

**Document 3: Sjabloon rapport**

*(OPM: haal deze titel weg bij het definitieve rapport)*

Configuration Management Database

*Het ontwerpen van een Configuration Management Databse met behulp van het Django framework*

Weyen Lars

Gerits Liesanne

De Raeve Hans en Spleesters Guy

Telloport <https://telloport.com>

Woord vooraf

Met trots presenteer ik u mijn graduaatsproef “*Configuration* *Management* *Database* met het *framework* Django”. Dit onderzoek is als onderdeel van mijn opleiding Programmeren aan de hogeschool PXL. Gedurende het tweede semester van het academie jaar 2022-2023 ben ik bezig geweest met het onderzoeken en schrijven van mijn graduaatsproef.

Mijn stage bedrijf had een overzicht nodig van al hun installaties, dit project heb ik op mij genomen. Om dit project te realiseren heb ik mij eerst vertrouwd maken met Python, aangezien Django een *framework* gebaseerd is op Python. Deze programmeer taal was nieuw voor mij maar door wat ik allemaal heb bijgeleerd op deze opleiding was ik er snel mee weg. Verder heb ik ook meer ervaring kunnen opdoen met de codeer talen waarmee ik al vertrouwd was zoals HTML, CSS en Javascript. Waardoor ik veel heb bijgeleerd met het onderzoeken en schrijven van mijn graduaatsproef zowel op persoonlijk als professioneel vlak.

Graag wil ik mijn waardering uitspreken aan mijn werkplekcoaches, Hans De Reave en Guy Spleesters, voor de kans die zij mij hebben gegeven om een waardevolle en leerrijke stage bij hen te volgen. Daarnaast wil ik hen bedanken voor de mogelijkheid om aan dit interessante project te werken en voor hun begeleiding gedurende dit proces. Ook wil ik Liesanne Gerits mijn PXL-coach bedanken voor de uitstekende begeleiding. Alsook voor de goede ideeën en kritische feedback op mijn opdrachten in verband met het graduaatsproef.

Tot slot wil ik nog mijn familie en vrienden bedanken die gedurende mijn graduaatsproef altijd voor mij klaar stonden.

Ik wens u veel leesplezier bij het doornemen van mijn graduaatsproef.

Lars Weyen

Inhoudsopgave

Inhoud

[Woord vooraf 2](#_Toc134692925)

[Inhoudsopgave 3](#_Toc134692926)

[1 Bedrijfsvoorstelling 4](#_Toc134692927)

[2 Projectvraag, onderzoeksacties en resultaten 4](#_Toc134692928)

[2.1 Situering probleemstelling 4](#_Toc134692929)

[2.2 Projectvraag en deelvragen 4](#_Toc134692930)

[2.3 (Onderzoeks-)acties 5](#_Toc134692931)

[2.4 Verzamelde resultaten 5](#_Toc134692932)

[2.5 Stakeholders 7](#_Toc134692933)

[2.6 Doelstellingen 7](#_Toc134692934)

[3 Conclusies en aanbevelingen 7](#_Toc134692935)

[3.1 Conclusies 7](#_Toc134692936)

[3.2 Aanbevelingen 8](#_Toc134692937)

[4 Technische beschrijving 8](#_Toc134692938)

[4.1 Techologieeën 8](#_Toc134692939)

[4.2 Tools 8](#_Toc134692940)

[5 Beschrijving uitwerking softwaretoepassing 8](#_Toc134692941)

[5.1 Aanpak 8](#_Toc134692942)

[5.2 Resultaat 9](#_Toc134692943)

[6 Persoonlijke reflecties en kritische kanttekeningen 9](#_Toc134692944)

[7 Referentielijst 10](#_Toc134692945)

[8 Bijlagen 11](#_Toc134692946)

# Bedrijfsvoorstelling

Telloport biedt een oplossing voor de contactloze uitwisseling van goederen voor B2B-klanten. Het ecosysteem van Telloport verzorgt het onbemand 24/7 leveren en ophalen van goederen via “slimme locaties”. Deze slimme locaties kunnen lockers zijn die uitgerust zijn met een terminal, de zogenaamde distriSPOTs, maar dit kunnen evengoed voertuigen zijn waarin tijdens de nacht goederen geleverd kunnen worden, intelligente rekken, slimme containers, etc.

# Projectvraag, onderzoeksacties en resultaten

## Situering probleemstelling

Elk slimme locatie bevat een aantal hardware en software componenten die via het cloud based platform ( het Telloport platform ) worden aangestuurd. Het platform regelt de business logica van welke goederen bevinden zich waar en wie heeft op welk moment toegang tot deze locaties om goederen te brengen of af te halen

Het beheer van al de hardware en software componenten wordt belangrijker naarmate het aantal slimme locaties toeneemt. Welke hardware componenten bevat iedere slimme locatie, hoe zijn deze geconfigureerd, welke hardwaren en software versies bevatten deze componenten, hoe zijn ze ingericht bij de klant, netwerkconfiguratie, etc. Vandaar de noodzaak om een module te ontwikkelen die geïntegreerd is met het bestaande Telloport platform om al deze componenten in kaart te brengen en te beheren.

## Projectvraag en deelvragen

Hoe kunnen de hardware en software componenten die aanwezig zijn op de slimme locaties van Telloport overzichtelijk in kaart gebracht en behoord worden met behulp van een webapplicatie die gebouwd zal worden met Python en Django waarbij er een integratie is met het bestaande Telloport platform binnen een semester tijd?

* Hoe behoudt Telloport momenteel het overzicht van alle hardware en software componenten die aanwezig zijn op de slimme locaties?
* Welke problemen brengt de huidige manier van werken met zich mee?
* Hoe kan een *configuration* *management* *database* ( CMDB ) een oplossing bieden voor deze problemen?
* Hoe maak ik het overzicht zo duidelijk mogelijk?
* Welke informatie over de hardware en software componenten moet er worden verzameld en opgeslagen in de CMDB?
* Hoe kan deze informatie op de meest efficiënte manier verzameld worden?
* Wat zijn de belangrijkste functionele en niet-functionele vereisten waarmee rekening moet worden gehouden bij het ontwikkelen van een CMDB?
* Hoe kunnen deze vereisten en de informatiebehoeften van Telloport worden geïntegreerd in een webapplicatie voor configuratiebeheer die gebouwd wordt met Python en Django?
* Hoe kan ervoor gezorgd worden dat de informatie die wordt opgenomen in de CMDB eenvoudig up-to-date gehouden kan worden?
* Hoe kan de interactie tussen de informatiestromen in de CMDB en het bestaande Telloport platform tot stand worden gebracht?
* Binnen welke termijn zal de webapplicatie operationeel zijn?

## (Onderzoeks-)acties

De huidige methode voor het opslaan van informatie over hardware- en softwarecomponenten is onderzocht bij Telloport. Een aantal problemen met de huidige werkwijze werden vastgesteld. Om deze problemen op te lossen werd de mogelijkheid onderzocht om een *Configuration* *Management* *Database* ( DMDB ) te implementeren. Om een duidelijk overzicht te krijgen van de structuur van de CMDB werd onderzoek gedaan naar de verschillende structuren van andere systemen.

Om de CMDB te ontwikkelen, werd er onderzocht welke hardware- en softwarecomponenten er moeten worden opgeslagen. Om deze informatie efficiënt te verzamelen, werd onderzocht wat de meest effectieve manier is om de informatie te verzamelen. Daarnaast werden functionele en niet-functionele verreisten geïdentificeerd die bij de ontwikkeling van de CMDB moesten worden aangepakt. Vervolgens werd er onderzocht hoe deze eisen en informatiebehoeften van Telloport kunnen geïntegreerd worden in een webapplicatie voor configuratiebeheer die gebouwd wordt met Python en Django.

Om ervoor te zorgen dat de informatie in de CMDB up-to-date blijft, werdt er onderzocht hoe deze informatie op zo een eenvoudig mogelijke manier bijgewerkt kan worden. Daarnaast werdt er gekeken hoe er een interactie kan gemaakt worden tussen de informatiestromen in de CMDB en het bestaande Telloport platform.

Tot slot werdt er onderzocht hoelang het zou duren om een operationele webapplicatie te bouwen aan de hand van de tijdsraming.

## Verzamelde resultaten

* Hoe behoudt Telloport momenteel het overzicht van alle hardware en software componenten die aanwezig zijn op de slimme locaties?
  + Er werdt geen overzicht gehouden.
* Welke problemen brengt de huidige manier van werken met zich mee?
  + Dit kan leiden tot tijdsverspilling, zonder een *configuration* *management* *database* kan het tijdrovend zijn om zelf alle informatie van de componenten bij te houden. Verder kan er ook een mismatch ontstaan tussen componenten en slimme locaties. Zonder een CMDB kan het moeilijk zijn compalititeit van componenten te controleren.
* Hoe kan een *configuration* *management* *database* ( CMDB ) een oplossing bieden voor deze problemen?
  + Door een *configuration* *management* *database* te implementeren maakt het onderhoud op slimme locaties makkelijker en overzichtelijker doordat alle informatie en specificaties opgeslagen zijn in de CMDB.
* Hoe maak ik het overzicht zo duidelijk mogelijk?
  + Elk slimme locatie heeft een overzicht met al zijn componenten waarbij ook een detail pagina is voorzien voor alle specificaties van dat component. Verder zijn er ook nog overzicht tabellen voorzien voor alle componenten en slimme locaties voor makkelijk gebruik.
* Welke informatie over de hardware en software componenten moet er worden verzameld en opgeslagen in de CMDB?
  + Bij de hardware componenten kan dit gaan over het serie en modelnummer en andere spicificaties zoals bij een voeding hoeveel Watt en welk formaat de voeding is. De software die word bijgehouden is vooral de Android versies die de computers gebruiken in de slimme locatie. Verder worden de locaties en klanten bijgehouden van de slimme locaties.
* Hoe kan deze informatie op de meest efficiënte manier verzameld worden?
  + De informatie van de slimme locatie zelf worden gesynchroniseerd met het bestaande Telloport platform.
  + Ook een mogelijkheid zou zijn geautomatiseerde gegevensverzameling via API’s die aangeboden worden door leveranciers.
  + Verder zou een QR-code of barcode die gescanned word door bijvoorbeeld een mobiele apparaat alle informatie kunnen linken naar de CMDB een mogelijkheid kunnen zijn.
* Wat zijn de belangrijkste functionele en niet-functionele vereisten waarmee rekening moet worden gehouden bij het ontwikkelen van een CMDB?
  + Functionele vereisten zoals het toe voegen van nieuwe componenten aan de applicatie, inclusief relevante informatie zoals serienummers en specificaties. Ook de functionaliteit om bestaande componenten makkelijk terug te vinden en hun gegevens, zodat er een duidelijk overzicht is van de componenten. Verder is er dan de functionaliteit om de locatie en informatie van de klant terug te vinden van de slimme locaties. Daarnaast de mogelijkheid om onderhoudsactiviteiten te beheren van de slimme locaties zoals reparaties, vervangingen en wie de onderhoud heeft uitgevoerd.
  + Niet functionele vereisten zoals een gebruikersvriendelijke interface die gemakkelijke navigatie en efficiënt gebruik mogelijk maakt. Verder is er ook nog de schaalbaarheid van de applicatie. De applicatie moet in staat zijn om een groeiend aantal slimme locaties en componenten te ondersteunen zonder verlies van prestaties.
* Hoe kunnen deze vereisten en de informatiebehoeften van Telloport worden geïntegreerd in een webapplicatie voor configuratiebeheer die gebouwd wordt met Python en Django?
  + Dit is mogelijk door het maken van gevensmodels in Django met specificaties en relaties van de slimme locaties en componenten. Verder worden Django views gebruikt voor de functionaliteit en templates om dit te weergeven op de webapplicatie.
* Hoe kan ervoor gezorgd worden dat de informatie die wordt opgenomen in de CMDB eenvoudig up-to-date gehouden kan worden?
  + Een gebruikersvriendelijke interface voor de applicatie, zodat er gemakkelijk en snel wijzigingen kunnen aangebracht worden.
  + Verder zou er een mogelijkheid kunnen zijn om geautomatiseerde gegevens binnen te halen van leveranciersdatabases. Waardoor de gegevens automatisch worden bijgewerkt aan de hand van externe bronnen.
* Hoe kan de interactie tussen de informatiestromen in de CMDB en het bestaande Telloport platform tot stand worden gebracht?
  + Het gebruik maken van de API van het platform om zo al bestaande informatie binnen tehalen zoals de slimme locaties en hun gegevens. Verder word deze informatie automatisch gesynchroniseerd waardoor de slimme locaties up-to-date zijn.
* Binnen welke termijn zal de webapplicatie operationeel zijn?
  + Naar het einde toe van deze graduaatsproef zal er een mini CMDB operationeel zijn die uitgebreid zal kunnen worden.

## Stakeholders

De opdracht wordt gerealiseerd voor de werknemers van Telloport.

## Doelstellingen

* Opstellen van het datamodel voor de verschillende componenten, welke informatie moet er opgeslagen worden, hoe verhouden de componenten zich tot elkaar in het Telloport ecosysteem, wat zijn configuratieparameters waarmee rekening gehouden moet worden.
* Overzicht van de componenten per slimme locatie.
* Detailpagina van de componenten met de verschillende parameters.
* Mogelijkheid om een component aan te maken, wijzigen en te verwijderen.
* Documentatie van de webapplicatie.
* Verslag met bevindingen.

# Conclusies en aanbevelingen

## Conclusies

In dit onderzoek is onderzocht hoe de hardware- en softwarecomponenten op de slimme locaties van Telloport overzichtelijk in kaart gebracht en beheerd kunnen worden met behulp van een webapplicatie die gebouwd zal worden met Python en Django waarbij er een integratie is met het bestaande Telloport platform binnen een semester tijd.

Uit het onderzoek blijkt dat het mogelijk is om hardware- en softwarecomponenten van de slimme locaties van Telloport op een overzichtelijke manier in kaart te brengen met behulp van een webapplicatie. Door het gebruiken van Python en het *framework* Django, kon er een overzichtelijke en gebruikersvriendelijke applicatie ontwikkeld worden.

De ontwikkeling vereisten een uitvoerige analyse van het bestaande platform en de vereisten van de webapplicatie. Door iteratief te werken en de feedback van de werkplekcoach op te nemen kon er een oplossing gerealiseerd worden die voldoet aan de vereisten.

De integratie met het bestaande Telloport platform is gelukt door middel van de API’s dat het platform gebruikt. Waardoor er een efficiënte en geautomatiseerde gegevens stroom van slimme locaties werd doorgeven naar de webapplicatie en de slimme locaties synchroniseert.

Het behalen van een werkende minimale CMDB binnen één semester tijd was een uitdaging, maar met een goede planning en motivatie kon de webapplicatie tijdig worden opgeleverd.

Dus dit onderzoek heeft aangetoond dat het mogelijk is om de hardware- en softwarecomponenten van de slimme locaties van het bedrijf overzichtelijk in kaart te brengen en te beheren door een webapplicatie die ontwikkelt is met Python en Django. De webapplicatie zorgt voor een efficiënte en geïntegreerde oplossing voor het beheer van Telloport’s slimme locaties, wat voor een verbeterde operationele effectiviteit en tijd besparing.

## Aanbevelingen

Tekst tekst tekst

# Technische beschrijving

## Techologieeën

De CMDB is ontworpen met de technologieën zoals Django, een open source webapplicatieframework geschreven in Python. Daarnaast zijn er ook verschillende programmeertalen gebruikt zoals HTML, CSS , Javascript en een beetje JQuery voor de Front-End.

## Tools

Er zijn verschillende tools gebruikt om de ontwikkeling te vergemakkelijken zoals Git, een versiebeheersysteem voor het beheren van de code, text-editor Visual Studio Code en draw.io voor het maken van een ERD schema. Ook zijn er verschillende Python packages gebruikt:

* Redis:
  + Wordt gebruikt om veel gebruikte gegevens te *cachen* en de prestaties van de toepassing te verbeteren.
* Numerize:
  + Wordt gebruikt in de applicatie om grote getallen te formateren naar een afkorting waardoor het makkelijker te begrijpen en te lezen is.
* Select2:
  + Wordt gebruikt om *autocomplete* functionaliteit te bieden voor zoekvelden in de applicatie. Dit bespaart tijd en maakt de applicatie gebruiksvriendelijker.

Tot slot is er ook nog Zoho Learn om documentatie en interactieve cursussen te schrijven.

# Beschrijving uitwerking softwaretoepassing

## Aanpak

De aanpak voor de ontwikkeling van de *configuration* *management* *database* bevat verschillende fases. Allereerst was er de analyse fase waarin er een grondige analyse werd gedaan van de vereisten en de mogelijke componenten die beheerd moeten worden in deze CMDB. Vervolgens werd de datamodel opgesteld met hoe de componenten zich verhouden tot elkaar in het Telloport ecosysteem, wat zijn de configuratieparameters waarmee rekening gehouden moet worden.

Vervolgens in de tweede fase werd de webapplicatie ontwikkeld op basis van het Django *framework*. Waaronder het programmeren van de verschillende functionaliteiten, het testen op bugs en het optimaliseren van de prestaties. Het resultaat van deze fase is een minimale webapplicatie die een overzicht kan tonen van een component en eventueel te wijzigen, aan te maken en te verwijderen.

Ten slotte in de laatste fase werd de documentatie geschreven van de webapplicatie in Zoho Learn en een verslag geschreven met al de bevindingen en aanbevelingen voor de toekomst.

## Resultaat

Het resultaat hiervan is een minimale CMDB webapplicatie die voldoet aan de vereisten van Telloport. Zij hebben toegang tot een interface waarmee ze slimme locaties, componenten en onderhoud kunnen beheren. De webapplicatie bied een oplossing die de tijdsbesparing en overzicht verbetert.

# Persoonlijke reflecties en kritische kanttekeningen

Tijdens mijn graduaatsproef heb ik een waardevolle leerervaring gehad. Door het project heb ik de mogelijkheid gekregen om nieuwe programmeer talen te leren Python en Django. Het realiseren van de webapplicatie was uitdagend maar leerzaam. Doordat ik nog geen kennis had van Python en Django was dit eerst even twijfelen of ik dit project zou doen. Maar met het doorlopen van de documentatie en het volgen van tutorials wist ik dat het me wel moest lukken. Ik realiseer me da het vermogen om nieuwe technologiën te leren veel hoger is geworden met het volgen van de richting Programmeren.

Tijdens het ontwikkelen van de webapplicatie kwam ik ook enkele uitdagingen en leerpunten tegen. Bijvoorbeeld bij het plannen en ontwerpen van de webapplicatie had ik meer tijd moeten nemen voordat ik begon aan de ontwikkelfase.

Al met al ben ik trots met de webapplicatie ik ontwikkeld heb met nieuwe programmeertalen waar ik nog geen kennis van had in binnen de bepaalde tijd. Het project heeft me veel bijgeleerd.

# Referentielijst

Programming with Mosh. (2020, 16 september). *Python for Beginners - Learn Python in 1 Hour* [Video]. YouTube. <https://www.youtube.com/watch?v=kqtD5dpn9C8>

Traversy Media. (2021, 29 september). *Python Django 7 Hour Course* [Video]. YouTube. <https://www.youtube.com/watch?v=PtQiiknWUcI>

Djangoproject [Django]. (z.d.). *Django documentation*. Djangoproject. <https://docs.djangoproject.com/en/4.2/>

*Redis*. (z.d.). Redis. <https://redis.io/>

*numerize*. (2018, 14 augustus). PyPI. <https://pypi.org/project/numerize/>

*Select2 - The jQuery replacement for select boxes*. (z.d.). <https://select2.org/>

# Bijlagen

*Je voegt hier je genummerde bijlagen toe (bijv de enquête die je hanteerde, de uitprint van bepaalde resultaten, ….) die belangrijk zijn om het verslag op een goede manier te kunnen ‘lezen’ en ‘interpreteren’.*

Tekst tekst tekst