Tom Gevers S1210334

SSD FTS

Algemeen overzicht

Scope:

Wat ik wel ga doen:

* Gebruikers kunnen registreren en aanmelden
* Gebruikers kunnen beschikbare festivals en bussen bekijken
* Gebruikers kunnen bussen bestellen
* Gebruikers kunnen bestelde bussen terug zien op profiel
* Gebruikers krijgen punten bij het bestellen van punten
* Gebruikers kunnen de punten gebruiken voor kortingen
* Admins kunnen festivals en bussen aanmaken

Wat ga ik niet doen:

* Gebruikers kunnen niet van taal switchen. De webapp blijft in Engels.
* Geen ingewikkelde betaalsysteem

Belangrijke stakeholders:

Festibus: is de opdrachtgever, oftewel financiëren het project en bepalen de eisen.

Festivalgangers: zijn de gebruikers en zij willen een makkelijk te gebruiken en betrouwbare manier hebben om bussen voor festivals te bestellen.

Functionele vereisten:

Klantenbeheer:

• Registratie, beheer en inzicht in klantgegevens en

reisgeschiedenis.

Reisplanning en Boekingen:

• Boeken van busreizen naar festivals.

• Weergave van beschikbare reizen en festivaldata.

Puntensysteem:

• Opbouwen van punten bij elke boeking.

• Inwisselen van punten voor voordelen zoals kortingen of VIP-toegang.

Niet functionele vereisten:

1. Gebruiksvriendelijkheid

* De applicatie moet intuitief en makkelijk te gebruiken zijn voor zowel festivalgangers als beheerders.
* UI moet responsief zijn voor mobiele apparaten, tablets en desktops.

2. Prestatie (Performance)

* Pagina's moeten binnen 2 seconden laden onder normale omstandigheden.
* De busplanning moet automatisch reageren op inschrijvingen binnen enkele seconden.

3. Schaalbaarheid

* Het systeem moet meer gebruikers en reizen aankunnen naarmate het groeit (bijv. meerdere festivals tegelijk).
* Modulaire ontwikkeling ondersteunt toekomstige uitbreiding.

4. Beveiliging

* Klantgegevens moeten veilig opgeslagen worden (bijv. versleutelde wachtwoorden).
* Alleen geautoriseerde gebruikers mogen toegang hebben tot beheermodules (denk aan authenticatie/rolbeheer).

5. Betrouwbaarheid

* Het systeem moet stabiel draaien zonder regelmatig te crashen.
* Back-ups moeten beschikbaar zijn in geval van fouten of dataverlies.

6. Onderhoudbaarheid

* De code moet goed gedocumenteerd en gestructureerd zijn zodat het systeem makkelijk aangepast of uitgebreid kan worden.
* Gebruik van een framework zoals Laravel zorgt voor onderhoudsvriendelijkheid.

7. Testbaarheid

* Het systeem moet eenvoudig te testen zijn (unit tests, integratietests).
* Testscenario's moeten eenvoudig te reproduceren zijn.

8. Beschikbaarheid

* De applicatie moet minimaal 99% beschikbaar zijn tijdens piekmomenten (zoals vlak voor grote festivals).

Interface Ontwerp:

Activity diagrams:

A screenshot of a chat

AI-generated content may be incorrect.

Wireframe:

A screenshot of a website

AI-generated content may be incorrect.

A screenshot of a website

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Technisch Ontwerp:

Programmeertalen: PHP 8.3.11, JavaScript, HTML5, CSS3

Frameworks: Laravel 11

Database: MySQL

OS: Windows 11

Code-Editor: Visual Studio Code

Versiebeheer: Git (Github)

Architectuurontwerp:

* Model
* View
* Controller

Beveiliging: Laravel Breeze

Teststrategie: PHPUnit

Sequence Diagram:

Trip boeken:

A diagram of a project

AI-generated content may be incorrect.

User Stories en Sprint Backlog

Trello boord:  
<https://trello.com/invite/b/686a87bd7e200553702264aa/ATTIab474850f9ab27f52ca9c0240960ceaa581EE693/wdp-extra>

Github  
<https://github.com/Tom-wdh/fts>