

Toni Mänty

**Konkurssivalvonta ohjelmistorobotiikan avulla**

## **Konkurssivalvonta ohjelmistorobotiikan avulla**

Toni Mänty  
Opinnäytetyö  
Kevät 2020  
Tietojenkäsittely  
Oulun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietojenkäsittely, Järjestelmäasiantuntija

---

Tekijä(t): Toni Mänty

Opinnäytetyön nimi: Konkursseivalvonta ohjelmistorobotiikan avulla

Työn ohjaaja: Sinikka Viinikka

Työn valmistumislukukausi ja -vuosi: Kevät 2020

Sivumäärä: 34

---

Opinnäytetyö käsittelee ohjelmistorobotiikka ja sen hyödyntämistä konkurssivalvonnassa. Työn toimeksiantajana toimii MPY Yrityspalvelut Oyj, jonka palveluksessa toimin ohjelmistokehittäjän tehtävässä. Opinnäytetyön aikana tehty ohjelmistorobotti on osa pidempää kehitystyötä ohjelmistorobotiikan alalla ja toimi ensimmäisenä projektina minulle aiheen parissa.

Työn tavoitteena oli luoda toimiva ohjelmistorobotti, joka hoitaa konkurssivalvonnan toistuvat ihmistä tarvitsemattomat tehtävät. Ohjelmistorobotti hoitaa yritystunnusten tarkistamisen tietokannasta päivittäin ja luo näistä raportin tehtävää hallinnoivalle henkilölle. Työ on tehty osana yrityksessä hoitamaani työtehtävää sovelluskehittäjänä ja siinä on ollut mukana muita yrityksen henkilöitä sekä ulkopuolinen konsultti.

Opinnäytetyössä on käytetty tietolähteenä useita eri keskustelupalstoja, erityisesti UiPathin omaa keskustelupalstaa. Tietolähteiksi on löytynyt myös muutamia artikkeleita sekä aiemmin tehtyjä tutkimuksia aiheesta. Opinnäytetyö on projektikuvaus ohjelmistorobotin kehitystyöstä.

Raportin käsittelemä projekti on saatettu päätökseen ja robotti on todettu valmiiksi sekä otettu käyttöön tuotannossa. Projektista saatiin myös hyödyllistä tietoa ohjelmistorobottien kehitykseen tarvittavista tiedoista ja taidoista.

---

Asiasanat: RPA, UiPath, konkurssivalvonta, ohjelmistorobotti, ohjelmistorobotiikka

## ABSTRACT

Oulu University of Applied Sciences  
Business Information Systems, System specialist

---

Author(s): Toni Mänty

Title of thesis: Robotic Process Automation and bankruptcy control

Supervisor(s): Sinikka Viinikka

Term and year when the thesis was submitted: Spring 2020      Number of pages: 34

---

This thesis covers software robotics and its utilization in bankruptcy supervision. The practical work is commissioned by MPY Yrityspalvelut Oyj, where I work as a software developer. The software robot made during the thesis is part of a longer development work in the field of robotic process automation and was the first project for me on the subject.

The aim of the work was to create a functional software robot that handles the recurring non-human tasks of bankruptcy supervision. The software robot handles the verification of company IDs from the database daily and generates a report of these to the person managing the task. The software development work has been done as part of my job as an application developer in the company and has involved coworkers from other areas of the company as well as an external consultant.

Several different discussion forums have been used as a source of information in the thesis, especially UiPath's own discussion forum. A few articles as well as previous research on the subject have also been found as sources of information. The thesis is a project description of the development work of the software robot.

The project covered by the report has concluded and the robot has been set to production. The project also provided useful information on the knowledge and skills needed to develop software robots.

---

Keywords: Robotic Process Automation, RPA, UiPath, bankruptcy, software robot

# SISÄLLYS

1	JOHDANTO .....	6
2	OHJELMISTOROBOTIIKKA .....	7
2.1	Hyödyt .....	7
2.2	Haasteet ohjelmistorobotiikassa .....	8
2.3	Ohjelmistorobottialustat .....	10
3	OHJELMISTOROBOTIIKAN SOVELLUSKOHTEET .....	12
4	TOIMEKSIANTAJA JA TOIMEKSIANTO .....	15
5	TOIMEKSIANNON TOTEUTUS .....	16
5.1	Ohjelmistorobotin kehityksen lähtökohdat .....	16
5.2	Ohjelmistorobotin kehittämisen kulku .....	17
5.2.1	Robotin alustus .....	18
5.2.2	Selaimen avaaminen ja kirjautuminen .....	25
5.2.3	Tietojen varsinainen käsittely .....	28
5.2.4	Raportin luonti ja lähettäminen .....	29
6	TESTAUS JA PROJEKTIN LOPPUTULOS .....	30
7	POHDINTA .....	31
	LÄHTEET .....	32

# 1 JOHDANTO

Ohjelmistorobotiikka on yleistynyt tapa automatisoida yrityksen triviaaleja toimintoja, ja tuoda näin työntekemiseen lisää tehokkuutta. Ohjelmistoroboteilla voidaan tehostaa työn tekemistä, joko luomalla itsenäisiä automaattisia robotteja tai luomalla robotti, joka toimii luonnollisen käyttäjän kanssa yhteistyössä. Tehostamalla työtä tällä tavalla voidaan saavuttaa pitkällä aikavälillä huomattavia säästöjä, varsinkin manuaalisissa ja toistuvissa tehtävissä.

Ohjelmistorobotiikassa ei käytetä fyysisiä robotteja, kuten perinteisessä automaatiossa yleisesti käytetään. Ohjelmistorobotti on tietokoneella toimiva virtuaalinen käyttäjä, joka toistaa määrättyjä toimintoja automaattisesti. Automatisointi ei kuitenkaan tapahdu itsestään, vaan on tärkeää heti alkaessa varmistaa vankka pohja sekä osaamisen että suunnittelun saralla. Prosessien automatisoimiseen menee aikaa, mutta pitkällä tähtäimellä katsottaessa se on lähes aina kannattavaa.

Ohjelmistoroboteista on useita erilaisia alustoja, jotka poikkeavat toisistaan jonkin verran käyttötavoista kehitystyökaluihin. Robotteja voidaan käyttää joko yritystoimintojen automatisoimiseen tai sovelluskehityksessä testaukseen. Tässä opinnäytetyössä käsitellään UiPath ohjelmistorobotialustaa ja sen hyödyntämistä yrityksen konkurssivalvonnan suorittamiseen. Ohjelmistorobotin kehittämistyötä ohjasi työnantajan antama työtehtävä.

Opinnäytetyön tavoitteena oli luoda toimiva ohjelmistorobotti hoitamaan yrityksen konkurssivalvonnan toistuvia tehtäviä. Tässä raportissa kuvataan prosessin eteneminen vaihe vaiheelta ja robottia varten tehty prosessin pilkkominen tarkoiksi toimintaohjeiksi. Perimmäisenä tarkoituksena projektilla on vahvistaa osaamista ohjelmistorobotiikan alalla, ja samalla luoda perusta manuaalisten toistuvien tehtävien minimoimiseen. Näin voidaan vapauttaa työntekijöitä yksinkertaisista, rutiininomaisista ja toistuvista tehtävistä mielekkäämpiin ja haastavampiin tehtäviin.

## 2 OHJELMISTOROBOTIIKKA

*“Robotic Process Automation is the technology that allows anyone today to configure computer software, or a “robot” to emulate and integrate the actions of a human interacting within digital systems to execute a business process.” (UiPath, 2019)*

Ohjelmistorobotiikalla (RPA) tarkoitetaan toistuvien prosessien ja työtehtävien suorittamista ohjelmallisesti (Hyytinen, 2018). Sen tehtävänä on suorittaa rutiininomaisia tehtäviä vapauttaen samalla työntekijän näistä tehtävistä mielekkäämpiin ja haastavampiin työtehtäviin. Ohjelmistorobottien tarkoitus ei siis ole korvata työntekijöitä, vaan toimia työkaluna kohti tehokkaampaa ja mielekkäämpää työtä.

Ohjelmistorobotit käyttävät järjestelmiä ihmisten tavoin eli ne käyttävät hiirtä ja näppäimistöä järjestelmien ohjaamiseen, kuten luonnollinen käyttäjä tekisi. Ohjelmistorobotteja voidaan luoda toimimaan itsenäisesti suorittaen annettuja tehtäviä tai toimimaan tiettyjen käyttäjän tekemien komentojen yhteydessä. Itsenäisesti toimivat robotit tarvitsevat oman työympäristönsä, koska ne käyttävät koneen resursseja (esim. hiirtä ja näppäimistöä) kaiken aikaa, jolloin luonnollinen käyttäjä ei voi käyttää samaa ympäristöä omien tehtäviensä tekemiseen.

RPA on suhteellisen helppo ottaa käyttöön, mutta parhaan hyödyn saamiseksi ihmisen täytyy sovittaa se IT arkkitehtuuriin ja infrastruktuuriin. Ohjelmoituna RPA:n toimintakyky vastaa vähintään kolmea kokopäiväistä työntekijää; laadukkaasti, ilman virheitä, tehden montaa asiaa yhtä aikaa ja vaihtaen nopeasti prosessista toiseen. Se vaatii vahvaa sisäistä kykyä skaalautuakseen ja saavuttaakseen jatkuvaa kehitystä. (Willcocks & Lacity, 2015.)

### 2.1 Hyödyt

IT-alalla ja datan käsittelyssä on lukemattomia rutiininomaisia tehtäviä, jotka ovat luonteeltaan yksinkertaisia ja toistuvia. Ohjelmistorobotiikan tavoitteena on tuoda tehokas työkalu tällaisten tehtävien suorittamiseen, jolloin luonnolliset työntekijät vapautuvat näistä tehtävistä mielekkäämpiin tehtäviin. Työtehtäviä, joita automatisoidaan ohjelmistorobotiikalla voi olla esimerkiksi tietojen syöttäminen lomakkeille, tietojen siirto järjestelmästä toiseen tai tietojen hakeminen jostain lähteestä (Hyytinen, 2018).

Edellä mainitut esimerkit ovat luonteeltaan yksinkertaisia ja triviaaleja tapauksia, joiden tekemiseen ei tarvita ihmisälyä tekemään päätöksiä. Sen sijaan ne voidaan suorittaa ohjelmallisesti, koska ne toimivat samalla kaavalla suorituksesta toiseen. Tällaisten tehtävien tekeminen päivästä toiseen käy ihmiselle nopeasti puuduttavaksi ja turhauttavaksi varsinkin, mikäli kyseiset tehtävät ovat päivittäisiä tai jopa useasti päivässä tehtäviä.

Robotit myös nopeuttavat eri tehtävien suorittamista, koska ne ovat tarkempia kuin ihmiset. Ohjelmistorobotille määritetään tehtävä tarkasti. Onnistuneen tehtävän määrittelyn jälkeen ohjelmistorobotit voivat toimia väsymättä ja nopeasti ympäri vuorokauden. Tietoja syöttäessä robotin ei tarvitse etsiä tiettyä kenttää, vaan se löytää sen esimerkiksi tietyn HTML-tagin tai kuvatunnistuksen avulla välittömästi. Ohjelmistorobotti myöskin syöttää tiedot nopeampaa, kuin mihin ihminen pystyisi ja voi toimia myös koko vuorokauden ajan, näin nopeutetaan huomattavasti prosessien läpimenoaika (CGI, 2019).

Inhimillisten virheiden vähentäminen tietojen käsittelyssä, on myös tärkeää ja taloudellista. Ohjelmistorobotit syöttävät 100-prosenttisen tarkasti annetut tiedot määrättyyn lomakkeeseen tai sovelukseen. (CGI, 2019.) Tämä ei tosin poista alkuperäisen datan virheellisyyttä, koska ne eivät osaa arvioida tietojen oikeellisuutta, vaan robotit syöttävät tiedot sellaisenaan, kuin ne on annettu. Tästä syystä lähtödatan oikeellisuus on erittäin tärkeää ja tähän tarvitaan joko ihminen suorittamaan tarkistustyötä tai mahdollisesti jonkinlainen ”tekoäly” ennen ohjelmistorobotin suorittamaan tiedonsiirtoa.

## **2.2 Haasteet ohjelmistorobotiikassa**

Ohjelmistorobotiikka ei kuitenkaan ole täysin ongelmaton teknologia, vaikka hyvin lupaava alku tälle teknologianalalle onkin nähtävissä. Teknologiana ohjelmistorobotiikka on varsin uusi ja lisäksi se voi aiheuttaa häiriöitä työprosesseissa ja kulttuurissa (Torikka, 2019).

### **Johdon epärointi ja työyhteisön ennakoasenteet**

On luonnollista, että uusi teknologia herättää epäilystä ja kysymyksiä. Etenkin sellainen teknologia, jolla on mahdollisuus ja tavoite poistaa kokonaan ihmisen tekemiä työtehtäviä (Torikka, 2019). Oh-



jelmistorobottien kehittäminen on lopulta monimutkainen ja pitkä prosessi. Tämä voi aiheuttaa epäilystä teknologian käytettävyydestä, joskin hieman perusteettomasti. Kuitenkin ohjelmistorobotiikan tavoitteena on luoda säästöjä pitkällä aikavälillä, eikä niinkään välittömästi tai ensimmäisen kuu-kauden aikana.

Ohjelmistorobotiikan vaikutus henkilöstöön on myös yksi syy, joka aiheuttaa epäröintiä sen käyttöönotossa. Lisäksi itse teknologiaan ei vielä luoteta riittävästi, jotta prosesseja uskallettaisiin antaa sen haltuun. (Torikka, 2019.) Arvon (2019, 108) tutkimuksen mukaan viestinnän avulla voidaan sekä vähentää henkilöstön pelkoja esimerkiksi työpaikan menettämisestä että lisäksi onnistua uusien ohjelmistorobotiikkaideoiden keräämisessä. Myös Willcocksin & Laciin 2015 mukaan RPA ei aiheuta juurikaan työpaikkojen menetyttä, paitsi luonnollisen poistuman kautta eli ihmisten, joiden työpaikkoja ei täytetä heidän pois lähtiessään.

### **Koulutuksen puute tai sen vajavaisuus**

Torikka (2019) viittaa tekstissään Angelo Poulikakokseen, jonka mukaan ”organisaatiot luottavat valitettavan usein kouralliseen ihmisiä. Nämä ihmiset ovat yleensä itse investoineet aikaa ja resursseja koulutukseensa. RPA-koulutuksen pitää olla tärkeyslistan kärjessä kaikille, jotka haluat olla tässä mukana.”

Arvon (2019, 104) tutkimuksessa yhdeksi haasteeksi nousi se, että ohjelmistorobotiikkaprojektiin osallistuvassa ryhmässä mallintajat eivät tunteneet automatisoitavaa työkulkua ja vastaavasti työkulun asiantuntijat eivät tunteneet ohjelmistorobotiikkaa. Torikan (2019) mukaan RPA-projek-teissa epäonnistuminen on todennäköisempää, mikäli koulutukseen ei panosteta tarpeeksi tai henkilöstöllä ei ole mahdollisuutta käyttää RPA-projekteihin riittävästi aikaa. Koulutuksen yksi olennainen osa on tuoda kaikkien tietoisuuteen se, mitä ohjelmistorobotiikan avulla voidaan tehdä sekä millaisia toimintoja on järkevä automatisoida.

Yrityksen on hyvä kehittää myös omia koulutusohjelmia, joilla RPA-asiantuntijat voivat sekä parantaa omaa osaamistaan sekä jakaa tietotaitoaan yrityksen muille työntekijöille, jotka voisivat omissa tehtävissään hyödyntää RPA:n mahdollisuuksia. Kouluttamalla muuta organisaatiota hel-potetaan asiantuntijoiden työtä. Kun mahdolliset RPA-projektit tulevat työpöydälle valmisteluvai-heen jälkeen, he voivat keskittyä olennaiseen eli robotin kehittämiseen.

## Väärät käyttötapaukset

RPA-kokeilua valmisteltaessa on tärkeä pitää mielessä käyttökohteen monimutkaisuus. Liian monimutkaisen käyttökohteen valinta kokeiluun hidastaa projektin etenemistä (Torikka, 2019), sekä voi antaa väärän kuvan koko teknologiasta ja näin ollen heikentää mahdollisuuksia käyttää ohjelmistorobotiikkaa tulevissa paremmissa käyttökohteissa. Tärkeintä on siis löytää tasapaino yksinkertaisen ja monimutkaisen käyttökohteen välillä sekä valitsemaan järkevä käyttökohde, jonka automatisoimisesta yritys hyötyy tarpeeksi.

Monesti yritykset hätäilevät automaatiohankkeiden aloittamisessa, jolloin mahdollisuuksien arviointi ja valmistelu jää vähäiseksi. Ohjelmistorobotiikka ei välttämättä sovellu kaikkiin automaatioideoihin ja on äärimmäisen tärkeää pitää järki mukana, kun uusia käyttökohteita kartoitetaan. Automatisoitavien kohteiden kartoitus voi myös paljastaa merkittäviä kehityskohteita prosesseissa, jolloin kyseessä ei ole pelkästään automatisointi vaan yleinen toiminnan tehostaminen (Torikka, 2019).

### 2.3 Ohjelmistorobottialustat

Ohjelmistorobottien luomiseen on tarjolla useita eri alustoja, joiden eroavaisuudet ovat yleisesti katsoen melko pieniä. Enimmäkseen erot liittyvät käyttöliittymiin sekä joidenkin alustojen välillä siihen, miten ohjelmistorobotit käsittelevät muita järjestelmiä. Ohjelmoinnin osaaminen ei ole pakollista työpöytäautomaation luonnissa, mutta mitä monimutkaisemmaksi automaatio menee, sitä enemmän on tarvetta ohjelmoinnille. (Laitila, 2018.) Ohjelmistorobottien lisensointi on valtaosalla tarjoajista samanlaista. Myöhemmin vertailuista roboteista, kaksi (WorkFusion ja UiPath) tarjoavat ilmaisen yhteisöversion, mutta yrityskäyttöä varten ne tulee päivittää maksulliseen versioon. Kolmas (SikuliX) ohjelmistorobotti on saatavilla ilmaiseksi kaikkien käyttöön.

Laitilan (2018) artikkelissa käsitellään kolmea eri ohjelmistorobottia, joista WorkFusion ja UiPath tarjoavat graafisen käyttöliittymän. Niissä robotin luonti tapahtuu suurimmaksi osaksi valmiilla elementeillä, eli käytetään valmiita nappeja tai komentojonoja, jotka voi muokata omia tarpeita vastaavaksi. SikuliXissäkin pääsee alkuun ilman ohjelmointia, mutta monimutkaisemmissa automaatioissa python-ohjelmointikielen osaaminen on pakollista.

Suurin ero ohjelmistoroboteissa on niiden tapa käyttää muita järjestelmiä. WorkFusion sekä SikuliX hyödyntävät ns. pinta-automaatiota, eli ne tunnistavat kuvien perusteella halutut elementit ja käyttävät niitä määritettyihin toimintoihin. Pinta-automaation heikkoutena on sen herkkyys käyttöliittymämuutoksille, eli pienetkin käyttöliittymämuutokset voivat rikkoa tehdyn robotin. UiPath sen sijaan käyttää ohjelmien rakenneosia, kuten internetsivujen html-koodin elementtejä tai Windows-sovel-lusten käyttöliittymärakenteita. (Laitila, 2018.)

### 3 OHJELMISTOROBOTIIKAN SOVELLUSKOHTEET

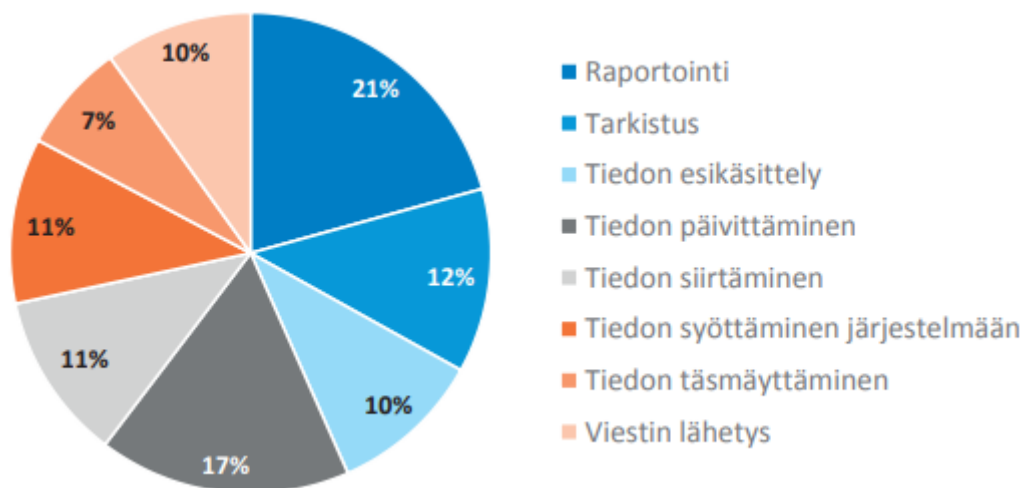
Ohjelmistorobotiikassa on potentiaalia nousta merkittäväksi teknologiaksi osaksi jokapäiväistä arkea, ja sen hyödyntämiseen löytyy jokin prosessi lähes jokaisesta organisaatiosta (Miettinen 2017, 46). Ohjelmistorobotiikkaa voidaan hyödyntää lähes missä tahansa selkeässä joka päiväisessä työssä. Yrityksissä on hyvä miettiä jo ohjelmistorobotiikan käyttöönoton alkuvaiheissa mitä muita toimintoja voidaan automatisoida. Riippuen käytetystä alustasta voidaan joitain robotin osia käyttää uudelleen muissa projekteissa. Tämä edellyttää sen, että ne on luotu tarpeeksi geneerisesti eikä niihin ole liitetty liian tarkkoja muuttujia tai muita yksilöiviä tekijöitä.

Kääriäisen, Aihkisalón, Halénin, Holmströmin, Jurmun, Matinmikon, Seppälán, Tihisen ja Tirrosen (2018, 10, 50) selvityksessä todetaan, että todennäköisesti ohjelmistorobotiikan soveltaminen tulee pääosin keskittymään organisaatioiden sisäisen tehokkuuden parantamiseen sekä jonkin verran myös organisaatioiden väliseen tiedonsiirtoon. Heidán mukaansa valtaosa käyttötapauksista kohdistuu tukiprosesseihin. Selvityksessä tarkasteltiin ohjelmistorobotiikan soveltamisen nykytilaa käyttötapausten kautta. Aineistona oli yli 878 käyttötapausta Suomesta, jotka on luokiteltu taulukossa 1.

Taulukko 1 Ohjelmistorobotiikan käyttötapauksen luokittelu suoritettavan tehtävän eli käyttötapaustyyppin mukaan (Kääriäinen ym., 2018, 10)

Työtehtävän kategoria	Selite
Raportointi	Erilaisten raporttien ja yhteenvetojen kokoaminen järjestelmistä
Tarkistus ja testaus	Tietojen oikeellisuuden tarkistaminen, järjestelmätestaukset
Tiedon esikäsittely	Kerätään, työstetään ja lajitellaan tietoa myöhempää käyttöä varten ihmiselle prosessin seuraavissa vaiheissa
Tiedon päivittäminen	Ylikirjoitetaan vanhaa tietoa uudella tai poistetaan vanhaa tietoa kokonaan. Tiedon laadun ylläpitäminen
Tiedon siirtäminen	Siirretään tai kopioidaan tietueita järjestelmästä toiseen, massatallennukset, arkistoinnit
Tiedon syöttäminen järjestelmään	Syötetään uusia tietueita järjestelmiin, esimerkiksi luodaan uusi asiakkuus tai työntekijä
Tiedon täsmäyttäminen	Verrataan kahden tai useamman tietolähteen tietoja keskenään
Viestin lähetys	Massapostitukset, sähköpostien lähetys, muistutukset, selvityspyynnöt

Kuviosta 1 näkee, että käyttötapaustyyppit voivat jakautua monenlaisiin tehtäviin. Selkeästi suurimpina tyyppinä ovat raportointi ja tiedon päivittäminen. Ne ovat tehtäviä, jotka vaativat paljon manuaalista triviaalia tiedon käsittelyä, johon ohjelmistorobotiikka on erinomainen työkalu.



Kuvio 1 Käyttötapaustyyppien jakautuminen (Kääriäinen ym., 2018, 10)

Miettinen (2017, 8) listaa Pro gradu -tutkimuksessaan joitakin esimerkkejä ohjelmistorobotiikan käyttökohteista. Muun muassa Birminghamin yliopistollisessa sairaalassa ohjelmistorobotiikan avulla siirretään läheteitä, testituloksia ja sairaskertomuksia kansallisen- ja sairaalan järjestelmän

välillä. Postilla ohjelmistorobotit auttavat henkilöstön palkkojen laskennassa hoitamalla öisin rutiinitöitä. Valtion talous- ja henkilöstöhallinnon palvelukeskus puolestaan on automatisoinut ostolas-  
kujen käsittelyyn ja palvelussuhdetietojen tarkastamiseen ja tallentamiseen liittyviä tehtäviä.

## 4 TOIMEKSIANTAJA JA TOIMEKSIANTO

Opinnäytetyön toimeksiantaja toimii MPY Yrityspalvelut Oyj, joka on vuonna 1888 perustettu tietotekniikkapalveluihin erikoistunut asiantuntijayritys. Yritys tarjoaa kuluttajille ja pienyrityksille kattavat laajakaista- sekä TV-palvelut Etelä-Savon alueella ja digitalkkaripalveluja koko Suomen alueella. Lisäksi yritysasiakkaille MPY tarjoaa tietoverkko- ja pilvipalveluja omassa konesalissa. Yritysasiakkaiden palveluihin kuuluu muun muassa tietotekniset asiantuntijapalvelut ja työn tekemisen ratkaisut.

Toimeksiannon tavoitteena on luoda ohjelmistorobotti, jonka tehtävänä on suorittaa konkurssivalvontaa, eli tarkistaa Suomen asiakastiedon tarjoaman konkurssi- ja yrityssaneerauslistan perusteella Lowellin verkkopalvelusta, onko listalla olevilla yrityksillä avoimia laskuja MPY:lle. Näistä kasataan raportti, joka lähetetään perinnästä vastaavalle henkilölle MPY:llä.

Ohjelmistorobotin tavoitteena on vähentää käsin tehtävää tarkastustyötä, ja näin nopeuttaa prosessin kulkua ja mahdollisia inhimillisiä virheitä tiedonsyötössä. Aikasäästöksi viikon ajalle on laskettu noin 1,5 tunnin säästö samalla vähentäen tästä tehtävästä vastaavan työntekijän manuaalista tiedonsyöttötyötä päivittäisistä työtehtävistä. Samalla vahvistetaan kehittäjän tuntemusta UiPath-työkalusta ja ohjelmistorobotiikasta tulevia työtehtäviä varten.

Toimeksiannon tehtävässä työkaluna toimii UiPath-ohjelmistorobottialusta, sekä sen tarjoamat kehitystyökalut ja siihen liittyvät lisäosat. Myöhemmässä kappaleessa käsitellään käytännön toteutuksessa käytettyjä työkaluja tarkemmin ja kerrotaan lisäosien tarkoituksesta enemmän.

Tehtävään ohjeistuksen on luonut työtehtävästä vastannut henkilö MPY:llä. Toimeksiannon edessä tehdään yhteistyötä kehittäjän ja vastaavan henkilön kanssa, jotta ohjelmistorobotista saadaan luotua sellainen kuin tarvitaan tehokkaasti ja mahdollisimman nopeasti.

## 5 TOIMEKSIANNON TOTEUTUS

### 5.1 Ohjelmistorobotin kehityksen lähtökohdat

Ohjelmistorobotin kehitys aloitettiin hahmottelemalla robotin runko prosessin omistajan laatiman toimintaohjeistuksen mukaan. Toimintaohjeistuksessa selvitetään yksityiskohtainen prosessinkuvaus sekä tietojen haun, loppuraportin luonnin ja lähetyksen ohjeistukset. Prosessin tulee edetä siten, että Suomen Asiakastieto lähettää jokaisena arkipäivänä robotin sähköpostiosoitteeseen raportin otsikolla Vahti, robotti avaa liitteenä olevan PDF-tiedoston. Vahti-raportin tietosisältö alkaa dokumentin toiselta sivulta Konkurssit -otsikon alta. Raportilla olevista yrityksistä robotti kopioi tiedon (y-tunnus, 8 numeroa), joka seuraa merkkejä "Y:".

Robotti avaa Firefox-selaimen ja Lowellin sivun ja kirjautuu määritetyillä käyttäjätunnuksilla. Avautuvalta Lowellin Access -palvelun etusivulta navigoidaan Laskutus -osioon. Mahdollisessa virhetilanteessa painetaan takaisinpainiketta ja klikataan uudelleen Laskutus-painiketta. Pdf:stä poimittu ensimmäinen kahdeksan merkin yritysasiakkaan y-tunnus syötetään kuvan Henkilö- / Y-tunnuskenttään, lisätään kahden viimeisen merkin väliin väliviivamerkki (-) ja painetaan oikealla puolella olevaa Siirry-painiketta. Mikäli kyseessä ei ole MPY:n asiakas, palvelu antaa ilmoituksen "Hakusi ei palauttanut yhtään osumaa" ja voidaan siirtyä hakemaan pdf:stä seuraava y-tunnus.

Tiedot kopioidaan sähköpostiin, joka on prosessin loppuraportti. Kopionnin jälkeen haetaan Vahti-raportilta seuraava y-tunnus, syötetään se edellisen tilalle, lisätään väliviiva ja klikataan Hae-painiketta. Kun konkurssit on käyty läpi, vastaava tarkastus tehdään Vahti-raportilla oleville yrityssaneerauksille. Kun kaikki Vahti-raportin y-tunnukset on käyty läpi Konkurssien ja Yrityssaneerausten osalta, robotti muodostaa loppuraporttina sähköpostin ja lähettää sen määrättyihin sähköpostiosoitteisiin. Kun loppuraportti on lähetetty, robotti kirjautuu ulos Lowell Access palvelusta ja sulkee Vahti-raportin (pdf).

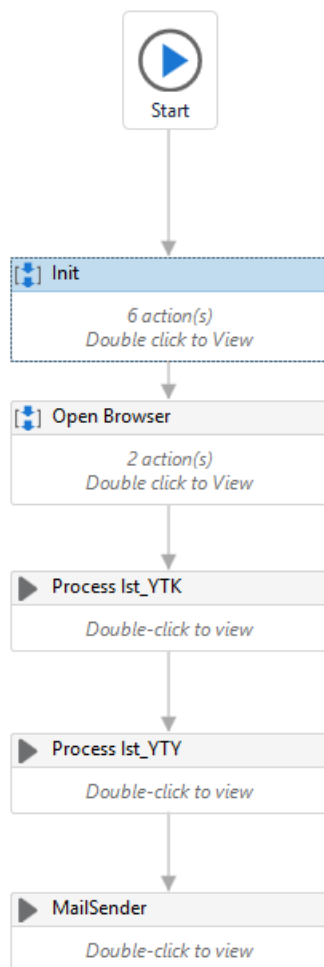


## 5.2 Ohjelmistorobotin kehittämisen kulku

En ollut aiemmin käyttänyt UiPathia, joten toimeksiannon aikana opettelin perusasioista lähtien sen käytön. Robotin kehitys tapahtui vaiheittain. Kehitystyön kulku oli pääosin robotin työnkulun mukainen, sillä se oli loogisin lähestymistapa tavoitteen saavuttamiseksi. Kehitykseen käytetystä ajasta valtaosa meni asiaan perehtymiseen ja kehitystekniikoiden selvittämiseen.

### Robotin työnkulku

Päätasolla robotti koostuu viidestä osasta. Näiden osien tehtävinä on joko valmistella data, prosessoida sitä tai koostaa käsitellystä datasta raportti jatkotoimia varten. Robotin työnkulussa jaetaan prosessi useampaan osaan, jotta myöhemmin testausvaiheessa on helpompi löytää mahdolliset virheet tai ongelmat robotin ajossa. Työnkulun jaottelu eri osa-alueisiin on yleisesti hyväksytty toimintamalli, sillä se helpottaa robottien luettavuutta ja ylläpitoa sekä mahdollistaa joidenkin robotin osien uudelleenkäyttämisen muissa projekteissa.

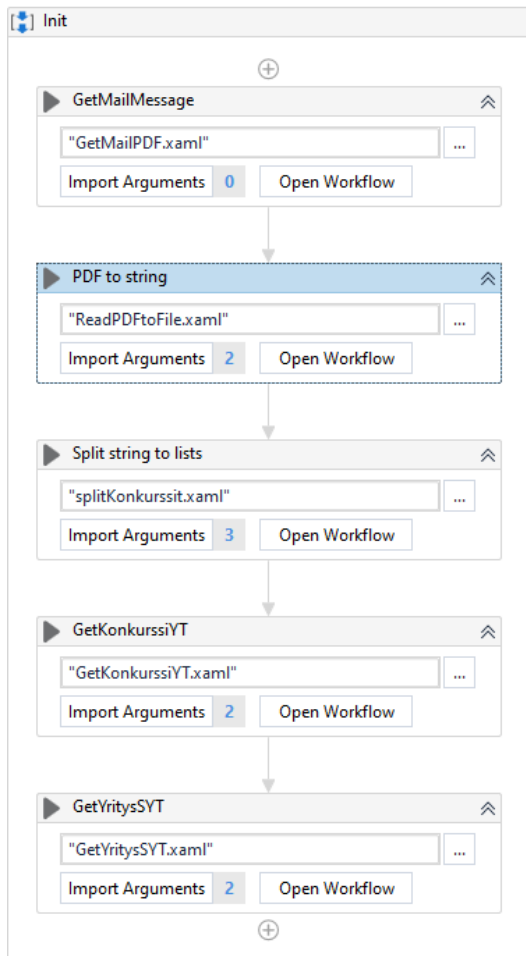


Kuva 1 Robotin työnkulun päätaso

Päätasolla muuttajat liittyvät enimmäkseen pysyvien tietojen keräämiseen, kuten käytettävät käyttäjätunnukset sekä salasanat, tiedostopolut tarvittavien tiedostojen tallentamiseen ja käsittelyyn sekä robotin tuottamien tietojen säilömiseen raportin luontia varten. Robotin päätasoo toimii sen toiminnan pohjana, jonka kautta suoritetaan alityönkulut sekä hallitaan robotin tarvitsemia ja käyttämiä tietoja/tietolähteitä.

### **5.2.1 Robotin alustus**

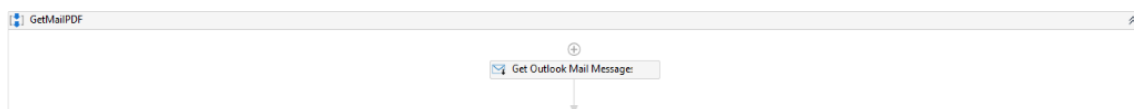
Init-osio sisältää tietojen hakuun ja valmisteluun liittyvät työnkulut. Osion nimi "Init" viittaa ohjelmiston ensimmäiseen toimintoon suorituksen alkaessa. Tällä yleensä tarkoitetaan tarvittavien tietojen hakua ja niiden käsittelyä myöhempiä toimintoja varten. Tässä projektissa Init-osion tehtävänä on hakea käytettävä lähtödata ja valmistella se prosessointia varten listoiksi. Nämä listat täytetään lähtödatasta poimituilla yritystunnuksilla ja ne on jaettu konkurssi- ja yrityssaneerauslistoiksi.



Kuva 2 Init-osio, tietojen haku, käsittely ja tallennus

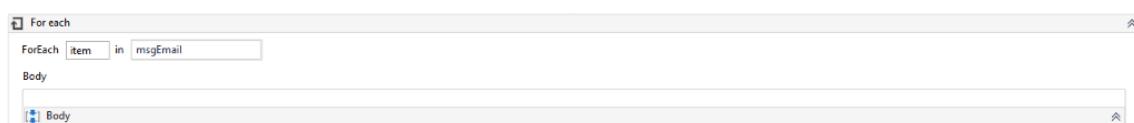
## Sähköpostin noutaminen Outlookista

GetMailPDF (kuva 3) työnkulku hakee Outlookista uusimman sähköpostiviestin, jonka otsikko sisältää sanan "Vahti." Sähköpostiviesti tallennetaan robotille määritettyyn muuttujaan (msgEmail),



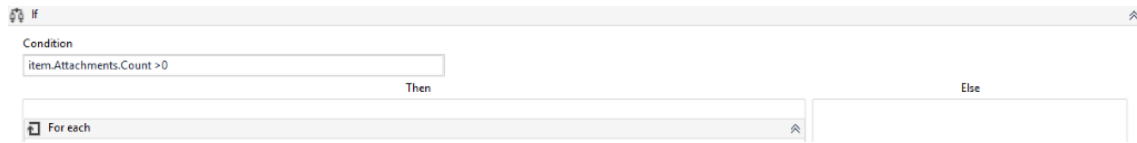
Kuva 3 Outlook viestin haku

Työnkulku käy läpi msgEmail muuttujan sisältämät viestit ja erottelee jokaisesta pdf päätteisen tiedoston (kuva 4). Nämä tiedostot tallennetaan robotin työkansioon käsittelyä varten.



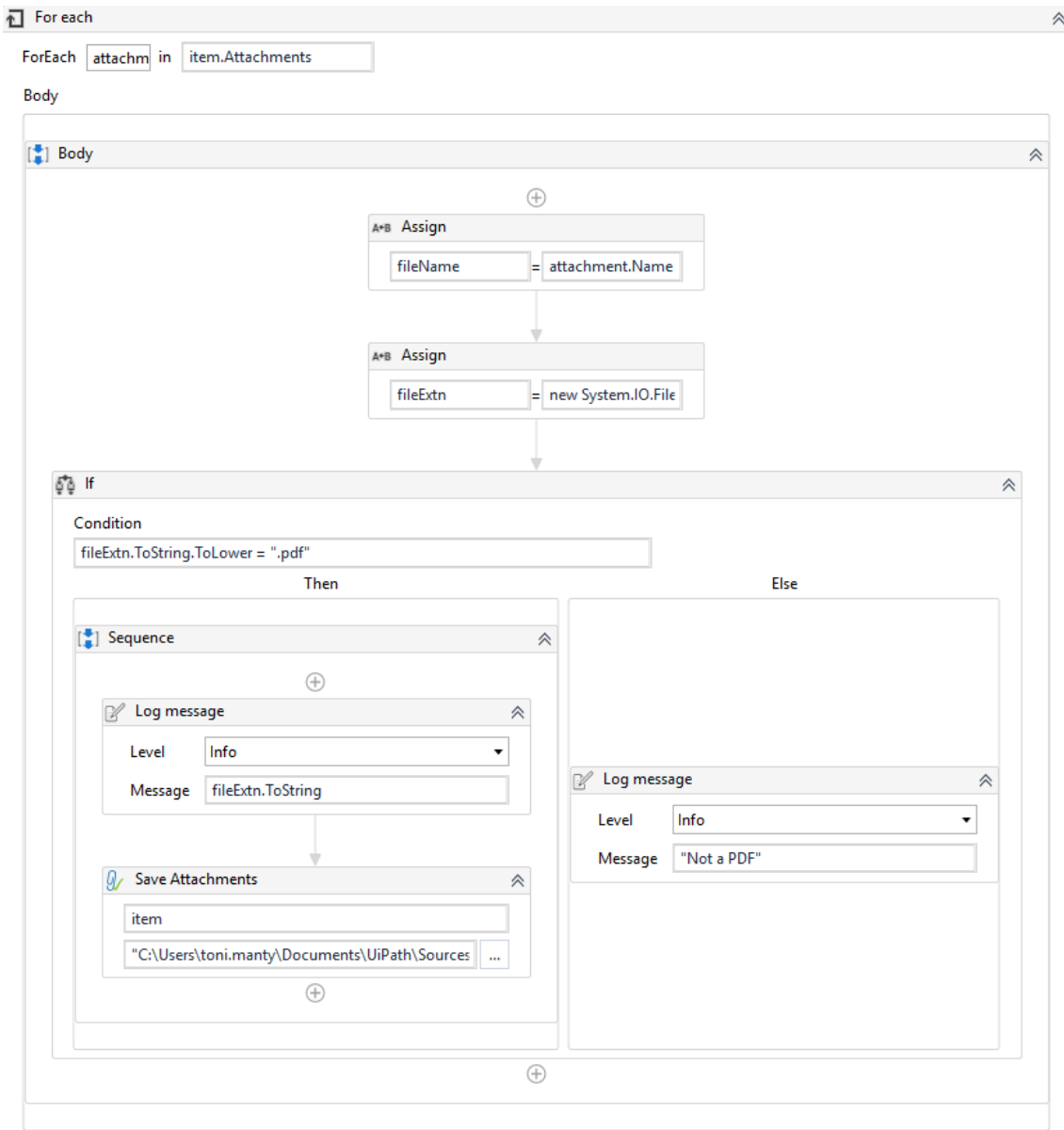
Kuva 4 Haettujen sähköpostien läpikäynti liitteiden perusteella

Edellä mainittu For each -käsky sisältää If-osion (kuva 5), joka tarkistaa liitteiden määrän tallennetuista sähköpostiviesteistä. Jos liitteitä löytyy vähintään yksi, siirrytään niiden tallentamiseen.



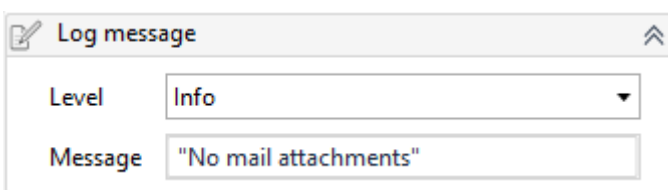
*Kuva 5 Löydettyjen liitteiden määrän tarkistus*

If-osion sisältämä For Each -toiminto (kuva 6) käy liitteet läpi tallentaen tiedoston nimen ja päätteän muuttujiin. Tämän jälkeen tarkistetaan, että tiedostopääte on ".pdf". Tiedoston päätteessä ".pdf", tiedosto tallennetaan robotille määriteltyyn työkansioon ja samalla kirjataan logiin tieto onnistuneesta tallennuksesta. Jos tiedoston pääte ei ole ".pdf", tiedosto hylätään ja tästäkin kirjataan tieto logiin.



Kuva 6 Löydettyjen liitteiden tallennus työkansioon

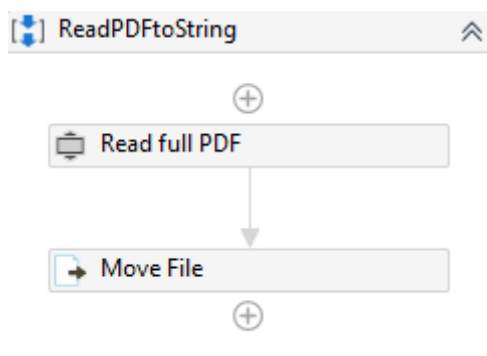
Kuvassa 5 esitettyyn If-osion Else käskyyn on määritetty Log Message -käsky (kuva 7), joka kirjaa logiin tiedon siitä, mikäli sähköpostiviestissä ei ole liitteitä. Samanlaista Log Message käskyä on käytetty edellä mainitussa For Each -toiminnossa, kirjaamaan tieto väärästä tiedostopäätteestä (kuva 6).



Kuva 7 Logiviestin tallennus liitteiden puuttuessa

## PDF-tiedoston luku muuttujaan

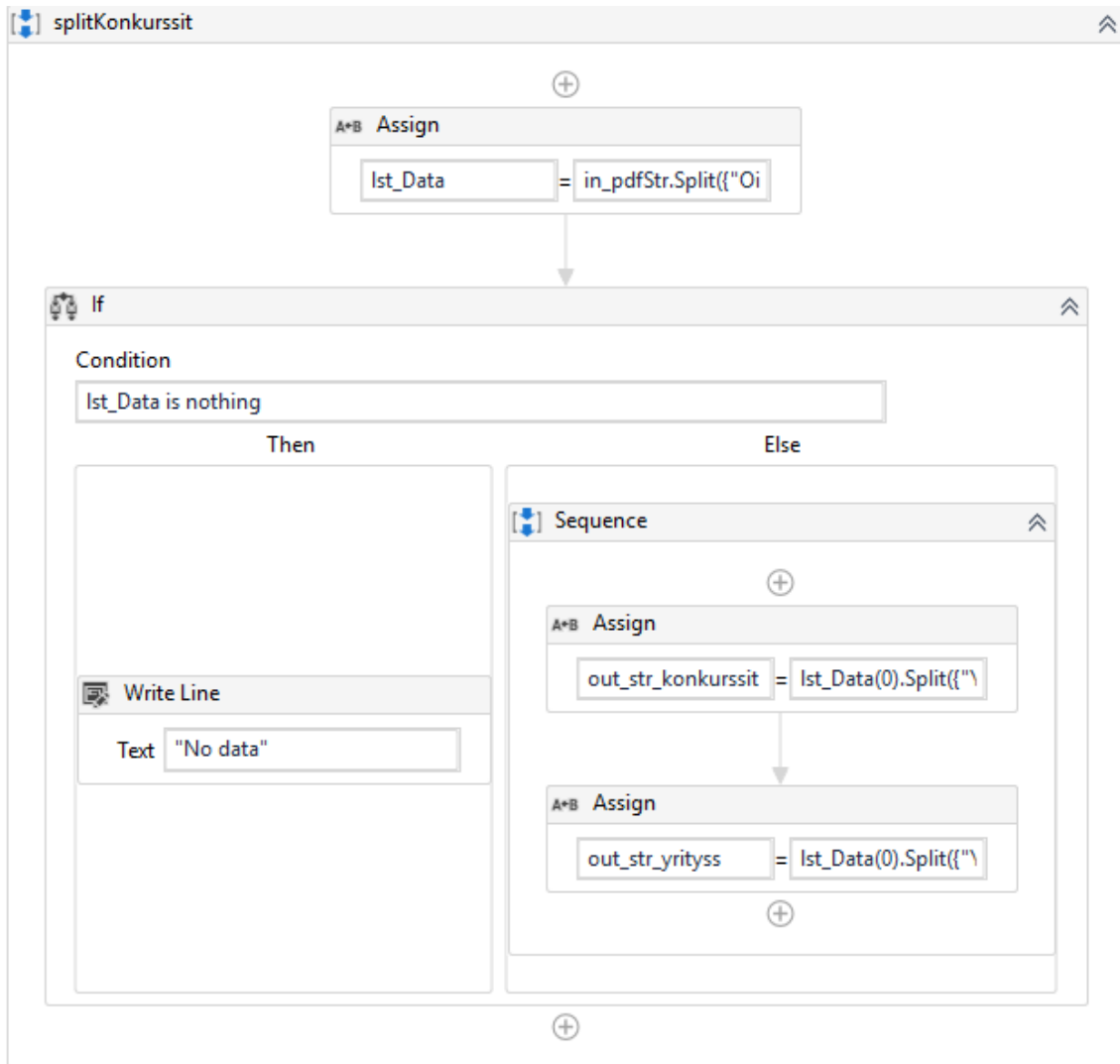
PDF to String työkulku (kuva 8) hakee tallennetun PDF-tiedoston, joka sisältää Suomen Asiakas-tiedon tarjoaman raportin yrityksistä ja niiden y-tunnuksista. Robotti poimii tekstistä tarvittavan tie-don ja tallentaa sen muuttujaan. Tämän jälkeen käsitelty PDF-tiedosto siirretään Valmiit-kansioon ja tiedoston nimeen lisätään päivämäärä, milloin se on käsitelty. Eroteltu teksti siirretään robotin päätasolle "out\_str\_pdfText" nimisen argumentin kautta. UiPathissa voidaan käyttää sekä muuttu-jia että argumentteja. Näiden erona on ainoastaan se, että argumentit toimivat eri työkulkujen välillä ja muuttujat ovat paikallisia. Eli argumenteilla voidaan siirtää muuttujien sisältämää tietoa työkulusta toiseen.



Kuva 8 Läpikäydyn PDF-tiedoston siirto Valmiit-kansioon

## Lähtödatan jako osiin

SplitKonkurssit-työkulku (kuva 9) jakaa aiemmin poimitun tekstin kahteen osaan sekä leikkaa pois ylimääräisen tekstin. Käsiteltävä teksti jaetaan kolmeen osaan, joista kaksi ensimmäistä eli konkurssit ja yrityssaneeraukset tallennetaan omiin muuttujiinsa. Ne siirretään robotin päätasolle jatkokäsittelyä varten. Kolmas osa muodostuu ylimääräisestä tekstistä, joka jätetään huomiotta.

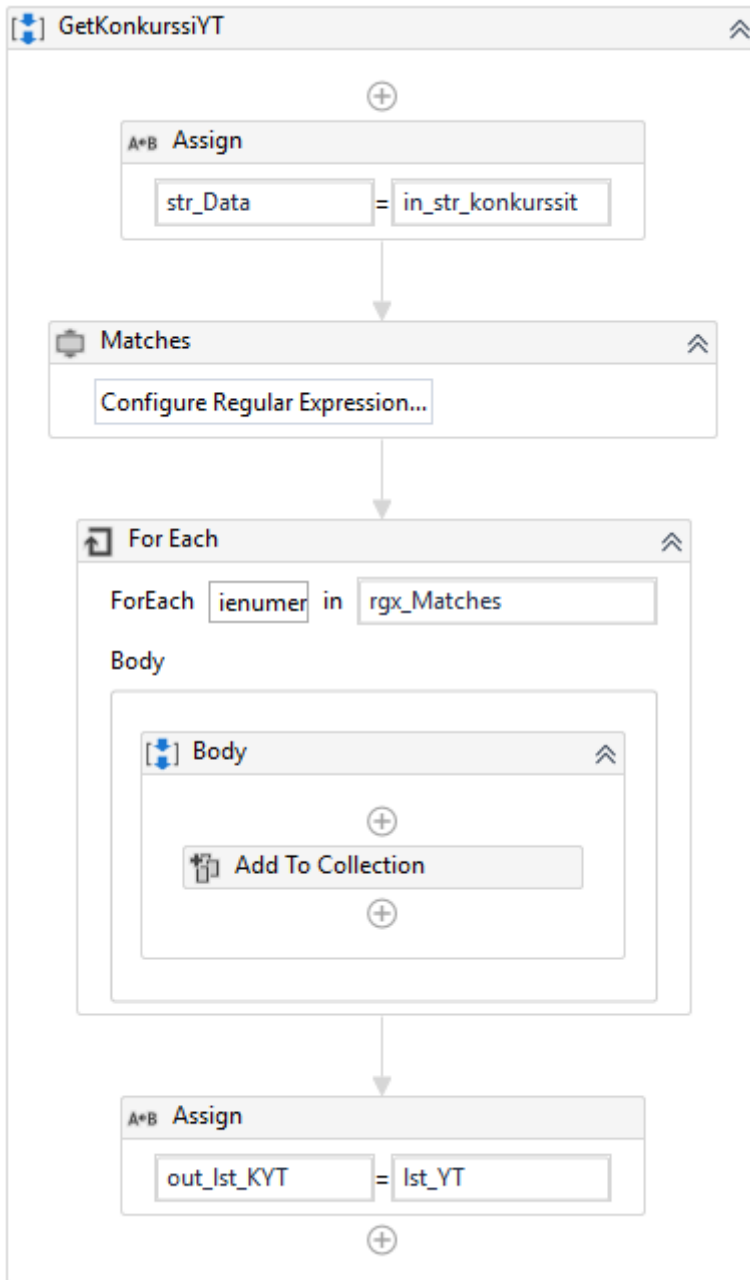


Kuva 9 Lähtödatan jako konkurssihin ja yrityssaneerauksiin

str\_Data on käsiteltävä PDF:stä saatu teksti. Ist\_Data on väliaikainen muuttuja, johon tallennetaan pilkottu teksti. If-toiminnoissa tarkastetaan sisältääkö Ist\_Data tietoa. Tiedon löytyessä Else-osiossa halutut osat tiedosta tallennetaan argumentteihin, joista se siirretään määritettyyn paikkaan.

### Konkurssien erottelu lähtödatasta

GetKonkurssiYT työnkulku (kuva 10) hakee konkurssit sisältävästä tekstistä kaikki yritystunnukset. Lisäksi se lisää viivan ennen viimeistä merkkiä, koska Vahti-raportti ei sisällä y-tunnuksessa olevaa viivaa. Tähän käytetään Regular Expression -lausetta, joka ajetaan For Each -toimintoon, jolla osumat tallennetaan listaan. Regular Expression -lause toimii sääntönä y-tunnuksen haussa sekä viivan lisäyksessä. For Each suorittaa edellä mainitun säännön mukaisen haun määritettyyn tekstiin, kunnes säännön mukaista osumaa ei enää löydy.

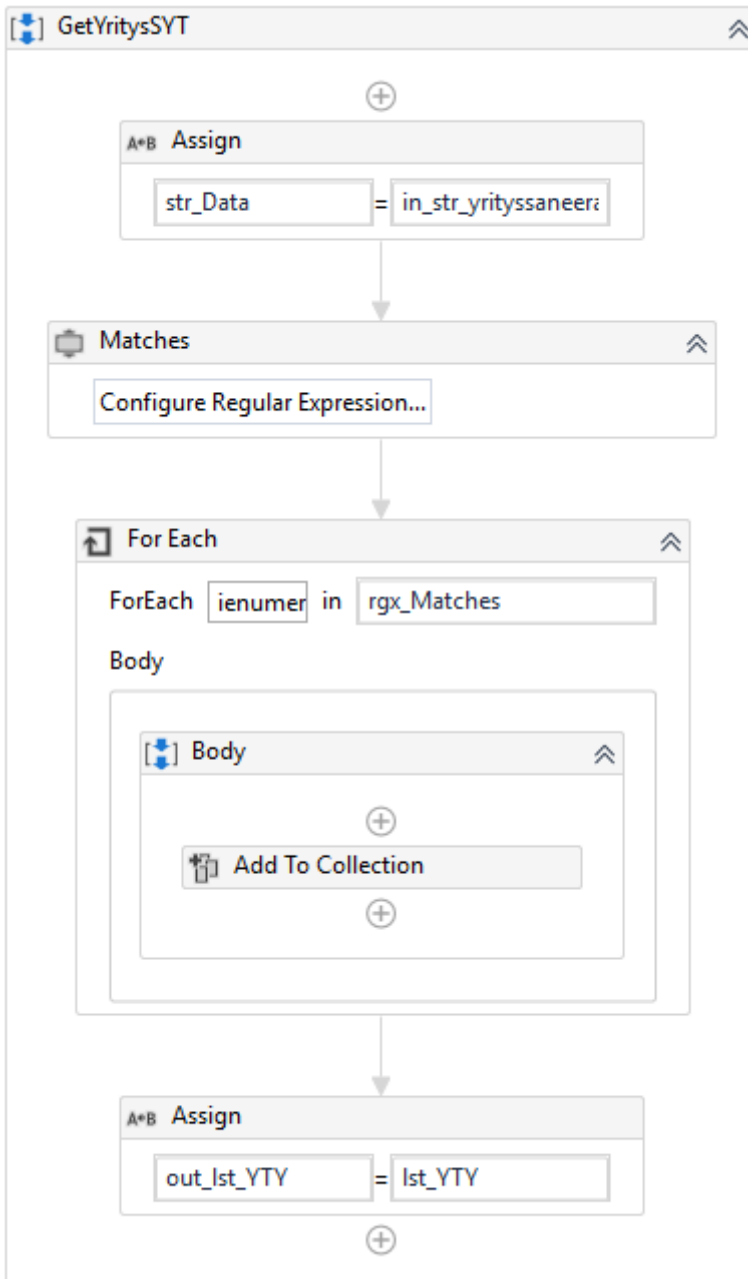


Kuva 10 Konkurssien y-tunnusten poiminta tekstistä

### Yrityssaneerausten erottelu lähtödatasta

GetYritysSYT työnkulku (kuva 11) hakee GetKonkurssiYT:n tavoin yrityssaneeraukset sisältävästä tekstistä kaikki yritystunnukset. Lisäksi se lisää viivan ennen viimeistä merkkiä, koska Vahti-raportti ei sisällä y-tunnuksessa olevaa viivaa.



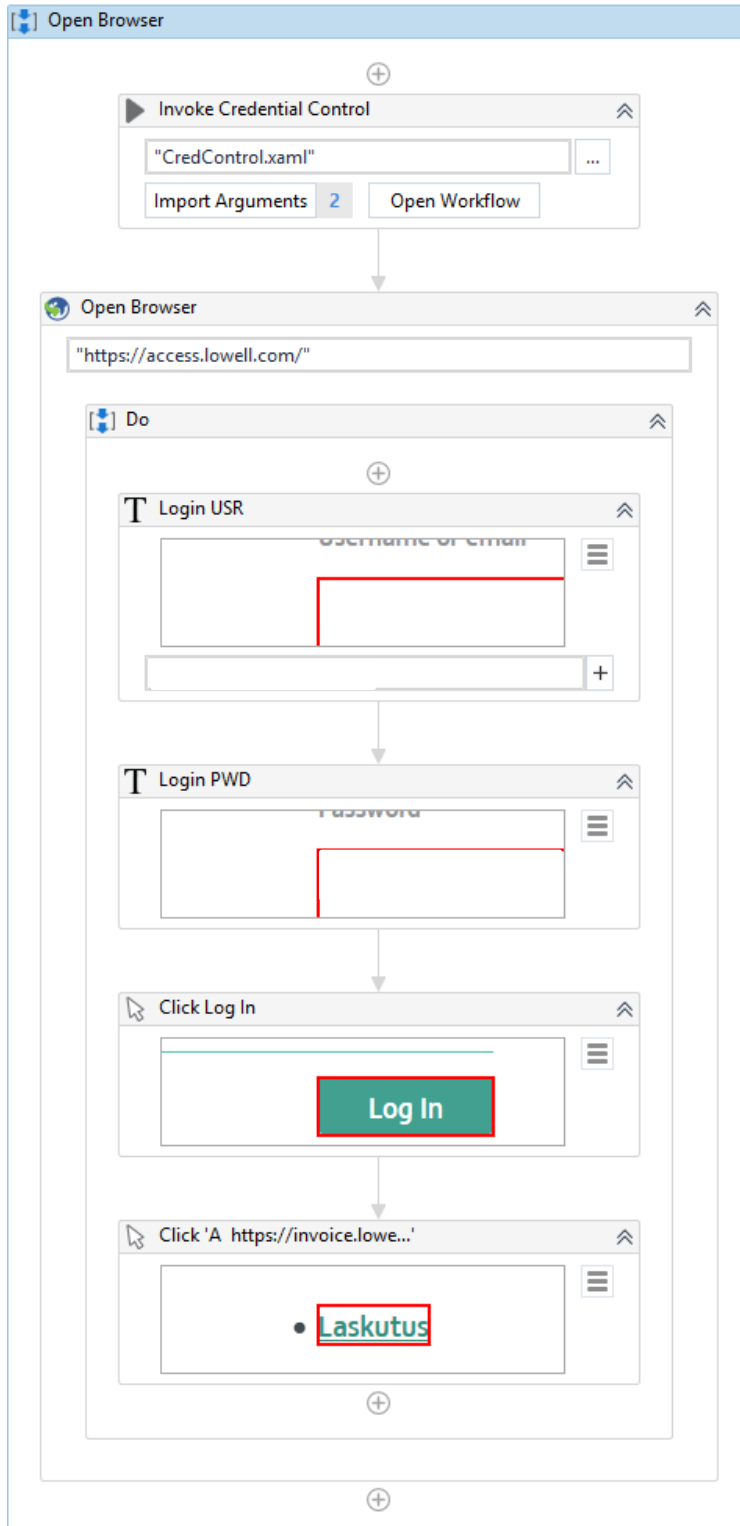


Kuva 11 Yrityssaneerausten y-tunnusten poiminta tekstistä

### 5.2.2 Selaimen avaaminen ja kirjautuminen

Open browser -työnkulku (kuva 12) sisältää sekä kirjautumistunnusten hallinnan että selaimen valmistelun. Ensimmäiseksi ajetaan Credential Control -työnkulku (Kuva 13), jossa tarkistetaan, onko

tarvittavat kirjautumistunnukset saatavilla Windowsin Credential Managerista. Jos kirjautumistun-  
nuksia ei löydy, robotti pyytää syöttämään tunnukset. Tämän jälkeen avataan Firefox-selain ja siir-  
rytään Lowellin Access-palvelun sivulle, jonka jälkeen robotti kirjautuu sisään ja siirtyy verkkosivun  
Laskutus osioon.

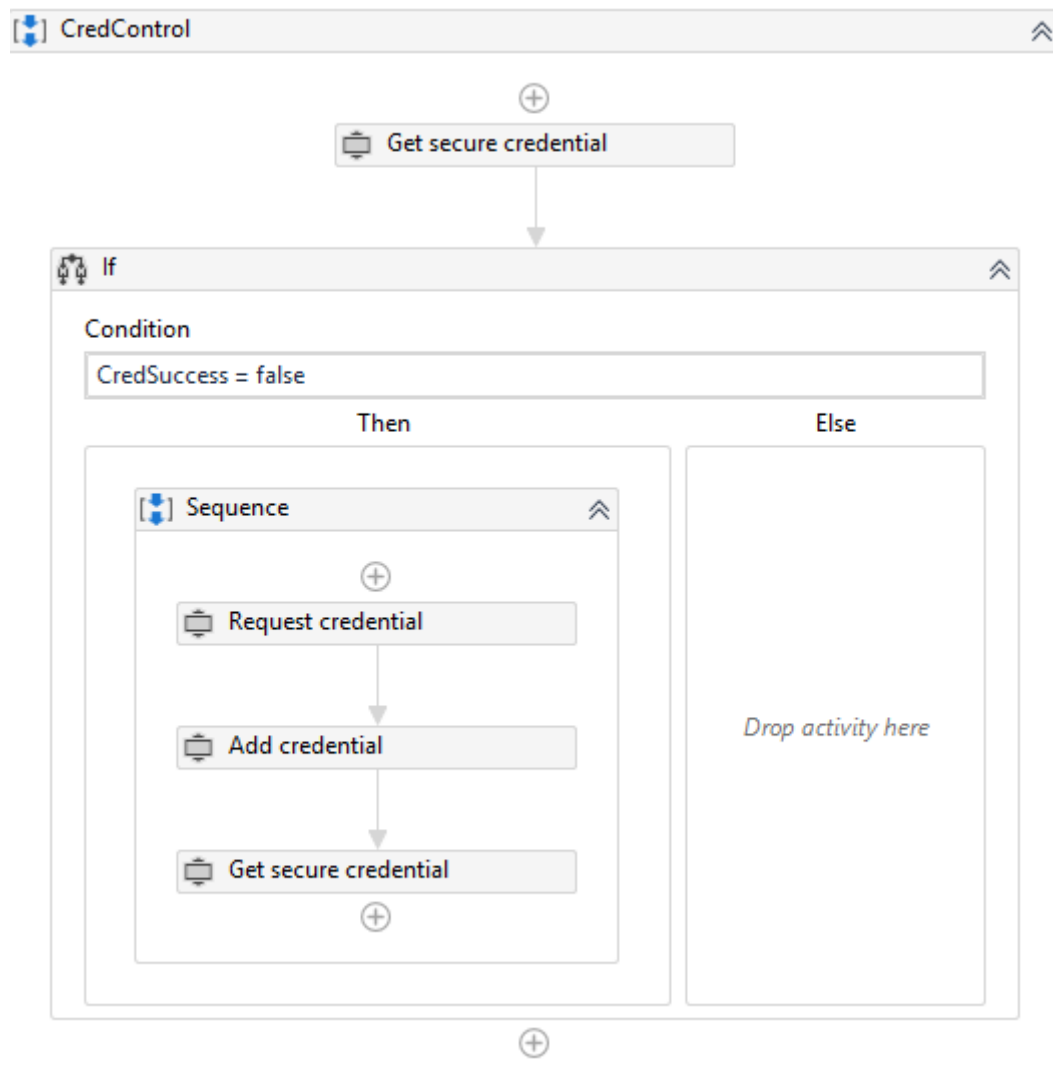


Kuva 12 Kirjautumistunnusten ja selaimen valmistelu

## Tunnusten hallinta

CredControl työnkulku (kuva 13) hoitaa tunnusten käsittelyn robotin sisällä. Tunnukset tallennetaan Windowsin Credential Manageriin, josta ne ovat myöhemmin saatavilla eri järjestelmien käyttöä varten.

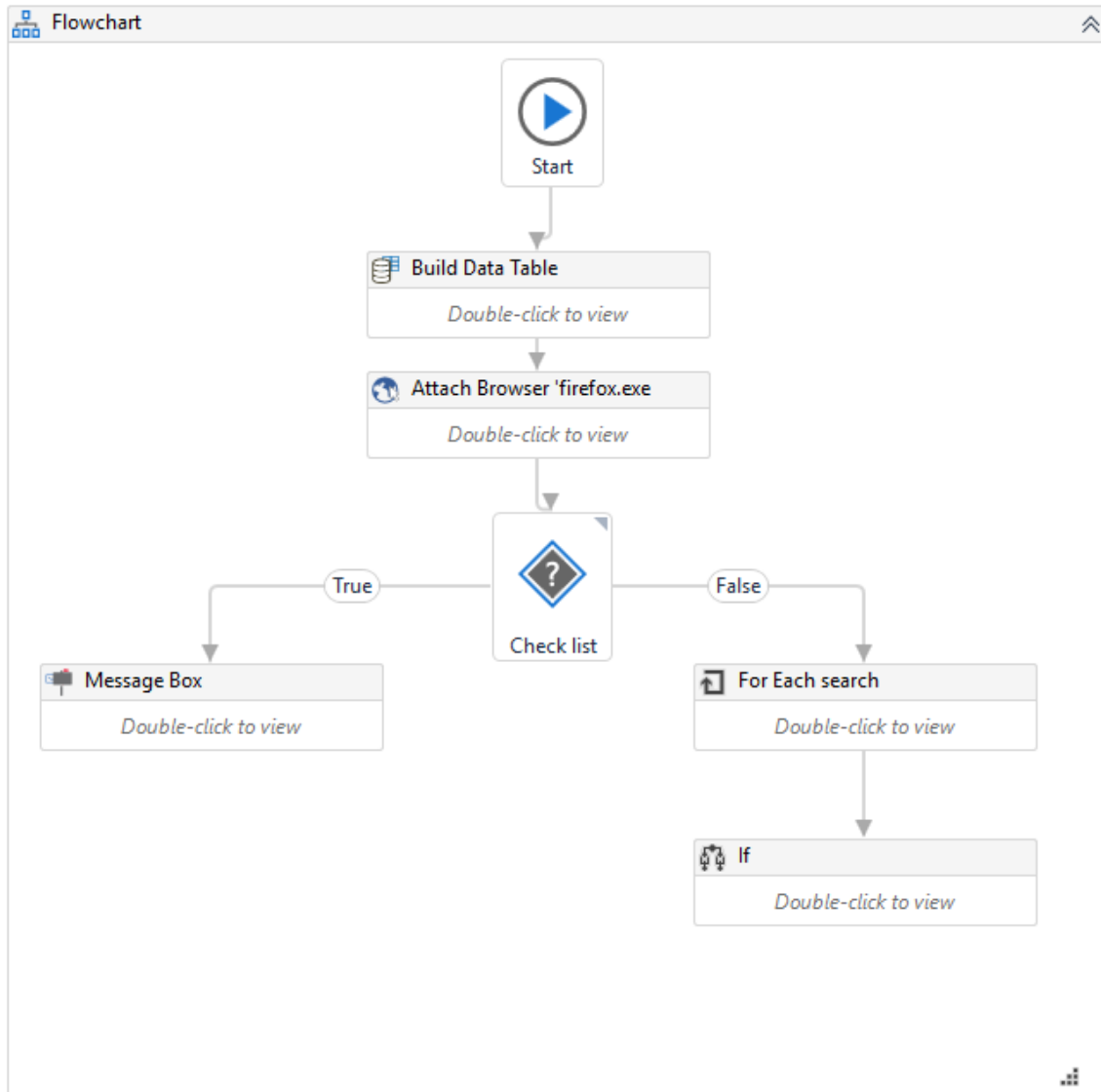
Jos tunnuksia ei löydy, pyydetään käyttäjää syöttämään tunnukset.



Kuva 13 Tunnusten haku ja saatavuuden tarkistus

### 5.2.3 Tietojen varsinainen käsittely

Listan käsittelijänä toimii ListProcessor työnkulku (kuva 14) hoitaa tietojen haun ja tulosten tallennuksen Lowellin järjestelmästä. Tietojen tarkistuksessa käytetään hyödyksi aiemmin luotuja listoja, jotka on täytetty Vahti-raportista poimituilla yritystunnuksilla.



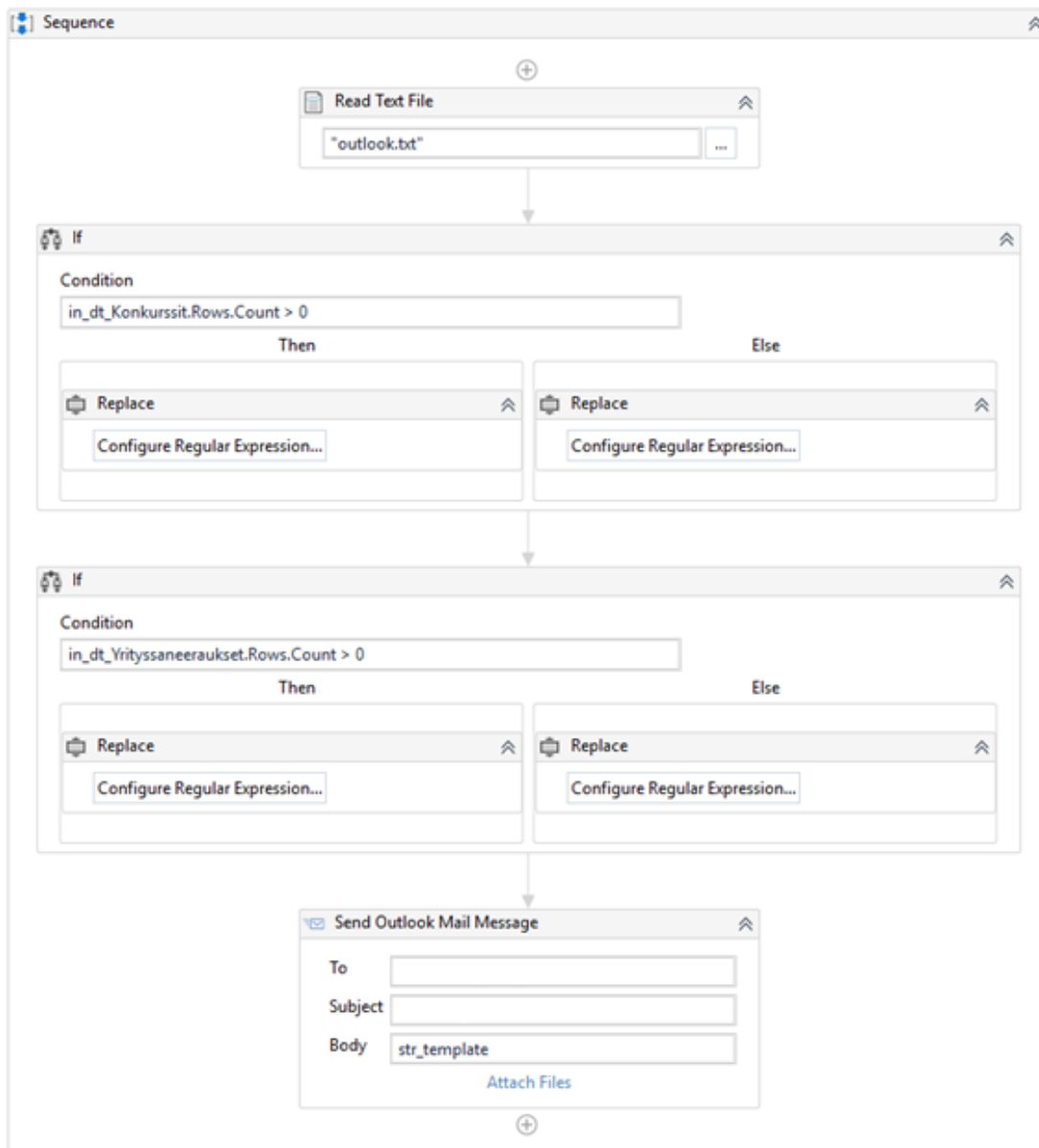
Kuva 14 Y-tunnusten tarkistusprosessi Lowell Access -palvelusta

Työnkulussa luodaan väliaikainen taulu, johon talletetaan hakutulokset Lowell Access -palvelusta. Tämän jälkeen yhdistetään avoimeen selaimen välilehteen, jotta robotilla on varmasti yhteys palvelun käyttöliittymään. Työnkulun seuraava osio tarkistaa käytettävän yritystunnuslistan sisältämät rivit. Jos lista on tyhjä, ilmoitetaan siitä ja siirrytään pois työnkulusta. Listan sisältäessä yritystunnuksia, aloitetaan niiden tarkistaminen palvelun käyttöliittymästä Lowellin tietokantaa vastaan.

Lopuksi työnkulku tarkistaa, onko yritystunnuslistan sisältämillä arvoilla löytynyt hakutuloksia. Mikäli näitä on, tallennetaan tulokset raportin luontia varten muuttuun ja siirrytään joko seuraavaan listaan tai raportin luontiin ja lähettämiseen.

#### 5.2.4 Raportin luonti ja lähettäminen

MailSender työnkulku (kuva 15) rakentaa raportin kerätyistä tiedoista ja lähettää sen sähköpostipohjaa käyttäen määrätyle henkilölle. Sähköpostin pohja on luotu tekstitiedostona robotin työkansoon, josta se luetaan tekstimuuttuun. Työnkulussa tarkistetaan hakutulokset listoittain (konkurssit, yrityssaneeraukset), joiden mukaan muokataan sähköpostipohjaa kertomaan tulosten lukumäärä. Sähköpostiin liitetään Excel-tiedosto, joka sisältää hakutuloksista otetut tiedot.



Kuva 15 Sähköpostin koonti ja lähetys

## 6 TESTAUS JA PROJEKTIN LOPPUTULOS

Robotin testauksessa on käytetty UiPathin työkaluista löytyvää Debug-työkalua. Sen etuna on reaaliaikainen listaus robotin toiminnoista ajon aikana, josta näkee helposti ongelmien sattuessa, missä kohtaa robotti kaatuu. Mielestäni debug-työkalun virheilmoitusten selkeydessä olisi kehitettävää. Varsinkin jotkut ongelmat on ilmoitettu hyvin geneerisesti, jolloin ne eivät anna tarpeeksi tietoa ongelman selvittämiseksi.

Suurin osa robotin testauksesta on tapahtunut vaiheittain ja niiden kehityksessä eri osien testauksessa. Debug-työkalulla robottia ajaessa voidaan määrittää, mitkä osat robotista ajetaan. Tällöin testausvaiheessa voidaan testata vain uusia toimintoja sekä niiden toimintaa olemassa olevan robotin kanssa. Kun halutaan testata pieniä muutoksia useasti, robotin ajoa pilkkomalla vähennetään testaukseen menevää aikaa.

Valmiiksi todetun robotin testaus suoritettiin puolittain tuotannossa pitämällä sekä manuaalinen tietojen tarkastus että robotin suorittama tarkastus käynnissä yhtä aikaa ja vertaamalla näiden tuloksia keskenään. Tässä vaiheessa tuli robotin ajossa esille muutamia isompia ongelmia, jotka jouduttiin korjaamaan ennen robotin tuotantoon ajoa. Robotti haki tuntemattoman syyn takia aina edellisen päivän Vahti-raportin sähköpostista, ja näin ollen se oli aina päivän myöhässä uusimmasta raportista. Ongelman syyksi selvisi robotin tapa käyttää Outlook ohjelmaa, jolloin Outlook ei ehtinyt hakea uusimpia sähköposteja palvelimelta. Ongelman ratkaistiin vaihtamalla sähköpostien hakuun käytetty Outlook-prosessi geneeriseen sähköpostinhaku-prosessiin, joka hakee sähköpostit suoraan palvelimelta käyttämättä Outlook-sovellusta. Outlookissa tätä varten luotiin uusi kansio, johon uudet raportit siirretään automaattisesti niiden saapuessa sähköpostipalvelimelle.

Toinen robotin ajossa sattunut ongelma oli ajoympäristöstä puuttuneet robotille määritettävät tunnukset, mikä aiheutti robotin ajon keskeytymisen puuttuvien oikeuksien vuoksi. Korjaus tälle oli yksinkertaisesti tunnusten lisääminen robottia varten, jonka jälkeen robotin ajo suoriutui loppuun asti ongelmitta.

Kun aiemmin sovittu testausaika saatiin päätökseen, siirrettiin robotti tuotantoon. Robotti ajetaan päivittäin Vahti-raportin saavuttua. Robotti on toiminut raportin kirjoitushetkellä useamman kuukauden lähes ongelmitta. Satunnaisina päivinä robotti lähettää kahden päivän raportit kerralla. Tähän ei ole löydetty syytä, mutta se on toistaiseksi jätetty odottamaan selvittelyä.

## 7 POHDINTA

Opinnäytetyön tarkoituksena oli kehittää ohjelmistorobotti hoitamaan konkurssivalvontaa yrityksen sisäisesti. Ohjelmistorobotin kehittäminen aloitettiin ilman mitään ennakkotietoja tai -osaamista, joka pitkitti projektin valmistumista huomattavasti. Projektin aikana suurin osa kehitysajasta kului kehitystyökalujen ja UiPathin syntaksin opettelussa sekä niiden soveltamisen opettelussa.

Teoriassa yksinkertainen prosessi huomattiin haasteellisemmaksi kuin alun perin ajateltiin. Prosessin pilkkominen pieniin osiin näytti käytännössä, kuinka paljon työtä yksinkertainenkin prosessi vaatii, jotta se voidaan suorittaa ohjelmistorobotilla. Prosessin pilkkominen tarkoittaa toimintaohjeistuksen laatimista, jossa prosessi kuvataan tarkasti klikkaus klikkaukselta. Tämä varmistaa sen, että ohjelmistorobotti toimii juuri siten kuin on tarkoitettu. Prosessin pilkkomiseen on käytettävä tarpeeksi aikaa, jotta ohjelmistorobotin kehittäminen sujuu ilman ylimääräisiä viivästyksiä. Tulevissa projekteissa tähän on kiinnitettävä erityistä huomiota.

Aikataulun venymisestä huolimatta ohjelmistorobotti saatiin valmiiksi ja päivittäiseen toimintaan. Tulevien projektien tekemistä helpottaa tämän projektin aikana karttunut perustieto UiPathista, sen työkaluista ja niiden soveltamisesta omiin tarkoituksiin. Projektin aikana isompien ongelmien kanssa auttoi ulkoinen asiantuntija, joka nopeutti vaikeiden ongelmien ratkaisussa ja näin nopeutti huomattavasti jo valmiiksi hidasta kehitystyötä. Jonkin asteinen peruskoulutus aiheeseen olisi ollut hyödyllinen ennen projektin aloitusta, mutta projektin yllättävä aloitus ei juurikaan jättänyt aikaa opiskella aihetta etukäteen.

Työkaluna UiPath on aloittelijaystävällinen, johtuen sen tavasta käyttää visuaalista editoria sekä koodia yhdessä. Tällä tavalla robotin rungon luominen on helpohko osuus, kunhan oppii hahmotamaan, miten robotti toimii käytännössä. Tämän jälkeen voidaan robotin toimintaa tarkentaa ja ohjata yksityiskohtaisemmaksi, kun runko on jo luotu valmiiksi. Jälkimmäinen osuus loi omaan kehitystyöhön suurimmat haasteet, mutta aktiivisen kehittäjäyhteisön kautta monesti löytyi samoja ongelmia, joita muut olivat jo ratkaisseet. Sieltä saadut vinkit auttoivat usein oman projektin etene- mistä.

## LÄHTEET

Arvo, R. 2019. Ohjelmistorobotiikan käyttö diagnostisten ohjausjärjestelmien automatisoinnissa, Case Verohallinto. Pro gradu –tutkielma. Turun yliopisto. Viitattu 26.10.2019, [https://www.utu-pub.fi/bitstream/handle/10024/148070/Arvo\\_Rebekka\\_Gradu.pdf?sequence=1](https://www.utu-pub.fi/bitstream/handle/10024/148070/Arvo_Rebekka_Gradu.pdf?sequence=1).

CGI 2019. RPA:n hyödyt. Viitattu 18.9.2019, <https://www.cgi.fi/fi/alykas-automaatio/ohjelmistorobotiikka>.

Hyytinen E. 2018. Ohjelmistorobotiikka vapauttaa ihmisen tekemään tehtäviä, joissa hän on yliver-  
tainen. Viitattu 26.10.2019, [https://www.codemen.fi/ohjelmistorobotiikka-vapauttaa-ihmisen-teke-  
maan-tehtavia-joissa-han-on-ylivertainen/](https://www.codemen.fi/ohjelmistorobotiikka-vapauttaa-ihmisen-teke-<br/>maan-tehtavia-joissa-han-on-ylivertainen/)  
[https://www.tivi.fi/uutiset/ota-ohjelmistorobotti-toihin-testasimme-3-vaihtoehtoa/1ca0f3a6-b979-  
39c4-8bd7-0e37f7b3b82b](https://www.tivi.fi/uutiset/ota-ohjelmistorobotti-toihin-testasimme-3-vaihtoehtoa/1ca0f3a6-b979-<br/>39c4-8bd7-0e37f7b3b82b)

Kääriäinen, J. (toim.), Aihkisalo, T., Halén, M., Holmström, H., Jurmu, P., Matinmikko, P., Seppälä  
T., Tihinen M., Tirronen J. 2018. Ohjelmistorobotiikka ja tekoäly – soveltamisen askelmerkkejä.  
Valtioneuvoston selvitys- ja tutkimustoiminta. Selvitys- ja tutkimustoiminnan sarja 65/2018.  
[http://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/161123/65-2018-Ohjelmistorobot-  
iikka%20ja%20tekoaly.pdf](http://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/161123/65-2018-Ohjelmistorobot-<br/>iikka%20ja%20tekoaly.pdf)

Laitila, T. 2018. Ota Ohjelmistorobotti töihin – testasimme 3 vaihtoehtoa. Viitattu 13.11.2019,  
[https://www.tivi.fi/uutiset/ota-ohjelmistorobotti-toihin-testasimme-3-vaihtoehtoa/1ca0f3a6-b979-  
39c4-8bd7-0e37f7b3b82b](https://www.tivi.fi/uutiset/ota-ohjelmistorobotti-toihin-testasimme-3-vaihtoehtoa/1ca0f3a6-b979-<br/>39c4-8bd7-0e37f7b3b82b)

Miettinen, M. 2017. Ohjelmistorobotiikka. Pro gradu-tutkielma. Itä-Suomen yliopisto. Viitattu  
27.10.2019, [http://epublications.uef.fi/pub/urn\\_nbn\\_fi\\_uef-20170986/urn\\_nbn\\_fi\\_uef-  
20170986.pdf](http://epublications.uef.fi/pub/urn_nbn_fi_uef-20170986/urn_nbn_fi_uef-<br/>20170986.pdf).

Torikka M. 2019. 5 syytä miksi ohjelmistorobotiikka epäonnistuu. Viitattu 26.10.2019,  
[https://www.tivi.fi/uutiset/5-syyta-miksi-ohjelmistorobotiikka-epaonnistuu/6f2e3e50-bafd-3e00-  
9f56-daae207c2439](https://www.tivi.fi/uutiset/5-syyta-miksi-ohjelmistorobotiikka-epaonnistuu/6f2e3e50-bafd-3e00-<br/>9f56-daae207c2439).



UiPath. 2019. What is Robotic Process Automation? Viitattu 18.9.2019, <https://www.uipath.com/rpa/robotic-process-automation>.

Willcocks, L. ja Lacity M. 2015. Businesses will increasingly use robots to deal with the explosion of data Viitattu 27.10.2019, <https://blogs.lse.ac.uk/businessreview/2015/09/15/businesses-will-increasingly-use-robots-to-deal-with-the-explosion-of-data/>