



Kannattavan ohjelmistorobotin luominen mikro- oyrittäjän asiakaslaskutukseen

Ilya Kolyasin

2023 Laurea



Laurea-ammattikorkeakoulu

Kannattavan ohjelmistorobotin luominen mikroyrittäjän asiakas- laskutukseen

Ilya Kolyasin
Liiketalouden koulutusohjelma
Opinnäytetyö
Kesäkuu, 2023

Ilya Kolyasin

Kannattavan ohjelmistorobotin luominen mikroyrittäjän asiakaslaskutukseen

Vuosi

2023

Sivumäärä

44

Tämän kehittämistyön tavoitteena oli auttaa mikroyrittäjän ratkaista haasteen asiakaslaskuttamisessa. Samalla on pyritty tehostaa yrityksen liiketoimintaa, parantaa asiakkaiden tyytyväisyyttä ja vähentää työntekijän ajankulutusta tähän toimintaan.

Kehittämistyössä selvitettiin, kuinka mikroyrittäjän liike-elämä helpottuu ohjelmistorobotin käyttöönottamisen jälkeen. Kehittäjänä kartoitin robotin käyttökohteen, suunnittelin sen toteuttamista, valmistin sen alusta loppuun asti. Kehitystyön aikana toteutin RPA_Invoice_Robot-ohjelmistorobotin UiPath Studion ohjelmistorobottiikan alustalla. Tämä robotti hoitaa asiakaslaskutuksen käyttäjän puolesta.

Aineistonkeruu toteutettiin haastattelemalla toimeksiantajaa videoimalla sen asiakaslaskuttamisen työprosessia. Videoiden analysoimisen aikana mitattiin asiakaslaskuttamisen vaiheiden suoritusajat. Hyödynsin niitä jatkossa, kun arvioin ja argumentoin ohjelmistorobotin käyttöä mikroyrityksen asiakaslaskutusprosessissa.

Ohjelmistorobotin valmistamisen jälkeen laskettiin ohjelmistorobotti-investointikannattavuutta. Lopputuloksena saatiin laskutusrobotti, joka nopeutti asiakaslaskutusprosessin ainakin seitsemän kertaa. Viimeisessä luvussa esitettiin jatkokehitysmahdollisuuksia asiakaslaskutusrobotille. Myös siellä havaittiin uusia ohjelmistorobottiikalle sopivia käyttökohteita mikroyrityksen liiketoimintaprosesseissaan.

Ilya Kolyasin

Creating a Profitable Software Robot for Microentrepreneur's Customer Invoicing

Year

2023

Pages

44

The purpose of this development work is to help microentrepreneurs solve the challenge of customer invoicing. At the same time, efforts were made to make the company's business more efficient, improve customer satisfaction and reduce the employee's time spent on this activity.

The development work investigated how the business life of a microentrepreneur becomes easier after the introduction of a software robot. As a developer, I mapped the robot's intended use, planned its implementation, and manufactured it from start to finish. During the development work, I implemented the RPA_Invoice_Robot software robot on the UiPath Studio software robotics platform. This robot handles customer invoicing on behalf of the user.

The data collection was carried out by interviewing the client and videotaping its customer invoicing work process. During the analysis of the videos, the execution times of customer invoicing phases were measured. I will use them in the future when I evaluate and argue the use of a software robot in the customer invoicing process of a microenterprise.

After the software robot was manufactured, the profitability of the software robot investment was calculated. The end result was a robot that sped up the customer invoicing process at least 7 times. In the last chapter, further development possibilities for the customer invoicing robot were presented. There, too, new applications suitable for software robotics were discovered in the business processes of the microenterprise.

Keywords: Customer Invoicing, Microenterprise, Robotic Process Automation, ROI, RPA, Ui-Path

Sisällys

1	Johdanto.....	6
1.1	Globalisaatio ja digitalisaatio	7
1.2	Toimeksiantajan esittely	7
2	Ohjelmistorobotiikka mikroyrittäjän asiakaslaskutuksessa	8
2.1	Asiakaslaskutuksen automatisoinnin vaihtoehdot	10
2.2	Ohjelmistorobotti investoinnin näkökulmasta	10
3	Kehittämistyön prosessi	11
3.1	Esiselvitys	12
3.1.1	Teemahaastattelu	12
3.1.2	Havainnointi	14
3.2	Määrittely	14
3.2.1	Benchmarking	15
3.2.2	Valmiit laskutustalot	15
3.2.3	Ohjelmistorobotiikan asiantuntijat	16
3.2.4	Ohjelmistorobotiikan alustat	16
3.2.5	UiPath Studuo-työkaluna	17
3.3	Pilottirobotti.....	20
3.3.1	Tietokantojen luominen	20
3.3.2	Pilottirobotin toteutus.....	21
3.3.3	Alustava kustannuslaskenta	24
3.4	Tuotantorobotti	25
3.4.1	Tietokantojen kehittäminen	26
3.4.2	Alirobottien luominen.....	28
3.4.3	Päärobotin toteutus.....	31
3.4.4	Viimeistely.....	35
3.4.5	Kustannuslaskenta	35
4	Tulokset	36
5	Pohdinta ja jatkokehitykset	38
	Lähteet.....	39
	Kuviot	42
	Liitteet	43

1 Johdanto

Tämän kehittämistyön tavoitteena on luoda taloudellisesti kannattavan ohjelmistorobotin mikroyrittäjän asiakaslaskutukseen. Tässä työssä samalla selvitän, kuinka mikroyrittäjän liike-elämä helpottuu, kun hän ottaa käyttöön yhden asiakaslaskutukseen suunnatun ohjelmistorobotin. Kehittämistyön tekijänä ja ohjelmistorobotin kehittäjänä kartoitan robotin käyttökohteen, suunnittelen sen toteuttamista, valmistan sen ja arvioin sitten mitä, missä ja milloin onnistuu sekä missä olisi vielä kehityskohteita.

Ensin kävimme keskustelu minun toimeksiantajani kanssa, jossa hän kertoi omista haasteistaan. Haastattelun aikana selvitin, kuinka tällä hetkellä yritys laskuttaa asiakkaitaan. Yhdessä me katsoimme läpi mahdollisia alternatiiveja sujuvammalle ja ketterämmälle laskutukselle. Asiakaslaskutuksen automatisointiin on olemassa monia vaihtoehtoja. Tämän projektin aikana käsiteltiin kolmea teknisesti erilaista ratkaisua, joista valittiin yhden. Tarkastelussa olivat lukuisat valmiit laskutustalot, ohjelmistorobotiikan asiantuntijat ja ohjelmistorobotiikan alustat. Ennen varsinaisen ohjelmistotyön alkamista videoin kameralla, miten yrittäjä suorittaa asiakaslaskuttamista ja litteroin tiedot tästä prosessista analyysitaulukkoon. Analysoinnin aikana lasken, kuinka monta työtuntia tästä manuaalista työstä kerääntyy yhden kuukauden aikana.

Näiden tietojen perusteella loin ohjelmistorobotin. Sitten laskin kaikkien väliprosessien yhteistyötuntien määrän manuaalisessa asiakaslaskutuksessa ja vertasin sen ohjelmistorobotin laskutukseen käyttämään aikaan. Tällä aineistolla laadin ohjelmistorobotti-investoinnin kannattavuuden arvioinnin.

Asiakaslaskutuksen robotin toimivuutta, sen tuomia hyötyjä ja mahdollisia haittoja käsittelemme opinnäytetyön viimeisessä luvussa pohdinta ja jatkokehitykset. Lisäksi siinä mietin jatkokehitystä asiakaslaskutusrobotille ja ohjelmistorobotiikalle muita sopivia käyttökohteita mikroyrittäjän liiketoiminnassaan.

Tämän kehittämisprojektin teoriaperustana toimii Taullin *Robotic The Process Automation Handbook*. Myös kehittämistyön tukena käytin pääasiassa *Älykäs taloushallinto*-kirjan. Lisäksi hyödynsin toimeksiantajan omia sisäisiä tiedostoja ja ohjelmistorobotiikasta lukuisia julkaituja tutkimuksia. Tämän toiminnallisen kehittämistyön tavoitteena oli luoda taloudellisesti kustannustehokkaan ja ammattimaisen projektin toteutus. Tämä oli uusi tietotekninen työkalu yrityksen tarpeeseen ja työelämän toimeksiantoon. Toimeksiantajan näkökulmasta ohjelmistorobotiikan käyttö yrityksen rutiiniprosessien suorittamisessa on merkittävä kilpailukeino. Työprojektin edistämisen aikana kaikki mahdolliset lähteet lisätään lähdeluetteloon.

Aiemman kokemuksen ohjelmointirobotiikasta sain työkokemukseni kautta sekä muutamista ohjelmointirobotiikan kursseista. Tämän lisäksi harrastan Java- ja Kotlin-ohjelmointikielisen mobiilisovelluksien kehittämistä.

1.1 Globalisaatio ja digitalisaatio

Globalisaation ja digitalisaation myötä nykymaailman liiketoiminta on paljon monimutkaisempi. Globaalitaloudessa suuret yritykset voivat liikkua rajojen yli ja vallata uusia markkina-alueita toisissa maissa. Tästä seuraa se, että suurten yritysten eli monopolistien on helpompi ottamaan käyttöön uusimmat ja kalleimmat teknologiat omassa liiketoiminnassaan. Yksi näistä uusista teknologioista on ohjelmistorobotiikka. Ohjelmistorobotiikka on suhteellisen uusi teknologia ja siksi sen asiantuntijoiden veloittama hinta on korkea. Useimmilla pienillä ja mikroyrityksillä ei ole varaa liian kalliisiin teknologioihin. Sitä paitsi niiden käyttöönotto yleisesti vaatii suuren määrän valmistelutöitä. Teknologioilla olisi mahdollista kasvattaa merkittävästi markkinaosuuttaan muiden samalla alalla oleviin kilpailuyrityksiin verrattuna. (Euroopan komissio 2017, 10-12.)

Digitalisaatio on mullistanut maailman jo kauan sitten, eikä tämä enää ole pelkkä megatrendi vaan arkipäiväasia. Digitalisaatiolla tarkoitetaan yrityksen toimintamallien, teknologian sekä prosessien kokonaisvaltaista muutosta kohti digitalisaation vaatimaa mallia. Digitalisaatio koostuu pienistä paloista, joiden avulla yrityksen liiketoimintaprosesseja tehostetaan ja samalla ajetaan yrityksen kilpailuetua. Yksi digitalisaation osa-alueista on ohjelmistorobotiikka, joka on noussut viime vuosina omana trendinä. Yleensä tämä käsite on vieras monille ihmisille, vaikka sanasta robotiikka olisi jo jonkunlaista käsitystä. Robotisaatiosta mieleen nousee esimerkiksi autoja valmistavat koneet eli robotit. Paitsi fyysisiä konerobotteja on myös ohjelmistorobotteja, jotka ajavat toimistoissa rutiinimaisia prosesseja arkipäiväisessä työelämässä. Ne käyttävät tietojärjestelmiä täysin samalla tavalla normaalin käyttäjän tapaan. Ohjelmistorobotiikka, eli Robotic Process Automation lyhyesti RPA, itsessään on rutiiniprosessien automatisoimista ohjelmistorobotin avulla. Lähtökohtaisesti ohjelmistorobotiikka käyttää prosessien suorittaessa samoja ohjelmia, joita käyttäisi ihminenkin. (Euroopan komissio 2017, 10-12.) Ohjelmistorobotin käyttöönoton aikana pitää ottaa huomioon EU:n yleisen tietosuojasetuksen (General Data Protection Regulation, GDPR). Tietosuojan sekä tietoturvan tärkeys on korostunut merkittävästi viime vuosien aikana erityisesti Euroopan unionin alueella. (Digital Workforce 2017.)

1.2 Toimeksiantajan esittely

Mikroyritykseksi määritellään yritys, jonka palveluksessa on alle 10 työntekijää ja sen vuosiliikevaihto on alle kaksi miljoonaa euroa tai taseen loppusumma enintään kaksi miljoonaa euroa. Tämän lisäksi sen täytyy olla riippumaton, eli sen pääomasta tai äänivaltaisista

osakkeista 25 prosenttia tai enemmän ei ole toisen yrityksen omistuksessa tai sellaisten yritysten yhteisomistuksessa. (Tilastokeskus 2022.)

Vaikka ei pienien yritysten nimiä tiedetä maanlaajuisesti, silti suurin osa Suomessa toimivista yrittäjistä ovat mikroyrittäjiä. Ne ovat arvokkaita työllistäjiä ja niiden toiminta on lähes kaikilla aloilla. Mikroyritysten osuus kaikista yritysmuodoista Suomessa oli jopa 92,4 % vuoden 2020 aikana. (Oulun yliopisto 2020.)

Tämän kehittämistyön toimeksiantaja on juuri mikroyritys X. Se sijaitsee pääkaupunkiseudun alueella ja toimii erittäin kilpailukykyisellä palvelualalla. Tämän mikroyrityksen ainoa työntekijä eli sen perustaja itse tekee kaikki yrityksensä työtehtävät.

Koska mikroyrityksessä työskentelee pieni määrä työntekijöitä tai yleensä vain itse yrittäjä niin hän kantaa hartioillaan koko työlaastin. Hän tekee kaiken työn itse yleensä ylimääräisellä työpanostuksella usein jättämällä huomiomatta viikonloppu- ja pyhäpäiviä. Kun hän ei itse kykene tai voi omia palveluita kattavasti tarjota, silloin hän palkkaa ulkopuolisia palvelun toimijoita.

Tavallisen mikroyrittäjän arkielämässä voi olla suuri määrä rutiininomaista työtä, jonka tekemällä hän käyttää omaa tietokoneensa. Yksi tylsistä, mutta silti hyvin tärkeistä yrityksen toiminnan kannalta työstä oleskelee asiakaslaskujen luonti ja lähettäminen asiakkailleen. Muutaman asiakaslaskun lähettäminen kuukauden aikana ei ole vaikeaa asiaa eikä vie paljon yrittäjän aikaa. Silloin, kun yrityksellä on kymmeniä asiakkaita, niin laskujen lähettämiseen alkaa kuulua jo omaa aikansa, minkä joutuu ottamaan muista liiketoiminnan prosesseista. Asiakaslaskujen luonti ja lähettäminen pitää olla aina ajoissa tehty eikä tämän voi suorittaa puutteellisesti. Huonosti palvelevan yrityksen asiakas ei enää pidä yritystä asiantuntijana tai pahimmassa tapauksessa voi käydä jopa niin, että asiakas ei enää halua jatkaa asiakkuuttaan tässä yrityksessä. Silloin se voi lähteä muille samalla alalla oleville yrittäjille eli suorasanaisesti kilpailijoille. Tästäkin syntyy lisää vaivaa, kun kilpailijat saavat uuden asiakassuhteen ei omista hyödyistä tai ominaisuuksista, vaan entisen yrittäjän puutteellisen palvelimisen takia. Fataalitapauksessa, jos laskutus tehdään puutteellisesti tai lainvastaisesti asiakas voi jopa kääntyä käräjäoikeuteen.

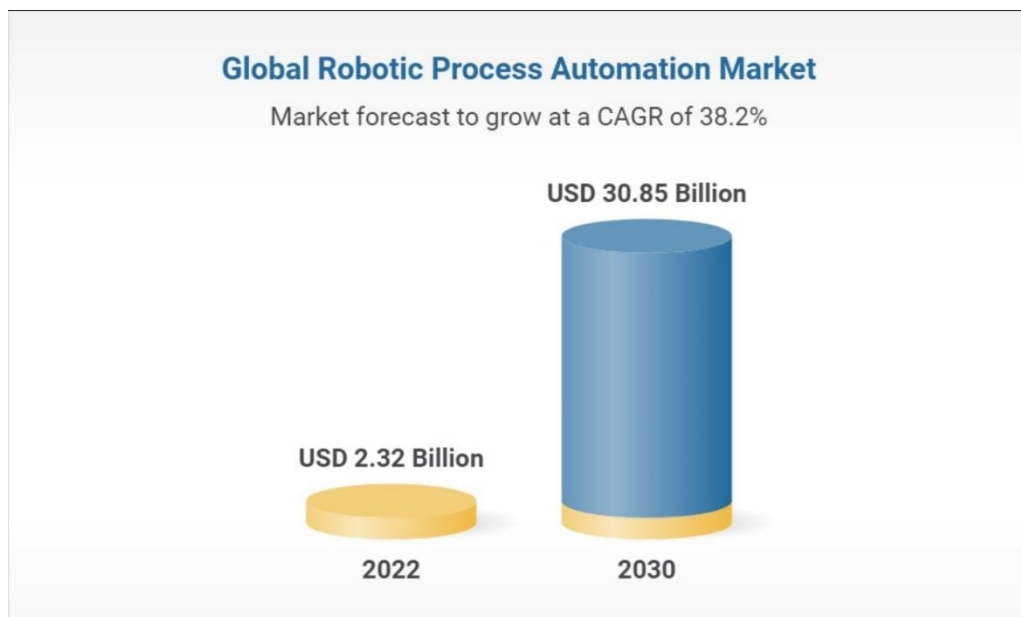
2 Ohjelmistorobotiikka mikroyrittäjän asiakaslaskutuksessa

Ohjelmistorobotiikka (Robotic Process Automation, RPA) on tietokoneella suoritettavien prosessien automatisointi. Ohjelmistorobotiikan alustalla luotu ohjelmistorobotti toimii tavallisen työntekijän mallilla ja useasti käyttää ohjelmien omat käyttöliittymät, kuten ihmisetkin käyttävät. Tämän tehokkaan digityöntekijän hyödyntämällä on helppo automatisoida useita digitaalisia töitä suhteellisen pienellä vaivalla. (Kaarlejärvi & Salminen 2018, 53.)

Ohjelmistorobotiikalla voi yleensä toteuttaa minkä vaan tietokoneella suorittavan prosessin. Tämän takia voisi kuvitella, että pitää kaikki toimistotyöt automatisoida. Silti monimutkaisen ja vaikean prosessin toteutus ohjelmistorobotiikan työkaluilla on hankalaa ja työllistävää. Myös voi olla, että tällaisen robotin suorittamiseen käytetyt resurssit eivät palaudu takaisin. Näin projektista ei tule lainkaan kustannustehokasta, vaan rahaa kulutetaan turhaan. Tämän takia on tarkistettava ensin mihin käyttökohteisiin ohjelmistorobotiikkaa sopisi parhaiten. Lisäksi on tehtävä alustava ROI (Return on Investment), jotta saa selvää, kuinka nopeasti ohjelmistorobotti maksaa itsensä takaisin. Muussa tapauksessa voi kustannukset olla turhaan suuria eikä robotin rakentaminen silloin ole enää kannattavaa. (Taulli 2020, 2-4.)

Oikein toteutettu ja rakennettu ohjelmistorobotti voi huomattavasti nopeuttaa rutiinisia työprosessia ja tuoda suuria säästöjä sen käyttävälle organisaatiolle. Lisäksi tällainen työkalu voi suuresti helpottaa työntekijöiden työelämän, parantaa työympäristön ilmapiirin ja vaikuttaa positiivisesti tyytyväisyyteen työpaikalla. Samalla se voi nostaa asiakkaiden tyytyväisyyttä saadusta palvelustaan. Kaikkien näiden tekijöiden kautta myös organisaation ammattilaistasonousee sen asiakkaiden silmissään. (Taulli 2020, 11-14.)

Maailman RPA (Robotic Process Automation) markkinan koko vuonna 2022 oli arvioiden mukaan 2,32 miljardia dollaria. Maailmanlaajuinen RPA (Robotic Process Automation) markkinakoon odotetaan nousevan 30,85 miljardiin dollariin vuoteen 2030 mennessä. Tämä merkitsee, että 2022-2030 ennustejakson keskimääräinen vuosittainen kasvu (CAGR eli Compound Annual Growth Rate) tulee olemaan 38,2 % (Investopedia 2022). Maailmanlaaja markkinakoon kehittämiskasvun ennuste on esitetty kuviossa 1 (Research and Markets 2022).



Kuvio 1: Maailman RPA (Robotic Process Automation) markkinakoon ennuste.

COVID-19-pandemia vaikutti negatiivisesti yritysten liiketoimintaan maailmanlaajuisesti, mikä johti liikevaihdon menetyksiin tilikaudella 2020. Yritysten liikevaihdon lasku johti tarpeeseen leikata yleiskustannuksia automatisoimalla liiketoimintaprosessejaan ja vähentämällä ylimääräisten tehtävien suorittamiseen osallistuvien työntekijöiden määrän. Ohjelmistorobotiikan tuottajat ja asiantuntijat hyödynsivät tämän mahdollisuutena tuoda robottiprosessiautomaatio (RPA) uusille yrityksille. (Research and Markets 2022.)

2.1 Asiakaslaskutuksen automatisoinnin vaihtoehdot

Asiakaslaskutus on jokaiselle yritykselle erittäin tärkeä toiminto. Riippumatta yrityksen koosta sen liikevaihto on tiukasti kiinni tulevista ja lähtevistä rahavirroista. Yrityksen maksuvalmius voi heikentyä huomattavasti, jos sen laskutusprosessissa on puutteita tai viiveitä. Tämän lisäksi laskutus on näkyvässä asiakkaille. Sen jatkuvasti huono suoritus voi pilata yrittäjän imagoa ja mainetta. (Kaarlejärvi & Salminen 2018, 120-121.)

Asiakkaiden laskutus eli myyntireskontran prosessi alkaa aina laskun laatimisesta. Tämän jälkeen valmiiksi täytetty lasku lähetetään asiakkaalle. Lopussa laskun maksusuorituksen jälkeen sen summa näkyy yrityksen pääkirjanpidossa. (Kaarlejärvi & Salminen 2018, 120-121.)

Yksi suosituimmista alueista ohjelmistorobotiikalle on taloushallinnon prosessien automatisointi. Kun taloushallinnon prosesseja halutaan automatisoida, niin täytyy niiden olla jo hyvin digitaalisessa muodossa. Tämä tarkoittaa, että taloushallinnon kaikkien transaktioiden on oltava sähköisessä muodossa ja kaikkiin tietoihin voi käsitellä sähköisesti. (Kaarlejärvi & Salminen 2018, 14-15.)

2.2 Ohjelmistorobotti investoinnin näkökulmasta

Jokaiselle yritykselle investointi on aina tärkeä toiminto, koska silloin se laittaa rahat kassasta projektiin, josta suunnitelman mukaan se hyödyntää useita vuosia. Yleisesti investointi yritykselle on pysyvä rahanpanostus eikä sen voi jälkikäteen peruuttaa. Tämän takia yrityksen on mietittävä vakavasti ainakin seuraavia seikkoja ennen investoinnin tekoa. Kuinka nopeasti tämä investointi voi maksaa itsensä takaisin ja mitkä ovat sen kokonaiskustannukset. (Jormakka, Koivusalo, Lappalainen & Niskanen 2015, 229.)

Investoinnin kannattavuutta voidaan laskea eri laskentamenetelmiä käyttäen. Tässä ovat muutamat vaihtoehdot kuten nettonykyarvo, nykyarvoindeksi, sisäinen korkokanta, takaisinmaksuaika ja muut (Jormakka ym. 2015, 229). Tässä projektissa ei ole paljon muuttuvia tekijöitä, mitkä olisi voineet vaikuttaa projektin kannattavuuslaskentaan. Tämän takia meille sopivin kannattavuuden laskentamenetelmä on takaisinmaksuaika. Investointi on omistajalleen kannattavaa, kun se tuottaa lisäarvoa (Jormakka ym. 2015, 235). Takaisinmaksuaika (Payback

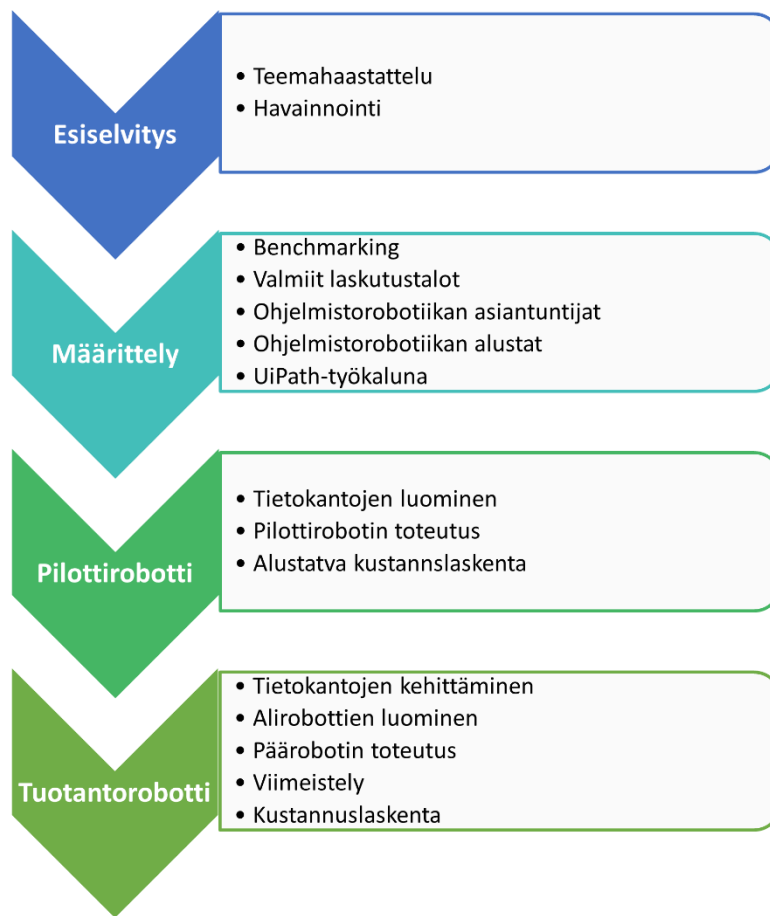
Period) on monessa yrityksessä suosittu menetelmä. Tässä menetelmässä selvitetään, missä ajassa investointi maksetaan takaisin vuotuisella nettokassavirralla. Kun käytän laskentamenetelmänä yksinkertainen takaisinmaksuaika, en tarvitse ottaa huomioon investoinnin korkoa ollenkaan. (Jormakka ym. 2015, 238-239.)

3 Kehittämistyön prosessi

Tämän kehittämistyöni prosessi oli monivaiheinen ja laaja. Ensin haastattelin minun toimeksiantajani ja selvittelin, mitkä haasteet hänellä oli asiakaslaskuttamisessa. Sen jälkeen tein havainnoinnin kuvaamalla videot kaikista laskuttamisen välivaiheista. Tämän aineistonkeruumenetelmän ansiosta selvitin, kuinka asiakaslaskutuksen suoritetaan tällä hetkellä. Seuraavaksi loin vertailutaulukon, johon kirjoitin aika-arvot jokaiselle vaiheelle. Tällä tavalla voi jatkossa vertailla ohjelmistorobotin ajossa käytetty aika. Tämä vertailutaulukko sopi myös investoinnin kannattavuuden laskemista varten.

Seuraavaksi pohdin, mikä ratkaisu sopisi parhaiten tämän asiakaslaskutushaasteeseen. Minulla oli yhteensä kolme ohjelmistorobotiikan vaihtoehtoa. Mahdollista olisi ostaa jo valmiin pakettiratkaisun sellaisenaan ja ottaa sen suoraan käyttöön. Toisena keinona olisi tilata ohjelmisto, mikä koostuu enemmän tai vähemmän itsenäisistä moduuleista. Viimeisenä olisi toimeksiantajan vaatimusten ja toiveiden mukaan räätälöity ohjelmistorobotti.

Kun sopiva UiPath Studio-työkalu ohjelmistorobotin luomista varten oli valittu, siirryin pilottirobotin toteuttamisvaiheeseen. Siellä määrittelin, minkälaisen ohjelmistorobotin täytyy olla, mitä sen on osattava ja mihin on pystyttävä. Mitä esivalmisteluja pitää tehdä ennen kuin aloitan varsinaisen robotin luonnin. Kun määrittely oli valmis, suunnitelman mukaisesti tein kaikki alustavat työt kuten tiedostojen muokkailut ja datan esimäärittelyt. Seuraavaksi toteutin pilottirobotin vaihe vaiheelta. Kun se oli valmis, testasin sen ja lisäsin vertailutaulukkoon puuttuvat aika-arvot. Näillä tiedoilla vertailin, kuinka tehokas tämä työkalu on asiakaslaskutusta varten ja miten paljon työaikaa se voi säästää. Esittelin toimeksiantajalleni alustavan kustannuksen tulokset ja hyväksynnän jälkeen jatkoin kohti tuotantorobotin toteutusta. Kehitin aikaisemmin luodut tietokannat, yksinkertaistamalla ne ja keskittämällä samaan Excel-tiedostoon. Tämän jälkeen lisäsin kaksi alirobottia kahdelle erilliselle prosessille. Seuraavaksi suoritin päärobotin kehittämällä pilottirobottia suureksi kokonaisuudeksi ja yhdistämällä siihen alirobotit. Viimeistelyvaiheessa kävin läpi jokaisen tuotantorobotin aktiviteetin läpi kommentoimalla sen tarvittaessa. Debuggausta varten lisäsin lokit ja Try Catch-rakenteet laadunvarmistamista varten. Kun tuotantorobotin testaus tuli viimeistään valmiiksi, jatkoin vertailla sen tehokkuuden vertailutaulukossa. Samalla laskin missä ajassa se maksaa itsensä takaisin. Viimeiseksi esitin toimeksiantajalleni tuotetun robotin asiakaslaskuttamista varten. Kuviossa 2 on lyhyesti esitetty kehittämistyön prosessin vaiheet.



Kuvio 2: Kehittämistyön prosessin kuvaus.

3.1 Esiselvitys

Esiselvityksen aikana analysoin nykytilannetta ja sain selville, miten asiakaslaskutuksen prosessi oli toteutettu aikaisemmin. Ensin puolistrukturoidulla teemahaastattelulla selvitin tarkasti, mitkä haasteet toimeksiantajalla oli asiakaslaskutuksessa. Sitten teknisen havainnoinnin avulla sain esiin asiakaslaskutuksen toteuttamisprosessit. Näillä tiedoilla loin vertailutaulukon, jota käytin tulevassa kustannuslaskennassa.

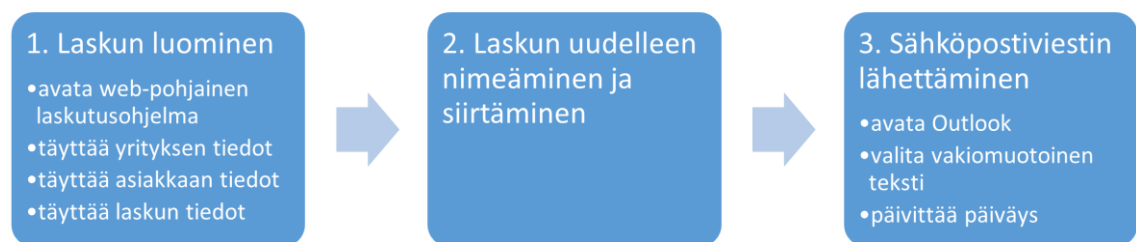
3.1.1 Teemahaastattelu

Tässä osiossa kerron teemahaastattelusta, minkä pidimme toimeksiantajan kanssa. Siellä selvisi, kuinka tällä hetkellä asiakaslaskutus oli toteutettu. Teemahaastattelusta sain tarkemman tiedon tutkimusongelmasta ja samalla mietimme toimeksiantajani kanssa vaihtoehtoja asiakaslaskutuksen vaikeuksien ratkaisuun.

Aineistonkeruumenetelmällä kerätään aineistoja omaa projektia varten. Tietojen etsimiseen on monta keinoja ja tässä vaiheessa minulle sopivin oli teemahaastattelu. Aineistonkeruumenetelmänä valitsin puolistrukturoidun teemahaastattelun. Valitsin sen, koska tarkoituksena oli

suunnata keskustelumme tutkimusaiheesta tavoitteeseen. Tällä menetelmällä pyritään kasvattaa ymmärrystä haasteen liittyvistä asioista. Yleisesti teemahaastattelua otetaan käyttöön, kun ei tunneta ilmiötä tai kun ilmiötä selittäviä teorioita ei ole käytettävissä. (Kananen 2015, 148-151.) Puolistrukturoidulle teemahaastattelulle on tyypillistä, että aihepiiri on tiedossa, mutta kysymysten tarkka muoto on vielä kysymyksen alla (Hirsjärvi, Remes & Sajavaara 2009, 208-209).

Haastattelussa selvisi, että tällä hetkellä toimeksiantaja käyttää ilmaista web-pohjaista laskutusohjelmaa. Tällä sovelluksella hän luo uusia laskuja asiakkailleen. Luonnin jälkeen laskut tallennetaan sähköisinä PDF-muodossa. Sitten niitä nimetään uudelleen asiakasnimien perusteella ja siirretään arkistointikansioihin. Seuraavaksi avataan Outlook-sähköpostisovellus web-selaimessa. Siinä generoidaan sähköpostiviestit manuaalisesti laittamalla kirjeen viestikenttään valmiiksi muokattu vakiomuotoinen teksti. Ainoastaan laskun eräpäiväys-kohdan tekstissä muutetaan laskunkohtaiseksi. Lopussa kirjeeseen liitetään oikean laskun PDF-muodossa ja lähetetään sähköpostikirjeen asiakkaalle. Suurin osa kaikista laskuista ovat vuokralaskut. Toimeksiantaja lähettää niitä asiakkaille noin 20-25 kappaletta kuukaudessa. Kuviossa 3 on lyhyesti esitetty asiakaslaskutusprosessin vaiheet ja niiden järjestys.



Kuvio 3: Asiakaslaskutusprosessin kuvaus.

Teemahaastattelussa esiin nousivat toimeksiantajan haasteet asiakaslaskutuksessa. Hän kertoi, että ei ehdi aina lähettää myyntilaskuja asiakkailleen ajoissaan. Tämän takia yrityksen tilillä välillä ei ole rahasummaa, minkä piti veloittaa palveluiden ja tuotteiden tuottamisesta asiakkaille. Lisäksi hän kertoi, että joskus laskujen täyttämisen vaiheessa tapahtuu virheitä. Näiden seurauksena laskun täyttämistä pitää aloittaa kokonaan uudestaan. Myös kävi ilmi, että välillä hän laittoi pienemmän ALV-kannan laskuun. Tämän takia veloitettava summa saattoi olla pienempi, kuin sen piti olla. Toisaalta laskun summa voi vahingossa olla suurempi. Muutaman kerran hän jopa lähetti asiakkaalle laskun, minne unohti lisätä alv:n mukaan. Tämän takia firman tilille ei tullut riittävää maksusummaa asiakkaalle toimitetusta palvelusta. Ei toimeksiantaja uskaltanut korjatun laskun lähettää perään, koska hän huomasi sen virheen vasta parin päivän jälkeen. Eikä halunnut menettää firman imagoa tästä virheestään.

Toimeksiantajan odotus tälle projektille oli sujuva asiakaslaskuttaminen ohjelmistorobotiikkaa hyödyntämällä. Samalla hän toivoi asiakaslaskuttamisessa käytetyn ajankulutuksen vähenvän huomattavasti.

3.1.2 Havainnointi

Valitsin toisena aineistonkeruumenetelmänä havainnointi, koska sen avulla voi saada välitöntä suoraa tietoa toimeksiantajan toiminnasta ja käyttäytymisestä. Havainnoinnilla päästään tarkkailemaan tarkkailukohteessa tapahtuvaa ilmiötä luonnollisessa ympäristössä. Muun muassa vuorovaikutuksen tutkimuksessa havainnointi on erinomainen aineistonkeruumenetelmä. (Hirsjärvi, Remes & Sajavaara 2009, 212-213.) Havainnointi voi myös nauhoittaa, sitten samaan tilanteeseen voi palata milloin vaan. Silloin aineistomäärä kasvaa, mutta sen tuotos on aito tallenne tilanteesta. (Kananen 2015, 139.) Tekninen havainnointi on varmin tapa tiedonkeruussa, jolloin tutkimusilmiö voidaan tallentaa kokonaisuudessaan ja palata aineistoon myöhemmin (Kananen 2015, 135).

Sovin havainnointiprosessista toimeksiantajani kanssa. Pääsin yrityksen työpisteeseen ja käytin ohjelman, jolla sain asiakaslaskutuksen prosessit videoimalla talteen. Tätä varten käytin kameraa ja sen omaa kuvausohjelmaa. Tallensin videon kaikista asiakaslaskutukseen liittyvistä pää- että välivaiheista.

Seuraavaksi videoanalyysin vaiheessa tein litteroinnin ja loin Microsoft Excel taulukkolaskentaohjelmassa vertailutaulukon. Litterointi on erilaisten tallenteiden, kuten äänitteiden, videoiden ja kuvien muokkaamista tekstimuotoon analysointia varten (Kananen 2015, 160). Tämän videomateriaalin analysoimalla sain syvän ymmärryksen, miten yrityksen asiakaslaskuttaminen tapahtuu tällä hetkellä. Saatujen videotallenteiden perusteella laadin vertailutaulukon. Tämän työllistävän prosessin jälkeen sain esitettyä jokaisen asiakaslaskutuksen välivaiheen kesto sekunneissa omalla rivillä. Taulukon vasemmalla puolella esittelin asiakaslaskutuksen manuaalivaiheiden kestot ja taulukon oikealla puolella automatisoidut vaiheiden kestot. Näin ollen oli helppo vertailla manuaalivaiheiden prosessit siihen automatisoituun ratkaisuun. Tällä tavalla oli helppo laskea kulutettu työaika per lasku keskimäärin.

3.2 Määrittely

Tässä vaiheessa aloitan koostaa esivaatimukset tälle projektille. Määrittelyllä tuon esiin mitä tästä projektista odotetaan. Muun muassa selvitän millä vaihtoehdoilla on mahdollista ratkaista toimeksiantajan haasteet ja automatisoida asiakaslaskutusprosessia. Tämän ratkaisun pitää nopeuttaa asiakaslaskutuksen ajonkulkua ja vähentää inhimillisten virheiden määrää. Lisäksi projektin toteutus ja käyttöönoton on oltava sujuva ja mutkaton, koska ei mikroyrityksellä ole ylimääräistä budjettia pitkien projektien toteuttamiselle.

3.2.1 Benchmarking

Käytin tässä vaiheessa tutkimusmenetelmänä benchmarking-menetelmää, eli vertailuanalyysiä. Ohjelmistorobotiikan valintaprossin aikana kävin järjestyksessä läpi kaikki mahdolliset vaihtoehdot asiakaslaskutuksen haasteen ratkeamiseen. Ennen kuin sopiva keino on valittu, mielessä on pidettävä koko ajan, mitä vaatimuksia tällä projektilla on. Mitä tämän ohjelmistoprojektin tulee tehdä ja mitä ominaisuuksia sillä pitää olla. (Laukkonen & Von Kügelgen 2020, 145.)

3.2.2 Valmiit laskutustalot

Yhtenä asiakaslaskuttamisen haasteen ratkaisuna voisi olla valmiin ohjelmistopalvelun käyttö näiden ohjelmistoiden tuottajista. Laskutustalon tuotteen osto, missä mukana ovat laskutukseen liittyvät toiminnot. Me kävimme läpi lukuisia laskutusohjelmia ja -sovelluksia. Näiden palvelutoimittajia on paljon Internetissä. Useissa laskutusohjelmistoissa on samankaltaiset ominaisuudet ja kyvykkyydet. Suurimmassa osassa niistä on valmiin laskupohjien käyttö laskujen luonnin yhteydessä. Kaikki nämä toiminnot nopeuttavat uuden laskun luomista ja lähettämistä. Mahdollisuus lähettää asiakkaille lasku välittömästi sähköposti- ja PDF-laskuna tai sitten verkkolaskuna. Kaikki laskut löytyvät laskuluettelosta, mikä helpottaa niiden hahmottamista. Lisäksi näissä laskutusohjelmistoissa on valmiit asiakas- ja tuoterekisterit, mihin voi tallentaa tietoja tulevaa käyttöä varten. Usealla on saatavilla myös tarjousten luonti ja lähetyksen kyvykkyydet. Lisäksi siinä on muitakin ominaisuuksia, joita on mahdollista hyödyntää omassa liiketoiminnassaan. Katsoimme monia palvelijoita pinnallisesti, mutta tarkemmin tarkasteltavaksi valittiin kolme: SimplBooks, Isolta ja Zervant. (SimplBooks 2021a; Isolta 2021a; Zervant 2021a.)

Monet näistä nettipalveluista mainostettu kuten ilmaiset laskutusohjelmat. Mutta todellisuudessa niissä on joko ilmainen kokeilujakso, mikä voi olla esimerkiksi 14 vuorokautta tai yhden kuukauden pituinen. Joskus ilmaisten laskujen määrä on rajoitettu esimerkiksi kymmeneksi kappaleeksi vuodessa. Yleisesti näiden palveluiden veloitus tapahtuu joko laskuttamalla hinnan per joka lasku tai sitten kuukausimaksuna, kun ostetaan tämän pakettipalvelun. (SimplBooks 2021b; Isolta 2021b; Zervant 2021b.)

Mikään näistä laskutusohjelmista ei sopinut toimeksiantajalleni, koska ne eivät loppujen lopuksi olleet täysin ilmaiset. Laskujen määrä mitä toimeksiantajani lähettää vuodessa on paljon suurempi kuin se mitä tarjotaan veloitusetta. Lisäksi ei hän tarvinnut muita hänelle ylimääräisiä laskutusohjelmien ominaisuuksia, vaan asiakkaan peruslaskutustoiminnon.

3.2.3 Ohjelmistorobotiikan asiantuntijat

Toisena vaihtoehtona voisi olla valmiin ohjelmistorobotiikan asiantuntijan työn osto. Näitä ammattilaisia on paljon tarjolla Internetissä. Suomessa suuria automatisaation ammattilaisia ovat esimerkiksi Knowit, CGI ja Mtech. Monilla tämän teknologian tarjoajilla on aika lailla samoja palveluja, mitkä ne myyvät omille asiakkailleen. Tällaisten työkumppanien avulla onnistuu toteuttamaan älykkäitä digitaalisia ratkaisuja, kuten esimerkiksi omia web- ja mobiilisolvelluksia, omia järjestelmiä, monenlaisia integraatio- ja rajapintaratkaisuja ja niin edelleen. (Knowit 2021; CGI 2021; Mtech 2021.)

Niiden keskimääräinen veloitus on aika korkea ja siksi nykypäivänä vain isoimmat yritykset voivat hyödyntää ohjelmistorobotiikkaa omissa liiketoimintaprosesseissaan. Tämä tarkoittaa, että tästä suhteellisen pienessä projektissa saattaa tulla liian suuret kustannukset toimeksiantajalleni. Tästä syystä hän ei näitä ohjelmistorobotiikan ammattilaisia tilannut, vaikka nämä olivat tiedossa jo etukäteen. Toisena perusteena on ohjelmistorobotiikan asiantuntijan tarpeettomuus juuri tässä projektissa, koska ei tämän pienen työn toteutus vaadi niin syvällistä ohjelmistorobotiikan osaamista.

3.2.4 Ohjelmistorobotiikan alustat

Seuraavaksi esiin tuli oman ohjelmistorobotin luonti omia tarpeita varten. Me kartoitimme ohjelmistorobotiikan erilaisia alustoja, joilla voisimme sellaista ohjelmistorobottia tuottaa. Ohjelmistorobotiikan alustoiden toimittajista esiin nousivat seuraavat toimijat: UiPath, Automation Anywhere, Blue Prism ja niin edelleen.

Ohjelmointiympäristön vaihtoehtoina olivat Blue Prism, Automation Anywhere ja UiPath Studio. Näistä valittiin UiPath, koska tämä on ilmainen softa, jossa on hyvin laaja tukiympäristö, eikä se vaadi aiempaa kokemusta ohjelmoinnista. Siinä on drag and drop (raahaa ja pudota) - graafisen käyttöliittymän toiminto, mikä helpottaa huomattavasti ohjelmistoprosessin työstämistä. Tämän alustan käyttöliittymä on intuitiivisesti selkeä ja yksinkertainen. Myös visuaalisesti toteuttava robottiympäristö auttaa hahmottaa robotin kulkua projektin aikana. Tämän ohjelmistoalustan suurena etuna on mahdollisuus siirtää erilaisia robotin aktiviteetteja raahaamalla ne hiirellä. Näistä osioista voi rakentaa täysmittaisen robottiprojektin. (UiPath 2022a.) Automation Anywhere A360 on pitkälti samankaltainen, mutta web-pohjainen ohjelmistorobotiikan alusta. Koska se on pilvipohjainen, se ei sopinut tähän projektin toteuttamistyökaluksi. (Automation Anywhere 2022.) Viimeisenä vaihtoehtona olevaa Blue Prism ei myöskään sopinut, koska sen pitkäaikainen käyttö on rajoitettu ilmaisella 30 päivän kokeiluversiolle (Blue Prism 2022).

Painavana syynä UiPathin valintaan oli se, että minulla oli jo kokemusta entuudestaan ohjelmistorobotiikasta UiPathin parissa opiskeluni ja työni aikana. Lisäksi tässä softassa on

suhteellisesti helppo luoda ohjelmistorobotteja drug and drop-tekniikalla. Sen ilmainen versio UiPath Studio sopii juuri näitä pieniä automatisoinnin tarpeita varten. Ohjelmistorobotin luonti oli päätetty tehdä toimivana ja taloudellista hyötyä tuovana. Lisäksi tämän robotin tarkoituksena oli selvittää kuinka helppo tai vaikea integroida automatiikan mikroyrityksen liike-toimintaprosesseihin ylipäätään. Tulevaisuudessa olisi mahdollista kartoittaa yrityksen liike-toimintaprosesseja lisää, joita voitaisiin myös automatisoida.

3.2.5 UiPath Studuo-työkaluna

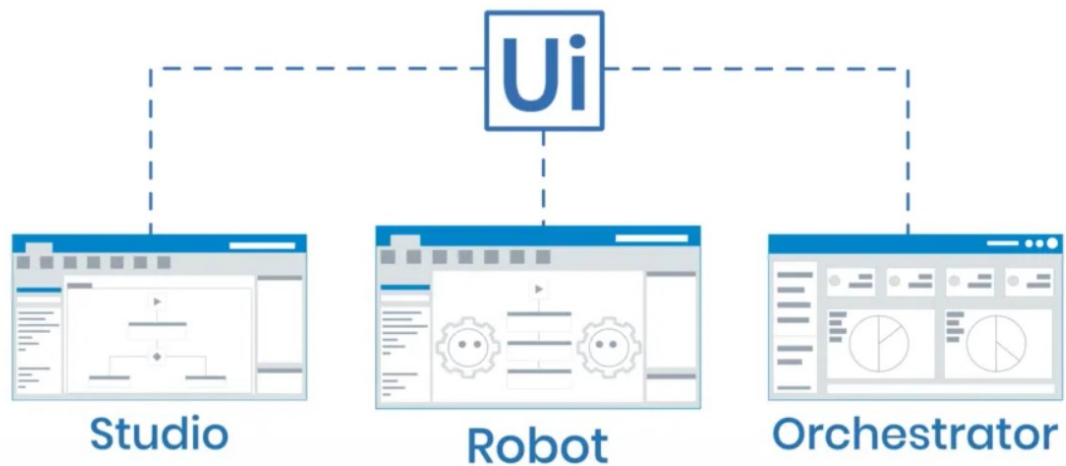
UiPath on yhdysvaltalainen maailmanlaajuinen ohjelmistoyritys, joka valmistaa robottiprosessiautomaation (RPA) ohjelmistoja. Sen perustivat Daniel Dinesin ja Marius Tircan vuonna 2005 Romaniassa. UiPathilla on lukuisia alueellisia toimistoja yli 40 maassa ympäri maailmaa. Niitä löytyy Euroopan ja Aasian useasta maasta, kuten Lontoosta, Pariisista, Bukarestista, Tokiosta, Singaporesta, Hongkongista, Münchenistä ja Madridista. Mutta sen pääkonttori sijaitsee New Yorkissa, Yhdysvalloissa. (Wikipedia. UiPath.) UiPathilla on useita suuria yritysasiakkaita, kuten Nasa, Uber, Deloitte, Xerox, PwC, Tata, Accenture, Sweco, Spotify ja IBM (UiPath 2022b). Myös sillä on suuria teknologisia kumppanuuksia, esimerkiksi Microsoftin, Oraclen, Abbyyn ja SAPin kanssa (UiPath 2022c). UiPathin liikevaihto vuonna 2023 oli 1.6 miljardia dollaria ja töissä oli 3833 työntekijää (Wikipedia. UiPath).

Kaikki UiPath-tuotteet ovat suunniteltu ja kehitetty turvallisuutta ajatellen. Tietoturva on sisäänrakennettu suoraan kehitysprosessin elinkaareen teettämällä sekä automaattisia tietoturvavarkistuksia että penetraatiotestauksia jokaisessa koontiversiossa. Kolmas riippumaton osapuoli Veracodelle tarkastaa reliisit korkeimmalla tasolla. Näin varmistetaan, että tietoturva on sertifioitu. (UiPath 2022d.)

UiPath Studio on ohjelmistopaketti, joka on kehitetty ohjelmistorobotiikkaan ja jakautuu kahteen eri versioon. Ilmainen Community Edition suunnattu RPA-ohjelmistokehittäjille ja maksullinen Enterprise RPA Platform suurille yrityksille. Jälkimmäisestä versiosta on myös lyhyt maksuton kokeilujakso. (UiPath 2022a.)

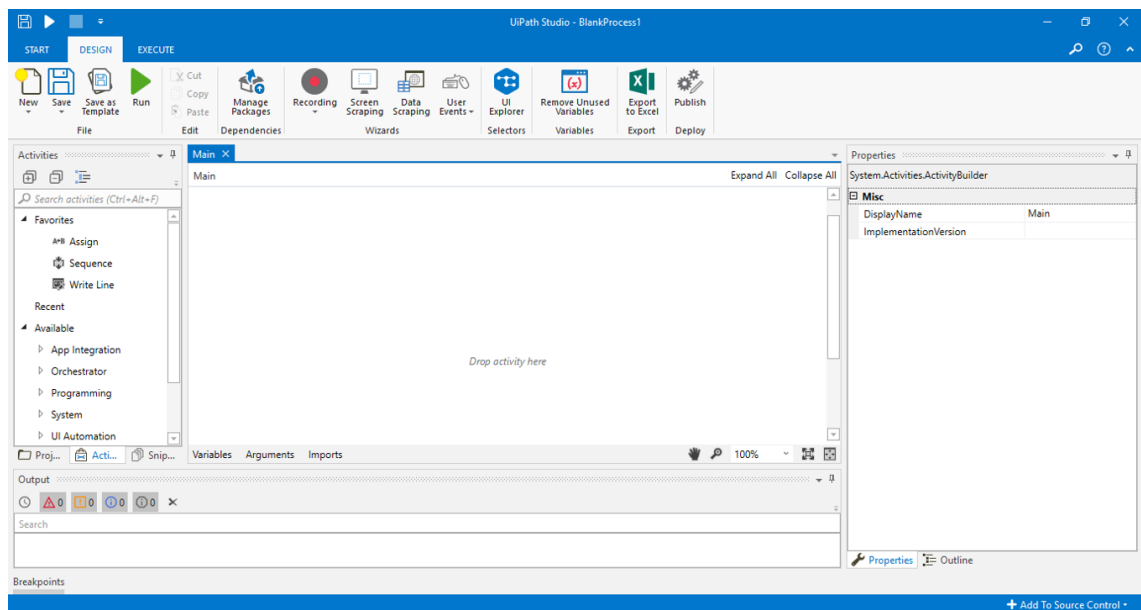
UiPath ohjelmiston sujuvan toiminnan takaamiseksi sille on laitteisto- ja ohjelmistovaatimukset. UiPath tukee vain Windows-käyttöjärjestelmiä versiosta 8.1 ylöspäin. Työasemalla täytyy olla vähintään .NET Frameworkin versio 4.6.1 asennettuna. Myös suorituskykyä varten on omia vaatimuksia. Tietokoneella sisämuistia pitää olla vähintään 4 gigatavua ja vähintään 3,5 gigatavua vapaa tilaa kovalevyllä. (UiPath. 2022f.)

UiPath Studio -ohjelmistopaketti koostuu neljästä osasta: UiPath Studiosta, UiPath Studio X:stä, UiPath Orchestratorista ja UiPath Robotista. UiPath Studio ja UiPath Studio X ovat varsinaisia ohjelmistorobottien ohjelmointiympäristöt, joissa robotteja luodaan ja määritellään. Kuviossa 4 on esitetty UiPath Studion toiminnan periaate. (UiPath 2022a.)



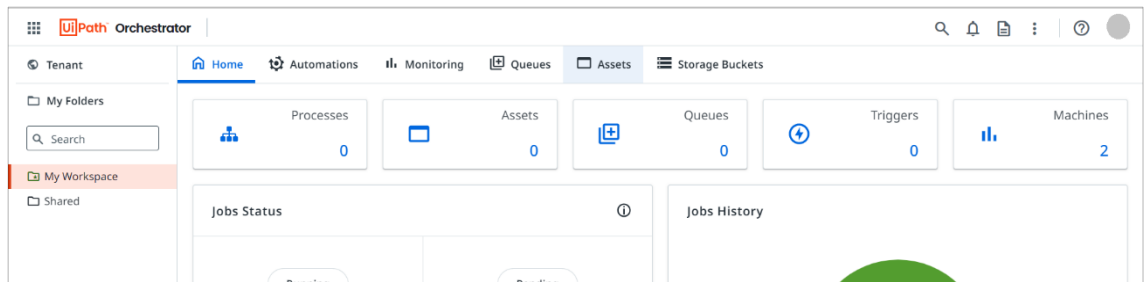
Kuvio 4: UiPath Studion toiminnan kuvaus.

Ympäristöjen erona on se, että Studiossa automatisoidaan prosessit ja Studio X:ssä tehtävät. Studio on suunnattu RPA ohjelmistokehittäjille ja Studio X on liiketoimintakäyttäjää varten eikä siinä ohjelmointitaitoja tarvita. Kuviossa 5 on esitetty UiPath Studion käyttöliittymä. (UiPath 2022a.)



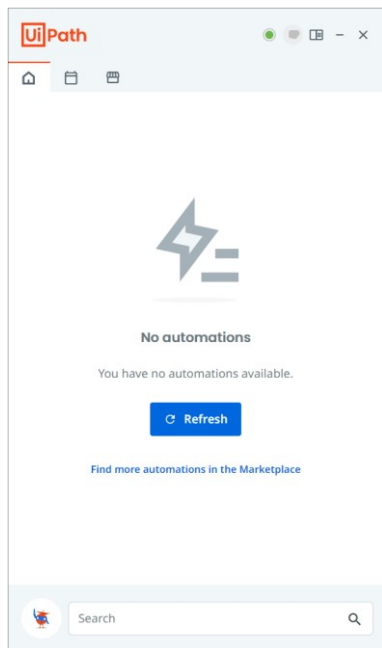
Kuvio 5: UiPath Studion käyttöliittymä.

UiPath Orchestratori toimii ohjelmistorobottien hallinnointityökaluna ja sijaitsee palvelimessa. Orchestratorin avulla roboteille annetaan käskyt, keskitetysti ohjataan niiden kulkua, sekä monitoroidaan niiden suorituksia. Tällä työkalulla on helppo skaalauttaa ohjelmistoprosesseja, hallinoida ja seurata suuria robottien määriä. Kuvio 6 esittää UiPath Orchestratorin käyttöliittymän.



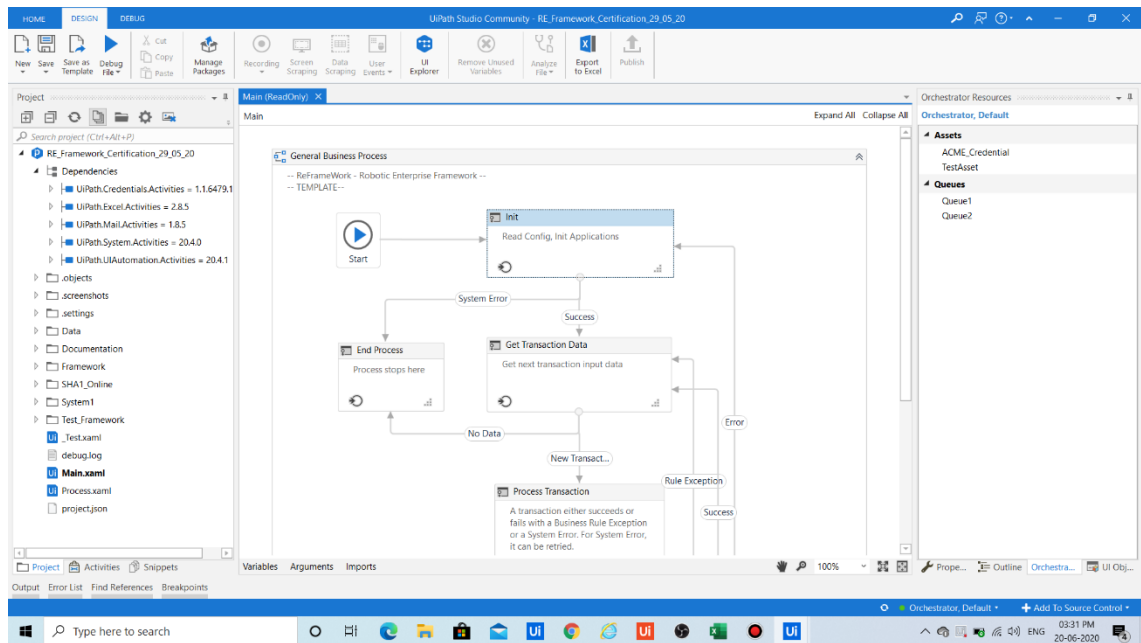
Kuvio 6: UiPath Orchestratorin käyttöliittymä.

UiPath Robot on pieni ohjelma, jonka avulla roboteilla voi automatisoida prosesseja paikallisen automatisoinnin lisäksi Orchestratorin virtuaaliympäristössä. Silloin valittu robottikone liitetään Orchestratoriin, jossa haluttu ohjelmistoprosessi voi käynnistää. Kuviossa 7 on esitetty UiPath Robot. (UiPath 2022a.)



Kuvio 7: UiPath Robot.

UiPath Studiolla luotavat prosessit koostuvat aktiviteeteista. Jokaisella on oma tarkoituksensa. Yhdellä aktiviteetilla voi esimerkiksi kirjoittaa arvot Excel-tiedostoon, avata jokin ohjelmisto, klikata painiketta hiirellä tai kirjoittaa tekstiä tiedostoon. Kaikilla aktiviteeteilla on omat ominaisuutensa ja asetuksensa, joita saa tarvittaessa muuttaa. Prosessin luominen tapahtuu vetämällä aktiviteetti vasemmalta laidalta suoraan Main.xaml työnkulkunäkymään. Kuvio 8 esittelee esimerkki luodusta aktiviteettikaaviosta UiPath Studiossa.



Kuvio 8: Esimerkki luodusta aktiviteettikaaviosta UiPath Studiassa.

Kuten muissa ohjelmointiympäristöissä myös UiPath Studiolla on käytössä omat muuttujat ja argumentit. Näissä voi tallentaa erilaiset arvot tulevaa käyttöä varten aktiviteeteissa. UiPath Studiolla on paljon muitakin sisäisiä työkaluja, joilla ohjelmistorobottien luominen suorituu helposti. (UiPath 2022a.)

3.3 Pilottirobotti

Ennen ohjelmistorobottin toteuttamista minun oli suorittava pilottirobotin ensin. Tämän robotin tehtävään kuuluu luoda vaan kolme laskua kerrallaan täyttämällä laskupohjat web-palvelimessa. Kaikkien tarvittavien tietojen sen on osattava hakea Excel-tiedostoista, joissa asiakastiedot säilyvät. Seuraavaksi oli testausvaihe, jossa kolmen laskun asiakkaille toimittamista pilottirobotin avulla testailin. Sen jälkeen arvioin pilottirobotin tehokkuutta vertailutaulukossa ja esittelin tulokset hyväksyttäväksi toimeksiantajalleni.

3.3.1 Tietokantojen luominen

Prosessien automatisointia varten laskutukseen kuuluvan datan täytyy olla mahdollisimman digitaalisessa muodossa. Kaikki yritykseltä lähtevät laskut ovat jo valmiiksi sähköisessä muodossa, koska niitä luodaan ilmaisen web-pohjaisen palvelun kautta. Niiden tuottamisessa käytetään Excel-taulukossa olevat asiakkaiden tiedot, kuten esimerkiksi Y-tunnukset jne. Niiden siirto järjestelmästä toiseen on vuosien aikana ollut turvallinen. Eikä suhteellisesti pienestä määrästä laskuista vuodessa ole kannattavaa ostaa laskutustalon palveluita. Ensisijaisesti loin valedatan tätä ohjelmistorobotti-projektia varten. Ei henkilökohtaisia tietoja saa käyttää testausta varten. Seuraavaksi tein kaksi erillistä Excel-tiedostoa. Ensimmäiseen laitoin kaikki

asiakkaiden omat tiedot omiin sarakkeisiin yhteen taulukkoon. Toiseen tiedostoon lisäsin jokaisen laskun omat tiedot, joissa on kuvaus palvelun tuottamisesta. Tämä selvitys toimitusta palvelusta on asiakaskohtainen ja siksi tekstien pituudet ovat erilaiset.

3.3.2 Pilottirobotin toteutus

Tässä kerron varsinaisen pilottirobotin teettämisestä. Ensimmäisenä loin DataBaseMyyjänTiedot-sekvenssin. Siellä luodaan Data Table:n, jossa on myyjän tiedot laskua varten. Myyjän tiedot eivät muutu, riippumatta siitä, mikä on sen asiakas ja siksi tietoja on mukavaa tallentaa omassa taulukossa. Tämä Data Table koostuu 12 sarakkeesta ja ne kaikki ovat String-tyyppisiä helppouden takia. Kun myyjän tietojen taulukko on luotu, robotti siirtyy eteenpäin. Kuviossa 9 on esitetty DataBaseMyyjänTiedot-sekvenssi.



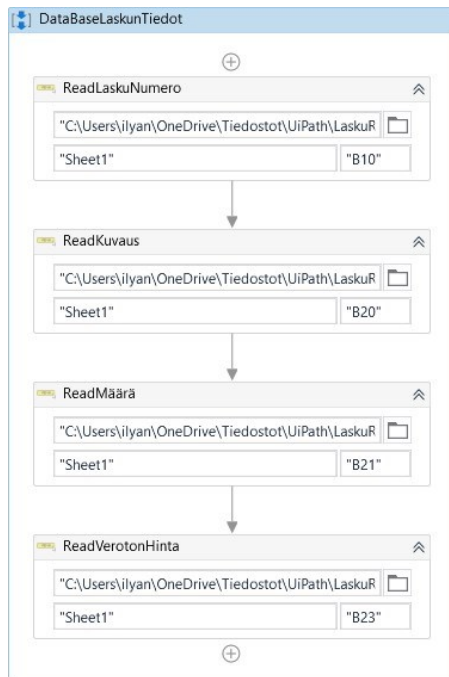
Kuvio 9: DataBaseMyyjänTiedot-sekvenssin kuvaus. (LaskuRobotti, Main.xaml, DataBaseMyyjän-Tiedot)

Seuraavaksi robotti kulkee DataBaseAsiakkaanTiedot-sekvenssiin. Tässä sekvenssissä se hakee asiakkaan tiedot ulkoisesta Excel-tiedostosta Read Cell-aktiviteettien avulla. Näin robotti tallentaa sisämuistiin 7 muuttujaan omat 7 arvoa. Kaikki tämän sekvenssin muuttujat ovat String-tyyppisiä. Kuviossa 10 on esitetty DataBaseAsiakkaanTiedot-sekvenssi.



Kuvio 10: DataBaseAsiakkaanTiedot-sekvenssin kuvaus. (LaskuRobotti, Main.xaml, DataBaseAsiakkaanTiedot)

Kun asiakkaan tiedot luettu läpi robotti siirtyy seuraavaan DataBaseLaskunTiedot-sekvenssiin. Nimensä mukaisesti siellä se hakee tarvittavat tiedot laskua varten. Niitä taas haetaan toisesta ulkoisesta Excel-tiedostosta Read Cell-aktiiviteettien avulla. Näin robotti tallentaa 4 muuttujaan omat 4 arvoa. Myös kaikki tämän sekvenssin muuttujat ovat String-tyyppisiä. Kuviossa 11 on esitetty DataBaseLaskunTiedot-sekvenssi.



Kuvio 11: DataBaseLaskunTiedot-sekvenssin kuvaus. (LaskuRobotti, Main.xaml, DataBaseLaskunTiedot)

Tämän jälkeen lisäsin LaskupohjanTäyttäminen-sekvenssin, missä robotti avaa oikean nettisivun Edge-browserissa. Tätä varten etukäteen asensin UiPath Studioon Microsoft Edge selaimen laajennuksen. Kun laskupohja on esillä, sen pitää täyttää oikeilla tiedoilla. Aluksi robotti täyttää yrityksen omat tiedot hakemalla ne omasta taulukosta DataBaseMyyjänTiedot-sekvenssistä. Niitä se laittaa Type Into-aktiiviteettien avulla oikeisiin kohtiin. Kun yrityksen tiedot hoidettu, robotti siirtyy syöttämään asiakkaiden omia, että laskujen tietoja. Se hakee ne suoraan aikaisemmin DataBaseAsiakkaanTiedot ja DataBaseLaskunTiedot aktiiviteeteillä määritetyistä muuttujista. Vielä laitoin kaikkiin Type Into-aktiiviteetteihin EmptyField-ominaisuus päälle. Sillä se tyhjentää jokaisen kentän arvot aina ennen sen täyttämistä. Tällä toiminnolla varmistetaan robotin laatua.

Kun kaikki laskun tiedot ovat valmiiksi täytetty laskupohjaan, robotti painaa Tallenna-painiketta Click-aktiiviteetillä. Pilottirobotti toistaa tämän syklin kolme kertaa, koska laskua on yhteensä vain kolme. Kuviossa 12 on esitetty koko pilottirobotin prosessien kuvaus.



Kuvio 12: Koko pilottirobotin prosessien kuvaus. (LaskuRobotti, Main.xaml)

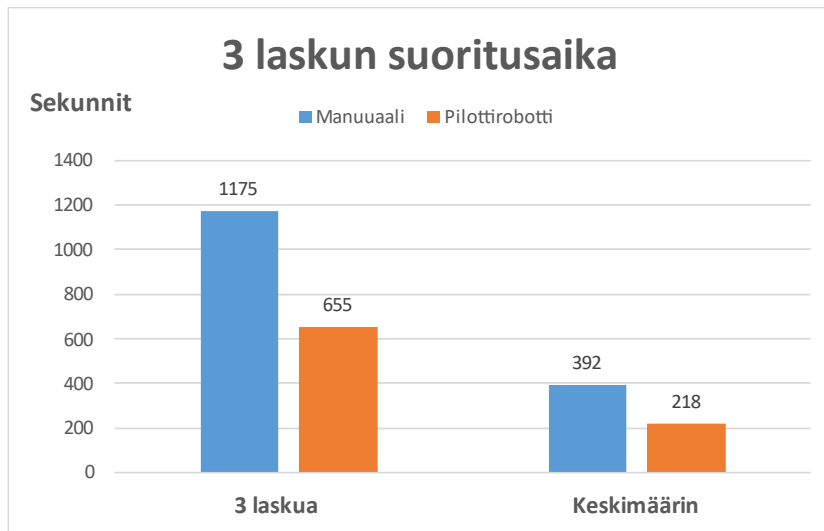
Seuraavaksi ovat taas manuaaliset prosessit, kuten laskujen uudelleen nimeäminen ja siirtäminen. Sen jälkeen avataan sähköpostipalvelun selaimessa ja lähetetään asiakkaille viestit laskujen kera. Näitä toimintoja ei ole automatisoitu vielä tässä projektin vaiheessa. Tämä pilottirobotti koostuu 4:stä sekvenssistä ja vain täyttää laskupohjat ja tallentaa valmiita laskuja. Kaikki seuraavat vaiheet tehdään manuaalisesti käsin.

3.3.3 Alustava kustannuslaskenta

Pilottirobotin testauksen jälkeen lisäsin vertailutaulukkoon oikealle puolelle kaikki robotin välivaiheet ja niiden kestot sekunneissa. Yksi välivaihe ”open browser” jäänyt pois, koska robotia käynnistämällä oikea web-sivusto aukeaa automaattisesti alle sekunnissa. Todellisen tuloksen arviointia varten lisäsin vielä yhden ”open UiPath and run robot”-välivaiheen alkuun. Luotujen laskujen uudelleen nimeäminen ja siirtäminen tapahtui nopeammin, koska niitä siirrettiin kerralla oikeaan paikkaan.

Kaikkien kolmen asiakaslaskun toimittaminen manuaalisesti kesti 1175 sekuntia eli 19 minuuttia ja 35 sekuntia. Keskimääräinen suoritus-aika per lasku oli 6 minuuttia 32 sekuntia. Pitää ottaa huomioon, että toimeksiantaja itse tekee asiakaslaskuttaminen ja hänen suoritus-aikansa on nopea. Jos tämän tekisi toinen työntekijä, niin laskuttamisprosessin kesto olisi pidentynyt merkittävästi.

Pilottirobottia käyttämällä kolmen asiakaslaskun toimitus kesti 655 sekuntia eli 10 minuuttia 55 sekuntia. Keskimääräinen suoritus aika per lasku oli 3 minuuttia 38 sekuntia. Näin ollen suoritus aika laski melkein kaksinkertaisesti. Kuvio 13 havainnollistaa saadut 3 laskun suoritusajat.



Kuvio 13: 3 laskun suoritusajat.

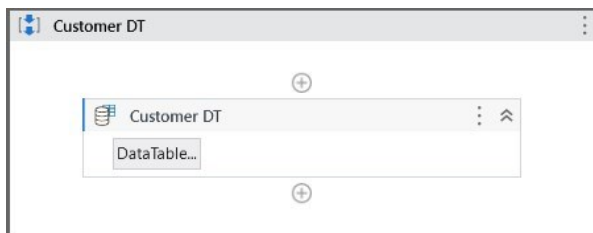
Periaatteessa ohjelmistorobotti tekee saman tehtävän samoilla työkaluilla ja samojen rajoitusten kautta, kuten fyysisesti oleva työntekijä. Mutta se ei pidä taukoja, ei erehdy eikä väsy ollenkaan. Seuraavaksi esittelin toimeksiantajalle valmiin prototyypin robotista, joka osasi täyttää oikeita asiakastietoja web-palvelimessa yksi lasku kerrallaan, kuten hänkin tekee laskun luomisen aikana. Toimeksiantaja hyväksyi pilottirobotin ja kehutti minua jatkamaan tuotantorobotin toteuttamista.

3.4 Tuotantorobotti

Tässä osassa kerrotaan varsinaisen tuotantorobotin tuottamisesta. Ensin käänsin edellisen pilottirobotin kaikki muuttujat ja sekvenssit englanniksi hyvän ohjelmointitavan mukaisesti. Seuraavaksi päivitin tietokannat. Tämän jälkeen loin kaksi alirobottia, joiden avulla Internet-yhteys voi tarkistaa ja asiakaslaskuja lähettää sähköpostitse. Sitten muokkasin päärobotin rakennetta ja sen arkkitehtuuria sujuvammaksi kokonaisuudeksi ja kytkin siihen alirobotit. Viimeistelyn vaiheessa kävin jokaisen aktiviteetin kommentoimassa ja samalla poistamassa ylimääräiset muuttujat. Sen jälkeen testasin valmiin ohjelmistorobotin toiminnan, jossa oli 10 asiakaslaskun jono. Viimeiseksi arvioin valmiin tuotantorobotin tehokkuutta vertailutaulukossa ja esittelin tulokset hyväksyttäväksi toimeksiantajalle.

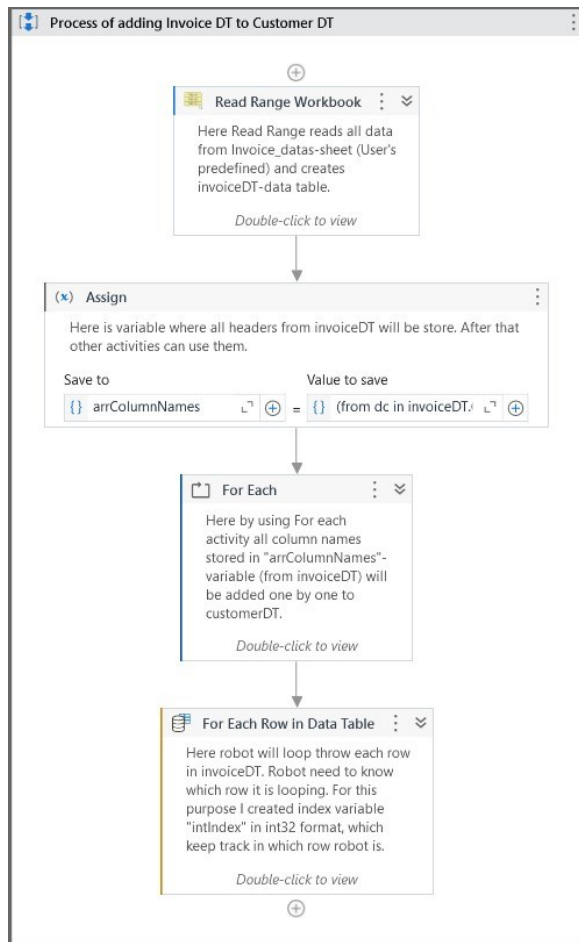
3.4.1 Tietokantojen kehittäminen

Tietokantojen kehittämisen aikana poistin Excel-tiedoston, jossa olivat asiakkaan omat tiedot. Sen sijaan loin robotin oman Data Taulukon Customer DT-sekvenssissä. Näin robotin ei enää tarvitse avata ja lukea kaksi erillistä Excel-tiedostoa vaan yhden, jossa on laskun omat tiedot. Asiakkaiden määrä on pysyvä ja uuden asiakasrivin lisääminen sisäiseen Data Taulukon on helppo. Samalla arkaluonteisia tietoja ovat hyvin suojassa. Alla on kuvio 14, jossa on Customer DT-sekvenssi esillä.



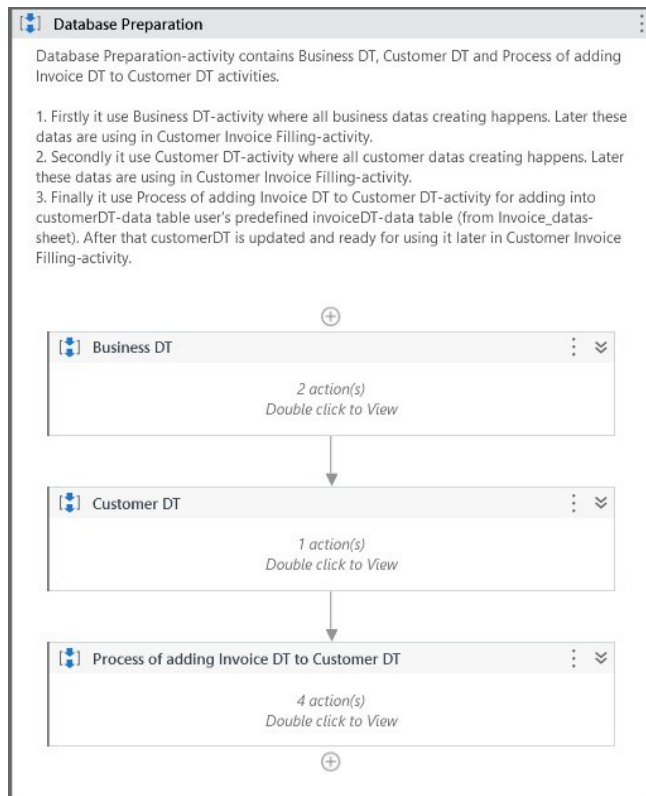
Kuvio 14: Customer DT-sekvenssin kuvaus. (RPA_Invoice_Robot, Main.xaml)

Seuraavaksi robotti lukee laskun omat tiedot Read Range Workbook-aktiviteetilla ja mutkikkaalla rakenteella lisää niitä aikaisempaan Customer DT-taulukkoon. Tällä tavalla robotti yhdistää kaikki asiakasdatat päivittämällä Customer DT-taulukon. Näin robotilla on käytössä kaksi sisäistä taulukkoa, joista se hakee arvot laskupohjan täyttämiseksi. Kuviossa 15 on esitetty, kuinka Customer DT-taulukon päivittäminen tapahtuu omassa sekvenssissä.



Kuvio 15: Customer DT-sekvenssin päivittäminen. (RPA_Invoice_Robot, Main.xaml)

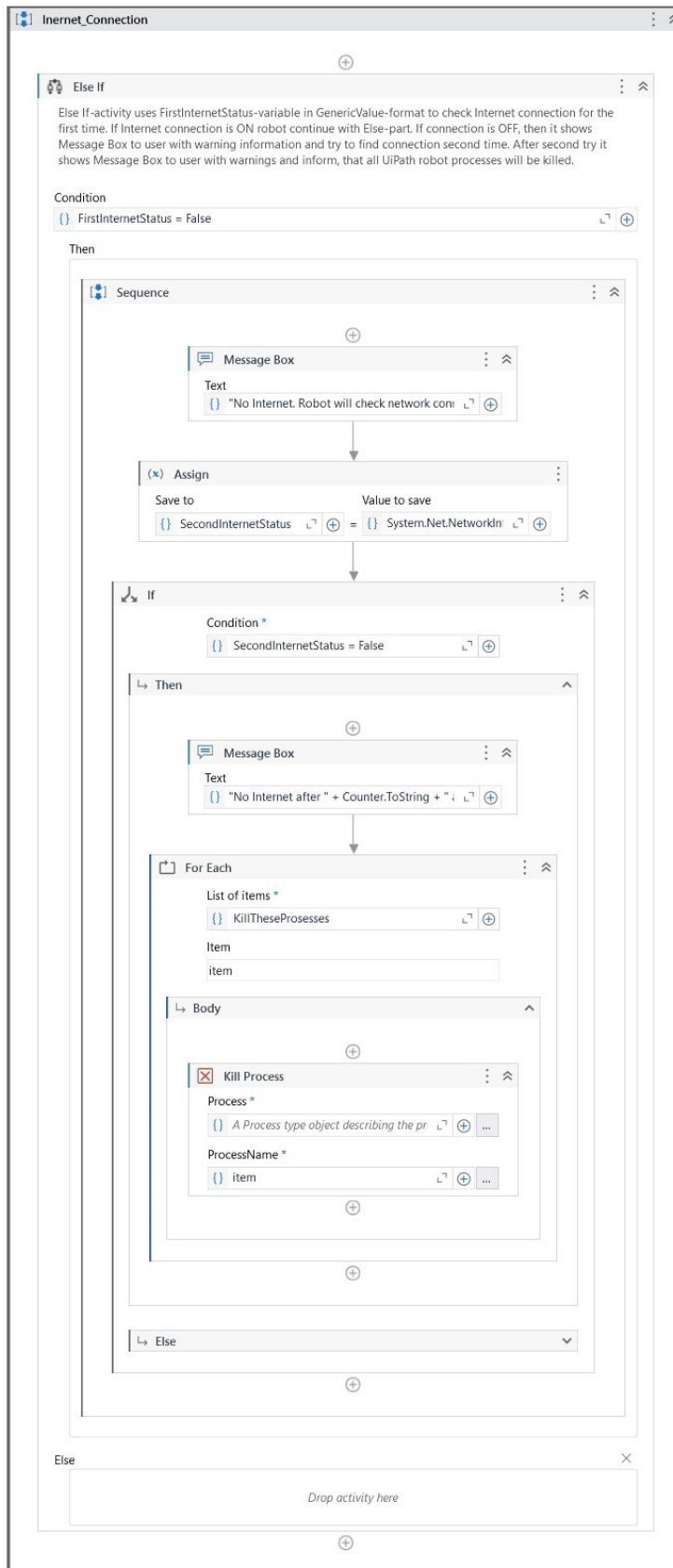
Laitoin kaikki tietokantoihin liittyvät sekvenssit omaan Database Preparation-sekvenssiin, jotta erottaa niitä muista päärobotin prosesseista. Alla kuviossa 16 on esitetty tämä sekvenssi.



Kuvio 16: Database Preparation-sekvenssin kuvaus. (RPA_Invoice_Robot, Main.xaml)

3.4.2 Alirobottien luominen

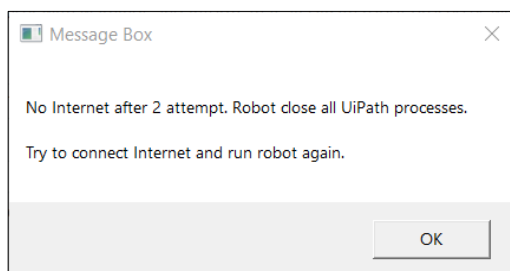
Ensin päärobotti käynnistää Invoke-aktiviteetilla ensimmäisen Inernet_Connection-alirobotin, jolla tapahtuu Internet-yhteyden tarkistus. Tämä alirobotti ei palauta eikä vastaanota mitään argumentteja. Argumentiksi tässä sanotaan muuttuja, jolla voi välittää arvot projektista toiseen. Käytin Else If-aktiviteetin ehtorakenteen valmistamiseksi. Sinne lisäsin GenericValue-tyyppisen FirstInternetStatus-muuttujan. Päärobotin prosessi jakautuu sen mukaan, onko Internet-yhteys päällä tai ei. Jos nettiyhteys on päällä, niin robotti kulkee Else-osioon ja sieltä palaa takaisin päärobottiin. Jos sitä ei ole, niin se käynnistää pari verkkoyhteyden tarkistusta ja sulkee kaikki UiPath-prosessit. Kuvio 17 esittelee tämä Inernet_Connection-alirobotin rakennetta.



Kuvio 17: Internet_Connection-alirobotin kuvaus. (Internet_Connection, Main.xaml)

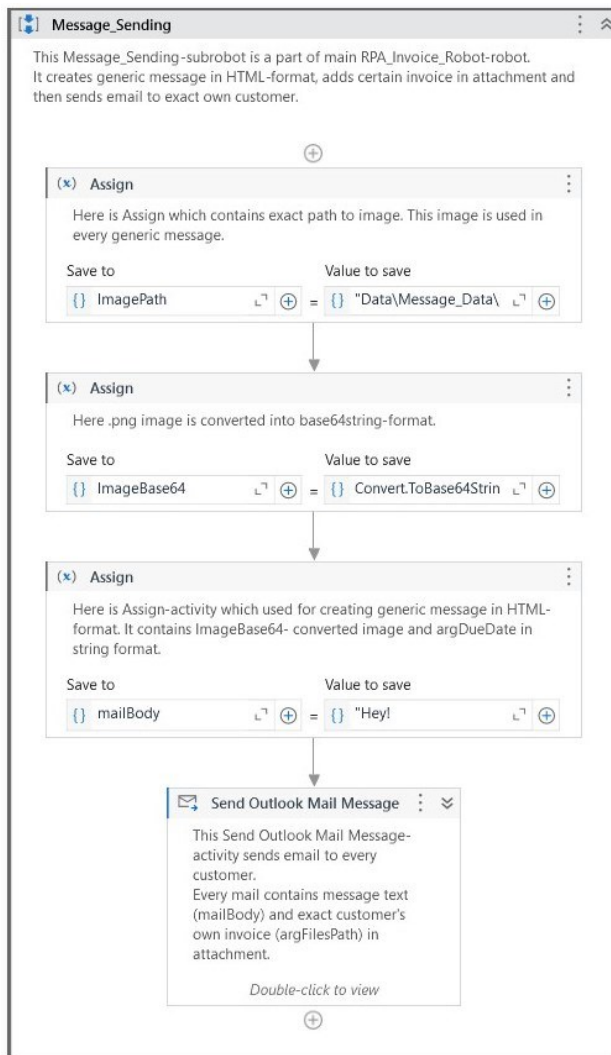
Mikäli Internet-yhteyttä ei ole pysyvästi, niin FirstInternetStatus-muuttujan arvo on False ja robotti ilmoittaa käyttäjälle Message Box:lla ”No Internet. Robot will check network connection again and continue process running, otherwise it close all processes.”.

Tämän jälkeen robotti tarkistaa Internet-yhteys vielä toisen kerran. Jos yhteyttä ei ole toisen yrityksen jälkeen, robotti kirjoittaa käyttäjälle viestin ”No Internet after 2 attempt. Robot close all UiPath processes. Try to connect Internet and run robot again.”. Seuraavaksi se lopettaa toimintansa käynnistämällä Kill Process-aktiviteetin For Each loopin kautta, koska pakko ”tappaa” 2 erillistä prosessia (”UiPath.Studio” ja ”UiPath.Executor”). Kuviossa 18 on esitetty viesti käyttäjälle toisen epäonnistumisen jälkeen.



Kuvio 18: Viesti käyttäjälle. (Internet_Connection, Main.xaml)

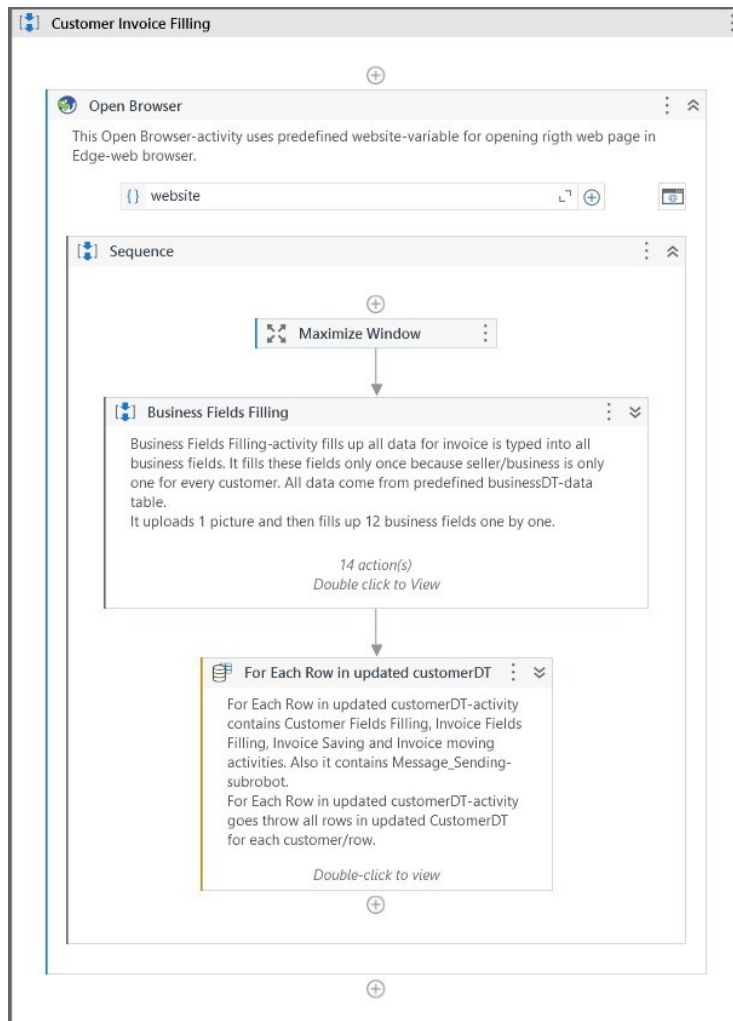
Seuraavaksi loin toisen Message_Sending-alirobotin, jolla sähköpostiviestin lähettäminen suorituu. Päärobotti kutsuu sen Invoke-aktiviteetilla ja toimittaa sille kaksi String-tyyppistä argumenttia. Ensimmäisessä argDueDate on laskun eräpäivä ja toisessa argFilePath on luotujen laskujen tarkka hakupolku. Kun alirobotti käynnistyy, se hakee ensisijaisesti logokuvan String-tyyppisestä ImagePath-muuttujasta ja muuttaa sen Base64-tyyppiseksi. Tällä toiminnolla voi kuvan upottaa tekstiin. Sitten on Assign-aktiviteetti, jossa on viestin standarditeksti valmiina. Siellä koodasin eräpäivän ja logokuvan suoraan tekstikenttään. Viimeinen aktiviteetti on Send Outlook Mail Message. Tässä määritellään sähköpostiviestin lähettäjän osoite, viestin otsikko ja teksti, sekä liitteen. Jokaiseen kirjeeseen liitetään oikean laskun PDF-muodossa ja lähetetään sähköpostikirjeen asiakkaille. Robotti ei lähetä viestit asiakkaille, vaan tallentaa ne Outlook-sovelluksen Luonnokset-kansioon manuaalitarkastusta varten. Kuviossa 19 on esitetty Inernet_Connection-alirobotin rakenne.



Kuvio 19: Message_Sending-alirobotin kuvaus. (Message_Sending, Main.xaml)

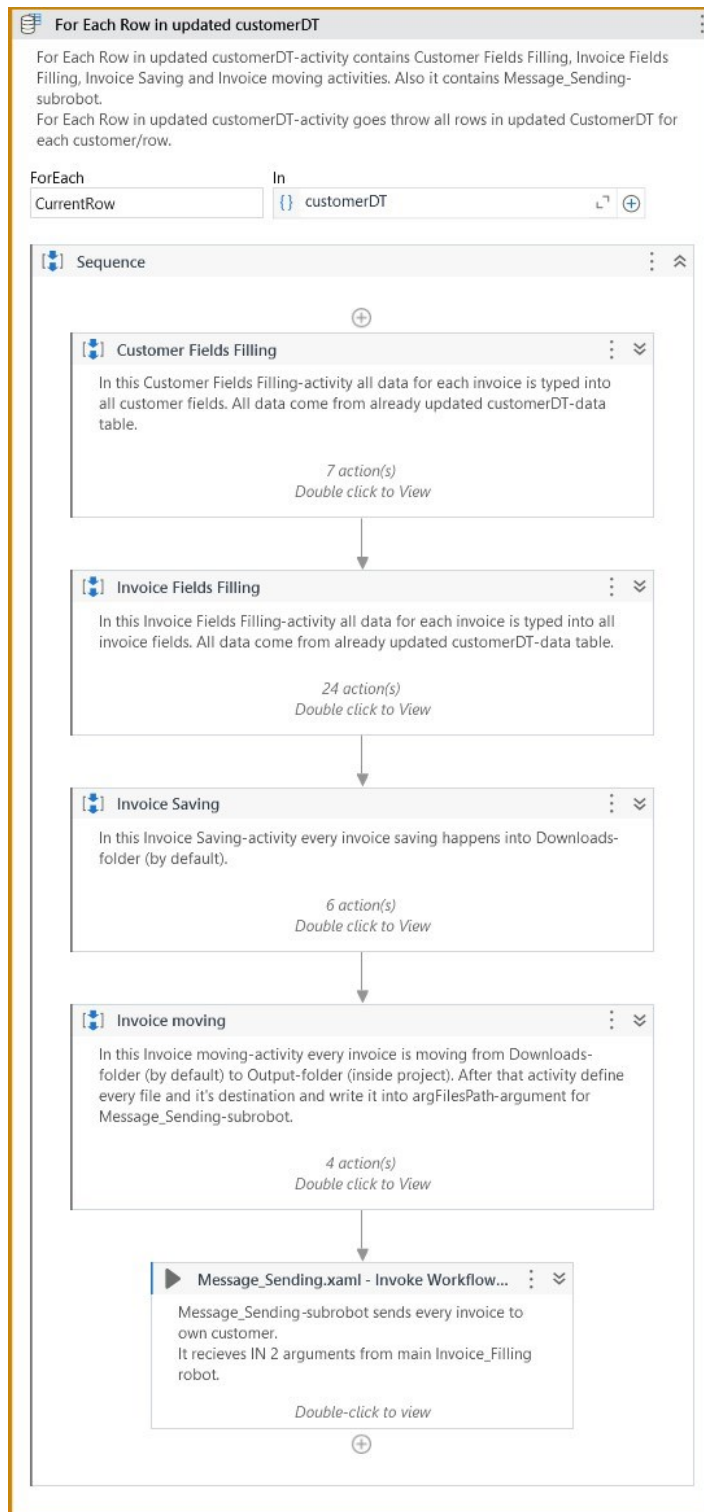
3.4.3 Päärobotin toteutus

Aloitin päärobotin toteutuksen muuttamalla pilottirobottia. Ensin muutin päärobotin kulkua, jakamalla Customer Invoice Filling-aktiviteetin kahteen osaan. Ensimmäisessä osassa robotti täyttää yrityksen omat tiedot ja laittaa niitä myyjän sarakkeisiin. Tämän se tekee vain kerran Business Fields Filling-aktiviteetissa. Tämän mekanismin takia myyjän tiedot ei tarvitse jokaisessa laskussa täyttää uudestaan. Kaikille asiakkaille on aina sama myyjä. Kuviossa 20 on esitetty Customer Invoice Filling-aktiviteetti.



Kuvio 20: Customer Invoice Filling-aktiviteetin kuvaus. (RPA_Invoice_Robot, Main.xaml)

Toisessa osassa on For Each Row in updated customerDT-aktiviteetti, joka itsessään sisältää vielä 5 aktiviteettiä. Tällä aktiviteetilla varmistetaan, jotta robotti täyttää laskupohjaan kaikki asiakkaat. Sillä For Each Row-rakenteelle ei ole väliä, miten paljon asiakkaita on jonoissa. Vaan se kirjoittaa jokaisen tiedot päivitetyistä aikaisemmin Customer DT-taulukosta yksi kerrallaan. Kuviossa 21 on For Each Row in updated customerDT-aktiviteetti.



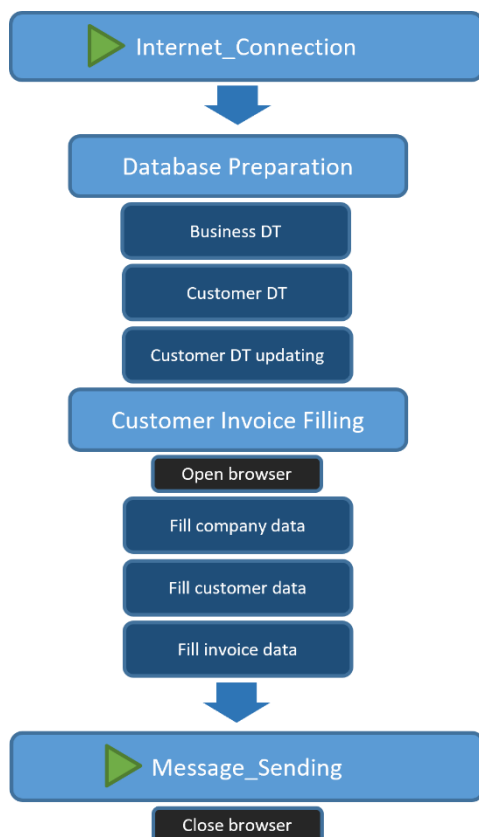
Kuvio 21: For Each Row in updated customerDT-aktiviteetin kuvaus. (RPA_Invoice_Robot, Main.xaml)

Näiden viiden aktiviteetin joukosta ensimmäisenä on Customer Fields Filling-aktiviteetti, jossa robotti täyttää asiakkaan tiedot. Toisena on Invoice Fields Filling-aktiviteetti, jossa jokaisen laskun omat tiedot kirjoitetaan.

Kun kaikki asiakkaan ja laskun tiedot syötetty laskupohjaan robotti jatkaa kulkukaavan mukaan seuraavaan Invoice Saving-aktiviteettiin. Siinä tapahtuu laskun tallentaminen oikealla nimellä. Nimen robotti generoi laittamalla kaksi erillistä muuttujaa yhteen stringCusName-muuttujaan ja erottamalla niitä pilkulla. Laskujen uudelleen nimeäminen on hyvä tulevaa arkistointia ja kirjanpitoa varten. Toki asiakkaallekin on mielekkäämpi saada laskun, mistä näkee hänen nimensä, tai vaikka palvelun-nimensä, ja eräpäivän aika. Näin asiakaskin pystyy käsitellä saamansa laskua helposti. Tällä pienellä toimilla saadaan parantaa asiakkaan tyytyväisyyttä ja samalla helpottaa saatujen laskujen monitorointia jatkossa.

Neljäntenä alkaa Invoice moving-aktiviteetti. Siellä äsken luotu oikeaniminen lasku siirtyy stringFilePathTo-muuttujan avulla Output-kansioon robottiprojektin sisälle. Tällä tavalla vältetään mahdollinen laskun poisto Downloads-kansiosta.

Viimeisenä aktiviteettina on Message_Sending-alirobotin kutsu Invoke-aktiviteetilla. Tämän alirobotin kulkuprosessista oli kerrottu aikaisemmassa luvussa. Kun kaikki nämä prosessit ovat valmiiksi suoritettu niin robotin sykli alkaa uudelleen siihen asti, kun kaikki laskut ovat tehty ja sähköpostikirjeet lähetetty. Lopussa robotti sulkee selaimen Close Application-aktiviteetilla. Kuviossa 22 on esitetty koko tuotantorobotin prosessit kaavalla.



Kuvio 22: Tuotantorobotin prosessien kuvaus.

Toimeksiantajan vastuulle jää sähköpostien manuaalinen tarkistus ja lähetys. Lähettäminen voi automatisoida ottamalla ruksin pois Message_Sending-alirobotin Luonnokset-kansion kohdalla.

3.4.4 Viimeistely

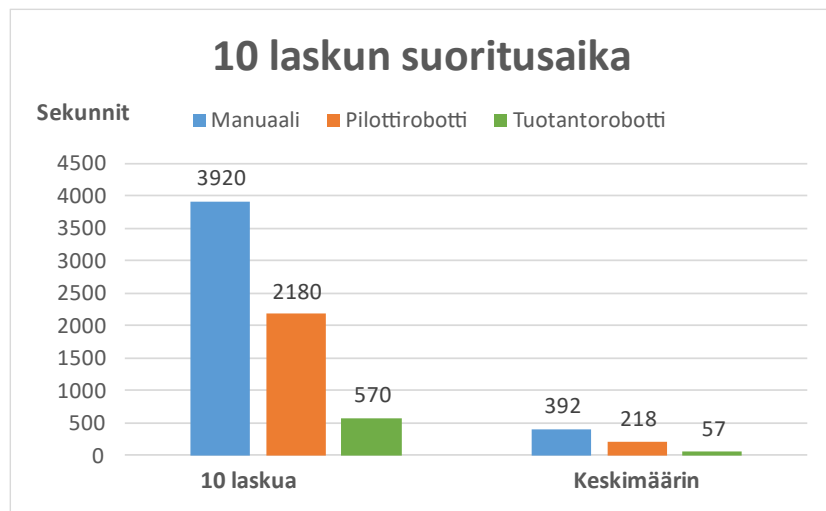
Viimeistelyn aikana lisäsin tuotantorobotin jokaiseen aktiviteettiin kuvaavia kommentteja. Samalla kävin nimeämässä aktiviteetteja uusiksi, jotta niiden nimitykset vastaisivat sisällön. Ilman kommentteja olisi hyvin vaikeaa sanoa, mikä tietyn aktiviteetin idea tai tavoite oli. Tietosuojaan mielessä robotti avaa selaimen incognito-tilassa. Asiakkaan omat tiedot ovat robotin omassa tietokannassa eikä niitä näy missään muualla. Excel-tiedosto, josta robotti hakee laskun omat tiedot, on salattu. Eli sen avaamisessa robotti käyttää salasanan. Tämä tarkoittaa, että siihen ei ole pääsyä ulkopuolisille. Lisäsin robottiin useita Log Message-aktiviteetteja kulkuprosessin kriittisissä kohteissa. Näiden avulla käyttäjä voi seurata robotin ajoa tarvittaessa. Mahdollista virheenkäsittelyä varten lisäsin pari Try Catch rakennetta. Näillä rakenteilla ohjelma ottaa kiinni tiettyjä virheitä tai poikkeuksia ja käsittelee niitä eikä kaadu saman tien poikkeustilanteessa. Jos ajon aikana virhe ilmestyy, niin Exception-osiolla käyttäjä saa esiin virheilmoituksen. Samanlainen viesti lähtee lokiin Log Message:lla debuggausta varten. Lisäsin vielä muutamia Timer-aktiviteetit, näiden avulla kulkuprosessin aikana robotti itse laskee tiettyjen prosessien kestot. Tämän hyödynnän, kun teen kustannuslaskelman tälle projektille. Viimeiseksi kirjoitin lyhyeen oppaan RPA_Invoice_Robot-tuotantorobotin käyttäjälle. Tässä ohjeessa kerrotaan, miten robotin prosessit toimivat ja mitä ohjelmia koneelle pitää asentaa.

3.4.5 Kustannuslaskenta

Tuotantorobotin testausvaiheen jälkeen lisäsin vertailutaulukkoon kaikki robotin välivaiheet ja niiden kestot sekunneissa omalle sarakkeelle. Välivaihe ”fill company data” jäänyt vaan ensimmäiselle laskulle. Kaikki muut 9 laskua valmistelemissa robotti ei koske laskupohjan myyjän puolta.

Tarkan tuloksen arviointia varten lisäsin yhden ”internet connection”-välivaiheen alkuun. Tämä tuli Internet_Connection-alirobotin lisäämisestä. Sen vaikutus tuotantorobotin suoritus-aikaan on mitätön, koska robottia käynnistämällä nettiyhteyden tarkistus tapahtuu alle sekunnissa. Robotti lisää tiedon yhteydestä lokiin Log Message-aktiviteetilla eikä käyttäjä edes huomaa sen.

Kaikkien 10 asiakaslaskun toimittaminen automaattisesti kesti yhteensä 570 sekuntia eli 9 minuuttia ja 30 sekuntia. Keskimääräinen suoritus-aika per lasku oli vaan 57 sekuntia. Näin ollen tuotantorobotin tehokkuus lisääntynyt merkittävästi, vaikka laskuja oli 3 kertaa enemmän. Kuvio 23 havainnollistaa saadut 10 laskun suoritusajat.



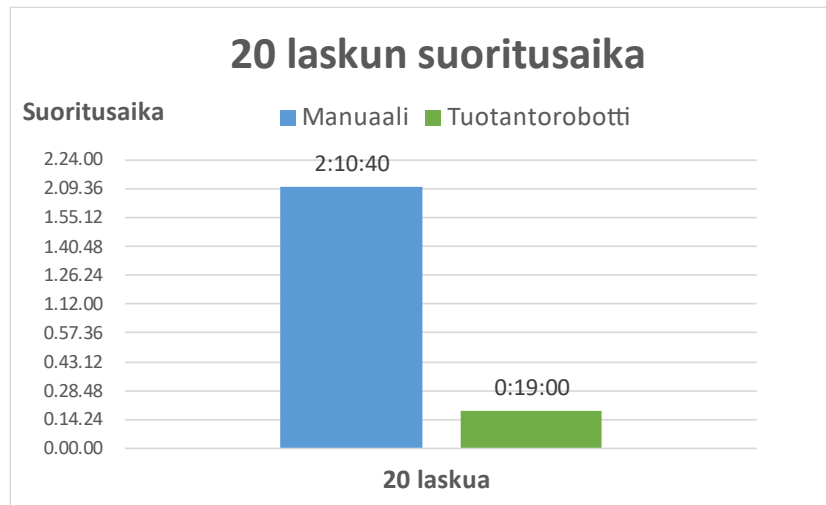
Kuvio 23: 10 laskun suoritusajat.

Vaiheiden suoritus nopeutui, koska robotin ei tarvinnut syöttää myyjän tietoja joka kerta uudestaan. Kun tuotantorobotti oli täysin käyttöönotettu ja testattu, esittelin sen toimeksiantajalleni taas. Hän yllättyi positiivisesti robotin tehokkuudesta ja hyväksyi tämän projektin.

4 Tulokset

Tämän kehittämistyön tuloksena syntyi tehokas tuotantorobotti, joka mahdollisti asiakaslaskutusprosessin automatisoinnin. Vuokralaskujen toimittaminen robotin käyttöönoton jälkeen suoriutuu huomattavasti nopeammin. Muiden laskujen toimitus jää edelleen toimeksiantajalle, koska niitä ei voi tehokkaasti automatisoida. Silti niiden määrä ei ole suuri ja siksi tuotantorobotin vaikutus mikroyrittäjän yksittäiseen liiketoimintaprosessiin on valtava.

Toimeksiantaja lähettää laskuja asiakkaille noin 20-25 kappaletta kuukaudessa. Näin 20 asiakaslaskun toimittaminen manuaalisesti olisi kestänyt vähintään $20 \times 392 = 7840$ sekuntia tai 2 tunti 10 minuuttia ja 40 sekuntia. Automatisoinnin jälkeen koko prosessin suoritus aika oli vaan 19 minuuttia. Asiakaslaskutusprosessi nopeutui melkein 7 kertaa. Samalla tuotantorobotilla ei ole inhimillisten virheitä. Suurella todennäköisyydellä manuaalisen laskuttamisen aikana olisi syntynyt näpyttelyvirheitä. Näiden aiheuttamista robotti välttää ja sillä sen tuottavuus on vielä parempi. Alla on kuvio 24, jossa on esitetty tuotantorobotin suoritus aika verrattuna manuaaliseen suoritus aikaan yhden kuukauden tasolla.



Kuvio 24: 20 laskun suoritusajat.

Koko ohjelmistorobotin toteuttamisaikana oli kulutettu 3 työpäivää, mikä tekee 22,5 työtuntia. Lasketaan robotin takaisinmaksuaika seuraavasti: kokonaan kulutettu aika jaetaan säästetyllä työajalla.

Säästetty työaika kuukaudessa saadaan erottamalla ihmisen suoritusajasta robotin suoritusajaa. Tämä tekee 1 tunti 51 minuuttia ja 40 sekuntia. Muutetaan nämä ajat sekunneiksi laskevista varten ja saadaan 81000 sekuntia ja 6700 sekuntia vastaavasti. Jaetaan $81000 / 6700 =$ noin 12,1 eli 12 kuukauden jälkeen robotti maksaa itsensä täysin takaisin.

Todennäköisesti, ettei toimeksiantaja pystyy suorittamaan laskuttamista ilman pientäkään taukoa. Jos tauot ja erilaiset keskeytykset otetaan huomioon laskennassa, niin silloin manuaaliseen suorittamiseen kuluva aika pidentyy. Tämän myötä robotin kustannustehokkuus korostuu ja takaisinmaksuaika lyhenee huomattavasti.

Ohjelmistorobotin käyttöönotto toi mukanaan myös kustannussäästöjä. Näin toimeksiantajan tuntiveloitus on 70 euroa tunnissa ilman ALV:tä. Silloin hänen rahansa säästö kuukausitasolla on 1 tunti 51 minuuttia ja 40 sekuntia kertaa 70 euroa = noin 130,28 euroa.

Toimeksiantajani oli aika yllättynyt kuinka helpolla tavalla nyt asiakaslaskutus voi olla hänelle. Eikä hänen enää tarvitse murehtia siitä, että hän voin näpyttelyvirheen tehdä vahingossa. Hän oli tyytyväinen, että voi panostaa enemmän tarjousten tekemiseen ja muutenkin työn tekemiseen kuin kuukausittaiseen rutiiniomaiseen asiakaslaskutukseen.

5 Pohdinta ja jatkokehitykset

Hyvänä kehityskohtana asiakaslaskutusrobotille voi olla ajastettu robotin käynnistys UiPath Orchestrator-palvelinalustalla. Toimeksiantaja oli pyytännyt minua heti kartoittamaan seuraavia prosesseja, joita voi automatisoida samalla työkalulla. Seuraavia liiketoimintaprosesseja, joita olisi tarpeellista automatisoida on tarjousten lähettäminen asiakkaille. Tämä vie suuressi aikaa toimeksiantajan työajalta ja yleisesti pilkkutarkkaa työtä. Se vaatii asiakkaan tarpeiden tuntemista ja niiden mukaan sitten asiakastarjous lähetettään. Tämä robotti voi tuottaa tarjouksen asiakkaalle sen perusteella, minkä tuotteen asiakasta kiinnostaa.

Asiakkaille tarjousten lähettämisessä on jotain samoja vaiheita kuin myös asiakaslaskutuksessakin. Ja kyllä siinä voi tämän tuotantorobotin osia käyttää. Mutta tämä uusi projekti vaatii taas omien laajempien tietokantojen luontia ja tarkempaa suunnittelua. Koska robotin kulkevia polkuja voi olla paljon enemmän, kuin tavallisessa asiakaslaskutuksessa. Siellä kaikki tarvittavat datat jo valmiiksi tiedossa ja valmiiksi omassa tietokannassa.

Ehkä tulevassa ohjelmistorobotissa voisi kokeilla käyttää tekoälyä, koska prosessien tuottaminen ei ole enää niin selvää ja yksinkertaistaa, kuten edellisessä esimerkissä. Mutta tuo on jo ihan toinen projekti. Siihen voisi jatkossa käyttää jotain valmiiksi tässä projektissa luotuja tietokantoja ja käyttää jotain samoja tekniikoita ja samoja periaatteita kuten edellisessäkin ohjelmistorobotti-projektissa. Valmiin opinnäytetyön esitin toimeksiantajalleni ja hän hyväksyi sen.

Lähteet

Painetut

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2009. Tutki ja kirjoita. Hämeenlinna: Kariston kirjapaino Oy.

Jormakka, R., Koivusalo, K., Lappalainen, J., Niskanen, M. 2015. Laskentatoimi. 4. uudistettu painos. Helsinki: Edita.

Kaarlejärvi, S. & Salminen, T. 2018. Älykäs taloushallinto - automaation aika. Helsinki: Alma Talent.

Kananen, J. 2015. Laadullinen tutkimus opinnäytetyönä. Näin kirjoitan opinnäytetyön tai pro gradun alusta loppuun. Jyväskylä : Jyväskylän ammattikorkeakoulu.

Laukkonen, V. & Kügelgen, M. 2020. Kaikki koodaa. Päivitä itsesi - Käytännön opas ajankohtaisiin digitaitoihin. Helsinki: Into Kustannus Oy.

Sähköiset

Andersson, C., Haavisto, I., Kangasniemi, M., Kauhanen, A., Tikka, T., Tähtinen, L. & Törmänen, A. 2016. Robotit töihin. Eva-raportti 2/2016. Helsinki: Nextprint Oy.

<http://www.eva.fi/wp-content/uploads/2016/09/Robotitt%C3%B6ihin.pdf>

Automation Anywhere 2022. Products. Community Edition. Viitattu 17.5.2022.

<https://www.automationanywhere.com/products/enterprise/community-edition>

Blue Prism 2022. SS&C Blue Prism Free Trial. Viitattu 17.5.2022. <https://www.blue-prism.com/free-trial>

CGI 2021. Älykäs automaatio. Ohjelmistorobotiikka. Viitattu 18.11.2021.

<https://www.cgi.com/fi/fi/alykas-automatio/ohjelmistorobotiikka>

Digital Workforce 2017. GDPR vaateet käytäntöön robotiikalla. Viitattu 18.7.2022. <https://digitalworkforce.com/fi/rpa-blogi/gdpr-vaateet-kaytanton-robotiikalla/>

Euroopan komissio 2017. Pohdinta-asiakirja globalisaation hallinnasta. Euroopan unioni. Bryssel. Viitattu 21.11.2022. https://ec.europa.eu/info/sites/default/files/reflection-paperglobalisation_fi.pdf

Investopedia 2022. Compound Annual Growth Rate (CAGR) Formula and Calculation. Viitattu 18.09.2022. <https://www.investopedia.com/terms/c/cagr.asp>

Isolta 2021a. Ominaisuudet. Viitattu 18.11.2021. <https://www.isolta.fi/ominaisuudet/>

Isolta 2021b. Hinnat. Viitattu 18.11.2021. <https://www.isolta.fi/hinnat/>

Knowit 2021. Älykäs automaatio. Ohjelmistorobotiikka (RPA). Viitattu 18.11.2021. https://www.knowit.fi/palvelut/solutions/alykas-automaatio/robotic-process-automation/?gclid=CjwKCAiAs92MBhAXEiwAXTi25-dcGg3opX0b4t1UQksQoA51SayWzNF22NOEWTNa33FGjWdfGDcoYxoCdNsQAvD_BwE

Mtech 2021. Palvelut. Viitattu 18.11.2021. <https://www.mtech.fi/palvelut/>

Oulun yliopisto 2020. Mikroyritysten ja työpaikkojen määrä nousi ensimmäisenä koronavuonna - Oulun yliopiston mikroyritysten tilastokatsaus vuosilta 2013-2020 julkaistu. Viitattu 18.9.2022. <https://www oulu.fi/fi/uutiset/mikroyritysten-ja-tyopaikkojen-maara-nousi-ensimmaisena-koronavuonna-oulun-yliopiston-mikroyritysten>

Pilkahduksia tulevaisuuteen - digitalisaation ja robotisaation mahdollisuudet. 10/2017. Valtiovarainministeriö, Helsinki 2017 https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/79260/Pilkahduksia_tulevaisuuteen.pdf?sequence=1&isAllowed=y

Research and Markets 2022. Robotic Process Automation Market Size, Share & Trends Analysis Report by Type, by Service, by Application, by Deployment, by Organization, by Region, and Segment Forecasts, 2022-2030. Viitattu 11.10.2022. https://www.researchandmarkets.com/reports/4668038/robotic-process-automation-market-size-share-and?utm_source=BW&utm_medium=PressRelease&utm_code=tcg6ll&utm_campaign=1601657+-+Global+Robotic+Process+Automation+Market+Analysis+%26+Forecast+Report+2021%3a+Market+Size+is+Expected+to+Reach+%2413.74+Billion+by+2028%2c+Registering+a+CAGR+of+32.8%25&utm_exec=chdo54prd

SimplBooks 2021a. Ominaisuudet. Viitattu 18.11.2021. <https://www.simplbooks.fi/ominaisuudet/>

SimplBooks 2021b. Hinnat ja käyttöönotto. Viitattu 18.11.2021. <https://www.simplbooks.fi/hinnat-ja-kayttoonotto/>

Taulli T. 2020. Robotic Process Automation Handbook: A Guide to Implementing RPA Systems. E-kirja. New York: Apress. Viitattu 22.11.2022

Tilastokeskus. Mikroyritys. Viitattu 18.9.2022. <https://www.stat.fi/meta/kas/mikroyritys.html>

UiPath 2022a. Product. Studio. Viitattu 17.5.2022. <https://www.uipath.com/product/studio>

UiPath 2022b. Resources. Automation Case Studies. Viitattu 17.5.2022.

<https://www.uipath.com/resources/automation-case-studies>

UiPath 2022c. Partners. Find a Technology Partner. Viitattu 17.5.2022.

<https://www.uipath.com/partners/technology-alliances>

UiPath 2022d. Trust and Security. Security. Viitattu 17.5.2022. <https://www.uipath.com/legal/trust-and-security/security>

UiPath. 2022e. Free Trial vs. Community. Viitattu 17.5.2022. <https://www.uipath.com/freetrial-or-community>

UiPath. 2022f. Studio. Hardware and Software Requirements - Standalone 2022.4. Viitattu

17.5.2022. <https://docs.uipath.com/studio/standalone/2022.4/user-guide/hardware-and-software-requirements>

Wikipedia. UiPath. Viitattu 19.5.2023. <https://en.wikipedia.org/wiki/UiPath>

Zervant 2021a. Ominaisuudet. Viitattu 18.11.2021. <https://www.zervant.com/fi/premium-palvelut/>

Zervant 2021b. Hinnasto. Viitattu 18.11.2021. <https://www.zervant.com/fi/hinnasto/>

Kuviot

Kuvio 1: Maailman RPA (Robotic Process Automation) markkinakoon ennuste.	9
Kuvio 2: Kehittämistyön prosessin kuvaus.	12
Kuvio 3: Asiakaslaskutusprosessin kuvaus.	13
Kuvio 4: UiPath Studion toiminnan kuvaus.	18
Kuvio 5: UiPath Studion käyttöliittymä.	18
Kuvio 6: UiPath Orchestratorin käyttöliittymä.	19
Kuvio 7: UiPath Robot.	19
Kuvio 8: Esimerkki luodusta aktiviteettikaaviosta UiPath Studiossa.	20
Kuvio 9: DataBaseMyyjänTiedot-sekvenssin kuvaus. (LaskuRobotti, Main.xaml, DataBaseMyyjän-Tiedot)	21
Kuvio 10: DataBaseAsiakkaanTiedot-sekvenssin kuvaus. (LaskuRobotti, Main.xaml, DataBaseAsiakkaanTiedot)	22
Kuvio 11: DataBaseLaskunTiedot-sekvenssin kuvaus. (LaskuRobotti, Main.xaml, DataBaseLaskunTiedot)	23
Kuvio 12: Koko pilottirobotin prosessien kuvaus. (LaskuRobotti, Main.xaml)	24
Kuvio 13: 3 laskun suoritusajat.	25
Kuvio 14: Customer DT-sekvenssin kuvaus. (RPA_Invoice_Robot, Main.xaml)	26
Kuvio 15: Customer DT-sekvenssin päivittäminen. (RPA_Invoice_Robot, Main.xaml)	27
Kuvio 16: Database Preparation-sekvenssin kuvaus. (RPA_Invoice_Robot, Main.xaml)	28
Kuvio 17: Internet_Connection-alirobotin kuvaus. (Internet_Connection, Main.xaml)	29
Kuvio 18: Viesti käyttäjälle. (Internet_Connection, Main.xaml)	30
Kuvio 19: Message_Sending-alirobotin kuvaus. (Message_Sending, Main.xaml)	31
Kuvio 20: Customer Invoice Filling-aktiviteetin kuvaus. (RPA_Invoice_Robot, Main.xaml)	32
Kuvio 21: For Each Row in updated customerDT-aktiviteetin kuvaus. (RPA_Invoice_Robot, Main.xaml)	33
Kuvio 22: Tuotantorobotin prosessien kuvaus.	34
Kuvio 23: 10 laskun suoritusajat.	36
Kuvio 24: 20 laskun suoritusajat.	37

Liitteet

Liite 1: Haastattelukysymykset, toimeksiantaja / asiakaslaskutus	44
--	----

Liite 1: Haastattelukysymykset, toimeksiantaja / asiakaslaskutus

Taustakysymykset:

1. Nimi, työpaikka
2. Työtehtävät, rooli yrityksessä
3. Työtausta lyhyesti
4. Koulutustausta

Haastattelukysymykset:

1. Miten tällä hetkellä asiakaslaskujen toimittaminen asiakkaille tapahtuu?
2. Mitä ohjelmia tai sovelluksia asiakaslaskujen luomisessa ja lähettämässä käytetään?
3. Missä muodossa laskuja ovat tällä hetkellä?
4. Missä paikassa ja muodossa asiakkaiden tietoja säilytetään?
5. Miten paljon samanlaisia laskuja lähetetään kuukaudessa?
6. Minkälaisia laskuja nämä ovat?
7. Kuka lähettää nämä laskut asiakkaille?
8. Mitkä ovat suurimmat haasteet asiakaslaskujen toimittamisessa asiakkaille?
9. Mitä toivot tästä ohjelmistorobotiikka-projektista?