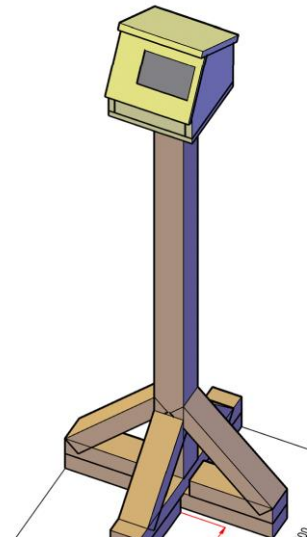


Onderzoek: Software – Project 56

Help de RDM-werkplaatsen om beter grip te krijgen op het gebruik van de werkplaatsen!



Projectleden:

- Matthijs Briel (0988991)
- Leandro de Nijs (1003440)
- Tom van Pelt (1003212)
- Gijs Kortlever (1003152)

Inleverdatum: 6 Februari 2022

Begeleiders: Erwin de Mos & S.M. Hekkelman

Bedrijfsbegeleider (Product Owner): F.W.J. Joosten Gladon

Gelegenheid: Eerste Kans

Inhoudsopgave

Wat is de beste programmeertaal voor de hardware?	3
Wat is de beste manier om data weer te geven?	3
Applicatie	3
Web applicatie	3
Conclusie	4
Hoe kunnen we het beste een webapplicatie maken?	5
Resultaten	6
Server	6
Webapplicatie	6
Conclusie en adviezen	7
Bronnen	8
Changelog	9

Wat is de beste programmeertaal voor de hardware?

- Python. Deze programmeertaal is erg groot en heeft heel veel mogelijkheden en kan direct draaien op een aantal microcontrollers waardoor deze microcontrollers beter aangestuurd kunnen worden. Omdat deze taal zo groot is, zijn er ook veel externe libraries voor beschikbaar die de connectie tussen de hardware en software makkelijker maken. Tevens is de taal ook aardig efficiënt en is het een erg toegankelijke programmeertaal.
- C++. Deze programmeertaal is ook erg groot en heeft ook veel mogelijkheden. Deze taal kan ook direct draaien op bijvoorbeeld een Raspberry Pi Pico. Ook zijn er veel externe libraries voor beschikbaar. De taal is efficiënter en stabielere dan Python maar wel moeilijker te programmeren.

Wat is de beste manier om data weer te geven?

Om dit te kunnen bepalen is er eerst gekeken naar een aantal opties. Hieruit zijn twee mogelijke opties ontstaan: een applicatie of een webapplicatie. Ieder van deze twee heeft zijn voor- en nadelen.

Applicatie

Voordelen:	Nadelen:
Minder veiligheidsrisico's	Moet geïnstalleerd worden
Sneller dan een website	Internet bereik noodzakelijk
Grotere functionaliteit aangezien ze toegang hebben tot de middelen van de computer waar die opstaat.	Moeilijk bij te houden/updaten
Geen hosting nodig	Alleen toegankelijk op een computer die de applicatie heeft geïnstalleerd
	Systeem gebonden (verschillende versies Windows / Mac)
	Vereist vrije ruimte

Web applicatie

Voordelen:	Nadelen:
Overall beschikbaar	Minder veilig
Makkelijk te laten zien tijdens een presentatie	Internet bereik noodzakelijk
Makkelijker toegankelijk voor meerdere personen	Hosting nodig
Beter up to date te houden	
Niet afhankelijk van systeem vereisten	
Veel documentatie over grafische weergave	
Cross-platform beschikbaar (telefoon, tablet, Mac, etc.)	

De eisen waar het data weergave platform in ieder geval aan moet voldoen zijn:

- Connectie met een database/API
- Grafische weergave van de data met grafieken
- Mogelijk maken om data te exporteren naar Excel
- Mogelijkheid voor veiligheid en toegang
- ~4 mensen moeten toegang hebben tot de data
- Overzichtelijk en snel

Conclusie

Uit het onderzoek blijkt dat het handigste is om gebruik te maken van een webapplicatie. Een webapplicatie is toegankelijker dan een applicatie die geïnstalleerd moet worden, omdat er makkelijk meer mensen op kunnen wat ook een vereiste is. Ook vormt een webapplicatie over het algemeen een betere ervaring voor de gebruikers. Een ander belangrijk punt is dat een webapplicatie cross-platform is. Cross-platform betekent dat het kan draaien op een telefoon, Windows Computer, Mac, etc. Dit is erg belangrijk omdat meerdere mensen op verschillende platformen erop moeten kunnen.

COMPARATIVE ANALYSIS

DESKTOP APP	VS	WEB APP
Not required	INTERNET CONNECTION	Required* <small>Some apps can function autonomously once loaded</small>
Manually downloaded and installed	INSTALLATION AND UPDATES	No installation. Automatically updates
Platform-dependent. Separate versions for each	PLATFORM DEPENDENCY	Cross-platform and portable
New licensed copy required for every system	DEPLOYMENT	You only need a browser and an account
Faster and has wider range of resource-heavy features	PERFORMANCE	Complex processes can be slower due to lag
Considered to be more secure	SECURITY	Considered to be less secure

Hoe kunnen we het beste een webapplicatie maken?

Om bovenstaande vraag te kunnen beantwoorden is er een interview gedaan met een expert op het gebied van programmeren: Jacques de Hooge. Jacques is een ervaringsdeskundige met bijna 50 jaar ervaring op het gebied van programmeren. Er zijn een aantal vragen aan hem gesteld en de samengevatte antwoorden zijn hieronder te lezen.

Allereest het maken van het front-end van een webapplicatie. De beste en makkelijkste manier om een front-end te maken is volgens Jacques door gebruik te maken van React. React is een library voor de programmeertaal Javascript, maar zou net zo goed een taal op zich kunnen zijn. Het is een library om gebruikersinterfaces te maken in Javascript. Er is heel veel documentatie over te vinden en volgens hem is het voor onze doeleinden de beste manier.

Samen met React raadt hij aan om gebruik te maken van Material-UI als template van de gebruikersinterface. Material-UI is een library die de mogelijkheid biedt om fundamentele en geavanceerde componenten snel, robuust en aanpasbaar te maken.

Als derde punt raadt Jacques aan om de server te programmeren in Python gebruikmakend van Flask of Django. Om de connectie te maken tussen de database en de webapplicatie is er een API nodig. "Een API (Application Programming Interface) is een software-interface die het mogelijk maakt dat twee applicaties met elkaar kunnen communiceren."¹ Jacques heeft ons aanbevolen om gebruik te maken van een RestAPI. "REST (Representational State Transfer) is een stijl van software architecture voor het ontwerp van netwerkapplicaties. Het fundamentele concept van een RESTful API is de 'resource'. Alle informatie die benoemd kan worden is een resource: een document, afbeelding, verzameling (collection) van resources, niet-virtueel object (bijvoorbeeld een persoon) etc. REST gebruikt een resource identifier om een resource te identificeren."²

Als laatste raadt Jacques aan om zo veel mogelijk berekeningen 'client side' uit te voeren. Door berekeningen client side uit te voeren wordt de server minder belast en blijft de webapplicatie snel mochten er meer mensen tegelijk op de webapplicatie zitten.

¹ <https://www.salesforce.com/nl/blog/2019/10/wat-is-een-api.html>

² <https://www.transip.nl/knowledgebase/artikel/2997-wat-is-een-rest-api/>

Resultaten

Server

De API is gemaakt in Python gebruik makende van de Flask library. Na enige problemen te hebben ervaren met de schoolserver is er overgestapt op een Linux VPS van Strato (z.d.). Deze oplossing biedt meer mogelijkheden omdat er hier directe toegang is tot zaken zoals de firewall.

Webapplicatie

Na een eerste versie van de webapplicatie in React te hebben gemaakt is er ontdekt dat dit niet ging werken. Geen van de projectleden had ervaring in React en Javascript of websites maken in het algemeen. Voordat er werd begonnen met het maken van een eerste versie in React wisten we al dat niemand ervaring had in React, maar er werd toch een poging gedaan. Met behulp van het video's en websites kwam er toch snel een eerste versie van de UI. Daarna kwam pas de echte uitdaging: functionaliteit toevoegen. Vooral de connectie maken met de API bleek erg lastig. Na veel uitzoek werk en proberen om problemen op te lossen leek het, het beste idee om opnieuw te beginnen.

De tweede (mogelijke) oplossing die geprobeerd is heet React-Admin. React-Admin is een framework gebaseerd op React en Material-UI maar specifiek gemaakt voor communicatie met API's. Klont veelbelovend dus. Maar na testen bleek dat toch niet geschikt was. React-Admin vereist namelijk een specifieke bouwstructuur van een API wat niet geschikt is voor de API die gebruikt wordt bij dit project. Volgende oplossing dan maar weer.

De derde (mogelijke) oplossing was Refine. Refine is ook een React Framework. Maar ook hiervoor zou de API volgens een specifieke bouwstructuur moeten zijn gemaakt. Inmiddels kwam de deadline steeds sneller inzicht en moest er toch een goede oplossing komen.

De vierde (mogelijke en veelbelovende) oplossing was Budibase. Budibase is een opensource webapplicatie builder om data apps te maken. Toen we deze oplossing wilden testen met de API werden er problemen gevonden met de schoolserver. De Firewall blokkeert namelijk verbindingen en aanvragen van buitenlandse IP-adressen blokkeert. Er is een mail gestuurd naar de serverbeheerder of hij het IP-adres van Budibase kon toelaten op de schoolserver. Deze mail is gestuurd in de week voor de kerstvakantie. Aan het eind van de kerstvakantie was er nog steeds geen antwoord en moest er een andere server geregeld worden. Dit was de server van Strato. Er kon dus getest worden met Budibase maar helaas voldeed Budibase niet aan een aantal andere eisen zoals data exporteren naar een Excel bestand. Volgende oplossing.

Inmiddels was het eind van de kerstvakantie en moest er toch echt een oplossing komen.

De vijfde (mogelijke) oplossing was Blazor samen met Radzen componenten (grafische weergave). Blazor is een framework waarin er geprogrammeerd wordt in C#. C# is een makkelijkere taal dan JavaScript en is dus sneller te begrijpen. Ook hierin is een eerste versie gemaakt maar ook in Blazor kan je geen data exporteren vanuit de API naar een Excel bestand.

De laatste (en gelukkig goede) oplossing is Microsoft Power BI. Power BI is bedoeld voor datavisualisatie en dus precies wat nodig is. Om een aantal onderdelen te gebruiken van Power BI die nodig zijn is de Pro licentie nodig, maar na overleg met onze Product Owner blijkt dat zij deze licentie heeft. De webapplicatie gaat na heel veel gedoe dus gemaakt worden in Microsoft Power BI.

(Bij vraag naar de bestanden voor de eerdere versies van de webapplicatie neem contact op met Leandro de Nijs, 1003440@hr.nl)

Conclusie en adviezen

Uiteindelijk, na veel zoekwerk en testen, is voor dit project de beste webapplicatie Microsoft Power BI. De Hogeschool Rotterdam maakt al intensief gebruik van de programma's van Microsoft en maakt volgens de Product Owner ook steeds meer gebruikt van Power BI. Na dit project kan de Hogeschool dus gemakkelijk de webapp overnemen om aan te passen en/of te onderhouden. Ook biedt Power BI nauwe samenwerking binnen andere Microsoftprogramma's zoals Excel en Teams.

Voor dit Project is de beste API de REST-API in Python gebruik makend van de Flask Library. Deze API zal data ophalen uit de SQLite3 database en de data sturen naar Power BI om grafisch weer te geven.

Bronnen

W3 Lab. (2021, 15 juli). *Why You Should Choose A Web Application Over A Desktop One*.

Geraadpleegd op 8 oktober 2021, van <https://w3-lab.com/choose-web-application-over-desktop/>

Digital Skynet. (2020, 3 juni). *Desktop App vs Web App: Comparative Analysis*. Geraadpleegd op 8

oktober 2021, van <https://digitalskynet.com/blog/Desktop-App-vs-Web-App-Comparative-Analysis>

Meta. (z.d.). *React – A JavaScript library for building user interfaces*. React. Geraadpleegd op 8

oktober 2021, van <https://reactjs.org/>

MUI. (z.d.). *MUI: The React component library you always wanted*. Geraadpleegd op 8 oktober 2021,

van <https://mui.com/>

TransIP. (z.d.). *Wat is een REST API? | TransIP*. Geraadpleegd op 8 oktober 2021, van

<https://www.transip.nl/knowledgebase/artikel/2997-wat-is-een-rest-api/>

VPS Linux bestellen vanaf € 5 | STRATO. (z.d.). STRATO AG. Geraadpleegd op 4 januari 2022, van

<https://www.strato.nl/server/vps-linux/>

React Admin. (z.d.). *Open-Source Framework for B2B Applications*. Geraadpleegd op 5 december

2021, van <https://marmelab.com/react-admin/>

refine | A React-based framework for building data-intensive applications in no time! | refine. (z.d.).

Refine. Geraadpleegd op 10 december 2021, van <https://refine.dev/>

budibase. (z.d.). <https://budibase.com/product>. Geraadpleegd op 13 december 2021, van

<https://budibase.com/product>

Radzen benefits. (z.d.). Radzen.com. Geraadpleegd op 26 december 2021, van

<https://www.radzen.com/features/>

Changelog

Versie	Datum	Aanpassing	Auteur
1.0	18/09/2021	Eerste versie	Leandro de Nijs
1.1	21/09/2021	Deelvraag deels beantwoord	Tom van Pelt
1.2	26/09/2021	Deelvraag verder beantwoord	Tom van Pelt
1.3	08/10/2021	Deelvragen verder beantwoord, tabellen en afbeeldingen toegevoegd, conclusie getrokken en bronnen toegevoegd	Leandro de Nijs
1.4	19/11/2021	Bronnen in APA-stijl gezet en lay-out opgeschoond	Tom van Pelt
1.5	10/01/2022	Grote toevoeging	Leandro de Nijs
1.6	11/01/2022	Bronnen toegevoegd	Leandro de Nijs
1.7	15/01/2022	Afbeeldingen op Voorblad toegevoegd	Tom van Pelt
1.8	30/01/2022	Kleine aanpassing	Leandro de Nijs
1.9	31/01/2022	Kleine aanpassing	Tom van Pelt