



RPA:n hyödyntäminen integraatiossa - Verkkokaupan tietojen kerääminen ohjelmistorobotiikan avulla

Henri Hertzberg

2024 Laurea



Laurea-ammattikorkeakoulu

RPA:n hyödyntäminen integraatiossa - Verkkokaupan tietojen kerääminen
ohjelmistorobotiikan avulla

Henri Hertzberg
Tietojenkäsittelytieteen ohjelma
Opinnäytetyö
Maaliskuu, 2024

Henri Hertzberg

RPA:n hyödyntäminen integraatiossa - Verkkokaupan tietojen kerääminen ohjelmistorobotiikan avulla

Vuosi

2024

Sivumäärä 31

Työn tavoite oli verkkokaupan tietojen kerääminen Ars Aloise ry:lle hyödyntäen ohjelmistorobotiikkaa. Paras keino sen saavuttamiseksi oli luoda ohjelmistorobotti, joka keräsi tiedot murto-osassa siitä ajasta mikä ihmiseltä kestää. Ohjelmistorobotin luominen tietojen keräämistä varten oli opinnäytetyön kehittämistehtävä. Valittu menetelmä ja palveluntarjoaja olivat Robocorp-RPA ja Robocorpin RPA-kirjastot, kuten esimerkiksi RPA.browser.

Keskeinen tuotos oli valmis ohjelmistorobotti, joka tallennettiin Github repositorioon ja sen tehtävänä oli kerätä tiedot verkkosivulta ja tehdä niistä jokaiselle teokselle oma PDF-tiedoista ja JPEG-kuva taideteoksesta.

Tuloksia arvioitiin toimeksiantajan palautteen perusteella. Korjaukset liittyivät siihen miten sopivaa ohjelmistorobotin keräämä tieto oli seuraavaan verkkokauppaan ja kehittämis ehdotusten perusteella ohjelmistorobottia ja tietoformaattia muokattiin tarvittaessa.

Asiasanat: RPA, ohjelmistorobotiikka, verkkokauppa, integraatio

Laurea University of Applied Sciences

Abstract

Degree Programme in Business Information Technology

Bachelor of Business Administration

Henri Hertzberg

Using RPA in integration - Extracting data from a webstore with robotic process automation methods

Year

2024

Pages

31

The purpose of this Bachelor's thesis was to collect information for the company Ars Aloise ry using RPA on their webstore. The best way to achieve this was to create a robotic process automaton (software robot) to complete the task in a fraction of time it takes a human to do. The development task for this work was to create a complete robot for the information collecting. The selected method to use was Robocorp-RPA with Robocorp RPA-libraries such as RPA.Browser.

The main result was a complete robot in a Github repository and the information collected for each work of art in PDF format-for the information and JPEG-format for the art-pieces in separate files.

The results were analysed based feedback from the thesis supervisor and the project giver and the suitability of the collected information. Conclusions and improvements were done based on the suitability of the data format for the new webstore.

Keywords: RPA, software robotics, webstore, integration

Sisällys

1	Johdanto.....	9
2	Taustaa	10
2.2	Opinnäytetyön tarkoitus	10
2.3	Tutkimuskysymykset	10
2.4	Ilmiön rajaaminen.....	10
2.5	Opinnäytetyön muoto	11
2.6	Hyödyt	11
2.7	Luotettavuus ja eettisyys	11
2.8	Keskeiset käsitteet.....	12
3	RPA	13
3.1	RPA:n hyödyt.....	14
3.2	RPA:n riskit.....	15
3.3	Valvottu vai valvomaton	15
3.4	Selainpohjainen automaatio	16
3.5	Sovelluspohjainen automaatio	16
3.6	Tulevaisuuden RPA	16
3.8	Robocorp ReMark.....	17
3.9	Ketterät ohjelmointimenetelmät	18
4	Tutkimusongelma.....	18
4.1	Strukturoimaton haastattelu	18
4.2	Kehittämismenetelmät	19
4.3	Käytetyt teknologiat ja osaaminen	19
5	Tulokset	20
5.2	Kehittämisprosessi	20
5.3	Ensimmäinen versio valmis	20
5.4	Toinen versio	21
5.5	Kolmas versio	22
6	Robotin toiminta	22
6.1	Aloitustaskit.....	23
6.2	Oliot yhteistyössä	23
6.3	Ensimmäinen olio	24
6.3.1	Toinen olio.....	24
6.3.2	Kolmas olio	25
6.3.3	Neljäs olio	26
6.3.4	Kehitysehdotukset.....	28
7	Pohdinta	29

8	Yhteenveto	30
9	Lähteet	30
10	Kuvat	32

1 Johdanto

Manuaalinen työ on aikaavievää, kallista ja sitä on usein paljon. Se voi olla vallihautojen kaivamista, potilaiden siirtelyä, suuren datamäärän käsittelyä tai muuta samanlaisena toistuvaa ja ennalta määriteltyä työtä. Varmaa on, että tällaisen työn teko helpottuu ja nopeutuu, kun käyttää oikeita työkaluja ja paljon työvoimaa. Tämä pätee myös IT-maailmassa, jossa on omat työkalunsa mutta työvoima on silti kallista.

IT-alan tapauksessa työkalut voivat toimia joskus myös työvoimana. Tietokoneen käyttö ja suuren datamäärän käsittely on luontevaa väsymättömälle ohjelmistorobotille, joka tarvitsee vain valmiiksi määritellyn tehtävän. Ohjelmistorobotiikan keinoin on mahdollista luoda työtehtävään suunniteltuja ohjelmistorobotteja, jotka suorittavat tehtävät automaattisesti ja usein ilman erillistä valvontaa. Robotteja voi kehittää niin monta kuin on tarve ja ne voidaan aikatauluttaa tekemään samat tehtävät päivittäin tai tarpeen mukaan.

Tämän opinnäytetyön ja hankkeen tarkoituksena on kerätä ohjelmistorobotilla tiedot verkkokaupan sivulta. Aihe tulee opinnäytetyön aihepajasta osana suurempaa projektia liittyen uuteen verkkokaupan siirtymiseen ja sen kehittämiseen. Tämä opinnäytetyö rajautuu projektin aloitukseen eli vanhan tiedon keräämiseen uutta verkkokauppaa varten. Opinnäytetyössä kerätään toimeksiantajan pyytämät tiedot heidän verkkokaupasta ja tallennetaan ne sellaiseen muotoon, joka hyödyttää uuden verkkokaupan tekijöitä. Ohjelmistorobotti navigoi sivulla kuin oikea käyttäjä ja automaattisesti käy läpi jokaisen kokoelman jokaisen teoksen ja tallentaa niistä JPEG-kuvan alkuperäisessä resoluutiossa sekä tallentaa teoksen kuvauksen PDF-tiedostoon. Projektissa käytetään ohjelmistorobottia, koska teoksia on yli 200 ja niiden tallentamiseen manuaalisesti kuluisi huomattavasti pidempi aika. Opinnäytetyön tietoperustana käytetään Robocorpin kehittämiä ohjelmistorobotiikan kirjastoja ja nettisivustoa, sekä Robocorp ReMark chatbottia, joka antaa osuva ja hyödyllisiä neuvoja suoraan Robocorpin tietokannasta.

2 Taustaa

Ohjelmistorobotiikka sovelluskehityksen omana uniikkina alueenaan on erityisen kiinnostava, koska sillä on suoria ja selkeitä käytännön sovellutuksia helpottamaan työtä joka muuten olisi erittäin helppoa, tylsää ja aikaavievää. Työhön on kuitenkin tehokas ratkaisu, kuten antaa tietokoneelle lisää valtaa suorittaa se loppuun hieman itsenäisemmin ilman käyttäjän väliintuloa. Robotti suorittaa tehtävänsä yleensä käyttäjän näkökulmasta ja rajapinnasta käsin, mutta paljon nopeammin. Se on siten hieno yhdistelmä automaatiota, optimointia ja tehokkuutta ohjelmoinnin keinoin.

2.1 Toimeksiantaja

Taideyhdistys Ars Aloise ry ja heidän Grafia13-niminen verkkokauppansa ovat opinnäytetyön toimeksiantaja ja kohdesivusto. Verkkokaupan sivulla kuvaillaan yhdistystä lyhyesti ”Kauppaa pyörittää taiteilijavetoinen yhdistys (Ars Aloise ry), joka järjestää näyttelyitä ja taidetapahtumia kotimaassa ja kansainvälisesti. Vuodesta 2012 lähtien kesäisin Galleria Linnankatu 13 Savonlinnassa.” (Grafia13 2024.)

2.2 Opinnäytetyön tarkoitus

Opinnäytetyön tavoite on avustaa yritystä saamaan tietonsa talteen vanhasta verkkokaupasta, jonka kehitystyö on loppunut. Opinnäytetyö tulee oikeaan käyttöön, koska toimeksiantaja aikoo vaihtaa verkkokauppansa alustaa ja siksi heidän tarvitsee saada olennaiset tiedot siirrettyä toiseen alustaan mahdollisimman vähällä manuaalisella vaivalla. Työhön pohdittiin myös liitettäväksi uuden verkkokaupan määrittelyä ja rakenteen pohtimista sen perusteella missä muodossa vanhat tiedot saadaan talteen vanhasta verkkokaupasta.

2.3 Tutkimuskysymykset

Haastattelun perusteella tutkimuskysymykseksi muotoutui miten ohjelmistorobotiikan keinoin saadaan mahdollisimman tehokkaasti kaikki tieto verkkokaupasta talteen hyvässä muodossa. Odotettu tulos on saada tarjolla olevasta 201 taideteoksesta kaikki tiedot käytettäväksi seuraavaan verkkokauppaan sopivassa muodossa. Sivustolla taiteilijoiden kokoelmissa on näkyvillä myös loppuunmyytyjä teoksia, jotka tulee jättää huomioimatta.

2.4 Ilmiön rajaaminen

Opinnäytetyö rajautuu yhden robotin tekemiseen ja siihen liittyvään kirjalliseen osuuteen. Robottia tullaan käyttämään yhdellä verkkokauppasivustolla. Robotin tulee kerätä kaikki haluttu tieto verkkokaupasta jättäen tarpeettomat tiedot, kuten loppuunmydyt teokset huomioimatta. Robotin täytyy myös tallentaa tiedot hyvässä muodossa koneelle ilman että se lähettää niitä mihinkään esim. sähköpostilla. Robotti on tarkoitus saada valmiiksi ja tallentaa

Github-repositorioon, jonka jälkeen robotin ja repositorion oikeudet annetaan toimeksiantajan ja muiden projektin jäsenten käyttöön.

Opinnäytetyö on osa suurempaa projektia saada koko verkkokauppa valmiiksi ja toimimaan, jotta toimeksiantaja voi jatkaa liiketoimintaansa. Tämä opinnäytetyö keskittyy projektin käyntiin saamiseen ohjelmistorobotin rakentamisella ja tietojen tallettamisella sekä mahdollisesti verkkokaupan suunnittelun alkuunsaattamisella.

2.5 Opinnäytetyön muoto

Opinnäytetyön muoto on kehittämistyöhanke, jonka on tilannut Ars Aloise taideyhdistys. He aikovat siirtää verkkokaupansa aiemmalta sivustolta heidän kannaltaan parempaan sivustoon. Itse verkkokaupan rakentaminen ja suunnittelu ovat vielä kesken, koska niille etsitään tekijöitä ja ohjelmistorobotti on suunnitteilla toimeksiantajalta saatujen vaatimusten perusteella. Tämän opinnäytetyön tuotos keskittyy ohjelmistorobotin kehittämiseen ja myöhemmin ehkä verkkokaupan suunnittelussa avustamiseen aikatauluista ja muiden tekijöiden löytymisestä riippuen.

2.6 Hyödyt

Hankkeen antaja hyötyy suoraan projektista, joka on osa suurempaa kokonaisuutta tehdä uusi verkkokauppa. Suurempaa projektia on tarjottu myös opiskelijoille, mutta sen katsottiin olevan niin laaja, että siihen tarvitsisi enemmän kuin yhden tekijän. Robotin valmistuminen ja tietojen talteen saaminen on kuitenkin tärkeä ensimmäinen askel koko projektin käynnistymiselle. Robotin valmistuessa säästetään paljon manuaalista työtä, jonka ohjelmistoautomaattiorobotti tekee huomattavasti ihmistä nopeammin ja tiedot saadaan haluttuun tiedostoformaattiin. Itse opinnäytetyön tekijä hyötyy myös, koska projektissa on mahdollista opiskella lisää RPA-metodeja ja tehdä mielenkiintoinen robotti.

2.7 Luotettavuus ja eettisyys

Toimeksiantajalle huomautettiin haastattelun yhteydessä, että opinnäytetyöt ovat julkisia ja siten projektista tulee kaikille julkinen. Projektia ja robottia tehdessä on pääsy vain verkkokaupan julkiseen puoleen, mikä riittää hyvin robotin tarkoituksiin. Toimeksiantajalta selvitettiin haastattelussa onko sivuston back-end puolella vielä tarpeellista tietoa, joka tulisi siirtää mutta julkinen tieto todettiin riittäväksi ja toimeksiantaja varmistaa että kaikki haluttu tieto on julkisesti nähtävillä kun robotin lopullinen ajo suoritetaan. Työn pohjalta syntyvän robotin oikeudet annetaan projektintekijöiden ja toimeksiantajan käyttöön sopimuksessa, joten kaikkien oikeuksia kunnioitetaan. Toimeksiantaja omistaa itse verkkosivuston, josta tiedot otetaan joten hänellä on oikeus omiin tietoihinsa ja teoksiinsa ja robotin tekijällä on oikeus kerätä julkiset tiedot sivustolta.

2.8 Keskeiset käsitteet

Käydään läpi kehityksen kannalta olennaisia termejä

RPA	Lyhenne robotic process automation. RPA on ohjelmistoteknologia, joka on kaikille helppo käyttöinen ja jota käytetään automatisoimaan digitaalisia työtehtäviä (Automation Anywhere 2024).
Selainpohjainen automaatio	Tarkoittaa selaimessa seikkailevaa robottia, joka toimii web-elementtien kautta.
Sovelluspohjainen automaatio	Tarkoittaa sovellusta ajavaa robottia, joka toimii usein kuvantunnistuksen kautta
Attended RPA	Tarkoittaa valvottua käynnistettävää robottia. Ihminen käynnistää operaation ja tekee valintoja (Robocorp 2024).
Unattended RPA	Tarkoittaa aikataulutettua robottia. Suorittaminen onnistuu ilman ihmistä (Robocorp 2024).
IPA	Lyhenne intelligent process automation. Tarkoittaa AI:ta ja koneoppimista hyödyntävää tulevaisuuden robottia (UIPath 2024).
Open source	Avoimeen lähdekoodiin perustuva
UI	User interface eli käyttöliittymä
OCR	Optical character recognition
Olio	Olio-ohjelmoinnin termistön mukainen itsenäinen osa ohjelmistoa, joka suorittaa tietyn toiminnon

3 RPA

”But when you look at each word, it’s understandable why RPA can be a bit confusing. For example, the word “robotic” does not refer to a physical robot - instead, it is about a software-based robot (or bot) that can automate human actions in the workplace (generally for white collar applications in clerical and administrative functions).” (Taulli 2020.)

Taulli avaa edellä hienosti kirjassaan mitä RPA-robotilla tarkoitetaan. Se on siis lähtökohtaisesti ohjelmointia esimerkiksi Pythonilla. RPA-ohjelmistoissa tavallisia ohjelmia kutsutaan boteiksi, mutta ohjelmat ovat niin erikoistuneita että ne ovat eriytyneet omaksi alakseen. Eriytymiselle on hyvä syy, koska uudella alalla on paljon kysyntää. Taullin (2020, 1) mukaan huhtikuussa 2019 suuri RPA-yritys UiPath keräsi 568 miljoonaa dollaria sijoituksia, sen arvotus nousi 110 miljoonasta dollarista 7 miljardiin dollariin, vuosittaiset tulot nousivat 8 miljoonasta dollarista 200 miljoonaan dollariin ja työntekijöiden määrä kasvoi 16 kertaiseksi 2500 työntekijään. Seuraavassa kappaleessa muutamia yrityksiä määrittelemässä RPA-termiä. (Taulli 2020.)

Automation anywhere RPA-yrityksen mukaan RPA on ohjelmistoteknologia, joka on kaikille helppo käyttöinen ja jota käytetään automatisoimaan digitaalisia työtehtäviä. Se on säädettävä ohjelmisto, jolla suoritetaan erilaisia prosesseja. Se työskentelee applikaatioiden ja systeemien kautta samalla tavalla kuin ihmiset, mutta virheettömästi, paljon nopeammin ja kellon ympäri. Käyttäjän tarvitsee vain määritellä tehtävä ja säännöt ja antaa automaation hoitaa loput. (Automation anywhere 2024.)

Nintexin mukaan RPA:n avulla voi luoda virtuaalista kellon ympäri toimivaa työvoimaa joka toimii nopeasti, joustavasti sekä tehokkaasti. Valvottu RPA auttaa työntekijöitä suorittamaan tehtävänsä tarkemmin ja tehokkaammin ja valvomattomat robotit voidaan käynnistää automaattisesti milloin vain ilman ihmiskontaktia. RPA:n hyötyjä ovat mm. inhimillisten virheiden poistaminen manuaalisessa datan käsittelyssä ja kalliiden virheiden korjaamisen välttämässä, työntekijöiden vapauttaminen monimutkaiseen työhön automatisoimalla yksinkertaiset työtehtävät ja työntekijöille tietokoneella tehtävän työtehtävän suorittamisen esittely askel askeleelta. (Nintex 2024.)

PEGA:n mukaan heidän RPA-tuotteensa hyödyt ovat nopeassa automaatiassa. Heidän tuotteensa lupaa automatisoida toistuvat työt ja auttaa työntekijää keskittymään asiakkaaseen, yhdistää erilaiset systeemit ilman että aikaa kuluu applikaatioiden välillä navigointiin sekä nopeuttaa prosesseja ja vähentää virheitä. (Pega 2024.)

3.1 RPA:n hyödyt

Taullin (2003, 11) mukaan konsulttiryitys A.T. Kearney on laskenut että keskimäärin ohjelmistorobotti maksaa kolmanneksen siitä mitä ulkomaalainen työntekijä (offshore employee) ja viidenneksen siitä mitä kotimainen työntekijä (onshore staff).

Ala on itsessään merkittävä osa ohjelmointia ja tarpeeksi erilainen käyttötarkoituksiltaan ollakseen selkeästi tunnistettava. Mitä hyötyä on sitten koko RPA-alasta? RPA:n tehokkuus perustuu sen automaattiseen robottimaiseen suorittamiseen missä tietokoneet ovat huomattavasti ihmisiä parempia eli tarkempia ja nopeampia. Tällaisia suorituksia ovat esimerkiksi suuren määrän tietoa liittämistä ja siirtämistä paikasta toiseen, nettisivun avaaminen ja kirjautuminen, sähköpostin ja liitteiden avaaminen, tietokantojen täyttö ja selailu, tietojen hakeminen lomakkeista tai dokumenteista, laskeminen ja työnkulkujen seuraaminen. (Taulli 2020, 3.)

RPA:n nopeuden ja tarkkuuden lisäksi sillä on toinen hyöty joka liittyy myös työn epäinhimillisyyteen. Taulli (2003, 88) listaa RPA:n kannalta optimaalisiksi töiksi työt jotka ovat tylsiä, aikaavieviä, itseään-toistavia, toistuvia, tarkasti säädeltyjä, selkeästi rajattuja, työläitä tietomäärältään, virheherkkiä, API-rajapintaa eli application programming interface käyttäviä, ihmiselle selkeyttämistä vaativia, herkkää dataa sisältäviä, suurta työntekijämäärää vaativia tai muuten organisaation toimintaa edistäviä.

Tylsissä tehtävissä työntekijöiden taidot menevät hukkaan, koska kuka tahansa voisi tehdä ne niiden mekaanisen luonteen vuoksi. Sellaisia tehtäviä voivat olla nappien painelu tai tietojen siirtely. Aikaavievät, toistuvat ja itseään-toistavat tehtävät ovat käytännössä täydellisiä RPA-kohteita, koska ne saadaan tehtyä nopeasti, standardin mukaan ja robottia voi uudelleenkäyttää lukemattomia kertoja samaan toimintoon. Tällaisia tehtäviä voisivat olla raporttien kirjoittaminen valmiilla kaavalla lisäten vain tunnusluvut. Selkeästi rajatut ja tiettyjä sääntöjä noudattavat tehtävät kuten lomakkeiden täyttö ja luku ovat myös sopivia kohteita, koska niihin voidaan soveltaa myös ohjelmoinnin ehtorakenteiden logiikkaa. Suurten tietomäärien käsittely API:en ja tietokantojen parissa työskennellessä, missä suurin riski virheille ovat inhimilliset virheet on myös omiaan RPA-roboteille, sillä niiden keskittymiskyky on rajaton. Mahdolliset tehtävät joissa data on muodossa, joka vaatii erityistä osaamista tulkintaan tai on arkaluontoista voidaan myös antaa robotille käsiteltäväksi, koska tällöin muokkaamiseen vaadittavat tiedot pysyvät luottamuksellisina. Roboteille sopii myös kaikki muut tehtävät, jotka vaativat suurta toistuvaa työpanosta tai muuten vapauttavat organisaation jäseniä muihin tuottavampiin töihin. (Taulli 2020, 88.)

3.2 RPA:n riskit

RPA:lla on kuitenkin rajoituksia ja vaatimuksia. Taulli (2003, 16) listaa riskeiksi omistajuuden hinnan, teknisen velan, yrityksen skaalauksen, kyberturvallisuuden, alan hypetyksen, valmistelut, teknologian rajoitukset sekä virtuaaliset ympäristöt.

Omistajuuden hinta riippuu aina liiketoimintamallista, koska ohjelmistorobotit ja niiden ylläpito täytyy ostaa aina palveluntarjoajalta tai kehittäjältä. Poikkeuksena tähän on itse tehdyt Open Source-pohjaiset sovellukset. Osa käyttää lisenssiä tai hintaa per robotti, mutta lisäkuluihin kuuluu ylläpito sekä käyttöönoton koulutus. Käyttöönottoon voidaan tarvita myös konsulttien apua. Teknisen velan käsite liittyy myös edelliseen, koska botit voivat kaivata muokkausta ja päivitystä prosessien muuttuessa. Tällöin myös botit täytyy sovittaa uuteen prosessin etenemiseen ja tunnistamiseen, mikä vaatii jatkuvaa huomiota. Yrityksen kasvaessa boteista saadaan lukuisia hyötyjä, mutta bottien määrän kasvaessa niiden hallitseminen tarkoittaa paljon työtä IT-osastolle. Bottien parissa työskennellessä on myös syytä huomioida kyberturva, jos ne käsittelevät arkaluontoista tietoa sekä fyysinen turvallisuus, jos ne hallitsevat esimerkiksi tuontantolaitteita. (Taulli 2003, 16.)

Alan korkeat odotukset voivat myös johtaa pieniin pettymyksiin, kun kaikki haluavat itselleen uusia botteja. Taullin (2003, 16) mukaan Pegan tekemä tutkimus huomasi että hyvän botin ja prosessin tekemisessä kestää keskimäärin 18 kuukautta, joista vain 39% tulee valmiiksi ajallaan. Tärkeä osa botin valmistusta on nimittäin tuntea tarkkaan automatisoitava tehtävä, koska botti toistaa myös huonot ratkaisut ja virheet täydellisesti joka kerta. RPA-teknologian rajoituksena on toistuvat rutiinitehtävät ilman päätöksentekoa. Kuitenkin botin prosessin voi pysäyttää päätöstä vaativaan kohtaan kuten valmiin ostoksen tai dokumentin tarkistukseen, jolloin ihminen voi puuttua asiaan. Myös AI eli artificial intelligence kehittyy jatkuvasti, jolloin siitä voidaan tulevaisuudessa saada hyvä päätöksentekijä täysin automaattiseen prosessiin. Viimeinen rajoitus on virtuaaliset ympäristöt eli etäyhteys työpöytään, jonka kautta botti voi toimia. Ne asettavat vielä ylimääräisen kerroksen ruudun ja tekstin tunnistamiseen ruudulta sekä itse tekstielementtiin käsiksi pääsemisen, mikä voi vaikeuttaa robotin toiminnan suunnittelua. (Taulli 2020, 16.)

3.3 Valvottu vai valvoton

Taullin (2020, 24) mukaan RPA-robotteja on kahdenlaisia, suomeksi ihmisen toimesta valvottuja ja valvomattomia eli attended ja unattended. Käytön aikana valvottu botti auttaa ihmistä tietyssä tehtävässä, kuten ensin kysyy tietoja ja sitten ohjaa puhelun alkutietoineen avustavaan puhelimeen vastaavalle henkilölle. (Taulli, T. 2020, 24). Toinen esimerkki avustetusta RPA:sta voisi olla chatbotti, joka yhdistää esimerkiksi verkkokaupan myyjälle keskustelun hakutietojen tai kysymyksenasettelun jälkeen.

Valvomaton RPA taas tarkoittaa että kyseessä on automaattinen robotti, joka käynnistyy tai käynnistetään ajastetusti, ohjelman tai sähköpostin kautta tai lomaketta lähetettäessä ja käy läpi ohjelmansa kokonaan ilman avustusta (Taulli. 2020, 24). Tämän opinnäytetyön ohjelmistorobotti tulee olemaan valvomaton robotti, joka kertaluontoisesti käy läpi ja tallentaa tiedot itsekseen sivulta, mutta vaatii käynnistykseen.

3.4 Selainpohjainen automaatio

Selainpohjaisella automaatiolla tarkoitetaan RPA-robottia, joka käyttää selainta ja työpöytää toimintaympäristönään. Tässä opinnäytetyössä on käytetty juuri selainpohjaista automaatiota hyödyntävää robottia. Tällainen robotti hyödyntää UI automaatiossa OCR eli optical character recognitionin eli tekstintunnistuksen keinoja. Robotti voi suunnistaa sivustolla käyttäen tekstiä, ikkunoita, kuvia, hiiren ja näppäimistön komentoja, sekä selaimen web-elementtejä kuten raahattavaa palkkia. Selaimessa robotti osaa myös tunnistaa varsinaisia web-elementtejä suoraan HTML, CSS tai Javascriptistä niiden nimien, tyyppien tai luokkien perusteella ja käyttää niitä referenssinä mistä painaa tai mitä tallentaa. Robotit osaavat myös aina aluksi itse avata selaimen ja navigoida URL:in kautta nettisivulle ja lopuksi sulkea ikkunan ja selaimen. (Taulli 2020, 151.)

3.5 Sovelluspohjainen automaatio

Sovelluspohjaisessa automaatiossa robotit käyttävät työpöytää ja avaavat jonkin sovelluksen käyttöliittymän ja navigoivat siellä haluttuihin paikkoihin suorittaen eri toimintoja. RPA-ohjelmistot toimivat hyvin työpöytä-ympäristössä luoden, kopioiden, yhdistellen tai siirrellen tiedostoja. Ne voivat tarkistaa, onko tiedostopolkua tai tiedostoa olemassa, leikata, kopioida ja siirtää tekstiä, avata ja sulkea esimerkiksi windows-sovelluksia, käsitellä muuttujia ja salasanoja ja jopa ajaa PowerShell-scriptejä. (Taulli 2020, 151.)

Robocorp-windows kirjastolla voidaan locatorien avulla käyttää kaikkia sovelluksia tunnistamalla niiden sisäisiä ikkunoita ja elementtejä, jolloin niitä voidaan painella. Kirjasto aloittaa siis Desktop tasosta, josta löytää WindowElementin. WindowElementin kautta etsitään ControlElement ja suunnistetaan sovelluksen ikkunan sisällä. ControlElement taas mahdollistaa työskentelemisen elementin kanssa, kuten napin löytämisen ja sen painamisen. (Robocorp 2024.)

3.6 Tulevaisuuden RPA

Tulevaisuuden RPA:ta kutsutaan intelligent process automationiksi eli IPA:ksi. koska se hyödyntää AI:ta ja muita sivistyneitä teknologioita tekemään päätöksiä edistämään automaatioprosessia (Taulli, 2020, 24). Tällöin saataisiin RPA:sta vieläkin automaattisempaa, kun voitaisiin laajentaa olosuhteita joissa robotti lähtee käyntiin ja vähentää ihmisen osuutta

lisäämällä päätösvaltaa ja kykyä tehdä parempia päätöksiä prosessin edetessä. IPA tai kognitiivinen RPA on viimeisintä RPA-teknologiaa, koska se hyödyntää toiminnassaan AI:ta joka oppii ajan kanssa paremmin esimerkiksi tulkitsemaan tietynlaisia dokumentteja. Sen avulla osasta valvotuista prosesseista ihmisen osuus voitaisiin poistaa kokonaan, koska AI olisi jo oppinut minkä päätöksen sen täytyy kussakin tilanteessa tehdä. Silloin RPA-ohjelmistot pääsevät käyttämään yhä enemmän omaa päätösvaltaa sekä tietämystään ja raja valvotun ja valvomattoman RPA:n välillä alkaa hälvetä (Tauli, 2020, 6.)

3.7 Python

Python on huippusuositettu open source- ohjelmointikieli, jota kuvaillaan heidän sivulla seuraavasti:” Python is powerful... and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open. -These are some of the reasons people who use Python would rather not use anything else.” (Python.org 2024.)

Pythonia voidaan käyttää serveripuolella verkko-ohjelmistoissa, RPA-työnkuluissa, tietokantojen muokkaamisessa, big datassa ja matematiikassa sekä ohjelmoinnissa prototyyppien tai ohjelmien kehittämiseen. Se toimii monella eri alustalla kuten Windows, Mac, Linux, Raspberry Pi ja on syntaksiltaan hyvin lähellä englanninkieltä. Se on suunniteltu vaatimaan vähemmän rivejä kuin muut kielet ja siihen kuuluu kääntäjä, eli sitä voi suorittaa testimielessä samalla kun siitä kehittää uusia versioita. Sitä voidaan myös käyttää proseduraalisesti, olio-ohjelmoinnin tyyllisesti tai funktionaalisesti. Muihin kieliin verrattuna se on kehitetty luettavuus edellä eli se vastaa englantia ja matemaattista kielenkäyttöä. Se käyttää rivinvaihtoa käskyjen lopussa toisin kuin jotkut muut kielet, jotka käyttävät puolipisteitä tai sulkuja. Se käyttää sisennystä ”näkyvyysmääreiden” määrittelyssä silmukoille, funktioille sekä luokille, missä toiset kielet käyttävät aaltosulkeita. (w3schools.com 2024.)

3.8 Robocorp ReMark

Robottia tehdessä ja Robocorpin sivustolla RPA-kirjastoja selatessa ja esimerkkejä etsiessä on Robocorpin omasta chatbotista ReMarkista paljon hyötyä. Chatbotin voi avata sivuston alalaidasta ja kysyä suoraan miten tietyllä kirjastolla voi saada aikaan asioita. ReMark antaa usein esimerkki-koodinpätkiä, joista näkee selkeämmin miten tietyn kirjaston syntaksi toimii. ReMark antaa myös ratkaisuja perustuen hieman eri kirjastoihin riippuen miten kysymyksen asettelee ja joskus antaa myös vaihtoehtoisia tapoja toteuttaa toiminnon samassa vastauksessa. Chatbotti kuitenkin tarjoaa pelkästään Pythoniin liittyviä ohjeita, koska sitä on kielletty enää opettamasta Robot frameworkin keywordeihin perustuvan syntaksin mukaisia ohjeita. Chatbotista on erittäin paljon hyötyä toimintojen suunnittelussa, toteuttamisessa ja syntaksin tarkistamisessa. Sen käyttö on hupaisa esimerkki uudesta tavasta oppia AI:n avulla,

varsinkin kun aihe on tehdä ohjelmistorobotteja toisen ohjelmistorobotin ohjeiden mukaan. (Robocorp Remark 2024.)

3.9 Ketterät ohjelmointimenetelmät

Ketterien menetelmien arvot perustuvat yksilöiden ja heidän välisten interaktioiden arvottamiseen prosessien ja työkalujen yläpuolelle. Tämä tarkoittaa että yksilöt ja heidän päätöksensä sekä mielipiteensä ovat tärkeimpiä prosessien eteenpäin vieviä voimia. He arvostavat myös toimivaa ohjelmistoa tarkan dokumentoinnin ylitse eli pääpaino pidetään prosessin etenemisessä liikaa dokumentointiin keskittymisen sijaan. Tärkeä painopiste agilessa on yhteistyö asiakkaan kanssa sopimusneuvottelujen sijaan. Tämä tarkoittaa tiivistä yhteistyötä asiakkaan kanssa, jotta ohjelmistosta saadaan reaaliajassa juuri sellainen kun asiakas ja sopimus määrittävät. Viimeinen huomio kehitysmenetelmässä on muutokseen reagointi suunnitelmien seuraamisen sijaan, jotta menetelmää tosissaan voisi kutsua ketteräksi ja että se voisi vastata muuttuviin haasteisiin ja tarpeisiin riittävän ajoissa. (Agilemanifesto 2024.)

4 Tutkimusongelma

Tässä luvussa käydään läpi tutkimusmenetelmät, kehittämismenetelmät ja käytetyt teknologiat. Projekti alkoi robotin vaatimusten selvittämisellä strukturoimattoman haastattelun avulla, jatkui tutun RPA-kirjaston valinnalla ja tarkentui käytetyn teknologian siirtymisellä Robot frameworkin keywordeistä Pythoniin.

4.1 Strukturoimaton haastattelu

Ennen varsinaisen robotin kehitystä täytyy saada tietoon vaatimukset ja rajoitukset robotille sekä sen hakemalle tiedolle. Tähän tarkoitukseen käytettiin videoyhteyden avulla suoritetta strukturoimatonta haastattelua. Strukturoimatonta haastattelua kutsutaan avoimeksi haastatteluksi, keskustelunomaiseksi haastatteluksi tai kliiniseksi haastatteluksi ja sen juuret ovat lääkäreiden ja pappien käyttämässä menetelmässä, jota nykyään käyttävät myös psykologit ja sosiaalityöntekijät. Strukturoimattomassa haastattelussa eli avoimessa haastattelussa olennaista on se ettei keskustelussa käytetä lomakkeita tai valmiita tarkasti määriteltyjä kysymyksiä. Haastattelussa haastattelija ja haastateltava keskustelevat avoimesti ja luontevasti tietyn aihepiirin sisällä paljolti haastateltavan ehdoilla. Haastattelijan roolia voi kuvata keskustelukumppaniksi, joka hienovaraisesti luotsaa keskustelua takaisin varsinaiseen aiheeseen. Haastattelijalta vaaditaan herkkyyttä kuunnella ja osata tarttua haastateltavan sanomisiin vieden näin keskustelua eteenpäin ja saaden mielenkiintoista aiheeseen liittyvää aineistoa. (Tietoarkisto 2024.)

4.2 Kehittämismenetelmät

Robotin tekemisen menetelmiä tullaan kuvaamaan Robocorpin kirjastojen ja robotin funktionaalisuuden kautta, kuten myös mahdollisesti lähdekoodin ja kommenttien kautta. Robotin toimintaa kuvataan myös selainperäisen RPA-automaation termien ja kirjallisuuden kautta. Suuri osa opinnäytetyöstä on itse robotin suunnittelu ja ohjelmakoodin kirjoittaminen.

4.3 Käytetyt teknologiat ja osaaminen

Robotin tekoon on käytetty RPA-palveluntarjoajista valitun Robocorpin palveluita ja kirjastoja. Ohjelmointi on suoritettu VSCode-alustalla käyttäen lisäosia Robocorp Code ja Robot framework language server. Lisäosat ovat helppokäyttöisiä ja tuttuja aiemman ohjelmistorobotiikka-kurssin pohjalta. Ohjelmointikielenä käytetään Robocorpille ominaista Pythonia. Aiemmin Robocorp robotteja voi tehdä myös heidän omalla Robot framework kielellä joka oli vähemmän ohjelmointiosaamista vaativa ja kansantajuisten keywordien pohjalta toteutettu, mutta siitä ollaan luopumassa. Itse robotti on tehty Python ohjelmointikieltä käyttäen, koska Robocorp on opinnäytetyön tekemiseen aikaan siirtymässä kokonaan pois helppolukuisemman heidän omakehittämänsä Robot framework-kielen käytöstä. Pythonin kautta voidaan kuitenkin hyödyntää RPA-kirjastoja, kunhan syntaksi noudattaa Pythonille tuttua rakennetta RPA-keywordien sijaan.

5 Tulokset

Tässä luvussa käydään läpi koko robotin kehittämisprosessi määritelmistä viimeisimpään versioon. Kehittämisprosessin alussa täytyi ratkaista muutamia toiminnallisuuteen liittyviä haasteita ja prosessin lopussa robottia muokattiin palautteen perusteella. Kappaleessa esitellään myös ohjelmiston tärkeimmät oliot ja kaavio koko sovelluksen toiminnan etenemisestä.

5.1 Haastattelu kohdeyrityksen kanssa

Varmistamaan ja tarkentamaan tietoa robotin vaatimuksista oli konsultoitava toimeksiantajaa Teams-yhteydellä. Haastatteluun osallistui ohjaava opettaja ja opinnäytetyön tekijä sekä toimeksiantajan edustaja eli Ars Aloise taideyhdistyksen hallituksen jäsen. Haastattelu tapahtui 12.2.2024 ja kesti noin tunnin, jonka aikana selvitettiin tarkemmin mitkä tiedot robotin tulisi kerätä. Haastattelussa kävi myös ilmi että julkinen yhteys verkkokaupan sivustolle olisi riittävä ja tarvittaessa back-end puolen tietoja laitetaan esille, kun robotin lopullinen ajo suoritetaan.

Toimeksiantajan vaatimukseen kuului saada verkkokaupan tuotteista - jotka ovat suurin osa fyysisiä litografisia taideteoksia - vähintään teoksen nimi, kuva, kuvaus, käytetty tekniikka, taiteilijan nimi, hinta, saatavuus, valmistusvuosi ja mahdollisesti myös verkkokaupan sivulle lisäysvuosi.

Ratkaistava tutkimusongelma on siten kerätä kaikki saatavilla oleva tieto teoksista verkkokaupan sivustolta. Ohjelmistorobotiikan keinoin käytetään käyttöliittymänä julkista sivua ilman kirjautumista.

5.2 Kehittämisprosessi

Ensimmäinen haaste robotin kehityksessä oli saada se navigoimaan sivustolla ja saattaa robotti löytämään kaikki yksittäiset taideteokset käymällä jokainen taitelija ja tämän kokoelma yksi teos kerrallaan läpi. Seuraava haaste oli tallettaa kaikki tiedot samassa tahdissa kuin niitä käydään läpi. Näiden ongelmien ratkaisemiseksi oli selvitettävä millä web-elementeillä pääsy kuhunkin sijaintiin onnistuu. Kolmas ongelma on saattaa tallennettu tieto muotoon, jota on helppo käsitellä ja käydä läpi mahdollisimman pienellä vaivalla. Tätä pohdittiin yhdessä opettajan kanssa ja päädyttiin erilliseen kuvaan ja PDF-tiedostoon teoksen teknisistä tiedoista.

5.3 Ensimmäinen versio valmis

Ensimmäisen version aikana tutustuttiin sivuston rakenteeseen ja selvitettiin tarvittavia web-elementtien nimiä ja niiden rakennetta. Ongelmaan navigoida kaikki tekijät ja teokset läpi oli

ensimmäisenä löydettävä ratkaisu. Lopulta todettiin että helpointa on suoraan siirtyä URL-osoitteen avulla etusivulle jokaisen kierroksen alussa ja jatkaa listan läpikäymistä siitä. Kuvassa 1 esitellään osa ohjelmaa, joka käy läpi sivuston kokoelmat ja siirtyy lopuksi etusivulle ja laskurissa seuraavan kokoelman alkuun.

```
while kohta < pituus:

    selain.go_to("https://holvi.com/shop/grafia13/")
    kokoelma = selain.get_webelements(locator="class:store-navigation-subsection-name")

    paina = kokoelma[kohta]
    selain.click_element(paina)

    kohta+=1
    Avaa_kokoelma()
    selain.go_back()
```

Kuva 1: Kokoelmien läpikäyjä-looppi

Toisena päivänä kun saatiin robotti käymään sujuvasti läpi kaikki tiedostot ne täytyi saada tallennettua PDF-muotoon kuvan kanssa. PDF kuitenkin kieltäytyi tunnistamasta ääkkösiä ja muutamaa teosten nimissä käytettyä merkkiä, jolloin suoritus keskeytyi aina saman teoksen kohdalla. Merkkimanipulaatiot ja suomalaisten ä- sekä ö- kirjainten muuttaminen a- ja o-muotoon ratkaisi ongelmat tilapäisesti.

5.4 Toinen versio

Toisen version luominen alkoi selvittämällä paremmin mitkä web-elementit, kuten CSS ja Javascript luokat, id:t tai nimet parhaiten kuvaavat tiettyjä elementtejä. Selkeän kuvan muodostuttua oli helppoa tehdä elementeistä listoja ja käydä niitä läpi yksi kerrallaan, koska ne sisälsivät vain niitä elementtejä mitä haluttiin käydä läpi. Elementtien rakenteen ollessa kunnossa täytyi saada myös robotin rakenne kuntoon, jolloin luovuttiin erillisistä taskeista ja siirryttiin käyttämään yhdessä toimivia olioita eli funktioita, jotka suoritettiin järjestyksessä ja jotka sisälsivät sisäkkäisiä looppeja listoihin perustuen. Tällöin oli helpompi päivittää kierros- ja kerta-muuttujia, jotka seurasivat läpikäytyjen looppien lukumäärää ja hallita muuttujien scopea. Sisimpien muuttujien seuraamiseen käytettiin globaaleja muuttujia. PDF-tiedoston luomiseen tarvittiin muutamia erikoisia kuva ja PDF kirjastoja, mutta kuvat ja merkit saatiin muokattua oikeaan muotoon ja PDF-tiedosto tallennettua. Erityisinä onnistumisina voi pitää URL:ista löytyneen pyydetyn lisäspäivämäärän löytyminen ja sen muokkaaminen sopivaan muotoon. Myös loppuunmyytyjen muodostaminen omaksi listakseen ja sen listan pituuden vähentäminen kokoelman kaikkien kuvien listasta käyttäen javascript luokkia oli hyvä keksintö, joka helpotti teosten valikoimista. Kuvassa 2 esitellään yksittäisten teosten läpikäynnin ratkaisu.

```

259 def Avaa_kokoelma():
260
261     teokset = selain.get_webelements(locator="class:store-item-image")
262     loppuunyydyt = selain.get_webelements(locator="class:store-item-sold-out")
263     miinus = len(loppuunyydyt)
264
265     kerta = 0
266     pituus2 = len(teokset)-miinus
267
268     while kerta < pituus2:
269         klikkaa = teokset[kerta]
270         selain.click_element(klikkaa)
271
272         kerta+=1
273         Talleta_tiedot()
274         selain.go_back()
275
276     loppuunyydyt == 0
277     kerta == 0
278

```

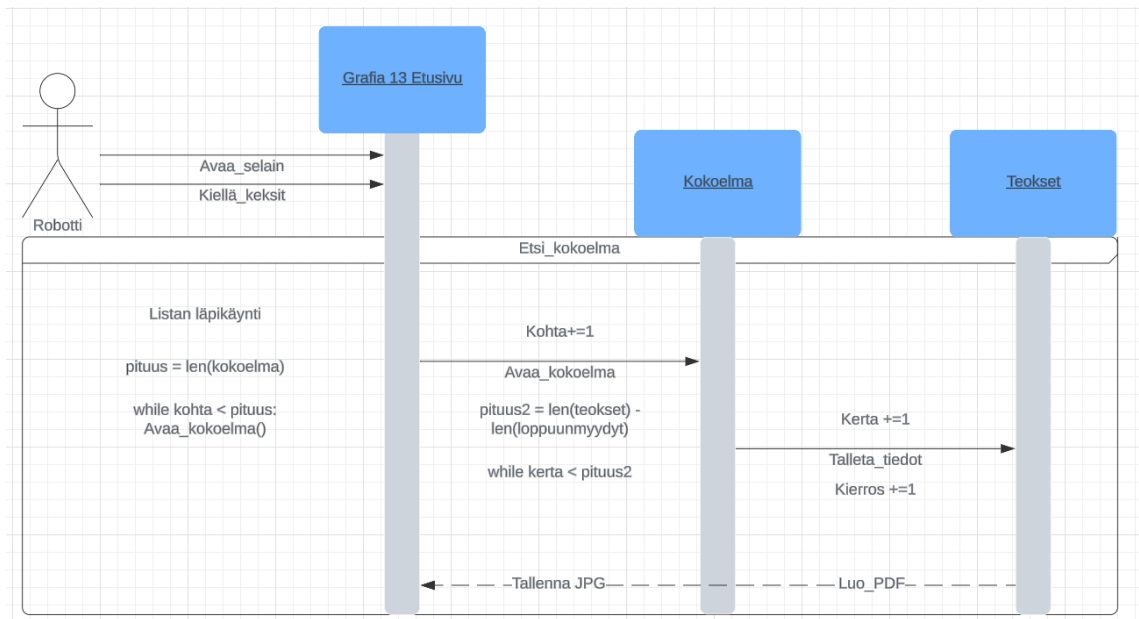
Kuva 2: Kahden listan erotus looppi.

5.5 Kolmas versio

Kolmas viimeistelty ja palautteen perusteella muokattu versio saatiin valmiiksi hyvässä aikataulussa. Siinä kuvat tallennettiin erikseen tiedoista JPEG-muodossa ja tiedot tallennettiin PDF-muotoon kuvatiedoston nimen kera. Kuvan alkuperäinen resoluutio saatiin talteen navigoimalla itse kuvan sivustolle, jossa näkyi pelkästään itse kuva alkuperäisessä resoluutiossaan ja siitä saatiin kuvankaappaus. Robotti ajettiin vielä varmuuden vuoksi kokonaan läpi, jotta nähtiin että kuvat ja tiedot tulevan talteen oikeassa muodossa ja että robotti toimii moitteettomasti koko listan läpi. Kaikkien noin 200 teoksen läpikäynnissä meni hieman aikaa, mutta robotti suoritti tietojen tallennuksen koneelle onnistuneesti.

6 Robotin toiminta

Kaikkiin työssä esiintulleisiin ongelmiin on ratkaisut olemassa Robocorpin eri RPA-kirjastoissa, kunhan niitä osaa tarpeeksi tehokkaasti hyödyntää ja soveltaa. Kyseinen robotti käyttää selain-ympäristöä ja siihen toimii parhaiten pohjana RPA.Browser.Selenium kirjasto Robocorpilta. Alussa tasks.py tiedostoon tulee import toiminnolla ottaa käyttöön tarvittavat kirjastot, kuten RPA.browser.Selenium ja luoda tarvittavat muuttujat työpöydälle, ikkunalle sekä selaimelle. Muuttujien luominen ja tasks-listan tekeminen ovat hyviä tapoja aloittaa robotin teko ja selkeyttää itselle koko robotin työnkulku. Kuvassa 3 näkyy robotin työnkulun kaavio.



Kuva 3: Robotin työnkulku

6.1 Aloitus taskit

Ensimmäinen task eli tehtävä, joista koko robotti koostuu on avata selain. Selaimen sai myös aukeamaan tietyllä resoluutiolla, jolla sen sai koko ruudun kokoiseksi. Tämä edellytti että tiesi oman näyttönsä resoluution.

Toinen task oli sulkea sivuston automaattisesti avaama valinta liittyen kekseihin etsimällä ”Hylkää kaikki” napin id ja käyttämällä `click_element` komentoa. Keksejä kysyvä pop-up ilmaantuu, kun sivusto avataan ensimmäisen kerran ja se täytyy sulkea, jotta robotti voi jatkaa sivustolle.

Kolmas task oli löytää tallennettavat tiedot verkkokaupasta. Tämä task kattaa kaikki vaadittavat toiminnot teoksen löytämiseksi kokoelmasta, sen kuvan tallentamiseksi ja tietojen lisäämiseksi PDF tiedostoon. Robotti käy tämän taskin sisällä kaikki kokoelmat ja teokset läpi yksi kerrallaan. Task koostuu 3 eri funktiosta eli oliosta, jotka keskenään toteuttavat kokoelmien läpikäynnin ja tietojen tallennuksen kahden sisäkkäisen loopin avulla.

6.2 Oliot yhteistyössä

Robotin näkökulmasta kaikki neljä funktiota, jotka navigoivat kokoelmia ja ottavat tiedot talteen ovat itsenäisiä taskeja, mutta käytännössä ne täytyi ohjelmoida erillisinä funktioina yhden ison taskin sisään. Syy tähän oli se että ne sisälsivät kaksi sisäkkäistä looppia, jotka käyttivät toistensa muuttujia ja siten oli helpompi hallita niin sanottuja nested looppeja ja niiden muuttujien scopea luomalla ne sisäkkäisiksi funktioiksi, jotka suorittivat toisensa järjestyksessä. Funktioita voi myös ajatella olioina olio-ohjelmoinnin oppien mukaan, koska

ne työskentelevät keskenään ja niillä kaikilla on tietty tarkoitus. Oliot oli järkevää sisällyttää saman taskin sisään myös siksi, koska ne määriteltiin toistensa sisällä, jonka jälkeen niitä kutsuttiin osana funktiota.

6.3 Ensimmäinen olio

Ensimmäinen olio on nimeltään `Etsi_kokoelma` ja sen tehtävä on etsiä web-elementin avulla jokaisen kokoelman nimi, joka on myös teosten taitelijan nimi. Funktio etsii jokaisen taiteilijan nimen ja tekee niistä listan, jonka se käy läpi nimi kerrallaan navigoiden ensin etusivulle ja sieltä painaen vuorossa olevaa kokoelmaa. Senjälkeen funktio ajaa seuraavan olion nimeltä `Avaa_kokoelma` ja etenee listassa. Kuvassa 4 ensimmäinen olio `Etsi_kokoelma`.

```

230 def Etsi_kokoelma():
231
232     kokoelma = selain.get_webelements(locator="class:store-navigation-subsection-name")
233     print(len(kokoelma))
234
235     pituus = len(kokoelma)
236     kohta = 0
237
238     while kohta < pituus:
239
240         selain.go_to("https://holvi.com/shop/grafia13/")
241         kokoelma = selain.get_webelements(locator="class:store-navigation-subsection-name")
242
243         paina = kokoelma[kohta]
244         selain.click_element(paina)
245
246         kohta+=1
247         Avaa_kokoelma()
248         selain.go_back()

```

Kuva 4: `Etsi_kokoelma`

6.3.1 Toinen olio

Sisällä kokoelmassa on kaikki tekijän teokset eli myynnissä olevat ja loppuunmyytyt teokset. Näistä haluttiin vain myynnissä olevat, joten kuvista etsittiin ne joilla on loppuunmyytyille tehty luokka ja niistä tehtiin lista. Tästä listasta tarvittiin kuitenkin vain pituus eli loppuunmyytyjen teosten määrä, joka vähennettiin kaikkien teosten listan pituudesta, jolloin saatiin pelkästään myynnissä olevien teosten lista. Loppuunmyytyt teokset olivat nimittäin järjestyksessä kaikki listan perällä. Listan ollessa valmis tehtiin uusi while-looppi, joka painaa jokaisen listan jokaisen teoksen vuorollaan auki ja ajaa seuraavan funktion nimeltä `Talleta_tiedot`, joka tallettaa vuorollaan teosten tiedot ja sen jälkeen palaa kokoelmaan. Kuvassa 5 toinen olio `Avaa_kokoelma`.

```

201  def Avaa_kokoelma():
202
203      teokset = selain.get_webelements(locator="class:store-item-image")
204      loppuunyydyt = selain.get_webelements(locator="class:store-item-sold-out")
205      miinus = len(loppuunyydyt)
206
207      kerta = 0
208      pituus2 = len(teokset)-miinus
209
210      while kerta < pituus2:
211          klikkaa = teokset[kerta]
212          selain.click_element(klikkaa)
213
214          kerta+=1
215          Talleta_tiedot()
216          selain.go_back()
217
218      loppuunyydyt == 0
219      kerta == 0

```

Kuva 5: Avaa_kokoelma

6.3.2 Kolmas olio

Seuraava suurempi funktio ottaa talteen kaikki tiedot jokaisen taiteilijan jokaisesta teoksesta eli koko kokoelmasta. Se aloittaa hakemalla teoksen sivulta sen nimen, joista sen täytyi korvata muutama erikoismerkki vastaavilla eri merkkikirjaston merkeillä, joita tallennettu PDF kieltäytyi tunnistamasta. Aiemmassa versiossa korvattiin myös ä ja ö-kirjaimet ennenkuin ne opittiin lisäämään PDF tiedoston merkistöön. Sen jälkeen otettiin talteen vaaditut tiedot eli kuvaus, kuva teoksesta sen alkuperäisessä resoluutiossa omalta sivultaan, hinta, ALV-teksti, tieto montako teosta on tarjolla - jos saatavilla - ja päivämäärä URL:ista jolloin teos on lisätty sivulle. Senjälkeen suoritettiin olio joka luo PDF-tiedoston kaikista tiedoista nimeltä Luo_PDF. Kuvat päätettiin tallentaa erikseen ja liittää niiden tiedostonnimet tietoihin. Kuvassa 6 olio Talleta_tiedot.

```

68 def Talleta_tiedot():
69
70     Teosnimi = selain.get_text('class:product-name.break-word')
71     string = str(Teosnimi)
72
73     #Matti Hintikan teoksen Ve'en edessä nimeä täytyi muokata PDF varten
74
75     korjaus = string.replace(' ', '')
76     string = korjaus
77     korjaus = string.replace('-', '-')
78     Teosnimi = korjaus
79     print(Teosnimi)
80
81     Kuvaus = selain.get_text('class:product-description.col-xs-12')
82     print(Kuvaus)
83
84     selain.set_screenshot_directory('Kuvat')
85
86     #Tallennusfunktio
87
88     kohde = selain.get_element_attribute('class:carousel-bgimage', 'style')
89     kohde = kohde[23:-3]
90     print(kohde)
91
92     selain.go_to(kohde)
93
94     selain.wait_until_element_is_visible('css:img')
95
96     selain.screenshot('css:img', 'Taideteos.jpg')
97     selain.go_back()
98
99     Hinta = selain.get_text('class:product-price')
100
101     string = str(Hinta)
102     korjaus4 = string.replace('€', 'e')
103     Hinta = korjaus4
104
105     print(Hinta)
106
107     ALV = selain.get_text('class=vat_notice.smallp')
108     print(ALV)
109
110     Tarjolla = selain.get_text('class:thincaps.product-item-stock')
111     print(Tarjolla)
112     print('\n')
113
114     Päivämäärä = selain.get_element_attribute('class:carousel-bgimage', 'style')
115     Päivämäärä = 'Lisatty v/kk/pp '+Päivämäärä[67:77]
116
117     global kuva
118     kuva = "Taideteos.jpg"

```

Kuva 6: Talleta_tiedot

6.3.3 Neljäs olio

Neljäs olio Luo_PDF ajetaan aina joka teoksen kohdalla kun sen kuva on tallennettu. Kuva nimetään ja sen nimi lisätään PDF-tiedoston alkuun, jotta sen tiedot voidaan helposti yhdistää kuvaan. Olio kerää sivustolta tallennetut tiedot ja lisää ne rivi kerrallaan PDF-tiedostoon. Lopuksi tiedosto nimetään numerojärjestyksessä ja kuva tallennetaan erikseen ja nimetään numerojärjestyksessä. Senjälkeen siirrytään seuraavaan kohtaan listassa, kunnes kaikki teokset on käyty läpi. Kuvassa 7 esitellään Luo_pdf olio.

```

120 def Luo_PDF():
121
122     global kierros
123
124     print(kierros)
125
126     pdf = FPDF()
127     pdf.add_page()
128
129     pdf.set_font('Arial', '', 12)
130
131     merkki = str(kierros)
132
133     pdf.cell(200, 10, txt='Tekniset tiedot', ln=True)
134     pdf.cell(200, 10, txt='Special characters: ä, ö', ln=True)
135     pdf.cell(200, 10, txt='Kuvan nimi', ln=True)
136     pdf.cell(200, 10, txt='Taideteos'+merkki+'.jpg', ln=True)
137     pdf.cell(200, 10, txt='Teoksen tiedot', ln=True)
138     pdf.cell(200, 10, txt=Teosnimi, ln=True)
139     pdf.cell(200, 10, txt=Kuvaus, ln=True)
140     pdf.cell(200, 10, txt=Päivämäärä, ln=True)
141     pdf.cell(200, 10, txt=Hinta, ln=True)
142     pdf.cell(200, 10, txt=ALV, ln=True)
143     pdf.cell(200, 10, txt=Tarjolla, ln=True)
144
145     tiedosto = 'Teostiedot'+merkki+'.pdf'
146
147     pdf.output(tiedosto)
148
149     #Kuva JPG
150     with Image.open(kuva) as img:
151
152         RGB = img.convert('RGB')
153
154         RGB.save('Taideteos'+merkki+'.jpg', format='JPEG')
155
156     kierros+=1

```

Kuva 7: Luo_PDF

Lopuksi näytetään vielä valmis ajo- kuva toimivasta robotista kuvassa 8. Kuvassa näkyy esimerkkitulostus sivustolta scrapetusta tiedosta tekstimuodossa, jossa se tallennetaan PDF-tiedostoon. Tulostuksessa näkyy myös yksittäisten teosten laskettu määrä sekä viimeinen olio, jonka tehtävä on sulkea selain suorituksen päättyessä ja tulostaa ”Selaimet suljettu”.

```
200
Tapani Mikkonen - Koli II
litografia, 2012, 56x76cm
https://cdn.holvi.com/media/poolimage_image/2023/11/21/c5157f68d13fff36b0cbbf53df7c8f71e46d777c_1100x1100_q85.jpg
600,00 e
Sis. ALV 10,00%

201
Selaimet suljettu
Kerätiedot status: PASS
=====
Robocorp Log (html): c:\Users\Henri\Desktop\Henrikansio 3.0\GitHub\Ohjelmistorobotiikka\Robotit\Grafia13 Robotti\output\log.html
```

Kuva 8: Valmis ajo ja esimerkkitulostus tiedoista

6.3.4 Kehitysehdotukset

Robotin ollessa valmis ja viimeistellesäni kirjoitusprosessia saatiin vielä kehitysehdotus verkkokauppaa suunnittelevalta tiimiltä. He toivoivat että vanhan verkkokaupan tiedot saisi myös CSV-tiedoston muodossa. Robottiin kehitettiin ratkaisu, jolla PDF-tiedostoon menevistä tiedoista tehdään myös erillinen CSV-tiedosto. CSV-tiedosto kootaan samalla kun tiedot otetaan talteen PDF-tiedostoa varten ohjelmistokoodissa. Kuvassa 9 näkyy uusiin kehitetty olio nimeltä CSV-luoja.

```

169 #CSV tiedoston luonti
170
171 teksti0 = 'Taideteos'
172 teksti1 = merkki
173 teksti2 = '.jpg'
174
175 sisältö0 = [teksti0+teksti1+teksti2]
176 sisältö1 = [Teosnimi]
177 sisältö2 = [Kuvaus]
178 sisältö3 = [Päivämäärä]
179 sisältö4 = [Hinta]
180 sisältö5 = [ALV]
181 sisältö6 = [Tarjolla]
182
183 tiedosto1 = 'GrafiaTeos'+str(kierros)+'.csv'
184
185 global tiedostonimi
186 tiedostonimi = tiedosto1
187
188 with open(tiedosto1, 'w', newline='') as tiedosto1:
189     kirjoittaja = csv.writer(tiedosto1)
190     kirjoittaja.writerow(sisältö0)
191     kirjoittaja.writerow(sisältö1)
192     kirjoittaja.writerow(sisältö2)
193     kirjoittaja.writerow(sisältö3)
194     kirjoittaja.writerow(sisältö4)
195     kirjoittaja.writerow(sisältö5)
196     kirjoittaja.writerow(sisältö6)
197
198 print(f"CSV {tiedosto1} valmis")

```

Kuva 9: CSV-luoja

7 Pohdinta

Heti aluksi oli syytä selvittää olisiko sellaista robottia mahdollista toteuttaa, jonka toimeksiantaja halusi. Samalla kun toimeksiantajan toimeksianto ja sivuston rakenne selvisivät kävi ilmi että robotti olisi hyvinkin mahdollinen toteuttaa RPA:n keinoin. Robotti tehtiin viikon pituisissa sprinteissä ja sen toimivuutta testattiin ajamalla sitä samalla kun sitä kehitettiin eteenpäin. Apua sai Robocorpin ReMark chatbotilta, jolta sai nopeasti selvitettyä miten tarvittavat kirjastot ja niiden syntaksi toimi.

Työskennellessä Agile menetelmien seuraaminen näkyi eniten muuttuvien vaatimusten ja uusien versioiden huomioimisessa koko kehityskaaren ajan. Ohjelmistoa lyhyen pituisissa sprinteissä mikä on tyypillistä ketterälle kehitykselle ja oltiin jatkuvasti yhteydessä ohjaavaan opettajaan mahdollisten muutosten ja valmiiden versioiden arvioimista varten. Pidimme lähes

viikoittaisia palavereja kirjoitustyön ja kehitystyön seurantaan varten. Ohjelmistoa muokattiin palautteen perusteella ja päätavoitteena pysyi toimivan ohjelmiston varmistaminen. Sprinteissä oli noin viikko väliä ja ensimmäinen kokonainen loppuun ajettava versio oli valmis kahden sprintin jälkeen. Kolmannella sprintillä parannettiin entisestään robotin rakennetta, kun saatiin parempi ymmärrys miten web-elementit ja locatorit toimivat yhdessä. Tärkeää oli tunnistaa oikeat web-elementit, joilla pääsi käsiksi oikeisiin kuviin ja tietoihin, joista taas sai tehtyä siistit listat.

8 Yhteenveto

Jälkeenpäin ajatellen olisi ehkä kannattanut pidempään suunnitella robottia ja tutkiskella sivustoa ja sen Javascript-rakennetta. Web-elementtien rakenteen selvittäminen oli kriittinen askel robotin navigaation toteutuksessa ja sen avulla oppi paljon sekä ohjelmistorobotiikasta että verkkosivujen kehityksestä. Myös robotin rakennetta olisi voinut miettiä ja suunnitella aluksi enemmän, koska siihen löytyi ainakin kaksi eri tapaa toteuttaa eli taskien tai pelkkien funktioiden avulla. Normaalisti robotit rakennetaan noudattamaan pieniä taskeja, mutta ohjelmat rakennetaan noudattamaan sisäkkäisiä funktioita ja muuttuja sekä loopeja. Sisäkkäisten funktioiden jakaminen omiin taskeihinsa siten että kaikki olisivat käyttäneet samoja muuttujia loopeissa vaatisi vielä enemmän ohjelmointiosaamista ja tietämystä ohjelmistorobotiikasta kuin tällä hetkellä on käytössä.

9 Lähteet

Agilemanifesto 2024. Manifesto for Agile Software Development. Viitattu 18.4.2024.

<https://agilemanifesto.org/>

Automation Anywhere 2024. What is Robotic Process Automation (RPA)? Viitattu 18.4.2024.

<https://www.automationanywhere.com/rpa/robotic-process-automation->

Grafia13 2024. Viitattu 26.3.2024. <https://holvi.com/shop/grafia13/>

Nintex 2024. Automate repetitive tasks with RPA. Viitattu 18.4.2024.

<https://www.nintex.com/process-automation/robotic-process-automation/>

Pega 2024. RPA software that scales beyond tasks. Viitattu 18.4.2024.

<https://www.pegacom/products/platform/robotic-process-automation>

Python.org 2024. Viitattu 6.4.2024. <https://www.python.org/about/>

- Robocorp 2024. Robocorp Remark. Viitattu 29.4.2024. <https://robocorp.com/blog/robocorp-launches-remark%0%9f%92%ac-a-generative-ai-assistant-for-automation-developers-2>
- Robocorp 2024. Robocorp.windows. Viitattu 6.4.2024. <https://robocorp.com/docs/python/robocorp/robocorp-windows>
- Taulli, T. 2020. The robotic process automation handbook : A guide to implementing rpa systems. Apress L. P. [ProQuest Ebook Central - Book Details](#)
- Tietoarkisto 2024. Avoin haastattelu. Viitattu 18.4.2024. https://www.fsd.tuni.fi/menetelmaopetus/kvali/L6_3_1.html
- UIPath 2024. Intelligent process automation. Viitattu 18.4.2024. <https://www.uipath.com/rpa/intelligent-process-automation>
- w3schools.com. Viitattu 6.4.2024. https://www.w3schools.com/python/python_intro.asp

10 Kuvat

Kuva 1: Kokoelmien läpikäyjä-looppi	21
Kuva 2: Kahden listan erotus looppi.	22
Kuva 3: Robotin työkulku	23
Kuva 4: Etsi_kokoelma	24
Kuva 5: Avaa_kokoelma	25
Kuva 6: Talleta_tiedot	26
Kuva 7: Luo_PDF	27
Kuva 8: Valmis ajo ja esimerkkitulostus tiedoista	28
Kuva 9: CSV-luoja	29