

Software Design Document

Festival travel system (Web development 2024-2025)

Auteur: Jesper Meuzelaar

Reviewer: Tim van Dijk

Table of Contents

1.0	<i>Introductie</i>	3
1.1	Doel van het SDD	3
1.2	Scope	3
1.3	Overzicht	4
1.4	Analyse	5
2.0	<i>User interfaces</i>	6
3.0	<i>Technisch ontwerp</i>	7
3.1	UML diagrammen	9
4.0	<i>Requirement analyse/functionaliteiten</i>	11
4.1	<i>Doel:</i>	11
4.2	Functionele eisen:	11
4.3	Niet functionele eisen:	11
5.0	<i>Functioneel ontwerp</i>	12
6.0	<i>Proces</i>	13

1.0 Introductie

1.1 Doel van het SDD

In dit Software-design-document zal ik beschrijven hoe mijn, Travel system Applicatie in elkaar zit. Zodat de client het ontwerp begrijpt en er ook meer uitleg kan worden gegeven aan bepaalde keuzes. Ook Zal ik in dit SDD-inzicht geven in het technische ontwerp van de Applicatie.

1.2 Scope

De Travel System-applicatie biedt een breed scala aan functies om zowel klanten als beheerders optimaal te ondersteunen. Klanten kunnen zich registreren, hun gegevens beheren en busreizen naar festivals boeken via een overzichtelijk platform dat ook inzicht geeft in beschikbare reis- en festivaldata.

Het systeem bevat ook een loyaliteitspuntenprogramma waarmee klanten bij elke reservering punten kunnen verzamelen. Deze punten kunnen worden ingewisseld voor voordelen zoals kortingen en VIP-toegang, waardoor de klantloyaliteit wordt bevorderd.

Beheerders hebben uitgebreide mogelijkheden voor het beheren van festivalgegevens, waaronder het toevoegen, bewerken, bekijken en verwijderen van festivals. Klantbeheer geeft inzicht in klantgegevens en reisgeschiedenis.

Planners kunnen ook profiteren van automatische triggers die inschrijvingen analyseren en busreizen plannen als er meer dan 35 inschrijvingen zijn. Loyaliteitspuntenprogramma's zijn ook geïntegreerd in het besluitvormingsproces, en beschikbare punten kunnen de bus planning en het gebruik beïnvloeden.

De applicatie zal worden ontwikkeld met een modulaire aanpak die flexibiliteit en uitbreidbaarheid garandeert. Door de functionaliteit op te delen in afzonderlijke modules kan elk onderdeel afzonderlijk worden verbeterd en uitgebreid. Het databaseontwerp ondersteunt deze modules met een schema dat zowel klantgerichte als administratieve taken stroomlijnt zonder de prestaties of gegevensintegriteit in gevaar te brengen.

1.3 Overzicht

Dit document zal een gedetailleerde weergave van het ontwerp op het technische vlak en ook op het visuele vlak weergeven, en beschrijft meteen de meest belangrijke aspecten aan deze applicatie. Het doel van het SDD is om alles zo duidelijk en overzichtelijk mogelijk te beschrijven.

- Onderdeel 2 – Software overview, een duidelijke descriptie van de functionaliteiten en het ontwerp van het project.
- Onderdeel 3 – User interfaces, uitleg over bepaalde design keuzes die zijn gemaakt tijdens het ontwerpen van de applicatie.
- Onderdeel 4 - Technisch ontwerp, een duidelijke uitleg van hoe de applicatie aan de backend werkt, hier zal ook de brug worden gevormd tussen de functionele eisen en de implementatie
- Onderdeel 5 – Functionaliteiten, een duidelijke lijst met functionele en niet functionele eisen.
- Onderdeel 6 – Proces van de ontwikkeling.
- Onderdeel 7 – Review van de SDD en de Code

1.4 Analyse

1.4.1 Projectdoelen

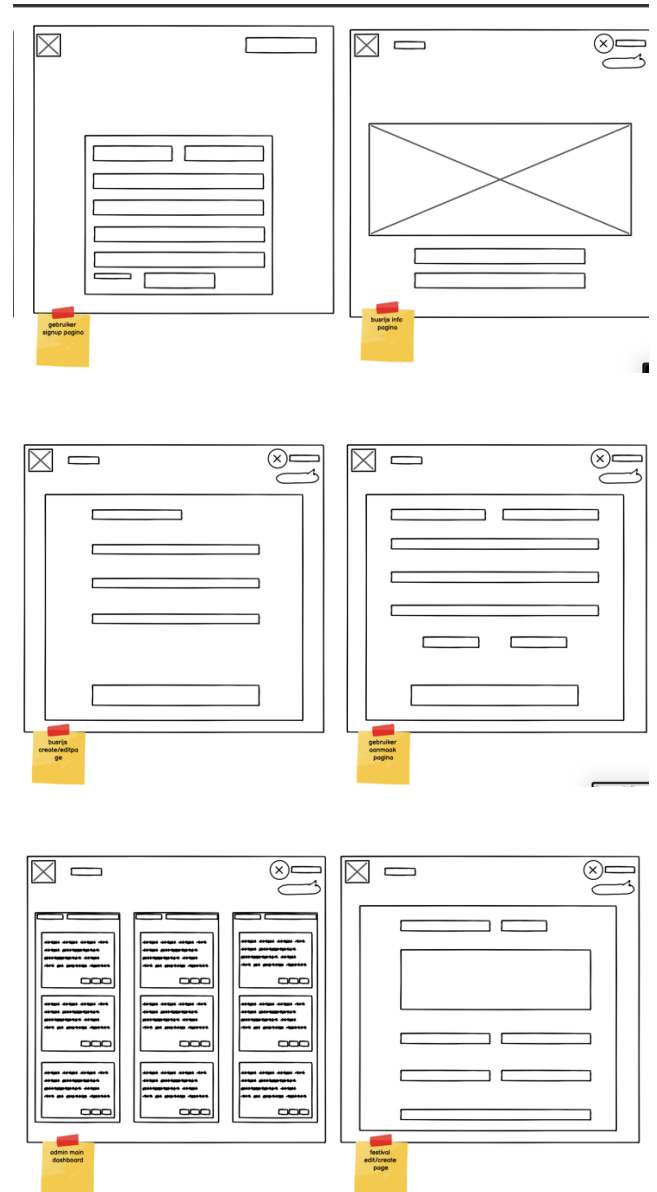
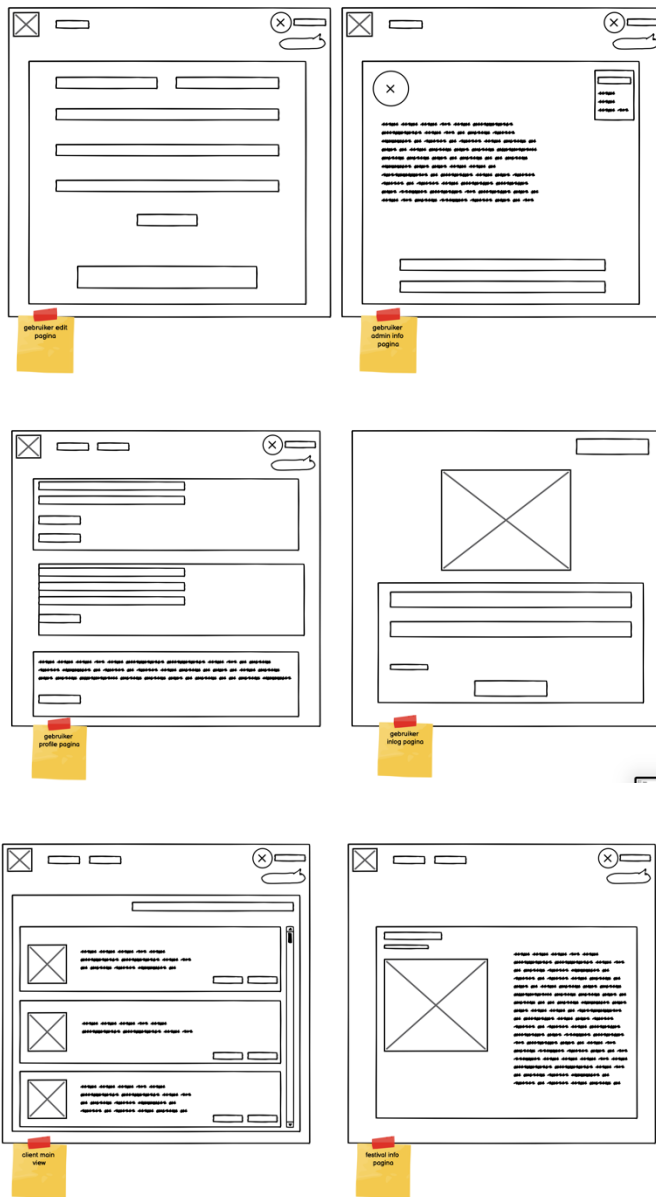
Het Festival Travel System heeft als doel om een gebruiksvriendelijk platform te ontwikkelen dat:

- De reiservaring van festivalgangers optimaliseert.
- Efficiënt klantbeheer en reisplanning mogelijk maakt.
- Realtime updates en een loyaliteitssysteem integreert.
- Dynamische bus planning implementeert op basis van vraag.

1.4.2. Stakeholders

- Busbedrijf (klant): Verantwoordelijk voor de uitvoering van reizen en planning. Het system moet hen helpen resources te beheren.
- Festivalgangers (eindgebruikers): Gebruiken het systeem om reizen te boeken en punten te sparen.
- Ontwikkelaar: Werkt aan het systeem en gebruikt dit project als leermiddel.

2.0 User interfaces



3.0 Technisch ontwerp

Technisch Ontwerp

Architectuur:

- **Frontend:** Een webgebaseerde gebruikersinterface gebouwd met HTML, CSS en JavaScript, ondersteund door Blade.
- **Backend:** Een PHP-framework Laravel verwerkt alle server-side logica en interactie met de database.
- **Database:** Een relationele database (bijvoorbeeld MySQL) met tabellen voor klanten, festivals, reizen en reis geschiedenis

Modules:

1 Klantenbeheer:

- Functionaliteiten: CRUD voor klantgegevens en authenticatie.
- Implementatie: Formulieren en laravel controllers en models voor interactie met de database.

2 Festivalbeheer:

- Functionaliteiten: CRUD-operaties voor festivalgegevens.
- Implementatie: Dashboard voor planners met aanroepbare functies.

3 Busplanning:

- Functionaliteiten: CRUD – operaties voor de gegevens en het linken aan specifieke Festivals
- Implementatie: Een dashboard voor de planners, met aanroepbare functies.

Technologieën:

- **Programmeertaal:** PHP voor backend en (blade/HTML)+Tailwind voor frontend.
- **Framework:** Laravel voor backendontwikkeling.
- **Database:** MySQL met tabellen voor gebruikers, reizen, festivals en reisgeschiedenis.
- **Google maps :** google maps api voor het weergeven van een route voor de festival beheerders

Databasemodel:

- **Tabellen:**

Entiteiten en Relaties

1. USERS Tabel (Gebruikers)

- **Doel:** Opslaan van gebruikersinformatie en authenticatiegegevens.
- **Primaire Sleutel:** id
- **Belangrijke Velden:**
 - first_name, last_name, username: Naam en gebruikersgegevens.
 - email
 - password: Wachtwoord (gehashed).
 - points: Puntensysteem voor karting bij transacties.
 - admin: Boolean die aangeeft of de gebruiker een admin is.
 - profile_picture: pad naar de profielfoto.
- **Relaties:**
 - Een gebruiker kan meerdere trips hebben (USER_TRIPS).

2. FESTIVALS Tabel (Festivals)

- **Doel:** Opslaan van festivalinformatie.
- **Primaire Sleutel:** id
- **Belangrijke Velden:**
 - name, location, description: Details over het festival.
 - date: Datum van het festival.
 - price: Prijs van tickets.
 - tickets: Aantal beschikbare tickets.
 - status: Status van het festival (bijv. active, sold).
 - image: URL of pad naar de afbeelding van het festival.
- **Relaties:**
 - Een festival kan meerdere reizen hebben (BUSREIZEN).
 - Een festival kan door meerdere gebruikers worden bezocht (USER_TRIPS).

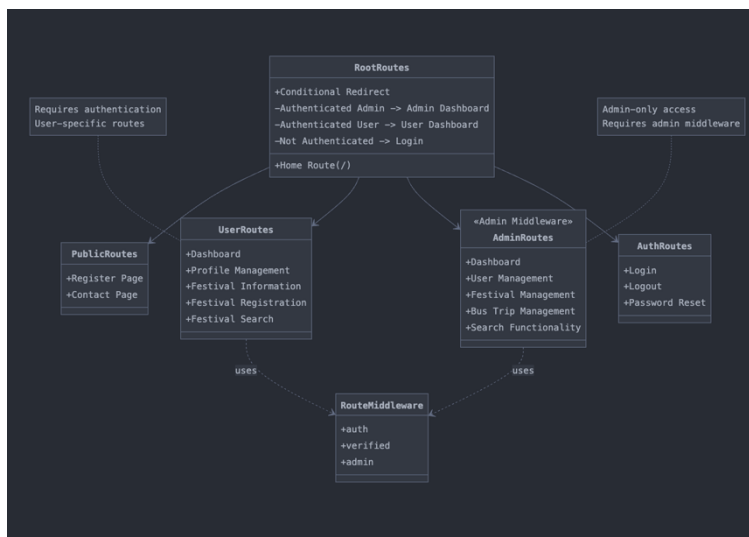
3. TRIPS Tabel (Gebruiker-Trips)

- **Doel:** Koppelt gebruikers aan festivals die zij bezoeken.
- **Primaire Sleutel:** id
- **Belangrijke Velden:**
 - user_id: Verwijzing naar de gebruiker (FK - USERS).
 - festival_id: Verwijzing naar het festival (FK - FESTIVALS).
 - Departure vertrek gekozen door gebruiker.
- **Relaties:**
 - Behoort tot een gebruiker.
 - Behoort tot een festival.

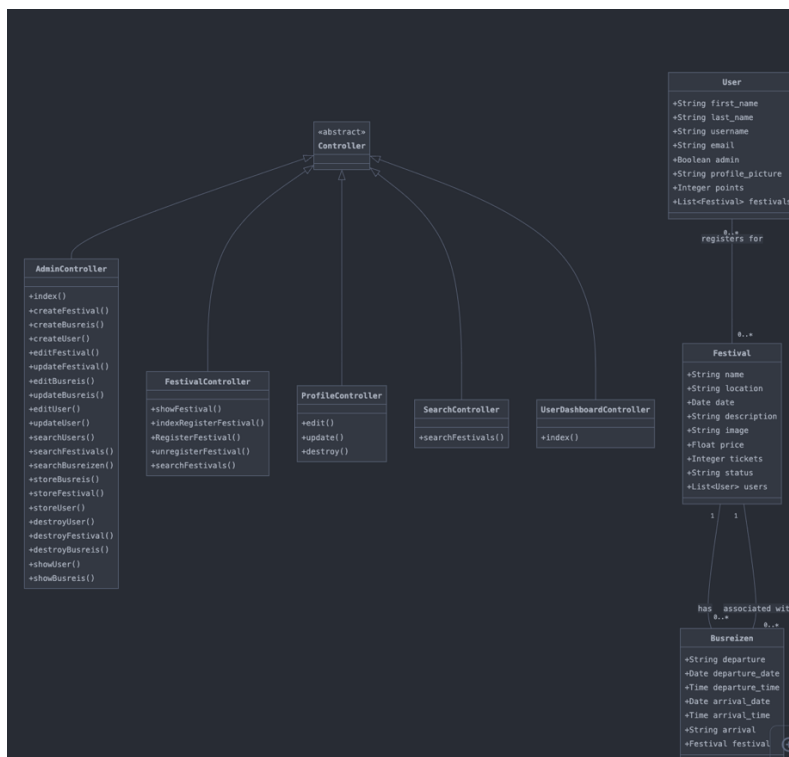
4. BUSREIZEN Tabel (Busreizen)

- **Doel:** Opslaan van informatie over busreizen naar festivals.
- **Primaire Sleutel:** id
- **Belangrijke Velden:**
 - departure, arrival: Vertrek- en aankomstlocatie.
 - departure_date, arrival_date: Data van vertrek en aankomst.
 - departure_time, arrival_time: Tijden van vertrek en aankomst.
 - festival_id: Verwijzing naar het bijbehorende festival (FK - FESTIVALS).
- **Relaties:**
 - Behoort tot één festival.

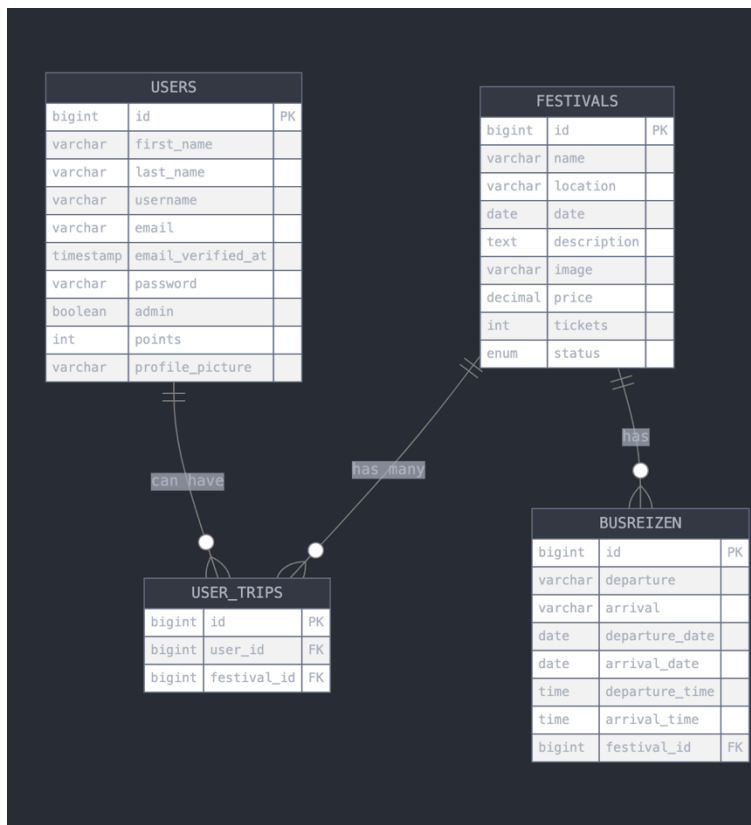
3.1 UML diagrammen



3.1.1 -Web.php



3.1.2 – Controllers



3.1.3 – ERD

4.0 Requirement analyse/functionaliteiten

4.1 Doel:

- Het optimaliseren van de reiservaring voor festivalgangers door een Gebruiksvriendelijk en efficiënt systeem te creëren.
- Het bieden van functionaliteiten voor het beheren van klanten, het plannen van reizen,
- En het uitvoeren van dynamische bus planning.

4.2 Functionele eisen:

- 1 Gebruikers moeten zich kunnen registreren en inloggen.
- 2 Klanten moeten beschikbare busreizen kunnen bekijken en boeken.
- 3 Het systeem moet loyaliteitspunten toekennen bij boekingen en klanten in staat Stellen deze in te wisselen.
- 4 Planners moeten festivals kunnen beheren (toevoegen, bewerken, verwijderen).

4.3 Niet functionele eisen:

- 1 Een schaalbare relationele database voor klantgegevens, festivalinformatie en Reisbeheer.
- 2 Realtime verwerking van boekingen en loyaliteitspunten.
- 3 Een intuïtieve webinterface die toegankelijk is via desktop.
- 4 Een snelle en goed ontworpen backend
- 5 Light house score van 90

5.0 Functioneel ontwerp

Overzicht van functionaliteiten:

1 Klantenbeheer:

- Registratie en inloggen van gebruikers.
- Inzicht in reisgeschiedenis en gespaarde loyaliteitspunten.

2 Reisplanning en boekingen:

- Weergave van beschikbare busreizen en bijbehorende festivalopties.
- Mogelijkheid voor klanten om reizen te boeken.

3 Loyaliteitspuntenprogramma:

- Automatische toekenning van punten bij elke boeking.
- Inwisselen van punten voor voordelen zoals kortingen.

4 Festivalbeheer:

- CRUD-functionaliteiten voor het toevoegen, bewerken, bekijken en verwijderen Van festivals.

Gebruiksscenario's:

Een klant boekt een festival en ontvangt punten.

Een planner voegt een nieuw festival toe aan het systeem.

Een festival wordt gelinkt aan een busreis.

6.0 Proces

1. SDD ontwerpen
2. Scope bepalen
3. Functionaliteiten beslissen
4. Mockups/wireframes maken
5. Analyse maken van Het project
6. Technisch ontwerp schrijven samen met de Software overview
7. GitHub repo aanmaken
8. Beginnen aan de code schrijven
9. Versie 1.0 afmaken
10. Code refactoreren
11. Versie 2.0