

# Opdrachtbeschrijving

ICT-B1.3 LU2 Individueel Minimum Viable Product

## Context

Voor deze opdracht ontwikkel je individueel een grafische 2D-applicatie waarbij applicatiedata van gebruikers op een veilige manier online wordt opgeslagen. Een praktijkvoorbeeld van een soortgelijke applicatie is [Gather](https://www.linkedin.com/posts/dtobon_accountingandaccountants-taxtwitter-activity-7250128013793595392-jW3A). Deze applicatie laat gebruikers een eigen 2D-omgeving inrichten met het doel om deze ruimte samen met andere gebruikers te benutten als een online kantoorruimte of conferentiegebouw.

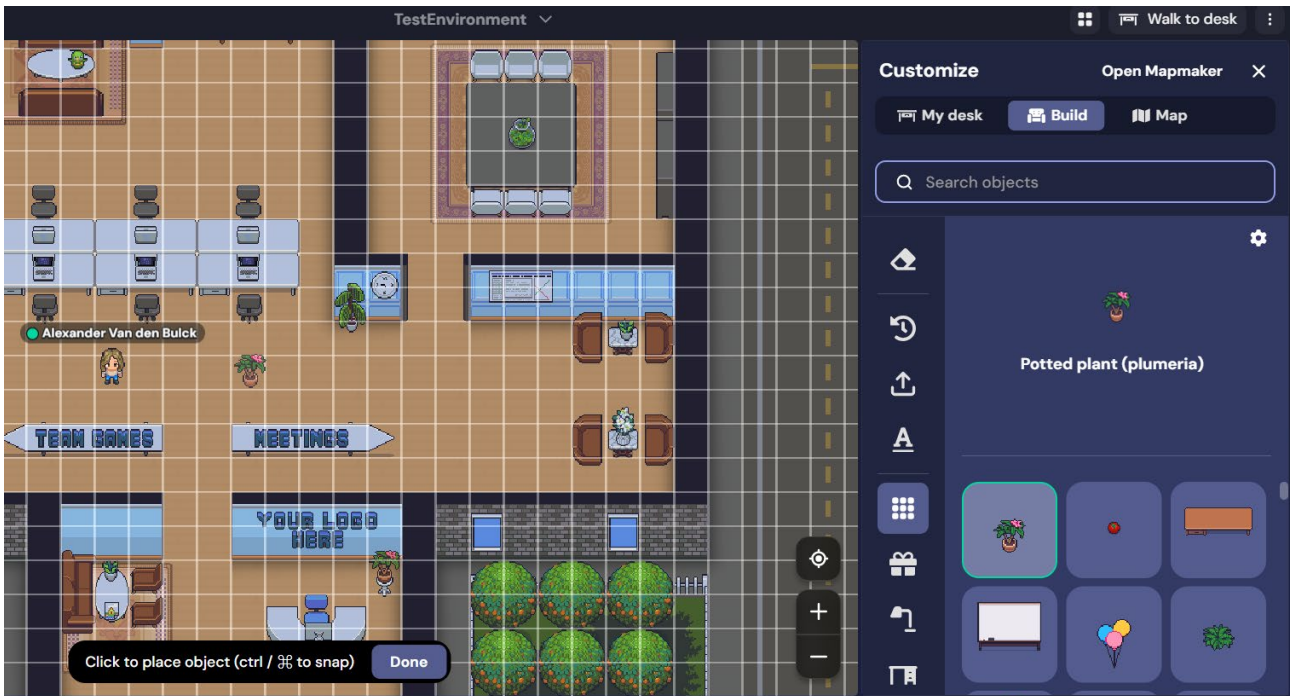


*Figuur 1: Voorbeeld digitale werkruimte in Gather.  
([https://www.linkedin.com/posts/dtobon\\_accountingandaccountants-taxtwitter-activity-7250128013793595392-jW3A](https://www.linkedin.com/posts/dtobon_accountingandaccountants-taxtwitter-activity-7250128013793595392-jW3A))*

Gather bevat talrijke interessante features die leerzaam en uitdagend zijn om na te bouwen. Voor deze opdracht gaan we specifiek de feature nabouwen waarmee je eigen 2D-omgevingen kan bouwen.

## 2D Environment Creator

Voor deze opdracht ga je een 2D Environment Creator applicatie ontwikkelen geïnspireerd door de gelijkaardige feature van Gather.



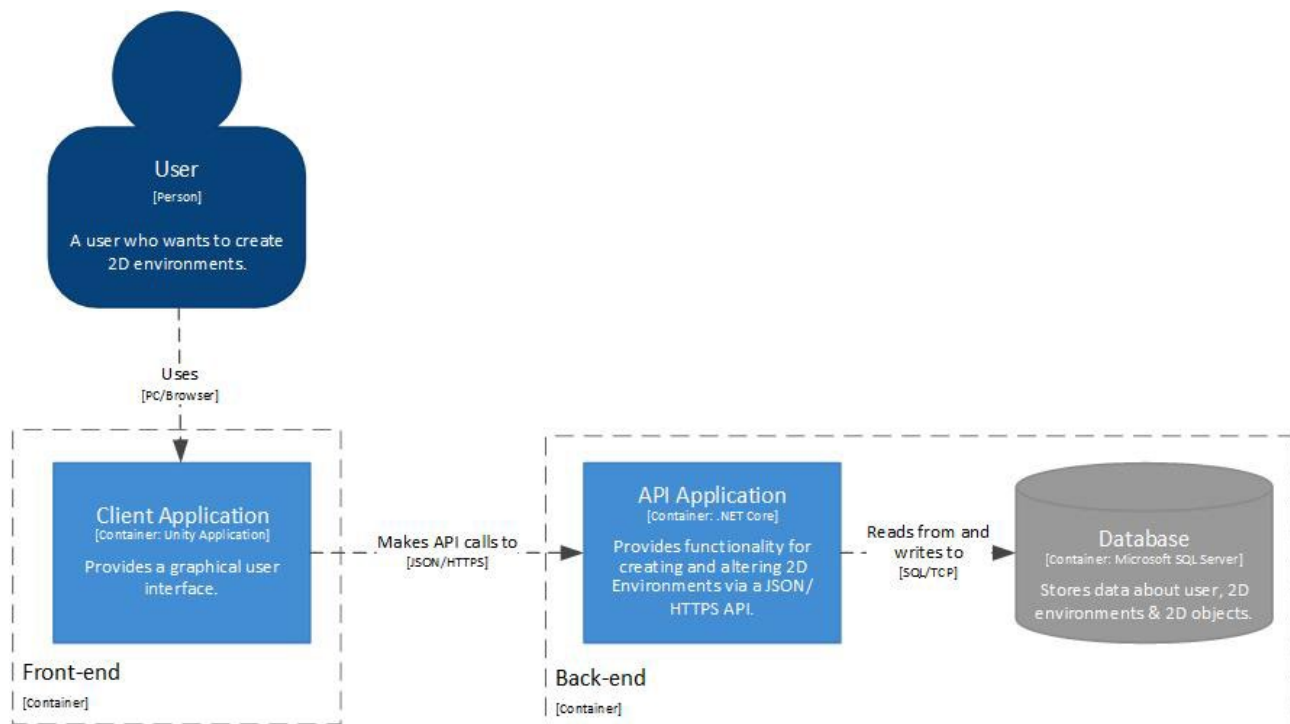
*Figuur 2: Customize feature van Gather.*

Jouw applicatie gaat gebruikers in staat stellen om hun eigen 2D-wereld in te richten op basis van een aantal grafische 2D-objecten die jij voor hen beschikbaar stelt. Data over de 2D-wereld van de gebruiker zal door jouw applicatie online opgeslagen worden zodanig dat gebruikers hun 2D-wereld niet kwijtraken zodra ze de applicatie sluiten.

## Architectuur

Jouw applicatie zal bestaan uit twee grote onderdelen:

- De front-end bestaande uit een applicatie gemaakt met de game engine Unity.
- De back-end bestaande uit een .NET Core API en een Microsoft SQL Server Database.



*Figuur 3: Beoogde architectuur*

De front-end bestaat uit een client-applicatie die de gebruiker een grafische interface aanbiedt om 2D-werelden aan te maken en in te richten met 2D-objecten.

De back-end bestaat uit een API en een database. De API biedt functionaliteit om gegevens over 2D-werelden op te kunnen halen en aan te passen. De database zorgt voor de opslag van data over gebruikers, 2D-werelden en 2D-objecten.

## Functionele eisen

De eisen voor deze applicatie kunnen we onderverdelen aan de hand van de [MOSCoW](#) methode.

### *Must have*

- Als gebruiker wil ik mezelf kunnen registreren op basis van een gebruikersnaam en wachtwoord.
  - Er bestaat nog geen gebruiker met deze gebruikersnaam. De gebruikersnaam is uniek.
  - Wachtwoord moet minimaal 10 karakters lang zijn.
  - Wachtwoord moet minstens 1 lowercase, uppercase, cijfer en niet-alphanumeriek karakter bevatten.
- Als gebruiker wil ik kunnen inloggen op basis van mijn gebruikersnaam en wachtwoord.
  - De gebruiker ziet een foutmelding wanneer de gebruikersnaam of wachtwoord niet correct is.
- Als gebruiker wil ik een nieuwe lege 2D-wereld aan kunnen maken.
  - De gebruiker moet ingelogd zijn.
  - De gebruiker moet een naam invoeren voor de nieuwe 2D-wereld.
  - De naam voor de nieuwe 2D-wereld mag niet identiek zijn aan de naam van een al bestaande 2D-wereld van de gebruiker.
  - De naam voor de nieuwe 2D-wereld is minimaal 1 karakter en maximaal 25 karakters lang.
  - De gebruiker mag niet meer dan 5 eigen 2D-werelden hebben.
  - De nieuwe 2D-wereld wordt opgeslagen.
- Als gebruiker wil ik een overzicht van mijn bestaande 2D-werelden kunnen bekijken.
  - De gebruiker moet ingelogd zijn.
  - Het overzicht toont de naam van de bestaande 2D-werelden van de gebruiker.
- Als gebruiker wil ik één van mijn bestaande 2D-werelden kunnen bekijken.
  - De gebruiker moet ingelogd zijn.
  - 2D-objecten die gekoppeld zijn aan deze 2D-wereld worden correct getoond op basis van de attributen van het 2D-object.
  - De gebruiker kan alleen eigen 2D-werelden bekijken.
- Als gebruiker wil ik een 2D-object aan mijn openstaande 2D-wereld kunnen toevoegen.
  - De gebruiker moet ingelogd zijn.
  - De gebruiker kan kiezen uit minimaal 3 beschikbare 2D-objecten.
  - Het nieuwe 2D-object wordt opgeslagen.
- Als gebruiker wil ik een door mij gemaakte 2D-wereld kunnen verwijderen.
  - De gebruiker moet ingelogd zijn.
  - 2D-objecten die gekoppeld zijn aan deze 2D-wereld worden ook verwijderd.

### *Should have*

- Als gebruiker wil ik de positie, rotatie of schaal van een bestaand 2D-object in een door mij gemaakte 2D-wereld kunnen aanpassen.
  - De gebruiker moet ingelogd zijn.
  - De aanpassing van het 2D-object wordt getoond.
  - De aanpassing van het 2D-object wordt opgeslagen.
- Als gebruiker wil ik een 2D-object in een door mij gemaakte 2D-wereld kunnen verwijderen.

- De gebruiker moet ingelogd zijn.
  - Het 2D-object wordt niet meer getoond.
  - Verwijderen van het 2D-object wordt opgeslagen.
- Als gebruiker wil ik de camera kunnen bewegen terwijl ik een 2D-wereld bekijk zodanig dat ik heel de 2D-wereld kan bekijken.
  - De gebruiker kan de camera naar links, rechts, boven en onder bewegen met de bijbehorende pijltjestoetsen op het toetsenbord.
- Als gebruiker wil ik de grootte van mijn 2D-wereld specificeren in maximale lengte (X) en maximale hoogte (Y) wanneer ik een nieuwe lege 2D-wereld aanmaak.
  - De maximale lengte moet een geheel getal zijn tussen 20 en 200.
  - De maximale hoogte moet een geheel getal zijn tussen 10 en 100.
  - De positie van 2D-objecten in deze 2D-wereld mag zich niet buiten de maximale lengte of maximale hoogte van de 2D-wereld bevinden.
- Als gebruiker wil ik een door mij gemaakte 2D-wereld kunnen delen met een andere gebruiker op basis van hun gebruikersnaam.
  - De gebruiker voert de gebruikersnaam in van de user waarmee ze hun 2D-wereld willen delen.
  - De gebruiker ontvangt geen feedback over de ingegeven gebruikersnaam\*, enkel bevestiging dat de 2D-wereld met deze user nu gedeeld is.
  - Indien de gebruikersnaam bestaat dan kan de gebruiker van dit account de gedeelde 2D-wereld vanaf nu bekijken.
- Als gebruiker wil ik dat een 2D-objecten een animatie afspeelt wanneer het in de 2D-wereld wordt geplaatst.
- Als gebruiker wil ik dat een 2D-object een animatie afspeelt wanneer ik erop klik.

\* Feedback over een ingegeven gebruikersnaam, zoals '*User bestaat niet*', geeft potentiële hackers informatie over welke gebruikersnamen wel of niet bestaan. Informatie delen zoals dit moet zoveel mogelijk beperkt worden terwijl negatieve invloed hiervan op de gebruikerservaring ook beperkt moet worden. Dit kunnen conflicterende belangen zijn.

#### *Nice to have*

- Als gebruiker wil ik dat door mij aangemaakte, aangepaste en verwijderde 2D-objecten pas worden opgeslagen wanneer ik kies om mijn uitgevoerde acties op te slaan in plaats van dat deze onmiddellijk opgeslagen worden.
  - Data over aangemaakte, aangepaste en verwijderde 2D-objecten worden pas opgeslagen wanneer de gebruiker op een Save-knop klikt.

## Niet-functionele eisen

De niet-functionele eisen bevatten eisen die niet rechtstreeks bijdragen aan de beschikbare functionaliteit van je applicatie. Ze beïnvloeden echter wel de kwaliteit van je applicatie.

### *Must have*

- Als developer wil ik dat er unittesten worden opgesteld voor de secure backend zodanig dat de betrouwbaarheid van de code verbeterd wordt.
  - Er zijn unit testen opgesteld voor minstens 3 acceptatiecriteria.
- Als developer wil ik dat er systeemtesten worden opgesteld voor de secure backend zodanig dat de betrouwbaarheid van het systeem verbeterd wordt.
- Als developer wil ik dat de secure backend gedeployed is in cloud zodanig dat deze online beschikbaar is.
- Als developer wil ik dat de code van de Unity applicatie onder version control staat op een git repository in cloud.
- Als developer wil ik dat de code van de secure backend onder version control staat op een git repository in cloud.
- Als developer wil ik dat dat een push naar de git repository van de secure backend zorgt voor het automatisch uitvoeren van de unittesten.
- Als developer wil ik dat een push naar de git repository van de secure backend zorgt voor het automatisch deployen van de API en database naar cloud indien de testen slagen.
- Als developer wil ik dat credentials in cloud op een veilige manier worden behandeld zodanig dat deze niet door personen buiten mijn developer team kunnen bekeken of gebruikt worden.
- Als developer wil ik dat de client applicatie met de API communiceert via een beveiligde HTTPS verbinding zodanig dat niet geautoriseerde personen of applicaties de verstuurd berichten niet kunnen lezen of aanpassen.
- Als gebruiker wil ik dat mijn gegevens beschermd worden tegen SQL injection.

### *Should have*

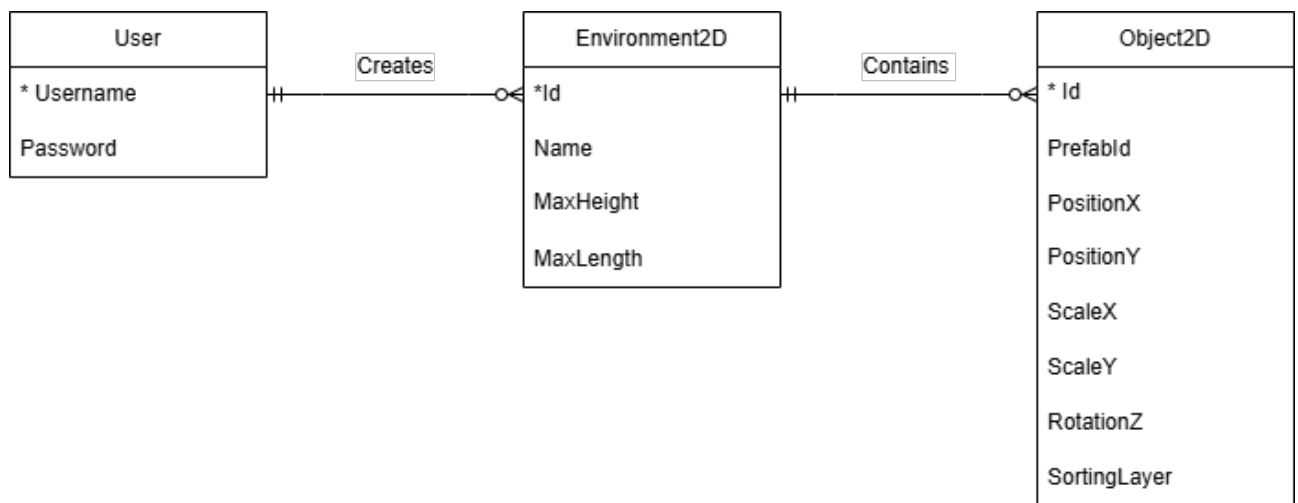
- Als developer wil ik dat er unittesten worden opgesteld voor de Unity applicatie zodanig dat de betrouwbaarheid van de code verbeterd wordt.

### *Nice to have*

- Als gebruiker wil ik dat de Unity applicatie beschikbaar is via browser als een webapplicatie zodanig dat ik deze applicatie niet op mijn computer hoeft te installeren.
- Als developer wil ik dat er systeemtesten worden opgesteld voor de Unity applicatie zodanig dat ik weet dat het systeem correct werkt.
- Als developer wil ik dat dat een push naar git repository van de Unity applicatie zorgt voor het automatisch uitvoeren van de unittesten.
- Als developer wil ik dat een push naar de git repository van de Unity applicatie zorgt voor het automatisch deployen van de API en database naar cloud indien de testen slagen.
- Als developer wil ik dat de Unity applicatie automatisch de token vernieuwd bij de secure backend wanneer de token verlopen is zodanig dat de gebruiker niet opnieuw in moet loggen wanneer de token verlopen is.

## Data model

Onderstaand data model dient als startpunt en verduidelijking van de opdracht. Je bent vrij om het data model aan te passen en uit te breiden om aan de requirements van de opdracht te voldoen.



*Figuur 4: ERD*



## Afbeeldingen

Voor deze opdracht ben je vrij om zelf te kiezen welke afbeeldingen je wenst te gebruiken. Je kan online verschillende gratis 2D asset packs vinden waarvan je de afbeeldingen kan gebruiken. Hieronder enkele voorbeelden:

- <https://kenney.nl/assets/category:2D>
  - <https://kenney.nl/assets/boardgame-pack>
  - <https://kenney.nl/assets/rpg-urban-pack>
  - <https://kenney.nl/assets/roqueline-indoors>
- [Open game art: 2D](#)
  - <https://opengameart.org/content/a-blocky-dungeon>
  - <https://opengameart.org/content/platformer-art-deluxe>
  - <https://opengameart.org/content/zelda-like-tilesets-and-sprites>
- [Unity asset store: 2D & free](#)