

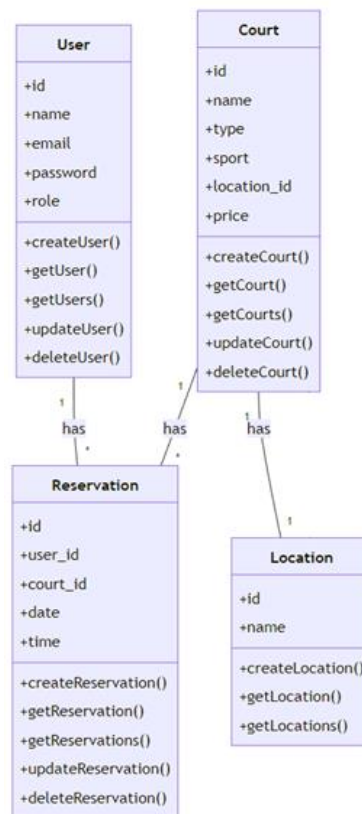
Document Webapplicatie

Naam: Norshin Mirza
Student Nummer: 538131

Table of Contents

1. Class diagram	3
2. sequence diagrams	4
3. Use Case Diagram	5
4. Wireframes.....	6
5. Endpoints.....	12
Courts Endpoints.....	12
Reservations Endpoints.....	14
Users Endpoints	17
Locations endpoints	18

1. Class diagram

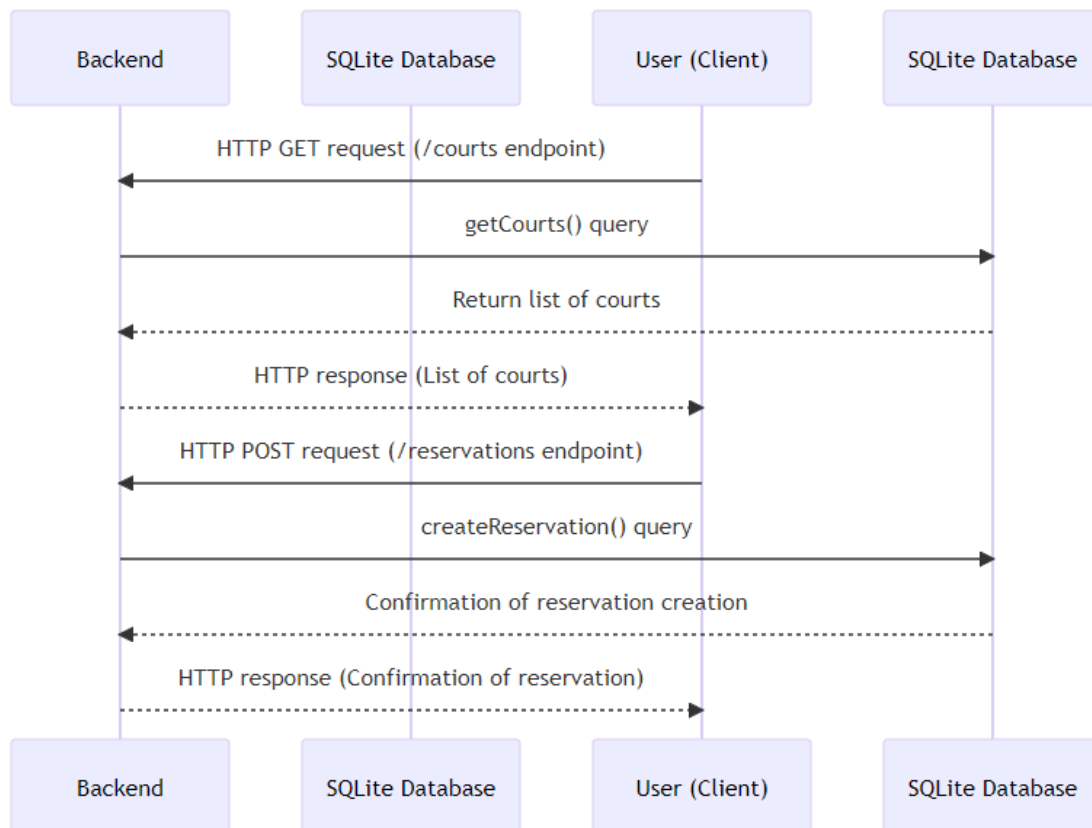


In dit diagram:

- De **User** klasse heeft attributen zoals id, name, email, password en role. Het heeft methoden zoals createUser, getUser, getUsers, updateUser en deleteUser. Een User kan meerdere Reservations hebben (one-to-many relatie).
- De **Court** klasse heeft attributen zoals id, name, type, sport, location_id en price. Het heeft methoden zoals createCourt, getCourt, getCourts, updateCourt en deleteCourt. Een Court kan meerdere Reservations hebben (one-to-many relatie). Een Court heeft een Location (one-to-one relatie).
- De **Reservation** klasse heeft attributen zoals id, user_id, court_id, date en time. Het heeft methoden zoals createReservation, getReservation, getReservations, updateReservation en deleteReservation. Een Reservation behoort tot een User (many-to-one relatie). Een Reservation behoort tot een Court (many-to-one relatie).
- De **Location** klasse heeft attributen zoals id en name. Het heeft methoden zoals createLocation, getLocation en getLocations. Een Location kan meerdere Courts hebben (one-to-many relatie).

Elke methode voert een specifieke actie uit die gerelateerd is aan zijn klasse. Bijvoorbeeld, de createUser methode in de User klasse maakt een nieuwe gebruiker aan, terwijl de getCourts methode in de Court klasse alle banen ophaalt. Deze methoden worden gebruikt om de functionaliteit van de applicatie te implementeren en om interactie met de database mogelijk te maken.

2. Sequence diagrams



Het proces begint wanneer de JavaScript code op de reserveringspagina van de gebruiker wordt uitgevoerd. Deze code stuurt een HTTP GET-verzoek naar de /courts endpoint van de backend om een lijst van alle beschikbare banen op te halen.

De backend ontvangt het verzoek en roept de `getCourts` functie in de `courtsController` aan. Deze functie maakt een query naar de SQLite database om alle banen op te halen. De SQLite database voert de query uit en stuurt de resultaten terug naar de backend.

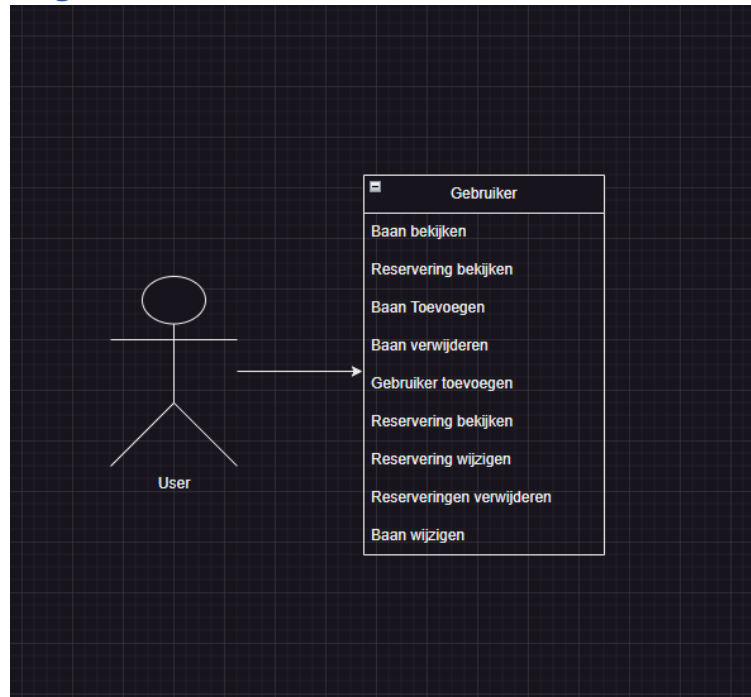
De backend verpakt de resultaten in een http-response en stuurt deze terug naar de frontend. De JavaScript code op de reserveringspagina van de gebruiker ontvangt de response, haalt de lijst van banen uit de response en toont deze aan de gebruiker.

Wanneer de gebruiker een baan en een tijdslot selecteert, hun persoonlijke gegevens invult en op de knop klikt om de reservering te maken, activeert dit een JavaScript functie. Deze functie verzamelt de ingevoerde gegevens en stuurt een HTTP POST-verzoek naar de /reservations endpoint van de backend om een nieuwe reservering te maken. De details van de reservering worden in de body van het verzoek opgenomen.

De backend ontvangt het verzoek en roept de `createReservation` functie in de `reservationsController` aan. Deze functie valideert de gegevens in het verzoek en maakt een nieuwe rij in de reserveringen tabel van de SQLite database met deze gegevens. De SQLite database maakt de nieuwe rij en stuurt een bevestiging terug naar de backend.

De backend verpakt de bevestiging in een http-response en stuurt deze terug naar de frontend. De JavaScript code op de reserveringspagina van de gebruiker ontvangt de response, controleert of de reservering succesvol was gemaakt en toont een bevestigingsbericht aan de gebruiker.

3. Use Case Diagram



Gebruikers kunnen verschillende acties uitvoeren. Ze kunnen alle beschikbare banen bekijken, wat hen een overzicht geeft van de opties die ze hebben voor het reserveren van een baan. Als ze een baan hebben gekozen die ze willen reserveren, kunnen ze een reservering maken voor een specifiek tijdslot op die baan.

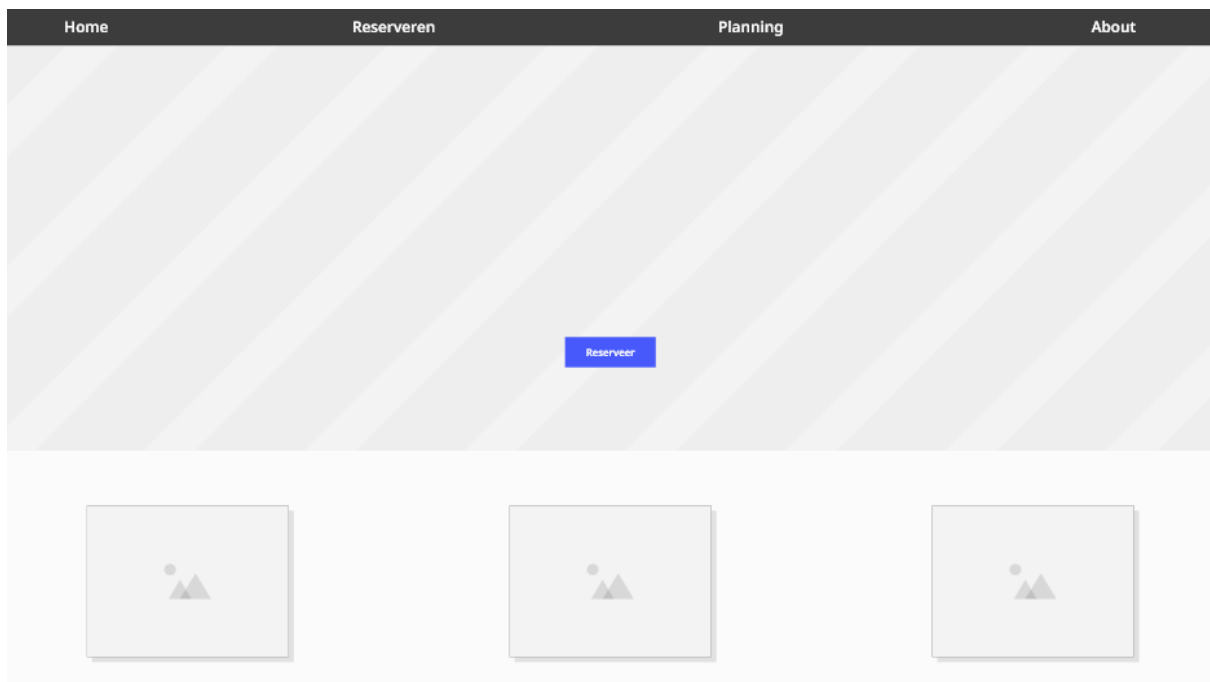
Gebruikers hebben ook de mogelijkheid om nieuwe banen toe te voegen aan het systeem. Ze kunnen de details van de nieuwe baan invullen en deze toevoegen, waardoor de baan beschikbaar wordt voor reserveringen door andere gebruikers. Als er een baan is die niet langer beschikbaar of nodig is, kunnen gebruikers ook een bestaande baan verwijderen uit het systeem.

Naast het beheren van banen, kunnen gebruikers ook een lijst van alle geregistreerde gebruikers in het systeem bekijken. Dit geeft hun een overzicht van wie er allemaal gebruik maakt van de applicatie.

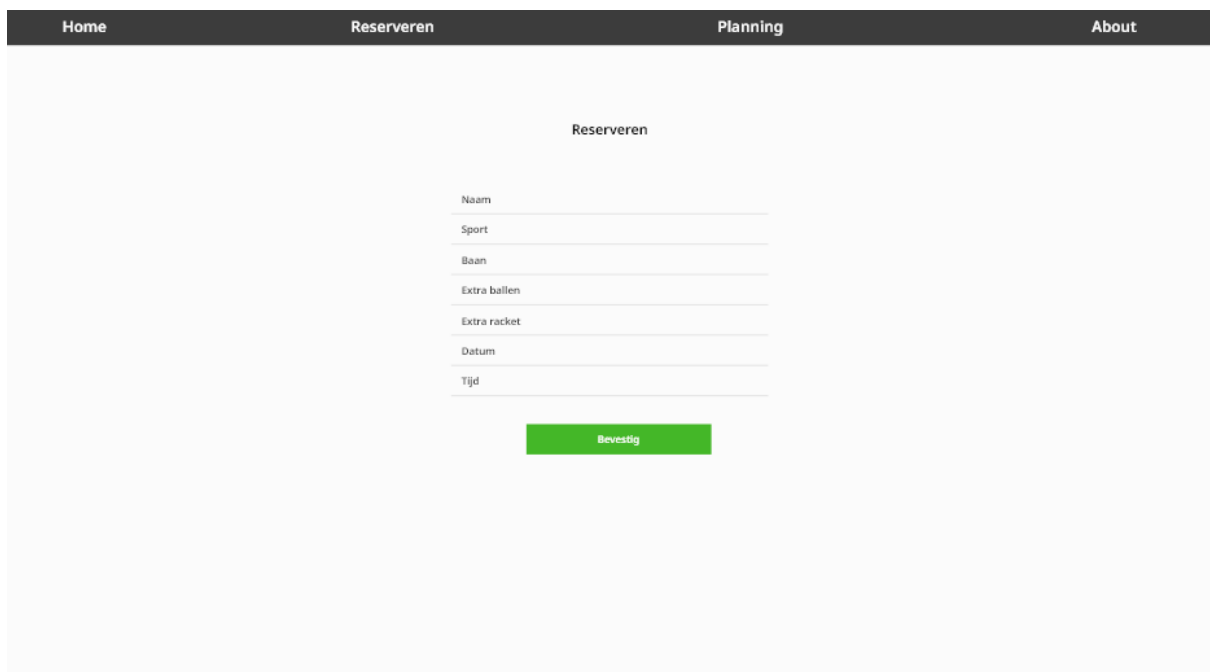
Wat betreft reserveringen, kunnen gebruikers de details van een specifieke reservering bekijken. Als ze wijzigingen willen aanbrengen in een bestaande reservering, zoals het veranderen van het tijdslot of de gekozen baan, kunnen ze de details van de reservering wijzigen. Als een reservering niet langer nodig is, kunnen ze ook een bestaande reservering verwijderen.

Tot slot kunnen gebruikers, als ze geïnteresseerd zijn in het bekijken van reserveringen die op een specifieke datum zijn gemaakt, de reserveringen filteren op datum. Dit geeft hun een overzicht van alle reserveringen die op die datum zijn gemaakt.

4. Wireframes



Dit wireframe vertegenwoordigt de startpagina van je applicatie. Het bevat meestal een logo of titel van je applicatie, een welkomstbericht en mogelijk een korte beschrijving van de functionaliteit. Het kan ook een navigatiebalk of links naar andere belangrijke pagina's bevatten.



Dit wireframe toont de pagina waar gebruikers een reservering kunnen maken. Het bevat een formulier met velden zoals naam, sport, baan, datum en tijd. Gebruikers kunnen deze velden invullen en op een knop drukken om de reservering te maken.

HomeReserverenPlanningAbout

Reserveren

Norshin

Padel

2

2

3

10-10-2023

12:30

Bevestig

Reservering bevestigd!

Dit wireframe toont de pagina wanneer de gebruiker zijn reservering heeft bevestigd. Er toont dan een groene tekst onder het formulier met “Reservering bevestigd!”.

HomeReserverenPlanningAbout

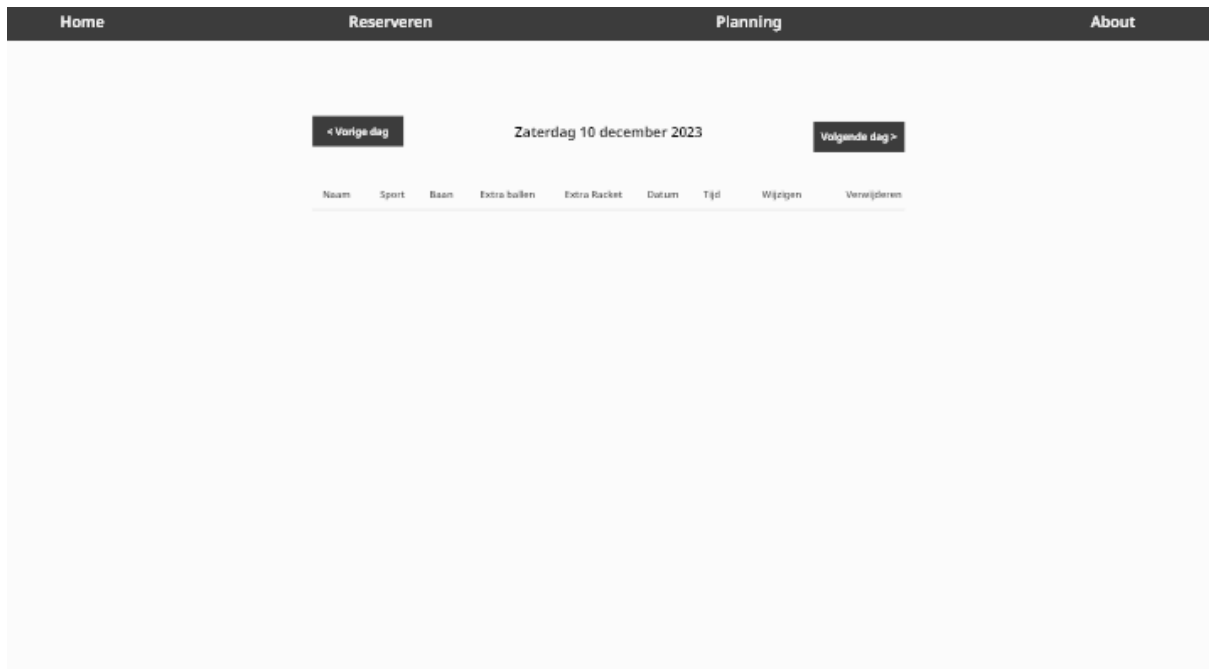
< Vorige dag

Zaterdag 10 december 2023

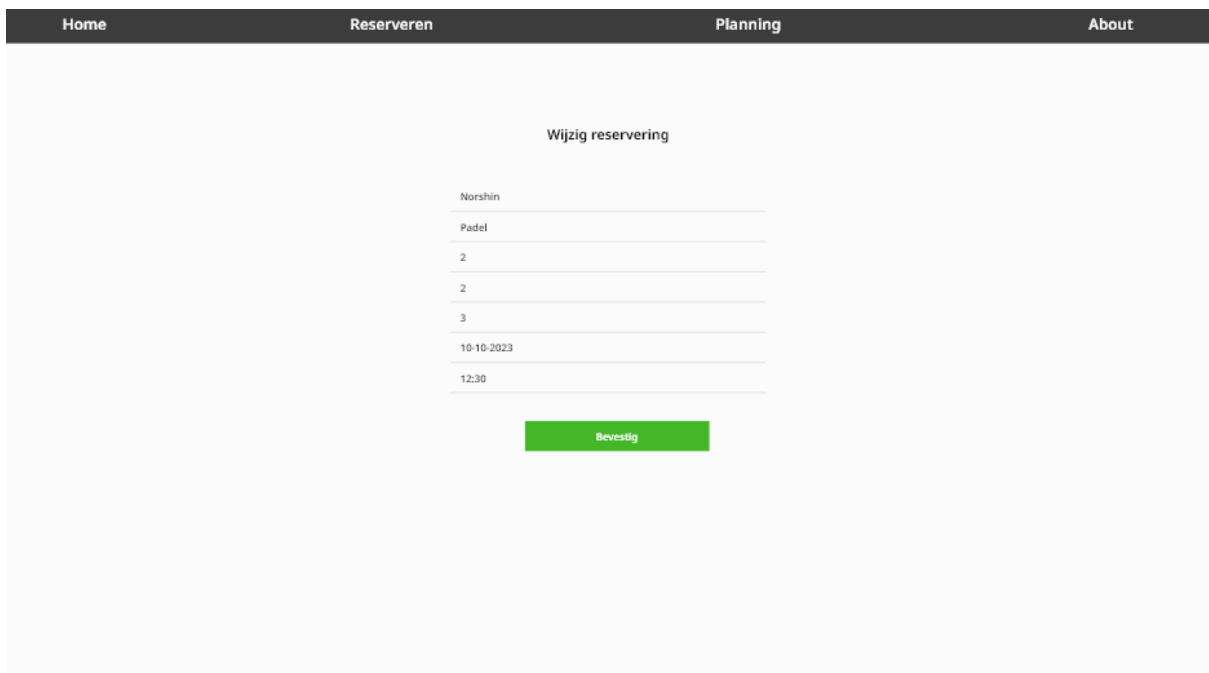
Volgende dag >

Naam	Sport	Baan	Extra ballen	Extra Racket	Datum	Tijd	Wijzigen	Verwijderen
Norshin	Padel	2	2	3	10-11-2023	12:30	Wijzigen	Verwijderen

Dit wireframe toont de pagina waar gebruikers de beschikbaarheid van banen of tijdschema's kunnen bekijken. Het kan een overzichtelijke weergave zijn van de beschikbare tijden per baan, met kleurcodes om de bezettingsstatus aan te geven.



Dit wireframe toont de pagina waar gebruikers een bestaande reservering kunnen verwijderen. Het bevat een lijst met alle reserveringen van de gebruiker of een overzicht van alle reserveringen in het systeem. Gebruikers kunnen op de "Verwijderen" knop klikken om de bijbehorende reservering te verwijderen.



Dit wireframe vertegenwoordigt de pagina waar gebruikers een bestaande reservering kunnen wijzigen. Het bevat een formulier met voorgeladen gegevens van de reservering die kan worden bewerkt. Gebruikers kunnen wijzigingen aanbrengen en deze opslaan door op een knop te klikken.

Home	Reserveren	Planning	About
Wijzig reservering			
<div> <div>Norshin</div> <div>Padel</div> <div>2</div> <div>2</div> <div>3</div> <div>10-10-2023</div> <div>12:30</div> </div>			
<div>Bevestig</div> <div>Reservering Gewijzigd!</div>			

Dit wireframe toont de pagina wanneer de gebruiker zijn reservering wijziging heeft bevestigd. Er toont dan een groene tekst onder het formulier met “Reservering bevestigd!”.

Home	Reserveren	Planning	About
About			
<p>At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.</p>			

Dit wireframe vertegenwoordigt de pagina waar je meer informatie kunt geven over je applicatie, zoals het doel, de missie, de geschiedenis of andere relevante informatie. Het kan ook afbeeldingen, citaten of testimonia bevatten om je boodschap te ondersteunen.

[Home](#)[Reserveren](#)[Planning](#)[About](#)[Register](#)[Banen](#)

Gebruikersnaam:

E-mail:

Account aanmaken

Hier worden nieuwe gebruikers toegevoegd en zal er zo uit zien

[Home](#)[Reserveren](#)[Planning](#)[Banen](#)[About](#)[Register](#)

Banen

Nieuwe baan

Type	Locatie	Verwijderen	Wijzigen
Padel	Oldenzaal	Verwijderen	Wijzigen
Tennis	Oldenzaal	Verwijderen	Wijzigen
Tennis	Enschede	Verwijderen	Wijzigen

Hier kan je de lijst zien van de banen die er op dit moment beschikbaar zijn.

[Home](#)[Reserveren](#)[Planning](#)[About](#)[Register](#)[Banen](#)

Wijzig een baan

Type:

Padel

Locatie:

Oldenzaal

Wijzig

Hier kan de gebruiker de banen die toegevoegd zijn wijzigen als dat nodig is.

[Home](#)[Reserveren](#)[Planning](#)[About](#)[Register](#)[Banen](#)

Voeg een nieuwe baan toe

Type:

Locatie:

Voeg toe

Hier kan de gebruiker nieuwe banen toevoegen.

5. Endpoints

Courts Endpoints

Template POST table:

GET	/courts		
Haalt alle banen op			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.	/	/	/
Responses:	Code	Description/ example if successful	
	200 OK	Succesvol opgehaald, retourneert een lijst van banen.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	

GET	/courts/byType		
Haalt bannen op basis van hun Type			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.	Court_Type	Tekst	Type baan die de user uitkiest
Responses:	Code	Description/ example if successful	
	200 OK	Succesvol opgehaald, retourneert een lijst van banen van een bepaald type.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	

GET	/courts/:id		
Haalt een specifieke baan op basis van het ID.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.	ID*	Integer	Het ID van de baan
Responses:	Code	Description/ example if successful	
	200 OK	Succesvol aangemaakt, retourneert de details van de baan.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	
	404 Not Found	Baan niet gevonden	

POST	/courts		
Maakt een nieuwe baan aan			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.			
Responses:	Code	Description/ example if successful	
	201 Created	Succesvol aangemaakt, retourneert de details van de nieuwe baan	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	
	400 Bad request	Een aanvraag was ongeldig of onvolledig.	

PUT	/courts/:id		
Wijzigt de details van een bestaande baan.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.	Id	Integer	Het id van de baan
Responses:	Code	Description/ example if successful	
	200 OK	Succesvol bijgewerkt, retourneert de bijgewerkte details van de baan.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	
	400 Bad request	E aanvraag was ongeldig of onvolledig.	

DELETE	/courts/:id		
Verwijdert een bestaande baan.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.	Id	Integer	Het id van de baan
Responses:	Code	Description/ example if successful	
	200 OK	Succesvol verwijderd.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	
	404 Not found	Baan niet gevonden.	

Reservations Endpoints

POST	/reservations		
Haalt alle reserveringen op.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.			
Responses:	Code	Description/ example if successful	
	200 Ok	Succesvol opgehaald, retourneert een lijst van reserveringen.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	

POST	/reservations		
Haalt alle reserveringen op.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.			
Responses:	Code	Description/ example if successful	
	200 Ok	Succesvol opgehaald, retourneert een lijst van reserveringen.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	

POST	/reservations/byDate		
Haalt reserveringen op op basis van hun datum.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.			
Responses:	Code	Description/ example if successful	
	200 Ok	Succesvol opgehaald, retourneert een lijst van reserveringen op een bepaalde datum.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	

POST	/reservations		
Maakt een nieuwe reservering aan			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.			
Responses:	Code	Description/ example if successful	
	201 Created	Succesvol aangemaakt, retourneert de details van de nieuwe reservering.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	
	400 Bad Request	De aanvraag was ongeldig of onvolledig.	

DELETE	/reservations/:id		
Verwijdert een bestaande reservering.			
Parameters:	Name	Type	Description
<i>Add a * to the name of required parameters.</i>	ID	Integer	Het ID van de reservering
Responses:	Code	Description/ example if successful	
	200 OK	Succesvol verwijderd	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	
	404 NOT found	Reservering niet gevonden.	

GET	/reservations/:id		
Haalt een specifieke reservering op op basis van het ID.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.	ID	Integer	Het ID van de reservering
Responses:	Code	Description/ example if successful	
	200 OK	Succesvol opgehaald, retourneert de details van de reservering.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	
	404 NOT found	Reservering niet gevonden.	

PUT	/reservations/:id		
Wijzigt de details van een bestaande reservering.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.	ID	Integer	Het ID van de reservering
Responses:	Code	Description/ example if successful	
	200 OK	Succesvol bijgewerkt, retourneert de bijgewerkte details van de reservering.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	
	404 NOT found	Reservering niet gevonden.	

Users Endpoints

GET	/users		
Haalt alle gebruikers op.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.			
Responses:	Code	Description/ example if successful	
	200 OK	Succesvol opgehaald, retourneert een lijst van gebruikers.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	

POST	/users		
Maakt een nieuwe gebruiker aan.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.			
Responses:	Code	Description/ example if successful	
	201 Created	Succesvol aangemaakt, retourneert de details van de nieuwe gebruiker.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	
	400 bad request	De aanvraag was ongeldig of onvolledig.	

POST	/users/check-email		
Controleert of een e-mailadres al in gebruik is door een andere gebruiker.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.			
Responses:	Code	Description/ example if successful	
	200 OK	Succesvol gecontroleerd, retourneert een bericht dat aangeeft of het e-mailadres al dan niet in gebruik is.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	
	400 bad request	De aanvraag was ongeldig of onvolledig.	

Locations endpoints

GET	/locations		
Haalt alle locaties op.			
Parameters:	Name	Type	Description
Add a * to the name of required parameters.			
Responses:	Code	Description/ example if successful	
	200 OK	Succesvol opgehaald, retourneert een lijst van locaties.	
	500 Internal Server Error	Er is iets misgegaan aan de serverzijde.	