

Architectuurdocument

Viewpoint

Cyrion van Dongen
Niels Boelens
Ruben Dieleman

1 DOCUMENTATIEHISTORIE

Versie	Wijzigingen	Auteur	Datum
0.1	Opzet Document	Ruben Dieleman	21 Sep 2020
1.0	Hoofdstuk 6 en 7	Wesley Verhagen	2 Nov 2020
1.1	Update 5 a.d.h.v. nieuwe Backlog	Cyrion van Dongen	23 Nov 2020
2.0	Update 6 en 7	Wesley Verhagen	10 Dec 2020
2.1	Update 4 en 5 a.d.h.v. nieuwste versie	Cyrion van Dongen	11 Jan 2021
2.2	Opmaak	Ruben Dieleman	15 Jan 2021

2 INHOUDSOPGAVE

1	DOCUMENTATIEHISTORIE	2
3	INLEIDING	4
4	PROJECTOMSCHRIJVING	5
4.1	STAKEHOLDERS	5
4.2	USER STORIES	5
5	ARCHITECTUUR	7
5.1	KLASSEN DIAGRAM	7
5.2	GLOBALE COMPONENTEN	7
5.3	CONTAINERS	8
5.4	FLOW	8
5.5	FRONT-END COMPONENTEN	9
6	SPECIFICATIE VAN INTERFACES	10
6.1	ENDPOINTS	10
6.2	MODELLEN	11

3 INLEIDING

Dit document biedt een overzicht van de architectuur van de applicatie. Er wordt van een hoog abstract level tot een laag concreet leven aanzichten getoond om verschillende aspecten van het systeem weer te geven.

De beslissingen die over de architectuur van de applicatie staan hier allemaal vastgelegd.

4 PROJECTOMSCHRIJVING

4.1 STAKEHOLDERS

De stakeholders van dit project zijn als volgt:

- Product owner
- Ontwikkelaars
- Eindgebruikers
 - Lezer
 - Gebruiker

De eindgebruikers zijn opgedeeld in twee groepen: de lezer en gebruiker.

De lezer wil een platform waar hij zich kan informeren over een onderwerp via anderen. Hierbij is het doel om zo veel mogelijk van verschillende oogpunten te zien te kunnen krijgen. Dit kan gedaan worden door discussies op te zoeken en te lezen. Zolang er niet is ingelogd, zal een eindgebruiker een lezer zijn.

De gebruiker moet deel kunnen nemen aan discussies en nieuwe discussies kunnen aanmaken. Hierbij kan hij ook andere Gebruikers helpen met hun informatie beter verspreiden door opmerkingen te beoordelen of een eigen opmerking te plaatsen. Hiernaast kan een gebruiker alles wat een lezer ook kan.

4.2 USER STORIES

ID	Actor	Omschrijving
US1	Lezer	Als de actor wil ik een lijst van discussies hebben, zodat ik makkelijk over een onderwerp kan informeren.
US2	Lezer	Als de actor wil ik op trefwoord kunnen zoeken naar discussies, zodat ik niet veel tijd hoeft te besteden aan het zoeken naar het onderwerp van mijn interesse.
US3	Lezer	Als de actor wil ik een aanvul functie tijdens zoeken, zodat het formuleren van het onderwerp makkelijk is.
US4	Software Engineer	Als actor wil ik dat tijdens het zoeken tags worden verwerkt, zodat de gebruiker makkelijker een onderwerp kan vinden dat hem interesseert.
US5	Gebruiker	Als de actor wil ik een discussie kunnen aanmaken, om de mening van anderen daarover te kunnen horen.
US6	Gebruiker	Als actor wil ik een beschrijving aan mijn discussies kunnen toevoegen, zodat ik buiten de titel om nog verder kan ingaan op het onderwerp.
US7	Gebruiker	Als actor wil ik wanneer ik een discussie aanmaak het tags met een onderwerp kunnen geven, zodat die makkelijker te vinden is.
US8	Gebruiker	Als de actor wil ik een bericht kunnen achterlaten, zodat ik mijn inbreng kan geven bij een discussie.
US9	Gebruiker	Als de actor wil ik een reactie kunnen plaatsen op het bericht van een ander, zodat ik me met die persoon verder kan verdiepen in de discussie.

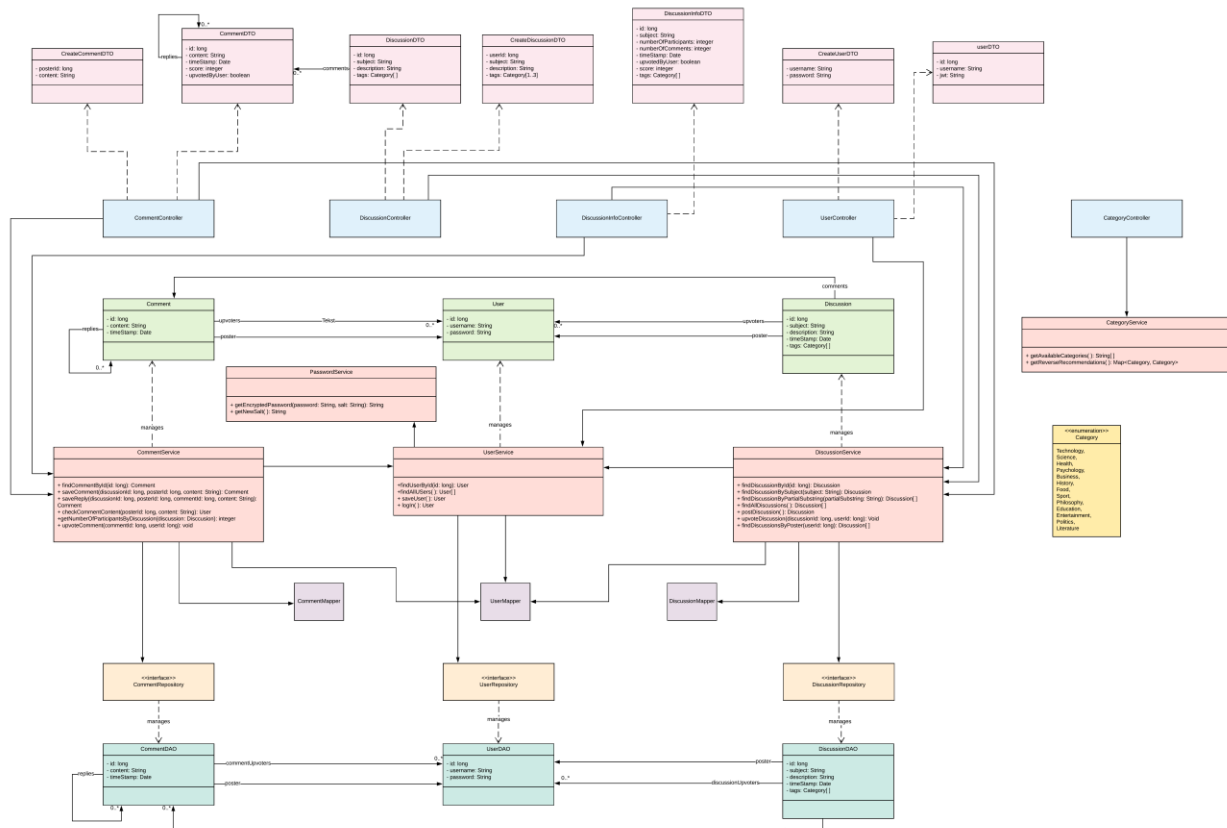
US10	Product Owner	Als actor wil ik berichten en discussies in het overzicht in willekeurige volgorde weergegeven worden, om nieuwe addities aan de discussie genoeg kans te geven opgemerkt te worden.
US11	Lezer	Als de actor wil ik een account aan kunnen maken, zodat ik zelf mijn mening kan delen of een vraag kan stellen.
US12	Gebruiker	Als de actor wil ik kunnen inloggen op mijn account wanneer ik dat niet ben, zodat ik weer kan deelnemen aan discussies.
US14	Gebruiker	Als de actor wil ik in mijn account kunnen zien welke discussies en berichten ik heb gemaakt, zodat ik sneller toegang terug heb tot discussies waar ik aan deelneem.
US15	Product Owner	Als actor wil ik dat gebruikers anoniem blijven, zodat er geen vooroordelen ontstaan op basis van wie iemand is.
US16	Product Owner	<p>Als actor wil ik dat gebruiker discussies geadviseerd worden die tegenstrijdig zijn met hun zoekopdracht, zodat ze eventueel interesse krijgen in nieuwe onderwerpen.</p> <p>Bijvoorbeeld, een gebruiker die naar discussies over sport zoekt krijgt politieke discussies als alternatief aangeboden naast zijn zoekopdracht.</p>
US17	Product Owner	Als de actor wil ik dat de gebruikers een overzicht van willekeurige discussies aangereikt krijgen, zodat kleinere groepen ook een kans hebben hun mening te vormen.
US18	Gebruiker	Als de actor wil ik mijn zoekresultaten kunnen filteren door tags te zoeken, zodat ik zo makkelijker het onderwerp van mijn interesse kan vinden.
US19	Product Owner	Als de actor wil ik dat de applicatie geïnternationaliseerd is, zodat het bruikbaar is voor een groter publiek. Ik wil de applicatie beschikbaar hebben in Engels en Nederlands.
US20	Product Owner	Als de actor wil ik dat het scoresysteem op berichten geïmplementeerd is, zodat het fatsoenlijkheid en constructief zijn bevorderd.
US21	Product Owner	Als de actor wil ik dat het scoresysteem op discussies geïmplementeerd is, zodat het fatsoenlijkheid en constructief zijn bevorderd.

5 ARCHITECTUUR

5.1 KLASSEN DIAGRAM

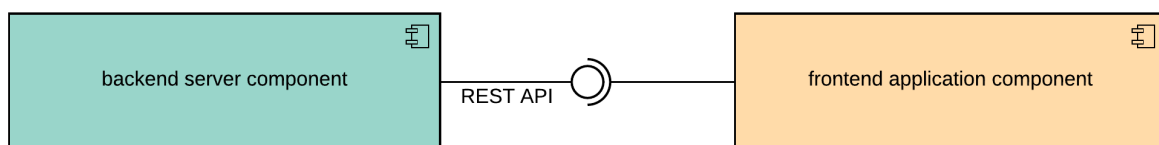
De structuur van de backend is hier weergegeven. Dit diagram is [hier](#) in detail te zien.

De structuur bestaat drie lagen: de API-controllers, de logica services, en de data repositories. Alle lagen gebruiken hun eigen objecten, de DTO's, modellen, en DAO's respectievelijk. De mapper klassen worden gebruikt om deze objecten naar de benodigde versie te kunnen omzetten.

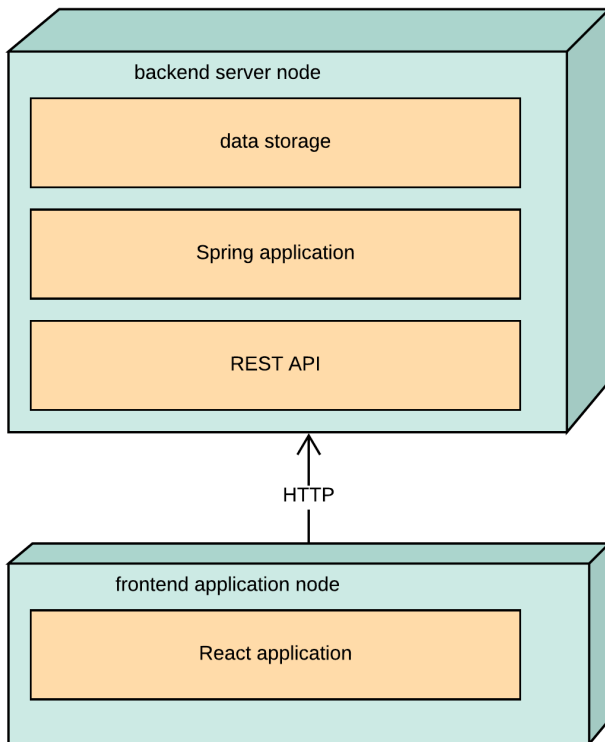


5.2 GLOBALE COMPONENTEN

De gehele applicatie is opgedeeld in deze componenten. De backend biedt een API aan om de communicatie tussen front- en back-end te verzorgen.

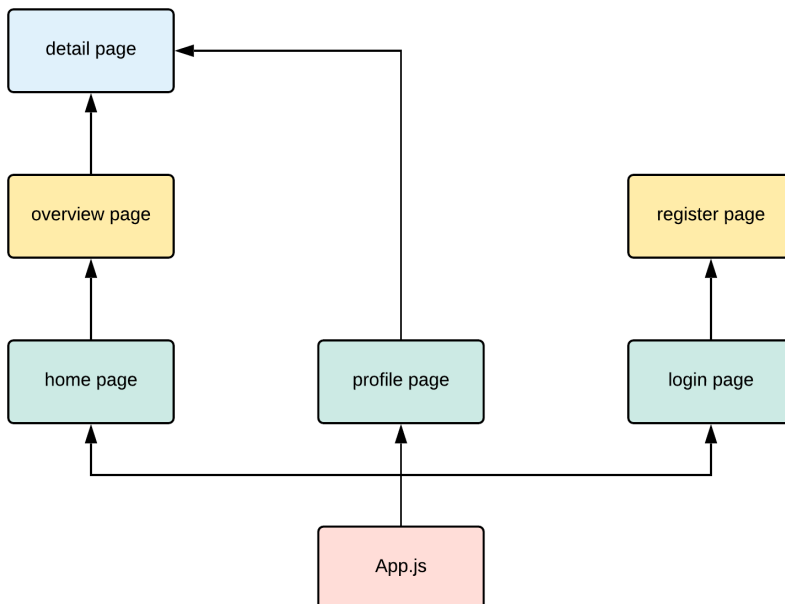


5.3 CONTAINERS



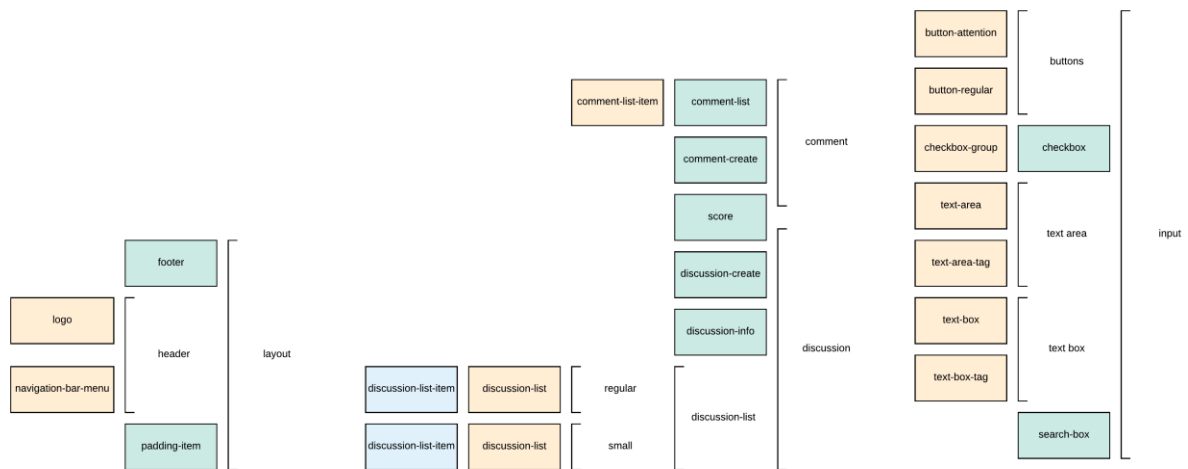
5.4 FLOW

De front-end pagina's volgen de volgende routes:



5.5 FRONT-END COMPONENTEN

De front-end bestaat uit de volgende componenten. Dit diagram kan [hier](#) in detail worden bekeken.



6 SPECIFICATIE VAN INTERFACES

Hier wordt de interface die gebruikt wordt om te communiceren tussen de front- en back-end in detail weergegeven. Alle weergegeven afbeeldingen kunnen in detail [hier](#) worden bekeken, of de API kan [hier](#) live bekeken te worden.

6.1 ENDPOINTS

Klik [hier](#) om in detail te bekijken.

Categories Endpoints for category interaction		▼
GET	/category	Get all available category values
GET	/category/reverse	Get a reverse recommendation for every category
Comments Endpoints for comment interaction		▼
POST	/comment/{commentId}/upvote	Upvote a comment
POST	/comment/{discussionId}	Post a comment on a discussion
POST	/comment/{discussionId}/reply/{commentId}	Post a comment on a discussion
GET	/comment/{userId}	Retrieve a user's comments by it's id
DiscussionInfo Endpoints for info about discussions		▼
GET	/discussion_info	Get info about all existing discussions
GET	/discussion_info/id/{discussionId}	Get info about a specific discussion by it's id
GET	/discussion_info/subject/{subject}	Get info about a specific discussion by it's subject
Discussions Endpoints for discussion interaction		▼
POST	/discussion	Create a new discussion
GET	/discussion/{discussionId}	Get a discussion by it's id
POST	/discussion/{discussionId}/upvote	Upvote a discussion
Users Endpoints for user interaction		▼
GET	/users	Get info about all existing users
POST	/users	Create a new user
GET	/users/{userId}	Get info about a specific user by it's id
POST	/users/login	Log in

6.2 MODELLEN

Klik [hier](#) om in detail te bekijken.

Models	
CommentDTO	<pre>{ content string id integer(\$int64) replies > [...] score integer(\$int32) timeStamp string(\$date-time) upvotedByUser boolean }</pre>
CreateCommentDTO	<pre>{ content string posterId integer(\$int64) }</pre>
CreateDiscussionDTO	<pre>{ description string subject string tags > [...] userId integer(\$int64) }</pre>
CreateUserDTO	<pre>{ password string username string }</pre>
DiscussionDTO	<pre>{ comments > [...] description string id integer(\$int64) subject string tags > [...] }</pre>
DiscussionInfoDTO	<pre>{ id integer(\$int64) numberOfComments integer(\$int32) numberOfParticipants integer(\$int32) score integer(\$int32) subject string tags > [...] timeStamp string(\$date-time) upvotedByUser boolean }</pre>
Pair«string,string»	<pre>{ first string second string }</pre>
ResponseEntity	<pre>{ body > [...] statusCode string Enum: statusCodeValue integer(\$int32) }</pre>
UserDTO	<pre>{ id integer(\$int64) jwt string username string }</pre>