



Functionele gebruikershandleiding

QURIO

JAN-PETER DHALLÉ EN AMBER VRANCKX

Inhoud

Inleiding	2
1 Authenticatie	3
1.1 Log in	3
1.2 Registreren	4
1.3 Wachtwoord vergeten	4
2 Quiz beheren	5
2.1 Quiz aanmaken	5
2.2 Quiz bewerken	6
2.3 Quiz klonen	6
2.4 Quiz archiveren	7
3 Vragen beheren	7
3.1 Vraag aanmaken	7
3.2 Vraag bewerken	8
3.3 Vraag klonen	8
3.4 Vraag verwijderen	9
4 Quiz spelen	9
4.1 Quiz starten	9
4.2 Quiz spelen	10
4.3 Resultaten voor een quiz bekijken	11
5 Quiz zoeken	12
6 Gebruiker zoeken	13
7 Profiel bekijken	13
8 Gebruikers uitnodigen voor een quiz	14
9 Feedback	14
9.1 Feedback geven	15
9.2 Feedback opvragen	15
10 Gebruikers beheren	16

Inleiding

Dit document bevat de genomen beslissen die wij gemaakt hebben tijdens de analyse en doorheen het ontwikkelproces van onze stageopdracht. Verder bevat het uitleg hoe de functionaliteiten achterliggend werken.

1 Authenticatie

1.1 Log in

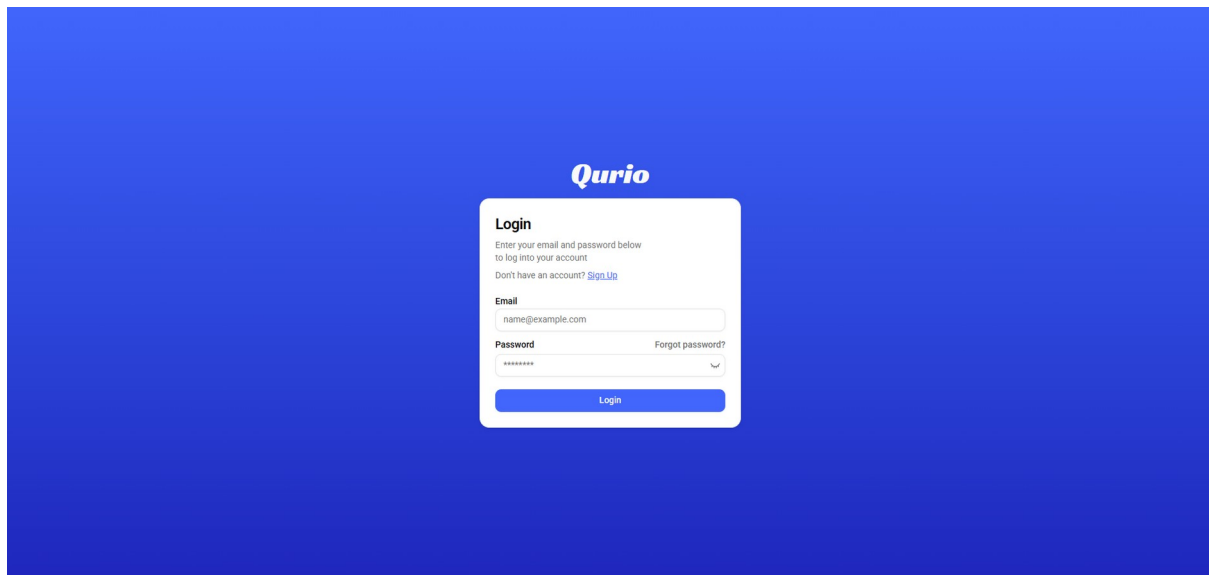
Een e-mail en wachtwoord zijn nodig voor in te loggen. Bij het inloggen wordt er eerst gecontroleerd of er een gebruiker met overeenkomstige e-mail bestaat. Hierna wordt het wachtwoord nagekeken en wordt het proces beëindigd, als het wachtwoord niet overeenkomt.

Als het wachtwoord juist is controleren we of de gebruiker al een *RefreshToken* heeft. Als dit niet het geval is, wordt er een nieuwe aangemaakt die voor 1 dag geldig is (deze waarde is aanpasbaar in de *properties*). De *RefreshToken* wordt ook opgeslagen in de database.

Hierna wordt de JWT-token gegenereerd. In deze token staat onder andere de rol (*scope*), naam en e-mail van de gebruiker. De JWT-token is geldig voor 5 minuten, ook dit is een property die aanpasbaar is.

Na de generatie van de *RefreshToken* en de JWT-token worden deze beiden meegestuurd in het antwoord naar de frontend.

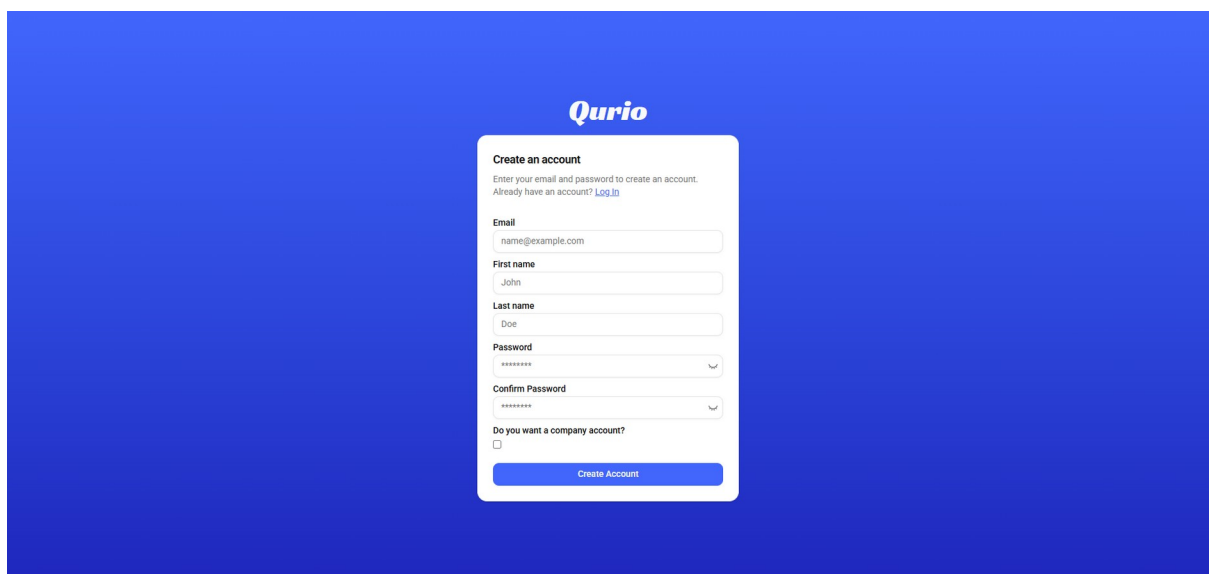
In de frontend wordt dit antwoord opgeslagen in een cookie. Ten slotte wordt deze data samen met de gebruikersgegevens ook opgeslagen in de *AuthContext* door middel van een *useEffect* in de site layout. Op deze manier kunnen we de data makkelijk opvragen binnen onze applicatie via de *useAuth*-hook.



1.2 Registreren

In de frontend vult de gebruiker alle gegevens in. Hierna wordt eerst gecheckt of er al een gebruiker is met het gegeven e-mailadres. Als dit het geval is stopt het registreren en wordt er een error getoond.

Als de gebruiker nog niet bestaat, wordt er een nieuwe gebruiker aangemaakt en opgeslagen. Als de gebruiker een bedrijf invult, dan wordt dit ook gemaakt. Hierna stuurt de backend de CREATED HTTP-code terug en is het proces voltooid.



1.3 Wachtwoord vergeten

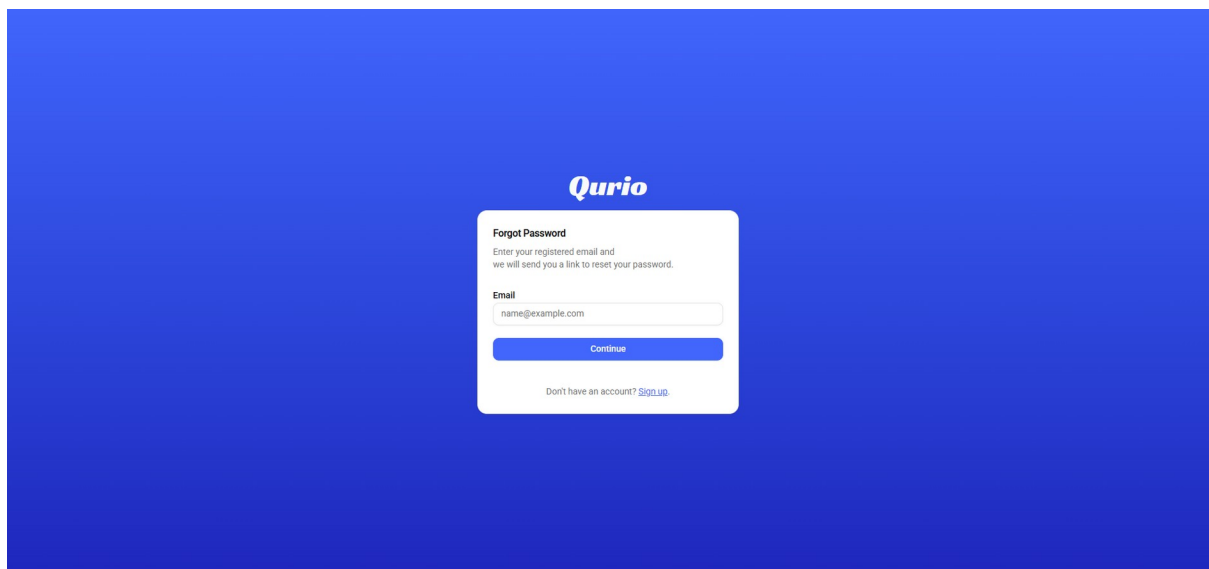
Een gebruiker kan een wachtwoord reset aanvragen door zijn e-mail in te geven bij de *forgot-password*-pagina. Vervolgens wordt er gecontroleerd of er een gebruiker bestaat met deze email en wordt er voor deze gebruiker een *PasswordResetToken* aangemaakt en opgeslagen in de database. Deze token is standaard een dag geldig, maar dit is aanpasbaar via een property.

Na het aanmaken van deze token verstuurt het systeem een e-mail met een knop. Deze knop linkt naar de *password-reset*-pagina, met de token in de URL.

Vervolgens wordt eerst gecontroleerd of de token bestaat en geldig is. Als dit het geval is kan de gebruiker een nieuw wachtwoord ingeven en dit indienen.

Na het indienen wordt er gecontroleerd of het token geldig is en vervolgens wordt het nieuwe wachtwoord gecodeerd met *bcrypt*. Ten slotte wordt de gebruiker geüpdatet in de database met het nieuwe wachtwoord.

De gebruiker kan nu inloggen met het nieuwe wachtwoord.



Qurio

Forgot Password
Enter your registered email and we will send you a link to reset your password.

Email

[Continue](#)

Don't have an account? [Sign up](#)

2 Quiz beheren

Elke ingelogde gebruiker heeft de mogelijkheid om zijn quizen te beheren. Hieronder wordt het aanmaken, bewerken, klonen en verwijderen van een quiz in detail besproken.

2.1 Quiz aanmaken

Elke ingelogde gebruiker kan een quiz aanmaken. Om een quiz aan te maken moet de naam, beschrijving, moeilijkheidsgraad, en één of meerdere categorieën en vragen verplicht ingevuld worden. De gebruiker kan de vragen uit de tabel selecteren.

De gebruiker kan er verder voor kiezen om een tijdslimiet, foto URL en een datum voor het uitbrengen van de score in te vullen. Bovendien zijn er enkele schakelaars, waarvan het berekenen van de score automatisch aan staat. Hier kan de gebruiker aangeven of de quiz wordt gepubliceerd, of de quiz zichtbaar wordt gemaakt en of de gebruiker onmiddellijk feedback krijgt.

Een bedrijfsklant, bedrijfsadministrator en algemene administrator kunnen bovendien een prijs toekennen aan de quiz.

Wanneer de gebruiker de vragen heeft geselecteerd komen deze onderaan de pagina te staan. Hier kan de gebruiker de vragen eventueel ordenen of kan hij er een ander gewicht voor de score toekennen.

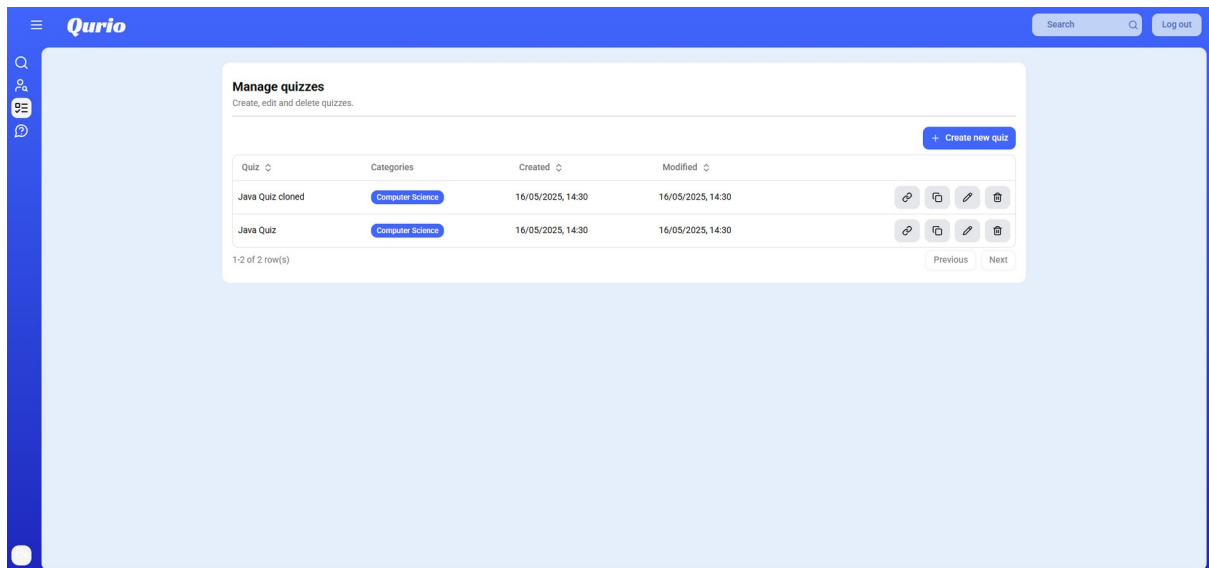
2.2 Quiz bewerken

Een gebruiker kan alleen een quiz bewerken als de quiz nog niet gepubliceerd is. Als de quiz gepubliceerd is kan het namelijk gespeeld worden en ontstaan er antwoorden van gebruikers die gelinkt zijn aan de vragen van de quiz. Als de quiz dan nog verder gewijzigd zou kunnen worden, zou dit een soort versiebeheer vereisen, wat momenteel niet voorzien is.

Wanneer een gebruiker een quiz wil bewerken, kan hij dezelfde opties en inputvelden als bij het aanmaken van een quiz wijzigen. Bij het wijzigen van een quiz, worden alle vragen verwijderd in de database en opnieuw aangemaakt zodat er zeker geen dubbele gegevens ontstaan. Hetzelfde gebeurt bij de categorieën.

2.3 Quiz klonen

Omdat het niet meer mogelijk is om na het publiceren van een quiz deze nog te bewerken, hebben we een functionaliteit voor het klonen van quizzes ontworpen. Deze functionaliteit maakt een nieuwe quiz aan in de database met de eigenschappen van de gekloonde quiz. De naam wordt gewijzigd naar de originele naam met “clone” achter. Zo is het meteen duidelijk voor de gebruiker wat de nieuwe en de oude quiz is.



2.4 Quiz archiveren

Een gebruiker heeft ook de mogelijkheid om een quiz te verwijderen. In praktijk wordt deze quiz dan op inactief gezet.

3 Vragen beheren

Net zoals het beheren van quizzes, kan elke ingelogde gebruiker eveneens vragen beheren. Hieronder worden het aanmaken, bewerken, klonen en verwijderen van vragen in detail toegelicht.

3.1 Vraag aanmaken

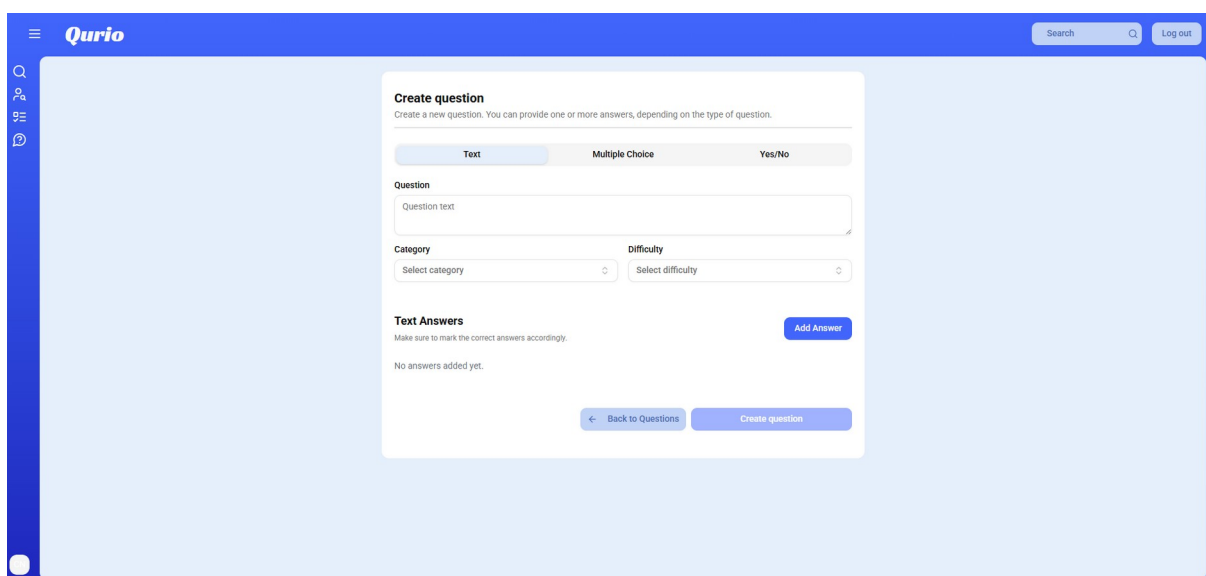
Bij het aanmaken van een vraag moet de gebruiker eerst kiezen welk type vraag hij wil aanmaken. Er is keuze tussen een tekstvraag, een meerkeuzevraag zonder giscorrectie, een meerkeuzevraag met giscorrectie en een ja/nee vraag.

Bij een tekstvraag moet de gebruiker verplicht de vraag, categorie, moeilijkheidsgraad en minstens één antwoordoptie meegeven. De gebruiker mag meerdere antwoordopties meegeven. Daarnaast is er de mogelijkheid om aan te geven of het antwoord hoofdlettergevoelig is of niet. Als het hoofdlettergevoelig is, wordt er

gecontroleerd of het antwoord van de gebruiker exact overeenstemt met het correcte antwoord. In het andere geval wordt enkel de tekst vergeleken.

Bij een meerkeuzevraag moeten dezelfde opties verplicht worden ingevuld als bij een tekstvraag. Het grote verschil zit bij de antwoordopties. Hier kan de gebruiker verschillende opties meegeven en aanduiden of de optie correct is of niet.

Een ja/nee vraag is qua structuur weer identiek hetzelfde als de vorige twee vragen. Hierbij moet de gebruiker ook aanduiden of het antwoord op de vraag “ja” of “nee” is.



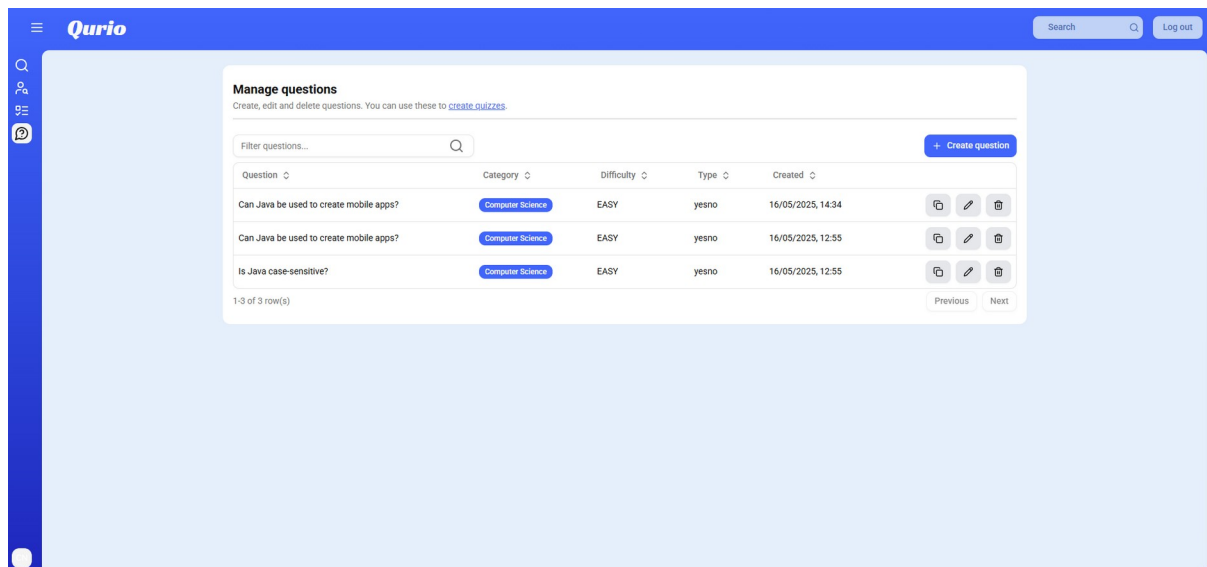
The screenshot shows the 'Create question' interface in the Qurio application. The interface has a blue header with the 'Qurio' logo and a search bar. A sidebar on the left contains navigation icons. The main content area is a light blue box with a white 'Create question' form. The form has three tabs: 'Text' (selected), 'Multiple Choice', and 'Yes/No'. Below the tabs, there is a 'Question' section with a text input field. Below that are 'Category' and 'Difficulty' dropdown menus. The 'Text Answers' section includes a note to mark correct answers and an 'Add Answer' button. At the bottom of the form are 'Back to Questions' and 'Create question' buttons.

3.2 Vraag bewerken

Als er een link is tussen een vraag en een niet-gearchiveerde quiz die wel gepubliceerd is, kan deze vraag niet bewerkt worden. In de andere gevallen kan dit wel en kunnen dezelfde opties als bij het aanmaken van een vraag gewijzigd worden. De antwoorden op de vraag worden dan verwijderd en opnieuw aangemaakt in de database, zodat we zeker geen dubbele gegevens hebben.

3.3 Vraag klonen

Bij het klonen van een vraag, wordt er een nieuwe vraag gecreëerd in de database met dezelfde eigenschappen. Er worden eveneens nieuwe antwoorden aangemaakt in de database op basis van de antwoorden die aan de gekloonde vraag hingen.



3.4 Vraag verwijderen

Een vraag kan niet verwijderd worden als het een link heeft met een niet-gearchiverde quiz die wel gepubliceerd is.

Als een vraag verwijderd wordt, wordt het in de praktijk op inactief gezet in de database. Het wordt dus niet verwijderd uit de database.

4 Quiz spelen

Het spelen van een quiz hebben we opgedeeld in drie grote onderdelen, namelijk de quiz starten, de quiz spelen en het bekijken van de resultaten.

4.1 Quiz starten

Wanneer een gebruiker naar de detailpagina van een quiz gaat kan men de quiz starten. Bij het starten van een quiz wordt er in de database een nieuw *UserQuiz*-object

aangemaakt. Er wordt eerst gecontroleerd of de quiz bestaat en of de gebruiker geïdentificeerd is via een code of account. Bij de creatie van de *UserQuiz*, wordt ook de starttijd opgeslagen. De eindtijd van de *UserQuiz* wordt pas bij het indienen van de quiz ingevuld. In de code gebruiken we dit bovendien om na te gaan of een quiz nog bezig is of is afgesloten.

De frontend gebruikt hierna de code van de *UserQuiz* voor het spelen van de quiz.

4.2 Quiz spelen

Na het starten van een quiz kan een quiz gespeeld worden. Het spelen van een quiz bestaat uit het indienen van antwoorden. Bij het navigeren naar een nieuwe vraag stuurt de frontend het antwoord op de vorige vraag door naar de server. Hierbij wordt er dan een nieuw *UserAnswer* object aangemaakt in de database.

Voor tekstvragen kan een *UserAnswer* niet verwijzen naar een antwoordoptie (*QuestionAnswer*) dus is dit *null*. Voor multiplechoice- en ja/nee-vragen is dit wel ingevuld, maar kan het antwoord zelf leeg zijn.

Vragen kunnen eveneens blanco ingediend worden. Hierbij is zowel het antwoord als de referentie naar een antwoordoptie leeg.

De frontend houdt ook bij welke vragen correct ingediend zijn tijdens het spelen van de quiz en plaatst een groen vinkje naast de vraag als deze ingediend is. Een gebruiker kan teruggaan en zijn antwoord aanpassen. In dit geval wordt de *UserAnswer* geüpdatet.

Bij het indienen van een quiz worden alle niet ingediende antwoorden nog een laatste keer ingediend. Omdat dit het laatste punt is gaat dit tot driemaal geprobeerd worden als het faalt.

Ten slotte wordt de *UserQuiz* afgesloten. Hierbij wordt de eindtijd ingesteld en gebeuren de scoreberekeningen, deze worden ook opgeslagen in de database.

De score voor de hele quiz wordt opgeslagen in de database. Vervolgens wordt er per antwoord opgeslagen of het correct is, hoeveel punten het krijgt, hoeveel punten er

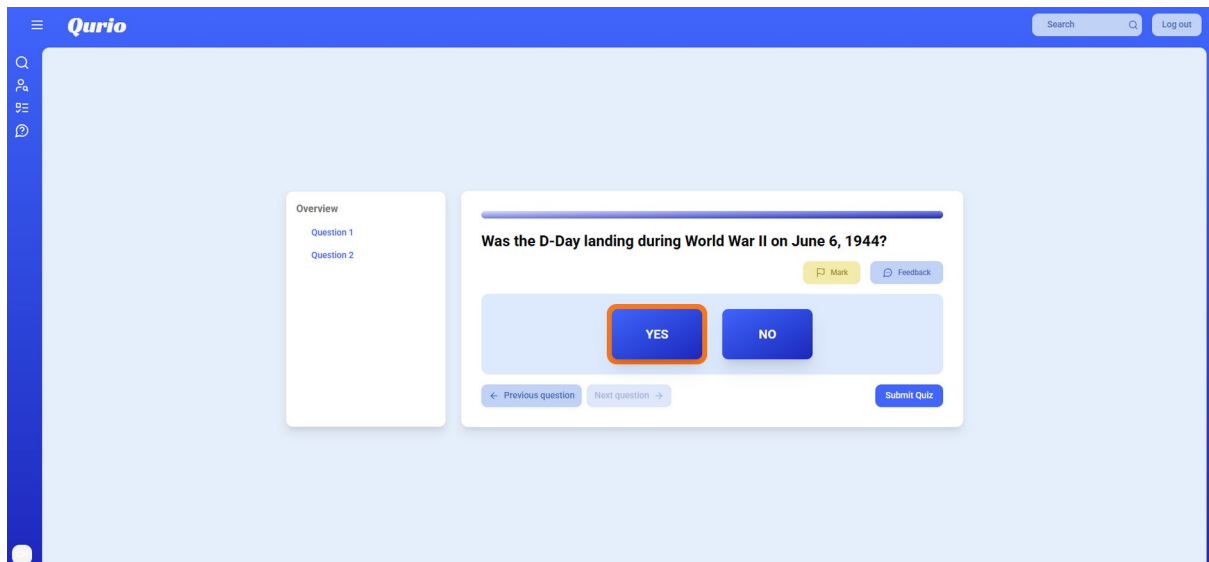
gescoord zijn op de vraag (vraag kan uit meerdere antwoorden bestaan), of de vraag correct is en de standaardscore voor een correcte vraag.

Het berekenen van de score en correctheid van vragen en antwoorden gebeurt via de *Scorer*. Elk vraagtype heeft zijn eigen implementatie voor het berekenen van de score. Voor een ja/nee-vraag wordt per antwoord gecontroleerd of deze klopt. Als deze klopt krijgt de gebruiker de waarde van het gewicht van de vraag als score anders 0. Bij een tekstvraag wordt gecontroleerd of het ingegeven antwoord exact overeenkomt met het juiste antwoord. Het proces zal beëindigd worden als er geen juist antwoord is voor de vraag. Er wordt hierbij ook gekeken naar de optie *caseSensitive* om de controle hoofdlettergevoelig of niet uit te voeren.

De applicatie heeft twee types multiplechoicevragen. De eenvoudigste is het de alles-of-nietsvraag. Hierbij moeten alle antwoorden juist zijn indien dit klopt dan krijgt de gebruiker het gewicht van de vraag als punten, anders 0.

Het laatste vraagtype is de multiplechoicevraag met giscorrectie. Hierbij gaan we de waarde van het gewicht van de vraag gedeeld door het aantal antwoordmogelijkheden steeds optellen of aftrekken van het totaal, afhankelijk van dat het juist of fout is. Hierbij is elk antwoord dat aangeduid is en ook juist is correct, maar ook niet aangeduide antwoorden die fout zijn worden als correct meegeteld.

Ten slotte voor het berekenen van de correctheid van een vraag maken we niet enkel gebruik van correct of incorrect, maar zijn er verschillende mogelijkheden. De mogelijkheden voor het bepalen van de correctheid van een vraag zijn: correct, fout, niet beantwoord, gedeeltelijk correct, onbekend.

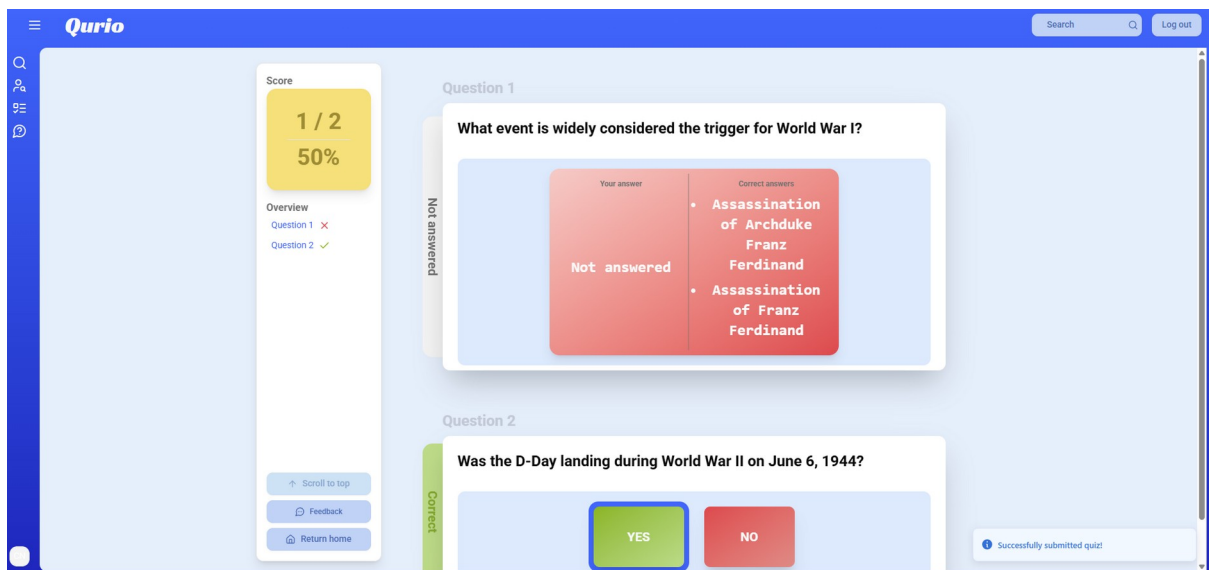


4.3 Resultaten voor een quiz bekijken

Na het spelen van een quiz worden de resultaten getoond. Hiervoor wordt een object dat alle resultaten (die berekend zijn bij het indienen van de quiz) verstuurd naar de frontend en daar weergegeven.

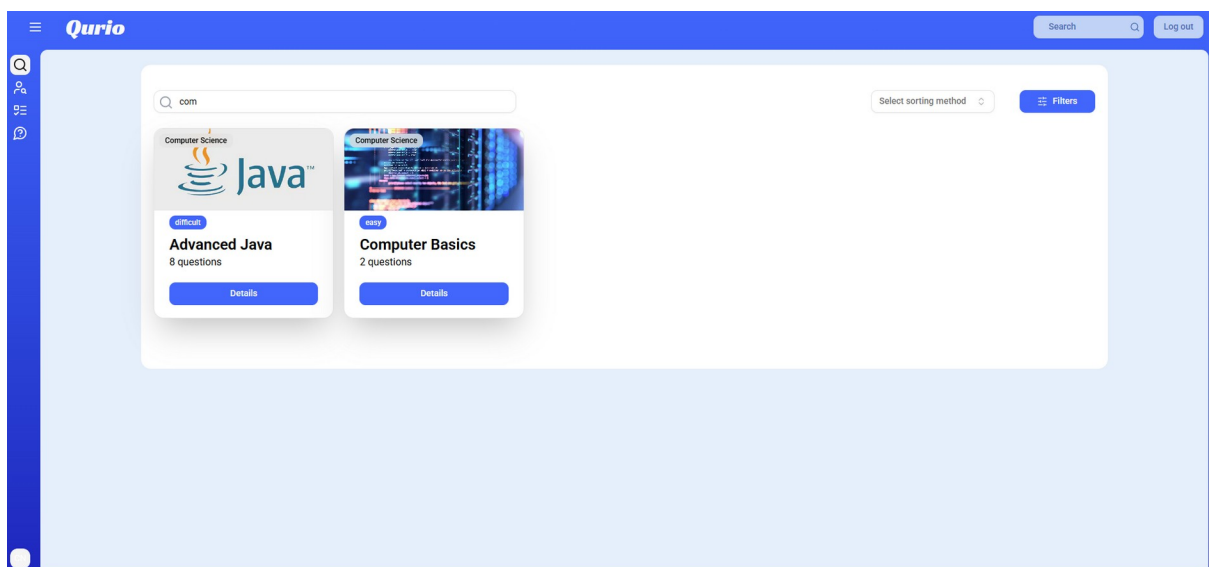
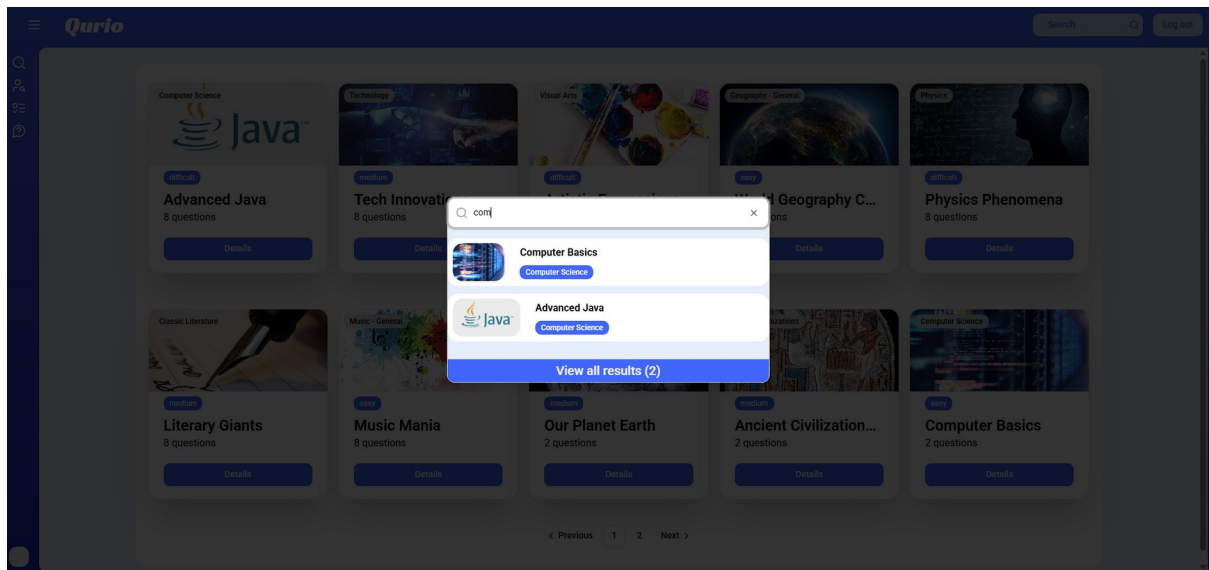
In de backend wordt ook steeds gecontroleerd of de quiz wel beëindigd is, voordat de resultaten berekend worden.

Het is ook nuttig om te vermelden dat de backend een onderscheid maakt tussen DTO's met resultaten en zonder resultaten. Op deze manier vermijden we gevallen waarbij we de oplossingen van de vragen opsturen naar de frontend voordat deze beschikbaar mogen zijn.



5 Quiz zoeken

Een ingelogde gebruiker kan een quiz op twee verschillende manieren zoeken. De eerste manier is via de zoekbalk. Hier kan de gebruiker zoeken op de naam van de quiz, de beschrijving of de naam van een categorie. Vervolgens worden er maximum vijf gevonden resultaten getoond. Wanneer de gebruiker op een resultaat klikt, wordt hij doorverwezen naar de startpagina. Onderaan de lijst met resultaten staat er eveneens een knop die de gebruiker doorverwijst naar de zoekpagina van de quiz. Dit is bovendien de tweede manier om een quiz te zoeken. Op deze pagina kan de gebruiker ook zoeken op tekst zoals in de zoekbalk. Daarnaast kunnen de quizzes gefilterd worden op moeilijkheidsgraad, prijs en categorie. Bovendien bevat deze pagina sorteermethodes. Een gebruiker kan zowel oplopend, als aflopend sorteren op datum of op populariteit.



6 Gebruiker zoeken

Een ingelogde gebruiker heeft de mogelijkheid om andere gebruikers op te zoeken. Dit kan via de voor- of achternaam. Nadien kan het profiel van deze gebruiker geraadpleegd worden.

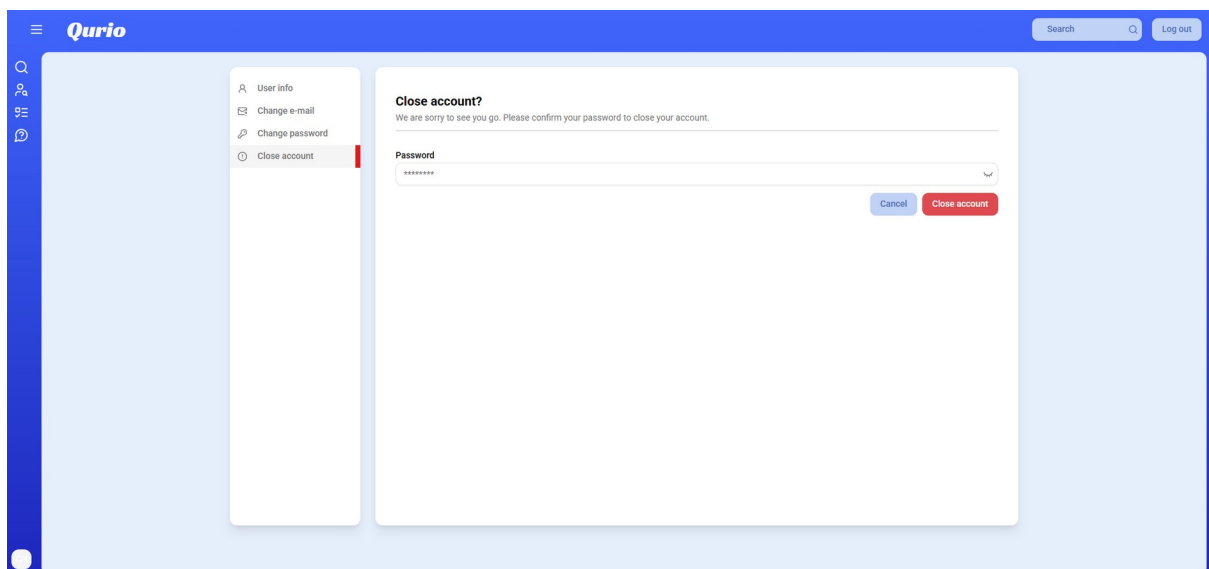
7 Profiel bekijken

Een gebruiker kan eveneens zijn eigen profiel raadplegen. Hierop staat zijn naam, de quizzen die hij heeft gemaakt en de quizzen die hij heeft gespeeld. Bovendien kan hij

naar de instellingen van zijn profiel navigeren. Hier kan de gebruiker meerdere instellingen aanpassen, zoals zijn naam, profielfoto, e-mailadres en wachtwoord.

Om het e-mail en het wachtwoord aan te passen, moet de gebruiker echter zijn wachtwoord meegeven om veiligheidsredenen.

Tot slot kan een gebruiker ervoor kiezen om zijn profiel te verwijderen. Hiervoor is eveneens het wachtwoord vereist. In de backend wordt de gebruiker en al zijn quizzen en vragen dan gearchiveerd.



8 Gebruikers uitnodigen voor een quiz

Het is ook mogelijk voor personen zonder account om een quiz te spelen via een uitnodigingslink. Deze link kan door de maker van een quiz gegenereerd worden. Bij het aanmaken wordt er een *SharedLink*-object met een unieke code opgeslagen in de database.

Na het aanmaken van de gedeelde link met hierin een code als URL-parameter, kan deze gebruikt worden om een quiz te spelen. Dit is mogelijk omdat de pagina's voor het

starten, spelen en bekijken van resultaten voor een quiz ook toegankelijk worden gemaakt als er een geldige *inviteCode* in de URL staat.

9 Feedback

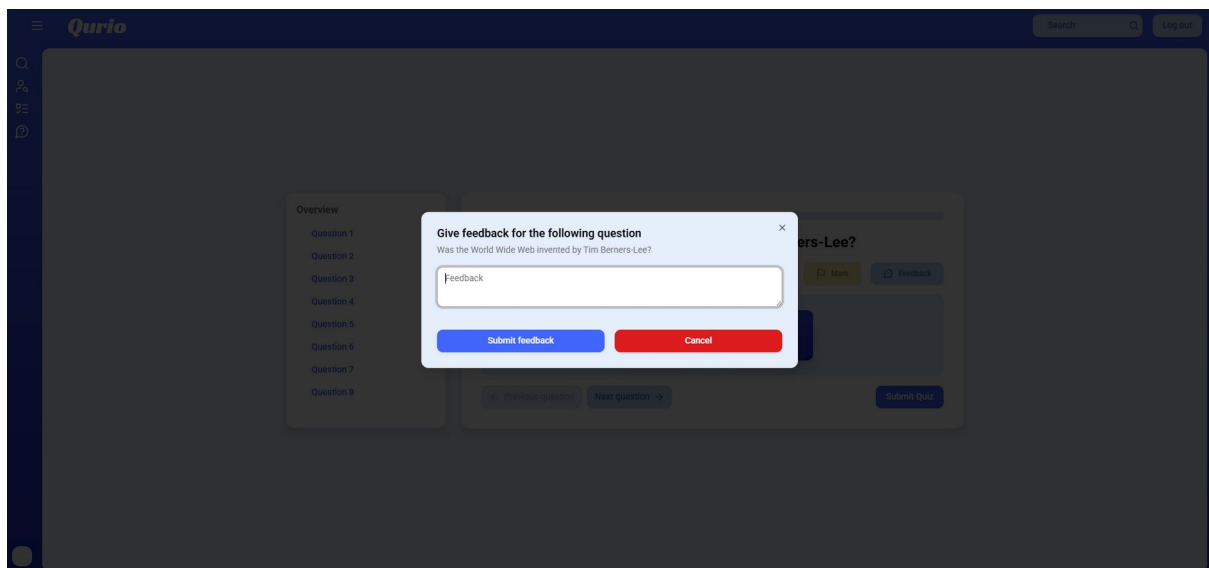
In Qurio is het mogelijk om feedback te geven zodat de eigenaar van de quiz hier rekening mee kan houden.

9.1 Feedback geven

Een gebruiker kan feedback geven op een quiz. Dit kan alleen als deze gebruiker de quiz al gespeeld heeft. Er kan eveneens feedback gegeven worden op een vraag van een quiz. Dit kan dan weer alleen als de gebruiker de quiz effectief aan het spelen is. Het geven van feedback kan eenvoudig door het invullen van het tekstveld op de dialoog dat verschijnt.

We hebben gekozen om de feedback te linken met een *UserQuiz*-object omdat er geen feedback kan bestaan wanneer de gebruiker de quiz nog niet heeft gestart. Er moet dus eerst een *UserQuiz*-object worden aangemaakt voor er feedback kan gegeven worden.

Bovendien hebben we beslist om de feedback eveneens te linken met een *QuizQuestion*-object zodat we zeker zijn dat de vraag aan een quiz hangt.



9.2 Feedback opvragen

De ingelogde gebruiker kan voor al zijn quizen de feedback opvragen. Bovenaan op deze pagina staat eerst informatie over de quiz, zodat het duidelijk is over welke quiz de feedback gaat. Hieronder is alle feedback die over de hele quiz gaat verzameld. Daaropvolgend bevindt zicht de feedback die over een specifieke vraag gaat. Deze zijn gegroepeerd op de vraag.

Wanneer er nieuwe feedback is, krijgt de gebruiker een melding. Zo is het voor de gebruiker steeds duidelijk als er nieuwe feedback is en moet hij dit niet steeds zelf zoeken. De feedback wordt op gelezen gezet wanneer de gebruiker deze heeft geopend.

10 Gebruikers beheren

Een administrator heeft toegang tot een pagina waarop hij alle gebruikers van de applicatie kan bekijken. De administrator kan een gebruiker zoeken op basis van de voor- of achternaam en de e-mail. Hij kan ook filteren op de actieve gebruikers of de inactieve gebruikers.

De administrator kan ervoor kiezen om een gebruiker te deactiveren. Hierbij worden ook alle quizen en vragen van deze gebruiker op inactief gezet in de database. Deze gebruiker heeft nadien geen mogelijkheid meer om in te loggen. De administrator kan

deze gebruikers terug toegang geven door ze opnieuw te activeren. De quizzen en vragen van de gebruiker blijven echter wel nog gearchiveerd.

11 Categorieën beheren

De administrator heeft de mogelijkheid om de categorieën te beheren. Deze categorieën zijn globaal zodat de gebruikers deze kunnen gebruiken bij het aanmaken van een vraag of een quiz.

Een administrator kan een nieuwe categorie aanmaken. Hiervoor dient hij een nieuwe unieke naam te kiezen. Meerdere categorieën met dezelfde naam worden niet toegestaan.

Na het aanmaken van een categorie, kan deze enkel verwijderd worden. Er is geen optie om deze te bewerken. Wanneer een categorie verwijderd wordt, wordt deze effectief helemaal uit de database verwijderd. Bij alle quizzen en vragen die deze categorie hadden, wordt deze verwijderd.