

Sprint – report

Start date	17/11/2025
End date	30/11/2025
Sprint #	1
Group	8
Students	Xander UijtdeHaag, Bart Geijtenbeek, Boaz Vaneveld, Rick de Cuijper, Valentijn van Grunsven
Topic	Organization and Motivation for Maintenance of Sustainable Schoolyards Organisatie en Motivatie voor Onderhoud van Duurzame Schoolpleinen

Related System requirements (sprint backlog)

Todo

- **Weekindeling kalenderscherm maken en stylen op de back-end**
Assignee: Xanderu07
- **C: Taken kunnen aanmaken in de agenda op de front-end**
Assignee: BoazVaneveld
- **C: Taken kunnen aanmaken in de agenda op de back-end**
Assignee: BoazVaneveld
- **R: Taken zijn zichtbaar volgens het Figma-prototype in de agenda op de back-end**
Assignee: BoazVaneveld
- **Alle aangemaakte taken in de kalender moeten zichtbaar en bruikbaar worden in de schatkaart-pagina zoals in het Figma prototype (Back-end)**
Assignee: BoazVaneveld
- **Maak de avatar-customisatiepagina volgens het Figma-prototype (Front-end)**
Assignee: rickdecuijper
- **Maak de avatar-customisatiepagina volgens het Figma-prototype (Back-end)**
Assignee: rickdecuijper

In Progress

- **Database maken op de back-end**
Assignee: BoazVaneveld
- **Authenticatie leren**
Assignee: bjhgeijtenbeek
- **Weekindeling bij het kalenderscherm maken op de front-end**
Assignee: Xanderu07
- **Registratie maken op de back-end**
Assignee: bjhgeijtenbeek

Testing

- **Unit tests maken voor je feature (minimaal 3 tests) voor CIN**

In Review

- **Activiteiten worden gereviewd**

Voeg toe aan de beschrijving wanneer afgerond: "*Reviewed by [Name]*"

Done

- **Architectural Component Diagram**
Assignee: *bjhgeijtenbeek*
- **Target Audience & Persona**
Assignee: *bjhgeijtenbeek*
- **Github klaarzetten voor het project (front-end en back-end)**
Assignee: *rickdecuijper*
- **Implementeer de unit tests in de dev-pipeline**
Assignee: *rickdecuijper*
- **CI/CD pipeline goed opgezet voor het project in GitHub**
Assignee: *rickdecuijper*

Tasks

Task	Assignee	Estimated hours	Spend hours	Progress
Repository voor frontend doeileinden opzetten	Rick	1 uur		100%
Repositories voor backend doeileinden opzetten	Rick	1 uur		100%
Github project bijwerken met huidige taken	Rick	4 uur		0%
Frontend en backend online on render	Valentijn	4 uur		
Authentication leren	Bart	8 uur		
Registratie maken	Bart	12 uur		
Inlog maken	Bart	12 uur		
Weekindeling bij het kalenderscherm maken	Xander	50 min		
Weekindeling kalenderscherm stijlen	Xander	8 uur		
CRUD-taken van kalender	Boaz	25 uur		
Database leren (prisma of SQL) (database als microservice)	Boaz	25 uur		

Taken die zijn aangemaakt moeten op de correcte plaats in de kalender verschijnen	Boaz	2 uur		
Knoppen voor pagina verandering aanmaken	Xander	1 uur		
Pagina met schatkaart aanmaken	Xander	30 min		
Schatkaart pagina stijlen	Xander	5 uur		
Taken die zijn aangemaakt op de kalender moeten ook op de schatkaart tevoorschijn komen	Boaz	2 uur		
Pagina voor de avatars maken	Rick	7 uur		
Knoppen voor customisatie opties voor avatar maken	Rick	7 uur		
Zorgen dat customisatie knoppen uiterlijk van de avatar veranderen	Rick	7 uur		

Retrospective summary

Start	Stand-downs inplannen met het hele team. Betere communicatie tussen het team Betere documentatie van werk Betere samenwerking tussen teamgenoten
Stop	Niet aan afspraken houden Niet op tijd zijn bij lessen
Keep	Goede takenverdeling tussen teamgenoten

Test (for sprint 2 and 3 only)

Summary of the test

Start	N/A
Stop	N/A
Keep	N/A

Demo

Summary of the demo

Start	Gebruikmaken van de front-end om onze code en designs te demonstreren.
Stop	Met gebruik van het Figma-prototype de front-end inzetten om onze code en designs te demonstreren.
Keep	Boaz verantwoordelijk maken voor het geven van demo's aan de klant.

Design documentation

Use Case Specification

{insert only one; what is the most important use-case for this sprint and complex enough to model}

Use case name	
Actor(s)	
Summary description	
Priority	
Status	
Pre-condition	
Post-condition	
Basic path	
Alternative paths	
Business rules	
Non-functional requirements	

Dataflow Diagram

{insert your Data Flow Diagram of the Use Case. Tip: It's important to show details of what data is needed / sent/received / transformed / stored and where (which entities = actors) but this is suggested to be at 'Context Diagram' level 0 for the Use Case itself (not the whole system). This will also help you in updating the ERD plus also for message design}

API specification and Message(s)

First Basic Response to Back end. No changes made to base code.

Request:

GET <http://localhost:3011/appointments/>

Response:

```
{
  "meta": {
    "count": 4,
    "title": "All appointments",
    "url": "/api/v1/appointments/"
  },
  "data": [
    "/appointments/1",
    "/appointments/2",
    "/appointments/3",
    "/appointments/4"
  ]
}
```

Release

[Release First release of the back end. · rickdecuijper/UVC-project-Back-End](#)

[Release First release of the front end. · rickdecuijper/UVC-Project-Front-End](#)

Pull request

Every student should code. In this table you relate an issue on Github with a task you mention above. When finishing an issue is a pull request should be created and one of the other students should review the pull request and either reject or accept the request. Every sprint a students should at least make two pull requests.

Nr	Issue	Task	Responsible	Reviewed
# 9	Weekindeling bij het kalenderscherm maken op de front-end	Weekindeling bij het kalenderscherm maken op de front-end	Xander	Boaz
# 10 # 12 # 13	Implementeer de unit tests in de dev-pipeline	Make unit test for the pipeline	Rick	Bart en Valentijn