontsnappen  -Gebruikt de app om nieuwe mensen te ontmoeten  -Gebruikt de app om internationale studenten kennis te laten maken met Nederlandse cultuur |
| Admin | -Gebruikt de app voor gebruikersbeheer  -Gebruikt de app om te testen |

## 

## Use Case Diagram

Het Use Case Diagram geeft een overzicht van alle functionele doelen van de actoren in een plaatje. Het toont ook afhankelijkheden tussen Use Cases en betrokkenheid van secundaire actoren.







## Use Cases

Hier volgen de tekstuele beschrijvingen van de Use Cases van de applicatie, volgens het ‘partly dressed template’. Elke Use Case wordt voorafgegaan door de (epic) User Story die het functionele requirement van de Actor specifieker formuleert. Daarna volgen een of meerdere (afgesplitste/afgeleide) User Stories die de fasering van de implementatie toelichten.

### XX-UC-001 “Beheert profielen & registratie”

|  |  |
| --- | --- |
| Use Case ID: | XX-UC-001 |
| Use Case Name: | *Beheer profiel & registratie.* |
| Primary Actor: | *De uitwonende en thuiswonende student. Actor 6.* |
| Scope: | *De studenten registreren d.m.v. een account aan te maken en beheren het profiel d.m.v. een foto toe te voegen.* |
| Stakeholders and Interest | *De administrator: het verzamelen van accountgegevens.*  *Het systeem: Opslaan van informatie.* |
| Pre-conditions: | De user is geïnteresseerd in de app en maakt een account aan.  De functie om je profiel aan te passen werken naar behoren. |
| Success Guarantee | *De student heeft een account aangemaakt. Vervolgens het profiel aangepast en een foto toegevoegd. Het systeem heeft de informatie opgeslagen en de administrator kan de gegevens beheren.* |
| Main Success Scenario | 1. Student maakt een account aan. 2. Systeem registreert gegevens. 3. Student past profiel informatie aan. 4. Student upload een foto. 5. Systeem slaat informatie op. |
| Extensions | *1a: Student verlaat account registratie. Flow wordt niet doorgezet.*  *2a, 5a: Systeem of pagina reageert niet.* 3a: Systeem voert wijzigingen niet door.  4a: 1. Afbeelding wordt niet herkend.  2. Afbeelding formaat is te groot/klein. |
| Special Requirements | De niet-functionele eisen:  Security:   1. Verzamelde data is alleen toegankelijk voor de administrator. 2. Handelingen kunnen worden herleid naar de administrator. 3. Identiteit wordt vastgelegd als er iets binnen het systeem wordt veranderd.   Maintainability:  Aanpassingen kunnen makkelijk gemaakt worden zonder dat andere componenten van het systeem beschadigd raken.  Reliability:   1. Het systeem werkt naar behoren.   Usability:   1. Wijzigingen moeten makkelijk worden toegepast. 2. Interface is duidelijk voor de gebruiker. |
| Open Issues | n.v.t |

### *2.4.2 XX-UC-002 “Matchmaking”*

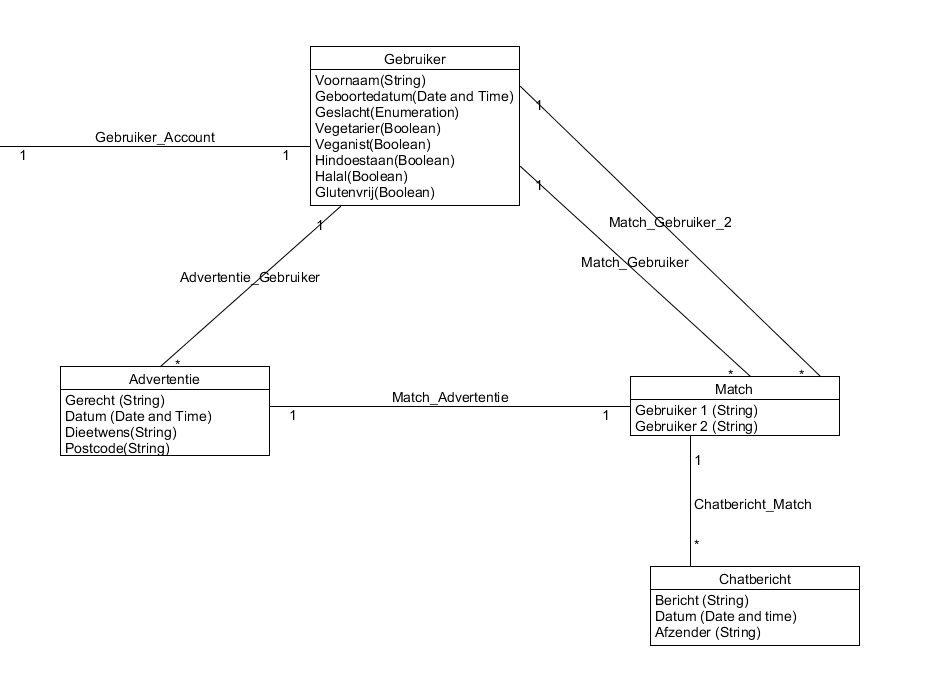
|  |  |
| --- | --- |
| Use Case ID: | XX-UC-002 |
| Use casename: | *Beheer profiel & registratie.* |
| Primary Actor: | *De uitwonende en thuiswonende student die gaan mee eten. Actor 1 en 3.* |
| Scope: | *De studenten die gaan mee eten kunnen op advertenties reageren, gebaseerd op filters, die geplaatst worden door de student die kookt(actor 2).* |
| Stakeholders and Interest | *De uitwonende student die kookt: een advertentie plaatsen waarop gebruikers kunnen reageren.*  *Het systeem: advertenties filteren en match maken.*  *De administrator: opslaan van voorkeuren.* |
| Pre-conditions: | Er moet een bestaande advertentie zijn waaruit actoren 1 en 3 kunnen kiezen.  Het systeem moet de match maken tussen de student die mee eet en de student die kookt. |
| Success Guarantee | *De kokende student plaatst een advertentie. De mee etende studenten filteren de lijst op basis van hun voorkeuren. Het systeem maakt de match tussen de kokende en mee etende student.* |
| Main Success Scenario | 1. Actor 2 plaatst een advertentie. 2. Actor 1 en 3 filteren de advertentie lijst. 3. Systeem filtert lijst. 4. Actor 1 en 3 kiezen advertentie geplaatst door actor 2. 5. Systeem maakt match. |
| Extensions | *1a: Advertentie wordt gecancelled.*  *2a: Geen advertenties op basis van filters. Actoren moeten opnieuw zoeken.* 3a: Systeem voert wijzigingen niet door.  4a: 1. Advertentie wordt door een systeemfout niet getoond.  2. Actoren 1 en 3 kiezen een andere advertentie.  5a: Systeem werkt niet naar behoren. |
| Special Requirements | De niet-functionele eisen:  Security:   1. Verzamelde data is alleen toegankelijk voor de administrator. 2. Handelingen kunnen worden herleid naar de administrator. 3. Identiteit wordt vastgelegd als er iets binnen het systeem wordt veranderd.   Maintainability:   1. Aanpassingen kunnen makkelijk gemaakt worden zonder dat andere componenten van het systeem beschadigd raken.   Reliability:   1. Het systeem werkt naar behoren.   Usability:   1. Wijzigingen moeten makkelijk worden toegepast. 2. Interface is duidelijk voor de gebruiker. |
| Open Issues | n.v.t |

### *2.4.3 XX-UC-003 “Match naar Diner”*

|  |  |
| --- | --- |
| Use Case ID: | XX-UC-003 |
| Use Case Name: | Samen eten. |
| Primary Actor: | De student die mee eet(actor 1 en 3). |
| Scope: | De student die gaat mee eten en de student die kookt hebben een match gemaakt. De actoren nemen contact op en spreken af. |
| Stakeholders and Interest | De uitwonende student die kookt: accepteren van match en afspraak maken.  Het systeem: Match maken en match geschiedenis opslaan.  De administrator: opslaan van voorkeuren. Gerapporteerde matches onderzoeken. |
| Pre-conditions: | Er moet een match zijn tussen de student die mee eet en de persoon die kookt.  Beide gebruikers moeten gebruik kunnen maken van de chat functie. |
| Success Guarantee | De kokende student plaatst een advertentie. De mee etende studenten filteren de lijst op basis van hun voorkeuren. Het systeem maakt de match tussen de kokende en mee etende student. |
| Main Success Scenario | 1. Student die mee eet neemt contact op met student die kookt. 2. Beide gebruikers zijn bereid om af te spreken. 3. Er wordt een datum afgesproken. |
| Extensions | 1a: 1. De student neemt geen contact op.  2. De chatfunctie werkt niet.  2a: Een van de gebruikers haakt af op het laatste moment.  3a: Er is geen moment waarop ze beide beschikbaar zijn. |
| Special Requirements | De niet-functionele eisen:  Security:   1. Verzamelde data is alleen toegankelijk voor de administrator. 2. Handelingen kunnen worden herleid naar de administrator. 3. Identiteit wordt vastgelegd als er iets binnen het systeem wordt veranderd.   Maintainability:   1. Aanpassingen kunnen makkelijk gemaakt worden zonder dat andere componenten van het systeem beschadigd raken.   Reliability:   1. Het systeem werkt naar behoren.   Usability:   1. Wijzigingen moeten makkelijk worden toegepast. 2. Interface is duidelijk voor de gebruiker. |
| Open Issues | n.v.t |

# Domain model.

## (Conceptueel) UML Domein Model



### Entiteiten:

**Gebruiker:** De entiteit Gebruiker bevat alle data die op je gebruikersprofiel wordt weergegeven. De attributen bepalen wat er allemaal op je profiel kan komen te staan.

**Advertentie**: De entiteit Advertentie wordt gebruikt om eet-advertenties mee te maken. Wat er op de allemaal komt te staan wordt bepaald door de verschillende attributen.

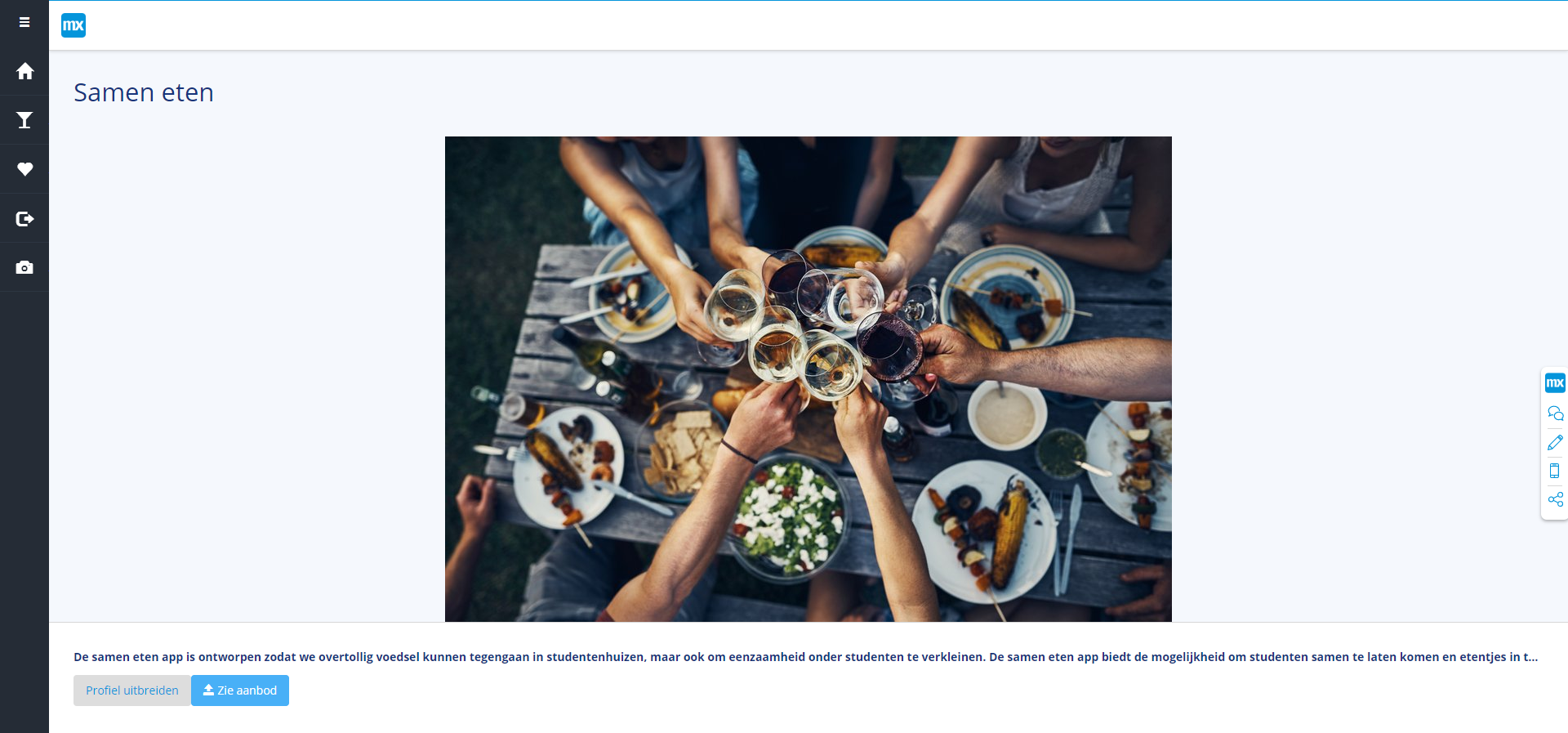
**Match:** De entiteit Match creëert in principe een ruimte waarin data kan worden uitgewisseld tussen twee gebruikers. De twee verschillende soorten attributen van Match bestaan dan ook uit de twee verschillende gebruikers.

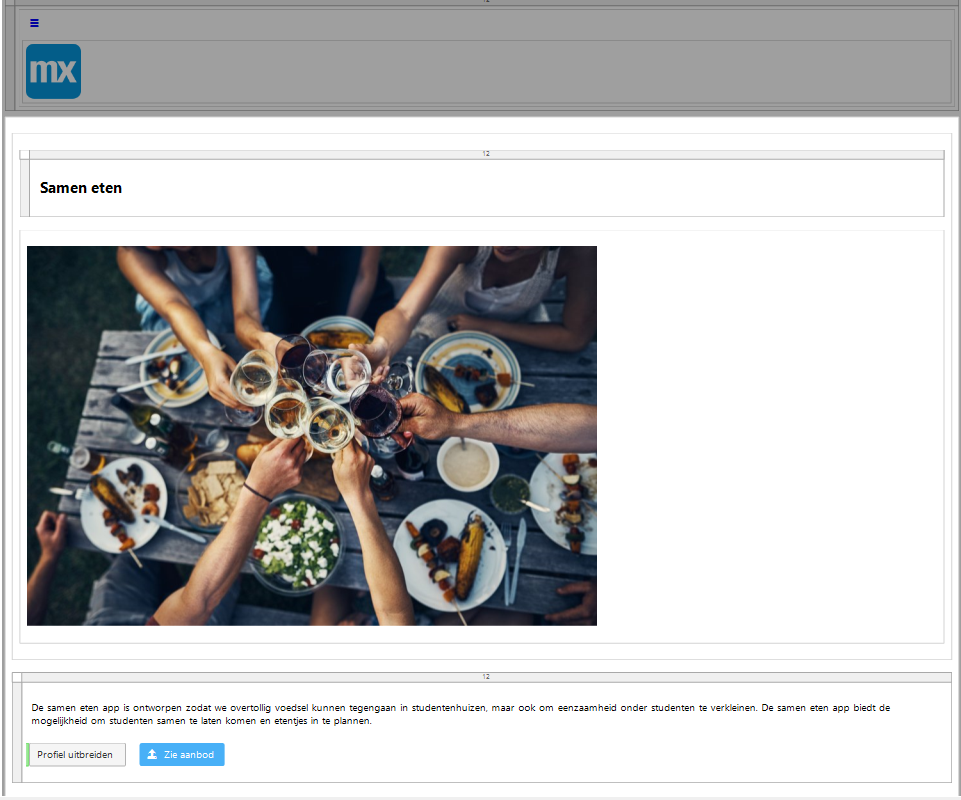
**Chatbericht:** De entiteit Chatbericht is de ruimte waarin deze twee gebruikers data kunnen uitwisselen in de vorm van chatberichten.

# Pages.

Hier vind je een overzicht alle pagina’s en de navigatie daarbinnen.  
Elke pagina is beschreven met de inhoud en interacties.

## *PG-naam1 Homepage*.

**

**

Deze pagina is de Homepagina van de app. Dit is de 1e pagina die de bezoeker ziet. Dit is de pagina waar de app geïntroduceerd wordt en deze pagina verwijst gebruikers verder de app in.

Deze pagina ondersteunt Use Case: als een gebruiker, wil ik op een homepage komen als ik de website bezoek, zodat ik kan zien wat de functionaliteit van de app is.

Je kunt naar deze pagina navigeren met de beschikbare link van de app.

### Gebruikersinteractie.

Op deze pagina zijn de volgende gebruikersinteracties mogelijk:

**Interactie 1**: Er is een profiel uitbreiden knop. Deze knop zorgt ervoor dat de gebruiker doorverwezen wordt naar een andere pagina waar de gebruiker gegevens kan uitbreiden.

**Interactie 2**: Er is een zie aanbod knop. Deze knop zorgt voor de interactie met de advertenties. De gebruiker wordt na het drukken op de knop doorverwezen naar een pagina met een overzicht van de advertenties.

### Vormgeving en styling

De volgende specifieke thema’s en CSS-styling regels zijn gebruikt bij de vormgeving van de pagina:

**Thema/regel 1**: De homepagina heeft een CSS styling voor de afbeelding op de pagina. Deze CSS styling is handmatig toegepast door middel van de margin. Verder is de pagina automatisch aangemaakt met het thema.

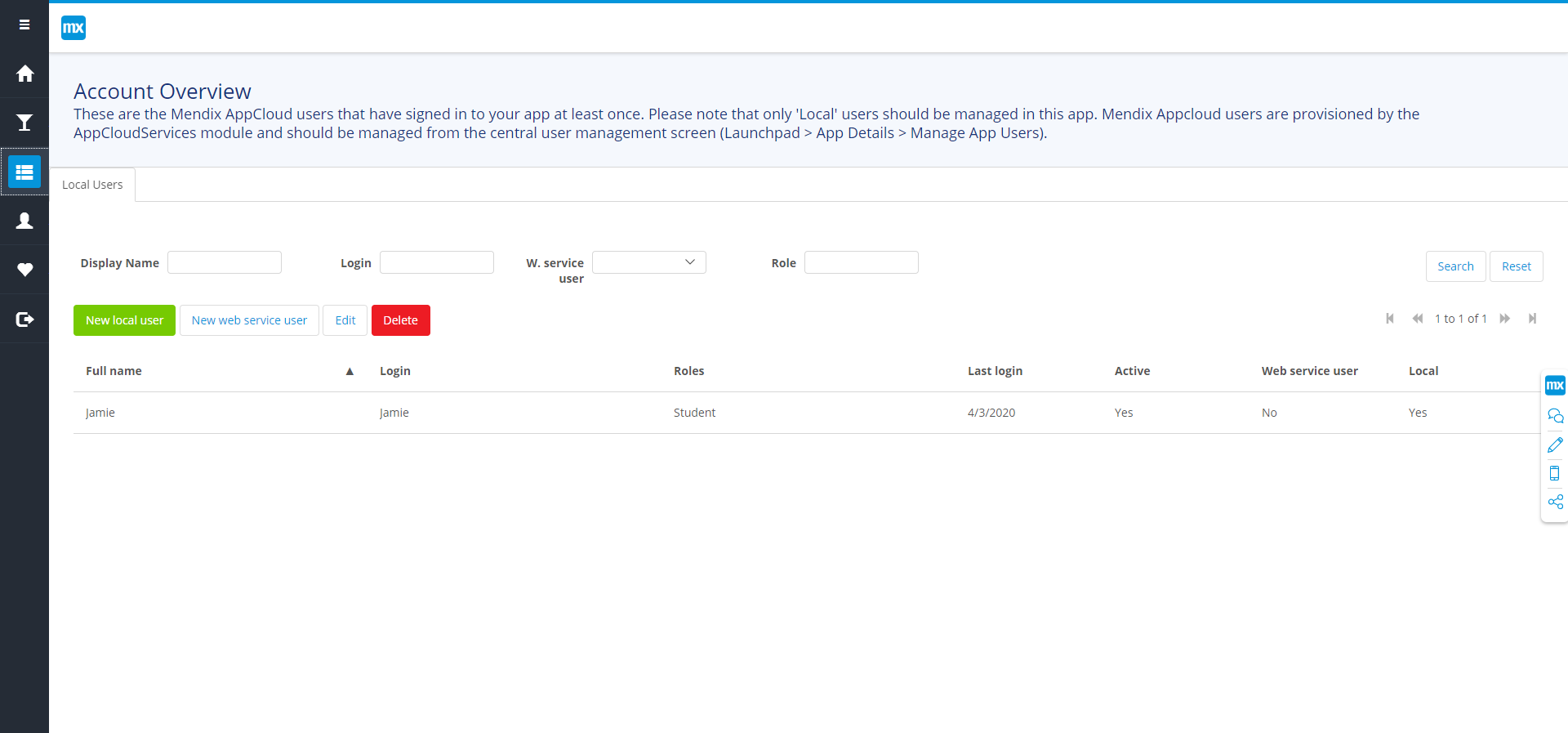
**Thema/regel 2**: De homepagina heeft een CSS styling voor de zie aanbod knop op de pagina. Deze CSS styling is handmatig toegepast door middel van de margin. Verder is de pagina automatisch aangemaakt met het thema.

### Security en Authorisatie

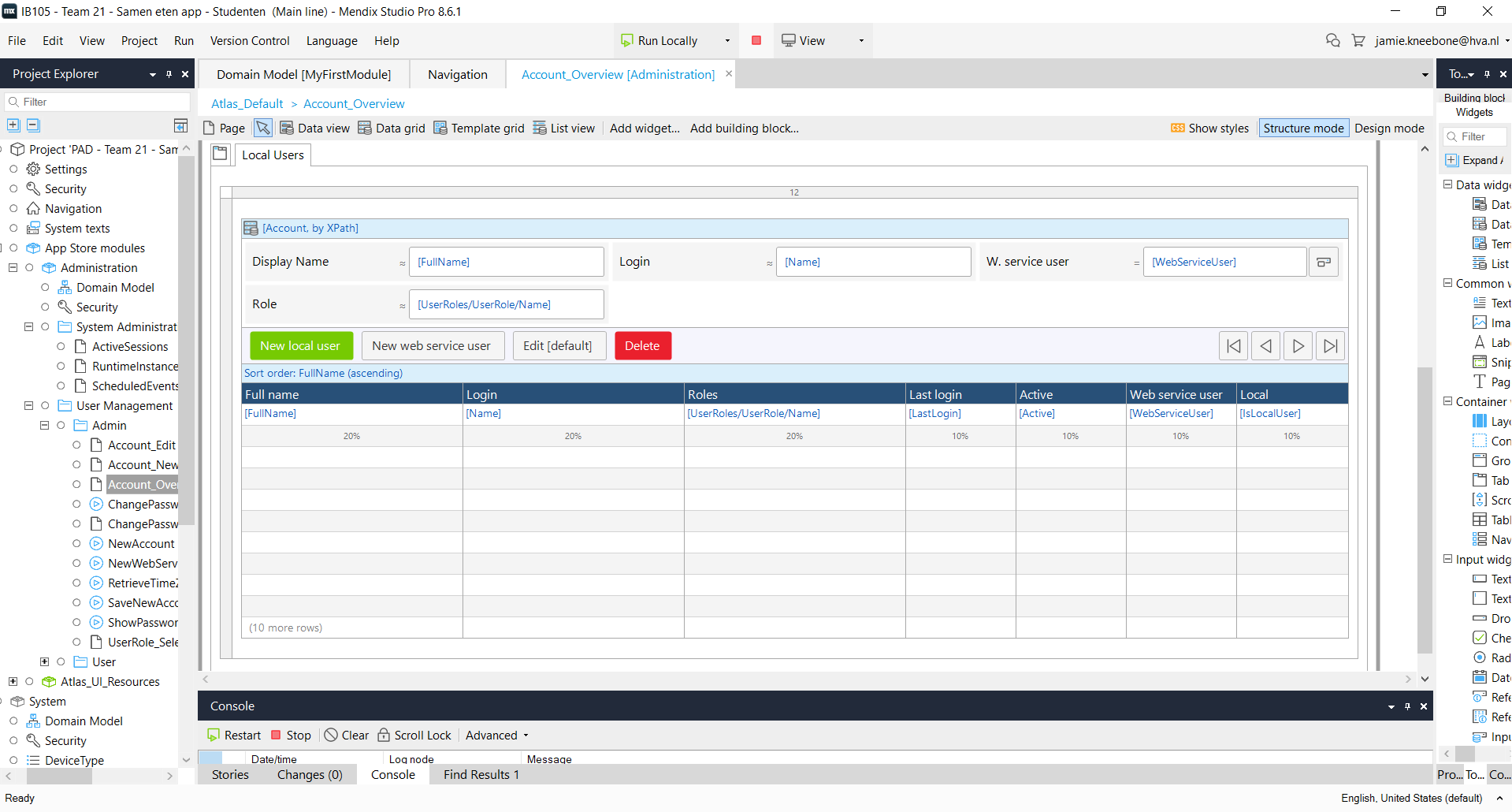
De volgende gebruikersrollen hebben recht van toegang tot de pagina:

**Rol 1**:Alle gebruikers hebben recht van toegang tot de pagina.

## *PG-naam2: Lijst gebruikers*.

**

**

**

**Beschrijving**

Op deze pagina zie je een overzicht van alle gebruikers, hier kan je ook nieuwe gebruikers met verschillende rollen in aanmaken en verwijderen. Op dit scherm zien we ook de data van gebruikers. Deze pagina is dus van groot belang voor de security van de applicatie en de administrator role is de enige die hier bij kan.

**User case**

Deze pagina ondersteunt Use Case:XX-UC-001 “Beheert profielen & registratie”

**Navigatie**Je kunt naar deze pagina navigeren met een druk op de knop: SYS Gebruikersbeheer in het menu links.

**Gebruikersinteractie**

Op deze pagina zijn de volgende gebruikersinteracties mogelijk

**Interactie1**: Druk op de groene knop: New Local User. Je vult hier data in voor het aanmaken van een nieuw systeem account.

**Interactie2**: De lege velden bovenin zijn bedoeld om te filteren op log in naam, rollen etc..

**Interactie3**: Klik op een profiel en daarna op delete, zo verwijder je een account.

**Interactie4**: Klik op een profiel en daarna op de edit knop, zo pas je een account aan.

…

### Vormgeving en styling

De volgende specifieke thema’s en CSS-styling regels zijn gebruikt bij de vormgeving van de pagina:

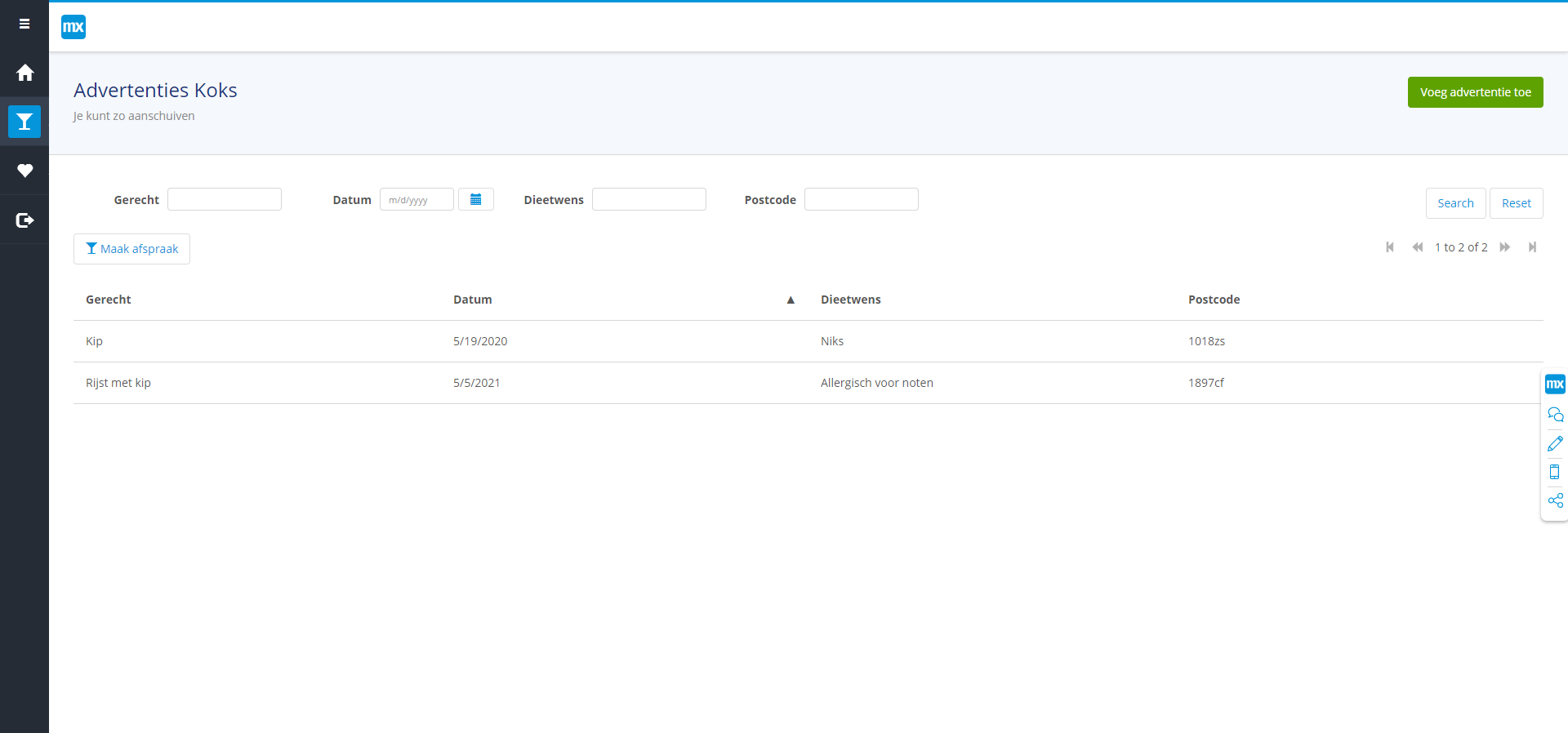
**Thema/regel1**:Deze pagina is automatisch aangemaakt en we hebben nog niet nagedacht over de styling. We weten wel dat we dit met css gaat gebeuren. Omdat de pagina automatisch is aangemaakt en van groot belang is is het ook beter om deze zo te laten. Misschien dat we later de kleuren nog gaan aanpassen.

### Security en Authorisatie

De volgende gebruikersrollen hebben recht van toegang tot de pagina:

**Rol1**: Alleen administrators hebben toegang tot deze pagina zodat normale gebruikers geen andere gebruikers kunnen verwijderen.

## *PG3 - Advertenties kokende studenten*

**

**

De pagina geeft een overzicht van alle advertenties, geplaatst door studenten die een maatje zoeken om mee te eten. De advertentie bevat een beschrijving van het gerecht, een datum, dieetwensen, een locatie en de advertentie wordt gekoppelt aan een gebruiker.

Deze pagina ondersteunt Use Case *XX\_UC\_002 Matchmaking.*Je kunt naar deze pagina navigeren met:

**actie1** knop “zie aanbod” op de homepage,

**actie2** Via het menu op elke pagina wordt genavigeerd naar het overzicht van alle advertenties.

### Gebruikersinteractie.

Op deze pagina zijn de volgende gebruikersinteracties mogelijk:

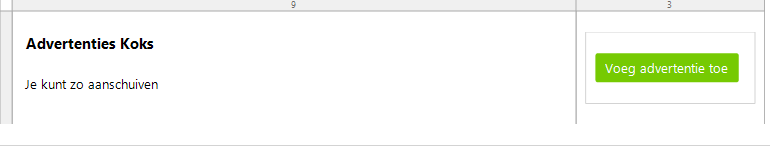
**Interactie1**: Er is een knop om een advertentie toe te voegen. Hier ontstaat een pop up/nieuwe pagina: “Toevoegen van een advertentie”. Zie 4.4

**Interactie2**: Er zijn filters waarmee kan worden gezocht naar specifieke wensen. Er kan worden gezocht op de omschrijving van het gerecht, de datum, de dieetwensen en de locatie (postcode).

**Interactie3:** Er zijn gerechten in een lijst weergegeven. Wanneer de gebruiker één van de advertenties selecteert en de knop “maak match” activeert, wordt de advertentie toegevoegd aan het overzicht van matches/favoriete advertenties.  
De gebruiker kan ook dubbelklikken op een advertentie. Dat geeft hetzelfde resultaat.  
De knop “maak match” en de dubbelklik verwijzen naar de microflow ACT\_CreateMatch.

### Vormgeving en styling

De volgende specifieke thema’s en CSS-styling regels zijn gebruikt bij de vormgeving van de pagina:



***Thema/regel1****: De lay out van de header is opgemaakt met CSS. De header 1 en ondertitel zijn met CSS dynamisch gemaakt en zo goed vorm te geven met elk apparaat. Mobiele weergave is geen probleem.*

***Thema/regel2****: De knop “Voeg advertentie toe” is gecentreerd in het midden van de kolom.*

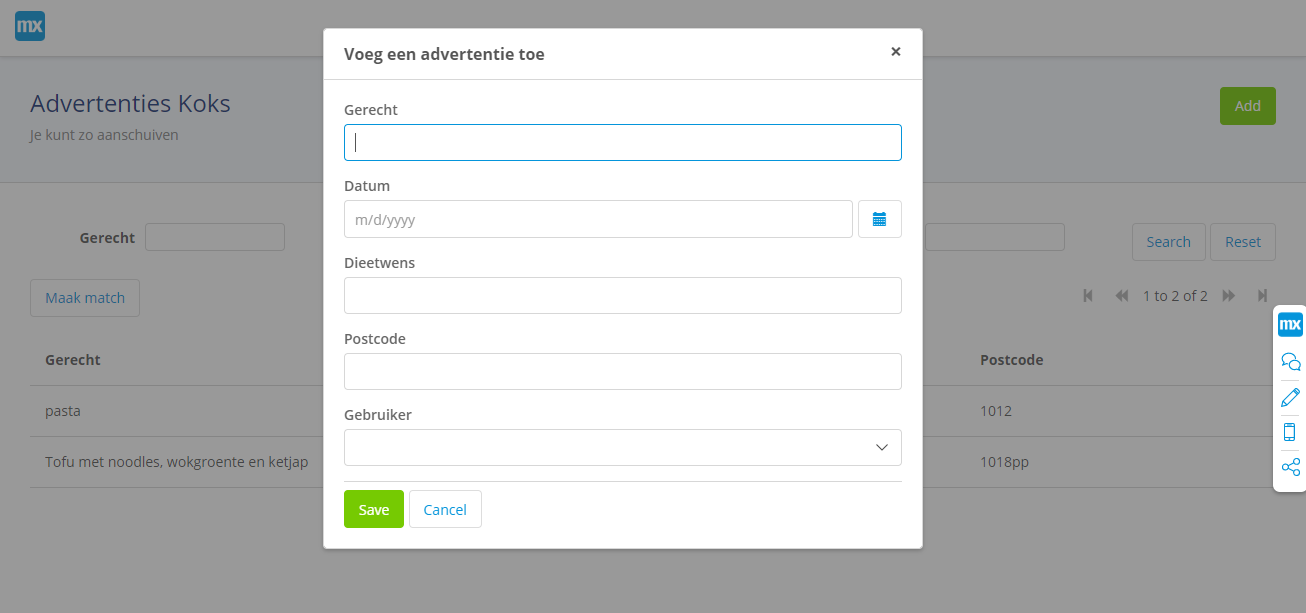
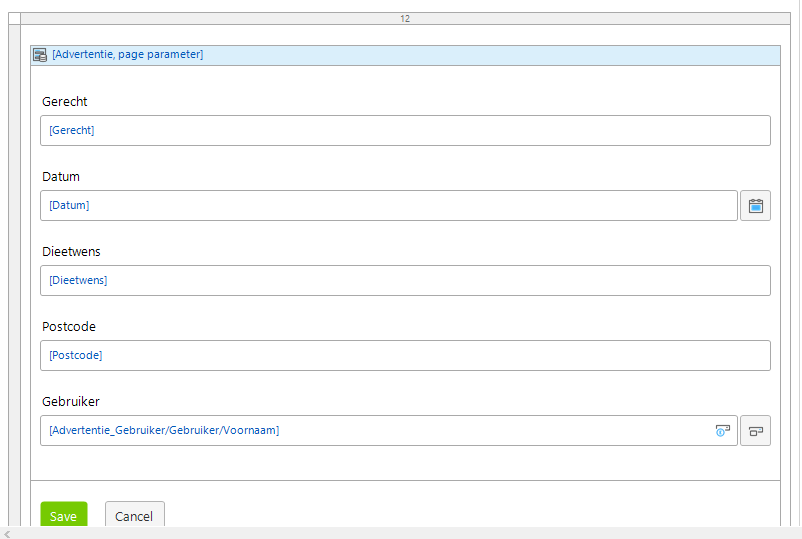
### Security en Authorisatie

De volgende gebruikersrollen hebben recht van toegang tot de pagina:

***Rol1****: Administrator. Om alle advertenties te kunnen inzien.*

***Rol2:*** *Student. Om een advertentie te kunnen plaatsen en te reageren op advertenties van andere studenten.*

## *PG4:* Toevoegen van advertentie

*  
*Deze pagina is een pop-up en komt voort uit het overzicht van advertenties (page3). Het is de pagina om een advertentie toe te voegen. De advertentie wordt opgeslagen in het domeinmodel en automatisch gekoppelt aan een gebruiker.

Deze pagina ondersteunt Use Case *XX\_UC\_002 Matchmaking*Je kunt naar deze pagina navigeren met

actie1 Knop “Add” op page3: AdvertentiesKokendeStudenten

### Gebruikersinteractie.

Op deze pagina zijn de volgende gebruikersinteracties mogelijk:

**Interactie1**: Het invullen van de gegevens omtrent de advertentie.

**Interactie2**: De gegevens opslaan in het domeinmodel en zodoende weergeven op page3: AdvertentiesKokendeStudenten.

**Interactie3**: Het aanmaken van de advertentie annuleren door op de “cancel” knop te drukken.

### Vormgeving en styling

De volgende specifieke thema’s en CSS-styling regels zijn gebruikt bij de vormgeving van de pagina:

Er is geen CSS-styling gebruikt in het maken van deze pagina.

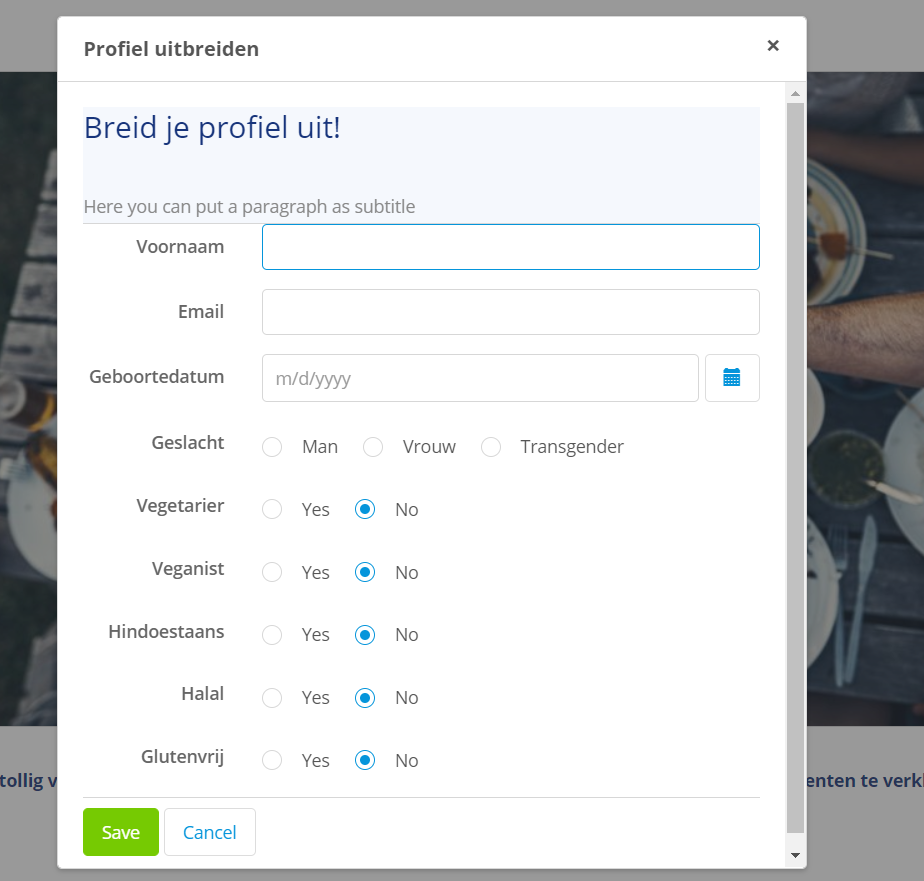
### Security en Authorisatie

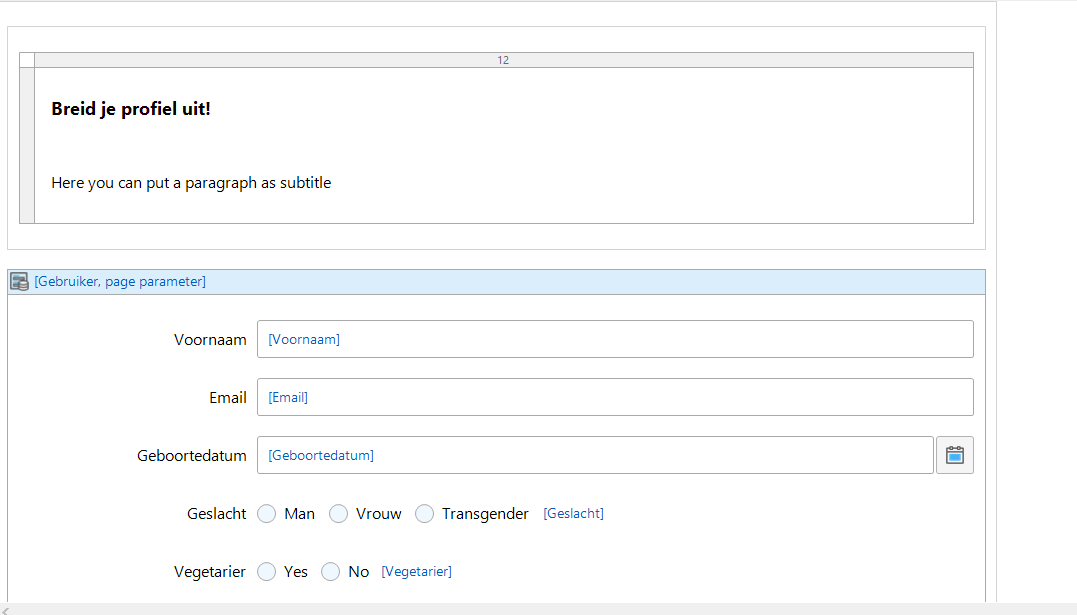
De volgende gebruikersrollen hebben recht van toegang tot de pagina:

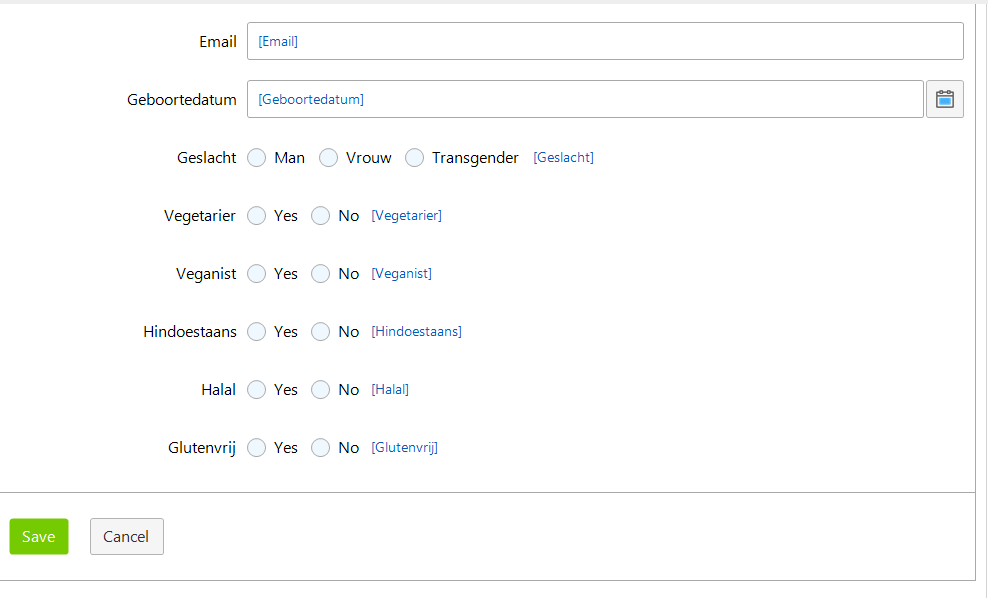
**Rol1**: Student. De student maakt advertenties aan.

De administrator heeft niets te maken met deze pagina.

## *PG-naam5:* ProfielGebruikerWijzigen

**

**

**

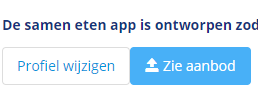
**Beschrijving**

Op deze pagina krijg je de optie om je profiel te personaliseren. Dit is nodig voordat je op andere advertenties kan reageren. Je vult gegevens over je naam, e-mail en geboortedatum in. Het leek ons ook handig om via je profiel mensen te laten weten of je nog speciale dieetwensen hebt, je krijgt daarom de optie om aan te geven of je vegetariër, veganist, hindoestaans, halal of glutenvrij prefereert.

**User case**

Deze pagina ondersteunt Use Case: XX-UC-001 “Beheert profielen & registratie”

**Navigatie**Je kunt naar deze pagina navigeren met een druk op de knop: Profiel Wijzigen, linksonder op de homepagina.



**Gebruikersinteractie**

Op deze pagina zijn de volgende gebruikersinteracties mogelijk

**Interactie1**: Voornaam, e-mail, geboortedatum en dieetwensen invullen door de gebruiker. Deze data is te herleiden uit het domeinmodel, entiteit Gebruiker. Geen microflows gebruikt of andere pop-ups.

**Interactie2:** Op save drukken om je profiel veranderingen op te slaan in het systeem.

**Interactie3:** Druk op cancel om het venster weer te verlaten en je wijzigingen niet door te voeren naar het systeem.

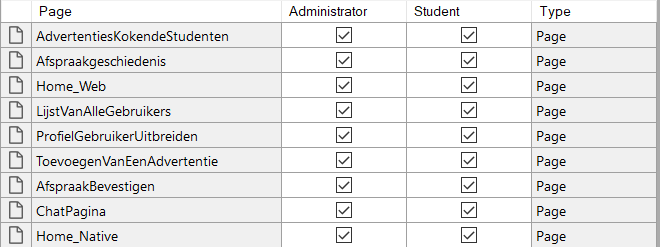
### Vormgeving en styling

De volgende specifieke thema’s en CSS-styling regels zijn gebruikt bij de vormgeving van de pagina:

***Thema/regel1****: Er is geen css gebruikt bij deze pagina*

### Security en Authorisatie

De volgende gebruikersrollen hebben recht van toegang tot de pagina:



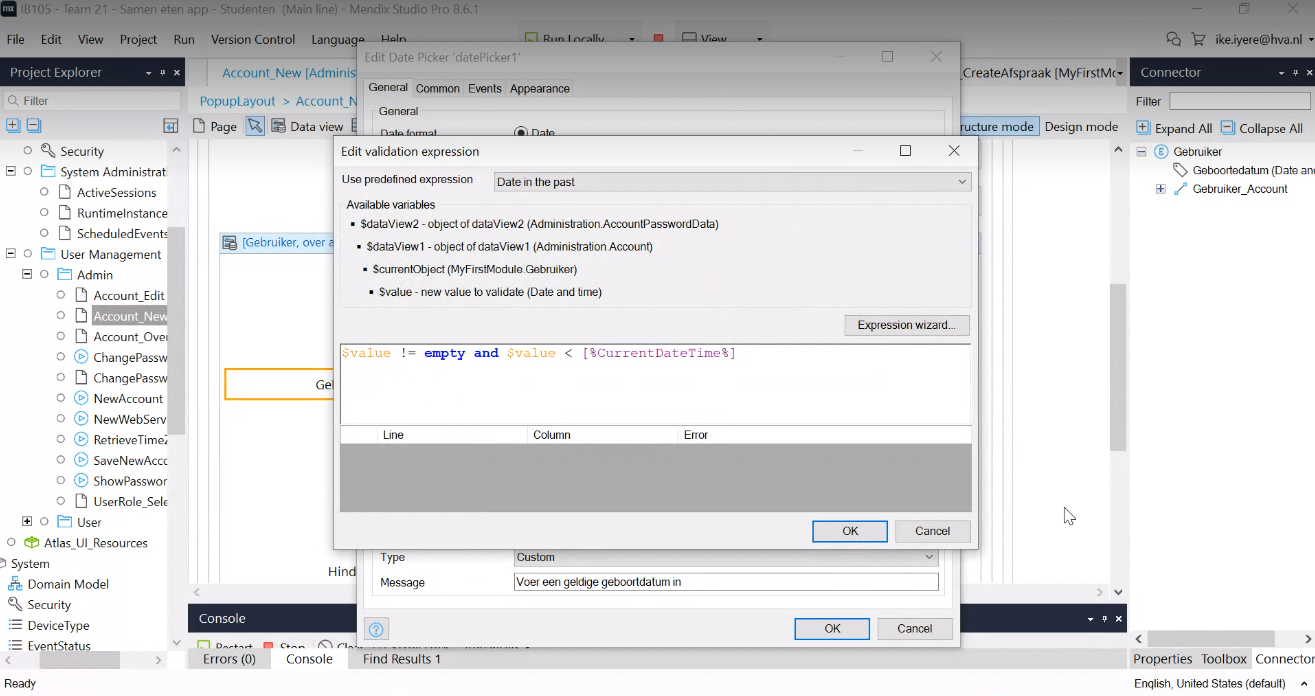
***Rol1 Administrator****:   
De administrator moet bij deze pagina om te kunnen testen.*

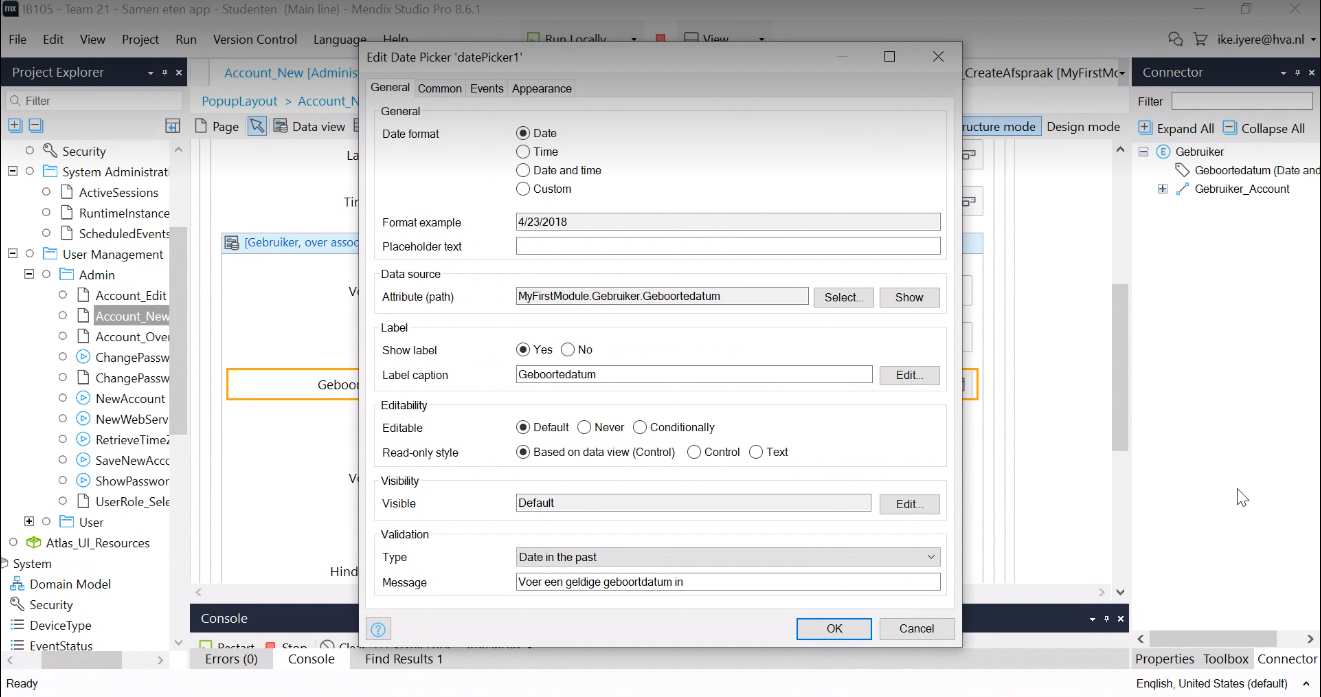
***Rol2 Student:*** *De student moet bij deze pagina kunnen komen om hun profiel aan te passen of uit te breiden.*

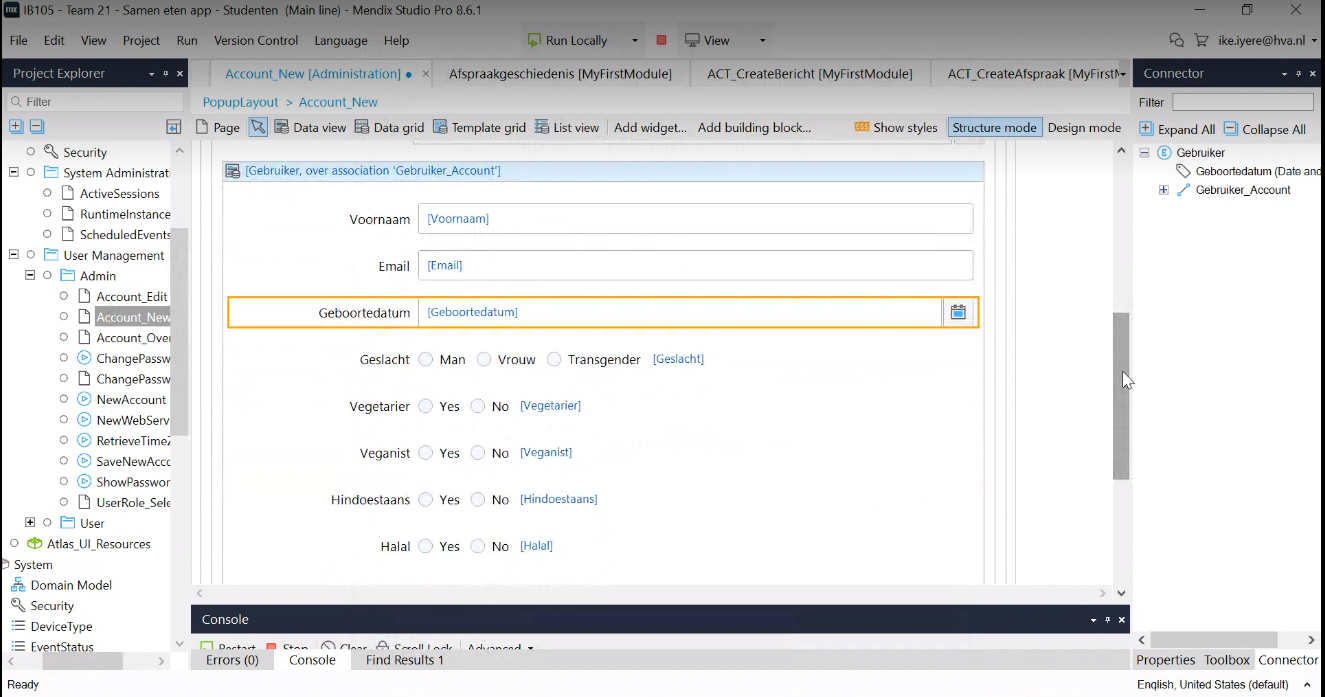
### Validaties:

Voor deze pagina is er gebruik gemaakt van een validatie bij het invoeren van een geboortedatum. Een geboortedatum in de toekomst wordt als foutief gezien. De user krijgt de volgende melding “ Voer een geldige geboortedatum in”.

Zie screenshots:

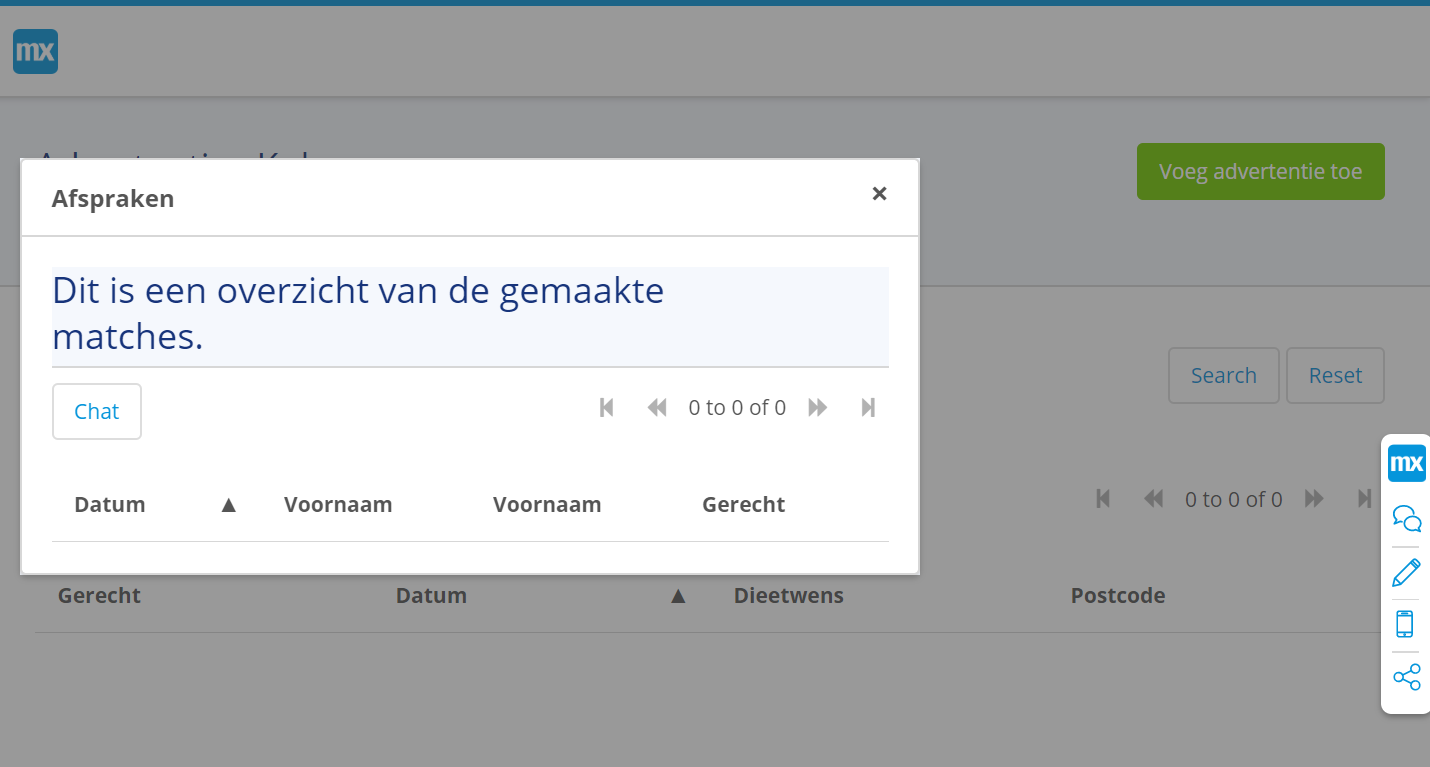


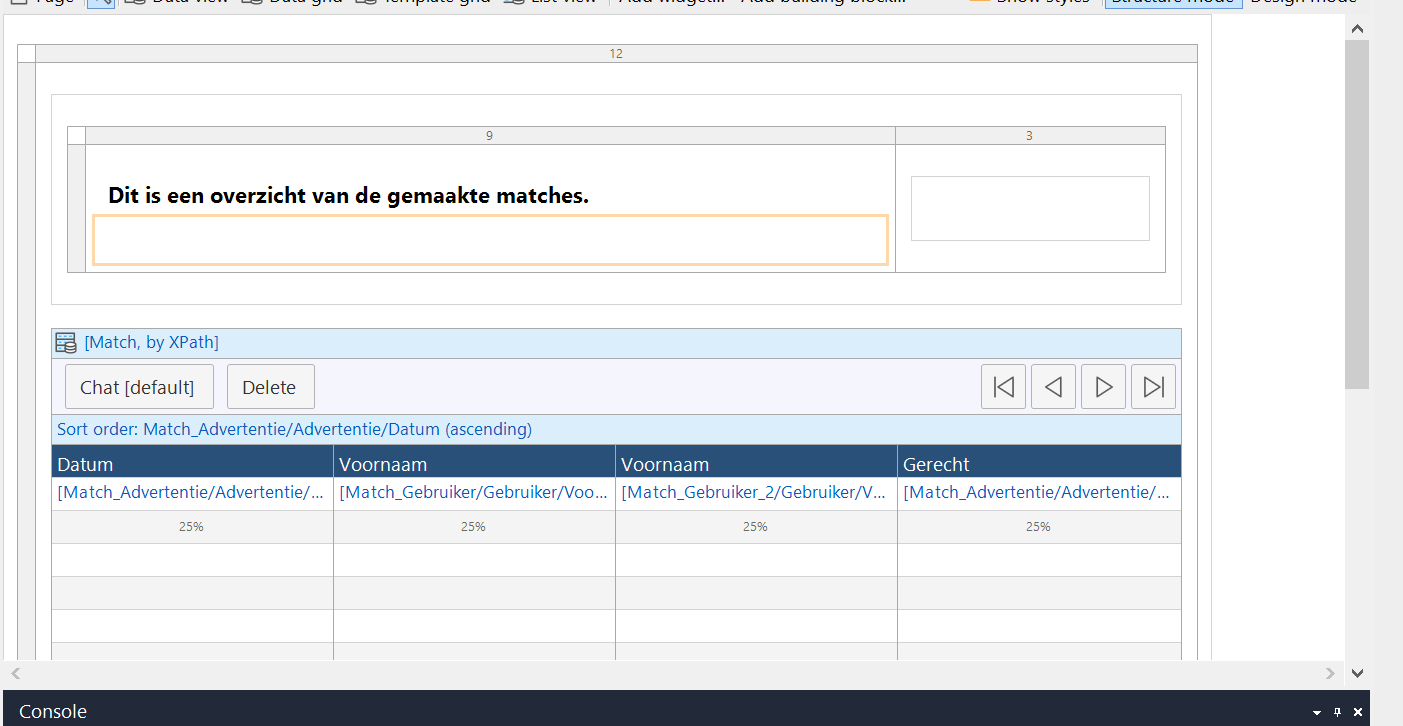




# 

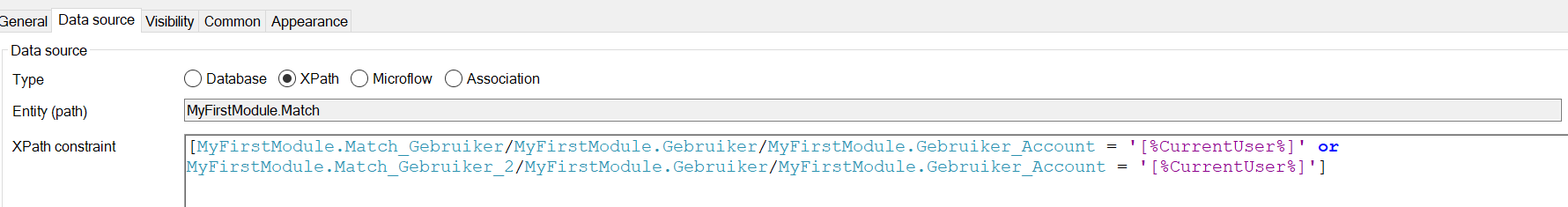
## *PG-naam5* Afspraakgeschiedenis



**

De afspraakgeschiedenis pagina is een pagina die de gemaakte matches laat zien. Deze pagina draagt een functionaliteit van opgeslagen informatie weergeven, zodat de gebruiker hiervandaan kan gaan chatten. De afspraakgeschiedenis pagina laat alleen de gemaakte matches zien van de gebruiker die op dat moment is ingelogd. Dit is door middel van een xpath.

Zie screenshot:



Deze pagina ondersteunt Use Case: Als een student wil ik een match kunnen maken met een student met dezelfde voorkeuren, zodat we samen kunnen gaan etenJe kunt naar deze pagina navigeren door op de homepagina naar het aanbod van advertenties te gaan. Vanuit het aanbod van advertenties een advertentie uitkiezen en met die een match maken.   
Vervolgens kan de gebruiker via dit overzicht op de advertentiepagina de knop ‘maak match’, die activeert microflow ‘ACT\_CreateAfspraak’ en zo navigeert de gebruiker naar de match geschiedenis pagina.

### Gebruikersinteractie.

Op deze pagina zijn de volgende gebruikersinteracties mogelijk:

**Interactie 1**: De gebruiker kan op deze pagina zoeken naar matches met bepaalde voorkeuren die de gebruiker als match heeft.

**Interactie 2**: De gebruiker kan op deze pagina een chat beginnen met 1 van de gemaakt matches. Deze interactie verwijst de gebruiker direct door naar een andere pagina.

### Vormgeving en styling

De volgende specifieke thema’s en CSS-styling regels zijn gebruikt bij de vormgeving van de pagina:

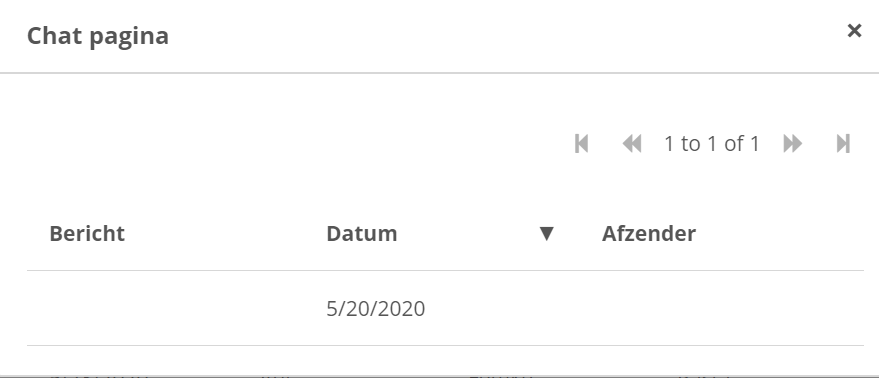
**Thema/regel 1**: De pagina is automatisch aangemaakt. Er wordt CSS gebruikt voor de dynamische ligging van knoppen en afbeeldingen, zodat deze er verzorgd uitzien.

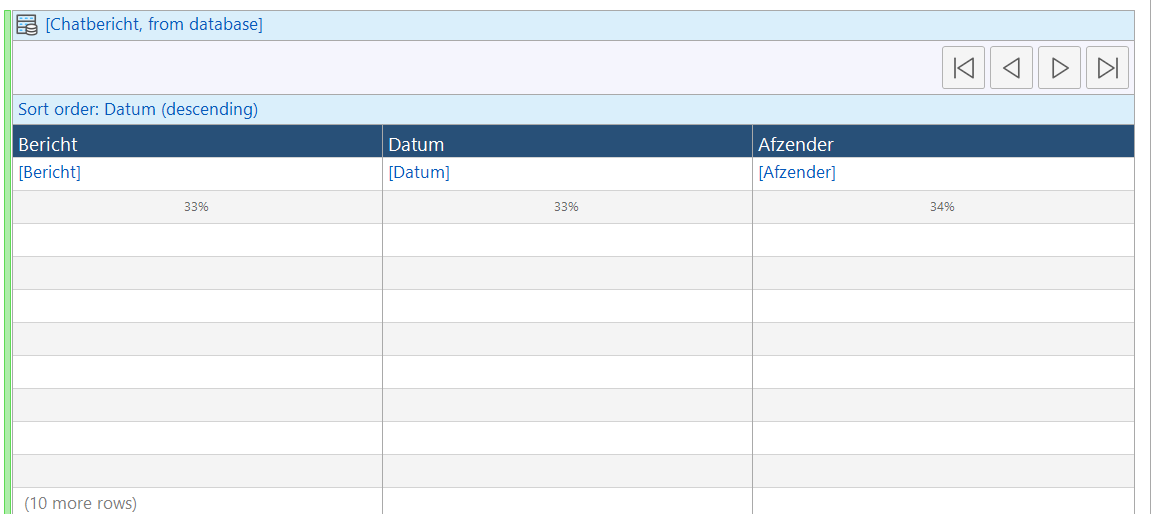
### Security en Authorisatie

De volgende gebruikersrollen hebben recht van toegang tot de pagina:

**Rol1**: De gebruikers hebben recht van toegang tot de pagina.  
**Rol2**: De administrator heeft recht van toegang tot de pagina.

## *PG-naam7:* ChatPagina



**

*Op de pagina is er de functie om chatberichten te sturen naar de persoon met wie je hebt gematched en de gespreksgeschiedenis zien.*

Deze pagina ondersteunt Use Case *XX\_UC\_003 “Match naar diner”.*Je kunt naar deze pagina navigeren door vanuit de match geschiedenis pagina op de knop chat te drukken.

### Gebruikersinteractie.

Op deze pagina zijn de volgende gebruikersinteracties mogelijk:

**Interactie1**: Gebruiker kan in een overzicht(data view) zien welke berichten verstuurd zijn, wanneer de berichten verstuurd zijn. ACT\_CreateBericht zorgt ervoor dat beide gebruikers in de chat zitten en de chatpagina opgehaald wordt.

**Interactie2**: Gebruiker kan berichten aanmaken, ontvangen en versturen.

### Vormgeving en styling

De volgende specifieke thema’s en CSS-styling regels zijn gebruikt bij de vormgeving van de pagina:

***Thema/regel1****:* De pagina is automatisch aangemaakt. Er wordt CSS gebruikt voor de dynamische ligging van knoppen en afbeeldingen, zodat deze er verzorgd uitzien.

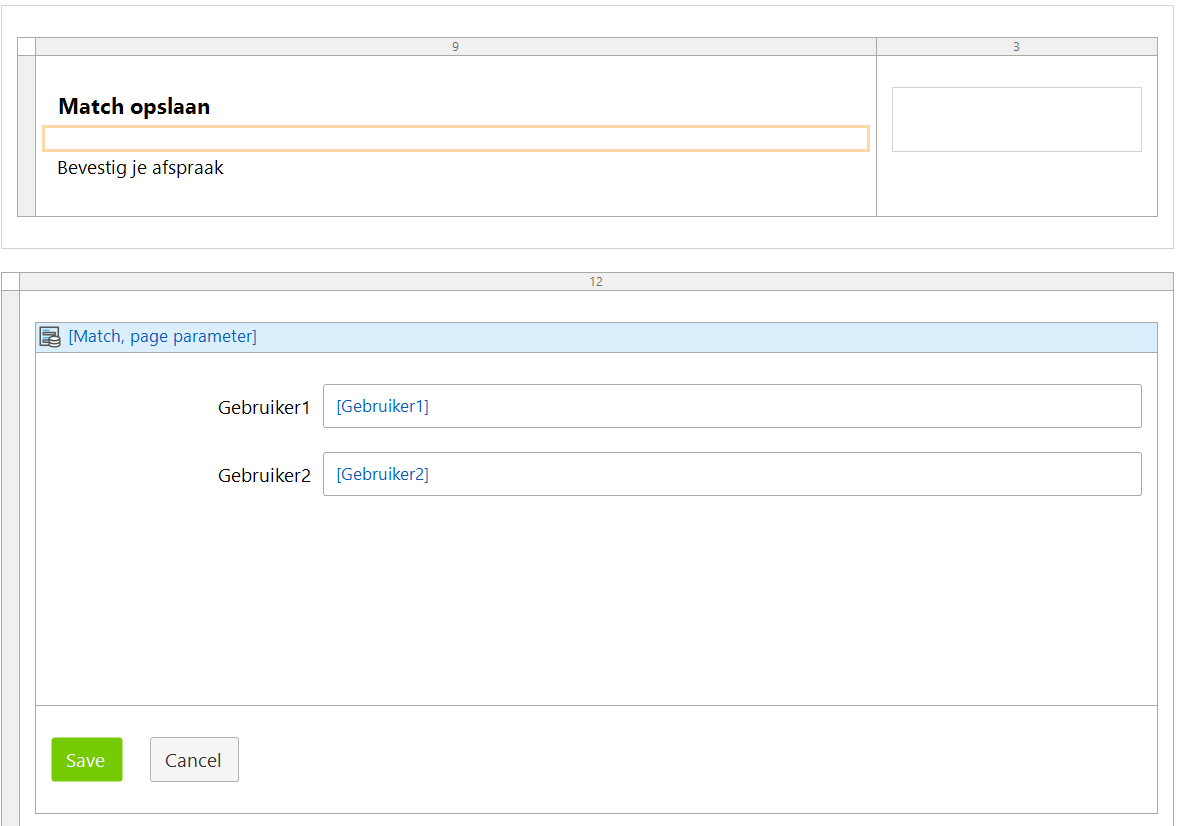
### Security en Authorisatie

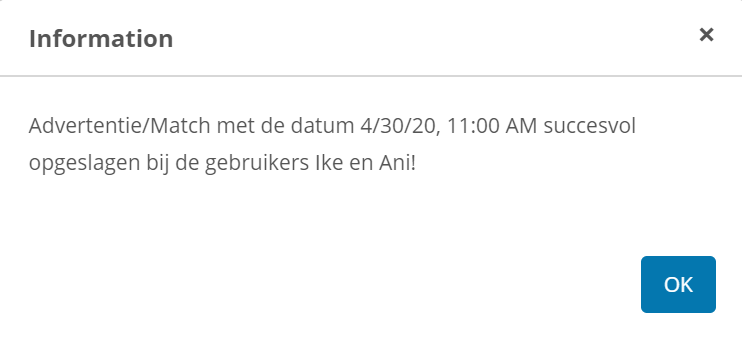
De volgende gebruikersrollen hebben recht van toegang tot de pagina:

***Rol1****:* Administrator, heeft inzicht in wat er op de pagina gebeurd en kan de pagina zelf aanpassen. Heeft de autorisatie om rechten te veranderen en inhoudelijke dingen toe te voegen.

***Rol2:***Student/gebruiker, kan chats versturen en teruglezen. Kan de pagina zelf niet veranderen.

## *PG-naam8:* Afspraak bevestigen

**

**

Deze pagina ondersteunt Use Case *XX\_UC\_003VanMatchNaarDiner.*Je navigeert naar deze pagina door vanuit de advertentiepagina op de knop maak afspraak te klikken en vanuit daar op de afspraak bevestigen pagina te komen.

### Gebruikersinteractie.

Op deze pagina zijn de volgende gebruikersinteracties mogelijk:

**Interactie1**: Gebruiker kan een afspraak met een andere gebruiker bevestigen door de gebruiker aan te klikken en op de knop save te drukken.

**Interactie2**: Gebruiker kan de interactie stopzetten door op de knop cancel te klikken.

### Vormgeving en styling

De volgende specifieke thema’s en CSS-styling regels zijn gebruikt bij de vormgeving van de pagina:

***Thema/regel1****:* Pagina is aangemaakt d.m.v. een data view template. Er wordt CSS gebruikt voor de dynamische ligging van knoppen en afbeeldingen, zodat deze er verzorgd uitzien. De knoppen zijn overzichtelijk en duidelijk.

### Security en Authorisatie

De volgende gebruikersrollen hebben recht van toegang tot de pagina:

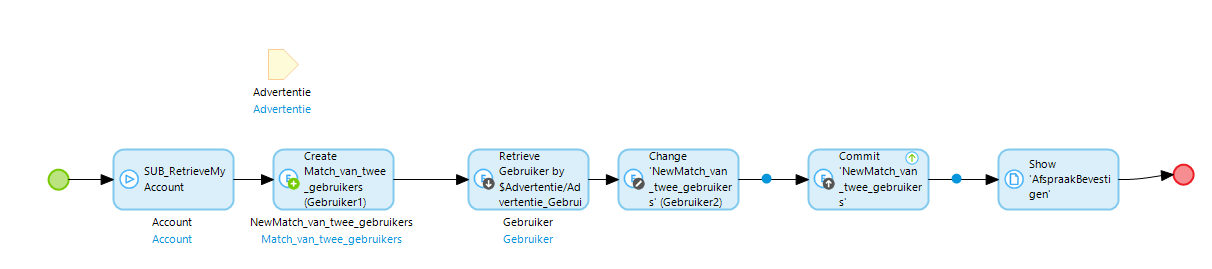
**Rol1**: Administrator, heeft inzicht in wat er op de pagina gebeurd en kan de pagina aanpassen. Heeft de autorisatie om rechten te veranderen en inhoudelijke dingen toe te voegen.

**Rol2:** Student/gebruiker, kan chats versturen en teruglezen. Kan de pagina zelf niet veranderen.

# Microflows

Hier vind je een overzicht van de geïmplementeerde microflows binnen de SamenEten app.

## ACT\_Create Afspraak

**

*Deze microflow maakt het mogelijk om een afspraak te maken tussen 2 gebruikers en deze afspraak is dus eigenlijk een match.*

### Invoer en uitvoer

**Parameter1***: Advertentie, deze parameter is een entiteit waarin de gegevens van een advertentie worden opgeslagen. Deze entiteit heeft een associatie met de entiteit Gebruiker & Afspraak.*

**Resultaat:** *Het resultaat van de microflow is dat de microflow een afspraak maakt tussen 2 gebruikers aan de hand van een advertentie.*

### Implementatie toelichting.

*Deze microflow begint met het ophalen van het account die op dat moment ingelogd is. Vervolgens creëert deze een match/afspraak met de ingelogde gebruiker en met de gebruiker van de advertentie. Deze match/afspraak wordt naar de database verstuurd door middel van een commit. Zodra dit goed gegaan is wordt er een afspraak bevestiging verstuurd.*

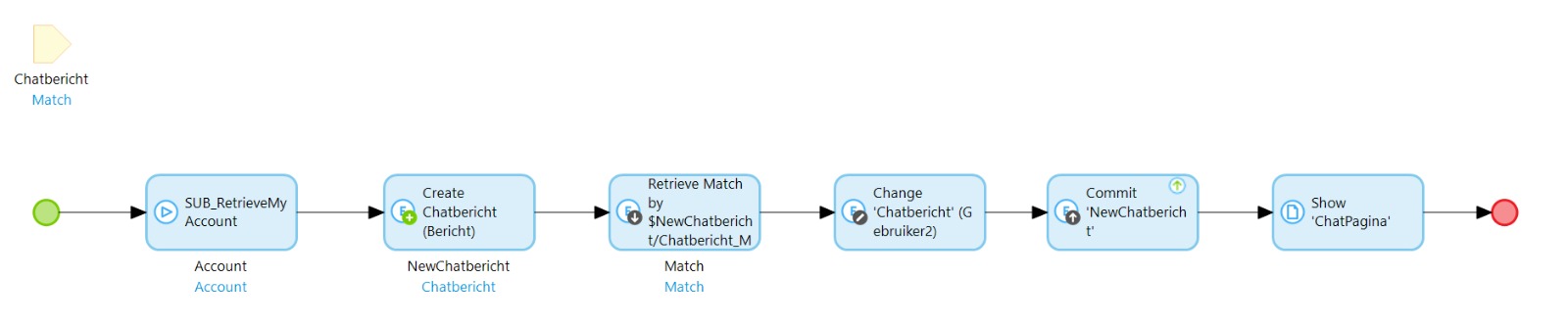
### Foutsituaties en exceptions.

*De microflow kan foutsituaties herkennen en geeft deze weer in de errors scherm binnen Mendix. Zo kan de Administrator deze errors herkennen en oplossen.*

### Security en Autorisatie

De gebruikersrollen *Student* en Administrator hebben recht van uitvoering van de microflow. *De administrator kan dit omdat hij de microflow moet kunnen testen. De student heeft ook recht van uitvoering omdat die de eindgebruiker is.*

## ACT\_CreateBericht



*Deze microflow maakt het mogelijk om een bericht aan te maken en te versturen tussen 2 gebruikers.*

### Invoer en uitvoer

**Parameter1***: Chatbericht, deze parameter is een entiteit waarin nieuwe berichten worden opgeslagen. De entiteit heeft een associatie met de entiteiten Gebruiker en Match.*

**Resultaat:** *Het resultaat van de microflow is dat er succesvol een bericht wordt aangemaakt en verstuurd naar de gebruiker met wie een match is gemaakt.*

### Implementatie toelichting

*De microflow begint met het ophalen van het account van gebruiker 1. Verder wordt er een bericht aangemaakt volgens de objecten, bericht, datum en afzender in de entiteit: Chatbericht. Vervolgens haalt de microflow de match tussen de twee gebruikers op zodat de berichten naar de juiste persoon worden verstuurd. Vervolgens wordt het aangemaakte chatbericht gecommit zodat het opgeslagen blijft en de microflow eindigt met het laten zien van de pagina ChatPagina.*

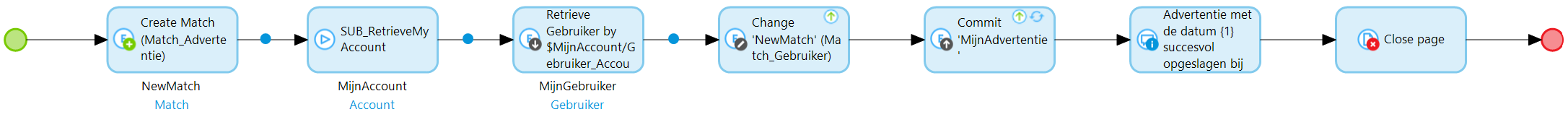
### Foutsituaties en exceptions

*In het geval dat er geen account kan worden opgehaald, geen match gevonden kan worden en de berichten niet gecommit zijn zal er een error scherm getoond worden binnen Mendix. Zo weet de administrator dat er iets verkeerd gaat.*

### Security en Autorisatie

De gebruikersrollen *Student* en Administrator hebben recht van uitvoering van de microflow. *De administrator kan dit omdat hij de microflow moet kunnen testen. De student heeft ook recht van uitvoering omdat die de eindgebruiker is.De administrator kan aanpassingen doen aan de microflow binnen mendix. Het systeem past de microflow toe op de pagina zodat de gebruiker het gewenste resultaat ziet.*

## *ACT\_SaveAdvertentie*



*Deze microflow maakt het mogelijk om een advertentie te creëren en op te slaan.*

### Invoer en uitvoer

**Parameter1***: MijnAdvertentie, deze parameter is een entiteit waarin nieuwe advertenties worden aangemaakt en opgeslagen. De entiteit heeft een associatie met de entiteiten Gebruiker en Match. Zo worden gebruikers gekoppeld aan aangemaakte advertenties en kan er een match gemaakt worden met een andere gebruiker.*

**Resultaat:** *Het resultaat van de microflow is dat er een advertentie aangemaakt wordt waar andere gebruikers een match mee kunnen maken.*

### 5.3.2 Implementatie toelichting

*De microflow begint met het creëren van een match met de associatie naar MijnAdvertentie. Verder wordt er een account opgehaald door middel van de RetrieveMyAccount microflow. Vervolgens haalt de microflow Mijn Gebruiker op. Daarna wordt Mijn Gebruiker geassocieerd met de nieuwe Match over de associatie Match\_Gebruiker. Tot slot wordt MijnAdvertentie gecommit en de advertentie met datum opgeslagen bij Gebruiker2. De microflow eindigt met het sluiten van de pagina.*

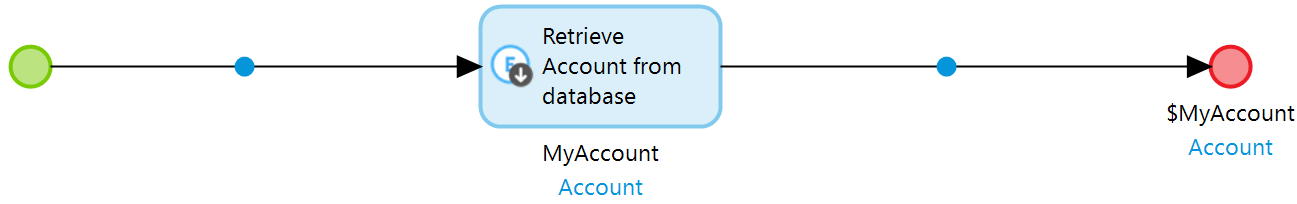
### Foutsituaties en exceptions.

*In het geval dat er geen account kan worden opgehaald, geen match gevonden kan worden en de berichten niet gecommit zijn zal er een error scherm getoond worden binnen Mendix. Zo weet de administrator dat er iets verkeerd gaat.*

### Security en Autorisatie

De gebruikersrollen *Student* en Administrator hebben recht van uitvoering van de microflow. *De administrator kan dit omdat hij de microflow moet kunnen testen. De student heeft ook recht van uitvoering omdat die de eindgebruiker is.De administrator kan aanpassingen doen aan de microflow binnen mendix. Het systeem past de microflow toe op de pagina zodat de gebruiker het gewenste resultaat ziet.*

## *SUB\_RetrieveMyAccount*.

*Deze microflow maakt het mogelijk om een account op te halen.*

### Invoer en uitvoer

**Parameter1***: MyAccount is een entiteit in de System.User die wordt opgehaald om een gewenste actie uit te voeren voor het account van de gebruiker.*

**Resultaat:** *Het resultaat van de microflow is dat er een account wordt opgehaald vanuit de database.*

### Implementatie toelichting.

*De microflow wordt alleen gebruikt voor het ophalen van accounts binnen de entiteit System.User. Het is een korte en simpele microflow die eindigt met het opgehaalde account.*

### Security en Autorisatie

De gebruikersrol Administrator heeft recht van uitvoering van de microflow. *De administrator kan dit omdat hij de microflow moet kunnen testen. De student heeft geen recht van uitvoering omdat die de account zelf niet hoeft op te halen of te testen. De administrator kan aanpassingen doen aan de microflow binnen mendix. Het systeem past de microflow toe op de pagina zodat de gebruiker het gewenste resultaat ziet.*

# Referenties

1. [www.mendix.com](http://www.mendix.com)*: official web-site of the mendix platform.*
2. <http://iso25000.com/index.php/en/iso-25000-standards/iso-25010> *standard product quality model.*