

Quintor Worqplace overdrachtsdocument

**Studenten:**

Milan Dol, 1756493

Saïd Shirre, 1726454

Gerson Mak, 1753045

Jan-paul van der Moolen, 1738045

Arutun Avedisyan, 1729476

Daan Docters van Leeuwen, 1760670

Mees Wieman, 1757924

Opdrachtgever: Pim Otte

Agile scrum coach: Ana Goessens

Versie: 2.0

Datum: 4-2-2022

Versiebeheer

Versie	Datum	Auteur(s)	Omschrijving
1.0	12-11-2021	Milan, Saïd, Gerson, Jan-paul, Arutun, Daan en Mees	<ul style="list-style-type: none">- Aanmaken van document- Het invullen van de volgende hoofdstukken: Inleiding, beschrijving korte artefacten, inloggegevens, deployment, product backlog, systeemdokumentatie, known bugs & beperkingen en bronnenlijst
2.0	04-02-2022	Milan, Saïd, Gerson, Jan-paul, Arutun, Daan en Mees	<ul style="list-style-type: none">- Het aanvullen en updaten van de inloggegevens- Het aanvullen van de known-bugs en beperkingen

Inhoudsopgave

Inhoudsopgave	2
Inleiding	3
Korte beschrijving artefacten	4
Angular front-end	4
Java back-end	4
MySQL database	4
Inloggegevens	5
Database	5
MySQL inloggegevens:	5
Webapplicatie	5
Webapplicatie inloggegevens:	5
Quintor tools	5
Quintor inloggegevens:	5
Deployment	6
Product backlog	9
Systeemdocumentatie	10
Angular front-end	10
Projectstructuur	10
Java back-end	11
Known Bugs en beperkingen	12
Bronnenlijst	13

Inleiding

Het afgelopen jaar zijn miljoenen mensen thuis komen te zitten door een globale pandemie. Dit heeft geleid tot een thuiswerk samenleving. Nu, eind 2021, zijn de kansen om naar werk te komen weer toe aan het nemen, maar dit neemt ook problemen met zich mee; Door alle regels rondom de pandemie zijn bedrijven geforceerd beperkte werkplekken vrij te geven op de werkvloer. Uit deze kwestie is de vraag ontstaan hoe je ervoor kan zorgen dat de open werkplek bij de willende medewerker terecht komt? Op dit gebied wilt Quintor gaan innoveren. Om dit te bereiken hebben ze een groep studenten van de Hogeschool Utrecht gevraagd, een systeem te ontwikkelen die de werkplek helpt moderniseren.

In dit overdrachtsdocument gaan wij, de groep studenten, u, de lezer, informeren over de oplossing en hoe het systeem in elkaar steekt. Zo kan er indien nodig, verder gewerkt worden met de oplossing die er dit blok is neergezet. Het document begint met een korte beschrijving over de artefacten (de zaken die deployable zijn). Vervolgens zal de benodigde informatie gegeven worden die vereist zijn om je te authenticeren in bepaalde systemen. Daarna wordt er een beschrijving gegeven over de deployment van de oplossing. Hierna volgt de product backlog en hierop volgend de overzicht van de (systeem)documentatie. Als laatst zullen de known bugs en beperkingen van het systeem besproken worden met daarna de decharge voor de zaken die het projectteam kwijt wil.

Door dit alles hopen wij een duidelijk beeld te schetsen over de oplossing, zodat de overdracht van de oplossing zo soepel mogelijk verloopt.

Korte beschrijving artefacten

Bij dit project komen een aantal onderdelen kijken die ervoor zorgen dat het alle artefacten als geheel functioneert. Hieronder volgt een korte beschrijving van wat deze onderdelen zijn en hoe je ze draait.

Angular front-end

De front-end is de worqplace website. Deze front-end maakt het makkelijk om reserveringen te maken en beheren. Het is in de Angular framework gemaakt en is deployable. Hij is te runnen door in de terminal van je IDE (Integrated Development Environment) de volgende commando's uit te voeren:

- npm install (dit installeert alle dependencies die benodigd zijn voor het project)
- ng serve (dit start de lokaal de applicatie op in een server)

De angular server draait normaal gesproken op port 4200.

Java back-end

De back-end is een API die het mogelijk maakt reserveringen in de database toe te voegen, uit te lezen, aan te passen en verwijderen. Hij is beschermd onder een login systeem met JWT tokens. Hij is gemaakt in Java en maakt gebruik van het Spring Boot framework. Hij is te runnen door de main applicatie file te executen in je IDE. Ook kun je de applicatie opstarten door in de hoofdmap van het project het volgende commando uit te voeren:

- mvnw spring-boot:run

De java back-end server draait normaal gesproken op poort 8080.

MySQL database

Voor de database is gebruikt gemaakt van MySQL. Hiervoor moet je online een MySQL server downloaden en configureren. Daarna kun je connecten met de server via MySQL client (e.g. MySQL Workbench, Navicat, Datagrip).

De MySQL database draait normaal gesproken op port 3306.

Inloggegevens

Binnen deze paragraaf staan een aantal inloggegevens. Deze gegevens zijn bedoeld voor development en zijn dus niet vertrouwelijk. Al deze gegevens kunnen worden aangepast. Met uitzondering op de inloggegevens van de server. Deze gegevens zijn van Quintor zelf en zullen per persoon verschillen.

Database

De applicatie verwacht een MySQL database waarmee verbonden kan worden. Er wordt geprobeerd te verbinden met deze database met de volgende inloggegevens.

MySQL inloggegevens:

Gebruiker: root

Wachtwoord: adminadmin

Deze gegevens kunnen worden aangepast in de backend ***application.properties*** file en in de ***docker-compose.yml*** file.

Webapplicatie

Wanneer de front-end, back-end en database alle drie draaiende zijn. Dan kun je naar de webpagina gaan op localhost:4200 of de ip-adress van de server waarop de webapplicatie gedeployed is. **Zorg er wel voor dat de data van het init.sql bestand in geladen is in de database** (te vinden in de directory: worqplace-backend/src/main/resources/data.sql). Alleen dan gelden de inloggegevens.

Webapplicatie inloggegevens:

Gebruikersnaam: admin@quintor.nl

Wachtwoord: Kaasje

Quintor tools

Er is gebruikt gemaakt van Jira (scrumboard), Bamboo (CI/CD), Sonar (Code quality) en Gitlab (Versiebeheer). Om toegang tot deze tools te krijgen, moet je contact opnemen met de beheerder.

Quintor inloggegevens:

Gebruikersnaam: "Je Quintor gebruikersnaam"

Wachtwoord: "Je Quintor wachtwoord"

Deployment

Vanuit Quintor hebben we geen server meegeleverd gekregen en de deployment is dus puur vanuit Docker gemaakt om de compatibility zo hoog mogelijk te houden. Hieronder staan gegevens over het Docker bridge network.

Docker bridge network

CIDR: 172.1.0.0/24

Gateway: 172.1.0.1

Containers app_network

- MySQL
 - Ports: 3306
 - IPv4: 172.1.0.2
- Openjdk 17 (Backend API)
 - Ports: 8080
 - IPv4: 172.1.0.3
- NGINX (Frontend)
 - Ports: 80
 - IPv4: 172.1.0.4

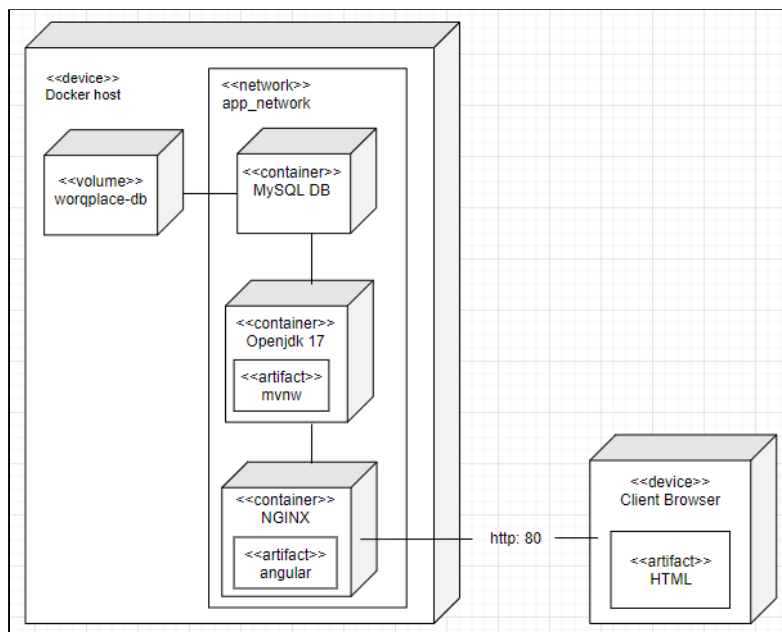
Database names

- worqplace

Protocollen

- HTTP
- TCP

In de onderste afbeelding staat de omgeving die na het stappenplan wordt gegenereerd.



Afbeelding 1: Deployment diagram

Omgeving uitrollen

Dependencies

- Docker engine¹
- Docker-compose²
- worqplace-frontend-development files³
- worqplace-backend-development files⁴

1. Maak eerst een folder genaamd app op de server of machine.
2. Zorg ervoor dat de worqplace-frontend-development en de worqplace-backend-development files in de folder zijn gedownload.
3. Zorg ervoor dat docker engine en docker-compose zijn geïnstalleerd. Daar kunnen deze websites bij helpen: docker engine: <https://docs.docker.com/engine/install/>. Docker-compose: <https://docs.docker.com/compose/install/>
4. Navigeer naar de backend folder. Hierin staat de docker-compose file en gebruik het volgende command: `docker-compose up`.

Als de stappen goed zijn uitgevoerd zal de omgeving automatisch worden opgezet. De database maakt een volume, zodat de data niet verloren gaat als de database container stop wordt gezet. Verder communiceert de backend met de database container via het Docker bridge netwerk. De frontend communiceert met de backend via API.

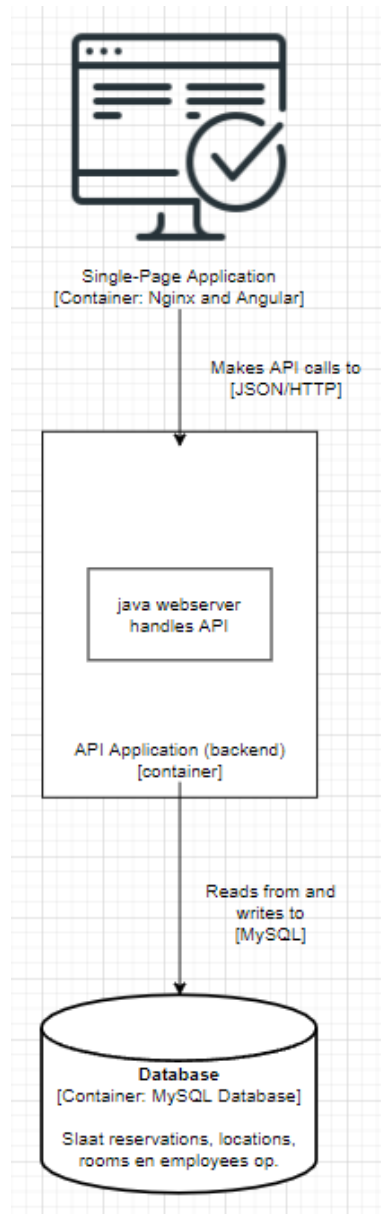
¹ *Install Docker Engine*. (2021, 11 november). Docker Documentation. Geraadpleegd op 12 november 2021, van <https://docs.docker.com/compose/install/>

² *Install Docker Compose*. (2021, 11 november). Docker Documentation. Geraadpleegd op 12 november 2021, van <https://docs.docker.com/compose/install/>

³ *Worqplace-backend*. (z.d.). Gitlab. Geraadpleegd op 12 november 2021, van <https://git.quintor.nl/SG-HogeschoolUtrecht/worqplace-backend>

⁴ *Worqplace-frontend*. (z.d.). GitLab. Geraadpleegd op 12 november 2021, van <https://git.quintor.nl/SG-HogeschoolUtrecht/worqplace-frontend>

In de onderstaande afbeelding staat de docker omgeving na het uitrollen van docker:



Product backlog

Voor de backlog is gebruik gemaakt van het projectmanagement tool Jira van Atlassian. Deze is terug te vinden via de volgende link:

<https://jira.quintor.nl/secure/RapidBoard.jspa?rapidView=609&projectKey=HI21&view=planning.nodetail&issueLimit=100>

Hieronder staat een foto van een moment eerder in het project aangezien het project nu afgelopen is en de backlog leeg is.

The screenshot displays a Jira Product Backlog interface. At the top, there is a list of 18 issues, each with a status icon (red square, green plus, or green up arrow), a priority icon (orange double chevron), an issue key, and a description. The issues are sorted by priority. Below this list, there is a section titled 'Backlog' with a sub-header '2 issues'. This section contains two issues, each with a green plus icon, a priority icon, an issue key, and a description.

Status Icon	Priority Icon	Issue Key	Description
Red square	Orange double chevron	HI21-87	Recurrence check for future reservations
Green plus	Orange double chevron	HI21-95	As a user I want feedback why my reservation is not possible
Green plus	Orange double chevron	HI21-65	As a developer I want Front end tests
Green plus	Orange double chevron	HI21-86	As a developer I want the frontend to be clearly documented
Green plus	Orange double chevron	HI21-82	As a developer I want my code to be sonar compliant
Green plus	Orange double chevron	HI21-54	As a developer I want continuous integration
Green plus	Orange double chevron	HI21-55	As a developer I want a sonarQube integration
Green up arrow	Orange double chevron	HI21-88	As a user I want an intuitive timepicker for all browsers
Green plus	Orange double chevron	HI21-91	As a user I want to be able to log in
Green plus	Orange double chevron	HI21-94	As a developer I want to be able to deploy
Green plus	Orange double chevron	HI21-67	As a developer I want a container diagram
Green up arrow	Orange double chevron	HI21-89	As a developer I want a deployment model with artifacts
Green plus	Orange double chevron	HI21-96	As a user I want the website to be user friendly
Green plus	Orange double chevron	HI21-97	As a user I want to have a mobile-friendly version of the website
Green plus	Orange double chevron	HI21-20	As a user I want to delete a reservation of mine
Green plus	Orange double chevron	HI21-39	As an admin I want to be able to see all reservations per location
Green circle	Orange double chevron	HI21-90	As a developer I want one time test setups in my method instead of a huge init method

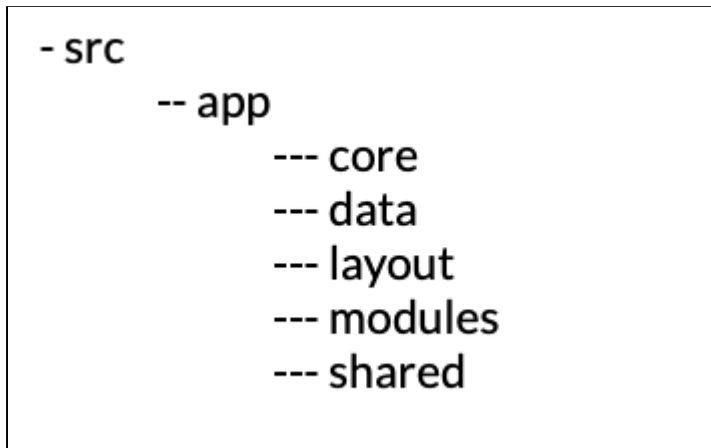
Backlog 2 issues			
Green plus	Orange double chevron	HI21-18	As a user I want to add my confirmed reservation to my calendar
Green plus	Orange double chevron	HI21-21	As a user I want to alter my reservation

Systeemdocumentatie

Angular front-end

Projectstructuur

De projectstructuur heeft onderhoudbare en flexibele structuur die te gebruiken is voor zowel kleine als grote projecten. Het is ontworpen door de angular community met de best-practises in gedachte.⁵



Afbeelding 3: De projectstructuur van de front-end in grote lijnen

In afbeelding 3 is de projectstructuur in grote lijnen te zien. De applicatie is ingedeeld in de volgende 5 directories:

- De **core directory** is verantwoordelijk voor resources die altijd aanwezig moeten zijn bij de levensloop van de applicatie (e.g authenticatie, http interceptors).
- De **data directory** is verantwoordelijk voor de communicatie met de backend en het ophalen van de data.
- De **layout directory** bevat html templates die gebruikt kunnen worden door andere componenten.
- De **modules directory** bevat applicatie features die opgehaald kunnen worden in de applicatie wanneer ze daadwerkelijk nodig zijn door de eindgebruiker. Dit zorgt voor lagere laadtijden (lazy loading design pattern).
- De **shared directory** bevat resources (componenten en services) die gedeeld kunnen worden door andere modules.

⁵ Structure Overview — Angular Folder Structure documentation. (z.d.). Angular-Folder-Structure. Geraadpleegd op 7 november 2021, van <https://angular-folder-structure.readthedocs.io/en/latest/>

Java back-end

Spring boot back-end

De back-end is gebouwd in combinatie met hibernate en spring boot om een rest applicatie neer te zetten. Het systeem is opgebouwd in verschillende lagen op basis van de functionaliteit die de laag heeft.

- **Presentation**
Deze laag handelt communicatie af tussen de gebruiker en het systeem. Hier leven de rest controllers.
- **Application**
Verzoeken uit de presentation laag worden naar deze laag gestuurd om afgehandeld te worden. Hier leven de service klassen, die logica uitvoeren op het domain en die opslaan via de data laag.
- **Domain**
Deze laag bevat de object georiënteerde representatie van de database. Deze klassen worden gebruikt om gegevens in op te slaan.
- **Data**
Door deze laag worden de domain klassen opgeslagen in de database. Hier leven de repositories die via Hibernate gegevens opslaan in de MySQL database.

Known Bugs en beperkingen

Known Bugs

- Bamboo faalt nog op sonar, omdat het niet de generated sources pakt.
- PITesting in de Backend Werkt niet op reservation integration test classes.
- Bij het op datum sorteren haalt eerst nog alle reserveringen op om te sorteren ipv de huidige lijst gebruiken.
- Er moet eerst een locatie geselecteerd zijn voordat er op datum gesorteerd kan worden.
- Het deleten van een reservatie doet een POST request i.p.v DELETE request en dat kwam door de CORS filter.

Beperkingen

- Bij het toevoegen, verwijderen of aanpassen van een reserveringen moet er eerst gerefreshed worden voordat de wijziging zichtbaar is.
- Er worden nog geen notificaties weergegeven bij het wijzigen/verwijderen van een reservering.
- Nginx container redirect niet naar de juiste locatie als je de url aanpast met een / locatie. Voorbeeld: worqplace.nl/admin. Dit geeft een Nginx foutmelding dat hij de pagina niet herkent.

Bronnenlijst

1. Install Docker Compose. (2021, 11 november). Docker Documentation. Geraadpleegd op 12 november 2021, van <https://docs.docker.com/compose/install/>
2. Install Docker Engine. (2021, 11 november). Docker Documentation. Geraadpleegd op 12 november 2021, van <https://docs.docker.com/engine/install/>
3. Structure Overview — Angular Folder Structure documentation. (z.d.). Angular-Folder-Structure. Geraadpleegd op 7 november 2021, van <https://angular-folder-structure.readthedocs.io/en/latest/>
4. Worqplace-backend. (z.d.). Gitlab. Geraadpleegd op 12 november 2021, van <https://git.quintor.nl/SG-HogeschoolUtrecht/worqplace-backend>
5. Worqplace-frontend. (z.d.). GitLab. Geraadpleegd op 12 november 2021, van <https://git.quintor.nl/SG-HogeschoolUtrecht/worqplace-frontend>