 Functioneel Ontwerp

Versie: 0.4.0  
Gemaakt door: Wouter Jansen

Datum: September 2016

**Versie beheer**

|  |  |  |
| --- | --- | --- |
| Versie | Wijzigingen | Datum |
| 0.1.0 | Eerste Versie | 30 Augustus 2016 |
| 0.2.0 | Scenario’s & Uitgangspunten toegevoegd | 11 September 2016 |
| 0.3.0 | Annulerings scenario toegevoegd, Database diagram toegevoegd | 19 September 2016 |
| 0.4.0 | Wijzigingsopdracht | 27 September 2016 |

Inhoudsopgave

[Achtergrond 4](#_Toc465297537)

[Actoren 5](#_Toc465297538)

[Use Case Diagram 6](#_Toc465297539)

[Klassendiagram 7](#_Toc465297540)

[Databasediagram 8](#_Toc465297541)

[Scenario’s 9](#_Toc465297542)

## Achtergrond

Op 23 Augustus 2016 werd mij verteld dat ik (Wouter Jansen) een conferentie website moet bouwen, deze website moet de huis-stijl hebben van ictinnovations.org, hierbij zijn ook nog aparte functies en eisen nodig, die in dit document vast zullen worden gesteld. Behalve dat er documentatie moet gemaakt worden voor de website moet de website ook actueel gebouwd worden

Het belang bij het correct afronden van deze opdracht zit in het feit dat als dit project correct kan worden afgerond ik klaar ben om de examens goed af te ronden.

Het nut van dit document is het noteren van alle functies in het systeem, en een goed overzicht te geven welke functies er in de website komen. Ook om aan te geven hoe de databases eruit komen te zien. Verder is dit vooral een document voor de developers die dit als controle en als naslag werk kunnen gebruiken.

## Actoren

Alle actoren die voorkomen in de website zijn:

Organisator

Spreker  
Bezoeker

Hieronder een kleine toelichting over de rollen van deze gebruikerstypen

**Organisator**

De organisator is degene met de hoogste ‘rang’ in het system, hij/zij heeft toegang tot alle organisator functies, maar kan ook functies zoals het reserveren van tickets uitvoeren. En het aanvragen van een tijdslot

**Spreker**

Een spreker is iemand die een tijdslot gaat gebruiken om te praten over een specifiek onderwerp, tijdens de registratie kan iemand aangeven of hij/zij een spreker is of niet. Hij heeft toegang tot functies zoals ticket reserveren, maaltijd reserveren en een tijdslot reserveren

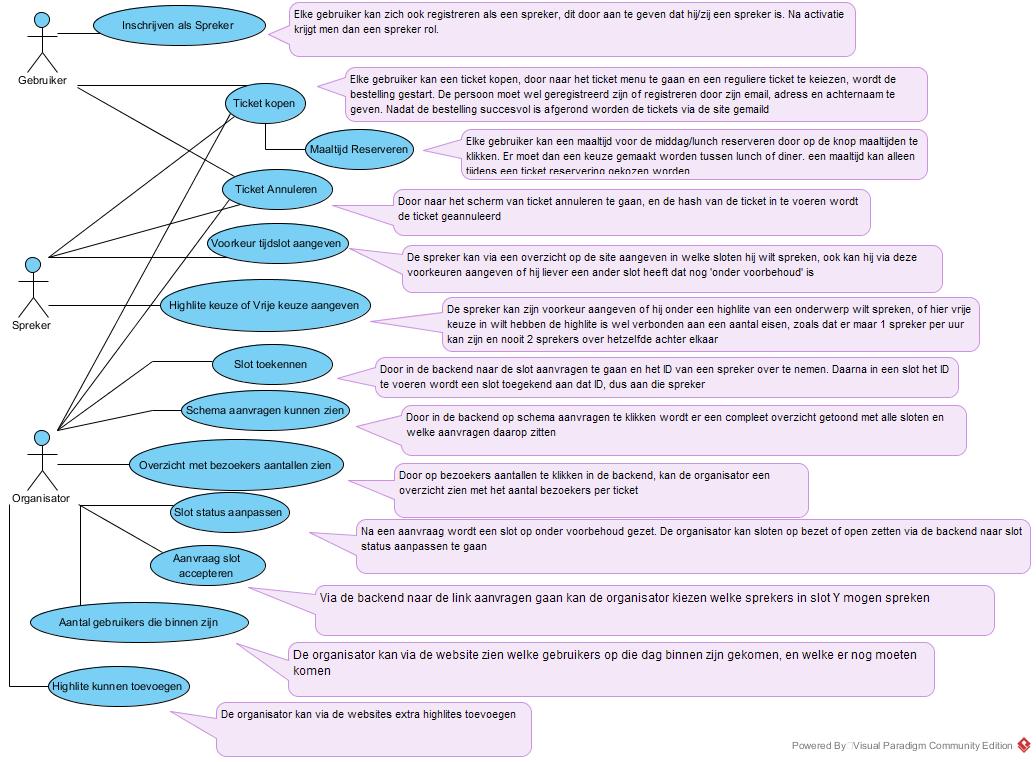
**Bezoeker**

Een bezoeker is in dit geval al geregistreerd op de website, de bezoeker heeft toegang tot de functies van een ticket bestellen en een maaltijd reserveren. Verder kan hij/zij zijn eigen bestelde ticket zien

## Use Case Diagram

Een use-case diagram is een diagram waarin het gebruik van het systeem wordt uitgelegd aan de hand van de verzoeken van buitenaf, in dit geval de Gebruiker, Spreker en organizators

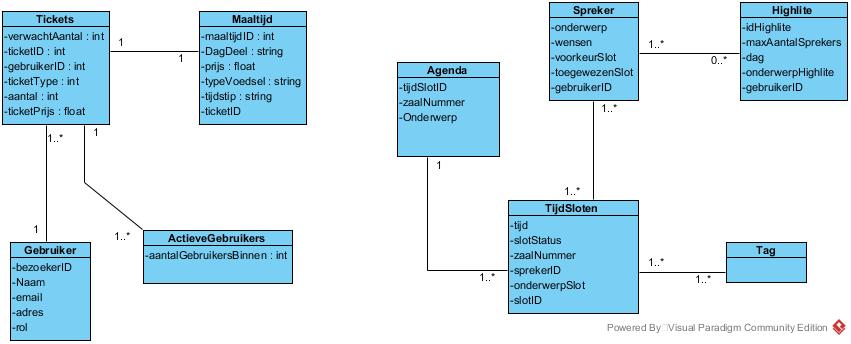
De bedoeling van het use case diagram is aantonen wie een bepaalde handeling kan uitvoeren door middel van het systeem, hierdoor worden de mogelijkheden per gebruiker in kaart gebracht



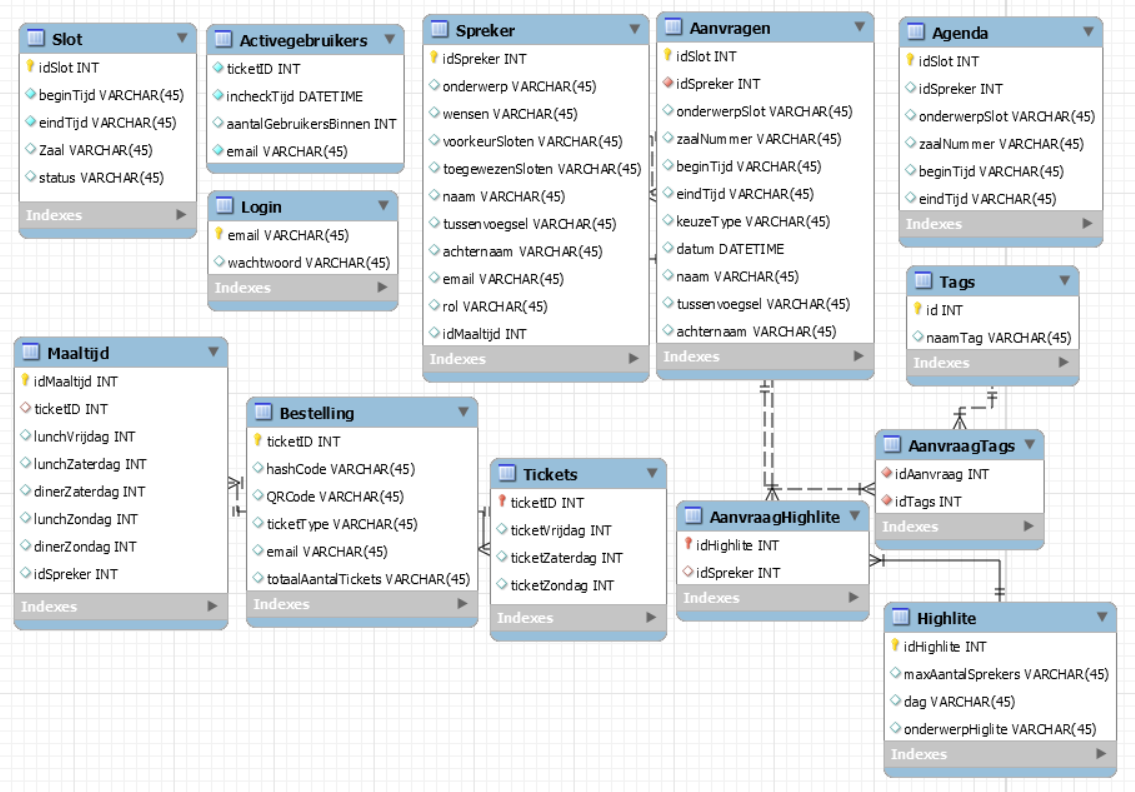
## Klassendiagram

Het klassendiagram is een weergave van connecties, waarin het duidelijk moet worden welke database interacties met elkaar worden uitgevoerd en hoe bepaalde klassen relaties met elkaar hebben.

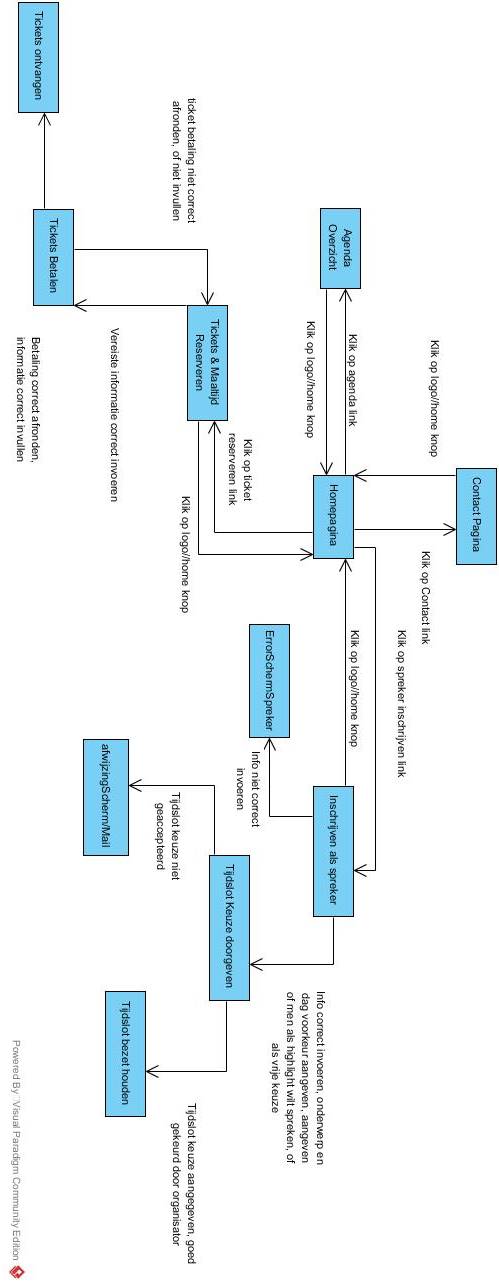
In het huidige scenario moet het duidelijk zijn welke waarden met elkaar hebben te maken en welke waarde nodig zijn voor het functioneren en aanmaken van een instantie van een klasse.



## Databasediagram



Schermboom



## Scenario’s

Code: A1  
Process: Reservering annuleren  
Rol: Bezoeker/Spreker  
Pre: De gebruiker is ingelogd, en heeft al een reservering geplaatst  
Post: De gebruiker heeft zijn bestaande reservering geannuleerd en zijn geld terug gekregen

**Primair Scenario**

|  |  |
| --- | --- |
| Stap | Activiteit |
| 1 | Bezoeker klikt op ticket annuleren |
| 2 | Systeem geeft de annuleren pagina |
| 3 | Bezoeker vult zijn hashcode in, die in de ticket staat |
| 4 | Systeem geeft vraagt om een bevestiging van de klant, en meld dat als de klant verder nog wilt bestellen niet garandeerd dat er nog plek is |
| 5 | Bezoeker bevestigd keuze |
| 6 | Zodra de annulering bevestigd is word de ticket verwijderd, het ticket aantal van desbetreffende dag omhoog gezet, en het geld wordt terug geschreven |

**Secundair Scenario**Code: B1

**Stap 4.1:** *de klant bevestigt zijn annulering niet*

De annulering wordt niet voltooid, dus blijft de ticket actief

Code: B2  
Process: Reservering plaatsen   
Rol: Bezoeker/Spreker  
Pre: De gebruiker is ingelogd  
Post: De gebruiker heeft tickets gekocht & een barcode ontvangen

**Primair Scenario**

|  |  |
| --- | --- |
| Stap | Activiteit |
| 1 | Bezoeker klikt op tickets |
| 2 | Systeem geeft de tickets pagina |
| 3 | Bezoeker kiest een dag & ticket combinatie, ook geeft hij aan welke maaltijd hij wilt |
| 4 | Systeem geeft de betaal pagina |
| 5 | Bezoeker controleert ingevulde data, en betaald |
| 6 | Zodra de betaling bevestigt is stuurt het systeem een email met daarin de ticket + QR code, samen met nog een code voor zijn maaltijd |

**Secundair Scenario**Code: B2

**Stap 3.1:** *Ticket & dag combinatie is niet meer beschikbaar*

* Systeem geeft de melding dat de tickets niet meer beschikbaar zijn

Helaas! De tickets zijn niet meer beschikbaar, kies een andere combinatie

**Stap 6.1:** *De betaling wordt niet afgerond*

* Systeem geeft de melding dat de betaling niet is afgerond en stuurt men terug naar de ticket pagina

Er is iets fout gegaan met de betaling, probeer het opnieuw!

Code: C1  
Process: Tijd slot aanvragen  
Rol: Spreker  
Pre: Gebruiker ingelogd als een spreker  
Post: Een aanvraag is geplaatst op de spreker zijn gekozen tijdslot

**Primair Scenario**

|  |  |
| --- | --- |
| Stap | Activiteit |
| 1 | De spreker gaat naar de ‘spreker registreer’ pagina |
| 2 | Systeem geeft de registratie pagina weer voor sprekers |
| 3 | Spreker vult al zijn informatie in, onder andere email en welke dag hij wilt spreken |
| 4 | Zodra de spreker zijn dag keuze geeft, geeft het systeem de highlites relatief aan die dag samen met een eventuele maaltijd keuze |
| 5 | De spreker kiest of hij voor een highlite wilt spreken of een vrije keuze, ook geeft hij aan of hij lunch of diner wilt consumeren |
| 6 | Als de spreker een vrije keuze heeft gekozen gaat het systeem direct door naar de sloten keuze pagina en geeft alle sloten voor die dag, anders gaat het systeem ook naar deze pagina maar haalt alleen sloten op die vrij zijn voor die highlite |
| 7 | De spreker selecteert een slot (Hij kan hierbij ook voorkeur aangeven voor een slot dat nog ‘onder voorbehoud’ is, een slot dat bezet is kan niet gekozen worden. Hierna submit hij zijn keuze |
| 8 | Het systeem slaat de keuze op, zet het slot onder voorbehoud en stuurt een bericht naar de organisator dat er een spreker een slot gekozen heeft |

**Secundair Scenario**Code: C1

**Stap 1.1 – 7.1 :** *Gebruiker sluit de browser of drukt op de terug knop*

* Systeem zal de eerder ingevulde gegevens niet opslaan, en worden dus weggegooid

**Stap 7.2:** *Gebruiker kiest een bezet slot, het systeem geeft het volgend bericht:*

* Dit slot is bezet! Het kan niet geselecteerd worden

**Stap 5.1:** *Gebruiker kiest een highlite die al vol is, het systeem geeft het volgend bericht:* - Deze highlite heeft al genoeg sprekers, kies een andere highlite of vrije keuze

Code: C2  
Process: Tijd slot toekennen aan spreker  
Rol: Organisator  
Pre: De organisator bevind zichzelf op de overzicht pagina van organisatoren, een van de geregistreerde sprekers heeft een tijdslot gekozen  
Post: Een spreker is toegekend aan zijn gekozen tijdslot

**Primair Scenario**

|  |  |
| --- | --- |
| Stap | Activiteit |
| 1 | De organisator gaat naar de ‘aangevraagde sloten’ pagina |
| 2 | Het systeem geeft het sloten overzicht pagina, en haalt alle recente aanvragen op |
| 3 | De organisator noteert gegevens van de spreker die hij wilt toekennen, en gaat naar de sloten toekennen pagina |
| 4 | Het systeem geeft de sloten toekennen pagina |
| 5 | De organisator zet de gegevens van de spreker in het gekozen slot en drukt op submit |
| 6 | Het systeem koppelt de spreker aan het gekozen slot, en zet zijn onderwerp + naam in de lijst |

**Secundair Scenario**Code: C2

**Stap 3.1, 4.1, 5.1:** *Gebruiker sluit de browser of drukt op de terug knop*

* Systeem zal de eerder ingevulde gegevens niet opslaan, en worden dus weggegooid

**Uitgangspunten**

Bij het onderdeel uitgangspunten wordt er in nog meer detail uitgelegd wat er nu daadwerkelijk gebeurd en wat noodzakelijk is voor het verrichten van bepaalde acties in de website, zowel database gericht als systeem gericht.

**Inloggen**

1. Een bezoeker van het systeem moet kunnen inloggen
2. Voor het aanmaken van een nieuwe gebruiker:
   1. De gebruikersnaam is voornaam + tussenvoegsel + achternaam, de gebruiker wordt als zodanig aangesproken
   2. e-mail: is uniek, wordt gecontroleerd via een bevestigings e-mail, pas na bevestigen heeft men toegang tot het systeem
   3. wachtwoord is minimaal 7 karakters groot/lang
   4. De rol kan gekozen worden tijdens het registreren (Spreker of Klant)
   5. De rol organisator wordt via het systeem gegeven
3. Een gebruiker heeft altijd 1 rol, hij kan er niet meerdere hebben
4. Een gebruiker logt in met e-mail en wachtwoord
5. Indien een gebruiker inlogt is de callback:
   1. null, als de combinatie niet juist
   2. null, als er velden leeg worden gelaten
   3. rol indien de combinatie juist is
   4. Er wordt gekozen om geen extra informatie te geven tijdens het inloggen, zo weten eventuele hackers niet of de info correct is
6. Data wordt opgeslagen in MySQL
7. CRUD: Create - Registreren  
    Read - Login  
    Update - Inlog gegevens wijzigen  
    Delete -
8. Een gebruiker blijft ingelogd tijdens het wisselen van pagina’s (sessie)
9. Er zijn enkele pagina’s die alleen toegankelijk zijn voor specifieke rollen (zoals het scherm van gebruikers die binnen zijn. Dit is alleen voor de organisator)

Benodigde schermen

1. Home
2. Registreren
3. Login
4. Backend(Profile)

Endpoints

/api/registratie (POST) handelt nieuwe gebruikers af: handle

/api/login (POST)

**Ticket Reserveren**

1. Een gebruiker van het systeem moet ingelogd zijn
2. Voor het aanmaken van een reservering zijn de volgende gegevens nodig:
   1. Een geldig ticket - dag combinatie, hiermee wordt bedoeld dat voor die dag de tickets nog niet zijn uitverkocht
   2. Gebruiker moet geactiveerd zijn
   3. De gebruiker heeft verder geen reserveringen openstaan(reserveringen die nog betaald moeten worden)
3. Een gebruiker heeft altijd een dag & ticket combinatie geselecteerd
4. Indien een gebruiker begint met het aanmaken van een nieuwe reservering is de callback:
   1. null, indien er geen tickets meer aanwezig zijn
   2. aantal kaartjes die nog beschikbaar zijn
5. Een gebruiker kan maar 1 reserving tegelijk plaatsen, daarmee wordt bedoeld dat hij maar kaartjes voor 1 dag & ticket combinatie(als het aantal beschikbaar is) kan reserveren na de reservering een nieuwe reservering kan beginnen voor een andere dag & ticket combinatie.
6. Een gebruiker moet de ticket reservering direct betalen
7. Als een dag voorbij is, is het niet meer mogelijk om tickets voor die dag te kopen
8. Data wordt opgeslagen in MySQL
9. CRUD: create - aanmaken van nieuwe reservering  
    read: - aantal beschikbare kaarten uitlezen  
    update: - aantal kaarten verminderen van die dag  
    delete:
10. Een gebruiker blijft ingelogd tijdens wisselen van pagina’s (sessie)

Benodigde schermen

1. Home
2. Ticket pagina
3. Reserverings pagina
4. Betaal pagina

Endpoints

/api/ homepagina

/api/tickets (post) dag & ticket pagina wordt gekozen, en tickets worden opgehaald  
/api/reservering (post) aantal tickets worden geselecteerd  
/api/betaling (get) er wordt opgehaald of de tickets betaald zijn of niet

**Maaltijd Reserveren**

1. Een gebruiker van het systeem moet ingelogd zijn
2. Voor het aanmaken van een maaltijd zijn de volgende gegevens nodig:
   1. Een geldig ticket - dag combinatie, hiermee wordt bedoeld dat voor die dag de tickets nog niet zijn uitverkocht
   2. Gebruiker moet geactiveerd zijn
   3. De gebruiker heeft verder geen reserveringen openstaan(reserveringen die nog betaald moeten worden)
3. Een gebruiker heeft altijd een dag & ticket combinatie geselecteerd
4. Indien een gebruiker begint met het aanmaken van een nieuwe reservering is de callback:
   1. null, indien er geen maaltijd-tickets meer aanwezig zijn
   2. aantal maaltijd-tickets die nog beschikbaar zijn
5. Een gebruiker kan maar 1 reserving tegelijk plaatsen, daarmee wordt bedoeld dat hij maar kaartjes voor 1 dag & maaltijd combinatie(als het aantal beschikbaar is) kan reserveren na de reservering een nieuwe reservering kan beginnen voor een andere dag & maaltijd combinatie.
6. Een gebruiker moet de maaltijd reservering direct betalen
7. Data wordt opgeslagen in MySQL
8. CRUD: create - aanmaken van nieuwe reservering  
    read: - aantal beschikbare maaltijd-tickets uitlezen  
    update: - aantal maaltijd-tickets verminderen van die dag  
    delete:
9. Een gebruiker blijft ingelogd tijdens wisselen van pagina’s (sessie)

Benodigde schermen

1. Home
2. maaltijd-ticket pagina
3. Reserverings pagina
4. Betaal pagina

Endpoints

/api/ homepagina

/api/tickets (post) dag & maaltijd pagina wordt gekozen, en maaltijd-tickets worden opgehaald  
/api/reservering (post) aantal maaltijd-tickets worden geselecteerd  
/api/betaling (get) er wordt opgehaald of de maaltijd-tickets betaald zijn of niet

**Highlite kiezen**

**1.** Een spreker moet zijn gegevens al hebben ingevuld, en een dag gekozen hebben. Dit wordt gedaan op de registratie pagina van een spreker

2. De volgende gegevens zijn dus al ingevuld: een geldig email, een dag en een highlite die nog slots vrij heeft

3. De slot die geselecteerd is mag niet hetzelfde onderwerp hebben als die ervoor of die erna. Slots die niet gekozen kunnen worden, worden dus ‘uitgezet’

4. Op vrijdag zijn er drie sprekers per highlight, op zaterdag is dat vijf

5. Alle data wordt opgeslagen in MySQL

6. CRUD: create – aanmaken van een spreker

Read – aantal beschikbare slots uitlezen

Update – een specifiek slot bezetten

Benodigde schermen

1. Home

2. Spreker pagina

3. Sloten pagina

Endpoints

/api/ homepagina

/api/sprekerpagina (post) email & dag worden doorgegeven, ook wordt de highlite gekozen relatief aan de dag, of een vrije keuze wordt doorgegeven

/api/sloten (get) sloten worden opgehaald gefiltered op dag en eventueel op highlite

/api/sloten(post) een slot wordt gekozen die de spreker wilt