



Mocked sensor portal

Plan van aanpak

Bachelor in de Toegepaste Informatica
keuzerichting Applicatieontwikkeling

Anton De Houwer

Academiejaar 2019-2020

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

INHOUDSTAFEL

INHOUDSTAFEL	2
INLEIDING	3
1 STAGEBEDRIJF	4
1.1 VanRoey.be	4
1.2 .NET team	4
2 AANLEIDING EN ACHTERGROND	5
3 DOELSTELLING	6
4 BUSINESS CASE	7
5 FASERING	8
5.1 Analyse	8
5.2 Development	8
5.2.1 Backend API	8
5.2.2 Front-End portal	8
5.2.3 'Keep Alive'-service	8
5.3 Deployment	8
5.3.1 Continuous integration / Continuous Deployment (CI/CD)	8
5.4 Documentatie	9
5.4.1 Gebruikershandleiding	9
5.4.2 Swagger	9
5.4.3 DocFx	9
5.5 Testing	9
5.5.1 Unit Tests	9
5.5.2 User Acceptance Testing (UAT)	9
6 STAGEBEGELEIDER	10
6.1.1 Pull requests	10
6.1.2 Demo & live code review	10
6.2 Begeleidende docent	10

INLEIDING

Als onderdeel van mijn laatste jaar in de opleiding IT Factory aan Thomas More te Geel, ben ik begonnen aan mijn stage. Tijdens mijn zoektocht naar een stageplaats, ben ik terecht gekomen bij VanRoey.be. Ik heb hierbij namelijk gekozen voor een stage in .NET, omdat ik een grote interesse in .NET heb en VanRoey.be voor mij geen onbekende was.

In dit document stel ik eerst mijn stagebedrijf voor. Vervolgens geef ik wat meer uitleg over de aanleiding en achtergrond van mijn stageopdracht en beschrijf ik de doelstellingen van de opdracht. Dan wordt de business case besproken. Daarna licht ik de fasering van mijn opdracht toe. En ten slotte vertel ik hoe mijn rapportering zal gebeuren.

1 STAGEBEDRIJF

1.1 VanRoey.be

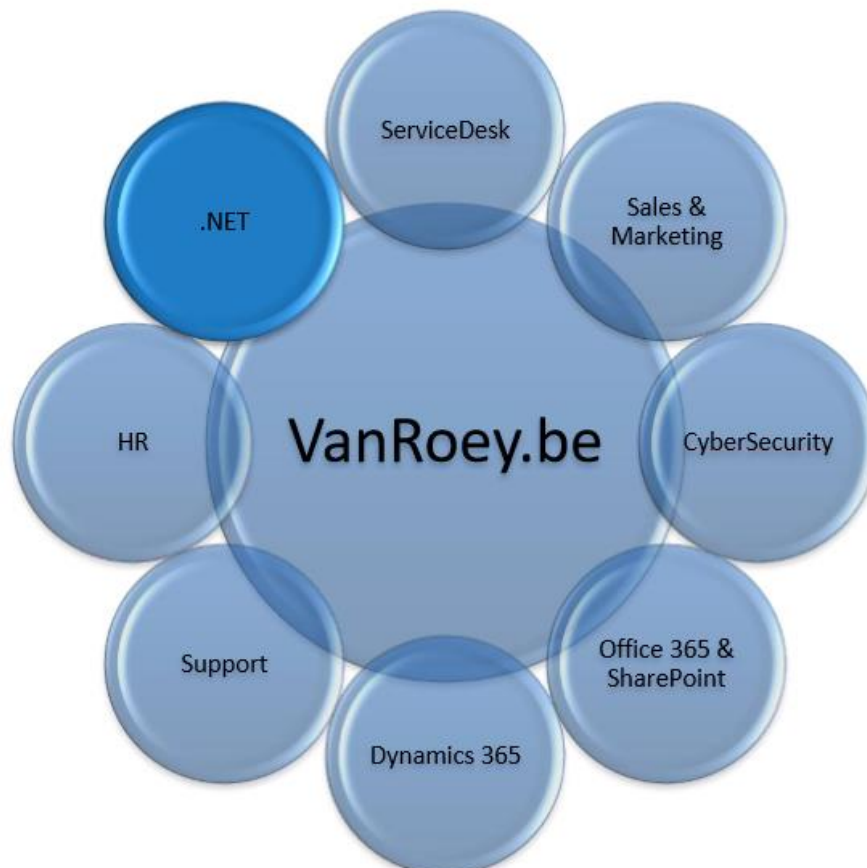
VanRoey.be is een sterk groeiend bedrijf dat vooral actief is in de IT-sector. Het bedrijf is ontstaan in 1993 in de garage van eigenaar Pat Van Roey. Ondertussen is het bedrijf uitgegroeid tot een NV met 4 kantoren in de provincie Antwerpen. In Geel is sinds 2019 het Digital Experience Center geopend waar bezoekers zich kunnen onderdompelen in de laatste nieuwe technologieën. Het hoofdkantoor bevindt zich in Turnhout, waar ik stage zal lopen.



1.2 .NET team



De specifieke afdeling waar ik stage zal lopen, is het .NET team. Dit team houdt zich bezig met het ontwikkelen van zowel interne als externe applicaties. Hierbij maakt het team (zoals de naam het zegt) gebruik van de .NET technologieën en frameworks van Microsoft. Als cloud-platform maakt het team gebruik van Azure.



2 AANLEIDING EN ACHTERGROND

Voor IoT projecten worden er soms sensoren gebruikt, een sensor maken en testen duurt echter veel te lang voor de developer die de berichten en/of gegevens nodig heeft om verder te kunnen gaan met zijn ontwikkeling. Hiermee verliest hij veel tijd en duurt het meestal heel lang voor een project klaar is wat ook niet goed is voor de klant. Dit probleem zou perfect kunnen opgelost worden als men deze sensoren sneller kan maken.

Uiteindelijk heeft VanRoey.be besloten om een mocked sensor portal te creëren. Dit portaal maakt het mogelijk om op een efficiënte manier virtuele sensoren aan te maken.

Op deze manier zullen projecten hierdoor veel sneller klaar zijn, waardoor er veel tijd zal worden bespaard. De klant zal ook vlugger tevreden zijn omdat zijn/haar aanvraag in kortere tijd zal worden gerealiseerd.



3 DOELSTELLING

Het project waar ik aan werk tijdens mijn stage is, zoals eerder aangehaald het creëren van een mocked sensor portal. Dit portaal maakt het mogelijk om virtuele sensoren aan te maken waardoor de programmeur niet moet wachten op de "echte" sensor die in de realiteit gemaakt wordt.

Dit portaal wordt opgesteld aan de hand van van meerdere faseringen die ieders een ander aspect van het project behandelt. Volgende faseringen zijn gedefinieerd:

- Analyse
- Development
 - Backend API
 - Front-End portal
 - *'Keep Alive'-service*
- Deployment
 - Continuous Integration / Continuous Deployment (CI/CD)
- Documentatie
 - Gebruikershandleiding
 - Swagger
 - DocFx
- Testing
 - Unit Tests
 - User Acceptance Testing (UAT)



Deze onderdelen worden in hoofdstuk 5 (fasering) uitgebreider toegelicht.

4 BUSINESS CASE

Wanneer het portaal klaar is voor productie, kunnen interne projecten dus ook sneller verlopen.

Daarnaast kunnen virtuele sensoren gedeeld worden voor andere developers. Als de sensor in realiteit nu ook af is kan hij de registratie van de virtuele sensor gewoon over nemen, en is de connectie meteen in orde. VanRoey.be zal hierbij ook kosten besparen door niet steeds een sensor te moeten aanschaffen voor projecten die voor klanten zijn.

Als de klant een webapplicatie wilt dat taken moet uitvoeren aan de hand van waardes die binnen komen van een IoT device, moet VanRoey.be hiervoor nu niet meer zelf een sensor in elkaar gaan steken maar kunnen zij gebruik maken van het Mocked Sensor Portal. Het gerdag van de Mocked sensoren zijn makkelijk te manipuleren.



5 FASERING

5.1 Analyse

Ik verdiep mezelf in Azure door middel van een cursus te volgen waarin de basis van Azure wordt uitgelegd. Hierna maak ik een analyse waar ik alle punten van beschrijf over hoe ik de opdracht ga uitwerken.

In deze analyse zet documenteer ik ook wat in er in scope zit en wat er out of scope zit. Ik leer ook goed na te denken over de tijd dat ik aan elk onderdeel van de opdracht ga spenderen. Dit noteer ik er dan telkens bij zodat ik aan het einde van mijn analyse een schatting heb van hoelang de opdracht in zijn geheel zal duren.

5.2 Development

5.2.1 Backend API

Ik ontwikkel een API in Asp.Net core, die data gaat wegschrijven in een SQL database, BLOB Storage en een Table Storage. Deze API gaat ook connectie moeten maken met een IoT Hub en/of IoT Central.

5.2.2 Front-End portal

Voor de Front-End ga ik gebruiken van Vue.js. Dit is een open-source JavaScript Framework dat gebruikt wordt voor het maken user interfaces en single-page applicaties. Mijn code zal ik schrijven in Typescript wat wel leuk is omdat dit helemaal nieuw voor mij is.

5.2.3 'Keep Alive'-service

Deze virtuele sensoren moeten geregistreerd worden bij een IoT Hub en/of IoT Central. Als de sensoren geregistreerd zijn moet ik er natuurlijk ook voor zorgen dat zij blijven draaien. Hiervoor zal ik een methode schrijven dat aan de hand van de frequentie van de sensor, zijn aanhangende payload zal versturen waardoor de sensor wakker blijft.

5.3 Deployment

5.3.1 Continuous integration / Continuous Deployment (CI/CD)

Vervolgens moet er Continuous Integration(CI) en Continuous Deployment(CD) opgezet worden.

CI gaat eerst het project helemaal testen. Deze zal meldingen geven als er iets fout loopt. Als dit het geval is, betekent het dat er iets moet aangepast worden in de code of configuratie.

Eens CI geslaagd is, zal CD ervoor zorgen dat de API/Function gehost wordt in de cloud.

5.4 Documentatie

5.4.1 Gebruikershandleiding

Voor dit portaal ga ik ook een gebruikershandleiding schrijven in word. Deze handleiding zal door iedereen perfect kunnen gebruikt worden die met het portaal wil werken.

5.4.2 Swagger

Swagger is een soort van documentatie voor de backend API. Met Swagger kunnen je de API gaan testen. Dit is vooral zeer handig voor mensen die in een project alleen aan de front-end werken. Het Front-End team kan met Swagger de API testen en weten zo wat er allemaal mogelijk is.

5.4.3 DocFx

Met DocFX kan ik ook voor een online handleiding zorgen. Zo kunnen ze Via /documentatie achter de url link van het portaal ook de online documentatie bekijken van DocFx.

5.5 Testing

5.5.1 Unit Tests

De code die ik zal schrijven, moet natuurlijk ook getest worden. Ik creëer hiervoor Unit-tests. Deze tests gaan de code uitvoeren en verwachten een bepaald resultaat van deze code. Is het resultaat niet wat de test verwacht, dan faalt de test.

Als iemand code in het project verandert, kunnen deze tests uitgevoerd worden. Als deze test dan faalt, weet ik dat er iets mis is en kan dit verbeterd worden.

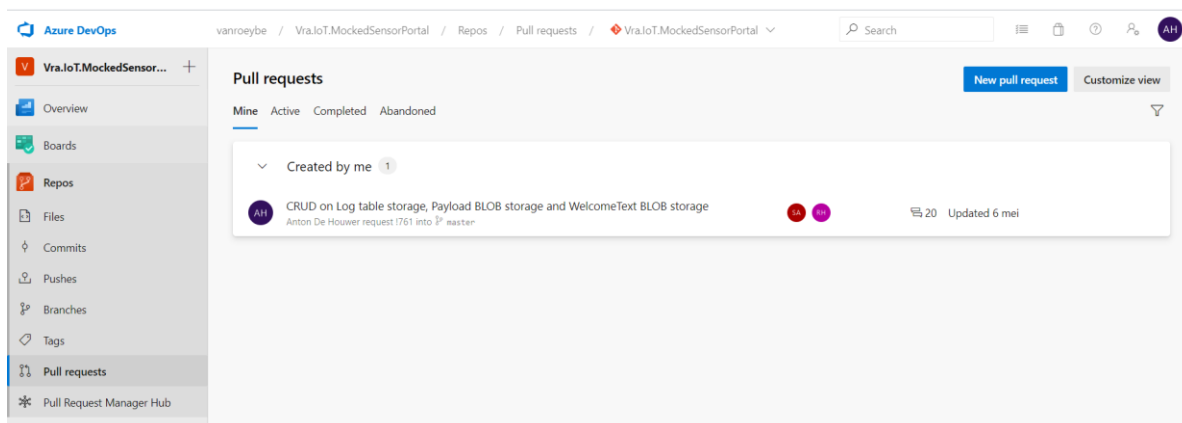
5.5.2 User Acceptance Testing (UAT)

Zodra mijn portaal klaar is ga ik dit eerst nog laten testen aan de gebruikers voor ik dit in productie zal stellen. Zo kunnen gebruikers ook nog laten weten wat ze nog graag zouden willen/niet willen hebben.

6 STAGEBEGELEIDER

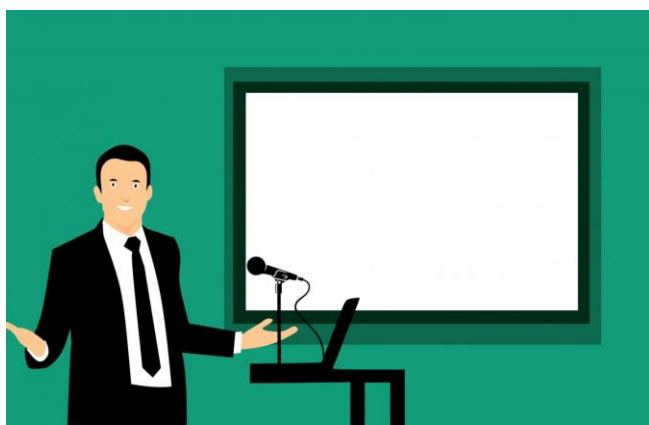
6.1.1 Pull requests

Tijdens het programmeren werk ik met Git. Git is een versiebeheersysteem dat het makkelijk maakt om in groep eenzelfde project te programmeren. Git werkt met takken (branches). Elke persoon werkt meestal op zijn eigen tak. Wanneer ik klaar ben met bijvoorbeeld een feature of functionaliteit te programmeren, wil ik deze code bij op de hoofdtak plaatsen. Daarvoor moet ik een pull request aanvragen. Hierbij kunnen alle teamleden (mijn stagementoren) de code die ik heb geschreven bekijken en becommentariëren. Als alle code goed is, wordt deze bij op de hoofdtak geplaatst.



6.1.2 Demo & live code review

Elke dinsdag en vrijdag namiddag heb ik een "live code review". Hierbij toon ik mijn code aan mijn stagebegeleiders. Ik leg uit wat de code doet en waarom ik het zo geschreven heb. Ik krijg hierbij ook direct feedback op. Als laatste is er ook nog het demo moment met iedereen van het team op vrijdag om de week af te sluiten. Tijdens dit moment toont ieder teamlid wat hij deze week geprogrammeerd/gerealiseerd heeft.



6.2 Begeleidende docent

Ik houd dagelijks een logboek (Word) bij. In dit document beschrijf ik kort wat ik die dag juist heb gedaan. Elke vrijdag van de week, stuur ik dit document door naar mijn stagebegeleider.