

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Rasmus Rugam 166056

**Automation Software for Bookkeepers -
Account Reconciliation by Verifying and
Matching Corresponding Documents and Bank
Statements**

Bachelor's thesis

Supervisor: Tarvo Treier
MSc

Tallinn 2021

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Rasmus Rugam 166056

**Automatiseerimistarkvara raamatupidajatele -
konto lepitamine vastavate dokumentide ja
pangaväljavõtete kontrollimise ning sobitamise
teel**

Bakalaureusetöö

Juhendaja: Tarvo Treier
MSc

Tallinn 2021

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Rasmus Rugam

18.05.2021

Abstract

The aim of this thesis is to develop a prototype at the MVP level that helps automate accounting reconciliation.

The main problem is the time-consuming communication between the accountant and the owner of the company at the end of month. This communication becomes necessary when documents have not been sent to the accountant. The accountant needs a document for each bank transaction to be able to correctly perform the respective accounting registrations and for legal compliance.

The result of the thesis is a completed prototype that can find a connection between the user's bank transfers and the documents sent to the program. User can also see which bank transfer does not have a document and it is possible to link a document manually if the program has not been able to decide whether a particular document is suitable for the given bank transfer or not.

This thesis is written in English language and is 50 pages long, including 7 chapter, 10 figures.

Annotatsioon

Automatiseerimistarkvara raamatupidajatele - konto lepitamine vastavate dokumentide ja pangaväljavõtete kontrollimise ning sobitamise teel

Antud lõputöö eesmärk on välja töötada MVP tasemel prototüüp, mis aitab automatiseerida raamatupidamise konto lepitamist, mis on käsitsi tehtav tegevus.

Põhiprobleemiks on raamatupidaja ja ettevõtte omaniku aeganõudev suhtlemine iga kuu lõpus. See suhtlus muutub vajalikuks, kui raamatupidajale pole dokumente saadetud. Raamatupidaja vajab iga pangatehingu kohta dokumenti, et oleks võimalik vastavaid raamatupidamisregistreid õigesti teostada ja juriidilistele nõuetele vastavuseks.

Lõputöö tulemuseks on valminud prototüüp, mis suudab leida seose kasutaja pangaülekannete ja programmile saadetud dokumentide vahel. Samuti saab vaadata, millisel ülekandel pole dokumenti ja on võimalik dokumenti käsitsi seostada, kui programm pole suutnud otsustada, kas konkreetne dokument sobib antud ülekandeks või mitte.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 50 leheküljel, 7 peatükki, 10 joonist.

List of abbreviations and terms

AIS	Account Information Services
AISP	Account Information Services Provider
API	Application Programming Interface
APP	Authorized Push Payment
AWS	Amazon Web Services
CEO	Chief Executive Officer
CSV	Comma-Separated Values
DOC	Document file
FCA	Financial Conduct Authority
GAAP	Generally Accepted Accounting Principles
GDPR	General Data Protection Regulation
GIF	Graphics Interchange Format
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
IT	Information technology
JPG	File format used for images and graphics
MVC	Model-View-Controller
MVP	Minimum Viable Product
NPM	Node Package Manager
OCR	Optical Character Recognition
PDF	Portable Document Format
PIS	Payment Initiation Services
PISP	Payment Initiation Services Provider
PNG	Portable Network Graphics, file format for images
SME	Small or Medium-sized Enterprise
SQL	Structured Query Language
SaaS	Software as a Service

TPP	Third Party Provider
TXT	Text Message
UI	User Interface
VAT	Value-Added Tax

Table of contents

1 Introduction	11
1.1 Background and problem.....	11
1.2 Task setting.....	12
1.3 Thesis overview	13
2 Available solutions	14
2.1 Costpocket	14
2.2 Envoice	14
2.3 Xero	15
2.4 Overview of Tools	15
2.5 Different solution.....	16
3 Open banking introduction	20
3.1 Payment Service Directive2	21
3.2 Account Information Services and Payment Information Services.....	21
4 Software requirements	23
4.1 Objective and users of the subsystem.....	23
4.2 Main processes	23
4.3 System requirements.....	24
4.3.1 Functional requirements	24
4.3.2 Non-functional requirements.....	24
4.4 Business rules	25
5 Developed software	26
5.1 Choice of development tools & frameworks	26
5.1.1 React	28
5.1.2 Node.js.....	30
5.1.3 Open banking API	30
5.1.4 Amazon Simple Email Service.....	31
5.1.5 Amazon Textract	31
5.2 Software architecture	32

5.2.1 Receiving and parsing email	33
5.2.2 Receiving bank statement and processing it.....	36
5.2.3 Retrieving data from document	38
5.2.4 Document data and bank statement	40
6 Difficulties and Learnings, Risks and Next Steps	43
6.1 Development difficulties and Learnings.....	43
6.2 Risks	43
6.3 Further developments	44
7 Summary.....	46
References	47
Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis	50

List of figures

Figure 1. Explanation of double entry bookkeeping	17
Figure 2. What accountants are wishing in automation and digitalization, the survey was conducted by FreeAgent in 2021, which involved 2500 accountants	19
Figure 3. Model-View-Controller architecture.....	28
Figure 4 Usage of popular JavaScript front-end frameworks	29
Figure 5. Schema describes the main architecture of the system	33
Figure 6. Receiving email and parsing its contents and storing it.....	35
Figure 7. Schema of getting permissions to read users bank statements.....	37
Figure 8 Obtaining user bank transactions.	38
Figure 9 Retrieving data from the document.....	40
Figure 10 Matching document data with bank transaction.....	42

1 Introduction

Autumn 2020, company named moduulo OÜ decided to develop new SaaS(Software as a service) software mainly focused for the start-ups and SME(Small or medium-sized enterprises) companies. The author of this thesis developed GUI(Graphical user interface) and main functionality.

1.1 Background and problem

Reconciliation is an accounting process that is used to ensure that the transactions in a company's financial records are correct and in agreement.[2] By doing that this verifies that accounts in the general ledger are consistent, accurate and complete. Account reconciliation is used to explain the differences between two financial records or account balances. If there is an unexplained difference between the two data sets it could be a sign of money laundering or theft. However some fluctuations may be tolerated depending on the time of payment and deposit.[1]

Every year asset, liability and equity accounts are carried forward, that is why account reconciliation is very important. During reconciliation, the accountant needs to compare the transactions registered in the internal records account with an external data from sources such as banks and credit card companies. Coordination between the two data sets must be consistent with each other and any mismatch must be noted in the account reconciliation statement.[2]

One of the main responsibilities of companies is to submit monthly accounting relevant data to the accountant – no company can escape.[3] The secret to a company's success is good accounting, which gives business owners a better understanding of the financial and economic situation and a better understanding of the situation.[4]

If the company has not registered as a VAT(Value-added tax) payer, only the Income and Social Tax return must be submitted for the months in which the company has made

taxable expenses and/or labour payments. The income and social tax return must be submitted by the 10th of each month.[3]

If the company has registered as a VAT payer, the company must submit income and social tax and VAT returns. The return must be submitted even if there is no economic activity, in which case both returns must be submitted in the amount of 0. The VAT return must be submitted by the 20th day of each month in which the turnover of the previous month is reported.[3]

To make costs smoother and safer, companies can and should take shortcuts to preparing accounts. Businesses spend too much time searching for documents and sending them to the accountant to receive feedback from the accountant. There are often situations where the accountant finds in the course of his work that some documents are missing, and the company has to start looking for the relevant documents. Software that makes it possible to find a connection between the company's payments or incomes and invoices received by e-mail is difficult to find or does not exist at all.

In order to simplify the work of entrepreneurs and bookkeepers, the corresponding software can be developed, which matches the invoice received by the company and the payment whether it is an incoming or outgoing payment.

1.2 Task setting

The goal of the thesis is to find a solution on prototype level for simplification of the accounting preparation in start-ups and SME The thesis starts with an overview of the legal framework around accounting, then shows the available tools for automation and drafts a way of automated processes.

The MVP(Minimum viable product) contains:

- Front-end website.
- Backend server with business logic
- Open Banking API connection
- AWS(Amazon Web Services)

The result of this thesis is a developed software which automates the match between banking statement and documents and gives the business owner a simple dashboard that

indicates the existence of missing documents, which documents are identified by the system or which documents he can verify and match manually.

1.3 Thesis overview

The work is divided into 5 main parts. Available Options (Chapter 2) describes the options available in the market Costpocket, Envoice, Xero, how they are similar to each other, and how the MVP software developed for this thesis differs from them. The introduction to Open Banking (Chapter 3) describes what is Open Banking, Payment Service Directive2, Account information services and payment information services. The software requirements section (Chapter 4) presents the objective and users of the subsystem, functional requirements, non-functional requirements, and business rules. Developed software (Chapter 5) describes the software itself, what were the development tools and frameworks, and software architecture. Difficulties, risks and next steps talks (Chapter 6) about development difficulties, risks and future developments.

2 Available solutions

2.1 Costpocket

Costpocket is an Estonian company founded in 2016, which today makes the company 5 years old. This company is currently mainly engaged in the digitization of expense documents and expense checks. The procedure is simple and smooth, user must take a picture of the corresponding document that is recommended to be digitized and upload it to the Costpocket environment, where it will be digitized for you. Digitized documents can be managed either in the phone app or in the cloud environment they provide. The company mainly offers two automatic text recognition - robotic digitization or human digitization. [5][6]

Both digitations have their pros and cons. Robotic digitation is very fast and the result can be obtained in just a few seconds, but the data is at least 70% correct. Human digitization is a bit more time consuming process it may take up to 24 hours, but the data can be more than certain, the accuracy of the data is promised to be up to 99.5%. [6]

The digitized documents can be forwarded to the appropriate accounting software that the user prefers to use, the choice is quite large.

2.2 Envoice

Envoice is an Estonian company founded in 2014 in November, which today makes the company age 6 years. This company offers many different options. Documents can be digitized by sending documents to an e-mail address, where invoices can also be sent to an e-mail address by your suppliers as well as employees. Documents can be uploaded directly from the computer or a picture of the document can be taken using the Envoice mobile application. The company uses its SmartExtract system to identify documents in order to obtain all the necessary information from the document. If the picture is too blurry or the document is badly scanned there is a chance that accountant must do it manually. Expense checks can be quickly photographed and indicated as an expense or shipping report within the company. It is possible to send purchase invoices and expense reports for internal coordination, where the company's manager could check, comment and confirm the respective expenses. It also offers the possibility to create sales invoices,

the use is made very convenient, you can enter pre-saved contacts, products and services immediately and forward the invoice immediately. The prepared invoice can also be submitted immediately to the accounting software. [7][8]

2.3 Xero

Xero is a cloud-based accounting software focused primarily on small businesses. Founded in New Zealand in 2006, Xero has today become one of the largest platforms with multiple branches around the world and thousands of employees. Today, it has taken a leading role in the cloud-based accounting market in New Zealand, Australia and England. Xero offers many different options - tracking and paying invoices, VAT returns, tracking projects, saving documents and processing them, etc. It also offers the ability to perform account reconciliation, but only in a manual way. This means that the bank statement must be obtained from the bank and imported into their application and in the correct form, or all payments must be entered manually. Xero requires that the bank statement must be in CSV format, all columns should have an explanation of what data is in that column, and a CSV file can contain up to 1000 lines. If the CSV file is larger than 1000 lines, a new file must be created. Once the bank statement has been uploaded to the environment, further work must be done manually. User must start looking for a large pile of bank payments and match them to your expenses or income, which is a very time-consuming process. Xero has made it possible to use mainly banking import with some New Zealand, Australian and UK banks, where their main market is.[9][10]

2.4 Overview of Tools

The aforementioned companies are mainly involved in document recognition and digitization. As a result of the applications, it is possible to send a digitized document to an accountant who must deal with the document further, to reconcile it with specific payment to which it relates. There is also a possibility to check your older documents. The data that is used is also written into the accounting system and particularly into main journal and general ledger.

The biggest of the previously mentioned companies is the Xero, which is capable of doing almost everything you need as a small business owner. Xero has created a different solution for digitizing documents called HubDoc. You can send DOC, PDF, GIF, JPG,

PNG, HTML and TXT files to this application. Where the corresponding activities are performed and the data obtained from it can be forwarded to Xeros, where the data can be further processed accordingly.[11]

2.5 Different solution

Looking at the statistics of the Center of Registers and Information Systems, it turns out that as of 1st of May 2021, 22712 companies have been registered in Estonia[12]. As mentioned in Chapter 1.1 above, no company can escape accounting[13]. It is difficult to say the exact number of how many of these companies have correct accounting and how many companies are in difficulty. All that author knows is that there are companies that have difficulty keeping accounting in order. For start-ups and small businesses who do not have enough experience in accounting, this can cause excessive problems and, as a result, a great need of time. Inadequate accounting can also lead to problems at national level.

This thesis focuses on accounting preparation and automation for accountants, start-ups and SME businesses, which is why only a general description of the respective business model is provided.

The purpose of the developed prototype software is to automate the accounting of accountants, start-ups and SME businesses in such a way that the business owner knows which documents correspond to the payment coming in or out of his account. The business owner must do the following - allow the software to read company's bank statement and forward the corresponding invoice to the email address provided by the system. From there, the application can already operate on its own. Business owner also has the ability to constantly monitor the process through the application. If the software has not sufficiently identified which document correspond to which bank transfer, then the user will be shown all possible documents that could correspond to the given bank transfer. These documents are displayed on the dashboard webpage, where he can then confirm them himself/herself.

There is no specific way to reconcile your account. However, GAAP(Generally Accepted Accounting Principles) require double entry. Double-entry bookkeeping is a very useful way to reconcile accounts. This means that the transaction is entered in the general ledger

and each financial transaction is posted to two accounts, a debit account and a credit account. Following this, even in the event of an accidental error, it is very easy to find where the error has occurred.[1]

DATE	DESCRIPTION	NOTES	TRANSACTION VALUE		ACCOUNT
			EXPENSE(DEBIT)	INCOME(CREDIT)	
01.05.2021	DEPOSITS		2000€		CASH
02.05.2021	New pressure washer			2000€	ASSET
03.05.2021	WORK:			500€	REVENUE
04.05.2021	WITHDRAW		500€		ACCOUNTS RECEIVABLE

Figure 1. Explanation of double entry bookkeeping

Double-entry bookkeeping is explained based on figure 1. Let us say X person is investing in the company 2000€. He spends the money on the pressure washer what is now considered as company asset, what gives value to the company. Now the person will do some work and receive credit for the work he has done, we call this revenue. As the owner of the company would like to receive that revenue, he has to do 500€ expense to the accounts receivable.

In account reconciliation the debit and credit values should balance each other, and the end result should be zero. However, this is not always the case, there might be situations where is possibility of earning more and not withdrawing the profit. Which leaves the company with a profit margin called net revenue profit.

As mentioned earlier, the target group consists of accountants, start-ups and SME owners. The developed software helps to eliminate every month activity where the business owner has to monitor the account statement and find the corresponding document, the system does it for him. This saves the business owner a huge amount of time that he/she can spend on something else, be it marketing or communicating with new customers. On the other hand, it also saves time for bookkeepers where they do not have to manually verify which document is corresponding to certain transaction. Business owners do the part of bookkeepers and bookkeepers can check that the company owner has all the documentation that has to be there. This opens up the possibility of doing your own accounting and most importantly, doing it properly.

The demand for software arose from the fact that many business owners complained that they were having trouble with their accounting every month. The problem was that at the end of each month they had to look up the relevant documents in their e-mails, where the money was spent and where the money came from and submit the relevant documents to the accountant or do the accounting themselves. As a result, it was necessary to spend a lot of time searching for all the documents on a monthly basis, and even if the business owner thought he had all the documents, it turned out that some documents were still missing and had to be searched. Communicating with the accountant by e-mail is a time-consuming process, as you never know when one of the parties will reply, whether all the documents have now been received or if some documents are still missing and which ones. Business owners complained that there was not any software to simplify their accounting, which would save them a lot of time.

"Having two companies in two countries (Estonia, Russia) selling SaaS products to Real Estate companies managing bigger-sized malls, I constantly fail to deliver all documents my different accountants need. Every month they alert me to forward missing invoices and purchase orders just before the deadline for VAT declaration happens. As a CEO(Chief Executive Officer), I'm in charge of developing my product, doing many sales activities, and managing all the other business-related issues of my company. Accounting preparation is regularly interrupting me in my monthly activities and takes so much precious time." – Denis Ivanov, Founder & CEO ORDIN.

"We are a young start-up in its seed stage, and as every company under Estonian law, we have to fulfil accounting obligations every month. As our processes are not all set up as they are in long-time established corporates, there is a big part of manual jobs, especially accounting preparation. Collecting all documents, checking if they all have been forwarded to our accountant, and then control if reports and VAT has been appropriately prepared takes a consequent amount of time every month. I wish I had a tool which automates this process for me and free up my time for more exciting work: Make Adact's customer happy and help their marketers create high engagement gamified campaigns." – Kalev Kärpuk, Founder & CEO adact OÜ.

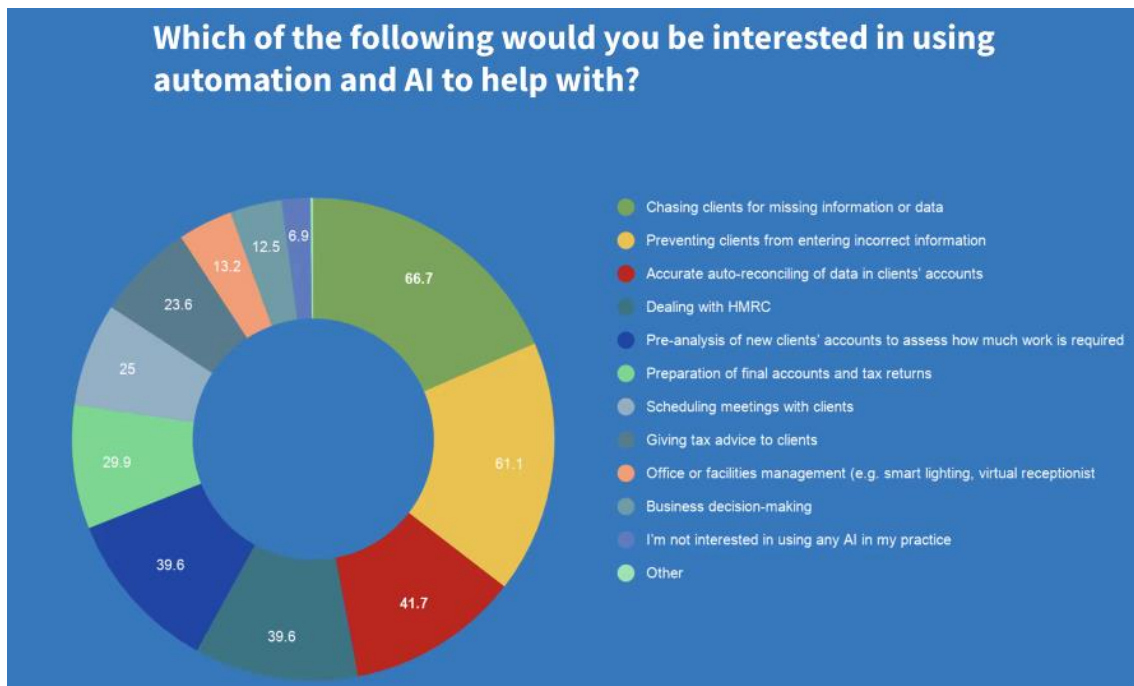


Figure 2. What accountants are wishing in automation and digitalization, the survey was conducted by FreeAgent in 2021, which involved 2500 accountants

The planned main result is to create a prototype solution on the MVP scale, which will help bookkeepers, start-ups, small and medium-sized enterprises to reduce the time spent doing accounting and preparing accounts by 30%. The solution includes a Front-end as well as a Back-end solution, where the user can perform the corresponding actions on the website.

During the specific development work, the answers to the following questions will be sought.

- Is the software able to automate the connection between the corresponding invoice and the bank transfer?
- What security considerations must be considered?
- Archiving to be legal compliant
- What if the system is not sure that this is the corresponding documents?
- How do we access the bank accounts?

3 Open banking introduction

On 8 October 2015, The European Parliament, on the proposal from the European commission, adopted an amended Payment Services Directive called PSD2. The new rules aim to promote the development and use of innovative online and mobile payments through open banking.[14] Open banking is a set of solutions that allows banks to share financial information electronically, securely, and only under the permission of clients. Granting access to its data and products via an API (Application Programming Interface). Allows TPP (Third Party Providers) to access financial information efficiently, which promotes the development of financial and non-financial applications and services.[15] Based on third-party providers field of industry, API's will then use shared data (and customer financial counterparty data). Uses can include matching customer accounts and transaction history with a variety of financial service choices, aggregating data from participating financial institutions and consumers to create marketing profiles or creating new transactions and account changes on behalf of the customer. [16]

Getting a loan becomes more streamlined and transparent. There will be no need of gathering information manually or filling in the applications from different sources. Open banking lets you request an offer from each service provider at the same time and you only have to provide your information once. The potential lender therefore be able to grab the information they need and make the best offer. In case applying loan for the company you will no longer have to submit your financial statements, bank itself can download all the necessary information from all the banks.[17]

On privacy matter, the customer must choose carefully about the TPP's usage, which may lead to creating a new standard or "trustmark". Open banking depends on data sharing, but you may want the details to be kept confidential. Luckily, open banking should not automatically decrease confidentiality or privacy. In order to secure and educate the customer of emerging threats, TPPs and banks will need to take action to inform about the risks of APP(Authorized Push Payment) fraud, phishing and other attacks.[18]

3.1 Payment Service Directive2

It all started in 2007 with the Payment Service Providers Directive, with the aim of contributing to the development of the European Union's single payments market in order to boost efficiency, competition and innovation in the European Union. [19]

To ensure consumer protection, increase competition and innovation in the market, and ensure payment market security, which is expected to ease the development of new payment methods and shops. The European Commission proposed an adjustment aimed at strengthening these objectives. Hence the name PSD2[19]

As stated in [17], PSD2 will organize and streamline two types of events that have occurred since the early adoption of PSD in 2007 and have become more popular in recent years: PIS(Payment Initiation Service) and on the other hand AIS(Account Information Service).

3.2 Account Information Services and Payment Information Services

Companies must complete a comprehensive application process with the FCA(Financial Conduct Authority) in order to be regulated as either an AISP(Account Information Service Provider) or PISP(Payment Initiation Service Provider) or both. The provision of both services is allowed to open banking suppliers, but many have only one authorization. Customer permission is necessary to access Open Banking, data is managed by AISP and PISP. This ensures that every AISP and PISP explicitly indicates to the end-user about how long and by whom data can be obtained. This digital consent journey is also the basis for data processing under GDPR(General Data Protection Regulation) for AISPs and PISPs.[20]

Being approved as AISP ensures that a corporation is able to seek for authorization to connect to a bank account and use that information to provide service. AISP businesses are allowed to only view bank account records on a “read-only” basis. They can look, but they can’t pass money to a consumers.[21]

Companies that are approved by PISP are allowed to make transfers on behalf of a client instead of merely displaying data on an account. PISP do this using their own devices by initiating direct payments to or from the payer’s bank account. This has prompted some

analysts from the industry to call AISP 'read-only' access to an individual account, whereas PISP provide 'read-write' access[20].

4 Software requirements

4.1 Objective and users of the subsystem

The purpose of the developed prototype at the MVP level is the creation and implementation of a new IT tool which does the automate accounting. The user of the software must be able to register and log in to the environment. The software is specifically designed for the user to use, which means that the user performs all the operations himself. For this subsystem, there is no administrator role that must validate user actions in the application. The user must be able to have access to read and act on their bank details if they wish to use the application as intended.

4.2 Main processes

The following list shows the main processes that can be performed in the software by the user:

- User registration
- Logging in to the application
- Registering a company
- Looking company information
- Adding suppliers (not necessary to fully operate on the application)
- Deleting suppliers (not necessary to fully operate on the application)
- Selecting the corresponding bank
- Logging into the bank system
- Giving the permission to read his bank account statements
- Looking his bank account statement
- Able to look possible matched documents
- Verify the documents

The main processes are the activities to ensure the operation of the application. The user is not obliged to use all the above main processes to achieve the desired result.

4.3 System requirements

4.3.1 Functional requirements

- User must be able to register
- User must get verification code to his/her email account
- User must verify his account on the website
- In case of forgetting password, user must be able to reset his password.
- Must get password reset code to email account
- Must be able reset password
- User must be able to log in.
- User must be able to register his/her company
- User must be able to see his/her company information
- User must be able to create new supplier
- User must be able to add bank account
- User must be able to choose between countries of the bank
- User must be able to choose the right bank of the country
- User must be able to connect to the bank
- User must be able to login in the bank interface
- User must be able to give access to read his bank statements
- User must be redirected back to the original webpage
- User must be able to choose between different bank accounts if he/she has multiple.
- User must be able to see possible match transactions
- User must be able to see verified match transactions
- User must be able to see possible matched documents to the corresponding transactions and the percentage
- User must be able to verify the corresponding document to the transactions

4.3.2 Non-functional requirements

- The application must be available and usable in the most popular web browsers without having to install any specific software.
- The application should be able to be used by at least 100 users at a time without any errors.

- When making a request, the response should not take more than 5 seconds to appear.
- It must not take more than 5 seconds for the system to save the data change.
- The principles of user interface design must be followed consistently.
- The user interface must have a main page that can be distinguished from other related activities.
- Required input fields in the user interface must be marked (for example, by adding * to the tag).
- The fields for adding data and changing data in the user interface must be distinguishable.
- User interface must be size responsive depending on the client screen resolution.
- In the application the data exchange between the client computer and the server must take place via the HTTPS protocol.

4.4 Business rules

- Each registered account has a unique code
- Each registered company has a unique code
- Each email sent to the application must correspond to a specific company
- Every 90 days business owner must give the rights to read their bank statement if the user wishes to use open banking feature.

5 Developed software

Before developing the application, it was necessary to think about how it should be developed. The decision was made that it should be developed as a web application and not as separate software that should be downloaded and installed on a computer. The main reasons for making this choice:

- The completed application will then be easier to integrate with other components to be completed in the future.
- The desktop application may be restricted by the hardware on which it is intended to be used. An application may be hampered by the need to build the application on several different operating systems, which requires additional resources. In case of an update to the software that requires more computer power for the application to run smoothly, then the user of the application must upgrade their hardware, which may result in the loss of one potential customer. Updating the software can also cause people problems who are not experienced enough in handling computers.[22]
- As a web application, the software can be used in most browsers and they run on both operating systems (Windows, iOS, etc.). Also, software updates are much easier to perform, in most cases the customer may not even notice when the software update has been performed, because he does not have to do anything. If the web software is well enough developed, it can also be accessed through the mobile phone.[23]

5.1 Choice of development tools & frameworks

In 2006, AWS began offering IT infrastructure services to businesses in the form of web services known as cloud computing. One of the dominant aspects of cloud computing is the ability to replace infrastructure start-up costs with low variable costs that suit the company. Using the cloud, companies no longer have to spend weeks or months planning to procure servers and other IT infrastructure. Using AWS, you can instantly start and configure many or even thousands of servers and deliver results faster.[24]

AWS is able to offer its customers significantly more variety of services and features than any other cloud provider. Starting with infrastructure technologies such as computing,

storage and databases, machine learning and artificial intelligence, data lakes and analytics and the Internet of Things. Today, AWS is one of the most flexible and secure cloud computing environments. To ensure security and meet security requirements, it supports a comprehensive set of cloud security tools with 230 security, compliance and management services and features to secure their core infrastructure used by the military, global banks and other highly sensitive organizations. AWS also supports 90 security standards and certificates of compliance, and all 117 AWS services that store customer data offer the ability to encrypt data.[25]

Claudia.js is a tool that helps implement Node.js projects in AWS Lambda and AWS API Gateway. Automates all vulnerable deployment and configuration operations of JavaScript and its associated API gateway, saving the developer time and setting everything up exactly as the developer expects. AWS Lambda and API Gateway are flexible, but configuring them can be tedious, especially in simple scenarios. Claudia automates all of these operations and uses standard NPM(Node Package Manger) packaging conventions that do not require a change in project layout. You can start the Lambda Micro service immediately and focus on development work without having to deal with AWS deployment workflows, all what one has to do is "claudia create" and Claudia compresses and processes code, grabs all dependencies, cleans up irrelevant resources, uploads to Lambda, configures the web APIs and updates version aliases.[26]

When the development of the software started, there were several questions about which database to use. Amazon DynamoDB was chosen mainly because of its scalability. DynamoDB is a no-SQL(Structured Query Language) database of core values and documents that provides single-digit millisecond performance on any scale. Performance is promised to handle more than 10 trillion queries per day and support more than 20 million queries per day. A fully managed, multi-region, multi-active and secure database with built-in security, backup and restore, and memory cache for all Internet-enabled applications.[27]

AWS Cognito provides key features to enable new web applications and mobile application users to manage and authenticate different use cases. Web applications typically require a valid username and password combination to successfully sign into the application. Modern authentication streams include more approaches to ensuring user authentication when using Cognito, this is no exception due to the capabilities and

features offered by Cognito. The service allows users to log in directly with their user data managed by AWS Cognito on behalf of web and mobile applications. Also allows you to sign in through third-party social networking applications such as Facebook, Amazon, Google. The service is designed to provide APIs and infrastructure for user management space for basic functions - authentication, authorization and user repository management with various operations for web and mobile applications.[28]

5.1.1 React

React is a JavaScript library created by Facebook software engineer Jordan Walke. Was invented to create fast and interactive user interfaces for web and mobile applications that handle time-changing data. React is a front-end library based on open-source components that is solely responsible for the application view layer. In the MVC (figure 3) architecture, the view layer is responsible for what the application looks like.[29]

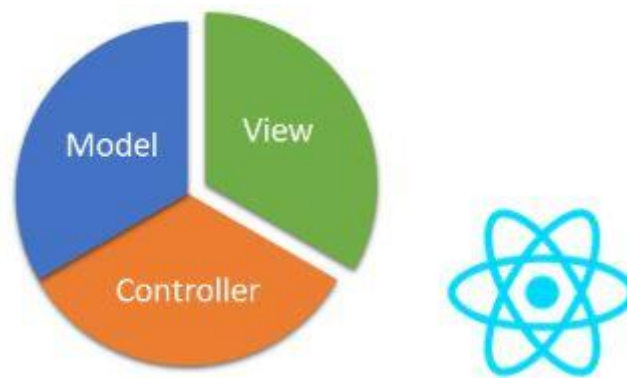


Figure 3. Model-View-Controller architecture

Angular is a TypeScript-based open-source JavaScript framework maintained by the Google Team and the community of individuals. Angular has been rewritten by the same team that built AngularJS. The main goal is to develop one-page applications. Angular aims to provide developers with a standard structure that allows its users to build large applications in a maintainable manner.[30][31]

Vue is a progressive JavaScript framework created by Evan You who worked for Google. Vue is mainly used to create single-page applications, web interfaces, and to develop desktop and mobile applications with the Electron framework. The name of the framework, Vue, comes from the word view, which sounds phonetically similar in English and corresponds to the traditional MVC architecture[32]. Vue is like a little mouse between two big elephants when we talk about React and Angular in web

development. How is Vue connected to the two major and popular frameworks? As mentioned earlier, Vue created Evan You when he worked on developing Google with Angular. It was born out of the need to build more efficient applications. Vue chose the best parts of Angular, for example, from the syntax of easy-to-use corner templates, but removed the supposed and complex stack that the corner needs and made it very functional. React also chose the best parts, such as two-way data linking, props concept, component-based approach. Combining the best parts of both frameworks, a new Vue framework was created. The creation of a good framework is also proved by the rapid gaining popularity among developers (figure 4). [33][34]

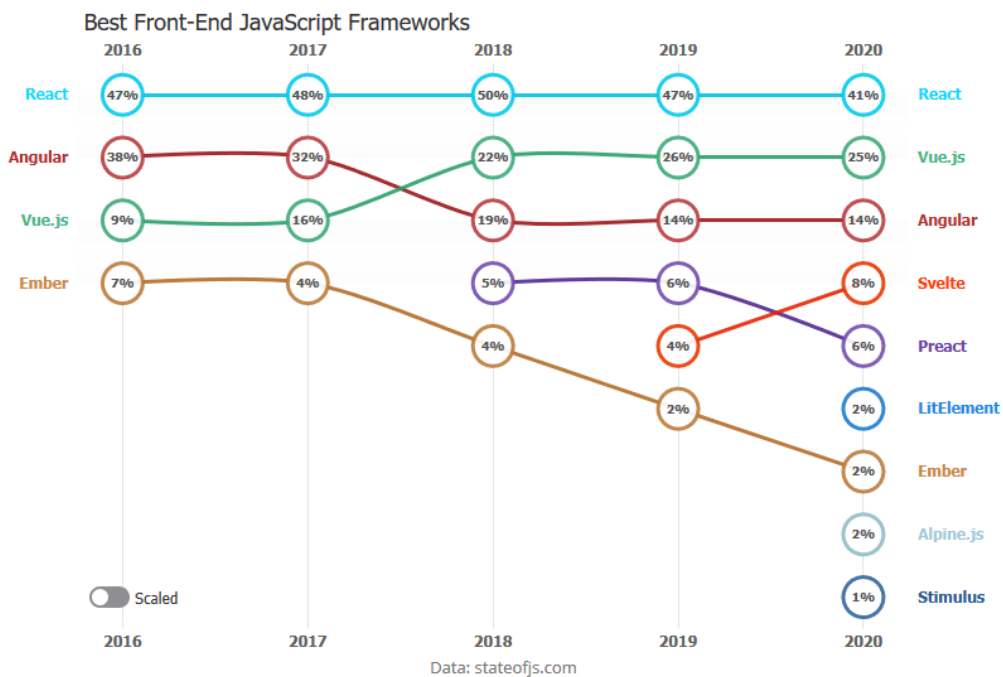


Figure 4 Usage of popular JavaScript front-end frameworks [35]

Looking at the 3 very popular frameworks mentioned above, it turned out to be a final decision that React framework will be used in the development. The reasons for this are as follows. Since author has used Angular-based development before, author did not like its complex structure and the first contact with Angular was rather unpleasant, then it was ruled out. Comparing Vue and React, it is hard to say which one author prefers, but author can say that he has used React before and was already familiar with some things, and it was a little easier for author to continue learning React rather than start learning Vue from scratch. It is also possible to connect developed system to the whole system more easily because the company uses React framework also for user interfaces. That also saves more

time to merge the two systems together rather than start developing the UI(User Interface) from scratch.

5.1.2 Node.js

Node.js is an open-source cross-platform JavaScript runtime and library for running web applications outside the client's browser. Node.js was developed by Ryan Dahl in 2009 and its latest version 15.14 was released in April 2021. Node.js is great for developers to use data-intensive applications because it uses an asynchronous event-driven model. Node.js is used to develop web chat applications, video streaming sites, one-page applications, and many other intensive web applications.[36]

With Claudia.js and Node.js it was possible to quickly set up the server without having to spend an insane amount of time for configuration and deploying, Claudia.js being used only for API Gateway deployment. It certainly played a role in being able to learn a new language that is on a popularity trend and highly demanded in the current job market. In the current company, from whom the idea for the topic of the dissertation came, developments are also being made with Node.js.

5.1.3 Open banking API

It was known that the interface had to be automatic through open banking in order to be as automated as possible. You should not manually extract your bank statements, import them into the correct format and then upload them. We had to look for the available Open Banking APIs, and it turned out that there were many.

- Citadele banka Eesti filiaal
- Coop Pank
- LHV Pank
- Luminor Pank
- SEB Pank
- Swedbank

However, accessibility proved to be very difficult. The first test was LHV bank, which offered an immediate opportunity to try and implement Sandbox. After a successful implementation, it became clear that it was not possible to test enough in the Sandbox environment and try to see if everything would work the way it should. In the sandbox

mode there was only three test payments, and they were very similar to each other. Sending them and researching whether it would be possible to access the Live environment with the moduulo company to get real data turned out to be a very long-lasting exchange of e-mails and it left a big question mark should author even continue with LHV. While waiting for the emails to confirm whether such a solution would be possible, Author found a platform where the Open Banking API is offered free of charge and the possibility is to receive data from several different banks from different countries in one implementation – Nordigen API. This solution was suitable because different banks have somewhat different APIs that require different implementation methods and would be time consuming and require more work.

5.1.4 Amazon Simple Email Service

Emails with the appropriate documents had to arrive in the application in order to be able to automate as much as possible and get as much information from these emails as possible. The SendGrid [39] service was launched, which then intercepted the e-mail sent to the corresponding e-mail address, and this data could be used and saved according to the situation. During the implementation, the corresponding emails were received and the data that needed to be received was received, but if the documentation was not sufficient, it was difficult to make the service work perfectly. There were various problems and this service was replaced by AWS Simple Email Service. This was because the service was better documented and did not have the same problems as SendGrid. All in all, it is better to have switched to the AWS Simple Email service, because it is known that we use the services of one provider and the accessibility is all in one place and not spread over different providers.

5.1.5 Amazon Textract

Amazon Textract is a machine learning service that automatically extracts text, handwriting and data from scanned documents, going beyond simple OCR(Optical Character Recognition) to identify, understand and extract data from forms and tables. When Amazon Textract was introduced in the AWS re: Invent Conference, it was said at the time that Textract would end the OCR industry. Many companies and organizations have to extract data from important documents, such as contracts or payment and tax documents. Prior to the introduction of Amazon Textract, companies followed the usual practice of either hiring a person or implementing OCR, which often requires

reconfiguration when changing the format of the document. Textract is a union of machine learning and OCR. It recognizes text, analyzes it and processes it in real time. Amazon engineers have trained Textract on millions of documents so that the machine can recognize as many documents as possible submitted by the user. This service allows you to retrieve data in minutes, hours or days.[37][38]

5.2 Software architecture

The final software consists of several components - Website, API, Nordigen API (Open Banking). The flowchart (Figure 5) describes the final workflow of the program. The user must be registered and have also registered their company. By using the website, user can give permission to read the company's bank statement, and the user can also see the bank data on the website. By sending the necessary documents to the respective e-mail, various operations are performed in the API in order to obtain the necessary data from the document and perform further operations. There are many different activities in the API - data storage, data reading, data parsing, etc. Finally, the user should see on the website which documents have been identified by the system and which documents he/she can confirm if the system has not been able to identify them sufficiently.

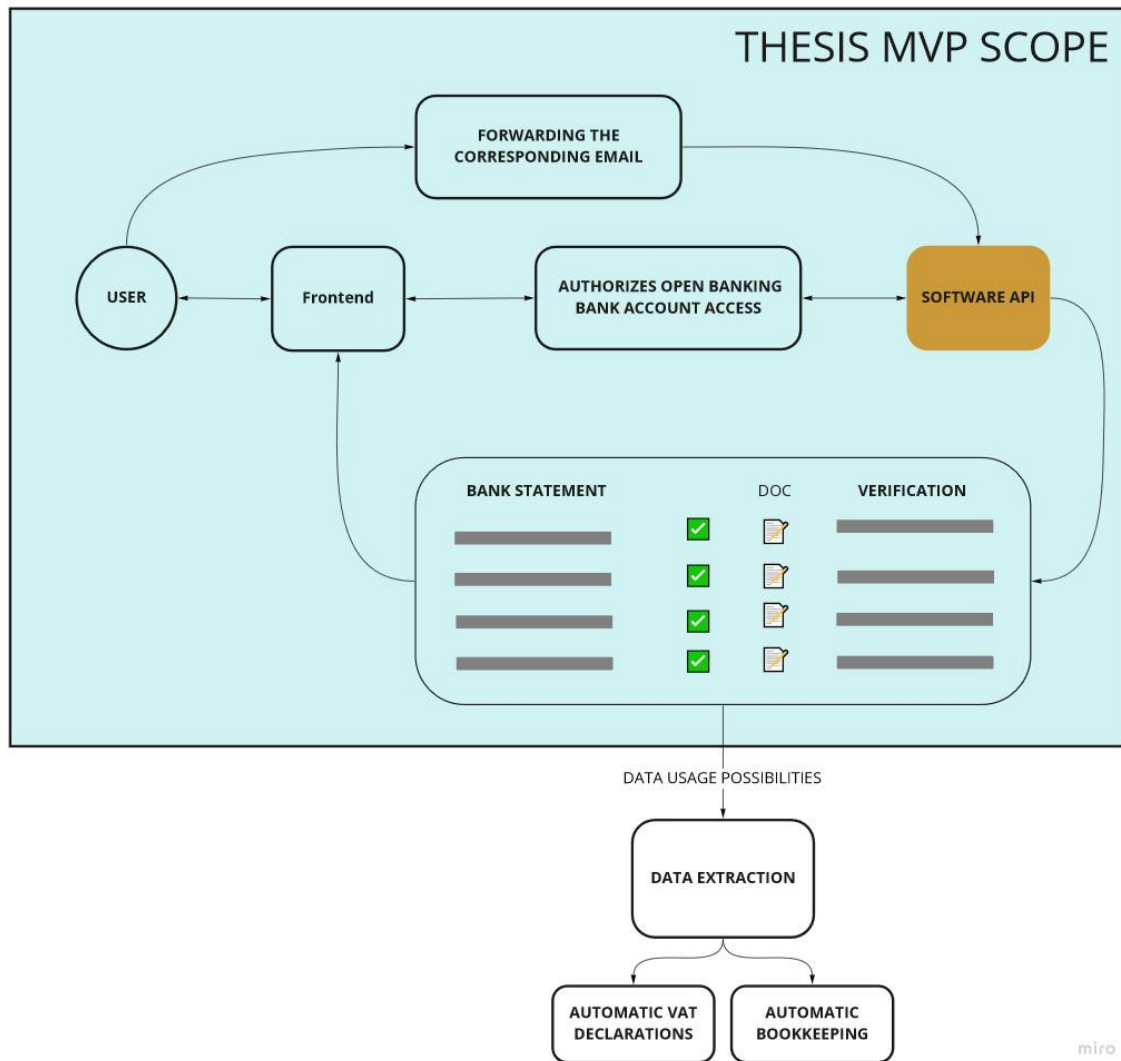


Figure 5. Schema describes the main architecture of the system

5.2.1 Receiving and parsing email

Looking at the flowchart (Figure 6), we can see how e-mail retrieval and parsing work. In order to send your email to the platform, you must first be a registered company. After that, the user gets a unique companyId for his company. The identifier is important because it will be part of the email to which the desired emails will be forwarded (example: companyId@moduulo.org). Once the user has forwarded the desired email to the corresponding email address, the AWS Simple Notification Service catches it, and the lambda function will be triggered. Lambda function will store the object into S3 bucket and sends the S3 object id to API. In the server the object id will be received, and the request is made to S3 bucket to get that object for parsing. When parsing out the object we will get the email where it was forwarded to. This allows us to identify the companyId so we will know which company forwarded it. From there, we have the option to take the

content and attachment of the email and save it in the correct company folder in the S3 bucket. In the database, we store the e-mail attached to the company and also store the following data – By whom it was sent, date, attachment in base64 format and content of the email.

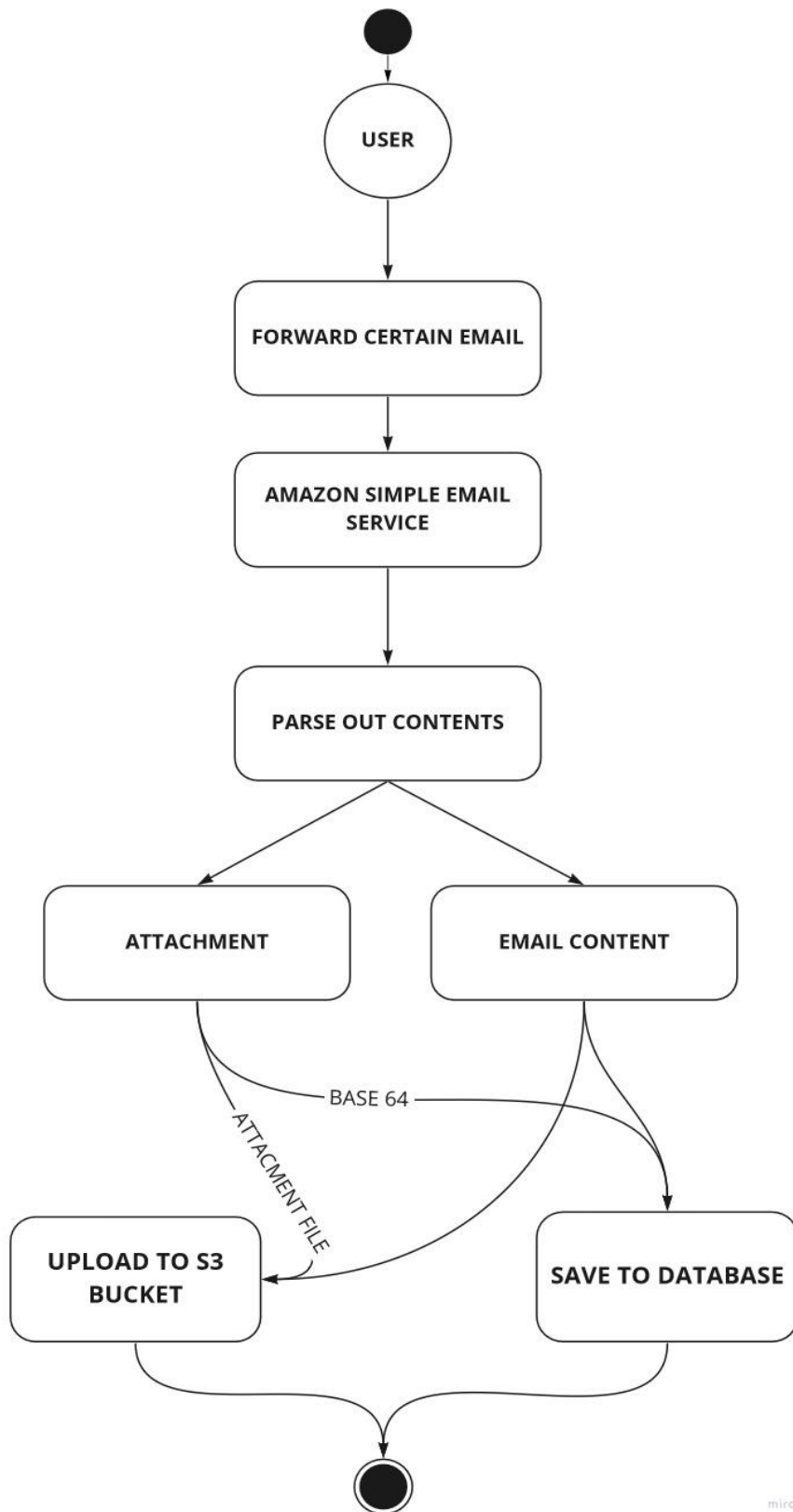


Figure 6. Receiving email and parsing its contents and storing it

5.2.2 Receiving bank statement and processing it

In order to connect your bank to the application and see your statement in the application, you need to perform the following actions. The schema of workflow (Figure 7) visually describes how the user has the ability to give permissions to read bank details. User must first choose the country where his bank is located, then he will have a choice of which banks to operate with. By choosing the right bank, he will be redirected to the Nordigen permission page, which will tell the user what will be received from the bank account and for how long. The user also agrees to Nordigen's Terms and Service and Privacy Policy by clicking the accept button. If the person agrees, it will be redirected to the respective bank page and if the user does not agree, there is the opportunity to return to the website and skip this process, but then it is not possible to receive the bank details. Being on the bank's website, the user can log in to the bank's portal using various identification methods and confirm that they consent to receive bank details. After confirmation, the user is redirected back to the website and then a unique identifier of the user's bank account is obtained, which then symbolizes a specific account - there can be several of them, which are stored in the database in the company table.

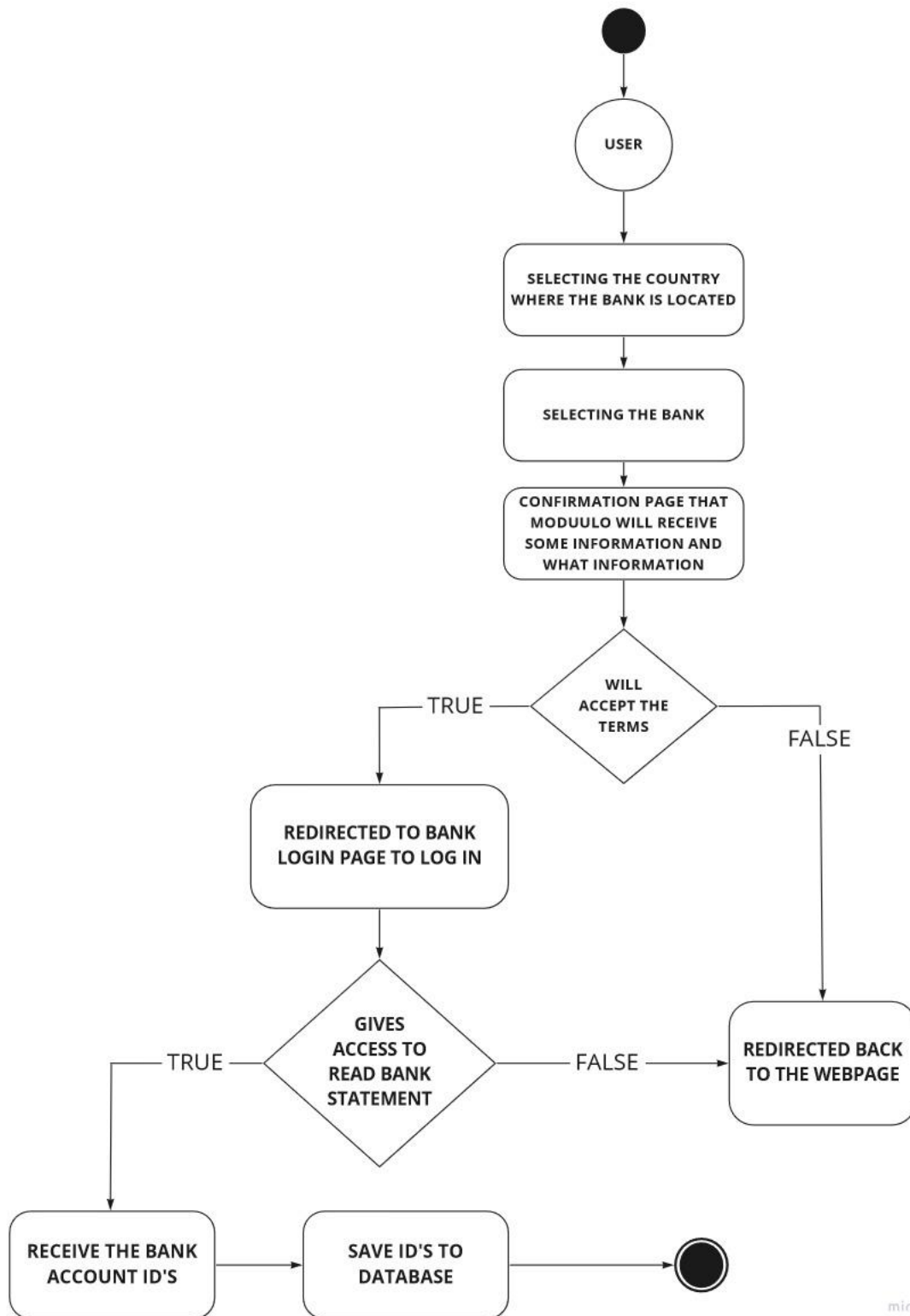


Figure 7. Schema of getting permissions to read users bank statements

When selecting the respective account, the unique identifier of this account is taken, which is used to query the path on the Nordigen API, which then returns the transactions of the given bank account (Figure 8).

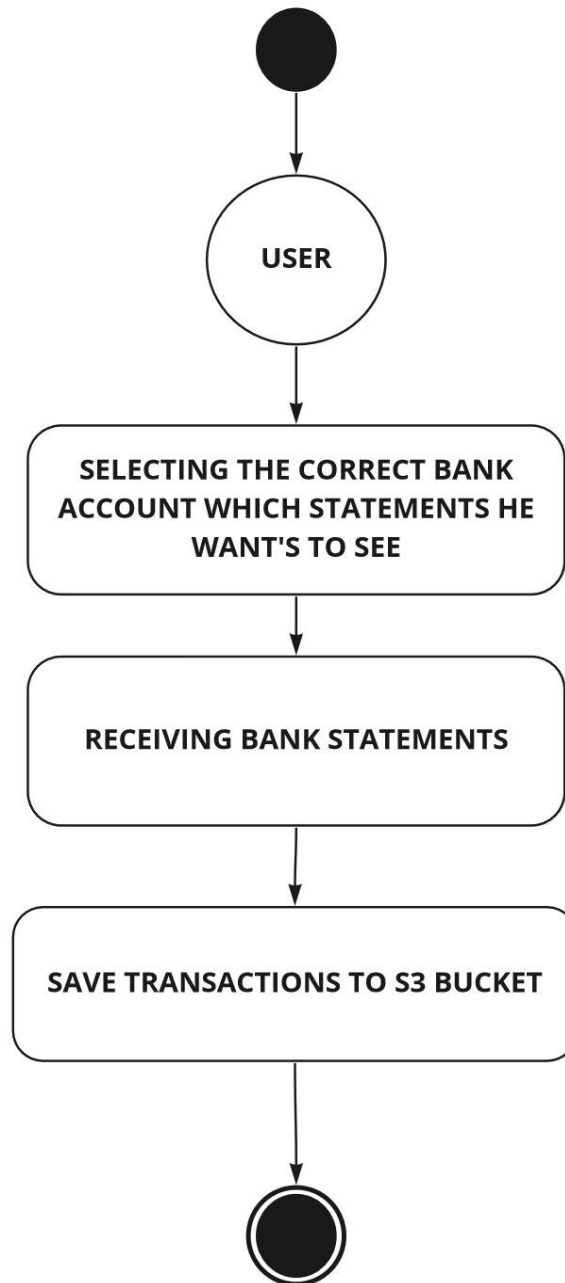


Figure 8 Obtaining user bank transactions.

5.2.3 Retrieving data from document

Section 5.2.1 above describes how the transmission of e-mail takes place and its processing, then author would not dwell on this context. Looking at the part where the attachment is stored in the S3 bucket in the corresponding company folder (Figure 9). When the attachment reaches the corresponding folder and a new document is detected, the Lambda function is launched which runs the Amazon Textract machine learning service. Which triggers a function that allows you to identify fields and values in the

document. If this document passes the function without any problems, it is announced through the AWS Simple Notification Service that another function should be started in Lambda. The main purpose of the second function is to identify the corresponding relations and write the results of the fields and values to a file and upload this file to the corresponding folder where the original document is located. The location of the completed file is saved in the email table originally created in the database. If coincidentally in the future, it would be necessary to find out in which folder and which file it is.

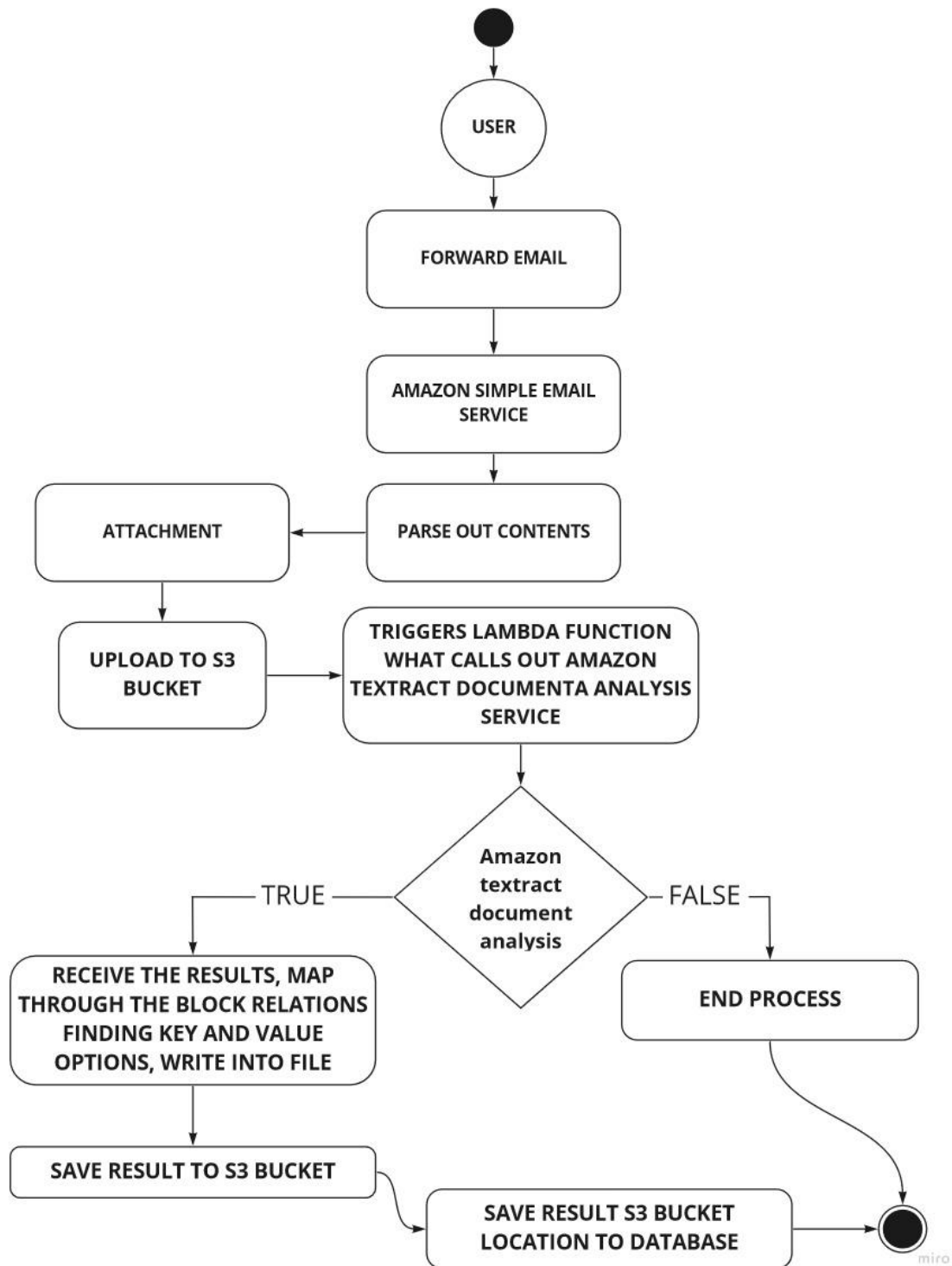


Figure 9 Retrieving data from the document

5.2.4 Document data and bank statement

Finding a link between document data and bank data is one of the most important in terms of application importance. It is a link that unites both parties and gives us the desired result, without which the part of automation that is to be automated would not work.

Figure 10 describes to us how a part works that brings together both parties and gives a visual result to the user. In the initial phase of the system, the user's bank statements are taken for the respective month. The second reference moment is the invoice key value pairs of the given user, which will be dealt with in the following comparisons. It is checked which key value pairs are associated with the invoice, which system was sent, so that we know where to save the data in case of successful identification. Coincidentally, if bank invoices should already exist in some invoices, they are stored as verified, usually not the first time. The loop starts by taking the data of one invoice and comparing it with the data of each transfer. Are the dates compatible with the given bank transfer and invoice details, etc. If the match between the bank data and the invoice data is more than 80%, it is stored in the database as well as in the API as a verified match. If by chance there should be less than 80% match, it will not be stored in the database and will be detected if no match is found. In this way, all invoice data is processed with transfers throughout the month, and if there is no more invoice data to go through, the received data is collected and displayed on the website to the user, where he can see it. What has been identified by the system and what documents can it certify if necessary.

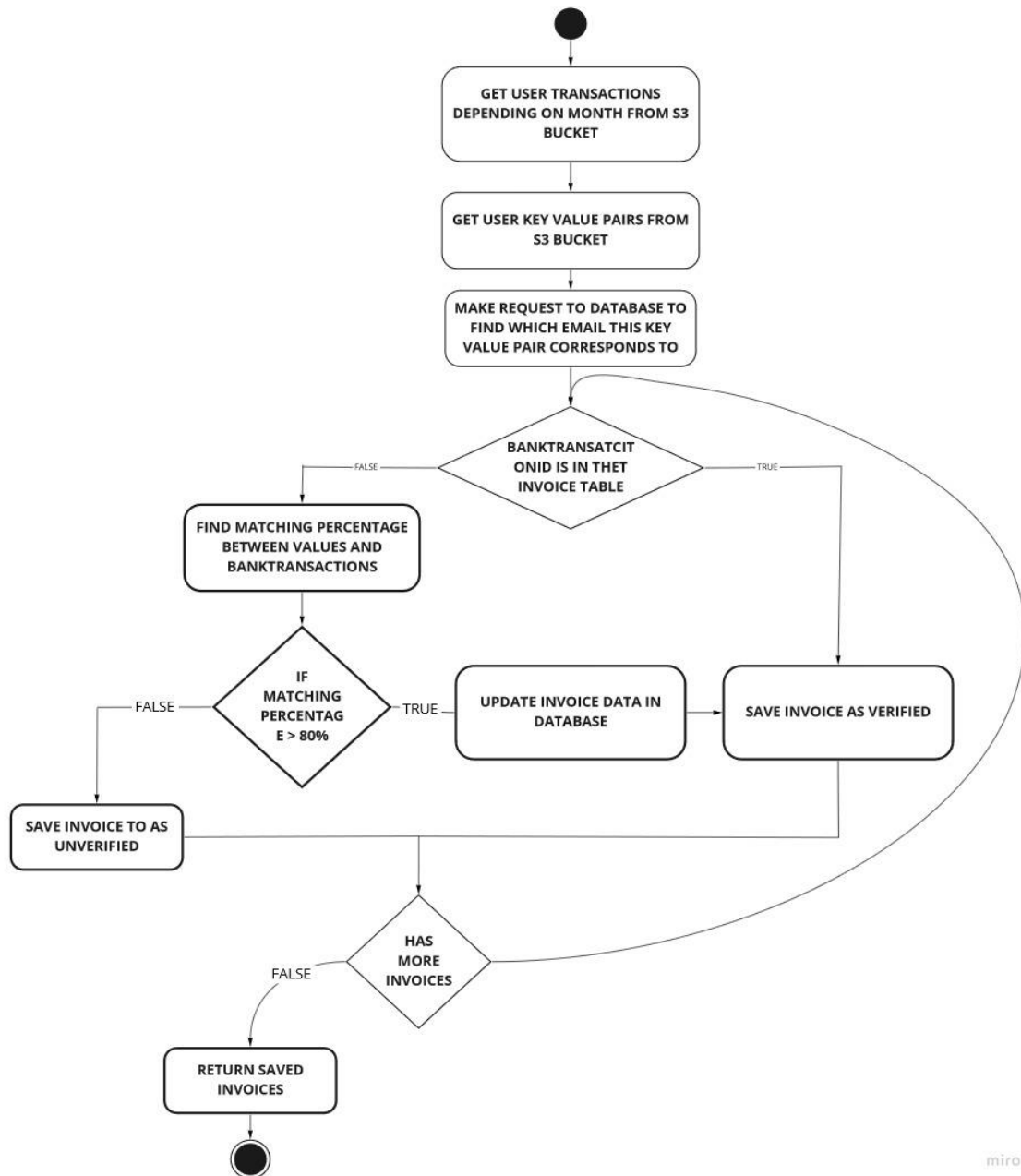


Figure 10 Matching document data with bank transaction

6 Difficulties and Learnings, Risks and Next Steps

6.1 Development difficulties and Learnings

Starting with the development work there was problems at every step on the way, starting with learning Node.js and React. A lot of analysis had to be done – how the software should look like and what the software should be capable of. A lot of time was invested in finding the right solution also the process was constantly changing and again analysing the changes took its time which made things difficult from time to time. For example, there were several times situations where the part was almost ready and the solution in the end just did not work out or fit as it should have. The solution had to be thrown into the trash so to speak and only the experience could be taken with it. Using AWS caused the most problems in doing development because of the lack of knowledge, there were a lot of situations where the solution should have worked but the rights were not granted, or the policies were wrong. Fortunately, the AWS documentation is well written for different development languages, which helped to find solutions to the problems. It took a lot of time to implement SendGrid because of the lack of documentation and there were a lot of problems with it, but it was a very big lesson in knowing what to do and how. When SendGrid was replaced by AWS Simple Email Service, its implementation was several times faster, as previous experience already existed. Another similar lesson was when LHV's open banking could be replaced with Nordigen solution. All the parts of this thesis were kind of a challenge because author had not done anything like this before, also had to think a lot about how the data should be stored and what the data flow should be like.

6.2 Risks

One of the most critical risks to a software solution is data breaches. Violation of customer data privacy may result in identity theft and consequent financial loss to customers. External attacks or unintentional leaks can also expose sensitive customer information - financial transactions, balances, bank account numbers. Coincidentally, if the user's account password should be leaked, the high risk is mitigated by the fact that the platform is AISP, which means that it is not possible to make any payments anywhere and the customer will not lose a single amount in such a case. However, this should be avoided

at all costs as sensitive user data is still at risk. That could possibly end up with getting a lawsuit.

Certainly, in the future, absolutely all data must be encrypted, if someone carries out a successful external attack, the data will not be as easily available but additional work must be done to access the original data what is no easy task. Luckily, Amazon offers a lot of different security measures, but the phrase "The application is just as strong as its creator" also applies. Which means that all possibilities are there, but if these possibilities are not used properly there is no use of it.

Definitely a regular penetration test should be applied to understand where the weaknesses of the application are, in order to improve the corresponding places so that the situations mentioned in the first paragraph do not happen. The customer data must be protected.

To mitigate the risks, two-factor authentication should definitely be introduced, so that situations do not occur where the customer's password should leak, and his data can be read and distributed, and unnecessary actions can be performed in the application.

6.3 Further developments

This application has many different directions where to further develop it and which components to implement it. The desire would be to develop a large platform where business owners and accountants have all the necessary functions to do their job more smoothly and, most importantly, save as much time as possible for both parties.

One of the following options that could be good for the user, would be a view where it is seen which documents have been successfully processed and which documents were problematic. Therefore, user has the opportunity to remove the problematic document and, if possible, re-upload it without having several documents and confusion. Machine-learning and Artificial Intelligence could be introduced to further augment document-transaction matching.

Second, make accountants have their own dashboard where they can manage the clients (companies) whose accounts they keep track of so that everything is officially in order as it should be. In this context, it would be good if the accountant could agree on times for

meetings with his client (company) and vice versa. For example, schedule an appointment through google calendar according to whether the other party has free time and the other party can approve or decline accordingly.

A very good feature would be to be able to configure your own email so that all incoming emails are forwarded to the application and the application recognizes which emails are correct and which emails should be processed and which emails are unnecessary and should not be processed. From there, also get the necessary information from the document and add it accordingly to the main journal. This as opposite to forward a particular email to the system (as it is done in the present MVP), which is a personal manual selection and task.

7 Summary

The aim of the dissertation was to develop an application that helps to automate accounting by finding a link between an invoice document and a bank statement, which is a web-based prototype at the MVP level. During the development, various software had to be implemented in order to achieve the desired result.

The created application is able to receive emails sent to a specific email and distinguish the necessary data from it, including the invoice document. The document can be viewed by the user on the website in the form in which it was transmitted, and the necessary information is obtained from the document in order to link it to the bank statement.

During the development, answers have been found to the hypothesis raised at the beginning of the dissertation. It is difficult to say exactly how much documents sent to the system can be automated, because the dissertation writer does not have his own company to test, how much can be automated and how much must be manually confirmed. Considering what security measures need to be put in place, it is certain that the data should be stored in encrypted form.

In order to obtain a bank statement in order to automate as much as possible, a solution was found in the form of open banking, where there is no need to perform manual operations through the bank, make a query and display data in some format and upload it to the platform for automation. This is done automatically, which only requires the user's consent. The bank statement is displayed to the user on the website together with the documents that are or may be related to this transfer.

A connection to the document and the bank statement is automatically found if there are enough common factors that are the basis for automation. If no connection is found, the user is shown possible options for what documents could be in connection with this, where the user can view and confirm with the click of a button and confirm that this particular payment is related to this document.

Account reconciliation has been automated upfront instead of after the bookkeeping process. Doing so results in a great win of time and resources on the accountant's side.

References

- [1] “Reconciliation Definition.” [Online]. Available: <https://www.investopedia.com/terms/r/reconciliation.asp>. [Accessed: 12-May-2021].
- [2] “Reconciling Account - Overview, Process, How It Works.” [Online]. Available: <https://corporatefinanceinstitute.com/resources/knowledge/accounting/reconciling-account/>. [Accessed: 12-May-2021].
- [3] [“Mida peab alustav ettevõtja teadma raamatupidamisest?- Sinufirma.ee.” [Online]. Available: <https://sinufirma.ee/mida-peab-alustav-ettevotja-teadma/>. [Accessed: 05-Oct-2020].
- [4] “Alustava ettevõtte raamatupidamine - kuidas toda algusest peale korras hoida?” [Online]. Available: <https://raamatupidaminevaikeettevottele.ee/artiklid/alustava-ettevotte-raamatupidamine-kuidas-hoida-korda/>. [Accessed: 05-Oct-2020].
- [5] “CostPocket OÜ (14016832) - teatmik.ee.” [Online]. Available: <https://www.teatmik.ee/en/personlegal/14016832-CostPocket-OÜ>. [Accessed: 12-May-2021].
- [6] “The easiest expense management app with OCR - CostPocket.” [Online]. Available: <https://costpocket.com/en>. [Accessed: 15-May-2021].
- [7] “ENVOICE OÜ (12749039) - teatmik.ee.” [Online]. Available: <https://www.teatmik.ee/en/personlegal/12749039-ENVOICE-OÜ>. [Accessed: 15-May-2021].
- [8] “Invoice - Manage your company’s invoices, expenses, & approvals online.” [Online]. Available: <https://invoice.eu/en/>. [Accessed: 15-May-2021].
- [9] “Accounting Software – Do Beautiful Business | Xero UK.” [Online]. Available: <https://www.xero.com/uk/>. [Accessed: 15-May-2021].
- [10] “Bank reconciliation in Xero.” [Online]. Available: <https://central.xero.com/s/article/Bank-reconciliation-in-Xero#:~:text=Bank reconciliation is the process,in Xero on the right>. [Accessed: 15-May-2021].
- [11] “About data extraction in Hubdoc.” [Online]. Available: <https://central.xero.com/s/article/About-data-extraction-in-Hubdoc-US-SG-SAROW#Whydatawontbeextracted>. [Accessed: 15-May-2021].
- [12] [“eRIK - Statistika.” [Online]. Available: https://www2.rik.ee/rikstatfailid/failid/tabel.php?url=21_05tg.html. [Accessed: 18-May-2021].
- [13] “Accounting Act – Riigi Teataja.” [Online]. Available: <https://www.riigiteataja.ee/en/eli/530102013006/consolide>. [Accessed: 18-May-2021].
- [14] “European Parliament adopts European Commission proposal to create safer and more innovative European payments.” [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/en/IP_15_5792. [Accessed: 06-Oct-2020].

- [15] “What Is Open Banking and How Will It Affect You?” [Online]. Available: <https://www.thebalance.com/what-is-open-banking-and-how-will-it-affect-you-4173727#citation-3>. [Accessed: 06-Oct-2020].
- [16] “Open Banking Definition.” [Online]. Available: <https://www.investopedia.com/terms/o/open-banking.asp>. [Accessed: 08-Oct-2020].
- [17] “What is open banking? | Luminor.” [Online]. Available: <https://www.luminor.ee/en/open-banking>. [Accessed: 08-Oct-2020].
- [18] “Delivering on the promises of Open Banking.”
- [19] “Everything you need to know about PSD2 | BBVA.” [Online]. Available: <https://www.bbva.com/en/everything-need-know-psd2/>. [Accessed: 30-Apr-2021].
- [20] “Open Banking: AISPs and PISPs explained.” [Online]. Available: <https://www.finextra.com/blogposting/16647/open-banking-aisps-and-pisps-explained>. [Accessed: 09-Oct-2020].
- [21] “What does AISP & PISP mean? | OpenWrks.” [Online]. Available: <https://www.openwrks.com/what-is-open-banking/aisp-pisp>. [Accessed: 09-Oct-2020].
- [22] “Desktop vs. Web Applications: A Deeper Look and Comparison.” [Online]. Available: <https://www.seguetech.com/desktop-vs-web-applications/>. [Accessed: 14-May-2021].
- [23] “The benefits of using web-based applications.” [Online]. Available: <https://www.geeks.ltd.uk/about-us/blog/details/eQU5Ip/the-benefits-of-using-web-based-applications>. [Accessed: 14-May-2021].
- [24] “About AWS.” [Online]. Available: <https://aws.amazon.com/about-aws/>. [Accessed: 15-May-2021].
- [25] “What is AWS.” [Online]. Available: <https://aws.amazon.com/what-is-aws/>. [Accessed: 15-May-2021].
- [26] “Deploy to AWS Lambda hassle-free.” [Online]. Available: <https://claudiajs.com/claudia.html>. [Accessed: 15-May-2021].
- [27] “Amazon DynamoDB | NoSQL Key-Value Database | Amazon Web Services.” [Online]. Available: <https://aws.amazon.com/dynamodb/>. [Accessed: 15-May-2021].
- [28] “What is Cognito in AWS? - Cloud Academy.” [Online]. Available: <https://cloudacademy.com/blog/what-is-cognito-in-aws/>. [Accessed: 15-May-2021].
- [29] “What is ReactJS: Introduction To React and Its Features.” [Online]. Available: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>. [Accessed: 15-May-2021].
- [30] “Pros and Cons of Angular Development Framework | AltexSoft.” [Online]. Available: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/>. [Accessed: 15-May-2021].
- [31] “What is Angular?: Architecture, Features, and Advantages.” [Online]. Available: <https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular>. [Accessed: 15-May-2021].
- [32] “The Pros and Cons of Vue.js | AltexSoft.” [Online]. Available: <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>. [Accessed: 15-May-2021].
- [33] “What is Vue.js, and Why is it Cool? – Linux Hint.” [Online]. Available: https://linuxhint.com/about_vue_js/. [Accessed: 15-May-2021].

- [34] “The Vue Handbook: a thorough introduction to Vue.js.” [Online]. Available: <https://www.freecodecamp.org/news/the-vue-handbook-a-thorough-introduction-to-vue-js-1e86835d8446/>. [Accessed: 15-May-2021].
- [35] “6 Best Front-end JavaScript Frameworks: Multi-Year Developer Surveys Visualized [2021 Update].” [Online]. Available: <https://jscharting.com/blog/best-javascript-front-end-frameworks-2021/>. [Accessed: 18-May-2021].
- [36] “What is Node.js: A Comprehensive Guide.” [Online]. Available: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs>. [Accessed: 15-May-2021].
- [37] “Amazon Textract | Extract Text & Data | AWS.” [Online]. Available: <https://aws.amazon.com/textract/>. [Accessed: 17-May-2021].
- [38] “What is Amazon Textract and Why is it a Game Changer? | by Webuters Technologies | Medium.” [Online]. Available: <https://medium.com/@webuters/what-is-amazon-textract-and-why-is-it-a-game-changer-77ae08c7125b>. [Accessed: 17-May-2021].
- [39] “Email Delivery Service | SendGrid.” [Online]. Available: <https://sendgrid.com/>. [Accessed: 18-May-2021].

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis¹

I Rasmus Rugam

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis Automation Software for Bookkeepers - Account Reconciliation by Verifying and Matching Corresponding Documents and Bank Statements, supervised by Tarvo Treier
 - 1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
 - 1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

18.05.2021

¹ The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.