

Ohjelmistorobotiikka



Kurssitehtävä

Karelia AMK, Campus Online

12/2021

Irina Norro

Karelia AMK
Campus Online

Tekijä	Irina Norro	Vuosi 2021
Työn nimi	Kurssitehtävä	
Työn ohjaaja	Joni Ranta	

TIIVISTELMÄ

Ohjelmistorobotiikan (5op) kurssin lopputehtäväksi valitsin suunnitella ja koodata ohjelmistorobotin, joka hakee tietojärjestelmästä tietoja annetuilla hakuehdoilla, vie hakutulokset Excel -tiedostoon ja lähettää tiedoston liitteenä sähköpostilla valitulle vastaanottajalle.

Tehtävä tehdään UiPath -ohjelmalla. Ulkopuoliseen tietojärjestelmään kirjaututaan salatuilla tunnuksilla, jotka on tallennettu UiPath Orchestratoriin. Orchestratorissa myös aktivoidaan robotin ajastus ja robotin käynnistys toimii tarvittaessa myös sitä kautta. Koodin lokitiedot tallennetaan omaan Excel -tiedostoon.

Avainsanat Ohjelmistorobotiikka, RPA, automatisointi

SISÄLLYS

1 JOHDANTO	1
2 MUUTUJAT, ARGUMENTIT, AKTIVITEETIT	2
3 ORCHESTRATOR.....	3
4 VAATIMUSMÄÄRITTELY	4
4.1 Asiakastarina.....	4
4.2 Sidosryhmät ja profiilikuvaukset	4
4.3 Asiakaspolku	5
4.4 Tuotteen yleiset vaatimukset	5
4.5 Palveluun liittyvät toiminnalliset vaatimukset	5
4.6 Palveluun liittyvät ei-toiminnalliset vaatimukset	5
5 PALVELUPOLKU.....	6
6 ESIVALMISTELU.....	6
7 ROBOTIN DOKUMENTOINTI	6
7.1 Kirjautuminen sivulle	7
7.2 Hakusanan syöttäminen	7
7.3 Lokitietojen tallennus	8
7.4 Hajautettu robotti	11
7.5 Hakutietojen lähetys sähköpostitse	14
8 DEBUGGAUS/TESTAUS	15
9 HUOMIOITA JA AJATUKSIA	16
LÄHTEET	17

1 JOHDANTO

Tässä kerron vähän tarkemmin tekemästäni kurssityöstä Ohjelmistorobotiikan (5op) kurssille. Tarkoituksena on suunnitella ohjelmistorobotti yhteisön käyttöön, esim. omaan työpaikkaan tai harrastusympäristöön. Tähän kehittämäni ohjelmistorobottiin sain inspiraation lukemastani opinnäytetyöstä.

Mika Anttilan (2020) opinnäytetyössä tutkittiin terveydenhuollon henkilöstön työhyvinvointia ohjelmistorobotin käyttöönoton myötä. Sairaanhoidopiirin munuaispoliklinikan asiantuntijahoitajien työhön kuului aamuisin mm. hakea manuaalisesti potilaiden laboratoriokäyntien vastauksia. Sairaanhoidajat selvittivät, ovatko potilaat käyneet saamansa lähetteen edellyttämässä laboratoriokokeessa. Mikäli potilas ei ole käynyt kokeessa, hänelle varattiin uusi aika tai muistutettiin asiasta. Jos kokeessa on käyty ja koetulos selvillä, potilaan tiedot lähetettiin lääkärille, joka antaa sen jälkeen lausunnon potilaan tuloksista.

Opinnäytetyö painottui tutkimaan laboratoriovastausten automaattisen ja manuaalisen tarkastamisen mielekkyyden vertailua, henkilöstön hyvinvointia ja ohjelmistorobotin soveltuvuutta kyseiseen tehtävään sekä säästettyjä kustannuksia. Robotin käyttöönoton myötä työn mielekkyys hoitajien keskuudessa koheni merkittävästi, kun aikaa ei mennyt joka aamu manuaalisen työn suorittamiseen, vaan aikaa jäi potilaille. Anttilan laskelmien mukaan hoitajilla ajankäytön tarve kyseiseen tehtävään väheni 16 %:iin aiemmasta. Robotin takaisinmaksuaika on noin 5 vuotta.

Tässä tehtävässä kehitän robotin, joka toimii samansuuntaisella, mutta yksinkertaistetulla periaatteella, jota Anttilan tutkimassa työympäristössä käytettiin.

Tässä omassa työssäni ei ollut pääsyä laboratorio- tai potilastietojärjestelmiin, joten tein sovelletun version, joka jäljittelee yksinkertaistaen tätä toimintoa.

Robotti

- kirjautuu verkkokauppaan (ulkopuoliseen tietojärjestelmään)
- hakee tuotteita annetulla hakuehdolla, tässä käytin hintarajauksia hakuehtona (tarkistaa potilaiden laboriotulokset)
- palauttaa tulokset rajauksen mukaan sähköpostiin tiedostoliitteenä.
- Robotti on ajastettu toimimaan joka aamu klo 7.

2 MUUTTUJAT, ARGUMENTIT JA AKTIVITEETIT

Tehtävässä käytetyt muuttujat ja argumentit on nimetty CamelCase muotoon. Muuttujien nimet ovat yksiselitteiset ja tehtäväänsä kuvaavat.

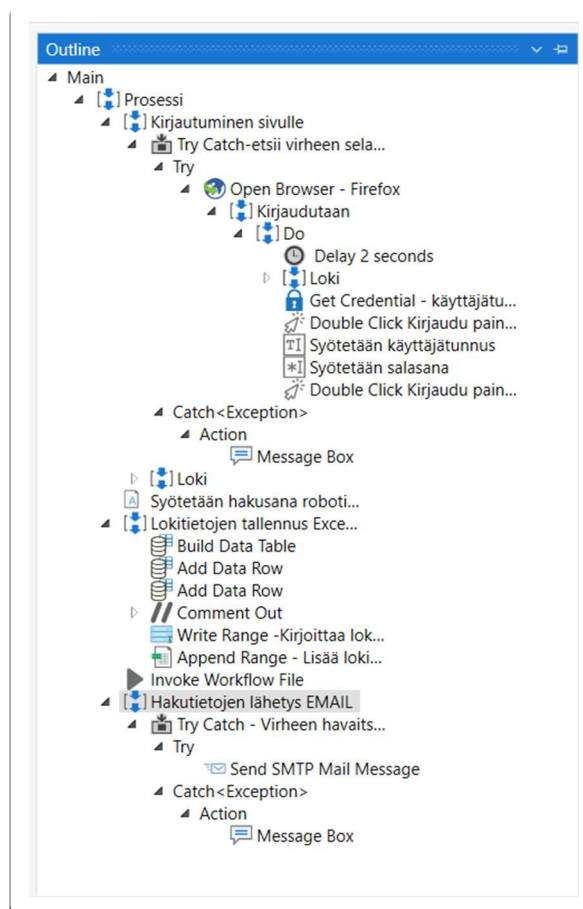
Name	Variable type	Scope	Default
brwSelain	Browser	Haku_kriteerillä	Enter a VB ex
dtTuoteHinta	DataTable	Haku_kriteerillä	Enter a VB ex
Create Variable			

Kuva 1. Muuttujat.

Name	Direction	Argument type	Default valu
argHakusana	In	String	Enter a VB e
Create Argument			

Kuva 2. Argumentit.

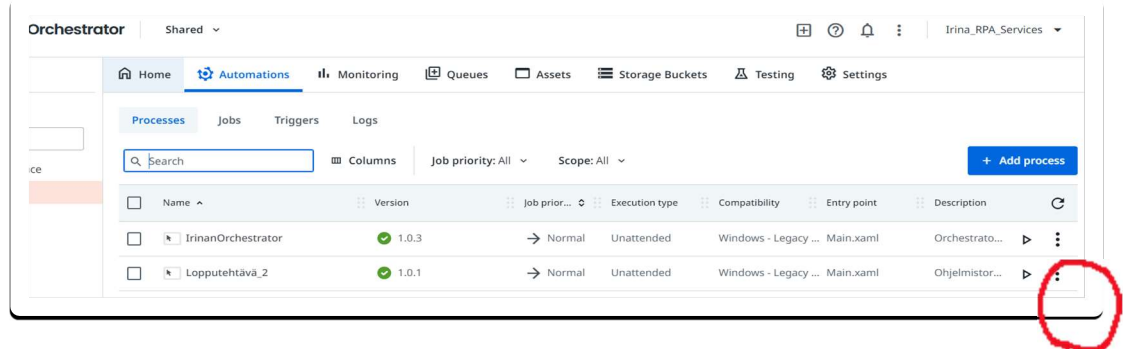
Aktiviteetit on nimetty selkeästi.



Kuva 3. Aktiviteetit.

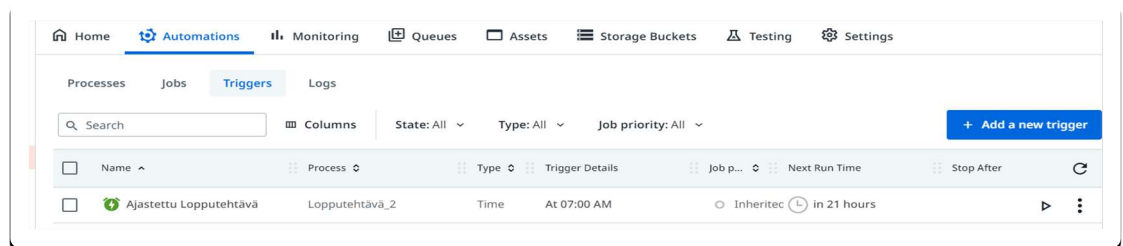
3 ORCHESTRATOR

Robotista on tehty prosessi Orchestratoriin ja sen voi käynnistää myös sieltä.



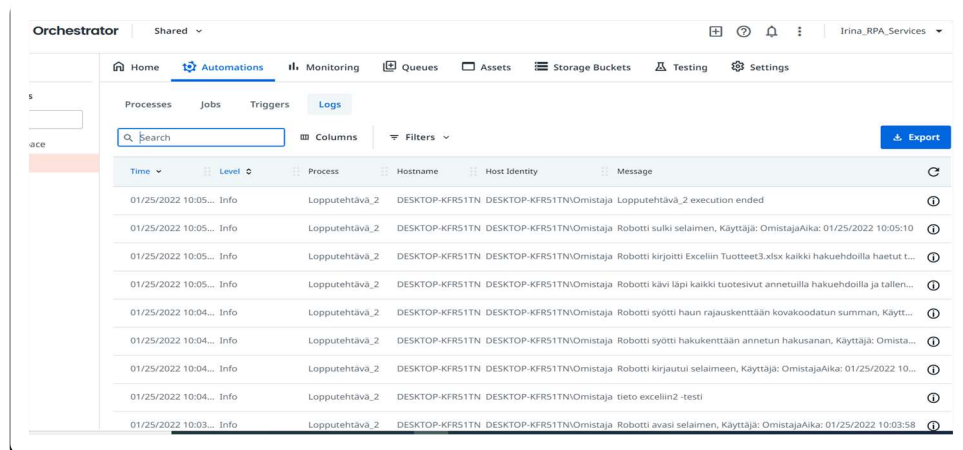
Kuva 4. Orchestrator. Prosessin käynnistys.

Robotti on ajastettu toimimaan aamuisin klo 7. Aikavyöhyke on huomioitu.



Kuva 5. Orchestrator. Ajastettu robotti.

Orchestrator tallentaa myös lokitiedot.



Kuva 6. Orchestrator. Lokitiedot.

4 VAATIMUSMÄÄRITTELY

Tässä käydään lyhyesti läpi, mitä ohjelmistorobotin rakentaminen vaatii vaatimusmäärittelyn tasolla.

4.1 Asiakastarina

Ohjelmistorobotti käynnistyy aamuisin klo 7.00. Robotti hakee hakuehdoilla (testissä käynyt potilas) tietoja järjestelmästä ja tekee siitä yhteenvedon Exceliin. Sairaanhoitaja-asiantuntija tulee töihin, avaa sähköpostilaatikon ja tarkistaa robotin lähettämän yhteenvedon laboratoriotuloksista. Hoitaja kirjautuu potilastietojärjestelmään ja ottaa yhteyttä niihin potilaisiin, jotka eivät vielä käyneet testeissä.

4.2 Sidosryhmät (Stakeholders) ja profiilikuvaukset (Profile descriptions)

Potilas

Munuaispoliklinikan potilas saa lähetteen tarkempiin tutkimuksiin. Potilas käy laboratoriotestissä ja menee sen jälkeen lääkärin vastaanotolle. Lähetteen ja testituloksen kirjautuu järjestelmään.

Henkilökunta

Sairaanhoitaja-asiantuntija kirjoittaa lähetteen potilaille. Hän tarkistaa testeissä käyneiden potilaiden tulokset ja varaa ajan lääkärille. Muihin potilaisiin hän ottaa yhteyttä ja muistuttaa tulevasta testiajasta tai varaa uuden laboratoriaoajan.

Lääkäri antaa lähetemääräyksen munuaisklinikan potilaalle. Ottaa vastaan potilaita ja antaa lausunnon laboratoriotestitulosten perusteella.

Ylläpito

Sairaanhoidopiirin IT-yksikkö ylläpitää sairaalassa käytettäviä ohjelmia, joihin henkilökunta kirjautuu. Auttaa tarvittaessa ohjelmien ja tietokoneiden kanssa. Ylläpitää ohjelmistorobotteja ja auttaa tarvittaessa niiden käytössä sekä selvittää vikatilanteita.

Hakkeri

Mustahattuhakkeri saattaa yrittää hakkeroida potilastietojärjestelmän ja selvittää potilaiden tietoja ja myydä niitä eteenpäin tai kiristää potilaita saamillaan tiedoilla.

4.3 Asiakaspolku (Customer Journey)

Ohjelma käynnistyy automaattisesti joka arkipäivä klo 7.00. Ohjelma tarkistaa testitulokset potilastietojärjestelmästä ja vertaa niitä lähetteisiin. Ohjelma lähettää koosteen sairaanhoitajalle. Hoitaja kirjautuu aamulla koneelle ja avaa koosteen sähköpostistaan. Hoitaja ryhtyy toimenpiteisiin koosteen perusteella.

4.4 Tuotteen yleiset vaatimukset

ID	Vaatimuksen kuvaus	Kategoria
001	Ilmoittaa prosessin virheistä lokitiedoissa.	Tekninen
002	Tarkistaa testeissä käyneet potilaat	Tekninen
003	Lähettää yhteenvedon sairaanhoitajalle	Tekninen

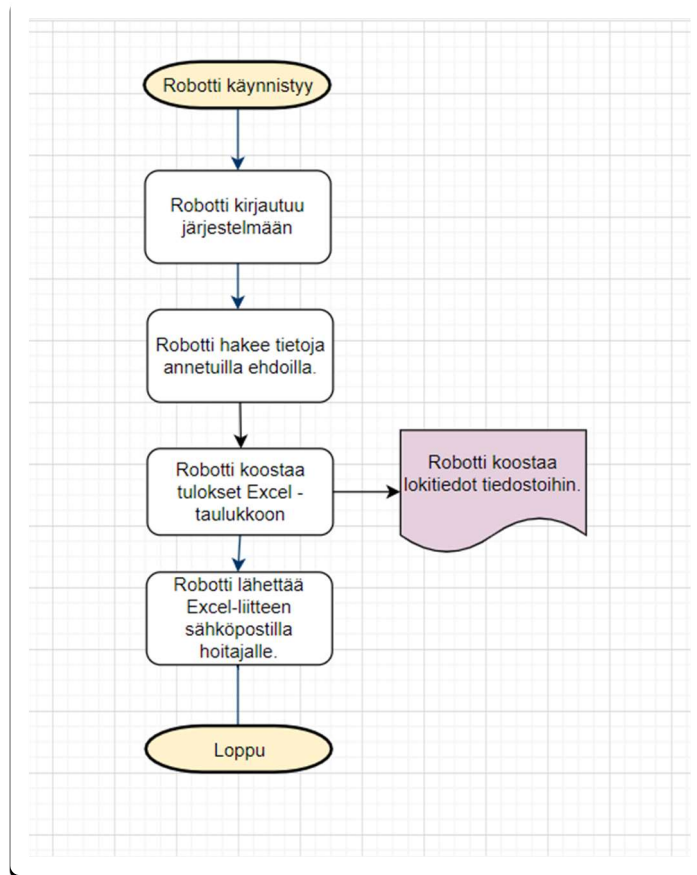
4.5 Palveluun liittyvät toiminnalliset vaatimukset (Functional Requirements)

ID	Vaatimuksen kuvaus	Kategoria
005	Selaimen nopeudelle sopiva toimintanopeus.	Tekninen
006	Yhteys pilvipalveluun (Orchestrator)	Tekninen

4.6 Palveluun liittyvät ei-toiminnalliset vaatimukset (Non-Functional Requirements)

ID	Vaatimuksen kuvaus	Kategoria
007	Hakutulosten täytyy olla varmoja.	Luotettavuus
008	Sähköpostilla lähetetyt potilaiden tiedot täytyy olla salattuja.	Tietoturva
009	Järjestelmä täytyy olla dokumentoitu niin, että pääkäyttäjän on helppo puuttua mahdollisiin virheilmoituksiin.	Käytettävyys

5 PALVELUPOLKU



Kuva 7. Palvelupolku. Vuokaavio.

6 ESIVALMISTELUT

Seuraavat asiat pitää olla tehtynä, jotta ohjelmoimani robotti olisi toiminnassa.

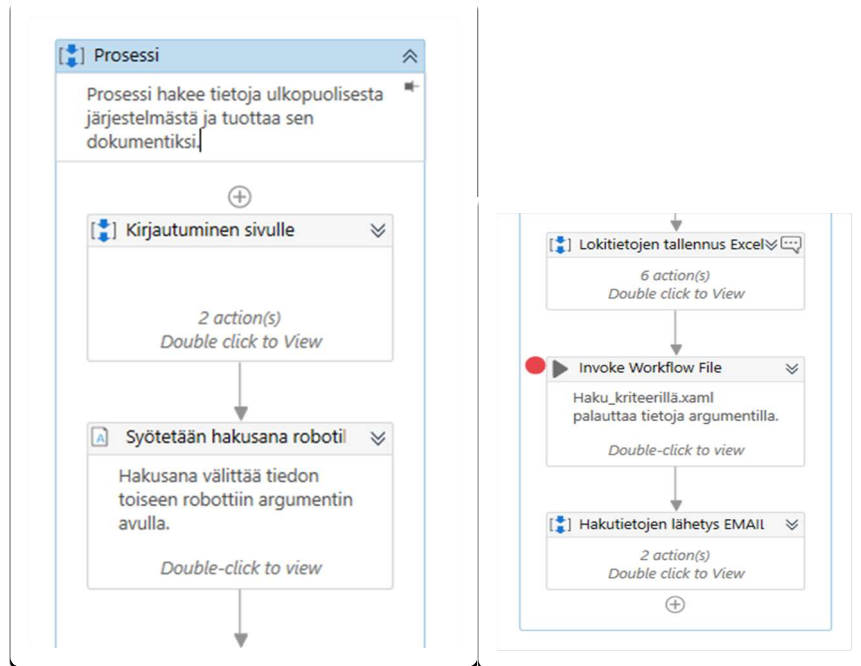
- Yhteys Internetiin ja Uipath asennettuna koneelle.
- Firefox Extension asennettuna UiPathiin.
- Kirjautumistunnukset luotu sivulle jimms.fi
- Orchestratoriin luotu Credential Assets, toimivat käyttäjätunnus ja salasana sivulle jimms.fi

7 ROBOTIN DOKUMENTOINTI

Robotin prosessi tiivistettynä. Robotissa on neljä vaihetta

1. Kirjautuminen sivulle

2. Hakusanan kysyminen käyttäjältä
3. Lokitietojen tallennus Exceliin
4. Hajautettu robotti, Haku_kriteerillä.xaml
5. Hakutietojen lähetys sähköpostitse



Kuva 8. Prosessi 1. Välivaiheet.

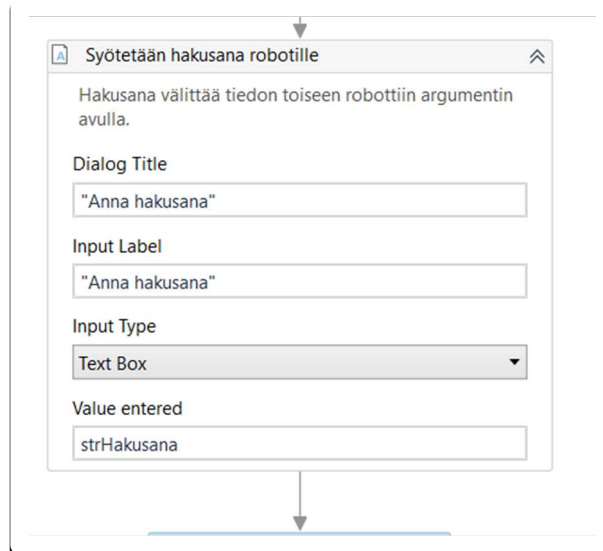
Kuva 9. Prosessi 2. Välivaiheet

7.1 Kirjautuminen sivulle

Kirjautuminen sivulle kuvattu tarkemmin Debuggaus -otsikon alla. Robotti kirjautuu selaimelle. Kirjautumisprosessi on TryCatchin sisällä. Robotti hakee kirjautumiseen tarvittavat käyttäjätunnuksen ja salasanan Orchestratorista. Tallentaa lokitiedot tapahtuneesta.

7.2 Hakusanan syöttäminen

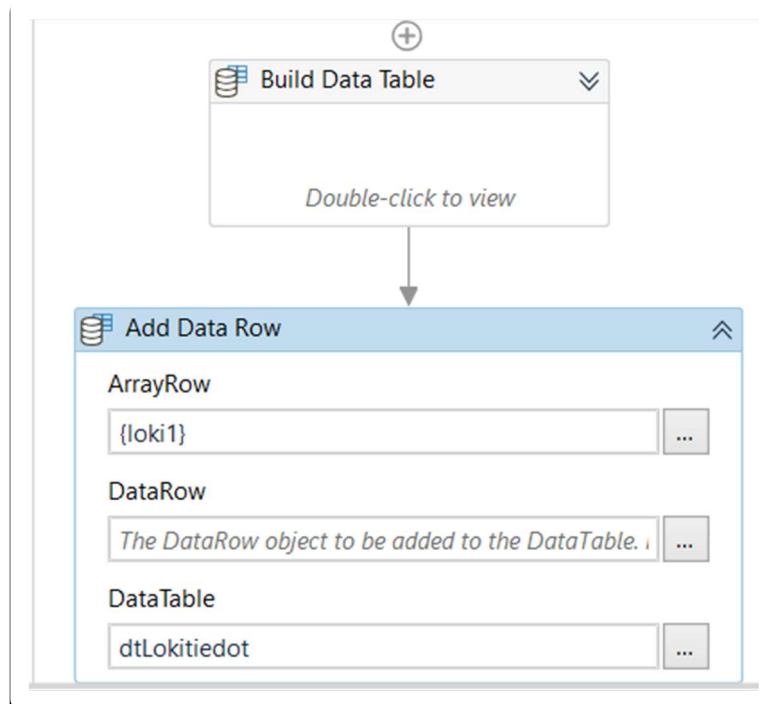
Robotti kysyy käyttäjältä hakusanaa, joka välitetään hajautetulle robotille. Robotti syöttää sen hakukenttään. Hajautetussa robotissa hakusana toimii argumenttina, joka palauttaa tietoa main.xamliin. Argumentti voisi olla joku fiksumpi tapa, joka toimisi itsenäisesti ilman käyttäjän väliintuloa.



Kuva 10. Hakusanan syöttäminen.

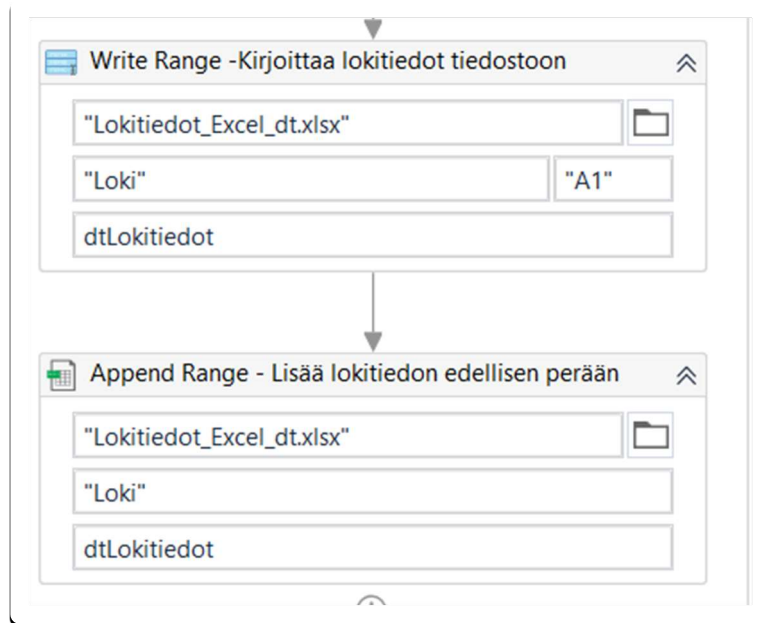
7.3 Lokitietojen tallennus

Tämä prosessi on kuvattuna tarkemmin .xaml -tiedostossa. Tässä lokitiedot tallennetaan .txt- ja .xlsx-muotoon. Lokitieto tallennetaan muuttujaan (loki1, loki2 jne.). Tässä oli tarkoitus viedä lokitieto Exceliin kolmeen eri sarakkeeseen, Käyttäjä, Teksti, Aika käyttämällä mahdollisesti splittausta. Sitä varten rakennettiin DataTable. En saanut tätä tässä muodossa kuitenkaan toimimaan, joten lokitieto tulee Excel-taulukkoon yhtenä rivinä.



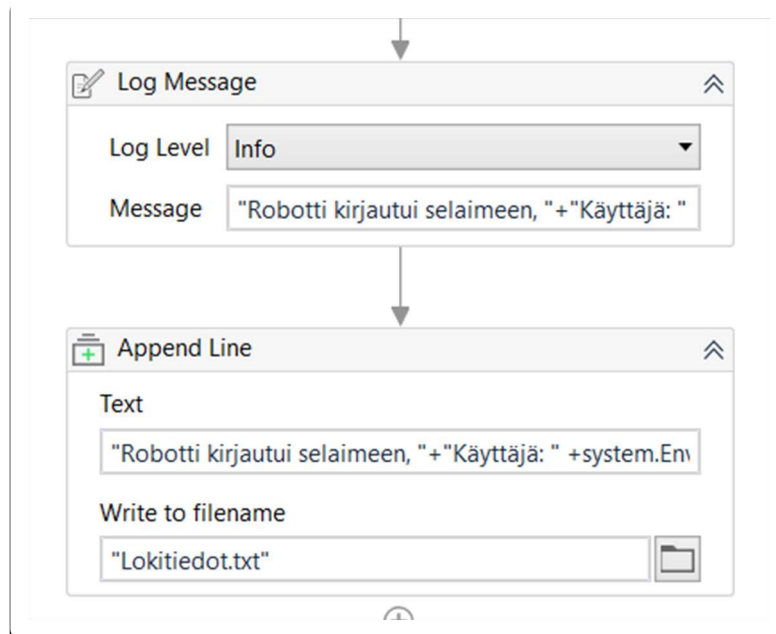
Kuva 11. Lokitiedon vieminen DataTableiin.

Tässä lokitieto vietään Exceliin. Tämä on toiminut siis kokeena, tarkistin, että lokitiedon välittäminen muuttujamuodossa toimii tarvittaessa.



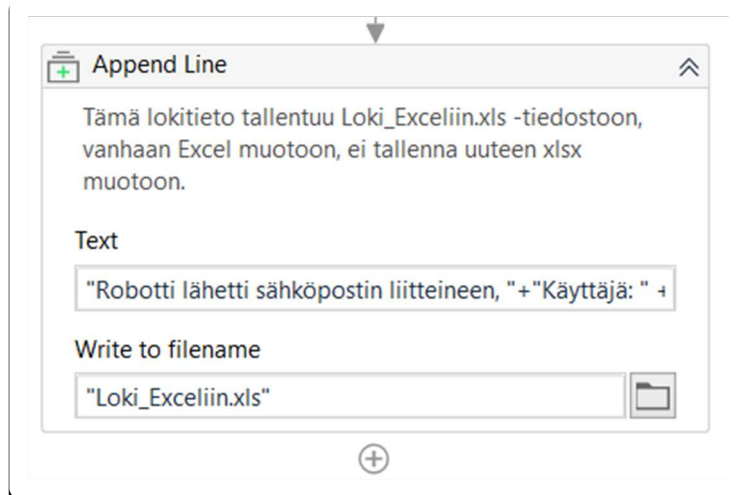
Kuva 12. Lokitiedon tallentaminen Exceliin.

Tein tämän myös toisella, yksinkertaisemmalla tavalla. Tässä lokitiedot vietään yksitellen tiedoston lokitietoihin ja tekstitiedostoon.

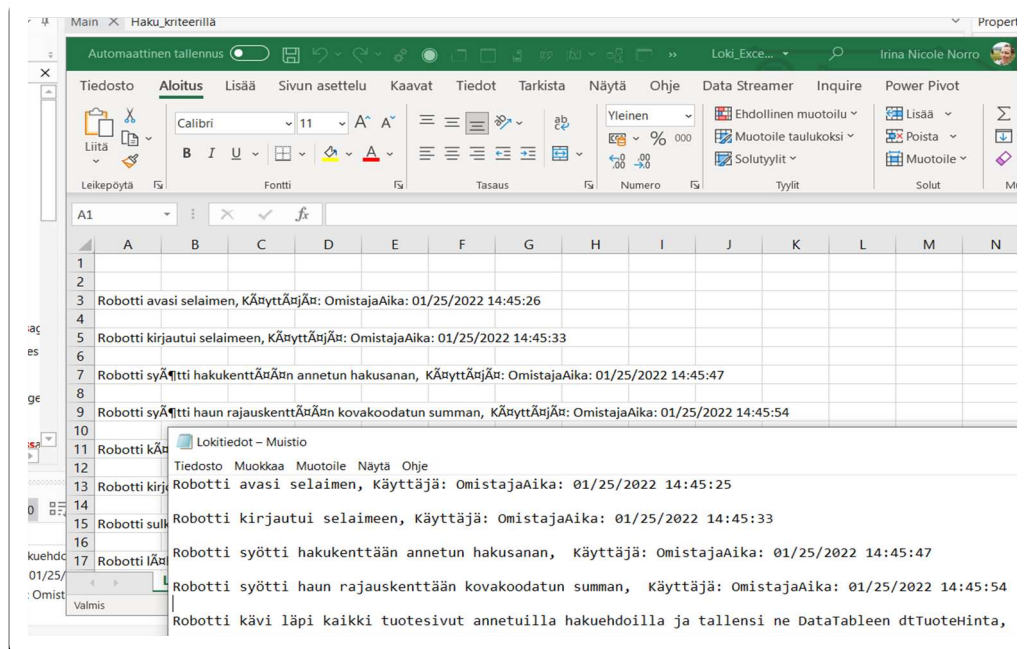


Kuva 13. Lokitiedon tallentaminen. Lokitiedot ja tekstitiedosto.

Lokitiedot voi tallentaa Excel -tiedostoon yksinkertaisella Append Line -aktiviteetilla.



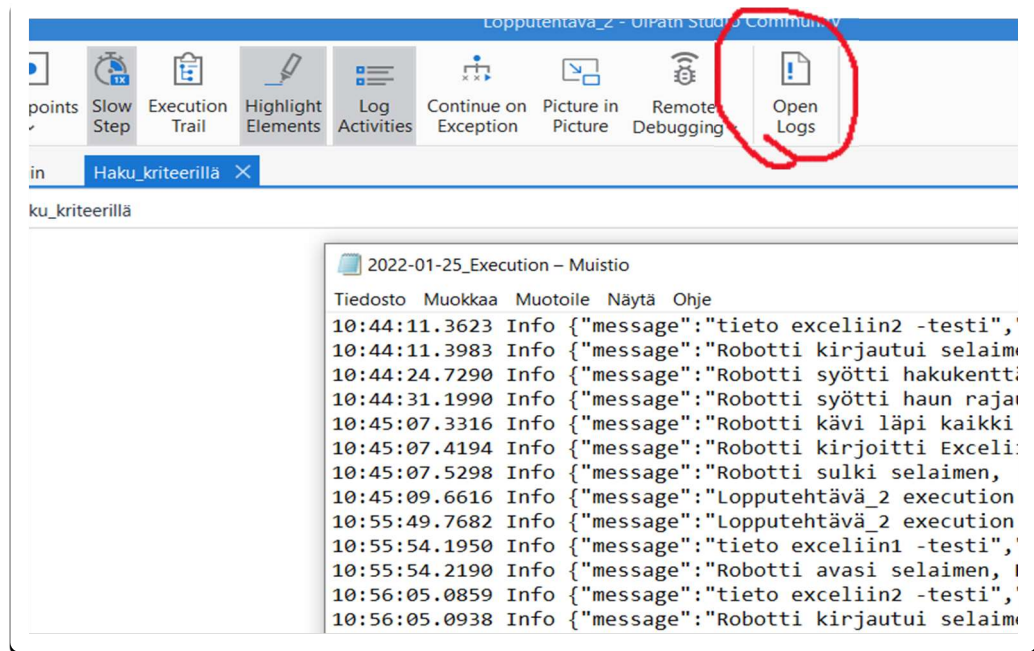
Kuva 14. Lokitiedon tallentaminen. Excel -tiedosto.



Kuva 15. Tallennetut lokitiedot tiedostoissa.

Testasin tässä lopputyössä monia tapoja tallentaa lokitietoja, siksi nämä on tässä neljässä eri muodossa. Jatkossa on yksinkertaisempaa käyttää paria eri tallentamismuotoa, sillä huomasi, että näissä menee helposti sekaisin.

Debug -välilehdellä näkyy myös lokitietoja. Lokitiedot avautuvat Open Logs painikkeesta.

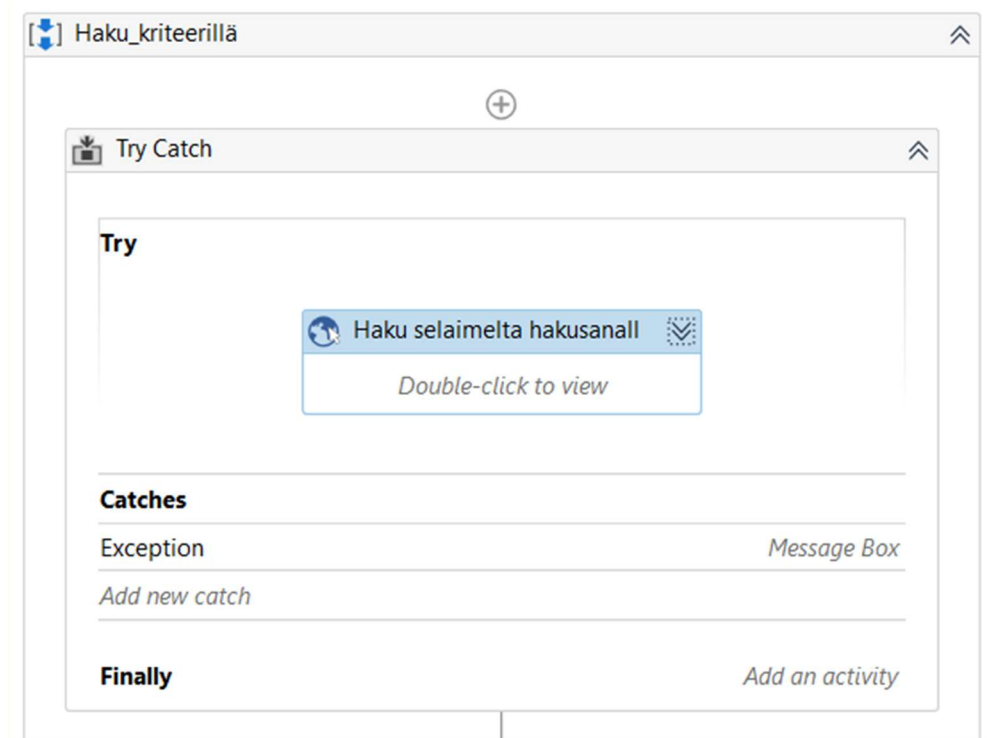


Kuva 16. Lokitiedot. Debug.

ff

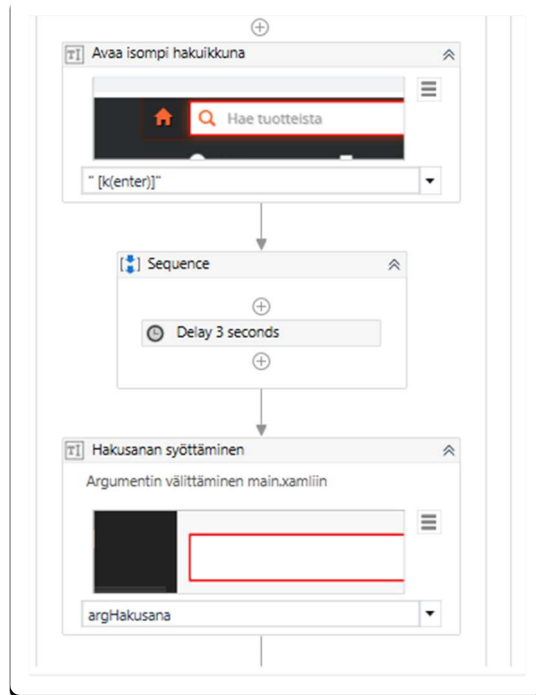
7.4 Hajautettu robotti

Hajautettu robotti on tallennettu nimellä Haku_kriteerillä.xaml. Robotti hakeutuu hakukenttään etsimään käyttäjältä kysytyllä hakusanalla tuotteita. Toiminto on TryCatchin sisällä.



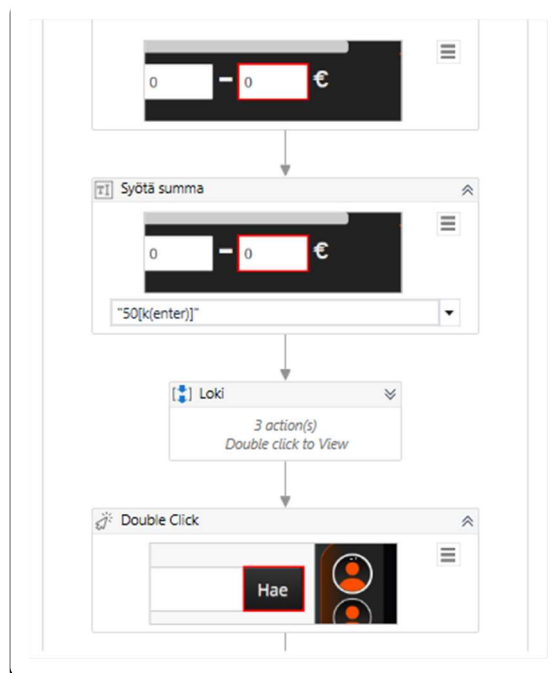
Kuva 17. Hajautettu robotii. Haku_kriteerillä.xaml.

Hakusanaa kysytään käyttäjältä main.xamlin puolella. Hakusana välitetään hajautetulle robotille argumentin avulla. Robotti syöttää käyttäjän antaman hakusanan selaimen hakukenttään.



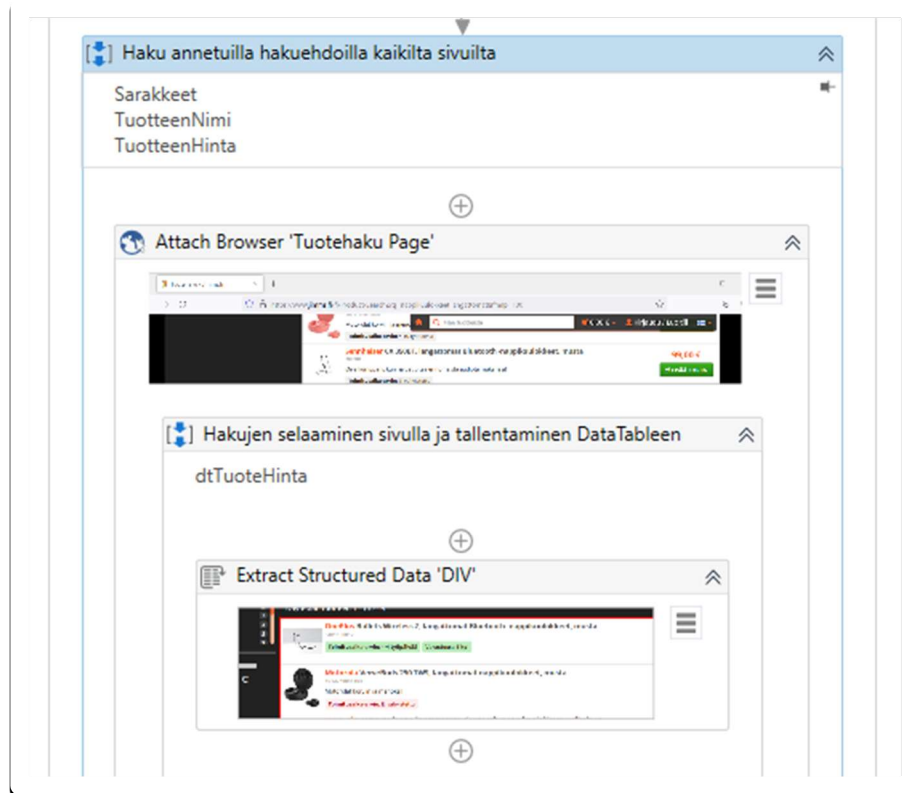
Kuva 18. Hakusanan syöttäminen.

Tässä robotissa halutaan filтерöidä haun tuloksia. Ja se tehdään rajamalla hakuä syöttämällä maximi summa, tässä se on kovakoodattuna (50).



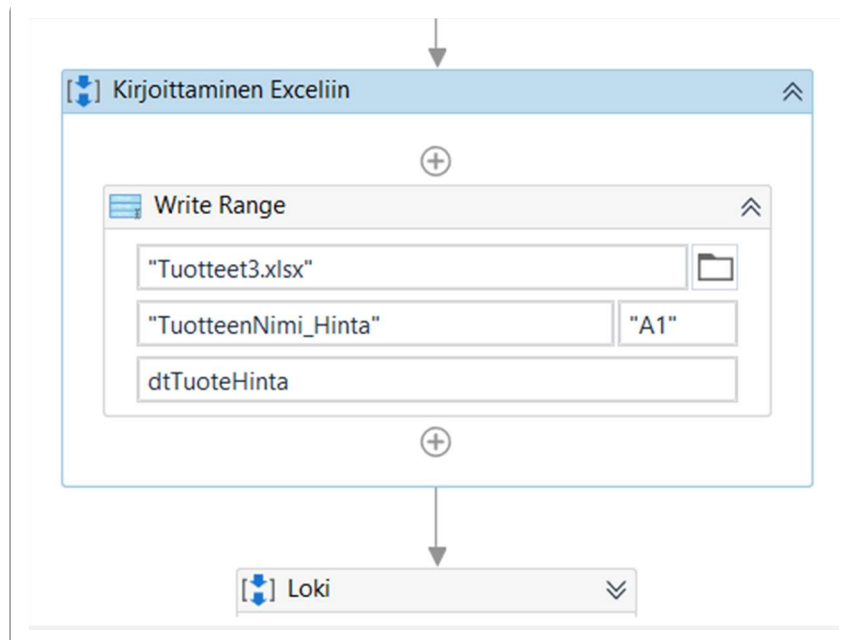
Kuva 19. Haun rajaus.

Haetut tuotteet sijoitetaan DataTableen dtTuoteHinta.



Kuva 20. Haetut tulokset DataTableen.

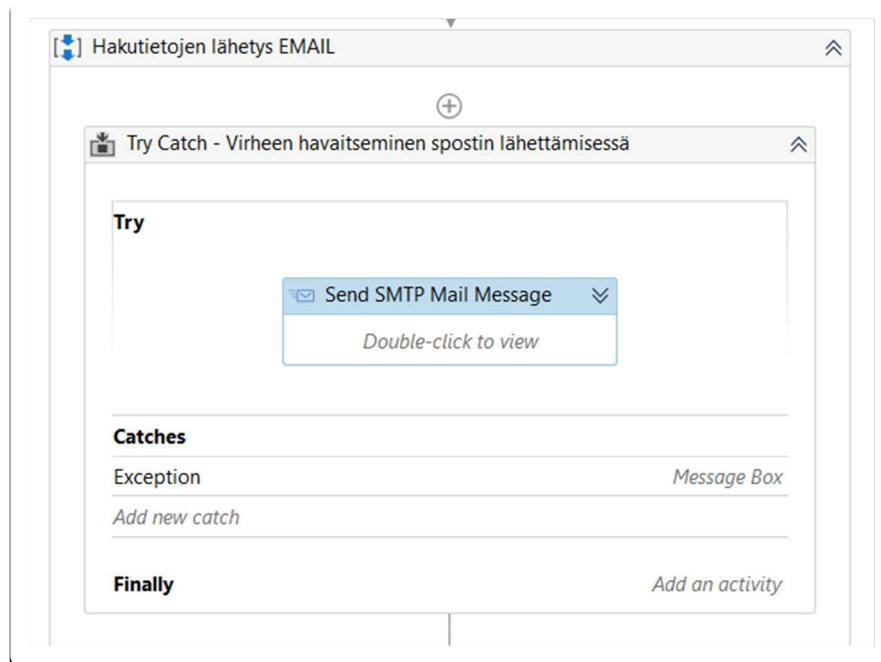
Rajauksella haetut tuotteet kirjoitetaan Exceliin.



Kuva 21. Tulostus Exceliin.

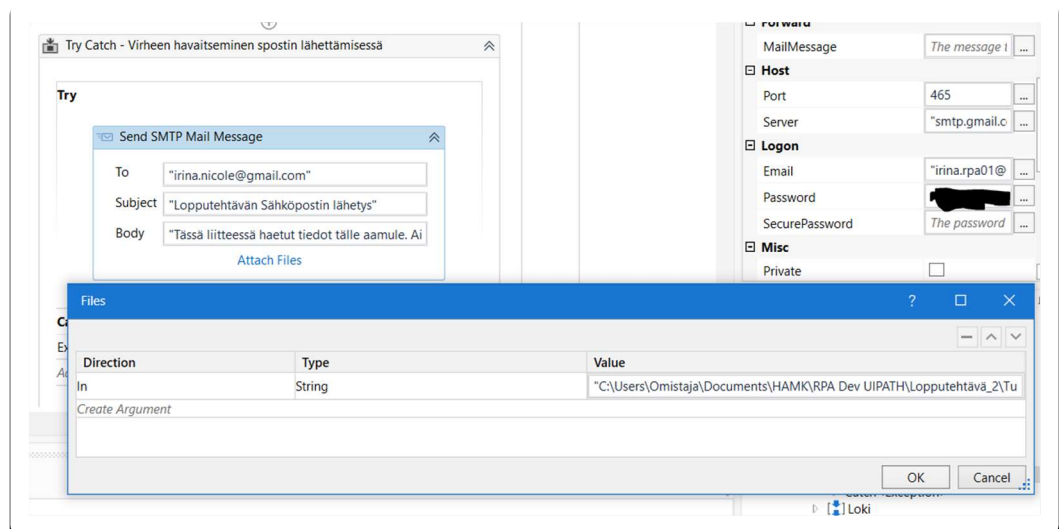
7.5 Hakutietojen lähetys sähköpostitse

Sähköpostin lähetys on TryCatchin sisällä.



Kuva 22. Sähköpostin lähetys.

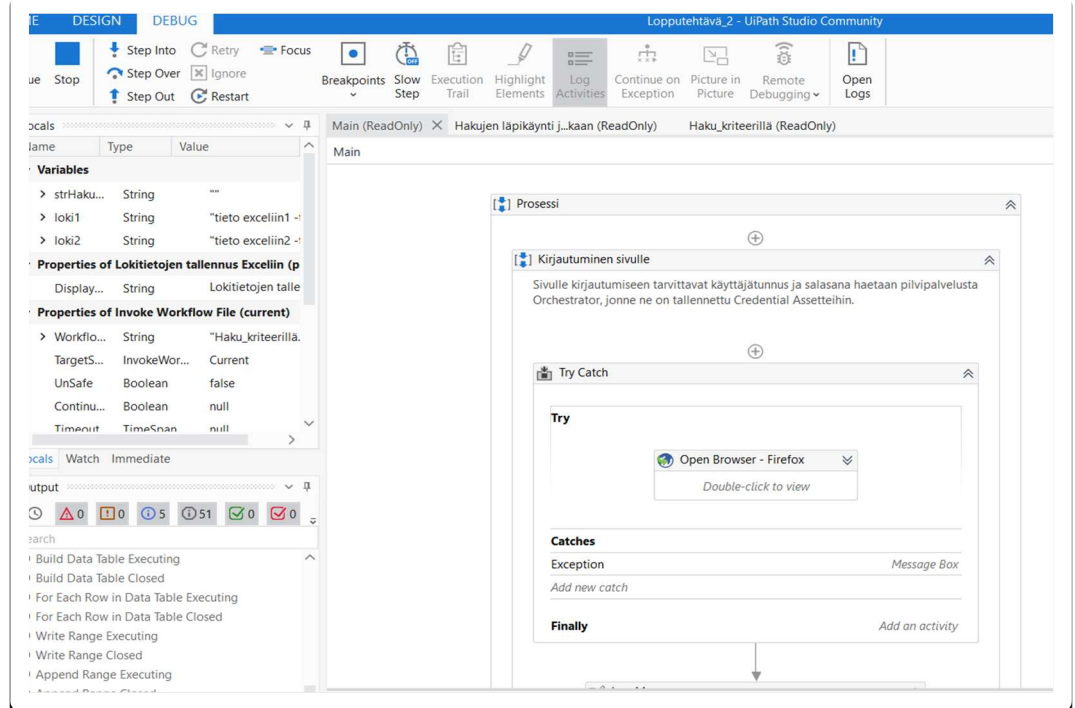
Sähköpostin mukana lähetetään liite.



Kuva 23. Sähköpostin lähetys. Liite

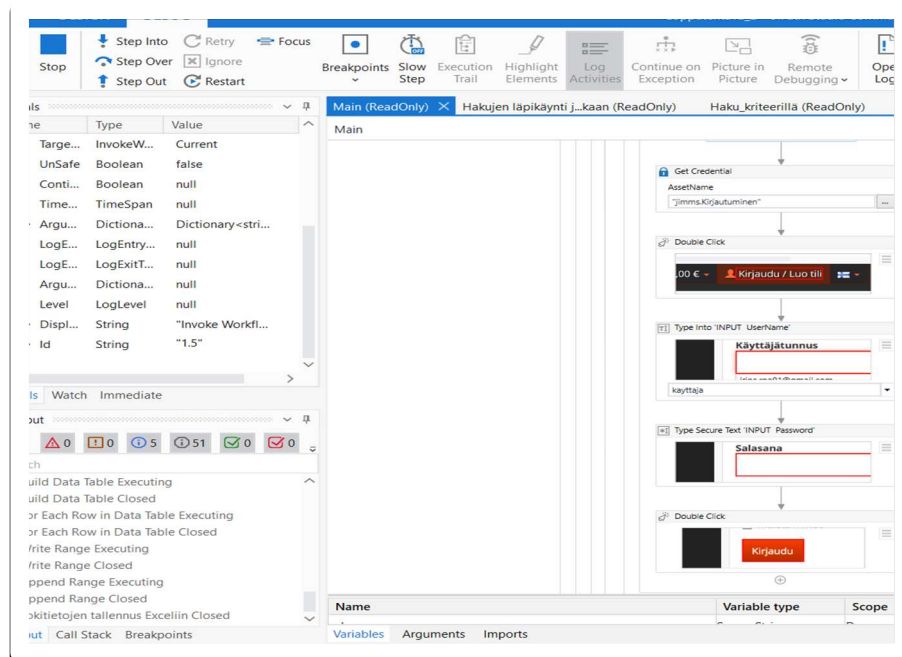
8 DEBUGGAUS/TESTAUS

Tässä dokumentoituuna robotin testaaminen. Debuggaaminen käynnistetään Debug välilehdeksi.



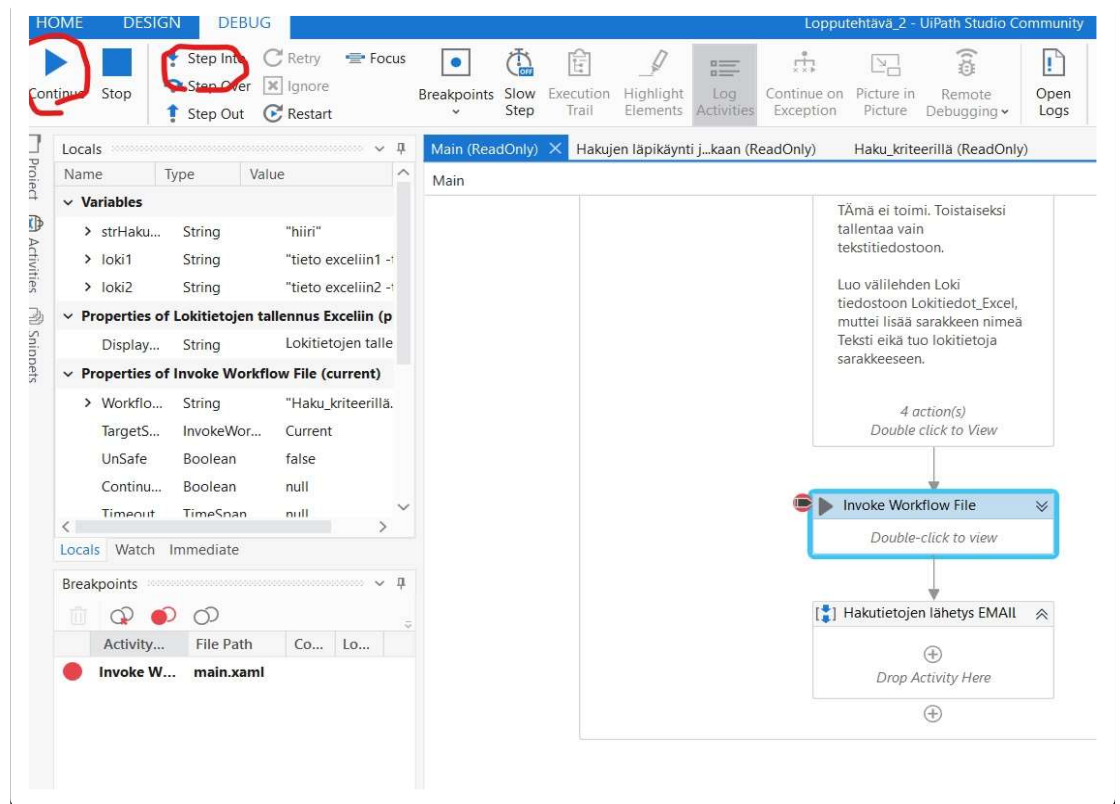
Kuva 24. Debuggaus.

Debuggeri käy koodin läpi aktiviteetti kerrallaan Breakpointtiin asti.



Kuva 25. Debug.

Breakpoint on laitettu hajautetun robotin kohdalle. Tähän debuggaus pysähtyy. Pääsen eteenpäin hajautetun robotin sisään painamalla Step into - tai Continue -painiketta.



Kuva 26. Debug. Breakpoint.

9 HUOMIOITA JA AJATUKSIA PROJEKTISTA

Tarkoituksena oli tehdä yksinkertaistettu versio alkuperäisestä robotista. Huomasin aika pian, että yksinkertaistettukin versio on melko vaativa tehdä näillä tiedoilla. Hakujen läpikäynnin oli tarkoitus olla hajautetussa robotissa (Hakujen läpikäynti ja valinta kriteerin mukaan.xaml) ja toteutettu For Each aktiviteetilla. Yritin tuoda hakutulokset selaimelta Exceliin ja käydä rivi kerrallaan läpi ja filttaröidä esim alle 100,00 maksavat tuotteet. Tämä osoittautui haasteeksi, sillä ilmeisesti ohjelma tuo selaimesta Exceliin luvut string -muodossa. Lisäksi hinnoittelun desimaalit pitää myös konvertoida doubleen. Käytin paljon aikaa ennen kuin pääsin tähän pisteeseen ja lopulta tein ratkaisun, mikä ainakin toimii (haun rajausta selaimella ja Data Scraping). Toivon kuitenkin vielä tulevaisuudessa oppivani muokkaamaan tuotua sisältöä halutunlaisesti.

Lisäksi tätä voi vielä hioa niin, että yhdessä tiedostossa olisi tuotenimikkeet ja toisessa tiedostossa haetut tuotteet, ohjelma vertaisi näitä toisiinsa ja palauttaisi ne nimikkeet mitä ei löydy jälkimmäisessä tiedostossa. Silloin

pääsisin käyttämään vertailuoperaattoreita, mitkä tässä tehtävässä jäi pois. Tämä vastaisi jo alkuperäistä ideaa tästä robotista.

Lisäksi kokeilin tallentaa lokitiedot Exceliin siten, että jokainen tieto menisi omaan sarakkeeseen, jotta Exceliä pystyisi käyttämään paremmin. Esimerkiksi käyttäjä, lokiteksti ja aika omiin sarakkeisiin. Tämänkin jouduin jättämään kesken. Tämän voisi varmaan tehdä splittaamalla, viedä ensin koko lokitieto sellaisenaan Exceliin ja sitten kehittää robotti, joka splittaa tiedon annetuilla parametreilla, esim. siirrä seuraavaan sarakkeeseen lopputeksti ensimmäisen pilkun jälkeen tms. Tämäkin jää ratkaistavaksi tulevaisuudessa.

Haastavinta tässä työssä onkin googlettaminen ja relevantin tiedon/tutoriaalin käyttö tai soveltaminen omaan käyttötarkoitukseen. Lisäksi minulle on vielä melko epäselvää, miten vb-expressioneita käytetään ja mitä niihin on ylipäättään tarkoitus syöttää, jotta pääsee haluttuun tulokseen.

Robotin rakentaminen ja kokeilu on ollut antoisaa, varsinkin kun huomaa asioiden toimivan joskus niin kuin toivoo. Raportin kirjoittaminen on ollut myös mukavaa, vaikkakin tekisi mieli jo kohta hypätä uuteen projektiin ja oppia lisää.

10 LÄHTEET

Anttila, Mika. TAMK. 2020. Opinnäytetyö
https://www.theseus.fi/bitstream/handle/10024/303040/Anttila_Mika.pdf?sequence=3&isAllowed=y (viitattu 21.1.2021)

Tonteri, Topias. JAMK. 2019. Vaatimusmäärittely RPA.
<https://gitlab.labranet.jamk.fi/Mysticons/core/-/wikis/Vaatimusmaarittely-RPA> (viitattu 22.1.2021)