

Diary thesis of Technical Support Engineer

Mikko Sevanto

Bachelor's Degree
Programme in Business Infor-
mation Technology
2021



Author(s) Mikko Sevanto	
Degree programme Business Information Technology	
Report/thesis title Diary thesis of technical support engineer	Number of pages and appendix pages 37
<p>This diary type thesis was written to follow a 10 week work period of a technical support engineer. The purpose of the thesis is to write daily reports of completed tasks at the place of employment and to analyze the daily reports in regards to progress and development.</p> <p>The thesis shows and explains the required tools and methods for completing a wide range of tasks as well as a clear understanding of the electronic invoicing world. The analysis was completed every week from the daily reports written for the week. The weekly analyses reveals different key concepts of system infrastructure and system communication by focusing on topics such as REST API, Amazon Web Services and SFTP.</p> <p>The thesis indicates clear development of skills in the Linux environment, scripting, relational databases as well as understanding the system infrastructure implemented at the place of employment.</p>	
Keywords IT Support, electronic invoicing, system integration, MySQL, REST API, scripting, Linux, AWS	

Table of contents

1	Introduction	1
2	Framework	3
2.1	Analysis of your current work	3
2.2	Interest groups at work.....	6
2.3	Interaction skills at work	6
3	Diary entries.....	8
3.1	Observation week 1	8
3.2	Observation week 2	10
3.3	Observation week 3	13
3.4	Observation week 4	16
3.5	Observation week 5	18
3.6	Observation week 6	20
3.7	Observation week 7	24
3.8	Observation week 8	26
3.9	Observation week 9	29
3.10	Observation week 10	32
4	Discussion and conclusions	34
	References	36

1 Introduction

This diary type thesis consists of daily entries of a technical support engineer at Apix Messaging Oy. The period for this thesis is April 19th – June 25th. The thesis content is highly centred around the daily tasks at the place of employment and includes weekly analyses which cover some main topics that are an integral part of the work that is being done. The goal of the weekly analysis is to understand how the work is rooted into different IT topics and how one can improve their skills through a more thorough understanding of related IT areas. The key IT skills required to successfully complete tasks in this diary type thesis are as follows:

- General programming knowledge in Python and Perl
- An understanding of Linux basics and how to navigate around a command line interface and how to use commands to execute actions.

Most of the processes used at work have been built using Perl and some using Python. My knowledge in Perl is limited but my Python knowledge is pretty good as I have learnt it in Haaga-Helia prior to starting this job. I wish to expand my Python knowledge as well as learn the syntax of Perl so I can understand some processes better. As invoices go through our process's problems can occur and when they do one must understand the code that is being executed during the processing of the invoice. What is causing the problem and what needs to be changed to get the desired outcome? As time goes on during this diary thesis, I am hoping to learn more about these two coding languages as I would like to be able to modify and possibly create new solutions to help me with my work.

I am familiar with Linux from my earlier Linux course at Haaga-Helia which is useful for my job. All the concrete tasks such as handling production data is done on a virtual computer on the command line interface. This is where all the necessary scripts are executed to do my work. Basic Linux commands are certainly useful, and I am hoping to get more comfortable with these as I gain more experience through the job. The most common Linux commands I use are (*B, Arturas, 2021*):

- cd (changing directory),
- mv (moving files)
- mkdir (making directories)
- rm (removing files)
- grep (searching through directories for a file)
- vim (text editor)
- ls (show contents of current directory)

- cp (copying a file, and placing it in another directory)
- sudo (executing tasks with administrative permissions)
- zip and unzip (for compressing or extracting files)
- ssh (remotely logging into another one of our computers to find certain types of data, for example scanning logs)
- exit (disconnecting from remote computer)
- less (show contents of file)

These help me get through most of my tasks. The rest of the commands I use are scripts written by our developers specifically for our processes. The main topics of discussion for the weekly analysis include system integration, REST API, scripting and Amazon Web Services.

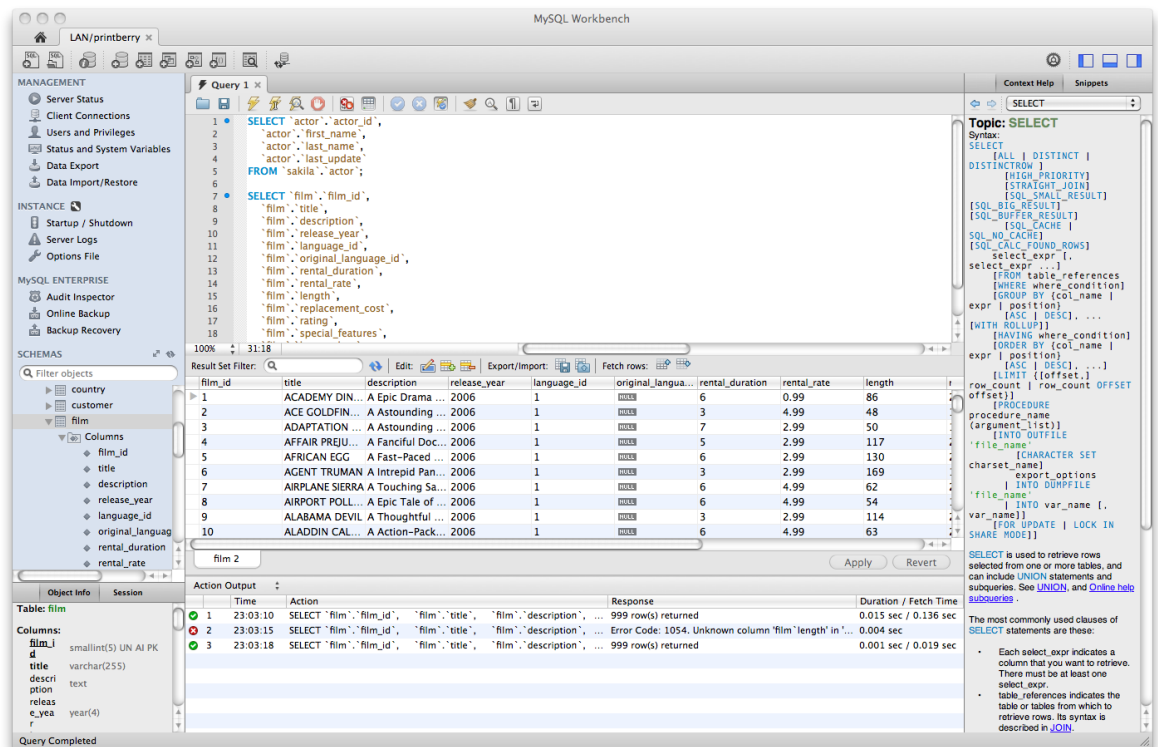
The company I work at employs around 20 people. The business provides a secure service for organizations to send their invoices through our servers where we process, validate and convert electronic invoices. These invoices then continue either directly to the recipient, the recipients billing system or to another service provider like Apix. Apix also provides software that our customers can use to draft, send, and receive invoices. The company also provides customizable and personalized solutions to fit the customer's needs. The job is highly technical and requires an in-depth amount of knowledge about the electronic invoicing world to be able to solve problems efficiently. There is no single universal format for the way data should be represented when sending an invoice and that is why the tasks can get very confusing and challenging. The purpose of our processes is to be able to successfully receive data from the sender's system, validate it and convert it into a format that is readable by the recipient's system. The channels that an invoice travels through are all determined by the rules that have been set for the specific sender and the recipient. These channel specifications are usually determined by the accounting systems of both sending and receiving ends. Channels can have their own unique and customer specific process for conversion and validation as well as more generic processes which are used by a wider range of customers. The format which Apix prefers is Finvoice 3.0 which is the most current version of this message format. Finvoice was "published by Finnish banks" (*Finanssiala, 2021*).

2 Framework

2.1 Analysis of current work

Most of our tasks are handled directly through command line interface. At the start of the day I connect to a virtual computer where we have most of our production data stored. A lot of my tasks involve using scripts developed by our developing team and with these tools I succeed in my work. Scripts allow me to directly view customer information, view invoice data, view routing, change routing, execute tests, import data, edit account details, administrate accounts and so much more. The importance of these scripts will surely be highlighted in the diary entries and weekly analyses as there are so many of them and they all have their own purpose in supporting my daily activities.

One of the main interfaces I use alongside command-line is MySQL workbench. MySQL is a visual tool to view, modify, administer, and handle data.



Typical view of MySQL workbench (MySQL, 2021)

In this relational database I can view transactions, event-history, user information and find unique ID's that can then be used to investigate data on the CLI. I can find a unique ID of an invoice and copy it into my temporary folder on Linux. All invoice data is in an XML file which upon viewing tells me a lot about a certain invoice. XML stands for Extensible Markup Language and is a way of representing human and machine-readable data in a logical manner through a set of standards. The purpose of XML is to hold data which can

then be processed by machines and humans. Data comes in defined fields and elements which can then be picked up and processed. Finvoice requires certain fields to come in certain order or the invoice will be rejected due to incorrect data. This same logic applies to other formats such as Teapps, which is another nationally accepted electronic invoice-format alongside Finvoice (*Edicom, 2019*).

Viewing the file allows me to see the format it arrived in, sender and recipient details as well as invoice rows which contains the actual invoice information. An example of how a Finvoice 3.0 XML file might look like:

Liitesanomani malli

```
<?xml version="1.0" encoding="ISO-8859-15" ?>
<FinvoiceAttachments xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FinvoiceAttachments.xsd" Version="1.0">
  <MessageTransmissionDetails>
    <MessageSenderDetails>
      <FromIdentifier>FI4043091020117999</FromIdentifier>
      <FromIntermediator>HELSEFIHH</FromIntermediator>
    </MessageSenderDetails>
    <MessageReceiverDetails>
      <ToIdentifier>003702048999</ToIdentifier>
      <ToIntermediator>003714377999</ToIntermediator>
    </MessageReceiverDetails>
    <MessageDetails>
      <MessageIdentifier>XS0604.48GPB2/212::attachments</MessageIdentifier>
      <MessageTimeStamp>2012-06-04T15:54:40+03:00</MessageTimeStamp>
      <RefToMessageIdentifier>XS0604.48GPB2/212</RefToMessageIdentifier>
    </MessageDetails>
  </MessageTransmissionDetails>
  <AttachmentDetails>
    <AttachmentIdentifier>YV1199015::attachments::a0824b7488d56e01cb98c9082ec 320489e
e99e26</AttachmentIdentifier>
    <AttachmentContent>iVBORw0KGgoAAAANSUUEGAAAAwAAAAACAYAAAB
WdVznAAAAAXNSR0IArs4c6QAAAPxJRFUKM+FkUGugkaQRF/DSEQWchnjKJyEPXGjIZ5GY7yLF3D1VdTghil3ReCfX0kvOt
Wvq5M2SWIgmwPghOX03XRdR1EUTkdTKiBe0/Xdf1lwvQzY7FYAHC9Xgkh/J3QNAImRl3XbDYb4jimaZr+TfrSfD7XbDbT/
X5X27bK81x5nn+P6AOcz2cBqqrqY+73e0mE51OpzGwXq8F6HK5KISgEIJut5ucc1qtVmPAOSegV2mayszknOsD2+1WgMx
sBEVRJEC73U6SZJKUZR
kAz+eTX8qyDDPj8XgQHq4HXq8X3vvPZ4cfXi6XtG3L8XiEsixl2vpPgIqi0BvI+/Yxt3r4KgAAAABJRU5ErkJggg==</At
tachmentContent>
    <AttachmentName>vokaali.png</AttachmentName>
    <AttachmentSecurityClass>SEI01</AttachmentSecurityClass>
    <AttachmentMimeType>image/png</AttachmentMimeType>

    <AttachmentSecureHash>a0824b7488d56e01cb98c9082ec320489ee99e26</AttachmentSecureHash>
  </AttachmentDetails>
</FinvoiceAttachments>
```

(*Finanssiala, Finvoice 3.0 Soveltamishje, 2021*)

Through MySQL workbench I can easily view the channels the invoice has come through and as it is a relational database, I can open a channel table and view the channel information by querying the respective channel. Here I can truly start investigating the processes a certain invoice has travelled through and how it has been modified by our pre-processor, converter, and post-processor. It is important to note here that electronic invoice operators such as Apix rarely touch the concrete data itself (monetary values, sender address, account numbers etc) but rather the way it is represented and readable by any system the receiver might be using.

My current work also holds a lot of simple and repetitive tasks that do not go that into the depths of our processes. This includes tasks such as changing passwords, e-mails and modifying customers routing. In Finland most e-invoice addresses are formed by the OVT number which always starts with the Finnish tax code 0037 and is followed by the recipient's organization number (*Maventa, 2021*). If the recipient's operator is a bank the address can be in the IBAN (International Bank Account Number) format as well. Our system retrieves this data from a public database (verkkolaskuosoiteisto.fi) every morning and updates this data into our system. If a recipient's address is not public it has to be informed to Apix and this is where I can then manually insert it into our system by running a script that takes the address (OVT or IBAN), organization number, organization name and operator ID (for example NDEAFIHH for Nordea). If receiving address is not found when someone is sending an invoice through our system, they will receive negative with error details. During the time of this diary thesis, I should encounter some problems that require more problem solving as well.

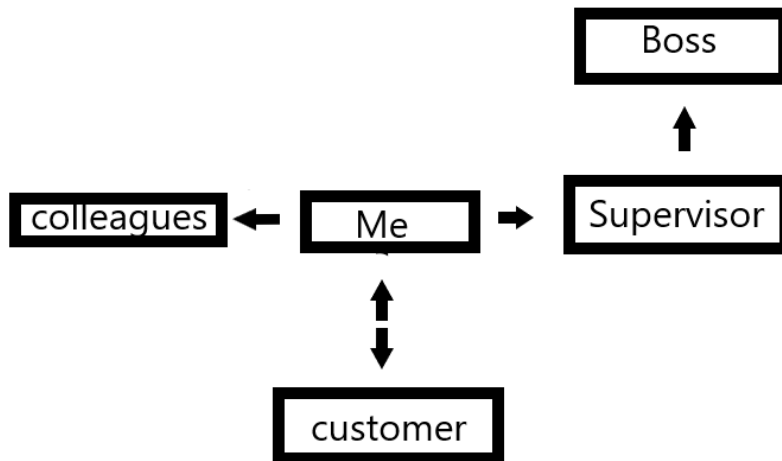
Currently at the start of this thesis I would assess myself as a beginning-stage actor. This is due to electronic invoicing being a concept of its own and experience is required to gain knowledge in the formats and channels of invoices and how data is manipulated during its time of travel from the sender to the recipient. During the beginning of the thesis, I heavily rely on my supervisor to teach me the appropriate ways for handling tasks. My prior knowledge in Linux and understanding of programming languages does certainly help me with my job. I hope to be an experienced specialist by the end of the thesis.

My professional development is at its early stages as this is the first truly technical job where I can put my already learned skills from school into use in the real world. My inexperience will most certainly shine at the beginning of the work but towards the end of the thesis I will have developed in Linux, MySQL, and programming as well as understanding API's and system integration. During this thesis I will focus on my technical development and problem-solving skills. The development during this thesis will not be heavily focused on programming but this is something that I will be able to develop in the future after gaining more ground in the tasks I need to complete during my junior stage as a technical support engineer.

The main literature and sources I have chosen for the thesis is Amazon Web Services for Dummies, Apix's own REST wiki, MySQL reference manual and Finvoice specifications and requirements instructions. All of these sources are central to helping me understand our processes, infrastructure and my daily tasks.

2.2 Interest groups at work

The interest external interest groups at work are customers and the internal interest groups at work are my supervisor, boss, and colleagues. The main interest of the customer is the service they receive when encountering issues with e-invoicing and to be able to trust our solutions. The interest of my colleagues with my work is that I can succeed in my tasks independently and that I do the required job. The interest of the supervisor lies on my capability to learn and succeed through tasks as well as being punctual and attentive when learning new things. The main interest of the boss is that I am independently and reliably completing my tasks which I am being employed to do.



2.3 Interaction skills at work

The interaction at work is mostly enjoyable. A lot of knowledge and experience is shared between developing, implementation and support departments and good communication creates higher efficiency turnover. Our atmosphere at the workplace is relaxed, open and requires individual competency as well as good teamwork skills. Most departments are around 3-5 people and if someone is uncertain of something there is always someone who will have an answer and at times if no answer is found we approach the situation as a team and work towards a solution together. It is useful to have this level of bonding and

sharing of knowledge because this way everyone can have a deeper understanding and broaden their perspective on the processes of other departments. Sometimes there are challenges in the workplace when everyone is busy and focused on their tasks. I find it important to read the room and understand when it is not a good time to disturb someone if the problem at hand is not detrimental to our business activities. Most of the interaction with colleagues is pleasant and the workplace is enjoyable.

The channels I use to interact with customers is by e-mail, live chat, or phone. With more experience comes confidence and I think the phone is the quickest way to solve simple solutions and it helps maintain good customer relations. When the problem requires more time for a solution, I can resort to e-mails with the customer so we can transfer files and test newly customized channels for example. Sometimes challenging and demanding customers can be difficult to deal with but with the right approach this can be handled by providing patient and knowledgeable service to the customer.

3 Diary entries

3.1 Observation week 1

Monday 19th April

The main objective of the first day of this thesis is to learn more about our software and familiarize myself with it. Going into this day I have now been working at the company for a little less than 2 weeks, so I still have a lot to learn.

As expected, the day was quite repetitive in terms of what I know and full of simple tasks such as adding contracts to customers. The term contract here means adding a service such as email scanning for example. Another simple task I did today was help a customer by finding invoices that they had sent and informing them which channels the invoices had gone through and what type of invoice it was. This invoice had been sent as a letter. A more interesting thing I did today was generate myself a private and public SSH key pair to gain access to a different server of ours. SSH stands for Secure Shell and is a protocol designed for authentication and securely accessing different systems over a network. (*Peter Loshin, 2021*)

Tuesday 20th April

Today I will finally focus on our software. I know through our software the customer will be able to handle their account details, draft, send, receive, and archive invoices. My goal is to get familiar with the functions so that in the future I can quickly assist customers with their needs regarding the usage of our software.

The day was surprisingly productive in terms of learning as I had a couple of customer cases where they required help changing their passwords for their accounts. Here I ran a script which automatically replaces the original password with a generated one.

As for my learning outcomes regarding the software I am certainly getting more comfortable with it than I was when I first started here. Learning the software functions and navigation is not as much of a technical goal but it is an important piece of knowledge required in my everyday work, therefore I am satisfied with my learning outcomes for today.

Wednesday 21st April

The day went quite well with a few bumps on the road. It was another quite redundant day in terms of my tasks as I am still trying to remember every detail on how to efficiently get through the tasks and what kind of scripts to use in what scenarios. For example, how we handle a rejected invoice is determined by a customer's default outgoing channel. For

some it is email, for others it might be paper and for some it might be just another e-invoice, or a notification of the rejection. For this I use different scripts with certain parameters that are required to complete the task successfully. The interesting thing that happened today was that our VPN went down for a good half an hour, and this prevented everyone from being able to do any work as we all got disconnected from our servers. One notable thing I learned today was how I can send and remove information from a website that holds public records of organizations and their electric invoice addresses. In a case such as today, one of our customers had changed their operator and wanted us to remove the routing they had with us. For this I ran a script that removed their services with us and asked the customer if they wanted us to keep their records in our archives. After this, I executed a new script and sent the information to the public database, in which the customers routing with us was finally removed. The update was instant and after refreshing the page the information disappeared from the website. All in all, it was a good day and I managed to learn some new things.

Thursday 22nd April

Today's objectives are centred on improving at handling basic tasks. I do not have anything extraordinary planned for today. The good thing about this job is that while there is a lot of repetition, there are also many unique and more challenging problems that arise as there is a multitude of reasons that an invoice is failing to go through. Today was not one of those days, however. I handled basic daily problems.

Friday 23rd April

My main objective today is to get a little more comfortable with MySQL. I have had almost zero experience using MySQL prior to getting employed at this company and it is an integral process of my daily work.

The main objective was certainly reached today. I used many different queries and tables to get the information I needed to complete my tasks. For example, running multiple queries using information such as a senders business ID, recipients business ID and the invoice number. With this information I can find an invoice in our event history and use certain unique IDs of the invoice to copy it into my temporary folder in Linux, then start digging through the data and investigate the material according to what I am trying to find. I think MySQL looks more confusing and challenging for a newcomer than what it really is.

Weekly analysis 1

During the past week I handled a lot of routine tasks and nothing quite out of the ordinary happened. The main area of improvement for me was certainly understanding more about

the relational database that we use. Considering I have never used MySQL or any database before, I had to start from the basics of working my way around a database. Our databases hold a lot of information in countless tables that all support different areas of the business. When a table is opened in MySQL, below the query section there will be fields, and these fields hold data. For a customer we might have data such as the customers address, the type of contract they are under and so forth. I can access a lot of this data on our computers without having to use MySQL but having it on a nice interface with quick and logical search functions due to the structured query language does ease the process a lot. Some data in these fields holds key information which can then be used to search for something else in a different table. A great example of this would be finding an invoice in the event history.

The data for a single invoice might hold sender details, timestamps, status, recipient details etc. When an SFTP customer receives an invoice, a unique storage id is created for each invoice. Even though the status in the event history table says delivered, it might not tell the whole truth. This storage id can be used in the storage table to see the path and directory the file has been dropped into, and whether the customer has retrieved the invoice along many other details. This is just one example showing the importance of a relational database.

Referring to the forementioned SFTP, SFTP stands for Secure File Transfer Protocol which uses encrypted data and an SSH keypair to securely transfer files between servers. Authentication happens when the clients private key finds its pair in the system. (*Conrad Chung, 2014*) Over all this week went well and I am pleased with my learning outcomes thus far.

3.2 Observation week 2

Monday 26th April

This day was alright, I was able to handle some live chat questions, finding invoices in our database and changing customers routing. I did not learn any new skills today, if anything I probably just gained some speed and further knowledge on how to handle certain repetitive tasks more efficiently.

Tuesday 27th April

It is another day where I do not have any clear objectives. Most likely the day will revolve around tasks I am familiar with and potentially learn something new, depending on how busy my supervisor is.

The day started off with the basic removing and adding of routes, rehandling invoices and sending some emails. One live chat example of today was where a customer wanted to know if they had received any invoices. I used my MySQL knowledge from earlier and retrieved this customers information using their business ID. With that I was able to get a list of invoices they had received and passed that information to the customer. Another customer was wondering why they were receiving paper invoices rather than e-invoices. I dug around the xml files of their invoices and found that their e-invoice address had been entered incorrectly therefore the process could not route the invoice and it had to go through their default outgoing channel, which in their case was by mail. The most interesting problem that occurred today was with a customer having duplicate problems with their invoices. They tried to send some invoices however these were rejected by our duplicate checker. What the duplicate checker does is check the document unique ID and check if that invoice has been sent already. If so, any further tries to send the invoice will be rejected. They were having some system problem on their side as the system kept on trying to force one invoice multiple times to our API, but it got rejected just as many times as the invoice having already been processed.

Wednesday 28th April

I was doing my basic tasks and after a couple of hours my supervisor wanted to show me some new problems and how to handle them. This case had to do with our scanning services. These invoices had been rejected due to the scanning service not being able to pinpoint the routing based off an eTag. An eTag is something that our customers can choose to use or not but generally it is just a tag that is mapped to a customer in our system, so when an invoice goes through the scanning system it can find the routing faster by linking the eTag to a customer and their address. The system generally improves scanning times as the scanner does not have to read a name, an address, a business ID and so forth. These eTag's did not exist however and the problem can be solved by running a script that essentially just adds another name for the original eTag. If future invoices use the same eTag the system will this time be able to spot the recipient and send the invoice through correctly.

Thursday 29th April

The main objective of today is trying to understand and hopefully solve a couple of yesterday's problems. A script that we commonly use to save data of invoices into our temporary directories was malfunctioning due to one of our developers' making changes to it. The way the script works is we use it to copy invoice data into our working directories so we can dig at invoice material and investigate the original and post processed files. The reason the script is so important is because it makes data investigation a lot faster by

making an accessible copy of invoice material rather than looking at the general data provided in MySQL. Another objective is trying to understand why certain tickets keep coming back to our ticket system with the same problem, even after the problem had been fixed.

Today we were able to fix the first problem that had occurred with the script. With the guidance of my supervisor, I made a backup of the script, opened the script, and reverted the changes our developer had made so that we could still use the script today for our necessary tasks. After that we moved the script to our production server so that others could use it as well while the developer was working on a more permanent solution. It was certainly a more challenging task, and I learned a little bit about Perl and how communication between our different server's work. Another bigger problem occurred today which was a system integration issue between an external accounting software and our systems. Some users were trying to establish a connection between the two systems but were unable to due to an error which is still being investigated.

Friday 30th April

I am hoping to find a solution for yesterday's system integration issue. Other than that, my goals are to just keep pushing through the daily tasks and expanding my knowledge and efficiency. As it turns out, the system integration issue was not on our side. The developing team tried to replicate the issue on our side of things but were unable to. This problem needs to be fixed on the other end.

Weekly analysis 2

This week turned out to be good for my learning and skill development. I'm slowly starting to have the routine tasks on lock and can get through them with ease. I have gained more understanding about our processes but do have a lot to learn. The main takeaway from this week for me has been scripting. A script is generally a program that upon execution follows a set of instructions to complete a certain task. (*Computer Hope, 2019*) Most of the scripts that I use have been written in Perl and some in Python. I am not particularly good at interpreting Perl as I have never encountered or learned it before. The presentation of code seems messier than the one of Python. I think the reason why Perl is so widely used for our scripts and processes can be found in this quote: "Perl was originally developed for the text processing like extracting the required information from a specified text file and for converting the text file into a different form." (*Geeksforgeeks, 2019*) I find this quote to hold particular importance due to the amount of text data handled at Apix. When invoices go through our processes the XML fields are filled with text, the Perl scripts can pick out this text and manipulate it into corresponding fields of a different format.

The main advantage of using scripts at my job is the automation of tasks. Using customized scripts for the job allows for efficiency and flexibility. For example, if I were to resend an invoice, I do not have to drop the invoice to a certain folder so it would be processed successfully. Instead, I can run a script that takes the unique id of the invoice, and it will be reprocessed through the same channel it came in and it will follow the rules of the process written for that channel. After it successfully goes through the validation and conversion process it is finally dropped to the correct outgoing directory. If it were not for the script, I would individually need to run the invoice through the right preprocessor, splitter, validator, converter, and post processor scripts before manually moving it to the right folder.

These tools increase effectivity in my daily tasks. I hope to be able to write some tools for any reoccurring problems I may face.

3.3 Observation week 3

Monday 3rd May

Today's main objective for me is to gain more independency in problem solving and understanding more about our processes.

One of the main things I did today was solve a problem where an invoice had been sent as an e-invoice instead of the intended email invoice. I started the process by searching for the invoice by its number in our database. After I copied the document ID into my working directory in my Linux environment. I then navigated into the invoice material and started looking through the original XML file to see if I could find the problem through there. I could not find the issue there at first, so I went look at the log files. Through the log file I could see all the stages the invoice had gone through. From the log I could already see that the invoice had been routed through the default channel, so I decided to check the validator script this file had gone through and compare it to the original xml file. I opened the script on CLI and searched for the word "email". And right enough there was a conditional statement in the script which would dictate the reason this invoice had gone as an e-invoice.

The problem was that the billing party had entered the word "Email" into a certain field. But this validator script only accepts the word written with all caps (EMAIL). So due to that miniscule detail, the invoice had been sent as an e-invoice. I am not familiar with Perl, but

I am satisfied for being able to identify the problem due to my general knowledge of programming. There are many different formats for electronic invoices and many different requirements, so I did not mess with the code in the script, I let it be and instructed the customer on how to fill in their invoice details correctly. I would say it was a successful day.

Tuesday 4th May

I have not set any objectives for this day. I will be going to work and doing my basic tasks, maybe something new will come along the way. This day was spent doing a lot of repetitive tasks. Changing customers routings, investigating invoices and so forth. The main thing that I learned today was the difference in how to handle one of the most common errors. The error occurs when the recipient's operator rejects an invoice due to the recipient not using said operator anymore. The most common way of handling this error is searching for the routing with a script, removing the route if found and then resending the invoice according to the recipients chosen to default out channel. However, this only applies for organizations. For B2C customers the method is taking the rejected invoices unique ID and using that to find parameter information in our database. What this information shows me is the required unique ID for B2C customers. This ID is then used as a quick query to locate the B2C customer in our database where we can manually delete their routing on MySQL.

Wednesday 5th May

Nothing out of the ordinary happened today. I started the day regularly by opening live chat, my ticket queue and logged on to one of our virtual computers where I handle most of my daily tasks. Here I spent most of my day on command line interface handling repetitive tasks such as adding or removing contracts, changing customers emails or passwords and adding more credit to their account.

Thursday 6th of May

The objective of today is learning about more complex tickets and how to handle them correctly. I learn something new every day but some of these tasks require a repeat process to truly acquire the skill. The main problem is that similar tasks need different solutions at times, and they can be hard to differentiate. I am hoping to go further than the tasks I already know comfortably and hoping to grow my perspective and understanding.

I spent the first half of the day solving my everyday tasks and assisting customers through live chat. For example, changing their user email or password or finding out what happened to an invoice they had sent. The second half of the day however was a bit more interesting than that. I think I cleared around 100 or more tickets with the assistance of my

supervisor. For the most part it was just running a script and for-looping the script through a text file where I had stored document unique IDs from identical tickets, and it would resend the these invoices automatically.

Friday 7th of May

The day held a lot of repetition due to trying to clear just one type of ticket queue and I managed to get down to about 70 tickets from 200 and more. Nothing new regarding those tickets popped up however there was a customer on live chat that had a more unique case. This customer wanted to make sure all their invoices would be sent as paper instead of e-invoices. This had a simple solution however as this can be done by the supplier adding the word TULOSTUS on to their invoice data and our processes will then direct it to mail.

Weekly Analysis 3

This week was progressive in terms of my learning. Looking back at the first day's handling of digging through the log file I was able to find the channel and the validator the channel uses. With this information I navigated into the validator directory and opened the validator script that had been run on this invoice. This is a learning lesson as I will most definitely search the log files more often when encountering errors. I was also able to identify the issue in the process even though the script had been written in Perl. I think one thing that I could have done differently here was get confirmation from the supervisor and edited the script so that it would indeed accept the word email as well. Why was it only written to accept "EMAIL" instead of lower caps as well? Would altering the code affect the outcome of the invoice in another way? Either way I could have made a backup copy of the original validator script and tested modified version to see if it would go through. This would have been a safe way to investigate and make changes to the script.

This week I also handled routine tasks which certainly develop my skills and confidence. I also automated a task by running a for loop on the command line. Here I would create a text file using Vim and paste document unique IDs and save the file. After this I would run the for loop on command line where it would loop each line from the text file and execute a given command. For the future I would like to write a script that would accept a text file as a parameter and a command as a parameter. This would make using the for loop faster in the long run as I would not have to type out in the terminal every time. The only concern I have for this is how would I properly implement a script that has its on parameters as a parameter to my for-loop script? Something to consider in the future.

3.4 Observation week 4

Monday 10th May

I am looking forward to today as I know me and a couple of other Junior employees are going to get a lesson from our CTO regarding our processes, how our systems work and how traffic flows through our systems.

It ended up being a busy day. Started the day off with learning about our systems and our APIs, I think that will be a good topic for the next weekly analysis as it is going to be a short week anyways due to a public holiday on Thursday. For a good part of the day after the lesson I did repetitive tasks and kept on clearing tickets to ease the never-ending workload. The rest of the day was spent assisting customers via live chat.

Tuesday 11th May

Today I had not set any goals for myself. I think it is going to be another repetitive day but maybe something will come along. Something interesting did come along as I had hoped for. Apart from the daily repetition I had a couple of interesting cases to deal with. One of our bigger international customers was having problems with an invoice not going through. Now this was an invoice that held over 300 PDF pages as attachments, so my initial thought was that the invoice is just too big and one of our processors is timing out. I copied the unique document details, found out the processor it is going through by running a script that shows channel data of an invoice. Once I found the name of the processor, I navigated to our processors directory and copied the script and connected to one of our test computers to test the processor on the file. After running it through it gave me an error where it stated that a "monetaryAmount" field was incorrect due to the numbers in the invoice being separated by dots instead of commas. I looked at the libraries that our script was using and found a function which would change any dots in the numeric fields to commas. After this I inserted the function and ran the invoices through its required processes again and this time it went through correctly.

Wednesday 12th May

-day off

Thursday 13th May

Public holiday

Friday 14th May

My supervisor is not at work today, so I am expecting to be working around my areas of knowledge, but I am interested in taking a challenge independently just to test my own skills.

For a good first half or so of the day I did repetitive and familiar tasks. Very simple stuff such as adding or removing routing, finding invoices etc. The second half of the day was more interesting as a customer through live chat was wondering why their invoices kept getting rejected. I asked for the invoice number and the recipient, with these details I quickly queried on MySQL to find the unique ID of the document. With the ID I was able to copy all the required data to my temporary folder and started my investigation. I started it by first looking at the error message which was complaining about an element being filled in incorrectly. I looked at the format of the invoice, in this case it was Finvoice 2.01. I opened the excel sheet for the requirements of Finvoice 2.01 and found that into this certain element, the letters HUR were required instead of the H that the customer had entered. I then guided the customer on how to fix the invoice and how to resend it again. It was not the most difficult task by any means, but I am still happy with being able to handle this task on my own without my supervisor being there. In conclusion it was a good day and I feel I took a leap towards dealing with more challenging tasks on my own.

Weekly Analysis 4

During this week I improved my areas of knowledge in terms of formatting and understanding more Perl. In electronic invoicing there are many formats with their own sets of rules and when they're first sent, they must pass their respective validator. The validator is dependent on the channel the invoice is coming through and each validator is built and modified for the format and version of the format that is coming through that channel.

On Friday the invoice was coming through a channel where the format was Finvoice with version 2.01. It is impossible for a junior like me to remember every rule and required element that is supposed to be included in the XML file therefore I had to resort to our implementation guidelines for Finvoice 2.01. Finvoice is the national standard in Finland and the most recent format is Finvoice 3.0. On Friday's case I was able to compare the original XML data of the invoice to the implementation guidelines. I would not have done this task differently as I did not have prior knowledge about the rules of the element that needed fixing. Solving this problem helped me understand more about formatting rules and improved my general understanding regarding the importance of sending correct data.

I also improved my understanding of Perl as I got to investigate a processor that was used

for an invoice. I was able to insert a module onto the script and called its function that checks and converts dots into commas as required by the format. I also made a back up of the script before the edits to ensure the previous version of the processor is still accessible. I then ran the newly edited processor into production using a script which updates the scripts on all our computers. I think this week overall was a good learning lesson even though I had two days off in the middle of the week.

3.5 Observation week 5

Monday 17th May

Today my main goal is assisting a customer through phone. I have now worked at the company for a month and a half and have not yet had my first phone call, so I am looking forward to that today.

Well as it turns out my first call was from one of our Norwegian customers. This call was regarding a case from the previous week where the customers invoices were running into a time out error. I had forgotten to send the updated version of the script into production, so the error was still happening for our customer. This time I made sure to check my edited script and moved it to the right directory and ran a command which made the script accessible and executable for all our production computers. It should work correctly now as I tested the script on an invoice already.

Tuesday 18th May

Today is going to be a busy day. One of our main servers is down and the backup server can only handle so much traffic on its own. This is resulting in a lot of time out errors when customers are trying to send their daily load of invoices for us to handle. The day was hectic, I did not have any objectives set for the day. Handled many calls regarding invoices not going through.

Wednesday 19th May

The problem with server downtime continues and traffic is clogged. Our development side managed to free the biggest load of files that was clogging the traffic, so most invoices are going through now. The problem here is that some of the invoices that did not go through still generated a duplicate check for themselves. When a customer tried to resend this, they would get a duplicate error. My main task of the day was clearing the duplicate queue by responding to customer requests and finding their invoice based on their organization and invoice number. After this I would run a script to resend the customers invoices.

Monday 17th May

- Sick day

Monday 17th May

- Sick day

Weekly analysis 5

This week was hectic due to the technical difficulties at work. With the server down-time being a big issue, I was thrown into the customer support side of the job, and I certainly felt inexperienced as at this point, I have only been working at the company for a little under two months.

Customers were continuously trying to send invoices and most of these were running into a time-out error due to the traffic on our end. The good thing about this was that everyone's problem was the same and the fix was rather easy as soon as we got our server up. I very quickly gained experience in handling multiple cases at once and it did get easier, and my confidence grew as I knew the solution and answer to the customers questions. Relating to a topic here I think an actual solution for the server down-time problems could be amazon web services.

AWS is an on-demand cloud computing platform that enables the user to increase computing power when required. From a business perspective the scalability that AWS offers means it is cost effective as cost is calculated through amount used. In a situation such as our servers being down the ability to increase power in an instant would significantly lower any potential costs and issues that comes along with hardware failure. AWS also has an option for REST API for transferring data between servers and they are also using an access and secret key for authentication. This is something Apix is already very familiar with as a similar protocol is in place with our own API. As in most cases security of data is an important factor when handling invoices.

AWS offers many different storage services and there is one which specifically stands out as a useful service. EBS (Elastic Block Storage) is a service which can be attached to any instance to increase volumes for handling data. (*Bernard Golden, 2013, 67-73*) This means storage volume could be increased by mounting it to one of our virtual machines that runs on Linux, then allowing to read and write data to the EBS. This also provides measures to prevent data loss as even if a database went down on Apix's end, EBS would not as it is running on Amazon's cloud platforms. This same EBS can be used to

mount to another instance to transfer the data into our own physical servers again. The storage is separate from our own servers therefore making it a reliable choice in case of any hardware failure.

This week has been an important learning lesson in the importance of properly functioning hardware and how it is necessary to have back up storage options instead of relying purely on the ones that are already being used. To let some traffic through the problem was solved by pausing thousands of invoices. I believe a better solution for business is to be able to increase the volumes we can handle rather than decrease the volumes we are currently receiving.

3.6 Observation week 6

Monday 24th May

Today I did some independent problem solving to a rather non complicated but tedious task. A customer had sent a list of invoices that I needed to check but the formatting was so off that it was impossible for me to easily copy paste 300 plus different invoice numbers. I took it upon myself to write a little python script which would read an integer from user input and increment it by one until the given end parameter. A very simple script that allowed me to print 300 plus invoice numbers with using a while loop. Once they were printed in a simple manner, I could copy paste it into my query. The reason I used python over Perl is simply that I know Python syntax better. The way I could have improved this script is to have it check and compare the contents of two different files. The reason being is that this list given to me contained 300 or so invoices, however I printed about 400 plus because my code incremented the invoice number by 1. Whereas this list was only requiring me to check invoice number 1001, 1005, 1006, 1009 for example. Therefore, incrementing it by one printed unnecessary invoice numbers for the task at hand.

Tuesday 25th May

Regular day today. Still clearing the queues from the previous week's problems. An interesting phone call case today regarding one of our major customers with a case regarding a system that was built for them. The way it works is that a company which wants to send a bill to our customer needs to get user details from our customer to access the system. This biller was concerned about security issues with this. However, there is no real security issue as our customer can create the login details on their end and the only

thing the biller can do on the system is draft and send bills that go directly to our customer. Our customer already confirms the supplier's identity before sending them their newly created unique login details. However, if something is off in the bill the customer does not have to pay the bill. Therefore, there is no real security issue with the way the system works. Another highlight for the day is that I have used my previously made tool two times today. It is proving to be a useful tool.

Wednesday 26th May

Today my goal is to work at things I know and hopefully get an exciting task.

I had an interesting task regarding a converter which was adding an empty XML element to the file. I was not able to understand why the empty element was being added and could not find any reasonable solution to stop it from happening, so I found an alternative way of handling the problem. I created a function which looked for an empty element in the document and if it found an empty element it would instantly remove it as it holds no data. I did not create this function in the converter but rather in the post processor. The reason I chose to do it in the post processor rather than the converter is because the converter is generally more difficult for me to understand as it picks up data from a certain format element and turns it into an acceptable set of data for another format. I figured the post processor was the safer option as it is already going through and checking the data to make sure it is acceptable after the conversion. The post processor is a good place to fix any problems the converter may create.

Thursday 27th May

On this day no major goals have been set.

I was clearing one of our ticket queues where the task is quite repetitive. These are scanned invoices that cannot be directed to the customer because of a missing eTag so the best method here was to map this eTag to the customer ID so future invoices would go through. I also cleared other basic tasks such as adding and removing customer details, changing user credentials.

Friday 28th May

-day off

Weekly Analysis 6

REST API is a very important part of our processes as we handle and transfer lots of data between client and server. A great example of the usage of REST API in our services is

system integration. A lot of customers use their own ERP or accounting systems for creating, receiving and sending invoices. The purpose here is to be the middleman as we provide the proper channels for receiving their invoice data in which we process, convert, and forward the invoice in the required format through the required channel which is defined by the receiver. The ERP software administrator can for example create an end user on our side using a PUT method. This query includes the to be end users' company and user details as parameters, for example business id and name. The response to the query will return the status of the query and if successful it will return required transfer keys which will then be used in another PUT query when sending an invoice for example. With the transfer details our processes can target the customer that is sending the invoice and if everything is in order with the request (all the sender details, receiver details and invoice format are valid) our server responds with the status, cost, number of invoices and other relevant data. The invoice then goes through the required preprocessing, conversion, and post processing before going out through the defined channel of the recipient. If the query is faulty the server's response will be negative and return error details to the sender of the query.

The REST API has other uses where other HTTP methods are used as well. In our case this could be a GET query that we receive and in this request the query could be asking for company details or invoice address details. If authentication is passed our server will respond with the requested details. The GET method is also used for receiving invoices. Here the parameters include more details about the user's software as there are different processes for handling invoices going to certain software's. (*Apix, 2018*) This is because many ERP software's have their own formats and preferences for ways of representing data. If authentication is passed the response will include the invoice. Here it is important to note that our REST API uses SHA-256 as a way for authentication between client and server where transfer key and id are the shared secret. With this secure hash algorithm, the parameters of the request are digested and calculated into a hash with the SHA-256 algorithm.

An example of a request for a recipients e-invoice address details:

Request:

Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Request version="1.0">
  <Content>
    <Group>
      <Value type="ReceiverName">Aalto Capital Oy</Value>
      <Value type="ReceiverYtunnus">2262882-7</Value>
    </Group>
  </Content>
</Request>
```

Response:

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Response>
  <Status>OK</Status>
  <StatusCode>2A00</StatusCode>
  <Content>
    <Group>
      <Value type="ReceiverName">Aalto Capital Oy</Value>
      <Value type="ReceiverYtunnus">2262882-7</Value>
      <Value type="ReceiverInvoiceAddress">003722628827</Value>
      <Value type="ReceiverOperator">Itella Information Oy</Value>
    </Group>
    <Group>
      <Value type="ReceiverName">Aalto Capital Oy</Value>
      <Value type="ReceiverYtunnus">2262882-7</Value>
      <Value type="ReceiverInvoiceAddress">FI0913783000220315</Value>
      <Value type="ReceiverOperator">Nordea</Value>
    </Group>
  </Content>
</Response>
```

(Rest Api for External Usage - Information for Application Developers - Apix -, 2021)

Regarding Wednesdays problem I do feel like I learned a lot about our processes, but I also think I could have handled the problem better. It would make more sense to find the problem in the converter and get rid of it there rather than create a solution in the post processor. Next time if a similar problem occurs and I cannot find the cause of the problem I would ask for assistance from my colleagues rather than create the solution in the post-processor. Overall, a productive week in terms of learning.

3.7 Observation week 7

Monday 1st June

Today I learned some new concepts regarding our B2C functions. Some of our users have consumers as their customers and these users require their company to be visible in consumer net banks. This is so that the consumer can subscribe to invoicing in their banks where they can instantly receive and confirm a recurring invoice and pay it off. Where we come in is again being the middleman for the safe transferring of this invoice. Once certain contracts are signed between the bank and the company, we send data to the banks to establish a connection between our services and the bank so that the consumers can receive the invoices directly to their bank when the supplier bills them.

My task here is to be in contact with the bank, add the supplier to our database using a script and then running scripts to create the query that is sent to the bank for opening the connection.

Tuesday 2nd June

Today I had a very interesting task that allowed me to understand more processes. One of our customers who uses SFTP for receiving invoices needed a new sftp user and path for a different type of invoice. To complete this task, I first created the sftp path with authorization matching the keys, user, password, host and port given by the customer. After this I created a new specific invoice address for receiving this different type of invoice and went to edit the post processor for this customer. In this post processor script, I created an else if conditional statement that searches through the incoming invoice data for the specific address I created earlier. If the invoice contained this address, the post processor would direct this invoice into the outgoing sftp path I created earlier. After this process I edited the file checker of this customer. What the file checker does is it automatically searches through these outgoing directories and if they find a certain path it will take the invoice and push it out to the customers side of the file transfer protocol according to the rules. The edit I made for this file checker involved in creating a new rule for transferring files from this new outgoing directory into the new incoming directory the customer had created on their side of the connection.

Wednesday 3rd June

An interesting thing that happened today was when I was creating a new message to send to the bank's web services, I had some wrong information in the data. I sent one message to one bank and realized my message contained invalid data. The problem here is that when I run the command in sudo to create this message based off data in a B2C

table in the database, the messages are placed in an outgoing directory from which a file checker checks for files and sends them automatically. After I realized my mistake, I moved the rest of the message files into my temporary folder. I chose to use the move command instead of removing as it is safer, considering I have already removed this directory once by accident.

Thursday 4th June

Today I have not set goals as summer is beginning and the workload is slowing down. I spent the day doing mostly repetitive tasks. I created credentials for some suppliers that use one of our bigger customer's portal. This portal is purely meant for one of our customers and their suppliers only. It is a working interface where the suppliers can create and send invoices directly to our customer. There is a certain level of authentication needed for the supplier to gain access here, but the system is mostly safe as it is built so that the recipient can review and decline any invoices if they do not recognize the supplier. However, we do only create the suppliers accounts upon the recipient's request.

Friday 5th June

Today I have one goal and that is getting authenticated for a different production machine. I was cleared by the government and now can learn a different systems network. Essentially things work the same and a lot of the same scripts can be run here. This system however is intended for only one of our customers. Many of my tasks revolve around support cases for this customer so getting authorized is important as I do not consistently have to ask someone else to retrieve the information. The system runs on a completely different VPN and has extra measures to gain access. This includes private certs, keys, authenticators and forced password changes. The level of security is important as the data handled on this system is extremely confidential.

Weekly Analysis 7

This week went well as I handled many different tasks. The B2C tasks are probably the most time-consuming repetitive tasks that I have had so far. First, I must import the supplier information into our database, after that I need to send the signed contracts to the bank and wait for them to open web services between Apix and the banks for the certain payee. This is opened using the organizations electronic address which in this case is their organizational numbers. If the channel is not opened transactions will not be allowed to go through. Once the bank returns to me, I need to wait a day before I can start sending actual data from our servers to the banks. If the payee had a previous account manager, they must send a message of change for their invoice notice into the bank with the details

I provide them. The details are the payees new electronic send address and intermediary code. This is so that the payees contract bank is changed to the correct one and their addresses are correct. This is done to ensure the end customer does not have to worry about receiving their bills. Once the change message has been sent the channels for the payees to send invoices through Apix and Nordea (Apix's contract bank) will be available within two days.

If the previous account manager refuses to send the change message, I need to send new payee notices into all of the Finnish banks. This is done with a script, and it picks up the data from the earlier database import. There are many things that can go wrong here which is the reason why it is so time consuming. A common problem might be the invoice reason code is incorrect. This is something that cannot be changed and must match the one that is already determined in the banks. The code is used as information for the reason of the invoice which the payee can then see in their bank. This could for example be a water bill. The B2C cases include a lot of communication between bank, Apix, payee and account manager. Every time I send data to the banks I need to check after two days if my data has been rejected, if it has, I need to inquire from the bank or make the correction if I know the problem. A good thing here is that the error codes from the banks are usually clear. There is not really any other way to go about these tasks except step by step with clear communication. I still have some uncertainties I hope to learn from my supervisor regarding this area as it can get very confusing. I do still feel like I learned a lot this week and I can handle most of the B2C cases on my own.

Regarding one of our more confidential portals where the suppliers can draft and send invoices, I am getting familiar with how the system works and can assist customers here when needed. We follow certain protocol for creating supplier accounts and handing the details out. Security regarding this portal is certainly important however the access given to the users is very limited to creating and sending invoices and the recipient must accept the invoices before they pay them. The system also does not allow any incorrectly formatted invoices to be drafted or sent as it follows a certain set of rules for the Finvoice 3.0 format. Overall week 7 was a good learning lesson and I am learning more as I go.

3.8 Observation week 8

Monday 8th June

Today I had to use a script in which I import B2C customers into our database. The script accepts a CSV file and organization number as parameters. The lines and information in

the CSV must be in a certain format such as: name, address, bank account number, in order for everything to run smoothly. There was a problem importing the first line of every CSV file as the CSV file contained hidden characters that I was unable to see on the file and even when opening to edit the CSV using Vim. The hidden characters were part of the UTF-8 standard and are called BOM (byte order mark). These are a sequence of bytes used as an indicator for the Unicode standard for encoding a file. (*Techopedia*) Luckily Vim had a function for this so I opened the CSV files individually and used a function called: `set nobomb`. (*Wincent, 2017*) This removed the BOM from the files and upon rerunning the script the first line of the CSV was imported to the database correctly.

Tuesday 9th June

More import scripts in use today. Yesterday I ran a script to import recipients for already existing B2C payee. Today I am running a script to import 50 new B2C payee for which I will import their recipients a month from now. This script takes the CSV as a parameter and the organization number. I had no problems running this script. I had to touch the code a little bit and hard code certain details such as the customer ID the payee belongs under. The way it works is that the main account holder sends the invoices for the payee so to get the channels working correctly it was a necessity for the script to know which customer the payee belongs to. The day went well as after running the scripts I handled regular tickets.

Wednesday 10th June

I have not set any goals for today. I am getting comfortable with my tasks and handle the routine tasks well. Hoping to stumble upon more challenging tasks.

I realized today that the script I ran on Monday was unsuccessful. I had edited the script slightly with my supervisor and we forgot to include one crucial aspect in the import script. The recipients that were imported into the recipient table were missing the payee business registry number that they belong to. This is important as when the main account holder sends the invoices for the payee, the recipient's details must match the details in our system, or the banks will reject the invoices. The fix here was to edit the script so that the organization number that is read from the parameter is turned into a registry number. The difference between organization number and registry number is that the organization number of Finnish businesses start with 0037 and end with the registry number. The registry number is 7 digits long and looks like the following example: 2224720-7. I found a module under our `sdttools` directory and inserted into the script. It removes the 0037 from the parameter that is read and adds the dash between the last two digits. This variable was then

read by the import sub function and the payee registry number was correctly inserted into the recipient table. I think the day was good overall as I learned more about our libraries that we can use to our advantage when modifying scripts. Having modules that can be inserted into scripts boosts efficiency.

Thursday 11th June

Today I had a customer call regarding a big number of invoices they apparently had not received. I asked the customer for their organization number, invoice numbers and send times to find the invoices. I found the invoices and copied their storage ID's, then I entered the storage IDs into sublime so that I could search the end of the line in sublime and insert quotation mark comma quotation mark (“,”) between each ID. I did this in SublimeText 3 as I am familiar with their search and replace function. (*Sublimetext documentation, 2017*) The reason I did this is because there were so many IDs there is no need to search them individually as it would take too long. The MySQL “where in” function takes in many parameters which are separated by quotation marks and commas. I found all 20 plus of them, and the status said they had already been fetched by the recipient's system from their outgoing directory. The customer wanted the invoices to be receivable again, so I pasted these storage IDs into a text file on my command line. I then ran a for loop through that file in which each line was its own variable, and for that variable the storage status would be changed using a script which takes the storage id as a parameter, in this case the variable in my for-loop. Overall, a good day and I am getting used to running this for-loop to automate some of my tasks.

Friday 12th June

Today my main task was figuring out a problem with an invoice being rejected due to incorrect characters in the invoice. The problem turned out to be some whitespaces in the invoice address details. The white spaces were “non-breaking-space” I was able to figure this out by opening the invoice data using Vim and hovering over the white spaces. As I hovered my cursor over and pressed “ga” on the keyboard vim would show me the hex code of the character. It showed 00a0 as the hex code, so I googled it and found out it is a non-spacing break. I used the insert option in Vim and manually deleted that character and added an actual space instead. After this I saved the file and moved it to the right outgoing directory where a file checker would retrieve this file and run it through the right process. I certainly learned new things today as I did not even know a white space could be such a problem. I think the only thing I could have done better here is to have investigated a way in which I can automatically find all the characters with hex code 00a0 and change

it into a space so I would not have to manually go through each one to change the character.

Weekly Analysis 8

This week was an important learning lesson in understanding what a script does and how it does it. On Monday I executed an import script which I thought would go successfully as the script printed all the details it had inserted into the database. I also checked the database after the import to ensure everything was inserted correctly. Carelessness was the true dealbreaker here as every column in the table was filled except one, yet I missed it. This was the most important column as it would be what creates the link between the payee and the recipient. As a result of this over thousand invoices were sent as paper instead of electronic invoices. Paper invoices costs more than electronic invoices and the customer was obviously not billed for our mistake of course.

Learning wise it was a good experience as I got to investigate and edit the script and I got to make the proper changes required to run the script successfully this time. A good thing to note here is that in the recipient table there is a comment section for each recipient. Before I ran the non-functional script, I had edited the insert comment function in the script so that the comment to be inserted would be "Manual Import" and the date of the import. This allowed me to quickly use a query in MySQL to search for every import that had this comment therefore allowing me to quickly remove the incorrect imports. I then finally edited the script and ran it again, this time the import being successful. I learned to be more careful during future imports, but I am glad I had made that little comment edit as it saved me quite a bit of time when distinguishing between wrong and correct imports on that day.

I also learned about the existence of a non-spacing break which looks like a whitespace but is not an acceptable form of data when sending invoices. I had encountered a similar error before where there were hidden characters however this was easier to solve as it did not require any editing of a script but rather a simple removing of space and adding an actual space.

3.9 Observation week 9

Monday 15th June

Today I created an FTP user. The way I did this was I went into MySQL Workbench and opened the ftp user's table. I copied a row and pasted it, set the ID of the user to NULL as the database will automatically calculate this. I edited the local path according to the

user's information, added the remote host and port. I then generated a password and created their directory for their outgoing files. After this I repeated the process in the file checker table and added a configuration in which the file checkers would now also be searching for any invoices through this new user's directory. I forwarded the user's name, password, and host details to the customer.

Tuesday 16th June

Today I created a couple of new customer accounts for one of our bigger customers. This wasn't the normal process of creation which could be done on our webpage or directly from the database. The way this works is I ran a script that took the organization number as a parameter. The script then searched for the business information from a public database and added the organization number and business name into a configuration file. This configuration file consists of business ID and name. After this I ran an export script which goes through this configuration file, and if some business ID does not match an already existing account in our database the export script would automatically create a dummy e-mail, create a password, transfer key and ID, routing and set the contracts that have been pre-determined by the admin user (main customer in this case). After doing all this and inserting the details into the database, the script automatically updated this information and made it accessible in all our production computers. I am not entirely sure why this is the way to add customers under this certain administrator account, but it was a learning experience as I got to read some more Perl as I always like to investigate a new script before execution, so I have a better understanding of the process.

Wednesday 17th June

Customer wanted to retrieve all their received invoices again from 5 days ago until today. I copied all the storage ids from the database and pasted them onto a text file in my Linux environment. After this I ran a for loop that goes through each line and runs a script in which the parameter is the storage id in each line of the text file. After execution all these invoices were retrievable again for the customer in their designated directory. This is quite basic scenario which I am used to, but it is always good to have some repetition to keep the skills fresh and the service efficient.

Thursday 18th June

Turns out that the implementation team had a bit of a problem when running a new update live yesterday. This update managed to butcher the pdf images that our system creates

when we process an invoice. The pdf images were being built with all the correct information and amount. The problem is that the invoice rows were completely missing from the images so there is no run down of what the recipient is being billed for. My task was to find all invoices that were sent during the time of the update going live and the time of the fix and resend the invoices through the right process with the image being generated properly.

Friday 19th June

Today I created quite a few accounts to one of our portals. This is a repetitive task and there is not many ways to go about it. The way authentication works here is the recipient who the portal is built for notifies us about a new supplier who needs credentials for the portal. We then create and forward the credentials for the portal where the supplier can draft and send invoices for free to the recipient. The service is paid for by the recipient. It is a rather unusual thing for a recipient to have their own portal, but it is a good tool for someone who receives a lot of invoices from overseas. A lot of international suppliers either use very different or outdated systems and do not have an understanding in electronic invoices.

Weekly Analysis 9

This week was generally a good week for developing my skills. This is because I mostly had a challenging task per day which I was able to complete without any major setbacks. On Monday I got to create some FTP users again which is not something I do every day and I had already forgotten the process for the task, so I had to follow a set of instructions left by my supervisor. Creating the FTP user helped me understand the importance of the directories being correctly named and linking them to the file checker configuration so the file checker knows where to look for this customers invoices. The file checkers are an automatic process that checks for invoices continuously and pushes them through the right processes determined by customer information and their respective channels.

Tuesday I was able to familiarize with Perl again as I got to investigate more Perl. The export script automatically forms an account and address based on the newly added customers to the configuration file. Wednesday, I developed more efficiency in repetitive tasks and more of my skills in Linux as I copied the storage IDs of invoices in MySQL, pasted them onto a new text file using Vim and ran a for loop where each line of the text file was read into a variable, and a script would then be executed on that variable until the very last line of text. Running the for loop on command line is a quick way to automate a bulk of tasks that require a process to be executed on them, this allows me to do other

tasks while I have one terminal automatically running a process for me. Thursday I was in busy contact with customers and for-looped document unique IDs to run them through the process again so a proper image would be generated. The important part here was to first make sure with the supplier that they wanted the invoice sent again. Friday, I spent some time creating accounts for one of the portals and unfortunately there is no possibility to for loop this task. Friday my skills did not develop that much, nor did I learn anything new.

3.10 Observation week 10

Monday 21st June

My goal for today and tomorrow is taking care of ongoing cases that I would rather not leave hanging over the holidays.

I managed to nudge one of my cases forward. A Norwegian customer wants to implement Finvoice 3.0 into their system as they only take Finvoice 2.01 currently. I took a couple of their supplier's invoices from earlier as an example and ran these through the right validator, preprocessor, converter and post processor (all determined by the incoming channel and outgoing channel). In this case I converted TEAPPS which is another format for invoices and instead of the converter of TEAPPS into FV2.01 I turned it into 3.0. I safely ran all the processes in my working directory and no files were moved to an outgoing directory meaning no invoices were sent. I sent these example XML files of the format they wanted.

Tuesday 22nd June

Last day of work for this week and for the thesis. I did not set goals today. The amount of customer requests is very little during summer it seems so mostly I have handled routine system generated tasks. I resent invoices as letter because of no recipient found, mapped some more eTag's and closed some accounts which I had set as a reminder for today.

Wednesday 23rd June

- day off

Thursday 24th June

- day off

Friday 25th June

-public holiday

Weekly Analysis 10

This week was more on the slow end. Most people are already on holidays for mid-summers and myself included as I went for holidays already on Wednesday. The most important take away from this week on a personal note was helping a customer with implementation of a new Format. Finland is generally quite technologically advanced in the

electronic invoicing industry considering we accept all the main formats such as Peppol BisBilling 3.0 which is used in 31 European countries and other main regions such as North America and Oceania.

The integrity of data must remain even if it is represented through a different format. All the key aspects of an electronic invoice cannot be changed but it is all a question of how the data is understood and then represented by the receiving system. Looking at my other tasks from the week I can see huge progress from the first weekly analysis to this final week analysis. Handling daily tasks has become easy and I can get through them efficiently. I have gotten very comfortable with all the daily scripts that I use, and I am looking forward to expanding my understanding by continuously taking on more challenging tasks.

4 Discussion and conclusions

At the beginning of the thesis, I considered myself a beginning stage actor. I feel that this statement was accurate considering my then skill level. After completing these 10 weeks I have most certainly developed in many technical areas. I think one of the main areas of development would be understanding the wide world of electronic invoicing. When I first entered into employment for Apix I had zero clue on what I was truly going to do and I had zero idea regarding the processes that electronic invoices go through. Having worked there for the previous 10 weeks this has completely changed. I now understand our position in the field and the purpose of our services. Altering data to fit a specific set of rules is not an easy task and can bring forth many problems which I as a technical support engineer have to solve.

Why did a certain invoice get rejected, why did a certain invoice get sent as paper instead of electronically? These two are some of the main questions I encounter at work every single day and sometimes the answer is straight forward and sometimes it is not. During my 10 weeks I have learned to handle more challenging versions of these problems independently. A basic version might be a routing is missing and the solution is easy, just add the routing and run the process again so the invoice will find its recipient. A more challenging version of this could be a problem in our process. What part of the process is throwing the error, why and how can it be fixed? First step here would be to copy the batch ID into my working directory. Then I would open the contents and specifically open the log file to see where the error is happening. Is it the validator, the splitter, the converter, or the post-processor? After figuring that I would need to open the actual process, for example the post-processor that is used for this incoming channel. Then I would open the actual XML file of the invoice, find the element where the supposed error occurs and compare this to the process which I would have open in another window. Here I can read Perl and see which function is the issue. Sometimes the error here could be an easy fix, such as creating a new function or a conditional statement or removing a function. After the process has been fixed the invoice could be executed again. During the beginning of my employment, I would not have been able to do any of this.

At other times the problem does not occur because of our processes but because of the customers or the supplier's processes. How can I distinguish between where the problem occurs and what is the right way of communicating this to the customer, what can we do to help the customer understand and fix the problems on their side? As a technical support engineer there really are unlimited problems that I face and unlimited answers to certain questions as well. This means that there is a lot of room for growth in the electronic

invoicing field specifically because the tasks are so specific. I would like to highlight here that even though the tasks are very specific to electronic invoicing, my technical skills have generally improved a lot. Before starting this thesis, I had only ever heard of Perl and now I can read it well. I have still a long way to go in knowing how to write it comfortably, but I understand it's syntax and can create simple solutions using this language.

This has a huge advantage for me in general as I my intention here is to grow my technical knowledge so I can apply my skills into areas which may not be in the electronic invoicing field. I have also gotten very comfortable with navigating and working around the Linux environment which is again something that can be applied to many jobs, not just this one. I am now comfortable with MySQL and relational databases and understand how data can be accessed by scripts and written into a database using scripts.

For the future I would like to still improve in Perl and get more comfortable writing Perl and working around its libraries to get the full advantage out of what they offer. Using libraries and their modules makes coding more efficient as tools already exist to offer solutions to problems which may be encountered. For example, in week 5 I used a module that already handled a conversion I was looking for, all I had to do was call the function in the right place. Analysing my work has proven to be helpful as it aided me in understanding how certain things could be done differently to create the same or a better outcome. It has also helped me track my level of development and shown me areas that I need more time to improve. In conclusion writing this thesis has given me insight into my own skills as well as into repeatedly thinking about my tasks. This in turn has given me an opportunity to use retrospective thinking as a tool for learning quicker.

References

Sahlberg, Veli-Matti, and Arto Jussilainen. 2021, Rest Api for External Usage - Information for Application Developers – Apix, URL: wiki.apix.fi/display/IAD/Rest+API+for+external+usage

Accessed: 20 June 2021

Oracle, 2021, Mysql 8.0 Reference Manual. URL: dev.mysql.com/doc/refman/8.0/en/

Accessed: 25 May 2021

Golden, Bernard, 2021, Amazon Web Services for Dummies. John Wiley & Sons Inc, 2015.

Mueller, John Paul., 2017, Amazon Web Services for Developers for Dummies. John Wiley & Sons, Incorporated, 2017.

B, Arturas, 2021, 35 Linux Basic Commands Every User Should KNOW URL:

www.hostinger.com/tutorials/linux-commands.

Accessed 16 May 2021

Finanssiala, 2021, Finvoice e-Invoicing Standard. URL: www.finanssiala.fi/en/topics/finvoice-standard/

Accessed: 1 June 2021

Edicomgroup, 2019, Finland to Compel Public Institutions to Receive Electronic Invoices.

URL: www.edicomgroup.com/en_US/news/12909-finland-to-compel-public-institutions-to-receive-electronic-invoices.html.

Accessed: 12 June 2021

Finanssiala, 2021. Finvoice 3.0 Soveltamisohje. URL:

https://file.finanssiala.fi/finvoice/Finvoice_3_0_soveltamisohje.pdf

Accessed: 20 June 2021

Visma, 2021. Mikä on OVT-TUNNUS JA Mitä Sillä Tehdään? URL: support.maventa.fi/fi/support/solutions/articles/77000474625-mik%C3%A4-on-ovt-tunnus-ja-mit%C3%A4-sill%C3%A4-tehd%C3%A4n-

[n-](https://support.maventa.fi/fi/support/solutions/articles/77000474625-mik%C3%A4-on-ovt-tunnus-ja-mit%C3%A4-sill%C3%A4-tehd%C3%A4n-)

Accessed: 20 May 2021

Loshin, Peter, 2021. What Is SSH (SECURE Shell)?. URL: searchsecurity.techtarget.com/definition/Secure-Shell.

[target.com/definition/Secure-Shell](https://searchsecurity.techtarget.com/definition/Secure-Shell).

Accessed: 10 June 2021

Chung, Conrad, 2014. SSH File Transfer Protocol (SFTP) Explained. URL: www.2brightsparks.com/resources/articles/ssh-file-transfer-protocol-explained.pdf

Accessed: 29 May 2021

Computer Hope, 2019. What Is a Script? URL: computerhope.com/jargon/s/script.htm.

Accessed: 29 May 2021

GeeksforGeeks, 2019. Introduction to Perl. URL: www.geeksforgeeks.org/introduction-to-perl/.

Accessed: 13 June 2021

Techopedia, 2016. What Is a Byte Order Mark (BOM)? URL: www.techopedia.com/definition/799/byte-order-mark-bom.

Accessed: 27 June 2021

Wincent, 2017. Remove BOM Marker from File with Vim. URL: wincent.com/wiki/Remove_BOM_marker_from_file_with_Vim

Accessed: 29 June 2021

SublimeText, 2017. Search and REPLACE – Single File. URL: sublime-text-unofficial-documentation.readthedocs.io/en/latest/search_and_replace/search_and_replace.html.

Accessed: 16 May 2021