



samk



Satakunnan ammattikorkeakoulu  
Satakunta University of Applied Sciences

SAKU HIETAVA

# **Huoltoraportin dokumentointi ohjelmistorobotin avulla**

SÄHKÖ- JA AUTOMAATIOTEKNIikka

## TIIVISTELMÄ

Hietava, Saku: Huoltoraportin dokumentointi ohjelmistorobotin avulla  
Opinnäytetyö, AMK  
sähkö- ja automaatiotekniikka  
Tammikuu 2023  
Sivumäärä: 32

Tässä opinnäytetyössä rakennettiin toimiva ohjelmistorobotti TruTekniikka Oy:lle. Ohjelmistorobotin tarkoitus on automatisoida huoltoraporttien dokumentointi. Raportit tulevat Outlook-sähköpostiosoitteeseen PDF-tiedostoina.

Ensimmäisessä osiossa tutustuttiin ohjelmistorobottiin, niiden historiaan sekä hyöty- että haittapuoliin.

Toisessa osuudessa suunniteltiin robotin toiminta ja se, millä työkalulla ohjelmistorobottia lähdettiin rakentamaan. Valittu työkalu oli Microsoftin Power automate web -työkalu.

Työn kolmannessa vaiheessa ohjelmistorobotti rakennettiin käyttäen Microsoft Power automate - sekä AI builderai -työkaluja.

Työn lopussa toimiva ohjelmistorobotti esitellään toimintatilanteessa.

Avainsanat: ohjelmistorobotiikka, RPA, AI Builder, Microsoft Power automate

## Abstract

Hietava, Saku: Documenting the maintenance report using a robotic process automation

Bachelor's thesis

Electrical and automation engineering

January 2023

Number of pages: 32

In this thesis we build a robotics process automation software for TruTekniikka Oy. RPA is meant to automatize reading and reporting of maintenance reports. Maintenance reports are sent to Outlook.

In the first phase, we got acquainted with history of RPA, and their advantages and disadvantages.

In the second part, the operation of the robot was planned, and which tool was used to start building the RPA. The chosen tool was Microsoft's Power automate web tool.

In the third phase of the work, the software robot is built using Microsoft power automate and AI builder.

Last phase will show completed software in use.

Keywords: robotic process automation, RPA, AI Builder, Microsoft Power automate

## ALKUSANAT

Tämä opinnäytetyö toteutettiin yhteistyössä TruTekniikka Oy:n kanssa. Haluan kiittää mahdollisuudesta rakentaa tämä ohjelmistorobotti, sekä minulle uskotusta luottamuksesta käsitellä yrityksen arkaluonteista materiaalia.

# SISÄLLYS

1 JOHDANTO .....	6
1.1 Työn selitys .....	6
1.2 TruTekniikka.....	6
1.3 Trumpf.....	6
2 OHJELMISTOROBOTTI .....	7
2.1 Mikä on ohjelmistorobotti.....	7
2.2 Historia .....	7
2.3 Ohjelmistorobotin hyödyt.....	8
2.4 Ohjelmistorobottiikan haitat .....	9
2.5 Ohjelmistorobotti-sovellukset.....	10
3 OHJELMISTOROBOTIN SUUNNITTELU JA RAKENTAMINEN.....	11
3.1 Ohjelmistorobotin suunnittelu .....	11
3.2 Työkalun valitseminen .....	12
3.3 Ohjelmistorobotin luonti.....	14
3.4 Ohjelmistorobotin luominen: raporttipohjan opettaminen.....	15
3.5 Sharepointin luonti.....	19
3.6 Excelin luonti ja muotoilu .....	20
3.7 Logiikan luonti .....	21
3.7.1 Laukaisin ja Execl-pohjan datan kopionti.....	22
3.7.2 Sähköpostin automatisointi.....	24
3.7.3 Excel-taulukon sisältö ja vuon haaroittaminen.....	27
4 OHJELMAN TOIMINTA KÄYTÄNNÖN TILANTEESSA.....	33
5 LUOMANI OHJELMISTOROBOTIN KEHITYS JA MAHDOLLISET VAARAT .....	35
6 YHTEENVETO.....	36
LÄHTEET.....	38

# 1 JOHDANTO

## 1.1 Työn selitys

Tämän opinnäytetyön tarkoituksena on luoda RPA eli ohjelmistorobotti automatisoimaan huoltoraporttien dokumentointia yritykselle TruTekniikka Oy. TruTekniikka Oy tuo Suomeen teollisuuskäyttöisiä laser-leikkauskoneita, jotka ovat saksalaisvalmisteisia. Valmistajana toimii Trumpf.

TruTekniikalla oli tarvetta järjestelmälle, joka automatisoi takuuhuoltoihin menevien kulujen laskemisen. Tämänhetkisessä järjestelmässä takuuhuoltokulut, joihin kuuluvat autolla ajettavat kilometrit, työntekijän palkka ja työntekijän yöpyminen, ylöskirjataan manuaalisesti. Tässä opinnäytetyössä tutkitaan ja vertaillaan erilaisia ohjelmistorobottiohjelmiä, ja rakennetaan vertailuun perustuen ohjelmistorobotti, joka vastaa TruTekniikan tarpeeseen sujuvoittaa takuuhuoltokulujen laskeminen.

## 1.2 TruTekniikka

TruTekniikka Oy kertoo itsestään nettisivulla lyhyesti: ”TruTekniikka Oy on ollut myymässä TRUMPFia ja tekemässä sitä tunnetuksi jo liki 40 vuotta. Yrityksenä TruTekniikka Oy aloitti TRUMPF-levytyökoneiden myynnin ja markkinoinnin Suomessa 1993.” (TruTekniikka Oy.)

## 1.3 Trumpf

Trumpf on vuonna 1923 perustettu saksalainen yritys, joka valmistaa metallintyöstökoneita ja se kuuluu maailman isoimpiin työstökoneiden valmistajiin.

Trumpf toimii jokaisessa maanosassa ja työskentelee yli 70 tytäryhtiön kanssa. Trumpf työllistää yli 14 500 työntekijää maailmanlaajuisesti ja on sitoutunut Pariisin ilmastositoumuksiin. (Trumpf, 2023.)

## 2 OHJELMISTOROBOTTI

### 2.1 Mikä on ohjelmistorobotti

RPA (Robotic Process Automation) eli ohjelmistorobotti on sovellus, jonka avulla on yksinkertaista luoda niin kutsuttuja robotteja tai botteja, jotka ohjelmoidaan matkimaan ihmisen toimintaa. Ohjelmistorobotit pystyvät tulkitsemaan ruutunäkymää. Ne voidaan myös opettaa tekemään oikeat hiiren liikkeet ja näppäimistön painallukset, navigoimaan tietokoneen järjestelmää ja kansioita, sekä lukemaan dataa eri tiedostoista ja internetistä. (UiPath.)

Tällä hetkellä ohjelmistorobotteja käytetään tietokoneella tehtäviin toisteisiin tehtäviin, joihin ihmistyövoiman käyttäminen voidaan nähdä resurssien tuhlausena: tällaisia tehtäviä ovat muun muassa erilaiset finanssialalla, telekommunikaatiossa ja terveydenhuollossa toistuvat tehtävät. Ohjelmistorobotteja käytetään myös tiedon raapimiseen internetistä. Erityisesti ohjelmistorobottiikka hyödynnetään teollisuudessa. Ohjelmistorobottimarkkinan odotetaan olevan 5 miljardia dollaria vuoteen 2024 mennessä. (Lawton, 2021.)

### 2.2 Historia

Ensimmäinen ohjelmistorobotti-sovellus kehitettiin 2000-luvun alussa, vaikka tämälantapaisia ohjelmia käytettiin jo 1990-luvun puolella tietokoneruutujen harvointiin, ja tiettyjen ohjelmistoroboteissa käytettävien teknologioiden juuret ovat seurattavissa aina 1960-luvulle. 2000-luvun alussa ohjelmistorobotteja

käytettiin pääasiallisesti uusien käyttöjärjestelmien käyttöönotossa tilanteissa, joissa uusi ja vanha järjestelmä piti saada keskustelemaan keskenään. 2000-luvun aikana ohjelmistorobottien kehitys jatkui, ja uusia käyttömahdollisuuksia keksittiin. Ohjelmistorobotit eivät kuitenkaan lyöneet itseään läpi ennen vuotta 2015, jolloin ohjelmistorobottien hyödyt oli saatu niin suuriksi, että niiden käyttöönotto alkoi olla yrityksille kannattavaa. Tämä johtuu suuresti tekoälykehityksestä, jota pystyttiin suoraan hyödyntämään ohjelmistorobottien toiminnan parantamisessa. (Robomotion, 2021.)

Kaiken kaikkiaan nykyaikaista ohjelmistorobotiikkaa voidaan pitää uutena teknologiana. Ohjelmistorobotiikan uutuus näkyy tällä hetkellä teknologian nopeana kehittymisenä. On lähes mahdotonta kuvitella, mitä mahdollisuuksia ohjelmistorobotiikalla on tulevaisuudessa.

### 2.3 Ohjelmistorobotin hyödyt

Ohjelmistorobottien tavoite on poistaa pitkäväteiset ja toisteiset työt ihmisiltä ja siirtää ne robotille. Ohjelmistorobotteja markkinoidaan yrityksille erityisesti neljään hyötyyn vetoamalla:

1. ”Vedä ja pudota” -tapaisella ohjelmistoympäristöllä henkilö ei tarvitse laajempaa ymmärrystä ohjelmoinnista. (IBM Cloud Education, 2022.)

2. Työntekijät voidaan vapauttaa työtehtävistä, jotka vaativat runsaasti toistoja. Tämä tuo yritykselle voittoa sekä työntekijöille mielekkäämpiä työtehtäviä, jossa työntekijä voi käyttää mietintä- ja päätösvaltaansa. (IBM Cloud Education, 2022.)

3. Ohjelmistorobotit eivät tarvitse lepotaukoja, joten yritys voi toimia kellon ympäri. Tämä nostaa yrityksen tuottoa sekä asiakkaiden tyytyväisyyttä. (IBM Cloud Education, 2022.)



4. Ohjelmistorobottien sääntöpohjaisen ohjelmistomallin avulla saadaan poistettua ihmisvirheet työkuvaan. Tarkkuutta vaativia töitä, esimerkiksi pitkien numerorivien kopioimista paperilta koneelle, ohjelmistorobotti tekee kellon ympäri väsymättä ja virhettä tekemättä. (IBM Cloud Education, 2022.)

Nämä hyödyt, joilla ohjelmistorobotteja markkinoidaan, ovat pitkälti totta, mutta herättävät myös tiettyjä kysymyksiä. Toimivat ohjelmistorobotit ovat kieltämättä yrityksille rahallisen voiton kannalta houkuttelevia, mutta työntekijöiden vapauttaminen toisteisista työtehtävistä ja siirtäminen mielekkäämpiin tehtäviin on pelkkää sanahelinää. Tulevaisuuden työelämää en lähde ennustamaan, mutta lienee vääjäämätöntä, että ainakin välittömällä aikataululla laaja ohjelmistorobottien käyttöönotto aiheuttaa työpaikkojen vähentymistä. On myös humaania huomioida erilaisten ihmisten tarpeet ja vahvuudet työelämässä; myös niiden, joille parhaiten nämä tarkkuutta vaativat toisteiset työt sopivat, on saatava työllistyä.

Markkinoinnissa tuodaan myös näkemys siitä, ettei ohjelmistorobottiohjelman käyttämiseen vaadita laajempaa ymmärrystä ohjelmoinnista. Tämä on hieman harhaanjohtava käsitys, sillä vaikkei ohjelmistorobotin käyttäminen vaadi koodustaitoja, vaatii se ymmärrystä algoritmien toiminnasta ja ohjelmien käyttäytymisestä.

## TÄTÄ KATSO

### 2.4 Ohjelmistorobotiikan haitat

Vaikka ohjelmistorobotit tuovat yleisesti ottaen enemmän hyötyä kuin haittaa, tulee silti ymmärtää ohjelmistorobottiteknologian haittapuolek. Käytännössä kuka vain voi rakentaa itse ohjelmistorobotin: tästä isoissa organisaatioissa seurauksena voi olla, että monta pientä koordinoimattomasti luotua robottia yhdessä luovat sekavan verkoston, jonka organisointi tulee mahdottomaksi. Näissä tilanteissa tulee tukeutua ammattilaisten hallitsemiin ohjelmistorobotteihin. (Pratt, M. K., 2021.)

Organisaatioiden tulisi myös objektiivisesti miettiä, onko työn automatisointi tarpeellista tai edes mahdollista. Monimutkaisten prosessien laatiminen ohjelmistorobotille sopivaan muotoon vaatii suunnittelua ja ymmärrystä tekniikasta. Kuten minkä tahansa muun ohjelman, myös toimivan ohjelmistorobotin luominen vaatii teknologian ymmärrystä hyvän lopputuloksen aikaansaamiseksi. Huonosti suunniteltu ja valmistettu ohjelmistorobotti ei nopeuta prosessia vaan hidastaa sitä. Virheellisten prosessien korjaaminen ja jatkuva koodin muokkaaminen tulee lopulta maksamaan enemmän kuin manuaalisesti tehty työ. (Pratt, M. K., 2021.)

## 2.5 Ohjelmistorobotti-sovellukset

Ohjelmistorobotti-sovellusten valmistajia on useita, mutta tässä opinnäytetyössä keskityn kahteen eri valmistajaan. UiPath ja Microsoftin Power Automate. UiPath on yksi suurimmista ohjelmistorobotti-sovellusten valmistajista, sekä itselleni entuudestaan tuttu, joten se valikoitui tarkasteltavaksi tästä syystä. Microsoftin Power Automate valikoitui toiseksi, sillä TruTekniikalla oli käytössään Microsoftin Office365-järjestelmä ja Microsoftin Power Automate oli helppo integroida toimimaan yhdessä sen kanssa. Valitsin tarkasteltavaksi kaksi eri järjestelmää, sillä opinnäytetyön laajuus huomioiden kaksi sovellusta tuntui sopivalta määrältä ja mahdollisesti kuitenkin vertailun.

## 3 OHJELMISTOROBOTIN SUUNNITTELU JA RAKENTAMINEN

### 3.1 Ohjelmistorobotin suunnittelu

Organization		Group	Technician	Mission no.:	Case no.:	Page:
TRUTEKNIikka		SERVICE		4438131	7118399	1/2

Activity report						
Machine		Number	Type			
		S08	TruLaser Cell 7040 (L13)			
Customer		Company	Street	Postal code:	City:	Customer number
					Vantaa	
					Country:	FI
Order data		Customer no.:	SAP service no.:	SAP consignment no.:	SAP spare part no.:	
Contact		Contact person:	Phone:			
Travel expenses		Receipt travel expenses (€):	Overnight stays	per (€):	Meal	
			1			
		Travel by car (km):	Return trip (km):			
		145	160			
Operating hours		Date:	NC:	Resonator / High pressure pump 1:	Resonator / High pressure pump 2:	
		10/22/2021	18037	17480	2586	
Security issues						
Customer comment						

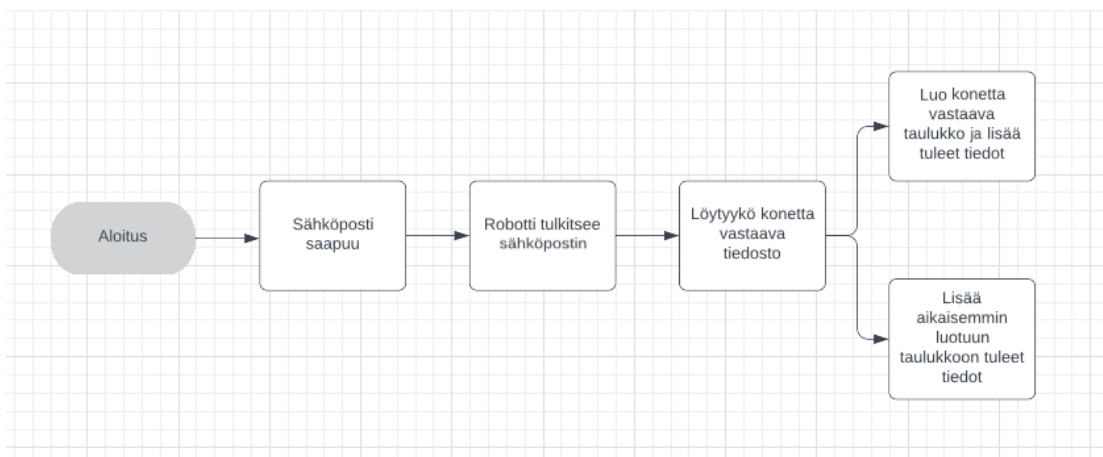
  

Times

Date	Start of travel time	Working time		End of travel time	Duration		Breaks		Mission no.	Passive travel time
		From	until		Work	Travel	Work	Travel		
10/22/2021		08:30	14:30	16:45	06:00	02:15			4438131	
10/25/2021	07:30	09:45	15:00	15:30	05:15	02:45			4438131	
10/26/2021	07:30	08:00	15:00	15:30	07:00	01:00			4438131	

(Kuva 1. Raporttipohja)

Kuvassa 1 on PDF-tiedosto, joka joudutaan tällä hetkellä ylöskirjaamaan manuaalisesti. Tavoitteena on luoda robotti, joka tulkitsee tiedostosta asiakkaan nimen, huollettavan koneen numeron, huoltotyöntekijän ajamat kilometrit, hotelliyöpymiset sekä kuluneet tunnit ja kirjaa ne järjestelmään.



(Kuva 2. Vuokaavio)

Vuokaaviossa (kuva 2) esitetään suunnitelma siitä, minkälainen ohjelmistorobotti tuli TruTekniikka Oy:lle luoda. Olen tehnyt suunnitelman perustuen yrityksen esittämään ideaan ja toiveisiin. Omaksi tehtäväkseni jäi suunnitella, miten idea toteutetaan käytännön tasolla. Seuraavaksi esittelen ohjelmistorobotille suunnittelemani toimintaperiaatteen.

Suunnittelemani ohjelmistorobotin toimintatapa on selkeästi ymmärrettävä vuokaavion (kuva 2) avulla. Tarkoitukseni oli, että taustalla robotti on jatkuvasti valppaana odottamassa heräämiskomentoa. Heräämiskomento on sellaisen sähköpostin saapuminen, joka täyttää robotille ilmoitetut parametrit. Tässä opinnäytetyössä parametrinä käytetään sähköpostin artikkelia. Näin vältetään siitä, että robotti yrittäisi ajaa väärää PDF-tiedostoja algoritmin läpi. Ohjelmistorobotti-ohjelmissa parametrejä voi antaa enemmänkin, esimerkiksi sähköpostiosoitteen, mutta opinnäytetyön tarpeeseen yksi parametri oli riittävä.

Seuraavaksi ohjelmistorobotti tulkitsee sähköpostiin tuleen PDF-tiedoston. Tiedostossa tärkeät tiedot ovat asiakkaan nimi, huolletun koneen numero sekä työntekijän ajokilometrit, hotellööiden määrä ja työtunnit.

Kolmanneksi robotti tarkistaa, löytyykö huolletun koneen nimeä vastaava taulukkoa ilmoitetun yrityksen Sharepoint-kansiosta. Mikäli koneen nimi löytyy, lisää robotti tulleet tiedot jo aikaisemmin luotuun taulukkotiedostoon. Mikäli taas kone on uusi yrityksen Sharepoint-kansiossa, koneelle luodaan uusi taulukkotiedosto, johon ohjelmistorobotti tallentaa sähköpostissa saamansa tiedot.

### 3.2 Työkalun valitseminen

Minun oli tehtävä ennen ohjelmoinnin aloittamista päätös siitä, millä työkalulla ohjelmistorobottia lähdettiin rakentamaan. Tärkeimpiä valintaperusteita käytettävälle ohjelmistorobotti-sovellukselle olivat työkalun ominaisuudet ja hinta.

UiPath ja Power automate tarjoavat molemmat ilmaista sovellusversiota, joka soveltuu erinomaisesti pienten ohjelmistorobottien rakentamiseen. Nämä versiot ovat kuitenkin yksinkertaisia, eivätkä sisällä niitä kaikkia työkaluja ja ominaisuuksia, joita tilattuun ohjelmistorobotin luomiseen tarvitsin. UiPathilla ja Power automatella on molemmilla tarjolla myös maksullisia versioita, joihin sisältyy monipuolisemmat työkalut ja ominaisuudet.

UiPath myy maksullista versiota, johon kuuluu useamman lisenssin rajaton käyttöoikeus, mikä tarkoittaa, että sovelluksessa on yhdellä käyttöoikeudella oikeus rakentaa kuinka monta bottia tahansa. UiPathin maksullisen pro-version aloitushinta on 420\$/kuukaudessa, mutta ostetun version hinta vaihtelee ostettujen lisenssien määrän perusteella (UiPath, 2023). Microsoftin Power automate taas sisältää usean eri hintaluokan sovellusvaihtoehtoja, joissa vaihtelua on niin sovelluksen sijainnin sekä lisenssien, flown ja robottien määrän suhteen. Hintahaitariksi Microsoft antaa 15-500\$ kuukaudessa. (Microsoft, 2023.) Käytännössä hintahaitarin ylärajaa on kuitenkin vaikea arvioida. UiPathin pro-versio palvelee siis etenkin isojen yritysten tarpeita, kun taas Microsoftin joustavimmat vaihtoehdot sopivat pienyrityksille ja yksityishenkilöille paremmin.

Sekä UiPathin että Power automaten pro-versioihin kuuluu perusteellinen dokumentaatio. Molemmat työkalut ovat yhdistettävissä Microsoft Outlookiin, joka on TruTekniikan käyttämä sähköpostipalvelu, jota kautta ohjelmistorobotti vastaanottaa huoltokulut sisältävät PDF-tiedostot. Molemmat sovellukset ovat myös yhdistettävissä Sharepointiin. Molempiin sovelluksiin kuuluu myös PDF-tiedostojen analysointiominaisuus.

Testatessani ohjelmia paljastui ongelma. TruTekniikalta tulevat PDF-tiedostot eivät olleetkaan vakionuotoisina vaan kuvina. Näin ollen UiPathin ja Power automaten PDF-analysointityökalut eivät pystyneet saamaan tiedostoista tarvittavaa tietoa. PDF-analyysityökalu ei siis ollut riittävä, sillä tarvitsin ohjelmistorobottiin ominaisuuden tulkita kuvista tietoa ja ymmärtää erilaisia raporttipohjia. Pelkkää konenäköä käyttämällä tiedoston tulkitseminen ei onnistunut, vaan

minun tuli löytää työkalu, jossa oli sekä konenäky että tekoäly. Power automatesissa oleva työkalu AI Builder sopi tarpeeseeni. Vastaavaa työkalua, jota olisin pystynyt työssäni käyttämään, ei löytynyt UiPathista.

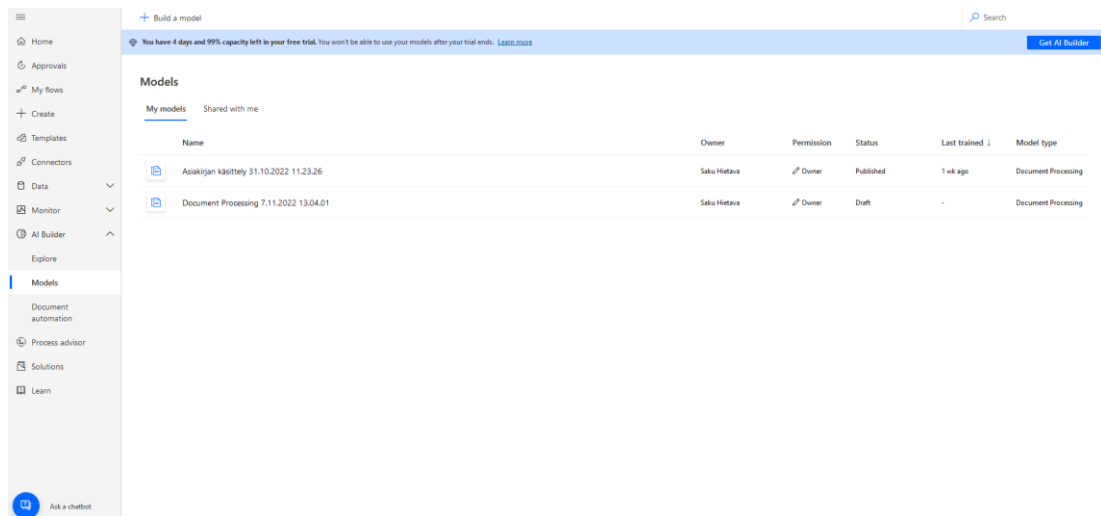
AI Builder on työkalu Microsoft Power Automatesissa, joka on suunniteltu optimoimaan liiketoimintaprosesseja. AI Builder mahdollistaa tietojen keräämisen ja prosessoinnin Power Automates -järjestelmän avulla. AI Builder sisältää laajan valikoiman prosessipohjia, jotka voi räätälöidä omaan käyttötarkoitukseen.

(Microsoft, 2022.) AI Builderia käyttämällä ohjelmistorobotti kykeni tulkitsemaan tiedon kuvamuotoisista PDF-tiedostoista ja lisäämään tiedot taulukkoon.

Opinnäytetyön ohjelmistorobottiin valikoitui lopulta siis Power Automates -sovellus perustuen AI Builder -työkaluun. Toinen painava syy oli se, että koulun kautta sain käytännöllisesti hankittua lisenssin Power Automates -proversioon. Power Automates toimi myös hieman sujuvammin Office365-palvelun kanssa, joten valinta UiPathin ja Power Automatesin välillä oli lopulta hyvin yksinkertainen.

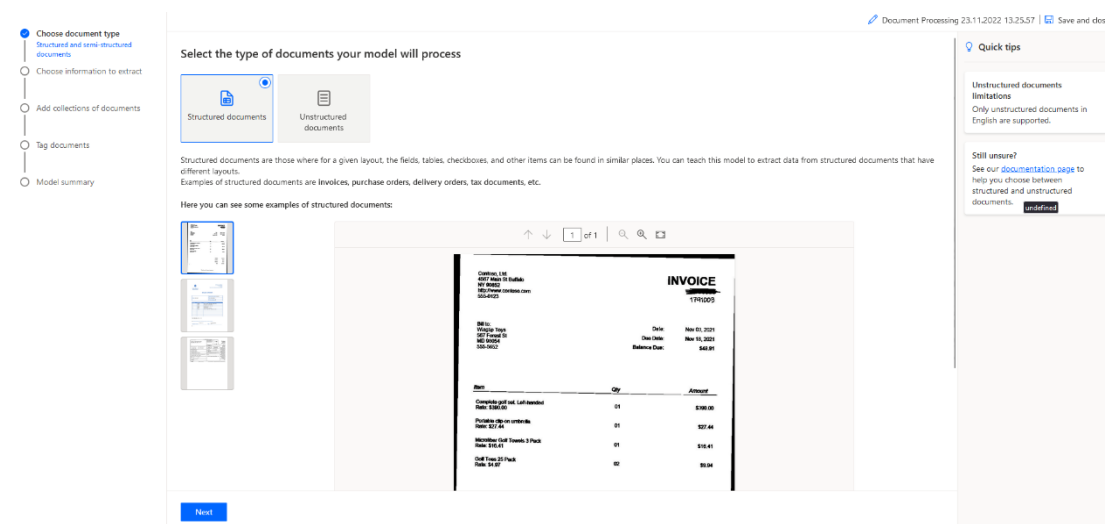
### 3.3 Ohjelmistorobotin luonti

### 3.4 Ohjelmistorobotin luominen: raporttipohjan opettaminen



(Kuva 3. Microsoft Power automaten perusnäkö) (Note: The caption text is partially cut off in the original image)

Ohjelmistorobotin luominen alkoi luomalla AI Builderilla tekoälymalli, jolle opetin raporttipohjan. Raporttipohja opetettiin tekoälymallille käyttämällä AI Builderin toimintoa Extract information from invoices, jonka tarkoituksena on opettaa PDF-pohjia tekoälylle. Extract information from invoices löytyy Power automaten verkkokäyttöliittymän sivuvalikosta kohdasta AI modules, Models.



(Kuva 4. Extract information from invoices)

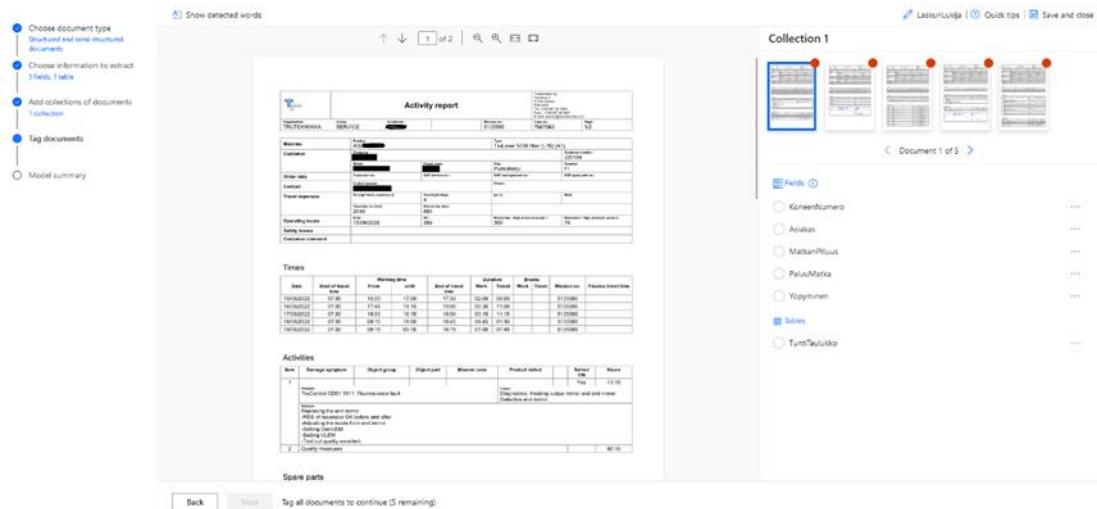
AI Builderilla voi käsitellä sekä jäsenneilyjä että jäsentämättömiä dokumentteja.

Jäsennellyillä dokumenteilla tarkoitetaan muodoltaan samanlaisia dokumentteja, kun taas jäsentämättömät dokumentit voivat olla muodoltaan monenlaisia. TruTekniikan raporttipohja on aina samanlainen, joten opinnäytetyössäni AI Builderissa oli tarpeen käsitellä vain jäsenneltyjä dokumentteja. Käytännössä raporttipohjan opettaminen tekoälylle valitsemalla AI Builderissa toiminto ”jäsennellyt dokumentit”.

Kun jäsennellyt dokumentit oli valittu, tuli valita, minkä näköistä tietoa dokumenteista otettiin. AI Builderissa on tieto on luokiteltu kolmeen parametriin: fields eli kenttä, checkbox, eli valintaruudut ja tables eli taulukot. Opinnäytetyössäni luotava ohjelmistorobotti tarvitsi viisi kenttää sekä yhden taulukon. Kenttien parametrit nimesin seuraavasti: KoneenNumero, Asiakas, MatkanPituus, PaluuMatka ja Yöpyminen. Taulukko tarvittiin kokonaistyötuntien merkitsemiseen. Taulukon nimeksi tuli TuntiTaulukko. Taulukon rivit nimesin TyoTunnit, AjoTunnit ja Paiva. Näin raporttipohjan perusta oli valmis.

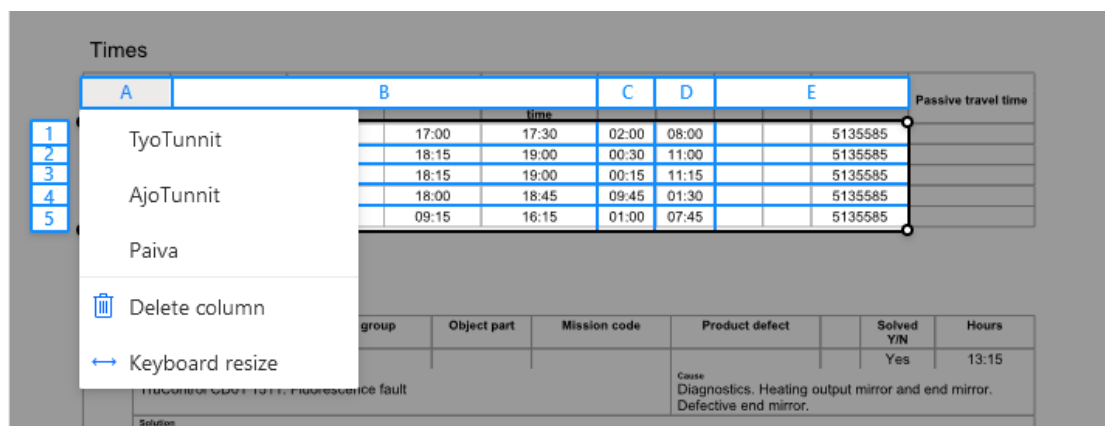
Seuraavassa vaiheessa latsin ohjelmaan viisi raporttipohjan mukaista dokumenttia. Tämä vaihe oli datan merkitsemistä tekoälyn opettamista varten. Lattuanani dokumentit maalasin parametrejä vastaavat alueet (kuva 5). Mitä enemmän ohjelmaan opetusdataa ladataan, sen suuremman tarkkuuden se varmentaa dokumenttien käsittelyyn, mutta omaan tarpeeseen viisi oli riittävä määrä.





(Kuva 5. Dokumenttien maalaaminen)

Dokumenttien maalaaminen on rakennettu AI Builderissa käyttäjäystävällisellä tavalla. Dokumentissa ylimaalaataan opetettava kohta ja valitaan, mitä parametria valinta vastaa. Ensin tein tämän kaikille ensimmäisessä vaiheessa kentiksi määrätyille parametreille.



(Kuva 6. Taulukon opettaminen)

Taulukko opetetaan pääasiassa samoin kuin kentät, mutta taulukkojen opettamiseen tulee muutama lisäaskel. Taulukko siis maalataan samoin kuin kentät, mutta taulukko jaetaan pysty- ja vaakasarakkeisiin. Tämän jälkeen ohjelma nimittää pystysarakkeet aakkosjärjestykseen, mutta tarpeellisia rivejä vastaavat kirjaimet korvataan aiemmin luodun taulukkopohjan mukaisesti. Niitä kirjaimella merkittyjä sarakkeita, joille ei itse anna tunnusta, ei ohjelmassa huomioida.

Toistin ylläkuvatut toimenpiteet kaikille viidelle dokumentille.

(Kuva 7. AI Builder on valmis käsittelemään dataa.)

Datan opettaminen tekoälymallille oli mahdollista datan merkitsemisen jälkeen. Ennen datan opettamista on tärkeää tarkistaa, että kaikki tiedot ovat kunnossa. Opettaminen on ohjelmassa automatisoitu toiminto, johon ei tarvitse itse puuttua. Opetusaika vaihtelee datan määrän ja monimutkaisuuden perusteella. Omassa työssäni viiden yksinkertaisen dokumentin opettaminen vei aikaa noin kymmenen minuuttia. Kun opetus on ajettu loppuun, kertoo ohjelma varmuutensa (confidence score) datan käsittelyn suhteen.

(Kuva 8. Varmuus käsittely)

Omassa työssäni ohjelma arvioi varmuutensa olevan 95%, mikä on toimivalle ohjelmistorobotille täysin hyväksyttävä tulos. Tässä vaiheessa Quick test -toiminnolla voi testata tekoälymallin toimivuutta syöttämällä uusia dokumentteja,

jotka tekoäly tulkitsee. Olin testin jälkeen itse tyytyväinen tekoälymallin toimintaan, joten julkaisin tekoälymallin Publish-toiminnolla, minkä jälkeen mahdollistui tekoälymallin käyttäminen ohjelmistorobotin algoritmeissa.


### 3.5 Sharepointin luonti

Sharepoint on Microsoft-organisaation web-pohjainen hallintajärjestelmä. Tässä opinnäytetyössä käytin järjestelmää kansioden ja xlsx (Microsoft Excel Open XML Spreadsheet) -tiedostojen hallintaan.

#### Luo ryhmäsivusto

Ryhmäsivusto yhdistetään Microsoft 365 ryhmään, joka antaa sivustollesi jaetun OneNote muistikirjan, ryhmän sähköpostiosoitteen ja ryhmäkalerin.

Yhdistä ryhmäsivusto Microsoft-tiimiin, jotta voit ottaa pysyvän keskustelun käyttöön valitsemalla ohjelmistopakettin siirtymisvalikosta ensin **Seuraavat vaiheet** ja sitten **Lisää reaaliaikainen keskustelu**. [Lisätietoja](#)



**Sivuston nimi \***

Sivuston nimi on käytettävissä.

**Ryhmän sähköpostiosoite \***

Ryhmän tunnus on käytettävissä.

**Sivuston osoite \***

Sivuston osoite on käytettävissä.  
https://oysamk.sharepoint.com/sites/OpinnaytetyoSivu

**Sivuston kuvaus**

**Yksityisyysasetukset**

**Valitse kieli**

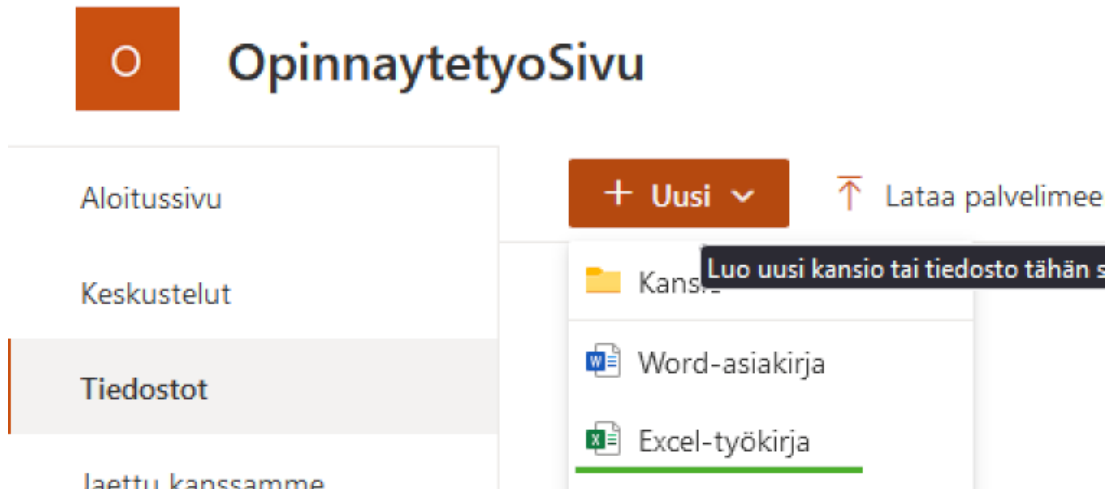
Valitse sivuston oletuskieli. Tätä ei voi muuttaa myöhemmin.

(Kuva 9. Sharepointin luonti)

Luodessa Sharepoint-sivua tein ryhmäsivun ja sivusta yksityisen.

Luotuani Sharepoint-sivun loin tyhjän excel-mallipohjan tiedostot-kansioon. Mallipohja toimi runkona, joka oli kopioitavissa tarpeen tullen.

### 3.6 Excelin luonti ja muotoilu



(Kuva 10, Excel-pohjan luonti)

	Paiva	TyoTunnit	AjoTunnit	KiloMetrit	PaluuKilometrit	HotelliYot	
1							Tuntihinta
2							Hotelliyo
3							Kilometrihinta
4							Tyotunnit yhteensa
5							Kilometrit
6							
7							Kokonais Hinta
8							- €

(Kuva 11. Excel-taulukko)

Loin excel-pohjaan kaksi taulukkoa. Ensimmäiseen taulukkoon tuli päivät ja työtunnit, jälkimmäiseen ajokilometrit ja hotelliyöt. Olisi ollut parempi, jos kaikki tiedot olisivat olleet yhdessä taulukossa, mutta raporttipohjan muodon vuoksi en saanut kaikkia tietoja toimimaan yhdessä taulukossa. Tässä työssä taulukojen nimet ovat Taulukko1 (tuntitaulukko) ja Taulukko2 (matkataulukko). Taulukko1 koostuu riveistä Paiva, TyoTunnit ja AjoTunnit. Taulukko2 koostuu KiloMetrit, PaluuKilometrit ja HotelliYot. Taulukoiden viimeiselle riville lasketaan kokonaiskulut. Kuvassa 11 näkyvät hinnat eivät vastaa TruTekniikka Oy:n hintoja, vaan ovat itse keksimäni ohjelmistorobotin demoa varten.

Sarakkeiden K1-K3 arvot ovat muuttumattomia. K4 ja K5 saavat arvonsa taulukko1 ja taulukko2 arvojen perusteella (Kuva11).

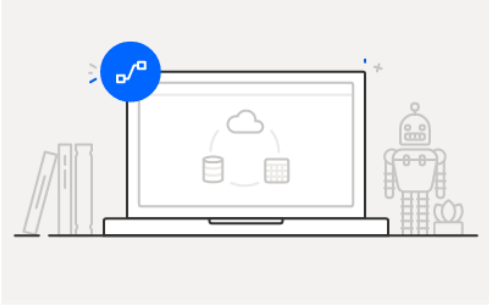
Rivi "Työtunnit yhteensä" saadaan funktiolla  $=SUMMA(B2:C999)$ , mutta tämä funktio yksin antoi kuitenkin väärän tuloksen, sillä se laski virheellisesti kellon-aikaa tavallisen yhteenlaskun sijaan. Solu piti muotoilla koodilla  $[t].mm.$ , minkä jälkeen rivi antoi oikean tuloksen.

Kilometrien laskeminen toimi heti oikein funktiolla  $=SUMMA(E2:F1000)$ , joka antoi oikein tulokseksi yhteenlasketut kilometrit. Funktiolla  $K4*K1*24$  sai työn kokonaishinnan. Funktiolla  $(SUMMA(G2:G999)*K2)$  laskettiin hotelliöiden kulut. Funktiolla  $K5*K3$  laskettiin kilometrikulut. Lopuksi kaikki summat laskettiin yhteen funktiolla  $=K4*K1*24+(SUMMA(G2:G999)*K2)+(K5*K3)$ , mistä muodostui kokonaishinta (K7). Kokonaishinta-solun muotoilin valuuttasoluksi.

### 3.7 Logiikan luonti

Aloitin ohjelmistorobotin logiikan luomisen avaamalla Power automaten. My flows -valikon kautta pääsi luomaan uuden flown, joka sisältää ohjelmistorobotin toimintalogiikan.

**Build an automated cloud flow** ✕







Free yourself from repetitive work just by connecting the apps you already use—automate alerts, reports, and other tasks.

Examples:

- Automatically collect and store data in business solutions
- Generate reports via custom queries on your SQL database

**Flow name**

**Choose your flow's trigger \*** ⓘ  
 Search or select a trigger from the list below to create a flow. (Required)

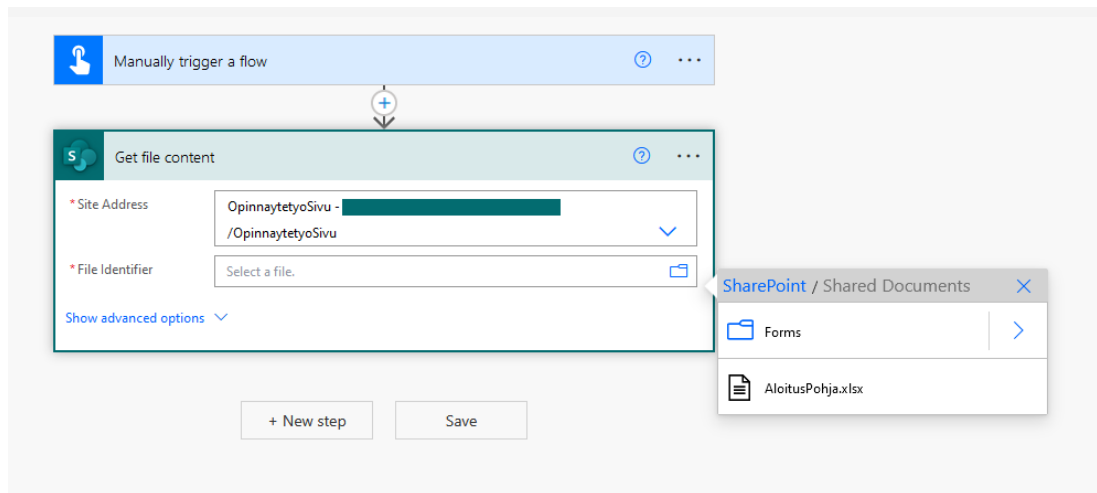
	When an email is flagged (V3) Office 365 Outlook	ⓘ
	When an email is flagged (V4) Office 365 Outlook	ⓘ
	When a new email arrives (V2) Outlook.com	ⓘ
<input checked="" type="checkbox"/>	When a new email arrives (V3) Office 365 Outlook	ⓘ
	When an email is flagged (V2) Outlook.com	ⓘ

(Kuva 12. New flow)

Power automate sisältää useita valmispohjia, jotka nopeuttavat ohjelman luomista. Oman työni tapauksessa valitsin pohjan *When a new email arrives (V3)*, mikä tarkoittaa sitä, että sähköpostin saapuminen herättää ohjelmistorobotin toimintaan.

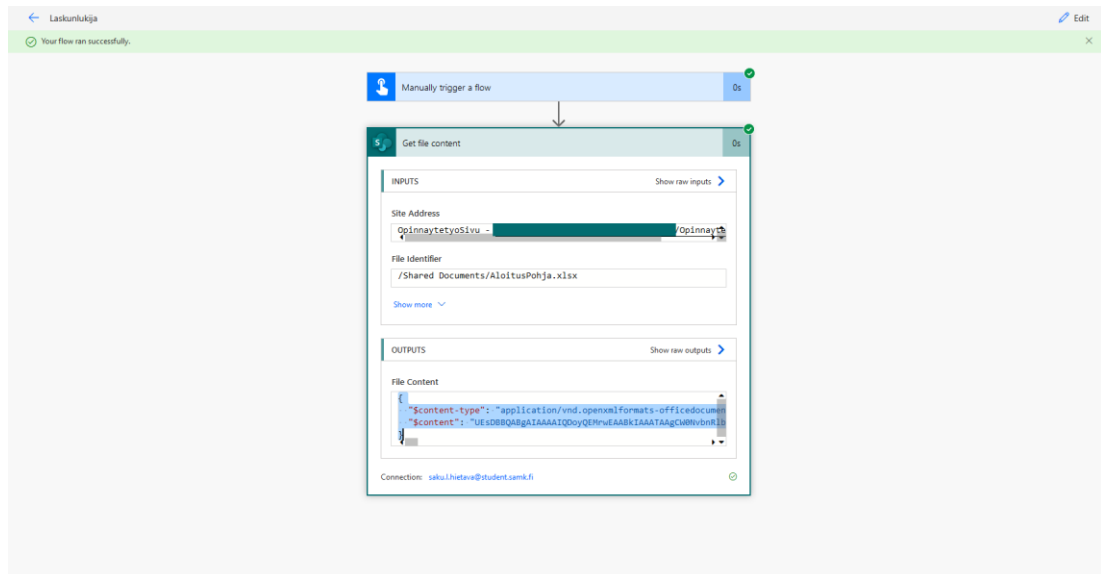
### 3.7.1 Laukaisin ja Execl-pohjan datan kopionti

Valmispohjassa laukaisimena toimi funktio *When a new email arrives (V3)*. Laukaisimella tarkoitetaan siis sitä funktiota, joka aktivoi ohjelmistorobotin toimintaan funktion ehdon täytyessä. Ensin valmispohjaan olisi pitänyt syöttää ohjelmistorobotin toiminnan herättävän sähköpostin arvot, eli siis ne tiedot, jotka olisivat täyttäneet funktion ehdot. En kuitenkaan vielä tehnyt tätä, vaan päätin toimia toisin. Valmispohja-valintani oli hieman epälooginen, mutta jätin sähköposti-triggerin jäämään työni taustalle odottelemaan. Samaan lopputulokseen olisin päässyt myös käyttämällä toisia reittejä, loogisin valinta olisi ollut valita ensin valmispohja *Manual trigger a flow*, mutta toimin nyt näin. Ihan ensin olin päättänyt käyttää Damien Birdin esittelemää tapaa excel-pohjien kopiointiin (DamoBird365, 2021).



(Kuva 13. Get file content -funktio.)

Poistin ensin valitsemastani valmispohjasta kaikki toiminnot. Seuraavaksi rakensin manuaalisen laukaisimen. Hain funktion Manually trigger a flow. Manuaalisella laukaisimella toiminnan voi käynnistää aina halutessaan käynnistämällä ohjelmistorobotin. Seuraavaksi lisäsin uuden askelen ohjelmaan: Get file content. Tällä käskyllä saadaan valitun tiedoston sisältämä data. Tässä tilanteessa haluttu data on aikaisemmin luomassani Excel-taulukossa. Polku, jota kautta tiedosto löytyy, tulee kertoa Get file content -funktiolle, joten lisäsin Site Address -valikkoon aiemmin luomani Sharepoint-sivuston URL-osoitteen. Valitsin Excel-pohjan File Identifier -valikkosta. Nyt ensimmäinen vaihe oli valmis, ja käynnistin ohjelmistorobotin.



(Kuva 14. File content)

Nyt pääsin käsiksi dataan, jonka Get file content oli saanut määrittelemästani Excel-tiedostosta. Data oli tietokoneelle ymmärrettävässä muodossa, ja kopioin sen väliaikaisesti tekstitiedostoon talteen. Kopio tuli luoda syystä, johon palaan seuraavassa luvussa 3.7.2 Sähköpostin automatisointi.

### 3.7.2 Sähköpostin automatisointi

Kun olin rakentanut laukaisimen ja kopioinut Excel-pohja datan, siirryin automatisoimaan sähköpostia. Palasin nyt 3.7.1 luvussa esittelemääni alkutilanteeseen funktioon When a new email arrives (V3), josta kuva 15.



When a new email arrives (V3)

Folder:

To:

CC:

To or CC:

From:

Include Attachments:

Subject Filter:

Importance:

Only with Attachments:

[Hide advanced options](#) ^

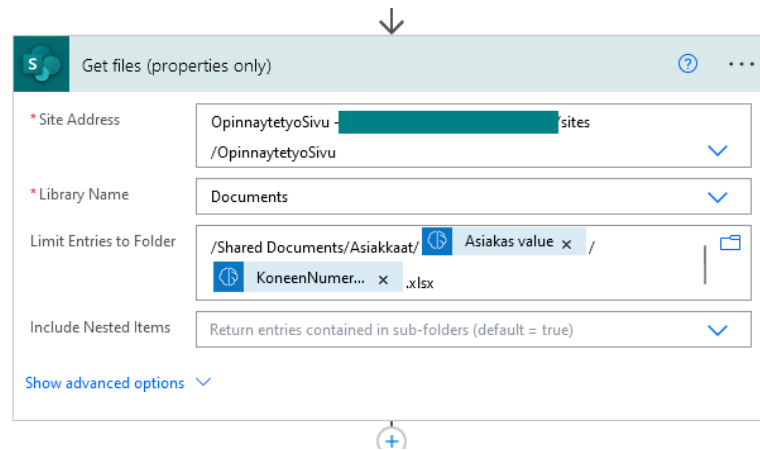
+ New step      Save

(Kuva 15. When new email arrives (V3).)

Tähän funktioon pystyi syöttämään kahdeksan parametriä. Näistä parametreista omassa työssäni tarpeellisia olivat Include Attachments, Subject Filter ja Only with Attachments. Nämä olivat tärkeitä, jotta ohjelmistorobotin tunnistamisissa sähköposteissa olisi tarvittava sisältö. Subject Filter tarkistaa sähköpostin nimen, ja Include Attachments ja Only with Attachments varmistavat, että sähköpostissa on mukana tiedosto.



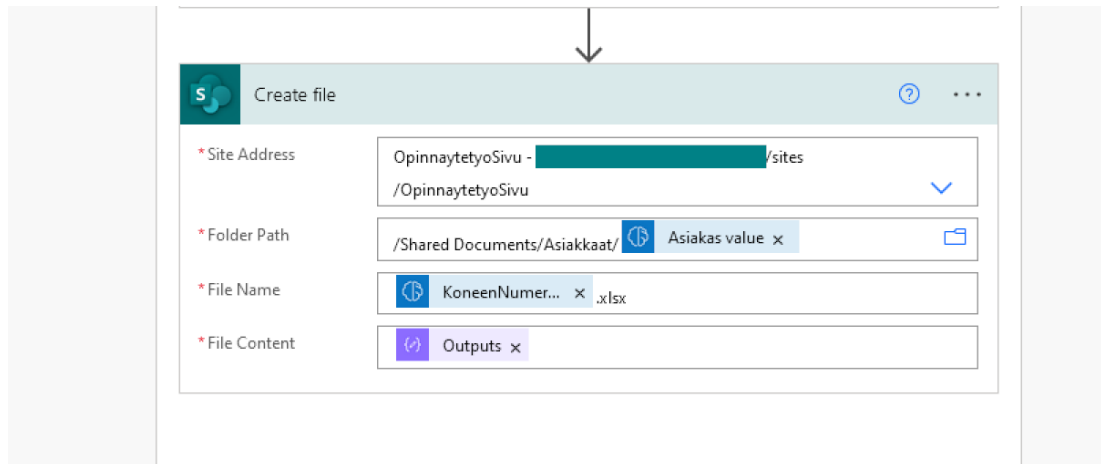
### 3.7.3 Excel-taulukon sisältö ja vuon haaroittaminen



(Kuva 17. Get file (properties only) -funktio.)

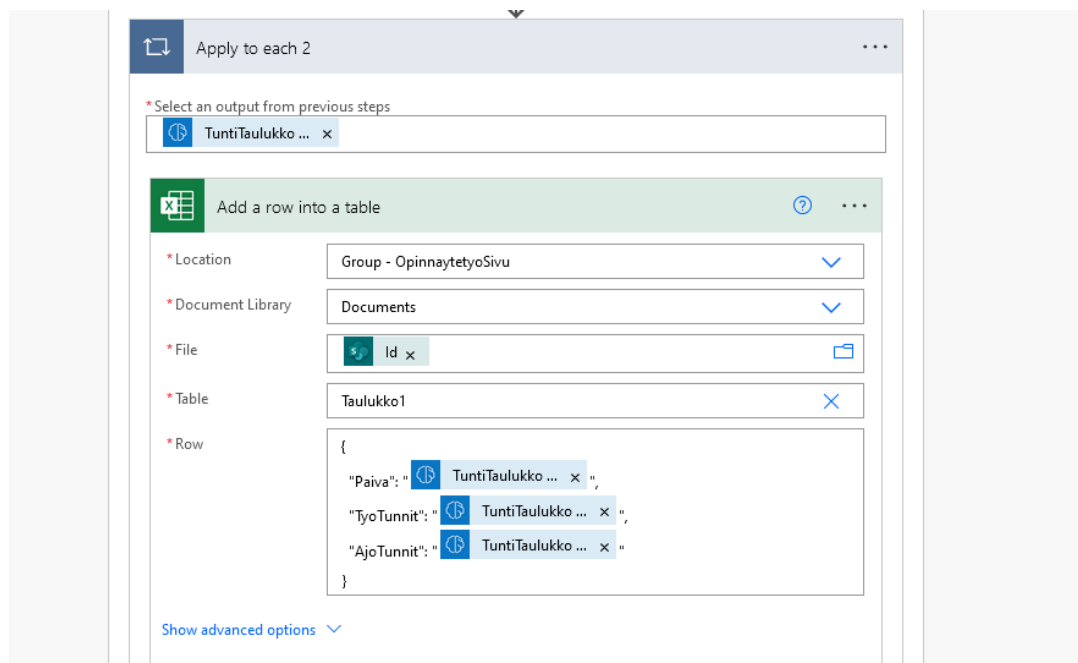
Compose-funktion jälkeen lisäsin Get file (properties only) -funktion. Funktio vastaa kysymykseen, onko haluttua tiedostoa halutussa paikassa. Tässä tapauksessa funktion kautta ohjelmistorobotti tunnistaa, löytyykö sähköpostiin saapuneessa PDF-tiedostossa ilmoitettu koneen nimi Sharepoint-kansiosta. Jos nimi löytyy, antaa funktio myöntävän vastauksen. Kun nimeä ei löydy, herjaa ohjelma virhettä, mikä aiheuttaa muutoksen logiikan kulkuun. Ensin rakentamani Compose-funktio on tarpeellinen tämän virheen ilmaantuessa. Jos halutun niminen taas tiedosto löytyi, jatkui logiikka vuon mukaisesti.

Get file (properties only) -funktiossa tiedostopolku rakennetaan tekoälyparametreilla, sillä polku on aina muuttuva.



(Kuva 18. Create file -funktio.)

Jatkoin logiikan luomista luomalla create file -funktion, jolla ohjelmistorobotti luo tiedoston. Funktiolle ilmoitetaan haluttu polku ja luotavan tiedoston nimi. Polun ja tiedoston arvot saadaan AI Builderin parametreista. Polkuna käytetään yrityksen nimeä. Luotavan tiedoston nimenä käytetään koneen nimeä. Luodun tiedoston datan sisältö saadaan kutsumalla luvun alussa luomani Compose-funktion sisältö luotuun tiedostoon (Outputs-arvo).



(Kuva 19. Excel-tilukointi)

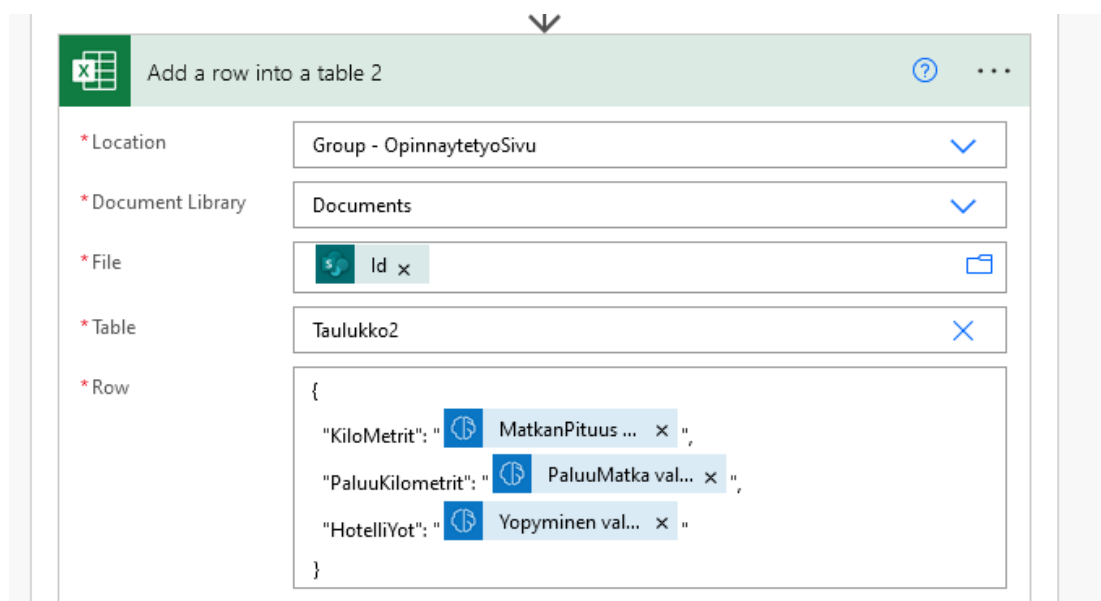
Seuraavassa vaiheessa rakensin Apply to each -silmukan, joka käy läpi kaikki tuntitaulukon sisältämät rivit. Silmukan sisään lisäsin funktion Add a row into a

table. Jotta funktio tietää, mitä Excel-taulukkoa lähtee käsittelemään, tuli tieto ilmoittaa funktiolle. Excel-tiedosto voidaan osoittaa Create file -funktion sisältämällä arvolla ID, jolloin polkua ei tarvitse erikseen kertoa. Add a row into a table -funktioille tuli ilmoittaa taulukko, johon halusin lisätä rivejä. Koska Compose-funktio loi täydellisen kopion Excel-taulukosta, oli mahdollista osoittaa alussa luotuihin taulukoihin ilman, että taulukkoa piti erikseen rakentaa.

Halusin käsittelemään Taulukko1-taulukon, jonka lisäsin Add a row into a table -funktioon. Ohjelmistorobotille tuli ilmoittaa taulukon rivit sekä näiden sisältämät arvot, joita olivat AI Builderin arvoista Paiva, Tyotunnit ja AjoTunnit. Rivien arvot kerrottiin koodilla, jonka ensimmäinen parametri ilmoitti solun, jälkimmäinen parametri solun arvon:

```
{
  "Paiva": "TuntiTaulukko Paiva",
  "TyoTunnit": "TuntiTaulukko TyoTunnit",
  "AjoTunnit": "TuntiTaulukko AjoTunnit"
}
```

Tämän jälkeen lisäsin rivit hotelliyöt ja ajokilometrit, joiden arvot oli AI Builder tunnistanut. Näitä arvoja ei tarvinnut käsitellä silmukan läpi, sillä rivejä oli mahdollista olla näillä arvoilla vain yksi.



(Kuva 20. Kilometrit ja hotelliyöt)

Muuten funktio on identtinen, mutta Taulukko1 sijasta valitaan Taulukko2.

Koodi on seuraava:

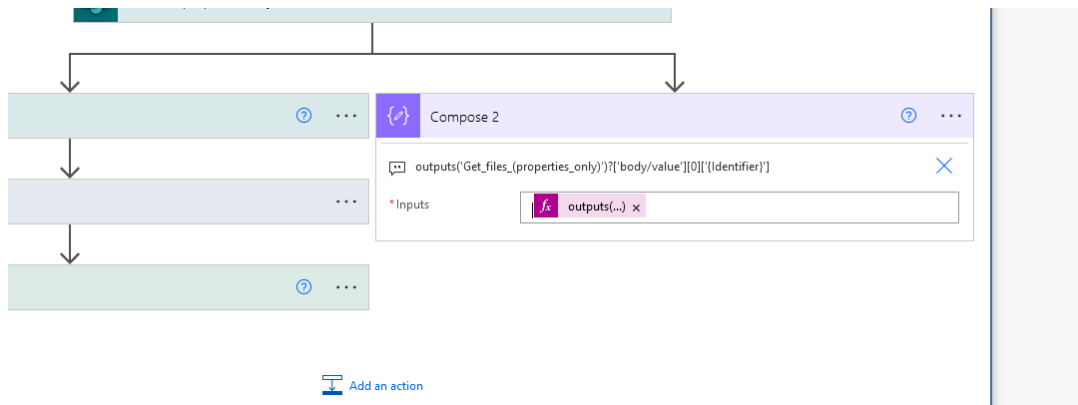
```
{  
  "KiloMetrit": "MatkanPituus value",  
  "PaluuKilometrit": "PaluuMatka value",  
  "HotelliYot": "Yopyminen value"  
}
```

Tähän mennessä olin tehnyt ohjelmistorobotin logiikasta toisen puolen valmiiksi. Toimintarakenne tähän mennessä oli:

1. PDF-tiedoston sisältämä sähköposti saapuu ja sisältää Subject Filterissä säädetyt parametrit, eli sähköpostin nimen.
2. AI Builder lukee PDF-tiedoston ja tallentaa valitut kentät käytettäviin parametreihin.
3. Get files (Properties only) tarkistaa onko PDF-tiedostossa nimettyä konetta Sharepoint-tiedostoissa.
4. Tiedostoa ei löydy, ja robotti luo uuden tiedoston ja täyttää Taulukko1 ja Taulukko2 halutuilla arvoilla.

Tässä vaiheessa logiikka tarkisti, löytyykö Sharepointista konetta, jonka nimi PDF-tiedostosta löytyi, mutta rakentaisi aina uuden tiedoston vanhan päälle ja poistaisi huoltojen historian.

Logiikkaan tuli siis seuraavaksi rakentaa haara, eli vaihtoehtoinen logiikan kulku niitä tilanteita varten, jolloin huollettavan koneen taulukkotiedosto löytyisi jo Sharepoint-kansiosta. Haaran tuli sijoittua Get File (Properties only) -funktion jälkeen. Lisäsin haaran valikosta toiminnolla Add a parallel branch, ja logiikka-vuossa uusi haara sijoittui oikealle puolelle.

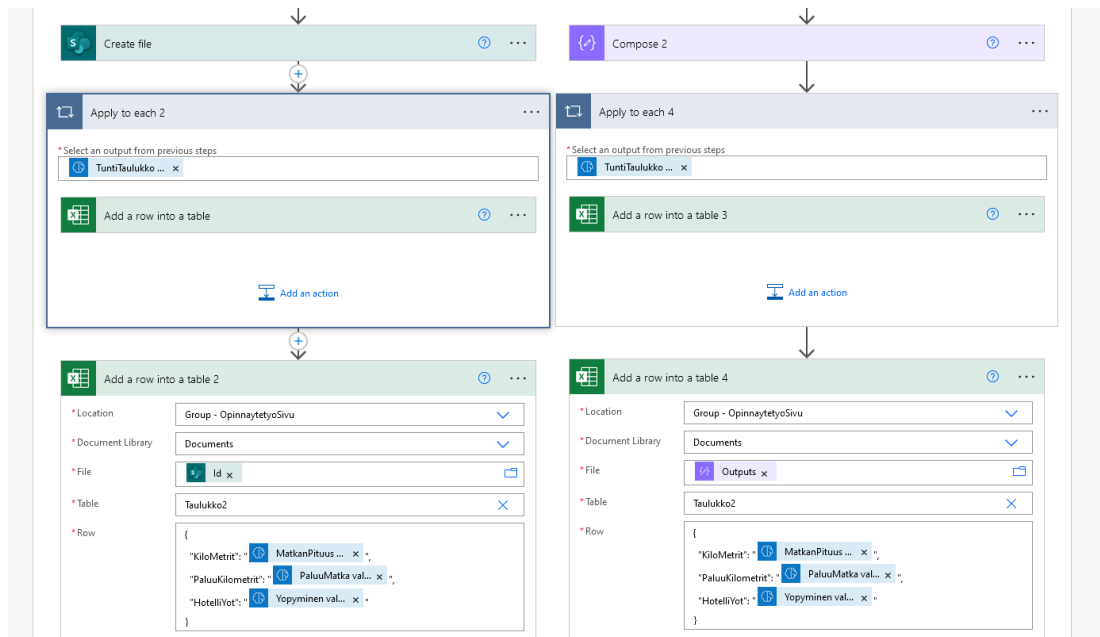


(Kuva 21. Get file (Properties only) -kysely)

Luotuun haaraan lisäsin Row a into a table -funktion datan lisäämiseksi jo luotuun taulukkotiedostoon. Ongelma syntyi, kun yritin osoittaa funktiolle tiedostopolkua, jossa taulukkotiedosto sijaitsi. Ohjelma kertoi, että syöttämäni polkua ei ollut olemassa. Virhekoodin ilmoittama polku oli kuitenkin halutun näköinen, ja kopioimalla ja käyttämällä tätä staattista polkua ohjelma löysi uuden yrityksen jälkeen halutun taulukkotiedoston. Minun tuli siis keksiä toinen tapa muuttuvan tiedostopolun kertomiseen funktiolle. Nyt sijoitin aiemmin tehtyyn haaraan Compose-funktion (s. 26). Compose-funktion kyselyn avulla sain tietoon aiemman funktion Get file properties only -outputin ID:n, jonka avulla pystyin osoittamaan Excel-funktiolle oikean tiedoston. Tämä menetelmä toimi.

ID-kyselyyn käyttämäni koodi: `output('Get file (Properties only)')?['body/value'][0]['Identifier']` (dremille, 2022).

Tästä eteenpäin loin logiikan samalla tavalla kuin ensin tekemäni vasemmalle sijoittuvan haaran, mutta oikealla puolella käytin tunnisteena Copose-funktion outputia Create file -funktion ID:n sijaan.

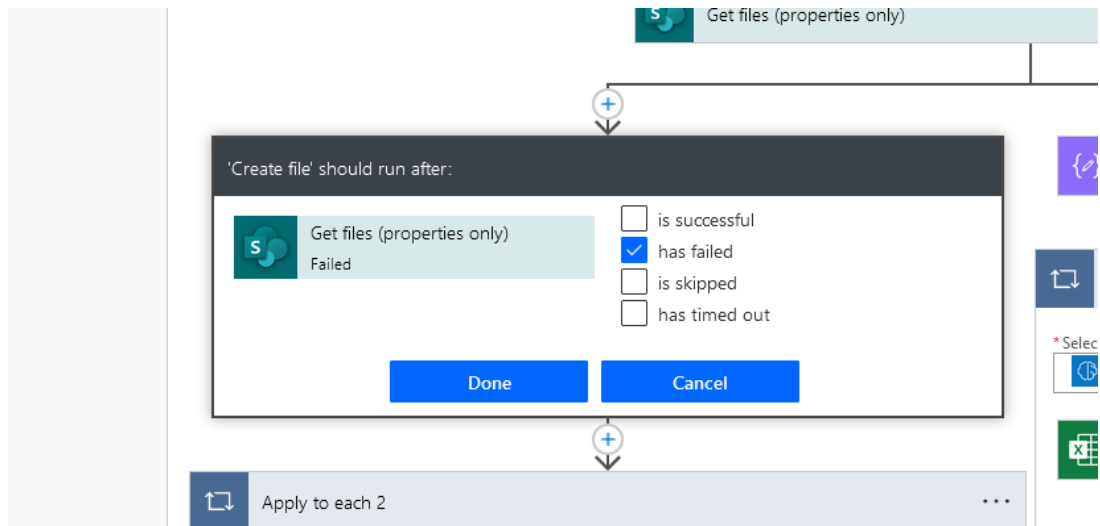


(Kuva 22. Logiikan haaroitus)

Nyt logiikka oli muuten valmis, paitsi haaroituskohdasta puuttui ehtoisuus. Logiikalle tuli vielä kertoa, missä tilanteessa oli seurattava vasemmanpuoleista, milloin oikeanpuoleista vuota. Ehtoisuuden rakensin virrehavainnon kautta.

Vuokaavion oletusarvona on, että jokainen funktio antaa hyväksytyyn vastauksen. Tilanteessa, jossa Get file (properties only) ei löydäkään kyselyn tiedostoa, tapahtuu virhe ja logiikka pysähtyy. Catch-toiminnon avulla ohjelma kuitenkin pystyy jatkamaan toimintaa. Funktioblokeissa olevaan ehtoon määritellään, missä tilanteissa funktioblokki suoritetaan. Power automate -ympäristössä vuokaavion kulun mahdollisuuksia on neljä, joista omassa työssäni tarvitsin kahta: "is successful" ja "has failed" (kuva 23).





(Kuva 23.)

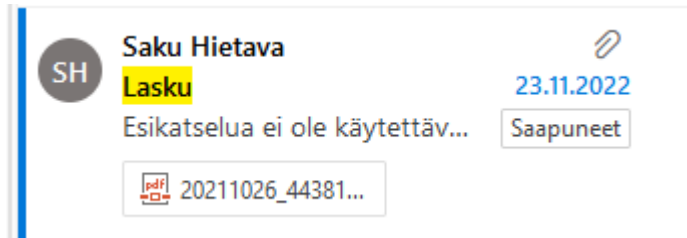
Minun tuli laittaa Create filen suoritusehtoihin ainoaksi valinnaksi ”has failed”, jolloin funktio suoritettiin vain, jos edellinen funktio Get files (properties only) ei löytänyt haluttua tiedostoa. Compose-funktion suoritusehtoihin ei tarvinnut tehdä muutoksia, sillä oletuksena oli, että edellisen funktion suorittamisessa ei tapahtunut virhettä. Nyt logiikka osasi tunnistaa, kumman vuon mukaisesti tuli kulkea.

Power automate sisältää myös If else -kyselyn, jota tässä tilanteessa olisi voinut hyvin käyttää, mutta näin tämän hieman turhaksi, sillä logiikan suorittamisen jälkeen tuleva raportti ilmoittaisi virheen sattuneen joka tapauksessa niissä tilanteissa, joissa uusi taulukkotiedosto luotaisiin. If-funktioblokki toisi siis vain yhden lisäpalikan vuohon ja hidastaisi toimintaa.

#### 4 OHJELMAN TOIMINTA KÄYTÄNNÖN TILANTEESSA

Nyt ohjelmistorobotti oli valmis demoa varten, ja seuraavaksi lähdin demoamaan ohjelmistorobotin toimintaa keksityssä tilanteessa.

Ensimmäiseksi lähetin sähköpostin Outlook-osoitteeseen, johon ohjelmistorobotti oli kiinnitetty. Sähköpostiin olin laittanut PDF-liitteen ja otsikoin viestin nimellä ”Lasku”, jotta sähköposti toimisi ohjelmistorobotin laukaisimena.



(Kuva 24. Lähetetty sähköposti)

Sähköpostin saavuttua ohjelmistorobotti lähti käsittelemään saapunutta viestiä.

28-day run history <span>ⓘ</span>		
Start	Duration	Status
Nov 23, 03:09 PM (18 sec ago)	00:00:18	Running
Nov 23, 02:42 PM (27 min ago)	48 ms	Test succeeded

(Kuva 25. Ohjelmistorobotin prosessointi)

Tiedostot > Asiakkaat > [Redacted]

Nimi	Muokattu	Muokkaaja	+ Lisää sarake
SOS [Redacted].xlsx	Muutama sekunti sitten	Hietava Saku	

(Kuva 26. Luotu Excel-taulukko)

Työtunnit	Ajotunnit	KiloMetrit	PaluuKilometrit	HotelliYot	Tunthinta	
6:00	2:15	145	160	1	40	
5:15	2:45				150	
7:00	1:00				0,1	
					Työtunnit yhteensä	24,15
					Kilometrit	305
					Kokonais Hinta	1 150,50 €

(Kuva 27. Excel- taulukon (kuva 26) sisältö)

Nyt lähetin ohjelmistorobottiin kinnitettyyn sähköpostiin toisen viestin, jonka liitetiedosto oli täsmälleen sama kuin ensimmäisessä viestissä.

TyoTunnit	AjoTunnit	KiloMetrit	PaluuKilometrit	HotelliYot	Tuntihinta
6:00	2:15	145	160	1	40
5:15	2:45				150
7:00	1:00	145	160	1	0,1
6:00	2:15				Tyotunnit yhteensa
5:15	2:45				Kilometrit
7:00	1:00				Kokonais Hinta
					2 301,00 €

(Kuva 28. Excel-taulukon sisältö toisen sähköpostin jälkeen)

Start	Duration	Status
Nov 23, 03:12 PM (1 min ago)	00:00:31	Succeeded
Nov 23, 03:09 PM (4 min ago)	00:00:34	Failed
Nov 23, 02:42 PM (31 min ago)	48 ms	Test succeeded

(Kuva 29. Ohjelmistorobotin tila kahden sähköpostin jälkeen)

Kuvassa 29 näkyvä ”failed”-virhe antaa hieman harhaanjohtavaa informaatiota, sillä se viittaa Get file -funktion ilmoitukseen siitä, ettei kyselyä tiedostoa löytynyt Sharepoint-kansiosta, jolloin funktio laukaisi virheen ja loi uuden tiedoston. Tämä virheviesti olisi näkynyt, vaikka haaroitus olisi rakennettu if-funktion kautta.

Demon perusteella totesin ohjelman toimivan aluksi luomani suunnitelman mukaan ja vastaavan TruTekniikan tarpeisiin.

## 5 LUOMANI OHJELMISTOROBOTIN KEHITYS JA MAHDOLLISET VAARAT

Luomassani ohjelmistorobotissa on vielä kehitettävää. Sharepoint-tiedoston nimi luomassani ohjelmistorobotissa on yrityksen nimi. Windows on kuitenkin tarkka tiedostojen nimistä, ja täten Sharepoint-tiedostojen nimien on noudatettava Windowsin sääntöjä. Jos PDF-tiedoston sisältö rikkoo jotakin näistä säännöistä, ei ohjelmistorobotti pysty toimimaan. Nimissä kiellettyjä merkkejä ovat ~ # % & \* { } \ : < > ? / + | ” (Anthony, 2018). Tähän olisi mahdollista kuitenkin rakentaa ratkaisu. Ratkaisuksi sopisi funktio, joka poistaisi PDF-tiedostosta

löytävän yrityksen nimen sisältämät kielletyt merkit. Esimerkiksi jos yrityksen nimi olisi Firma\firma, joka itsessään tarkoittaisi Windowsille, että Firma-nimisestä kansioista löytyisi firma-niminen kansio, käyttäisi ohjelmistorobotti yrityksestä nimeä Firmafirma, jolloin robotti kykenisi suorittamaan toimintonsa. En kuitenkaan luonut tällaista funktiota ohjelmistorobottiin välttääkseni työn paisumisen.

Luomaani ohjelmistorobottiin liittyy tietoturvariski kuten aina uusia järjestelmiä käyttöönottaessa. Vaikka suurin osa tietomurroista tapahtuu ihmisvirheiden ja sosiaalisen manipuloinnin kautta (Rosencrance ja Bacon, 2021), on tärkeää pitää ohjelmistorobotteja silmällä. Tietoturvasyistä tulee salata, mitkä sähköpostiosoitteet ja avainsanat on täytyttävä, jotta ohjelmistorobotti lukee sähköpostin ja herää käyntiin. Jos nämä tiedot vuotaisivat, voisi joku hyvin tehdä kiusaa ja tukkia ohjelmistorobotin lähettämällä turhia raportteja. Ns. vaarallisemman hakkeroinnin riskit luomassani ohjelmistorobotissa näen kuitenkin pieninä, sillä AI Builderin olisi vaikea tunnistaa Excel taulukkoon tarkoitettua haitallista makroa, sillä se eroaisi niin paljon opetettujen tiedostojen ulkonnäöstä.

## 6 YHTEENVETO

Saavutin työssäni halutun tavoitteen, ja sain luotua ohjelmistorobotin, joka toimi toivotulla tavalla. Aikaa, joka meni huoltotakuu-taulukon manuaaliseen täyttöön, ei olla laskettu, mutta lienee selvää, että ohjelmistorobotti suorittaa työn huomattavasti nopeammin. Kuvan 29 perusteella näkyy, että ohjelmistorobotti suorittaa toiminnon 30-35 sekunnin kuluessa. Ihminen ei voi suoriutua nopeammin. Nopeus ei ollut kuitenkaan ainoa motiivi ohjelmistorobotin luomiseen, vaan tarkoituksena oli ensisijaisesti automatisoida ihmiselle tylsä ja pitkäväteinen työ. Tässä ohjelmistorobotti onnistui. TruTekniikka Oy:n järjestelmässä ohjelmistorobottia ei olla vielä otettu käyttöön, mutta tulevaisuudessa

se aiotaan ottaa testikäyttöön ja toivottavasti myöhemmin ihan varsinaiseen käyttöön asti.

## LÄHTEET

Anthony. (3.12.2018.) File naming restrictions in sharepoint. <https://sharepointstuff.com/2018/12/03/file-naming-restrictions-in-sharepoint/>.

Bratt, M. K.. (1.3.2021.) What are the advantages and disadvantages of RPA?. <https://www.techtarget.com/searchcio/feature/What-are-the-advantages-and-disadvantages-of-RPA>.

DamoBird365. (3.8.2021.) Create a new Excel File in Power Automate and dynamically populate with a Table and Rows #Excel [video]. YouTube. [https://www.youtube.com/watch?v=RB\\_ySjhm9Sg&t=139s&ab\\_channel=DamoBird365](https://www.youtube.com/watch?v=RB_ySjhm9Sg&t=139s&ab_channel=DamoBird365).

dremille. (19.1.2022.) Get File content error using Get files (properties only). Haettu 11.12.2022 osoitteesta <https://powerusers.microsoft.com/t5/General-Power-Automate/Get-File-content-error-using-Get-files-properties-only/td-p/1424528>.

IBM Cloud Education. (22.10.2020.) What is Robotic Process Automation (RPA). Haettu 1.12.2022 osoitteesta <https://www.ibm.com/cloud/learn/rpa>.

Lawton, G.(23.4.2021.) Robotic process automation (RPA). TechTarget. Haettu 1.12.2022 osoitteesta <https://www.techtarget.com/searchcio/definition/RPA>.

Microsoft. (2023.) Power Automate Pricing. Haettu 1.12.2022 osoitteesta <https://powerautomate.microsoft.com/en-us/pricing/>.

Microsoft. (27.09.2022.) AI Builderin yleiskatsaus. Haettu 1.12.2022 osoitteesta <https://learn.microsoft.com/fi-fi/ai-builder/overview#release-status>.

Robomotion (4.4.2021.) History of Robotic Process Automation (RPA). <https://www.robomotion.io/blog/history-of-rpa/>.

Rosencrance, L., Bacon, M. (2021) .Tech Target. Social engineering. Haettu 1.12.2022 osoitteesta <https://www.techtarget.com/searchsecurity/definition/social-engineering>.

Trumpf. Responsibility. Haettu 1.12.2022 osoitteesta [https://www.trumpf.com/en\\_INT/company/responsibility/](https://www.trumpf.com/en_INT/company/responsibility/).

TruTekniikka Oy. Haettu 14.1.2023 osoitteesta. <https://www.trutek.fi/yritys/>.

UiPath. (2022.) Robotic Process Automation. Haettu 1.12.2022 osoitteesta <https://www.uipath.com/rpa/robotic-process-automation>.