

Teemu Vaittinen

Ohjelmistorobotiikan hyödyntäminen julkishallinnon virastossa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Ohjelmistotekniikka

Insinööriytyö

28.3.2018

Tekijä Otsikko Sivumäärä Aika	Teemu Vaittinen Ohjelmistorobotiikan hyödyntäminen julkishallinnon virastossa 50 sivua + 6 liitettä 28.3.2018
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	tieto- ja viestintäteknikka
Ammatillinen pääaine	ohjelmistotuotanto
Ohjaajat	tietohallintopäällikkö Sakari Karstu yliopettaja Auvo Häkkinen
<p>Insinöörityön tavoitteena oli selvittää ohjelmistorobotiikan tuomia mahdollisuuksia julkishallinnon virastolle ja tarkentaa Robot Frameworkin sopivuutta ohjelmistorobotiikan kehitystyökäkaluna. Asiakkaalle, eli Patentti- ja rekisterihallitukselle (PRH) tärkeintä oli nopeuttaa itseään toistavia työprosesseja ohjelmistorobotiikan avulla ja saada sitä kautta pienennettyä kustannuksia.</p> <p>Insinöörityössä sujuvoitettiin PRH:n sisäistä IT-tuen asiakaspalvelua automatisoimalla mobiilikaranteenitickettien käsittelyä ohjelmistorobotiikalla. Toteutus tehtiin ensin Robot Frameworkilla ja sitten UiPath-ohjelmistolla niiden vertailua varten. Lisäksi LEI-tunnuksia hakevien yritysten laskutusprosessia automatisoitiin Robot Frameworkilla tehdyllä toteutuksella. Toteutusten havaittiin nopeuttavan rutiiniprosessien käsittelyä huomattavasti. Ne paransivat myös prosessien kanssa työskennelleiden työntekijöiden tyytyväisyyttä välittömästi. Pitemmän ajan säästöistä laskettiin suuntaa antavia ennusteita, mutta osa säästöistä näkyy vasta ajan kanssa parempana asiakastytyväisyytenä ja työilmapiirin parannuksena.</p> <p>Lopputuloksena saatiin kaksi käyttöönotettua työprosesseja automatisoivaa ohjelmistorobottia, jotka on molemmat tehty käyttäen Robot Frameworkia. Niiden lisäksi saatiin paljon tietoa Robot Frameworkin soveltuvuudesta ohjelmistorobotiikkaan. Sillä pystyy tehdä suurimmaksi osaksi kaikki asiat, mitä ohjelmistorobotiikkaan vaaditaan. Jos jokin puute löytyy, niin sen voi paikata kehittämällä laajennus itse tai käyttämällä kolmannen osapuolen kehittämää avoimen lähdekoodin ratkaisua.</p> <p>Projekti kannusti PRH:n työyhteisöä miettimään, mitkä asiat voisi hoitaa ohjelmistorobotiikalla. Kehitysehdotuksia ja sovelluskohteita tuli kasvavaan tahtiin, mitä enemmän tietoisuus projektista levisi. Ohjelmistorobotiikan käyttökohteita havaitaan paremmin, kun ymmärretään, millaisia mahdollisuuksia se tuo työprosessien automatisoimiseksi.</p>	
Avainsanat	Robot Framework, RPA, ohjelmistorobotiikka, ohjelmistorobotti, UiPath

Author Title	Teemu Vaittinen Utilization of robotic process automation in a public government office
Number of Pages Date	50 pages + 6 appendices 28 March 2018
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Professional Major	Software Engineering
Instructors	Sakari Karstu, Chief Technology Officer Auvo Häkkinen, Principal Lecturer
<p>The aim of the thesis was to explore the capabilities of using robotic process automation in a government office and to specify Robot Framework's compatibility as a development tool for robotic process automation. For the client, which was the Finnish Patent and Registration Office (PRH), the most important thing was speeding up repetitive work processes to reduce expenses.</p> <p>PRH's internal IT-support was streamlined by automating mobile quarantine tickets by using RPA. It was first executed with Robot Framework and then UiPath's software for comparison. Additionally, the billing process of customers who apply for LEI codes was automated with a software robot developed with Robot Framework. Software robots sped up routine processes significantly. They also increased the immediate satisfaction of employees that had to work with said processes. Some estimations of long-term benefits were calculated but some of them will manifest over time in the form of improved customer satisfaction and working atmosphere.</p> <p>The outcome was the deployment of two software robots developed with Robot Framework. Along with them came a lot of information about Robot Framework's compatibility with RPA. It is capable of doing most things that are required for robotic process automation. Should there be an absence of a feature, a user-made solution or a third party open source solution can offer said missing feature.</p> <p>The project encouraged PRH's work community to consider what processes could be automated with RPA. The number of development and implementation suggestions grew at an increasing rate as the awareness of the project increased. The uses of RPA are identified more clearly, when the possibilities of it are realized.</p>	
Keywords	Robot Framework, RPA, robotic process automation, software robot, UiPath

Sisällys

Lyhenteet

1	Johdanto	1
2	Robot Framework ohjelmistorobotiikan työkaluna	3
2.1	Robot Frameworkin esittely	3
2.2	Esivaatimukset	5
3	Mobiilikaranteenitikkettien käsittelyn automatisointi	7
3.1	Toteutus	8
3.2	Kokeen arviointi	14
3.2.1	Hyödyt	16
3.2.2	Heikkoudet	18
3.3	Jatkokehitys	19
4	UiPath ohjelmistorobotiikan työkaluna	20
4.1	UiPathin esittely	20
4.2	Esivaatimukset	22
5	Mobiilikaranteenitikkettien käsittelyn automatisointi	23
5.1	Toteutus	23
5.2	Toteutusten vertailu	28
5.2.1	Hyödyt	30
5.2.2	Heikkoudet	30
5.3	Jatkokehitys	31
6	LEI-laskujen käsittelyn automatisointi	31
6.1	Toteutus	33
6.2	Robotin arviointi	39
6.2.1	Hyödyt	43
6.2.2	Heikkoudet	44
6.3	Jatkokehitys	45
7	Yhteenveto	46

Liitteet

Liite 1. Karanteenitietin käsittelyrobotti

Liite 2. Versiotiedot

Liite 3. LEI-laskuja käsittelevä robotti

Liite 4. LEI-laskutus Excel-tiedoston luontiskripti "excelinluonti.py"

Liite 5. Sähköpostin lähetysskriptit – "lahetaemail.py" ja "__init__.py"

Liite 6. Firefox-selaimen sulkemisskripti "close_ff.py"

Lyhenteet

ATDD	Acceptance Test-Driven Development on asiakkaan näkökulmaa lähtökoh- tana käytävä hyväksymistesteihin perustuva kehitys.
CSS	Cascading Style Sheets on verkkosivun tyyliohjeisto, joka määrää, millai- sena verkkosivu halutaan esittää katsojan selaimella. Sen avulla määrite- tään yhteiset muotoilusäännöt verkkosivustolle fonttien ominaisuuksista ja verkkosivun taustaväreistä komplekseihin animoituihin elementtiin. Sen perussyntaksi koostuu valitsimesta, ominaisuudesta ja arvosta.
ESMTP	Extended Simple Mail Transfer Protocol on määritelmä laajennuksista SMTP-protokollaan, jossa yhteys sähköpostipalvelimeen luodaan EHLO- komennolla. Palvelin vastaa listaamalla tukemansa laajennukset, jos pal- velin ylipäänsä ESMTP:tä tukee.
HTML	Hypertext Markup Language on standardoitu kieli, millä verkkosivut on kir- joitettu. Teksti on rakenteista ja sisältää erilaisia elementtejä peräkkäin ja sisäkkäin. Elementit erotetaan toisistaan kulmasulkeilla merkityillä tunnis- teilla.
IDE	Integrated Development Environment on ohjelmointiympäristö, joka on oh- jelma, jolla suunnitellaan ja toteutetaan ohjelmistoa. Ohjelmointiympäristö sisältää tavanomaisesti työkalut ohjelman suorittamiseen, kirjoittamiseen ja testaamiseen.
LEI	Legal Entity Identifier on kansainvälinen yhteisötunnus, minkä yritys tai yh- teisö tarvitsee tehdäkseen kauppvoja pörssiosakkeilla. Tunnus haetaan PRH:lta.
OCR	Optical character recognition on teknologia, jonka avulla tunnistetaan ja tulkitaan tekstiä sähköisesti.
PRH	Patentti- ja rekisterihallitus on valtion virasto, joka hoitaa Suomen yhdistys- ten rekisterit sekä myöntää patenteja ja tavaramerkkejä.

RPA	Robotic Process Automation on ohjelmistorobotiikka, ja se on teknologia, minkä avulla automatisoidaan rutiininomaisia prosesseja tietokoneella tehtävässä työssä.
RIDE	Robot Integrated Development Environment on Robot Frameworkin ohjelmointiympäristö.
SFTP	SSH File Transfer Protocol on protokolla, minkä avulla tiedostoja käsitellään turvallisesti Secure Shell -yhteydellä kahden järjestelmän välillä.
SMTP	Simple Mail Transfer Protocol on protokolla, minkä avulla lähetetään sähköpostiviestejä eri sähköpostipalvelimien välillä. Yhteys luodaan HELO-komennolla, jonka mukana ilmoitetaan lähettäjän domain-nimi tai IP-osoite.
SSH	Secure Shell on verkkoprotokolla, minkä tarkoitus on salata tietoliikenne erilaisilla salaus- ja tunnistusmenetelmillä. Suojattu kaista kahden järjestelmän välillä muodostetaan käyttäjäautentikoinnilla, jota varten käyttäjä tarvitsee salausavainparin.
UML	Unified Modeling Language on standardoitu graafinen mallinnuskieli, jota käytetään ohjelmistosuunnittelussa kuvaamaan ohjelman rakennetta, käyttäytymistä ja vuorovaikutusta.
Valtori	Valtion tieto- ja viestintätekniikkakeskus, joka tarjoaa toimialariippumattomia tietotekniikan palveluita valtionhallinnolle.
XML	Extensible Markup Language on standardoitu kieli, millä kuvataan tietoa rakenteellisessa muodossa. Teksti muistuttaa rakenteeltaan HTML-kieltä, mutta sivunkuvauksen sijaan se kuvaa tiedon rakennetta ilman etukäteen määriteltyä koodia.
XPath	XML Path Language on kieli, millä ositetaan XML-dokumenttien eri osia toisistaan. Käytetään usein hakemaan jokin tietty elementti XML-dokumentin puumaisesta rakenteesta.

- VYVI Valtion yhteinen viestintäratkaisu on valtion keskitetty IT-palvelu, jonka tarkoitus on tarjota valtionhallinnon organisaatioille yhtenäiset viestintävälineet.
- YTJ Yritys- ja yhteisötietojärjestelmä on Verohallinnon ja PRH:n yhteinen palvelu, jossa voi ilmoitetut tiedot menevät yhdellä ilmoituksella molemmille viranomaisille.

1 Johdanto

Suurin osa nykyaikaisesta toimistotyöstä tehdään tietokoneella – työntekijä tulkitsee tietoa selaimelta, syöttää tietoa erilaisiin kenttiin ja välittää tietoa sähköpostilla. Ohjelmistorobotiikan avulla voisi suorittaa tehtävästä riippuen osan työntekijän tekemistä rutiinistöistä, mutta ala ja sen tuomat edut ovat vielä hieman tuntemattomia työmarkkinoilla (11). Insinööriyön aiheena on selvittää käyttökohteita ja hyödyntämismahdollisuuksia ohjelmistorobotiikalle julkishallinnon virastolle, tässä tapauksessa siis PRH:lle, eli Patentti- ja rekisterihallitukselle sekä kokeilla toteuttaa joitakin niistä. PRH harkitsee ohjelmistorobotiikan laajempaa käyttöönottoa työprosessien sujuvoittamiseksi.

Ohjelmistorobotiikka, eli Robotic Process Automation lyhyesti RPA, on tietokoneella suoritettavien rutiiniprosessien automatisoimista ohjelmistorobotilla, joka käyttää prosessien suorittamiseen lähtökohtaisesti samoja ohjelmia, joita ihminenkin käyttäisi. Sen tarkoitus on korvata ihmisen paikka itseään toistavien ja ennustettavien töiden teossa. Kun ohjelmistorobotille määritellään tarkat toimintaperiaatteet ja säännöt, nopeuttaa se prosessin käsittelyä, parantaa dokumentointia ja vähentää virheiden määrää. Ohjelmistorobotiikan tarkoituksena on opettaa robotti simuloimaan ihmisen tekemiä toimenpiteitä mahdollisimman tarkasti muuttamatta itse työprosessia.

RPA:n maailmanlaajuinen markkinakoko oli vuonna 2015 arvioiden mukaan noin 125,2 miljoonaa dollaria, vuonna 2016 se oli kasvanut yli 600 miljoonaan dollariin. (19, 17) Markkinaennusteiden mukaan markkinakoko kasvaisi yli 12 miljardiin vuoteen 2024 mennessä (18).

Työn aihepiiristä ei ollut aiempaa kokemusta virastossa, joten taustatieto hankittiin itsenäisellä opiskelulla. Ohjelmointitaitausta on koulutuksen ja harrastuneisuuden kautta muun muassa Pythonin, Javan ja C-ohjelmointikielistä. Ohjelmistorobotiikan kehitysalustoja kuvaillaan helposti opittaviksi ja esivaatimuksena ei yleensä vaadita ohjelmointitaitausta. Tarkoituksena on myös arvioida eri ohjelmistorobotiikan kehitystyökalujen käytettävyyttä ja soveltuvuutta organisaatioon, jonka henkilöstöllä ei vielä ole ohjelmistorobotiikasta juurikaan kokemusta. Mainittakoon myös, että PRH:lla oli työjakson aikana kaksi ulkoista konsulttia testausautomaation parissa, joilta saatiin myös hieman apua alkuun pääsyyn.

Periaatteessa ohjelmistorobotiikan avulla voisi automatisoida melkein minkä tahansa tietokoneen käyttöprosessin, mutta se ei välttämättä maksa itseään takaisin. Monimutkaisten, kognitiivista päättelykykyä vaativien prosessien automatisointi on hyvin monimutkaista, minkä takia niiden automatisointi olisi erittäin työlästä. Vaikka työlään toteutuksen saisikin käytettävään kuntoon, jää robotin toimintojen onnistumisprosentti alhaiseksi tai kehityskustannukset nousevat tähtitieteellisiksi. Siksi on tärkeää tunnistaa ohjelmistorobotiikan käyttökohteet oikein ja arvioida niiden kannattavuus ennen toteutuksen suunnittelua.

Jos ohjelmistorobotiikkaa yrittää käyttää sellaiselle kohteelle, mille ei ole tehty kunnollista kustannus-hyötyanalyysia, niin lopputulos saattaa olla vain marginaalisesti hyödyllinen. Silloin kustannukset yleensä ylittävät ratkaisun tuomat hyödyt ja todellista lisäarvoa lopputuloksesta ei saada. (14.)

Onnistunut ohjelmistorobotiikan sovelluskohde säästää yleensä satoja, ellei tuhansia työtunteja sekä parantaa työn tarkkuutta sekä nopeutta. Onnistumiskriteerinä on myös robotin kehitykseen kuluneiden resurssien takaisinmaksuaika, joka on aika, jolloin robotin toteutuksesta kertyneet säästöt ylittävät sen kehityskustannukset. Jos aika on kovin pitkä, niin automaation kohde valittiin ehkä huonoin perustein. Hyötyihin lasketaan ajallisten ja rahallisten säästöjen lisäksi myös työviihtyvyyden parantaminen, käsiteltävien prosessien laadun parantuminen ja skaalautuvuus.

Kokeilun toteuttamiseen valittiin ohjelmointiympäristöksi Robot Framework sisäisen sopimisen sekä avoimen lähdekoodin kokeilunhalun perusteella ja UiPath EERA-konsultointiyhtiön selvityksen (21) perusteella. Päätökseen vaikutti myös se, että Robot Frameworkilla oli tehty toimivia ohjelmistorobotteja Liikenteen turvallisuusvirasto Trafissa. Robot Framework oli myös entuudestaan tuttu PRH:lla. Kolmantena vaihtoehtona ohjelmointiympäristöksi oli vielä BluePrism, mutta sen kokeiluun ei ollut aikaa. Päämääränä oli määrittää eri vaihtoehtojen soveltuvuus PRH:n työprosessien automatisointiin, arvioida niiden rajat ja niiden tuomat mahdollisuudet sekä säästöt. Tavoitteena on lisäksi kokeilla testausautomaatioon tarkoitettun Robot Frameworkin mahdollisuuksia ohjelmistorobotiikassa.

Insinööriyössä suunniteltiin ja toteutettiin ohjelmistorobotiikan pilotti käsittelemään PRH:n verkkopalvelimen kautta toimivan tikettijärjestelmän tikettejä. Sen tuottamia hyötyjä ja menetelmän heikkouksia arvoitiin toteutuksen lopuksi. Tikettejä käsittelevä robotti

toteutettiin myös UiPathin ohjelmistolla vertailun vuoksi. Sen jälkeen suunniteltiin ja toteutettiin LEI-tunnuksien laskutusprosessia automatisoiva ohjelmistorobotti, jonka sekä hyviä että huonoja puolia pohdittiin PRH:n kannalta.

2 Robot Framework ohjelmistorobotiikan työkaluna

Ensimmäiseksi ohjelmistorobotiikan testauskohteeksi oltiin alustavasti sovittu työnantajan kanssa työpaikan IT-helpdeskin mobiilikaranteenitikkettien käsittelyn automatisoinnin. Tikettijärjestelmä sijaitsee verkkosivulla, joka on suomalaisen ohjelmistoyhtiön Efecten Oyj:n kehittämä palvelu (7), minkä kautta PRH hallinnoi IT-palveluitansa. Automaatiotapauksen säästöennusteet eivät olleet korkeat, koska karanteenitikkettejä luotiin tavanomaisena työpäivänä enintään viisi kappaletta, yhden käsittelyyn kuluu vain pari minuuttia. Kohde valittiinkin lähinnä sen yksinkertaisuuden vuoksi ja sen tarkoituksena oli auttaa tutustumaan Robot Frameworkiin ja sen ohjelmointiympäristöön.

Lähtökohtana on siis testausautomaatioon laajalti käytetty työkalu, jonka käyttö ohjelmistorobotiikan saralla on ehkä kyseenalaista (11), mutta työkalu ei maksa mitään, niin sen kokeilukynnys on matala ja säästömahdollisuudet korkeat. Varsinkin kun huomioon ottaa, ettei laajalti käytössä olevia avoimen lähdekoodin ohjelmistorobotiikkatyökaluja ole vielä oikein olemassa.

2.1 Robot Frameworkin esittely

Robot Framework sai alkuperäisen konseptinsa Pekka Klärckin lopputyöstä Data-Driven and Keyword-Driven Test Automation Frameworks vuonna 2006. Työkalun tarkoitus on auttaa luomaan yksinkertaisia ohjelmiston manuaalista tai automaattista testausta varten olevia testitapauksia avainsanapohjaisella ohjelmiston testausympäristöllä. (5.) Testitapauksen suoritusta seurataan joko reaaliaikaisesti tai jälkikäteen tarkastelemalla suoritustotteita ja lokitiedostoja. Robot Framework ei itsessään sisällä mitään ulkopuolisen suorituksen seurannan mahdollistavaa työkalua.

Robot Framework valittiin, koska PRH:lla on siihen osaamista testausautomaation puolelta, joten se on hieman tuttu ympäristö PRH:lle. Jatkokehityksen tai ylläpidon voi siis hoitaa mahdollisesti PRH:n oma työntekijä tutussa kontekstissa. Valintaan vaikutti myös

avoimen lähdekoodin ratkaisun tuoma ulkoisten lisäosien suuri määrä. Jos jollekin toiminnolle ei löydy valmista kirjastoa, niin sellainen voidaan tarvittaessa ohjelmoida itse. Robot Framework on hyvin joustava tuettujen alustojen ja ohjelmointikielten kanssa. Se tukee pääasiassa Pythonia, mutta myös Java-alustalla toimivaa Jythonia sekä .NET-kielten alustalla toimivaa IronPythonia. Vaikka Robot Framework on itsessään suunnattu testiautomaatioon (2), on sen käyttö selain- ja työpöytäautomaatiota varten mahdollista juuri ulkoisten lisäosien yhteensopivuuden ansiosta.

Joka kerta, kun testitapauksista koostuva kokonaisuus ajetaan Robot Frameworkilla, syntyy Hypertext Markup Language, lyhyesti HTML-formaatissa oleva loki, joka sisältää tietoa suoritetuista testitapauksista. Lokitus tapahtuu kahdessa muodossa: sekä ylemmän tason raporttiedostona, että yksityiskohtaisena lokitiedostona. Suorituksesta on myös saatavilla xUnit-yhteensopiva Extensible Markup Language, eli XML-tiedosto, mutta ohjelman suorituksen yhteydessä pitää näin erikseen määrittellä. Huomionarvoista on, että xUnit-raportointi ei erota kriittistä ja epäkriittistä testitapausta toisistaan, vaan käsittelee kaikki testitapaukset vain sen onnistumisen tai epäonnistumisen perusteella. (3.)

Käyttäjä voi itse päättää lokituksen tason Robot Frameworkin avainsanoilla määrittämällä lokiviestien tason. Jokaisen lokitetun tiedon tason voi määrittää erikseen avainsanoilla WARN, INFO, DEBUG tai TRACE (Taulukko 1). Näiden lisäksi on myös FAIL, jota vain Robot Framework voi käyttää itse.

Taulukko 1. Robot Frameworkin version 3.0.2 lokitasot korkeimmasta pienimpään. Mitä alempi lokitaso, sitä yksityiskohtaisemmin tiedot kirjataan lokiin.

Robot Frameworkin lokituksen tason avainsana	Käyttö
FAIL	Käytetään, kun minkä tahansa avainsanan käyttö epäonnistuu. Vain Robot Frameworkin omassa käytössä.
WARN	Käytetään varoitusten antamiseen. Varoitukset näkyvät myös konsoli-ikkunassa ja suoritusotteen virheet -osiossa lokitiedostossa. Ei vaikuta suorituksen onnistumiseen.
INFO	Lokituksen oletustaso. Oletuksena tätä alemman tason lokitukset eivät näy suoritusotteessa.
DEBUG	Virheiden tarkistusta varten oleva lokitustaso, jossa näkyy myös kirjastojen sisäinen toiminta. Jos tällä tasolla avainsanan suoritus epäonnistuu, niin näyttää missä kohtaa koodia epäonnistuttiin.
TRACE	Yksityiskohtaisempi virheiden tarkistustaso. Avainsanojen argumentit ja palautusarvot kirjataan automaattisesti lokiin tällä tasolla.

Valmiin ohjelman lokitaso kannattaa pitää yleisesti INFO-tasolla, varsinkin jos koodi sisältää arkaluontoista tietoa, sillä tarkemman tason lokitus tosiaan kirjaa kaiken tapahtuman ylös, vaikka olisi määritellyt tietueen arkaluontoiseksi muuttujaksi Password-avainsanalla.

Jos alimmallakaan lokituksen tasolla ei löydy haluttua tietoa, niin vielä yhtenä mahdollisuutena on tutkia Robot Frameworkin sisäistä lokitiedostoa. Sitä ei normaalisti luoda suorituksen yhteydessä, mutta sen voi laittaa päälle asettamalla ROBOT_SYSLOG_FILE-ympäristömuuttujan ja sille halutun sijainnin polun. Robot Frameworkin systeemiloki sisältää samat lokitasot kuin tavanomaisesti luotu lokitiedosto, mutta FAIL-tason tilalla on ERROR-taso. Systeemiin kirjataan tietoa prosesseista, käytetyistä kirjastoista, suoritetuista testitapauksista ja tulosteesta.

Tarkemmat lokitasot tuottavat huomattavan määrän lokia. Lokin määrän optimoimiseksi on määriteltävä lokitustaso huolella, varsinkin jos robottia suoritetaan tiheällä aikavälillä. Lokitiedostojen koon ja sisällön hallinta onnistuu antamalla ohjelmistorobotille erilaisia suoritusparametreja, joilla voi jakaa lokin eri osiin, estää lokin tai sen eri osien luontia kokonaan tai säätää, mitkä asiat kirjataan lokiin ja mitkä eivät.

2.2 Esivaatimukset

Toisin kuin yleisimmillä ohjelmistorobotiikan kehitysalustoilla, Robot Frameworkin käyttöön vaaditaan käytännössä hieman ohjelmointitaitojen osaamista. Varsinkin siinä tapauksessa, jos jokin avainsana tai toiminnallisuus ei löydy sisäänrakennettuna kehitystyökalusta. Vaikka toiminnallisuus löytyisi kolmannen osapuolen kehittämänä, niin sen asennuksessa, käyttöönotossa ja oikean käytön hallinnassa suositellaan ohjelmoinnin perustoimintojen ymmärtämistä (9). Teoriassa on mahdollista kehittää ohjelmistorobotti käyttämällä pelkästään Robot Frameworkin sisäänrakennettuja toimintoja, mutta käytännössä jokainen työpöytäohjelma vaatii automaattiohjausta varten jonkun erillisen kirjaston.

Testitapausten suunnitteluun ja kirjoittamiseen käytettiin Robot Integrated Development Environment -kehitystyökalua RIDEa, jonka toiminta edellytti wxPythonin asennusta (6). Valtaosa robotin toiminnoista suoritetaan Efecte-tikettijärjestelmällä, selaimen kautta. Jotta ohjelmoitava robotti saisi ohjattua verkkoselaimen toimintoja, vaatii se selaimen

tukeman automaatiokirjaston, jota varten valittiin Selenium WebDriver ja sen käyttöä edellyttävä Selenium-kirjasto (8).

Selenium-kirjasto täydentää myös robotin suorituslokeja ottamalla poikkeustilanteen satuesssa kuvakaappauksen selainikkunasta. Näiden avulla on helpompi todeta jälkikäteen, missä tilassa robotin suoritus on ollut poikkeustilanteen tapahtumahetkellä. Kuvakaappaukset generoidaan ”Capture Page Screenshot” -avainsanalla automaattisesti tai käyttäjän kutsumana. Kuvat liitetään automaattisesti Robot Frameworkin luomaan raporttiin ja ovat nähtävillä sieltä suoraan seuraavan esimerkkikuvan (kuva 1) tapaisesti.

TEARDOWN BuiltIn.Run Keyword And Ignore Error Onko Nakymassa Kasiteltavia Tiketteja 00:00:05.403
Documentation: Runs the given keyword with the given arguments and ignores possible error.
Start / End / Elapsed: 20180213 14:20:12.936 / 20180213 14:20:18.339 / 00:00:05.403

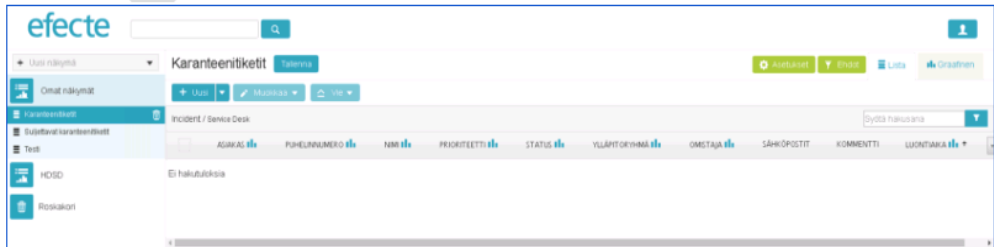
KEYWORD Onko Nakymassa Kasiteltavia Tiketteja 00:00:05.402
Start / End / Elapsed: 20180213 14:20:12.937 / 20180213 14:20:18.339 / 00:00:05.402

KEYWORD ExtendedSelenium2Library.Set Selenium Implicit Wait 5 seconds 00:00:00.002

KEYWORD ExtendedSelenium2Library.Wait Until Element Is Visible \${TIKETTILISTAN_ENSIMMAINEN_ALKIO} 00:00:05.394
Documentation: Waits until element specified with 'locator' is visible.
Start / End / Elapsed: 20180213 14:20:12.939 / 20180213 14:20:18.333 / 00:00:05.394

KEYWORD ExtendedSelenium2Library.Capture Page Screenshot 00:00:00.382
Documentation: Takes a screenshot of the current page and embeds it into the log.
Start / End / Elapsed: 20180213 14:20:17.949 / 20180213 14:20:18.331 / 00:00:00.382

14:20:18.331 INFO



14:20:18.333 FAIL ValueError: Element locator '//*[@id="workspace_table_0"]/tbody/tr[1]' did not match any elements.

Kuva 1. Esimerkkiote Robot Frameworkin suoritusraportista, missä on mukana Selenium-kirjaston ottama automaattinen kuvakaappaus poikkeuksen tapahtumahetkellä. Suoritusraportti on luotu mobiilikaranteenitikkien käsittelyn automatisointityön kehityksen aikana.

Kuvan 1 tapauksessa poikkeuksen aiheutti etsityn elementin puuttuminen kyseiseltä sivulta. Automaattisesti otettujen kuvakaappausten koko on hyvin pieni – keskimäärin alle 50 kilotavua, minkä seurauksena yhden suorituksen seurauksena syntyvien lokien koko saattaa paisua hieman projektin koon kasvaessa.

Tikettijärjestelmän verkkosivu käyttää AngularJS-elementtejä eikä perus-Selenium-kirjasto tarjoa työkaluja niiden käsittelyyn, joten sitä varten tarvittiin Extended Selenium2 -kirjasto (4). Selenium-kirjaston versioksi valittiin vanha 1.8.0-versio ExtendedSelenium2Libraryn yhteensopivuuden takaamiseksi. Selaimeksi valittiin Google Chrome,

koska sen toimimaan saaminen oikealla verkkoajurilla oli vaivattomin. Vaihtoehtoina oli myös Internet Explorer ja Mozilla Firefox, mutta IE:n toiminnan takaaminen on vaikeampaa kuin muiden selainten verkkoajurien kanssa Trafin IT-asiantuntija Jukka Vuennon mukaan. (1.)

3 Mobiilikaranteenitickettien käsittelyn automatisointi

Työntekijän kännykkä joutuu mobiilikaranteeniin, jos yhteys Tiedon sähköpostihallinnoinnin ja mobiililaitteen välillä katkeaa. Mobiililaitteen ensimmäisellä käyttöönottokerralla yhteys luodaan karanteeniticketin muodossa mobiililaitteen ja sähköpostitilin välille. Tämä tarkoittaa myös, että aina kun henkilö vaihtaa kännykkää joutuu uusi kännykkä karanteeniin. Yhteys saattaa katketa mobiililaitteen versiopäivitysten vuoksi tai sähköpostin liian monen peräkkäisen virheellisen kirjautumisyhteyden seurauksena, mikä aiheuttaa myös mobiilikaranteenin.

Mobiilikaranteeniticketin käsittelyn työnkulku alusta loppuun:

1. Työntekijän kännykän sähköposti joutuu mobiilikaranteeniin.
2. Työntekijä saa asiasta karanteeniviestin Outlookin kautta.
3. Karanteeniviesti välitetään sähköpostilla PRH:n tikettijärjestelmän sähköpostiosoitteeseen.
4. Tikettijärjestelmä vastaanottaa välitetyn sähköpostin uutena IT-tuen tiketinä.
5. PRH:n tukihenkilö välittää karanteeniviestin Valtion tieto- ja viestintätekniikkakeskuksen, lyhyesti Valtorin (10) valtion yhteisen viestintäratkaisun-, eli VYVI-sähköpostin tukeen.
6. Tieto käsittelee tukipyynnön, avaa työntekijän mobiilikaranteenin ja ilmoittaa PRH:n ticketin kautta ticketin ratkaisusta.
7. PRH:n tukihenkilö sulkee ticketin saatuaan vahvistuksen mobiilikaranteenin raukeamisesta.
8. Työntekijä saa ilmoituksen tikettinsä valmistumisesta ja mobiilikaranteenin raukeamisesta ja voi taas jatkaa kännykkänsä sähköpostin käyttöä.

Ideaalitapauksissa työnkulku on aina sama, mutta joskus työntekijä ei seuraa hänelle saapuneen karanteeniviestin ohjeistusta oikein ja seurauksena on virheellisesti tehty ticketti, joka ei sisällä mobiilikaranteeniticketin olennaisinta tietoa: mobiililaitteen laitetunnusta. Robotin on tarkoitus automatisoida PRH:n tukihenkilön roolisuoritus työkulusta sekä tunnistaa ja sulkea virheellisesti tehdyt ticketit ja ilmoittaa ticketin luoneelle henkilölle oikeat toimenpiteet.

3.1 Toteutus

Robotin työnkulku:

1. Avaa Chrome-selain Efecten kirjautumissivulle.
2. Kirjaudu robotin Active Directory, eli AD-tunnuksilla Efecten palveluun.
3. Valitse käsittelemättömien karanteenitickettien näkymä.
4. Jos listalla on yhtään tikettiä, niin avaa listan ensimmäinen tiketti muokattavaksi. Jos ei, niin hyppää kohtaan 8.
5. Jos tiketti sisältää tekstin ”Laitteen tunnus”, niin oletetaan ticketin olevan oikein tehty. Jos tekstiä ei löydy, niin oletetaan, ettei tiketti ole oikein tehty, sulje tiketti, ilmoita ticketin luoneelle henkilölle ticketin kommentti-kenttään lisäohjeet ticketin oikein luomiseksi ja siirry kohtaan 8.
6. Aseta ticketin omistajaksi Roope Robotti1, statukseksi ”Odottaa ulkopuolista”, tyyppiä ”Palvelupyyntö”, palveluryhmäksi ”Yleiset IT-palvelut /Helpdesk”, palveluksi ”Työasema- ja käyttäjätukipalvelut” ja kirjoita ticketin toimenpide-kenttään, että tiketti on käsitelty automaattisesti.
7. Lähetä tiketti Tiedon VYVI:n sähköposti –tuelle ja tallenna tickettiin tehdyt muutokset.
8. Valitse suljettavien karanteenitickettien näkymä.
9. Jos listalla on yhtään tikettiä, niin avaa listan ensimmäinen tiketti muokattavaksi. Jos ei, niin jatka kohtaan 10.
10. Jos Tieto on vastannut tickettiin sähköpostilla ja se sisältää tekstin ”Tiketti on ratkaistu”, niin sulje tiketti. Jos tekstiä ei löydy, niin jatka seuraavaan askeleeseen muokkaamatta tikettiä.
11. Kirjaudu ulos Efecten palvelusta ja sulje selain.

Ensimmäiseksi haasteeksi ilmeni mobiilikaranteenitickettien tunnistaminen ja erottaminen muista tiketeistä. Tunnistamista varten haastateltiin monen vuoden työkokemuksen omaavia tickettijärjestelmää käyttäneitä tukihenkilöitä. Haastattelun perusteella todettiin, että suuri osa karanteenitiketeistä on välitetty oikein ja niiden otsikkona on ”VL: Mobiililaitettasi on tilapäisesti estetty synkronoimasta Exchange ActiveSyncin kautta, kunnes järjestelmänvalvoja sallii laitteen pääsyn”. Lopuilla vapaamuotoisemmilla itsetehdyillä karanteenitiketeillä on lähes poikkeuksetta otsikossa sana ”karanteeni”.

Karanteenitickettien erottelu muista tiketeistä onnistui helpoimmin luomalla Efecten ticketinäkymän, jossa oli näkyvyyssehtoina, että ticketin Nimi-kenttä sisältää ”*karanteeni” tai ”*mobiililaitettasi” ja ticketin Status on käsittelemätön. Jos ticketin Status on käsittelemätön, niin oletetaan, ettei tikettiä ole vielä kukaan aukaissut käsiteltäväksi. Seuraavassa kuvassa 2 ovat näkyvillä suodatusehdot.

The image shows a search filter interface with two sections. The top section contains three filter rows. The first row has 'Nimi' (Name) set to 'sisältää' (contains) with the value '*karanteeni'. The second row has 'TAI' (OR) selected. The third row has 'Nimi' set to 'sisältää' with the value '*mobiililaitetasi'. Below the first row is a 'Lisää ehto' (Add condition) button. Below the second row is a 'Lisää ehto' button. The third row has 'Status' (Status) set to 'sama kuin' (same as) with the value '1 - Käsittelemä...'. Below the third row is another 'Lisää ehto' button.

Kuva 2. Robotin karanteenitickettien hakuehdot. Kuvakaappaus tikettijärjestelmän suodatusehdoista.

Kuvan 2 hakuehdot tallennettiin näkymään nimeltä "Karanteeniticketit". Kun hakuehtojen edessä on tähtimerkki, niin se hyväksyy merkeiksi mitä tahansa haettavaa edeltäväksi tekstiksi. Ilman sitä hakuehtoihin ei sisältyisi tikettejä otsikoilla, joissa sana "karanteeni" on yhdyssanan jälkimmäinen osa.

Kun robotti käsittelee hakuehdot täyttävän tiketin, niin seuraava askel on odottaa Tiedon vastausta ja sulkea tiketti ratkaisuviestin saavuttua. Sen jälkeen tiketti pitäisi olla aina samassa tilassa, eli tiketin Status on "Odottaa ulkopuolista", Omistaja on Robotti1 Roope ja tiketissä on lukemattomia sähköposteja. Ensikäsitteilyn läpikäyneitä tikettejä varten luotiin toinen oma tikettinäköymä, jonka suodattimiksi asetettiin samat Nimi-kentän sisällöt kuin aiemmin ja lisäksi edellä mainittujen tiketin tilan mukaiset suodatusasetukset. Seuraavassa kuvassa 3 ovat näkyvillä suodatusehdot.

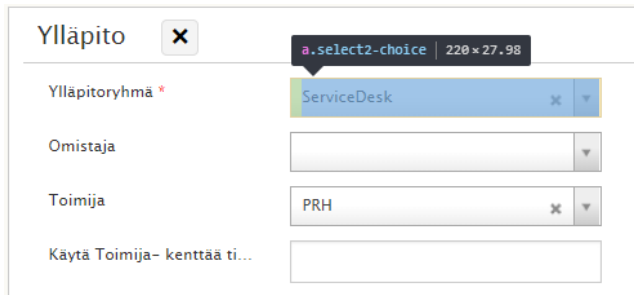
Kuva 3. Kuvakaappaus suljettavien karanteenitickettien suodatusasetuksista.

Kuvan 3 hakuehdot tallennettiin näkymään nimeltä ”Suljettavat karanteenitikit”. Kun oikeat suodattimet tallennettiin omiksi näkymikseen, niin seuraava vaihe oli ohjelmoida robotin logiikka tunnistamaan verkkosivun tärkeät elementit ja klikkaamaan niitä oikeissa järjestyksissä. Selenium-kirjaston käyttö selaimen hallintaan vaatii verkkoelementin tunnistamiseen ja käyttöön tunnistimen, joka annetaan haettavan elementin ominaisuudesta. Sitä varten tutkittiin tikkopalvelua selaimen ”Tarkista”-työkalulla, millä pystyy analysoida tarkistettavan verkkosivun elementtien ominaisuuksia sivun lähdekoodista.

Elementtien tunnisteiden poimintaan tai robotin prosessin työpolun tallentamiseen olisi voinut myös käyttää Seleniumin omaa ohjelmointiympäristöä Firefox-selaimelle olevalla lisäosalla. Sen toimintaa kokeiltiin toteutuksen alussa, mutta elementtien tunnisteet poimittiin usein liian epätarkasti ja niitä piti käsin muuttaa tarkemmaksi. Vaihtoehtona oli myös Chrome-selaimen Katalon Automation Recorder, mikä oli toiminnaltaan hyvin samanlainen Seleniumin lisäosan kanssa, mutta siinä havaittiin kaatumisongelmia ja toiminnan nykimisongelmia, joten senkin käytöstä luovuttiin pian.

Vaikka tunniste voi olla mikä tahansa verkkoelementin ominaisuus tai osa verkkosivun lähdekoodista, kannattaa se virheiden välttämiseksi olla jokin ominaisuus, millä löytyy

vain yksi elementti. Jos tunniste löytyy useasta verkkoelementistä, pysähtyy etsintä ensimmäisen tunnisteeseen omaavan elementin kohdalle. Usein paras tunniste on elementin yksilöity id, mutta tikettijärjestelmän verkkosivuelementit on luotu joltain geneeriseltä pohjalta. Tästä syystä joillekin elementeille piti löytää jotain muuta yksilöivää. Esimerkiksi tikettiä muokatessa pudotusvalikkoelementit eivät sisällä itsessään mitään toisistaan erottuvaa. Kuvaan 4 on poimittu yksi monista pudotusvalikoista, jolla ei ole yksilöivää id:tä.



Kuva 4. Kuvakaappaus muokkaustilassa olevasta tiketin osasta. Elementin tarkistajan tiedot korostettuna avaamattomasta pudotusvalikosta. Kaikki tikettijärjestelmän pudotusvalikot ovat "select2-choice"-elementtejä ilman yksilöivää tunnistetta.

Pudotusvalikon yläelementti sisältää kyllä id:n, mutta se luodaan dynaamisesti sivun latautuessa, joten sen arvo vaihtelee. Sitäkään ei siis voitu käyttää luotettavasti oikean pudotusvalikon tunnistamiseen. Sen sijaan tunnistettiin ensin oikea rivi, miltä hakea pudotusvalikkoa. Oikea rivi tunnistettiin rivin sisältämän tekstin perusteella. Pudotusvalikon hausta tehtiin oma avainsana, mikä ottaa argumenttina rivin otsikon tekstin ja pudotusvalikon valinnan tekstin ja klikkaa löydettyjä elementtejä.

```
Tee Valinta Pudotusvalikosta
  [Arguments]      ${arg_rivin_nimi}      ${arg_valinta}
  [Documentation] Argumenttina ensin rivin nimi ja toisena valinnan teksti

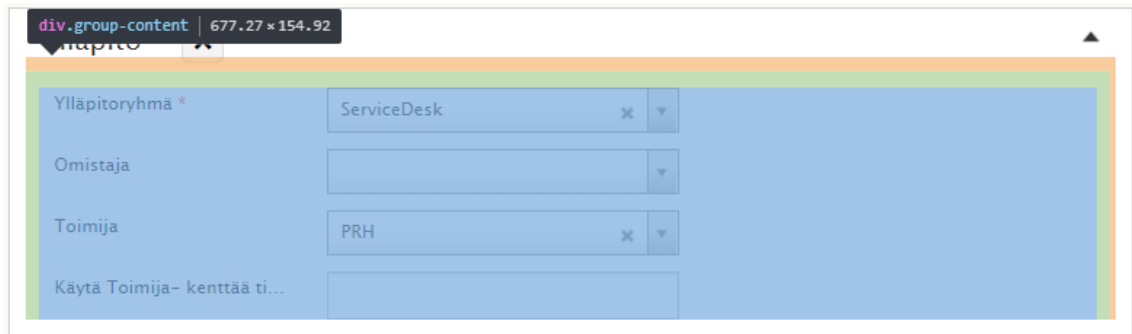
  Scroll Element Into View  /*[@class="group-content"]/div[*][.//text()='${arg_rivin_nimi}']

  Click Element  /*[@class="group-content"]/div[*][.//text()='${arg_rivin_nimi}']/*[contains(@id,'::data.gaId')]

  Click Element  /*[@class="select2-results"]/li[*][.//text()='${arg_valinta}']
```

Esimerkkikoodi 1. Avainsana "Tee Valinta Pudotusvalikosta" vierittää argumenttina saamansa rivin ensin näkyville, sitten klikkaa siitä pudotusvalikon auki. Lopuksi klikkaa auki olevasta pudotusvalikosta argumenttia vastaavaa valintaa. Elementtien tunnisteet haetaan XML Path Language, eli XPath-kyselyinä, jolla käydään Extensible Markup Language, lyhyesti XML-tietoa läpi.

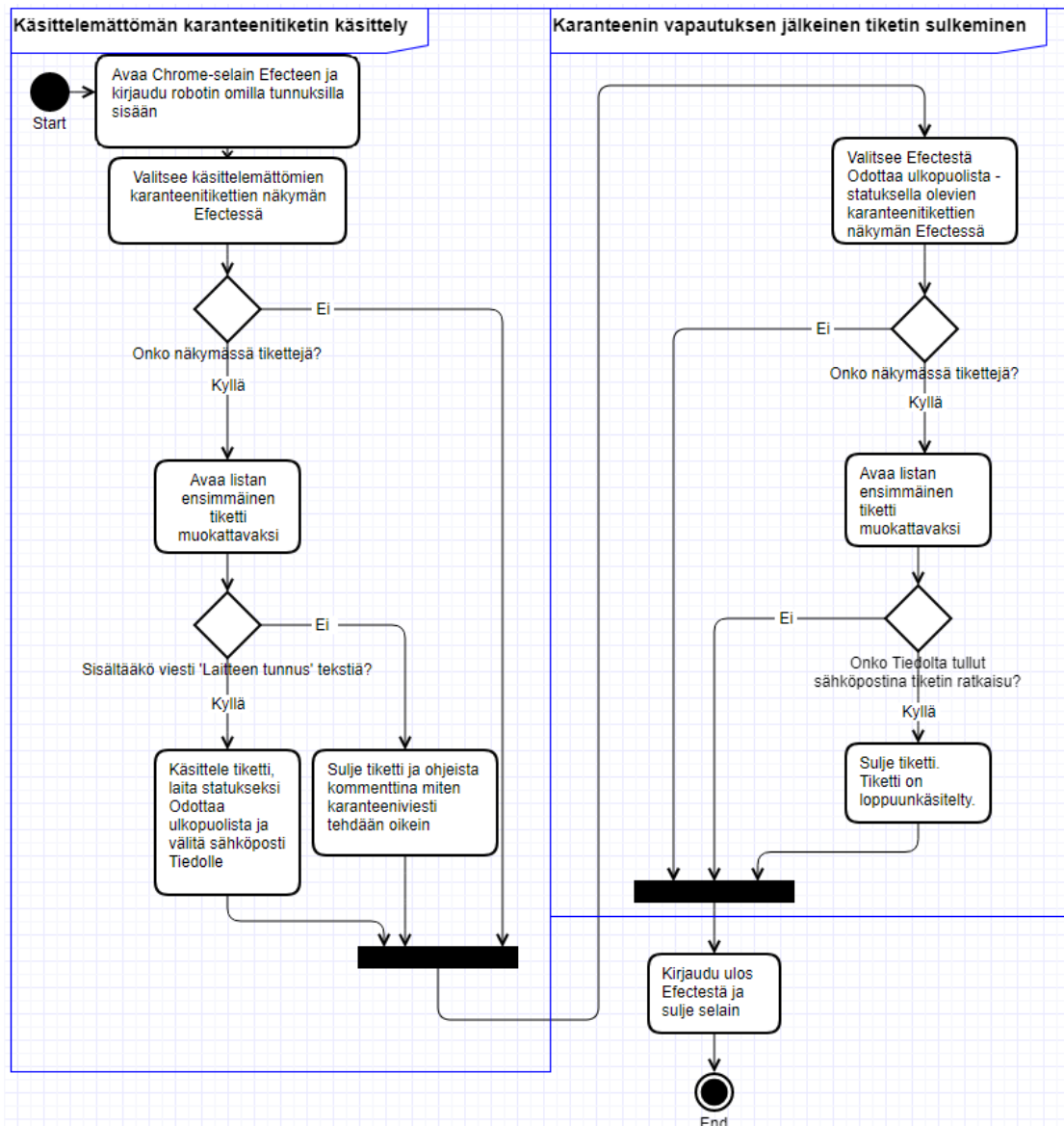
Tunnistamiseen ei riittänyt pelkästään rivin tekstin etsintä ja kentän sisältämän pudotusvalikon elementti. Sisäkkäisiä elementtejä on tikkettijärjestelmän sivulla niin paljon, että oikean tekstirivin löytämiseen piti spesifioida, minkä ylemmän elementin sisältä riviä etsitään. Ylempi elementti on rakenteeltaan taulukon muotoinen, joka sisältää haettavat rivit "div"-HTML-tunnisteen mukaisissa osioissa. Ylempi elementti on korostettu seuraavassa kuvassa 5.



Kuva 5. Elementin tarkistuksella korostettu "group-content"-elementti, joka sisältää rivien elementit. Tämä elementti on verkkosivulla jo yli 20 toisen elementin sisällä.

Hyvin syvälle sisäkkäisten elementtien löytäminen saattaa epäonnistua tarkentamatta ylempiä elementtejä, vaikka elementin XPathin kopioisi suoraan valitusta elementistä. XPath-viite halutulle elementille on helppo kopioida selainten elementtien tarkistuksen kautta. Menetelmää käytettiin pääasiallisena tapana saada verkkoelementtien tunnistuksista kiinni.

Robotin käyttöönoton jälkeen robotin toimintalogiikasta tehtiin aktiviteettikaavio (kuva 6), mistä näkee robotin loogisen suoritusjärjestyksen ehtoineen.

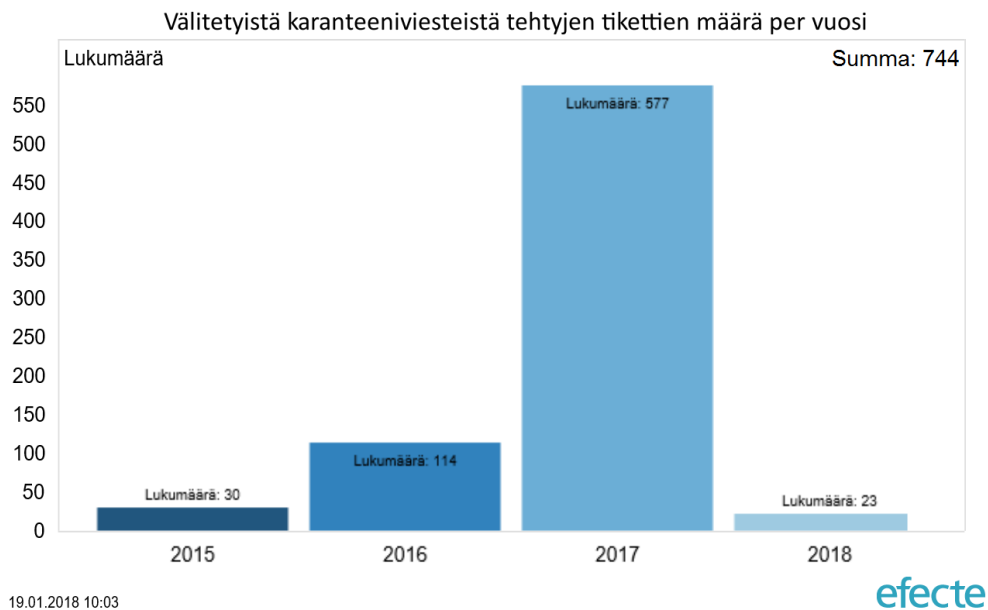


Kuva 6. Karanteenitickettien käsittelyrobotin aktiviteettikaavio, joka kuvaa robotin toiminnan käynnistyksestä loppuun.

Aktiviteettikaaviosta (kuva 6) näkee robotin toiminnan kaksi päälohkoa: käsittelemättömän karanteeniticketin käsittelyn ja karanteenin vapautuksen jälkeinen ticketin sulkeminen. Molemmassa logiikka on aika suoraviivaista ja toiminta haarautuu sen mukaan, onko valitussa näkyvässä tikettejä näkyvillä. Toinen haarautuminen tapahtuu ticketin käsittelyn alussa ticketin sisällön mukaan. Logiikan haarautuminen tehtiin avainsanalla "Run Keyword If", joka suorittaa parametrina sille annetun avainsanan, jos määritelty ehto täyttyy.

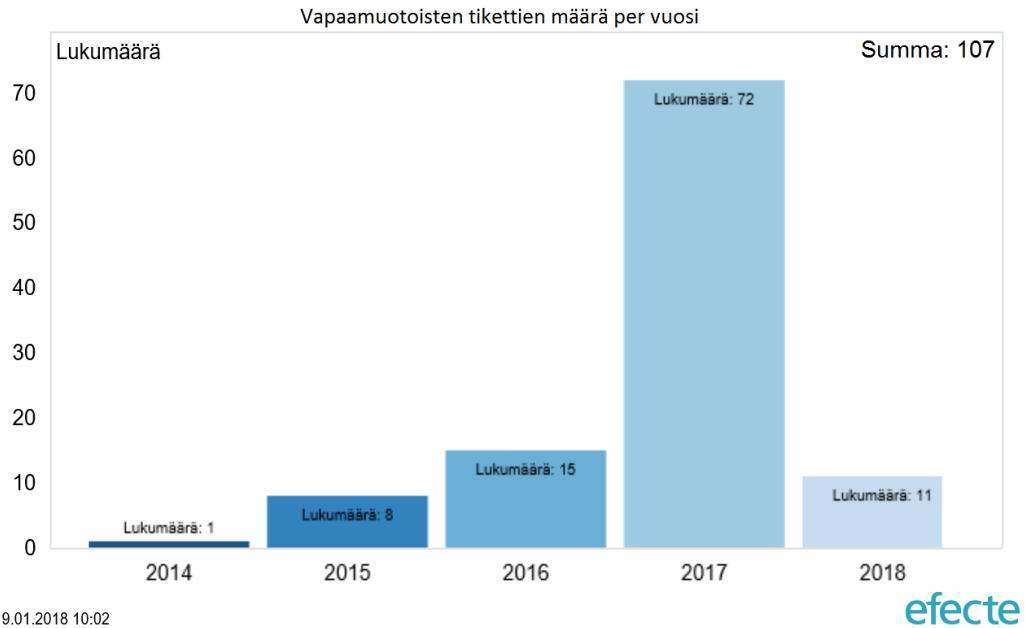
3.2 Kokeen arviointi

Efecten omia tilastointityökaluja käyttäen luotiin kaksi pylväsdiagrammia. Ensimmäinen (kuva 7) sisältää oikein välitettyjen tikkettien kokonaismäärän per vuosi. Toinen (kuva 8) sisältää vapaamuotoisesti luodut karanteeniticketit per vuosi hakusanalla ”*karanteeni”. Tähtimerkki ennen hakusanaa tarkoittaa, että hakuun otetaan mukaan kaikki hakusanaa edeltävät merkit.



Kuva 7. Oikein välitetyistä karanteeniviesteistä luotujen tikkettien lukumäärä per vuosi. Luotu Efecten tilastointityökalulla ja suodattamalla kaikkien olemassa olevien tikkettien otsikoista merkkijono ”*mobiililaittasi”.

PRH siirtyi organisaationa käyttämään pääasiallisina työpuheliminaan kännyköitä 1.10.2017, jolloin viimeisetkin lankapuhelimet kerättiin pois käytöstä. Muutos ei toki tapahtunut kokonaan yhdessä päivässä vaan vuodesta 2015 alkaen hiljattain kiihtyvään tahtiin, mikä näkyy karanteenitikkettien määrän kasvuna. Vertailun ja tikkettien kokonaismäärän tarkemmaksi arvion saamiseksi haettiin vastaavat tiedot myös vapaamuotoisten tikkettien lukumääristä. Kuvassa 8 on diagrammi vapaamuotoisten karanteenitikkettien määrästä per vuosi.



Kuva 8. Vapaamuotoisten karanteenitikkettien lukumäärä per vuosi. Luotu Efecten tilastointityökalulla ja suodattamalla kaikkien olemassa olevien tikkettien otsikoista merkkijono ”*karanteeni”.

Vuoden 2014 yksittäinen vapaamuotoinen karanteenitiketti selittyy tilastovirheenä. Kyseinen tiketti löytyy otsikolla ”Työasema viruskaranteenissa”, mikä täyttää hakukriteerit sisältämällä ”karanteeni”-merkkijonon otsikossaan. Moisia virheellisesti karanteenitikeiksi tulkittavia tapauksia on arvioiden mukaan noin yksi sadasta, mikä on hyväksyttävän vähän verrattuna vapaamuotoisten tikkettien määrään. Jos vuoden 2017 karanteenitikkettien määrä vastaa suunnilleen seuraavien vuosien määräluokkaa, niin vuodessa tulkittaisiin vapaamuotoisista karanteenitikeistä noin yksi kappale väärin. Vuoden 2018 alussa vapaamuotoisten karanteenitikkettien määrää paisuttaa robotin toiminnan testausta varten luodut tikit. Robotin testausta varten luotiin viisi erillistä testitikettiä.

Koska tilastoa karanteenitikkettien määrästä ei ole pidemmältä ajalta koko henkilöstön kännyköiden käyttöönoton jälkeen, niin on vaikea arvioida keskiarvoa vuosittaiselle karanteenitikkettien määrälle. Voisi kumminkin spekuloida, ettei karanteenitikkettien lukumäärä ylittäisi vuoden 2017 yhteismäärää, joka on vapaamuotoisten ja oikein luotujen karanteenitikkettien summa siltä vuodelta $72 + 577 = 649$. Vuoden 2017 lukumäärää paisuttaa uusien kännyköiden käyttöönotossa tapahtuneet karanteenivapautukset, joten voisi olettaa vuosittaisen määrän olevan ehkä sata tai kaksisataa tikkettä vähemmän kuin kännyköiden käyttöönottovuonna. PRH:lla on henkilöstöpalvelun mukaan 398 aktiivista palvelussuhdetta ja ylipäänsä virassa olevia henkilöitä on 406.

3.2.1 Hyödyt

Karanteeniticketin keskimääräinen aktiivinen käsittelyaika käsittelemättömästä suljetuksi on PRH:n tukihenkilöiden arvioiden mukaan kaksi-kolme minuuttia. Sama aktiivinen työaika robotin suorituksella on keskimääräksi laskettu suorituslokien perusteella 40 sekuntia. Jos vuodessa luodaan 600 karanteenitickettiä, niin robotti säästää kaksi minuuttia kerrottuna tikettien määrällä ja kolme minuuttia kerrottuna tikettien määrällä välisen aktiivisen työajan. Molemmat jaetaan 60:lla, jotta lopputuloksena on helpommin tulkittavat työtunnit.

$$\frac{\text{työajan käyttö (minuuttia)} * \text{tikettien määrä vuodessa}}{60 \text{ (minuuttia)}} = \text{työtuntia} / \text{vuosi}$$

Lasketaan arvion mukainen aktiivisen työajan maksimaalinen ja minimaalinen säästö:

$$\text{minimisäästöt} = \frac{2 \text{ (minuuttia)} * 600}{60 \text{ (minuuttia)}} = 20 \text{ työtuntia} / \text{vuosi}$$

$$\text{maksimisäästöt} = \frac{3 \text{ (minuuttia)} * 600}{60 \text{ (minuuttia)}} = 30 \text{ työtuntia} / \text{vuosi}$$

Karanteenitickettien automatisointi siis säästää vuodessa 20–30 aktiivista palkallista työtuntia. Toisaalta todellista työaikasäästöä on vaikea arvioida, koska kiireellisluontoisten karanteenitickettien käsittely keskeyttää usein muiden prosessien käsittelyn tukihenkilöiltä mahdollisesti aiheuttaen ajatusvirheitä tai -katkoja. Jos arvioidaan karanteeniticketin normaalin käsittelyn lisäksi aiheuttavan kahden minuutin palautumisviiveen muihin asioihin joka kerta, niin säästettyä työaika vuodelta kertyy 20 tuntia lisää, jolloin kokonaismäärä työaikasäästöistä olisi 40–50 tuntia.

Robotin suunnitteluun ja ohjelmointiin kului arvion mukaan noin kaksi kokonaista työpäivää. Arvioon ei ole laskettu mukaan uuden ohjelmointiympäristön tutustumiseen kulunutta aikaa. Yhteensä koko robotin suunnitteluun, ohjelmointiin, käyttöönottoon ja ohjelmointiympäristöön tutustumiseen kului hieman yli kaksi viikkoa. Robot Frameworkin asiantuntijalle samanlaisen robotin kehitys kestäisi varmaan alle yhden työpäivän. Jos oletetaan asiantuntijan kehittävän robotin karanteenitickettejä samassa ympäristössä, niin

robotti maksaisi itsensä takaisin noin kahdessa-kolmessa kuukaudessa, mutta vuosittaiset tuotot jäävät näillä arvioilla alle 1 000 euron, vaikka tukihenkilön tuntipalkaksi olettaisi 17 €.

Keskimääräinen odotusaika karanteenitketin luonnista sen aukaistavaksi käsittelyyn on 10–15 minuuttia. Sama odotusaika on robotin käyttöönoton jälkeen enintään viisi minuuttia yksittäiselle tiketille. Robotti käynnistyy aina viiden minuutin välein tasatunnista alkaen ja käsittelee yhden käsittelemättömän karanteenitketin odottamaan Tiedolta karanteenin vapautusta, jonka jälkeen se käsittelee yhden Tiedon ratkaiseman tiketin. Jos kokonaissäästöjen mukaan lasketaan myös tiketin luoja odottamisaika, joka on arviolta 10 minuuttia vähemmän per tiketti robotin käsittelyssä, niin lisäsäästöjä kertyy vielä 100 työtuntia.

Arvioitu bruttosäästö työtunneissa on yhteensä:

$$\text{minimisäästöt} = \frac{(2 + 10 + 2 (\text{minuuttia})) * 600}{60 (\text{minuuttia})} = 140 \text{ työtuntia/vuosi}$$



$$\text{maksimisäästöt} = \frac{(3 + 10 + 2 (\text{minuuttia})) * 600}{60 \text{ minuuttia}} = 150 \text{ työtuntia/vuosi}$$

Tulokseksi saatiin 140–150 työtuntia. Jos työtunnit muutetaan tilastokeskuksen ilmoittaman vuoden 2014 keskipalkan (12) mukaan rahaksi, niin vuosittaissäästöiksi saadaan 140 h * 16,4 €...–150 * 16,4 €, eli 2 296–2 460 €. PRH:n kokoiselle virastolle, jossa tuotot mitataan miljoonissa, on kyseisen tapauksen mahdollistamat säästöt hyvin pieniä.

EERAn tekemän selvityksen perusteella tärkeitä ohjelmistorobotiikan hyötyjä ovat myös prosessin tehokkuus ja nopeus, laadun parantuminen sekä henkilöstötyytyväisyys (21). Näitä robotti parantaa käsittelynopeudellaan ja metodisella käsittelylogiikallaan. Kaikki tiketit käsitellään aina saman logiikan mukaan ja niiden käsittelyn jättäminen robotille parantaa henkilöstötyytyväisyyttä sekä helpdeskillä että tikettipalvelua käyttävillä työntekijöillä.

3.2.2 Heikkoudet

Robotin suurin tekninen heikkous on sen epävalmius tikettitulvan käsittelyyn. Robotin tikettien käsittelynopeus on yksi tiketti per viisi minuuttia. Jos karanteenitikettejä tulee käsiteltäväksi enemmän kuin yksi viidessä minuutissa, niin tikettien odotusaika pitenee. Kesäkuun 6. päivä vuonna 2017 oli yksi kahdesta päivästä, jolloin karanteenitikettien määrä per päivä on ollut suurimmillaan, eli 14. Kyseisenä ruuhkapäivänä luotiin pahimmillaan kolme karanteenitikettiä peräkkäin saman minuutin aikana. Jos robotti olisi käsitellyt ruuhkapäivän tiketit, niin uusin niistä kolmesta peräkkäin luodusta tiketistä joutuisi odottaa käsittelyn aukaisua pahimmillaan $4 * 5$ minuuttia, eli 20 minuuttia. Seuraava kuva (kuva 9) sisältää kyseisen ruuhkapäivän kaikki tiketit lisätietoineen.

	NIMI 	LUONTIAIKA  ↓
1	ID: 760524 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 15:45
2	ID: 760525 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 15:45
3	ID: 760526 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 15:45
4	ID: 760523 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 15:35
5	ID: 760514 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 15:10
6	ID: 760515 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 15:10
7	ID: 760509 Työpuhelin karanteenissa	07.06.2017 14:55
8	ID: 760505 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 14:50
9	ID: 760506 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 14:50
10	ID: 760498 Kännyn karanteeni	07.06.2017 14:35
11	ID: 760436 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 11:05
12	ID: 760411 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 09:55
13	ID: 760384 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 08:20
14	ID: 760381 VL: Mobiililaitettasi on tilapäisesti estetty synl	07.06.2017 08:15

Kuva 9. Kuvakaappaus karanteenitiketeistä, jotka on luotu 7.6.2017. Kyseisenä päivänä luotiin ja käsiteltiin 14 karanteenitikettiä, mikä on tähän asti suurin määrä karanteenitikettejä päivässä. Tikettien omistajat ja käsittelijät on piilotettu näkymästä.

Toinen yhtä ruuhkainen päivä oli 11.9.2017, mutta sinä päivänä on luotu pahimmillaan vain kaksi tikettiä saman minuutin aikana. Päiviä, jolloin karanteenitikettien määrä on ollut 10 tai yli on ollut tähän asti 12 kappaletta. Kyseisten päivien määrä on pieni, vaikka kaikki tapahtuivat samana vuonna. Ruuhkapäivien karanteenitikettien luontiaikoja ana-

lysoidessa huomattiin, ettei tikettien luontimäärä per minuutti juuri ylitä kolmea. Jos oletetaan, ettei se ylitä kolmea vähemmän ruuhkaisina päivinäkään, niin robotin tiketin käsittelyaika ei veny yli 20 minuutin. Ruuhkapäiviä pitäisi tulla muutenkin vähemmän tulevaisuudessa, olettaen että ruuhkapäivät johtuivat uusien kännyköiden käyttöönotosta.

Projektin uudelleenkäytettävyyden taso on melko matala ja valtaosa siihen luoduista avainsanoista toimii vain selaimen kautta Efecten palvelun käytössä. Jos projekti olisi toteutettu UiPathilla tai Blue Prismillä, olisi mahdollista uudelleen käyttää siihen käytettyjä objekteja tai prosessinkulkua.

Vaikka työssä pyrittiinkin ottamaan huomioon tarpeen mukainen joustavuus verkkoelementtien löytämisessä, robotin toimivuutta ei voida taata selainpäivityksen tai Efecten palvelun päivityksen jälkeen. Selainpäivitys voi aiheuttaa yhteensopivuusongelmia verkkoajurin kanssa ja estää robotin toiminnan täysin ennen kuin verkkoajurin päivitetään uuden selaimen kanssa yhteensopivaksi. Efecten tikettien hallintapalvelu on ilmeisesti rakennettu automaattisesti AngularJS-elementeistä ja rikkoo hieman hyvien verkkosivujen nimeämiskäytäntöä jättämällä joitakin elementtejä sitomatta omiin tunnisteisiin. Se aiheutti hankaluuksia oikeiden elementtien etsinnässä ja saattaa vaikuttaa odottamattomalla tavalla Efecten palvelun päivityksen jälkeiseen robotin toimintaan.

Automaation kohde ei myöskään täytä yleisiä ohjelmistorobotiikan vaatimuksia pienellä ajan ja rahan säästöillä. Jos karanteenitickettien käsittelyn automatisoinnin tarkoitus olisi ollut pelkästään säästää resursseja, niin toteutusta ei olisi kannattanut tehdä. Automaatiotapauksen tarkoitus oli vaivan säästön lisäksi tarjota yksinkertainen lähtökohta ohjelmistorobotiikan käyttöön ja siinä se onnistui. Ja vaikka suorat rahalliset hyödyt olivat marginaalisia, PRH:lla on nyt käytössään hyvä pohja muidenkin tikettien automatisointiin. Tästä on hyvä aloittaa vaativampiin automaatioprojekteihin.

3.3 Jatkokehitys

On mahdollista, että iso osa työkännyköistä joutuu karanteeniin jonkun ison ohjelmistopäivityksen vuoksi. Suuressa osassa kännyköitä on sama käyttöjärjestelmä ja muu ohjelmisto, jolloin ohjelmistopäivitys vaikuttaisi kaikkiin näihin kännyköihin. Jos tällainen kauhuskenaario tapahtuisi, niin karanteenitickettejä luotaisiin nopeassa ajassa mahdollisesti yli 300 kappaletta. Robotilla kestäisi jonon purkamiseksi $300 * 5 = 1\ 500$ minuuttia,

eli noin vuorokausi. Tuossa niin sanotussa painajaiskenaariossa on ymmärrettävää, että käsittely kestää kauan, koska se kestäisi kauan ihmiseltäkin, mutta käsittelyaika voisi olla robotilta paljon nopeampikin. Jatkokehitysideana voisi päivittää robotin logiikan niin, että robotti käsittelee avoimia tikettejä niin pitkään kuin niitä löytyy avoimien karanteenitikkettien näkymästään.

Tikettijärjestelmään tulee myös lähes päivittäin sähköpostin salasanan resetointi-tikettejä, joiden käsittelyn automatisointi säästäisi aikaa ja vaivaa tukihenkilöiltä sekä mahdollisesti sujuvoittaisi asiakkaan uuden salasanan saannin. Suurimpana haasteena siinä varmaan tulisi oikeiden tikkettien tunnistaminen, koska tiketit tulevat aika vapaassa muodossa asiakkailta. Työnkulku olisi kuitenkin yleensä hyvin samanlainen ja karanteenitikkettejä käsittelevästä robotista saisi hyvän pohjan tikkettien käsittelyä varten. Sama pätee mihin tahansa muunkinlaiseen tikkettien käsittelyn automatisointiin. Karanteenitikkettejä käsittelevä robotti on ohjelmoitu niin uudelleenkäytettäväksi kuin se on Robot Frameworkilla mahdollista.

4 UiPath ohjelmistorobotiikan työkaluna

Yhtiönä UiPath aloitti uransa New Yorkissa vuonna 2005 kehittää ohjelmistorobotiikan ohjelmistoa, jonka tarkoitus on olla turvallinen, helppokäyttöinen ja tehokas. Se on vasta vuoden 2015 jälkeen saanut merkittävämpää osaa RPA-markkinoilla. Siitä johtuen sen tuotteiden käyttäjiltä koottuja referenssitietoja ei ole paljon. Mainitsemisen arvoista on, että National Aeronautics and Space Administration NASA on aloittanut ohjelmiston käytön Shared Services Center -siivessään, mikä tarkoittaa, että UiPathin ohjelmisto täyttää NASA:n turvallisuusvaatimukset. Merkittäviä yhteistyökumppaneita UiPathilla on muun muassa Accenture Oy, Oracle Corporation, Atos ja ABBYY, jotka ovat auttaneet yhtiön nousussa digitaalitekniikan markkinoilla. (22.)

4.1 UiPathin esittely

UiPath on ohjelmistorobotiikkaan kehitetty ohjelmistopaketti, joka koostuu kolmesta osasta: UiPath Studiosta, UiPath Orchestratorista ja UiPath Robotista.

UiPath Studio on ohjelmistorobottien varsinainen ohjelmointiympäristö, missä määritellään automatisoitava työ käyttäen prosessijohtamiseen pohjautuvia työnkulkua kuvaavia kaavioita. Kaaviot voi luoda manuaalisesti askel kerrallaan tai käyttäen UiPath Studion sisäänrakennettuja työnkulun tallennustyökaluja, missä ohjelmisto tallentaa tietokoneella tehdyt toiminnot ja tallentaa ne toistettavana sekvenssinä. Työsekvenssin tallennukselle on eri tilat yksinkertaisen toiminnon suorittamiseen tai yhden sovelluksen käyttöön, työpöytätoimintojen-, verkkosovellusten- ja virtuaaliympäristön käytön tallennukseen. UiPath Studioon kuuluu sisäänrakennettuna käyttäjätoimintoja yleisien käyttöliittymien hallintaan sekä Excelin ja SAP-finanssiprosessien hallintaohjelman käyttöön. Näitä sisäänrakennettuja toimintoja kutsutaan Studiossa aktiviteeteiksi. (24.)

UiPath Orchestrator on palvelimella toimiva ohjelmistorobottien hallinnointityökalu, jolla annetaan roboteille käskyt, ohjataan keskitetysti niiden toimintaa, sekä monitoroidaan suorituksia ja lokeja. Se on skaalautuvuusratkaisu, joka helpottaa suurempien robottimäärien hallinnointia ja seuranta. Orchestrator tarjoaa myös tietoturvallisen paikan säilyttää robottien käyttämää arkaluontoista dataa, kuten asiakastietoja tai käyttäjätunnustietoja.

UiPath Robot on UiPathilla kehitetty ohjelmistorobotti, jonka sisäänrakennettuihin ominaisuuksiin kuuluu Googlen, Microsoftin ja ABBYY:n kehittämät OCR-teknologiat. Sen avulla roboteilla voi automatisoida prosesseja paikallisen automatisoinnin lisäksi virtuaaliympäristössä. Virtuaaliympäristön käyttö on roboteille haaste, koska sitä käyttävä käyttöliittymä simuloi käytettävän virtuaaliympäristön käytännössä videokuvana, minkä ohjaaminen ei onnistu ilman käytettävien asioiden sijaintien kovakoodausta tai jonkin sortin kuvan koneellista tulkintaa.

Robotin jokaisesta suorituksesta ja UiPath Studion toiminnasta tehdään lokitiedostot oletuksena käyttäjän AppData/Local/UiPath/Logs-kansioon LOG-tiedostomuodossa. Kaikki virheet ja varoitukset merkataan myös Windowsin omaan tapahtumalokiin. Studion lokitus perustuu NLogiin, joka on avoimen lähdekoodin .NET-pohjainen lokisyntaksi. Lokiin pystyy vaikuttaa muokkaamalla NLog-konfiguraatitiedostoa tai käyttämällä Studion sisäänrakennettuja aktiviteetteja Log Message ja WriteLine. WriteLine-aktiviteetin lokitaso on Trace-tasolla ja Log Message-aktiviteetin lokitason voi valita. Lokitasot epätarkimmasta tarkimpaan (taulukko 2): Off, Critical, Error, Warning, Information, Trace, Verbose.

Taulukko 2. UiPath Studion lokitasot epätarkimmasta tarkimpaan. Mitä alempi lokitaso, sitä tarkemmin tiedot kirjataan lokiin. Taulukko on otettu mukailen UiPathin dokumentaatiosta (24).

Lokitaso	Oletuksena lokiin päätyy
Off	Ei mitään. Loki on pois päältä.
Critical	Kaikki tieto, minkä taso on Critical tai korkeampi. Käytössä kriittisten virheiden lokitukseen.
Error	Kaikki tieto, minkä taso on Error tai korkeampi. Käytössä virheiden raportointiin.
Warning	Kaikki tieto, minkä taso on Warning tai korkeampi. Käytössä erilaisten varoitusten lokitukseen.
Information	Kaikki tieto, minkä taso on Information tai korkeampi. WriteLine-aktiviteetti raportoidaan tällä tasolla ja Log Message, jos se on asetettu Trace-tasoa yleemmäksi.
Trace	Kaikki tieto, minkä taso on Trace tai korkeampi. Aktiviteeteista ja muuttujista on tietoa tällä tasolla.
Verbose	Kaikki tieto, minkä taso on Verbose tai korkeampi. Tarkemmat tiedot aktiviteettien aloituksista sekä lopetuksista ja kaikki tiedot käytetyistä argumenteista sekä muuttujista.

Suorituksen aloitus ja lopetus, robotin ja Orchestratorin väliset transaktioiden aloitukset ja lopetukset sekä aktiviteettien suoritustiedot ovat oletuksena lokin alimmalla Verbose-tasolla. Lokin minimitaso on suorituslokeissa ja Studion omassa lokissa Trace-tasolla. Lokitapahtumat siirtyvät myös Orchestratoriin, jossa niitä voi selata robottikohtaisesti ja voi tuoda taulukkona .csv-tiedostoksi.

4.2 Esivaatimukset

UiPathin toiminnan takaamiseen sille on ohjelmisto- ja laitteistovaatimuksista, jotka on spesioitu UiPathin verkkosivuilla. Kuva 10 on kuvakaappaus virallisista ohjelmistovaatimuksista.

Important!

Please note that Studio is not compatible with machines that have the KB4055002 update on Windows 7 SP1 and Windows Server 2008 R2 SP1, and stops working. If you have this operating system version, we recommend you manually uninstall this .Net framework update and install the next version, [KB 4055532](#).

Supported Versions	
Operating System	<ul style="list-style-type: none"> Windows 7 (update KB2533623) Windows 10 Windows Server 2008 R2 Windows Server 2012 R2 Windows Server 2016
.NET Framework	<ul style="list-style-type: none"> minimum: 4.5.2 recommended: 4.6.1

Kuva 10. Kuvakaappaus UiPathin sivuilla sijaitsevista ohjelmistovaatimuksista. Huomiona myös epäsopivuus Windows 7 KB4055002-päivitysversion kanssa.

Toisin kuin Robot Framework, UiPath tukee vain Windows-käyttöjärjestelmiä Windows 7:n päivityksestä KB2533623 ylöspäin. Lisävaatimuksena tarvitaan vähintään .NET Frameworkin versio 4.5.2 asennettuna.

Suorituskykyä täytyy koneelta myös löytyä kohtuullisesti UiPath-ohjelmiston pyörittämiseen. Kuvassa 11 on kuvakaappaus virallisista laitevaatimuksista.

	Minimum	Recommended
CPU	1.4GHz 32-bit (x86)	Dual Core 1.8GHz 64-bit
RAM	4 GB	4 GB

Kuva 11. Kuvakaappaus UiPathin sivuilla sijaitsevista laitteistovaatimuksista, jotka ovat aika matalat nykyajan standardien mukaan.

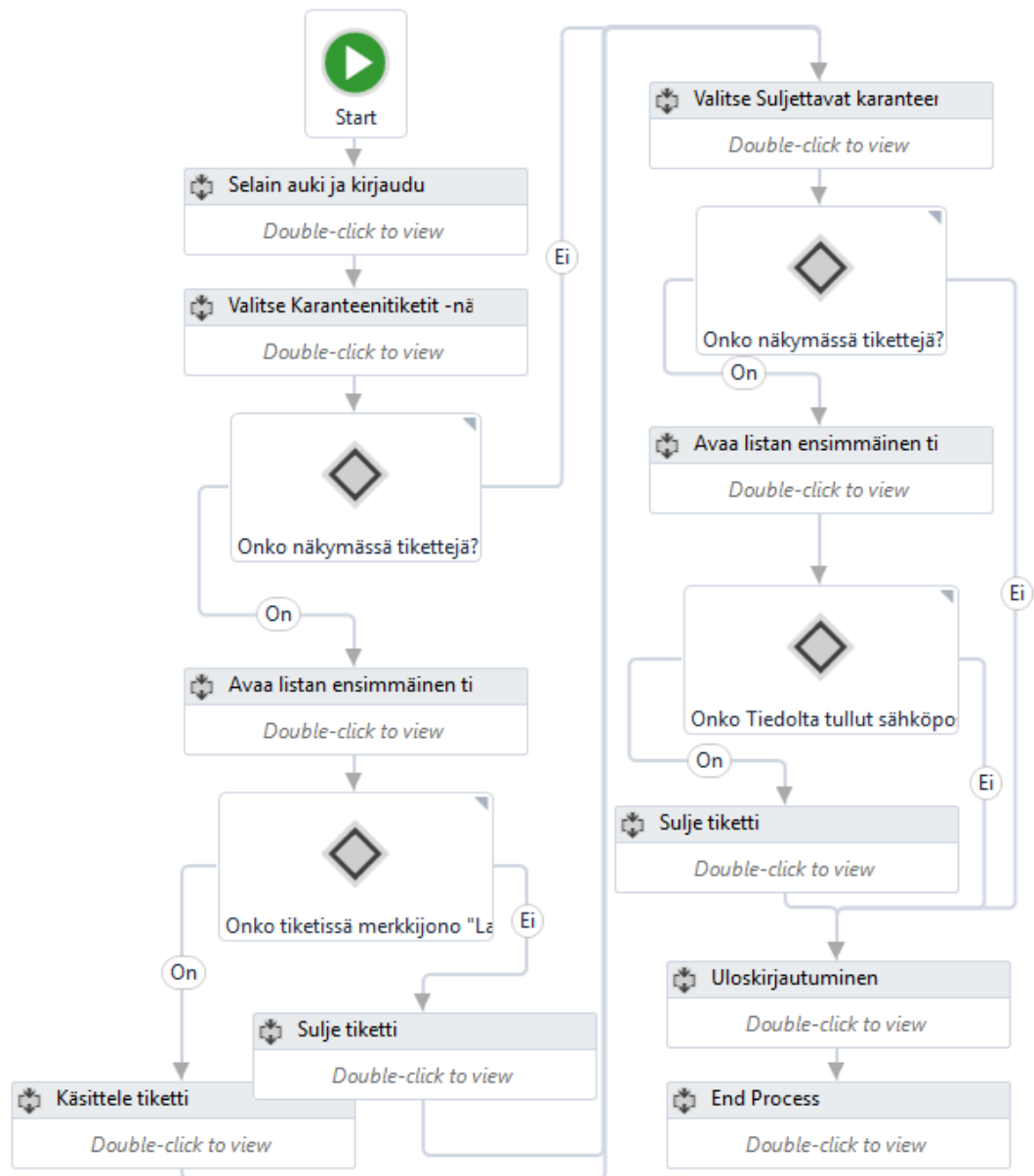
Käytännössä kaikki tietokoneet PRH:lla täyttävät sekä laite- että ohjelmistovaatimukset paria erikoistapausta lukuun ottamatta, niin yhteensopivuuksia ei tarvinnut paljon pohtia. Tarkistettavaksi jäi vain oikean .NET Framework -versio, joka on helppo asentaa, jos se puuttuu. UiPath Studion asennus oli helppo ja nopea operaatio verrattuna Robot Frameworkiin. Tarvittavat verkkoajurit sai myös asennettua kehitystyökalun sisällä Setup-välilehdeltä Setup Extensions -valikosta.

5 Mobiilikaranteenitickettien käsittelyn automatisointi

Karanteenitickettien käsittely -tapaus vaatimuksineen toistettiin UiPath-ohjelmistolla. Tarkoituksena verrata Robot Frameworkilla kehitettyä robottia UiPath-robottiin. Sama automaation hyödyntämiskohde valittiin, jotta performanssierot on helpompi mitata ja aikaa uuden hyödyntämiskohteen löytämiseen ja siihen tutustumiseen ei ollut.

5.1 Toteutus

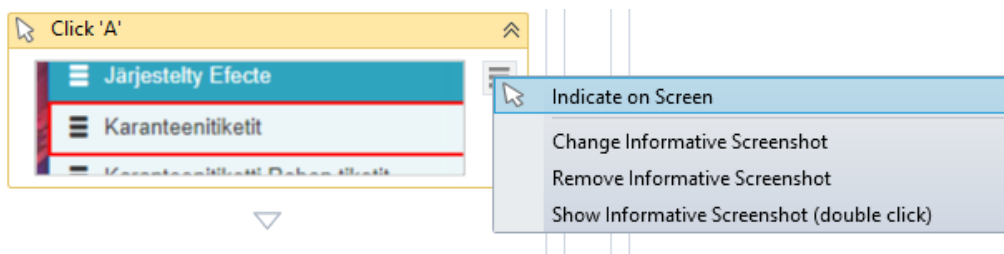
Robottia lähdettiin kehittämään samalla aktiviteettikaaviolla (kuva 6) kuin Robot Frameworkilla kootun robotin toimintalogiikkaa. Lähtökohta oli hyvä, sillä aktiviteettikaavion pohjalta sai hyvän luurangon robotin toiminnalle, se oli helppo luoda UiPath Studion kaavioidin pohjautuvalla logiikan rakennustyökaluilla. Kuva 12 sisältää UiPath Studiossa luodun aktiviteettikaaviota pohjana käytetyn tickettien käsittelylogiikan.



Kuva 12. Kuvakaappaus UiPath Studiosta luodusta aktiviteettikaaviosta. Jokaisen tehtävän pystyy avata ja muokata sen sisältämiä tapahtumia. Esimerkiksi "Selain auki ja kirjaudu" sisältää toiminnot selaimen aukaisemiselle Efecten verkkosivuille sekä verkkojärjestelmään kirjautumista varten tarvittavien elementtien klikkailut.

Toimintojen ohjelmointi onnistui joko nauhoittamalla esimerkkitoimintaa toistamalla robotille halutut toiminnot itse tai valitsemalla toiminnot Studion omista aktiviteeteistä ja huolehtimalla oikeiden elementtien löytämisestä sekä toiminnon suoritusjärjestyksestä. Vaikka toimintojen nauhoittaminen oli kätevää, jouduttiin aika usein puuttumaan sekvenssiin vaihtamalla jotain elementtien tunnistusominaisuuksia. Efecten tikettijärjestelmän elementtien tunnistusten puutteet aiheuttivat tämänkin toteutuksen aikana ongelmia

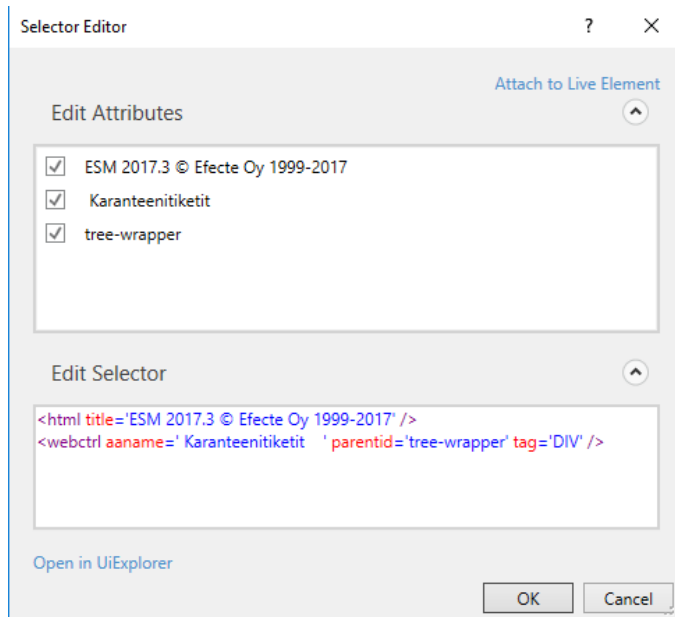
varsinkin tikettien pudotusvalikkojen kanssa. UiPath Studiolla pystyi näyttämään toiminnolle elementin myös suoraan valitsemalla toiminnolle ”Indicate on Screen” -valinnan (kuva 13). Tässä tapauksessa haluttu käyttöliittymän osa valitaan klikkaamalla sitä ruudulta.



Kuva 13. Kuvakaappaus UiPath Studion sisäänrakennetusta ”Indicate on Screen” –valinnasta, jolla valitaan robotin toiminnon kohde-elementti klikkaamalla sitä manuaalisesti itse. Kuvan toiminto on klikkaa elementtiä Karanteenitikit-näkymä.

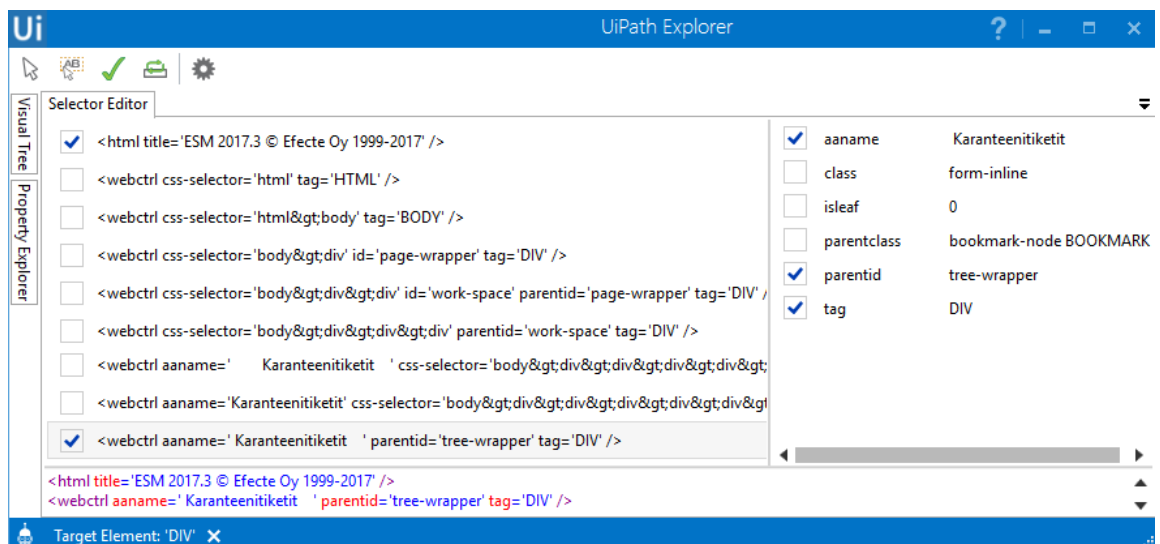
Toiminto nimetään automaattisesti kohde-elementin mukaan, mutta sitä voi vapaasti muuttaa luettavuuden parantamiseksi. Kuvan 13 toiminto nimellä ”Click 'A'” ei kuvaa toimintoa tarpeeksi hyvin ja ilman havainnollistavaa kuvakaappausta sen sisältönä, olisi vaikea päätellä, mitä verkkoelementtiä oikein klikataan. Valitusta elementistä otetaan valinnan yhteydessä pieni kuvaileva kuvakaappaus, joka on kuvassa 13 ympyröity Karanteenitikit-näkymä Efecte-tikettijärjestelmässä. Hyvänä käytäntönä kannattaa nimetä toimintojen nimiin kohde-elementit kuvaavilla nimillä. Kuvan tapauksessa toiminto uudelleennimettiin ”Click 'Karanteenitikit - näkymää'”, joka kuvaa toiminnon tehtävää paremmin ja säilyttää alkuperäisen aktiviteetin nimen sellaisenaan.

Elementtien löytämistä varten Studio pitää kirjata omassa UiExplorerissa kaikista tietokoneella auki olevista ikkunoista ja niiden sisältämistä elementeistä sekä niiden ominaisuuksista. Se mahdollistaa elementtien tunnistamisen klikkauksella. Kun elementtiä klikkaa ”Indicate on Screen” -toiminnolla, se tunnistetaan Explorerissa ja sille valitaan automaattisesti yksilöivät ominaisuudet. Valitsijaan ne tallennetaan XML-tekstinä. Joskus automaattisesti valitut ominaisuudet eivät ole tarpeeksi yksilöiviä, joten valintoja pääsee muokkaamaan Explorerissa kuvan 14 mukaisesti.



Kuva 14. Kuvakaappauksessa muokattavana on tikkijärjestelmä-verkkosivun Karanteeniticketit-näkymän valintaelementti. Valitsijaan on valittu verkkosivun ikkunan otsikko ”ESM 2017.3 @ Efecte Oy 1999-2017”, elementin tekstisisältö ”Karanteeniticketit”, elementin yläelementin id ”tree-wrapper” ja elementin HTML tagi ”DIV”.

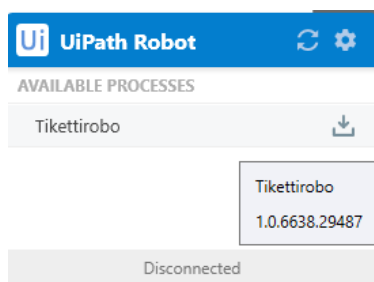
Elementtejä pääsee valikoimaan tarkemmin valitsemalla ”Open in UiExplorer” -valinnan, jolloin elementti pitää valita reaaliaikaista tunnistamista varten ruudulta klikkaamalla. Explorerista näkee kaikki tunnisteet ja sieltä voi valita, mitkä niistä valitaan hakukriteereiksi valitsijalle.



Kuva 15. Kuvakaappaus UiPathin elementtistauksesta, johon on valittu Karanteeniticketit-näkymän valintaelementti. Valittuina yksilöivinä ominaisuuksina on aaname, parentid ja DIV. Niiden lisäksi UiExplorer tunnistaa elementin class HTML-tagin, isleaf XML-attribuutin ja parentclass yläelementin luokan.

Elementin tunnistavan valitsijan pystyy tekemään kokonaan manuaalisestikin UiExplorerin kautta, mutta se on yleensä työläämpää. Välillä siltä ei voi välttyä, koska valitsijan automaattinen generointi ei aina ole täydellinen. Esimerkiksi tikettiä muokatessa verkkosivulla on useita samanlaisia pudotusvalikkoelementtejä, joiden erottaminen toisistaan on haaste yksilöllistävien tunnisteiden puutteen vuoksi. Koska yksilöiviä tunnisteita ei ole tai ne muuttuvat eri käyttökerroilla, piti elementit löytää sijainnin tai absoluuttisen Cascading Style Sheets, lyhyesti CSS-polun perusteella.

Robotin pystyy käynnistää komentoriviltä, UiPath Studion Run-valinnalla ja UiPath Orchestratorilla. Robotin käynnistämiseen Orchestratorilla täytyy luoda yhteys tietokoneeseen, jolla robotti sijaitsee ja Orchestratorin verkkosivun välillä. Se luodaan antamalla Orchestratorista robotille yksilöity avain ja Orchestratorille koneen nimi, jolla robotti sijaitsee. Seuraavassa kuvassa (kuva 16) on kuvakaappaus julkaistusta tikettejä käsittelevästä robotista UiPath Robot -hallintatyökalussa. Työkalu täytyi Windows 10:ssä käynnistää manuaalisesti sijainnista %Userprofile%\AppData\Local\UiPath viimeisimmästä app-kansiosta ajamalla UiRobot.exe –tiedoston.



Kuva 16. Kuvakaappaus UiPath Robot –käyttöliittymästä, jonka kautta on tarkoitus yhdistää robotteja Orchestrator-palvelimelle. Robotti päätyy itsestään listalle, kun valmiin robotin julkaisee Studion puolelta. Kuvassa ei ole vielä luotu yhteyttä Orchestratorin ja robotin välille.

Community Edition sisältää rekisteröintioikeudet kahteen development-luokituksen robottiin Orchestrator-palvelun kautta (25). Robotin seuranta tapahtui UiPath Orchestratorin kautta, joka Community Editionin käyttöä varten sijaitsee osoitteessa <https://demo.uipath.com/>. Orchestratoriin kirjautuneena siirryttiin Robot-sivulle ja lisättiin karanteenitikettejä käsittelevä robotti järjestelmään. Sitä varten Orchestratorille annettiin tietokoneen nimi, käyttäjätunnus, jolla robotti ajetaan ja käyttäjätunnuksen salasana. Robotin yksilöity rekisteröintiavain otettiin robotin lisäsisikunasta (kuva 17) ja syötettiin UiPath Robotin asetuksiin yhteyden luomiseksi. Robotin tyyppiä valittiin development.

The image shows a configuration window for adding a robot in Orchestrator. The window is split into two tabs: 'CONFIGURATION' and 'RUNTIME'. The 'CONFIGURATION' tab is selected. The fields are as follows:

- Key ***: bf2db08d-b38e-424d-9a56-2746c545d94e
- Machine ***: DESKTOP-PMFQGCB
- Name ***: MirBot
- Domain\Username ***: DocMircea
- Password**:
- Type**: NonProduction
- Description**: (empty)

At the bottom of the configuration area, there is a checkbox labeled 'Create another'. Below the configuration area are two buttons: 'CANCEL' and 'PROVISION'.

Kuva 17. Orchestratorin ikkuna, josta lisätään järjestelmään robotti. Pakolliset kentät on merkitty '*'-merkillä. Kuva saatu suoraan UiPathin dokumentaatiosta (24).

Robotti tarvitsee käyttäjäoikeudet tietokoneeseen, jolla se suoritetaan, joten Orchestratoriin pitää syöttää käyttäjätunnustiedot. Salasana suojataan 256-bittisellä salausmenetelmällä ja säilötään Orchestratorin tietokannassa. On myös mahdollista säilyttää arkaluontoiset tiedot Orchestratorin tietokannan sijaan CyberArk Enterprise Password Vault -maksullisessa palvelussa.

5.2 Toteutusten vertailu

Karanteenitikkien automatisoinnin yleiset hyödyt on käyty läpi jo osassa 3.2 Kokeen arviointi, joten tässä osiossa vertaillaan vain Robot Frameworkilla ja UiPathilla tehtyjen robottien ominaisuuksia. Taulukkoon 3 on koottu ominaisuusvertailut UiPathista ja Robot Frameworkista.

Taulukko 3. Ominaisuusvertailu UiPathin ja Robot Frameworkin välillä.

Ominaisuudet	UiPath	Robot Framework
Versionhallinta	Orchestratorin kautta, kuuluu lisenssiin.	Puuttuu. Tarvitsee oman ratkaisun tai kolmannen osapuolen toteutuksen (esim. github tai sourceforge).

Robottien hallinnointi	Robotit ilmoittavat tilastaan Orchestratorille. Robotit voi käynnistää ja asettaa toimintaan etänä verkkopalvelun kautta.	Keskitettyä hallinnointityökalua ei ole, mutta mahdollisuudet semmoisen kehittämiseen ovat saatavilla.
Raportointi	Orchestratoriin yhdistettynä, robotit jakavat lokitiedot luettavaksi keskitettyyn palveluun. Lokien luettavuus parempaa vasta jäsentelyn jälkeen, mutta kelpollista ilman sitäkin.	Lokitiedostot hyvin luettavissa ja niihin päätyvä tieto on lähes täysin käyttäjän kontrolloitavissa. Jos raportit haluaa yhdistää tiettyihin suorituskertoihin, niin se onnistuu lähinnä muuttamalla lokitiedostojen nimeämiskäytäntöä. Lokitiedosto syntyy vasta robotin suorituksen päätyttyä.
Robotin keskeyttäminen	Mahdollista sekä Orchestratorin että Studion kautta ajaessa.	RIDEn kautta käynnistettynä se on mahdollista joissakin tapauksissa, mutta ei kovin luotettavaa. Muuten mahdollisuudet keskeyttää robotin toiminta ovat hyvin rajoitetut.
Robotin työkalut käyttöliittymien hallintaan	Sisäänrakennettuna laaja valikoima toimintoja erilaisten käyttöliittymien ohjaamiseen.	Robot Frameworkin sisäänrakennetut ominaisuudet erittäin rajalliset. Lähes kaikki toiminnallisuudet ulkoisten tahojen tekemiä.
Mukautettujen toimintojen luonti	Mahdollista C#-skriptin muodossa, joka pitää muuttaa NuGet-paketiksi ja asentaa UiPath Studiossa käyttöön. (Ei mahdollista Community Editionissa)	Tukee laajan kielivalikoiman skriptien suoritusta avainsanana. Itse tehdyn skriptin voi yleensä sellaisenaan tuoda nimellä Robot Framework –projektin käyttöön. Tuodut skriptit käännetään luettavaksi automaattisesti Robot Frameworkin omalla kääntäjällä.
Hinta	Lisenssistä joutuu maksamaan vuosittain. Hinta määritellään asiakaskohtaisesti.	Ilmainen, perustuu vapaaseen lähdekoodiin.
Tietojen suojaus	Orchestratorin käyttöön kuuluu tietopankki, minne voi säilöä arkaluontoista tietoa. Tieto salataan vahvalla, 256-bittisellä salausmenetelmällä.	Salasanojen piilotus lokitiedostoista mahdollista. Muuten puutteellista - turvallisen ratkaisun joutuu suunnittelemaan itse.
Käyttäjätuki/käyttäjäreferenssit	Dokumentaatio, esimerkkiprojekteja, video-ohjeistuksia ja tekninen tuki sähköpostin välityksellä. Käyttäjäreferenssejä ei vielä ole paljon.	Virallinen dokumentaatio ja joitakin esimerkkejä on vapaasti käytettävissä. Käyttäjäreferenssejä hyvin paljon, koska Robot Framework on suosittu testausautomaatiossa. Vähän referenssejä ohjelmistorobotiikan käytössä.
Ohjelmointiympäristön päivitykset	Onnistuu Studion kautta suoraan ja tarvittaessa automaattisesti.	Täytyy tehdä manuaalisesti komentokonsolin kautta.
Käyttöliittymän elementtien joustava tunnistaminen	UiExplorerilla pystyy valitsemaan tarvittaessa joustavammat elementit. Vaihtoehtona liian geneeristen elementtien paikantamiseen on myös ankkuroida jokin tunnistettava elementti ja paikantaa geneerinen elementti sen mukaan.	XPathilla pystyy sitomaan elementtejä muihin elementteihin, jos geneeristen elementtien löytämisessä on ongelmia.

Kummankaan RPA-vaihtoehdon elementtien tunnistaminen ei ole aukoton selviytymään käyttöliittymän muutoksista. Vaikka molemmilla on mahdollisuudet joustavaan elementtien löytöön, täytyy kummankin kanssa välillä tyytyä löytämään elementti absoluuttisen

sijainnin perusteella. Molemmilla on yhtä hyvät mahdollisuudet sen välttämiseksi. Vaikka elementin tunnistaisi hyvinkin joustavalla logiikalla, joutuu käyttöliittymäpäivitysten jälkeen varmistaa, että robotti pystyy vielä suorittamaan toimintonsa ongelmitta.

5.2.1 Hyödyt

Toimintojen haarautuminen oli ainakin helpompi visuaalisesti hahmottaa UiPath Studiolla, kuin Robot Frameworkin RIDEllä. Pelkästään robotin logiikan ohjelmointi aktiiviteettikaavion tasolla oli mahdollista ilman perus ohjelmoinnin ymmärrystä – siihen riitti prosessimallien tietämys. Prosessimallista oli muutenkin suoraviivaisempi aloittaa toimintosekvenssien ohjelmointi.

Orchestrator-palvelun ajastettujen toimintojen helppous, robottien seurannan keskittäminen ja versionhallinta kokonaisuutena on kieltämättä toimiva ratkaisu UiPathin ohjelmistolle. Robot Frameworkissa moinen ratkaisu pitäisi itse kehittää. Sähköposti-ilmoitusten lähetyksillä ja paikallisella lokien seurannalla voidaan joitakin robotteja seurata, mutta isomman robottiverkoston hallinnointiin se ei ole kestävä ratkaisu.

5.2.2 Heikkoudet

Vaikka UiPathin ohjelmiston ominaisuutena mainostetaan, ettei sen käyttö vaadi ohjelmointitaitoja, niistä oli ainakin huomattavaa hyötyä aktiviteettien toiminnan ymmärtämisessä ja robotin logiikan määrittelyssä. Itse tehtyjen aktiviteettien ohjelmoiminen ja varsinkin elementtien löytäminen ja tunnistaminen yksilöivillä ominaisuuksilla olisi varmasti haastavaa henkilölle, jolla ei ole kokemusta ohjelmoinnista. UiPath on ainakin lähempänä yleisen ohjelmistorobotiikan ohjelmointitaidon vaatimuksia kuin Robot Framework.

UiPathin generoimien lokitiedostojen selaaminen on hieman vaikeampaa, kuin Robot Frameworkin vastaavien. Jos niistä haluaa analysoida robotin prosessin kulkua, niin lokitasoksi pitää asettaa joku Information-tasoa alempi vaihtoehto, jonka seurauksena tietoa tulee lokiin jo niin paljon, että sen tulkintaa vaatii paljon aikaa tai oman jäsentelyn tiettyjä asioita haettaessa. Ongelmaa voi helpottaa käyttämällä WriteLine tai Log Message –aktiviteetteja.

Kaikkien UiPath-ohjelmistolla tehtyjen robottien takaisinmaksuaika pitenee, kun joutuu ottamaan huomioon UiPath-lisenssistä johtuvat kulut.

5.3 Jatkokehitys

Jos PRH päättää ostaa UiPath-ohjelmiston lisenssin, niin jatkossa kannattaa koota yhteen ohjelmistorobotiikan tiimi. Se voisi sisältää robottien ylläpitäjiä, kehittäjiä ja suunnittelijoita. Tietenkin kannattaa aloittaa pienellä, parin hengen ryhmällä ja laajentaa tarvittaessa. UiPath Orchestrator tarjoaa hyvät työkalut moisen tiimin toimintojen hallintaan ja seurantaan muokattavilla käyttäjärooleillaan. Administrator-rooli on ainoa, minkä käyttöoikeuksia ei voi muokata, mutta roolin omaava henkilö voi hallinnoida muiden roolien oikeuksia ja käyttäjämääriä.

6 LEI-laskujen käsittelyn automatisointi

Legal Entity Identifier -tunnuksen, eli LEI-tunnuksen tilanneiden PRH:n asiakkaiden määrä on noussut huimaa tahtia, eikä PRH:lla ole tarvittavia resursseja vastaamaan uusien tunnusten kysynnästä johtuvaa laskutusten selvitysten määrää. Yritys tai yhteisö tarvitsee LEI-tunnuksen tehdäkseen kauppoja pörssiosakkeilla ja sen saadakseen pitää täyttää hakemus ja lähettää se PRH:lle. Hakemus käsitellään PRH:lla ja lasku LEI-tunnuksesta lähetetään asiakkaan laskutusosoitteeseen. Laskujen käsittelyn nopeuttamiseksi toivottiin apua ohjelmistorobotiikalta. Koska PRH ei ollut tehnyt vielä päätöstä UiPath-ohjelmiston lisenssin hankinnasta, oli käytännössä ainoa vaihtoehto toteuttaa robotti Robot Frameworkilla.

Työn aloittamisen aikana jonottavien hakemusten määrä oli arvioiden mukaan vähän alle 10 000 ja niitä käsitteli pieni viiden hengen ryhmä muun työn ohella. Uusia hakemuksia tulee käsiteltäväksi pienryhmän mukaan keskimäärin 100–200 kappaletta päivässä. Hakemuksia käsitellään tuotteliaan työpäivän aikana noin 100 kappaletta, joten syytä tahdin kiristykselle tai lisäävulle ainakin löytyy. Uudet asiakkaat toimivat kuvan 18 ohjeen mukaan.



Miten teet hakemuksen?

1. Täytät hakemuksen tiedot netissä PRH:n LEI-palvelun ennakkojulkaisussa. Hakemuslomakkeessa on seuraavat vaiheet:

- Täytät emoyhteisöjen tiedot tai ilmoitat tietojen puuttumisen
- Täytät hakemuksen muut tiedot
- Vahvistat ja kerrot, kuka allekirjoittaa

Tutustu tarkemmin hakemuksen vaiheisiin klikkaamalla ylläolevia linkkejä.

2. Tulostat ja allekirjoitat hakemuksen

3. Lähetät hakemuksen mahdollisine liitteineen PRH:een

- nopeinta on lähettää skannattu hakemus sähköpostissa osoitteeseen LEI(at)prh.fi
- jos haluat lähettää hakemuksen kirjepostissa, osoite on:

Patentti- ja rekisterihallitus
LEI-rekisteri
00091 PRH

Kuva 18. PRH:n asiakkaille kohdistettu kuvaus LEI-tunnuksen hakemuksen lähettämisestä.

Tunnusta hakeva asiakas tulostaa PRH:n hakemuksen, täyttää sen tiedoillaan ja nimi- kirjoituksellaan, jonka jälkeen lähettää täytetyn hakemuksen PRH:lle sähköpostina. Se, missä muodossa hakemus on sähköpostin liitteenä, on jätetty asiakkaan tulkinnan vaaraan, joten hakemukset tulevat hyvin vapaamuotoisina – joskus kännykällä otettuna kuvana hakemuksesta, joskus skannattuna paperina. Käytännössä ainoa vaihtoehto tekstin tulkinnalle PDF-tiedostoista tai kuvista olisi käyttää mahdollisimman hyvää optista tekstin tunnistusteknologiaa Optical character recognition, lyhyesti OCR, mutta se ei takaa virheettömyyttä tai välttämättä edes riittävää oikein tulkittujen osuutta kokonaismäärästä. Tästä syystä on vaikea korvata ihminen kokonaan LEI-laskujen käsittelyn prosessista.

Yhteisten neuvottelujen jälkeen päätimme asianomaisten kanssa, että lähdetään ainakin ensin automatisoimaan kaikki muut osat paitsi ihmisen tekemä hakemusten tietojen tulkinta. Muut osat sisältävät tekstin kopioimista selaimelta, valikoista valitsemista, Excel-kaavion täyttöä ja tiedostojen siirtoa välityspalvelimelle, jotka ovat melko yksinkertaisia toteuttaa ohjelmistorobotiikalla. Työprosessi vaikuttaa ohjelmistorobotiikalle sopivalta, mutta oli hankala arvioida ennen työn aloittamista, sisältääkö prosessi liikaa kognitiivista päättelyä virheiden käsittelyä varten.

6.1 Toteutus

LEI-laskujen käsittelyn työnkulku:

1. Kirjautu Virre-tietopalveluun, joka sisältää LEI-asiakashallinnan laskutustietoineen sekä viralliset rekisteritiedot yrityksistä, säätiöistä ja yrityskiinnityksistä.
2. Tulkitse asiakkaan LEI-hakemus ja luo tietojen perusteella uusi asiakasmerkintä järjestelmään. Tiedot tallennetaan Virreen, jotta LEI-vuosimaksut voidaan laskuttaa automaattisesti Virren ja laskutusjärjestelmän rajapinnan kautta.
3. Hae Virren asiakashaulla kaikki tilaltaan passiiviset asiakkaat, joiden LEI-tunnuksia ei ole laskutettu ja valitse niistä ensimmäinen käsiteltäväksi.
4. Avaa toinen selainikkuna LEI-ylläpitosovelluksen osoitteeseen ja kirjautu käyttäjätunnuksella sisään.
5. Kopioi passiivisen asiakkaan Y-tunnus Virrestä ja syötä toiseen selainikkunaan, jossa on Yritys- ja yhteisötietojärjestelmä YTJ:n LEI-ylläpitosovellus.
6. Hae Y-tunnuksella asiakkaan tiedot ylläpitosovelluksesta ja lisää asiakkaan asiakastunnus Virrestä LEI-sovelluksen tietoihin.
7. Nouda Virrestä asiakkaan loput tiedot ja kirjaa ne oikeisiin kohtiin laskutuspyynnön mukaiseen Excel-tiedostoon.
8. Lopuksi aseta asiakkaan tilaksi ”Aktiivinen”.
9. Toista askeleet 3-8, kunnes passiivisia asiakkaita ei enää löydy ja lähetä asiakastiedoista koostettu Excel-tiedosto Valtion talous- ja henkilöstöhallinnon palvelukeskukseen, eli Palkeisiin käsiteltäväksi.

Robotin on tarkoitus automatisoida kohdat 3–9 työnkulusta. Robotti tarvitsee Virreä ja LEI-ylläpitosovellusta käyttääkseen molempiin omat käyttäjätunnukset. Työnkulkuun ei ole lueteltu eri kohtia, missä robotin täytyy tehdä virhetarkistuksia. Vaikka Robot Framework tarjoaa työkalut virheiden käsittelyyn ja havaitsemiseen, on niiden käyttö toimintojen haarautumiseen hieman työlästä. Tästä syystä työn yksi suurimmista haasteista oli virheiden käsittely ja ääritapausten huomiointi.

Ainoa tapa varmistua robotin suorituksen onnistumisesta on tarkistaa suoritusraportti joka päivä suorituksen jälkeen, mutta käytäntö ei täytä ohjelmistorobotiikan hallintakriteerejä. Järkevintä useamman robotin hallinnan kannalta olisi, jos kaikki robotit ilmoittaisivat toimistaan jonkin ihmiselle tarkoitetun hallintatyökalun kautta. Asiaa koetettiin parantaa parantamalla robotin virheilmoitustapaa. Virheen sattuessa robotti lähettäisi virheilmoituksen sähköpostin välityksellä LEI-tunnuksia hallinnoiville henkilöille tai robotin ylläpitäjälle.

Virhetapauksia, joita tuli ottaa huomioon:

- Virreen on voitu kirjata Y-tunnus väärin, joten LEI-ylläpitosovelluksesta ei löydy kyseisellä tunnuksella tietoja.
- LEI-sovellukseen ei ole välttämättä tehty hakemusilmoitusta Virreen kirjattulla y-tunnuksella, joten ylläpitosovelluksesta ei löydy kyseisestä asiakasta.
- Virren sekä LEI-sovelluksen käyttäjätunnustietojen salasana vanhenee aina kolmen kuukauden välein. Tätä varten tarvitaan joko käyttäjätunnus, minkä salasana ei vaihdu tai salasanan vanhentuessa pitää manuaalisesti vaihtaa robotin tietoihin uusi salasana.
- Laskutukseen lähetettävällä Excel-tiedostolla on tarkat merkkien maksimimäärät ja muotokriteerit lähes joka asiakastieto-kentälle.
- Virre-palvelun käyttökotot. Robotin voisi kytkeä pois päältä katkojen ajaksi ja käynnistää uudelleen, mutta paras ratkaisu olisi robotin itsenäinen reagointi asiaan.

Jos Virressä olevan asiakkaan y-tunnuksella ei löydy tietoja LEI-sovelluksesta, niin robotti lähettää asiasta virheilmoitus sähköpostilla. Virheilmoitukset lähetetään PRH:n laskutusosastolle ja ongelmatapaukset hoidetaan käsin sitä kautta. Virheilmoitukseen liitetään tarvittaessa poikkeuksen aiheuttaneen asiakkaan Y-tunnus. Jos robotin kirjautumisyritys ei mene läpi Virreen tai LEI-ylläpitosovellukseen, niin robotti lähettää virheilmoituksen, jossa kehoitetaan tarkistamaan robotin käyttäjätunnuksen salasanan voimassaolo ja resetoimaan se tarvittaessa. Resetoinnin yhteydessä pitää muuttaa robotin tietoihin uusi salasana.

Jos robotin suorituksessa ilmenee jokin muu virhe, niin robotti lähettää yleisen virheilmoituksen sähköpostin kautta robotin ylläpitäjälle. Sähköpostin luontia varten tehtiin oma avainsana ”Virheilmoitus Spostin Lahetys”.

*** Settings ***

```
Suite Teardown Run Keyword If Any Critical Tests Failed    Virheilmoitus
Spostin Lahetys # Lähetä virheilmoitus sähköpostina, jos tulee kriittisiä virheitä
```

*** Keywords ***

```
Virheilmoitus Spostin Lahetys
[Arguments]    ${postin_otsikko}=${SPOSTIN_OTSIKKO} ${postin_sisalto}=${SPOSTIN_YLEISSISALTO}    ${postin_saaaja}=ylläpitäjänosoite    # Argumenttina annetaan lähetettävän sähköpostin sisältö. Jos ei anneta, niin lähettää viestin oletussisällöllä ylläpitäjälle.
[Tags]    virheilmoitus    emailsahkopostin lahetys
Laheta Email    ${LAHETTAJA}    ${postin_saaaja} ${postin_otsikko}    ${postin_sisalto}
```

Esimerkkikoodi 2. Virheilmoitus Spostin Lahetys –avainsanan sisältö ja virheentarkistuksen konteksti Robot Frameworkissa. Argumenttina avainsanaa kutsuttaessa voi antaa lähetettävän sähköpostiviestin otsikon, sisällön ja vastaanottajan.

Virheilmoitus Spostin Lahetys -avainsanaa kutsutaan suorituksen lopussa, jos robotin missään kriittisessä testitapauksessa tapahtuu virhe. Samalla avainsanalla lähetetään tieto y-tunnuksesta, millä ei löydy LEI-ylläpitosovelluksesta tietoja. Silloin avainsanalle annetaan argumenttina viestin sisältö. Jos argumenttia ei anneta, niin sähköpostin sisältö on oletusarvoisesti \${SPOSTIN_YLEISSISALTO}-muuttuja, joka sisältää oletusvirheen tekstin. Laheta Email -avainsanan toteutus löytyy EmailLibrary-kirjastosta, mikä on sopivaksi muokattu versio seleniummasterissa olevasta GmailEmailLibrarysta (16).

```
import smtplib
from email.MIMEMultipart import MIMEMultipart
from email.MIMEBase import MIMEBase
from email.MIMEText import MIMEText
from email import Encoders
import os

class LahetaEmailTyokalu(object):

    ROBOT_LIBRARY_SCOPE = 'Global'

    def __init__(self):
        print u'tyokalu, jolla lähetetään sähköpostia'

    def laheta_email(self, lahettaja, vastaanottaja, otsikko, sisalto):
        msg = MIMEMultipart()

        msg[u'From'] = lahettaja.encode('utf-8')
        msg[u'To'] = vastaanottaja.encode('utf-8')
        msg[u'Subject'] = otsikko.encode('utf-8')

        msg.attach(MIMEText(sisalto, "plain", "utf-8"))

        mailServer = smtplib.SMTP("palvelimenosoite", 25)
        mailServer.ehlo()
        mailServer.sendmail(lahettaja, vastaanottaja, msg.as_string())

        mailServer.close()
```

Esimerkkikoodi 3. Skriptissä käytetään Simple Mail Transfer Protocol, lyhennettynä SMTP, jolla sähköpostiviestejä välitetään palvelimelta toiselle. Itse viesti sisältöineen enkoodataan Multipurpose Internet Mail Extensions, lyhyesti MIME, -standardin mukaan. Välityspalvelimen osoite on PRH:n oma, jota voi käyttää vain sallitut ip-osoitteet. Robotin tietokoneen ip-osoite lisättiin hyväksytyjen lähittäjien listalle.

LahetaEmailTyokaly-luokassa on määritelty Robot Frameworkin näkyvyysalue Global-arvoksi, mikä tarkoittaa, että luokasta luodaan vain yksi instanssi ohjelman käytettäväksi. Tämä takaa vain yhden ja saman luokan instanssin käytön. Laheta_email-funktio ottaa

parametreiksi lähettäjän, vastaanottajan, sähköpostin otsikon ja sisällön. Niistä se koostaa MIME-standardilla sähköpostin. SMTP-yhteys aloitetaan kutsumalla ehlo()-funktiota, joka käynnistää kommunikaation laajempaa Extended SMTP:tä käyttäen, jos mahdollista. Vaikka välityspalvelin ei tukisi ESMTP:n käyttöä, vastaa palvelin sitten käyttämään SMTP:tä. Sähköpostiviesti lähetetään merkkijonona välityspalvelimen kautta vastaanottajalle.

Yleisen virhetapauksen sattuessa täytyy lukea robotin suorituksesta seuraavaa lokitiedostoa ja selvittää tarkemmin virheen alkuperä. Yleisiä virheitä ei pitäisi tapahtua kovin usein vakaalla verkkoyhteydellä, kunhan robottia suorittavalla tietokoneella on tarpeeksi resursseja toimintojen vakaaseen suoritukseen.

Seuraavaksi on listattuna kaikki virhesähköpostitapaukset sisältöineen, niiden otsikot alluviivattuna, viestin sisältö sen perässä ja lopuksi viestin vastaanottaja:

- Jos Virreen kirjautuminen estyy:
"Laskuttajarobotin kirjautuminen Virreen epäonnistui - LEI-laskutusrobotti ei päässyt kirjautumaan Virreen! Robotin yhteydessä Virreen on jokin ongelma tai käyttäjätunnuksen tiedot ovat vanhentuneet tai vaihdettu. Tarkista onko robotin käyttäjätunnuksen salasana vanhentunut. Jos on, niin salasana täytyy resetoida ja syöttää robotille uusi salasana vanhan tilalle." – helpdeskin ylläpitoon
- Jos LEI-sovellukseen kirjautuminen estyy:
"Laskuttajarobotin kirjautuminen LEI-ylläpitosovellukseen epäonnistui - LEI-laskutusrobotti ei päässyt kirjautumaan LEI-ylläpitosovellukseen! Robotin yhteydessä LEI-sovellukseen on jokin ongelma tai käyttäjätunnuksen tiedot ovat vanhentuneet tai vaihdettu. Tarkista onko robotin käyttäjätunnuksen salasana vanhentunut. Jos on, niin salasana täytyy resetoida ja syöttää robotille uusi salasana vanhan tilalle." – helpdeskin ylläpitoon
- Jos Virressä asiakkaalla on virheellinen asiakasryhmä valittuna:
"LEI-laskutusrobotti havaitsi poikkeustapauksen - \${ASIAKASTUNNUS}, \${ytunnus_kopio}: Asiakasryhmäksi Virren asiakashallinnassa on valittu Kotitaloudet tai Valtion virasto tai laitos eikä laskutusexcelille voida muodostaa asiakasnumeroa. Asiakasryhmä voi olla vain Elinkeinoelämä/Kunnat tai kuntayhtymät/Muut (esim. säätiöt tai yhdistykset). Asiakasta ei lisätty laskutusexceliin. Asiakas on statukseltaan aktiivinen Virressä." – laskutuksen sähköpostiin
- Jos yritystä ei löydy LEI-sovelluksesta Virren y-tunnuksella:
"LEI-laskutusrobotti havaitsi poikkeustapauksen - \${ASIAKASTUNNUS}, \${YTUNNUS}: Y-tunnuksella ei löytynyt tietoja LEI-sovelluksesta, koska Y-tunnusta ei ole rekisteröity millekään yritykselle. TARKISTA Y-TUNNUS! Asiakasta ei lisätty laskutusexceliin. Asiakas on statukseltaan aktiivinen Virressä. – laskutuksen sähköpostiin

- Jos yritykselle ei ole tehty LEI-hakemusta:
"LEI-laskutusrobotti havaitsi poikkeustapauksen - \${ASIAKASTUNNUS}, \${YTUNNUS}: Yritykselle ei ole myönnetty LEI-tunnusta. TARKISTA Y-TUNNUS! Mikäli Y-tunnus on syötetty oikein, LEI-tunnus on jäänyt rekisteröimättä. LEI-tunnus tulee rekisteröidä ennen hakemuksen laskuttamista. Asiakasta ei lisätty laskutusexceliin. Asiakas on statukseltaan aktiivinen Virressä." – laskutuksen sähköpostiin
- Jos yritykselle ei voitu tallentaa asiakastunnusta LEI-sovelluksessa:
"LEI-laskutusrobotti havaitsi poikkeustapauksen - \${ASIAKASTUNNUS}, \${YTUNNUS}: Asiakasnumeroa ei voitu tallentaa LEI-ylläpitosovellukseen, koska emoyhtiön tai tytäryhtiön konsernisuhdetiedoissa on puutteita. ASIAKASTUNNUKSEN VOI TALLENTAA LEI-YLLÄPITOSOVELLUKSEEN SEN JÄLKEEN, KUN KONSERNISUHDETIEDOT ON PÄIVITETTY. Asiakas lisättiin laskutusexceliin." – laskutuksen sähköpostiin
- Jos yritystä ei löydy LEI-sovelluksesta Virren puolelta poimitun lisätiedotkentän sisällöllä (jonka pitäisi olla Y-tunnus tai tyhjä kaikissa tapauksissa):
"LEI-laskutusrobotti havaitsi poikkeustapauksen - \${ASIAKASTUNNUS}, \${YTUNNUS}: Y-tunnuksella ei löytynyt tietoja LEI-sovelluksesta. TARKISTA VIRREN LISÄTIEDOT-KENTTÄÄN SYÖTETTY Y-TUNNUS! Asiakasta ei lisätty laskutusexceliin. Asiakas on statukseltaan aktiivinen Virressä." – laskutuksen sähköpostiin
- Jos yrityksellä ei ole voimassa olevaa rekisteröintiä LEI-sovelluksessa:
"LEI-laskutusrobotti havaitsi poikkeustapauksen - \${ASIAKASTUNNUS}, \${YTUNNUS}: Yrityksellä ei ole voimassa olevaa rekisteröintiä. TARKISTA Y-TUNNUS! Asiakasta ei lisätty laskutusexceliin. Asiakas on statukseltaan aktiivinen Virressä." – laskutuksen sähköpostiin
- Jos robotti kohtaa virheen, joka johtaa testitapauksen epäonnistumiseen:
"LEI-laskutusapuri on kohdannut virheen - LEI-laskuja käsittelevä robotti on kohdannut virheen. Ota yhteys järjestelmänvalvojaan. Päivittäistä Exceliä ei ehkä luotu oikein. Tarkista robotin loki C:/Robotin_loki/ tietokoneelta adminsrv3." – helpdeskin ylläpitoon

Kirjautumisvirheitä ei pitäisi tapahtua salasanan vanhenemisestä, mutta tarkistus on hyvä jättää siltä varalta, että jokin muu syy estää kirjautumisen onnistumisen. Se voi ainakin aiheutua verkkosivujen käyttökatkoista tai yhteysongelmista kirjautumisyrityksen aikana. Kulloinkin käsiteltävän asiakkaan Y-tunnus on robotille tallennettu \${YTUNNUS}-muuttujaan tai \${ytunnus_kopio}-muuttujaan siinä tapauksessa, että Virressä lisätiedotkenttä ei ole tyhjä. Muuttujien sisältö tulostuu näkyviin sähköpostin saajalle.

Työhön valittiin muuten hyvin samanlaiset työkalut kuin karanteenitikkien käsittelyä varten, mutta Firefox-selaimen käyttöä varten käytettiin myös sitä tukeva verkkoajuri GeckoDriver 0.17.0. Tiedoston siirtoa varten käytettiin pysftp 0.2.9 –Python kirjastoa (13), jolla mahdollistettiin SSH File Transfer Protocol, eli SFTP-protokollan käyttö. Sen

tarkoitus on mahdollistaa asiakastiedoista luodun Excel-tiedoston siirto Palkeisiin. Xlwt-kirjastoa (15) käytettiin Excel-tiedoston kirjoittamiseen. Tämän vaihtoehtona kokeiltiin ensin XlsxWriteriä, mutta tiedostomuoto piti olla Excelin 97/2000/XP/2003 XLS-formaatissa, minkä luonti ei onnistunut kuin xlwt-kirjastolla. Työkalut pysyivät hyvin samanlaisina edellisen työn kanssa, koska silloin robottia voidaan suorittaa samalla koneella kuin karanteenitikettejä käsittelevä robottia, säästäten resursseja.

Robotin toimintaa testattiin keksityillä asiakastiedoilla, jotka tehtiin varta vasten testaamaan erilaiset poikkeustapaukset. Palkeilla oli robotin testausvaiheessa käynnissä uuden tiedostopalvelimen käyttöönotto, joka hidasti tiedostonsiirtoyhteyden kokeilua noin viikon verran. Uusien yhteystietojen ja sftp:n käyttöön oikeuttavilla käyttäjätunnusten saannin jälkeen kokeiltiin yhteyttä Python-skriptillä.

```
import pysftp

def LahetaPalkeisiin():
    paikallinenpolku = 'C:/python_kokeilut/%s', % tiedostonimi
    etapolku = '/Tulot/%s', % tiedostonimi
    cnopts = pysftp.CnOpts()
    cnopts.hostkeys = None
    with pysftp.Connection(host="etapalvelimenosoite", username="kayttajatunnus", password="salasana", cnopts=cnopts) as sftp:
        sftp.put(paikallinenpolku, etapolku)

    sftp.close()
```

Esimerkkikoodi 4. Excel-tiedoston luontiin lisättiin LahetaPalkeisiin-metodi, jota Robot Frameworkista kutsutaan avainsanalla Laheta Palkeisiin. Käyttäjätunnus ja palvelimen osoitteen tiedot on piilotettu esimerkkikoodista.

Metodissa määritellään ensin polut mihin ja mistä tiedosto siirretään, jonka jälkeen avataan sftp-yhteys etäpalvelimelle ja siirretään tiedosto pysftp:n put-komennolla. Lopuksi yhteys suljetaan.

Robotin on tarkoitus suoriutua kerran päivässä työajan ulkopuolella, kun työntekijät ovat saaneet tulkittua asiakkaiden hakemuksia ja luoneet niiden pohjalta passiivisia asiakkaita Virreen. Suoritusta varten piti valmistella oma tietokone, jonne asennettiin tarvittavat kirjastot robotin ajoa varten. Tietokoneella on Windows Server 2012 R2 Standard – käyttöjärjestelmä, mutta se ei estä robotin tarvitsemien kirjastojen toimintaa.

Omalla tietokoneellaan robotin käynnistää Windowsin oma Task Scheduler-ohjelma, millä pystyy ajoittamaan toimintojen tai ohjelmien käynnistymisiä. Sillä on ajoitettu suorittamaan robotti konsolikomennolla `"pybot --outputdir C:\Robotin_loki\%date:~`

4,4%%date:~-7,2% --timestampoutputs C:\python_kokeilut\LEI_Laskutus_Apuri.robot". Parametrilla "outputdir" määritetään lokitiedostojen tallennussijainti, joka tässä tapauksessa luodaan dynaamisesti kuukauden mukaan muotoon VVVVMM, missä V:t kuvaavat vuotta ja M:t kuukautta. Jos sijaintia ei määritä, niin oletuksena luodaan raportit suoritus sijaintiin. Parametrilla "timestampoutputs" mahdollistetaan lokitiedostojen automaattinen nimeäminen tarkkojen aikaleimojen mukaan.

6.2 Robotin arviointi

Ennen kuin robotti laitettiin tuotantoon, analysoitiin robotin kehityksen aikana suoritusraporteista keskimääräisen päivittäisen suoritusajan ja tehokkuuden. Kuva 13 sisältää onnistuneesta testiajasta syntyneen lokin tietoja.

TEST	Asiakastietojen käsittely	00:04:18.939
Full Name: LEI Laskutus Apuri.Asiakastietojen käsittely		
Tags: asiakastiedot, virre		
Start / End / Elapsed: 20180220 10:57:35.139 / 20180220 11:01:54.078 / 00:04:18.939		
Status: PASS (critical)		
+	KEYWORD	00:00:00.047
\${apumuuttuja} = Tarkista Onko Passiivisia		
+	KEYWORD	00:00:00.000
BuiltIn. Set Suite Variable \${ONKO_EXCEL_AUKI}, \${False}		
+	KEYWORD	00:00:00.000
BuiltIn. Set Suite Variable \${JATKETAANKO}, \${apumuuttuja}		
+	KEYWORD	00:00:00.015
BuiltIn. Run Keyword If \${JATKETAANKO}==\${True}, Avaa Excel Kirjoitukseen		
+	KEYWORD	00:00:24.140
BuiltIn. Run Keyword If \${JATKETAANKO}==\${True}, Avaa Selain Leisovellukseen Ja Kirjaudu		
-	FOR	00:03:47.033
\${asiakastapaus} IN RANGE [1 1000]		
Start / End / Elapsed: 20180220 10:57:59.341 / 20180220 11:01:46.374 / 00:03:47.033		
+	VAR	00:00:22.193
\${asiakastapaus} = 1		
+	VAR	00:00:20.788
\${asiakastapaus} = 2		
+	VAR	00:00:21.660
\${asiakastapaus} = 3		
+	VAR	00:00:20.883
\${asiakastapaus} = 4		
+	VAR	00:00:20.811
\${asiakastapaus} = 5		
+	VAR	00:00:21.421
\${asiakastapaus} = 6		
+	VAR	00:00:21.082
\${asiakastapaus} = 7		
+	VAR	00:00:19.851
\${asiakastapaus} = 8		
+	VAR	00:00:22.000
\${asiakastapaus} = 9		
+	VAR	00:00:36.344
\${asiakastapaus} = 10		
+	KEYWORD	00:00:00.000
BuiltIn. Log Ei löytynyt enempää passiivisia asiakkaita, lopetetaan suoritus		
+	TEARDOWN	00:00:07.704
Sulkemistoiminnot		

Kuva 19. Kuvakaappaus kymmenen testiasiakkaan käsittelyn jälkeisestä suorituslokista. Jokaisen kutsutun avainsanan ja sen sisältämän toiminnan voi aukaista tarkemmin tarkasteltavaksi. Lokissa auki asiakastapaukset, jotka käsiteltiin for-silmukassa.

Robot Frameworkin testiautomaatioluonteen vuoksi testisarja nimeltä "LEI Laskutus Apuri", joka on käytännössä ohjelmistorobotti, sisältää kolme testitapausta "Virre kirjau-

tuminen”, ”Virre asiakashaku” ja ”Asiakastietojen käsittely”, jotka kuvaavat robotin päätoimintoja. Virre kirjautuminen sisältää Chrome-selaimen avaamisen Virre-palveluun ja sinne sisäänkirjautumisen. PRH:n verkossa kaikki liikenne kulkee suojatun välityspalvelimen kautta, niin robotin avatessa selaimen, sen täytyy ensin selvittää välityspalvelimen tiedot. Selvitys aiheuttaa viiveen Virre kirjautuminen -toimintoon. Samanlainen viive tulee Avaa Selain Leisovellukseen ja Kirjautu -avainsanan yhteydessä, kun siinä aukaistaan Firefox-selain ensimmäisen kerran.

Robotin yhteenlaskettu suoritus aika kaikkine toimintoineen oli kymmenen testiasiakkaan käsittelyyn 5 minuuttia, 5 sekuntia ja itse asiakkaiden käsittely kesti noin 21 sekuntia per asiakas. Robotti varmistaa, joka asiakkaan jälkeen, onko käsiteltäviä asiakkaita enää listalla, ja tulkitsee, ettei asiakkaita ole enää käsiteltäväksi, kun se ei löydä verkkosivulta enää hakutuloksien elementtejä. Hakutuloksien elementtejä etsitään Wait Until Element Is Visible -avainsanalla, joka odottaa aikakatkaisuun asti, että elementti löytyisi sivulta. Aikakatkaisun odotus aiheuttaa viivästyksen viimeisen asiakastapauksen kohdalla. On mahdollista, että aikakatkaisun aiheuttaa verkkosivun hidus lataus tai verkko-ongelmat, mutta käsittelemättä jääneet asiakkaat jäävät pahimmassa tapauksessa vain seuraavalle suorituskerralle.

Viiden työntekijän ryhmä luo työpäivän aikana noin 100 uutta passiivista asiakasta. Yhden asiakkaan käsittelyyn robotilla kestää keskimäärin noin 21 sekuntia. Kaikkien asiakkaiden käsittelyyn menisi päivittäin keskimäärin 21 sekuntia kertaa asiakkaiden määrä päivän päätteessä. Jaetaan tulos vielä 60:lla, että saadaan tulos minuuteiksi.

$$\text{Robotin keskimääräinen suoritus aika päivässä} = \frac{21 \text{ sekuntia} * 100}{60 \text{ sekuntia}} = 35 \text{ minuuttia}$$

Keskimääräinen suoritus aika työpäivän asiakkaiden käsittelyyn on siis noin 35 minuuttia, johon lisätään robotin alku- ja lopputoimet, mikä on enintään pari minuuttia sen päälle. Robot Frameworkin järkevän käytön rajoja koeteltiin, kun testitapausten koko ja suoritus aika pitenevät, koska kehitysympäristö toimii parhaiten lyhyiden erinäisten testitapausten suoritukseen. Pitempi suoritus aika nostaa myös yhteysongelmista tai muista aikakatkaisuista johtuvien virheiden todennäköisyyttä.

Jotta robotin höytyjä voidaan arvioida paremmin, täytyy LEI-hakemusten tilastoja analysoida tarkemmin. Kuvasta 20 nähdään hakemusten nousu alkaen jo vuoden 2017 elokuussa. Tiedot kattavat LEI-hakemusten ja laskutusten määrän ajalta 2015–2018 maaliskuun 15. päivään asti. PRH alkoi myöntää tunnuksia vuoden 2014 alusta.

LEI-hakemusten laskutusvolyymi 2015-2018

Nimike	Vuosi	tammi	helmi	maalis	huhti	touko	kesä	heinä	elo	syys	loka	marras	joulu	Kokonaistulos
LEI-tunniste hakemus (alv)	2015	47	24	31	10	42	27	18	10	21	27	10	32	299
LEI-tunniste hakemus (alv)	2016	0 *	49	18	17	19	72	20	12	21	9	17	39	293
LEI-tunniste hakemus (alv)	2017	25	12	17	53	84	40	200	269	479	388	558	529	2 654
LEI-tunniste hakemus (alv)	2018	1 051	1 003	910 **										2 964

* Laskutusjärjestelmän vaihtumisen vuoksi tammikuussa 2016 ei laskutusta

** Laskutusrobotin käyttöönotto 28.2.2018. Maaliskuun laskutusmäärät ajalta 1.-15.3.

LEI-hakemusten volyyymi 2015-2018

Ilmoitustyyppi	Vuosi	tammi	helmi	maalis	huhti	touko	kesä	heinä	elo	syys	loka	marras	joulu	Kokonaistulos
LEI-hakemus	2015	54	20	45	12	52	22	14	13	21	28	11	31	323
LEI-hakemus	2016	26	24	22	16	26	65	12	16	18	12	34	20	291
LEI-hakemus	2017	24	15	28	49	82	69	190	401	2 702	4 539	2 939	3 435	14 473
LEI-hakemus	2018	1 720	937	407 *										3 064

* Hakemusmäärä ajalta 1.-15.3.

Kuva 20. Kuvakaappaus LEI-hakemusten ja laskutusten määrästä vuosilta 2015-2018 maaliskuuhun 15.3. asti.

LEI-tunnusten kysynnän kasvun selittää kriteerimuutos tunnusten vaatimuksille. Niitä ruvettiin vaatimaan vuoden 2017 lopussa yhteisöiltä, jotka tekevät sijoituspalvelujen tarjoajille toimeksiantoja julkisen kaupankäynnin kohteina olevista rahoitusvälineistä. Marras-kuusta 2017 alkaen LEI-tunnus vaadittiin johdannaista koskeviin toimeksiantoihin ja 3.1.2018 alkaen vielä laajemmin rahastoja, osakkeita ja joukkovelkakirjoja koskeviin toimeksiantoihin. Myös tunnusten hinta aleni 1.12.2017 alkaen, mutta sen ei ole nähty kasvattaneen kysyntää oleellisesti. (26.)

Laskutusjonossa olevien määrän voi laskea vähentämällä LEI-hakemusten volyymin määrän niiden laskutusvolyymistä. LEI-hakemuksia on laskutettu yhteensä $299 + 293 + 2\,654 + 2\,964 = 6\,210$ kappaletta ja LEI-hakemuksia on tehty yhteensä $323 + 291 + 14\,473 + 3\,064 = 18\,151$ kappaletta. (kuva 20.) Laskua jonottavien asiakkaiden määrä on siis $18\,151 - 6\,210 = 11\,941$ kappaletta.

Vuoden 2018 tammikuussa käsiteltiin suurin määrä hakemuksia ilman robotin apua. Jos siitä lasketaan PRH:n LEI-hakemusten laskutuksen maksiminopeus, niin saadaan 1 051 jaettuna kuukauden päivien määrällä. Silloisessa tammikuussa oli 31 päivää, mutta lasketaan laskutusnopeus kuukauden keskimääräisen päivien määrän mukaan.

$$\text{Laskutusnopeus ilman robottia} = \frac{1\,051 \text{ laskua}}{30 \text{ päivää}} \approx 35 \text{ laskua/päivä}$$

Saman nopeuden laskeminen robotin käyttöönoton jälkeen on vielä hieman epätarkkaa pienen otteen takia, mutta sen voi silti laskea arviona vertailun vuoksi. Lasketaan se vuoden 2018 maaliskuulta, jolloin robotti on ollut toiminnassa sen kuun alusta asti, eli 910 jaettuna käsittelypäivien määrällä.

$$\text{Laskutusnopeus robotin kanssa} = \frac{910 \text{ laskua}}{15 \text{ päivää}} \approx 61 \text{ laskua/päivä}$$

Laskutusnopeus näyttäisi melkein kaksinkertaistuneen verrattuna tammikuun 2018 tulokseen, joka oli silloinen huippu käsittelynopeudelle. Tietenkin tämä on vain karkea arvio robotin vaikutuksesta, joka tarkentuu ajan kanssa. On todennäköistä, että laskutusnopeus vaihtelee laskuja käsittelevän tiimin aktiivisuuden mukaan, eli lomaperiodit, sairastumiset ja muut hidastavat tekijät vaikuttavat laskutusnopeuksiin.

Uusien LEI-hakemusten vuosittainen määrä tulee ennusteiden mukaan laskemaan vuosien 2018–2019 aikana muutamaan tuhanteen. Silloin uusia hakemuksia jätettäisi noin $5\,000 / 365 \approx 13,7$ kappaletta päivässä, jolloin jono hupenisi vauhdilla $61 - 13,7 = 47,3$ laskua per päivä. Koko jonon hupenemiseen kuluisi sen arvion mukaan jonon kokonaismäärä jaettuna käsittelynopeudella.

$$\text{Jonon hupenemiseen kuluva aika} = \frac{11\,941 \text{ laskua}}{47,3 \text{ laskua/päivä}} \approx 252,5 \text{ päivää}$$

Jonon hupenemiseen kuluisi maaliskuun 2018 käsittelynopeuden mukaan 252,5 päivää. Käsittelynopeus pitäisi kiihtyä kesän aikana kesätyöntekijän avun seurauksena. Ennusteet ovat täysin LEI-tunnisteiden myöntämistä hallinnoivan Global Legal Entity Identifier Foundationin kriteerien muutosten varassa. Jos LEI-tunnusten kriteerejä tiukennetaan entisestään, niin hakemusten määrä saattaa nousta taas odottamattomasti. (26.)

Tapauksen toteuttamisessa kesti yli viisi viikkoa, mikä venyttää kehityskustannuksia melko paljon. Aikaa kului eniten kommunikointiin eri tahojen kanssa, varsinkin Palkeiden vaatimusten selventämisessä ja testaamisessa kesti monta viikkoa. On ymmärrettävää, että ison organisaation järjestelmän käyttöön tarvitsee tietoa monelta eri ulkoiselta ja sisäiseltä taholta. Suoraan ohjelmointiin ja sen testaamiseen kului arvioiden mukaan melkein kaksi viikkoa. Vaikka toteutus ja työnkulku olivat monimutkaisemmat kuin karanteenitikkien käsittelyrobotilla, ei se ollut kumminkaan vaikeimmasta päästä. Tapaus oli realistisempi näyttö tiukemmilla vaatimuksillaan ohjelmistorobotiikan käyttökohteeksi. Se sisälsi myös realistisen määrän kehityskeskusteluja asiakkaan kanssa sekä selvittelyä käytettävän järjestelmän yhteensopivuudesta ja yhteisistä käytännöistä.

6.2.1 Hyödyt

Ihmisen suoritusnopeus samoille tehtäville kestää asiakkaita käsittelevien työntekijöiden mukaan noin kolme minuuttia per asiakas. Olettaen, että ihminen käsittelee tasaisella vauhdilla tauotta 100 asiakasta, niin kokonaiskäsittelyajan pystyy laskemaan samalla tavalla, kuin robotin käsittelyajan muuttamalla minuutit sekunneiksi. Kolme minuuttia on 180 sekuntia.

$$\begin{aligned} \text{Ihmisen keskimääräinen suoritus aika päivässä} &= \frac{180 \text{ sekuntia} * 100}{60 \text{ sekuntia}} = 300 \text{ minuuttia} \\ &= 5 \text{ tuntia} \end{aligned}$$

Todennäköistä on, ettei ihmisen suoritus pysyisi tasaisena, kun työelämässä tulee monesti pieniä keskeytyksiä tauoista sähköpostin tarkisteluun ja muuhun työsosiaaliseen toimintaan. Jos keskeytykset ja tauot otetaan huomioon, niin sadan asiakkaan käsitteelyyn kuluva aika lähestyy kokonaisen kahdeksan tunnin työpäivän tehtävää. Jos robotti jatkaa työtään vuoden säästään 5–8 työtuntia per työpäivä, niin aikasäästöiksi kertyy työaikasäästö per päivä kertaa työpäivien määrä viikossa kertaa viikkojen määrä vuodessa. Vuodessa on keskimäärin 52,177457 viikkoa.

$$\begin{aligned} \text{Minimaaliset työaikasäästöt vuodessa} &= 5 \text{ työtuntia} * 5 * 52,177457 \\ &\approx 1304 \text{ työtuntia/vuosi} \end{aligned}$$

$$\begin{aligned} \text{Maksimaaliset työaikasäästöt vuodessa} &= 8 \text{ työtuntia} * 5 * 52,177457 \\ &\approx 2\,087 \text{ työtuntia} / \text{vuosi} \end{aligned}$$

Robotti säästää toiminnallaan arviolta 1 304-2 087 työtuntia vuodessa. Jos työtunnit lasketaan Suomen vuoden 2016 julkishallinnon keskipalkan (23) mukaan rahaksi, niin vuosittaissäästöiksi saadaan 1 304 h * 14,33 €...–2 087 h * 14,33 €, eli 18 686,32–29 906,71 €. Robotin pitäisi maksaa itsensä takaisin vain parin kuukauden käytön jälkeen, mikä on erittäin hyvä tulos sellaisenaan.

Hyötyjen mukaan täytyy laskea työilmapiirin parannus, kun yksitoikkoiset työt siirtyvät robotille, sekä työn kirjanpidon parannus robotin suorituslokien muodossa ja käsittelyprosessin virheettömyyden parantaminen. Näiden ominaisuuksien arvoa on vaikea mitata ja niiden arviointiin vaaditaan pitempää tyytyväisyys- ja suoritusseurantaa. Alustavasti LEI-tunnuksia hakevien asiakkaiden käsittelijöiltä on tullut positiivista palautetta.

6.2.2 Heikkoudet

Tämänhetkisen toteutuksen ehdoton heikkous on sen seurantamahdollisuudet. Jos jokin odottamaton virhe tapahtuu kesken robotin työskentelyn, voi asiakkaiden käsittely jäädä kesken ja Excel-tiedosto voi jäädä tallentamatta tai lähettämättä Palkeisiin. Robotti tekee virheilmoituksen sattuneesta ja lokista voi saada käsiteltyjen asiakkaiden tiedot, mutta ne täytyy kopioida käsin Excel-tiedostoon, jos robotilla on ollut jokin virhe sen käsiteltyssä.

Sekin on teoriassa mahdollista, että jokin muu taho, kuten Windows-päivityksistä johtuva pakotettu tietokoneen uudelleenkäynnistys tai verkkoyhteyden katkeaminen väärään aikaan tarkoittaisivat, ettei robotti kykenisi raportoimaan suorituksensa kesken jäämisestä ulospäin. Tätä ehkäistään pitämällä robotti mahdollisimman kontrolloidussa työympäristössä.

Robotin käyttämät verkkosivut, Virre ja LEI-ylläpitosovellus, latautuvat ajoittain epäjohtomukaisen hitaasti. Testauksen ja käyttöönoton aikana jouduttiin useista eri vaiheista nostamaan odotusten aikakatkaisujen pituuksia. Teoriassa se voi nostaa robotin suoritusaikaa yli kolminkertaisesti, jos verkkosivut latautuisivat hyvin hitaasti pitemmän aikaa. Pitkiä, yli 30 sekunnin latausaikoja havaittiin kolmessa tapauksessa 400:stä. Hidastukset

voivat johtua hetkellisistä verkko-ongelmista palvelimella tai robotin tietokoneella. Ongelma kierrettiin nostamalla monessa odotusvaiheessa aikakatkaisun rajaksi 40–50 sekuntia, mutta kaikista paras ratkaisu ongelmaan olisi, että robotti yrittäisi toimintoa uudelleen sivun uudelleenlatauksen jälkeen.

Käyttöönoton alussa robotin toiminta pysähtyi LEI-sovellukseen kirjautumisen aikakatkaisun vuoksi, kun se oli siinä vaiheessa vain 15 sekuntia, joten yhden päivän suoritus jäi tekemättä sen takia. Passiivisia asiakkaita luotiin kumminkin normaaliin tahtiin, mikä aiheutti seuraavana päivänä uuden ongelman: Virren yrityshaku ei näytä tuloksia ollenkaan, jos tulosten määrä ylittää tietyn maksimirajan. Tätä ei ollut otettu huomioon, joten robotti tulkitse, ettei käsiteltäviä asiakkaita vain ollut siltä päivältä. Ongelma kierrettiin väliaikaisesti puuttamalla robotin toimintaan manuaalisesti suorituksen aikana ja vaihtamalla Virren hakukriteereiksi passiivisten elinkeinoelämän asiakkaat, joiden määrä ei ollut liikaa verkkosivun näytettäväksi.

6.3 Jatkokehitys

Jos tai kun PRH siirtyy LEI-tunnusten käsittelyssä täysin sähköiseen lomakkeiden käsittelyyn, niin robotin toiminnan voisi laajentaa kattamaan myös lomakkeiden pohjalta uusien asiakkaiden luonnin Virreen. Jos sen jälkeen saisi vielä poikkeustapausten käsittelyn täysin automaattiseksi, niin ihmisen rooliksi jäisi enää ylläpito ja monitorointi. PRH:n suunnitelmana on palkata kesätyöntekijä, joka tulkitsee asiakaslomakkeita ja luo niiden pohjalta asiakkaita. Sen toivotaan jouduttavan asiakasjonon käsittelyä entisestään.

Varsinaista monitorointia voisi parantaa suorittamalla robottia paikallisesti Windows Task Schedulerin sijaan esimerkiksi jatkuvan integraation ohjelmistolla Jenkinsillä, jossa testitapausten hallinnointi tapahtuisi keskitetyllä palvelimella. Jenkinsiin on olemassa lisäosia, jotka mahdollistavat sähköposti-ilmoitusten automaattisen lähettämisen suoritettavan asian statuksen mukaan.

Käyttöönoton jälkeen robotin toiminnassa löytyi joitakin virheitä ja pari poikkeustilannetta, mitä ei ollut otettu huomioon. Näiltä olisi välttytty paremmin käymällä tarkemmin läpi eri suoritusvaiheiden poikkeustilanteita robotin tilaajan ja käytettyjen verkkopalvelujen kehittäjien kanssa. Toisaalta pitkiin keskustelu- ja kehityksperiodeihin ei aina ole tar-

peeksi aikaa ja robotin käyttämät työkalut voivat muuttua ajan myötä, joten robotti tarvitsee ylläpitoa vähintään palvelujen päivitysten yhteydessä. Jatkuvasta ylläpidosta on myös hyötyä mahdollisten ongelmien diagnosoinnin ja korjaamisen kanssa.

Jos LEI-ylläpitosovelluksen käyttö jatkaa yhteensopivuuttaan vain Firefox-selaimella, niin Firefoxin versiosta 55 eteenpäin joudutaan todennäköisesti miettimään Selenium-kirjaston tilalle jokin muu rajapinta, sillä sen kehittäjät ovat virallisesti ilmoittaneet projektin lopettamisesta. Vaikka Firefoxin verkkoajuri Geckodriver tukee Firefoxin versioita 55:stä ylöspäin, se ei takaa yhteensopivuutta ainakaan kovin pitkään Selenium-kirjaston osalta, jos vapaaehtoiskehittäjät eivät pidä yllä virallisten Selenium-kirjaston kehittäjien jättämää projektia. (20.)

Joitakin ongelmia Firefoxin ja Geckodriverin yhteiskäytössä havaittiin jo LEI-tunnuksia automatisoivan robotin testivaiheessa. Kun Firefox-selaimen sulkee Selenium2Libraryn Close Browser -avainsanalla, niin selain kaatuu ja ilmoittaa, että Firefox lakkasi toimimasta. Ongelma kierrettiin sulkemalla sekä Firefox-selain että Geckodriver Python-skriptillä, joka kutsuu käyttöjärjestelmän taskkill-komentoa (Liite 6). Uusien standardien mukana voi tulla ongelmaan ratkaisu uuden ohjaimen muodossa. Chromen käytöllä LEI-sovelluksen kanssa ei havaittu ongelmia, vaikkei sitä virallisesti tueta. Niin tai näin, tilanetta yhteensopivuudesta täytyy seurata.

7 Yhteenveto

Insinöörityössä pyrittiin selvittämään ohjelmistorobotiikan hyödyntämismahdollisuuksia julkishallinnon virastolle, PRH:lle. Havaittiin, että suuria hyötyjä voi saada nopeallakin aikataululla. Lisäksi selvitettiin Robot Frameworkin mahdollisuuksia ohjelmistorobotiikan saralla. Robot Frameworkia verrattiin UiPathin ohjelmistorobotiikan kehitysohjelmistoon vertailukelpoisuuden vuoksi. Ohjelmistorobotiikan hyödyntämiskohteita havaittiin syntyvän sitä mukaa, mitä enemmän tietoisuus ohjelmistorobotiikasta levisi työympäristössä.

Hyödyntämiskohteiden ehdotuksia oli insinöörityön lopussa kerätty jonoksi asti eri organisaation osista. Jokaisesta ehdotuksesta pitäisi erikseen arvioida kannattavuus, mitä ei tehty insinöörityön aikana. Insinöörityön aloitusvaiheessa keksitylle hyödyntämiskohdelle, laskujen kohdennuksen automatisoinnille, ei ollut aikaa, kun LEI-tunnusten laskutuksessa tarvittiin pikaisempaa apua.

Työssä kehitetyt robotit autoivat toiminnallaan parantamaan PRH:n työtoimien sujuvuutta. Toteutuksista kävi ilmi, kuinka Robot Framework sopeutuu ohjelmistorobotiikan kehitysalustaksi. Ohjelmistorobotiikan yleisistä hyödyistä sekä robottiyksilöiden tuomista paikallisemmista hyödyistä saatiin paljon tietoa ja niiden arviointia varten esitettiin varteenotettavia tapoja. Jokaisessa ohjelmistorobotiikan kohteessa täytyy arvioida erikseen painoarvot välittömille ja välillisille hyödyille. Työn tuloksista voi olla hyötyä tahoille, jotka harkitsevat ohjelmistorobotiikan käyttöönottoa ja pohtivat sopivia käyttökohteita.

Insinööriyön teon aikana opittiin paljon ohjelmistorobotiikan käyttöönottoaskelista. Käyttökohteen ajallisia säästöarvioita on vaikea antaa, ennen robotin alkuvaiheen testiajoa. Jos robotin suoritusnopeudella on painoarvoa, kannattaa suunnitella jo alkuvaiheessa nopea toteutus ja ajaa arvion pohjatiedoiksi testiajoja. Arvioiden tarkkuus nousee varmasti kokemuksenkin myötä.

Robot Frameworkilla pystyy toteuttamaan hyvin pitkälle kaiken, minkä UiPath-ohjelmistollakin, mutta sen omat seuranta- ja hallinnointipuutteet täytyy paikata omilla ratkaisulla. LEI-tunnusten automatisoinnissa se tehtiin sähköposti-ilmoitusten kautta, mutta keskitetyn hallinnointipuolen ratkaisu jäi pohdintavaiheeseen. Jos haluaa valmiin RPA-ohjelmistopakettin, niin UiPath tarjoaa siihen toimivan ratkaisun lisenssin hintaa vastaan. Jos haluaa ilmaisen ja vapaan lähdekoodin RPA-rungon, niin Robot Framework tarjoaa laajat mahdollisuudet kaiken itse toteuttamiseen.

Kehitettyjen ohjelmistorobottien toimintaa esitettiin parilla demonstraatiovideolla PRH:n yhteisten palvelujen, asiakaspalvelujen ja viestinnän tulosalueiden kehittämispäivillä 1.3.2018 Lohjan Kisakalliolla. Esitys oli päivän viimeisenä ja siitä huolimatta työyleisön reaktio oli innostunut ja positiivista palautetta annettiin aiheesta paljon. Se inspiroi varmasti monessa ohjelmistorobotiikan sovellustapoja ja kohteita. Sekin vahvistaa, että ohjelmistorobotiikalle tosiaan on tarvetta.

Työn lopussa kyseltiin automaatiotapauksiin osallistuneilta henkilöiltä palautetta. Se oli kaikin puolin positiivista. Robotteja ylistettiin luotettavuudesta, niiden tuomat avut koettiin hyödyllisiksi ja määrittely koettiin helpoksi. Tyytyväisiä oltiin myös käyttöönoton nopeuteen ja edullisuuteen.

PRH:lla voidaan päättää oman RPA-ohjelman jatkamisesta tämän työn havaintojen perusteella.

Lähteet

- 1 What to install for RIDE automation. 2016. Verkkoaineisto. <<http://gnome-lab.com/wordpress/?p=58>>. Luettu 1.12.2017.
- 2 Robot Frameworkin kotisivut. 2017. Verkkoaineisto. <<http://robotframework.org>>. Luettu 5.12.2017.
- 3 Robot Framework User Guide. 2017. Verkkoaineisto. <<http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html>>. Luettu 5.12.2017.
- 4 Huang, Richard. 2016. Extended Selenium 2 (WebDriver) web testing library for Robot Framework. Verkkoaineisto. <<https://github.com/rickyperc/robotframework-extendedselenium2library>>. Luettu 8.12.2017.
- 5 Klärck, Pekka. 2006. Data-Driven and Keyword-Driven Test Automation Frameworks. Diplomityö. <<http://eliga.fi/writings.html>> Luettu 1.12.2017.
- 6 RIDE. 2016. Verkkoaineisto. <<https://github.com/robotframework/RIDE>>. Luettu 5.12.2017.
- 7 Efecte-ohjelmistoyhtiön kotisivut. 2018. Verkkoaineisto. <<http://www.efecte.com/>>. Luettu 2.1.2018.
- 8 SeleniumHQ Browser Automation. 2017. Verkkoaineisto. <<http://www.seleniumhq.org/>>. Luettu 5.12.2017.
- 9 Faught, Danny. Keyword-Driven Testing. 2004. Stickyminds: Verkkoartikkeli <<https://www.stickyminds.com/article/keyword-driven-testing>>. 5.11.2004. Luettu 11.1.2018.
- 10 Valtion tieto- ja viestintätekniikkakeskuksen kotisivut. 2018. Verkkoaineisto. <<http://www.valtori.fi/>>. Luettu 16.1.2018.
- 11 Pesonen, Teemu. Robotic Process Automation, a new interesting growth opportunity for quality assurance companies? Vala Group: Verkkoartikkeli. <<https://www.valagroup.com/fi/2017/03/robotic-process-automation-new-interesting-growth-opportunity-quality-assurance-companies/>>. 8.3.2017. Luettu 10.1.2018.
- 12 Suomen virallinen tilasto (SVT): Yksityisen sektorin tuntipalkat [verkkajulkaisu]. ISSN=1798-3916. 2014. Helsinki: Tilastokeskus [viitattu: 23.1.2018]. Saantitapa: <http://www.stat.fi/til/ystp/2014/ystp_2014_2015-09-17_tie_001_fi.html>. Luettu 18.1.2018.

- 13 Hinrichs, Jeff. Pysftp 0.2.9. 2016. Verkkoaineisto. <<https://pypi.python.org/pypi/pysftp>>. Luettu 1.2.2018.
- 14 Ohjelmistorobotiikka ja BPM – Opas eroavaisuuksista ja mahdollisuuksista. 2018. Alfame: Verkkoaineisto. <<http://www.alfame.com/ohjelmistorobotiikka-ja-bpm-opas>>. Luettu 30.1.2018.
- 15 Kiseliou, Roman. 2017. Xlwt documentation. Verkkoaineisto. <<https://xlwt.readthedocs.io/en/latest/>>. Luettu 5.2.2018.
- 16 Send Email in Robot Framework Python Using Gmail. 2014. Seleniummaster: Verkkoaineisto. <<http://seleniummaster.com/sitecontent/index.php/selenium-robot-framework-menu/selenium-robot-framework-python-menu/228-send-email-in-robot-framework-python-using-gmail>>. Luettu 1.2.2018.
- 17 Robotic Process Automation Market Growth – RPA Industry Share, Size Report 2024. 2017. Global Market Insights: Verkkoaineisto. <<https://www.gminsights.com/industry-analysis/robotic-process-automation-market>>. 8.2017. Luettu 8.2.2018
- 18 IT Robotic Automation Market. Transparency Market Research: Verkkoaineisto. <<https://www.transparencymarketresearch.com/it-robotic-automation-market.html>>. 11.2016. Luettu 8.2.2018.
- 19 Robotic Process Automation (RPA) Market Size & Share Report, 2024. 2018. Grand View Research: Verkkoaineisto. <<https://www.grandviewresearch.com/industry-analysis/robotic-process-automation-rpa-market>>. 2.2018. Luettu 23.2.2018.
- 20 Firefox 55 and Selenium IDE. 2017. Official Selenium Blog: Verkkoaineisto. <<https://seleniumhq.wordpress.com/2017/08/09/firefox-55-and-selenium-ide/>>. 9.8.2017. Luettu 5.2.2018.
- 21 Ohjelmistorobotiikan (RPA) hyödyntäminen Valtiokonttorissa. 2017. EERA Oy: Selvitys Valtiokonttorille. Sisäinen raportti. 30.10.2017. Luettu 27.10.2017.
- 22 UiPathin kotisivut. 2018. Verkkoaineisto. <<https://www.uipath.com/>>. Luettu 13.2.2018.
- 23 Suomen virallinen tilasto (SVT): Kuntasektorin palkat [verkojulkaisu]. ISSN=1799-0203. 2016, Liitetaulukko 8. Kuntasektorin tuntipalkat toimialaluokituksen mukaan 2016. Helsinki: Tilastokeskus [viitattu: 1.3.2018]. Saantitapa: <http://www.stat.fi/til/ksp/2016/ksp_2016_2017-05-04_tau_014_fi.html>. Luettu 1.3.2018.
- 24 The UiPath Studio Guide. 2018. Verkkoaineisto. <<https://studio.uipath.com/>>. Luettu 13.2.2018.

- 25 UiPath Community Edition License Agreement. 2018. Verkkoaineisto.
<<https://www.uipath.com/community-license-agreement>>. Luettu 22.2.2018.
- 26 Järvenpää, Juha. 2018. Tiimiesimies, YY/Hallintotiimi, Patentti- ja rekisterihallitus.
Sähköpostikeskustelu 20.3.2018.

Karanteeniticketin käsittelyrobotti

```

*** Settings ***
Documentation      Viimeksi päivitetty 6.2.2018 klo 9:22
Suite Setup       Set Suite Variable   ${ONKO_NAKYMASSA_TIKETTEJA}
                  ${EMPTY}
Suite Teardown    Close Browser
Library           ExtendedSelenium2Library # Pitää tukea Angularia (Efecten Angular-elementtien vuoksi), siksi valittu ExtendedSelenium2Library uusimman SeleniumLibrary:n sijaan. Toimii vain robotframeworkin selenium versiolla 1.8.0

*** Variables ***
${EFECTE_TIKETTINAKYMA}    /*[@id="workspace_filter_111"] # Järjestellyn Efecten tikettilistan otsikon xpath
${EFECTE_LOGIN_URL}      http://efecte.prh.fi/tuotanto/Start.do?NoNTLMAuth=true
                          # Efecten kirjautumissivu
${TIKETTILISTAN_ENSIMMAINEN_ALKIO}    /*[@id="workspace_table_0"]/tbody/tr[1]
                          # Tikettilistan ensimmäinen rivi
${TIKETIN_MUOKKAAKAIKKIA_NAPPAIN}    //button[@name="editingAllinDataCard"]
                          # Avatun ticketin Muokkaa kaikkia -näppäimen xpath
${TIKETIN_TALLENNA_NAPPAIN}          /*[@id="efecteDcSaveButton"] # Muokattavan ticketin vihreän tallenna-näppäimen xpath
${UUSI_VIESTI_NAPPAIN}              /*[@id="workspace-handler-attribute-410"]/div[2]/div/div[2]/div[1]/div[1]/button[1] # Avatun ticketin Email-osan Uusi viesti -näppäimen xpath
${TIKETIN_PAIVITYSTIEDOT}          /*[@id="group-466"] # Avatun ticketin Päivitystiedot-osan xpath
${LAHETA_SPOSTI_NAPPAIN}           /*[@id="dcButtonSendMail"] # Avatun ticketin Email-osassa muokattavan sähköpostin vihreä Lähetä-näppäin
${SELAIN}                          chrome # Käytettävä selain
${OMAT_NAKYMAT}                    /*[@id="tree-wrapper"]/div/ul[1]/li[1]/div/ul # Efecten Omat näkymät -alavalikon xpath
${EFECTE_PROFIILIPAINIKE}          /*[@id="user-profile-button"] # Efecten oikean yläkulman profiilin hallinnan näppäimen -xpath
${TUNNUS}                          *****
${S_SANA}                          *****
${VIRHEELLISEN_TIKETIN_KOMMENTTI}  Tämä on automaattinen vastaus. Karanteeniviesti tarvitaan kokonaisuudessaan karanteenin purkamiseksi. Välitä se Outlookilla osoitteeseen 'postiosoite' Tämä tiketti on suljettu.

*** Test Cases ***
Efecte_login
    [Documentation] Avaa Chromen Efecten login-sivulle ja odottaa, että Efecten tikettinäköma aukeaa
    [Tags]          login
    [Setup]
        Set Suite Variable   ${ONKO_VAARIN_TAYTETTY}   ei
        Open Browser        ${EFECTE_LOGIN_URL}    ${SELAIN}
        Input Text          /*[@name="userID"]    ${TUNNUS}
        Input Password      /*[@name="userPassword"]    ${S_SANA}
        Click Button        /*[@id="loginButton"]
        Wait Until Element Is Visible    ${EFECTE_TIKETTINAKYMA} 10

Oikean_nakymän_valinta
    [Documentation] Valitsee Omat näkymät -valikosta Karanteeniticketit -nimisen rivin ja klikkaa sitä
    [Tags]          nakyma
    [Setup]
        Aseta OnkoNakymassaTiketteja on
        Klikkaa Nakymaa Karanteeniticketit
        Wait Until Element Is Visible    ${EFECTE_TIKETTINAKYMA}
    [Teardown] Run Keyword And Ignore Error Onko Nakymassa Kasiteltavia Tiketteja

```

```

Tiketin_aukaisu_muokkaukseen
  [Tags]      tiketti
  [Setup]    Log   Onko tikettejä näymässä: ${ONKO_NAKYMASSA_TIKETTEJA}
  [Timeout]
  Run Keyword If '${ONKO_NAKYMASSA_TIKETTEJA}' =='on'   Tiketin Aukaisu
Muokkaukseen
  ... ELSE No Operation
  [Teardown] Run Keyword If Test FailedSulje Virheellisesti Muotoiltu
Tiketti

Tiketin_kasittely
  [Documentation] Valitsee tiketin statuksen, omistajan, tyyppin, palvelu-
ryhmän, palvelun ja merkkää toimenpiteiksi tekstiä.
  [Tags]      tiketti
  [Setup]    Log   Onko tikettejä näymässä: ${ONKO_NAKYMASSA_TIKETTEJA}
  [Timeout]
  Run Keyword If '${ONKO_NAKYMASSA_TIKETTEJA}' =='on' and '${ONKO_VAA-
RIN_TAYTETTY}'=='ei' Aja Tiketin Kasittelyavainsanat
  ... ELSE No Operation
  [Teardown]

Suljettavien_karanteenitickettien_nakymän_valinta
  [Documentation] Valitsee Omat näymät -valikosta Karanteeniticketit -nimi-
sen rivin ja klikkaa sitä
  [Tags]      nakyma
  [Setup]    Aseta OnkoNakymassaTiketteja   on
Set Suite Variable   ${ONKO_TIKETTI_RATKAISTU} on
Klikkaa Nakymaa Suljettavat karanteeniticketit
Wait Until Element Is Visible   ${EFECTE_TIKETTINAKYMA}
  [Teardown] Run Keyword And Ignore Error   Onko Nakymassa Kasiteltavia Ti-
ketteja

Suljettavan_tiketin_aukaisu_muokkaukseen
  [Tags]      tiketti
  [Setup]    Log   Onko tikettejä näymässä: ${ONKO_NAKYMASSA_TIKETTEJA}
  [Timeout]
  Run Keyword If '${ONKO_NAKYMASSA_TIKETTEJA}' =='on'   Suljettavan Tike-
tin Aukaisu Muokkaukseen
  ... ELSE No Operation
  [Teardown] Run Keyword If Test FailedSet Suite Variable
${ONKO_TIKETTI_RATKAISTU} ei

Suljettavan_tiketin_kasittely
  [Documentation] Valitsee tiketin statuksen, omistajan, tyyppin, palvelu-
ryhmän, palvelun ja merkkää toimenpiteiksi tekstiä.
  [Tags]      tiketti
  [Setup]    Log   Onko tikettejä näymässä: ${ONKO_NAKYMASSA_TIKETTEJA}
  [Timeout]
  Run Keyword If '${ONKO_NAKYMASSA_TIKETTEJA}' =='on' and '${ONKO_TI-
KETTI_RATKAISTU}'=='on'   Aja Suljettavan Tiketin Kasittelyavainsanat
  ... ELSE No Operation
  [Teardown]

Profiilin_navigointi
  [Tags]      logout
  Tee Valinta Profiilihallinnasta Kirjautu ulos

*** Keywords ***
Tee Valinta Pudotusvalikosta
  [Arguments]   ${arg_rivin_nimi}   ${arg_valinta}
  [Documentation] Argumenttina ensin rivin nimi ja toisena valinnan teksti
Scroll Element Into View   /*[@class="group-con-
tent"]/div[*][.//text()='${arg_rivin_nimi}']
Click Element   /*[@class="group-con-
tent"]/div[*][.//text()='${arg_rivin_nimi}']/*[contains(@id,'::data.gaId')]

```

```
Click Element    /*[@class="select2-results"]/li[*][.//text()='${arg_valinta}']
```

Klikkaa Nakymaa

```
[Arguments]      ${arg_nakymän_nimi}
Wait Until Element Contains    ${OMAT_NAKYMAT} ${arg_nakymän_nimi}
Click Element    /*[@id="tree-wrap-per"]/div/ul[1]/li[1]/div/ul/li[*][.//text()='${arg_nakymän_nimi}']
```

Valita Tiketti Spostina Tiedolle

```
Tee Valinta Pudotusvalikosta    Sähköpostipohjat    HD: Sähköposti
(VYVI)
Wait Until Element Is Visible    ${UUSI_VIESTI_NAPPAIN}
Click Button    ${UUSI_VIESTI_NAPPAIN}
Wait Until Element Is Visible    ${LAHETA_SPOSTI_NAPPAIN}
Scroll Element Into View    ${TIKETIN_PAIVITYSTIEDOT}
Click Element    ${LAHETA_SPOSTI_NAPPAIN}
```

Tee Valinta Profiilihallinnasta

```
[Arguments]      ${arg_valinnan_nimi}
[Documentation] Tekee argumenttina saadun valinnan Efecten oikean yläkul-
man profiilinäppäimestä.
Element Should Be Visible    ${EFECTE_PROFIILIPAINIKE}
Click Element    ${EFECTE_PROFIILIPAINIKE}
Wait Until Element Is Visible    /*[@id="profile-content"]/ul/li[*][.//text()='${arg_valinnan_nimi}']
Click Element    /*[@id="profile-content"]/ul/li[*][.//text()='${arg_valinnan_nimi}']
```

Kirjoita Tiketin Tekstikenttaan

```
[Arguments]      ${arg_rivin_nimi}    ${arg_teksti}
[Documentation] Kirjoittaa tiketin tekstikenttään tekstiä. Ensimmäinen
argumentti on rivin nimi ja toinen kirjoitettava teksti. Olettaa, että tiketti
on avattu muokkausta varten.
Element Should Be Visible    ${TIKETIN_TALLENNA_NAPPAIN}
Scroll Element Into View    /*[@class="group-content"]/div[*][.//text()='${arg_rivin_nimi}']
Input Text    /*[@class="group-content"]/div[*][.//text()='${arg_rivin_nimi}']//textarea    ${arg_teksti}
```

Onko Nakymassa Kasiteltavia Tiketteja

```
Set Selenium Implicit Wait 10 seconds
Wait Until Element Is Visible    ${TIKETTILISTAN_ENSIMMAINEN_ALKIO}
Set Selenium Implicit Wait 0 seconds
[Teardown] Aseta OnkoNakymassaTiketteja    Aseta OnkoNakymassaTiketteja
ei
```

Tiketin Aukaisu Muokkaukseen

```
Element Should Be Visible    ${TIKETTILISTAN_ENSIMMAINEN_ALKIO}
Click Element    ${TIKETTILISTAN_ENSIMMAINEN_ALKIO}
Wait Until Element Is Visible    ${TIKETIN_MUOKKAAKAIKKIA_NAPPAIN}
Element Should Be Visible    /*[@id="workspace-handler-attribute-1503"]
/div[2]/div/div/div/div/div/button
Click Element    /*[@id="workspace-handler-attribute-1503"]
/div[2]/div/div/div/div/div/button
Page Should Contain    Laitteen tunnus
Click Element    ${TIKETIN_MUOKKAAKAIKKIA_NAPPAIN}
Wait Until Element Is Visible    ${TIKETIN_TALLENNA_NAPPAIN}
```

Tiketin Muokkaus

```
Tee Valinta Pudotusvalikosta    Status    3 - Odottaa ulkopuolista
Tee Valinta Pudotusvalikosta    Omistaja    Robottil Roope
Tee Valinta Pudotusvalikosta    Tyyppi    Palvelupyyntö
Tee Valinta Pudotusvalikosta    Palveluryhmä    Yleiset IT-palvelut
/HelpDesk
```

Tee Valinta Pudotusvalikosta Palvelu Työasema- ja käyttäjätukipalvelut

Kirjoita Tiketin Tekstikenttaan Toimenpide Tämä tiketti on käsitelty automaattisesti Robot Frameworkilla
[Teardown]

Tiketin Tallennus

Element Should Be Visible \${TIKETIN_TALLENNA_NAPPAIN}
Click Element \${TIKETIN_TALLENNA_NAPPAIN}
Wait Until Element Is Not Visible \${TIKETIN_TALLENNA_NAPPAIN}

Suljettavan Tiketin Aukaisu Muokkaukseen

Element Should Be Visible \${TIKETTILISTAN_ENSIMMAINEN_ALKIO}
Click Element \${TIKETTILISTAN_ENSIMMAINEN_ALKIO}
Wait Until Element Is Visible \${TIKETIN_MUOKKAAKAIKKIA_NAPPAIN}
Page Should Contain Tiketti on ratkaistu
Log Tiketti on ratkaistu, suljetaan tiketti.
Click Element \${TIKETIN_MUOKKAAKAIKKIA_NAPPAIN}
Wait Until Element Is Visible \${TIKETIN_TALLENNA_NAPPAIN}

Suljettavan Tiketin Muokkaus

Tee Valinta Pudotusvalikosta Status 5 - Suljettu

Sulje Virheellisesti Muotoiltu Tiketti

Click Element \${TIKETIN_MUOKKAAKAIKKIA_NAPPAIN}
Wait Until Element Is Visible \${TIKETIN_TALLENNA_NAPPAIN}
Tee Valinta Pudotusvalikosta Status 5 - Suljettu
Tee Valinta Pudotusvalikosta Automaattiviestit Ä,,lä lähetä
Tee Valinta Pudotusvalikosta Omistaja Robottil Roope
Tee Valinta Pudotusvalikosta Tyyppi Palvelupyyntö
Tee Valinta Pudotusvalikosta Palveluryhmä Yleiset IT-palvelut

/HelpDesk

Tee Valinta Pudotusvalikosta Palvelu Työasema- ja käyttäjätukipalvelut

Kirjoita Tiketin Tekstikenttaan Toimenpide Tämä tiketti on käsitelty automaattisesti Robot Frameworkilla

Kirjoita Tiketin Tekstikenttaan Kommentti \${VIRHEELLISEN_TIKETIN_KOMMENTTI}

Tiketin Tallennus

Set Suite Variable \${ONKO_VAARIN_TAYTETTY} on

Aseta OnkoNakymassaTiketteja

[Arguments] \${arg_uusi_arvo}
Set Suite Variable \${ONKO_NAKYMASSA_TIKETTEJA} \${arg_uusi_arvo}
[Teardown] Log Onko tikettejä näkymässä: \${ONKO_NAKYMASSA_TIKETTEJA}

Aja Tiketin Kasittelyavainsanat

Run Keywords Tiketin Muokkaus Valita Tiketti Spostina Tiedolle
Tiketin Tallennus

Aja Suljettavan Tiketin Kasittelyavainsanat

Run Keywords Suljettavan Tiketin Muokkaus Tiketin Tallennus

Versiotiedot

- Python 2.7.14 (32-bit)
- wxPython 2.8.12.1 (32-bit)
- RIDE 1.5.2.1
- Robot Framework 3.0.2
- Selenium2Library 1.8.0
- ExtendedSelenium2Library 0.9.1
- Google Chrome 63.0.3239.132 (Official Build) (64-bit)
- ChromeDriver 2.34
- Windows 10 –käyttöjärjestelmä

- Mozilla Firefox 52.0 Extended Support Release (32-bit)
- GeckoDriver 0.17.0 (32-bit)
- pysftp 0.2.9
- xlwt 1.3.0

- UiPath Studio v2018.1 (Community edition)

LEI-laskuja käsittelevä robotti

```

*** Settings ***
Documentation      Aja tämä robotti täysillä oikeuksilla konsolista komennolla
'pybot --outputdir C:\Robotin_loki\%date:~-4,4%%date:~-7,2% --timestampoutputs
C:\python_kokeilut\LEI_Laskutus_Apuri.robot''
Suite Teardown    Run Keyword If Any Critical Tests Failed      Virheilmoitus
Spostin Lahetys # Lähetä virheilmoitus sähköpostina, jos tulee kriittisiä vir-
heitä
Library           ExtendedSelenium2Library # Pitää tukea Angularia (Virren
Angular-elementtien vuoksi), siksi valittu ExtendedSelenium2Library uusimman
SeleniumLibraryyn sijaan. Toimii vain robotframeworkin selenium versiolla 1.8.0
Library           excelinluonti.py      # Python-tiedosto, joka sisältää
avainsanat excel-tiedoston luontia ja täyttämistä varten
Library           EmailLibrary          # Sähköpostin lähettämistä varten tehty
python-kirjasto
Library           close_ff.py          # Python-skripti, joka sulkee Firefox-se-
laimen (tätä käytettiin, koska firefoxin versioepäsopivuus aiheutti selaimen
kaatumisen, kun kutsuttiin Close Browser -avainsanaa)

*** Variables ***
${ALV_TUNNUS}     ${EMPTY}
${ASIAKASNRO}    ${EMPTY}
${ASIAKASHALLINTA_URL} https://virre.prh.fi/novus/customerManagement
${ASIAKASTUNNUS_KENTTA_LEI} /*[@id="Asiakastunnus"]
${ASIAKASRYHMA} ${EMPTY}
${ASIAKKAAN_KIELI} ${EMPTY}
${ASIAKKAAN_LISATIEDOT} ${EMPTY}
${ASIAKKAAN_NIMI}  ${EMPTY}
${ASIAKKAAN_TARKENNE} ${EMPTY}
${ASIAKASTUNNUS}  ${EMPTY}
${BRUTTO}         N
${HAUN_VIRHETEKSTI} ${EMPTY}
${HELPDESKIN_SPOSTI} helpdeskinosoite
${KIELIKOODI}     ${EMPTY}
${LAHETTAJA}     noreply(at)prh.fi
${LASKUTTAJA}    507001A108
${LASKUTUKSEN_SPOSTI} laskutuksenosoite
${LEI_PERUUTA_NAPPAIN} /*[@id="btnPeruuta"]
${LEI_KTUNNUS}   *****
${LEI_SSANA}     *****
${LEI_KTUNNUS_KENTTA} /*[@id="Kayttajatunnus"]
${LEI_SSANA_KENTTA} /*[@id="Salasana"]
${LEI_LOGIN_NAPPAIN} //button[@type="submit"]
${LEI_YTUNNUSHAKU} /*[@id="Ytunnus"]
${LEI_SOVELLUS_URL} http://10.132.114.5/LeiYllapito
${MAARA}         1
${MENEKO_LASKUTUKSEEN} ${EMPTY}
${NIMIKE}        2995
${ONKO_EXCEL_AUKI} ${EMPTY}
${ONNIISTUIKO_KIRJAUTUMINEN_LEI} ${EMPTY}
${POSTIOSOITE}   ${EMPTY}
${POSTINUMERO}  ${EMPTY}
${POSTITOIMIPAIKKA} ${EMPTY}
${PRH_NAYTOT_VALIKKO} /*[@id="main-navigation"]/ul/li[6]/a
${SELAIN}        chrome
${SPOSTIN_OTSIKKO} LEI-laskutusapuri on kohdannut virheen
${SPOSTIN_YLEISSISALTO} LEI-laskuja käsittelevä robotti on kohdannut vir-
heen. Ota yhteys järjestelmänvalvojaan. Päivittäistä Exceliä ei ehkä luotu oi-
kein. Tarkista robotin loki C:/Robotin_loki/ tietokoneelta adminsrv3.
${VALITTAJATUNNUS} ${EMPTY}
${VERKKOLASKUOSOITE} ${EMPTY}

```



```

${VIRHEELLINEN} virheellinen
${VIRRE_MUOKKAA_NAPPAIN} /*[@id="_eventId_modifyCustomerBtn"]
${VIRRE_HYVAKSY_NAPPAIN} /*[@id="_eventId_confirmBtn"]
${VIRRE_VAHVISTA_NAPPAIN} /*[@id="_eventId_acceptSaveCustomerBtn"]
${VIRRE_KTUNNUS_KENTTA} /*[@id="inUserName"]
${VIRRE_SSANA_KENTTA} /*[@id="inUserPwd"]
${VIRRE_KTUNNUS} *****
${VIRRE_SSANA} *****
${VIRRE_URL} https://virre.prh.fi/novus/home?execution=els1
${VIRRE_LOGIN_NAPPAIN} /*[@name="Login"]
${VIRRE_LOGOUT_NAPPAIN} /*[@name="Logout"]
${YRITYSHAKUUN_NAPPAIN} /*[@href="companySearch"]
${YRITYSHAUN_TILA} /*[@id="stateCode"]
${YKSIKKOHINTA} 110,00
${YHTEYSHENKILO} ${EMPTY}
${YRITYSHAUN_NIMI} /*[@id="name"]
${YRITYSHAUN_HAKUNAPPAIN} /*[@id="customerSearchCriteria"]/div[*]/div/button[@id="_eventId_searchCustomerBtn"]
${YTUNNUS} ${EMPTY}
${YTUNNUS_LISATIETO} ${EMPTY}

*** Test Cases ***
Virre kirjautuminen
    [Documentation] Avaa selaimen Virren osoitteeseen ja kirjautuu käyttäjätunnuksella sivustolle. Odottaa Logout-näppäimen löytyvän sivulta onnistuneen kirjautumisen jälkeen.
    [Tags] virrelogin
    Open Browser ${VIRRE_URL} ${SELAIN}
    Input Text ${VIRRE_KTUNNUS_KENTTA} ${VIRRE_KTUNNUS}
    Input Password ${VIRRE_SSANA_KENTTA} ${VIRRE_SSANA}
    Click Button ${VIRRE_LOGIN_NAPPAIN}
    ${onnistui_login}= Run Keyword And Return Status Wait Until Element Is Visible ${VIRRE_LOGOUT_NAPPAIN}
    Run Keyword If '${onnistui_login}'=='${False}' Virheilmoitus
    Spostin Lahetys Laskuttajarobotin kirjautuminen Virreen epäonnistui LEI-laskutusrobotti ei päässyt kirjautumaan Virreen! Robotin yhteydessä Virreen on jokin ongelma tai käyttäjätunnuksen tiedot ovat vanhentuneet tai vaihdettu. Tarkista onko robotin käyttäjätunnuksen salasana vanhentunut. Jos on, niin salana täytyy resetoida ja syöttää robotille uusi salasana vanhan tilalle.
    ${HELPDESKIN_SPOSTI}
    [Teardown] Run Keyword If Test Failed Close All Browsers # Sulkee selaimen, jos kirjautuminen ei onnistunut

Virre asiakashaku
    [Documentation] Siirtyy Virren yrityshakuun
    [Tags] virre
    Go To ${ASIAKASHALLINTA_URL}
    Wait Until Element Is Visible /*[@class="yrityshaku"]
    Scroll Element Into View ${YRITYSHAUN_TILA}
    Click Element ${YRITYSHAUN_TILA}
    Click Element ${YRITYSHAUN_TILA}/option[@value="pas"]
    Scroll Element Into View ${YRITYSHAUN_HAKUNAPPAIN}
    Click Element ${YRITYSHAUN_HAKUNAPPAIN}

Asiakastietojen kasittely
    [Documentation] Teardownina, eli tämän testitapauksen jälkeen suoritetaan Sulkemistoiminnot -avainsana.
    [Tags] virreasiakastiedot
    ${apumuuttuja}= Tarkista Onko Passiivisia
    Set Suite Variable ${ONKO_EXCEL_AUKI} ${False}
    Set Suite Variable ${JATKETAANKO} ${apumuuttuja}
    Run Keyword If '${JATKETAANKO}'=='${True}' Avaa Excel Kirjoitukseen
    Run Keyword If '${JATKETAANKO}'=='${True}' Avaa Selain Leisovellukseen Ja Kirjau
    : FOR ${asiakastapaus} IN RANGE 1 1000

```

```

\ Exit For Loop If      '${JATKETAANKO}' == '${False}'
\ ${apumuuttuja}= Kasittely Looppi
\ Set Suite Variable   ${ONKO_EXCEL_AUKI}   ${True}
\ Set Suite Variable   ${JATKETAANKO}   ${apumuuttuja}
\ Exit For Loop If      '${JATKETAANKO}' == '${False}'
Log Ei löytynyt enempää passiivisia asiakkaita, lopetetaan suoritus
[Teardown] Sulkemistoiminnot

```

*** Keywords ***

Asiakkaan Avaus Muokkaukseen

```

[Tags] virremuokkaus virre
Page Should Contain Asiakashaku
Click Element /*[@id="customerSearchResults"]/tbody/tr[1]/td[1]/a
Wait Until Page Contains Asiakkaan tiedot
Scroll Element Into View ${VIRRE_MUOKKAA_NAPPAIN}
Click Button ${VIRRE_MUOKKAA_NAPPAIN}
Wait Until Page Contains Asiakkaan muokkaaminen

```

Asiakastietojen Kopsaus Virresta

[Documentation] Ensin kopioidaan asiakastiedot Virrestä ja käsitellään ne, sitten suoritetaan lei-sovellustoiminnot. Sieltä palataan takaisin tähän avainsanaan, joka kirjaa asiakkaan laskutusexceliin, jos \${MENEEOO_LASKUTUKSEEN} on True.

```

[Tags] asiakastiedot excelin taytto virre
${ytunnus_apu}= Get Element Attribute /*[@id="customerInstance.ycode"]@value
Set Suite Variable ${YTUNNUS} ${ytunnus_apu}
${ASIAKKAAN_NIMI} Get Element Attribute /*[@id="customerInstance.name"]@value
${ASIAKKAAN_TARKENNE} Get Element Attribute /*[@id="customerInstance.extension"]@value
${asiakastunnus_apu} Get Element Attribute /*[@id="customerInstance.insignia"]@value
Set Suite Variable ${ASIAKASTUNNUS} ${asiakastunnus_apu}
${ASIAKASRYHMA} Get Text /*[@id="customerInstance.groupCode"]/option[@selected="selected"]
${ASIAKKAAN_KIELI} Get Text /*[@id="customerInstance.languageCode"]/option[@selected="selected"]
${POSTIOSOITE} Get Element Attribute /*[@id="customerInstance.contactStreetAddress"]@value
${POSTINUMERO} Get Element Attribute /*[@id="customerInstance.contactZipCode"]@value
${POSTITOIMIPAIKKA} Get Element Attribute /*[@id="customerInstance.contactCity"]@value
Comment Kielikoodin ehdot
${KIELIKOODI} Set Variable If '${ASIAKKAAN_KIELI}' == 'suomi' FI
'${ASIAKKAAN_KIELI}' == 'ruotsi' SV '${ASIAKKAAN_KIELI}' == 'englanti'
... EN '${ASIAKKAAN_KIELI}' != 'ruotsi' and '${ASIAKKAAN_KIELI}' != 'englanti' and '${ASIAKKAAN_KIELI}' != 'suomi' EN
${ALV_TUNNUS} Get Element Attribute /*[@id="customerInstance.eInvoicingVatId"]@value
${VERKKOLASKUOSOITE} Get Element Attribute /*[@id="customerInstance.eInvoicingAddress"]@value
Comment Asiakasryhmän ehdot, joiden mukaan määräytyy asiakasnumero
${ASIAKASNRO} Set Variable If '${ASIAKASRYHMA}' == 'Elinkeinoelämä' and
'${VERKKOLASKUOSOITE}' == '' 99000001 '${ASIAKASRYHMA}' == 'Elinkeinoelämä' and
'${VERKKOLASKUOSOITE}' != '' 99000002 '${ASIAKASRYHMA}' == 'Kunnat ja kuntaliittymät' and
'${VERKKOLASKUOSOITE}' == ''
... 99000003 '${ASIAKASRYHMA}' == 'Kunnat ja kuntaliittymät' and
'${VERKKOLASKUOSOITE}' != '' 99000004 '${ASIAKASRYHMA}' == 'Muut (esim. säätiöt ja yhdistykset)' and
'${VERKKOLASKUOSOITE}' == '' 99000003
'${ASIAKASRYHMA}' == 'Muut (esim. säätiöt ja yhdistykset)' and '${VERKKOLASKUOSOITE}' != ''

```

```

... 99000004 '${ASIAKASRYHMA}' != 'Muut (esim. säätiöt ja yhdistyk-
set)' and '${ASIAKASRYHMA}' != 'Kunnat ja kuntaliittymät' and '${ASIAKASRY-
HMA}' != 'Elinkeinoelämä' ${VIRHEELLINEN}
${VALITTAJATUNNUS} Get Element Attribute /*[@id="customerIns-
tance.eInvoicingIntermediatorId"]@value
${YHTEYSHENKILO} Get Element Attribute /*[@id="customerIns-
tance.contactPersonName"]@value
Comment HUOM päivämäärää tarvitaan myös laskun hinnan määrittämiseen
${ASIAKKAAN_LISATIEDOT} Get Text /*[@id="customerInstance.additio-
nInfo"]
${ytunnus_kopio} Set Variable ${YTUNNUS}
Set Suite Variable ${YTUNNUS_LISATIEETO} ${YTUNNUS}
Run Keyword If '${ASIAKKAAN_LISATIEDOT}' != '' Set Suite Variable
${YTUNNUS} ${ASIAKKAAN_LISATIEDOT}
Comment TÄ,,HÄ,,N LEIKÄ,,SITTELY
Run Keyword If '${ASIAKASNRO}' != '${VIRHEELLINEN}' Leisovellus Toi-
minnot ${ONNISTUIKO_KIRJAUTUMINEN_LEI}
Switch Browser 1
Run Keyword If '${ASIAKASNRO}' == '${VIRHEELLINEN}' Virheilmoitus
Spostin Lahetys LEI-laskutusrobotti havaitsi poikkeustapauksen
${ASIAKASTUNNUS}, ${ytunnus_kopio}: Asiakasryhmäksi Virren asiakashallin-
nassa on valittu Kotitaloudet tai Valtion virasto tai laitos eikä laskutusex-
celille voida muodostaa asiakasnumeroa. Asiakasryhmä voi olla vain Elinkei-
noelämä/Kunnat tai kuntayhtymät/Muut (esim. säätiöt tai yhdistykset). Asia-
kasta ei lisätty laskutusexceliin. Asiakas on statukseltaan aktiivinen Vir-
ressä. ${LASKUTUKSEN_SPOSTI}
... ELSE IF '${MENEKO_LASKUTUKSEEN}' == '${False}' Log Asia-
kas ei mene laskutukseen. Käsitellään seuraava asiakas...
... ELSE Uusi Asiakas ${ASIAKASNRO} ${ASIAKKAAN_NIMI} ${ASI-
AKKAAN_TARKENNE} ${POSTIOSOITE}
... ${POSTINUMERO} ${POSTITOIMIPAikka} ${KIELIKOODI} ${ytunnus_kopio}
${ALV_TUNNUS} ${VERKKOLASKUOSOITE}
... ${VALITTAJATUNNUS} ${YHTEYSHENKILO} ${YKSIKKOHINTA}
Comment Log Variables

```

Tarkista Onko Passiivisia

[Documentation] Tarkistaa löytyykö Virren asiakashaun tuloksena yhtään asiakasta. Palauttaa suorituksensa jälkeen tuloksen true tai false sen mukaan, jos asiakkaita löytyy tai ei.

```

[Tags] passiivisten tarkistus virre
${onkolisaa}= Run Keyword And Return Status Wait Until Page Contains
Hakutulokset timeout=15
Run Keyword If '${onkolisaa}' == '${False}' Page Should Contain Element
/*[@id="errors"]
[Return] ${onkolisaa}

```

Virre Asiakkaan Aktivointi

[Documentation] Tekee tarpeelliset toimet Virressä asiakkaan statuksen muuttamiseksi aktiiviseksi.

```

[Tags] statuksen vaihto virre
Scroll Element Into View /*[@id="customerInstance.stateCode"]
Click Element /*[@id="customerInstance.stateCode"]
Click Element /*[@id="customerInstance.stateCode"]/option[@va-
lue="akt"]
Tallenna Muutokset Virreen

```

Tallenna Muutokset Virreen

[Documentation] Painaa Virressä 'Hyväksy' ja 'Hyväksy', jotta muutokset tallentuu Virren asiakastietoihin.

```

[Tags] virrevahvista ja hyväksy
Scroll Element Into View ${VIRRE_HYVAKSY_NAPPAIN}
Click Button ${VIRRE_HYVAKSY_NAPPAIN}
Wait Until Page Contains Asiakkaan tiedot
Scroll Element Into View ${VIRRE_VAHVISTA_NAPPAIN}
Click Element ${VIRRE_VAHVISTA_NAPPAIN}

```

Wait Until Page Contains Asiakashaku

Kasittely Looppi

```
[Tags] for loop asiakkaiden kasittely
Switch Browser 1
Asiakkaan Avaus Muokkaukseen
Virre Asiakkaan Aktivointi
Asiakkaan Avaus Muokkaukseen
Asiakastietojen Kopsaus Virresta
Tallenna Muutokset Virreen
Scroll Element Into View ${YRITYSHAUN_HAKUNAPPAIN}
Click Element ${YRITYSHAUN_HAKUNAPPAIN}
${apumuuttuja}= Tarkista Onko Passiivisia
Set Suite Variable ${JATKETAANKO} ${apumuuttuja}
Run Keyword If '${JATKETAANKO}'=='${True}' Scroll Element Into View
/*[@id="search-result"]/div
Run Keyword If '${JATKETAANKO}'=='${True}' Wait Until Element Is Visible
/*[@id="customerSearchResults"]
[Return] ${JATKETAANKO} # Palauttaa True, jos jatketaan. Palauttaa
False, jos haku antaa lisää asiakkaita käsiteltäväksi.
```

Leisovellus Ytunnushaku

```
[Tags] lei-sovellus haku
Switch Browser 2
Go Tohttp://10.132.114.5/LeiYllapito/Etusivu
Wait Until Element Is Visible ${LEI_YTUNNUSHAKU}
Input Text ${LEI_YTUNNUSHAKU} ${YTUNNUS}
Click Button //button[@type="submit"]
Comment /*[@class="input-group"]//button
${onnistuiko_haku}= Run Keyword And Return Status Wait Until Page
Contains Tunnustiedot timeout=40
Run Keyword If '${onnistuiko_haku}'=='${True}' Leisovellus Avaa Muokkaus
... ELSE Jos Haku Epaonnistuu
[Teardown]
[Return] ${onnistuiko_haku} # Arvona false, jos haku ei onnistu. Ja
true, jos haku onnistuu. Lähetä sähköpostilla Y-tunnus, jos sillä ei löytynyt
lei-sovelluksesta asiakasta.
```

Leisovellus Avaa Muokkaus

```
[Tags] lei-sovellus asiakastunnus tallennus
Switch Browser 2
Go Tohttp://10.132.114.5/LeiYllapito/Tunnukset/Muokkaa?toimepide=muokkaus
${asiakastunnus_kentan_sisalto}= Get Element Attribute
/*[@id="Asiakastunnus"]@value
Input Text ${ASIAKASTUNNUS_KENTTA_LEI} ${ASIAKASTUNNUS}
Run Keyword If '${asiakastunnus_kentan_sisalto}'=='${True}' Leisovellus Asia-
kastunnus Lisays
... ELSE Jos Asiakkaalla On Jo Asiakastunnus
```

Sulkemistoiminnot

```
[Tags] excelin lahetys excelin tallennus uloskirjautuminen se-
lainten sulku
Run Keyword If '${ONKO_EXCEL_AUKI}'=='${True}' Sulje Ja Tallenna Excel
Run Keyword If '${ONKO_EXCEL_AUKI}'=='${True}' Palkeisiin Lahetys
Click Button ${VIRRE_LOGOUT_NAPPAIN}
Wait Until Element Is Visible ${VIRRE_LOGIN_NAPPAIN}
Run Keyword If '${ONKO_EXCEL_AUKI}'=='${True}' Leisovellus Uloskirjautu-
minen Ja Selaimen Sulku
Close Firefox
Switch Browser 1
Close Browser
```

Virheilmoitus Spostin Lahetys

```

[Arguments]      ${postin_otsikko}=${SPOSTIN_OTSIKKO}    ${postin_si-
salto}=${SPOSTIN_YLEISSISALTO}    ${postin_saaaja}=kehittajanosoite    # Ar-
gumenttina annetaan lähetettävän sähköpostin sisältö. Jos ei anneta, niin lä-
hettää viestin oletussisällöllä.
[Tags]           virheilmoitus    email    sahkopostin    lahetys
Laheta Email    ${LAHETTAJA}    ${postin_saaaja}    ${postin_otsikko}    ${pos-
tin_sisalto}

Leisovellus Toiminnot
[Arguments]      ${onnistuiko_kirjautuminen}
[Tags]           kirjautuminen    lei-sovellus
Run Keyword If  '${onnistuiko_kirjautuminen}'=='${True}'    Leiso-
vellus Ytunnushaku
[Teardown]

Leisovellus Uloskirjautuminen Ja Selaimen Sulku
[Tags]           lei-sovellus    lei-sovellus    uloskirjautuminen
Switch Browser  2
Go Tohttp://10.132.114.5/LeiYllapito/Kirjautuminen/KirjauDUlos
Wait Until Element Is Visible    ${LEI_LOGIN_NAPPAIN}
Comment        Close Browser    Close Browser    firefoxin bugin takia aiheuttaa
firefoxin kaatumisen. Versiolla 52.6.0
Close Firefox

Avaa Selain Leisovellukseen Ja KirjauDU
[Tags]           lei-sovellus    kirjautuminen
Open Browser    ${LEI_SOVELLUS_URL}    firefox
Input Text     ${LEI_KTUNNUS_KENTTA}    ${LEI_KTUNNUS}
Input Password ${LEI_SSANA_KENTTA}    ${LEI_SSANA}
Click Button   ${LEI_LOGIN_NAPPAIN}
${onnistuiko_login}= Run Keyword And Return Status    Wait Until Element
Is Visible ${LEI_YTUNNUSHAKU}
Run Keyword If '${onnistuiko_login}'=='${False}'    Virheilmoitus
Spostin Lahetys Laskuttajarobotin kirjautuminen LEI-ylläpitosovellukseen epä-
onnistui    LEI-laskutusrobotti ei päässyt kirjautumaan LEI-ylläpitosovelluk-
seen! Robotin yhteydessä LEI-sovellukseen on jokin ongelma tai käyttäjätunnuk-
sen tiedot ovat vanhentuneet tai vaihdettu. Tarkista onko robotin käyttäjätun-
nuksen salasana vanhentunut. Jos on, niin salasana täytyy resetoita ja syöttää
robotille uusi salasana vanhan tilalle.    ${HELPDESKIN_SPOSTI}
Run Keyword If '${onnistuiko_login}'=='${False}'    Fail    LEI-
sovellukseen kirjautuminen ei onnistunut! Virheilmoitus lähetetty asiasta säh-
köpostilla.
Set Suite Variable    ${ONNISTUIKO_KIRJAUTUMINEN_LEI}    ${onnistuiko_lo-
gin}

exceldebug
[Documentation] Tätä avainsanaa käytettiin vain excel-tiedoston luomisen
debuggaamiseen.
... Avaa excel-tiedoston kirjoitukseen, kirjoittaa sinne kaksi asiakas-
tapausta (joilla vain ykkösiä ja kakkosia kenttien arvoina). Sulkee ja tallen-
taa tiedoston, jonka se lähettää lopuksi palkeisiin.
[Tags]           debug
Set Log Level    TRACE
Avaa Excel Kirjoitukseen
Uusi Asiakas    1    1    1    1    1    1
... 1    1    1    1    1    1
... 1
Uusi Asiakas    2    2    2    2    2    2
... 2    2    2    2    2    2
... 2
Sulje Ja Tallenna Excel
Laheta Palkeisiin

Jos Yritystä Ei Loydy
[Tags]           lei-sovellus    virheposti

```

```

Virheilmoitus Spostin Lahetys LEI-laskutusrobotti havaitsi poikkeusta-
pauksen ${ASIAKASTUNNUS}, ${YTUNNUS}: Y-tunnuksella ei löytynyt tietoja LEI-
sovelluksesta, koska Y-tunnusta ei ole rekisteröity millekään yritykselle.
TARKISTA Y-TUNNUS! Asiakasta ei lisätty laskutusexceliin. Asiakas on statuk-
seltaan aktiivinen Virressä. ${LASKUTUKSEN_SPOSTI}

```

Jos Yrityksen Hakemus Puuttuu

```

[Tags] lei-sovellus virheposti
Virheilmoitus Spostin Lahetys LEI-laskutusrobotti havaitsi poikkeusta-
pauksen ${ASIAKASTUNNUS}, ${YTUNNUS}: Yritykselle ei ole myönnetty LEI-tun-
nusta. TARKISTA Y-TUNNUS! Mikäli Y-tunnus on syötetty oikein, LEI-tunnus on
jäänyt rekisteröimättä. LEI-tunnus tulee rekisteröidä ennen hakemuksen laskut-
tamista. Asiakasta ei lisätty laskutusexceliin. Asiakas on statukseltaan ak-
tiivinen Virressä. ${LASKUTUKSEN_SPOSTI}

```

Jos Haku Epaonnistuu

```

[Documentation] Jos yrityksen haku epäonnistuu lei-sovelluksessa, niin
otetaan virheteksti ylös (jos semmoinen on) ja tehdään toimenpiteitä sen pe-
rusteella. Mitään tähän päätyvää asiakasta ei kirjata laskutusexceliin.
[Tags] lei-sovellus virheposti haarautumislogiikka lasku-
tuspaatos
${haun_virheteksti} Set Variable ${EMPTY}
Run Keyword And Continue On Failure Ota Virheteksti Talteen
Log Virhetekstin sisältö: ${HAUN_VIRHETEKSTI}
Run Keyword If '${HAUN_VIRHETEKSTI}'=='Yritykselle täytyy tehdä ensin
hakemusilmoitus Operatiivisessa. LEI-tunnuksen myöntäminen ei onnistu tois-
taiseksi ilman sitä.' Jos Yrityksen Hakemus Puuttuu
... ELSE IF '${HAUN_VIRHETEKSTI}'=='Yritystä ei löydy.' Jos
Yritystä Ei Löydy
... ELSE IF '${HAUN_VIRHETEKSTI}'=='Yrityksellä ei ole voimassa ole-
vaa rekisteröintiä.' Jos Yrityksellä Ei Ole Rekisterointia
... ELSE Jos Ytunnus On Virheellinen
Comment SUITE VARIABLE SEN MUKAAN PISTETÄÄ,,Ä,,NKÄ- ASIAKAS LASKUTUSEX-
CELIIN VAI EI
Set Suite Variable ${MENEKO_LASKUTUKSEEN} ${False}

```

Jos Ytunnus On Virheellinen

```

[Tags] lei-tunnus virheposti
Run Keyword If '${YTUNNUS_LISATIETO}'=='Virheilmoitus Spostin Lahetys
LEI-laskutusrobotti havaitsi poikkeustapauksen ${ASIAKASTUNNUS}, ${YTUN-
NUS}: Y-tunnuksella ei löytynyt tietoja LEI-sovelluksesta, koska Y-tunnusta ei
ole rekisteröity millekään yritykselle. TARKISTA Y-TUNNUS! Asiakasta ei li-
sätty laskutusexceliin. Asiakas on statukseltaan aktiivinen Virressä. ${LAS-
KUTUKSEN_SPOSTI}
... ELSE Virheilmoitus Spostin Lahetys LEI-laskutusrobotti havaitsi
poikkeustapauksen ${ASIAKASTUNNUS}, ${YTUNNUS}: Y-tunnuksella ei löytynyt
tietoja LEI-sovelluksesta. TARKISTA VIRREN LISÄ,,TIEDOT-KENTTÄ,,Ä,,N SYÄ-TETTY Y-
TUNNUS! Asiakasta ei lisätty laskutusexceliin. Asiakas on statukseltaan aktii-
vinen Virressä. ${LASKUTUKSEN_SPOSTI}

```

Jos Tallennus Ei Onnistu Lei

[Documentation] Asiakas jatkaa laskutukseen normaalisti poikkeuksesta huolimatta.

```

[Tags] virheposti lei-sovellus
Virheilmoitus Spostin Lahetys LEI-laskutusrobotti havaitsi poikkeusta-
pauksen ${ASIAKASTUNNUS}, ${YTUNNUS}: Asiakasnumeroa ei voitu tallentaa LEI-
ylläpitosovellukseen, koska emoyhtiön tai tytäryhtiön konsernisuhdetiedoissa
on puutteita. ASIAKASTUNNUKSEN VOI TALLENTAA LEI-YLLÄ,,PITOSOVELLUKSEEN SEN
JÄ,,LKEEN, KUN KONSERNISUHDETIEDOT ON PÄ,,IVITETTY. Asiakas lisättiin lasku-
tusexceliin. ${LASKUTUKSEN_SPOSTI}

```

Palkeisiin Lahetys

[Documentation] Yrittää lähettää exceltiedoston palkeisiin ja tarkistaa sen onnistumisen. Jos se epäonnistuu, niin asetetaan testitapaus epäonnistuneeksi. Jos onnistuu, niin lähetetään vahvistussähköposti asiasta oletus sähköpostin saajalle (joka on Teemu Vaittinen).

```
[Tags] excelpalkeet
${onnistuiko_lahetys_palkeisiin}= Run Keyword And Return Status Laheta
Palkeisiin
Run Keyword If '${onnistuiko_lahetys_palkeisiin}'=='${True}' Vir-
heilmoitus Spostin LahetysLEI-tunnisteiden laskutusexcel lähetetty onnistu-
neesti Tänään lähetettiin palkeisiin laskutettavaksi excel tiedosto onnis-
tuneesti. Tiedoston löydät Robotin_loki -kansion alta
... ELSE Fail
```

Leisovellus Asiakastunnus Lisays

[Documentation] Yrittää tallentaa muutokset lei-sovellukseen ja tarkistaa onnistuuko operaatio. Jos se ei onnistu, niin suoritetaan avainsana Jos Tallennus Ei Onnistu Lei. Jos onnistuu, niin jatketaan vain suoritusta.

```
[Tags] lei-sovellus asiakastunnus
Click Element //button[@type="submit"][.//text()='Valmis']
${meniko_muutokset_lapi}= Run Keyword And Return Status Wait
Until Element Is Visible ${LEI_PERUUTA_NAPPAIN}
Go Tohttp://10.132.114.5/LeiYllapito/Tiedot/TallennaTiedot
Comment Go Tohttp://10.132.114.5/LeiYllapito/Tiedot/HylkaaMuutokset
${onnistuiko_tallennus}= Run Keyword And Return Status Wait
Until Page Contains Tunnustiedot timeout=60
Run Keyword If '${onnistuiko_tallennus}'!='${True}' Jos Tallennus Ei
Onnistu Lei
Go Tohttp://10.132.114.5/LeiYllapito/Etusivu
Set Suite Variable ${MENEKO_LASKUTUKSEEN} ${True}
```

Jos Asiakkaalla On Jo Asiakastunnus

```
[Tags] virheposti lei-sovellus
Virheilmoitus Spostin Lahetys LEI-laskutusrobotti havaitsi poikkeusta-
pauksen ${ASIAKASTUNNUS}, ${YTUNNUS}: Y-tunnuksella on jo tallennettu asia-
kasnumero LEI-sovellukseen. TARKISTA Y-TUNNUS! Mikäli Y-tunnus on oikein,
tutki, miksi LEI-ylläpitosovellukseen on jo syötetty asiakasnumero. Asiakasta
ei lisätty laskutusexceliin. Asiakas on statukseltaan aktiivinen Virressä.
${LASKUTUKSEN_SPOSTI}
Set Suite Variable ${MENEKO_LASKUTUKSEEN} ${False}
```

Ota Virheteksti Talteen

```
${haun_virheteksti} Get Text //div[@class="alert alert-danger valida-
tion-summary-errors"]
Set Suite Variable ${HAUN_VIRHETEKSTI} ${haun_virheteksti}
```

Jos Yrityksellä Ei Ole Rekisterointia

```
Virheilmoitus Spostin Lahetys LEI-laskutusrobotti havaitsi poikkeusta-
pauksen ${ASIAKASTUNNUS}, ${YTUNNUS}: Yrityksellä ei ole voimassa olevaa re-
kisterointia. TARKISTA Y-TUNNUS! Asiakasta ei lisätty laskutusexceliin. Asia-
kas on statukseltaan aktiivinen Virressä. ${LASKUTUKSEN_SPOSTI}
```

LEI-laskutusexcel-tiedoston luontiskripti

```
#!/usr/bin/python
# -*- coding: utf8 -*-

import xlwt
import pysftp
from datetime import datetime

workbook = 0
worksheet = 0
vaakarivi = 0
pystyrivi = 0
pvm = datetime.now().strftime('%d%m%Y')
tiedostonimi = '5070_'+pvm+'_LEI-laskutus.xls'

def AvaaExcelKirjoitukseen():
    global vaakarivi
    global pystyrivi
    global workbook
    global worksheet

    #Luo Excel-tiedosto ja lisaa se worksheetiksi

    workbook = xlwt.Workbook(encoding='utf-8')
    worksheet = workbook.add_sheet("myyntitilaustiedosto")

    otsikkotiedot = (
        ["T",
        "Tilauslaji",
        "Myyntiorg.",
        "Jakelutie",
        "Sektor",
        "Myyntitsto",
        "Viitelasku",
        "Asiakasnro",
        "Nimi",
        "Nimi2",
        u"Lähiosoite",
        "PostiNo",
        "Toimipaikka",
        "Maakoodi",
        "Kielikoodi",
        u"Henkilötunnus",
        "Y-Tunnus",
        "ALV-tunnus",
        "OVT-tunnus",
        u"Välittäjä-tunnus",
        "Laskutusasiakasno",
        "Laskutusasiakas nimi",
        "Maksaja-asiakasno",
        "Maksaja-asiakas nimi",
        "Toimitusasiakasno",
        "Toimitusasiakas nimi",
        "Palvelun luontipvm",
        "Hinnoittelu pvm",
        "Laskun pvm",
```



```

        "Maksuehto",
        "Laskuttaja",
        "Asiaviite",
        "Tilausnumero",
        "Sopimuspvm",
        "Sopimusnumero",
        u"Tiliöintiviite",
        u"Työmaa-avain",
        "Poikk. veron maaramaa",
        u"Otsikkoteksti: Otsikkomuistio 1 (tekstilaji 0002) tulostuu ennen
riveja. Huom. Kirjoita teksti katkeamattomasti, eti rivivaihtoja eikä alt+en-
ter painikkeita",
        u"Otsikkoteksti: Maksuperusteteksti (tekstilaji Z000) tulostuu las-
kun loppuun. Huom. Kirjoita teksti katkeamattomasti, ei rivivaihtoja eikä
alt+enter painikkeita",
        "Nimike",
        u"Määrä",
        u"Määryksikköä",
        u"Yksikköhinta",
        "Brutto/Netto",
        "Rivityyppi",
        u"Tulosyksikkö",
        "Nimikkeen nimitysteksti (laskutusaihe)",
        u"Riviteksti: Rivimuistio (tekstilaji 0002), tulostuu laskulle rivin
jälkeen",
        "PRR-osa",
        "TaKP-tili",
        "Suorite",
        "Toiminto",
        "Alue / kunta",
        "Seurantakohde 1",
        "Seurantakohde 2",
        "Laskuliitteet"]
    )

    for otsikko in (otsikkotiedot):
        worksheet.write(vaakarivi, pystyrivi, otsikko)
        pystyrivi += 1

    vaakarivi = 1

def UusiAsiakas(asiakasno, asiakkaannimi, tarkenne, lahiosoite, postino, pos-
titoimipaikka, kielikoodi, ytunnus, alvtunnus, ovtunnus, valittajatunnus, yh-
teyshenkilö, yksikköhinta):
    global vaakarivi
    global pystyrivi
    global workbook
    global worksheet

    asiakastiedot = (
        ["1",
        "Z001",
        "5070",
        "02",
        "01",
        "A108",
        "",
        asiakasno,
        asiakkaannimi[:35].strip(),
        tarkenne[:35].strip(),
        lahiosoite[:35].strip(),
        postino[:10].strip(),

```

```
postitoimipaikka[:35].strip(),
"FI",
kielikoodi[:2].strip(),
"",
ytunnus,
alvtunnus[:20].strip(),
ovttunnus[:18].strip(),
valittajatunnus[:18].strip(),
"",
"",
"",
"",
"",
"",
"",
"",
"",
"",
"",
"",
"507001A108",
yhteyshenkilö[:35].strip(),
"",
"",
"",
"",
"",
"",
"",
"",
"",
"2995",
"1",
"",
yksikkohinta, #mahdollinen virhe, jos ei oteta päivämäärää huomioon
"N",
"",
"",
"",
"",
"",
"",
"",
"",
"",
"",
"",
"",
"",
"",
"",
""]
)

pystyrivi = 0

for tieto in (asiakastiedot):
    worksheet.write(vaakarivi, pystyrivi, tieto)
    pystyrivi += 1

vaakarivi += 1

def SuljeJaTallennaExcel():
    global workbook
    workbook.save(tiedostonimi)

def LahetaPalkeisiin():
    paikallinenpolku = 'C:/Robotin_loki/%s' % tiedostonimi
    etapolku = '/Tulot/%s' % tiedostonimi
    cnopts = pysftp.CnOpts()
```

```
cnopts.hostkeys = None
with pysftp.Connection(host="*****", username="*****", password="*****", cnopts=cnopts) as sftp:
    sftp.put(paikallinenpolku, etapolku)

sftp.close()
```

Sähköpostin lähetysskriptit – lahetaemail.py ja __init__.py

```
# -*- coding: utf8 -*-
#lahetaemail

import smtplib
from email.MIMEMultipart import MIMEMultipart
from email.MIMEBase import MIMEBase
from email.MIMEText import MIMEText
from email import Encoders
import os

class LahetaEmailTyokalu(object):

    ROBOT_LIBRARY_SCOPE = 'Global'

    def __init__(self):
        print u'työkälu, jolla lähetetään sähköpostia'

    def laheta_email(self, lahettaja, vastaanottaja, otsikko, sisälto):
        msg = MIMEMultipart()

        msg[u'From'] = lahettaja.encode('utf-8')
        msg[u'To'] = vastaanottaja.encode('utf-8')
        msg[u'Subject'] = otsikko.encode('utf-8')

        msg.attach(MIMEText(sisälto, "plain", "utf-8"))

        mailServer = smtplib.SMTP("palvelimenosoite", 25)
        mailServer.ehlo()
        mailServer.sendmail(lahettaja, vastaanottaja, msg.as_string())

        mailServer.close()

# -*- coding: utf8 -*-
#__init__.py

from lahetaemail import LahetaEmailTyokalu
__version__ = '0.9'

class EmailLibrary(LahetaEmailTyokalu):

    ROBOT_LIBRARY_SCOPE = 'GLOBAL'
```

Firefox-selaimen sulkemisskripti "close_ff.py"

```
#!/usr/bin/python
# -*- coding: utf8 -*-
import os

def close_firefox():
    os.system('taskkill /f /im geckodriver.exe')
    os.system('taskkill /f /im firefox.exe')
```