

Furnify Home

Team 02

Alexandra Ganseman
Nathan Salabiaku
Thomas Lonneville
Wiebe Vandendriessche
Xander Vanparys



Inhoud

Lijst met figuren	3
Verklarende woordenlijst.....	4
Inleiding.....	5
Context (Wiebe Vandendriessche)	5
Probleemstelling (Wiebe Vandendriessche).....	5
Verbindende alinea (Thomas Lonneville)	5
Doelstelling (Thomas Lonneville)	6
Hoofdstuk 1: Gebruikersaspecten (Xander Vanparys).....	7
Hoofdstuk 2: Systeemarchitectuur	9
Activiteitendiagram (Nathan Salabiaku)	9
Toestanddiagram (Alexandra Ganseman)	10
Hoofdstuk 3: Testplan	13
Hoofdstuk 5: Installatiehandleiding	15
Referentielijst	17

Lijst met figuren

Figuur 1: Use case-diagram.....	8
Figuur 2: Activiteitendiagram	9
Figuur 3: Toestandsdiagram	10
Figuur 4: Deeltoestanden vragenlijst ruimte	11
Figuur 5: Deeltoestanden vragenlijst functionaliteit	12

Verklarende woordenlijst

Expedition DO!

Initiatief van de UGent om studenten en onderzoekers te ondersteunen en inspireren zodat ze cruciale ondernemende vaardigheden ontwikkelen met als doel een businessidee vorm te geven.

start-up

Een start-up is een doorgaans snelgroeiend bedrijf dat aan een marktbehoefte wil tegemoetkomen door het ontwikkelen van een uitvoerbaar bedrijfsmodel rond een innovatief product of dienst, bedrijfsproces of een platform.

configuratietool

Een tool voor het samenstellen van een product uit verschillende basisbouwstenen, de keuze en schikking van componenten die samen een systeem met de gewenste functionaliteit moeten vormen.

Gebruikersinterface

Uiterlijk van een app dat door de gebruiker wordt gezien.

JavaScript

Een scripttaal waarmee webpagina's interactief en dynamisch kunnen worden. Een deel van een script bestaat uit een aantal opdrachten die de browser uitvoert.

Responsieve lay-outs

Responsieve lay-outs passen zich dynamisch aan verschillende schermformaten aan om een optimale gebruikerservaring te bieden op verschillende apparaten, zoals desktops, tablets en smartphones.

React

React is een JavaScript-bibliotheek voor het bouwen van gebruikersinterfaces.

Three.js

Three.js is een populaire JavaScript-bibliotheek die wordt gebruikt voor het maken en weergeven van 3D-graphics in webapplicaties.

Inleiding

Context (Wiebe Vandendriessche)

Furnify Home werd opgericht door Joachim Schouten en Isaï Cornelis. Volgens de website is het een Gentse start-up die ontstond dankzij het 'Expedition DO!' programma van de Universiteit Gent. In 2022 overtuigde hun concept de jury en begonnen de ondernemers met hun start-up Furnify Home. Het jaar 2023 bracht bekroningen als een van de snelst groeiende start-ups binnen 'Start it @KBC', en een derde plaats als beste studentenonderneming van België (Furnify, z.j.).

Furnify richt zich op de innovatieve sector van modulaire, plaatsbesparende meubels. Door de voortdurende evolutie van woningontwerpen, waar ruimtes steeds compacter worden, komt de noodzaak voor efficiënte inrichtingsoplossingen. Furnify Home legt zich toe op het optimaliseren van leefruimtes door multifunctionele modules te ontwerpen, waaronder slaapkamers, bureauruimtes en inloopkasten. Zo wordt elke vierkante meter optimaal benut.

Probleemstelling (Wiebe Vandendriessche)

Het opmeten van de ruimtes op locatie is een niet zo duurzame oplossing. Het vraagt veel tijd en heeft een ongewenste CO2-uitstoot als gevolg door het telkens heen en weer rijden. Gezien de groeiende aantal klanten, ook in Nederland, is het van essentieel belang om dit te vereenvoudigen. Deze uitdaging staat centraal in de vraagstelling. Het persoonlijk bezoeken van klanten blijkt niet langer de meest praktische en efficiënte aanpak, waardoor een dringende behoefte ontstaat aan een slimme digitale oplossing. De focus ligt op het ontwikkelen van een geavanceerd instrument om dit proces te optimaliseren. Furnify wil een 3D-kamerervaring voor de gebruiker. Zo kan deze zelf zijn afmetingen en wensen online doorsturen. Furnify hoeft dan zelf niet meer ter plaatse te komen voor opmetingen.

Verbindende alinea (Thomas Lonneville)

In het eerste hoofdstuk van dit verslag wordt er toegelicht hoe de vragenlijst eruitziet en hoe deze vragen het 3D-model beïnvloeden vanuit het standpunt van de gebruiker. In het daaropvolgende hoofdstuk wordt er aan de hand van een activiteitendiagram besproken welke processen er door het systeem zullen worden uitgevoerd. Tot slot komen de verschillende toestanden waarin de webapplicatie zich kan bevinden aan bod. Dit zal worden geïllustreerd met toestandsdiagrammen.

Doelstelling (Thomas Lonneville)

Zoals eerder vermeld, ligt de focus op het ontwikkelen van een geavanceerd instrument. Het doel is een gebruiksvriendelijke webapplicatie te creëren die zowel op gsm als op laptop functioneert. De applicatie dient uit twee grote componenten te bestaan, namelijk een vragenlijst en een 3D-visualisatie van de kamer.

De webapplicatie zal volledig *front-end* werken. Hiervoor wordt er gebruik gemaakt van React, een JavaScript-bibliotheek die wordt gebruikt om dynamische gebruikersinterfaces te bouwen (Describing UI, z.j.). Het heeft als voordeel dat codecomponenten kunnen worden hergebruikt. Dit maakt het mogelijk om de vragenlijst op een eenvoudige manier in verschillende secties te verdelen, waardoor de applicatie overzichtelijk blijft. De opdrachtgever verwacht dat de vragenlijst een vijfde van het scherm inneemt. Om het gebruiksgemak op een gsm te verbeteren zal dit worden opgelost door het gebruik van een uitklapbare balk.

Voor de 3D-visualisatie zal er gebruikgemaakt worden van Three.js. Dit is een JavaScript-bibliotheek die wordt gebruikt om 3D-figuren te creëren en af te beelden op een website (Coleman, 2017). Hiermee kan ook aan de wens van de opdrachtgever worden voldaan om de ruimte van de gebruiker in meerdere perspectieven te tonen.

Hoofdstuk 1: Gebruikersaspecten (Xander Vanparys)

Om het heen en weer rijden van klant tot klant te vermijden, biedt Furnify een digitale configuratietool aan. Mensen die geïnteresseerd zijn in de meubels van Furnify kunnen zelf de afmetingen van hun kamer en mogelijke obstakels invoeren. Op die manier kan de online tool bepalen of de meubels in hun kamer passen en hoeveel ruimte bespaard zal worden.

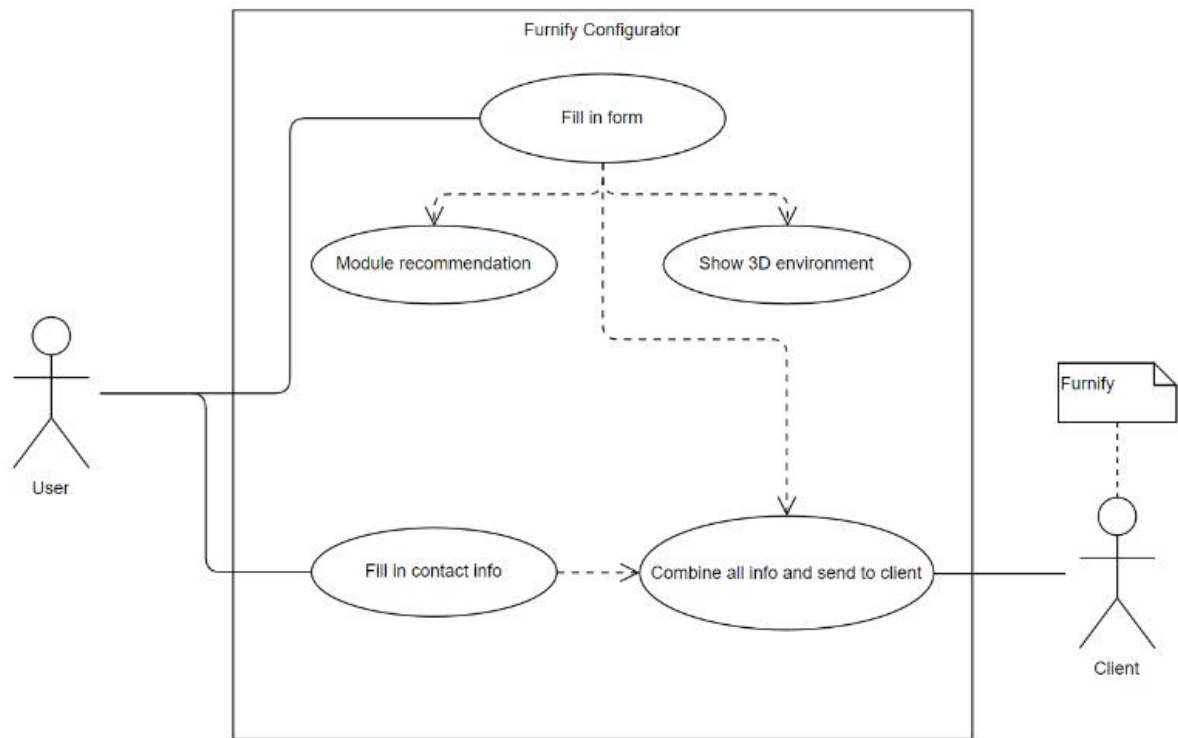
Wanneer een gebruiker de website bezoekt, wordt deze een vragenlijst gepresenteerd. Het doel van deze lijst is om de voorkeuren en eisen van de gebruiker vast te stellen. De verzamelde informatie omvat:

- De belangrijkste functies die de woonruimte moet bieden, zoals een bed, opbergruimte, bureau, enzovoort;
- De gewenste indeling van de ruimte, bijvoorbeeld of modules gebruikt moeten worden als scheidingswand of centraal in de ruimte geplaatst moeten worden;
- De voorkeur voor materiaalkeuze van de modules;
- De afmetingen van de kamer en eventuele obstakels die aanwezig zijn;
- De matrasvoorkeur van de gebruiker, indien een bed wordt geselecteerd;
- Eventuele andere specifieke wensen of voorkeuren die voor Furnify van belang zijn om te weten;

Op basis van deze voorkeuren en rekening houdend met de natuurlijke lichtinval wordt een combinatie van modules voorgesteld.

Zoals afgebeeld in Figuur 1, komt er een 3D-omgeving tevoorschijn terwijl de gebruiker de vragenlijst invult. Met elk gegeven antwoord past de omgeving zich aan om de ingevoerde voorkeuren steeds nauwkeuriger weer te geven. Na het invullen van de afmetingen verschijnt er een lege kamer. Uiteindelijk zal de gebruiker een voorstel krijgen voor de plaatsing van de modules. Dankzij de 3D-tool kan de gebruiker een goed beeld schetsen van hoe zijn kamer eruit zal zien met de modules erin. De gebruiker kan interageren met deze omgeving om de ruimte volledig naar wens in te richten.

Als de gebruiker tevreden is met de configuratie van de modules in de ruimte, kan deze zijn contactgegevens invullen. De gegevens van de vragenlijst gaan naar Furnify voor het uitwerken van een passende oplossing. Op deze manier is er een efficiëntere samenwerking en krijgt de klant een versneld inzicht in het potentiële eindresultaat.

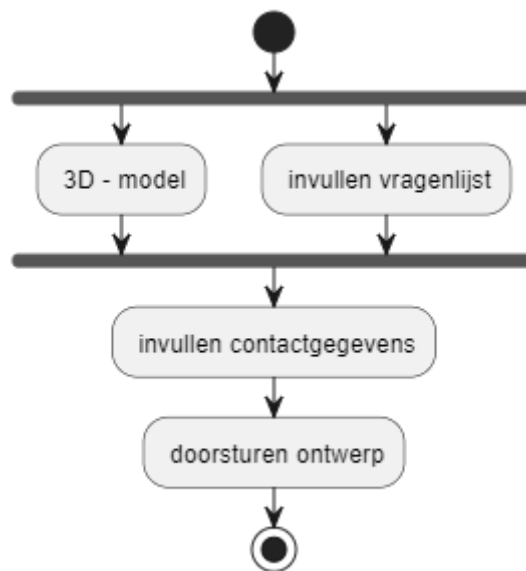


Figuur 1: Use case-diagram

Hoofdstuk 2: Systeemarchitectuur

Activiteitendiagram (Nathan Salabiaku)

Figuur 2 toont het activiteitendiagram van de webapplicatie. Het diagram begint met het invullen van de vragenlijst, geprogrammeerd in React.js. React.js werd gekozen omdat het gemakkelijk te implementeren is. Bovendien levert React.js ook veel tools om een responsieve webapplicatie te maken. De vragenlijst omvat vijf verschillende statussen: ruimte, functionaliteit, specificatie, contactgegevens en bevestigen. Gebruikers kunnen hiermee de benodigde informatie voor hun woonruimte invoeren. De vragenlijst is ontworpen met responsieve lay-outs om een optimale gebruikerservaring te bieden op alle apparaten. Tijdens het invullen van de vragenlijst worden de gegevens bijgehouden in contextklassen.



Figuur 2: Activiteitendiagram

Parallel aan het invullen van de vragenlijst wordt ook een 3D-model van de woonruimte gegenereerd met behulp van Three.js. Een opensource bibliotheek voor het maken van 3D-modellen in Javascript. Het model wordt gemaakt op basis van de ingevulde gegevens in de vragenlijst. Hierbij is het mogelijk om verschillende modules in de woonruimte te plaatsen. De aangeboden modules variëren afhankelijk van de informatie die de gebruiker heeft ingevoerd in de vragenlijst. Dit geeft de gebruiker een volledig beeld van hoe de meubels eruit zullen zien in zijn ruimte. Het 3D-model wordt ook ontworpen met een responsieve lay-out.

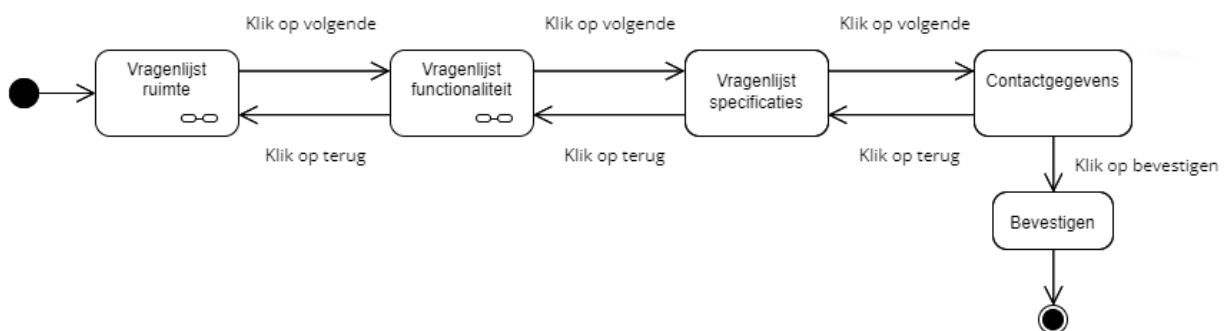
Wanneer de gebruiker tevreden is met het resultaat, kan deze zijn contactgegevens invullen. Dit stelt Furnify in staat om contact op te nemen met de klant. De ingevoerde gegevens worden samen met de rest van de vragenlijst bijgehouden in de contextklassen.

Nadat de vragenlijst is ingevuld en het 3D-model is voltooid, worden de ingevoerde gegevens vervolgens naar Furnify verzonden via e-mail. Op deze manier kan het bedrijf de gegevens verwerken volgens de behoeften van de gebruiker.

Toestandsdiagram (Alexandra Ganseman)

Op Figuur 3 zijn de vijf verschillende toestanden, die de vragenlijst kan aannemen, zichtbaar. Wanneer de gebruiker de website opent, zal deze als eerste de vragenlijst over de ruimte te zien krijgen. Op het moment dat de gebruiker op volgende drukt zal de toestand veranderen naar de vragenlijst over de functionaliteit. Dit kan opnieuw veranderd worden door op de knop “volgende” te klikken, maar de gebruiker kan er ook voor kiezen om terug te keren naar de vorige toestand door op de knop “terug” te klikken. De volgende toestand in dit geval is de vragenlijst over de specificaties, waarnaar de gebruiker ook kan kiezen om door te gaan naar de contactgegevens. Als alle gegevens ingegeven zijn, kan de gebruiker klikken op bevestigen om het proces van de vragenlijst af te ronden. Op dat moment zal de vragenlijst een bevestigingsscherm worden.

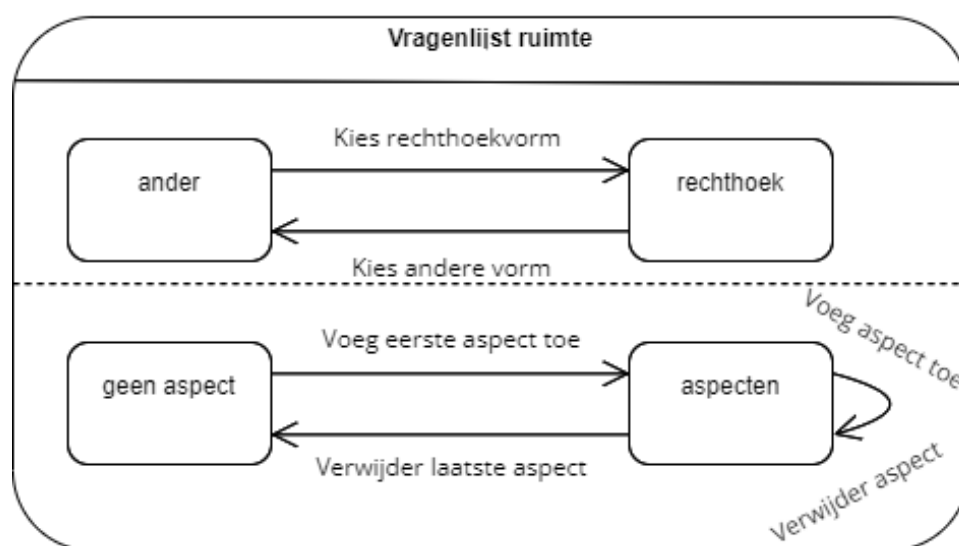
De overgangen tussen de verschillende toestanden gebeuren enkel wanneer de gebruiker de knoppen “volgende” of “terug” aanklikt. Wanneer deze de vragen invult, over de specificaties of de contactgegevens, zal de toestand hetzelfde blijven. De vragen over de ruimte en functionaliteit bevatten echter deeltoestanden, die veranderen afhankelijk van het antwoord op de vragen.



Figuur 3: Toestandsdiagram

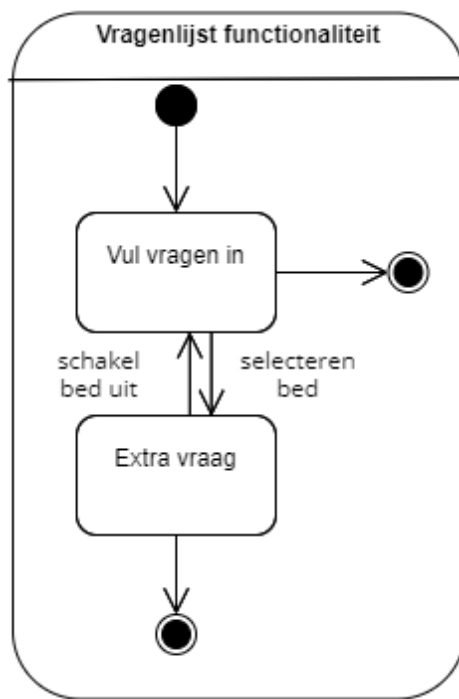
De deeltoestanden van de vragenlijst over de ruimte zijn zichtbaar op Figuur 4.

Afhankelijk van de vorm van de ruimte, is er een ander formaat van gegevens nodig. Voor beide van deze formateringen is er een andere deeltoestand voorzien. Daarnaast krijgt de gebruiker ook de optie om door te geven welke aspecten er zich in de ruimte bevinden. Deze aspecten kunnen onder andere een stopcontact of een verwarming zijn. Hiervoor worden ook twee toestanden voorzien. Een eerste waarbij er geen aspecten aanwezig zijn en een tweede waarbij er één of meerdere aspecten toegevoegd zijn.



Figuur 4: Deeltoestanden vragenlijst ruimte

Ook de vragenlijst over de functionaliteit heeft deeltoestanden, zie Figuur 5. De gebruiker kan doorgeven wat noodzakelijk is in de ruimte. Enkel wanneer deze opgeeft dat deze nood heeft aan een bed zal er een extra vraag verschijnen. Deze vraag zal gaan over de keuze van matras die de gebruiker wenst.



Figuur 5: Deeltoestanden vragenlijst functionaliteit

Hoofdstuk 3: Testplan

De front-end applicatie wordt getest aan de hand van *end-to-end-testing*. Hiermee is het mogelijk het gedrag van een eindgebruiker na te bootsen en te bekijken of de applicatie op de interactie reageert zoals ervan wordt verwacht. Deze keuze is gemaakt omdat *end-to-end testing* de volledige samenwerking tussen alle componenten test, waardoor ook afzonderlijke componenten indirect worden getest door de resultaten van de samenwerking.

De tests zijn geautomatiseerd wat het voordeel biedt dat bij toevoegingen van nieuwe features de tests steeds opnieuw kunnen worden uitgevoerd. Volgens Test Guild (2023) zijn de top 3 automatisatie testtools Selenium, Cypress en PlayWright. Voor dit project is Cypress gebruikt.

Cypress is een opensource *end-to-end testing framework* dat ook gebruikt kan worden voor *component*-, *integration*- en *unit testing*, hoewel dat laatste minder eenvoudig is. Dit *framework* ondersteunt het testen van JavaScript en TypeScript toestanden, wat het ideaal maakt voor dit project.

In tegenstelling tot Selenium is het namelijk eenvoudiger en sneller in gebruik, doordat het directe toegang heeft tot de DOM van de browser zonder een externe *web driver* nodig te hebben. Dit zorgt voor snellere en efficiënter communicatie met de browser tijdens het uitvoeren van tests. Bovendien biedt Cypress de mogelijkheid om *realtime* mee te kijken met de testuitvoering terwijl deze plaatsvindt, waardoor ontwikkelaars eenvoudiger fouten kunnen opsporen dan bij PlayWright.

Een nadeel van Cypress is dat het enkel kan testen op Chromium-gebaseerde browsers. Het is dus niet mogelijk om de tests te laten uitvoeren via Safari. Dit wordt opgelost door Apple-gebruikers de webapplicatie te laten uitproberen.

Met de geschreven tests wordt getracht de handelingen van de eindgebruiker zo nauwkeurig mogelijk na te bootsen. Dit omvat onder andere het toevoegen van obstakels en het vervolgens verslepen ervan in de 3D-ruimte. Ook wordt nagegaan of, afhankelijk van de browsertaal, de correcte tekst steeds wordt weergegeven. De testsuite bevat ook controles voor foutieve invoer zoals een telefoonnummer met te veel cijfers, letters of negatieve waarden bij een afmeting. Belangrijk is ook het behoud van data. Zo mogen aangebrachte wijzigingen niet verdwijnen bij het navigeren naar een volgende sectie van de vragenlijst.

Hoewel Cypress zich focust op DOM-elementen wordt ook de 3D-visualisatie getest. Hiervoor is er nood aan een extra Cypress-plugin. De controle gebeurt dan aan de hand van regressie waarbij het verkregen beeld wordt vergeleken met een snapshot.

De tests worden tijdens het ontwikkelproces voornamelijk gebruikt om na te gaan of nieuwe toevoegingen andere delen niet beschadigen. Deze worden uitgevoerd na iedere commit, maar kunnen ook lokaal worden uitgevoerd. Nieuwe tests worden uitgeschreven na het toevoegen van een nieuwe feature. Tijdens het uitwerken van een feature worden eventuele fouten vaak met behulp van logging opgespoord.

Het nabootsen van de handelingen van de eindgebruiker is uiteraard niet voldoende. Daarom worden personen, die niets afweten van het ontwikkelproces, gevraagd om de applicatie uit te testen en het van feedback te voorzien. Dit gaat niet enkel over mogelijke fouten, maar ook over de gebruiksvriendelijkheid van de applicatie.

Hoofdstuk 5: Installatiehandleiding

Applicatie bereiken

De applicatie is een pure front-end webapplicatie en vereist dus geen externe software of scripts. Een standaard webbrowser, zoals Chrome, Firefox of Safari volstaat om de applicatie te bereiken. Deze is momenteel ontplooid op een Linux-server van de UGent met IP-adres: 157.193.171.41. Surfen naar de website kan dus via gewoonweg <http://157.193.171.41>.

Applicatie uittesten in ontwikkelingsomgeving.

Tijdens de ontwikkeling van de applicatie werd gebruik gemaakt van Vite. Vite is een lokale ontwikkelingsserver en wordt vaak gebruikt bij React-projecten. Volg de volgende stappen om de applicatie lokaal uit te testen:

1. Zorg dat de *gitlab-repository* lokaal staat.
 - a. Open een terminal venster.
 - b. Voer het volgende commando uit: *git clone [repository-name]*
2. Ga in een terminal venster naar de map van de *repository*.
3. Voor de meest recente versie van de code te verkrijgen: *git pull*
4. Voer het volgende commando uit om in de juist map te staan: *cd ./project*
5. Installeer de juiste *packages*: *npm install*
6. Run de applicatie lokaal: *npm run dev*

Applicatie zelf ontplooien op de server

Zelf de site *builden* en ontplooien op bijvoorbeeld een *apache2-webserver* is ook iets wat mogelijk wordt gemaakt door Vite. Dit kan via volgende commando's:

1. Clone het project op de server.
2. Ga naar de map: */project*.
3. Installeer de juiste *packages*: *npm install*
4. Voer *npm run build* uit.
5. In de root van de server voer deze commando's uit om de site op de apache2 server te zetten:
 - a. *sudo rm -rf /var/www/html/**
 - b. *sudo cp -r furnishy2/project/dist/* /var/www/html*
6. Om de *productionbuild* lokaal te previewen, gebruik: *npm run preview*

Referentielijst

Coleman, B. (2017). The Beginner's Guide to Beginning Three.js. Geraadpleegd op 19 februari 2024 via <https://medium.com/@benjamin.c.coleman/the-beginners-guide-to-beginning-three-js-c36b8947c2aa>

Describing the UI. (z.j.). Geraadpleegd op 21 februari 2024 via <https://react.dev/learn/describing-the-ui>

Furnify. (z.j.). Geraadpleegd op 20 februari 2024 via <https://www.furnifyhome.eu>

The Top 3 Automation Testing Tools (2023 Edition). (2023). Geraadpleegd op 17 maart 2024 via <https://testguild.com/top-3-automation-testing-tools/>