Proeven Van Bekwaamheid verslag



Toleco BV.

Jeroen Zonneveld

Michael Bloemraad

3-6-2024

**Inhoudsopgave**

[**Inleiding: 4**](#_heading=h.gjdgxs)

[**B1-K1: Realiseert software 5**](#_heading=h.30j0zll)

[*K1.1 Eisen en wensen 5*](#_heading=h.1fob9te)

[WP1 - De eisen en wensen zijn verwerkt in de user stories. 5](#_heading=h.3znysh7)

[WP2 - De user stories voldoen aan de criteria (wie, wat, waarom en realistisch). 6](#_heading=h.2et92p0)

[*K1.2 Ontwerp 7*](#_heading=h.tyjcwt)

[WP1 - De user stories zijn vertaald naar een passend, eenduidig en volledig ontwerp (sluit aan op wensen en eisen). 7](#_heading=h.3dy6vkm)

[WP2 - Er is gebruikgemaakt van relevante of toepasselijke schematechnieken (bijv. activiteitendiagram, klassendiagram, ERD, use case diagram). 7](#_heading=h.1t3h5sf)

[WP3 - De gemaakte keuzes in het ontwerp zijn onderbouwd met steekhoudende argumenten, waarbij rekening is gehouden met bijv. ethiek, privacy en security. 9](#_heading=h.4d34og8)

[*K1.3 Realisatie 9*](#_heading=h.2s8eyo1)

[WP1 - Er is voldoende inhoud van de user stories gerealiseerd binnen de gestelde/geplande tijd. 9](#_heading=h.17dp8vu)

[WP2 - De opgeleverde functionaliteiten voldoen aan de eisen en wensen zoals omschreven in de betreffende user story. 10](#_heading=h.3rdcrjn)

[WP3 - De kwaliteit van de code is goed. Dit uit zich onder andere in: OOP, objectstructuur, MVC, validatie, efficiëntie, foutafhandeling en terugkoppeling, security (veilig programmeren). 12](#_heading=h.26in1rg)

[WP4 - De code is gestructureerd opgesteld volgens code conventions. 16](#_heading=h.lnxbz9)

[WP5 - De code is verzorgd, leesbaar, gestructureerd en voorzien van zinvol commentaar. 17](#_heading=h.35nkun2)

[WP6 - Versiebeheer is effectief toegepast. 20](#_heading=h.1ksv4uv)

[WP7 - De voortgang is bewaakt en de juiste keuzes/afwegingen zijn gemaakt op basis van prioriteiten. 20](#_heading=h.44sinio)

[*K1.4 Testen 21*](#_heading=h.2jxsxqh)

[WP1 - De testcases in het testplan sluiten aan op de user stories en bevatten alle scenario's. 21](#_heading=h.z337ya)

[WP2 - De stappen, het gewenste resultaat en testdata zijn benoemd. Niet alleen het hoofdscenario, maar ook alternatieve scenario’s. 23](#_heading=h.3j2qqm3)

[WP3 - Het testrapport bevat de juiste resultaten en conclusies. 23](#_heading=h.1y810tw)

[*K1.5 Verbetervoorstellen 25*](#_heading=h.4i7ojhp)

[WP1 - De juiste verbetervoorstellen zijn gedaan vanuit het testen. 25](#_heading=h.2xcytpi)

[WP2 - De juiste verbetervoorstellen zijn gedaan vanuit de oplevering. 25](#_heading=h.1ci93xb)

[WP3 - De juiste verbetervoorstellen zijn gedaan vanuit de reflectie. 25](#_heading=h.3whwml4)

[**B1-K2: Werkt in een ontwikkelteam 26**](#_heading=h.2bn6wsx)

[*K2.1 Overleggen 26*](#_heading=h.qsh70q)

[WP1 - De kandidaat neemt actief deel waarbij relevante onderwerpen worden ingebracht en de juiste vragen worden gesteld. 26](#_heading=h.3as4poj)

[WP2 - De kandidaat stemt regelmatig en tijdig af met projectteamleden en opdrachtgever over de voortgang en eventuele knelpunten. 27](#_heading=h.1pxezwc)

[WP3 - De gemaakte afspraken zijn eenduidig vastgelegd. 27](#_heading=h.49x2ik5)

[WP4 - De kandidaat houdt zich aan gemaakte afspraken. 27](#_heading=h.2p2csry)

[*K2.2 (Deel)oplevering 28*](#_heading=h.147n2zr)

[WP1 - De kandidaat presenteert een overtuigend, duidelijk, beargumenteerd verhaal, afgestemd op de doelgroep. 28](#_heading=h.3o7alnk)

[WP2 - De kandidaat stelt gerichte vragen om te controleren of de betrokkenen goed geïnformeerd zijn over het opgeleverde werk. 34](#_heading=h.23ckvvd)

[WP3 - De kandidaat reageert adequaat op feedback. 34](#_heading=h.ihv636)

[*K2.3 Reflecteren 35*](#_heading=h.32hioqz)

[WP1 - De kandidaat benoemt zowel positieve als verbeterpunten van het proces van zowel eigen als teamprestaties. 35](#_heading=h.1hmsyys)

[WP2 - De kandidaat reageert adequaat op de ontvangen feedback. 35](#_heading=h.41mghml) *36*

# Inleiding:

**Het begin van Toleco?**

De samenwerking van Ruud en Toine begon in 2008. Deze samenwerking intensiveert in 2010 als Ruud de interne (stage)begeleider wordt van Toine, die dan start met zijn deeltijdopleiding Logistics Engineering. Het is het begin van vele werk-inhoudelijke gesprekken en in de jaren die volgen het opleveren van vele oplossingen in de dagelijkse bedrijfsvoering. Ruud en Toine groeien apart van elkaar in verschillende functies door de rangen van de organisatie. Het contact blijft bestaan als Ruud en Toine hun werkzaamheden vervolgen bij andere werkgevers. Het altijd samen blijven brainstormen over proces- en organisatieontwerp, masterdatabeheer, data-acquisitie en vormen van rapportages met de bedoeling om de bedrijfsvoering te verbeteren brengt Ruud en Toine eerst maandelijks, later wekelijks en vervolgens dagelijks bij elkaar.

In 2023 besluiten Toine Leidelmeijer en Ruud Overmars samen te gaan werken in Toleco BV.

**Wat doet Toleco?**

Toleco biedt gespecialiseerd advies op het gebied van proces- en organisatieontwerp en in het digitaliseren van bedrijfsvoering, met een focus op de levensmiddelen producerende industrie, met name MKB-bedrijven.

Hun aanpak omvat het ontwikkelen, produceren en implementeren van (op maat gemaakte) software oplossingen, waarbij ze ook aanzienlijke aandacht besteden aan databeheer en ondersteuning. Daarnaast bieden ze deskundig project- en interim-management om ervoor te zorgen dat de systemen en informatiestromen optimaal kunnen functioneren en dat het team de nodige ondersteuning krijgt om succesvol te zijn in de digitale transformatie van de organisatie, waardoor ze efficiënter en effectiever kunt opereren.

De eigenaar van Toleco is gespecialiseerd in reflex 3000. Dit is een speciaal ontwikkeld softwarepakket gebouwd voor de foodindustrie. Toleco bouwt dan ook scripts om de logistieke processen te vertalen naar een passende ERP-inrichting ( enterprise resource planning ). na een tijdje werd er gerealiseerd dat het maken van een ERP ook vroeg om procesmatige en organisatorische veranderingen nodig waren.

Tijdens mijn stage ben ik vooral bezig geweest met het helpen van opbouwen van de backend / automatisering van toleco zijn software om meer bedrijven te kunnen helpen.

Python is de meest gebruikte taal binnen Toleco, dit werkt namelijk snel en makkelijk en je kan er van alles mee doen, van webapps tot applicaties en scripts voor automatisering van data.

In toleco werken ze volgens de Scrum methodiek. Dit houdt in dat we in sprints van een tot twee weken. De stukken die we opleveren zijn allemaal kleine deeltjes en aanpassingen die langzaam verschillende verbeteringen en automatiseringen zijn voor bedrijven. Daarnaast werken we aan een backend om zo de automatisering makkelijker te maken.

# B1-K1: Realiseert software

## K1.1 Eisen en wensen

### WP1 - De eisen en wensen zijn verwerkt in de user stories.

Toine en Ruud zijn de hoofdcontactpersonen in het bedrijf en gaan met de klanten in gesprek aan de hand daarvan komt er een vergadering hoe wij de klant kunnen helpen. Vaak komen er ook verzoeken vanuit Toine zelf, dit komt omdat het bedrijf net begonnen is en er nog een hoop moet gebeuren zoals front en back -end.

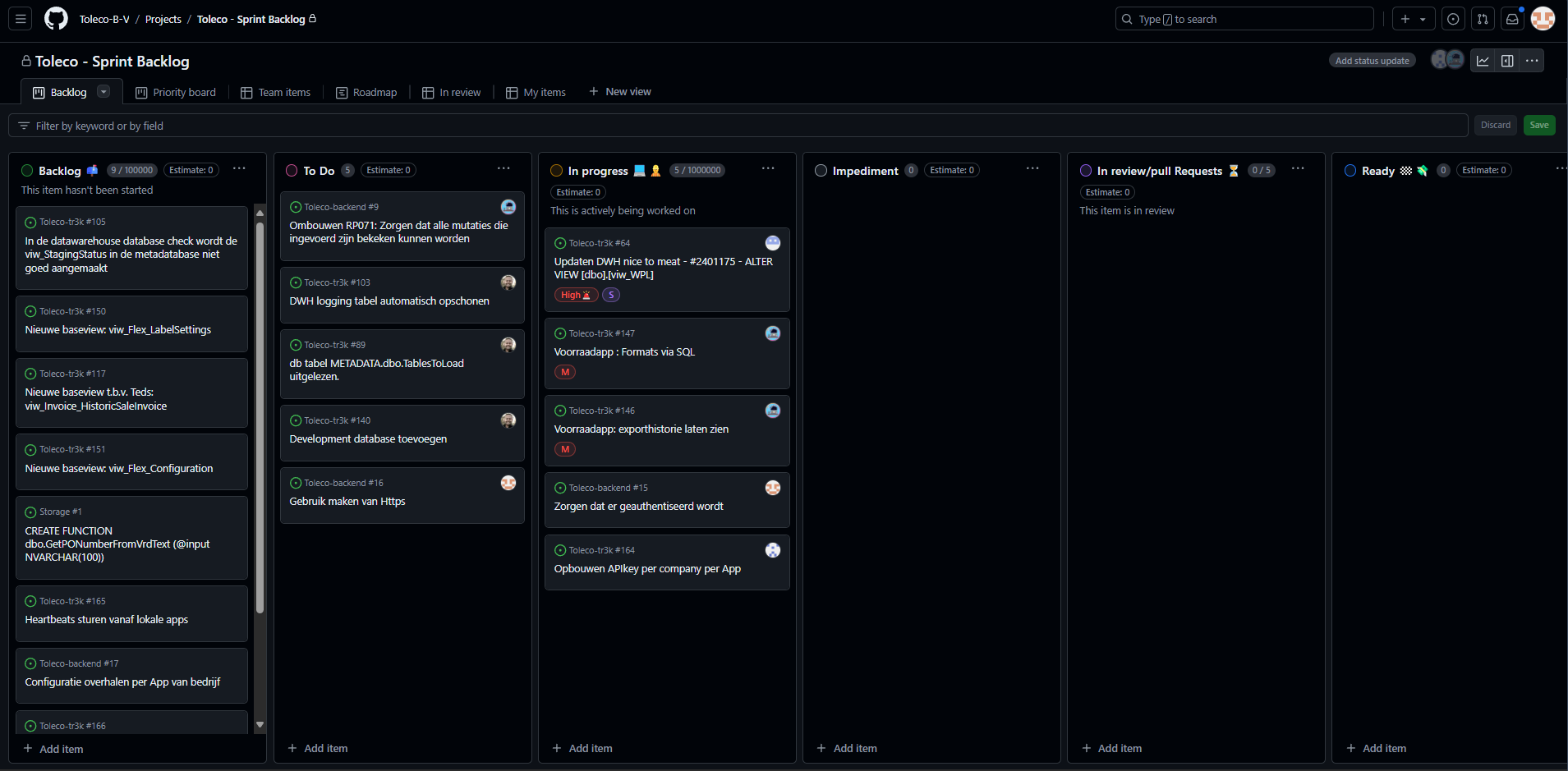
Het moment dat ik hier kwam stagelopen waren ze net klaar met de basis voor het beginnen van uitwerken van de backend waar alle services en apps moeten gaan draaien.

Aan de hand hiervan zijn we met de developers in gesprek gegaan, Er is besproken dat de backend online moet gaan draaien via een onlineservice genaamd ‘Heroku’.

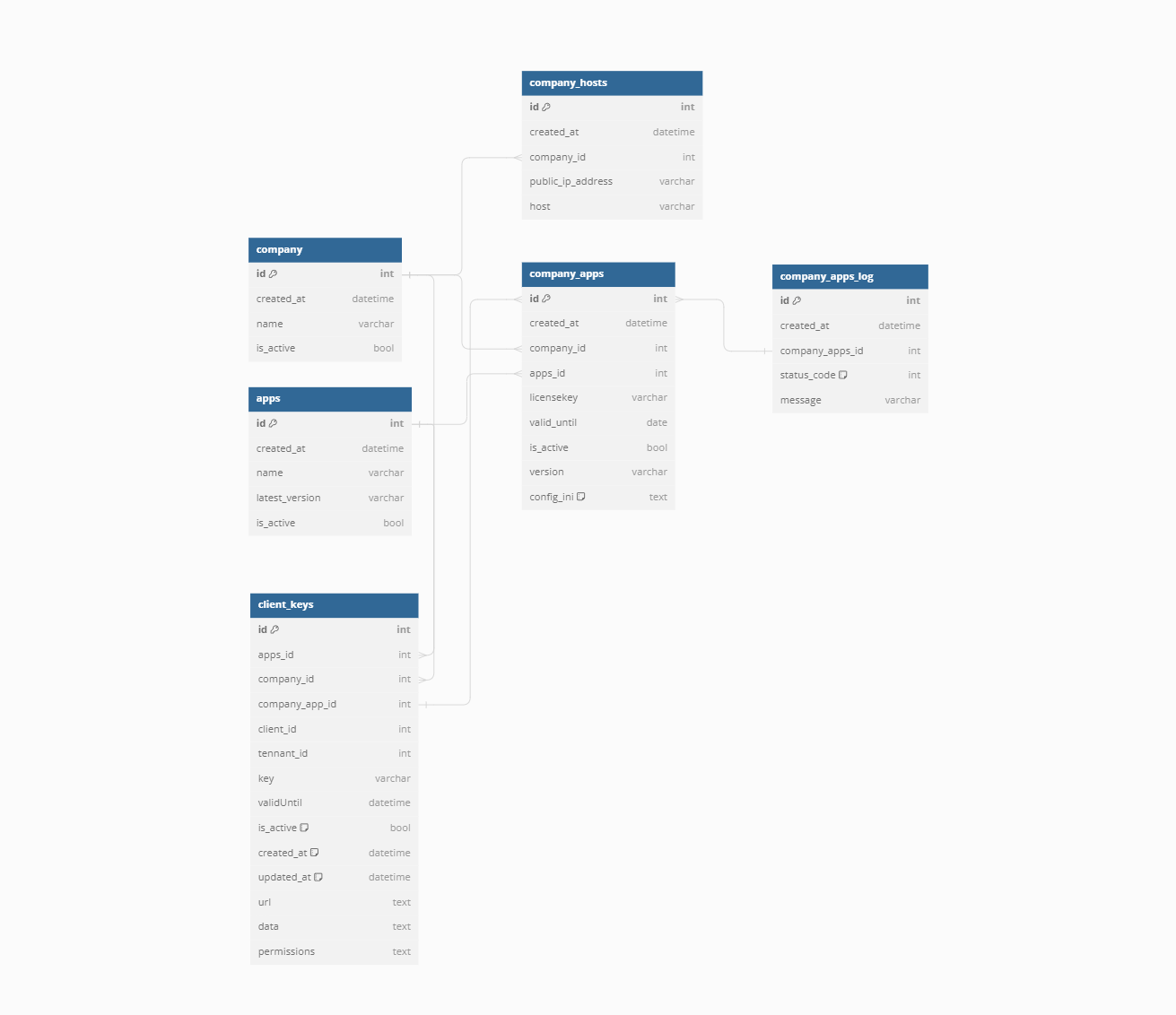
Heroku is een online cloud applicatie service. Dit houdt in dat we onze applicaties online willen laten draaien zodat klanten er op een makkelijke manier bij kunnen.

Mijn eerste taak hierin is, het opzetten van de connectie van de backend naar Heroku,

Anders gezegd ik moet zorgen dat de backend kan draaien via Heroku.

*Afbeelding 1: Een voorbeel hoe wij ons werk verdelen aan de hand van Scrum Methode, dit word gebruikt tijdens de bespreking onder alle leden van toleco)*

Voordat we begonnen met het opzetten van de backend en de connectie naar heroku moest eerst het database model gemaakt worden. Samen met de andere developers hebben we een database model gemaakt. Aan de hand van dit database model hebben we ongeveer uitgewerkt hoe we het einddata model voor onze backend willen gaan opzetten.

*Afbeelding 2 – beginnend Database Model*

In afbeelding 2 zei je ons beginnend database model. Aan de hand van deze uitwerking hebben we aan de hand van python en het framework Django een database gemaakt.

### WP2 - De user story's voldoen aan de criteria (wie, wat, waarom en realistisch).

Aan de hand van de scrum meetings worden alle dingen die in de eerste tab( back log ) staan altijd ingekeken om te kijken welke dingen nog gedaan kunnen worden. Dan gaan we stap voor stap langs wie wat gaat doen. Met Het tabje (back log) dat aan jou gelinkt is heb je 2 weken de tijd om te kijken of je het afkrijgt en of hoever je komt. Lukt dit niet kan er altijd gekeken worden waar je op vast loopt en hoeveel tijd je nog nodig hebt. Realistisch gezien omdat het vaak kleine stapjes zijn kun je meerdere dingen doen per week. Soms krijg je ook een groter project naar je geschoven en hier doen je dan net wat langer over.

## K1.2 Ontwerp

### WP1 - De user story's zijn vertaald naar een passend, eenduidig en volledig ontwerp (sluit aan op wensen en eisen).

Het ontwerp dat we gemaakt hebben is gecontroleerd door iemand die remote werkt voor het development team van toleco. Aansluitend op ons idee is het passend, eenduidig en volledig ontworpen.

Zoals ik in het vorige werkproces vermeld heb ben ik vooral bezig geweest met het backend gedeelte van het nieuwe idee van toleco hiervan. Aan de hand van het database ontwerp wordt er gekeken naar de eisen en wensen die vertaald zijn in het functioneel ontwerp.

### WP2 - Er is gebruikgemaakt van relevante of toepasselijke schematechnieken (bijv. activiteitendiagram, klassendiagram, ERD, use case diagram).

Toleco werkt samen met Scrum techniek dit is een methode dat je een korte periode hebt om een bepaald iets af te krijgen en of zoveel mogelijk probeert te komen.

Elke begin van de maandag hebben we een meeting waarin word besproken wie wat gaat doen.

Op maandag 15 juli heb ik een meeting gehad met de Developers, daarin hebben we besproken hoe we de backend willen gaan aanpakken na dat de backend draait op heroku.

Aan de hand van die meeting heb ik speciaal voor dit werkproces een activiteitendiagram gemaakt over de connectie en de beveiliging van onze apps en de backend. (zie bijlage 1) Voor het diagram heb ik Unified Modeling Language (UML) gebruikt. Op school is mij UML in A diagram of a company

Description automatically generated

*Bijlage 1 - Activiteiten diagram*

### WP3 - De gemaakte keuzes in het ontwerp zijn onderbouwd met steekhoudende argumenten, waarbij rekening is gehouden met bijv. ethiek, privacy en security.

Wat betreft ethiek en privacy hoeven wij geen rekening te houden bij het technische ontwerpen/bouwen. Dit omdat er geen gebruik gemaakt wordt van productiedata in de ontwikkelomgeving.

Voor security daarentegen moeten we rekening houden met de gebruikersrechten.

Voor binnen ons bedrijf is eigenlijk niets zeer streng beveiligd, alles is hier verdeeld in groepen en elke groep heeft zijn toegang tot zijn eigen omgeving, maar het is niet dat alles wat IT doet blokkeren voor de mensen die er wat minder van weten. Verbinding van buitenaf naar ons API Endpoints is wel allemaal geblokkeerd ten zij je een APIKey beschikt.

Een ApiKey is een volgorde van cijfers en letter die automatisch aangemaakt worden.  
Per company/ klant krijgen ze een eigen key zonder deze key kan je geen toegang krijgen tot onze gegevens er services die we uiteindelijk via de backend willen gaan laten draaien.

## K1.3 Realisatie

### WP1 - Er is voldoende inhoud van de user stories gerealiseerd binnen de gestelde/geplande tijd

Bij Toelceo werken we in sprints van een week. Gedurende deze weken werken we aan de tickets die je hebt gekregen bij de scrum meeting op de maandag. De maandag erop wordt er gekeken waar je staat, of je vast loopt en of hoelang het nog kan duren.

Tot nu toe is alles goed verlopen en heb ik de hulp kunnen krijgen als ik vastzat.

Soms heb je grootte projecten in je lijstje staan en kan deze gesplit worden onder meerdere mensen. Vaak zijn het dus allemaal kleine projectjes die samen uiteindelijk een groot applicatie gaat vormen.

Met de Scrum methodiek wordt ingeschat hoelang jij nog doet over het project,

Aan de hand van de ticket wordt er geschat hoelang jij ongeveer zal doen over de ticket die je hebt gekregen. Aan het begin dat ik hier kwam stage lopen was dit 1 week, dit is nu veranderd naar 2 weken.

### WP2 - De opgeleverde functionaliteiten voldoen aan de eisen en wensen zoals omschreven in de betreffende user story.

Elke woensdag en vrijdag heb ik een gesprek met een senior developer genaamd Boris.

Dit doen we online via teams samen kijken we na wat ik gedaan heb en of ik nog tegen dingen ben aangelopen. Hier bespreek ik samen met hem wat de volgende plannen zijn voor de backend. De plannen zijn allemaal door besproken met Toine.  
Er wordt ook besproken op de wensen zijn vervuld over de dingen die zijn besproken,

Omdat we vaak een gesprek samen hebben kan er eigenlijk weinig gebeuren met het idee dat er bedacht is voor de backend. Als er dingen zijn die lastig op te lossen zijn gaan boris en ik samen met toine zitten en bespreken we het met ze alle hoe we het probleem kunnen oplossen. Dan leggen we de ideeën op tafel aan de hand daarvan maken we een beslissing.

**Api key beveiliging:**

Een van de complexere codes die ik opgepakt heb was het maken en laten werken van een api key. Het probleem was dat wanneer er verbinding word gemaakt met onze api endpoints oftewel verbinding met onze database er geen beveiliging was die dit tegen hield iedereen kon dus data sturen en ontvangen als ze de juiste url wisten te vinden. Ik heb een API Key autorisatie gebouwd zodat er geen toegang meer is vanaf buiten af ze kunnen nu niks opvragen of versturen naar onze database. Dit is een stuk code die ik gebruikt heb :

A screen shot of a computer program

Description automatically generated

*Afbeelding 4a – ApiKey Check*

*Op deze afbeelding zie je een deel van mijn code voor de apikey.*

Ik zal in het kort uitleggen wat deze code ongeveer in houdt,

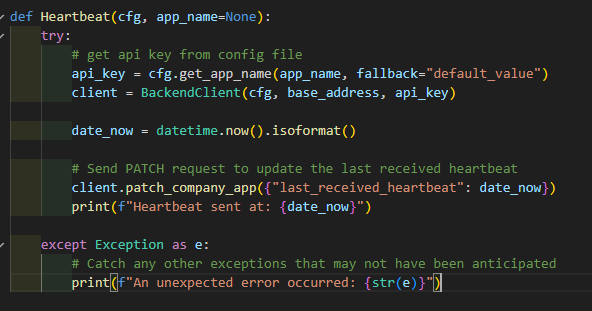
Regel 13 en 14 word er gekeken of de persoon/bedrijf die een request stuurd in bezit is van een apikey dit gaat via een header genaamd X-API-Key. Als er geen apikey aanwezig is zal lijn 16 en 17 geactiveerd worden en krijg je een error dat er geen apikey is meegegeven.

Vanaf lijn 19 decrypt python de api key en controllerd of de apikey geledig is.  
decrypten is uit elkaar halen van de key op een speciaale volgorde. En als de code niet klopt dan krijg je terug dat de api key die mee gestuurd is niet geldig is.

### WP3 - De kwaliteit van de code is goed. Dit uit zich onder andere in: OOP, objectstructuur, MVC, validatie, efficiëntie, foutafhandeling en terugkoppeling, security (veilig programmeren).

Hier volgen een aantal screenshots van stukken code die ik geschreven heb:

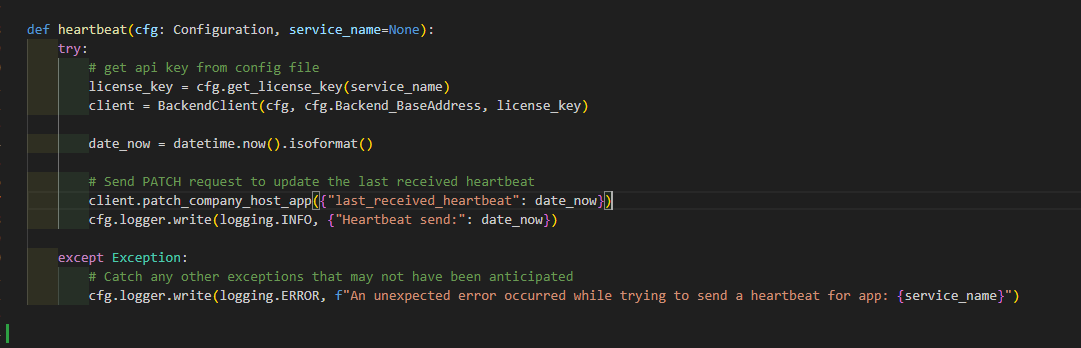
*Afbeelding 5 – Kwaliteitscontrole checks voor het wordt doorgestuurd naar een andere programeer omgeving.*

*Afbeelding 6 – Een stukje van de heartbeat plus wat aantekeingen zo dat je kunt zien wat wat doet*  


Wanneer we stukken code toevoegen aan bestaande code zetten we commentaar erbij over wat het allemaal inhoud. Dit maakt het overzichtelijk voor iedereen die later nog in de code wil gaan werken. En of de code moet gaan na kijken.

Ik ben bezig geweest met het bouwen van een Heartbeat die een signaal stuurt naar onze backend. Elke keer als de heartbeat gestuurd wordt word er een tijd meegegeven. Dat is de exacte tijd dat de heartbeat verstuurd word. Als de tijd langer dan 1 min niet geüpdatet is kunnen wij weten dat de service (applicatie) bij de klant is afgesloten of een error heeft.

*Afbeelding 7 – het stuk code waar de heartbeat word opgeroepen.*



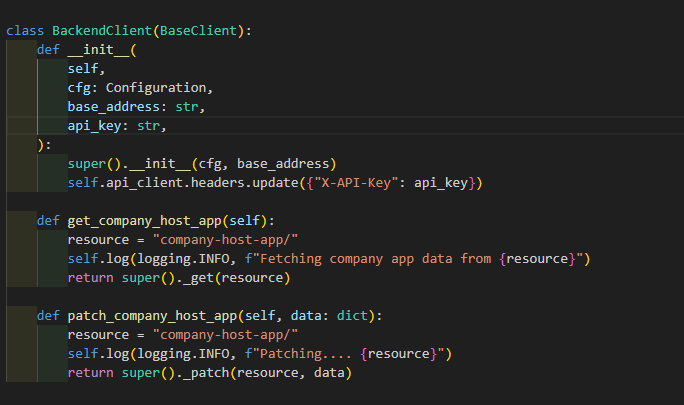
#### Heartbeat uitleg:

Wat doet de heartbeat

* heartbeat(cfg, service\_name)
  + De heartbeat is een singaal vanuit de klanten naar ons backend toe, aan de hand van dit singaal weten we of de service bij hun nog werkt en of uitgevallen is.

Op sommige plekken heb ik geen commentaar gezet omdat dat simpelweg niet nodig is.  
En spreekt de code voor zichzelf. Het is een regel die we sinds kort hebben toegevoegd om comentaar bij lastige stukken code te implementeren. Dit komt omdat niet alle code voor iedereen duidelijk is en zo makkelijker overzicht kan maken. Maar met als voorbeeld date.now dat zijn stukkjes die voor zichzelf spreken.

**Backend requests makkelijk maken**

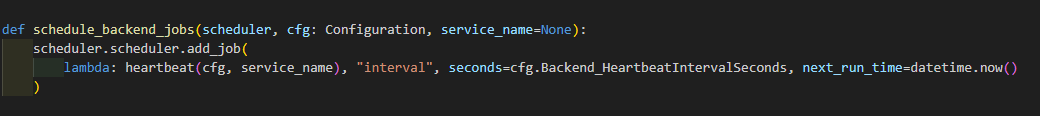
*Afbeelding 8 – Backend requests ( \_get en \_ patch)*  


Deze backend client heb ik gemaakt om dat we op een makkelijke manier moesten gaan kijken om dingen vanuit de backend op te kunnen halen zonder steeds dezelfde code te moeten gaan schrijven. Dus zijn we samen met het team bij elkaar gaan zitten en hebben we het probleem op tafel gegooid. Na een korte overleg zijn we op conclusie gekomen om een oud bestaand script ( apiclient ) om te gooien en zo meer functionaliteit uit dat script te kunnen halen. Het is nu zo omgegooid dat je nu makkelijk nieuwe requesten kan toevoegen en of gebruiken. Daarnaast is het veel over zichtelijker voor iedereen.

#### Functies:

Een functie die ik gemaakt heb is de “schedule\_backend\_job” functie. Deze functie heb ik gemaakt om makkelijker de heartbeat om de 15 seconden te laten runnen in de services die bij de klant draaien. Dit is hoe het eruit ziet:

*Afbeelding 9 – Tijd job voor de heartbeat*



### WP4 - De code is gestructureerd opgesteld volgens code conventions.

Binnen Toleco is er aan het begin van het programmeren in dit bedrijf niet heel veel tijd en aandacht voor geweest, steeds meer veranderingen word toegevoegt. Met namen linters en formatters als linter word nu gebruik gemaakt van Flake8 en als formatter word Black gebruikt, Deze 2 zijn speciaal gericht naar een manier van coderen genaamd pep 8

Voor het formateren van de code gebruiken we een standaard formatter gemaakt door meerdere mensen op het internet en ondersteund door Windows. Hierdoor weten we dat atijd alles dat gepusht word op de juiste manier geformatteerd is en geen eigen geleerde manier van coderen. Dat kan namelijk in de toekomst als je meer medewerkers hebt een groote puinhoop worden.

#### PBI-13709 KAM (Kwaliteit, Arbo, Milieu):

Voorheen werden er niet veel checks gedaan die controllerde hoe de code was en of het wel op de juiste manier geformateerd was , sinds kort laten we dit doen bij elke merge request.

*Afbeelding 10 – een github controlle die afgaat als het er een stuk code niet voldoet aan de waarde van onse style* A screenshot of a computer

Description automatically generated

Hierboven is te zien hoe we hier gebruik van maken.

### WP5 - De code is verzorgd, leesbaar, gestructureerd en voorzien van zinvol commentaar.

Binnen visual studio code maken wij gebruik van flake8 en Black. Wat dit voornamelijk doet is het toevoegen van standaarden aan ons codebase, waardoor alle developers de zelfde regels en stijl hebben en zo makkelijk elkaars code kunnen lezen.

Zoals te zien is in de code die ik gemaakt heb in het vorige werkproces is het gestructureerd en heb ik er commentaar bij gezet met wat het doet. Hierdoor is het makkelijk navigeren in lange stukken code. Hier nog wat meer voorbeelden van code die ik ontwikkeld heb met commentaar erbij. Op afbeelding 7 is ook een mooi voorbeeld te zien van goed geformatteerde code die ik geschreven heb met commentaar erbij.

### WP6 - Versiebeheer is effectief toegepast.

Aan het begin dat ik hier kwam werken bij toleco en hoe her er nu voor staat is er een hoop veranderd. Aan het begin dat ik hier kwam werd er al netjes met github gewerkt zelf wist ik hier nog niet zo heel veel van. Bij alle opdrachten die je kreeg moest je een nieuwe brench aan maken op de development brench, main was een brench waar de software die werkte vandaan gehaald werd dit werd ook wel voor de klant gebruikt. Elke keer als je een brench aanmaakte noemde je heb als voorbeeld 24098 dit betekend jaar 2024 brench nummer 98  
dit moesten we bij houden in een excel bestand.

Later is dit veranderd om dat er een slecht overzicht was met het werken in brences want developers gingen soms dubbel werken. Hoe het er nu uitziet is duidelijk en overzichtelijk.  
tijdens de sprint worden er kaartjes aangemaakt deze kaartjes hebben een nummer nu plaats je het wat de code die je gaat schrijven is: fix, hot fix, feature etc. Daarna plaats je het nummer van het kaartje de bij jou opdracht hoort. Hierna schrijf je of het voor de backend is of niet. Zie foto:   
A screenshot of a computer program

Description automatically generated

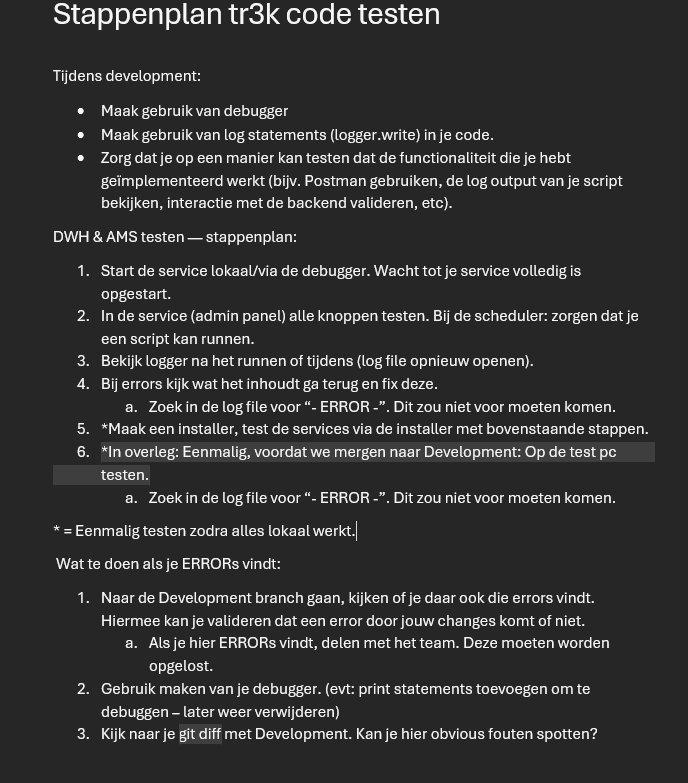
### WP7 - De voortgang is bewaakt en de juiste keuzes/afwegingen zijn gemaakt op basis van prioriteiten.

Elke woensdag bespeek ik met boris of ik de juiste keuzes maak tijdens het programmeren.  
hij begeleidt me hier op de juiste manier mee als hij ziet dat er dingen de verkeerde richting op zou gaan zal hij me er op aanspreken. Preoriteiten waren er niet heel veel aan begin van de weken dat ik hier kwam werken, Werd ik vrij gelaten met kleine opdrachtjes later toen we begonnen met de backend werd ik mee getrokken. En krijg ik meer verantwoording en preoriteiten bij sommige projecten.

## K1.4 Testen

### WP1 - De testcases in het testplan sluiten aan op de user stories en bevatten alle scenario's.

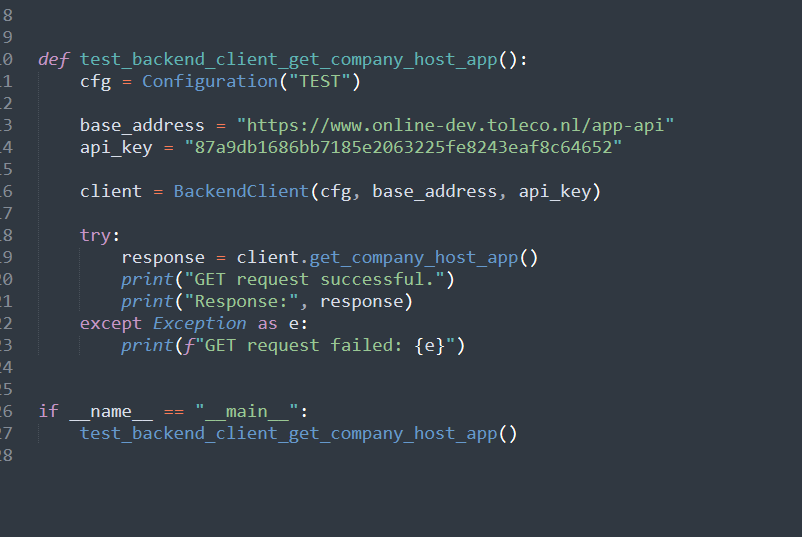
Hier bij toleco maken we geen gebruik van testplannen.  
hier voor heb ik een test plan gemaakt die andere mensen kunnen gebruiken om hun gemaakte software te kunnen testen en hoe ze dit kunnen testen alles staat er netjes in uitgeschreven. Hier bij toleco hebben we ook een software genaamd DWH dit moetst ook soms getest worden maar dat moet op een speciaale manier. Hier hebben boris en ik samen naar gekeken en een handleding heschreven hoe je dit kunt installeren en kan testen.

  
*word bestand waar in de meeste tips en stappen plan instaat voor testen van je code en DWH*

### WP2 - De stappen, het gewenste resultaat en testdata zijn benoemd. Niet alleen het hoofdscenario, maar ook alternatieve scenario’s.

Zoals je hier boven ziet heb ik de meest mogelijke scenarios op een duidelijke manier opgeschreven met stappen plannen. Dit zodat elke nieuwe stagair dit makkelijk kan gebruiken.

Als programmaeur ben je uiteindelijk zelf verantwoordelijk voor je eigen gescherven stukje code. Dingen die op niet makkelijk kon testen heb ik vaak zelf een test voor geschreven maar dit kwam zeer minimaal voor, als voorbeeld moest ik testen op de apikey waar bezig was ook dadelijk werkte ik moest hier een makkelijke database vebinden maken met een api key en zonder.



### WP3 - Het testrapport bevat de juiste resultaten en conclusies.

Uit de tests heb ik gevonden dat dat aan het begin de code die ik gemaakt had niet klopte en bepaalde dingen vergeten was. Uiteindelijk na veel trail en error ben ik er achter gekomen waar mijn foute zaten en ben ik op de juiste resultaten geindigd.  
dit stukje code hebben we nu ook toegevoegd in een map genaamd test

Hier in kan je testen vinden die al een keer gemaakt zijn en je kan ze zo gebruiken.

Om zo altijd achter het juiste relutaat te komen.

## K1.5 Verbetervoorstellen

### WP1 - De juiste verbetervoorstellen zijn gedaan vanuit het testen.

### WP2 - De juiste verbetervoorstellen zijn gedaan vanuit de oplevering.

### WP3 - De juiste verbetervoorstellen zijn gedaan vanuit de reflectie.

# B1-K2: Werkt in een ontwikkelteam

## K2.1 Overleggen

### WP1 - De kandidaat neemt actief deel waarbij relevante onderwerpen worden ingebracht en de juiste vragen worden gesteld.

**Stand up:**

Elke ochted gaan we in een rondje staan en vertellen we wat we gaan doen die dag, en je kan ook aangeven of je vast zit dat word besproken waneer je een afspraak kan inplannen

**Sprint meeting**

Om de 2 weken gaan we met ze alle zitten om te bespreken wie wat gaat doen.   
hierbij nemen we alle kaartjes die in de backlog door om te kijken welke hooge of laage preoriteiten hebben. Aan de hand daarvan verdelen we de kaartjes.

**Rationeel overleg**

Het rationeel overleg word 1 keer om de zoveel maanden gehouden er word dan besproken hoe het met de klanten gaat. En of er nog dingen kunnen verbeteren binnen bepaalde bedrijven

**DWH** ( DATAWAREHOUSE )   
Over de datawarehouse hebben we ook een meetingen gehouden hierin werden verteld wat de toekomst plannen waren voor deze service, we gingen met ze alle brainstormen of er nog op misschien nieuwe dingen konden komen.

### WP2 - De kandidaat stemt regelmatig en tijdig af met projectteamleden en opdrachtgever over de voortgang en eventuele knelpunten.

Elke woensdag en vrijdag heb ik een gesprek met Boris, hij is een soort van de lead developer die hier momenteel werkt dit gaat allemaal virtueel want hij is in het buitenland momenteel.  
deze gesprekken zijn harstike fijn want alles wat ik niet weet kan ik op een kladblok schrijven en hem er over vragen hoe bepaalde dingen zitten in de code of waarom hij bepaalde dingen op een bepaale manier aanpassen.

### WP3 - De gemaakte afspraken zijn eenduidig vastgelegd.

Alle afspraken die gemaakt worden worden in outlooks agenda opgeslagen.   
je plant een afspraak in om een tijd dat de debetrevende niks te doen in en plant een afspraak onder oveleg.

### WP4 - De kandidaat houdt zich aan gemaakte afspraken.

Alle afspraken die in mijn contract stonden houd ik me aan,

Ik doe geen dingen die niet toegstaan zijn zonder te vragen.

Aan het begin werd er niet heel erg gelet op de manier waarop geprogrammerd werd nu de linter en black fotmatter er is houdt ik me strict bij de regels.

## K2.2 (Deel)oplevering

### WP1 - De kandidaat presenteert een overtuigend, duidelijk, beargumenteerd verhaal, afgestemd op de doelgroep.

### WP2 - De kandidaat stelt gerichte vragen om te controleren of de betrokkenen goed geïnformeerd zijn over het opgeleverde werk.

### WP3 - De kandidaat reageert adequaat op feedback.

## K2.3 Reflecteren

### WP1 - De kandidaat benoemt zowel positieve als verbeterpunten van het proces van zowel eigen als teamprestaties.

### WP2 - De kandidaat reageert adequaat op de ontvangen feedback.