2024 - 2025

|  |
| --- |
|  |
| Plan Van AanpakWachtlijsten |
|  |

Jonas Quintiens
Student Bachelor in de Toegepaste Informatica – Applicatieontwikkeling

Inhoudsopgave

[Inleiding 3](#_Toc195548332)

[Business case: 3](#_Toc195548333)

[Technische Analyse 4](#_Toc195548334)

[Huidige Architectuur 4](#_Toc195548335)

[Risico Analyse 4](#_Toc195548336)

[Database Ontwerp 4](#_Toc195548337)

[E-mailnotificaties 4](#_Toc195548338)

[Business Logic 5](#_Toc195548339)

[Functionele Analyse 5](#_Toc195548340)

[Use cases: 5](#_Toc195548341)

[Conclusie 8](#_Toc195548343)

[Doelstellingen 8](#_Toc195548344)

[Stappenplan 8](#_Toc195548345)

[Stap 1: Opzetten van het Database model 8](#_Toc195548346)

[Stap 2: Implementatie van de Wachtlijstfunctionaliteit 8](#_Toc195548347)

[Stap 3: E-mailnotificaties 9](#_Toc195548348)

[Stap 4: Betalingsverwerking 9](#_Toc195548349)

[Stap 5: Beheer en Monitoring 9](#_Toc195548350)

[Stap 6: Testen en Deployen 9](#_Toc195548351)

# Inleiding

Binnen de bestaande applicatie is er behoefte ontstaan aan een wachtlijstfunctionaliteit voor zowel evenementen als groepsinschrijvingen. Dit stelt gebruikers in staat zich alsnog aan te melden wanneer een evenement of groep volgeboekt is, met de mogelijkheid om door te schuiven zodra er een plek vrijkomt.

De implementatie zal starten met een **Minimum Viable Product (MVP)**, waarbij de focus ligt op het ontwikkelen van de basisfunctionaliteit voor wachtlijsten bij evenementen. Deze eerste versie bevat enkel de noodzakelijke onderdelen: het inschrijven op de wachtlijst, het versturen van uitnodigingen zodra er een plaats vrijkomt, en de mogelijkheid tot betalen en definitieve inschrijving.

Na deze MVP-fase zal het systeem verder worden uitgebreid. Op basis van gebruikservaringen en feedback wordt in volgende ontwikkelingscycli de functionaliteit verfijnd en uitgebreid, onder andere naar groepsinschrijvingen en aanvullende features zoals herinneringen, geautomatiseerde opvolging en beheertools.

###### Mindmap:



### Business case:

Het is de bedoeling dat gebruikers zich kunnen inschrijven voor een wachtlijst. De wachtlijst kan toebehoren aan een evenement of inschrijvingsperiode. Als er een plaats vrij komt krijgt de gebruiker een mail met een betalingsuitnodiging. Als deze betaald is dan wordt de wachtlijst inschrijving omgezet naar een echte inschrijving.

# Technische Analyse

### Huidige Architectuur

De bestaande applicatie is gebouwd met Django. Dit betekent dat we gebruik kunnen maken van Django's models, views en signalen om de wachtlijstfunctionaliteit te implementeren. De gebruikersregistratie en betaling kunnen geïntegreerd worden met bestaande authenticatie- en betalingssystemen. De e-mailconfiguratie en Stripe-integratie zijn al aanwezig in de applicatie.

### Risico Analyse

| **Risico** | **Impact** | **Maatregel** |
| --- | --- | --- |
| Gebruiker reageert niet op uitnodiging | Hoog | Gebruik een timeout en schakel automatisch door naar de volgende persoon |
| Meerdere plekken komen tegelijk vrij | Middel | Voeg logica toe om meerdere gebruikers tegelijk uit te nodigen of automatische uitnodigen te versturen |
| Gebruiker betaalt niet of betaling mislukt | Hoog | Voeg fallback-mechanisme toe, zoals herinneringen of annulering na X uur |

### Database Ontwerp

Om een Minimum viable product te bekomen

Wachtlijst model met relatie naar evenement en inschrijvingsperiode

Event methoden voor uitnodingen te versturen

### E-mailnotificaties

Django's ingebouwde e-mailfunctionaliteit kan worden gebruikt om gebruikers op de wachtlijst te informeren zodra een plek vrijkomt. Omdat de e-mailconfiguratie al aanwezig is, hoeft enkel de logica voor de notificaties geïmplementeerd te worden.

### Business Logic

* Wanneer een event vol zit, kunnen gebruikers zich aanmelden voor de wachtlijst.
* Als een plaats vrijkomt, krijgt de eerste gebruiker in de wachtrij een e-mail.
* De gebruiker heeft een bepaalde tijd om te betalen voordat de uitnodiging vervalt.
* Na betaling wordt de wachtlijstinschrijving omgezet naar een definitieve inschrijving.
* Indien de gebruiker niet binnen de gestelde tijd betaalt, wordt de volgende persoon op de wachtlijst benaderd.

# Functionele Analyse

### Use cases:

**Functionaliteit:**
Als lid kan ik mij inschrijven op een wachtlijst voor een evenement wanneer het volledig volzet is, zodat ik op de hoogte word gebracht als er een plaats vrijkomt.

**Normaal verloop:**

1. Het lid bezoekt de evenementpagina waarop staat dat het evenement volzet is.
2. Het systeem biedt de mogelijkheid aan om op de wachtlijst te komen.
3. Het lid klikt op “Inschrijven op wachtlijst”.
4. Het systeem vraagt het e-mailadres van het lid.
5. Het lid vult het e-mailadres in en bevestigt.
6. Het systeem bevestigt de succesvolle inschrijving op de wachtlijst.
7. De inschrijving wordt geregistreerd in het systeem.

**Alternatief verloop:**

* **Fout e-mailadres:** Als het lid een ongeldig e-mailadres invoert, toont het systeem een foutmelding en vraagt om een geldig adres in te geven.
* **Reeds ingeschreven:** Als het lid reeds op de wachtlijst staat, krijgt het een melding dat het al is ingeschreven.

**Functionaliteit:**
Als lid kan ik mij inschrijven wanneer ik een uitnodiging ontvang, door een ticket te kopen binnen een beperkte geldigheidsduur.

**Normaal verloop:**

1. Het lid ontvangt een uitnodiging via e-mail.
2. De uitnodiging bevat een unieke link naar het inschrijvingsformulier en een deadline.
3. Het lid opent de link en ziet de beschikbare ticketopties.
4. Het lid selecteert een ticket en voert de gevraagde gegevens in.
5. Het lid betaalt het ticket.
6. Het systeem bevestigt de succesvolle inschrijving.
7. De plaats wordt uit de vrije voorraad gehaald.

**Alternatief verloop:**

* **Deadline verstreken:** Als de link na de deadline wordt geopend, toont het systeem dat de uitnodiging is verlopen.
* **Betaling mislukt:** Als de betaling faalt, wordt het ticket niet gereserveerd en krijgt het lid een foutmelding.

**Functionaliteit:**
Als beheerder kan ik voor een evenement of inschrijvingsperiode een wachtlijst inschakelen.

**Normaal verloop:**

1. De beheerder logt in op het beheerdersportaal.
2. De beheerder opent de instellingen van het gewenste evenement.
3. De beheerder schakelt de wachtlijstoptie in.
4. Het systeem activeert de wachtlijst en toont deze optie aan bezoekers van de evenementpagina.
5. Inschrijvingen worden vanaf dan geregistreerd op de wachtlijst.

**Alternatief verloop:**

* **Geen evenement geselecteerd:** Als er geen evenement is geselecteerd, toont het systeem een foutmelding.
* **Wachtlijst al actief:** Indien de wachtlijst al actief is, geeft het systeem aan dat dubbele activering niet nodig is.
* **Wachtlijst activeren bij het aanmaken van een nieuw evenement of** inschrijvingsperiode.

**Functionaliteit:**
Als beheerder kan ik leden manueel selecteren van de wachtlijst en hen uitnodigen om zich in te schrijven, ongeacht automatische triggers.

**Normaal verloop:**

1. De beheerder logt in op het systeem.
2. De beheerder navigeert naar de wachtlijst van een specifiek evenement.
3. De beheerder selecteert één of meerdere leden uit de lijst.
4. De beheerder klikt op “Verstuur uitnodiging”.
5. Het systeem stuurt automatisch een uitnodigingsmail naar de geselecteerde leden.
6. De uitnodigingen worden geregistreerd als “verstuurd”.

**Alternatief verloop:**

* **Geen leden geselecteerd:** Als er geen leden zijn geselecteerd, toont het systeem een waarschuwing.
* **Fout in verzending:** Als het e-mailsysteem niet bereikbaar is, toont het systeem een foutmelding.

**Functionaliteit:**
Wanneer er plaatsen vrijkomen (bv. door annulering of uitbreiding), stuurt het systeem automatisch een uitnodiging naar de eerstvolgende leden op de wachtlijst.

**Normaal verloop:**

1. Er komt een plaats vrij voor een evenement.
2. Het systeem detecteert de wijziging (vb. door annulering of verhoging van de capaciteit).
3. Het systeem selecteert automatisch het eerstvolgende lid of leden op de wachtlijst.
4. Het systeem genereert uitnodigingen met een beperkte geldigheid.
5. Het systeem stuurt deze uitnodigingen via e-mail.
6. De status van de leden wordt bijgewerkt naar “uitgenodigd”.

**Alternatief verloop:**

* **Geen wachtlijst:** Als er geen wachtlijst is voor het evenement, wordt geen actie ondernomen.
* **Technische fout bij e-mail:** Als het systeem de e-mails niet kan versturen, registreert het een fout en toont een waarschuwing aan de beheerder.

Use Case Diagram:

# A diagram of a company  AI-generated content may be incorrect.

# Conclusie

### Doelstellingen

Een werkende wachtlijstfunctionaliteit opleveren voor evenementen.

E-mails automatisch verzenden wanneer er een plek vrijkomt.

Betalingen via Stripe verwerken en succesvolle inschrijvingen registreren.

De MVP zodanig opzetten dat deze later uitbreidbaar is naar groepsinschrijvingen.

Minimale technische overhead en goede integratie met de bestaande Django-architectuur.

### Stappenplan

#### Stap 1: Opzetten van het Database model

* Definieer het Waitlist model
* Definieer de relaties Event en User in Django ORM.
* Voeg migraties toe en pas deze toe op de database.

#### Stap 2: Implementatie van de Wachtlijstfunctionaliteit

* Voeg een mogelijkheid toe voor gebruikers om zich in te schrijven op een wachtlijst als een event vol is.
* Beheer de logica voor het vrijmaken van een plaats en het toewijzen van een uitnodiging aan de eerste in de rij.

#### Stap 3: E-mailnotificaties

* Schrijf en test e-mailtemplates voor uitnodigingen en herinneringen.
* Integreer de e-maillogica met de bestaande e-mailconfiguratie.

#### Stap 4: Betalingsverwerking

* Gebruik de bestaande Stripe-integratie om betalingen te verwerken.
* Verwerk succesvolle betalingen en converteer de inschrijving van wachtlijst naar definitieve deelname.

#### Stap 5: Beheer en Monitoring

* Voeg een admin-dashboard toe om de status van de wachtlijst in te zien.
* Log belangrijke acties zoals inschrijvingen, uitnodigingen en betalingen.

#### Stap 6: Testen en Deployen

* Voer unittests en integratietests uit.
* Deploy de wijzigingen en monitor gebruikersfeedback.