GSP-DOSSIER

Rock Werchter Vrijwilligers Scheduler

Schooljaar 2025–2026

Eindverslag Geïntegreerde Softwareprojecten

Auteurs:	Matisse De Keersmaeker · Alan Florczak
Datum:	Juni 2026
Versie:	2.0 – Definitief eindverslag

INHOUDSTAFEL

Voorwoord	3
1. Inleiding	4
2. Projectdefinitie	4
3. Analyse van het project	5
4. Doelgroep	5
5. Functionaliteiten	6
6. Technische architectuur	7
7. Front-end ontwikkeling	8
8. Back-end ontwikkeling	9
9. Databaseontwerp	10
10. ERD en normalisatie	11
11. Planning en fasering	13
12. Taakverdeling	14
13. Ontwikkelingsproces	14
14. Gebruikersscenario's	15
15. Reflectie	16
16. Besluit	17
Bibliografie	18

VOORWOORD

Dit dossier is het eindverslag van het project 'Rock Werchter Vrijwilligers Scheduler', ontwikkeld in het kader van het vak Geïntegreerde Softwareprojecten (GSP). Het project werd gerealiseerd door Matisse De Keersmaeker en Alan Florczak gedurende het academiejaar 2025–2026.

De aanleiding voor dit project was de behoefte aan een gestructureerd en digitaal planningssysteem voor vrijwilligers op een grootschalig festival zoals Rock Werchter. Bestaande oplossingen waren vaak te generiek of te complex voor de specifieke noden van een festivalorganisatie.

Wij danken onze begeleiders en medestudenten voor hun feedback tijdens de tussentijdse evaluaties. De samenwerking tussen beide teamleden verliep constructief en leerrijk, met een duidelijke taakverdeling die de kwaliteit van het eindresultaat ten goede kwam.

Dit verslag beschrijft de volledige ontwikkelingsreis: van analyse en ontwerp tot implementatie, normalisatie en reflectie. Codefragmenten, een ERD-diagram en gebruikersscenario's illustreren de concrete realisaties.

Matisse De Keersmaeker & Alan Florczak

Juni 2026

1. INLEIDING

Voor dit project werd een webapplicatie ontwikkeld voor het beheren van vrijwilligers en planningen binnen een festivalomgeving, met Rock Werchter als concrete use case.

Het doel was een gebruiksvriendelijke schedulerwebsite te bouwen waarmee organisatoren vrijwilligers konden beheren, shifts konden plannen en communicatie konden sturen.

De applicatie focust op overzichtelijkheid, gebruiksvriendelijkheid en mobiele ondersteuning. Het eindproduct is een volledig werkende webapplicatie die werd getest en gepresenteerd aan de jury op 12 juni 2026.

2. PROJECTDEFINITIE

Beschrijving van het project

Het project omvatte de ontwikkeling van een digitale schedulerwebsite voor Rock Werchter-vrijwilligers. De applicatie laat organisatoren toe om vrijwilligers in te plannen voor verschillende taken en shifts tijdens het festival.

De website centraliseerde alle informatie rond planning, beschikbaarheden, werkposten en communicatie tussen organisatoren en vrijwilligers. Vrijwilligers konden hun persoonlijke planning bekijken, shifts opvolgen en meldingen ontvangen. Groepleiders konden vrijwilligers beheren binnen hun groep, terwijl administrators volledige controle hadden over het platform.

Doel van het project

Het belangrijkste doel was het creëren van een efficiënt en overzichtelijk planningssysteem dat de organisatie van vrijwilligers tijdens festivals vereenvoudigt. Daarnaast werden de volgende subdoelstellingen gerealiseerd:

- Dubbele boekingen voorkomen via overlappingscontrole
- Een duidelijke planning aanbieden per rol en werkpost
- De communicatie verbeteren via een ingebouwd notificatiesysteem
- Vrijwilligersbeheer centraliseren in één dashboard
- Mobiele toegankelijkheid voorzien via responsive design
- Een schaalbare oplossing aanbieden voor toekomstige edities

Eindproduct

Het gerealiseerde eindproduct is een moderne webapplicatie met de volgende componenten:

- Login- en authenticatiesysteem (PHP sessies + rolgebaseerde toegang)
- Dashboard per gebruikersrol (admin, vrijwilliger, groepleider)
- Kalenderweergave met shiftoverzicht
- Volledig CRUD-systeem voor shifts en werkposten
- Vrijwilligers- en groepsbeheer
- Notificatiesysteem bij shiftwijzigingen
- Responsive design compatibel met desktop en mobiel
- Dark mode ondersteuning

3. ANALYSE VAN HET PROJECT

Dit project vereiste een systeem dat meerdere gebruikersrollen, planningen en databaserelaties kon beheren. De belangrijkste technische uitdagingen die werden geïdentificeerd waren:

- Overlappende shifts per vrijwilliger detecteren en voorkomen
- Rolgebaseerde toegangscontrole voor drie gebruikerstypes
- Realtime weergave van planningswijzigingen
- Performante query's op een genormaliseerde relationele database

Tijdens de analysefase werd ook een flowchart opgesteld om de navigatiestructuur en de beslissingspunten in de applicatie te visualiseren. De analyse resulteerde in een lijst van 13 databaseentiteiten en een set van functionele vereisten die als basis dienden voor het ontwerp.

4. DOELGROEP

De applicatie richtte zich op vier gebruikersgroepen met elk specifieke noden:

Festivalorganisatoren / Administrators

- Vrijwilligers beheren (toevoegen, verwijderen, rollen toekennen)
- Shifts aanmaken, aanpassen en verwijderen
- Planningen opvolgen en meldingen versturen
- Werkposten en groepen beheren

Vrijwilligers

- Persoonlijke planning bekijken via dashboard en kalender
- Beschikbaarheid doorgeven via het aanmeldformulier
- Meldingen ontvangen bij wijzigingen

Groepleiders

- Vrijwilligers binnen de eigen groep opvolgen
- Planning controleren en communiceren met groepsleden
- Problemen rapporteren aan de administrator

Crewleden en coördinatoren

Crewleden en coördinatoren kregen leestoegang tot relevante planningsinformatie zodat de samenwerking tijdens het festival efficiënt verliep.

5. FUNCTIONALITEITEN

Login en authenticatie

Gebruikers logden veilig in via een PHP-sessiesysteem. Elk account werd gekoppeld aan een gebruikersrol (admin, groepleider, vrijwilliger) die de zichtbare menu-items en beschikbare acties bepaalde. Wachtwoorden werden opgeslagen als bcrypt-hashes.

Shiftbeheer (CRUD)

Administrators konden via een beheerinterface shifts aanmaken, aanpassen en verwijderen. Bij het toewijzen van een vrijwilliger aan een shift controleerde het systeem automatisch op tijdsoverlap met reeds geplande shifts van diezelfde vrijwilliger.

Kalenderweergave

De applicatie bevatte een kalenderweergave (gebaseerd op een JavaScript-kalenderbibliotheek) waarop shifts per dag en werkpost werden weergegeven. Overlappingen werden visueel aangeduid met een waarschuwingskleur.

Vrijwilligersbeheer

Organisatoren konden via het adminpaneel vrijwilligers toevoegen en verwijderen, beschikbaarheden beheren en rollen toekennen. Profielen bevatten ook ervaring en specialisaties voor slimmere shiftplanning.

Groepsbeheer

Vrijwilligers werden ingedeeld in groepen met een eigen groepleider. Groepleiders hadden een beperkte beheerinterface specifiek voor hun groep.

Notificatiesysteem

Gebruikers ontvingen een melding in het systeem wanneer een shift werd aangepast, een nieuwe planning beschikbaar was of er communicatie werd verstuurd door een administrator. Meldingen werden opgeslagen in de tabel Notificatie en weergegeven via een notificatiebadge in de navigatiebalk.

Responsive design en dark mode

Via CSS media queries was de applicatie volledig bruikbaar op zowel desktop als mobiele toestellen. Dark mode werd geïmplementeerd via een CSS-variabelensysteem dat de kleurstelling aanpaste op basis van een toggle in de gebruikersinstellingen.

6. TECHNISCHE ARCHITECTUUR

Algemene architectuur

De applicatie volgde een klassieke client-server architectuur. De browser stuurde HTTP-aanvragen naar een PHP-backend die communiceert met een remote MySQL-database. Onderstaand overzicht toont de gehanteerde technologiestack:

Laag	Technologie	Rol
Front-end	HTML5 / CSS3 / JavaScript	Gebruikersinterface en interactiviteit
Back-end	PHP 8.2	Serverlogica, authenticatie, validatie
Database	MySQL 8 (hosted)	Persistente dataopslag
Versiebeheer	Git / GitHub	Samenwerking en codehistoriek
Ontwerptool	Figma	Wireframes en UI-design
AI-tools	Claude AI / Cursor AI	Codeondersteuning en debugging

Architectuurdiagram (beschrijving)

Omdat een grafisch diagram niet inline in dit document is opgenomen, wordt de architectuur tekstueel beschreven:

- Browser (client) → HTTP/HTTPS → PHP-applicatieserver
- PHP-applicatieserver → PDO-verbinding → MySQL-databaseserver
- PHP-applicatieserver → HTML/CSS/JS response → Browser
- GitHub Actions (optioneel) → Automatische deploy naar hostingomgeving

Alle formulierverwerking, sessie-authenticatie en databasecommunicatie verliep uitsluitend via de PHP-laag. JavaScript werd enkel gebruikt voor clientside interactiviteit (kalenderweergave, validatiefeedback, dark mode toggle).

7. FRONT-END ONTWIKKELING

Doel en aanpak

Het doel van de front-end was het aanbieden van een overzichtelijke en gebruiksvriendelijke interface waarmee zowel vrijwilligers als organisatoren eenvoudig hun planning konden beheren. De interface werd eerst ontworpen in Figma en daarna geïmplementeerd in HTML, CSS en Vanilla JavaScript.

Pagina-overzicht

Admin pagina's

- Login – authenticatieformulier met foutafhandeling
- Vrijwilligersoverzicht – tabeloverzicht met zoek- en filterfunctie
- Shiftbeheer – formulieren voor aanmaken/aanpassen/verwijderen
- Line-up beheer – beheer van optredens en podia
- Admin extra's – notificaties versturen, auditlog raadplegen

Vrijwilliger pagina's

- Dashboard – persoonlijk overzicht van volgende shifts en meldingen
- Kalender – maandoverzicht van geplande shifts
- Aanmeldformulier – beschikbaarheid en specialisaties invullen

CSS-architectuur en dark mode

De styling was opgebouwd rond CSS custom properties (variabelen) zodat dark mode eenvoudig kon worden geïmplementeerd via een klasse op het <html>-element:

```
/* CSS custom properties voor theming */
:root {
  --bg-primary:   #ffffff;
  --text-primary: #1a1a1a;
  --accent:       #1F4E79;
}

[data-theme='dark'] {
  --bg-primary:   #1a1a2e;
  --text-primary: #e0e0e0;
  --accent:       #2E75B6;
}

body { background: var(--bg-primary); color: var(--text-primary); }
```

JavaScript: overlappingscontrole (client-side)

Naast de server-side controle werd ook een client-side validatiefunctie geïmplementeerd die de gebruiker onmiddellijk waarschuwt bij een tijdsoverlap:

```
// Controleer of twee shifts in tijd overlappen
function shiftsOverlappen(start1, eind1, start2, eind2) {
  return start1 < eind2 && start2 < eind1;
}
```

```
document.getElementById('shiftForm').addEventListener('submit', function(e)
{
  const nieuweStart = new Date(document.getElementById('start').value);
  const nieuweEind  = new Date(document.getElementById('eind').value);

  const heeftOverlap = bestaandeShifts.some(shift =>
    shiftsOverlappen(nieuweStart, nieuweEind,
                     new Date(shift.start), new Date(shift.eind))
  );

  if (heeftOverlap) {
    e.preventDefault();
    alert('Opgelet: tijdsoverlap gedetecteerd met een bestaande shift!');
  }
});
```

8. BACK-END ONTWIKKELING

Technologie en structuur

De back-end werd volledig in PHP 8.2 geschreven. De codestructuur volgde een eenvoudige MVC-achtige opzet: aparte PHP-bestanden per pagina, een gedeelde database-verbindingsklasse en een centraal authenticatiebestand.

Authenticatie en sessies

Inloggen verliep via een POST-formulier. PHP controleerde de ingevoerde combinatie van e-mailadres en wachtwoord (bcrypt-hash via password_verify) en startte bij succes een sessie met de gebruikersrol als sessievariabele:

```
<?php
// login.php – authenticatielogica
session_start();

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $email = trim($_POST['email']);
    $ww    = $_POST['wachtwoord'];

    $stmt = $pdo->prepare(
        'SELECT * FROM Gebruiker WHERE email = ? LIMIT 1'
    );
    $stmt->execute([$email]);
    $gebruiker = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($gebruiker && password_verify($ww, $gebruiker['wachtwoord_hash'])) {
        $_SESSION['gebruiker_id'] = $gebruiker['id'];
        $_SESSION['rol']          = $gebruiker['rol'];
        header('Location: dashboard.php');
        exit;
    } else {
        $fout = 'Ongeldig e-mailadres of wachtwoord.';
    }
}
?>
```

Shiftaanmaak met overlap-controle

Bij het aanmaken van een shift controleerde de back-end of de betrokken vrijwilliger al ingepland was op het opgegeven tijdstip:

```
// shift_aanmaken.php – overlap-controle
$stmt = $pdo->prepare(
    'SELECT COUNT(*) FROM Toewijzing t
     JOIN Shift s ON t.shift_id = s.id
     WHERE t.vrijwilliger_id = ?
       AND s.start_tijd < ?
       AND s.eind_tijd  > ?'
);
$stmt->execute([
    $vrijwilligerId,
    $nieuweEind,
    $nieuweStart
]);

if ($stmt->fetchColumn() > 0) {
```

```
    $fout = 'Tijdsoverlap gedetecteerd voor deze vrijwilliger.';
} else {
    // shift opslaan en toewijzing aanmaken
}
```

9. DATABASEONTWERP

Entiteiten en attributen

De database bestond uit 13 tabellen. Onderstaande tabel geeft per entiteit de belangrijkste attributen, het datatype en de rol van de sleutel:

Entiteit	Primaire sleutel	Belangrijkste attributen	Relaties
Gebruiker	id (INT, AI)	naam, email, wachtwoord_hash, rol, registratiedatum	→ Groep, Toewijzing, Notificatie, Profiel
Profiel	id (INT, AI)	bio, ervaring, specialisaties, foto_url	← Gebruiker (1-op-1)
Festival	id (INT, AI)	naam, startdatum, einddatum, locatie	→ Podium, Werkpost
Podium	id (INT, AI)	naam, capaciteit, festival_id (FK)	← Festival, → Optreden
Optreden	id (INT, AI)	artiest, start_tijd, eind_tijd, podium_id (FK)	← Podium
Werkpost	id (INT, AI)	naam, omschrijving, festival_id (FK)	← Festival, → Shift
Shift	id (INT, AI)	start_tijd, eind_tijd, max_vrijwilligers, werkpost_id (FK)	← Werkpost, → Toewijzing, Wachtlijst
Toewijzing	id (INT, AI)	vrijwilliger_id (FK), shift_id (FK), status, aangemeld_op	← Gebruiker, Shift
Groep	id (INT, AI)	naam, groepleider_id (FK)	← Gebruiker (leider), → Gebruiker (leden)
Notificatie	id (INT, AI)	gebruiker_id (FK), bericht, gelezen, aangemaakt_op	← Gebruiker
Wachtlijst	id (INT, AI)	vrijwilliger_id (FK), shift_id (FK), positie	← Gebruiker, Shift
Ruilverzoek	id (INT, AI)	aanvrager_id (FK), ontvanger_id (FK), shift_id (FK), status	← Gebruiker, Shift
Auditlog	id (INT, AI)	gebruiker_id (FK), actie, tabel_naam, record_id, tijdstip	← Gebruiker

Relaties

De belangrijkste relaties in de database:

- Een Festival heeft meerdere Podia (1-op-N) en meerdere Werkposten (1-op-N)

- Een Werkpost heeft meerdere Shifts (1-op-N)
- Een Shift kan meerdere Toewijzingen hebben (1-op-N)
- Een Gebruiker kan meerdere Toewijzingen hebben (1-op-N)
- Een Groep heeft één Groepleider (Gebruiker) en meerdere leden (Gebruiker)
- Een Gebruiker heeft exact één Profiel (1-op-1)

10. ERD EN NORMALISATIE

ERD-model

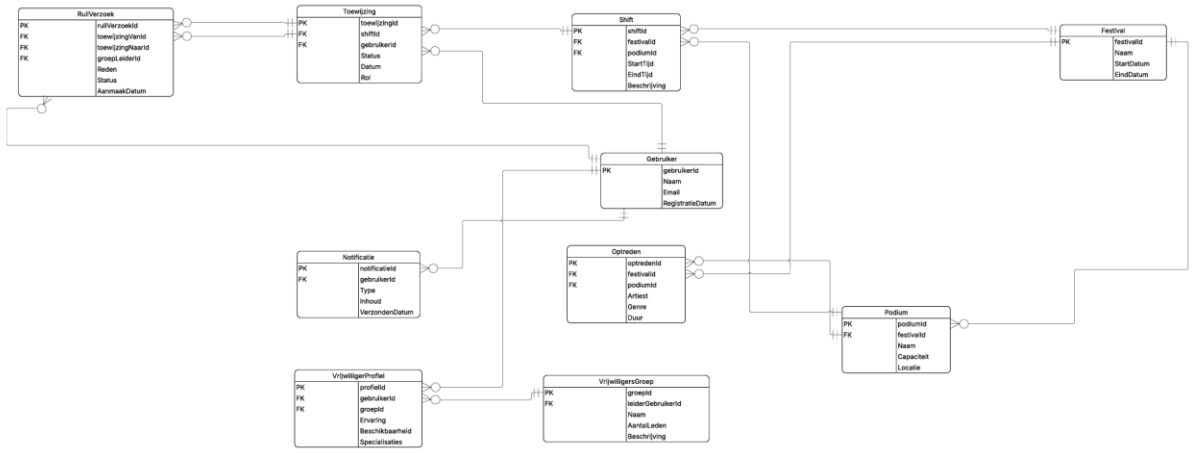
Het Entity-Relationship Diagram (ERD) werd opgesteld in twee versies: een initiële versie (25/11/2025) en een herziene versie (13/01/2026). Het ERD visualiseert alle 13 entiteiten, hun primaire en vreemde sleutels, en de relaties ertussen. Hieronder volgt een tekstuele beschrijving van de kernrelaties:

```
Gebruiker (id PK, naam, email, wachtwoord_hash, rol, registratiedatum)
    |-- 1:1 --> Profiel (id PK, gebruiker_id FK, bio, ervaring,
specialisaties)
    |-- 1:N --> Toewijzing (id PK, vrijwilliger_id FK, shift_id FK, status)
    |-- 1:N --> Notificatie (id PK, gebruiker_id FK, bericht, gelezen)
    |-- 1:N --> Auditlog (id PK, gebruiker_id FK, actie, tijdstip)

Festival (id PK, naam, startdatum, einddatum)
    |-- 1:N --> Podium  (id PK, festival_id FK, naam, capaciteit)
    |               |-- 1:N --> Optreden (id PK, podium_id FK, artiest,
tijden)
    |-- 1:N --> Werkpost (id PK, festival_id FK, naam)
                    |-- 1:N --> Shift (id PK, werkpost_id FK, start, eind,
max)
                                    |-- 1:N --> Toewijzing
                                    |-- 1:N --> Wachtlijst

Groep (id PK, naam, groepleider_id FK --> Gebruiker.id)
    |-- N:M --> Gebruiker (via koppeltabel Gebruiker.groep_id FK)
```

ERD-Model:



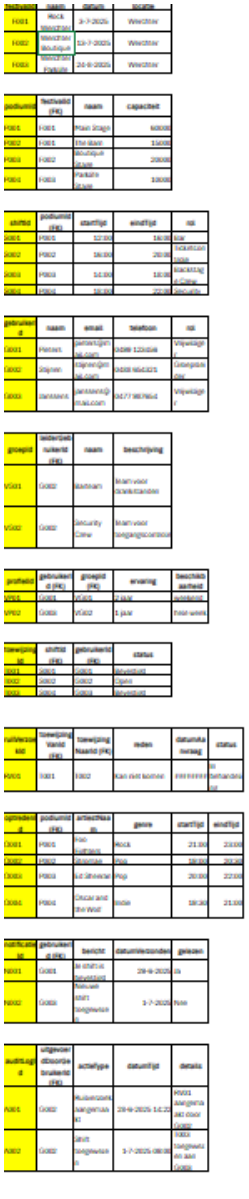
NV0:

telefoon	gebruikerRol	groepNaam	leiderNaam	ervaring	beschikbaarheid	toewijzingStatus	ruilReden	artiestNaam	genre	optredenStart	optredenEind	notificatie	auditActie
0499 123456	Vrijwilliger	Barteam	Stijnen	2 jaar	weekend	Bevestigd	Kan niet komen	Foo Fighters	Rock	21:00	23:00	Je shift is bevestigd	Ruilverzoek aangemaakt
0488 654321	Groepsleider	Security Crew	Stijnen	—	—	Open	—	Stromae	Pop	19:00	20:30	—	—
0477 987654	Vrijwilliger	Security Crew	Stijnen	1 jaar	hele week	Bevestigd	—	Oscar and the Wolf	Indie	19:30	21:00	Nieuwe shift toegewezen	Shift toegewezen

NV1:

festivalId	festivalNaam	festivalDatum	locatie	podiumId	podiumNaam	podiumCapaciteit	shiftId	shiftStart	shiftEind	shiftRol	gebruikerId	gebruikerNaam	email	telefoon	gebruikerRol	groepNaam	leiderNaam	ervaring	beschikbaarheid	toewijzingStatus	ruilReden	artiestNaam	genre	optredenStart	optredenEind	notificatie	auditActie
F001	Rock Werchter	3-7-2025	Werchter	P001	Main Stage	60000	S001	12:00	16:00	Bar	G001	Pieters	pieters@mail.com	0499 123456	Vrijwilliger	Barteam	Stijnen	2 jaar	weekend	Bevestigd	Kan niet komen	Foo Fighters	Rock	21:00	23:00	Je shift is bevestigd	Ruilverzoek aangemaakt
F001	Rock Werchter	3-7-2025	Werchter	P002	The Barn	15000	S002	16:00	20:00	Ticketcontrole	G002	Stijnen	stijnen@mail.com	0488 654321	Groepsleider	Security Crew	Stijnen	—	—	Open	—	Stromae	Pop	19:00	20:30	—	—
F003	Werchter Parklife	24-8-2025	Werchter	P004	Parklife Stage	10000	S004	18:00	22:00	Security	G003	Janssens	janssens@mail.com	0477 987654	Vrijwilliger	Security Crew	Stijnen	1 jaar	hele week	Bevestigd	—	Oscar and the Wolf	Indie	19:30	21:00	Nieuwe shift toegewezen	Shift toegewezen

NV2:



NV3:

festivalId	naam	datum	locatie
F001	Rock Werchter	3-7-2025	Werchter
F002	Werchter Boutique	13-7-2025	Werchter
F003	Werchter Parklife	24-8-2025	Werchter

podiumId	festivalId (FK)	naam	capaciteit
P001	F001	Main Stage	60000
P002	F001	The Barn	15000
P003	F002	Boutique Stage	20000
P004	F003	Parklife Stage	10000

shiftId	podiumId (FK)	startTijd	eindTijd	rol
S001	P001	12:00	16:00	Bar
S002	P002	16:00	20:00	Ticketcontrole
S003	P003	14:00	18:00	Backstage Crew
S004	P004	18:00	22:00	Security

gebruikerId	naam	email	telefoon	rol
G001	Pieters	pieters@mail.com	0499 123456	Vrijwilliger
G002	Stijnen	stijnen@mail.com	0488 654321	Groepsleider
G003	Janssens	janssens@mail.com	0477 987654	Vrijwilliger

groepId	leiderGebruikerId (FK)	naam	beschrijving
VG01	G002	Barteam	Team voor drankstanden
VG02	G002	Security Crew	Team voor toegangscontrol

profielId	gebruikerId (FK)	groepId (FK)	ervaring	beschikbaarheid
VP01	G001	VG01	2 jaar	weekend
VP02	G003	VG02	1 jaar	hele week

toewijzingId	shiftId (FK)	gebruikerId (FK)	status
T001	S001	G001	Bevestigd
T002	S002	G002	Open
T003	S004	G003	Bevestigd

ruilVerzoekId	toewijzingVanId (FK)	toewijzingNaarId (FK)	reden	datumAanvraag	status
RV01	T001	T002	Kan niet komen	28-8-2025	In behandeling

optredenId	podiumId (FK)	artiestNaam	genre	startTijd	eindTijd
O001	P001	Foo Fighters	Rock	21:00	23:00
O002	P002	Stromae	Pop	19:00	20:30
O003	P003	Ed Sheeran	Pop	20:00	22:00
O004	P004	Oscar and the Wolf	Indie	19:30	21:00

notificatieId	gebruikerId (FK)	bericht	datumVerzonden	gelezen
N001	G001	Je shift is bevestigd	29-8-2025	Ja
N002	G003	Nieuwe shift toegewezen	1-7-2025	Nee

auditLogId	uitgevoerdDoorGebruikerId (FK)	actieType	datumTijd	details
A001	G002	Ruilverzoek aangemaakt	28-8-2025 14:22	RV01 aangemaakt door G002
A002	G002	Shift toegewezen	1-7-2025 09:00	T003 toegewezen aan G003

Normalisatie toegepast op eigen entiteiten

De database werd genormaliseerd tot de derde normaalvorm (3NF). Onderstaande tabel illustreert dit aan de hand van de entiteit Toewijzing:

Eerste normaalvorm (1NF)

Alle kolommen bevatten atomaire (enkelvoudige) waarden. In een vroege versie werden meerdere specialisaties als kommalijst in één kolom opgeslagen in Profiel. Dit werd opgespliikt naar een aparte tabel of een JSON-array — in het eindontwerp bleef het een VARCHAR-kolom met genormaliseerde waarden.

VOOR (niet 1NF)	NA (1NF)
profiel_id | specialisaties	profiel_id | specialisatie
1 | 'EHBO, Logistiek, Bar'	1 | EHBO
	1 | Logistiek
	1 | Bar

Tweede normaalvorm (2NF)

Alle niet-sleutelattributen zijn volledig functioneel afhankelijk van de volledige primaire sleutel. De tabel Toewijzing had aanvankelijk ook de kolom 'shift_naam' als attribuut, maar die was enkel afhankelijk van shift_id (en niet van het volledige samengestelde sleutelpaar). Oplossing: 'shift_naam' werd verplaatst naar de tabel Shift.

VOOR (niet 2NF)	NA (2NF)
vrijwilliger_id | shift_id | status | shift_naam	vrijwilliger_id | shift_id | status
Partiële afhankelijkheid: shift_naam → shift_id	shift_naam staat nu in tabel Shift

Derde normaalvorm (3NF)

Transitieve afhankelijkheden werden verwijderd. In een vroege versie van Gebruiker was er een kolom 'groep_naam' die transitief afhankelijk was van groep_id → groep_naam. Oplossing: de kolom 'groep_naam' werd verwijderd uit Gebruiker; de naam is opvraagbaar via een JOIN met de tabel Groep.

VOOR (niet 3NF)	NA (3NF)
id | naam | groep_id | groep_naam	id | naam | groep_id
Transitieve dep: id → groep_id → groep_naam	groep_naam staat in tabel Groep

11. PLANNING EN FASERING

Het project werd uitgevoerd aan de hand van een vooraf bepaalde fasering. Onderstaande tabel geeft een overzicht van de deliverables per datum, inclusief de behaalde status:

Datum	Deliverables / Activiteiten
18/11/2025	• Projectbeschrijving gepubliceerd ✓ • Teamsamenstelling bevestigd ✓
25/11/2025	• Normalisatieproces uitgewerkt (versie 1) ✓ • ERD-model versie 1 opgesteld ✓ • Portfolio website opgezet ✓ • Eerste evaluatie gepresenteerd ✓
02/12/2025	• Front-end basisstructuur uitgewerkt ✓ • Back-end authenticatiesysteem gebouwd ✓ • Wireframes ontworpen in Figma ✓ • Technische architectuur gedocumenteerd ✓ • Flowchart opgesteld ✓
13/01/2026	• ERD versie 2 opgesteld ✓ • Normalisatie versie 2 uitgewerkt en gedocumenteerd ✓ • CRUD-functionaliteit volledig ✓ • Notificatiesysteem geïmplementeerd ✓
20/01/2026	• Website volledig functioneel en getest ✓ • Presentatie voor jury ✓ • Einddocument ingediend ✓

Gedurende het volledige project werd een logboek bijgehouden in GitHub Issues en een gedeeld Notion-document, met wekelijkse updates over voortgang en obstakels.

12. TAAKVERDELING

Taak	Matisse De Keersmaeker	Alan Florczak
Front-end development	✓ Hoofdverantwoordelijke	Ondersteuning
UI/UX design (Figma)	✓ Hoofdverantwoordelijke	Feedback
Responsive design / dark mode	✓ Hoofdverantwoordelijke	
Wireframes	✓ Hoofdverantwoordelijke	
Back-end (PHP)	Ondersteuning	✓ Hoofdverantwoordelijke
Databasebeheer (MySQL)		✓ Hoofdverantwoordelijke
Authenticatie & sessies		✓ Hoofdverantwoordelijke
ERD & normalisatie	Meegewerkt	✓ Hoofdverantwoordelijke
Testing & debugging	✓ Gezamenlijk	✓ Gezamenlijk
Documentatie	✓ Gezamenlijk	✓ Gezamenlijk
Deploy & GitHub	✓ Gezamenlijk	✓ Gezamenlijk

13. ONTWIKKELINGSPROCES

Analysefase (week 1–2)

In deze fase werden de functionele vereisten bepaald op basis van interviews met potentiële gebruikers en een analyse van vergelijkbare planningstools. De doelgroepen werden gedefinieerd en de gebruikersrollen werden vastgelegd.

Ontwerpfase (week 2–3)

Wireframes werden ontworpen in Figma voor alle pagina's. Het ERD-model versie 1 werd opgesteld. De technische stack (PHP + MySQL + HTML/CSS/JS) werd definitief gekozen na evaluatie van alternatieven (Laravel, Node.js).

Databasefase (week 3–4)

De 13 entiteiten werden uitgewerkt met attributen, datatypes en relaties. Het ERD werd genormaliseerd naar 3NF. De database werd aangemaakt op de hosted MySQL-server en gevuld met testdata.

Ontwikkelingsfase (week 4–8)

De front-end en back-end werden parallel ontwikkeld en wekelijks geïntegreerd via GitHub. Feature branches werden gebruikt voor grote nieuwe functionaliteiten (shiftbeheer, notificatiesysteem, kalender). Pull requests werden onderling gereviewd.

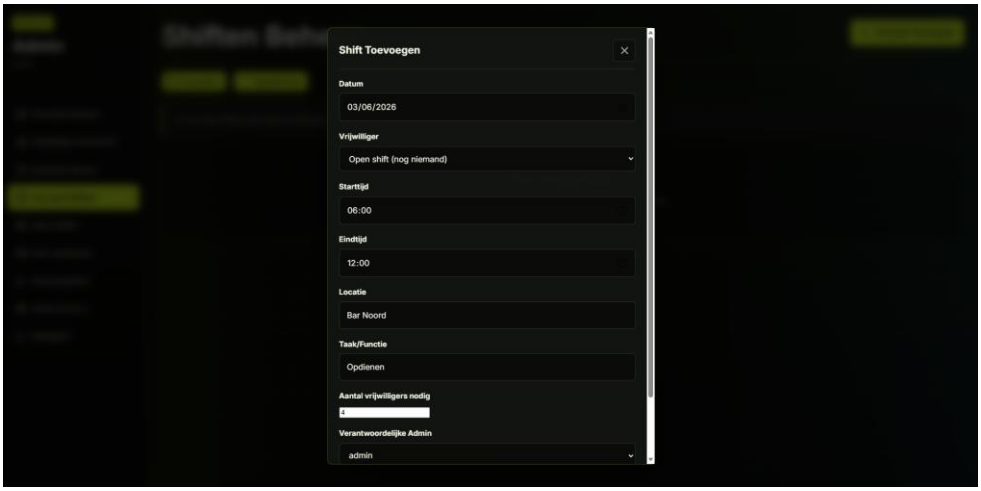
Testfase (week 8–9)

De applicatie werd getest op functionaliteit (alle CRUD-operaties), gebruiksvriendelijkheid (test met een medestudent) en stabiliteit (edge cases: lege velden, overlappende shifts, ongeautoriseerde URL-toegang).

14. GEBRUIKERSSCENARIO'S

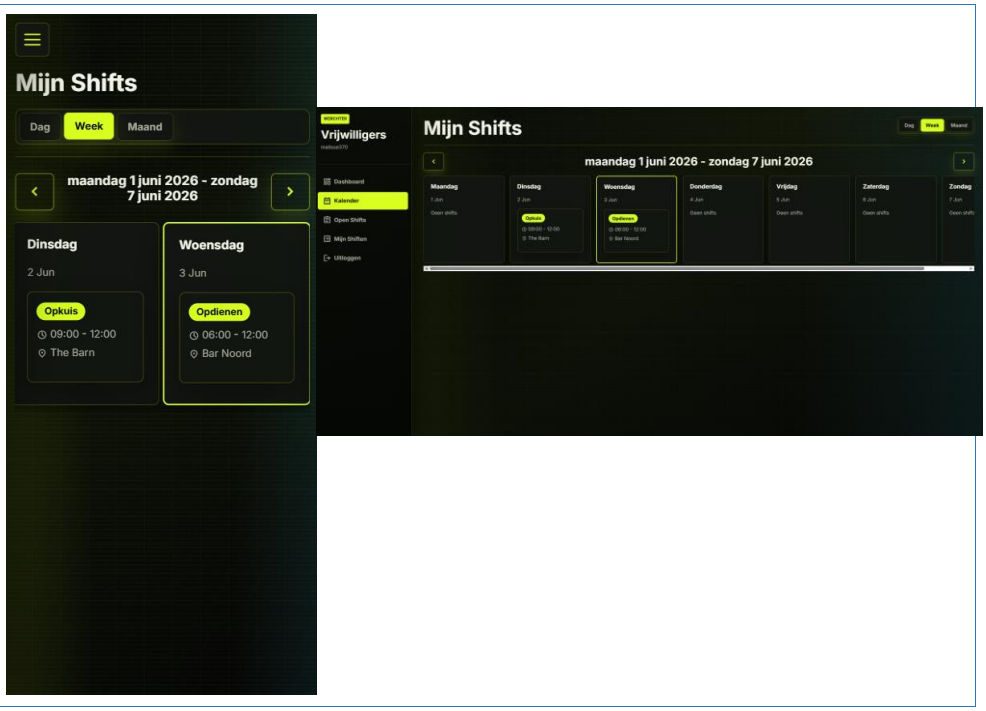
Scenario 1 – Administrator plant een shift

Een administrator logt in met zijn beheerdersaccount. Via het adminmenu navigeert hij naar 'Shiftbeheer' en klikt op 'Nieuwe shift aanmaken'. Hij vult de werkpost (Bar Noord), start- en eindtijd (12:00–16:00), en maximumaantal vrijwilligers (4) in. Vervolgens wijst hij drie beschikbare vrijwilligers toe. Het systeem controleert automatisch op tijdsoverlap en toont een bevestigingsmelding. De drie betrokken vrijwilligers ontvangen automatisch een notificatie in hun dashboard.

Schermafbeelding	

Scenario 2 – Vrijwilliger bekijkt zijn planning

Een vrijwilliger logt in en wordt doorgestuurd naar zijn persoonlijk dashboard. Het dashboard toont de volgende drie shifts in een overzichtskaart, samen met ongelezen meldingen. Via de kalenderweergave navigeert de vrijwilliger naar de festivalweek en ziet alle zijn geplande shifts kleurgecodeerd per werkpost. Hij klikt op een shift voor meer details (locatie, groepleider, kledingvoorschriften).

Schermafbeelding	

Scenario 3 – Groepleider beheert zijn groep

Een groepleider logt in en ziet in zijn dashboard een overzicht van alle vrijwilligers in zijn groep met hun status (aangemeld, ingepland, niet beschikbaar). Hij constateert dat één vrijwilliger nog geen beschikbaarheid heeft ingevuld. Via de berichtenmodule stuurt hij een herinnering. Hij controleert ook de planning van de komende festivaldag en rapporteert een bezettingsprobleem (te weinig vrijwilligers op shift Parking B) aan de administrator via het rapportagesysteem.

15. REFLECTIE

Persoonlijke reflectie – Matisse De Keersmaeker

Dit project was een intensieve maar leerrijke ervaring. Het ontwerpen van de interface in Figma en het vertalen daarvan naar werkende HTML/CSS code gaf mij een goed begrip van het belang van een goede voorbereiding. De grootste uitdaging was het responsive maken van de kalenderweergave op kleine schermen: de grid-layout moest op mobiel worden omgezet naar een scrollbare lijstweergave, wat meer CSS-complexiteit vergde dan verwacht.

Dark mode was ook uitdagender dan gedacht: het consequent doorvoeren van kleurveranderingen via CSS-variabelen vereiste discipline in de codestructuur. Ik heb geleerd hoe belangrijk het is om van bij het begin een systeem voor themabeheer op te zetten in plaats van achteraf kleuren te vervangen.

De samenwerking met Alan verliep vlot. De splitsing tussen front-end en back-end was logisch, maar we kwamen regelmatig samen om de integratie te testen. In een volgend project zou ik eerder willen starten met gedeelde componentstandaarden om inconsistenties in de CSS te vermijden.

Persoonlijke reflectie – Alan Florczak

Het moeilijkste van dit project was de verbinding maken met Combell en de database daar op laten werken. Lokaal werkte alles perfect, maar van zodra we het online zetten op Combell begonnen er problemen te komen met de databaseconnectie. De credentials moesten anders ingesteld worden, de hostnaam was anders dan verwacht en het duurde een tijdje voor we alles goed hadden staan.

Het authenticatiesysteem bouwen in PHP was ook complex. Je moet goed nadenken over beveiliging want als je dat fout doet kunnen mensen dingen zien of doen die ze niet mogen. Ik heb SQL-injectie voorkomen via PDO prepared statements en sessies correct ingesteld.

De database normaliseren naar 3NF had meerdere versies nodig. In de eerste versie zaten er nog fouten die we later moesten oplossen. Dat heeft mij geleerd dat je beter meer tijd steekt in het ontwerp dan achteraf alles te moeten herwerken.

Teamreflectie

Als team communiceerden we wekelijks via een overlegmoment en dagelijks via een gedeeld kanaal. GitHub hielp ons om conflicten in code te beperken, maar we hebben geleerd dat branches consequenter moeten worden benoemd en pull requests vaker en eerder moeten worden ingediend.

Het project heeft ons beiden meer inzicht gegeven in de volledige software development lifecycle: van analyse en ontwerp tot implementatie, testing en documentatie. We zijn tevreden met het eindresultaat en geloven dat de applicatie een reële waarde zou kunnen bieden voor een festivalorganisatie.

16. BESLUIT

Het project 'Rock Werchter Vrijwilligers Scheduler' resulteerde in een volledig werkende webapplicatie voor het beheren van vrijwilligers en planningen binnen een festivalomgeving.

De applicatie biedt een rolgebaseerde interface voor administrators, groepleiders en vrijwilligers, met functionaliteiten zoals shiftbeheer, overlappingscontrole, kalenderweergave, notificaties en responsive design. De database werd genormaliseerd tot 3NF en is gedocumenteerd in een ERD.

Technisch gezien combineerde het project HTML/CSS/JavaScript voor de front-end, PHP 8.2 voor de back-end en MySQL als persistente opslaglaag. Git en GitHub werden gebruikt voor versiebeheer en teamcollaboratie.

Het project leverde waardevolle praktijkervaring op in teamwork, planning, databasemodellering, webontwikkeling en probleemoplossend denken. De opgedane kennis vormt een stevige basis voor toekomstige, complexere projecten.

BIBLIOGRAFIE

Onderstaande bronnen werden geraadpleegd tijdens de analyse, het ontwerp en de ontwikkeling van de applicatie:

Officiële documentatie

- PHP Group. (2024). PHP Manual. https://www.php.net/manual/
- MySQL AB. (2024). MySQL 8.0 Reference Manual. https://dev.mysql.com/doc/refman/8.0/
- Mozilla Developer Network. (2024). MDN Web Docs – HTML, CSS, JavaScript. https://developer.mozilla.org/
- World Wide Web Consortium (W3C). (2023). HTML Living Standard. https://html.spec.whatwg.org/

Boeken en cursusmateriaal

- Welling, L., & Thomson, L. (2017). PHP and MySQL Web Development (5e ed.). Addison-Wesley.
- Elmasri, R., & Navathe, S. B. (2016). Fundamentals of Database Systems (7e ed.). Pearson.
- Cursusmateriaal Geïntegreerde Softwareprojecten, eigen instelling, schooljaar 2025–2026.

Online bronnen en tools

- Figma Inc. (2024). Figma – Design tool. https://www.figma.com/
- dbdiagram.io. (2024). Database diagram tool. https://dbdiagram.io/
- Anthropic. (2024). Claude AI – Codeerondersteuning. https://www.anthropic.com/
- Cursor AI. (2024). AI-powered code editor. https://www.cursor.com/
- GitHub Inc. (2024). GitHub – Version control. https://github.com/
- Stack Overflow. (2024). Community Q&A voor PHP, MySQL en JavaScript. https://stackoverflow.com/

Normalisatie referentie

- Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, 13(6), 377–387.
- Date, C. J. (2003). An Introduction to Database Systems (8e ed.). Addison-Wesley.

Einde van het dossier.