

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Kristian Laupa  
175006IDDR

# **Mikroettevõtte finantsaruandluse halduse automatiseerimine**

Diplomitöö

Juhendaja: Lembit Viilup  
PhD

Tallinn 2021

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Kristian Laupa

\_\_\_\_\_

allkiri

17.05.2021

## **Annotatsioon**

Käesoleva töö peamiseks eesmärgiks on luua ettevõttele spetsiifiline ja mugav halduse tarkvara. Töö käigus analüüsiti võimalikke lahendusi vastavalt ettevõtte nõuetele ja kitsendustele. Töö käigus valmis mitu prototüüpi, mille põhjal valmis lõpptoode.

Töö tulemuseks on tarkvaraline lahendus, mis võimaldab ettevõtte töötajatel dokumentide haldamine automatiseerida. Tarkvara lisab töötajatele vajalikud read igasse dokumenti ja liidab kokku kõik eraldi seisvad dokumendid ühtseks dokumendiks.

## **Abstract**

### **Automation of Micro-Enterprise Financial Reporting Management**

The aim of this thesis is to create a software for company specific and easy to use management. During the project different possible solutions were analyzed with given specifications and restrictions from the company. During the project multiple prototypes were done and the final solution based on the previous prototypes.

The result of the project is a software solution, which gives company employees a possibility to automate document management. Necessary lines for the employees are added by the software to each document and all the separate documents are combined into a single document.

## Lühendite ja mõistete sõnastik

|                 |   |
|-----------------|---|
| Automatsioon    | Automatiseerimine on tehnoloogiate loomine ja rakendamine kaupade ja teenuste tootmiseks, inimese minimaalse sekkumisega  |
| Fail            | Arvuti välismällu salvestatav või kopeeritav ja sealt kustutatav kindla nimega andmekogum   |
| Folder          | <i>Folder</i> ehk digitaalne kaust on võrreldav füüsilise kaustaga, millesse salvestatakse dokumente, samas arvutikaustad võimaldavad veel salvestada ka erinevaid tüüpi faile [11] |
| GUI             | <i>Graphical user interface</i> ehk graafiline kasutajaliides on arvutiprogramm, mis võimaldab inimesel suhelda arvutiga sümbolite, visuaalsete metafooride ja osutusseadmete abil  |
| Microsoft       | Tehnoloogiaarendusettevõte, mis arendab, toodab, litsentseerib ja müüb mitmesuguseid IT-seadmeid ja tarkvara  |
| Microsoft Excel | Tabelarvutussüsteem ehk tabelarvutus- ja tabeltöötlusprogramm   |
| OpenPyxl        | Python'i teek, mis võimaldab Microsoft Excel'i faile lugeda ning töödelda [8]   |
| OS module       | Python moodul, mis pakub funktsioone kataloogi(kausta) loomiseks ja eemaldamiseks, selle sisu kuvamiseks ja muutmiseks ning tuvastamiseks [5]                                       |
| Pakett          | <i>Package</i> ehk pakett on oma olemuselt Python moodulite kogum, sisaldades erinevaid Python script'e   |
| Pandas          | Python'i pakett, mida kõige enam kasutatakse andmeteaduse / andmete analüüsi ja masinõppe ülesannete jaoks [7]  |
| Python3         | Intepreteritud kõrgema taseme programmeerimiskeel [2]   |

|         |  |
|---------|--|
| Skript  | Skript on sisuliselt programm ehk masinale ette antavate käskude jada. [13]  |
| Teek    | <i>Library</i> ehk teek on oma olemuselt on funktsioonide, makrode, klasside, moodulite vms komponentide kogum, mida saab programmis vajadust mööda kasutada |
| Tkinter | Python standard graafilise liidese teek, mis pakub koos Python3'ega kiiret ja lihtsat viisi graafilise kasutajaliidese loomist [9]                           |
| Windows | Enim tuntud ja kasutatav arvutite operatsioonisüsteem  |
| .xlsx   | Dokumendifail, mis on Microsoft Exceli standardvormingus failitüüp   |

# Sisukord

|   |           |
|---|-----------|
| <b>Mõisted</b>  | <b>5</b>  |
| <b>Sissejuhatus</b>   | <b>9</b>  |
| <b>1 Ülesande püstitus</b>                                      | <b>11</b> |
| <b>2 Analüüs</b>  | <b>12</b> |
| 2.1 Funktsionaalsed nõuded loodavale lahendusele . . . . .      | 12        |
| 2.2 Mittefunktsionaalsed nõuded loodavale lahendusele . . . . . | 13        |
| 2.3 Skoop . . . . .   | 13        |
| 2.4 Valitud programmeerimiskeelte kasutuselevõtt . . . . .      | 15        |
| 2.4.1 Tarkvara programmeerimiskeel . . . . .                    | 15        |
| 2.4.2 Kasutajaliidese programmeerimiskeel . . . . .             | 15        |
| 2.5 Alternatiivsed lahendused . . . . .                         | 16        |
| 2.6 Sarnaste lahenduste võrdlus . . . . .                       | 16        |
| 2.7 Nõutud dokumendi nime veergu lisamise loogika . . . . .     | 18        |
| 2.8 Nõutud dokumentide liidestamise loogika . . . . .           | 18        |
| 2.9 Nõutud kasutajaliidese loogika . . . . .                    | 19        |
| 2.10 Töö planeerimine . . . . .                                 | 20        |
| <b>3 Automaatika arendus</b>                                    | <b>22</b> |
| 3.1 Dokumendi nime veergu lisamise tarkvara loomine . . . . .   | 22        |
| 3.2 Dokumentide liidestamise tarkvara loomine . . . . .         | 23        |
| 3.3 Graafilise kasutajaliidese tarkvara loomine . . . . .       | 23        |
| <b>4 Sihtgrupp</b>  | <b>25</b> |
| <b>5 Tulemuste hinnang</b>                                      | <b>26</b> |
| <b>Kokkuvõte</b>  | <b>27</b> |
| <b>Kasutatud kirjandus</b>                                      | <b>28</b> |
| <b>Lisad 1 Lihtlitsents</b>                                     | <b>30</b> |
| <b>Lisad 2 Dokumendi nime lisamise loogika kood</b>             | <b>31</b> |

|  |           |
|--|-----------|
| <b>Lisad 3 Dokumentide liidestamise loogika kood</b>   | <b>32</b> |
| <b>Lisad 4 Graafilise kasutajaliidese loogika kood</b> | <b>33</b> |



## Sissejuhatus

Antud diplomitöö raames töötatakse välja mugav lahendus auditeerimis- ja raamatupidamisteenusete osutamiseks, võttes abiks manuaalselt tehtava töö automatiseerimine. Ettevõttes teletakse peamiselt raamatupidamise- ja auditeerimisteenusete pakkumisega. Vastavaid teenuseid pakutakse juba vähemalt 120 mikro- ja väikeettevõttele, kes on oma raamatupidamise kajastamise ja auditeerimise mikroettevõtte hoolde usaldanud. Eestis, kus ettevõtte pesitseb ja teenuseid pakub on tööjõu olemasolu limiteeritud. See tähendab, et ettevõtte jätkusuutlikkuse jaoks on hädasti vaja mingi osa töö automatsiooni. Sellest tulenevalt on vaja ettevõtte spetsiifilist dokumentide majandamise automaatikat, mis vähendaks ebaefektiivse ja korduva töö tegemist.

Dokumentide automaatika teenust on võimalik alati sisse osta ning ostetud teenus on tihti väga hästi tehtud ja lihtsasti kasutatav, aga finantsiliselt kulukas. Vaatamata väga heast teenusest, firma peamiseks eesmärgiks ei ole luua täislahendust, vaid automatiseerida kindlaid vaheetappe. Automaatika peamiseks eesmärgiks on manuaalse ja korduva töö automatiseerimine vaheetappides.

Lõputöö lõppeesmärgiks on luua finantsaruandluse halduse tarkvaraline lahendus, mis automatiseerib raamatupidajate ja auditeerijate aega nõudvat ja korduvat manuaalset tehtavat tööd. Töö raames seadis töö autor endale 3 alameesmärgi. Esiteks, luua dokumentide käsitlemisel automaatika, mis võtab dokumendi nime ja lisab võetud väärtuse määratud kohta dokumendis. Lisaks luua loogika, mis tagab automaatika töökindluse.

Teiseks, luua mitmete failide kokkütõstmise loogika, mis tõstab kokku eraldi seisvate failide sisud kokku üheks suureks failiks. Kolmandaks, leida viis kuidas firma töötajatel oleks antud tarkvara kõige lihtsam kasutada, arvestades tavakasutaja poolset lähenemist.

Esimese eesmärgi saavutamiseks uuriti võimalikke olemasolevaid lahendusi. Asuti prototüüpima esimesi lihtsustatud versioone. Seejärel uuriti prototüüpide nõrkusi ning leiti neile töökindlamad alternatiivsed lahendused ning tehti uued katsetused. Lisaks analüüsiti võimalikke lahendusi, et luua võimalikult stabiilne ja terviklik süsteem. Selle eesmärgi nimel loodi koostöö, suhtlus dokumendi ja programmeerimiskeele vahel.

Teise eesmärgi saavutamiseks kasutati sarnaselt esimese eesmärgi lähenemisviise. Tarkvarale on ette määratud asukoht, kus kontrollitakse mis tüüpi faile esineb kaustas. See järel tarkvara filtreerib temale määratud failid välja, võttes arvesse ainult kindlat failitüüpi. Viimasena luuakse uus tühi dokument, kuhu kopeeritakse kokku eelnevalt välja filtreeritud failide sisud.

Kolmanda eesmärgi saavutamiseks võeti arvesse firma töötajate soovid, luua piisavalt lihtne GUI ehk kasutajaliides ning hoida eraldi funktsioonina esimeses ja teises kirjeldatud eesmärgid.

Esimeses peatükis esitatakse töös käsitletava ülesande püstitus ning sellele järgneb analüüs. Analüüsis kirjeldatakse funktsionaalseid ja mitte funktsionaalseid nõudeid loodavale lahendusele, räägitakse töö skoobist, kirjeldatakse tehtuid valikuid, töös kasutatud loogikaid, alternatiivseid valikuid, sarnaste lahenduste võrdlusi ning töö planeerimisest. Kolmandas peatükis kirjeldatakse tarkvaralist ülesehitust ning kasutajaliidese loomisest. Neljandas peatükis kirjutatakse lähemalt, kes võiksid veel antud sihtgruppi kuuluda ja diplomandi lahendusest kasu lõigata. Viiendas peatükis kirjeldatakse tehtud töö tulemust millele järgneb töö kokkuvõte.

Töö on vormistatud dokumentide ettevalmistussüsteemiga  $\text{\LaTeX}$ [15].

## 1 Ülesande püstitus

Mikroettevõttes, kus diplomand abi osutab, pakutakse raamatupidamis- ja auditeerimisteenust. Teenuseid pakutakse nii Eestis kui ka välismaal. Sellest tulenevalt peab olema pakutud teenus kiire ja kvaliteetne. Lisaks on oluline hoida oma mainet positiivsena, et mikroettevõttel oleks võimalik kasvada.

Efektiivsemalt tööaja kasutamist ja vajalike töötundide kokku hoidmiseks on ettevõttele vajalik manuaalse ja korduva töö automatsiooni, mis:

- võimaldab klientide saadetud dokumentidele vajalike ridade juurde lisamist;
- võimaldab klientide erinevad dokumendid kokku tõsta üheks suureks dokumendiks;
- on firmale pikemas perioodis alati kasutatav ja muudetav vajadusele;
- on tavakasutajatele lihtne kasutada.

Selliseid nõudeid täidab finantsaruandluse halduse automatsioon. Sellest tulenevalt otsustaski ettevõtte, et neil on vajadus automatsiooni järele. Ettevõtte seisukohaks on automatsiooni võimalikult kiire tööle saamine ja kasutamine. Sellest tulenevalt tuleb teha diplomandil tema jaoks kõige lihtsam ning nõuetele vastav finantsaruandluse halduse automatsiooni tarkvaraline lahendus.

Selle lõputöö käigus arendatakse finantsaruandluse halduse automatsioon ehk tarkvaraline lahendus ning lihtsasti kasutatav kasutajaliides, mis täidavad eelnevalt mainitud nõudeid.

## 2 Analüüs

Vastavas peatükis „Analüüs“ kirjeldatakse eraldi punktidenä finantsaruandluse halduse automatsiooni funktsionaalseid ja mittefunktsionaalseid nõudeid, olemasolevate lahenduste ülevaade, kirjeldatakse skoopt, põhjendatakse valitud programmeerimiskeele kasutuselevõttu ning alternatiivsed lahendused. Samuti tuuakse välja sarnaste lahenduste võrdlusi diplomandi loodud lahendusega, kirjeldatakse ettevõtte poolt nõutud dokumentide manageerimise loogikaid ning nõutud loogikast loodavale kasutajaliidesele, viimase punktina selgitatakse lähemalt üldisest töö planeerimisest.

### 2.1 Funktsionaalsed nõuded loodavale lahendusele

Funktsionaalseid nõudeid loodavale lahendusele vaadeldakse firma seisukohast.

Finantsaruandluse halduse automatsiooni funktsionaalne lahendus peab täitma järgmiseid punkte:

- automatsioon peab töötama erinevates Windows operatsioonsüsteemide töökohtades;
- automatsioonile peab olema määratud kindel *folder* ehk kaust töötaja arvutis;
- automatsioon peab suutma eristada Microsoft Excel .xlsx faile teistest kaustas olevatest failidest;
- automatsioon peab suutma töödelda erinevaid Microsoft Excel .xlsx faile;
- automatsioon peab suutma kopeerida faili nime;
- automatsioon peab suutma kopeeritud faili nime sisestama määratud veergu vastavas dokumendis;
- automatsioon peab suutma kokku tõsta ühtseks failiks mitmete Microsoft Excel .xlsx failide sisu;
- kasutajaliides peab koosnema kahest erinevalt seisvast nupust funktsiooni rakendamiseks;
- kasutajaliides peaks teavitama, kui funktsiooni rakendamise nuppu on vajutatud;
- vältida mitmete erinevate rakenduste installeerimist Windows operatsioonsüsteemide töökohtades ehk töötajate arvutis;
- tervik lahendus peaks olema loodav samas programmeerimiskeeles.

Tuginedes firma funktsionaalsetele nõuetele, on diplomandi poolne funktsionaalne lahendus toetunud vastavale loetelule.

## **2.2 Mittefunktsionaalsed nõuded loodavale lahendusele**

Mittefunktsionaalseid nõudeid loodavale lahendusele vaadeldakse samuti firma seisukohast.

Finantsaruandluse halduse automatsiooni mittefunktsionaalne lahendus peab täitma järgmisi punkte:

- tavakasutajate jaoks peab kasutajaliides olema lihtsasti kasutatav;
- tavakasutajate nõudmisel ja vajadusel peab olema kohalikul administraatoril võimalus lahendust muuta;
- tavakasutajate jaoks peab kasutajaliides olema loogiliselt arusaadav;
- automatsioonil peavad puuduma programmi hoolduse ja kasutamise kulud;
- automatsioon peab olema kiire ja efektiivne;
- automatsiooni kasutajaliidese välimus ei mängi rolli;
- automatsiooni kasutajatel ei teki kohustusi lahenduse kasutamisel;
- automatsioon peab olema pikaajaliselt kasutatav.

Tuginedes firma mittefunktsionaalsetele nõuetele, on diplomandi poolne mittefunktsionaalne lahendus toetunud vastavale loetelule.

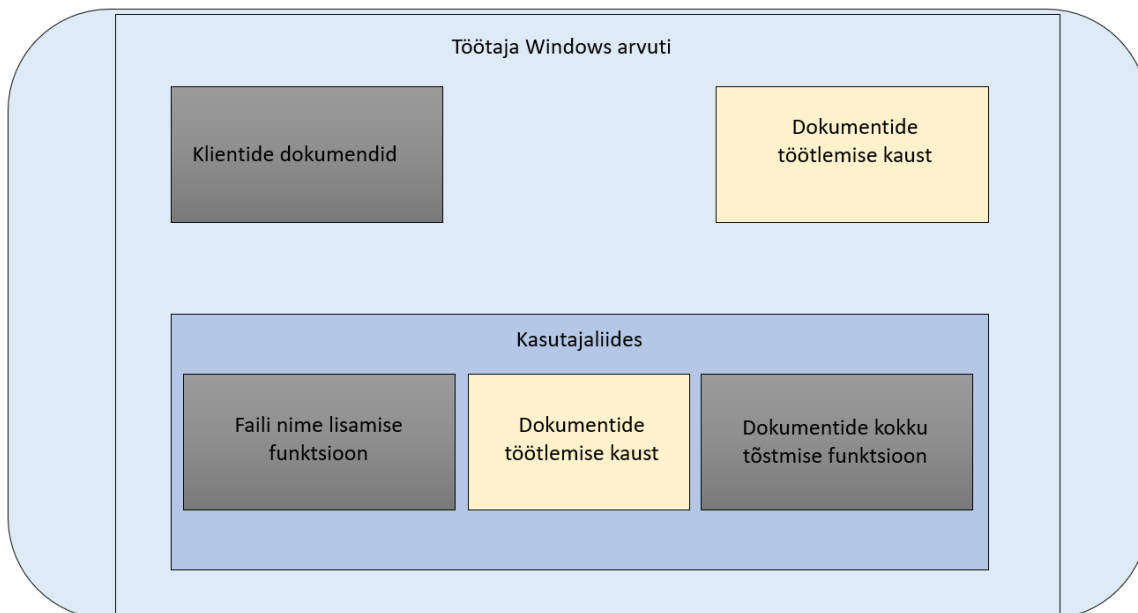
## **2.3 Skoop**

Mikroettevõtte finantsaruandluse halduse automatsiooni arenduse välja töötamine toimus paralleelselt mikroettevõtte töötajatega suheldes. Mikroettevõtte töötajad määrasid nii kasutajaliidese kui ka tarkvaralised piirangud ning nõuded. Kogu lahendus on välja töötatud diplomandi poolt, arvestades ettevõtte poolseid nõudeid (vt. peatükk 2.1) ja (vt. peatükk 2.2).

Finantsaruandluse halduse automatsiooni terviklik ülesehitus koosnes:

- Täidetavate nõuete analüüsist ja nende määramisest;
- Kasutajaliidese disain:
  - tavakasutaja jaoks võimalikult lihtne kasutada;

- tavakasutaja jaoks loogiline ülesehitus;
  - Python3 baasil töötav;
  - funktsioonid määrata erinevate nuppudena kasutajaliideses;
  - vältida mitmete rakenduste installeerimist töötajate töökohtadesse.
- Tarkvara arendus:
    - Python3 baasil töötav;
    - erinevatele dokumentidele faili nime juurde lisamise loogika;
    - erinevate dokumentide kokku tõstmise loogika üheks dokumendiks;
    - kasutajaliidese ja tarkvara vahelise suhtluse loogika.



*Joonis 1: Finantsaruandluse halduse automatsiooni blokkdiagramm.*

Eelnevalt nimetatud tehtud töödest jääb lõputöö skooopi vähesemal määral kasutajaliidese disain ja rohkemal määral tarkvara arendus. Joonisel 1 on kirjeldatud terve finantsaruandluse halduse automatsiooni ülesehitus ehk skoop.

Joonisel on kujutatud reaalset olukorda, kus töötaja on oma Windows'i arvutis avanud diplomandi poolt loodud kasutajaliidese, mis avaneb töötaja arvutis eraldi rakendusena. Kasutajaliidese on kaks eraldi seisvat nuppu, faili nime lisamise funktsiooni nupp ning dokumentide kokku tõstmise funktsiooni nupp. Mõlemale nupule on määratud töötlemise kindlas kaustas olevaid dokumente töötaja arvutis. Eelnevalt tõstab töötaja vajalikud dokumendid dokumentide

töötlemise kausta ning seejärel kasutab töötaja diplomandi loodud kasutajaliidese funktsioone. Vastavalt vajadusele saab töötaja iseseisvalt otsustada, milline funktsioon parasjagu vajalik on.

## **2.4 Valitud programmeerimiskeelte kasutuselevõtt**

Ettevõttes on välja kujunenud ja loodud erinevaid Python skripte, mille eesmärk on samuti raamatupidajate ja auditeerijate töö lihtsustamine. Vastavad skriptid on loodud seelses ettevõttes töötavana IT-tugi poolt. Suurt rolli välja valitud programmeerimiskeele kasutuselevõttes mängib ka ettevõtte IT-tugi eelnev kogemuspagas Python'is.

### **2.4.1 Tarkvara programmeerimiskeel**

Eelnevalt sai mainitud programmeerimiskeele kiiruse tähtsus ja töötavus Microsoft Windows operatsioonisüsteemi masinates (vt. peatükk 2.1) ja (vt. peatükk 2.2). Sellest tulenevalt tekkisid kitsendused programmeerimiskeele valikul. Valituks sai programmeerimiskeel Python [12].

Loodav tarkvara peab olema kiire ning olema efektiivne samaajaliselt mitmete dokumentide manageerimisel. Tarkvara peab olema suuteline mõistma Microsoft Excel (.xlsx) tüüpi faile. Lisaks peab oskama vastavaid faile muuta ning töödelda vastavalt vajadusele. Suureks boonuseks on programmeerimiskeeles olemasolevaid Python'i paketid ja teegid, mis võimaldavad Microsoft Excel failide manageerimist. Lisaks peab tarkvara võimaldama töötamist mitmete Microsoft Excel failidega samaaegselt.

Lisaks on Python väga universaalne programmeerimiskeel, mis on äärmiselt lihtsalt ja kiirelt kirjutatav ning käivitatav. Samuti leidub Python'i kohta väga palju erinevaid dokumentatsioone ja õpetusi. Python'i eeliseks on diplomandi eelnev kogemuste pagas.

Tööd alustati Python programmeerimiskeeles. Töö käigus selgus, et teatud vajaminevad paketid ja teegid ei toetanud vanemat versiooni Python'ist, kuid need olid olemas uuemas Python3'es. Lisaks ei erine väga palju Python3 Python'ist ja selle õpiköver pole suur. Sellest tulenevalt on töö käigus loodud kood kirjutatud Python3's.

### **2.4.2 Kasutajaliidese programmeerimiskeel**

Eelnevalt sai mainitud kasutajaliidese tähtsust loogikas, kui ka lihtsuse ülesehitusel ning vältivust erinevate rakenduste installeerimist töötajate Windows operatsioonisüsteemide töökohtadesse (vt. peatükk 2.1) ja (vt. peatükk 2.2). Sellest tulenevalt tekkisid kitsendused kasutajaliidese programmeerimiskeele valikul. Valituks sai Python3'e juba eelnevalt sisse ehitatud ja installeeritud Tkinter [9].

Loodav graafilise kasutajaliidese tarkvara peab olema võimalikult lihtsasti arusaadav ja mõis-

tetav, kuna ettevõtte IT-tugi peab olema võimeline töötajate nõudmisel muudatusi sisse viima. Tkinter ehk graafilise kasutajaliidese ülesehitus ning visuaalsus peab olema igale-ühele üheselt mõistetav. Samuti peab graafiline kasutajaliides suuteline olema eelnevalt loodud tarkvara funktsioone käivitada ja seda ainuüksi kasutajaliideses olevate nuppude peale klikkamisega ehk hiireklahvi vajutamisega.

Töö alustamise hetkel polnud diplomand kokku puutunud Python3 Tkinter keelega ning otsustati väiksema õpikõvera kasuks. Vähesee kogemuse ja projekti kiiruse tõttu luuakse võimalikult lihtne ja loogilise ülesehitusega graafiline kasutajaliides. Võttes arvesse kõiki nõutud nõudeid loodavale kasutajaliidese lahendusele.

## **2.5 Alternatiivsed lahendused**

Kuna sarnaseid automatsioone müüakse või pakutakse teenustena, siis koheselt olemasolevat valmis lahendust ainult tarkvaraliselt olemas pole. Lisaks nõuaks sellise tarkvara automatsiooni eksisteerimine kindlaid juhiseid ja vajadusi vastava ettevõtte töötajate poolt. Töös käsitletud finantsaruandluse halduse automatsioon on lihtsustatud ning pole standardne lahendus.

Olemasolevad finantsaruandluse halduse automatsioonid varieeruvad võimaluste, lisavõimaluste ja hinna poolest. Enamus raamatupidamistarkvarad pakuvad küll sarnaseid teenuseid, küll aga mitte täpselt sellist loogikat ja ülesehitust, mida ettevõttel just vastava ülesande täitmiseks vaja läheb.

Ettevõtte kasutab Eestis laialdaselt levinud raamatupidamistarkvara Standard Books [3]. Sellest raamatupidamistarkvarast saab võtta eeskujuga. Läbi vastava tarkvara saadavad ettevõtte kliendid raamatupidajatele ning auditeerijatele finantsaruandeid. Nähes, millises vormingus finantsaruanded ettevõtte töötajateni jõuab, saab diplomandi loodavas automaatikas toetuda ülesehituse vormingus, millisena Microsoft Excel .xlsx faile saadetakse. Teades, milline näeb välja saadetud finantsaruandluse dokument, saab luua halduse automaatika, mis teeb ära vajalikud protseduurid dokumentidega, enne kui dokumendid on valmis kasutamiseks Microsoft Excel Pivot Table's ehk liigendtabelis [16].

## **2.6 Sarnaste lahenduste võrdlus**

Olemasolevate sarnaste lahenduste ülevaateid vaadeldakse Microsoft Excel'i [4] poolt pakutavaid lahendusi ja võrreldakse diplomandi poolt välja töötatud lahendusega. Tuuakse välja sarnaste lahenduste miinused ning diplomandi loodud automatsiooni plussid.

Microsoft Excel'i pakutavad lahenduste miinused:



- puudub täielik lahendus, mida mikroettevõtte töötajad vajavad;
- lisa funktsioonid on tasulised;
- tavakasutaja jaoks raskesti kordineeriv ja mõistetav;
- uuemad funktsioonid vajavad Microsoft 365 olemasolu ja esmaklassilist versiooni;
- aeglane, kui on vajadus mitmete suuremate failide kokku tõstmist;
- sõltub olemas olevast Excel'i versioonist;
- töötajate sõnul on lihtsam käsitsi andmeid kopeerida, kui kasutada vastavaid Microsoft Excel'i poolt pakutavaid funktsioone;
- pidev manuaalne dokumentide käsitlemine.

Diplomandi pakutava lahenduse plussid:

- diplomandi välja töötatud lahenduses puuduvad programmi hoolduse ja kasutamise kulud;
- on tavakasutaja jaoks lihtne ja loogiline kasutada;
- lahendus teeb ära just niipalju, kui nõutud;
- ei sõltu olemas olevatest Excel'i versioonidest;
- töötab kiiresti ja efektiivselt ka suuremate failide töötlemisega;
- kasutajatel ei teki kohustusi lahenduse kasutamisel;
- on pikaajaliselt kasutatav;
- tavakasutajate nõudmisel ja vajadusel on kohalikul administraatoril võimalus lahendust muuta.

Võrreldes Microsoft Excel'i pakutavaid lahenduste miinuseid ja diplomandi pakutava lahenduse plussid on selgelt arusaadav, et mikroettevõttel on hädavajalik võtta kasutusse diplomandi poolt välja töötatud lahendust.

Sellest tulenevalt hoiab ettevõtte kokku vajalike töötunde, tagab tööliste rahulolu ning võimaldab töötajatel efektiivsemalt oma tööaega kasutada.

## 2.7 Nõutud dokumendi nime veergu lisamise loogika

Finantsaruandluse halduse automatsiooni üheks eesmärgiks on ettevõtte klientide poolt saadetud ja ette antud dokumentide nimedega manageerimine. Klientide poolt saadetud dokumentide nimi koosneb osaliselt kuupäevast, millest omakorda saab tuletada dokumendis sisalduva informatsiooni loodud tähtaega. Vastav info on äärmiselt vajalik ning kasulik ettevõtte töötajate ülesannete täitmisel.

Arvestades, et dokumendi nimest võetav kuupäev on alati korrektne, on võimalik luua selle kasutamiseks automatsioon. Sellest tulenevalt saab automatsiooniga kopeerida dokumendi nime ning kasutada seda kliendi poolt saadetud dokumendis. Olles kopeerinud dokumendi nime saab automatsiooniga luua uue veeru dokumendis ja nimetada selle päisrida kui „Kuupäev“. Järgmise sammuna lisab automatsioon eelnevalt kopeeritud dokumendi nime määratud veergu ning võrdsustab väärtuse dokumendis olevate ridade arvuga.

Selle osa automatsiooni kokkuvõtvaks ülesandeks oli läbi käia kõik klientide poolt saadetud dokumendid ning manageerida neid täpselt nii nagu sai eelnevalt mainitud. Järgnevalt on kujutletud tervik protsess läbi graafilise joonise (vt. joonis 2).



Joonis 2: Dokumendi nime veergu lisamise loogika tegevus kujutletud graafiliselt.

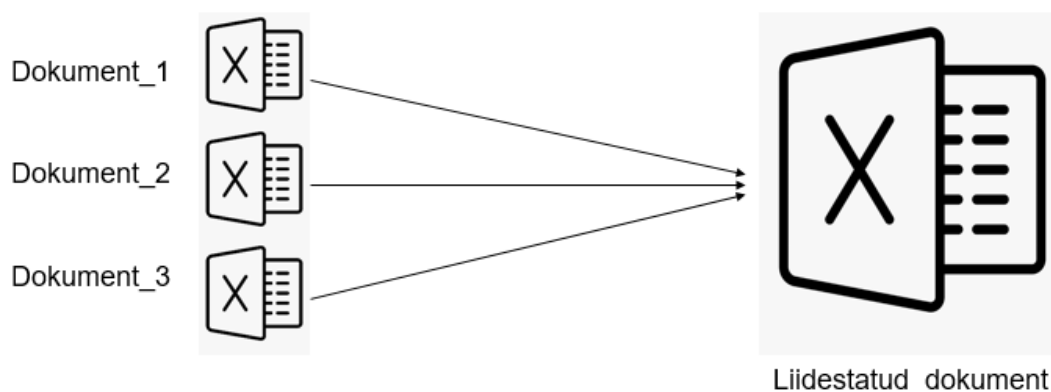
## 2.8 Nõutud dokumentide liidestamise loogika

Finantsaruandluse halduse automatsiooni teiseks tähtsaks eesmärgiks on ettevõtte klientide poolt saadetud eraldi seisvate dokumentide sisu liidestamine üheks terviklikuks dokumendiks. Ettevõtte kliendid saavad vastavaid dokumente iga kuu kohta, mis on eraldatud eraldi dokumentidena iga päeva kohta kuus. Samuti tuleks vastavate dokumentidega eelnevalt läbi viia protsess dokumendi nime veergu lisamisega (vt. peatükk 2.7) ja alles siis viia läbi dokumentide sisude liidestamine. Vastava sisude kokku liidestamine on äärmiselt vajalik ettevõtte töötajate ülesannete täitmisel.

Arvestades, et eraldi seisvate dokumentide sisuline ülesehitus on alati sarnane, saab luua

nende liidestamiseks automatsiooni. Liidestamise automaatika peaks ükshaaval kõik kliendi poolt saadetud dokumentide sisud enda jaoks kopeerima. Kui vajalikud dokumendid on automatsiooni poolt läbi töötatud, on järgmiseks sammuks uue dokumendi loomine, millele on määratud nimi ning dokumendi failitüüp peab kattuma eelmiste eraldi seisvate dokumentide tüübiga. Olles loonud uue tühja dokumendi, kopeeritakse eelnevalt salvestatud dokumentide sisud uute dokumenti ümber, sisaldades terve ühe kuu informatsiooni.

Selle osa automatsiooni kokkuvõtvaks ülesandeks oli klientide poolt saadetud dokumentide sisude liidestamine kokku uute samasse failitüüpi dokumenti. Järgnevalt on kujutletud tervik protsess läbi graafilise joonise (vt. joonis 3).



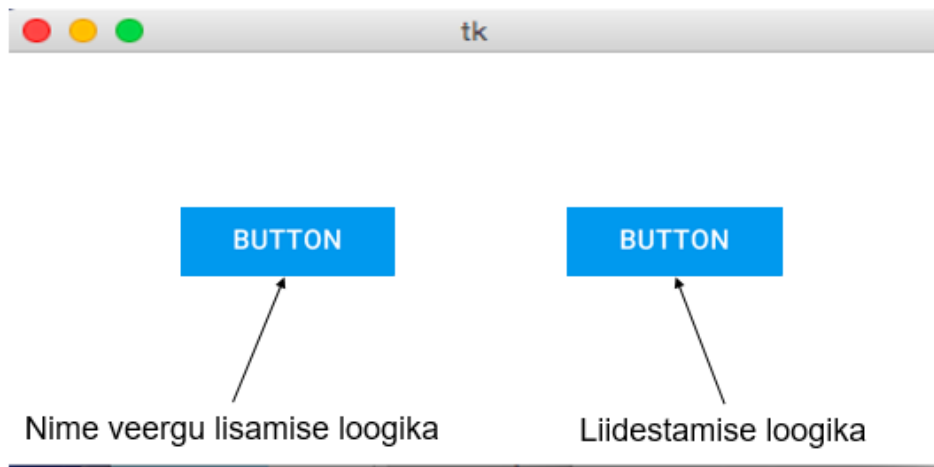
Joonis 3: Dokumentide liidestamise saavutamise loogika kujutletud graafiliselt.

## 2.9 Nõutud kasutajaliidese loogika

Finantsaruandluse halduse automatsiooni kolmas ehk viimane eesmärk on vajadus luua võimalus ettevõtte töötajatel kasutada dokumendi nime veergu lisamise loogikat (vt. peatükk 2.7) ja dokumentide liidestamise loogikat (vt. peatükk 2.8) läbi graafilise kasutajaliidese. Kasutajaliidese vajalikus tuleneb ettevõtte töötajate teadmatuses, kuidas kasutada ning käivitada arvuti operatsioonisüsteemides skripte.

Teades, et graafiline kasutajaliides peab võimaldama käivitada kahte eraldi seisvat loogikat on selge, et kasutajaliides peab minimaalselt koosnema kahest erinevast nupust. Üks nupp on määratud dokumendi nime veergu lisamise loogika jaoks ning teine nupp dokumentide liidestamise loogika jaoks. Käivitades graafilise kasutajaliidese peab tekkima kaks erinevat nuppu, millele on võimalik arvuti hiirega peale klikata. Klõkkides kummalegi nupule, peab käivituma vastav loogika.

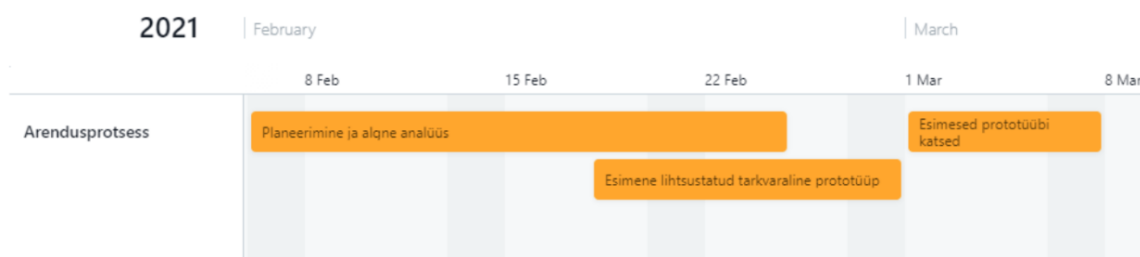
Selle osa automatsiooni kokkuvõtvaks ülesandeks oli kokku tuua kahe erineva loogika võimalused ning neid käivitada läbi graafilise kasutajaliidese. Järgnevalt on kujutletud tervik protsess läbi graafilise joonise (vt. joonis 4).



Joonis 4: Kasutajaliidese saavutamise loogika kujutatud graafiliselt.

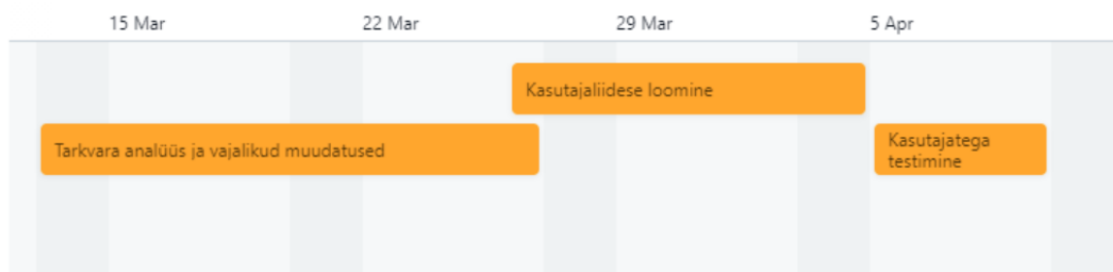
## 2.10 Töö planeerimine

Töö ja planeering algas veebruaris 2021. Alustuseks uuriti mis on tarkvaralised võimalused ja



Joonis 5: Tegevuskava esimene pool.

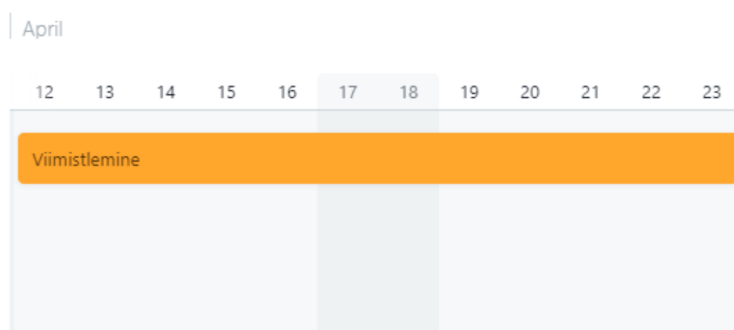
kuidas peab finantsaruandluse halduse automatsioon töötama (vt. joonis 5). Samuti kohtuti ning arutleti ettevõtte töötajatega, mis on nende poolsed ootused ja kuidas võiks üldine lahendus välja näha.



Joonis 6: Tegevuskava teine pool.

Esimene prototüüp valmis 2021 veebruari lõpp. 2021 märtsi esimesel nädalal toimus olemasoleva prototüübiga esimesed tarkvara testimised (vt. joonis 5). Peale õnnestunud testimisi, viidi läbi järjekordne tarkvara analüüsimine ja vajalike muudatuste sisse viimine märtsi keskel

2021 (vt. joonis 6). Märtsi teine pool ja aprilli esimene nädal kulus kasutajaliidese loomisele ja tarkvaraga ühendamiseks. Aprilli teisel nädalal alustati tulevaste automatsiooni kasutajatega tehtud töö testimist (vt. joonis 6). Aprilli keskel algas terviklahenduse viimistlemine. Viimane viimistlus toimus osaliselt veel aprilli lõpus 2021 (vt. joonis 7).



*Joonis 7: Tegevuskava kolmas pool.*

Kõige intensiivsem arendus toimus veebruari keskelt aprilli lõpuni 2021, mil valmis peamine osa terviklahendusest. Edaspidi on lisatud väiksemaid veaparandusi ning viidud sisse kliendi poolseid soove.

### 3 Automaatika arendus

Selles peatükis kirjeldatakse etappide kaupa, kuidas finantsaruandluse halduse automatsiooni arendati, arvestades firma poolt määratud nõudeid ja vajadusi (vt. peatükk 2.1 ja 2.2). Peatükk jaguneb kolmeks erinevaks etapiks, esimene etapp kirjeldab dokumentidele vajaliku info juurde kirjutamise tarkvara loomisest, teine etapp kirjeldab dokumentide liidestamise tarkvara loomisest ning kolmas ehk viimane etapp kirjeldab graafilise kasutajaliidese tarkvara loomisest.

#### 3.1 Dokumendi nime veergu lisamise tarkvara loomine

Esimeseks eesmärgiks antud töös on luua dokumentidele autonoomne nime lisamise tarkvaraline funktsioon. Tarkvara peamiseks ülesandeks on dokumendi nimi kopeerida ning lisada see määratud väljadele dokumendis. Dokumendi nime veergu lisamise loogikas tuli arvestada ette antud ettevõtte poolsete nõuetega (vt. peatükk 2.1 ja 2.2). Tarkvara funktsiooni loomisel võeti eeskuju peatükk 12, raamatust „*Automate the Boring Stuff with Python, 2nd Edition*“ [14].

Autonoomse nime lisamise tarkvara loomisel kasutatakse Python'i poolt pakutavat pakendit ehk *package*, mis oma olemuselt on Python'i moodulite kogum, sisaldades erinevaid Python'i *script'e*. Lisaks kasutatakse veel Python'i teeki, mis oma olemuselt on funktsioonide, makrode, klasside, moodulite või muude selliste komponentide kogu, mida saab programmis vajadust mööda kasutada.

Tarkvaralises lahenduses kasutati Python OS mooduleid [5] ja OpenPyxl teeki [8]. Python OS mooduleid kasutati dokumentide lokaliseerimiseks määratud kaustas ehk folderis. Python OpenPyxl teeki kasutati .xlsx faili tüüpide manageerimisel. Moodulite ja teeki koostööl on võimalik luua lahendus, kus avatakse üks haaval .xlsx failitüüpi dokument ning tehakse määratud tegevused. Esmalt leitakse kõik dokumendid vastavalt dokumendi tüübile ning manageeritakse üks haaval läbi. Manageerimisel liigutakse kindlale leheküljele ehk *sheet'*ile dokumendis, minnakse edasi määratud veerule samas dokumendis ja määratakse veeru päiserea pealkirjaks "Kuupaev". Viimaks kontrollitakse ridade arvu suurust, alustades teisest reast, jättes välja kõikide veergude päiseread. Sellele järgneb dokumendi nime kopeerimine veergu "Kuupaev", alustades teisest reast kuni lehekülje maksimum ridade arvuni ning tehtud muudatused salvestatakse originaal ehk samas dokumendis (vt. Lisad 2).

Vastavate pakettide ja teekide abil sai tehtud lahendus teoks. Tehtud lahendus rahuldab kõiki ettevõtte töötajate vajadusi ja nende poolseid ootusi.

## 3.2 Dokumentide liidestamise tarkvara loomine

Teiseks eesmärgiks antud töös on luua dokumentide autonoomne liidestamise tarkvaraline funktsioon. Tarkvara peamiseks ülesandeks on määratud kaustas dokumentide sisude kokku liidestamine üheks dokumendiks. Dokumentide liidestamise tarkvaras tuli arvestada ette antud ettevõtte poolsete nõuetega (vt. peatükk 2.1 ja 2.2). Tarkvara loomisel võeti eeskujuna Python Pandas kasutusjuhendist ja näidetest [6].

Autonoomse dokumentide liidestamise tarkvara loomisel kasutatakse Python'i poolt pakutavat pakendit ehk *package*, mis oma olemuselt on Python'i moodulite kogum, sisaldades erinevaid Python'i *script'e*. Lisaks kasutatakse veel Python'i teeki, mis oma olemuselt on funktsioonide, makrode, klasside, moodulite või muude selliste komponentide kogu, mida saab programmis vajadust mööda kasutada.

Tarkvaralises lahenduses kasutatakse Python OS mooduleid [5] ja Python Pandas teeki [7]. Python OS mooduleid kasutatakse dokumentide lokaliseerimiseks määratud kaustas. Python Pandas teeki kasutati .xlsx dokumentide sisu liidestamiseks. Mooduli ja teeki koostööl on võimalik luua lahendus, kus kopeeritakse kõikide dokumentide sisud ning liidestatakse kokku uues dokumendis. Esmalt leitakse ja filtreeritakse kõik dokumendid vastavalt dokumendi tüübile välja. Järgnevalt luuakse samat tüüpi uus ja tühi dokument, kuhu kopeeritakse kokku filtreerimise tulemusena leitud dokumendid ja nende sisud. Kui eraldi seisvate dokumentide sisud on uute dokumenti ümber kopeeritud, määratakse dokumendile nimi ning see salvestatakse (vt. Lisad 3).

Vastavate pakettide ja teekide abil sai tehtud lahendus teoks. Tehtud lahendus rahuldab kõiki ettevõtte töötajate vajadusi ja nende poolseid ootusi.

## 3.3 Graafilise kasutajaliidese tarkvara loomine

Kolmandaks ehk viimaseks eesmärgiks antud töös on luua graafilise kasutajaliidese tarkvaraline funktsioon. Tarkvara peamiseks ülesandeks on kokku tuua dokumendi nime veergu lisamise tarkvara funktsioon (vt. peatükk 3.1) ja dokumentide liidestamise tarkvara funktsioon (vt. peatükk 3.2). Graafilise kasutajaliidese tarkvaras tuli arvestada ette antud ettevõtte poolsete nõuetega (vt. peatükk 2.1 ja 2.2). Tarkvara loomisel võeti eeskujuna Python Tkinter kasutusjuhendist ja näidetest [10].

Graafilise kasutajaliidese loomisel kasutatakse Python3'e juba sisse installeeritud ja pakutavat teeki. Vastav teek on Python3'e standard graafilise liidese teek, mis pakub koos Python3'ega kiiret ja lihtsat viisi graafilise kasutajaliidese ehk GUI rakenduste loomist.

Tarkvaras kasutatakse Python3 teeki nimega Tkinter [9]. Tkinter'i abil saab luua kasutajatele töölauarakenduse, mis koosneb ja kuvab erinevaid klikatavaid nuppe rakenduses. Nuppudele peale vajutamisel käivitatakse just need funktsioonid, mis on vastavale nupule määratud. Toetudes sellisele tarkvarale saame töölauarakendusele luua kaks erinevat nuppu. Ühele nupule määrame dokumendi nime veergu lisamise käivitava funktsioon (vt. peatükk 3.1) ning teisele nupule määrame dokumentide liidestamise käivitava funktsiooni (vt. peatükk 3.2).

Sellise lahendusega oleme loonud tavakasutajale väga lihtsalt kasutatava kasutajaliidese, mis võimaldab kasutajal mõne nupu vajutamisega kogu nõutud töö ära teha (vt. Lisad 4). Tehtud lahendus rahuldab kõiki ettevõtte töötajate vajadusi ja nende poolseid ootusi.



## 4 Sihtgrupp

Diplomandi poolt pakutud dokumentidega seonduv automatiseeriv lahendus võiks olla ka abiks teistele väiksematele ettevõtetele. Sellisteks sihtgruppideks võiks lugeda näiteks hooajalised ettevõtted, kes pakuvad sõltuvalt hooajast enda poolseid kaupu või teenuseid. Sellised ettevõtted pigem ei soovi maksta kallite igakuiste raamatupidamistarkvarade eest. Vastavad tarkvarad on kindlasti hästi tehtud ja on suureks abiks, aga hooajalise ettevõttena tahetakse pigem võimalikult palju tööd ise ära teha ning kulutused minimaalsena hoida, et hooajaliselt võimalikult suur kasum teenida.

Diplomandi välja töötatud lahendust saab muuta vajadusel ettevõtte spetsiifiliseks, kuna ettevõtted käituvad erinevalt ja igaüks majandab firma siseseid tegevusi enda moodi. Olenemata kuidas keegi ettevõttes dokumentide majandamisega tegeleb, saab alati dokumentide majandamisel tööd automatiseerida. Olgu selleks erinevate dokumentide liidestamine või vajaliku info juurde lisamine dokumenti. Sellise töö automatiseerimine diplomandi loodud lahenduse näitel ei maksa ettevõtte jaoks mitte midagi. Sellised lahendused pigem aitavad just tüütut, manuaalset tehtavat tööd automatiseerida ja vajalikku tööaega kokku hoida, samuti tööaega efektiivsemalt kasutada ja keskenduda rohkem parema teenuse pakkumisele.

## 5 Tulemuste hinnang

Töö tulemuse kvaliteeti näitab automatsiooni kiirus ja imelihtne kasutajaliidese kasutus. Varasemalt ettevõtte töötajate aega nõudvat ja korduva töö manuaalset dokumentide manageerimist on nüüdseks tehtav palju kiiremini ning automaatselt.

Eelnevalt kulus töötajatel sellise ülesandes täitmiseks ligikaudu 20 minutit ja seda iga kliendi kohta eraldi. Kasutades loodud automatsiooni, tehakse töö ära kuni poole minutiga. Lihtne matemaatiline arvutus näitab, et töötajad hoiavad nüüd iga kliendiga vastava ülesande täitmisel kokku kuni 97.5% rohkem aega kui varem.

Töö automatiseerimisel on ka märgatav rahaline väärtus. Võimalusi on mitu - kuna tööaeg väheneb, siis ka tööjõudu ei ole sellises mahus enam vaja. Seega hoiaks kokku tööjõumaksudelt, mis Eestis teadagi on väga kõrged. Teine võimalus näiteks, töötajad saaksid võidetud ajaga teenindada uusi kliente ning seega hoopis rohkem raha sisse tuua ettevõttele.

Mikroettevõtte hoiab kokku seega igal aastal vähemalt 200 töötundi, mis on võrdeline enam-vähem 1740 euroga, kui võtta arvesse vastavas valdkonnas töötava inimese palga keskmine tunni hind 8.7 eurot [1]. Tänu sellele hoiab ettevõtte aastas kokku suure summa raha ning töötajad saavad rohkem keskenduda oma tavatööle ning tööaega efektiivsemalt ära kasutada.

## **Kokkuvõte**

Tehtud töö tulemusena valmis mikroettevõttele spetsiifiline finantsaruandluse halduse automatiseerimise tarkvara. Automatiseerimise juhtimine on teostatud kaheosalisena, milles üks osa võtab ja kopeerib dokumendi nime ning kirjutab selle dokumendile lisaks ja teine osa tõstab eraldi seisvate dokumentide sisud kokku uueks ühtseks dokumendiks. Kasutajaliidese lahendus lõi viisi, kus mõlemad eelnevalt mainitud funktsioonidel on vastavaks tegevuseks eraldi nupud, mille peale vajutamisel aktiveeritakse vastava funktsiooni tegevus.

Töö valmimisaeg on firmale suurepärane, sest puudusid tähtajad vastava lahenduse valmis saamiseks, ainult diplomandi kohustused hoida kinni kuupäevadest diplomitöö tegemisel. Töö tellinud mikroettevõtte on rahul automatiseerimise tarkvaralise lahendusega, sest lahendus täidab tellitud töö tingimusi ja soove just nii nagu algselt kokkulepitud sai.

Valminud lahenduse kiirus on ligikaudu 97.5% kiirem, kui eelnevalt sama töö tegemine manuaalselt. Kasutajate sõnul nõudis vastava tööülesande läbimine iga kliendi kohta 20 minutit tööd. Sama töö tehakse nüüd ära poole minutiga. Mikroettevõttel on kaudselt 50 sellist firmat, kellega vastav protsess läbi tehakse ja seda igal kuul. Vastava töö manuaalselt tegemisel nõudis igal kuul ligikaudu 17 tundi tööd, võttes kasutusse automatiseerimise, tehakse nüüdseks vastav töö ühes kuus ära ligikaudu poole tunniga.

Lisaks ajalisele kokkuhoiule on ka märgatav kokkuhoid rahas. Areneva ettevõtte jaoks on tähtis hoida erinevate kulutuste määr minimaalsena. Sarnaste manuaalsete tööde automatiseerimine aitab kindlasti ettevõttel edasi areneda ja nutikamaks muutuda.

## Kasutatud kirjandus

- [1] *Eesti riigis majanduse, rahanduse ja raamatupidamise valdkonnas töötava inimese palgavahemik.* [Kasutatud 30.04.2021 11:56]. 2021. URL: <https://www.palgad.ee/salaryinfo/majandus-rahandus-ja-raamatupidamine>.
- [2] *Kõrgema taseme interpreteeritud programmeerimis keel.* [Kasutatud 12.04.2021 21:03]. 2021. URL: <https://realpython.com/python-introduction/>.
- [3] *Laialdaselt Eestis levinud raamatupidamistarkvara.* [Kasutatud 08.05.2021 16:47]. 2021. URL: <https://www.excellent.ee/standard-books/>.
- [4] *Microsoft Excel võimalused mitmete failide kokku tõstmiseks ühte faili kokku.* [Kasutatud 03.04.2021 10:27]. 2018. URL: <https://professor-excel.com/merge-excel-files-combine-workbooks-one-file/>.
- [5] *Python3 moodul, mis pakub funktsioone operatsioonisüsteemiga suhtlemiseks.* [Kasutatud 05.04.2021 09:21]. 2021. URL: <https://docs.python.org/3/library/os.html>.
- [6] *Python3 Pandas teeki kokkuvõtlik kasutusjuhend.* [Kasutatud 28.04.2021 13:11]. 2021. URL: [https://pandas.pydata.org/docs/user\\_guide/index.html](https://pandas.pydata.org/docs/user_guide/index.html).
- [7] *Python3 teek, mis võimaldab andmemassiivide haldamist ja töötlemist.* [Kasutatud 05.04.2021 19:27]. 2021. URL: <https://pandas.pydata.org/about/index.html>.
- [8] *Python3 teek, mis võimaldab xlsx dokumente managerida.* [Kasutatud 05.04.2021 12:14]. 2021. URL: <https://openpyxl.readthedocs.io/en/stable/>.
- [9] *Python3 teek, võimaldab kiiret ja lihtsat viisi graafilise kasutajaliidese loomiseks.* [Kasutatud 05.04.2021 19:27]. 2021. URL: <https://docs.python.org/3/library/tk.html>.
- [10] *Python3 Tkinter teeki kokkuvõtlik kasutusjuhend.* [Kasutatud 29.04.2021 15:49]. 2021. URL: <https://docs.python.org/3/library/tkinter.html>.
- [11] *Sisukas ja kokkuvõtlik kirjeldus digitaalse kausta kohta.* [Kasutatud 12.05.2021 08:17]. 2015. URL: <https://techterms.com/definition/folder>.
- [12] *Sisukas ja kokkuvõtlik kirjeldus programmeerimiskeele Python kohta.* [Kasutatud 11.04.2021 09:03]. 2021. URL: <https://www.python.org/doc/essays/blurb/>.
- [13] *Sisukas ja kokkuvõtlik kirjeldus skripti kohta.* [Kasutatud 12.05.2021 21:27]. 2011. URL: <https://viki.pingviin.org/Skript>.

- [14] Al Sweigart. *Automate the Boring Stuff with Python, 2nd Edition*. No Starch Press, 2019. Ptk 12.
- [15] *Teksti ettevalmistus süsteem*. [Kasutatud 10.04.2021 17:39]. 2021. URL: <https://www.overleaf.com/>.
- [16] *Tööriist andmete arvutamiseks, summeerimiseks ja analüüsimiseks*. [Kasutatud 09.05.2021 11:11]. 2021. URL: <https://support.microsoft.com/et-ee/office/pivottable-liigendtabeli-loomine-t%C3%B6%C3%B6leheandmete-anal%C3%BC%C3%BCsimiseks-a9a84538-bfe9-40a9-a8e9-f99134456576#OfficeVersion=Windows>.

## Lisad 1 Lihtlitsents

Mina, Kristian Laupa

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Mikroettevõtte finantsaruandluse halduse automatiseerimine", mille juhendaja on Lembit Viilup.

1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.

3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

17.05.2021

## Lisad 2 Dokumendi nime lisamise loogika kood

```
import os
import openpyxl

def add_dates_to_file():

    # location of the folder
    for file_name in os.listdir('.'):

        # loop through the files
        if file_name.endswith('.xlsx'):

            # open & manage the files
            workbook = openpyxl.load_workbook(file_name)
            sheet = workbook['Sheet1']

            # add given value to the fifth cell,
            # starting from 2-nd row till the max rows
            for row in range(2, sheet.max_row + 1):
                # insert value to cell E1
                sheet['e1'] = 'kuupaev'
                add_date = file_name[:-5]
                add_date_cell = sheet.cell(row, 5)
                add_date_cell.value = add_date

            # save the modifications
            workbook.save(file_name)
```

### Lisad 3 Dokumentide liidestamise loogika kood

```
import os
import pandas

def merge_the_files():

    # location of the folder
    excel_loc = os.path.abspath('.')

    # list of all the files in folder
    files = os.listdir(excel_loc)

    # read each file data into dataframe
    data = pandas.DataFrame()

    # loop through the files
    for file in files:
        if file.endswith('.xlsx'):
            #append rows to the given dataframe
            data = data.append(pandas.read_excel(file),
                               #no index values used along the concatenation axis
                               ignore_index=True)
    data.head()

    # combine all the data
    data.to_excel('merged_files.xlsx')
```



## Lisad 4 Graafilise kasutajaliidese loogika kood

```
from tkinter import *
from tkinter import messagebox
from add_date import add_dates_to_file
from merge_xslx import merge_the_files

window = Tk()

#window size in pixels
window.geometry("250x200")

#window can't be resized
window.resizable(0, 0)

window.title("Excel_automation")

# If button is clicked display msg "Dates are added!"
def show_msg_date():
    messagebox.showinfo('Message', 'Dates_are_added!')

# If button is clicked display msg "Documents are merged!"
def show_msg_merge():
    messagebox.showinfo('Message', 'Documents_are_merged!')

# Button called Add_dates and it's functions
btn1 = Button(window, text="Add_dates", height=5, width=10,
command=lambda: [add_dates_to_file(), show_msg_date()])
.place(x=10, y=50)

# Button called Merge_excels and it's functions
btn2 = Button(window, text="Merge_excels", height=5, width=10,
command=lambda: [merge_the_files(), show_msg_merge()])
.place(x=150, y=50)

window.mainloop()
```