Opinnäytetyö AMK

Konetekniikka

2023

Ville Huopainen

Laskutuksen automatisointi ohjelmistorobotiikan avulla

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Opinnäytetyö AMK | Tiivistelmä

Turun ammattikorkeakoulu

Konetekniikka

2023 | 22 sivua

Ville Huopainen

Laskutuksen automatisointi ohjelmistorobotiikan avulla

Opinnäytetyön tavoitteena on kehittää automatisointiratkaisu ohjelmistorobotiikkaa käyttäen. Tavoitteena on myös dokumentoida ja ottaa käyttöön suunniteltu ohjelmistorobotti.

Opinnäytetyön toteuttamisen työkaluksi valittiin on UiPath Studio, koska se on käytössä toimeksiantoyrityksessä. UiPath Studion komponenteilla voidaan kasata ohjelmistorobotti, joka matkii ihmisen toimia tietokoneella. Dokumetointityökaluksi valikoitui UiPath:in Task Capture, joka on käytössä toimeksiantoyrityksessä. Dokumentointi suoritetaan rakentamalla robotin toiminnasta vuokaavio, jota täydennetään kuvin ja selityksin.

Tuloksena opinnäytetyöstä saatiin robotti, joka hakee henkilöille Toggl-tuntikirjausjärjestelmästä laskutettavat työt ja muodostaa niistä laskutuspohjan Google Driveen. Lisäksi kyseinen prosessi dokumentoitiin Task Capturella.

Opinnäytetyön lopuksi kehitetty ohjemistorobotti ajettiin suorittamaan tehtävä. Toimeksiantaja hyväksyi robotin tekemän työn tulokset.

Asiasanat:

Ohjelmistorobotiikka, UiPath, Task Capture, Rajapinta, Google Drive, Toggl

Bachelor’s Thesis | Abstract

Turku University of Applied Sciences

Mechanical Engineering

2023 | 22 pages

Ville Huopainen

Automation of invoicing with Robot Process Automation

The aim of the thesis was to develop an automation solution using robotic process automation. The goal was also to document and implement the planned software robot.

UiPath Studio was chosen as the tool for implementing the thesis, because it is used by the company that commissioned the thesis. A software robot that imitates human tasks on the computer can be created with the components of UiPath Studio.

UiPath’s Task Capture, which is used in the company, was selected as the documentation tool. The documentation process was performed by building a flowchart of the robot’s operation, which is supplemented with pictures and explanations.

The output of the thesis was a robot that retrieves invoiced work from Toggl, a time tracking software, and creates an invoicing template in Google Drive. The process in question was documented with Task Capture.

As a result of the thesis, the software robot was run to complete the task. The company approved the results of the robot’s work.

Keywords:

RPA. UiPath, Task Capture, API, Google Drive, Toggl

Sisältö

Käytetyt lyhenteet tai sanasto **6**

1 Johdanto **7**

2 Ohjelmistorobotiikka ja ohjelmointirajapinta **8**

2.1 Ohjelmistorobotiikan hyödyt 8

2.2 Ohjelmointirajapinta 8

3 Robotin suunnittelu ja kehitys **9**

3.1 Lähtökohta 9

3.2 Arkkitehtuuri 9

3.3 Robotin kehitys 11

4 Robotin testaus, dokumentointi ja käyttöönotto **19**

4.1 Robotin testaus 19

4.2 Dokumentointi 19

4.3 Käyttöönotto 19

5 Yhteenveto ja pohdinta **20**

Lähteet **22**

Liitteet

Liite 1. Liitteen otsikko.

Kuvat

Kuva 1. Vuokaavio robotti 1:n toiminnasta. Tehty Task Capturella. 10

Kuva 2. Alkudata-Excel 11

Kuva 3. Laskutuspohja 11

Kuva 4. Alkutiedon haku 13
Kuva 5. Laskutuspohjan hakeminen 14
Kuva 6. Kuukausi-infon haku 15
Kuva 7. Toggl-järjestelmästä haettu yhteenveto 15
Kuva 8. Toggl-tiedoston käsittely 16
Kuva 9. Tiedonsiirto Google Driveen 18

Käytetyt lyhenteet tai sanasto

Lyhenne	Lyhenteen selitys
API	Application programming interface, ohjelmointirajapinta
Config	Configuraton file, palvelinkokoonpanotiedosto
RPA	Robotic process automation, ohjelmistorobotiikka
URL	Uniform Resource Locator, verkko-osoite
Xaml	Extensible Application Markup Language, Xaml-merkintäkieli

1 Johdanto

Opinnäytetyössä toteutetaan laskutusautomaatiorobotti UiPath Studio -työkalulla. Tämä opinäytetyö toteutetaan Rajobit Oy:lle, joka on kasvava ohjelmistorobotteja palveluntarjoajana tuottava teknologiayritys.

Ohjelmistorobotiikka (robotic process automation, RPA) on nouseva teknologia, jossa tietokoneohjelma jäljittelee ihmisen toimintaa tietokoneella. Näin pystytään automatisoimaan rutiininomasia toimintoja, kuten tietojen syöttöä järjestelmästä A järjestelmään B. RPA vapauttaa aikaa ihmisen manuaalisista töistä, eli ihmiset voivat keskittyä monimutkaisempiin ja arvokkaampiin työtehtäviin. Lisäksi ohjelmistorobotti on nopeampi kuin ihminen, eikä se tee virheitä. RPA tarjoaa skaalaavuutta ja joustavuuta liiketoiminnalle. Sen avulla voidaan nopeasti reagoida kasvavaan työmäärään ja muuttuviin vaatimuksiin ilman suuria investointeja uusiin järjestelmiin tai ohjelmistohin.

Kiinnostukseni ohjelmistorobotiikkaa kohtaan heräsi viime vuonna. Aloin selvittämään, mihin kaikkeen sitä voi hyödyntää ja miten. Huomasin ohjelmistorobotiikan lisäävän työn mielekkyyttä poistamalla aikaavieviä rutiineja sekä parantavan työn tuottavuutta. Opinnäytetyön avulla saan suunniteltua ja toteutettua automaatioratkaisun toimeksiantajalle ja näin poistamaan rutiinin, joka vie useita tunteja aikaa kehittäjälaskutuksen tekijöiltä. Lisäksi opinnäytetyö ohjelmistorobotiikasta saattaa tuoda teknologian sellaisten ihmisten tietoon ketkä eivät sitä vielä tiedä käyttää.

Opinnäytetyössä käsitellään aluksi ohjelmistorobotiikkaa, rajapintaa sekä sen käsitteitä ja tavoitteita.

2 Ohjelmistorobotiikka ja ohjelmointirajapinta

Ohjelmistorobotiikassa robotti tarkoittaa ohjelmistopohjaista järjestelmää, joka matkii ihmisen toimia. Lyhyesti ohjelmistorobotiikka sisältää ohjelmiston käytön, joka jäljittelee ihmisen toimintaa kun se on vuorovaikutuksessa tietokoneen sovelluksen kanssa ja suorittaa sääntöihin perustuvia tehtäviä. (Tripati 2018, 9-10)

2.1 Ohjelmistorobotiikan hyödyt

Ohjelmistorobotiikka tarjoaa korkeampilaatuista palvelua ja parempaa tarkkuutta. Robotti tekee vähemmän inhimillisiä virheitä ja noudattaa sääntöjä, joten työn laatu on parempaa. Lisäksi inhimillisen virheen löytäminen ja korjaaminen on helpompaa, koska robotti kertoo mikäli virhe on tapahtunut. Tämä taas lisää parempaa tiedon hallintaa, joka tuottaa parempia analyyseja ja parempaa ongelmanratkaisua. (Tripati 2018, 12)

Ohjelmistorobotti vaatii pienemmät kustannukset kuin ihminen. Ohjelmistorobotti vastaa kolmea ihmistä; robotti työskentelee 24 tuntia vuorokaudessa ja ihminen kahdeksan. Käytettävyyden ja tuottavuuden ansiosta toimenpiteen kustannukset ovat laskeneet huomattavasti. Lisäksi ohjelmistorobotti on nopea. Se pystyy toimimaan niin nopeasti kuin sivusto tai järjestelmä pystyy vastaamaan. (Tripati 2018, 12-13)

2.2 Ohjelmointirajapinta

Ohjelmointirajapinta on joukko sääntöjä, joiden avulla sovellukset voivat kommunikoida keskenään. Se toimii kahden sovelluksen välillä. Tässä tapauksessa UiPath on käyttösovellus, johon haetaan rajapinnan kautta halutut tiedot OneDrive:sta ja tulevaisuudessa Toggl-järjestelmästä. (IBM 2023. What is an API?)

3 Robotin suunnittelu ja kehitys

3.1 Lähtökohta

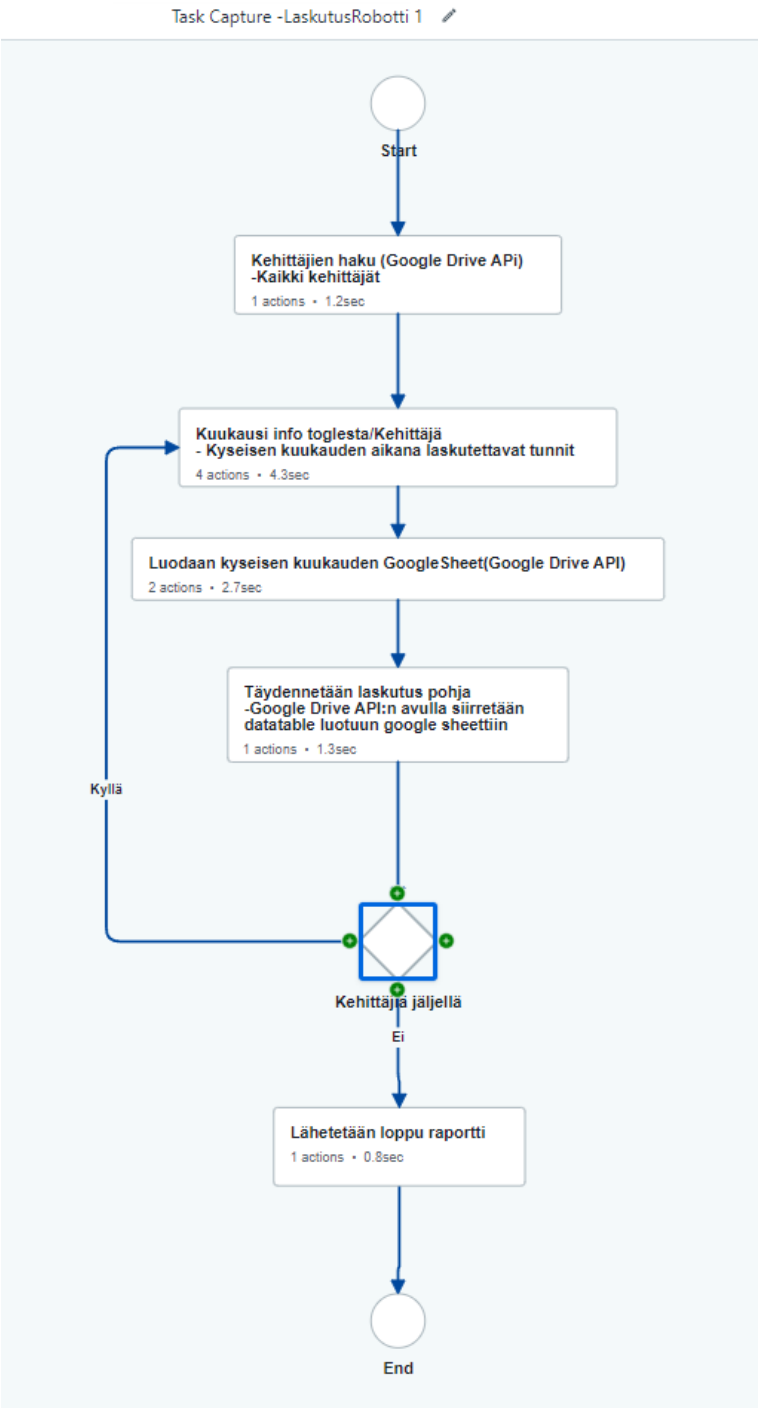
Toimeksiantaja tarvitsee robotin, jolla voidaan automatisoida laskutus. Tällä hetkellä laskutus vie ihmisen aikaa noin vuorokauden kuukaudessa. Prosessissa kehittäjät aluksi hakevat Toggl-järjestelmästä Excel:iin laskutettavat työnsä ja sieltä tiedot viedään Talenom Onlineen, jossa muodostetaan lasku asiakaskohtaisesti. Lisätyötä tässä luo asiakaalle se että useampi kehittäjä voi laskuttaa samaa asiakasta.

Olen suunnitellut automatisoinnin koostuvan kahdesta robotista. Ensimmäinen robotti täyttää kehittäjien laskutus-Excelit eli muodostaa kyseiselle kuukaudelle laskutuspohjan Google Driveen.

Tämän jälkeen toinen robotti hakee täydennetyn Excelin, jonka kehittäjä on todennut paikkansapitäväksi ja ottaa siitä ylös laskutettavan asiakkaan ja laskutettavan hinnan. Tämän jälkeen robotti vie tiedot Talenom Onlineen ja tarkistaa onko asiakkaalle avointa laskua. Jos näin ei ole, robotti luo uuden laskun. Opinnäytetyö keskittyy tämän ensimmäisen robotin kehittämiseen.

3.2 Arkkitehtuuri

Arkkitehtuuri robottiin suunnitellaan UiPath:in Task Capturella. Kyseessä on UiPath:in oma sovellus, joka on vuokaavion kaltainen suunnittelu- ja dokumentointiohjelma. Robotin alkuvaihessa ohjelmaan laitetaan oma idea vuokaaviona. Alla kuva kehittäjälaskutusrobotin ensimäisestä osuudesta (Kuva 1.).



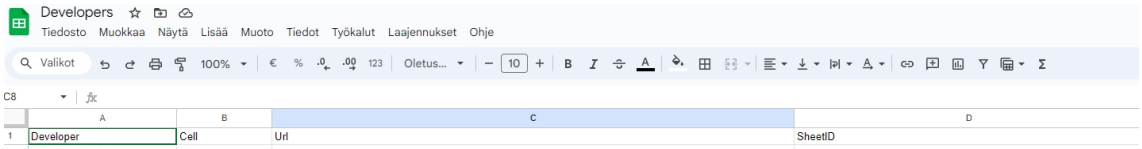
Kuva 1. Vuokaavio robotti 1:n toiminnasta. Tehty Task Capturella.

Robotti kehitetään toimeksiantajan omaan pohjaan. Tämä tarkoittaa sitä, että robottiin ei tarvitse tehdä kaikkea alusta alkaen. Kehitykseessä luodaan 1) sisään- ja uloskirjautuminen Toggl-järjestelmään, 2) tuntikirjausten haku Toggl-

järjestelmästä sekä 3) Excel-tiedostojen haku ja takaisinvienti Google Drive:sta. Käytän kehittämisessä Google Workspacen omaa rajapintaa, johon on jo tehty valmiiksi komponentit. Lopuksi sijoitan kehtittämäni xaml:it pohjan oikeisiin kohtiin ja täydennän argumentit.

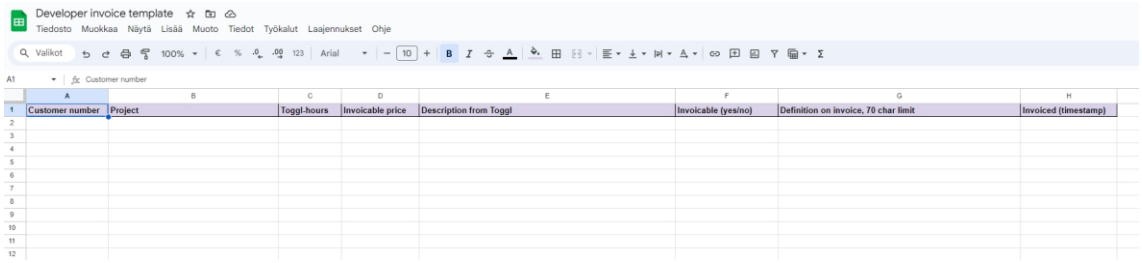
3.3 Robotin kehitys

Aloitin robotin kehtityksen kehittämällä alkudata-välilehden (Kuva 2.), josta löytyvät kehittäjät, tarvittava URL-osoite sekä SheetID:t, jolla löydetään kyseisen kehittän laskutus-Excel.



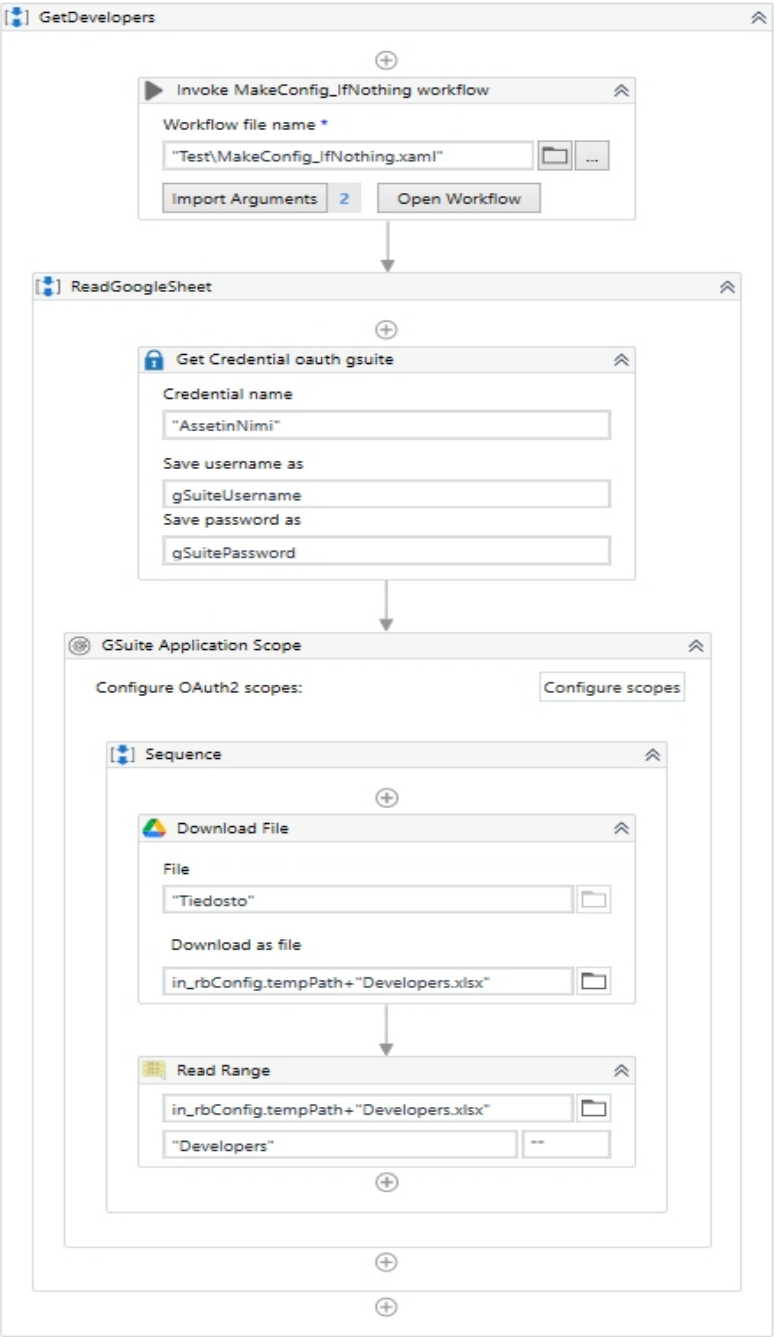
Kuva 2. Alkudata-Excel

Seuraavaksi kehitin laskutuspohjan (Kuva 3.), jota käytetään jokaisessa laskutusyhteenvedossa. Siitä löytyy asiakasnumero, projektin nimi, Toggl-tunnit, laskutettava hinta, kuvaus Togglesta, tieto laskutettavuudesta, laskulle tuleva teksti ja aikaleima jolloin kyseinen rivi on laskutettu.



Kuva 3. Laskutuspohja

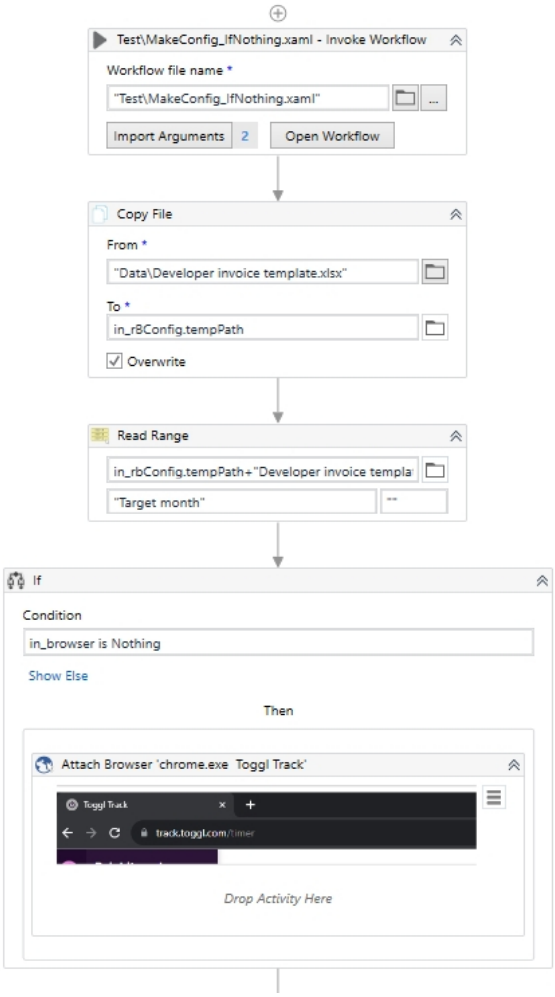
Aloitin kehittämällä Xaml:in, jossa haetaan Googlen rajapitaa käyttäen Developers-välilehti. Googlen rajapintakomponentit ovat toimeksiantajan omat. Xaml:in nimesin toimeksiantajan käytänteiden mukaisesti ”GetDevelopers”. Ensiksi haen Orchestratorista käyttäjän ja salasanan. Sen jälkeen toimeksiantajan rajapintakomponentilla haetaan haluttu tiedosto download file -komponentilla, johon täydennetään fileID joka saadaan kyseisen välilehden URL:sta. Tiedosto tallennettaan hyvien käytänteiden mukaisesti config:issa olevaan väliaikaiseen kansioon. Lopuksi tallennetu Excel luetaan, ja siitä viedään kehittäjät robotin työjonoon. (Kuva 4.)



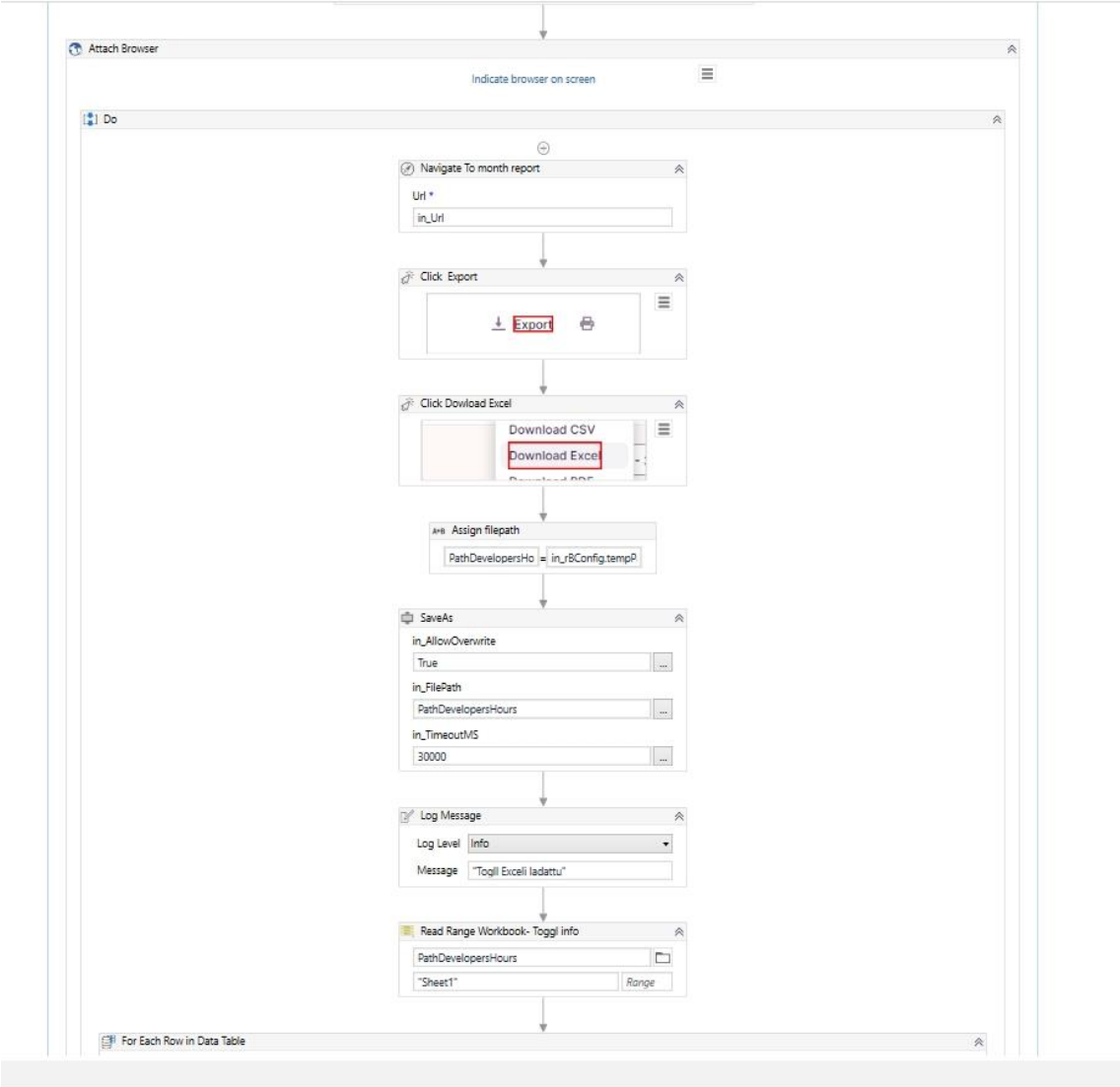
Kuva 4. Alkutiedon haku

Vaiheesa kaksi, jossa haetaan tiedot Toggl-järjestelmästä, kehitin Xaml:in GetDevelopersHoursInfo. Aluksi haen robotin datakansioon tallennetun laskutuspohjan (Kuva 3. ja Kuva 5.), jonka luen datataulukoksi Toggl-järjestelmän tietoja varten. Seuraavaksi navigoin Developers-tiedostosta saadun kehitäjäkohtaisen linkin avulla (Kuva 6.) Togglessa

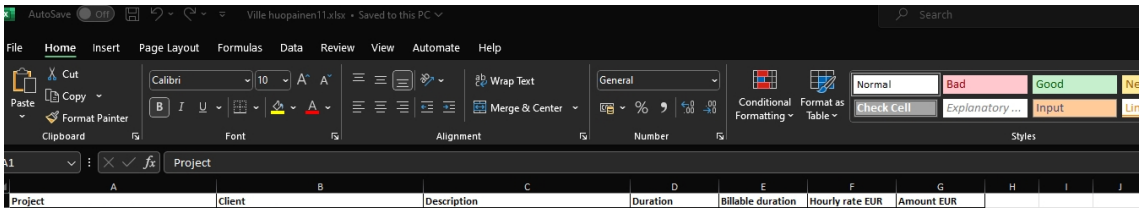
kuukausiyhteenvetoon (Kuva 7.) ja lataan sen. Tiedosto tallennetaan toimeksiantajan omalla SaveAs-komponentilla. Ladatun tiedoston luen myös datataulukkoon (Kuva 6.).



Kuva 5. Laskutuspohjan hakeminen

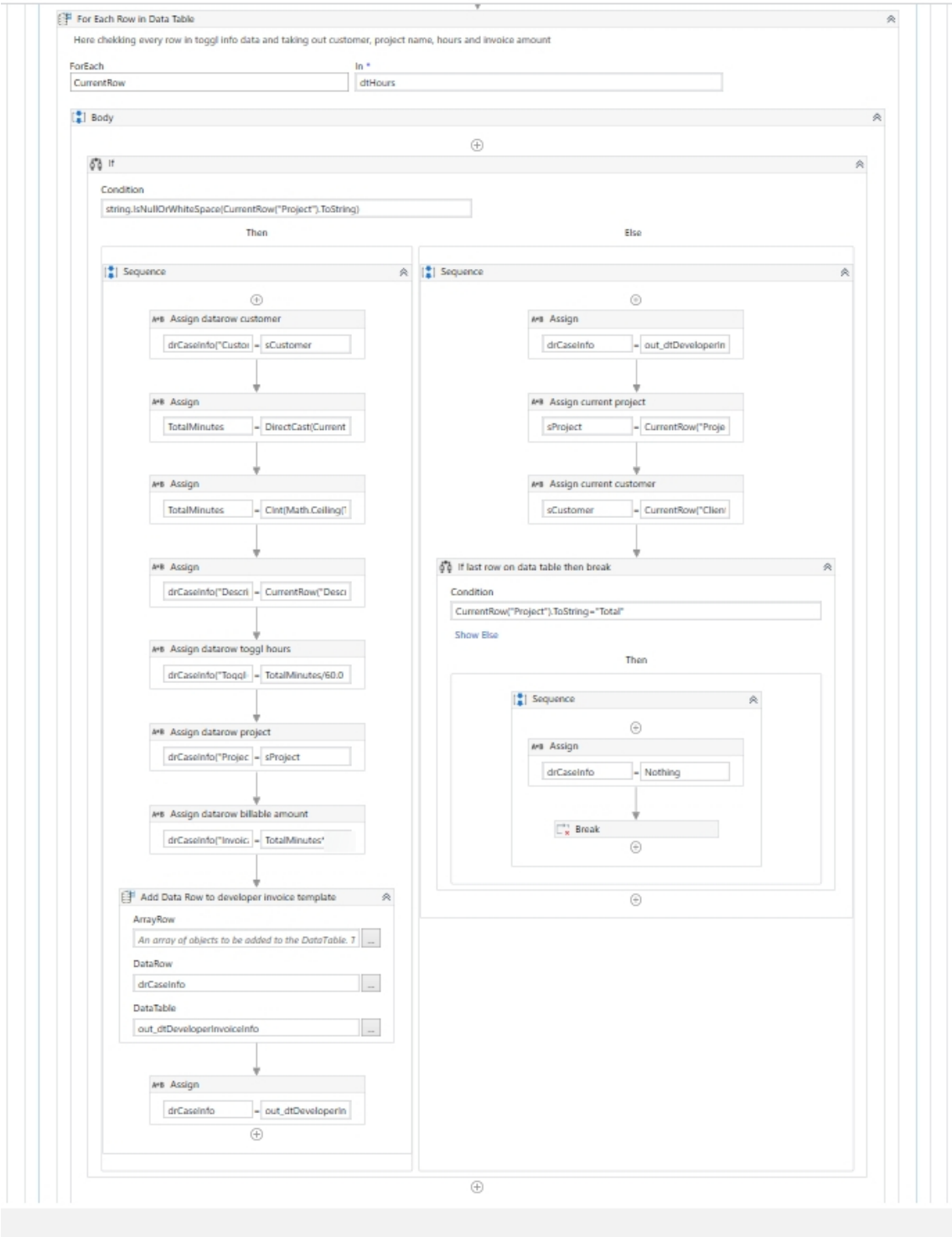


Kuva 6. Kuukausi-infon haku



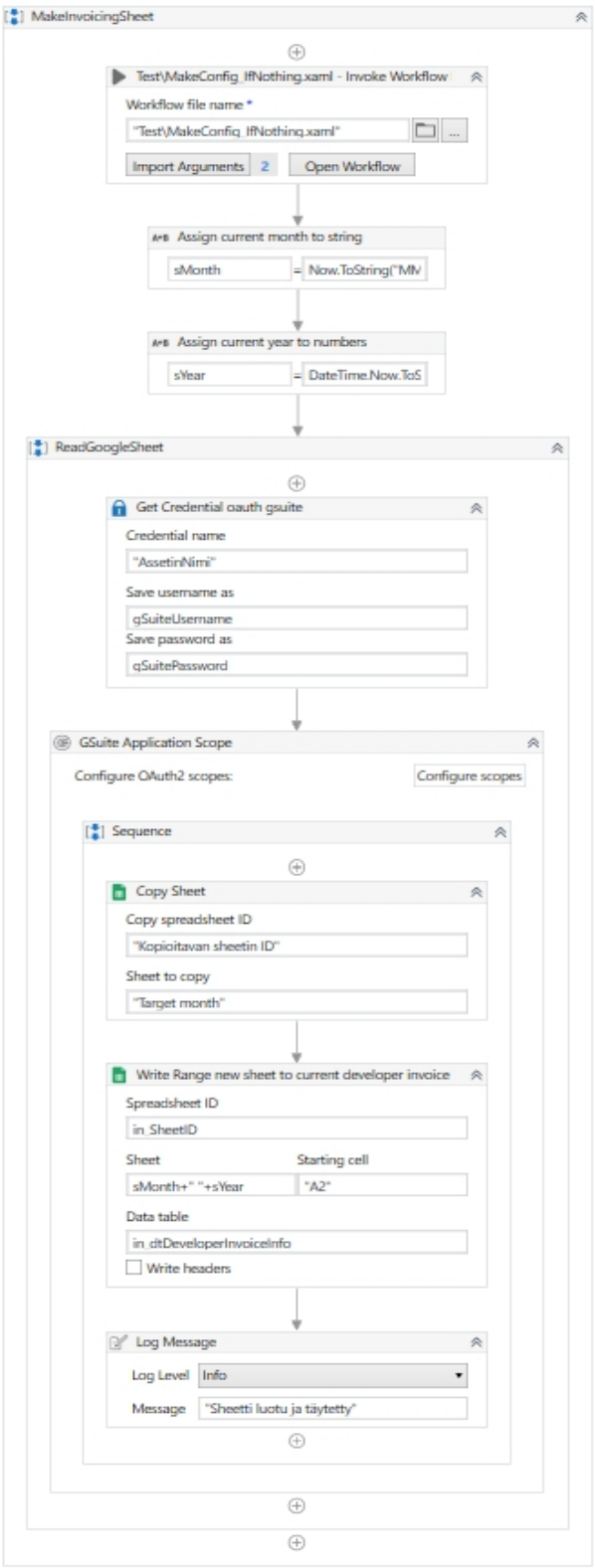
Kuva 7. Toggl-järjestelmästä haettu yhteenveto

Taulukosta (Kuva 7.) haen rivikohtaisesti Project-, Client description- ja Duration -tiedot. Tämän pohjalta käyn koko taulukon läpi ja lisään halutut tiedot datariviin. Datarivin kirjoitan Laskutuspohja-tiedostosta saatuun datataulukkoon (Kuva 8.).



Kuva 8. Toggl-tiedoston käsittely

Viimeisessä osiossa kehitin Xaml:in, jossa siirretään Toggl-tietojen pohjalta luotu datataulukko kehittäjän välilehteen. Xaml on nimetty ”MakeInvoicingSheet” toimeksiantajan käytänteiden mukaisesti. Tässä Xaml:ssa otan muuttujiin talteen kuluvan kuukauden ja vuoden, jonka jälkeen haen Orchestrator-palvelimesta käyttäjätunnuksen ja salasanan. Niillä pääsen käyttämään Microsoftin rajapintaa. Rajapinnalla haen jo aikaseimmin mainitun laskutuspohja-Excelin ja kopioin sen kyseisen kehittäjän Excel-tiedostoon uudeksi välilehdeksi. Lopuksi siirrän aikaisemmin luodun datataulukon (Kuva 8.) juuri luodun pilvipohjaisen Excelin päälle.



Kuva 9. Tiedonsiirto Google Driveen

4 Robotin testaus, dokumentointi ja käyttöönotto

Robotin testaus tehdään samanaikaisesti kehittämisen yhteydessä. Dokumentointi tehdään, kun robotti on kehitetty alustavasti. Robotti otetaan käyttöön, kun se on testattu.

4.1 Robotin testaus

Robotin testaus suoritetaan toimeksiantajan hyvien käytänteiden mukaan, eli jokaisen Xaml:in on oltava itsenään uudelleen ajettava. Testaus suoritetaan kolmessa vaiheessa Xaml-tasolla. Ensin jokainen Xaml ajetaan yksinään läpi. Toiseksi suoritetaan prosessitason testaus. Sitten kehitetty robotti testiajetaan alusta loppuun, minkä jälkeen se voidaan ajaa tuotannossa. Robottiin on tähän mennessä tehty kolme Xaml-tason testiä, prosessitason testit sekä tuotantotesti. Tuotantotesti suoritettiin ensimmäisen tuotantoajon yhteydessä. Tuotantotesti meni onnistuneesti läpi ja lopputulos oli haluttu.

4.2 Dokumentointi

Robotti on dokumentoitu Task Capture:lla. Aiemmin tehdyssä arkkitehtuurin suunnittelussa on dokumentointipohja. Jokaiseen vuokaavion tehtävään on selitetty ja havainnollistettu mitä niissä tehdään. Dokumentista tuli usean sivun pituinen. Sen löytää Liitteestä 1.

4.3 Käyttöönotto

Robotti 1 otettiin käyttöön testauksen yhteydessä 1.12.2023. Robotti ajettiin valvotusti, ja robotin tekemän työn oikeellisuus tarkastetiin. Otettiin myös talteen mahdolliset korjaukset versiota 1.0.1 varten. Robotti on otettu käyttöön, mutta lisävaatimuksia en ole vielä saanut. Versio 1.0.1 otetaan käyttöön seuraavaan ajoon.

5 Yhteenveto ja pohdinta

Opinäytetyön aiheena oli kehittää toimeksiantajalle ohjelmistorobotti, joka automatisoi kehittäjälaskutuksen. Alkupohdinnoissa todettiin, että automatisointi tarvitsee tehdä kahdella robotilla, jotta prosessi olisi mahdollisimman luotettava, hallittava ja helposti käyttöön otettava. Ensimmäinen robotti täydentää kehittäjä-Excelin, jonka jälkeen toinen robotti vie tarkastetusta kehittäjä-Exceistä tiedot Talenom Onlineen ja luo laskut Excelien pohjalta. Tämän opinnäytetyön puitteissa kehitettiin näistä roboteista ensiksi mainittu. Toinen robotti tullaan kehittämään vuoden 2024 aikana.

Ensimmäinen robotti on saatu kehitettyä ja testattua. Tuotantoajo suoritettiin valvotusti, ja robotti on todettu toimivaksi. Toimeksiantaja on hyväksynyt robotin ja on tyytyväinen sen toimintaan. Voimme siis todeta, että ensimmäinen robotti oli onnistunut ja lisäarvoa tuottava. Seuraavaksi aloitetaan toisen robotin suunnittelu ja toteutus.

Robotin suunnittelu oli helppo ja selkeä prosessi. Kaikki vaiheet kehittämisestä testaukseen ja dokumentointiin menivät suunnitelmien mukaan ja aikataulussa. Robotin kehitys onnistui hyvin, mutta olisin voinut aloittaa sen kehittämisen aikaisemmin, jotta olisin ehtinyt tekemään sen käyttäen Toggl-rajapintaa. Robotti on kuitenkin nykyisellään riittävä ja vastaa toimeksiantajan vaatimuksia sekä tarpeita. Työn laajuus on opinnäytetyöksi sopiva.

Robotin käyttöönotto sujui myös suunnitellusti eli valvotuissa olosuhteissa ja tulokset tarkastaen. Käyttöönottoprosessi tehtiin toimeksiantajan asettamien hyvien käytänteiden mukaisesti, ja se oli yksityiskohtaisesti määritelty sekä ohjeistettu. Odotan ensimmäisen robotin parannusehdotuksia, jotka saatuani tiedän miten ensimmäistä robottia voidaan jatkokehittää.

Robotin tulokset kiertävät tällä hetkellä toimeksiantajalla jatkokehitysmielessä. Jos aikataulutus olisi ollut erilainen, niin olisin voinut suunnitella ja kehittää myös toisen robotin hoitamaan laskutuksen alusta loppuun. Ensimmäisen robotin osuus on kuitenkin kokonaisuuden kannalta merkityksellisin, ja kahden

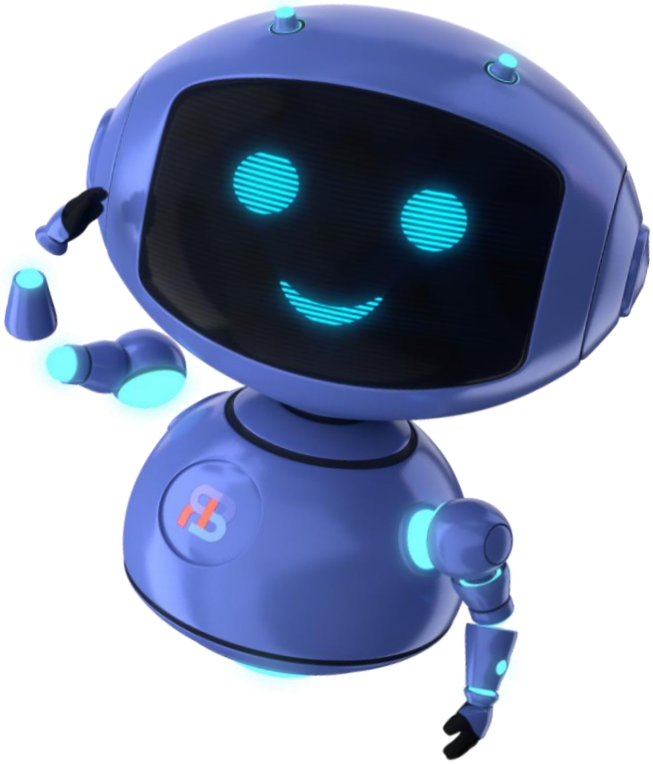
robotin kokonaisuus olisi ollut haastavaa käsitellä ottaen huomioon opinnäytetyön laajuus. Kokonaisuudessaan voidaan todeta, että robotti täyttää kaikki sille asetetut vaatimukset ja odotukset.

Lähteet

Tripathi, A. 2018. Learning Robotic Process Automation: Create Software Robotis and Automate Business Processes with the Leading RPA Tool – Uipath. Birmingham: Packt Publishingn, Limited. Vaatii kirjautumisen palveluun. Viitattu 29.5.2023.

IBM 2023. What is an API? Viitattu 15.11.2023. https://www.ibm.com/topics/api

Rajobit



Prosessi:

LaskutusRobotti 1

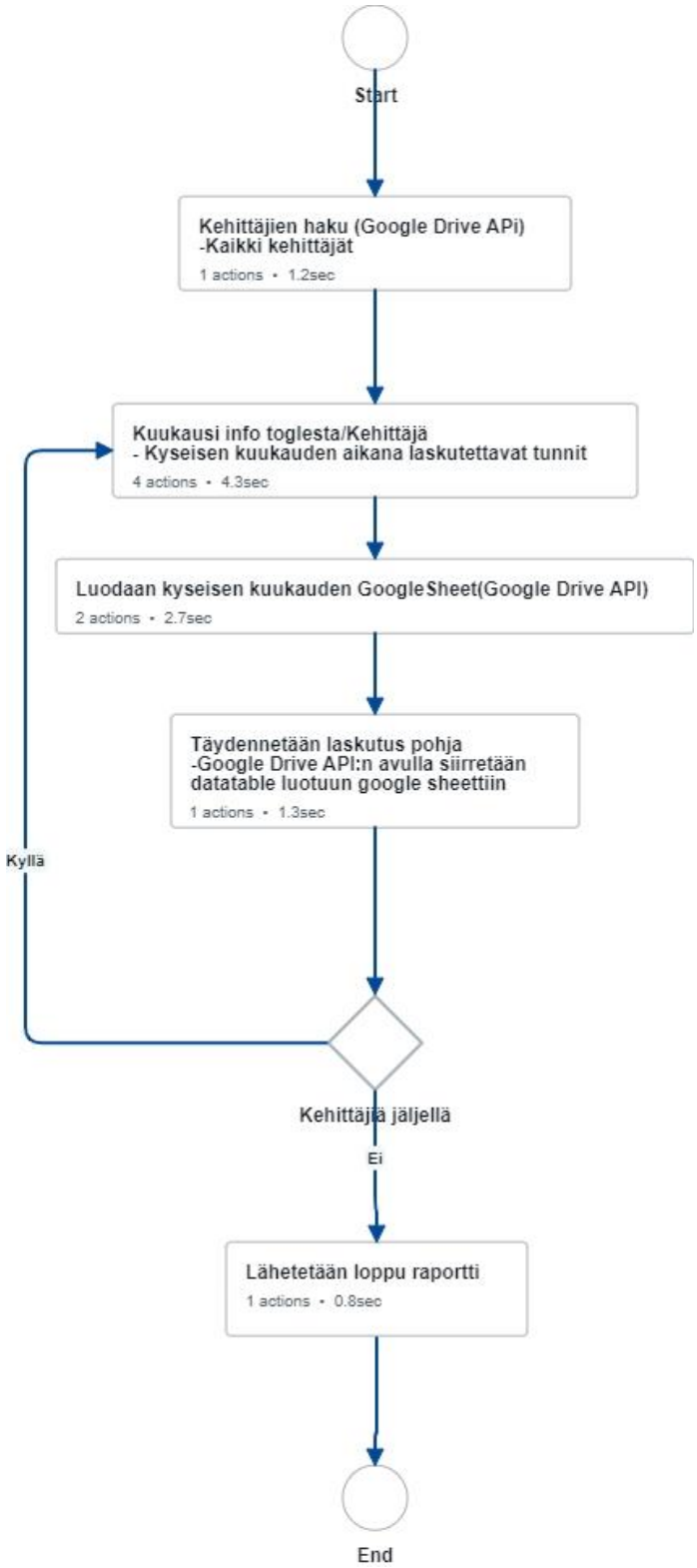
Tekijä:

Ville Huopainen

Kuvaus:

Tässä dokumentissa käydään läpi prosessi LaskutusRobotti 1.

Prosessikaavio



Kehittäjien haku (Google Drive API) - Kaikki kehittäjät

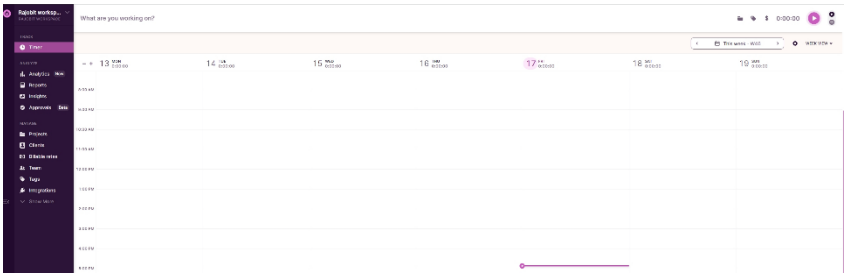
Developers Google sheet, jossa alkudata

Haettavat tiedot:

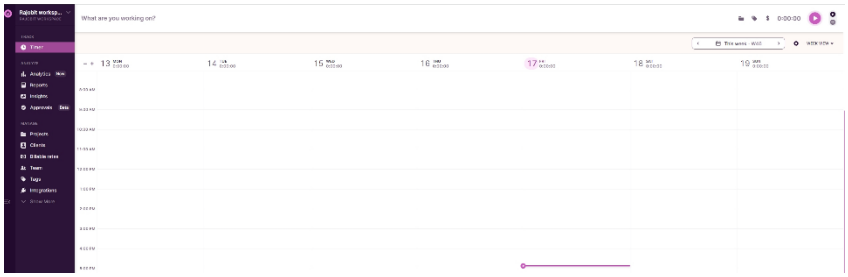
Kehittät, URL:it ja SheetID:t

Kuukausi-info Toggl:sta/Kehittäjä - Kyseisen kuukauden aikana laskutettavat tunnit

Kirjaudutaan Toggl-järjestelmään valmiilla komponentilla

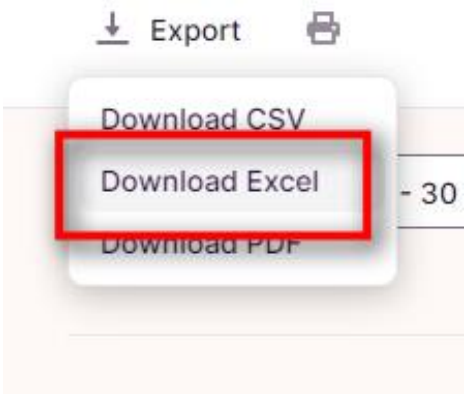


Navigoidaan kehittäjän kuukausi-infoon URL:lla

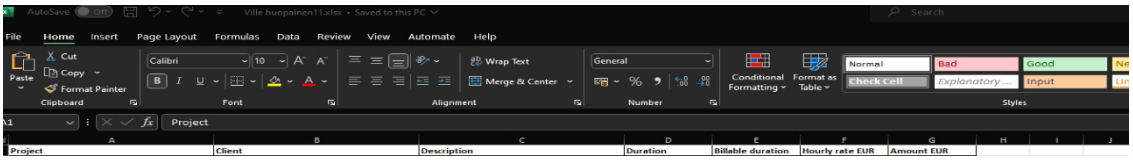


Klikataan “Dowload Excel”

-Tiedosto tallenetaan valmiilla komponentilla

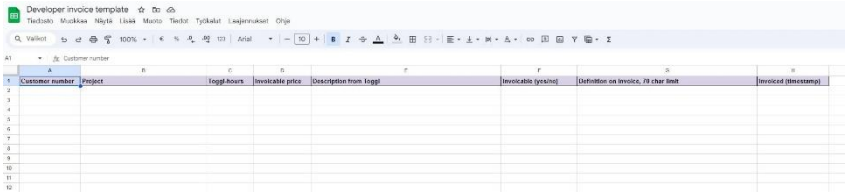


Kuva tallennetusta Excelistä, josta tiedot kerätään talteen



Luodaan kyseisen kuukauden GoogleSheet (Google Drive API)

Kopioidaan kuvan laskutuspohja



Siiretään kopioitu laskutuspohja kyseisen kehittäjän laskutusvälilehdelle

Lisätään kopioitu pohja nimellä ”kyseinen kuukausi ja vuosi”

Täydennetään laskutuspohja - Google Drive API:n avulla siirretään datataulukko luotuun Google Sheettiin

Tuodaan Toggl-järjestelmästä haettu ja siivottu datataulukko kyseisen kuukauden laskutuspohjaan

Kehittäjiä jäljellä

If 'Kyllä' then go to '2. Kuukausi info Toggl:sta/Kehittäjä
- Kyseisen kuukauden aikana laskutettavat tunnit'
If 'Ei' then go to '6. Lähetetään loppuraportti'
Onko työjonossa vielä kehittäjiä jäljellä

Lähetetään loppuraportti

Robotti lähettää kyseisen kaltaisen raportin päättyneestä ajosta

Rajobitin pohja muodostaa raportin

00000_Laskutusrobotti1 ajo päättyi onnistuneesti

robotuser@rajobit.com

Ten kaikki työtä:
Onnistuneesti käsitettyjä: 1.

Tyyskentelin yhteensä 2 minuutia 34 sekuntia.

Onnistuneet käsittelyt

Referenssi	sensiitinArvo
Ville Huopainen: 11	Laskutus Excelä täydennetty onnistuneesti

Rajobit

End