|  |  |
| --- | --- |
|  | |
| Project  Case 4 | |
|  | **Application Development**  Naam: Gilles Bosmans – r0899258 |
|  |
| Academiejaar 2023-2024  Campus : Geel |



Inhoudstafel

[Inhoudstafel 3](#_Toc153738504)

[Inleiding 4](#_Toc153738505)

[1 Ontwikkeling en oplossing 5](#_Toc153738506)

[1.1 Version control 5](#_Toc153738507)

[1.2 Structuur 5](#_Toc153738508)

[1.3 De applicatie 5](#_Toc153738509)

[1.3.1 Wat ik gemaakt heb 5](#_Toc153738510)

[1.3.2 Hoe ik dit gedaan heb 5](#_Toc153738511)

[1.3.2.1 Datadiagram 6](#_Toc153738512)

[1.3.2.2 Branche structuur 6](#_Toc153738513)

[1.3.2.3 Releases 6](#_Toc153738514)

[1.3.2.4 Reusable style resource 6](#_Toc153738515)

[1.3.2.5 API – endpoint 6](#_Toc153738516)

[1.4 Demo 7](#_Toc153738517)

[1.4.1 De administrator 7](#_Toc153738518)

[1.4.2 De gebruiker 9](#_Toc153738519)

[2 Demonstratie via YouTube 11](#_Toc153738520)

[3 Hoofdbronnen 12](#_Toc153738521)

[besluit…. 13](#_Toc153738522)

Inleiding

Voor de opdracht “case study” voor het opleidingsonderdeel DEVOPS & SECURITY, zal ik een applicatie ontwikkelen voor een specifieke casestudy. Mijn aandacht is gericht op Case 4, waarbij ik een WPF-applicatie (Windows Presentation Foundation) zal creëren met integratie van SQLite, Dapper als ORM (Object-Relational Mapper), en MaterialDesignThemes.

Het doel van deze casus is het bouwen van een Desktop GUI met een sterke focus op het gebruik van een SQLite database. Tijdens de ontwikkeling zal ik de principes van overerving implementeren, data schrijven met behulp van een Object-Relational Mapper (ORM), en de applicatie integreren in een GitHub Actions CI/CD pipeline. Deze pipeline maakt het mogelijk om de applicatie als een artifact te downloaden, inclusief de essentiële .exe-file.

# Ontwikkeling en oplossing

## Version control

Om gebruik te kunnen maken van version control gebruik ik GitHub.

Link: <https://github.com/GillesBosmans/DevOps-CaseStudy-Case4.git>

## Structuur

DevOps-CaseStudy-Case4

│

├── .github

│ └── workflows

│ └── deploy.yaml #Workflow voor deployen van de applicatie.

│

├── dal

│ ├── interfaces # Interfaces voor de repositories

│ └── repositories # Repositories voor het ophalen van data

│

├── models # Modelklassen

│

├── wpf # Design en code voor de user interactie

│

├── .gitignore # Gitignore configuratie

├── GB\_AutoParts.sln # Visual Studio Solution-bestand

└── README.md # Documentatie over het project

## De applicatie

### Wat ik gemaakt heb

Het idee dat ik had was het creëren/ ontwikkelen van een systeem waarbij ik de mogelijkheid heb om onderdelen afkomstig van sloopauto’s te catalogiseren en vervolgens deze onderdelen te verkopen. Door gebruik te maken van GitHub-actions wil ik een wpf.exe hebben voor het runnen van de applicatie.

### Hoe ik dit gedaan heb

Ik ben begonnen met het creëren van de structuur. Hierna heb ik de GitHub-action toegevoegd en gebruik gemaakt van branches in GitHub. Ik ben begonnen met het maken van een datadiagram. Hierna heb ik mijn models gemaakt en ben ik begonnen met het maken van mijn WPF schermen. Toen dit af was, heb ik interfaces gemaakt voor het creëren van een structuur voor mijn repositories die ik nodig heb voor het ophalen van data. Vervolgens heb ik mijn Sqlitebaserepository die ik nodig heb voor de configuratie van mijn database gecreëerd. Toen dit allemaal klaar was, kon ik beginnen met het schrijven van mijn repositories voor het daadwerkelijk ophalen van data, hierbij heb ik ook een ORM gebruikt (Dapper). Eenmaal de code voor alle repositories geschreven was, ben ik begonnen met het programmeren van de WPF-schermen voor de communicatie tussen de user en de database. Als laatste heb ik nog enkele bugs opgelost en heb ik de applicatie van mijn development branche naar mijn main gezet door middel van een pull request, dit had als respons dat de action is beginnen runnen en er een release beschikbaar is gekomen.

#### Datadiagram

Afbeelding met tekst, diagram, Parallel, Plan

Automatisch gegenereerde beschrijving

#### Branche structuur

Afbeelding met schermopname, diagram, lijn, Perceel

Automatisch gegenereerde beschrijving

#### Releases

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

#### Reusable style resource

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

#### API – endpoint

Ik heb ook een endpoint gemaakt met Node, Express en Mongoose voor een dubbel verificatie, dit gebeurd op het moment dat de user inlogt.

Afbeelding met schermopname, tekst, software, Multimediasoftware

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

## Demo

De applicatie beschikt over 2 (gebruikers) delen:

### De administrator

Standaard login 🡪 wordt gecreëerd bij het runnen van de applicatie

Username: [admin@gbautoparts.com](mailto:admin@gbautoparts.com)

Password: Admin

Afbeelding met tekst, schermopname, Lettertype, logo

Automatisch gegenereerde beschrijving

Eenmaal aangemeld:

Afbeelding met voertuig, wiel, Landvoertuig, band

Automatisch gegenereerde beschrijving

Nu kunnen we een voertuig aanmaken – kan nadien ook verwijderd worden als er geen relaties meer zijn (als er geen onderdelen meer zijn):

Afbeelding met tekst, schermopname, scherm, software

Automatisch gegenereerde beschrijving

Op dit moment heeft het voertuig nog geen onderdelen:

Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving

Onderdelen toevoegen aan het voertuig + eventueel nieuwe categorieën aanmaken – ook kunnen we het onderdeel bewerken of verwijderen:

Afbeelding met tekst, schermopname, scherm, software

Automatisch gegenereerde beschrijving

Na een klik op filter wordt het onderdeel opgehaald:

Afbeelding met tekst, schermopname, software, scherm

Automatisch gegenereerde beschrijving

### De gebruiker

Standaard login 🡪 wordt gecreëerd bij het runnen van de applicatie

Username: [kassa1@gbautoparts.com](mailto:kassa1@gbautoparts.com)

Password: Kassa1

Username: [kassa2@gbautoparts.com](mailto:kassa2@gbautoparts.com)

Password: Kassa2

Bij het maken van een typo wordt er een error getoond, in dit geval “Email is niet geldig”.

Afbeelding met tekst, schermopname, Lettertype, logo

Automatisch gegenereerde beschrijving

Eenmaal aangemeld:

Afbeelding met voertuig, wiel, Landvoertuig, band

Automatisch gegenereerde beschrijving

Het onderdeel dat we als Admin zojuist toegevoegd hebben, is nu zichtbaar:

Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving

Toevoegen aan bestelling (indien je meer stuks wilt toevoegen dan aanwezig zal er een melding komen “Er is maar ‘1’ in stock.”):

Afbeelding met tekst, software, Computerpictogram, Webpagina

Automatisch gegenereerde beschrijving

Bestelling geplaatst:

Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving

Bestelling verwerkt “Bestelling plaatsen gelukt” en product out of stock:

Afbeelding met tekst, schermopname, software, scherm

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving

# Demonstratie via YouTube

- Op welke YouTube link we een demo kunnen bekijken

# Hoofdbronnen

* Publishing/Deploying WPF Application ([video](https://www.youtube.com/watch?v=VIlDni8-iWM&t=465s))
* IAmTimCorey ([video](https://youtu.be/gSfMNjWNoX0?si=6pvFZUa3RhePpqvD))
* WPF Tutorial Reusable Style Resources ([website](https://wpf-tutorial.com/wpf-application/resources/))
* Datamanipulatie vanuit een programma (ZX0544)
* HTTPCLIENT ([website](https://www.stevejgordon.co.uk/sending-and-receiving-json-using-httpclient-with-system-net-http-json))

besluit….

Dit project heeft me niet alleen uitgedaagd om een applicatie op te zetten en te deployen, maar het heeft me ook getriggerd om verder te gaan dan de basis. Dit door gebruik te maken van interfaces en Reusable Style Resources (UIStyling) alsook meerdere projecten te gebruiken in één solution om een overzichtelijke structuur te creëren en het aanspreken van een endpoint door middel van een API die ik gemaakt heb met Node, Express en Mongoose.