

# Project Beschrijving:

## **Titel:**

Trein abonnement systeem

## **Beschrijving:**

Met het abonnement systeem kan een gebruiker abonnementen kopen per regio die binnenin die regio geldig zijn. Zodat hij daar gebruik kan maken van de treinen. Ook kan de gebruiker tickets kopen die geldig zijn doorheen alle regio's.

De gebruiker moet zich eerst registreren op onze website en dan kan kiezen tussen een 6 maanden abonnement, een 3 maanden abonnement, een maand abonnement of een 10 beurtenkaart. De gebruiker kan ook kiezen in welke regio hij dit abonnement kan aankopen. Een gebruiker kan per regio slechts 1 abonnement aanschaffen, hij kan wel in meerdere regio's op het zelfde moment een abonnement hebben. Er is ook een optie voor kortingen, waarbij de user een kortingscode kan invullen en zo korting kan krijgen op zijn aankoop. Ook kunnen ze een gewoon ticket kopen. Bij de aankoop word er een bestelling geplaatst die dan word bijgehouden op de gebruiker zijn profiel. Mits deze ouder zijn dan 1 jaar worden ze van het profiel afgehaald en ergens gearchiveerd.

# Story 1: Registratie en Login:

Als een gebruiker,  
kan ik mij registreren op de website en inloggen op mijn account,  
zodat ik abonnementen en tickets kan kopen en mijn profiel kan beheren.

## **Wireframe:**

### **Registration page:**

**Request Type:** POST

**URL:** URL/register

**Request Body:** Een gebruiker object

**Input fields:**

- Naam
- Voornaam
- E-mailadres
- Wachtwoord
- Bevestig wachtwoord

**Response Code:** 200

**Knoppen :** Een knop “Account aanmaken” voor je registratie te bevestigen.

**Links:** Een mededeling met “Heb je al een account? Log in.” Met een link op “log in” die verwijst naar de login pagina.

### **Login page:**

**Titel:** “Inloggen op uw apparaat”

**Input fields:**

- E-mailadres
- Wachtwoord

**Knoppen:** Een knop “Inloggen” voor in te loggen na het ingeven van je gegevens.

**Links:** Een mededeling met “Nog geen account? Registreer nu”. Met een link op “Registreer nu” die verwijst naar de registratie pagina.

## **Acceptatiecriteria:**

- Alle velden moeten ingevuld zijn.
- E-mailadres moet een @ bevatten.
- Wachtwoord moet langer dan 8 char's zijn en moet letters, cijfers en een speciaal teken bevatten.
- Als user met E-mail al bestaat → Foutmelding
- Bij incorrect inloggegevens → Foutmelding

## User story 2: Abonnementen kiezen en kopen.

Als een gebruiker met een account, kan ik een abonnement kiezen van 6 maanden, 3 maanden, 1 maand of een 10-beurtenkaart. Zodat ik gebruik kan maken van de treinen met mijn abonnement. Ik kan maar 1 abonnement hebben per regio. Maar meerdere abonnementen verspreid over de regio's.

### Wireframe

#### Aankoop pagina:

**Titel:** "Kies uw Abonnement"

**Request Type:** POST

**URL:** URL/abonnement

**Request Body:** Een Abonnement object

#### Abonnement types:

- 6 maanden
- 3 maanden
- 1 maand
- 10-beurtenkaart

**Response Code:** 200

**Regio selectie:** Dropdown menu met beschikbare regio's

**Knop:** "Aankoop voltooien"

#### Bevestiging pagina:

**Titel:** "Uw abonnement is succesvol aangekocht!"

#### Details:

- Gekozen abonnementstype
- Gekozen regio
- Vervaldatum
- Prijs
- Aankoopdatum

**Knop:** "Ga naar profiel"

#### Acceptatiecriteria:

- Gebruiker kan kiezen tussen 4 formules(6 maanden ,3 maanden ,1 maand of een 10 beurtenkaart)
- Gebruiker kan een regio kiezen waar het abonnement geldig is.
- Gebruiker kan maar 1 abonnement per regio hebben.
- Gebruiker moet zijn abonnementen kunnen beheren op zijn account.
- Alle velden zijn verplicht

## User Story 3: Tickets kopen.

Als een gebruiker, kan ik als ik geen abonnement heb of in de regio's waar ik geen abonnement heb een ticket kopen met start en eindbestemming zodat ik voor mijn traject dat ik met de trein ga afleggen de trein kan nemen.

### Wireframe

#### **Ticket aankoop pagina:**

**Titel:** "Koop een ticket"

**Request Type:** POST

**URL:** URL/ticket

**Request Body:** Een Ticket object

#### **Ticket details en traject:**

- Begin station
- Bestemming eind station
- Prijs in euro

**Knop:** "Koop ticket"

**Response Code:** 200

#### **Bevestiging pagina**

**Titel:** "Uw ticket is succesvol aangekocht!"

#### **Details:**

- Type: Los ticket
- Geldigheidsdatum
- Begin station, locatie eindstation
- Prijs

**Knop:** "Ga naar profiel"

**Output succes:** Het gekochte ticket

#### Acceptatiecriteria:

- Gebruiker kan een los ticket aanschaffen voor te reizen door alle regio's.
- Het ticket word voor 24u bijgehouden op het account van de gebruiker.
- Na gebruik van het ticket word deze verwijderd van het account en gearchiveerd.

## User Story 4: Bestelling en betalingen.

Als een gebruiker,  
kan ik mijn bestellingen en abonnementen/tickets inzien op een apart tabblad.  
Zodat de conducteur mijn abonnement en of ticket/10beurten kaart kan scannen met  
een QR code. En dat ik kan zien wanneer deze aankopen geplaatst zijn. Scannen moet  
niet functioneel zijn maar kan een uitbereiding zijn.

### Wireframe

#### Order History Page:

**Titel:** "Mijn aankopen"

**Request Type:** GET

**URL:** URL/bestellingen

#### Request Parameters:

- userID

#### Bestelling lijst:

- Tabel met rijen voor elke bestelling, met kolommen voor:
  - Bestelling type (Abonnement of Ticket)
  - ID
  - Datum van aankoop
  - Prijs
  - Vervaldatum
  - Indien 10Bk hoeveel beurten er over zijn.
  - QR code die ID opgeeft.

#### Gearchiveerde bestellingen:

- Sectie onderaan voor gearchiveerde bestellingen ouder dan 1 jaar

**Response Code:** 200

**Output succes:** De aankopen

#### Acceptatiecriteria:

- Gebruiker kan een overzicht van zijn abonnementen/tickets/10-beurtenkaart zien op een aparte pagina.
- In het overzicht staan ook alle details van de bestelling erbij. Zoals aankoopdatum en tijd, prijs, geldigheidsdatum en hoeveel beurten er over zijn.
- Bestellingen die vervallen zijn worden automatisch gearchiveerd.

## User Story 5: Abonnementen archiveren.

Als een systeembeheerder kan ik abonnementen en tickets die vervallen zijn automatisch archiveren zodat de applicatie voor de gebruikers overzichtelijk blijft en dus alleen actieve abonnementen en tickets toont.

### Wireframe

#### Archiveer systeem:

**Titel:** "Archivering van vervallen abonnementen en tickets"

**Request Type:** PUT/POST

**Response Code:** 200

#### Acceptatiecriteria:

- Abonnementen en tickets die vervallen zijn worden automatisch naar het archief verplaatst.

#### Gearchiveerde data:

- Zichtbaar in het gebruikersprofiel, maar ook toegankelijk voor administratieve doeleinden.

## User Story 6: Kortingen

Als een gebruiker kan ik bij aankoop van een abonnement/ticket een kortingscode invullen indien ik deze heb zodat ik minder moet betalen voor mijn aankoop. En dus goedkoper kan reizen met de trein. Dit word gewoon toegevoegd bij het aankopen van producten.

### Wireframe

#### Korting systeem:

**Request Type:** POST

**Request Body:** Een Abonnement of Ticket object

#### Velden:

- Code
- IsActive
- DiscountAmount

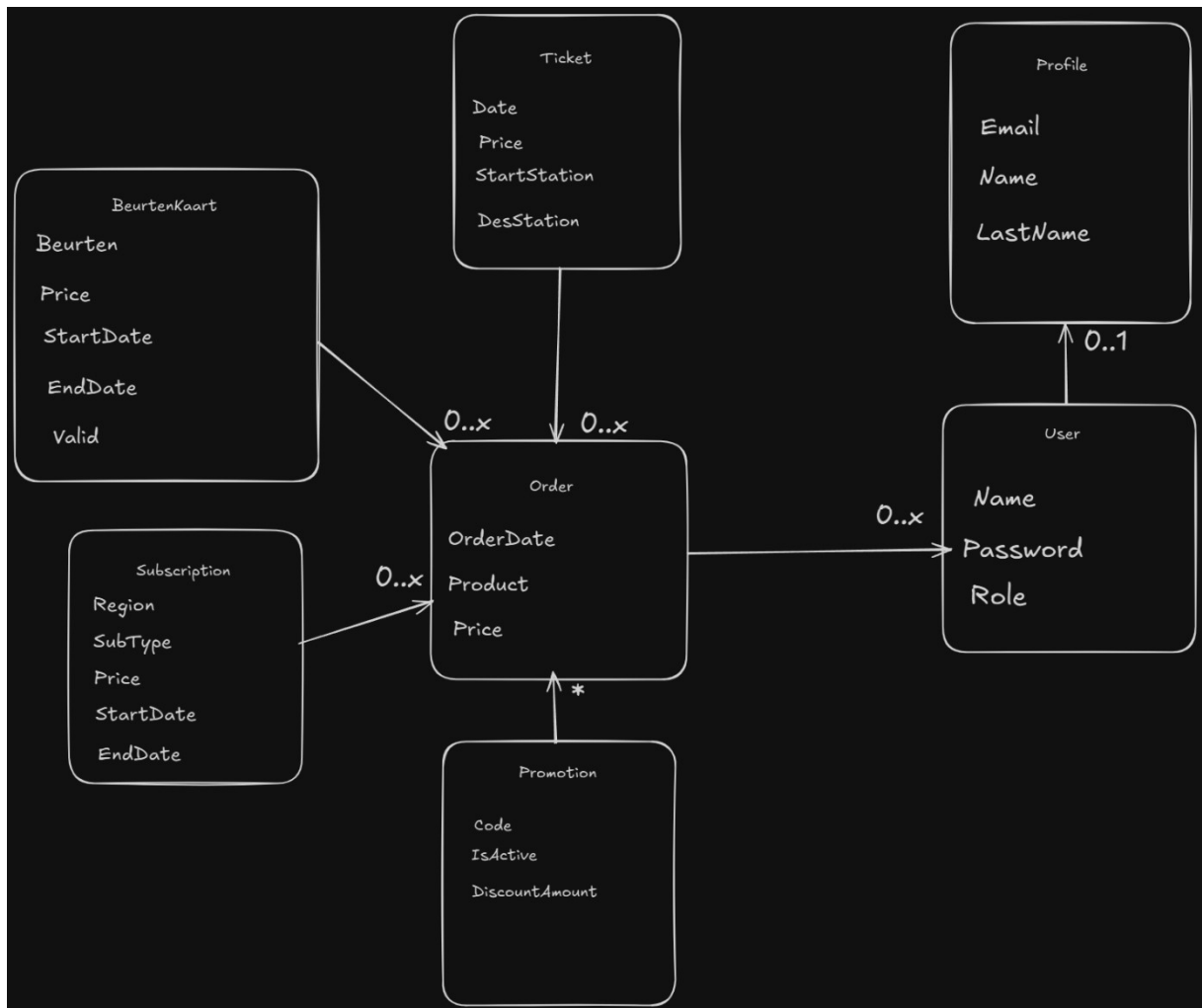
**Response Status:** 200

**Output success:** De afgetrokken korting.

#### Acceptatiecriteria:

- Code moet geldig zijn
- Velden zijn niet verplicht

## Domain model:





# Conceptueel & Logisch Database Model:

