

**LUO**

**De Ridder Pieter-Jan**

Multimedia Technology 2018 - 2019

Karel de Grote hogeschool Antwerpen

Inhoud

[1 Concept 2](#_Toc11151864)

[2 Planning 2](#_Toc11151865)

[3 Functionaliteiten 3](#_Toc11151866)

[4 User cases 4](#_Toc11151867)

[Case 1 4](#_Toc11151868)

[Case 2 4](#_Toc11151869)

[5 Branding 5](#_Toc11151870)

[Logo 5](#_Toc11151871)

[Kleuren 5](#_Toc11151872)

[5](#_Toc11151873)

[6 Technologieën 6](#_Toc11151874)

[Vue 6](#_Toc11151875)

[Firestore 6](#_Toc11151876)

[Vue Router 7](#_Toc11151877)

[Vuex 8](#_Toc11151878)

# 1 Concept

Luo is een leuke frisse quiz applicatie die aanleunt tegen de nieuwe PWA technologie. Iedereen heeft de mogelijkheid om een quiz aan te maken en te delen op het overzicht zodat iedereen deze kan spelen. Ik zelf hou mij bezig met natuurfotografie en vind het leuk om hier continue over bij te leren. Om dit process wat eenvoudiger te maken kwam ik op het idee om de kennis die ik heb vergaard in een quiz vorm te steken zodat ik deze dagelijks terug kan spelen om zo mijn kennis over dat bepaald onderwerp op te frissen en te verbeteren. Ik heb dus geen bepaalde doelgroep waar ik mij specifief op focus. Iedereen die nieuwigheden wilt bijleren en zijn ervaringen delen met anderen, kan dat doen door gebruik te maken van Luo.

# 2 Planning

Voor mij gaat plannen met niets beter dan een pen en een notietieblokje. Ik vind dit zeer handig om snel feedback op te schrijven die je hebt verkregen van personen rondom je en ideeën die je plots binnen komen te schieten snel op te kunnen schrijven. Zo heb ik mijn brainstorm process, de functionaliteiten en de grote lijnen van de applicatie hoe deze in elkaar komt te zitten verwerkt. Toen ik dit had uitgewerkt heb ik via GitHub board mijn planning daar neergezet in de vorm van issues. Elk issue heeft een eigen nummer bv LUO-022. Wanneer ik commit naar GitHub verwerk ik deze nummer mee in mijn commit. Indien je op een later punt iets wil contoleren kan je via deze nummer in je commit history makkelijk terug vinden wat je toen had gedaan.

# 3 Functionaliteiten

Tijdens het plannen heb ik uiteraard ook genoteerd wat ik kwa features terug wil zien in de applicatie. Hier zie je enkele use cases:

* Iedereen moet in staat zijn om een quiz te spelen.
* Indien je quizzes wenst te maken zal je je moeten registreren.
* Het moet mogelijk zijn om multiple choice quizzes aan te kunnen maken. In de toekomst ga ik er meer voorzien.
* Op het einde van een quiz kan je deze liken.
* Elke gebruiken moet een quiz kunnen reporten indien deze de regels overtreed.
* Quizzes met een QR code kunnen aanmaken.
* QR codes kunnen scannen om dan de bijhorende quiz te kunnen spelen.
* Als gebruiker een categorie kunnen voorstellen.
* Andere gebruikers kunnen volgen zodat je een melding krijgt wanneer deze een nieuwe quiz hebben gemaakt.
* Foto’s en geluid kunnen toevoegen aan elke vraag.

# 4 User cases

Om een beter voorbeeld te geven waar Luo allemaal voor kan gebruikt worden som ik even enkele cases op.

## Case 1

Een werkneemster van een museum wil haar bezoekers een leukere ervaring laten beleven. Ze besluit om haar kennis die ze doorheen de jaren heeft vergaard om te zetten in leuke quizzes. Door de handige QR code functie die in Luo te vinden is kan ze deze codes af printen en op verschillende plaatsen in haar museum ophangen. Hierdoor kunnen bezoekers tijdens hun rondleiding deze QR codes scannen om meer informatie over dat bepaald onderwerp te vergaren en de informatie op een leukere manier verwerken.

## Case 2

We volgen een klas die momenteel in hun examenperiode zitten. Tijdens deze periode worden er telkens 100’tal samenvattingen van de ene student naar de andere verstuurd. Dit is niet altijd even praktisch en de kans bestaat dat er foute informatie in staat waardoor iedereen op het examen wordt gebuisd. Luo kan hierbij helpen! Zo is het mogelijk voor de studenten om hun samenvattingen als quiz in de applicatie te steken. Via de QR code kunnen ze deze makkelijk delen met de rest van de klas. Indien er iemand niet akkoord is met bepaalde informatie die er in staat kunnen ze deze vraag reporten zodat de maker van de quiz op de hoogte wordt gesteld van zijn fout.

# 5 Branding

Voor ik begon aan Luo wou ik dat de applicatie zeer fris en strak overkwam. Vooral veel wit met hier en daar een accent kleur. Ik heb veel designs geprobeerd omdat ik niet echt iets kreeg gemaakt in combinatie met foto’s. In mijn definitieve design maak ik gebruik van cards, ik vind dat het design nu goed tot zijn recht komt.

## Logo

Het logo is ontworpen in de primaire kleuren van Luo. In het logo is subtiel de L, U en O verwerkt.



## Primaire Kleuren

## 

#BA42CC

#09A129

#F21B3F

#7326D6

## Fonts

In de applicatie maak ik gebruik van het font ‘**Montserrat**’.

De volgende lettergroottes worden gebruikt. (als rem in de applicatie)

**22px 24px 21px 16px 13px 10px**

# 6 Technologieën

Voor Luo heb ik gebruik gemaakt van het front-end framework Vue in combinatie met Vue Router, Firestore en Vuex. Ik heb Firestore gebruikt als database omdat ik voornamelijk met front-end zaken bezig ben en mij daar volop op wou focussen.

## Vue

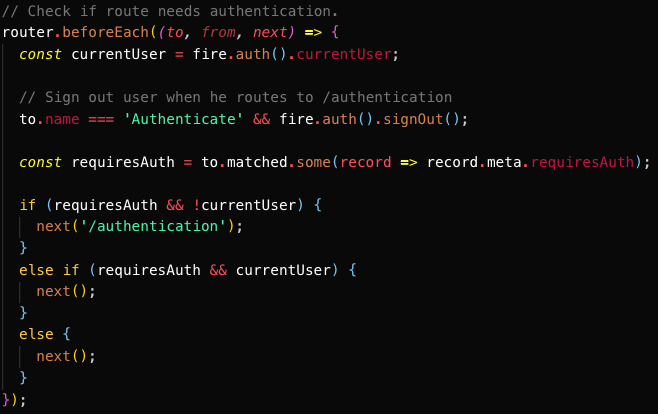
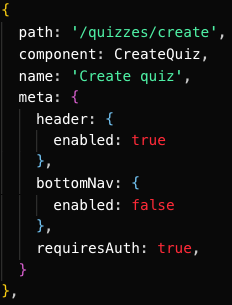
Vue is veruit mijn favoriete framework om applicaties mee te ontwikkelen. Ik heb lang gebruik gemaakt van React maar heb mij uiteindelijk voorgenomen om Vue ook eens onder de loep te nemen. Ik gebruik het nu bijna voor elk project. Daarom was het voor mij geen moeilijke keuze om te beslissen wat ik ging gebruiken voor Luo. Ik ben hier het meest vertrouwd in.

## Firestore

Ik heb dit al enkele keren gebruikt en vind het heel fijn om mee te werken. Het leukste aan Firestore vind ik de real time updates, je hebt toegang tot de onSnapshot() functie die continue gaat “luisteren“ naar updates. Door het reactivity system van Vue zal je applicatie dus ook live worden bijgewerkt. Naast Firestore heb je ook de mogelijkheid om gebruik te maken van cloud functions, dit zijn functies die op de servers van Firebase zelf worden uitgevoerd. Ik gebruik deze om bepaalde calculaties te doen indien er een waarde in Firestore veranderd. Zo zal de variabele likes automatisch verhogen indien er een user id wordt opgeslagen in de likedBy array van een quiz.

## Vue Router

Een package waarmee je routing kan toevoegen in je applicatie. In de Route.js file kan je routes specificeren waaraan je aan elke route een component kan hangen dat op zijn manier dient als root component. In deze file controleer ik ook of de gebruiker via Firebase is ingelogd of niet. Dit is nodig omdat er enkele routes moeten worden beveiligd tegen gebruikers die niet ingelogd zijn zoals quizzes aanmaken, etc.



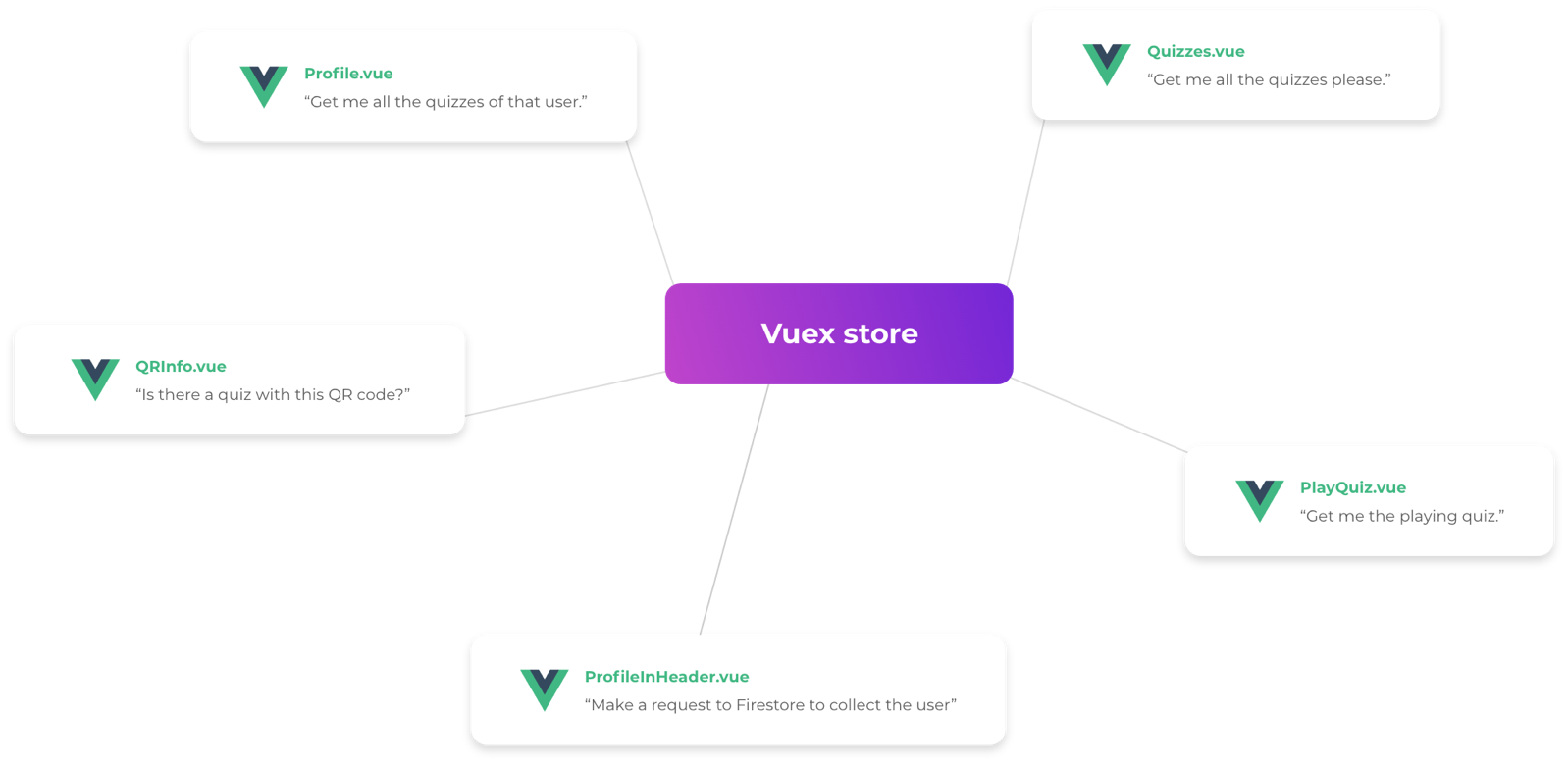
Figuur 1

Figuur 2

Op *Figuur 1* zie je een voorbeeld van een route in Route.js die naar de pagina gaat om een quiz te kunnen aanmaken. Zoals je op dit figuur kan zien staat er in de meta tag een key genaamd “requiresAuth”. In *Figuur 2* Hebben we dan toegang tot deze meta variabele en kunnen we beslissen of de gebruiker hier al dan niet toegang tot heeft.

## Vuex

Ook gemaakt door de ontwikkelaars van Vue en komt ook standaard bij applicaties die ik opstart. Dit is een zeer handige package die er voor zorgt dat je een globale state ter beschikking hebt, ook wel de store genoemd. Deze store is de plaats waar ik al mijn fetch request naar Firestore heb geplaatst. Doordat deze data in de store is opgeslagen, heeft elk component dit ter beschikking.



Netfliy

# 8 Usability tests

Ik heb de applicatie door enkele personen laten testen en daar zijn de volgende resultaten uit gekomen.

* Wanneer je een quiz aan het spelen bent en je zit met een toestel dat niet zo hoog is wordt alles op elkaar gepropt.
  + Opl: Ik heb de foto bovenaan iets smaller gemaakt in de hoogte en op deze foto een knopje voorzien die er voor zorgt dat de foto hoger wordt wanneer er op wordt geklikt. Ook is het mogelijk om te scrollen door de antwoorden.
* Dit probleem deed zich ook voor op de Profile en QuizEnd page.
  + Opl: Header smaller maken in de hoogte en op de plaatsen waar het kan er voor zorgen dat je kan scrollen.
* Wanneer je een vraag aan het maken bent en je na het toevoegen van je antwoorden op submit drukt ga je naar de vogende stap waar je een correct antwoord moet selecteren. Wanneer je in deze stap zit is het knopje “add answer” nog steeds zichtbaar van de vorige stap. Ik had van verschillende mensen de feedback gekregen dat dit knopje beter weg kan wanneer men een correct antwoord moet selecteren.
* Wanneer je een vraag aan het maken bent en je na het toevoegen van je antwoorden op enter klikt zou je naar de volgende stap moeten gaan waar je een correct antwoord moet selecteren.
  + Opl: Enter click event toevoegen die de volgende stap opent.
* Het modal om een vraag toe te voegen iets robuuster maken. Wanneer we perongeluk naast dit modal klikken zijn we onze vraag kwijt.
  + Opl: Data die de gebruiker heeft ingevuld in de Vuex state opslaan zodat wanneer we perongeluk naast het modal klikken en we drukken terug op “create question” die data terug wordt ingevuld.
* Op de profile page staat een stat die aangeeft hoeveel quizzes een bepaalde gebruiker al heeft gespeeld. Deze informatie is voor velen niet relevant.
  + Opl: Ik heb deze informatie vervangen door een stat die aangeeft hoeveel followers deze gebruiker heeft.

# 9 Timesheet

* 4 april 2019
  + Aanmaken GitHub repository. <https://github.com/PeeJeeDR/luo>
* 5 april 2019
  + Logo ontwerp van LUO.
  + Wireframe + design van splash screen.
* 7 april 2019
  + Wireframing + design van authentication page.
* 10 april 2019
  + Wireframing + design quiz overview.
  + Design van app components.
* 12 april 2019
  + Research doen welke technologieën ik ga gebruiken.
  + Vue project aanmaken.
  + Header component toegevoegd.
* 13 april 2019
  + Firebase toevoegen aan het project.
  + Authentication page toegevoegd.
  + Je kan jezelf registreren op firebase.
* 16 april 2019
  + Routes beveiligen voor niet ingelogde gebruikers.
* 20 april 2019
  + Form validation op het login/registreren formulier.
  + Na het registreren de user opslaan in de database (Firestore).
* 21 april 2019
  + Quizzes table toegevoegd in de database.
  + Quizzes tonen in het overzicht.
* 24 april 2019
  + Speel knop toegevoegd aan de navigatie onderaan.
* 29 april 2019
  + Een test page toegevoegd om enkele acties uit te voeren op de database.
* 30 april 2019
  + Manifest file toegevoegd zodat de app kan geplaatst worden op het startscherm van je telefoon.
* 1 mei 2019
  + Styling navigatie onderaan.
  + Navigatie onderaan in de app toevoegen.
  + Header toevoegen in de applicatie.
  + Quiz genereren in de test api.
* 2 mei 2019
  + Loading state in Vuex toevoegen voor wanneer de quizzes aan het laden zijn.
* 4 mei 2019
  + Populaire en nieuwe quizzes tonen in het overzicht van de app.
* 5 mei 2019
  + Pagina transitions toegevoegd.
  + Categorieën toegevoegd zodat je hier op kan filteren.
* 6 mei 2019
  + De pagina en het modal toegevoegd voor het aanmaken van vragen.
* 7 mei 2019
  + Antwoorden kunnen toevoegen en deze opslaan in Vuex.
  + Quizzes via het formulier toevoegen aan de database.
* 10 mei 2019
  + De titel, omschrijving en public state kunnen kiezen wanneer men een nieuwe quiz aanmaakt.
  + Foto’s kunnen uploaden in de database.
* 12 – 14 en 18 mei 2019
  + Code refactor van modals.
  + Quiz kunnen opslaan met een geselecteerde categorie.
  + Een custom image kunnen toevoegen aan je quiz.
* 19 en 20 mei 2019
  + Wireframing + design van profiel pagina.
  + Informatie van de gebruiker die ingelogd is laten zien op deze pagina.
* 21 mei 2019
  + Mogelijk om je profielfoto aan te passen.
  + Header animeren wanneer we navigeren naar een andere pagina.
* 22 – 23 mei 2019
  + Code refactor van modals.
  + Wanneer je een quiz opslaat in de database komt de id van deze quiz in een array “quizzesMade” van die bepaalde user in de database.
  + Eerste Firestore cloud function toegevoegd die gaat tellen hoeveel elementen er in de array “quizzesMade” zitten en dat resultaat gaat opslaan in een nieuwe variabele.
* 24 mei 2019
  + Quizzes ophalen uit de database aan de hand van de id van de gebruiker en dit resultaat tonen op de profiel pagina.
* 26 mei 2019
  + Wireframing + design van quizzes spelen.
  + Quiz informatie modal toegevoegd in de applicatie.
* 27 mei 2019
  + Quiz kunnen spelen.
  + Einde van een quiz toegevoegd.
* 28 mei 2019
  + De id van de gebruiker opslaan die een quiz heeft gespeeld in de array “playedBy” van die bepaalde quiz. Met een cloud function wordt ook geteld hoeveel elementen deze bevat.
* 29 mei 2019
  + Gebruikers die niet ingelogd zijn en dus ook geen id hebben krijgen er een toegewezen via LocalStorage om de playedBy array van een quiz te kunnen vullen.
  + Gebruikers die niet ingelogd zijn kunnen geen quizzes aanmaken.
  + Quizzes kunnen liken na het spelen van een quiz.
* 30 – 31 mei 2019
  + Valdatie toegevoegd wanneer je een vraag wilt aanmaken.
  + Je kan nu quizzes reporten.
  + Likes tonen van een gebruiker op de profiel pagina.
* 1 juni 2019
  + Vragen tijdens het aanmaken kunnen deleten.
* 2 juni 2019
  + Logout knop in sidebar toegevoegd.
  + Gebruiker kan een categorie voorstellen. Deze komt in de database te zitten.
* 3 – 4 juni 2019
  + Code refactoring.
  + QR code quizzes en scanner toegevoegd.
  + CSS fixes en bug fixing.
  + Optie om quizzes te deleten.
  + Filter quizzes die zijn verwijderd..
* 5 juni 2019
  + Bug fixing.
  + Sample images toevoegen aan de hand van een geselecteerde categorie.
* 6 - 7 juni 2019
  + Css fixes.
  + Notifications
* 8 juni 2019
  + Code refactor.
  + Mogelijk om andere users te volgen.
* 9 juni 2019
  + Offline firestore caching.
  + Bug fixing.
* 10 juni 2019
  + Geluiden toegevoegd aan de applicatie.
* 11 juni 2019
  + Wanneer de gebruiker een quiz heeft aangepast en dan op back drukt vraagt de applicatie of de gebruiker zeker is van zijn keuze.
  + Antwoorden door elkaar schudden wanneer we een quiz aan het spelen zijn.