

# Wout Van Cleemput (202181098)

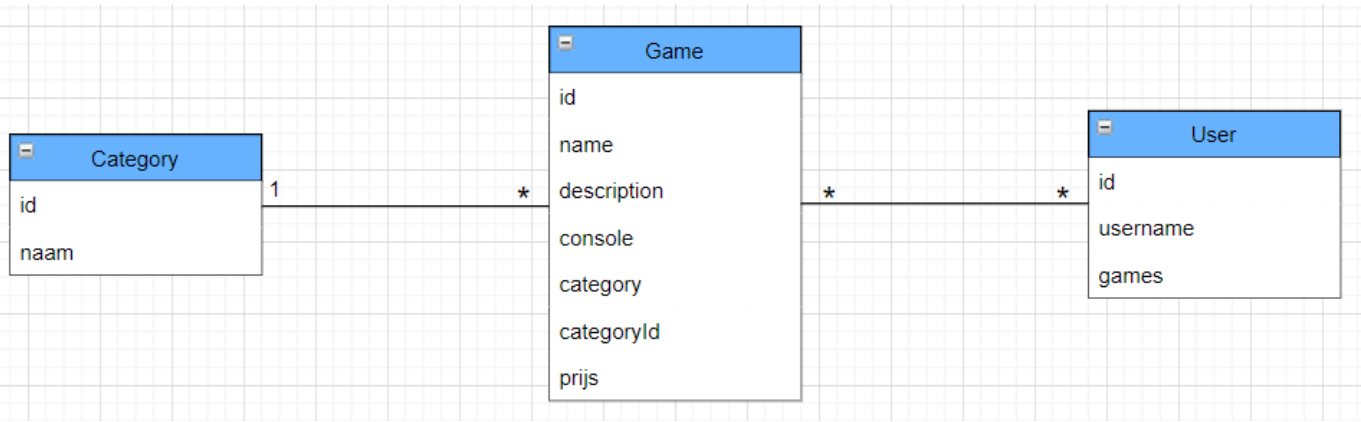
- ☒ Web Services: [GITHUB URL](#)
  - [GitHub repository](#)
  - [Online versie](#)

## Logingegevens

- Gebruikersnaam/e-mailadres: e2e-testing@gameapp.be
- Wachtwoord: Test1234!

## Projectbeschrijving

Mijn project gaat over een website dat games verkoopt.




## Screenshots

Voeg enkele (nuttige!) screenshots toe die tonen wat de app doet.

The screenshot shows the GAMEONLINE website. The header has the logo and navigation links: Home, Winkelkar, Wout Van Cleemput, and Log Out. The main content area shows search results for 'Among Us'. On the left, there are filters for 'Categorie' (All, Action, Adventure, Party, Platformer, RPG, Sandbox, Shooter, Sports) and 'Console' (Selecteer een console). The search results show 'Among Us' with a description, platform (PC), party size, and price (€0.00). Below it, 'Assassin's Creed Valhalla' is partially visible.

 Zoekbalk

 GAMEONLINE

[Home](#)[!\[\]\(41f056cec2ea9da453f2d4b9c95284fa\_img.jpg\)](#)[Winkelkar](#)[!\[\]\(12a79dbbf604ba7c31ab212ea78f3d62\_img.jpg\)](#)

[Wout Van Cleemput !\[\]\(a4c0e6a71283188729f66a69a953f4b5\_img.jpg\)](#)[Log Out](#)[!\[\]\(8bcb2da1a49862cdfda785db9124ba26\_img.jpg\)](#)

Taal wijzigen

**Naam**

**Foto URL**

**Beschrijving**

**Console**

**Categorie**

Selecteer een categorie

**Categorie ID**

**Prijs**

Spel Toevoegen

GAMEONLINE

Home

Winkelkar

Wout Van Cleemput

Log Out

Change language

Name

Image URL

Description

Console

Category

Category ID

Price

Add Game

Name is required.

Image URL

Image URL is required.

Description

Description is required.

Console

Console is required.

Category

Category is required.

Category ID

Category ID is required.

Price

Price is required.

Add Game

3 / 7

[laal wijzigen](#)

**Naam**

**Foto URL**

**Beschrijving**

**Console**

**Categorie**  

Selecteer een categorie ▼

**Categorie is verplicht.**

**Categorie ID**

**Category ID moet een getal zijn.**

**Prijs**

**Prijs moet een getal zijn.**

**Validation Messages**  
Dit spel bestaat al in de database.

## API calls

### Categories

- `GET /api/categories`: alle categorieën ophalen
- `GET /api/categories/:id`: categorie met een bepaald id ophalen
- `POST /api/categories`: een nieuwe categorie toevoegen
- `PUT /api/categories/:id`: een bestaande categorie bijwerken
- `DELETE /api/categories/:id`: een categorie verwijderen

### Games

- `GET /api/games`: alle games ophalen
- `GET /api/games/:id`: game met een bepaald id ophalen
- `POST /api/games`: een nieuwe game toevoegen
- `PUT /api/games/:id`: een bestaande game bijwerken

- **DELETE /api/games/:id**: een game verwijderen

## Gebruikers

- **GET /api/users**: alle gebruikers ophalen
- **GET /api/users/:id**: gebruiker met een bepaald ID ophalen
- **POST /api/users**: een nieuwe gebruiker toevoegen
- **PUT /api/users/:id**: een bestaande gebruiker bijwerken
- **DELETE /api/users/:id**: een gebruiker verwijderen
- **GET /api/users/:id/games**: Haal de games op die aan een gebruiker zijn gekoppeld.

## Behaalde minimumvereisten

### Web Services

- **data laag**
  - ☒ voldoende complex (meer dan één tabel, 2 een-op-veel of veel-op-veel relaties)
  - ☒ één module beheert de connectie + connectie wordt gesloten bij sluiten server
  - ☒ heeft migraties - indien van toepassing
  - ☒ heeft seeds
- **repository laag**
  - ☒ definieert één repository per entiteit (niet voor tussentabellen) - indien van toepassing
  - ☒ mapt OO-rijke data naar relationele tabellen en vice versa - indien van toepassing
- **servicelaag met een zekere complexiteit**
  - ☒ bevat alle domeinlogica
  - ☒ bevat geen SQL-queries of databank-gerelateerde code
- **REST-laag**
  - ☒ meerdere routes met invoervalidatie
  - ☒ degelijke foutboodschappen
  - ☒ volgt de conventies van een RESTful API
  - ☒ bevat geen domeinlogica
  - ☒ geen API calls voor entiteiten die geen zin hebben zonder hun ouder (bvb tussentabellen)
  - ☐ degelijke autorisatie/authenticatie op alle routes
- **algemeen**
  - ☒ er is een minimum aan logging voorzien
  - ☒ een aantal niet-triviale integratietesten (min. 1 controller  $\geq 80\%$  coverage)
  - ☒ minstens één extra technologie
  - ☒ maakt gebruik van de laatste ES-features (async/await, object destructuring, spread operator...)
  - ☒ duidelijke en volledige README.md
  - ☒ volledig en tijdig ingediend dossier en voldoende commits (de meeste commits zijn nog van vorig jaar <https://github.com/Web-IV/2223-webservices-WoutVC>)

# Projectstructuur

## Web Services

```
C:. | createServer.js | index.js |
+---core | logging.js | serviceError.js |
+---data | | index.js | |
| +---migrations | | 202212201521_createGameTable.js | | 202304251635_createCategoryTable.js | |
202312221455_createUserTable | |
| ---seeds | 202105251900_resetDb.js | 202212201527_games.js | 202304251636_category.js |
202312221455_users | +---models | Category.js | Game.js | sequelize.js | User.js |
+---repository | category.js | game.js | user.js |
+---rest | index.js | _categories.js | _games.js | _health.js | _validation.js | _users.js |
+---service | category.js | game.js | health.js | user.js | ---tests games.spec.js
```

## Extra technologie

### Web Services

Voor web services heb ik met eslint gewerkt en geprobeert met sequelize te werken maar enkel eslint werkt.

<https://www.npmjs.com/package/eslint> (<https://sequelize.org/docs/v6/getting-started/>)

## Testresultaten

### Web Services

Het test of er games, users en categories toegevoegd kunnen worden en verwijderd. Er wordt ook getest als je een game kunt krijgen met alleen een id te geven, en of het mogelijk is om een game bij te werken (updaten).

Voor de users en categorieën wordt er getest als je alle users/categorieën kan zien en dat je een user/categorie kan zien met een bepaalde id.

```
PASS src/___tests___/games.spec.js (7.991 s)
  ✓ GET /api/games should return all games (460 ms)
  ✓ PUT /api/games/:id should update an existing game (430 ms)
  ✓ GET /api/games/:id should return a single game by ID (270 ms)
  ✓ DELETE /api/games/:id should delete an existing game (726 ms)
  ✓ GET /api/users should return all users (702 ms)
  ✓ GET /api/users/:id should return a single user by ID (117 ms)
  ✓ GET /api/category should return all categories (163 ms)
  ✓ GET /api/category/:id should return a single category by ID (337 ms)
  ✓ DELETE /api/category/:id should delete an existing category (167 ms)

Test Suites: 1 passed, 1 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        8.078 s
Ran all test suites.
```

## Gekende bugs

### Web Services

- Sequalize en swagger werkt niet

## Wat is er verbeterd/aangepast?

### Web Services

- Geprobeert om swagger te implementeren
- 8 testen toegevoegd
- De User entiteit toegevoegd