



Faculteit Bedrijf en Organisatie

Ontwikkeling van de Proof of Concept voor een nieuw IT asset management programma voor AZ Glorieux
gebouwd met Power Apps

Thomas Van den Hauwe

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Benjamin Vertonghen
Co-promotor:
Siemen Vanneste

Instelling: AZ Glorieux

Academiejaar: 2019-2020

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Ontwikkeling van de Proof of Concept voor een nieuw IT asset management programma voor AZ Glorieux
gebouwd met Power Apps

Thomas Van den Hauwe

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Benjamin Vertonghen
Co-promotor:
Siemen Vanneste

Instelling: AZ Glorieux

Academiejaar: 2019-2020

Tweede examenperiode

Woord vooraf

De bachelorproef die voor u ligt was het sluitstuk van de opleiding Toegepaste Informatica, afstudeerrichting Netwerken/Systeembeheer.

IT betekent voor mij een constante verbetering en stroomlijning van hoe we dingen doen. In teken hiervan heeft Low-code veel potentieel en ik ben blij om mijn proef in dit domein te hebben kunnen maken.

Ik wil mijn co-promoter, Siemen Vanneste bedanken zowel voor de kans om stage te mogen lopen bij AZ Glorieux als voor het voorstellen van een tof bachelor proef onderwerp.

Ook wil ik mijn promotor, Benjamin Vertonghen bedanken voor de constructieve feedback doorheen het onderzoek.

Mijn ouders wil ik bedanken voor hun steun en geduld.

Veel leesplezier.

Samenvatting

Het ziekenhuis AZ Glorieux zou graag PowerApps willen gebruiken om een oude business applicatie (LanReview, ITAM) te vervangen. Specifiek wordt bekeken hoe geschikt PowerApps is voor deze rol.

Dit onderzoek is relevant omdat het algemene zakelijke landschap gekenmerkt is door een ontoereikendheid aan de vraag om software en het is interessant om te zien of low-code zijn beloofde snellere softwareontwikkeling kan waarmaken.

Eerst wordt de omgeving van AZ Glorieux verkend en wordt LanReview in detail besproken. Het begrip low-code wordt onderzocht, in het bijzonder wordt er een actueel beeld van geschetst. Het PowerApps platform wordt onder de loep genomen.

Er wordt een requirementsanalyse gemaakt. De low-code markt wordt onderzocht en met de requirements in het achterhoofd wordt er een low-code (voor powerusers) platform gekozen voor de primaire proof-of-concept en hierna een low-code (voor ontwikkelaars) platform voor de secundaire proof-of-concept. Dit zijn PowerApps en Outsystems geworden.

In de volgende delen worden de proof-of-concepts voor beiden uitgewerkt. Dit wordt gedaan door elke requirement te implementeren en te documenteren.

Tenslotte komt het besluit dat PowerApps niet de beste optie is voor deze case en wordt genuanceerd voor wat het wel en niet geschikt zou zijn. Dit wordt gedaan voor zowel PowerApps (no-code) als Outsystems (low-code).

Inhoudsopgave

1	Inleiding	15
1.1	Probleemstelling	16
1.2	Onderzoeksvraag	16
1.2.1	Hoofdonderzoeksvragen	16
1.2.2	Deelonderzoeksvragen	16
1.3	Onderzoeksdoelstelling	17
1.4	Opzet van deze bachelorproef	17
2	Stand van zaken	19
2.1	AZ Glorieux omgeving	19
2.1.1	Omgeving	19
2.1.2	Toekomst	19
2.1.3	IT Asset Management	20

2.1.4	Databronnen	21
2.2	LanReview	21
2.2.1	Visual Basic	21
2.2.2	LanReview werking	23
2.3	Low Code	23
2.3.1	Basics	23
2.3.2	Geschiedenis	27
2.3.3	Nood / Voordelen	28
2.3.4	Kritiek / Nadelen	29
2.3.5	Markt en Evolutie	30
2.3.6	Voorgaand onderzoek	32
2.4	Microsoft Power Platform: Power Apps	33
2.4.1	Wat?	33
2.4.2	Drie soorten apps	34
2.4.3	Belangrijke Onderdelen	34
2.4.4	IDE overzicht	36
2.4.5	Recente Wijzigingen	36
2.4.6	Pricing, Licencing	37
2.4.7	Uitbreidingsmogelijkheden	38
2.5	Microsoft Power Platform: Power Automate	38
3	Methodologie	39
3.1	Requirementsanalyse	39
3.1.1	Functioneel VS Niet-functioneel	39
3.1.2	MoSCoW-methode	40

3.1.3	Long List	41
3.1.4	Short List	42
3.1.5	Besluit	48
4	POC: Power Apps	49
4.1	Voorbereiding	49
4.1.1	Data en data weergave	50
4.2	SharePoint Configuratie	50
4.3	Model van Opstelling	51
4.4	Requirements	51
4.4.1	Must have	51
4.4.2	Should have	61
4.4.3	Nice to have	64
4.5	Custom Connector	66
4.5.1	Microsoft Graph	67
4.5.2	Azure API App (ASP.NET)	68
4.5.3	Blank Custom Connector	71
5	POC: Outsystems	75
5.1	Voorbereiding	75
5.1.1	Van 0 tot de eerste app uitvoering	75
5.1.2	IDE en begrippen	75
5.2	Model van Opstelling	77
5.3	Requirements	77
5.3.1	Must have	77

5.3.2 Should have	87
5.3.3 Nice to have	91
6 Conclusie	93
A Onderzoeksvoorstel	95
A.1 Introductie	95
A.2 State-of-the-art	97
A.3 Methodologie	98
A.4 Verwachte resultaten	99
A.5 Verwachte conclusies	100
Bibliografie	101

Lijst van figuren

2.1	Voorbeeldweergave van de Visual Basic IDE (Speed, 2020)	22
2.2	LanReview algemeen/filterbaar overzicht	23
2.3	LanReview detailoverzicht	24
2.4	Citizen Development statistieken (McKendrick, 2017)	26
2.5	Vergelijking van doorlooptijd (Richardson & Rymer, 2016)	28
2.7	Forrester voorspelde markt evolutie (J. Rymer, 2018)	32
2.8	Overzicht Microsoft Power Platform (Microsoft Docs, 2019d)	33
2.9	Overzicht van de formulebalk (Microsoft Docs, 2019a)	35
2.10	Overzicht van Power Apps Studio (Microsoft Docs, 2017)	37
4.1	Model van de PowerApps proof-of-concept.	51
4.3	De Automate flow waarmee SharePoint gefilterd wordt.	56
4.4	Voorbeeld van SharePoint filtering in de PowerApp.	57
4.5	Overzicht van de beperktere mobiele PowerApps POC	59
4.6	Instelling voor aantal op te halen rijen	60
4.7	Recurrente flow met SharePoint databron.	61
4.8	Recurrente flow met Excel databron.	62
4.9	Een voorbeeld van de mail.	63

4.10 Een voorbeeld van vastpinnen aan de toolbar.	65
4.11 Dynamisch aanpassen van layout aan filter condities.	65
4.12 Microsoft Graph in het Microsoft 365 platform (Microsoft Docs, 2020d) 67	
4.13 Microsoft Graph in de Connected Services	68
4.14 Teruggegeven error bij het maken van requests.	72
4.15 Resultaat van de Graph query's.	73
4.16 Gebruik van de Custom Connector in een flow, de acties van deze connector zijn groen.	74
5.1 Overzicht van de Outsystems Service Studio.	76
5.2 Model van de Outsystems proof-of-concept.	77
5.3 Hoofdscherm Outsystems POC	79
5.4 Detailscherm Outsystems POC	80
5.5 Uitbreiding van de standaard formuliervalidatie.	80
5.6 Overzicht van de filtering voorbereidingen.	82
5.7 De Filter actie.	83
5.8 De SQL query.	84
5.9 De client en server actie om een rapport te downloaden.	84
5.11 Voorbeeld van de POC met een mobiele layout.	87
5.12 Overzicht van de configuratie en het resultaat van de Mail Timer.	89
5.13 Weergave van het autorisatieproces. (MS Login Connector Team, 2019)	90

Lijst van tabellen

2.1 Vergelijking van het traditionele en low-code ontwikkelingsproces (Revell, 2020)	26
2.2 Overzicht van de marktleiders	31
3.3 Overzicht van beschikbare plannen (Pohl, 2019)	43
3.5 Prijsmodel Outsystems (Outsystems, g.d.)	45
3.7 Prijsmodel Mendix (Mendix, g.d.)	46
3.8 Prijsmodel Salesforce (Salesforce, g.d.-a)	47
4.1 Meerprijs bovenop huidige licentie	52
4.2 Overzicht navigaties tussen schermen	52

1. Inleiding

Het IT Team van AZ Glorieux heeft een oog op de toekomst. Grote voorbeelden hiervan zijn een push om nodige toestellen te migreren naar Windows 10, een geleidelijke adoptie van Intune en de vervanging van een handvol oude business applicaties. Dit onderzoek staat in het teken van deze vernieuwingsslag, specifiek zal er getracht worden een bijdrage te leveren in de context van het laatste voorbeeld.

LanReview, een binnenshuis ontwikkeld assetmanagement programma dat onontbeerlijk is voor de IT Helpdesk, is aan vernieuwing toe. De laatste ontwikkeling is van enige tijd geleden en het is niet logisch meer om dit terug op te nemen voornamelijk wegens de verouderde codebase (Visual BASIC 3.0). De vervangende applicatie moet bij voorkeur gemaakt worden met Microsoft Power Apps en ondersteunend ook Power Automate. De vraag wordt met andere woorden gesteld wat voor potentieel dit platform heeft en hoe het past in hun omgeving.

Dit is niets speciaals. Bedrijven kampen al jaren met een niet in te vullen vraag naar software en dit zal in de toekomst alleen maar blijven toenemen. Low-code platformen (waaronder het Microsoft Power platform) stellen zichzelf als het beste gereedschap om dit probleem aan te pakken.

Dat is wat onderzocht zal worden. Specifiek wordt gekeken hoe gepast het PowerApps gereedschap is om het LanReview probleem op te lossen. Dit wordt gedaan aan de hand van een proof-of-concept.

Het is belangrijk om te weten dat het onderzoek zicht beperkt tot low-code. Er zal bijvoorbeeld geen proof-of-concept op de traditionele manier geprogrammeerd worden.

Kennis van onderstaande technologieën is een meerwaarde maar niet verplicht om de tekst te kunnen begrijpen:

- SQL
- REST API's
- C# en het ASP.NET Framework.

1.1 Probleemstelling

De probleemstelling in zijn simpelste vorm kan herleid worden naar:

'Kan PowerApps gebruikt worden om LanReview te vervangen?'

De meerwaarde die het onderzoek (het antwoord op deze vraag) biedt aan AZ Glorieux:
De proof-of-concept kan een startpunt zijn voor de definitieve oplossing. Op z'n minst zal duidelijk worden hoe PowerApps in te zetten is en voor welke scenario's. Dit kan met andere woorden helpen bij de vernieuwing van andere applicaties en het kan de drempel verlagen voor het praktisch gebruik van PowerApps in scenario's waar men er anders niet aan gedacht zou hebben.

1.2 Onderzoeksraag

1.2.1 Hoofdonderzoeksraag

- Is het mogelijk een vervanger voor LanReview te bouwen met Power Apps die elk de vier hoofdfunctionaliteiten ondersteunt en op zijn minst drie vierde van de overige functionaliteiten kan ondersteunen?
- Is Power Apps werkelijk de beste keuze hiervoor of is er meerwaarde in een volledig gerealiseerd IT asset management pakket? Is er alternatief een beter geschikt low-code platform?

1.2.2 Deelonderzoeksraag

- Is de Proof of Concept eenvoudig uit te breiden met nieuwe functionaliteiten? Is dit aanvaardbaar voor het IT team van AZ Glorieux?
- Is Power Apps robuust genoeg om meer complexe functionaliteit te ondersteunen? Is het met een zelfgeschreven uitbreiding bijvoorbeeld mogelijk om de remote desktop functionaliteit te verwezenlijken?
- Is de gebruikte methode op zijn minst deels bruikbaar om andere applicaties voor de IT van AZ Glorieux te bouwen? Er worden twee cases onderzocht: één voor een telefoonboek (legacy business applicatie met een lage moeilijkheidsgraad) en één om het potentieel van Power Apps te demonstreren. Dit is opgenomen in de volgende onderzoeksraag.
- Is er een use case voor nieuwe of experimentele functionaliteiten in Power Apps zoals integratie met Teams of beperkt toepassen van AI via de AI Builder?
- De Proof of Concept zal nauw samenwerken met SCCM, is het concreet mogelijk om in de nabije toekomst ook samen te werken met Intune?

1.3 Onderzoeksdoelstelling

De hoofddoelstelling bestaat uit het opstellen van een competente proof-of-concept. Hoe competent deze is wordt bepaald door de voldoening aan de twee hoofdonderzoeksvragen. Hiernaast moet verdere ontwikkeling zo toegankelijk mogelijk gemaakt worden. Met andere woorden de proef en bijbehorende documenten moeten een hoge mate van praktisch nut hebben voor het IT Team om niet-gerealiseerde functionaliteit in de proof-of-concept en toekomstige functionaliteiten toe te kunnen voegen.

1.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomain, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 4 wordt de primaire proof-of-concept, gebouwd met PowerApps, overlopen.

Hoofdstuk 5 behandelt de secundaire proof-of-concept, gemaakt met Outsystems.

In Hoofdstuk 6, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2. Stand van zaken

2.1 AZ Glorieux omgeving

2.1.1 Omgeving

AZ Glorieux is een groot regionaal ziekenhuis met een domein waarin er meer dan 1000 toestellen zitten waarover controle en overzicht nodig is. In deze soort omgeving is er geen marge voor fouten en moet respons direct kunnen zijn. Zowel Servers als client computers gebruiken Microsoft Windows besturingssysteem varianten. De softwarecatalogus bestaat uit een ruime selectie van zowel medische als ondersteunende programma's. Relevant is dat Office gebruikt wordt, zowel de gewone versie (offline) als Office 365. Er ligt een grote nadruk op documentering. Hiervoor wordt SharePoint gebruikt.

2.1.2 Toekomst

Migratie naar Windows 10 is een lopend proces. Veel bedrijven doen dit en proberen dit te versnellen wegens de komende end-of-life status van Windows 7 (Microsoft Support, 2020). Per case kan dit tijdrovend zijn om softwarecompatibiliteit te garanderen.

Verspreiding van nieuwe technologieën: interesse in het Power Platform, implementatie van Microsoft Teams, Intune wordt steeds meer gebruikt als aanvulling van SCCM (System Center Configuration Manager, zie Subsectie 2.1.3). Er worden steeds meer mobiele devices opgenomen in het domein.

Wat digitale transformatie betreft wordt sommige verouderde, huiseigen software vervangen. Concreet gaat het om tools gemaakt met Visual Basic. Er wordt gestreefd naar een

hogere mate van automatisatie. Het is wenselijk dat zoveel mogelijk software gedeployed kan worden vanuit SCCM.

2.1.3 IT Asset Management

Dit onderzoek valt in het domein van IT Asset Management. Hieronder volgen enkele definities om dit begrip te verduidelijken.

IAITAM (g.d.) geeft een algemene definitie:

IT Asset Management is a set of business practices that incorporates IT assets across the business units within the organization. It joins the financial, inventory, contractual and risk management responsibilities to manage the overall life cycle of these assets including tactical and strategic decision making.

Gartner (g.d.) stelt:

IT asset management (ITAM) provides an accurate account of technology asset lifecycle costs and risks to maximize the business value of technology strategy, architecture, funding, contractual and sourcing decisions.

Ivanti (2018) reikt onderverdelingen aan. Er kan een onderscheid gemaakt worden tussen Fysieke, Digitale, Software, Mobile en Cloud IT Asset Management. Een beschrijving voor de voor het onderzoek toepasbare categorie, de fysieke, geldt:

The discovery and inventory of hardware including PCs, laptops, printers, copiers, and any other device used for IT and data management purposes.

Er werd verwacht dat de IoT (Internet of Things) trend IT Asset Management programma's zal laten evolueren om met deze devices om te kunnen gaan. Dit betekent meer assets en meer asset types. Hier zijn voordelen mee verbonden: efficiënter resource gebruik, verhoging van operationele efficiëntie en productiviteit. Er is betere controle mogelijk op veiligheid en compliance. Onderhoud en herstel kunnen geautomatiseerd worden. (Badnakhe, 2020).

IT asset management oplossingen zijn vaak een aanvulling op het werken van SCCM. Een belangrijke functie van SCCM is het verschaffen van informatie. In sommige gevallen wordt functionaliteit van SCCM opgeroepen of zijn database geraadpleegd. Door de belangrijke rol van SCCM loont het om in groter detail te gaan.

Droogenbroot en Scheiris (2016) beschrijft het als volgt:

System Center Configuration Manager, afgekort als SCCM of ConfigMgr, is een onderdeel van de Microsoft System Center Suite en staat in voor het

beheer en inventarisatie van pc's en servers. Daarenboven is het mogelijk toepassingen uit te rollen, software updates te installeren en compliance te controleren en op te lossen.

De primaire functies zijn het creëren van computer images, massa deployment van images, distribueren van silent installaties van software applicaties, cross campus managen van software en natuurlijk inventarisatie en organisatie van eigen hardware en devices.

SCCM functioneert aan de hand van een aantal sleuteltechnologieën: PXE booting, de SCCM agent geïnstalleerd op client pc's die data voedt aan SCCM en Task Sequences waarmee een hele reeks taken uitgevoerd kunnen worden. (Spitze, 2019).

Eerder werd gesproken over de relatie tussen ITAM oplossingen en SCCM. Waar traditionele oplossingen een aanvulling of uitbreiding zijn op SCCM is LanReview eerder een vereenvoudiging. Het is toegespitst op inventarisatie en organisatie van eigen hardware en devices.

De reden dat SCCM zo verspreid is is omdat het gratis inbegrepen is in enterprise licensing van Microsoft. Het is bijvoorbeeld inbegrepen in Microsoft 365 licentie. (Microsoft Docs, 2020b)

2.1.4 Databronnen

Een overzicht van enkele databronnen die AZ Glorieux gebruikt. Deze zullen later in het onderzoek aan bod komen.

- SQL Server
- Microsoft Excel en Access
- SharePoint

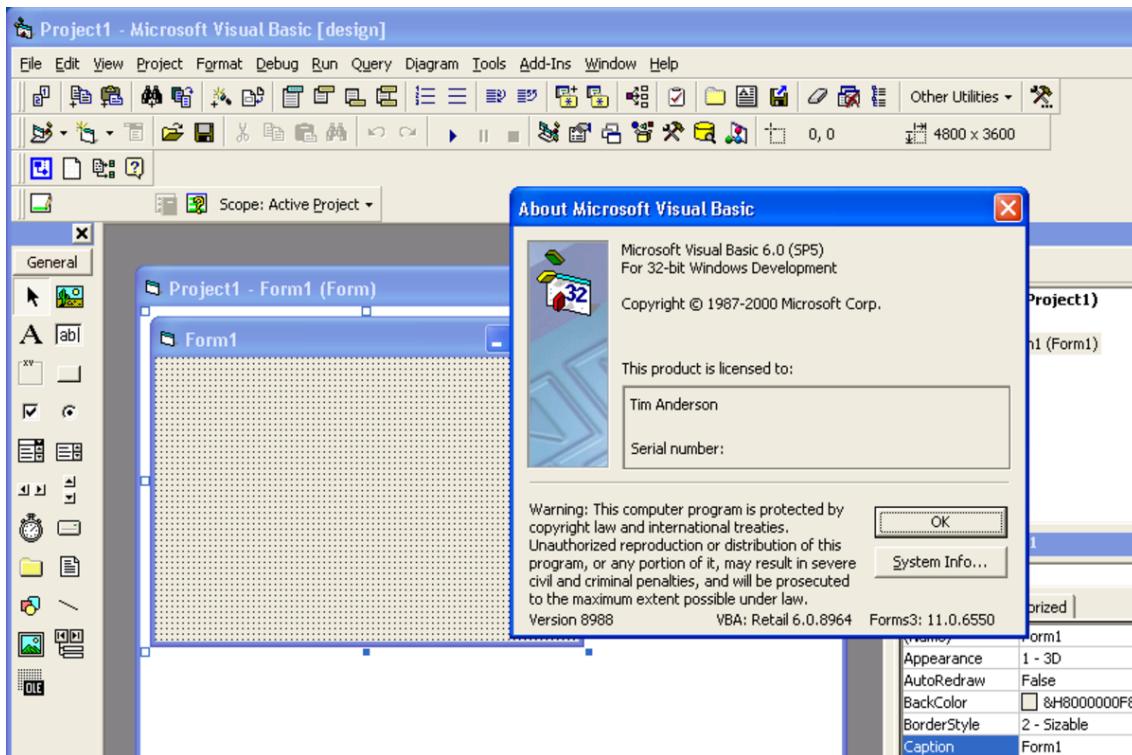
2.2 LanReview

2.2.1 Visual Basic

LanReview werd gebouwd in Visual Basic (VB). Dit is een event-gedreven programmeertaal en ontwikkelomgeving van Microsoft waarmee men code kan aanpassen door drag-en-drop van objecten en door het gedrag en uiterlijk van deze objecten te definiëren. VB komt van de BASIC programmeertaal. Het is een RAD (Rapid Application development) platform. Het werd voornamelijk gebruikt voor prototyping en als front-end voor databases. De laatste versie, Visual Basic 6.0, stamt van 1999.

Het grote voordeel is snelheid van ontwikkeling. Er zijn ook een aantal nadelen. Het had veel geheugen nodig en was niet geschikt voor programma's die veel proceskracht nodig hadden, zoals games. Het is ook beperkt tot het Microsoft besturingssysteem.

(Rouse, 2019)



Figuur 2.1: Voorbeeldweergave van de Visual Basic IDE (Speed, 2020)

Geschiedenis

- **1964:** BASIC geformuleerd door John Kemeny en Thomas Kurtz.
- **1987:** Architect/programmeur Alan Cooper bedenkt voorloper van VB genaamd Tripod.
- **1988:** Bill Gates koopt de rechten voor Tripod.
- **1991:** Visual Basic 1.0 geïntroduceerd.
- **1998:** Visual Basic 6.0 geïntroduceerd.
- **2002:** .NET framework.
- **2008:** Einde extended support voor Visual Basic 6.0.

(Grigoris, 2014)

Vandaag

Uitvoering van VB applicaties blijft ondersteund op moderne Windows versies. Het maken van nieuwe echter niet (Microsoft Docs, 2018). Niettegenstaande heeft men manieren gevonden om deze alsnog te installeren. (Brust, 2015)

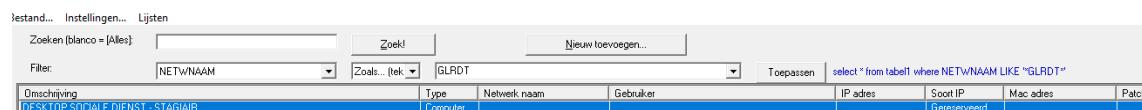
Er is nog een kleine maar vocale groep aanhangers van VB. Dit komt omdat ze de officiële opvolger, Visual Basic .NET, niet waardig vonden voornamelijk omdat de afhankelijkheid van .NET een extra abstractielaaq was. Enkele voorbeelden hiervan zijn:

- Visual Basic behoud blijft hoog. Opvallend aanhoudend succes voor een ‘dode’ tool.

(ISpliter, 2014).

- Er werd een petitie gemaakt voor verdere ontwikkeling van VB. (classicvb petition, 2005)
- Er worden nog steeds applicaties mee ontwikkeld (Ippolito, 2018).

2.2.2 LanReview werking



The screenshot shows the LanReview application's main interface. At the top, there are buttons for 'Bestand...', 'Instellingen...', and 'Lijsten'. Below these are search fields: 'Zoeken (blanco = [Alles])' and 'Zoek!', and a 'Nieuw toevoegen...' button. A 'Filter:' dropdown is set to 'NETWNAAM' with a dropdown arrow, and a 'Zoals... (tek.' dropdown is set to 'GLRDT'. To the right of these are 'Toepassen' and 'select * from tabel1 where NETWNAAM LIKE "%GLRDT%"' buttons. The main area is a table with columns: 'Omschrijving', 'Type', 'Netwerk naam', 'Gebruiker', 'IP adres', 'Soort IP', 'Mac adres', and 'Patch'. One row is visible, showing 'DESKTOP SOCIALE DIENT - STAGIAIR' in the 'Omschrijving' column and 'Computer' in the 'Type' column.

Figuur 2.2: LanReview algemeen/filterbaar overzicht

LanReview is een belangrijke tool van de Helpdesk en wordt dagelijks gebruikt. Het houdt zijn informatie over assets in het domein bij in een Access databank. Deze informatie werd origineel geëxporteerd uit SCCM. De applicatie heeft twee views: de primaire view is een filterbaar overzicht van devices. Voor elke entry kan een meer gedetailleerde view opgeroepen worden waar alle datavelden te zien zijn. Aan de hand van de filter functionaliteit worden SQL-achtige query's gebouwd die dan uitgevoerd worden op de Acces databank.

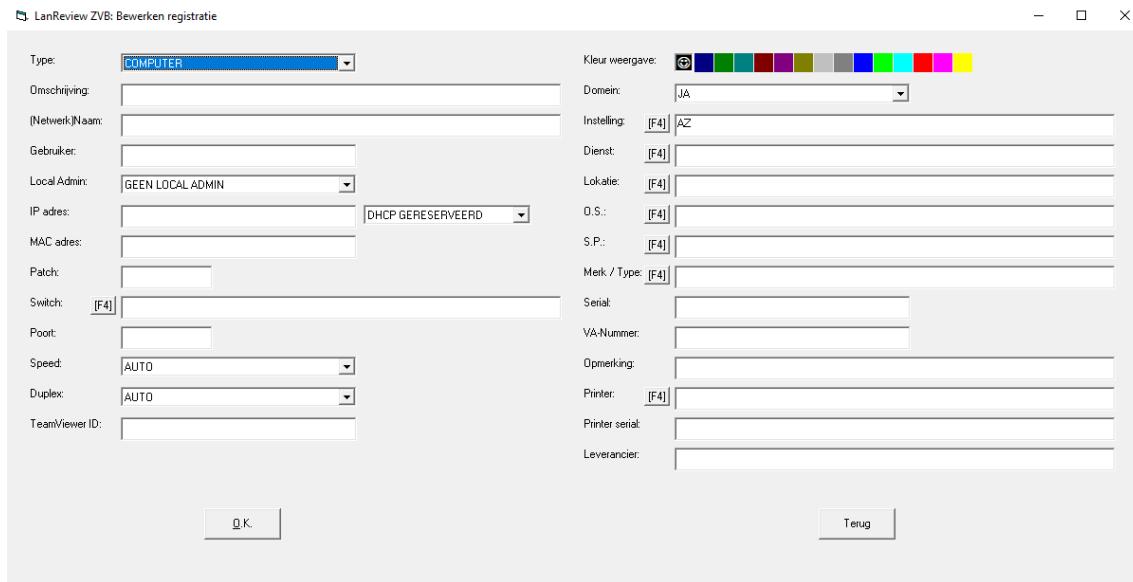
Vaak voorkomende use cases:

- Het overnemen van iemand zijn PC die de helpdesk contacteert met een vraag om assistentie. Het is een startpunt voor remote management.
- Het genereren van rapporten. Er kunnen voorgebouwde query's uitgevoerd worden op de data waarvan de resultaten onder andere naar Excel geëxporteerd kunnen worden.
- Filteren van de dataset. Bijvoorbeeld een overzicht van alle laptops van een bepaald model tonen die nog steeds met Windows 7 werken.
- End-of-life beheer van toestellen. Via kleurmarkeringen is duidelijk of een device nog niet gebruikt wordt, in gebruik is of niet meer gebruikt wordt/in storage is.

2.3 Low Code

2.3.1 Basics

De POC zal gebouwd worden met een low-code platform. Cloud services maken hier een deel van uit. Moderne low-code platformen zijn gecategoriseerd als PaaS (Platform as a Service). In geval van IaaS (infrastructure as a service) leeft de volledige infrastructuur in de cloud. PaaS kan gezien worden als een extra laag gebouwd bovenop IaaS waarmee applicatieontwikkelaars software kunnen programmeren in de cloud om later aan te kunnen bieden als SaaS (Software as a Service). Het is een tussenlaag gefocust op programmeurs. (Nucleus, 2017)



Figuur 2.3: LanReview detailoverzicht

Definiëren van wat low-code zoal inhoudt kan best begonnen worden met een formele definitie:

It is an application platform that supports rapid application development, one-step deployment, execution and management using declarative, high-level programming abstractions, such as model-driven and metadata-based programming languages. They support the development of user interfaces, business logic and data services, and improve productivity at the expense of portability across vendors, as compared with conventional application platforms.

(Vincent, Iijima, Driver & Natis, 2019)

De bedoeling van low-code is om applicaties sneller te kunnen maken en ontwikkeling voor een grotere groep toegankelijk te maken. Typisch gezien wordt er een WYSIWYG (What You See Is What You Get) interface gebruikt waarin visuele componenten geconfigureerd worden via drag-and-drop. Vaak is er ondersteuning om aangepaste code toe te voegen indien het platform bepaalde functionaliteit out of the box niet ondersteunt. (Kissflow, 2018a)

Low-code is een software ontwikkelingsfilosofie die staat voor snellere aflevering van applicaties tegen zo weinig mogelijk manueel coderen. Praktisch gezien worden applicaties opgezet en geconfigureerd in een grafische user interface aan de hand van visuele modellering. Ontwikkelaars sparen tijd uit omdat ze geen werk meer moeten doen aan infrastructuur of herimplementatie van patronen. Ze hoeven zich slechts te focussen op de unieke 10% van een applicatie. (Revell, 2020)

Kissflow legt in zijn omschrijving nadruk op een grotere toegankelijkheid en Outsystems focust meer op het kunnen vermijden van herhalend werk.

Een low-code platform bestaat typisch uit volgende delen:

- Een visuele IDE (Integrated Development Environment).
- Connectoren naar back-ends of services.
- App Lifecycle Management. Hiermee bedoeld geautomatiseerde tools voor build, debug, en deploy. Ook beheer van de app tijdens test, staging en productie.

(Revell, 2020)

Low-code VS No-code

Een no-code platform is een gespecialiseerde versie van een low-code platform waar men manuele aanpassingen zoveel mogelijk wil vermijden, al dan niet door het pre-builden van de nodige visuele componenten.

Bloomberg (2017) nuanceert de verschillen door te focussen op de eindgebruiker:

- Low-code: De volgende generatie van applicatie ontwikkeling dat het werk van professionele ontwikkelaars versnelt en stroomlijnt.
- No-code: Self-service applicatie assembly voor business users die citizen developers worden (Zie Subsectie 2.3.1).

Low-code development VS traditional software development

Citizen developers

De survey uitgevoerd door McKendrick (2017) geeft de volgende definitie:

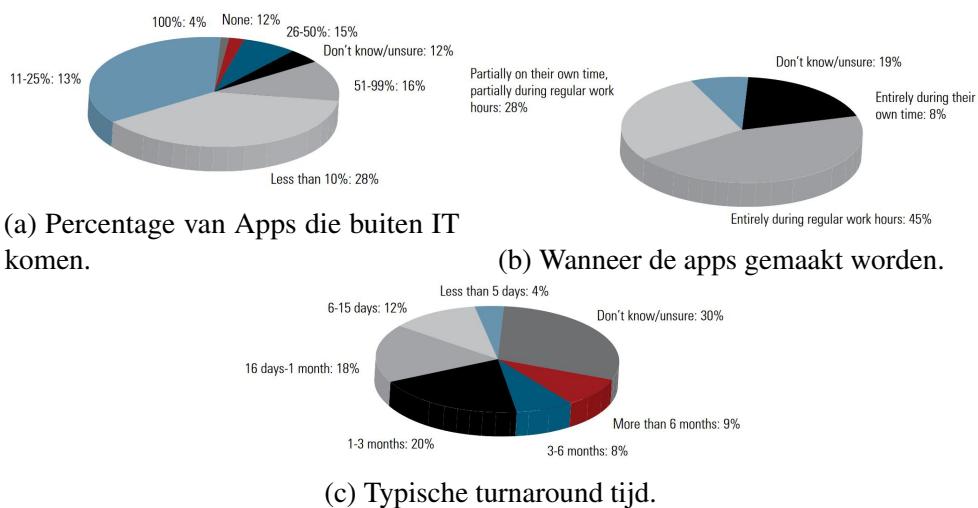
We define citizen developers as business users, not part of IT departments or contracted IT services, who build and use their own scripts, programs, algorithms, or interfaces designed to perform business functions or support business processes.

Er zijn enkele interessante statistieken uit voortgekomen:

- Snelheid van applicatie aflevering en delen van data/analytics worden aanzien als zwakke gebieden van IT support.
- 76% van de respondenten zegt dat op zijn minst een deel van de gebruikte applicaties buiten het IT departement komen.
- 45% van citizen development activiteiten worden tijdens de werkuren gedaan, de grootste motivatie hiervoor (42%) is omdat het IT departement te traag is.
- In slechts 17% van de gevallen duurt ontwikkeling van een citizen developer applicatie langer dan drie maand.

	Traditional	Low-Code
1	requirements bepalen	requirements bepalen
2	architectuur plannen	derde partij API's selecteren
3	selectie van back-end framework, bibliotheken, data stores en API's	app workflow, data models en user interface tekenen in de visuele IDE
4	selectie van front-end framework	API's verbinden
5	keuze van deployment stack, opzetten van CI (Continuous Integration) en operatieplan creëren	indien nodig handgeschreven code toevoegen aan front-end of aanpassing van automatisch gegenereerde SQL query's
6	wireframes en prototypes maken	gebruikersacceptatietests
7	ui coderen in gekozen Javascript framework	deployment naar productie, hierna updates eenvoudig doorduwen
8	falende tests schrijven	
9	modellen definiëren en koppelen aan data stores	
10	business logic definiëren en coderen	
11	views maken om JSON data te voorzien/ontvangen van en naar de front-end	
12	implementeren van workflows	
13	integreren van derde partij API's via gepubliceerde interface of support bibliotheek	
14	herhalen tot tests slagen	
15	testen voor security, performance, kwaliteit en gebruikersacceptatie	
16	deploy, patch, monitor en update tot levenseinde van de app	

Tabel 2.1: Vergelijking van het traditionele en low-code ontwikkelingsproces (Revell, 2020)



Figuur 2.4: Citizen Development statistieken (McKendrick, 2017)

Mythes

Er bestaan enkele mythes rondom low-code:

- Minder geschikt voor professionele gebruikers dan voor citizen developers.
→ In de praktijk wordt low-code vaker gebruikt door professionele ontwikkelaars.

- De nood om te programmeren wordt geëlimineerd.
→ Het is vaak mogelijk om eenvoudigere applicaties te bouwen zonder code. Een typische business applicatie zal echter extra programmatie nodig hebben voor integratie met andere applicaties en databases, ook om aangepaste algoritmen te kunnen gebruiken. Deze use cases worden als extensies, externe programmatie of als scripts toegevoegd.
- low-code betekent kleine schaal.
→ Er zijn genoeg praktijkvoorbeelden die dit tegenspreken. Neem bijvoorbeeld Sapphire HMS. (Bashar, 2017)

(Richardson & Rymer, 2016)

2.3.2 Geschiedenis

De oorsprong van low-code kan best bekeken worden vanuit het perspectief van de veranderingen in complexiteit en bruikbaarheid van programmeertalen doorheen de jaren. Er zijn enkele grote sprongen waarneembaar:

- **Fortran**: niet intuïtief, nuttig voor wetenschappelijke en numerieke computing.
- **Cobol**: Focus buiten wetenschappers of wiskundigen. Syntax staat dichter bij Engels. Het hielp bij het vinden van oplossingen naar business taken.
- **C**: Geschreven met Engelse syntax. Bruikbaar in een grote variatie aan applicaties.
- **Internet**: Er volgde een nood aan vereenvoudiging uit de komst van het internet en de groeiende populariteit van webapplicaties. Men is kleinere en eenvoudigere scripts beginnen gebruiken in plaats van te steunen op complexe programmeertalen. Er was een focus op functie. Applicaties moeten aan een snellere pas ontwikkeld kunnen worden en talen moeten eenvoudig genoeg zijn om dit te ondersteunen.
Een voorganger van low-code dat ook uit het internet is voortgekomen is het content management systeem. Wordpress is een bekend voorbeeld. Er zijn gelijkenissen in ideologie: het laat toe om de inhoud van een website aan te passen met een minimale hoeveelheid programmeren. Er worden modules gebruikt voor vaak voorkomende requirements. (Kissflow, 2018a)

(Kissflow, 2018b)

De ideologie achter low-code bestaat al een lange tijd. Er zijn cyclussen waar te nemen waarin de populariteit piekte. Een iteratie van 10 jaar geleden is 4gl (Fourth Generation Language). Een iteratie van 20 jaar geleden is RAD (Rapid Application Development).

Visuele en declaratieve development tools bestaan al decennia maar elke iteratie zijn er minder technische skills nodig om complexere business apps te kunnen maken die ook kunnen omgaan met moderne scenario's. Moderne low-code platformen overstijgen de limieten van hun voorgangers op een aantal manieren:

Ze zijn meer ‘open’. Vaak worden platform API's en adapters ondersteund. Soms is het mogelijk om Java of .NET code te genereren als maatregel tegen vendor lock-in. Sommige platformen gebruiken zelf open source development frameworks zoals AngularJS, Apache

Cordova en Bootstrap.

De platformen zelf zijn ‘completer’. De vorige generaties waren eerder tools. Moderne low-code platformen ondersteunen de volledige applicatie levenscyclus.

Betere integratiemogelijkheden. De meeste low-code platformen laten toegang toe tot externe data. Vaak zijn er API's voor integratie met externe applicatie modules.

(Richardson & Rymer, 2016)

2.3.3 Nood / Voordelen

Enterprise	Result	Code	Low-code
US government (Affordable Care Act)	Document compliance module	100 person-months	5 person-months
British insurance provider	Agent portal	Unknown*	10 days to minimum viable product (MVP)
Call center operator	Customer-specific app	4 months	3 weeks
Spanish insurance provider	Web channel and administration system	2.7 years (estimated)	13 weeks

*The project was on the technology management backlog list for years with little hope of ever starting

Figuur 2.5: Vergelijking van doorlooptijd (Richardson & Rymer, 2016)

Het verkooppunt van low-code is hogere snelheid van applicatieontwikkeling. Naast deze snelheid zijn er nog andere voordelen:

- Betere collaboratie tussen business en IT: Business shareholders zien hun visie sneller vorm nemen en kunnen aanpassingen sneller doorvoeren. Business regels zijn geïntegreerd in het bouwen van de applicatie.
- Focus op functionaliteit: Er wordt geen tijd verspild met schrijven van herhalende code.
- Transparante deployment: Er zijn geen aanpassingen meer nodig om verschillende architecturen te kunnen ondersteunen.
- Lange termijn gebruik: Apps zijn eenvoudig aan te passen.
- Enkele code base: Minder ruimte voor fouten en de code base wordt ook niet vervuild met tijd.

(van Schetsen, 2016)

Marktleidende platformen hebben de volgende eigenschappen:

- Eenvoudige visuele configuratie.
- Veel integratie opties.
- Mobiel compatibel.
- Schaalbaar.
- Support over de volledige app-levenscyclus.

(Kissflow, 2018a)

Vanuit het perspectief van de klant wordt duidelijk welke waardepropositie er is als eerst naar de omgeving gekeken wordt. Bedrijven moeten steeds meer kunnen omgaan met disruptieve innovatie en wijzigend klantgedrag. De vraag naar applicaties is vaak groter dan wat geleverd kan worden. Er zijn strategieën om hier mee om te proberen gaan zoals outsourcing, hackathons of voorgebouwde softwareoplossingen. Low code is een nieuwe manier om er mee om te kunnen gaan. De waardepropositie voor de business en het tech management is:

- Apps visueel configureren in plaats van manueel te coderen. Ontwikkeling kan zo mogelijk met een factor van 5 tot 10 versneld worden.
- Echte requirements en zo echte waarden vinden. Met low-code kunnen snel minimum viable products gebouwd worden om klantenideeën te toetsen.
- Live-trial van business ideeën tegen lage of onbestaande kost. Ideeën kunnen snel omgezet worden in een werkend prototype dat gedeployed en getest kan worden op de markt.
- Werkende prototypes kunnen in vijf minuten omgezet worden in productie apps. Het komt niet voor dat een app opnieuw gebouwd moet worden om een hoger volume of hogere diversiteit van gebruikers aan te kunnen.
- Meer development talent mogelijk. Developers zonder formele achtergrond kunnen door het visuele aspect nu ook business applicaties bouwen.

(Richardson & Rymer, 2016)

Digitale transformatie

Digitale transformatie is kortweg het heruitvinden van bedrijfsvoering in het digitale tijdperk. Digitale technologie wordt gebruikt om nieuwe of bestaande bedrijfsprocessen te (her)maken als antwoord op veranderingen in business- en markt requirements. Dit heeft ook effect op bedrijfscultuur en klantervaring. (Salesforce, g.d.-b)
Digitale transformatie wordt steeds belangrijker als arena waarin business een voordeel kan krijgen op de competitie. Low-code wordt aanzien als een krachtig wapen in dit proces.

2.3.4 Kritiek / Nadelen

Er zijn een aantal valkuilen waar rekening mee moet gehouden worden bij de keuze van een platform. De implementatie kan falen en het gewenste resultaat kan uitblijven als de klant geen concreet plan heeft over hoe low-code te integreren is in zijn portfolio. Het is ook aangeraden om op te passen indien er een kleinere vendor gekozen wordt. Het is een veranderende markt en deze hebben een lagere stabiliteit. (Richardson & Rymer, 2016)

Een mening is dat de business kant zijn mentaliteit ten opzichte van IT moet aanpassen. Al decennia probeert met de nood van programmeren terug te drijven en in functie hiervan worden tools gemaakt. Bedrijven zijn hiertoe aangetrokken om kosten te kunnen besparen. Maar de achterliggende kennis blijft nodig, vooral als er iets fout gaat. Niet alles kan visueel gedaan worden, development talent blijft dus nog nodig. De aanpassing in mindset die gevraagd wordt is dat business het feit aanvaart dat software ontwikkeling duur en

moeilijk is en dat er geen schortcuts voor zijn. (Reselman, 2018)

Er kan gesteld worden dan niet elk low-code platform daadwerkelijk 'low-code' is en dat men hiervoor moet oppassen:

- **NIET** low-code: Het platform genereert statisch code en verstopt deze effectief voor de gebruiker. De code is niet flexibel en schaalt niet. Dit staat digitale transformatie in de weg en verhoogt technical debt in plaats van dit te verlagen.
- **WEL** low-code: Het platform is flexibel en schaalbaar. De volledige levenscyclus van de app moet ondersteund worden. Achterliggend wordt een model gedreven architectuur gehanteerd. De flexibiliteit komt doordat de model componenten geabstraheerd worden naar XML. De applicatie kan ook makkelijker gewijzigd worden en is eenvoudig schaalbaar.

(Shiah, 2018)

Shadow IT

Gebruik van no-code tools in de handen van citizen developers kan mogelijk leiden tot shadow IT. Dit komt voor wanneer de IT organisatie te traag en procesgebonden is om snel genoeg te kunnen reageren op de noden van business users en deze buiten IT gaan om zelf een app bouwen. Zonder overzicht en coördinatie kunnen security kwetsbaarheden en compliance overtredingen verspreiden. Deze apps kunnen ook redundant en van lage kwaliteit zijn. (Bloomberg, 2017)

2.3.5 Markt en Evolutie



(a) Forrester Wave (J. R. Rymer & Koplowitz, 2019) (b) Gartner Magic Quadrant (Vincent Iijima, Driver & Natis, 2019)

Het business model van low code platformen is anders dan dat van traditionele platformen zoals Java of .NET. Er wordt uitgegaan van geleverde in plaats van beloofde business waarde. Zo zijn veel platformen gratis of tegen lage kost toegankelijk. Inkomsten worden

Forrester	Salesforce	Microsoft	Mendix	Outsystems	Kony
Gartner	Salesforce	Microsoft	Outsystems	Mendix	Appian

Tabel 2.2: Overzicht van de marktleiders

gegenereerd uit benoemde gebruikers, gedeployde apps en nodige capaciteit. (Richardson & Rymer, 2016)

Wat marktsegmentatie betreft heeft er zich op relatief korte tijd veel evolutie voorgedaan. Er kunnen drie grote stappen onderscheid worden:

1. Vijf marktsegmenten waarvan de grootste general purpose is. Het doel is om een grote variatie aan web- en mobiele applicaties te kunnen bouwen. Ter ondersteuning hiervan is er declaratieve tooling voorzien voor de creatie, integratie, deployment, life-cycle management en distributie van applicaties. Hiernaast waren ook nog segmenten voor process, database, request-handling en mobiele apps.
2. De samensmelting van segmenten naar general purpose.
3. Splitsing om te focussen op de soort gebruiker (business user of professionele ontwikkelaar).

Er worden een aantal marktontwikkelingen verwacht:

- **Consolidatie.** Grote verkopers zullen low-code platformen overnemen.
- **Marktsegment veranderingen.** Low code for mobile verdwijnt en low-code for process groeit naar general purpose segment.
- De volgende innovatie zal **ondersteuning voor IoT** zijn.

(Richardson & Rymer, 2016)

J. R. Rymer en Koplowitz (2019) identificeert vijf marktleiders. Opvallend is dat Microsoft in een korte periode naar een leidingspositie is gesprongen. De bevindingen van Richardson en Rymer (2016) sluiten hierbij aan.

Prospecten voor de komende vijf jaar:

- Tegen 2024 zal drie vierde van grote enterprises ten minste vier low code development platformen gebruiken voor zowel IT app development als citizen development initiatieven.
- Tegen 2024 zal low-code applicatie development verantwoordelijk zijn voor 65% van totale applicatieontwikkeling activiteit.

(Vincent e.a., 2019)

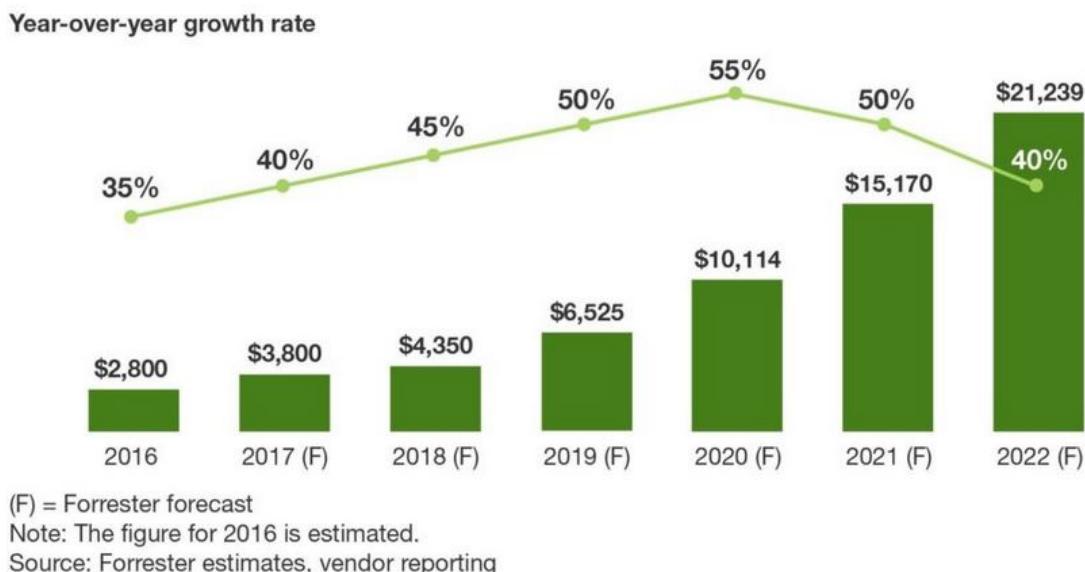
De verschuiving van traditionele software ontwikkeling naar low-code software ontwikkeling is terug te vinden in huidige jobaanbiedingen¹².

¹<https://stagebank-hbo-ict.irp.nl/internships/12543/>

²<https://stagebank-hbo-ict.irp.nl/internships/12378/>

Er wordt verwacht dat de low-code markt tegen 2022 21 biljoen dollar waard zal zijn. Er wordt ook voorspeld dat de jaarlijkse groei van 50% zich verder zal zetten. (J. Rymer, 2018)

Projected low-code development platform market growth



Figuur 2.7: Forrester voorspelde markt evolutie (J. Rymer, 2018)

2.3.6 Voorgaand onderzoek

Er is al redelijk wat praktisch onderzoek gevoerd die het potentieel van low-code illustreert maar het grootste succesverhaal is dat van Sapphire HMS. Dit is een low-code applicatie binnen de gezondheidszorg. Het ministerie van gezondheid van Koeweit had nood aan een nieuw ziekenhuis managementsysteem en Advanced Technology Company werd ingezet om dit uit te voeren. Na initieel falen om iets manueel te bouwen werd gekeken naar low-code platformen en was het mogelijk om de eerste implementatie na zes maanden in gebruik te nemen. (Bashar, 2017)

Proof of Concepts

Er werd onderzoek gedaan naar de mogelijke meerwaarde van low-code in een KMO omgeving. Uit vier marktleidende platformen werd Outsystems gekozen. Er werden POC's voor vaak voorkomende scenario's mee opgesteld. De bouw was snel, eenvoudig en zonder obstructions. De nodige investering van tijd, geld en kennis was beperkt (Devloo, 2018).

Microsoft Power Apps werd gebruikt om enkele administratieve bedrijfsprocessen van verhuisbedrijf De Borger te optimaliseren. Het resultaat werd effectief in gebruik genomen en de tijdsinst sinds implementatie werd berekend en significant bevonden. De prijs was aanvaardbaar. Er was slechts 5 euro per maand nodig om een aangepaste connector te kunnen gebruiken. (Spriet, 2019)

In een ander geval werd Power Apps gebruikt om een mobiele reporting applicatie te maken voor gebruik in de bouwsector. Er werd ook gebruik gemaakt van Power Automate om afbeeldingen te verwerken en SharePoint om deze op te slaan. (Aho, 2018)

Binnen de gezondheidszorg werd er een case study gedaan naar hoe onderzoeksdata verzameld kon worden door gebruik van een low-code platform, uiteindelijk werd er voor Mendix gekozen. De veiligheid van het platform werd als goed bevonden, relevante privacywetten waren gerespecteerd en authenticatie werd ondersteund en was eenvoudig te implementeren. Er werd verhoogde productiviteit waargenomen. (Totterdale, 2018)

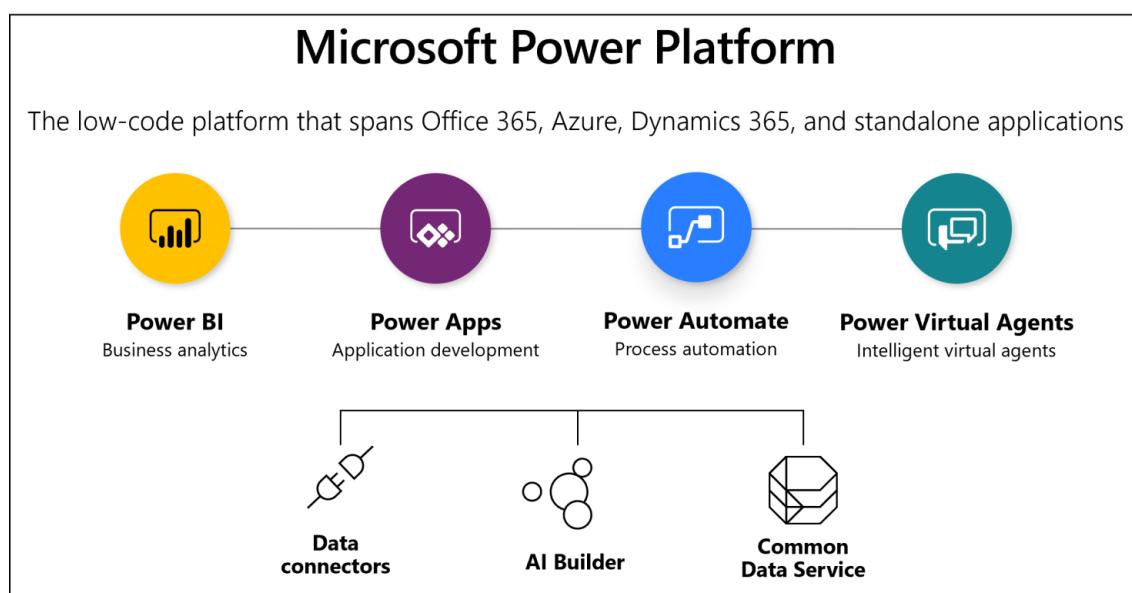
Enquêtes

Er werd onderzocht of het mogelijk zou zijn om een EHR (Electronic Health Records) systeem te bouwen met low-code om te gebruiken op nationaal niveau. Dit was naar aanleiding van het Akson planningsproject voor een nieuw EHR systeem in Noorwegen. Er werden kwalitatieve interviews afgangen bij mensen met expertise in low-code en mensen met kennis van het Akson project. (Ness & Hansen, 2019)

Low-code en traditionele softwareontwikkeling werden met elkaar vergeleken aan de hand van interviews met medewerkers bij een Fins software consultancy bedrijf. Er kon besloten worden dat zowel de medewerkers als de klanten oplossingen gebouwd met low-code verkozen. (Virta, 2018)

2.4 Microsoft Power Platform: Power Apps

2.4.1 Wat?



Figuur 2.8: Overzicht Microsoft Power Platform (Microsoft Docs, 2019d)

Power Apps is een verzameling van apps, services en connectors die snelle ontwikkeling mogelijk maakt van zakelijke apps. Voor databehoeften is er het onderliggend gegevensplatform CDS (Common Data Services) beschikbaar. Er kan ook gekozen worden uit verschillende online of on-premises gegevensbronnen via Connectors. De apps zijn responsief en kunnen zowel in een browser als mobiel (telefoon/tablet) uitgevoerd worden. (Microsoft Docs, 2019g)

2.4.2 Drie soorten apps

Canvas apps

Populairste en meest toegankelijke variant. Er is een focus op het grafische, zo worden de apps via een WYSIWYG (What You See Is What You Get) manier gebouwd. Daarmee is er volledige controle over de User Experience. Er is zowel tablet als mobile layout mogelijk. Men kan verder connecteren met meerdere databronnen. De resultaten zijn taak- en rol gebaseerde apps. (Pragmatic Works, 2019)

Model Driven apps

In tegenstelling tot Canvas apps begint men hier bij het data model. Deze data moet ook in de CDS leven. Het doel is meestal om complexe business processen te visualiseren. Het resultaat zijn dan ook end-to-end business applicaties. (Pragmatic Works, 2019)

Portals

Waar Canvas en Model apps naar binnen, naar interne business gebruikers gericht is, is Portals naar buiten gericht. Met portals kunnen extern gefocuste websites gebouwd worden waarmee gebruikers buiten de organisatie kunnen inloggen met een waaier van identiteiten en onder andere data in de CDS kunnen bekijken of aanpassen. Indien toegelaten kan inhoud ook anoniem gebrowset worden. Er zijn zowel interne als externe gebruikers mogelijk. (Microsoft Docs, 2020e)

2.4.3 Belangrijke Onderdelen

Common Data Service

De Common Data Service dient voor de opslag en beheer van business applicatie data in de cloud. Deze data wordt opgeslagen als een set van entiteiten. Een entiteit op zijn beurt bestaat uit records, analoog met tabellen in een database. Voor typische scenario's zijn er standaardentiteiten aanwezig, er kan ook gekozen worden om een aangepaste entiteit te maken. Er is logica en validatie mogelijk in de vorm van berekende velden, businessregels, workflows en business process flows. Het 'common' aspect: de databronnen zijn gelinkt, als data in de ene bron aangepast wordt zal de relevante data in de andere bron automatisch aangepast worden. (Microsoft Docs, 2019d)

Connectors

Data die een app nodig heeft is opgeslagen in een databron maar om deze te kunnen gebruiken moet er eerst een verbinding mee tot stand gebracht worden. In Power Apps wordt er de term Connector voor gebruikt. De beschikbare Connectors vallen in twee categorieën: enerzijds voor populaire services en anderzijds on-premises data opslag. Een Connector voorziet een manier om te praten met de data en kan in de app zelf dan ofwel tabellen of acties voorzien.

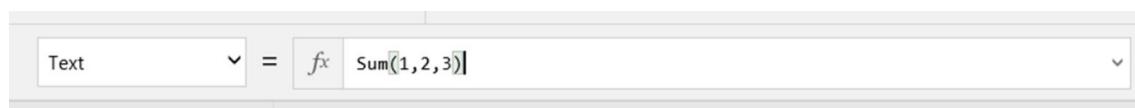
Een overzicht van de belangrijkste Connectors:

- Common Data Service
- Cloud storage: Voor elke grote provider. Bijvoorbeeld Google Drive of OneDrive.
- Excel: om Excel data te kunnen tonen in de app. Requirements zijn dat de data geformatteerd moet worden als tabel en het bestand in de Cloud opgeslagen moet zijn.
- Office 365 (Outlook): Onder andere mogelijk om mails te tonen, versturen, beantwoorden of te deleten.
- Office 365 (Users): Toegang tot de gebruikersprofielen in de organisatie.
- Oracle DB
- Power BI
- SharePoint
- SQL Server
- Twitter

(Microsoft Docs, 2020c)

On-premises data gateway: Om on-premises databronnen te kunnen gebruiken binnen een Power App is moet er een data gateway voor aanwezig zijn. Dit is een soort van brug tussen de on-premises data en de Microsoft cloud services die veilige dataoverdracht garandeert. (Microsoft Docs, 2019f)

Formules



Figuur 2.9: Overzicht van de formulebalk (Microsoft Docs, 2019a)

Logica en werken met gegevens is mogelijk met Excel-achtige expressies. In Excel worden formules gebruikt om cellen te vullen of grafieken en tabellen te maken. In Power Apps worden vergelijkbare formules gebruikt om besturingselementen in plaats van cellen te configureren. Concreet ook om om te gaan met user input. De granulariteit is: een app bestaat uit UI controls → elke control heeft properties → per property kunnen formule(s) ingesteld worden. (Microsoft Docs, 2019a)

Overig

Nadat werk aan een app opgeslagen is moet deze gepubliceerd worden om bruikbaar te zijn voor andere leden van de organisatie.

Versiebeheer is beperkt aanwezig. Er kan eenvoudig teruggegaan worden naar eerder gepubliceerde versies van een app.

UI Tests zijn mogelijk met de Test Studio maar op het moment van schrijven is dit nog experimenteel. De low-code begrippen uit PowerApps worden toegepast op het schrijven, organiseren en automatiseren van tests voor canvas apps. Zo kunnen tests geschreven worden via PowerApps expressies maar men kan ook gebruik maken van de recorder om app interacties op te vangen en er automatisch expressies uit te genereren. Deze tests kunnen dan teruggespeeld worden om app functionaliteit te valideren.

Gebruikte concepten en terminologie komen overeen met gangbare testing frameworks zoals bijvoorbeeld JUnit voor Java.

Meer uitleg aan de hand van deze terminologie.

- **Test Cases:** Test cases bestaan uit een serie instructies of acties genaamd teststappen. Test cases worden uitgevoerd om te controleren of apps of specifieke features in de app werken zoals verwacht. Deze teststappen zijn geschreven in de Power Apps expressietaal.
- **Test Suites:** Gebruikt om test cases te groeperen.
- **Test Assertions:** Gebruikt om te valideren of het verwacht resultaat overeen komt met het verkregen resultaat. Evalueert naar true of false.

(Microsoft Docs, 2019c)

Tot op heden is er nog geen IT Asset Management app gebouwd in Power Apps die als template terug te vinden is. Wel bestaat er een Asset checkout app. Dit staat er dicht genoeg bij om te kunnen gebruiken als referentie. (Meganathan, 2019)

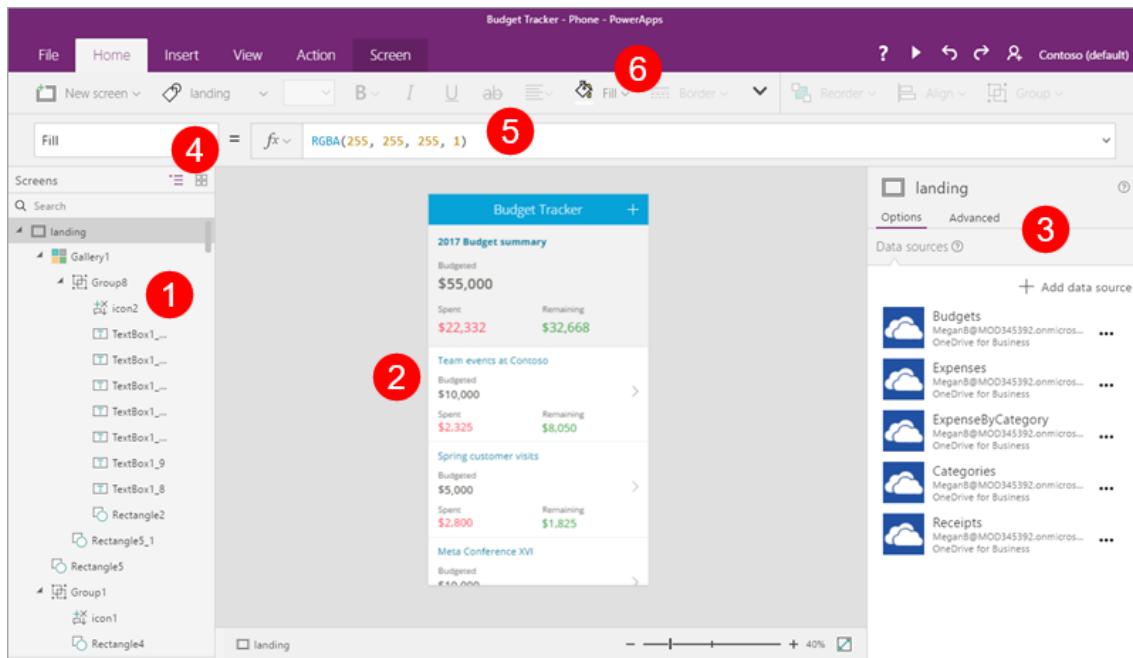
2.4.4 IDE overzicht

De onderdelen van Power Apps Studio uitgelicht volgens Figuur 2.10:

1. Links: Hiërarchische view van de app screens en controls.
2. Centraal: Huidig app scherm.
3. Rechts: Geavanceerde layout opties en databronnen.
4. Property lijst voor geselecteerde control.
5. Formulebalk.
6. Ribbon.

2.4.5 Recente Wijzigingen

De Test studio valt hierbij (zie subsectie 2.4.3).



Figuur 2.10: Overzicht van Power Apps Studio (Microsoft Docs, 2017)

De AI Builder is in publieke preview sinds 10 juni 2019. Dit maakt het mogelijk om AI te gebruiken in Power Apps met minimale technische kennis.

Een voorbeeld van het typische verloop: Een AI model type kiezen → Data verbinden (uit CDS) → AI model aanpassen naar noden (data filteren, scheduling) → AI model trainen (gebeurt automatisch) → De inzichten van het AI model gebruiken doorheen het Power Platform.

Wat AI modellen betreft zijn er enkele keuzes: enerzijds kan er een aangepast model gebruikt worden (voorspelling, formulier verwerking, object detectie, tekst classificatie), anderzijds kan geopteerd worden voor een template. In dat geval is er geen data nodig, het is gebouwd (en getraind) door Microsoft. Mogelijke keuzes zijn: Business card lezer, sleutel zin extractor, taaldetectie, tekstherkenning, gevoelsanalyse. (Microsoft Docs, 2019e)

2.4.6 Pricing, Licencing

- Power Apps voor Office 365.
- Plan 1
- Plan 2
- Power Apps for Dynamics.

(Pohl, 2019)

Elke entry in de lijst heeft meer functionaliteit dan de voorgaande.

In Power Apps voor office 365 is het enkel mogelijk Canvas apps te maken. Het aantal connectors is beperkt. De on-premise connectors bijvoorbeeld zijn niet aanwezig.

2.4.7 Uitbreidingsmogelijkheden

Power Apps studio is bedoeld als no-code omgeving gefocust op business gebruikers. Binnen deze suite is er geen optie om aangepaste code te gebruiken. De enige manier om dit mogelijk te maken is via Custom API's, Azure functions of Azure API apps³. Praktisch gezien is dit mogelijk door een project te maken in Visual studio en een OpenAPI definitie te voorzien. Als deze app in de Azure cloud staat kan deze ook opgeroepen worden vanuit Power Apps (Jugo, 2019)

2.5 Microsoft Power Platform: Power Automate

Power automate is een service voor het maken van geautomatiseerde workflows tussen apps en services. Een flow kan gemaakt worden door stappen aaneen te schakelen. Per stap kan een actie of conditie ingesteld worden. In deze condities kunnen data operaties geconfigureerd worden, er zijn ook expressies mogelijk. Naast het zelf bouwen van een flow zijn er sjablonen aanwezig voor gangbare automatisatie cases. Flows kunnen gebruikt worden vanuit Power Apps. (Microsoft Docs, 2019b)

Er zijn vijf soorten flows:

- Geautomatiseerde flow: De flow wordt gestart vanuit een bepaalde trigger.
- Button flows: De flow wordt manueel gestart. Vaak wordt dit gebruikt vanop een mobiel device.
- Scheduled flows: De flow herhaald / op een schema uitvoeren.
- Approval flows
- UI flows: Een vorm van UI automatisatie, een optie indien er geen API of connector aanwezig is (bijvoorbeeld oudere toepassingen). Voor gebruik met Windows desktop en webapplicaties. Momenteel nog in experimentele fase. Het kan gezien worden als een eenvoudige manier om een script te bouwen. De werking is dat de gebruiker een actie/set stappen opneemt, hier wordt dan een aanpasbare flow voor gegenereerd. (Microsoft Power Automate, 2019)

Beperkingen

Er is geen aangepaste code of scripting ondersteund.

³Volgens het antwoord van een Microsoft medewerker op het Power App forum: <https://powerusers.microsoft.com/t5/Power-Apps-Ideas/add-your-own-js-in-powerapps-or-call-external-js-file-easily/idi-p/869>

3. Methodologie

3.1 Requirementsanalyse

De bedoeling van deze analyse is uitvinden hoe goed Power Apps voldoet aan de requirements en mogelijks ontdekken of er een alternatief bestaat dat overwogen kan worden. Als PowerApps in de toekomst effectief gebruikt zal worden, voornamelijk door het IT en Helpdesk team, is het dan niet logisch om dit alternatief te laten conformeren aan de bedoelde eindgebruiker, zijnde professionele IT'ers? Dit idee zal meespelen bij de besluitvorming.

3.1.1 Functioneel VS Niet-functioneel

De requirements werden opgesteld tijdens herhaalde gesprekken met zowel de co-promotor als een lid van het Helpdesk team dat de app effectief zou gaan gebruiken. De resultaten zijn hieronder opgesplitst in functionele¹ en niet-functionele² requirements.

- **Functionele requirements**

- Overzicht kunnen geven van belangrijkste info voor elk toestel in het netwerk.
- Rapporten kunnen genereren.
- PC's bedienen vanop afstand (Remote Desktop Protocol kunnen oproepen).
- Gerichte/basis taken kunnen automatiseren.
- Bruikbaar zijn buiten het domein.
- Mobiel bruikbaar zijn.

¹Functionele requirement: gewenst gedrag van het systeem

²Niet-functionele requirement: kwaliteitseis waaraan het systeem moet voldoen

- Nieuwe types toestellen opnemen.
- Randapparatuur opnemen (= koppeling tussen toestellen).
- Revisie van elk dataveld per toesteltype.
- Data opslaan in SharePoint (cloud).
- Samenwerken met SCCM en/of synchroniseren met en data uit de SQL database kunnen gebruiken.
- Barcodes kunnen scannen (toegang hebben tot camera, barcode functionaliteit ingebouwd).
- AI functionaliteit.
- Het Ping commando kunnen oproepen.
- Command line toegang hebben tot PC's.
- Intune integratie.

- **Niet-functionele requirements**

- Prijs (geen prijsstijging voor de te implementeren case of voor het aantal gebruikers).
- Future proof³ zijn.
- Diverse GUI verbeteringen (specifiek kleurmarkeringen, tabbladen)/robuust GUI ontwerp ondersteunen.
- Performant zijn (specifiek met grote hoeveelheden entries/rijen kunnen omgaan).
- Leercurve moet degelijk zijn (het gebruik ervan moet aanslaan na het onderzoek).
- Veiligheid (een must, nodig omwille van gevoelige bedrijfsdata).

3.1.2 MoSCoW-methode

De volgende stap is prioriteiten stellen onder de requirements. Hiervoor wordt de MoSCoW-methode⁴ gebruikt.

- **Must have**

- Geen of beperkte meerprijs.
- Overzicht kunnen geven van belangrijkste info voor elk toestel in het netwerk.
- Rapporten kunnen genereren.
- PC's bedienen vanop afstand (Remote Desktop Protocol kunnen oproepen).
- Mobieltje bruikbaar zijn.
- Future proof zijn.
- Performant zijn.
- Veiligheid.

- **Should have**

- Gerichte/basis taken kunnen automatiseren.

(Specifieke case: een wekelijks rapport opstellen en rondmailen.)

³Hiermee bedoeld een algemene combinatie van achterliggende technologie, innovatie, consistentie en marktaanwezigheid

⁴Must have: eis moet terugkomen in het eindresultaat, Should have: eis is zeer gewenst maar het product is bruikbaar zonder, Could have: eis komt aan bod als er genoeg tijd is, Won't have: komt niet aan bod, is voor de toekomst of een vervolgproject (Wikipedia, 2020)

- Bruikbaar zijn buiten het domein.
- Nieuwe types toestellen opnemen.
- Randapparatuur opnemen.
- Revisie van elk dataveld per toesteltype.
- Data opslaan in SharePoint (cloud).
- Leercurve moet degelijk zijn.
- **Could have (nice to have)**
 - Samenwerken met SCCM en/of synchroniseren met en data uit de SQL database kunnen gebruiken.
 - Diverse GUI verbeteringen/robuust GUI ontwerp ondersteunen.
 - Barcodes kunnen scannen.
 - AI functionaliteit.
 - Het Ping commando kunnen oproepen.
- **Won't have**
 - Command line toegang hebben tot PC's.
 - Intune integratie.

3.1.3 Long List

Er zijn veel platformen om uit te kiezen⁵. Degenen hieronder opgeliijst werden besproken in de Forrester Wave (J. R. Rymer & Koplowitz, 2019) en het Gartner Magic Quadrant (Vincent e.a., 2019). De low-code markt kent een sterke groei en is zeer bewogen (zie sectie 2.3.5). Het eerste gekozen criteria om de lijst te filteren is bedoeld om hiermee om te gaan, het kan immers aangenomen worden dat een platform bestempeld door Gartner of Forrester als leider voldoende future proof is. Het andere gekozen criteria betreft het licentiemodel. Power Apps is opgenomen in de Office 365 licentie van AZ Glorieux, initieel gebruik ervan betekent geen dus geen meerprijs, een concurrent moet met andere woorden iets gelijkaardig kunnen aanbieden.

Naam	Leider		
	Forrester	Gartner	Freemium
AgilePoint			
Appian			x
Betty Blocks			
bpm'online			
Clear Software			
GeneXus			
Kony	x		x
K2			
Kintone			
MatSoft			x
Mendix	x	x	x
Microsoft	x	x	(x)
Outsystems	x	x	x

⁵ 57 volgens een oplijsting van TrustRadius

Oracle		
Pega		
Progress Software		
ProntoForms		
Quick Base		
Salesforce	x	x
ServiceNow		x
Skuid		
Thinkwise		
TrackVia		x
WaveMaker		
Zoho		x

Het resultaat is dat er van de leiders (zie tabel 2.2 voor een gefocuste weergave) slechts enkelen overblijven die in aanmerking komen.

3.1.4 Short List

Hier worden de overgebleven platformen in diepte met elkaar vergeleken maar in plaats van elke requirement af te toetsen werden deze eerst herleid naar hun achterliggende begrippen (bijvoorbeeld: mobile ondersteuning, automatisatie functionaliteit, security, leercurve en dergelijke).

Gemaakte stellingen zijn afkomstig uit PCmagazine.com reviews van elk platform en zijn aangevuld met informatie uit de officiële documentatie per platform.

Microsoft

Power Apps is deel van het Power platform (zie afbeelding 2.8). Het bestaat sinds 2016 en is daarmee het jongste platform in de vergelijking maar heeft een snelle ontwikkeling ondergaan en is op slechts enkele jaren tijd een marktleider geworden.

Werd diepgaand behandeld in de Stand van Zaken. (specifiek in sectie 2.4). Een kort overzicht:

Gefocust op business users. Apps worden gebouwd met de Power Apps Studio in de cloud. Deze omgeving is volledig grafisch, er is geen code editing. Uitbreidingen kunnen enkel ingevoegd worden via Azure Functions of Azure web apps en worden geschreven in .NET. Logica is gedeclareerd met een Excel-achtige formule taal. Er is een grote catalogus aan connectors naar externe service of dataopslag. Verbinding maken met SharePoint data is eenvoudig met de Connector ervoor. Automatisatie is mogelijk via Power Automate waarin geautomatiseerde stromen van aaneengeschakelde acties geconfigureerd kunnen worden. AI werd recent geïntroduceerd, dit component heet AI Builder, naast zelf trainen van gangbare modellen biedt Microsoft enkele voortgetrainde aan. Nog in bèta is het testing framework genaamd UI Test Studio. Power Apps zijn gefocust op interne business

gebruikers, voor externe bestaat er Portals. De leercurve is hoger dan van de concurrentie maar beschikbare documentatie en tutorials zijn talrijk en matuur. Performantie en vooral database performantie is lager dan van de concurrentie, er staat ook standaard een limiet op het aantal rijen dat per keer opgehaald kan worden (500). Microsoft toont toewijding aan het platform door constante ontwikkeling maar sommige licentiewijzigingen zijn minder goed ontvangen.

- **Forrester**

- *Positief:* Maturiteit bereikt, krachtige features en grote catalogus aan integratie adapters.
- *Negatief:* Verwarrend productaanbod en licentiëring. Voor elk product van het Power platform is bijvoorbeeld een aparte licentie nodig.

- **Gartner**

- *Positief:* Eenvoudige drag-en-drop design tool en Excel-achtige expressietaal. Snelle productie deployment mogelijk. Sterke toewijding aan LCAP markt getoond.
- *Negatief:* Model apps niet altijd beste oplossing. Verwarrende licentiëring.

Licentiëring:

	PA for Office 365	PA Plan 1	PA plan 2	PA for Dynamics 365
Canvas apps	x	x	x	x
Model driven apps			x	x
Common Data Services gebruik mogelijk		x	x	x
Standaard connectors	x	x	x	x
Premium Connectors		x	x	x
Custom connectors		x	x	x
On-premises connectors		x	x	x

Tabel 3.3: Overzicht van beschikbare plannen (Pohl, 2019)

Outsystems

Outsystems werd opgericht in Lissabon in 2001, zit momenteel in 11 landen en telt 1228 medewerkers.

Er is een focus op professionele ontwikkelaars, het is gemakkelijker om met code te werken. Het is mogelijk om op elk moment te switchen van de grafische omgeving naar de code editor. Iets anders dat dit professioneel aspect ondersteunt is dat de IDE (genaamd Service Studio) offline geïnstalleerd dient te worden. De layout ervan komt sterk overeen met die van PowerApps en Mendix. Andere grote onderdelen zijn de Forge (repository voor gebruikersgemaakte apps en plug-ins) en de Integration Studio waar extensions geschreven worden.

Een nadeel van de technische focus is dat er geen volledige cloud no-code omgeving aanwezig is. Een ander nadeel is de hogere leercurve maar er is een uitgebreid aanbod

van documentatie, tutorials online cursussen en webinars aanwezig in de OutSystems University om hier mee om te gaan.

Het platform is gebouwd in .NET. Bouwen van mobiele of webapplicaties is ondersteund maar er moet gekozen worden tussen de twee.

Een app is gestructureerd in modules: mobile (of web), service, library en extension modules. Dit om hergebruik aan te moedigen.

Automatisatie functionaliteit is beperkt tot Timers. Het is enkel mogelijk om bepaalde acties op een tijdschema uit te voeren. Wat AI betreft zijn er connectors naar Azure Luis en Azure ML. Inhuis voorziet het Outsystems.AI programma taalanalyse componenten (sleutelzin detectie, gevoelsanalyse).

UI design is minder geavanceerd dan bij de concurrentie en er is meer werk nodig om het gewenste resultaat te krijgen.

SharePoint gebruik is enkel mogelijk via de REST API. Dit is een grote beperking ten opzichte van Power Apps, waar het een standaard Connector is.

Het platform is veilig en voldoet aan verscheidene certificeringen.

Om gerust te stellen voor vendor lock-in⁶ is het mogelijk om de applicatie te genereren in .NET.

(Marvin, 2017b) en aangevuld met informatie uit de officiële documentatie⁷

- **Forrester**

- *Positief:* Krachtigste aanbod van features en consistent met introduceren van nieuwe features. Globale aanwezigheid.
- *Negatief:* Soms manuele code nodig voor integratie. Complex prijsmodel dat potentiële klanten kan afschrikken.

- **Gartner**

- *Positief:* Sterke visie en innovatie. Gedeelde Forge componenten hoog beoordeeld. Bovengemiddeld op gebied van productiviteit en modernisatie van bestaande applicaties.
- *Negatief:* Niet competitief voor bouwen van procesgeoriënteerde apps. Minder toegankelijk voor citizen developers. Prijzen kunnen snel stijgen afhankelijk van gebruik, dit gebruik wordt op een verwarringende manier berekend.

Mendix

Mendix werd opgericht in Nederland in 2005 en werd in 2018 overgenomen door Siemens voor 730 miljoen dollar. Er zijn enkele grote partnerships aangegaan, onder andere met SAP die Mendix resellt als het SQL Cloud Platform.

Net als bij Outsystems is de doelgroep eerder professionele ontwikkelaars. Er is zowel een volledige no-code cloud omgeving aanwezig (Mendix Studio) als een uitgebreidere variant (Mendix Studio Pro) die offline geïnstalleerd moet worden, eerder gericht op developers, aan wie het dan mogelijk gemaakt wordt om te coderen in Java.

Er is een groot aanbod aan voorgebouwde templates en componenten die Microsoft en

⁶Gedwongen een product of service moeten blijven gebruiken, onafhankelijk van kwaliteit, omdat het (financieel) niet praktisch is om er vanaf te stappen (Cloudflare, g.d.)

⁷<https://success.outsystems.com/Documentation>

Licentiëring:

	Free	Basic	Standard	Enterprise
Unlimited apps	(x) ⁸	x	x	x
Gebruikers	100	1000	Geen limiet	Geen limiet
On-premises of private cloud			x	x
Support		8x5	8x5	24/7
Aantal omgevingen	1	3	3	5
CI & deployment		x	x	x
Maandelijkse prijs	Gratis	>\$4000	>\$10000	Custom

Tabel 3.5: Prijsmodel Outsystems (Outsystems, g.d.)

Outsystems evenaren. De UI filosofie is om te starten met design en wireframes, dan het model te maken met logica en workflows die in dat model passen. De apps zijn responsief. Het scherm wordt automatisch aangepast tussen smartphone, tablet of desktop views.

Database integratie is beter dan bij Outsystems. Database wijzigingen worden automatisch herkend in de app, bij Outsystems is dit niet het geval. Specifiek naar SharePoint toe is verbinding net als bij Outsystems beperkt tot een REST API.

App creatie is algemeen gezien gestroomlijnder dan bij Outsystems. Er kan na afloop gedeployed worden naar verschillende cloud omgevingen. App aanpassingen zijn eenvoudig (versionering is voorzien) maar voor elke nieuwe versie van het platform moet er gemigreerd worden.

Testing en analytics zijn geavanceerder dan bij concurrenten zoals bijvoorbeeld PowerApps waar dit nog in bèta zit. Bij Mendix is een hoge mate van maturiteit bereikt, tests worden bijvoorbeeld automatisch uitgevoerd.

Over specifieke onderdelen: Buzz is een portaal dat dient als sociaal intranet waar collaboratie begint. SCRUM is ingebouwd in het platform. Integraties, plugins, gebruikersapps zijn in de Mendix App Store te vinden, dit aspect is even matuur als varianten van Microsoft en Outsystems.

Wat AI functionaliteit betreft is er integratie mogelijk met IBM Watson.

Security is als verwacht voor een marktleider:

(Marvin, 2017a) en aangevuld met informatie uit de officiële documentatie⁹

- **Forrester**

- *Positief*: Trendsetter in features, krachtig aanbod. Goede ondersteuning voor app levenscyclus, uitbreidingen naar CI toe. Sterke partners: SAP, Siemens en IBM.
- *Negatief*: Extra code nodig voor integraties. Services voor content management binnen apps is iets zwakker. Prijzen voor platform adoptie moeilijk te voorspellen.

⁹<https://docs.mendix.com/>

- **Gartner**

- *Positief:* Competitief sterk door invloed van middelen na overname door Siemens. Aantrekkelijk voor enterprise door autoscaling, high availability en lage latency. Aparte ontwikkelomgeving afhankelijk van bedoelde gebruiker. AI ondersteunde ontwikkeling. Mogelijk complexe applicaties te maken. Tevredenheid van professionele ontwikkelaars.
- *Negatief:* Na overname door Siemens zou toewijding aan Mendix kunnen veranderen weg van mainstream LCAP. Minder bruikbaar door citizen developers. Prijzen en contract flexibiliteit slecht beoordeeld.

Licentiëring:

	Free edition	Single app	Professional	Enterprise
Number of environments	1	2	2	3
Horizontal scaling				x
Support for CI & deployment				x
On-premises				x
App user limit	unlimited	unlimited	unlimited	unlimited
Number of apps	unlimited ¹⁰	unlimited	unlimited	unlimited
Price	Free	>\$1917	custom	custom

Tabel 3.7: Prijsmodel Mendix (Mendix, g.d.)

Salesforce

Salesforce is een Amerikaans cloud-georiënteerd software bedrijf met hoofdzetel in San Francisco, Californië. Het verleent sinds 1999 CRM¹¹ services en complementaire enterprise applicaties waaronder het Lightning low-code platform.

Het platform biedt opties aan voor zowel low-code als traditionele ontwikkeling. Er is een focus op CRM en het doelpubliek van de apps is business gebruikers. Voor externe gebruikers bestaat er de Lightning Community Builder. Dit is analoog met Microsoft zijn PowerApps en Portals.

De apps kunnen enkel vanuit het Salesforce portaal gebruikt worden. Hosting gebeurt altijd in de Salesforce cloud. Apps zijn standaard responsief.

De Lightning App Builder is een point-en-klik low-code oplossing. Ondersteunend zijn er de Process Builder om datamodelen op te stellen en de Schema Builder voor complexe geautomatiseerde logica. In de Schema Builder is gebruik van de binnenshuis ontwikkelde programmeertaal Apex ook mogelijk. De syntax ervan komt overeen met Java. Vanuit Apex kan data gemanipuleerd worden aan de hand van SOQL (Salesforce Object Query Language).

¹¹https://nl.wikipedia.org/wiki/Customer_relationship_management

De UI wordt gebouwd met componenten uit het Lightning UI framework. Er is een sterke focus op herbruikbaarheid.

Er kunnen onder andere REST en SOAP API's gebruikt worden maar dit moet ingesteld worden via Apex.

Net als bij Outsystems kan app bouw geassisteerd worden door AI. Salesforce doet dit met Einstein. Hierbuiten kan Einstein ook gebruikt worden voor traditionele toepassingen in dit domein zoals AI assistentie en image herkenning met de Vision API. Chatbots bouwen is mogelijk met Einstein Bots.

Voor professionele ontwikkelaars is er het Heroku web development platform. Hier is men vrij om te kiezen welke programmeertaal gebruikt wordt.

Het heeft de meest uitgebreide feature-set en de beste aanbieding van community componenten en apps in de App Exchange. De keerzijde hiervan is dat de features overweldigend kunnen worden (feature bloat, UI clutter).

Er zijn trails (tutorials) beschikbaar op de trailhead interactieve leer-en training site maar de pagina's zijn niet altijd up-to-date.

Het is mogelijk dat database aanpassingen problemen geven in de app. (Marvin, 2017c)

Salesforce biedt een gratis Developer Edition aan. Dit heeft alle features en is bedoeld om te leren. Er is ook een proefversie mogelijk. De betalende opties zijn te zien in Tabel 3.8

- **Forrester**

- *Positief:* Grote partnerlijst en security certificaten.
- *Negatief:* Code moeten gebruiken om aan noden te kunnen voldoen.

- **Gartner**

- *Positief:* App Exchange, innovatie (onder andere Einstein en IoT mogelijkheden)
- *Negatief:* Zowel Lightning als Apex moeten gebruiken om requirements uit te kunnen werken.

Licentiëring:

	Platform Starter	Platform Plus
Custom objecten	10	110
Lightning App Builder	x	x
AppExchange	x	x
Process Automatie	x	x
Lightning Console	(x) \$25 extra	x
Maandelijkse prijs	\$25	\$100

Tabel 3.8: Prijsmodel Salesforce (Salesforce, g.d.-a)

Appian

Net als Salesforce is dit een grote speler. Functioneel is het even sterk als Outsystems en Mendix maar helaas is er geen gratis licentie mogelijk.

Temenos Quantum (Kony)

Dit platform is door Vincent e.a. (2019) bestempeld als een leider en heeft een gratis licentie. Wat serieuze overweging tegenhoudt is de recente overname (en naamswijziging).

3.1.5 Besluit

Er zijn geen redenen gevonden om weg te stappen van PowerApps als primaire kandidaat om de POC mee uit te werken ook al kan het schrijven van uitbreidingen omslachtig worden. Er zijn twee concurrenten gevonden: Outsystems en Mendix. Wat functionaliteit betreft zijn ze inwisselbaar. Beiden zijn gefocust op professionele ontwikkelaars en hebben aantrekkelijke gratis licenties. De grootste belemmering betreft het gebruik van SharePoint als dataopslag. Waar het bij PowerApps standaard ondersteund is als Connector moet er in Mendix en Outsystems hun geval een REST API gebruikt worden. Bij gebruik van één van beiden is het dus interessanter om een alternatieve opslagmethode te gebruiken.

Na overweging is voor Outsystems gekozen om de secundaire POC mee uit te werken. Wat doorslag gaf is de gratis versie die van Outsystems net iets beter is.

Bovendien vallen de resource noden van de POC volledig binnen de limieten van het plan, in Subsectie 5.3.1 staat dit uitgelicht.

4. POC: Power Apps

4.1 Voorbereiding

De POC is gemaakt in de cloud omgeving van PowerApps met de Office 365 licentie van HoGent en AZ Glorieux. De beperkte mogelijkheden van deze licenties werden omzeild door gebruik van het Community Plan dat alle functionaliteitsrestricties opheft in een persoonlijke ontwikkelomgeving. Concreet beperkt de Office 365 licentie uitbreidingen en toegang tot lokale resources. Eigen geschreven logica is mogelijk via custom connectors, deze moeten gehost of op zijn minst gedeclareerd zijn in Microsoft Azure. Dankzij het Azure Student plan via HoGent werd de nodige cloud capaciteit verschafft.

Eens de nodige licenties aanwezig zijn is er toegang tot de Power Apps ontwikkelomgeving. Om een canvas app te maken zijn er verscheidene opties:

- **Beginnen vanuit data:** Er wordt een gegevensconnector gekozen die de app data verschaft. Op basis van deze data wordt een app gegenereerd met drie schermen: een overzichtscherm met lijstweergave van de items, een detailscherm dat geopend wordt na klikken op een item uit het vorige scherm dat meer datavelden toont en een edit scherm, geopend vanuit het detailscherm waar men de datavelden kan aanpassen. Het detail- en edit scherm toont de data via een formulier control.
- **Blanco beginnen:** Een app ontwikkelen vanuit een letterlijk leeg canvas.
- **Sjabloon:** Er zijn sjablonen voorzien voor gangbare scenario's.

Er is ook keuze tussen telefoon en tablet layout. De huidige case leent zich tot de eerste optie maar hierin is men beperkt tot de telefoon layout. Een tablet layout is beter geschikt voor de hoeveelheid data die getoond moet kunnen worden en de extra grafische controls nodig voor sommige requirements. Om deze reden werd voor een leeg canvas met tablet

layout gekozen.

Meer over app gebruik: apps worden geopend vanuit het PowerApps portaal. Op Windows 10 kan via de Power Apps Store app snelkoppelingen voor de apps toegevoegd worden aan Start. Analoog hiermee kan op een smartphone via de Power Apps app snelkoppelingen aan het thuisscherm toegevoegd worden.

4.1.1 Data en data weergave

Om data te kunnen gebruiken moet de connectie ermee toegevoegd worden in de app. Er werden een aantal verschillende connecties gebruikt tijdens het verloop, deze kunnen gegroepeerd worden per datatype:

- Excel
 - OneDrive for Business
 - Excel Online
 - Google Drive
- SharePoint
 - SharePoint Connector

Deze data wordt in de app gevisualiseerd aan de hand van Galerijen en formulieren. De flow waarmee een galerij geconfigureerd wordt heeft een bepaalde structuur:

Gegevensbron selecteren → indeling wijzigen → weer te geven velden aanpassen.

Dit is niet zo speciaal maar het is inbegrepen omdat het een fundamenteel deel is van app configuratie in PowerApps.

Voor de meeste connectors wordt data ingelezen als Table. Als dit niet mogelijk is worden API calls gebruikt. Een sterk punt is dat in geval van relationele data de nodige joins automatisch gedaan worden. (Lindhorst, 2018)

Naast Tables zijn er uiteraard variabelen. Deze kunnen een globale context of context per scherm hebben en worden impliciet gedeclareerd.

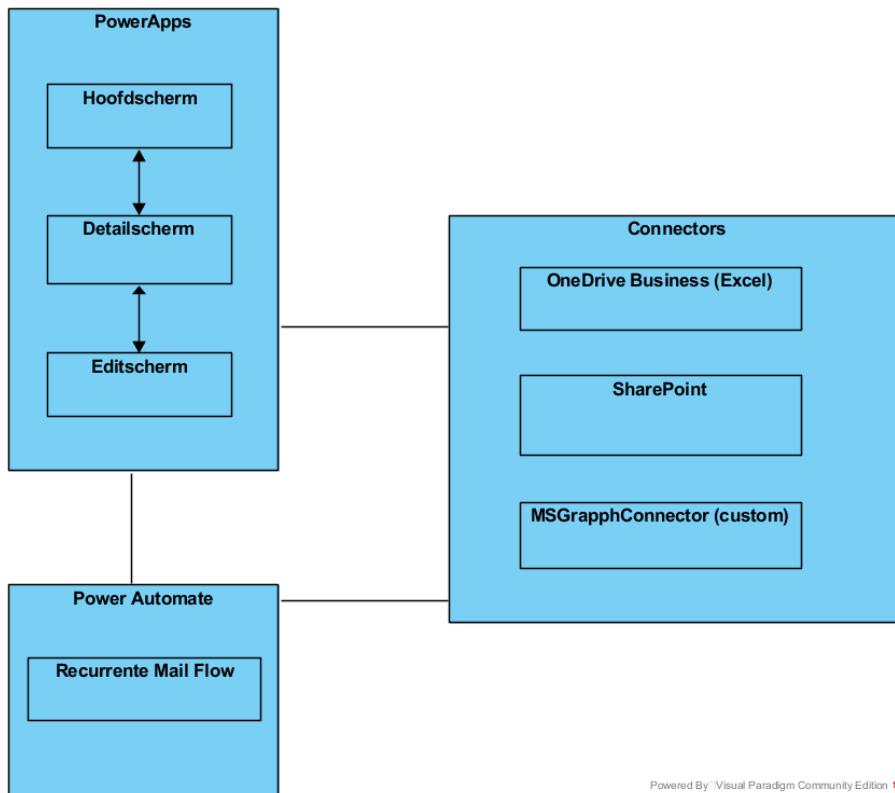
4.2 SharePoint Configuratie

Voor het bouwen van de POC werd een testsite voorzien. De asset data werd in een enkele Lijst geplaatst. Er werd beperkte validatie ingesteld.

Het is de bedoeling dat in de definitieve versie Content Types gebruikt zullen worden: in grote lijnen een overkoepelend type met gemeenschappelijke kolommen en dan unieke subtypen (pc/Laptop, Switch, Server).

Dit zal geen effect hebben op de PowerApps POC omdat er automatische Lookups gedaan worden.

4.3 Model van Opstelling



Figuur 4.1: Model van de PowerApps proof-of-concept.

4.4 Requirements

4.4.1 Must have

Prijs

Tijdens het maken van de POC werd gevonden dat de enige requirements die niet uitgewerkt kunnen worden met de bestaande licentie het complex filteren en rapportage zijn (Custom Connector nodig). Indien er niet vanaf gedaan kan worden is er een aanvullend stand alone plan mogelijk. Dit houdt in dat individuele gebruikers applicaties (twee apps en één portal) zonder functionaliteit beperkingen kunnen uitvoeren voor €8.40 per maand¹. Deze app wordt door vier personen op de helpdesk gebruikt, dat brengt het totaal op €33.60.

Overzicht kunnen geven van belangrijkste info voor elk toestel in het netwerk

Zoals reeds besproken in Sectie 4.1 kan deze functionaliteit gegenereerd worden. Ook al is er voor tablet layout gekozen is het aantal realistisch weer te geven

¹<https://powerapps.microsoft.com/en-us/pricing/>

PA met complexe functionaliteit	NEE	JA
<= 2 apps	-	>= €33.60
>2 apps	-	>= €134.40

Tabel 4.1: Meerprijs bovenop huidige licentie

velden in het overzichtsscherm beperkt en werden de netwerknaam, omschrijving en dienst geselecteerd. Visuele indicatie van status was belangrijk in LanReview maar in plaats van de hele tekst van een rij in te kleuren is gekozen voor een gekleurd bolletje aan het hoofd van elke rij.

Data wordt aan de serverkant door SharePoint gevalideerd, het resultaat van deze validatie wordt getoond in de app zelf en dit is aangevuld met beperkte client side validatie. Dit komt neer op het aanwezig moeten zijn van de netwerknaam, het MAC-adres en device serial. In het detail- en edit scherm worden respectievelijk detailform en editform controls gebruikt. Elk veld hierin wordt voorgesteld door een data card. Het is de 'required' eigenschap van een data card die uitwijst of een veld ingevuld moet zijn.

De flow van deze requirement wordt duidelijk aan de hand van het overzicht van navigaties tussen de schermen:

Beweging	Code
home → detail	Navigate(DetailScreen; ScreenTransition .None)
home → edit	NewForm(Editform);;Navigate(EditScreen; ScreenTransition .None)
home ← detail	Back()
detail → edit	EditForm(Editform) ;; Navigate(EditScreen; ScreenTransition .None)
detail ← edit	Back()

Tabel 4.2: Overzicht navigaties tussen schermen

Filtering / rapportage

Filtering (met Excel databron):

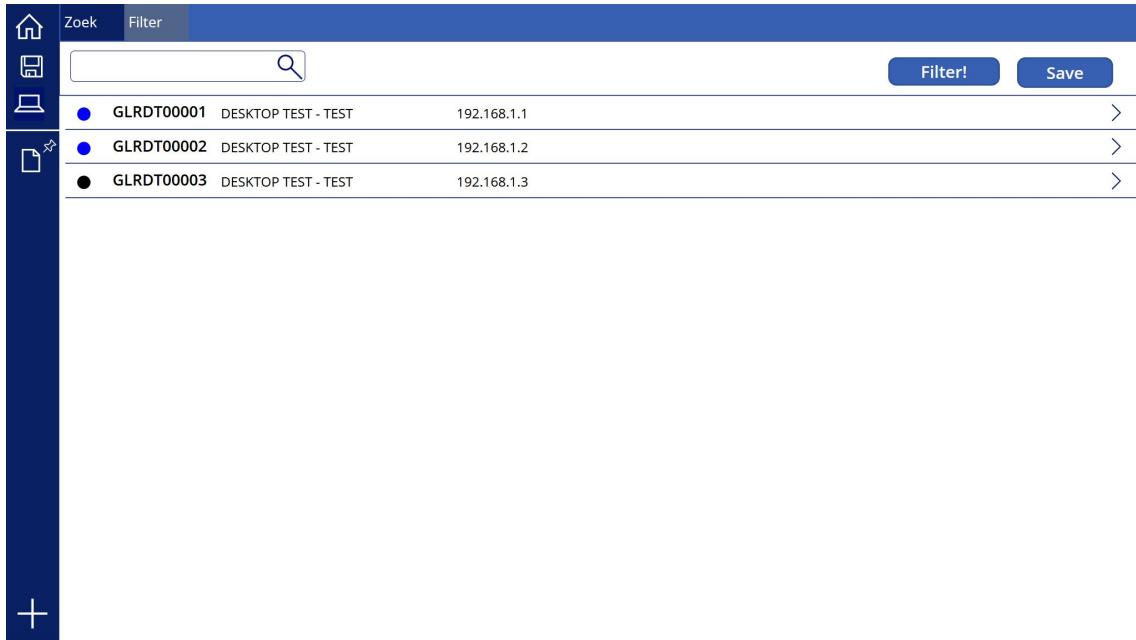
Er moet een onderscheid gemaakt worden tussen 'zoeken' en 'filteren'. Zoeken is het matchen van tekst in een zoekveld aan een vooropgesteld aantal kolommen elke keer deze tekst wijzigt. Implementatie ervan in PowerApps is eenvoudig te doen met één formule:

```
| Search(Tabel1_1; SearchInput.Text; "NETWNAAM"; "NAAM"; "IP")
```

In geval van filtering moet het resultaat voldoen aan bepaalde condities. In PowerApps is een Filter formule aanwezig. Een voorbeeld implementatie kan zijn:

```
| Filter(Tabel1_1; MODEL = 'HP COOLBOOK' && OS = 'Windows 10')
```

Dit oogt statisch. In LanReview is het mogelijk de kolom, operator en de filterwaarde in te stellen. Bovendien kunnen meerdere condities aaneengeschakeld worden. Een



(a) Hoofdscherm

The screenshot shows a PowerApp interface with a blue header bar. On the left is a vertical navigation bar with icons for Home, List, and Document, and a plus sign for adding new items. The main area has a table titled 'Tabel 1' with various columns and rows of data. At the top right are icons for star, trash, and edit.

DIENST	INST	IP
HELPDESK	AZ	192.168.1.1
KLEUR	LOKATIE	MAC
BLAUW	HELPDESK	08.00.27.8F.43.60
NAAM	NETWNAAM	OPM
DESKTOP TEST - TEST	GLRDT00001	eerste testentry
OS	PATCH	POORT
WINDOWS 10	N001	1
RECTYPE	SERIAL	SOORTIP
COMPUTER	SP44RG5F77	DHCP GERESEERVEERD
SP	TYPEMERK	SWITCH
	HP COOLBOOK	KAST A SWITCH 1
USER		
TESTUSER		

(b) Detailscherm

gelijkaardige manier om filter data te verschaffen in PowerApps is door een galerij te gebruiken waarbij elke rij een dropdown heeft voor de te filteren kolom, een dropdown voor het nodige vergelijkingsteken en een aan te passen tekstveld. Elke keer men een conditie toevoegt wordt deze data opgeslagen als rij in een Tabel.

Wat nu met mogelijkheden om de filter effectief uit te voeren? Er zijn een aantal opties, telkens met stijgende complexiteit.

1. In PowerApps zelf met behulp van formules.

Dit is niet mogelijk omdat we de te filteren kolom niet kunnen bepalen aan de hand van een variabele, er is geen string substitutie mogelijk. Het is ook niet mogelijk om het aantal condities dynamisch toe te wijzen (in het 'Filter' commando)).

→ **NEE**

2. Via een Power Automate flow

- Filteractie 'Een lijst maken met rijen in een tabel' (ODATA filter-query)²

Wel: query kan als argument worden meegegeven.

Niet: 'and' operaties zijn niet ondersteund

- Filteractie 'Matrix filteren' (flow expressie)³

Wel: 'and' operaties zijn ondersteund

Niet: query kan niet als argument worden meegegeven.

→ **NEE**

3. Door gebruik van een Custom Connector.

→ **Ja** (zie Sectie 4.5)

Ter verduidelijking: een samengestelde query in zowel ODATA als flow expressie:

- **ODATA:** (NETWNAAM eq 'GLRDT00001')and (IP eq '192.168.1.1')
- **Flow expressie:** @and>equals(item() ?['NETWNAAM'], 'GLRDT00001'),equals(item() ?['IP'], '192.168.1.1'))

Dit beoogde soort filtering is overigens out of the box aanwezig in model based apps. (Microsoft Docs, 2020a)

Als het dynamische aspect van filtering opgegeven wordt is het nog steeds mogelijk om 'statisch' te werken. Daarmee bedoeld een aantal voorgebouwde queries, voor gangbare scenario's zoals ze bestaan in LanReview zijn zonder problemen op te nemen in PowerApps.

Filtering (met SharePoint databron):

Filtering met SharePoint kan anders aangepakt worden, om dit aan te tonen worden de uitvoeringsopties opnieuw overlopen.

1. In PowerApps zelf met behulp van formules.

→ **NEE**

2. Via een Power Automate flow

ODATA filters uitgevoerd op SharePoint items hebben geen beperking op het aantal via 'and' aaneengekoppelde operaties, de filter kan dus in Power Automate uitgewerkt worden.

→ **JA**

De belangrijkste operaties van de flow (Figuur 4.3) zijn:

²Filtering uitgelegd onder 'System Query Option \$filter' <http://docs.oasis-open.org/odata/odata/v4.0/errata03/os/complete/part2-url-conventions/odata-v4.0-errata03-os-part2-url-conventions-complete.html>

³<https://docs.microsoft.com/en-us/power-automate/use-expressions-in-conditions>

- **Items ophalen:** De ODATA filter query wordt als enkel string argument vanuit PowerApps doorgegeven naar de flow en is bruikbaar in het 'FilterQuery' veld.
- **Reactie:** Als eerst een test antwoord werd voorzien als voorbeeld om het schema uit te genereren wordt er een bruikbare json naar Powerapps teruggegeven.

Gebruik in de PowerApp (Figuur 4.4):

De kolommen van SharePoint zijn enkel via een speciale soort naam toegankelijk, er moet mapping gebeuren. Daarvoor wordt een tabel gebruikt waarbij 'colPA' de kolomnaam is voor weergave in de PowerApp en 'colAUT' de bruikbare kolomnaam is in Automate.

```
Set( ColNamesTable ; Table({ colPA :"NETWNAAM" ; colAUT :" Title " } ; { colPA :"NAAM" ; colAUT :"yh3z" } ; [..])) // entries voor eerste twee kolommen getoond .
```

Elke keer op 'formatquery' geduwd wordt:

```
OnSelect => Collect( FormattedQueryHTTP ; LookUp( ColNamesTable ; colPA = colNameDropdown_1. SelectedText. Value ) . colAUT & " " & LookUp( OperatorsTable ; eqPA = eqNameDropdown_1. SelectedText. Value ) . eqAUT & " " & FilterValueText_1 & " " )
```

De waarden worden gemapt naar odata aanvaardbare waarden (via lookup), aan elkaar geplakt en toegevoegd aan tabel FormattedQueryHTTP als een enkelvoudige filter operatie. Wanneer de query uitgevoerd wordt ('dynquery' knop):

```
OnSelect => Set( HTTPFilterResults ; 'PowerApps-knop' . Run( Left( Concat( FormattedQueryHTTP ; Value & " and " ) ; Len( Concat( FormattedQueryHTTP ; Value & " and " ) ) - 5 ) ) )
```

De waarden van de tabel worden geconcateneerd met ' and' ertussen. De laatste 'and' wordt ervan gehaald via len (.., -5). De flow wordt uitgevoerd.

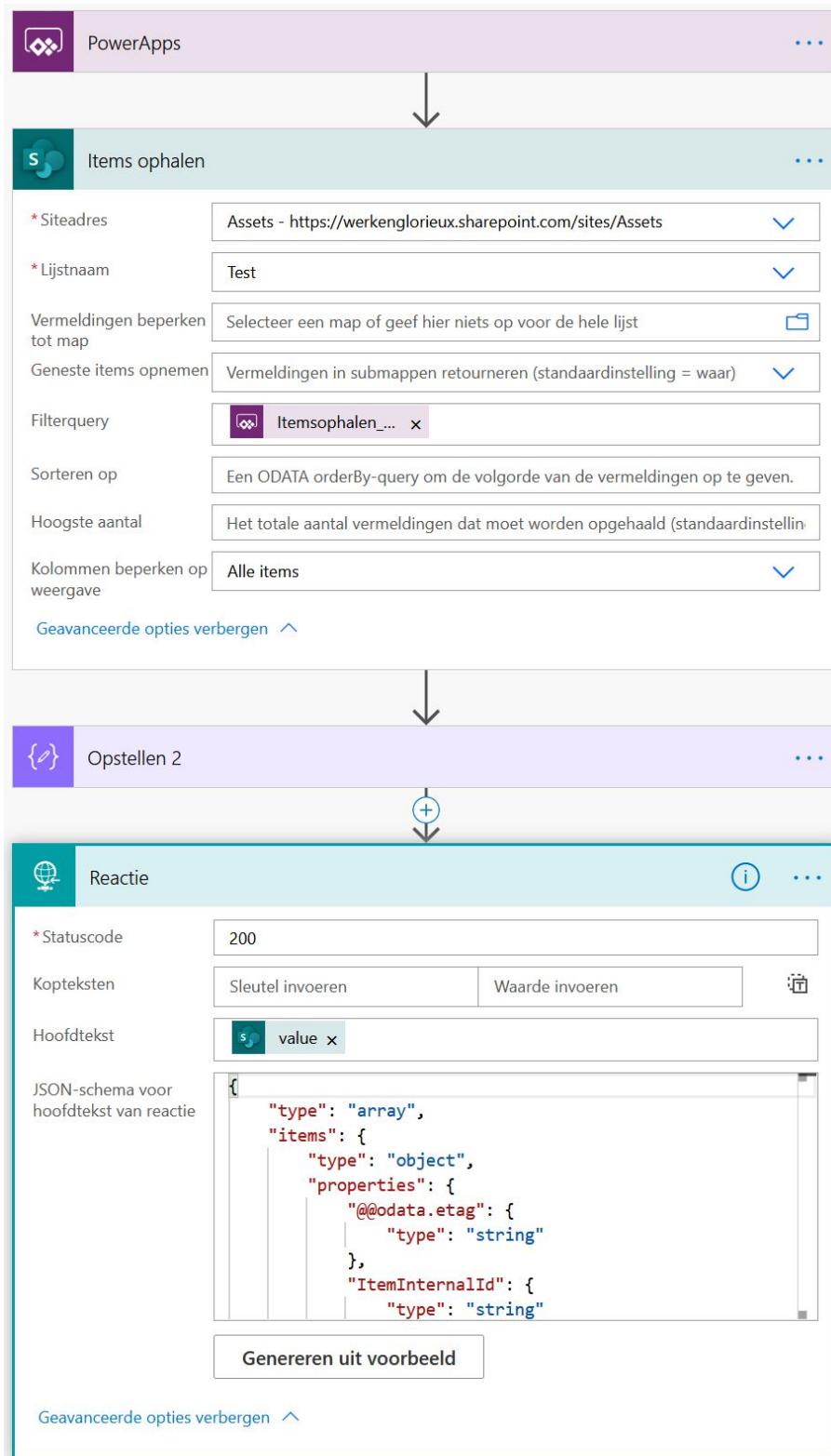
Reporting:

Het is niet mogelijk om data met een Power App lokaal op te slaan. De workaround is om de resultaten weg te schrijven naar een Excel bestand in OneDrive dat specifiek dient voor reporting.

Remote Desktop/Ping

Dit stelt een probleem. In LanReview is het mogelijk te rechtsklikken op een item en een remote desktop sessie te beginnen of een eenvoudige ping uit te voeren. Achter de schermen wordt dit met de command prompt uitgevoerd. In Powerapps is het om te beginnen niet mogelijk om te rechtsklikken. Custom code uitvoering is standaard niet ondersteund. Zelf een custom connector biedt hier geen oplossing omdat de app in de cloud leeft en aan serverzijde uitvoeren van RDP of ping heeft geen nut. Er is een workaround nodig.

PowerApps heeft een SaveData() methode om lokaal data op te slaan. Een typisch scenario is dat er geen internet is, app bewerkingen kunnen tijdelijk lokaal opgeslagen worden



Figuur 4.3: De Automate flow waarmee SharePoint gefilterd wordt.

om definitief uit te werken als gedetecteerd wordt dat er terug internetverbinding is. Een eerste logische beperking is dat de SaveData() niet benaderbaar is buiten PowerApps,

The screenshot shows a SharePoint search interface. At the top, there are filter options: 'Zoek' (Search), 'Filter', 'USER' (selected), 'is gelijk aan' (is equal to), 'TESTUSER'. Below the filters, there are two rows of search terms: 'jlco eq 'HELPDESK'' and 'moww eq 'TESTUSER''. On the right side of the interface are three buttons: 'resetArgs', 'formatQuery', and 'dynamicQuery'. Below these buttons, the search results are displayed in a table:

	GLRDT0001	DESKTOP TEST - TEST	HELPDESK	192.168.1.2	>
●	GLRDT0002	DESKTOP TEST - TEST	HELPDESK	192.168.1.3	>
●	GLRDT0003	DESKTOP TEST - TEST	HELPDESK	192.168.1.4	>

Figuur 4.4: Voorbeeld van SharePoint filtering in de PowerApp.

een tweede beperking is dat deze functionaliteit niet mogelijk is wanneer de app in een webbrowser en niet in een smartphone/tablet gebruikt wordt. Het idee om een IP-adres naar een configuratiebestand weg te schrijven om dan buiten de PowerApp te gebruiken om naar een pc te RDP'en of pingen stopt hier.

Het is mogelijk om een data gateway op te zetten. Dit zorgt voor een verbinding met on-premises data. Dit stelt een actie beschikbaar in Power Automate waarmee een bestand lokaal aangemaakt kan worden. Via een workaround kan er PowerShell mee uitgevoerd worden. (Luca, 2017)

Hier is niet voor gekozen om deze redenen: extra kosten, resource gebruik en configuratie voor een in verhouding kleine use case. Bovendien zou de data gateway ingesteld moeten worden op een gedeelde locatie, anders zou deze data gateway op de pc van elke PowerApp gebruiker moeten staan.

Een deel van het hierboven staande is bruikbaar, namelijk het idee om het IP-adres van het te pingende/rdp'ende toestel voor te bereiden op een pc lokaal zodat de laatste stap manueel uitgevoerd kan worden. Dit moet in een bestandsformaat zijn waar Power Apps vlot mee overweg kan. Excel is een optie.

De uitwerking wordt:

In OneDrive een Excel file aanmaken met een kolom voor elke gebruiker van de app. De header waarde is de voornaam van de gebruiker. Er is maar één rij aanwezig waar per kolom het huidige te pingende/rdp'ende IP-adres in staat. Instellen dat dit bestand lokaal gesynchroniseerd wordt (OneDrive is standaard Windows 10 functionaliteit).

De gebruiksflow is:

1. Het detailscherm openen van de beoogde pc.
2. Op de knop duwen om het IP-adres weg te schrijven.

```
OnStart => Set(Gebruiker; First(Split(User().FullName;" ")).Result)
OnSelect => Patch(ipTable; First(Filter(ipTable; user = Gebruiker))
    ;{ ip: SelectedDevice.IP })
```

De formule gekoppeld aan 'OnStart' haalt de volledige naam van de ingelogde PowerApps gebruiker op en splitst deze waarna enkel het eerste resultaat (voornaam) wordt opgeslagen als 'Gebruiker' variabele.

De formule gekoppeld aan het 'OnSelect' property haalt het IP-adres uit het detail-formulier op en schrijft het weg naar de Excel tabel 'iptable' onder de kolom met de matchende gebruikersnaam.

3. Lokaal kan nu een script uitgevoerd worden dat dit Excel bestand inleest en het relevante commando uitvoert (keuze hiervan hangt af van de overeenkomende snelkoppeling).

```
$excepad = "C:\Users\Thomas\OneDrive – Hogeschool Gent\rdpping_ips.xlsx"
$iptable = Import-Excel -Path $excepad
$ip = $iptable.Where({$_.user -eq $env:USERNAME}).ip

if ($args[1] -eq "ping"){
    ping $ip
} elseif ($args[1] -eq "rdp"){
    Start-Process "$env:windir\system32\mstsc.exe" -ArgumentList "/v:$ip"
} else{
    // extra commando
}
```

Het 'import-excel' commando is mogelijk met de ImportExcel module⁴.

Voor het gemak wordt de uitvoering van het nodige commando gedaan via een snelkoppeling, de verwijzingen zijn:

```
ping => C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
      -NoExit -command "& 'H:\Documenten_TOSH\HoGent_Docs\2019–2020\
      Batchelorproef\C#\ apps\localtools.ps1' -commando ping"
RDP => C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -
      NoExit -command "& 'H:\Documenten_TOSH\HoGent_Docs\2019–2020\
      Batchelorproef\C#\ apps\localtools.ps1' -commando rdp"
```

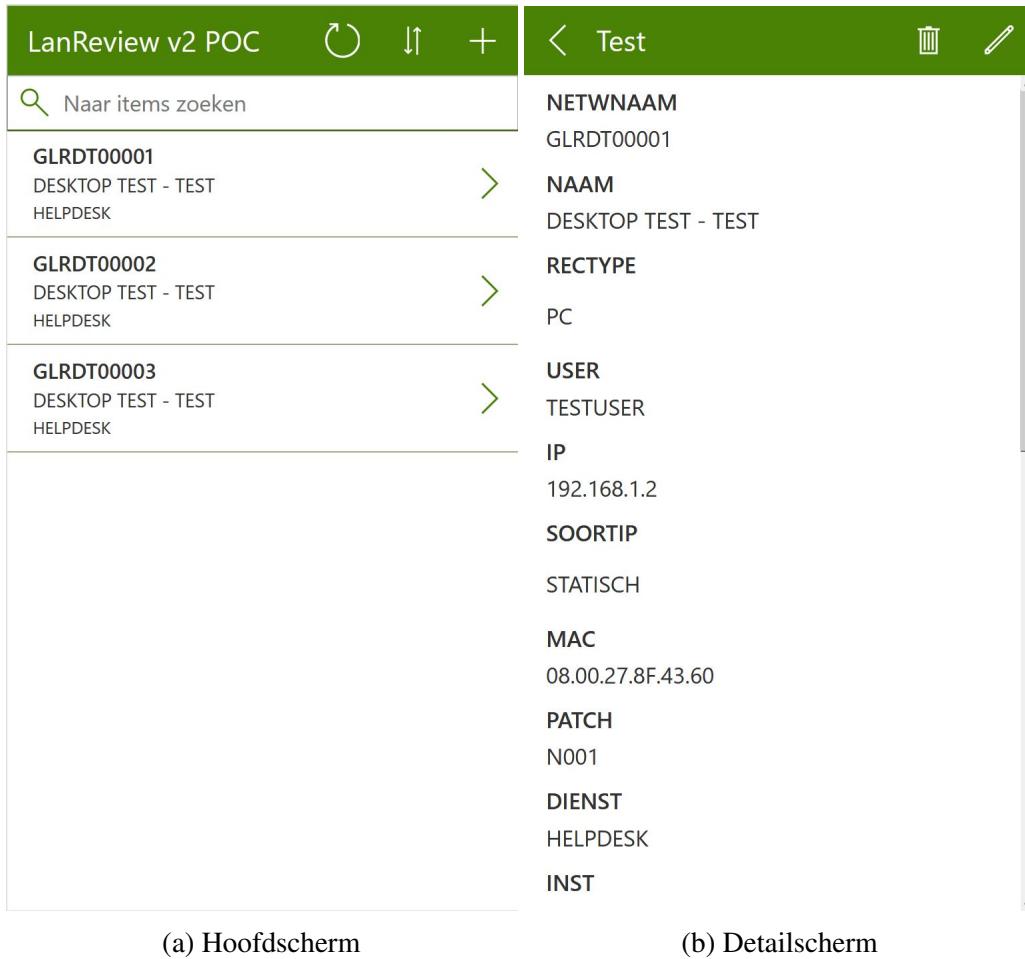
Het nadeel is dat door deze extra overhead er 3-5 seconden gewacht moet worden voor het bestand gesynchroniseerd is en de nodige snelkoppeling uitgevoerd kan worden met het juiste IP-adres.

Mobiel bruikbaar zijn

Een Power App is onafhankelijk van platform of devicetype maar er wordt bij creatie wel gevraagd op welke layout er gefocust zal worden en het is zo dat de app er te groot of te klein uit zal zien als de app later gebruikt wordt op het toesteltype waar het niet voor ontworpen werd.

Om deze reden werd er een versie van de app gemaakt specifiek voor mobiel gebruik met een beperkte functionaliteitsset (Figuur 4.5). Deze app zal voornamelijk gebruikt worden om snel een overzicht op te vragen van de belangrijkste info van een pc en om het toevoegen van nieuwe toestellen te vergemakkelijken met barcode scanning functionaliteit.

⁴<https://github.com/dfinke/ImportExcel>



(a) Hoofdscherm

(b) Detailscherm

Figuur 4.5: Overzicht van de beperktere mobiele PowerApps POC

Future proof zijn

Zowel J. R. Rymer en Koplowitz (2019) als Vincent e.a. (2019) stelden dat klanten de licentiëring van producten in het Power platform verwarring vonden. Hierbovenop komt dat deze licentiëring onderhevig kan zijn aan verandering (Pohl, 2019).

Microsoft is een sterke aanwezigheid maar hetzelfde kan niet altijd gezegd worden over hun aangeboden producten (Bott, 2018).

De low-code markt is nog steeds turbulent volgens de eerder aangehaalde reports. Als gekeken wordt naar de cyclische aard van low-code en de voorgangers beschouwd worden (vb, COBOL, 4GL, RAD) stelt de vraag zich of dit vroeg of laat ook voorbijgestreefd zal worden. (Reselman, 2018)

Performant zijn

Een cloud applicatie die een lijst gebruikt in SharePoint of een Excel file in OneDrive zal niet even performant zijn als een lokale applicatie die een lokale Access databank gebruikt. Er zijn echter een aantal technieken om data operaties te optimaliseren zoals data voorladen met 'ClearCollect', calls parallel uitvoeren met 'ConcurrentCall' en laden van

data voor niet-zichtbare UI controls uitstellen met de 'Delayed Load' optie. (Andaluissi, 2018)

Er is nodige inefficiëntie geïntroduceerd in de app door gebruik van workarounds, bijvoorbeeld (subsectie 4.4.1).

Er is kritiek dat gegenereerde code niet even geoptimaliseerd kan zijn als specifiek geschreven code. (Shiah, 2018)

Een belangrijke aanpassing die gemaakt moet worden is om de app 2000 kolommen op te laten halen in plaats van de standaard ingestelde 500. Er zijn meer dan 1000 toestellen en zoek- en filteroperaties zouden anders maar op de eerste 500 toegepast worden.



Figuur 4.6: Instelling voor aantal op te halen rijen

Security

Eens een app gepubliceerd is kan gekozen worden wie de app mag gebruiken. Dit kan ook op groepsniveau gebeuren door te publiceren naar een bepaalde Environment. Goedgekeurde gebruikers hebben toegang met hun Office 365 login, authenticatie gebeurt met andere woorden via Azure AD. Er moet opgepast worden dat deze gebruikers over de nodige permissies beschikken voor de aanwezige connectors om ontbrekende functionaliteit te vermijden.

Er zijn enkele technieken om granulaire permissies in te stellen, die van Dunnam (2019) wordt hier uitgelegd:

Scenario: Enkel bepaalde gebruikers mogen een formulier aanpassen.

Toepassing: Een (SharePoint) lijst opslaan met de e-mailadressen van de gebruikers met bevoegdheid. Bij app start controleren of men in deze lijst zit. In de app zal de editknop enkel maar voor hun zichtbaar zijn.

Concreet:

```
| OnStart => ClearCollect(acceptedApprovers, Filter(Approvers, Title = User().FullName))
| Visibkle (editknop) => If(IsEmpty(acceptedApprovers), false, true)
```

Bovenstaande werd niet ingewerkt maar iets dergelijk zou later gebruikt kunnen worden om de app open te stellen voor een groter publiek (buiten enkel de IT).

4.4.2 Should have

Gerichte/basis taken kunnen automatiseren

De belangrijkste case hiervoor is dat er een wekelijks overzicht moet worden gemaakt voor alle apparaten die langer dan een maand de status 'te schrappen' gekregen hebben, gekenmerkt door een 'KLEUR' veld met waarde 'BLAUW'. Een toestel krijgt de 'te schrappen' status wanneer ze verwijderd/vervangen worden uit de actieve omgeving en in opslag worden geplaatst. Het is dan de gewoonte om een maand te wachten voor een toestel effectief verwijderd wordt (in geval van ontbrekende data of een functioneel probleem in de nieuwe oplossing).

Dit is als flow geïmplementeerd, zie Figuur 4.7



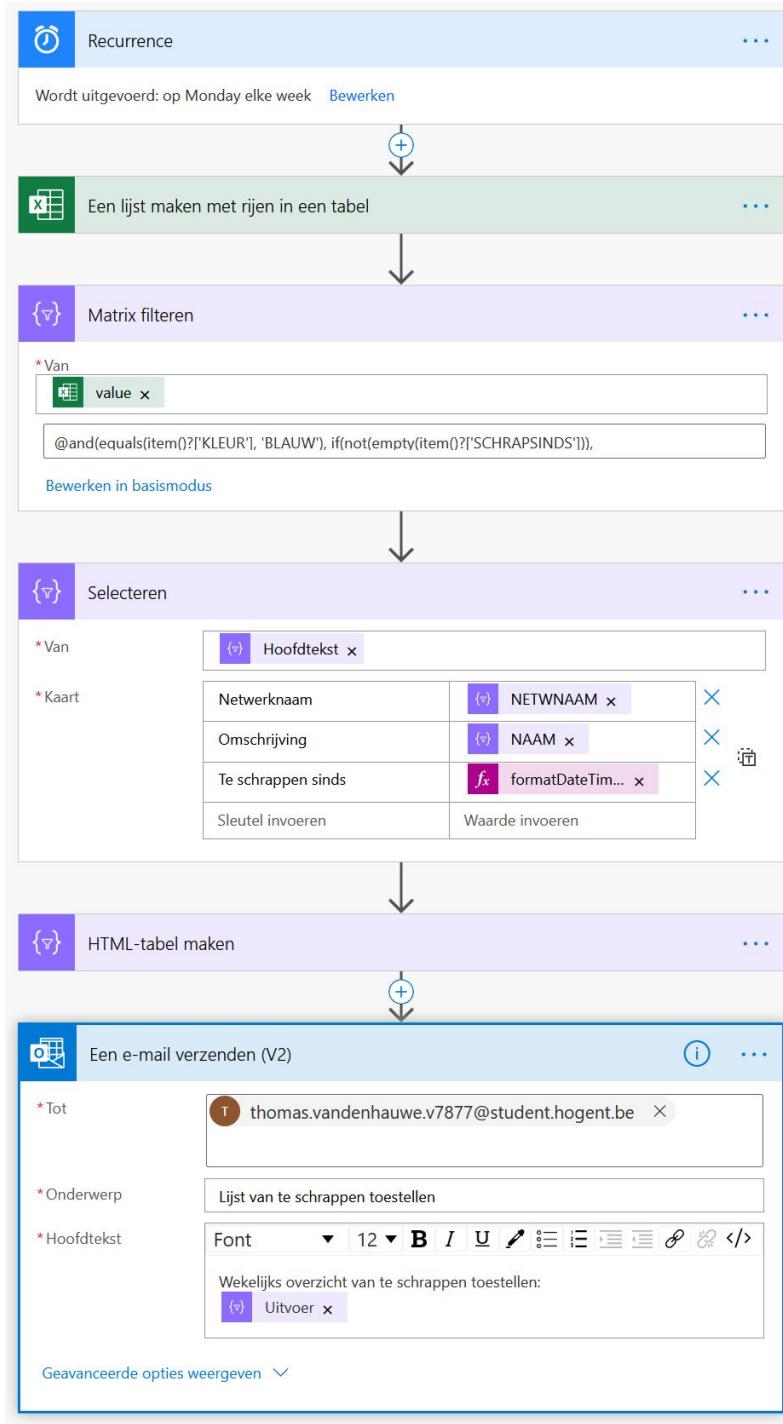
Figuur 4.7: Recurrente flow met SharePoint databron.

Er is ook een flow uitgewerkt met Excel als databron. Het verschil is dat een datum wordt teruggegeven als integer en dat enkele formules aangepast moeten worden om hiermee om te gaan (Figuur 4.8).

- 1. Recurrence:** Het is een recurrente flow. Het tijdschema is wekelijks, maandag om 3.00u.
- 2. Een lijst maken met rijen in een tabel:** In deze stap worden de rijen uit de nodige Excel tabel enkel opgehaald, filtering gebeurt in de volgende stap. Er wordt dus niet met ODATA gefilterd maar via flow expressies.
- 3. Matrix filteren:** Om te voldoen moet KLEUR BLAUW zijn, het SCHRAPDATUM veld mag niet leeg zijn en moet een datum bevatten van meer dan een maand geleden.

```

SharePoint => @and( equals(item()['b0et']['Value'], 'BLAUW'), if(
    not(empty(item()['SCHRAPDATUM'])), lessOrEquals(item()['SCHRAPDATUM'],
    addDays(utcNow(), -31)), false))
Excel => @and( equals(item()['KLEUR'], 'BLAUW'), if(not(empty(item()['SCHRAPSINDS'])), lessOrEquals(addDays('1899-12-30', int(
  
```



Figuur 4.8: Recurrente flow met Excel databron.

```
|     item () ?[ 'SCHRAPSINDS' ] ) ) , addDays( utcNow () , -31) , false ))
```

Excel geeft datums terug als integers van het aantal dagen sinds 1900. Als dit toegevoegd wordt aan '1899-12-30' is de correcte datum terug opgesteld en kan dit vergeleken worden met de huidige datum `utcNow()`. Dit werd besproken in de

PowerApps forums⁵

4. **Selecteren:** De in de mail te gebruiken kolommen uit het resultaat van de filter halen. Zo kunnen ze gebruikt worden in de volgende stap.
5. **HTML-tabel maken:** Het resultaat van de vorige stap wordt als argument opgegeven. Er wordt een HTML-tabel opgesteld.
6. **Een e-mail verzenden (V2):** Het reële adres zal dat van de helpdesk zijn. In de body van de mail kan de HTML-tabel ingevoegd worden.

Het resultaat is dat iedereen op de helpdesk automatisch een mail krijgt met het overzicht.
Figuur 4.9



Wekelijks overzicht van te schrappen toestellen:

Netwerknaam	Omschrijving	Te schrappen sinds
GLRDT00002	DESKTOP TEST - TEST	01-04-2020

Figuur 4.9: Een voorbeeld van de mail.

Bruikbaar zijn buiten domein

De app leeft in de cloud. Het enige dat nodig is zijn geldige Office 365 credentials en een internetverbinding.

SharePoint requirements

Requirements die met SharePoint te maken hebben zijn:

- Nieuwe types toestellen opnemen.
- Randapparatuur opnemen.
- Revisie van elk dataveld per toesteltype.
- Data opslaan in SharePoint (cloud).

Deze worden besproken in Sectie 4.2

Leercurve moet degelijk zijn

(Dit deel is gebaseerd op persoonlijke ervaringen van de auteur)

- **Uiterlijk → Gemakkelijk:** Uiterlijk is eenvoudig te configureren met drag-en-drop. Layout eigenschappen moeten niet vaak aangepast worden doordat controls in positie

⁵<https://powerusers.microsoft.com/t5/Building-Flows/Excel-Online-Date/td-p/134200>

'springen'. Groepen visuele controls kunnen gekopieerd en geplakt worden.

- **Data en variabelen → Opletten:** Databegebruik kan uitdagend zijn indien niet goed ingepland. Gebruik van variabele is eenvoudig maar het overzicht erover kan verloren gaan omdat ze gedeclareerd worden bij het eerste gebruik
- **Gedrag en uitbreidingen → Moeilijk:** Gedrag instellen is moeilijk, formules hebben een leercurve. Uitbreidingen schrijven is ook moeilijk, voorkennis van REST principes was nodig.

Als een case out of the box ondersteund is op PowerApps zal een business gebruiker er geen problemen mee hebben. Zeker niet als ze voorkennis van Excel hebben.

4.4.3 Nice to have

Samenwerken met SCCM en/of synchroniseren met en data uit de SQL databank kunnen gebruiken

Dit werd niet geïmplementeerd maar er is wel een mogelijke oplossing voor gevonden. In essentie gaat het om een script dat volgens een tijdschema uitgevoerd wordt op de SCCM server dat SCCM cmdlets gebruikt om asset data op te vragen die weggeschreven worden naar een SharePoint omgeving.(Ziehnert, 2020)

Het vorige betreft synchronisatie. Alternatief kan de AdminService REST API gebruikt worden om rechtstreekse toegang te krijgen tot SCCM data vanuit PowerApps of Automate. (Gross, 2019)

De configuratie hiervan valt buiten de scope van het onderzoek, de grote stappen illustreren dit:

SMS Provider API (SCCM) → SMS Provider rol (site server) → Cloud Management Gateway → certificaat → Azure App registratie → Custom Connector

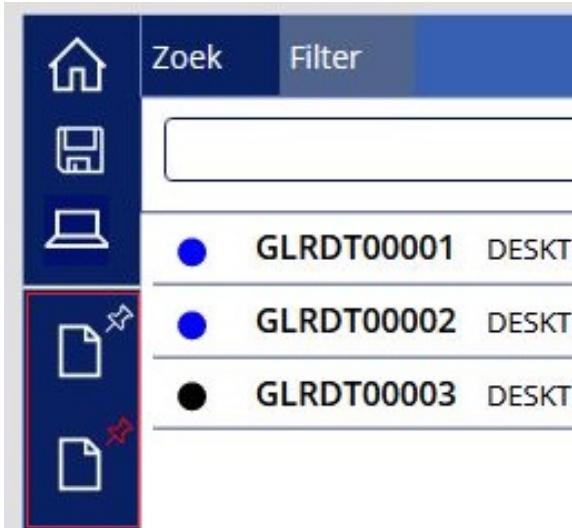
In de POC zijn er bovendien geen requirements die nood hebben aan directe toegang tot SCCM en de Custom Connector brengt extra kosten mee.

Diverse GUI verbeteringen/robuust GUI ontwerp ondersteunen

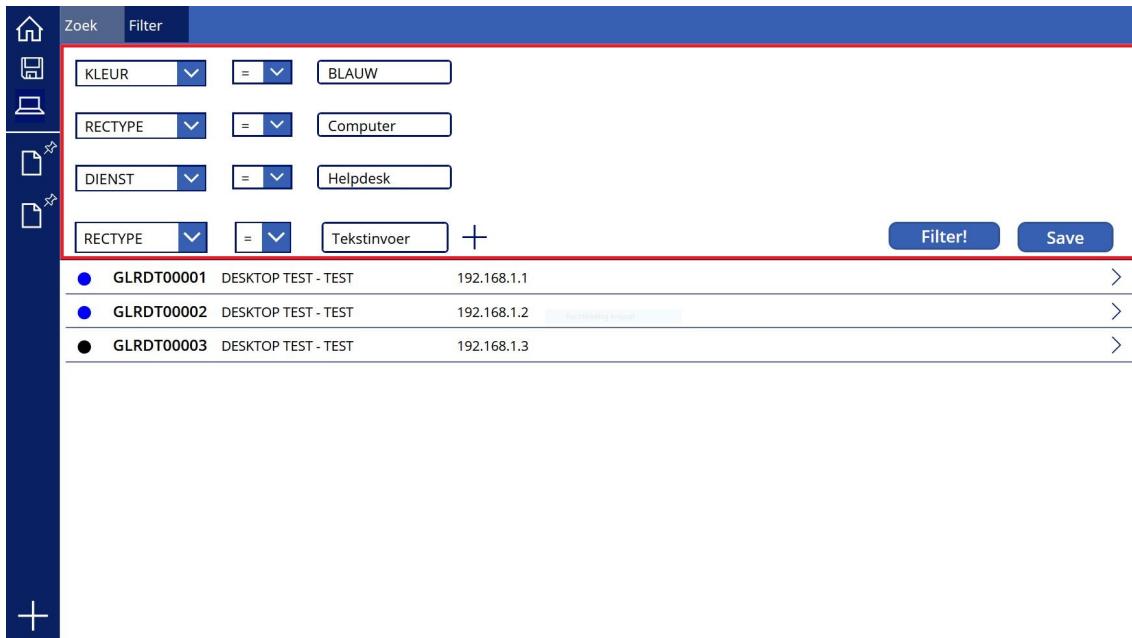
Het is mogelijk items vast te pinnen op de toolbar. Toevoegen gaat door rechtsboven op het 'pin' symbool te klikken in de detailweergave van een pc. Losmaken van de toolbar is analoog door op de rode 'pin' rechtsboven van het icoon te klikken. Als gehovered wordt over de toolbar wordt de Netwerknaam ook getoond. (Figuur 4.10)

Voor duidelijkheid zijn de zoek in filter velden van elkaar gescheiden door tabs. Tabs worden gemaakt met Gallerij controls.

De filter layout is ook gemaakt met een Gallerij control. De layout past zich dynamisch aan als er filtervelden worden toegevoegd. (Figuur 4.11)



Figuur 4.10: Een voorbeeld van vastpinnen aan de toolbar.



Figuur 4.11: Dynamisch aanpassen van layout aan filter condities.

Barcodes kunnen scannen

Er is een control beschikbaar specifiek om barcodes in te scannen. Dit maakt het proces om deze functionaliteit toe te voegen aan de app zo eenvoudig als het aanmaken van een Label. Er dient dus geen speciale connector toegevoegd of geconfigureerd te worden. Standaard kan deze scanner enkel gebruikt worden op een tablet of smartphone maar via de instellingen kan een alternatieve webbarcode scanner gekozen worden.

Het gebruiksscenario is dat men tijd wil besparen bij het toevoegen van nieuwe toestellen door de barcode en serial in te scannen in plaats van manueel over te typen (en de kans op fouten te vergroten). Toestellen worden soms in bulk aangekocht dus de potentiële

tijdswinst is aanzienlijk.

Dit is geïmplementeerd door een barcode knop toe te voegen aan de MAC en SERIAL data cards in het editschermer. Indien er op de knop geduwd wordt opent de camera, wordt de barcode waarde uitgelezen en toegevoegd aan het relevante tekstveld.

Ondersteunend moeten maar twee property waarden aangepast worden.

- Button → OnScan: Als de scanner gebruikt werd wordt de uitgelezen waarde toegevoegd aan een context variabele.

```
| UpdateContext({ ScannedMac: BarcodeScanner1 . Value })
```

- TextField → Default: Als er een ScannedMac variabele aangemaakt werd wordt deze als tekst gebruikt in plaats van de standaardwaarde.

```
| If ( IsBlank ( ScannedMac ) ; Parent . Default ; ScannedMac )
```

AI functionaliteit

Er werd geen enkele requirement opgesteld die AI nodig zou hebben maar de mogelijke opties werden toch even verkend.

Zoals reeds besproken werd in Subsectie 2.4.5 zijn er verschillende opties. Via AI Builder kan een model getraind worden of kan een voorgebouwd model gebruikt worden voor een typisch scenario. Gebruik van AI Builder is prijzig (€421.70 per unit/maand), er is een calculator⁶ om exacte berekeningen te maken. Een alternatief is om AI functionaliteit te introduceren aan de hand van een specifieke connector.

Een eenvoudige voorbeeldcase die zo'n connector gebruikt (bijvoorbeeld Azure Computervision⁷) is om de Netwerknaam uit te lezen van het label die op een PC plakt met tekstherkenning. Indien het aantal API calls beperkt blijft is het gebruik van deze connector bovendien gratis.

4.5 Custom Connector

Een Custom Connector is de enige optie om code of een niet-ondersteunde databron te introduceren in PowerApps. Er zijn twee varianten onderzocht:

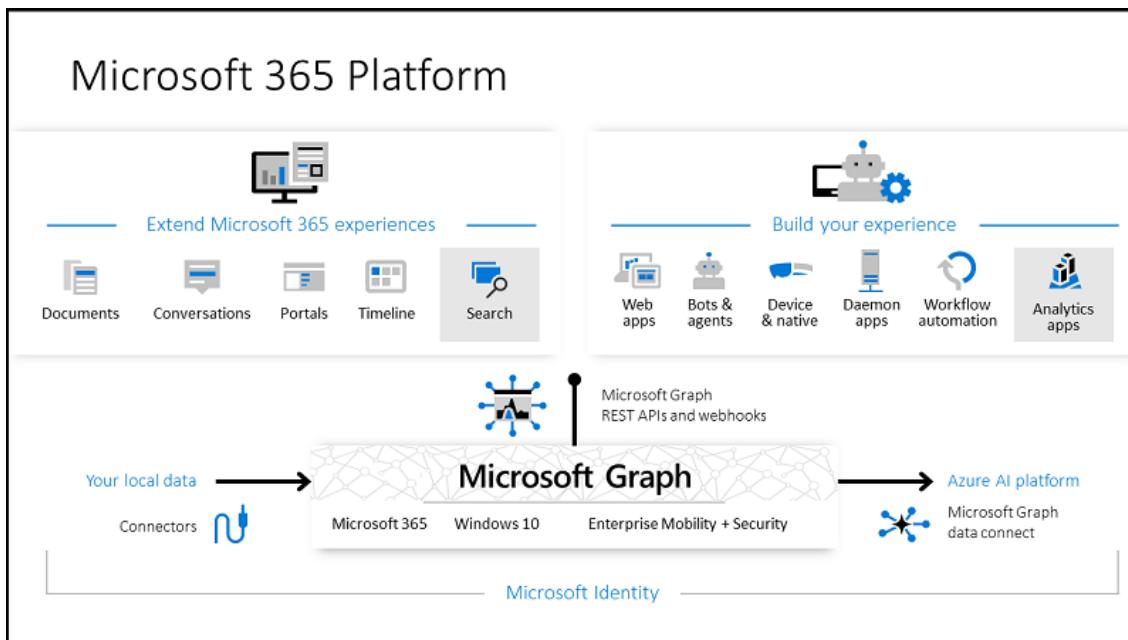
- Azure API App (Swagger definitie)
- Blank Custom Connector

Het doel is om een complexe filter uit te werken. Een eerste idee was om alle data naar de API te posten en de filtering binnen de app zelf uit te werken. Hier werd snel vanaf gestapt naar de Microsoft Graph API.

⁶<https://powerapps.microsoft.com/nl-nl/ai-builder-calculator/>

⁷<https://azure.microsoft.com/nl-nl/services/cognitive-services/computer-vision/>

4.5.1 Microsoft Graph



Figuur 4.12: Microsoft Graph in het Microsoft 365 platform (Microsoft Docs, 2020d)

Microsoft Graph is een REST⁸ API waarmee allerhande data uit Office 365 services opgevraagd kan worden. De specifieke case is het filteren van Excel data in OneDrive. Er volgt een overzicht van de nodige stappen en bijbehorende requests om dit te verwezenlijken. In dit eenvoudige voorbeeld wordt gefilterd op een pc met Netwerknaam 'GLRDT00001'. Niet-relevante header waarden zijn weggelaten.

1. Een sessie creëren in de Excel file met data. Dit is nodig om de filter effectief toe te kunnen passen in stap 3. In de response body is de sessie-id te vinden.

```
POST https://graph.microsoft.com/v1.0/me/drive/root:/testdata.xlsx
      :/workbook/createsession
BODY => {persistChanges: true}
```

2. Reeds bestaande filters verwijderen moesten deze nog aanwezig zijn.

```
POST https://graph.microsoft.com/v1.0/me/drive/root:/testdata.xlsx
      :/workbook/worksheets('Blad1')/tables('tabell1')/clearFilters
```

3. De filter toepassen. De sessie-id moet meegegeven worden als header waarde. In de URL is de kolomnaam te vinden 'columns('NETWNAAM')'. De operator en eigenlijke filterwaarde staan in de request body '"criterion1": "=GLRDT00001"'.

```
POST https://graph.microsoft.com/v1.0/me/drive/root:/testdata.xlsx
      :/workbook/worksheets('Blad1')/tables('tabell1')/columns(
      'NETWNAAM')/filter/apply
HEADER => workbook-session-id: {session-id}
BODY =>
{
  "criteria" :
```

⁸REpresentational State Transfer <https://restfulapi.net/>

```

    { "filterOn": "custom",
      "criterion1": "=GLRDT00001"
    }
  }

```

4. Het resultaat van de filter opvragen. Enkel de 'values' tag volstaat, daarom wordt erop gefilterd in de visibleView met 'rows?\$select=values'

```
GET https://graph.microsoft.com/v1.0/me/drive/root:/testdata.xlsx
:/workbook/worksheets('Blad1')/tables('tabel1')/range/
visibleView/rows?$select=values
```

4.5.2 Azure API App (ASP.NET)

Via Microsoft Graph is er toegang tot de data maar nu is er vanuit de Power Apps zelf toegang nodig tot Microsoft Graph. De optie die de meeste vrijheid biedt is het maken van een .NET API in Visual Studio, dit wordt naar Azure gedeployed (als Azure API App) en later geïmporteerd in PowerApps als Custom Connector aan de hand van een Swagger definitie. Er is ook de keuze of de app gebouwd wordt in ASP.NET framework of .NET Core. Om praktische redenen werd gekozen om de ASP.NET framework variant te maken, het is namelijk mogelijk om de nodige Azure app registratie vanuit Visual Studio zelf uit te voeren. In .NET zijn er aantal klassen beschikbaar waarmee Microsoft Graph bewerkingen uitgevoerd kunnen worden waarvan de belangrijkste 'GraphServiceClient' is. Het gebruik ervan wordt verder uitgelegd.

Voorbereiding: Azure App registratie

Connected Services

Add code and dependencies for one of these services to your application



Figuur 4.13: Microsoft Graph in de Connected Services

In de solution staat in de 'Connected services' lijst een optie om te verbinden met Office 365 services via de Microsoft Graph (Figuur 4.13). In deze wizard wordt een nieuwe Azure App registratie aangemaakt (indien nog niet bestaand) en kunnen de nodige permissies

aan de hand van 'scopes' bepaald worden. De scopes om OneDrive bestanden te kunnen gebruiken zijn:

- User.Read
- Files.Read
- Files.Read.All
- Files.ReadWrite
- Files.ReadWrite.All

Na afloop zijn er enkele belangrijke stukken gegevens ingevoegd in de Web.config, een korte verklaring:

- **TenantId:** Identifier van de Active Directory gebruiker/tenant, ook de id van de map waar de resources zich bevinden.
- **ClientId:** Identifieert de app registratie in Active Directory.
- **ClientSecret:** Secret geassocieerd met de ClientId.

Nu kan er overgegaan worden op de implementatie. Er zijn twee technieken toegepast om de 'GraphServiceClient' klasse te configureren.

De eerste is gebaseerd op de techniek gevonden op CodeProject⁹:

```
//
public static async Task<GraphServiceClient> GetGraphServiceClient()
{
    var authentication = new
    {
        Authority = "https://graph.microsoft.com",
        Directory = WebConfigurationManager.AppSettings["ida:TenantId"],
        Application = WebConfigurationManager.AppSettings["ida:ClientId"],
        ClientSecret = WebConfigurationManager.AppSettings["ida:ClientSecret"]
    };

    var app =
        ConfidentialClientApplicationBuilder.Create(authentication.Application)
            .WithClientSecret(authentication.ClientSecret)
            .WithAuthority(AzureCloudInstance.AzurePublic,
                authentication.Directory)
            .Build();

    var scopes = new[] { "https://graph.microsoft.com/.default" };

    var authenticationResult = await app.AcquireTokenForClient(scopes)
        .ExecuteAsync();

    var graphServiceClient = new GraphServiceClient(
        new DelegateAuthenticationProvider(x =>
    {
        x.Headers.Authorization = new AuthenticationHeaderValue(
            "Bearer", authenticationResult.AccessToken);
    })
);
```

⁹Demystifying Microsoft Graph - <https://www.codeproject.com/Tips/5249834/Demystifying-Microsoft-Graph>

```

    return Task.FromResult(0);
});
return graphServiceClient;
}

```

De tweede gebruikt een techniek beschreven op CSharpCorner¹⁰:

```

// 
public static async Task<GraphServiceClient> GetGraphServiceClient2()
{
    var authentication = new
    {
        Authority = "https://graph.microsoft.com/",
        Directory = WebConfigurationManager.AppSettings["ida:TenantId"],
        Application = WebConfigurationManager.AppSettings["ida:ClientId"],
        ClientSecret = WebConfigurationManager.AppSettings["ida:ClientSecret"],
        GraphResourceEndPoint = "v1.0",
        Instance = WebConfigurationManager.AppSettings["ida:AADInstance"],
        Domain = WebConfigurationManager.AppSettings["ida:Domain"]
    };
    var graphAPIEndpoint =
        $"{authentication.Authority}{authentication.GraphResourceEndPoint}";
    var newAuth = $"{authentication.Instance}{authentication.Directory}";
    // var newAuth2 = $"{authentication.Instance}{authentication.Domain}";

    AuthenticationContext authenticationContext = new
        AuthenticationContext(newAuth);
    Microsoft.IdentityModel.Clients.ActiveDirectory.ClientCredential
        clientCred
    = new Microsoft.IdentityModel.Clients.ActiveDirectory
        .ClientCredential(authentication.Application,
            authentication.ClientSecret);
    Microsoft.IdentityModel.Clients.ActiveDirectory.AuthenticationResult
        authenticationResult
    = await
        authenticationContext.AcquireTokenAsync(authentication.Authority,
            clientCred);
    var token = authenticationResult.AccessToken;
    var delegateAuthProvider = new
        DelegateAuthenticationProvider((requestMessage) =>{
            requestMessage.Headers.Authorization = new
                AuthenticationHeaderValue("bearer", token.ToString());
            return Task.FromResult(0);
        });

    var graphClient = new GraphServiceClient(graphAPIEndpoint,
        delegateAuthProvider);
    return graphClient;
}

```

Het verschil tussen beide zit in de dependencies die gebruikt worden om een instantie van 'GraphServiceClient' te bouwen. De stappen die ze uitvoeren zijn echter hetzelfde en

¹⁰Integrate Microsoft Graph With .NET CORE Web APIs - <https://www.c-sharpcorner.com/blogs/integrate-microsoft-graph-with-net-core-web-apis>

kunnen als volgt beschreven worden:

1. Nodige variabelen declareren voor onder andere de ClientId, TenantId, ClientSecret.
2. De app/authenticatie(context) bouwen aan de hand van deze gegevens.
3. Een token genereren.
4. Een instantie van GraphServiceClient aanmaken en returnen.

Deze instantie wordt teruggegeven naar een Controller klasse (hier FilterController) waar de HTTP-operaties in gedeclareerd worden. In onderstaand voorbeeld worden alle aanwezige items in OneDrive teruggegeven:

```
// GET api/values/5
public async Task<string> Get(string filter)
{
    try
    {
        GraphServiceClient client = await
            MicrosoftGraphClient.GetGraphServiceClient2();
        var resultaat = await
            client.Users["db4fef52-9274-49e2-846c-1f325c4b9d7c"].Drive.Root
            .Children.Request().GetAsync();

        return resultaat.ToString();
    }
    catch (MsalUiRequiredException)
    {
        //
```

Als bovenstaande test query uitgevoerd wordt via de Swagger UI wordt er een fout teruggegeven. (Zie Figuur 4.14) Het probleem is dat het hier aangemaakte soort token applicatie permissies in plaats van gedelegeerde permissies nodig heeft. Voor gedelegeerde permissies is een ingelogde gebruiker nodig. Applicatie permissies zijn in Azure enkel in te stellen met een Administrator account.

Het sjabloon¹¹ uit de officiële Graph documentatie gebruikt deze soort token.(Microsoft Docs, 2020d) Dit is echter niet mogelijk voor een API die in Power Apps opgeroepen moet worden. Er is een andere oplossing nodig.

4.5.3 Blank Custom Connector

Er kan aangenomen worden dat als de Custom Connector in Power Apps zelf gemaakt wordt (via de optie 'Create from blank') en de nodige Microsoft Graph requests rechtstreeks gedeclareerd worden, dat de permissies van de ingelogde PowerApps gebruiker genomen worden om deze later uit te voeren, dat het ingevoegde soort token met andere woorden gedelegeerde permissies zal hebben.

Vergeleken met hoe het in de vorige sectie ging moeten de stappen in Azure manueel

¹¹GitHub GraphSDK - <https://github.com/rogreen/GraphSDK/blob/master/GraphSDKDemo/Views/MainPage.xaml.cs>

The screenshot shows the Swagger UI for a custom connector. At the top, there's a navigation bar with 'swagger' on the left, a URL input field containing 'https://localhost:44354/swagger/docs/v1', a 'api_key' input field, and a 'Explore' button on the right. Below this, the title 'PA_Custom_Connector_3' is displayed. A 'Filter' section is open, showing a GET request for '/api/Filter'. In the 'Parameters' table, there is one entry: 'filter' with a value of 'test', categorized as 'query' and 'string'. Below the table are buttons for 'Try it out!' and 'Hide Response'. Under the 'Curl' section, a command is provided: 'curl -X GET --header 'Accept: application/json' 'https://localhost:44354/api/Filter?filter=test''. The 'Request URL' section shows the full URL 'https://localhost:44354/api/Filter?filter=test'. The 'Response Body' section displays the following JSON error message:

```
{
  "Message": "An error has occurred.",
  "ExceptionMessage": "Code: Authorization_RequestDenied\r\nMessage: Insufficient privileges to complete the operation.\r\nInne",
  "ExceptionType": "Microsoft.Graph.ServiceException",
  "StackTrace": "   bij Microsoft.Graph.HttpProvider.<SendAsync>d__18.MoveNext()\r\n--- Einde van stacktracing vanaf vorige l
}
```

Figuur 4.14: Teruggegeven error bij het maken van requests.

uitgevoerd worden.

1. Nieuwe app registratie maken.
2. Een Client secret aanmaken.
3. Scope permissies instellen, dit was: User.Read Files .Read Files .Read.All Files .ReadWrite Files .ReadWrite.All

Als nu de optie 'Create from blank' geselecteerd wordt start een wizard:

General → Security → Definition → Test

In de 'Security' tab is het belangrijk OAuth v2.0 authenticatie te kiezen met als id provider 'Azure Active Directory'. Dan komt het neer op het invullen van de gegevens die net in Azure aangemaakt zijn. Het echte werk begint in de 'Definitie' tab. De operaties uit Subsectie 4.5.1 werden eerst in Microsoft Graph Explorer¹² getest, hierna worden de request en response gegevens gebruikt om een actie te maken. Deze actie kan licht aangepast worden wat variabelen gebruik betreft zodat bijvoorbeeld de filterargumenten uit PowerApps correct kunnen doorgegeven worden naar deze connector.

¹²<https://developer.microsoft.com/en-us/graph/graph-explorer>

Het is handig dat in de laatste stap de acties getest kunnen worden. De resultaten hiervan zijn ingevoegd. (Figuur 4.15)



Figuur 4.15: Resultaat van de Graph query's.

In de PowerApp worden deze acties opgeroepen via formules gekoppeld aan de 'OnSelect' property van de 'Filter !' knop.

```
Set(filterSessionID ; MSGraphConnector.CreateSession({ persistChanges : true
    }) . id ) ;;
MSGraphConnector.ClearFilters() ;;
ForAll(filterVelden ; MSGraphConnector.FilterApply(veld.Value ;
    filterSessionID ; { filterOn : "custom" ; criterion1 : gelijkTeken.Value &
    filterTekst })) ;;
Set(filterResultaatItems ; MSGraphConnector.GetResult({ '$select' : " values
    " }) . value );;
```

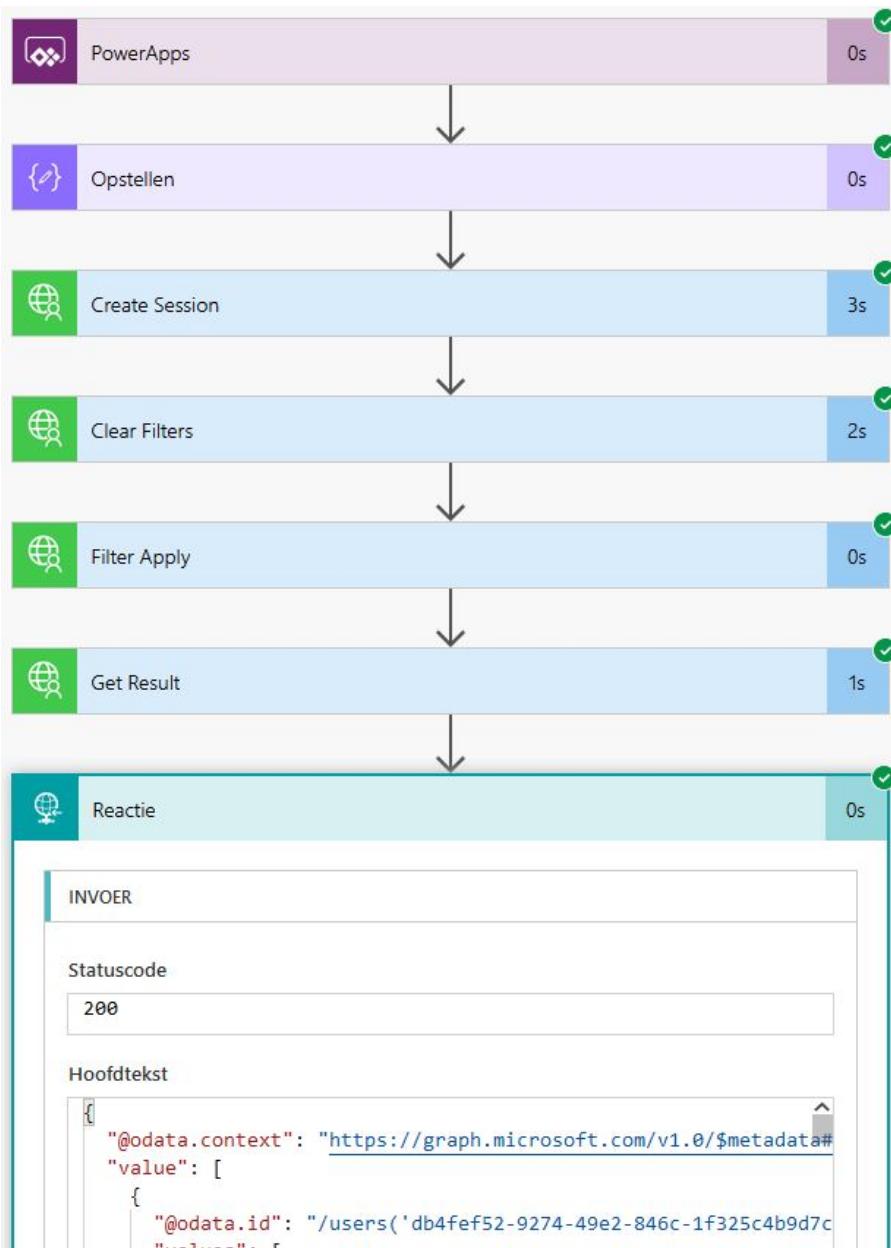
1. Een Excel workbook sessie starten en de gereturnde id koppelen aan een variabele.
2. Vorige filters verwijderen.
3. Voor elke rij in de 'filterVelden' tabel wordt de filter actie oproepen. De aanvaarde argumenten in volgorde zijn: kolomnaam, sessie id en een record waarin de stringwaarde van de operator en filterwaarde aan elkaar geplakt worden (via '&').
4. Er wordt een variable gedeclareerd dat het relevant stuk return json toegewezen krijgt via de 'GetResult' actie.

Theoretisch ziet dit er goed uit, de tests ervoor slagen en de eerste actie geeft de sessie id succesvol terug. Jammer genoeg falen de andere. Er wordt telkens een 404 (resource not found) error teruggegeven.

De gevonden workaround is om de oproepen naar de Custom connector uit te laten voeren vanuit een flow. Figuur 4.16 toont de succesvolle uitvoering hiervan.

Enkele opmerkingen die het gebruik van de flow verduidelijken:

In de powerapps wordt de flow uitgevoerd via de 'Filter !' knop. De filterargumenten moeten als invoerparameters meegegeven worden. Deze moeten formatting krijgen. De resultaten van 'GetResult' worden teruggestuurd in 'Reactie' en zijn als json toegankelijk vanuit de PowerApp mits het schema eerst gedefinieerd werd.



Figuur 4.16: Gebruik van de Custom Connector in een flow, de acties van deze connector zijn groen.

5. POC: Outsystems

5.1 Voorbereiding

5.1.1 Van 0 tot de eerste app uitvoering

De eerste stap is om een account aan te maken, hierna kan de Outsystems software geïnstalleerd worden (Service Studio als gewone IDE en Extension Studio om uitbreidings te maken).

In de persoonlijke omgeving kan naast het maken van een nieuwe app community componenten van de Forge ook rechtstreeks geïnstalleerd worden.

Bij het maken van een app zijn er vijf keuzes: reactive of traditionele web app, tablet of telefoon app en ten laatste ook een service. Er is voor reactive gekozen omdat er in de POC in PowerApps twee versies gemaakt zijn en de app dus op meerdere platformen moet kunnen gebruikt worden. Na de creatie van de app wordt ook een module aangemaakt.

De eenvoudigste manier om data te introduceren is door een Excel file op te laden. De data hieruit wordt naar een cloud instantie van SQL Server opgeslagen.

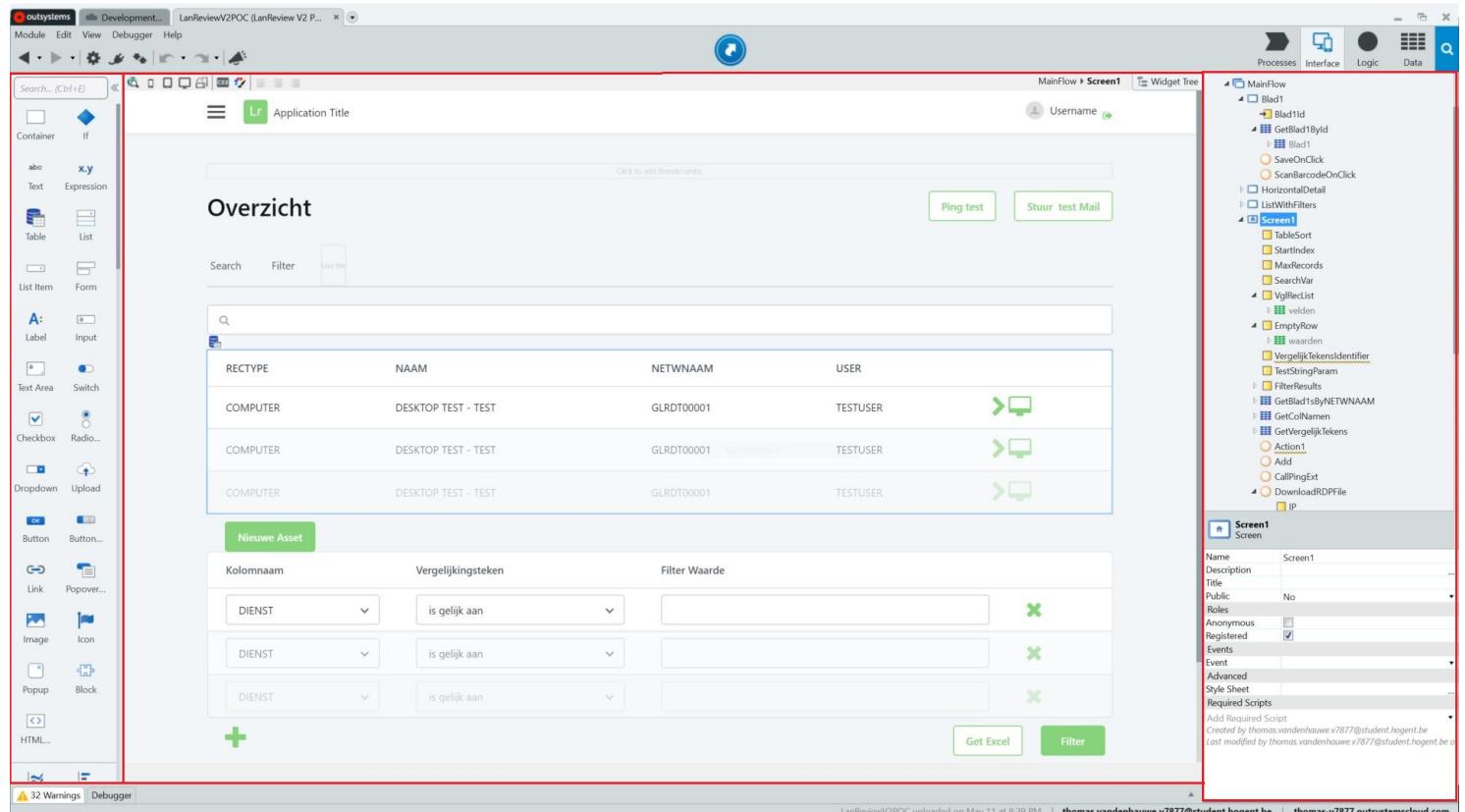
Als de app gestart wordt moeten Outsystems credentials opgegeven worden.

Dit is de basis om de requirements uit te kunnen beginnen werken maar eerst nog wat uitleg bij het gebruik van de IDE.

5.1.2 IDE en begrippen

Er zijn een aantal functies aanwezig, hier uitgelegd per mogelijk weergavepaneel (Zie Figuur 5.1):

- **Centraal:** Hier kunnen de UI flows (onderlinge verhouding van de schermen), ontwerpweergave van een scherm of een actie/methode weergave staan.
- **Links:** Een overzicht van de beschikbare UI controls of actie stappen.
- **Rechts:** Context views voor de processen, interface, de logica en de data. Aangevuld met een properties paneel wanneer toepasselijk.



Figuur 5.1: Overzicht van de Outsystems Service Studio.

Er zijn een aantal begrippen die gekend moeten zijn bij het maken van applicaties.

- **TrueChange Debugger:**

Zoals verwacht van een debugger kunnen breakpoints gezet worden en zijn tijdens uitvoering de waarden van de actieve variabelen in te kijken. Het interessante is dat hiernaast de app geanalyseerd wordt en performantie en security aanbevelingen teruggegeven worden.

- **One Click Publish:**

Stappen uit traditionele software ontwikkeling om een app beschikbaar te stellen (verifiëren, compileren, uploaden en publiceren van code) worden geautomatiseerd met één knop. Als de verificatie faalt moeten de fouten eerst uit de app gehaald worden. Dit betekent ook dat wijzigingen altijd online worden gezet voor het resultaat bekijken kan worden.

- **AI Assisted development:**

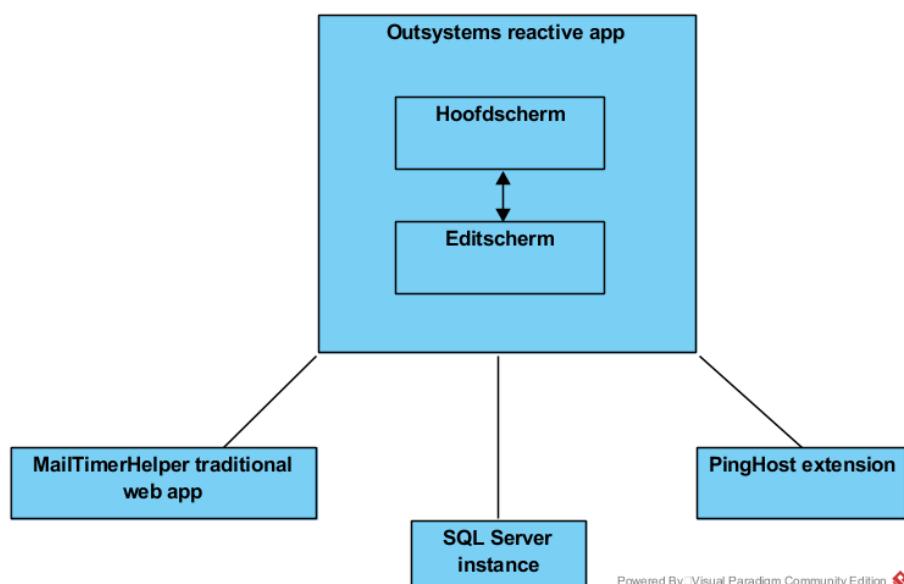
Tijdens ontwikkeling wordt AI gebruikt om onder andere sleutelwoorden en app context te analyseren en op basis hiervan acties aan te bevelen. Een voorbeeld van het gebruik ervan in de POC was dat wanneer een actie aangemaakt werd in de lijst

op het hoofdscherm er voorspeld werd dat er een detailweergave nodig was. Dit detailscherm en bijbehorend formulier werd automatisch gegenereerd.

- Aggregate VS SQL Query:

Er zijn twee manieren om data te queryen. Een aggregate is eenvoudig te declareren en beter geoptimaliseerd in Outsystems dan een gewone SQL query. Die is echter flexibeler en het gebruik ervan wordt aangeraden wanneer iets niet mogelijk is met een aggregate. Beiden werden gebruikt in de POC en komen verder aan bod.

5.2 Model van Opstelling



Figuur 5.2: Model van de Outsystems proof-of-concept.

5.3 Requirements

5.3.1 Must have

Prijs

De Personal Environment¹ (gratis versie) legt geen beperkingen op de bruikbare functionaliteit of hoe lang deze omgeving beschikbaar is. Dit betekent echter niet dat er geen limieten zijn. De hamvraag is of deze limieten aanvaardbaar zijn voor het gebruiksscenario van de POC. Ze worden daarom één per één afgetoetst:

- Maar één ontwikkelaar in de omgeving, er is geen samenwerking mogelijk.

¹What's an OutSystems personal environment? - https://success.outsystems.com/Support/Personal_Environment/What's_an_OutSystems_personal_environment

- De app is niet zodanig complex dat meerdere personen nodig zijn om te ondersteunen of verder uit te bouwen.
- Database capaciteit is gelimiteerd tot 2GB.
 - De bij te houden informatie over de meer dan 1000 assets valt hier nog steeds ruim onder.
- Indien de app een maand niet gebruikt wordt gaat de omgeving 'in slaap' en moet de app opnieuw gegenereerd worden voor deze opnieuw gebruikt kan worden.
 - De app zal dagelijks gebruikt worden dus dit zal nooit een risico worden.
- Cloud resources worden gedeeld met andere gebruikers, er is geen service garantie.
 - De app eist geen intens resource gebruik.
- Er is geen harde limiet op het aantal toegestane app gebruikers maar als de vorige items in beschouwing genomen worden zal een app bruikbaar zijn tot maximum 100 personen.
 - Maximum zes personen zullen de app gebruiken.

De marktleidende low-code platformen gefocust op professionele ontwikkelaars zijn duurder dan de concurrentie. Mendix en Outsystems rekenen meer dan €2000 voor hun basisplan (Zie Tabel 3.5 en 3.7). Het was daarom belangrijk dat de POC binnen de limieten bleef, de meerprijs zou niet worden aanvaard.

Overzicht kunnen geven van belangrijkste info voor elk toestel in het netwerk

Dit wordt gedaan via een hoofdscherm met een overzicht van de assets en een detail/edit-scherm dat meer informatie toont (Zie Figuur 5.3 en 5.4). Waar dit bij PowerApps gegenereerd kan worden uit data kunnen dezelfde resultaten eenvoudig bekomen worden met behulp van AI geassisteerde acties.

1. Een nieuw scherm aanmaken en kiezen voor een lijstweergave sjabloon.
2. De data slepen naar het tabel element volstaat na wat kolomaanpassingen om een gewenste weergave van de data te krijgen. Achter de schermen wordt een aggregate aangemaakt die de rijen ophaalt uit de databank.
3. Als een knop aangemaakt wordt in de rij zal als actie aanbevolen worden dat er een nieuw scherm wordt aangemaakt met een detailweergave. In de achtergrond wordt de id van het rij item meegegeven om in het detailscherm een aggregate in te stellen die deze id gebruikt om alle data van het item op te halen en te tonen.
4. In het detailscherm kan de layout aangepast worden naar wens en kan validatie ingesteld worden.

Dit instellen van validatie is belangrijk. Omdat de data recent werd geïmporteerd naar de SQL instantie zijn er geen specifieke regels op ingesteld. Als een externe databron gebruikt werd (bijvoorbeeld SharePoint) had dit wel het geval kunnen zijn.

Outsystems heeft automatische client side validatie. (OutSystems Docs, g.d.) In de formulier widget heeft elk inputveld geassocieerd met een kolom een nodig inputtype (zoals mail, datum, tekst) toegewezen gekregen, elke input is standaard ook verplicht (via het 'mandatory' property). Het formulier zelf heeft een Valid boolean property dat op false wordt gezet als validatie in één van de inputs zou falen. Controle hierop gebeurt in de

The screenshot shows a web-based application titled 'LanReviewV2POC'. At the top right, there is a user profile for 'Thomas Hauwe'. Below the title, the word 'Overzicht' is displayed. On the left, there are 'Search' and 'Filter' buttons. A search bar contains a magnifying glass icon. To the right of the search bar are two green buttons: 'Ping test' and 'Stuur test Mail'. The main area is a table with the following data:

RECTYPE	NAAM	NETWNAAM	USER	
COMPUTER	DESKTOP TEST - TEST	GLRDT00001	Thomas	
COMPUTER	DESKTOP TEST - TEST	GLRDT00002	TESTUSER	
COMPUTER	DESKTOP TEST - TEST	GLRDT00003	TESTUSER	
		GLRDT00004		
		GLRDT00004		
		GLRDT00001		

At the bottom left of the table area is a green button labeled 'Nieuwe Asset'. Below the table, a small note says '1 to 6 of 6 items'.

Figuur 5.3: Hoofdscherm Outsystems POC

client actie gekoppeld aan de knop om toe te voegen of te wijzigen. Het is deze actie die gewijzigd moet worden volgens het specifieke scenario. (Zie Figuur 5.5)

Gaat het om het toevoegen van een nieuwe entry of wijzigen van een bestaande? Indien het een nieuwe is wordt gekeken of het ingevulde Netwnaam uniek is via de server actie 'NetwNaamUnique'. Deze zal een aggregate uitvoeren waarin een filter toegepast is die kijkt of de meegegeven Netwnaam matcht aan een waarde uit de NETWNAAM kolom. Indien de netwerknaam uniek is wordt verdergegaan met de formuliervalidatie. Indien niet wordt een error message gekoppeld aan de textinput van de Netwnaam en stopt de actie.

Filtering / rapportage

Filter

Zoeken en filteren is in tabbladen gescheiden en ook hun resultaten worden in aparte Table controls weergegeven.

→ Search:

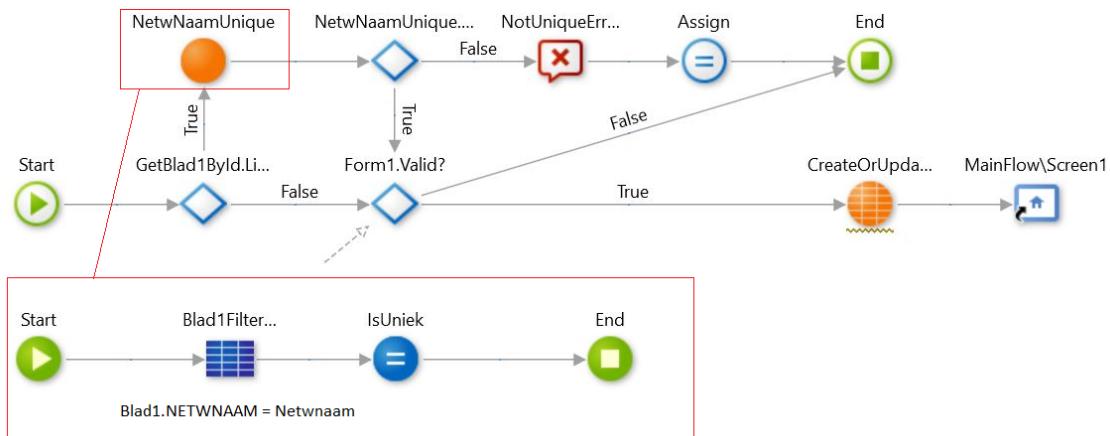
Voorbereidend wordt er een filter ingesteld in de reeds bestaande aggregate (die items ophaalt om weer te geven in het hoofdscherm) die een zoekvariabele zal matchen aan elke kolom. Er wordt een zoekveld toegevoegd aan het scherm en deze zoekvariabele wordt eraan gekoppeld. Als OnChange event wordt een client actie uitgevoerd die de aggregate ververst (en de filter opnieuw uitvoert met de gewijzigde zoekvariabele).

LanReviewV2POC Thomas Hauwe

Detailweergave

RECTYPE	INST		
COMPUTER	AZ		
NAAM	DIENST		
DESKTOP TEST - TEST	HELPDESK		
NETWNAAM *	LOKATIE		
GLRDT00001	HELPDESK		
USER	OS		
Thomas	WINDOWS 10		
IP	SP		
192.168.1.1	2		
SOORTIP	TYPEMERK		
DHCP GERESERVEERD	HP COOLBOOK		
MAC *	Scan MAC	SERIAL *	Scan Serial
08.00.27.8F.43.60		SP44RG5F77	
PATCH	OPM		
N001	eerste testentry		
SWITCH	KLEUR		
KAST A SWITCH 1	BLAUW		
POORT			
1			
Save			

Figuur 5.4: Detailscherm Outsystems POC



Figuur 5.5: Uitbreiding van de standaard formuliervalidatie.

```
| Blad1.NETWNAAM like "%" + SearchVar + "%" or Blad1.RECTYPE like "%" +
|   SearchVar + "%" or ... or Blad1.KLEUR like "%" + SearchVar + "%"
```

→ Filter:

- **Vorm en datatypen:**

De filter bouwen gebeurt via een dynamisch groeiende lijst waarbij elke rij bestaat uit een keuzeveld voor de kolom, de soort vergelijking en de tekst om op te filteren. Er moeten enkele datatypen voorbereid worden om dit mogelijk te maken:

- Statische entiteit die alle kolomnamen bevat.
- Statische entiteit die alle mogelijke vergelijkingen/operators bevat.
- Structuur die de filterwaarden van een rij zal vasthouden. Dit is een samengesteld type, het bevat:
 - * Identifier van de entiteit met kolomnamen.
 - * Identifier van de entiteit met operators.
 - * Een Text variabele met de filterwaarde.

In de interface moeten nog twee variabelen aangemaakt worden: een Record List van de 'FilterVelden' (zal elke filterwaarde bevatten) en een Record van 'FilterVelden' (stelt een lege rij voor). Ten laatste moet wat hierboven aangemaakt is correct toegewezen worden aan de properties van de dropdowns.(Zie Figuur 5.6) Het Advanced Filter Builder² Forge component werd gebruikt om hier wijs uit te geraken.

- **Query:**

De 'Filter' knop voert de 'FilterOnClick' client actie uit die op zijn beurt de 'Filter' server actie uitvoert. De Record List wordt als argument meegegeven.

De 'Filter' actie (Zie Figuur 5.7):

1. Controleren of de Record List leeg is. Indien ja, stopt de uitvoering.
2. De eerste entry naar een string converteren.

```
| GetColNamen( VglRecList [0]. velden . ColNaamAtt ) . ColNamen . Naam +
|   " " + GetVergelijkTekens( VglRecList [0]. velden .
|     VergelijkTekenAtt ) . VergelijkTekens . Teken + " ' " +
|       VglRecList [0]. velden . FilterWaarde + " '' "
```

3. Controleren of er meerdere records zijn. Indien ja, wordt er gelooped en wordt de string verder aangevuld via onderstaande expressie:

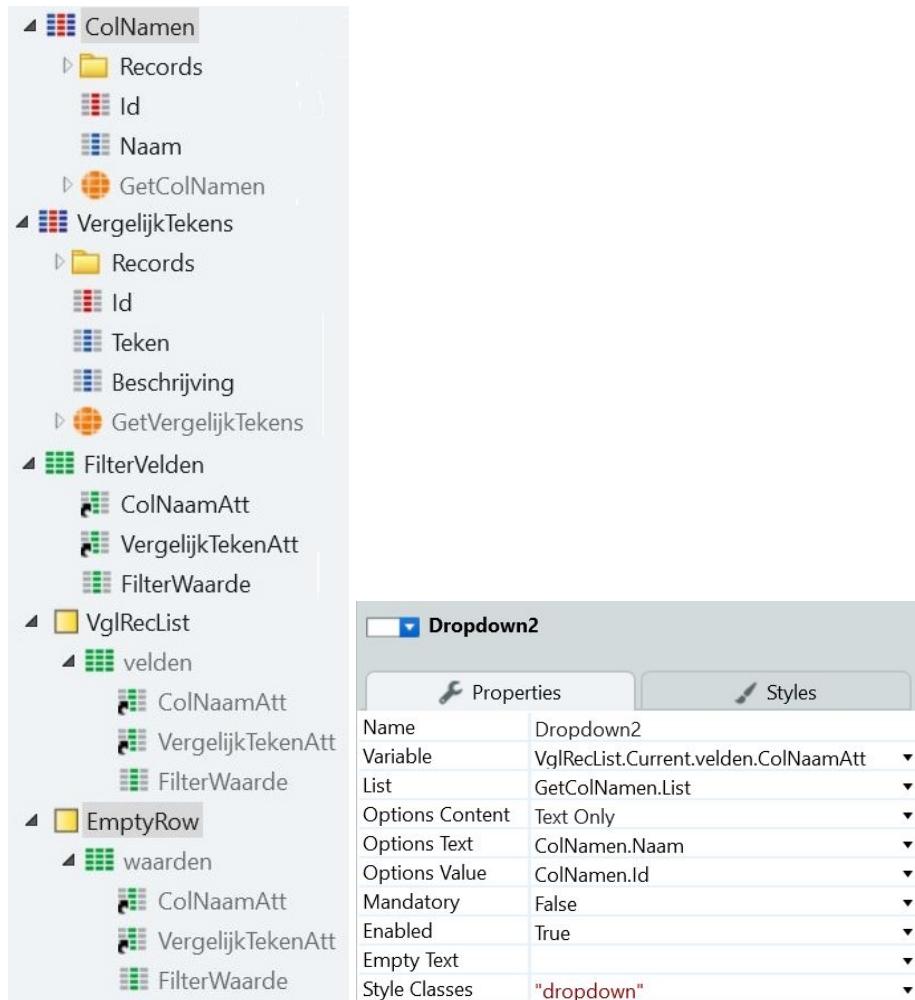
```
| TestStringParam + " and " + GetColNamen( VglRecList . Current .
|   velden . ColNaamAtt ) . ColNamen . Naam + " " +
|     GetVergelijkTekens( VglRecList . Current . velden .
|       VergelijkTekenAtt ) . VergelijkTekens . Teken + " ' " +
|         VglRecList . Current . velden . FilterWaarde + " '' "
```

4. De SQL query uitvoeren. Een opmerking hierbij is dat de 'expand inline' optie actief moet staan om de gebouwde string effectief te kunnen gebruiken als argument. De query zelf is een simpele WHERE met door AND aaneengeschakelde evaluaties. (Figuur 5.8)

De gereturnde filterresultaten worden in een aparte Table geplaatst. Dit maakt het exporteren als report later makkelijker.

²<https://www.outsystems.com/forge/component-overview/365/advanced-filter-builder>

De gevonden beperkingen zijn dat het evalueren met sommige kolommen niet juist werkt en dat de performantie lager is dan bij een aggregate.



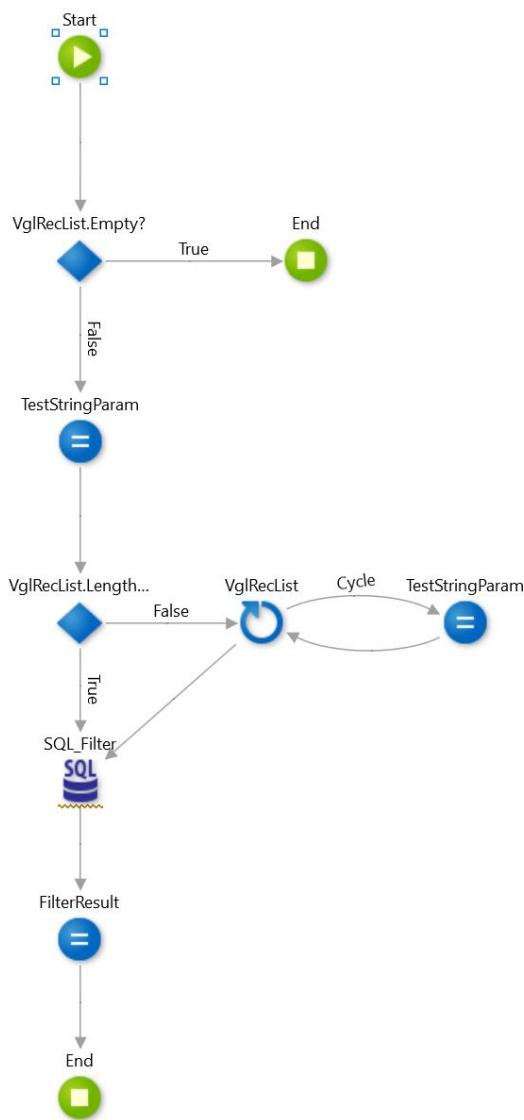
- (a) Overzicht van de datastructuren en variabelen. (b) Hoe de data aan de dropdown met kolommen gekoppeld wordt.

Figuur 5.6: Overzicht van de filtering voorbereidingen.

Reporting

Downloadacties zijn goed ondersteund. Door een download control als eindpunt van een actie te zetten wordt de geconfigureerde inhoud ervan aangeboden in een download prompt. De specifiek geconfigureerde acties om dit te bereiken worden opgeroepen wanneer er een gewenste dataset gefilterd staat en op de 'Get Excel' knop gedrukt werd.

- GetExcel (Client actie):** Het filterresultaat naar de server actie doorgeven en de gereturnde Excel file te downloaden zetten.
- PrepareExcel (Server actie):** De filterresultaten converteren naar een Excel bestand via de 'Record List to Excel' stap.



Figuur 5.7: De Filter actie.

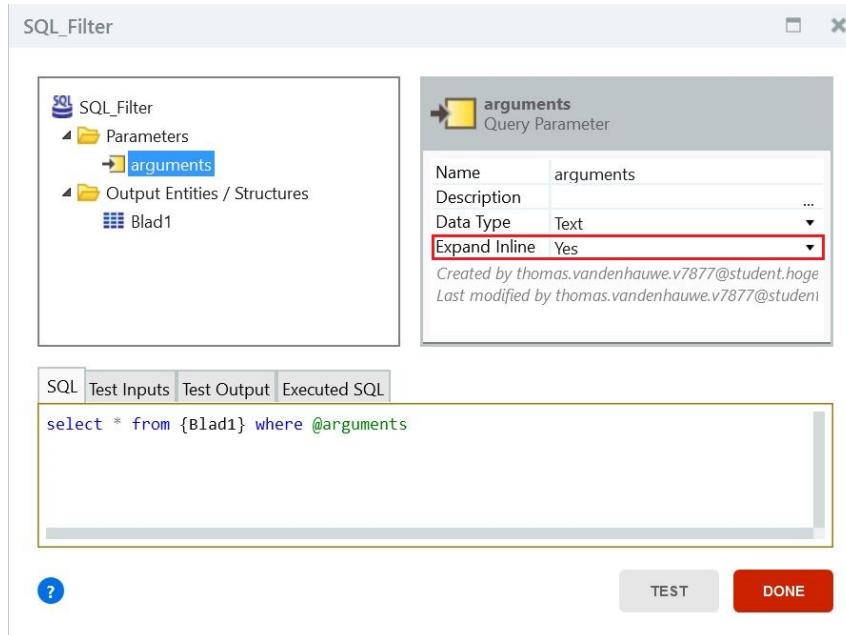
RDP / Ping

RDP (Remote Desktop Protocol)

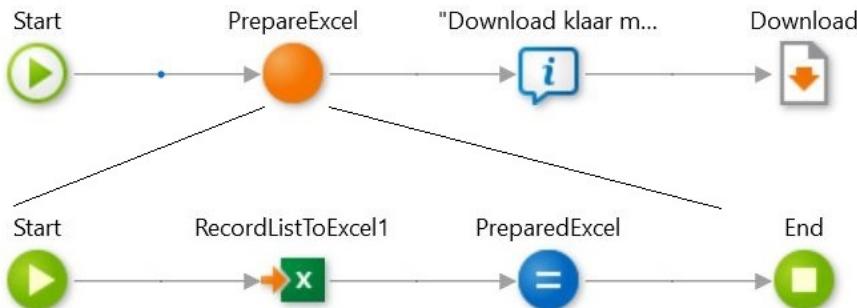
Omdat de POC in de cloud draait is deze functionaliteit niet direct op te roepen, mogelijke workarounds:

- **Workaround 1:** Client side javascript, specifiek een ActiveX plugin gebruiken om shell commando's uit te voeren³.
→ 'ActiveX not defined' error. Dit is enkel mogelijk met oudere versies van Internet Explorer. Het is ook logisch dat dit niet meer ondersteund is vanuit security aspect.

³How to Run cmd.exe with parameters from javascript - <https://stackoverflow.com/questions/10186813/how-to-run-cmd-exe-with-parameters-from-javascript>



Figuur 5.8: De SQL query.

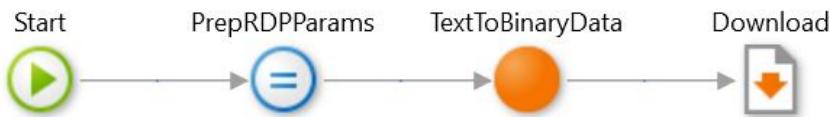


Figuur 5.9: De client en server actie om een rapport te downloaden.

- **Workaround 2:** Een 'link' control en de href property laten wijzen naar een lokale RDP file.
→ Niet flexibel genoeg: de rdp config file zou elke keer aangepast moeten worden met het relevante IP-adres.
- **Workaround 3:** De webbrowser waar de POC in werkt configureren om custom urls te kunnen gebruiken (zoals rdp://). Vanuit de app naar dat soort url navigeren via de ingebouwde 'RedirectToUrl' functie.
→ Te veel extra configuratie buiten Outsystems en twijfelachtige ondersteuning.
- **Workaround 4:** Een RPD file maken voor de gewenste pc en downloaden/uitvoeren. Dit past delen toe zoals gezien in het RDP File Creator⁴ Forge component. Er is ook een dependency nodig om text naar binaries te converteren.
Er worden drie gegevensstukken gecombineerd: IP-adres van de doel-pc, naam van de bevoegde rdp gebruiker en het sjabloon van een rdp configuratiebestand.

⁴https://www.outsystems.com/forge/component_overview.aspx?projectid=574&projectname=rdp-file-creator&

In de client methode wordt het sjabloon ingevuld met de eerste twee waarden en hierna geconverteerd naar een binary. Ten laatste wordt dit bestand als download aangeboden. In de meeste webbrowsers is het als voorkeur in te stellen om bepaalde bestandstypen direct te openen na download. Dat maakt het tijdsverlies bij elke uitvoering miniem.



(a) Client actie om een RDP file te genereren.

```

disable menu anims:i:1
disable themes:i:0
disable cursor setting:i:0
bitmapcachepersistable:i:1
full address:s:" + IP + "
audiomode:i:0
redirectprinters:i:0
redirectcomports:i:0
redirectsmartcards:i:1
... . . .
  
```

(b) Deel van de RDP configuratie waar het IP-adres en gebruikersnaam ingevuld worden.

Ping

De basis van het probleem bij het uitwerken van RDP geldt ook hier. Toestellen zijn enkel benaderbaar in het lokale netwerk, de ping moet dus uitgevoerd kunnen worden aan de client kant. Outsystems maakt uitvoeren van client side javascript mogelijk maar er werd geen oplossing gevonden om Ping op deze manier te laten slagen.

Plan B is om een extensie te maken. In de persoonlijke omgeving moet dit in .NET en hier is het onder andere mogelijk om de Pinger⁵ klasse te implementeren. Een grote opmerking hierbij is dat de app on-premises gehost moet worden om dit echt te laten werken.

Stappen om de extensie aan te maken:

1. Een nieuwe extensie aanmaken in Integration Studio.
2. De actie 'PingHost' declareren. Er zijn twee parameters nodig:
→ ipaddress (Text)
← pingable (Boolean)
3. Visual Studio openen met de 'Edit Source Code .NET' knop.
4. Een methode schrijven die de Pinger klasse implementeert (Zie Listing 5.1).

⁵ ASP.NET C# Helper Class to execute a ICMPv4 PING to any remote host using hostname or IP Address
- <https://www.ryadel.com/en/asp-net-c-sharp-icmpv4-ping-remote-host-hostname-ip-address/>

5. Net als in de Service Studio One Click-Publish gebruiken om de extensie op te slaan en beschikbaar te maken.
6. De extensie als dependency toevoegen aan de POC maakt de PingHost actie bruikbaar.

Deze eenvoudige methode maakt een instantie van de Pinger klasse, stuurt de ping naar het IP-adres van de invoerparameter en evalueert het resultaat op succes om dit als boolean uitvoer terug te sturen.

```
public void MssPingHost( string ssipaddress , out bool sspingable ) {
    sspingable = false ;
    using ( Ping pinger = new Ping() )
    {
        try
        {
            PingReply reply = pinger.Send(ssipaddress) ;
            sspingable = reply.Status == IPStatus.Success ;
        }
        catch ( PingException e )
        {
            sspingable = false ;
        }
    }
}
```

Listing 5.1: Pinger implementatie

Mobiel bruikbaar zijn

Het gekozen soort app is reactive om zowel op desktop als mobiele layout bruikbaar te zijn maar er werd niet verwacht dat er bij deze keuze (tussen Traditional Web, Reactive en Mobile) functionaliteit verloren zou gaan. Dit was wel het geval:

- **Automatisatie:** Het maken van een Timer die mail verstuur is enkel mogelijk in een Traditional Web App, er was een workaround nodig (Zie Subsectie 5.3.2).
- **Barcode Scanning:** Een Mobile App is nodig om de Barcode Scanner te kunnen gebruiken (Zie Subsectie 5.3.3).

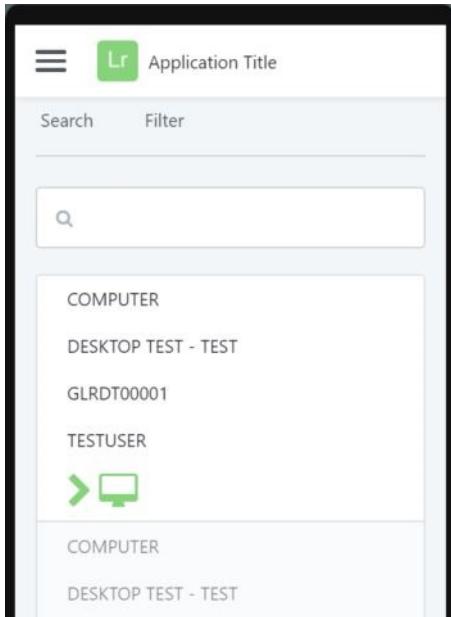
Future proof zijn

Een indicator is de marktpositie. Zowel Gartner als Forrester bestempelen Outsystems als leider (Zie Tabel 2.2). Dit spreekt positief over de bestaanszekerheid.

Ook hun traject over de jaren geeft vertrouwen. Outsystems is opgericht in 2005 en sinds dan is het product en de licentiëring ervan consistent gebleven.

Hierbovenop komt dat er steeds vernieuwingen worden geïntroduceerd⁶.

⁶Product Releases and Updates - <https://www.outsystems.com/whats-new/>



Figuur 5.11: Voorbeeld van de POC met een mobiele layout.

Performant zijn

Een app in de cloud is meestal niet zo snel als een lokale applicatie. Zeker ophalen van data over het internet heeft een invloed op performantie.

Hoe weegt deze POC dan op tegen de PowerApps POC?

In Subsectie 5.3.1 wordt een beeld gevormd van de beperkingen. Vooral de punten over gedeelde resources en 'in slaap gaan' van de omgeving zijn relevant ten opzichte van performantie. PowerApps kent deze beperkingen niet. Dit is natuurlijk te verwachten als een gratis plan (Outsystems) vergeleken wordt met een betalend plan (PowerApps).

Er kan dus besloten worden dat PowerApps consistentere performantie zal bieden.

Security

De POC is toegankelijk met geldige Outsystems credentials. Alternatieve user management (bijvoorbeeld Azure Active Directory) is mogelijk maar niet in de gratis versie en in geval van Azure moet de App on premises gehost worden.

5.3.2 Should have

Gerichte/basis taken kunnen automatiseren

Net als bij de PowerApps POC bestaat de case uit het wekelijks genereren en uitsturen van een rapport met toestellen die langer dan een maand 'te schrappen' staan.

Voor herhalende taken zijn Timers in te stellen. Het probleem is dat het in Traditional Web apps mogelijk is om mails te versturen in acties maar dat deze functionaliteit voor Reactive Web apps (nog) niet ondersteund is. Als workaround is gekozen om een aparte Traditional

Web app te maken die het uitvoeren van deze stappen overneemt.

Voorbereidend moet de omgeving in Service Center geconfigureerd worden om e-mail te kunnen versturen. Dit gebeurt in Administration > Email. Meest notabel zijn de te gebruiken SMTP Server en standaard afzenderadres. Het is ook mogelijk om voor testscenario's al de mails te redirecten naar een testlijst.

Eens een nieuwe Traditional Web app is aangemaakt in de Service Studio volstaat het om:

1. Wat logica betreft: één server actie declareren met één 'Send Email' stap.
2. De mail body configureren: belangrijkst is een Table dat het overzicht van de devices zal bevatten.
3. De asset databank uit de hoofd POC module als dependency declareren en toevoegen aan de Table. Dit maakt er automatisch een aggregate voor aan in een nieuwe Preparation stap. Elke keer de mail opgesteld wordt zal deze Preparation stap eerst uitgevoerd worden.
4. In deze aggregate wordt een filter ingesteld om de juiste toestellen terug te geven.
| AddMonths(testdata.SCHRAPSINDS,1) <= CurrDateTime() and testdata.KLEUR = "BLAUW"
5. Een timer instellen om de actie uit stap één op een tijdschema uit te voeren (Zie Figuur 5.12)

Bruikbaar zijn buiten domein

De POC en databank zijn cloud gehost. Enkel internettoegang en Outsystems credentials zijn nodig om de POC te kunnen gebruiken.

SharePoint requirements

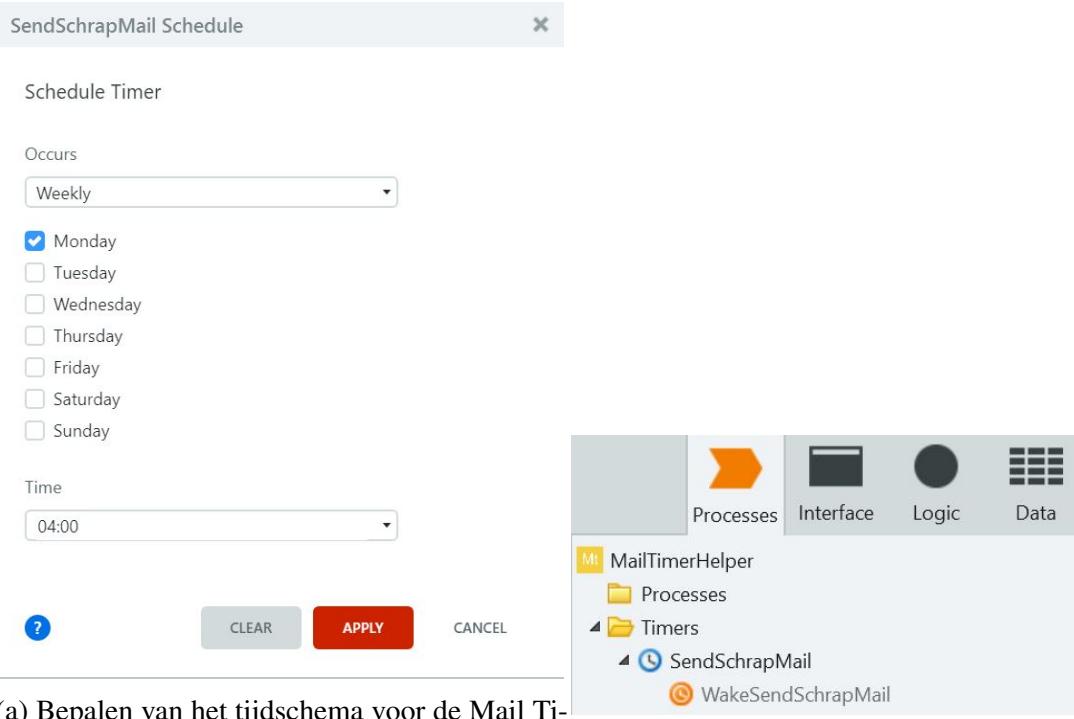
Requirements die met SharePoint te maken hebben zijn:

- Nieuwe types toestellen opnemen.
- Randapparatuur opnemen.
- Revisie van elk dataveld per toesteltype.
- Data opslaan in SharePoint (cloud).

Dit wordt besproken in Sectie 4.2.

Onderstaande aanpak werd onderzocht maar na afloop niet toegepast. Voor het huidige scenario voldoet de standaard voorziene SQL instantie en zijn er geen merkbare voordelen die de extra complexiteit verantwoorden.

Er is geen connector zoals in PowerApps dat rechtstreekse toegang geeft tot de SharePoint data. De verbinding zal met andere woorden via de beschikbare REST api's moeten verlopen. SharePoint heeft zijn eigen '_api' toegang, alternatief is er ook de nieuwere Microsoft Graph. Omdat in de PowerApps POC Graph reeds gebruikt werd (Subsectie 4.5.1) wordt er hier opnieuw voor gekozen. Iets dat daar aan bod kwam was het verschil tussen delegated



(a) Bepalen van het tijdschema voor de Mail Timer.

(b) Mail Timer context overzicht.



Lijst met te schrappen toestellen:		
NETWNAAM	NAAM	DIENST
GLRDT00002	DESKTOP TEST - TEST	HELPDESK
GLRDT00001	DESKTOP TEST - TEST	HELPDESK

(c) Voorbeeld van het resultaat.

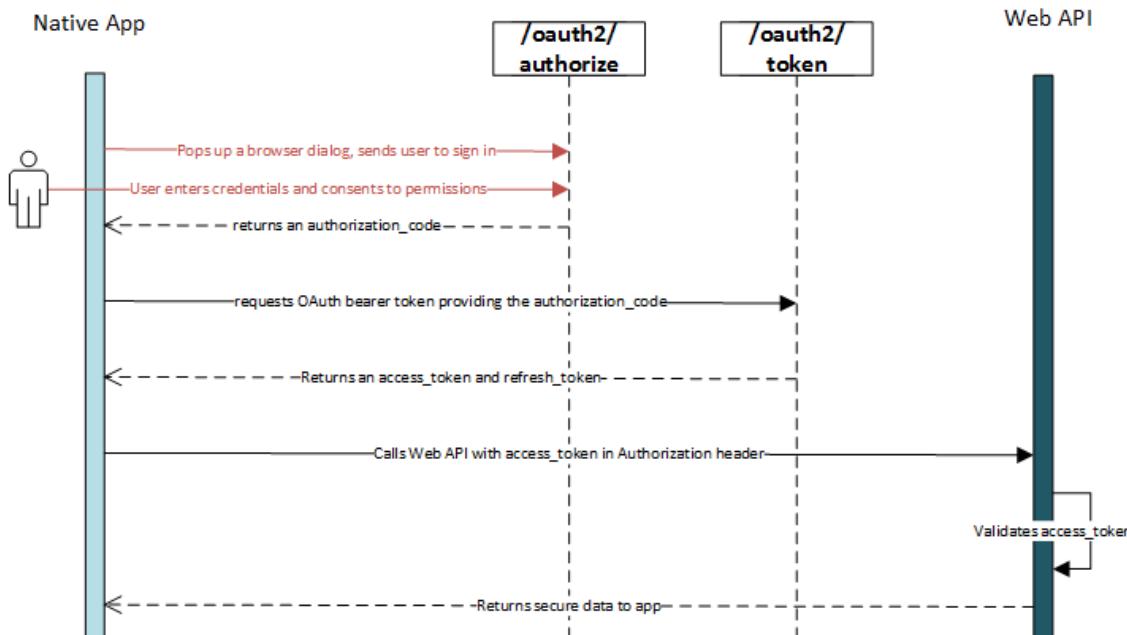
Figuur 5.12: Overzicht van de configuratie en het resultaat van de Mail Timer.

en application permissions. Ook hier wordt uitgegaan van delegated permissions gezien dit de veiligere optie is (Windows login nodig).

Alvorens er data opgevraagd kan worden moet het autorisatieproces (hier OAuth V2) doorlopen worden (Figuur 5.13).

De implementatie ervan in Outsystems gaat als volgt:

1. Azure
 - (a) App registratie.
 - (b) Client secret aanmaken.
 - (c) Scope definiëren. Voor SharePoint is dit: Sites .Manage.All Sites .Read.All SitesReadWrite .All
 - (d) Redirect URL naar de Outsystems app instellen. Structuur is https ://< outsystems



Figuur 5.13: Weergave van het autorisatieproces. (MS Login Connector Team, 2019)

- omgeving>/<app naam>/<afhandeling pagina>
2. Autorisatie request doen (\ authorize eindpunt). Komt overeen met inloggen in Windows account en goedkeuren van de scope.
← Autorisatiecode wordt teruggegeven.
 3. Token aanvragen met de autorisatiecode (\token eindpunt)
← Token wordt teruggegeven.

Dit token moet aanwezig zijn in elke request naar de SharePoint data via Microsoft Graph.

Het is echter mogelijk een deel van deze stappen door te geven naar een Forge component (zoals Microsoft Login Connector Reactive (MS Login Connector Team, 2019)). De implementatie daarvan is een pak eenvoudiger: in de standaard login actie van de POC wordt de methode uit de extensie geplaatst die het autorisatieproces start 'GetOAuth2AuthenticationURL'. Praktisch gezien wordt na de Outsystems login nu ook de Microsoft login uitgevoerd. Na afloop is het token beschikbaar om te gebruiken in REST operaties. Alleen de voorbereiding in Azure moet nog steeds gedaan worden.

Leercurve moet degelijk zijn

(Dit deel is gebaseerd op persoonlijke ervaringen van de auteur)

De algemene leercurve van Outsystems ligt hoger dan bij PowerApps, dit is logisch omdat het om een andere doelgroep gaat.

Als er specifieker gekeken wordt naar de oorzaken van moeilijkheid tussen beide platformen:

- **Outsystems:** Gebruik van data en datatypen kan in sommige scenario's overweldigend zijn.

gend zijn voor een beginner.

- **PowerApps:** Gebruik van formules en vooral sommige technieken die er mee gebruikt moeten worden.

Over hulpmateriaal:

De documentatie is goed en de community is actief. De instructievideo's zijn ok. Deze zijn van Outsystems zelf maar soms verouderd.

Community apps uit de Forge zijn ook handig. Componenten kunnen ingekijken worden specifiek om technieken aan te leren. Veel forge apps zijn als demo bedoeld of hebben een companion demo app.

5.3.3 Nice to have

Samenwerken met SCCM en/of synchroniseren met en data uit de SQL databank kunnen gebruiken

Dit valt buiten de scope van het onderzoek.

Diverse GUI verbeteringen/robuust GUI ontwerp ondersteunen

Om de GUI te bouwen wordt het UI framework genaamd Silk UI⁷ gebruikt. Er is een grote focus op Drag-en-drop.

Een overzicht van Silk UI componenten, die telkens meer controle over detail geven:

Screen templates → thema editor → UI patronen/controls voor gangbare use cases → Style properties → CSS editor

Afhankelijk van hoeveel tijd men wil spenderen is elk aspect van de UI met andere woorden aan te passen.

De gemaakte POC is reactive. Een samenvatting van de ervaringen bij het bouwen van de UI is dat er veel controle is over details maar dat drag-en-drop warrig kan worden als er veel containers/componenten op het scherm zijn.

Barcodes kunnen scannen

Dit is mogelijk via het officiële Forge Component van Outsystems⁸.

Na het importeren van de barcode scanner als dependency is het gebruik ervan analoog met de variant in PowerApps.

1. Een knop toevoegen bij de Text Input horend bij de kolom waar de later ingescande waarde ingevuld zal worden. Dit is in de Form widget op het detailscherm.

⁷<https://silkui.outsystems.com/>

⁸<https://www.outsystems.com/forge/component-overview/1403/barcode-plugin>

2. Een Client actie declareren op het On Click event die 'ScanBarcode' oproept en de 'ScanResult' parameter aan een variabele koppelt.
3. Deze variabele instellen als inhoud van de doel Text Input (MAC of SERIAL).

Bij de uitwerking werden snel problemen vastgesteld. Omdat de POC een Reactive Web app is zijn de nodige Apache Cordova componenten niet beschikbaar. Dit is beperkt tot een Mobile app en de POC zou als dusdanig moeten herschreven worden om de Barcode Scanner effectief te kunnen gebruiken.

AI functionaliteit

Geen enkele van de opgestelde requirements heeft AI nodig. Dit werd niet verder onderzocht.

6. Conclusie

Dit onderzoek werd uitgewerkt om antwoord te kunnen geven op de twee hoofdonderzoeks vragen, deze waren:

- *Is het mogelijk een vervanger voor LanReview te bouwen met Power Apps die elk de vier hoofdfunctionaliteiten ondersteunt en op zijn minst drie vierde van de overige functionaliteiten kan ondersteunen?*
- *Is Power Apps werkelijk de beste keuze hiervoor of is er meerwaarde in een volledig gerealiseerd IT asset management pakket? Is er alternatief een beter geschikt low-code platform?*

Hoewel sommige requirements eenvoudig uit te werken waren (barcode scanning, automatisatie) waren voor anderen workarounds nodig die na uitwerking nog steeds tekort schoten vergeleken met wat de originele LanReview mogelijk was (RDP/ping, filtering). Indien men tijdens de uitwerking van een app workarounds en custom connectors moet beginnen gebruiken is het daarom aangeraden om een andere weg uit te gaan. Door de nodige tijdsinvestering kan evengoed iets opgesteld worden in een traditionele ontwikkelomgeving en het sterkste voordeel van low-code is daarmee tenietgedaan.

Dit wil niet zeggen dat PowerApps geen meerwaarde biedt voor deze case. Voor het maken van een strikt mobiele versie met een beperktere functionaliteitsset is het zeer geschikt. Bijvoorbeeld voor een telefoonboek app zou het ook perfect zijn.

Voor het maken van een complexere business app is Outsystems de beste keuze. De leercurve is niet dermate hoger dan die van PowerApps en indien de app beperkte resources en gebruikersbestand nodig heeft moet bovendien geen betalend plan aangegaan worden. Dit is het geval voor de LanReview POC maar er zijn nog steeds beperkingen mogelijk door de aard van de applicatie (cloud). Als lokale resources nodig zijn (RDP/ping) is dit net

als bij PowerApps een hekelpunt. Het besluit hierbij is dat als AZ Glorieux een complexe app nodig heeft Outsystems een goede optie is maar als vervanger van LanReview is het niet toereikend.

De meest passende oplossing voor het vernieuwen van de desktopversie van LanReview is om de broncode over te zetten naar een WPF of UWP .NET Core desktop applicatie of soortgelijk.

Low-code (PowerApps) heeft wel degelijk een toekomst in AZ Glorieux maar er moet rekening gehouden worden met requirements:

- Eenvoudige requirements → Low-code voor citizen developers (no-code) - **PowerApps**.
- Complexe requirements → Low-code voor ontwikkelaars - **Outsystems**.
- Specifieke requirements, grote integratie met host systeem nodig → Traditionele desktop app.

Laat dit onderzoek daarom een leidraad zijn voor het maken van deze toekomstige keuzes.

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1 Introductie

Het IT Team van AZ Glorieux heeft een oog op de toekomst. Twee grote voorbeelden hiervan zijn een push om nodige toestellen te migreren naar Windows 10 en een geleidelijke adoptie van Intune als aanvulling van System Center Configuration Manager (SCCM). Tijdens mijn stage heb ik hier een bescheiden bijdrage aan kunnen leveren. Het beoogde onderwerp voor mijn bachelor proef is verbonden met deze vernieuwing. LanReview, een binnenhuis ontwikkeld assetmanagement programma dat onontbeerlijk is voor de IT Helpdesk, is aan vernieuwing toe. De laatste ontwikkeling is van enige tijd geleden en het is niet logisch meer om dit terug op te nemen voornamelijk wegens de verouderde code base (Visual BASIC 3.0). De vervangende applicatie moet gemaakt worden met Microsoft Power Apps en ondersteunend ook Power Automate.

Meer achtergrond over de IT voorziening van AZ Glorieux verklaart deze keuze. Het is een groot regionaal ziekenhuis met een netwerk dat meer dan 1000 toestellen telt. Het domein is gebouwd met System Center Configuration Manager. Zowel Clients als Servers gebruiken vormen van het Microsoft Windows besturingsysteem. Microsoft Office wordt op de meeste pc's gebruikt. Power Apps is ook opgenomen in het Office365 plan.

Een overzicht van de functies van LanReview die zeker ook in zijn opvolger aanwezig dienen te zijn:

- Overzicht geven van belangrijke informatie voor elke toestel in het domein.
- Rapporten kunnen genereren.
- Pc's bedienen vanop afstand.

Er zijn ook enkele volledig nieuwe functies beoogd:

- Mobiel bruikbaar zijn.

Overige functionaliteit:

- Grottere mate van Automatisatie introduceren met Power Automate.
- Buiten het domein bruikbaar zijn.
- Nieuwe types toestellen opnemen.
- Relevante Randapparatuur opnemen, deze in relatie stellen met de gekoppelde PC.
- Revisie van datavelden per toestel.
- Diverse Grafische User Interface (GUI) verbeteringen: kleuren gebruiken, tabbladen.
- Met barcode's kunnen werken. Bv de mac adres barcode inscannen om een nieuwe entry in de applicatie te maken.

Er zal een requirementsanalyse uitgevoerd worden via de MoSCoW-methode (Wikipedia, 2020). Dit staat verder beschreven in Methodologie.

Verdere uitleg van de hoofdfuncties van LanReview via enkele use cases:

- Er komt een telefoon binnen van een dokter die een probleem heeft met één van de medische programma's. Het computernummer wordt doorgegeven. We zoeken op dit nummer in LanReview en krijgen een overzicht van de PC in kwestie. Het is nu mogelijk te rechtsklikken en de pc over te nemen om het probleem verder op te lossen.
- Er werden nieuwe laptops aangekocht. We nemen deze op in LanReview door op z'n minst een niet toegewezen computernummer te kiezen samen met het mac-adres. Aanvullende zaken zijn: een beschrijving, model, locatie, ip-nummer.
- Een model laptop dient geüpdatet te worden naar Windows 10. Binnen LanReview queryen we op dit model en krijgen we een overzicht van elk toestel terug.

De hoofddoelstelling bestaat uit het opstellen van een competente Proof of Concept. Hiernaast moet verdere ontwikkeling zo toegankelijk mogelijk gemaakt worden. Met andere woorden de proef en bijbehorende documenten moeten een hoge mate van praktisch nut hebben voor het IT Team om niet gerealiseerde functionaliteit in de proof of concept en toekomstige functionaliteiten toe te kunnen voegen.

De onderzoeksvragen liggen in lijn met het bovenstaande:

- Is het mogelijk een vervanger voor LanReview te bouwen met Power Apps die elk de vier hoofdfuncties ondersteunt en op z'n minst drie vierde van de overige functionaliteiten kan ondersteunen?
- Is Power Apps werkelijk de beste keuze hiervoor of is er meerwaarde in een volledig

gerealiseerd IT asset management pakket? Is er alternatief een beter geschikt low-code platform?

- Is de Proof of Concept eenvoudig uit te breiden met nieuwe functionaliteiten? Is dit aanvaardbaar voor het IT team van AZ Glorieux?
- Is Power Apps robuust genoeg om meer complexe functionaliteit te ondersteunen. Is het met een zelf geschreven uitbreiding bijvoorbeeld mogelijk om de remote desktop functionaliteit te verwezenlijken?
- Is de gebruikte methode op z'n minst deels bruikbaar om andere applicaties voor de IT van AZ Glorieux te bouwen? Er worden twee cases onderzocht: één voor een telefoonboek (legacy business applicatie met een lage moeilijkheidsgraad) en één om het potentieel van Power Apps te demonstreren. Dit is opgenomen in de volgende onderzoeksvraag.
- Is er een use case voor nieuwe of experimentele functionaliteiten in Power Apps zoals integratie met Teams of beperkt toepassen van AI via de AI Builder?
- De Proof of Concept zal nauw samenwerken met SCCM, is het concreet mogelijk om de nabije toekomst ook samen te werken met Intune?

A.2 State-of-the-art

LanReview is eigenlijk een view op SCCM. Het is uit die databank dat de gegevens per toestel komen. Deze gegevens werden aangevuld met extra data velden en opgeslaan in SharePoint lijsten. Om te rapporteren worden SQL achtige queries gebruikt op deze lijsten. Het is mogelijk om op elk veld van de entries te filteren. De data van elke entry kan ook aangepast worden. Er is een optie aanwezig om pc's over te nemen. Het onderliggende remote desktopprotocol wordt opgeroepen vanuit LanReview om dit te verwezenlijken.

De term assetmanagement werd gebruikt om LanReview te beschrijven maar waar LanReview een view is op SCCM kan aan traditioneel IT Asset Management Program meestal gezien worden als alternatief voor SCCM. In dergelijke programma's is buiten inventarisatie ook voorziening voor netwerk discovery, analytics, security en meer. Gartner (Gartner, 2020) beschrijft het als volgt: *IT asset management (ITAM) provides an accurate account of technology asset lifecycle costs and risks to maximize the business value of technology strategy, architecture, funding, contractual and sourcing decisions.*

Het IT asset management landschap is al enige tijd in evolutie door verspreiding van smartphone technologie en IoT (Badnakhe, 2020), dit maakt de keuze voor een nieuwe en flexibele technologie als Power Apps logisch.

PowerApps is een low code ontwikkelingsplatform dat niet programmeurs toelaat om business apps te maken. Het is mogelijk om canvas of modelgestuurde applicaties te bouwen (Knight, 2019). Indien iets niet visueel gedaan kan worden is gebruik van een Excel achtige query taal mogelijk (Owen, Lindhorst & Vivek, 2019). Uitbreiding is mogelijk via Connectoren naar externe services. Uitbreidingen zijn ook volledig zelf te bouwen in C Sharp (Vivek, Hecke, Guderian, Daly & Lattimer, 2019). Recente innovaties binnen het Power platform zijn introductie van AI mogelijkheden aan de hand van Virtual

Agents en een grotere integratie met Teams (Cunningham, 2019). Vooral dit laatste is interessant omdat AZ Glorieux recent ook Teams is begonnen introduceren.

Automatisatie in een PowerApp kan verwezenlijkt worden via Power Automate, dat recent een naamwijziging heeft ondergaan van Flow (Weare, 2019) Volledig binnen de GUI is het mogelijk een automatisatieproces te maken bestaand uit aaneengeschakelde acties en condities.

Een populaire databron voor een PowerApp is SharePoint. Integratie gebeurt eenvoudigweg via de Connector ervoor. (Owen, Vivek, Manwaring, Blythe & Delano, 2019)

A.3 Methodologie

Er gaan twee onderzoekstechnieken gebruikt worden: een Proof of Concept voor de vervanger van LanReview gebouwd met PowerApps en een vergelijkende studie tussen beide applicaties. De vergelijking zal focussen op analyseren van requirements, bekijken hoe deze uitgewerkt kunnen worden en hierna ook illusterend vergelijken met behulp van simulaties.

Er zal gewerkt worden in fasen.

1. De Basics: Een globaal beeld schetsen van de IT omgeving van AZ Glorieux en hierna uitdiepen wat LanReview nodig heeft om te kunnen werken (Specifiek de gebruikte databronnen). Achtergrondinformatie toereiken voor de gevonden technologieën en de technologieën die we zullen gebruiken om onze POC te bouwen (Power Apps, Power Automate).
2. LanReview Reviewen: Hiermee bedoelt het interne werken van LanReview in kaart brengen tot in de nodige details. Helaas zullen we hierbij niet op codeniveau geraken omdat de broncode niet beschikbaar is. Dit betekent ook dat geen code overgenomen kan worden naar de POC maar dat enige originaliteit nodig zal zijn bij het uitwerken van de functionaliteiten. Deze en de vorige stap zal ondersteund worden met afbeeldingen. Er zal in de eerste stappen gefocust worden op toegankelijkheid en duidelijkheid.
3. Requirementsanalyse: Met behulp van de MoSCoW-methode gaan er prioriteiten gesteld worden in het aanzienlijk aantal beoogde functionaliteiten. De basis hiervan werd reeds gelegd tijdens gesprekken specifiek hiervoor gehouden tijdens de Stageperiode met de co-promotor en collega's die de applicatie praktisch zouden gaan gebruiken.
4. Alternatieven: Hoewel er hoogstwaarschijnlijk niet afgeweken zal worden van PowerApps zal de nodige aandacht geven worden aan alternatieven. Welke volledig uitgewerkte proprietary softwarepakketten bestaan er die LanReview kunnen vervangen? Uit de vergelijking van Finances Online blijkt dat er keuze genoeg is (FinancesOnline, 2020). Is er serieuze concurrentie voor PowerApps vanuit het Open Source kamp? Binnen PowerApps: is SharePoint de juiste technologie om onze

data op te slaan? De bedoeling hiervan is hoofdzakelijk inspiratie op te doen voor het uitwerken van de Proof of Concept. Als er echter uit dit onderzoek een superieure oplossing gevonden wordt zal dit voorgelegd worden aan de opdrachtgever.

5. Bouwen van de Proof of Concept: voor de praktische ontwikkeling van de POC laten we ons leiden door de voorganger en door de gebruikelijke technieken opgedaan in stap 1. De testfase zal uitvoerig zijn, de POC moet namelijk op termijn inzetbaar zijn binnen AZ Glorieux. Testen van code is niet aan de orde in een low-code ontwikkelplatform buiten de zelf geschreven uitbreidingen ervoor. Geautomatiseerd de UI testen is ook nog niet mogelijk doordat het Power Apps test Framework nog in de experimentele fase zit (Microsoft, 2020b). Tests zullen dus manueel opgesteld en uitgevoerd moeten worden. Indien toegelaten zou het ook behulpzaam zijn als de eindgebruikers bij tussenmomenten de Proof of Concept uitprobeerden en feedback gaven over hun ervaring.
6. Een vergelijking tussen LanReview met de POC. Wat zijn de gelijkenissen en verschillen? De uitwerking van belangrijke functionaliteiten zullen onderling vergeleken worden. In het bijzonder: Zijn er beperkingen in Power Apps gevonden waardoor ingeboet werd aan functionaliteit?
7. Aandacht voor de toekomst. De uitbreidingsmogelijkheden moeten gepeild worden. Extra functionaliteit moet zo vlot mogelijk geïntroduceerd kunnen worden door het IT Team. Is het mogelijk PowerApps elders toe te passen in AZ Glorieux via een soortgelijke methode als voor onze POC. Mogelijk wordt dit uitgelegd via een specifieke case. Hoe moet de POC aangepast worden om op termijn samen te kunnen werken met Intune zoals het nu nauw met SCCM samenwerkt? In dit gedeelte of één apart ervoor zal ook een practisch aspect aanwezig moeten zijn voor het IT Team van AZ Glorieux dat als documentatie zal moeten dienen of alternatief, verwijzingen moet hebben naar bestaande documentatie.

Praktischere zaken:

Wat tools betreft moet specifiek een PowerApps ontwikkelomgeving opgezet worden. Dit kan via de webapplicatie voor PowerApps als deel van het Office365 pakket. Alternatief zou het ook mogelijk moeten zijn om het softwarepakket "PowerApps Studio" te installeren. Als het maken van klassediagrammen niet uitgebreid mogelijk zou zijn vanuit PowerApps zal Visual Paradigm gebruikt worden.

Over AZ Glorieux, een belangrijke noot:

Als tijdens het uitwerken van de proef het AZ Glorieux netwerk of iets ermee te maken nodig zou zijn werd reeds voorgesteld dat ik terplaatse mag werken.

A.4 Verwachte resultaten

Tastbare resultaten:

- Een comprehensieve requirementsanalyse. Als hier niet de nodige aandacht wordt gegeven bestaat het risico van de "verkeerde" POC te bouwen. Een goede basis is

belangrijk dus er wordt verwacht dat de requirementsanalyse uitgebreid is.

- Logischerwijze zullen niet alle requirements opgenomen kunnen worden in de POC, toch wordt verwacht dat de belangrijkste uitgewerkt zijn. Ruimer gezien wordt verwacht dat de POC met weinig aanpassing de originele applicatie, LanReview, kan gaan vervangen.
- Een bijkomend resultaat is een set instructies of een stappenplan, de vorm kan nog wijzigen maar er zal een document gebouwd worden dat het IT Team kan gebruiken om verder te werken met PowerApps, om ontwikkeling van de POC over te kunnen nemen.

Met betrekking tot de POC:

Er wordt verwacht dat het opzetten van een PowerApps applicatie weinig problemen geeft, dit is uiteindelijk het doel van low code platformen zoals PowerApps. Er wordt wel enige uitdaging verwacht bij het uitwerken van functionaliteit die niet direct voorzien is. Uitbreidingen schrijven, vooral als het op automatisatie aankomt, kan zich moeilijk tonen. Er moet rekening mee gehouden worden in de planning.

Er wordt overigens verwacht dat er niet afgeweken zal worden van een combinatie van PowerApps, Power Automate en SharePoint voor het bouwen van de POC.

Indien dit gemeten wordt zal er vermoedelijk geen significant verschil zijn in performantie tussen de POC en LanReview.

A.5 Verwachte conclusies

De conclusies, geformuleerd als antwoorden op de onderzoeks vragen, zijn als volgt:

- Er wordt verwacht dat een vervanger voor LanReview gebouwd kan worden met LanReview. SharePoint kan gebruikt blijven als data bron maar SCCM (dus SQL Server) verbinding is mogelijk beperkt door licentie wijzigingen. (Microsoft, 2020a)
- Power Apps is de beste keuze in ons scenario. Zelf als hiervoor betere software gevonden werd zal de bijkomende kostprijs niet verantwoord kunnen worden.
- Het is mogelijk de POC af te leveren op een manier waarmee verder ontwikkeling ondersteund wordt.
- PowerApps is robuust genoeg om de hoofdfunctionaliteiten te ondersteunen, al dan niet met zelfgemaakte uitbreidingen.
- Er wordt verwacht dat de gebruikte techniek ook andere toepassingen kan vinden binnen AZ Glorieux.
- Een aanpassing naar Intune in plaats van of in combinatie met SCCM is doenbaar. De nodige tijds en moeite investering is aanvaardbaar.

Samenvattend wordt verwacht dat de POC een meerwaarde is voor AZ Glorieux en dat deze kan groeien tot waarde voor LanReview.

Bibliografie

- Aho, L. (2018). *Mobile data collectionand storing solution for Microsoft environments.*
- Andaloussi, M. S. (2018, januari 18). Performance considerations with PowerApps. Verkregen 18 mei 2020, van <https://powerapps.microsoft.com/en-us/blog/performance-considerations-with-powerapps/>
- Badnakhe, R. (2020, januari 23). Why Are More Businesses Adopting IoT for Asset Management? Verkregen 6 april 2020, van <https://dzone.com/articles/why-more-businesses-are-adopting-iot-for-asset-man>
- Bashar, L. (2017, juni 28). Introducing Sapphire, the Hospital Management System Built with OutSystems. Verkregen 6 april 2020, van <https://www.outsystems.com/blog/introducing-sapphire-hospital-management-system.html>
- Bloomberg, J. (2017, juni 22). Don't Confuse Low-Code with No-Code. Verkregen 6 april 2020, van <https://intellyx.com/2017/06/22/dont-confuse-low-code-with-no-code/>
- Bott, E. (2018, april 9). In Memoriam: All the consumer products Microsoft has killed off. Verkregen 18 mei 2020, van <https://www.zdnet.com/pictures/in-memoriam-all-the-consumer-products-microsoft-has-killed-off/>
- Brust, D. (2015, september 14). Installing Visual Basic/Studio 6 on Windows 10. Verkregen 6 april 2020, van <http://blog.danbrust.net/2015/09/14/installing-visual-basic-studio-6-on-windows-10/>
- classicvb petition. (2005, maart 8). A PETITION FOR THE DEVELOPMENT OF UNMANAGED VISUAL BASIC AND VISUAL BASIC FOR APPLICATIONS. Verkregen 6 april 2020, van <https://web.archive.org/web/20141024003217/http://www.classicvb.org/petition/>
- Cloudflare. (g.d.). What Is Vendor Lock-In? | Vendor Lock-In and Cloud Computing. Verkregen 18 april 2020, van <https://www.cloudflare.com/learning/cloud/what-is-vendor-lock-in/>

- Cunningham, R. (2019, november 4). Power Apps Top 10: innovation areas that are reshaping app development. Verkregen 14 februari 2020, van <https://powerapps.microsoft.com/en-us/blog/power-apps-top-10-innovation-areas-that-are-reshaping-app-development/>
- Devloo, M. (2018). *Kan een low-code development-platform een substantiële meerwaarde betekenen voor kleine en middelgrote bedrijven in de IT-sector?*
- Droogenbroodt, W. & Scheiris, J. (2016, april 11). Efficiënt beheer van uw PC-park met Microsoft System Center Configuration Manager. Verkregen 6 april 2020, van <https://www.savaco.com/nl/nieuws/detail/efficiënt-beheer-van-uw-pc-park-met-microsoft-system-center-configuration-manager>
- Dunnam, A. (2019, augustus 16). PowerApps Security Trimming Options. Verkregen 24 mei 2020, van <https://techdailychronicle.com/powerapps-security-trimming-options/>
- FinancesOnline. (2020, februari 27). Top 12 Most Popular IT Asset Management Software. Verkregen 2 maart 2020, van <https://financesonline.com/it-asset-management/>
- Gartner. (g.d.). It Asset Management (itam). Verkregen 6 april 2020, van <https://www.gartner.com/en/information-technology/glossary/it-asset-management-itam>
- Gartner. (2020). It Asset Management (itam). Verkregen 14 februari 2020, van <https://www.gartner.com/en/information-technology/glossary/it-asset-management-itam>
- Grigoris, R. (2014, maart 27). Where Is the Successor to Visual Basic? Verkregen 6 april 2020, van <https://www.newsmax.com/RichardGrigoris/Visual-Basic-Microsoft-programmers/2014/03/27/id/562161/>
- Gross, A. (2019, februari 12). The Microsoft Endpoint Configuration Manager Administration Service Guide. Verkregen 19 mei 2020, van <https://www.asquaredozen.com/2019/02/12/the-system-center-configuration-manager-adminservice-guide/>
- IAITAM. (g.d.). What is IT Asset Management (ITAM)? Verkregen 6 april 2020, van <https://iaitam.org/what-is-it-asset-management/?q=what-is-it-asset-management&v=d3dcf429c679>
- Ippolito, I. (2018, september 20). VB6 Desktop Icon Size Changer. Verkregen 6 april 2020, van <http://www.planetsourcecode.com/vb/scripts>ShowCode.asp?txtCodeId=75311&lngWId=1>
- ISplitter. (2014, januari 14). Visual Basic 6.0: A giant more powerful than ever. Verkregen 6 april 2020, van <https://www.codeproject.com/Articles/710181/Visual-Basic-6-0-A-giant-more-powerful-than-ever>
- Ivanti. (2018, oktober 10). What Is IT Asset Management? Verkregen 6 april 2020, van <https://www.ivanti.com/blog/what-is-it-asset-management>
- Jugo, A. (2019, december 1). Extending PowerApps and Flow, part 2: Extending PowerApps language with Azure API Apps. Verkregen 8 april 2020, van <https://blog.sharedove.com/adisjugo/index.php/2018/12/01/extending-powerapps-and-flow-part-2-extending-powerapps-expression-language-with-azure-api-apps/>
- Kissflow. (2018a, april 17). Low-Code Development Platforms – Launchpads to Build Powerful Apps Easily. Verkregen 6 april 2020, van <https://kissflow.com/rad/low-code/low-code-development-platform-launchpad-build-apps-easily/>
- Kissflow. (2018b, mei 2). The History of Low-Code Platforms: How Development Changed Forever. Verkregen 6 april 2020, van <https://kissflow.com/rad/low-code/history-of-low-code-development-platforms/>

- Knight, B. (2019, mei 14). PowerApps Canvas vs Model-driven Applications. Verkregen 14 februari 2020, van <https://blog.pragmaticworks.com/powerapps-canvas-vs-model-driven-applications>
- Lindhorst, G. (2018, mei 9). Advanced app building with PowerApps expressions and rules (Build 2018). Verkregen 18 mei 2020, van <https://www.youtube.com/watch?v=qRpD1w6gMPk>
- Luca, S. (2017, december 1). Upload and run an On Premises PowerShell script from Microsoft Flow. Verkregen 18 mei 2020, van <https://sergeluca.wordpress.com/2017/12/01/upload-and-run-a-remote-powershell-script-from-microsoft-flow/>
- Marvin, R. (2017a, juli 31). Mendix Review. Verkregen 18 april 2020, van <https://www.pcmag.com/reviews/mendix>
- Marvin, R. (2017b, juli 31). OutSystems Review. Verkregen 18 april 2020, van <https://www.pcmag.com/reviews/outsystems>
- Marvin, R. (2017c, juni 12). Salesforce App Cloud Review. Verkregen 18 april 2020, van <https://www.pcmag.com/reviews/salesforce-app-cloud>
- McKendrick, J. (2017). *THE RISE OF THE EMPOWERED CITIZEN DEVELOPER - 2017 LOW-CODE ADOPTION SURVEY*. Unisphere Research.
- Meganathan, A. (2019, juli 26). Build Asset Checkout App Using Microsoft PowerApps. Verkregen 8 april 2020, van <https://www.c-sharpcorner.com/article/build-asset-checkout-app-using-microsoft-powerapps/>
- Mendix. (g.d.). Pricing. Verkregen 19 april 2020, van <https://www.mendix.com/pricing>
- Microsoft. (2020a, januari 7). Power Apps and Power Automate licensing FAQs. Verkregen 2 maart 2020, van <https://docs.microsoft.com/en-us/power-platform/admin/powerapps-flow-licensing-faq>
- Microsoft. (2020b, januari 10). Power Apps test framework. Verkregen 2 maart 2020, van <https://docs.microsoft.com/en-us/power-platform-release-plan/2019wave2/microsoft-powerapps/test-framework>
- Microsoft Docs. (2017, juni 12). Create a canvas app to manage projects. Verkregen 8 april 2020, van <https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/sharepoint-scenario-build-app>
- Microsoft Docs. (2018, januari 18). Support Statement for Visual Basic 6.0 on Windows. Verkregen 6 april 2020, van <https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-basic-6/visual-basic-6-support-policy>
- Microsoft Docs. (2019a, januari 3). Aan de slag met het canvas-app-formules in Power apps. Verkregen 8 april 2020, van <https://docs.microsoft.com/nl-nl/powerapps/maker/canvas-apps/working-with-formulas>
- Microsoft Docs. (2019b, november 5). Get started with Power Automate. Verkregen 8 april 2020, van <https://docs.microsoft.com/en-us/power-automate/getting-started>
- Microsoft Docs. (2019c, november 18). Test Studio (experimental). Verkregen 8 april 2020, van <https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/test-studio>
- Microsoft Docs. (2019d, juni 12). Wat is Common Data Service? Verkregen 8 april 2020, van <https://docs.microsoft.com/nl-nl/powerapps/maker/common-data-service/data-platform-intro>
- Microsoft Docs. (2019e, september 6). What is AI Builder? Verkregen 8 april 2020, van <https://docs.microsoft.com/en-us/ai-builder/overview>

- Microsoft Docs. (2019f, oktober 16). What is an on-premises data gateway? Verkregen 8 april 2020, van <https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/gateway-reference>
- Microsoft Docs. (2019g, juli 15). What is Power Apps? Verkregen 8 april 2020, van <https://docs.microsoft.com/en-us/powerapps/powerapps-overview>
- Microsoft Docs. (2020a, april 2). Create or edit filters in model-driven app views. Verkregen 18 mei 2020, van <https://docs.microsoft.com/en-us/powerapps/maker/model-driven-apps/create-edit-view-filters>
- Microsoft Docs. (2020b, februari 12). Frequently asked questions for Configuration Manager branches and licensing. Verkregen 6 april 2020, van <https://docs.microsoft.com/en-us/configmgr/core/understand/product-and-licensing-faq>
- Microsoft Docs. (2020c, maart 19). Overview of canvas-app connectors for Power Apps. Verkregen 8 april 2020, van <https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/connections-list>
- Microsoft Docs. (2020d, mei 19). Overview of Microsoft Graph. Verkregen 19 mei 2020, van <https://docs.microsoft.com/en-us/graph/overview>
- Microsoft Docs. (2020e, januari 17). What is Power Apps portals? Verkregen 8 april 2020, van <https://docs.microsoft.com/en-us/powerapps/maker/portals/overview>
- Microsoft Power Automate. (2019, november 4). Learn about robotic process automation in Power Automate. Verkregen 12 april 2020, van https://www.youtube.com/watch?v=bZrrdoytTH0&feature=emb_title
- Microsoft Support. (2020, januari 15). Ondersteuning voor Windows 7 beëindigd op 14 januari 2020. Verkregen 12 april 2020, van <https://support.microsoft.com/nl-be/help/4057281/windows-7-support-ended-on-january-14-2020>
- MS Login Connector Team. (2019, oktober 17). Microsoft Login Connector Reactive (Documentation). Verkregen 19 mei 2020, van https://www.outsystems.com/forge/Component_Documentation.aspx?ProjectId=6933&ProjectName=microsoft-login-connector-reactive
- Ness, C. & Hansen, M. E. (2019). *Potential of low-code in the healthcaresector* (master-scriptie, Norwegian School of Economics).
- Nucleus. (2017, januari 16). Wat is het verschil tussen IaaS, PaaS, SaaS en UaaS? Verkregen 6 april 2020, van <https://www.nucleus.be/blog/uptime-as-a-service/verschil-iaas-paas-saas-en-uaas/>
- Outsystems. (g.d.). Pricing and Editions. Verkregen 19 april 2020, van <https://www.outsystems.com/pricing-and-editions/>
- OutSystems Docs. (g.d.). Validate the fields of a form. Verkregen 19 mei 2020, van https://success.outsystems.com/Documentation/11/Developing_an_Application/Design_UI/Forms/Validate_the_fields_of_a_form
- Owen, A. F., Lindhorst, G. & Vivek, K. (2019, december 19). Formula reference for Power Apps. Verkregen 14 februari 2020, van <https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/formula-reference>
- Owen, A. F., Vivek, K., Manwaring, M., Blythe, M. & Delano, L. (2019, maart 4). Connect to SharePoint from a canvas app. Verkregen 14 februari 2020, van <https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/connections/connection-sharepoint-online>

- Pohl, T. (2019, september 27). Overview of the new pricing models for Microsoft PowerApps and Flow. Verkregen 8 april 2020, van <https://blog.atwork.at/post/Microsoft-PowerApps-and-Flow-licensing>
- Pragmatic Works. (2019, april 3). PowerApps Canvas vs Model-Driven Applications. Verkregen 8 april 2020, van https://www.youtube.com/watch?v=fhIlC1mSO_4&feature=emb_title
- Reselman, B. (2018, januari 31). Why the promise of low-code software platforms is deceiving. Verkregen 6 april 2020, van <https://searchsoftwarequality.techtarget.com/opinion/Why-the-promise-of-low-code-software-platforms-is-deceiving>
- Revell, M. (2020, januari 16). What Is Low-Code? [2020 Update]. Verkregen 6 april 2020, van <https://www.outsystems.com/blog/what-is-low-code.html>
- Richardson, C. & Rymer, J. R. (2016, januari 15). *Vendor Landscape: The Fractured, Fertile Terrain Of Low-Code Application Platforms*. Forrester.
- Rouse, M. (2019, juli). Visual Basic (VB). Verkregen 6 april 2020, van <https://searchwindevelopment.techtarget.com/definition/Visual-Basic>
- Rymer, J. (2018, augustus 8). Why You Need To Know About Low-Code, Even If You're Not Responsible For Software Delivery. Verkregen 12 april 2020, van <https://go.forrester.com/blogs/why-you-need-to-know-about-low-code-even-if-youre-not-responsible-for-software-delivery/>
- Rymer, J. R. & Koplowitz, R. (2019, maart 13). *The Forrester Wave: Low-Code Development Platforms For ADD Professionals, Q1 2019*. Forrester.
- Salesforce. (g.d.-a). Platform Pricing. Verkregen 27 mei 2020, van <https://www.salesforce.com/editions-pricing/platform/>
- Salesforce. (g.d.-b). What Is Digital Transformation? Verkregen 6 april 2020, van <https://www.salesforce.com/products/platform/what-is-digital-transformation/>
- Shiah, J. (2018, april 27). Why “low-code development” isn’t always truly “low-code”. Verkregen 6 april 2020, van <https://www.itproportal.com/features/why-low-code-development-isnt-always-truly-low-code/>
- Speed, R. (2020, maart 12). Microsoft throws a bone to those unable to leave the past behind: .NET 5 support on the way for Visual Basic. Verkregen maart2020, van https://www.theregister.co.uk/2020/03/12/visual_basic_net_5/
- Spitze, P. (2019, oktober 7). Microsoft SCCM and the alternatives in 2019. Verkregen 6 april 2020, van <https://www.software2.com/resource-centre/microsoft-sccm-and-the-alternatives-in-2019>
- Spriet, J. (2019). *Hoe kunnen de administratieve systemen van verhuizingen De Borgerge-optimaliseerd en efficiëntergemaakt worden?*
- Totterdale, R. L. (2018). CASE STUDY: THE UTILIZATION OF LOW-CODE DEVELOPMENT TECHNOLOGY TO SUPPORT RESEARCH DATA COLLECTION. *Issues in Information Systems*, 19(2), 132–139.
- van Schetsen, A. (2016, mei). Guest View: Five reasons low-code development is no longer optional. Verkregen 6 april 2020, van <https://sdtimes.com/agile/guest-view-five-reasons-low-code-development-no-longer-optional/>
- Vincent, P., Iijima, K., Driver, M. & Natis, Y. (2019, augustus 8). *Magic Quadrant for Enterprise Low-Code Application Platforms*. Gartner.
- Virta, T. (2018). *Relation of low-code development to standard software development: Case Biit Oy* (masterscriptie, Lappeenranta University of Technology).

- Vivek, K., Hecke, P., Guderian, B., Daly, J. & Lattimer, J. (2019, maart 27). Use plug-ins to extend business processes. Verkregen 14 februari 2020, van <https://docs.microsoft.com/en-us/powerapps/developer/common-data-service/plug-ins>
- Weare, K. (2019, december 23). Microsoft Rebrands Flow Service to Power Automate, Adds Robotic Process Automation Capabilities. Verkregen 14 februari 2020, van <https://www.infoq.com/news/2019/12/Power-Automate-RPA/>
- Wikipedia. (2020, maart 1). MoSCoW-methode. Verkregen 2 maart 2020, van <https://nl.wikipedia.org/wiki/MoSCoW-methode>
- Ziehnert, N. (2020, januari 28). Building PowerApps for ConfigMgr - The Frugal Way (Part 1). Verkregen 18 mei 2020, van <https://z-nerd.com/blog/2020/01/27-building-powerapps-for-configmgr-the-frugal-way/>