Handleiding voor developers

Mijn naam is Robin Vervoorn en ik was scrummaster/notulist van dit project.  
Ik regelde merendeels de documentatie van het project, dus ik kan jullie het best begeleiden over hoe jullie als het nieuwe development team verder kunnen met het ontwikkelen van dit project.

# Wat houdt het project in?

Ik denk dat jullie al waarschijnlijk uitleg hebben gekregen over wat dit project is, maar eigenlijk nog een vaag idee van wat jullie moeten doen.

Als het nog goed klopt, dan is Martin Borsje jullie Product Owner (PO). Hij heeft zelf een generiek idee van wat hij wilt, maar hij weet eigenlijk nooit precies wat hij echt wil hebben. Jullie taak wordt hem uiteindelijk goed laten realiseren wat hij nou eigenlijk wil. Hiervoor zullen jullie later sprintwissels houden.

In dezelfde folder waar jullie deze handleiding hebben gevonden zitten 2 bestanden:

* Handleiding-devs (deze handleiding)
* Kwaliteiteisen-document-CDTC.pdf

Hierin staat alles documentatie die je nodig hebt om verder te kunnen gaan met dit project.

Ik raad eerst aan om "kwaliteiteisen-document-CDTC.pdf" effe goed door te lezen. Dit is het kwalificatie dossier van dit project. Hierin heb ik alles gedocumenteerd wat relevant is voor het project.   
Meest belangrijke onderdelen in het kwalificatie dossier zijn:

* 2.1 Doel / 2.2 Aard / 2.3 Doelgroepen
* 3.1 Verantwoordelijkheden / 3.2 Definition of Done (DoD)
* 4.1 MVC / 4.3 MPA / 4.4 Keuze Architectuur en Softwarepatronen
* 5.1 Django / 5.4 Frameworks en Libraries / 5.5 Hoofdcomponenten / 5.6 Systeemomgeving
* 6.1 Logische Databasestructuur (Dit is een concept)
* 7.0 Coding Standards / 7.1 Branching & Commits / 7.2 Overige Standards
* 8.1 Look & Feel

In deze documentatie zal je ook nog wat terugvinden over een Microsoft Azure server/database.   
Hier zullen jullie niet veel aan hebben aangezien dit op naam staat van Versiebeheer (Azure).   
Wij probeerde dit te gebruiken als live server, maar de klant wil geen subscription service gebruiken.  
Als oplossing hebben we van Martin Borsje een Raspberry PI geleend.   
Dit is in het bezit van Stefano Verhoeve of Martin Borsje.   
Deze Raspberry PI heeft alle benodigde onderdelen al geïnstalleerd.

# Opzetten van project

Het eerste wat jullie moeten doen is het overnemen van de huidige repository. Dit kan op 2 manieren gedaan worden:  
- Het forken van de originele repository.  
- Een .zip downloaden van de originele repository en een nieuwe maken.  
(Vraag effe na aan jullie SLB'er welke van de twee het beste is)

Zodra je een eigen repository heb, moet je de repository clonen in je eigen directory/folder.

Eerst moet je Python met PIP gaan installeren.  
Ga naar: <https://www.python.org/downloads/> en download de meeste recente versie van Python 3.11 (Verwijder eventueel oudere/nieuwere versies van Python)

Zorg ervoor dat PIP ook geïnstalleerd wordt. PIP is Python's Package manager.   
Als je niet zeker weet of PIP geïnstalleerd is, dan voer "pip --version" uit in CMD of Windows Powershell. Je kan in dezelfde command "pip" vervangen met "py" of "python" om te kijken welke versie van Python jouw systeem gebruikt.

Zoiezo moet je ook effe Django zelf installeren.

Als zowel Python/PIP als Django geïnstalleerd is, dan wordt het tijd om de packages/libraries te downloaden.  
Open CMD in de project folder en voer deze command uit: "pip install -r requirements.txt".  
Dit gaat alle packages downloaden die in de requirements.txt gedefinieerd zijn.  
(Let op!!! Sommige packages kunnen mislukken met downloaden. Probeer de command opnieuw)

Zorg ervoor dat je een eigen .env bestand maakt en hier in de directory plaats:  
A screenshot of a computer

Description automatically generated  
(Bekijk afbeelding hierboven)

Dit is wat er in de .env moet zitten:  
A black screen with white text

Description automatically generated  
(Bekijk afbeelding hierboven)

Nu wordt het tijd om een database op te zetten. De huidige database gebruikt MySQL.  
Jullie zullen een database schema moeten creëren lokaal.   
(Let op dat er een verschil is tussen de database zelf en een database schema!!!)

Zorg ervoor dat je een tool gebruikt om MySQL databases in te kunnen kijken zoals:  
- PHPmyAdmin van Ampps  
- MySQL Workbench

Als het goed is zouden jullie tenminste één van de tools moeten hebben. Toch raad ik aan om MySQL workbench te gebruiken, zodat jullie later de database connectie van de Raspberry PI in kunnen kijken als die live gaat.

Voor jullie lokale database raad ik aan om de standaard database te gebruiken die jullie ook hebben gebruikt voor de rubyquest opdracht  
De standaard inloggevens voor die database zijn:  
Username: "root"  
Password: "mysql" of "password"

Als dit niet werkt, dan heb je waarschijnlijk andere inloggegevens. Vraag dit aan jullie SLB'er.

Zodra je in de database bent, dan moet je een nieuw schema aanmaken genaamd "cdtc\_db" volgens de DB\_NAME in het .env bestand.  
(De naam moet gelijk zijn aan "cdtc\_db" want anders kan je de server later niet lokaal gebruiken. Bekijk hiervoor de code van settings.py)  
Verander DB\_USER en DB\_PASS in de .env bestand naar je eigen database user en password als die nog niet gelijk staan aan elkaar.

Nu dat de database opgezet is, hoef je alleen nog een paar commands te runnen in de project folder.  
Open CMD in de project folder, en run de volgende commands:

* "py manage.py makemigrations". Hiermee maak je de migration files als die nog niet aanwezig zijn. Zolang het project nog niet live is, mag je de migrations verwijderen en opnieuw aanmaken.  
  LET OP!!! Dit is heel belangrijk. Als de applicatie ooit live komt te staan, dan moet je een nieuwe migration file maken voor elke verandering in de models en seeders. Django leest migration files maar één keer uit en daarna niet meer.
* "py manage.py migrate". Hiermee migrate je alle nieuwe migration files met changes in de database. Elke migration file wordt maar één keer uitgelezen, dus geen migration files verwijderen tenzij je de database schema gaat droppen.
* "py manage.py seeders" Hiermee worden alle seeders uitgevoerd en nepdata aangemaakt in de database.
* "py manage.py runserver 8000" Hiermee kan je de live server starten en de applicatie bekijken met de URL die in de terminal wordt vermeldt.

Nu heb je alle stappen gevolgd en zou alles moeten werken. Als iets nog niet helemaal werk, dan volg de stappen opnieuw en anders vraag het aan jullie SLB'er.

Jullie gaan zoiezo een backlogsysteem nodig hebben. Ik heb een kopie gemaakt van ons backlogsysteem gemaakt in trello. Kopieër dit Trello-bord voor jullie zelf.

Hier is de link naar het Trello-Bord: <https://trello.com/b/tkCeXHFG/cdtc-copy>

NOTE: Voor het gebruik maken van de MQTT broker script. Vul de juiste database credentials in "mysqlConnection.py" en gebruik de comment variables in "Read\_data\_file.py". Daarna run de "Read\_data\_file.py" script.

Ik wens jullie verder succes met het vervolgen van dit project.