



**Kauno technologijos universitetas**

Informatikos fakultetas

**Finansinių dokumentų automatinis turinio nuskaitymas  
naudojant dirbtinio intelekto ir RPA technologijas**

Baigiamasis magistro studijų projektas

---

**Vilius Kerutis**

Projekto autorius

**Doc. Agnė Paulauskaitė - Tarasevičienė**

Vadovė

---

**Kaunas, 2022**



**Kauno technologijos universitetas**

Informatikos fakultetas

# **Finansinių dokumentų automatinis turinio nuskaitymas naudojant dirbtinio intelekto ir RPA technologijas**

Baigiamasis magistro studijų projektas

Programų sistemų inžinerija (6211BX011)

---

**Vilius Kerutis**

Projekto autorius

**Doc. Agnė Paulauskaitė -  
Tarasevičienė**

Vadovė

**Dalia Čalnerytė**

Recenzentas

---

**Kaunas, 2022**



**Kauno technologijos universitetas**

Informatikos fakultetas

Vilius Kerutis

## **Finansinių dokumentų automatinis turinio nuskaitymas naudojant dirbtinio intelekto ir RPA technologijas**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Viliaus Keručio, baigiamasis projektas tema „Finansinių dokumentų automatinis turinio nuskaitymas naudojant dirbtinio intelekto ir RPA technologijas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

---

(vardą ir pavardę įrašyti ranka)

---

(parašas)

## TECHNINĖ UŽDUOTIS

Sukurti programinės įrangos robotą, kuris automatizuotų finansinių dokumentų turinio nuskaitymo procesą imituojant fizinio asmens veiksmus bei įpročius. Sukurtos programinės įrangos tikslas – panaudojant RPA (angl. *Robotic Process Automation*) bei dirbtinio intelekto technologijas robotizuoti fizinių asmenų atliekamą rutininį darbą, kas leistų sumažinti užsakovų veiklos kaštus tiesiogiai susijusius su reikalingų etatų išlaikymu, pagreitinti užduočių įvykdymo laiką bei sumažinti klaidų skaičių įtakojamą žmogiškojo faktoriaus. Sukurta programinė įranga apima šiuos tris pagrindinius procesus:

1. Finansinių dokumentų parsisiuntimas iš *SharePoint* repozitorijos į lokalią direktoriją ir užkrovimas į *Blue Prism* esančią eilę. Parsiuntus dokumentus yra atmetami dokumentai, kurie nėra PDF formato.
2. Finansinių dokumentų PDF formato keitimas į PNG naudojant *ImageMagick* įrankį, o vėliau PNG formato keitimas į TXT naudojant *Tesseract-OCR* įrankį. Iš vieno PDF dokumento atsiradusių kelių TXT failų apjungimas į vieną tekstinį kintamąjį. Galiausiai, iš tekstinio kintamojo yra išfiltruojama reikiama informacija pagal taisyklių faile aprašytas taisykles.
3. Ataskaitos apie atliktą darbą sugeneravimas ir išsiuntimas už procesą ir programinį robotą „atsakingiems žmonėms“. Atsakingais žmonėmis – yra vadinami fiziniai asmenys, kurių vykdomas procesas yra robotizuojamas bei asmenys, kurie prižiūri programinės įrangos robotą ir užtikrina sklandų proceso veiklą įvykdymą.

Sukurtai programinei įrangai atsižvelgiant į užsakovų keliamus reikalavimus turi būti atliktas testavimas, kuris parodytų, kad programa geba atpažinti bent 80 procentų laukų teisingai. Testavimas turėtų vykti su dar nematytais finansiniais dokumentais. Tam, kad atpažinimo tikslumo įvertis būtų kiek įmanoma teisingas, reikia testuoti su kiek įmanoma daugiau skirtingų finansinių dokumentų. Finansinių dokumentų pateikimas testavimui yra užsakovų atsakomybė.

Vilius Kerutis. Finansinių dokumentų automatinis turinio nuskaitymas naudojant dirbtinio intelekto ir RPA technologijas. Magistro studijų baigiamasis projektas / vadovė doc. Agnė Paulauskaitė - Tarasevičienė; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Programų sistemų inžinerija.

Reikšminiai žodžiai: RPA, robotas, AI, Finansinių dokumentų nuskaitymas.

Kaunas, 2022. 70 p.

## **Santrauka**

Šiandieniniame pasaulyje ir taip sparčiai besivystant technologijoms versle vis daugiau uždavinių yra sprendžiama pasitelkiant dirbtinį intelektą ir robotizaciją. Pastaraisiais metais robotų integravimas tiek į buitinę, tiek į pramoninę erdvę tapo kone kasdienybe, juolab vertinant, kad robotai gali būti fiziniai arba programiniai. Tačiau verslo ir paslaugų teikimo sektoriuje atsiradęs robotinio procesų automatizavimas (RPA) yra ganėtinai „jauna“ ir „perspektyvi“ technologija, kurios sprendimai taikomi rutininiams darbams automatizuoti – robotizuoti žmogaus atliekamus veiksmus IT sistemose. Finansinių dokumentų automatinio turinio nuskaitymas naudojant dirbtinio intelekto ir RPA technologijas yra itin aktuali tema šiuolaikinio verslo pasaulyje. Tokių RPA sistemų tikslas apima esminių dokumento laukų (data, suma, tiekėjas, pardavėjas ir pan.) aptikimą ir informacijos nuskaitymą. Nors rinkoje yra nemažai komercinių sprendimų orientuotų šio uždavinio sprendimui, tačiau kokybiškai nuskaityti ir atpažinti visus reikiamus laukelius nepriklausomai nuo finansinio dokumento struktūros vis dar yra iššūkis.

Baigiamajame darbe yra pateiktas finansinių dokumentų automatinio turinio nuskaitymo problemos sprendimas pasitelkiant RPA (angl. Robotic Process Automation) ir dirbtinio intelekto technologijas.

Tiriamąjį darbo rezultatą programa, kuri geba efektyviai ir tiksliai atpažinti nurodytus laukelius nepriklausomai nuo finansinio dokumento struktūros panaudojant pažangiausias dirbtinio intelekto bei RPA technologijas taip siekiant dar labiau padidinti programos efektyvumą.

Baigiamojo darbo dokumento įvade yra pateikiamas problemos aprašymas, jos aktualumas, išsikelti tikslai ir uždaviniai šiai problemai spręsti. Analizės dalyje apžvelgiamos tiek RPA metodikos, tiek alternatyvūs būdai rutininių darbų automatizavimui. Šioje dalyje taip pat nagrinėjami rinkoje egzistuojantys produktai (programinės įrangos, sistemos, prototipai ir pan.), įvardinant jų silpnąsias ir stipriąsias puses, taikymo galimybes, ribojimus ir funkcionalumo detales. Projektinėje darbo dalyje pateikiamas siūlomo sprendimo - sukurtos sistemos aprašymas pradedant reikalavimų specifikacija ir baigiant programinio roboto projekto detalizacija. Sukurtos sistemos kokybės analizė ir tobulinimo galimybės pateiktos tyrimo dalyje. Sukurtos ir įdiegtos programinės įrangos bei jos patobulinimų eksperimentinis tyrimas pateiktas eksperimentinėje dalyje. Paskutiniame dokumento skyriuje pateikiami rezultatai, išvados ir ateities išvalgos.

Vilius Kerutis. Automatic reading of the content of financial documents using artificial intelligence and RPA technologies. Master's Final Degree Project / supervisor doc. Agnė Paulauskaitė - Tarasevičienė; Informatics Faculty, Kaunas University of Technology.

Study field and area (study field group): Computer Sciences, Software Systems Engineering.

Keywords: RPA, robotization, robot, AI.

Kaunas, 2022. 70 p.

### **Summary**

In today's world of so rapid technological advancement in business, more and more challenges are being addressed through artificial intelligence and robotization. In recent years, the integration of robots into both domestic and industrial applications has become almost commonplace, especially given that robots can be physical or software. However, Robotic Process Automation (RPA), which has emerged in the business and service industry, is a relatively "young" and "promising" technology that uses solutions to automate routine work - robotizing human actions in IT systems. The automatic reading of the content of financial documents using artificial intelligence and RPA technologies is a very relevant topic in today's business world. The purpose of such RPA systems includes the detection of essential fields of a document (date, amount, supplier, vendor, etc.) and the retrieval of information. Although there are a number of commercial solutions on the market focused on solving this problem, it is still a challenge to qualitatively read and recognize all the required fields regardless of the structure of the financial document.

This thesis presents a solution to the problem of reading of the content of financial documents automation using RPA (Robotic Process Automation) and artificial intelligence technologies.

The result of the research work is a program that is able to recognize the specified fields efficiently and accurately, regardless of the structure of the financial document, using the most advanced technologies of artificial intelligence and RPA, thus further increasing the efficiency of the program.

The introduction to the thesis paper contains a description of the problem, its relevance, the objectives pursued and the tasks to be undertaken to solve the problem. The analysis section reviews both RPA methodologies and alternative ways to automate routine work. This section also examines the products currently available on the market (software, systems, prototypes, etc.), identifying their weaknesses, strengths, application capabilities, limitations, and functionality. In the practical part of the work, there is a description of the proposed solution - the developed system starting with the specification of requirements and ending with the detailed design of the software robot. The analysis of the quality of the developed system and the possibilities for improvement are presented in the research part. An experimental study of the developed and deployed software and its improvements is presented in the experimental part. The final section of the document presents results, conclusions, and future insights.

## Turinys

<b>Turinys</b> .....	<b>7</b>
<b>Lentelių sąrašas</b> .....	<b>9</b>
<b>Paveikslų sąrašas</b> .....	<b>10</b>
<b>1. ĮŽANGA</b> .....	<b>11</b>
<b>2. ANALITINĖ DALIS</b> .....	<b>13</b>
2.1. Srities apžvalga.....	13
2.1.1. OCR veikimas .....	13
2.1.2. Taikomos technologijos, jų veikimo principai ir apribojimai .....	14
2.1.3. <i>Tesseract-OCR</i> veikimas .....	15
2.2. Technologijų pasirinkimo pagrindimas .....	16
2.3. Egzistuojančių rinkoje programų sistemų savybių kiekybinis ir kokybinis palyginimas .....	19
<b>3. PROJEK TINĖ DALIS</b> .....	<b>21</b>
3.1. Panaudojimo atvejų vaizdas .....	21
3.1.1. Panaudojimo atvejų sąrašas .....	21
3.2. Sistemos statinis vaizdas .....	24
3.2.1. Apžvalga.....	24
3.2.2. Paketų detalizavimas .....	25
3.3. Sistemos dinaminis vaizdas .....	26
3.3.1. Būsenų diagrama .....	26
3.3.2. Veiklos diagramos .....	27
3.3.3. Sekų diagramos .....	31
3.3.4. Bendradarbiavimo diagrama .....	36
3.4. Išdėstymo vaizdas.....	36
3.5. Duomenų vaizdas .....	37
<b>4. TYRIMO DALIS</b> .....	<b>38</b>
4.1. Kokybės vertinimas .....	38
4.1.1. Atlikto darbo kokybės analizės tikslai.....	38
4.1.2. Kokybės vertinimo procesas.....	38
4.1.3. Vertinimo rezultatai.....	39
4.2. Sistemos parametrų tyrimas .....	41
4.3. Greitaveikos tyrimas.....	43
<b>5. EKSPERIMENTINĖ DALIS</b> .....	<b>46</b>
5.1. Testavimo rezultatai ir išvados .....	46
5.2. Laukų atpažinimo tikslumas.....	48
5.3. Apdorojimo laikų palyginimas .....	49
<b>6. IŠVADOS</b> .....	<b>51</b>
<b>7. LITERATŪRA</b> .....	<b>52</b>
<b>8. TERMINŲ IR SANTRUMPŲ ŽODYNAS</b> .....	<b>54</b>
<b>Priedai</b> .....	<b>55</b>
1. priedas. Užpildyta apklausos anketa.....	55
2. priedas. Straipsnis.....	58

<b>1</b>	<b>Introduction .....</b>	<b>58</b>
<b>2</b>	<b>Materials and Methods .....</b>	<b>60</b>
2.1	Employed Technologies .....	62
2.2	Developed Algorithm .....	63
<b>3</b>	<b>Experiments .....</b>	<b>64</b>
<b>4</b>	<b>Discussion .....</b>	<b>67</b>
<b>5</b>	<b>Conclusions .....</b>	<b>67</b>
	<b>References.....</b>	<b>68</b>



## Lentelių sąrašas

<b>2.1 lentelė</b> Automatizavimo įrankių apžvalga.....	17
<b>2.2 lentelė</b> Programų sistemų palyginimas.....	19
<b>3.1 lentelė</b> <i>PA 1. Išsiųsti ataskaitą</i> detalioji specifikacija.....	21
<b>3.2 lentelė</b> <i>PA 2. Nuskaityti taisyklių failą</i> detalioji specifikacija.....	22
<b>3.3 lentelė</b> <i>PA 3. Sugeneruoti ataskaitą</i> detalioji specifikacija.....	22
<b>3.4 lentelė</b> <i>PA 4. Įrašyti apdorotą informaciją</i> detalioji specifikacija .....	23
<b>3.5 lentelė</b> <i>PA 5. Nuskaityti finansinį dokumentą</i> detalioji specifikacija .....	23
<b>3.6 lentelė</b> <i>PA 6. Apdoroti finansinį dokumentą</i> detalioji specifikacija .....	24
<b>4.1 lentelė</b> Kokybės vertinimo rezultatai .....	40
<b>4.2 lentelė</b> Visų tyrimų apibendrinta rezultatų lentelė .....	45
<b>5.1 lentelė</b> Finansinių dokumentų tikslumas priklausomai nuo dokumentų kiekio .....	48
<b>5.2 lentelė</b> Fizinio asmens ir RPA apdoravimo laikų palyginimas.....	49
<b>5.3 lentelė</b> Fizinio asmens ir RPA vidutinio dokumento apdoravimo laiko palyginimas .....	50

## Paveikslų sąrašas

<b>2.1 pav.</b> Laukų pažymėjimas ir atpažinimas taikant <i>OCR</i> .....	13
<b>2.2 pav.</b> Vaizdinis <i>OCR</i> pavyzdys .....	13
<b>2.3 pav.</b> „Nanonets“ įrankio naudojimosi principas .....	14
<b>2.4 pav.</b> <i>Tesseract</i> veikimo principas .....	15
<b>3.1 pav.</b> Projekto panaudojimo atvejų diagrama (PAM)Panaudojimo atvejų sąrašas .....	21
<b>3.2 pav.</b> Sistemos paketų diagrama .....	25
<b>3.3 pav.</b> Projekto klasių diagrama .....	25
<b>3.4 pav.</b> <i>DocumentProcessing</i> paketo klasių diagrama.....	25
<b>3.5 pav.</b> <i>FileManagement</i> paketo klasių diagrama.....	26
<b>3.6 pav.</b> <i>RuleManagement</i> paketo klasių diagrama.....	26
<b>3.7 pav.</b> Projekto būsenų diagrama .....	27
<b>3.8 pav.</b> <i>Taisyklių failo nuskaitymo</i> veiklos diagrama .....	27
<b>3.9 pav.</b> <i>Finansinio dokumento nuskaitymo</i> veiklos diagrama .....	28
<b>3.10 pav.</b> <i>Finansinio dokumento apdorojimo</i> veiklos diagrama .....	28
<b>3.11 pav.</b> <i>Ataskaitos išsiuntimo</i> veiklos diagrama .....	29
<b>3.12 pav.</b> <i>Apdorotos informacijos įrašymo</i> veiklos diagrama.....	30
<b>3.13 pav.</b> <i>Ataskaitos sugeneravimo</i> veiklos diagrama .....	31
<b>3.14 pav.</b> <i>Taisyklių failo nuskaitymo</i> sekų diagrama.....	32
<b>3.15 pav.</b> <i>Finansinio dokumento nuskaitymo</i> sekų diagrama.....	32
<b>3.16 pav.</b> <i>Finansinio dokumento apdorojimo</i> sekų diagrama .....	33
<b>3.17 pav.</b> <i>Ataskaitos išsiuntimo</i> sekų diagrama.....	33
<b>3.18 pav.</b> <i>Apdorotos informacijos įrašymo</i> sekų diagrama .....	34
<b>3.19 pav.</b> <i>Ataskaitos sugeneravimo</i> sekų diagrama.....	35
<b>3.20 pav.</b> <i>Finansinio dokumento nuskaitymo</i> bendradarbiavimo diagrama.....	36
<b>3.21 pav.</b> Proceso diegimo diagrama .....	36
<b>3.22 pav.</b> Dalykinės srities duomenų bazės klasių diagrama .....	37
<b>4.1 pav.</b> Vidutinio dokumento apdorojimo laiko priklausomybė nuo suglaudavimo kokybės .....	41
<b>4.2 pav.</b> Tikslaus laukų atpažinimo priklausomybė nuo suglaudavimo kokybės.....	42
<b>4.3 pav.</b> Vidutinio dokumento apdorojimo laiko priklausomybė nuo DPI.....	42
<b>4.4 pav.</b> Tikslaus laukų atpažinimo priklausomybė nuo DPI.....	43
<b>4.5 pav.</b> Vykdyto laiko priklausomybė nuo dokumentų kiekio .....	44
<b>4.6 pav.</b> Vidutinės trukmės vieno dokumento apdorojimui priklausomybė nuo dokumentų kiekio .....	44
<b>5.1 pav.</b> Taisyklių failo datos nustatymo pavyzdys .....	46
<b>5.2 pav.</b> Testavimo rezultatų pavyzdys .....	47
<b>5.3 pav.</b> Viso dokumento ir atskirų laukų tikslumo palyginimas.....	48
<b>5.4 pav.</b> Fizinio asmens ir RPA vykdymo laikų palyginimas .....	49
<b>5.5 pav.</b> Vidutinės trukmės vieno dokumento apdorojimui priklausomybė nuo dokumentų kiekio .....	50

## 1. IŽANGA

Finansinių dokumentų automatinio turinio nuskaitymas naudojant dirbtinio intelekto ir robotinio procesų automatizavimo arba procesų robotizavimo (RPA) technologijas yra itin aktuali šiuolaikinio verslo tema. Šio darbo tikslas yra kokybiškai nuskaityti ir atpažinti visus reikiamus laukelius nepriklausomai nuo finansinio dokumento struktūros. Rinkoje siūlomi sprendimai yra itin priklausomi nuo dokumento struktūros, t.y. jeigu paduodamas finansinis dokumentas, kurio struktūra yra kitokia nei dokumentų, su kuriais sprendimas veikė teisingai, tai laukų atpažinimo efektyvumas ženkliai sumažėja. Šio tiriamojo darbo rezultatas yra programa, kuri geba efektyviai ir tiksliai atpažinti nurodytus laukelius nepriklausomai nuo finansinio dokumento struktūros. Dėl šios priežasties siūlomas sprendimas išsiskiria iš rinkoje esančių, nes jo pritaikomumas yra daug platesnis. Tikslui įgyvendinti yra naudojamos pažangiausios dirbtinio intelekto technologijos susietos su RPA technologijomis, o tai dar labiau padidina programos efektyvumą. Sukurto sprendimo pridėtinė vertė – automatizuojamas fizinių asmenų darbas, mažinami įmonių kaštai, atsisakoma rutininio darbo specifikos.

Kuriamo produkto potencialūs vartotojai dirba pakankamai didelėse įmonėse (500 darbuotojų ir daugiau), kurios pardavinėja prekes ar paslaugas ir už tai išrašo sąskaitas. Taip pat tai yra darbuotojai dirbantys įmonėse, kuriose klientai gali užsisakyti prekes ar paslaugas internetinėje erdvėje. Taip būna sugeneruota užsakymo sąskaita, kurią darbuotojai turi apdoroti. Siūlomas produktas yra kuriamas tam, kad darbuotojams nereiktų fiziškai apdoroti dokumentų, o tai būtų padaroma automatiškai. Siūlomo produkto vartotojai turėtų pasižymėti minimaliu kompiuteriniu raštingumu tam, kad sugebėtų programai nurodyti reikiamus apdoroti finansinius dokumentus ir gebėtų panaudoti programos rezultatą – pagrindinę finansinių dokumentų laukų informaciją. Produkto naudotojai yra žmonės, dirbantys finansų skyriuose. Taip pat tai gali būti asmenys, kurie dirba pardavimų skyriuose. Programos naudotojai turėtų gerai išmanyti finansinių dokumentų struktūrą bei teikiamą informaciją tam, kad gebėtų pritaikyti programos išvestus rezultatus. Taigi, programos vartotojai neturėtų išsiskirti ypatingomis savybėmis, bet turėtų būti gerai susipažinę su finansiniais dokumentais bei turėtų minimalių kompiuterinio raštingumo žinių.

Dabartinė būsimų vartotojų problema yra laiko sąnaudos skirtos finansiniams dokumentams apdoroti. Kuriamas produktas galėtų visiškai panaikinti šią problemą, o vartotojai galėtų užsiimti prasmingesne, didesnio intelektualinio mąstymo reikalaujančia veikla. Tokiu būdu būtų keliamas darbuotojų produktyvumas ir efektyvumas. Darbuotojų pasitenkinimas darbu išaugtų pastebimai, nes mažai kas nori dirbti nuobodų bei pasikartojantį darbą.

Galime teigti, kad efektyvių, nuo finansinio dokumentų struktūros nepriklausančių sprendimų laiko sąnaudų sumažinimo poreikiui patenkinti šiuo metu rinkoje nėra arba yra, bet nepakankamai efektyvūs.

Produkto paskirtis yra pakankamai aiški – finansinių dokumentų nuskaitymas ir apdorojimas. Didžiausi reikalavimai iš vartotojo perspektyvos būtų patogumas, greitis bei tikslumas. Siūlomas sprendimas tenkintų visus šiuos reikalavimus, nes sukurtas produktas yra itin patogus naudotojui, labai greitas lyginant su fizinių asmenų užduočių atlikimo greičiu bei tikslus. Šis produktas galėtų tapti itin sunkiai pakeičiamas kasdieninėje verslo pasaulio veikloje.

**Šio darbo tikslas** - sukurti programinės įrangos robotą, kuris automatizuotų finansinių dokumentų turinio nuskaitymo procesą imituojuant fizinio asmens veiksmus bei įpročius panaudojant RPA (angl.

*Robotic Process Automation*) bei dirbtinio intelekto technologijas. Tokiu būdu robotizuoti fizinių asmenų atliekamą rutininį darbą, taip sumažinant užsakovų veiklos kaštus tiesiogiai susijusius su reikalingu etatų išlaikymu, pagreitinti užduočių įvykdymo laiką bei sumažinti klaidų skaičių įtakojamą žmogiškojo faktoriaus. Tikslui pasiekti išsikelti šie uždaviniai:

1. Apžvelgti esamus RPA sprendimus (komercinius) bei ištirti jų galimybes, išryškinant privalumus ir trūkumus automatizuotam finansinių dokumentų nuskaitymui.
2. Surinkti finansinių dokumentų pavyzdžius sudarant reprezentatyvią ir įvairialypę finansinių dokumentų (sąskaitų faktūrų) duomenų imtį *.pdf* formatu.
3. Išskirti svarbiausiu sąskaitų faktūrų laukus, kurie dažniausiai suvedami į apskaitos sistemą.
4. Automatiškai surasti ir nuskaityti išskirtus informacinius dokumento laukus panaudojant OCR technologiją bei dirbtinio intelekto metodus.
5. Atlikto sukurto sprendimo testavimą vertinant surastos informacijos tikslumo, greitaveikos metrikas.
6. Integruoti RPA technologijas sukuriant programinį robotą atsakingą už ataskaitos apie atliktą darbą sugeneravimą ir išsiuntimą nurodytiems asmenims.

Baigiamąjį darbą sudaro penki pagrindiniai skyriai, išvados, literatūra, terminų, santrumpų žodynas bei priedai. Šio tiriamojo darbo tematikoje autorius publikavo dvi publikacijas: viena bakalauro studijose „*Service quotation generation using robotic process automation technology*“, 2020, ir antroji šiais metais pavadinimu „*Intelligent Invoice Documents Processing Employing RPA Technologies*“, 2022.

## 2. ANALITINĖ DALIS

### 2.1. Srities apžvalga

Populiariausias ir verslo pasaulyje dažniausiai taikomas sprendimas finansiniam dokumentams apdoroti yra optinis simbolių atpažinimas (angl. Optical Character Recognition – OCR) [1] naudojamas kartu su dirbtiniu intelektu. Daugelio sprendimų principas yra labai panašus: sistemai yra perduodamas dokumentas, tame dokumente yra pažymimi visi laukai, kurie turėtų būti atpažinti, OCR technologija nuskaityto pažymėtas vietas ir toks sužymėtas dokumentas (žr. 2.1 pav.) yra perduodamas atpažinimo modeliui mokytis. Vėliau yra perduodami realūs nauji dokumentai ir modelis geba atpažinti pateiktą informaciją. Verslo pasaulyje tai yra priimtinausias sprendimas [2], tačiau šis sprendimas yra visiškai priklausomas nuo dokumento struktūros. Jeigu yra pakeičiama finansinio dokumento struktūra, tai modelis jau negali atpažinti visų laukų. Modelis atpažįsta tik kelis, kurių radimas yra itin paprastas, pavyzdžiui, galutinę užsakymo sumą – ji dažniausiai rašoma dokumento apačioje, po visu užsakymu, po žodžių „iš viso“ (angl. total amount). Taigi, šis informacijos finansiniuose dokumentuose atpažinimo metodas yra plačiai taikomas, tačiau nėra itin stabilus, kai dokumento struktūra yra kintanti. Sprendimas puikiai veikia, jeigu dokumento forma yra pastovi, laukai yra tvarkingai užpildomi.

The diagram illustrates the OCR process on an invoice. On the left, a scanned invoice is shown with various fields highlighted in colored boxes: 'Your company name' (green), '123 Your Street, City, State, Country, ZIP Code' (blue), '564-555-1234, your@email.com, yourwebsite.com' (purple), 'BILLED TO, Client Name, Street address, City, State Country, ZIP Code' (pink), 'Invoice number' (magenta), and 'DATE OF ISSUE, mm/dd/yyyy' (green). In the center, there is a 'DOC' icon. On the right, a 'LABELS' list is shown with the following items:

Label	Value
buyer_address	City, State Country ZIP Code
buyer_name	Client Name
invoice_amount	\$2000
invoice_date	mm/dd/yyyy
invoice_number	00001
payment_due_date	MM/DD/YYYY
seller_address	123 Your Street City, State, Country

2.1 pav. Laukų pažymėjimas ir atpažinimas taikant OCR

#### 2.1.1. OCR veikimas

OCR principas veikia panašiai kaip ir žmogaus matymas bei ženklų atpažinimas [3], [4]. Žmogaus akys skaitant tekstą atpažįsta šviesesnes ir tamsesnes vietas, o smegenys atrenka simbolius pagal jau žinomus simbolius. Kompiuteris gali atlikti tą patį veiksmą, tačiau jam yra reikalingas teksto vaizdas. Kitaip tariant, tekstas privalo būti skaitmeninis. Turint vaizdą, OCR bando vaizdą paversti tekstu, nes pats vaizdas – nesvarbu ar tai bet kokia fotografija, ar PDF formato tekstinis dokumentas – jam nesiskiria ir yra tik pikselių visuma (žr. 2.2 pav.).



2.2 pav. Vaizdinis OCR pavyzdys

OCR yra labai naudingas, nes vaizdą paverčia tekstu, o tai padidina bet kokių veiksmų, susijusių su informacijos apdorojimu, skaičių [5]. Vaizdas verčiamas tekstu, nes dažniausiai reikiama iš to išgauti informaciją [6]. Turint tekstą galima atlikti paieškas, filtruoti tekstą ar paprasčiausiai atpažinti, kokia informacija yra pateikta, tą informaciją panaudoti kitur ir pan.

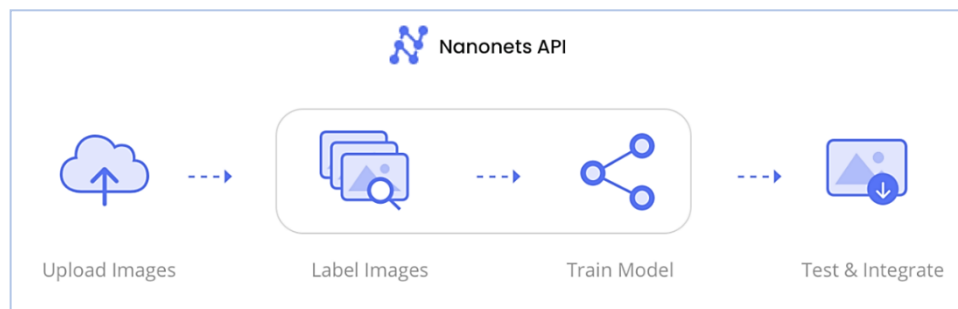
Pats atpažinimas yra ganėtinai sunkus, nes vien skaitmeninėje erdvėje egzistuoja skirtingi šriftai ir rašto stiliai, o jeigu dar yra atpažįstami vaizdai, kuriuose matomas ranka parašytas tekstas, tai tampa dar sudėtingiau apdoroti tokį vaizdą, nes kiekvienas žmogus turi skirtingą rašyseną. Tam, kad būtų atpažinta kuo daugiau įvairiausių vaizdų ir paverčiama juos į tekstą, yra naudojami du metodai:

1. šablono atpažinimas (angl. *pattern recognition*). Veikimo principas labai paprastas – OCR logikai yra perduodami kiekvieno žinomo simbolio įvairiausi šablonai. Modelis, gavęs naują vaizdą simboliams atpažinti, taiko šablonus, kol yra randamas tinkantis ir vaizdas yra paverčiamas tekstu.
2. funkcijų aptikimas (angl. *feature detection*). Šis būdas yra daug sudėtingesnis už pirmąjį, tačiau žymiai plačiau panaudojamas. Jis vadovaujasi linijų braižymo taisyklėmis, pavyzdžiui, jeigu yra dvi susikertančios viename taške linijos, o žemiau jų yra dar viena linija, kertanti prieš tai minėtas linijas dviejuose taškuose, tai galima teigti, kad tai yra „A“ simbolis. Šis metodas yra glaudžiai susijęs su neuroniniais tinklais ir simboliu atpažinimas vyksta būtent tokiu principu.

Ranka parašytų tekstų vaizde atpažinimas vyksta taikant tik funkcijų aptikimo metodą [7], [8]. Tai nėra itin stabilus metodas, nes ranka rašyto rašto atpažinimas kartais tampa sunkia užduotimi net žmogui. Savaiame suprantama, kad kompiuteriai atpažinti tokį raštą yra dar sunkiau.

### 2.1.2. Taikomos technologijos, jų veikimo principai ir apribojimai

Vienas populiariausių nemokamų dokumentų (daugiausiai specializuojasi į finansinius dokumentus) apdorojimo įrankių – „Nanonets“ [9]. Šis internetinis įrankis yra ganėtinai lengvai suprantamas ir juo naudotis paprasta. Jo veikimas yra pagrįstas OCR ir dirbtinio intelekto technologijomis (žr. 2.3 pav.). Naudotojai gali patys išmokyti modelį paduodami jam testinius dokumentus ir parodydami, kas ir kur randasi (žr. 2.1 pav.). Vėliau yra paduodami realūs, nauji dokumentai ir modelis geba atpažinti anksčiau parodytus laukus. Šis įrankis buvo ištirtas su dešimtimis finansinių dokumentų ir veikė puikiai, tačiau, padavus tuos pačius dokumentus kitokia struktūra, modelio laukų atpažinimas suprastėjo daugiau kaip penkiasdešimt procentų. Buvo prieita prie išvados, kad šis įrankis puikiai veikia nurodytos struktūros finansiniams dokumentams, tačiau nelabai naudingas, kai dokumento struktūra yra keičiama. Vadinasi, kad jeigu yra paduodama daug ir įvairių dokumentų, tai šis įrankis nėra efektyvus, nebent bus nurodomi kiekvienos naujos struktūros laukai.



2.3 pav. „Nanonets“ įrankio naudojimosi principas

Šis įrankis deklaruoja, kad dokumentams apdoroti naudoja konvoliucinius neuroninius tinklus (CNN), tačiau kokios architektūros šie tinklai nėra aišku ir negalima išbandyti, nes pačio įrankio išeities kodas nėra laisvai prieinamas.

Yra ir daugiau OCR grįstų įrankių, atliekančių dokumentų apdorojimą bei naudojančių tokias pat technologijas kaip ir prieš tai aprašytas įrankis:

1. „Conexiom“ [10],
2. „Klippa“ [11],
3. „Logikia“ [12],
4. „FastFour“ [13].

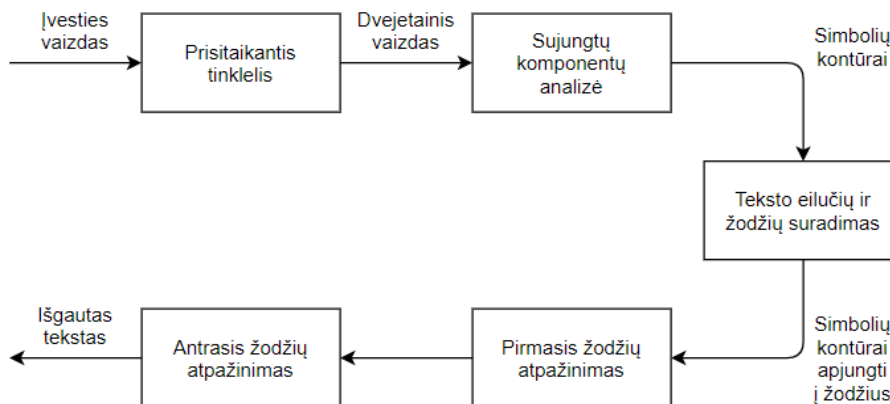
Jų skirtumai aptarti 2.3. skyriuje.

### 2.1.3. Tesseract-OCR veikimas

*Tesseract* yra atvirojo kodo teksto atpažinimo (OCR) variklis, prieinamas pagal *Apache 2.0* licenciją [14], [15]. Šis variklis gali būti naudojamas tiesiogiai arba (programinės įrangos kūrimo tikslais) naudojant *API* spausdintam tekstui iš vaizdų išgauti [16]. Jis palaiko įvairiausias kalbas. *Tesseract* neturi implementuotos naudotojo sąsajos, tačiau internete galima rasti naudotojo sąsajų, kurios implementuoja *Tesseract*. *Tesseract* yra suderinamas su daugeliu programavimo kalbų ir platformų. Šis variklis gali būti naudojamas kartu su esama maketo analize, norint atpažinti tekstą dideliame dokumente, arba gali būti naudojamas kartu su išoriniu teksto detektoriumi tekstui iš vienos teksto eilutės atvaizdo atpažinti [17].

Žemiau yra pateiktas *Tesseract* veikimo principas (žr. 2.4 pav.). Čia matome, kad *Tesseract* varikliui yra vaizdas pateikiamas kaip įvestis. Vaizdas yra perkeliamas ant prisitaikančio tinklelio, o šis vaizdą paverčia dvejetainiu [18]. Tada yra vykdoma sujungtų komponentų analizė, kurios rezultatas yra simbolių kontūrai. Pagal šiuos kontūrus yra bandoma surasti teksto eilutes ir žodžius. Radus teksto eilutes ir žodžius, simbolių kontūrai yra sujungiami į pilnus žodžius. Šie žodžiai praeina du žodžių atpažinimo lygius [19] ir išgautas tekstas yra išvedamas kaip *Tesseract* variklio rezultatas.

Pridėjus naują mokymo priemonę ir išmokius modelį su daugybe duomenų ir šriftų, „Tesseract“ pasiekia geresnių rezultatų. Vis dėlto, rezultatai yra nepakankamai geri, kad būtų galima dirbti su ranka rašytu tekstu ir išskirtiniais šriftais [20]. Eksperimentiniu keliu yra galimybė tikslinti arba perkvalifikuoti viršutinius sluoksnius tam, kad būtų pasiekti geresni rezultatai [21].



2.4 pav. *Tesseract* veikimo principas

## 2.2. Technologijų pasirinkimo pagrindimas

Projektui įgyvendinti buvo reikalingi du etapai. Šiame skyrelyje yra aprašomi kiekvienas atskirai.

### Pirmasis etapas. DI srities technologijos.

Dirbtinio intelekto sprendimai yra įgyvendinti naudojant *Python* programavimo kalbą, nes yra priimta, kad tai yra patogiausia ir daugiausiai galimybių turinti kalba kai yra dirbama su įvairiausiais duomenimis skirtingose formose [22]. Kadangi projekto pagrindinė dalis susideda iš dirbtinio intelekto, tai *Python* programavimo kalba yra nepakeičiama norint sukurti funkcionalų ir veikiantį sprendimą. Programavimo aplinka yra itin svarbi tam, kad būtų sukurtas anksčiau minėtas sprendimas. Šiam tikslui aplinka turėtų būti intuityvi, patogi naudoti bei pritaikyta darbui su duomenimis, dirbtiniu intelektu. Būtent todėl buvo pasirinkta *Tesseract-OCR* [23] programinės įrangos kūrimo biblioteka, kuri yra skirta būtent tokių projektų realizacijai. Ši biblioteka yra suderinama su daugybe duomenų mokslo bibliotekų ir šios platformos pakanka projektui visiškai realizuoti. Taip pat šią biblioteką labai lengva derinti su *PyCharm* įrankiu, skirtu darbui su *Python* programavimo kalba. Tai yra populiariausias ir patraukliausias šiuo metu rinkoje egzistuojantis darbui su duomenimis ir dirbtiniu intelektu sprendimas. Biblioteka taip pat buvo lyginama su alternatyvomis. Apžvelgus žemiau minimas platformas, buvo priimtas programinės įrangos kūrimo aplinkos sprendimas.

1. Viena iš mažiau žinomų alternatyvų yra *Keras* [24]. Teigiama, kad tai yra itin patogus įrankis, pavyzdžiui, tam, kad būtų sukurta nedidelis neuronų tinklas, pakanka vos poros kodo eilučių. Tačiau, šis įrankis turi itin aukšto lygio aplikacijų programavimo sąsają (angl. *Application Programming Interface* – API), vadinasi, kad atliekant sudėtingesnes užduotis šio įrankio naudojimas tampa žymiai sunkesnis. *Tesseract-OCR* yra labai panašus įrankis, tačiau sunkumai, su kuriais yra susiduriama naudojant *Keras*, yra pakankamai puikiai išspręstos. Taigi, šis įrankis yra netinkamas kūrėjams, su nedidele patirtimi dirbtinio intelekto srityje, tačiau šis įrankis gali būti itin naudingas šios srities profesionalams kaip pilnai funkcionuojanti alternatyva.
2. Kitas įrankis, kuris galėtų būti naudojamas kaip alternatyva – *Theano* [25]. Iš esmės, tai *Python* biblioteka, kuri yra itin gerai pritaikyta darbui su skaitinėmis reikšmėmis ir gilioju mokymusi. Tačiau *Theano* yra labai priklausoma nuo matematinių funkcijų bei turi daugybę panašumų (iš darbo su skaitinėmis reikšmėmis pusės) su *NumPy* ir *MatLab*. Vadinasi, norint realizuoti sudėtingesnį sprendimą, reikia šį įrankį naudoti kartu su kitais įrankiais, o tai nėra patogiu. Šis įrankis labai naudingas iš matematinės pusės (dėl funkcijų ir plačių galimybių). Taip pat, tai yra viena seniausių giliojo mokymosi bibliotekų, todėl šiomis dienomis galima rasti daug naujesnių ir patogesnių alternatyvų.
3. Dar viena alternatyva – *Torch* [26]. Tai itin greitas įrankis, kuris yra paremtas moduliais. Naudotojas gali labai lengvai pridėti reikiamus modulius arba ištrinti nereikalingus. Taip sprendimo valdymas tampa itin patogus ir labai nesudėtingai suprantamas. Didžiausias minusas šio įrankio yra realizacijos kalba – *Lua*. Šiomis dienomis didžioji dauguma duomenų mokslo sprendimų yra realizuojami *Python* programavimo kalba, o tai reiškia, kad sprendimai, realizuoti su *Lua*, nėra didelės vertės. *Lua* nėra populiarus, todėl pagalbos internete ieškant naudojimosi dokumentacijos ar pavyzdinių sprendimų problemai spręsti nėra labai daug. Tokie



sprendimai yra nenaudingi mąstant apie ateities perspektyvas, nes gerų specialistų, kurie galėtų palaikyti ir plėtoti sukurtą sprendimą nėra daug.

4. Populiareesnė alternatyva – *infer.NET* [27]. Tai *Microsoft* sukurtas įrankis. Šiuo įrankiu galima išspręsti daugybę dirbtinio intelekto problemų. Šio įrankio pilnai užtenka, kai yra dirbama su DI – nėra reikalinga papildomų įrankių integracija ar specifinių bibliotekų įtraukimas. Įrankis atrodo tikrai gera ir stipri alternatyva, tačiau šiuo metu tai yra uždaro kodo programinė įranga, o tai reiškia, kad naudotojai negali laisvai naudotis šiuo įrankiu. Įrankis gali būti naudojamas tik edukaciniais tikslais. Šiuo metu jokios kitos komercinės licencijos nėra suteikiamos. Būsimas projektas galėtų būti kuriamas edukaciniais tikslais, tačiau po sukūrimo projekto vertė taptų nuline, nes įrankis negalėtų būti naudojamas ateityje, o sprendimas taptų nebepalaikomu ir nenaudojamu.

Taigi, apžvelgus visas alternatyvas, yra matomas aiškus lyderis – *Tesseract-OCR*. Visos kitos alternatyvos turi labai stiprių pranašumų, tačiau kartu ir bent vieną nenuginčijamą minusą, kuris tampa kertinis mąstant apie kūrimo galimybes.

### **Antrasis etapas. RPA technologijos.**

Šiuo metu rinkoje egzistuoja daugybė įrankių, kurie specializuojasi į procesų automatizavimą. Su kiekvienais metais šių įrankių skaičius auga, nes auga ir paklausa procesų automatizavimui. Nors technologijų rinkoje yra labai daug pasirinkimų, tačiau robotizavimo srityje yra trys aiškūs lyderiai: *Blue Prism* [28], *UiPath* [29] ir *Automation Anywhere* [30].

*Automation Anywhere* įrankis yra labiau pagrįstas scenarijų rašymo principu, o *UiPath* ir *Blue Prism* turi vaizdo procesų dizainerius (angl. *drag and drop designers*). Tai reiškia, kad norint naudoti *Automation Anywhere*, reikia geresnių programavimo įgūdžių. *Automation Anywhere* ir *UiPath* turi makro komandų įrašymo įrenginius – funkciją, kuri pagreitina proceso atvaizdavimą, o *Blue Prism* įrankyje šios funkcijos nėra dėl pakankamai pasenusios technologijos. *Blue Prism* atstovai teigia, kad taip yra dėl saugumo apribojimų. Tačiau įrašymo įrenginys yra tik naudinga priemonė, kurios galima ir nenaudoti. Įrašytus veiksmus taip pat galima koreguoti ir keisti pagal asmeninius poreikius. Be to, yra specifinių skirtumų, susijusių su automatizavimo metodais. Renkantis įrankį svarbu peržiūrėti, koks procesas bus automatizuojamas, jo sudėtingumą bei specialistų, kurie implementuos procesą, patirtį. *UiPath* buvo sukurtas remiantis klientų išvalgomis, todėl galima teigti, kad klientams, kurie turi mažai žinių robotizavimo srityje, geriausia yra rinktis būtent šį įrankį dėl jo naudojimo paprastumo ir teikiamų plačių galimybių. Kognityvinių gebėjimų ir pakartotinio naudojimo požiūriu *Automation Anywhere* yra geriausias įrankis rinkoje. Kliento - serverio architektūra, lyginant su kitais įrankiais, yra daug geresnė naudojant *Blue Prism*. Taip pat šis įrankis renkamas, kai yra poreikis didelio masto robotų diegimui, nes turi labai gerą valdymo vartotojo sąsają ir yra atsparus trikdžiams [31].

RPA technologijos pasirinkimą lėmė saugumo, patikimumo ir naudojimo paprastumo kriterijai. Bendras visų pagrindinių automatizavimo įrankių palyginimas pateiktas žemiau (2.1 lentelė).

**2.1 lentelė** Automatizavimo įrankių apžvalga.

Kriterijus	Automation Anywhere	UiPath	Blue Prism
Reikalaujamos žinios	Geras scenarijų rašymo suvokimas	Grafinio dizainerio išmanymas	Grafinio dizainerio ir objektinio programavimo pagrindai

Kriterijus	Automation Anywhere	UiPath	Blue Prism
Pakartotinis programos dalių panaudojimas	Yra	Yra	Yra
Vidinė Talpa	Priklauso nuo duomenų bazės talpos	Iki 250 robotų	Priklauso nuo duomenų bazės talpos
Tikslumas	Pakankamai didelis visose terpėse	Labai didelis <i>Citrix</i> robotizavime, o kitur - vidutinis	Didelis visose terpėse
Robotai	Įvairūs	Įvairūs	Tik išorinių ofisų
Lygiagretumo galimybė	Stipriai apribota	Yra, tačiau nepatikima dideliuose projektuose	Yra ir labai padidina vykdymo greitį
Įrašymo funkcija	Yra	Yra	Nėra
Architektūra	Kliento - serverio	Paremta žiniatinkliu	Kliento - serverio
Prieinamumas	Tik siūlomoje kompiuterinėje aplikacijoje	Galima naudoti aplikacijoje, naršyklėje ar telefone	Tik siūlomoje kompiuterinėje aplikacijoje
Proceso kūrimas	Paremtas scenarijais	Grafinis dizaineris	Grafinis dizaineris
Pamatinė technologija	<i>Microsoft</i>	<i>Kibana, Elasticsearch, Microsoft - sharepoint</i>	C#
Patikimumas	Aukštas	Vidutinis	Labai aukštas
Kaina*	Didelės diegimo išlaidos	Maža kaina nedidelėms įmonėms, papildomai mokama už valdymo kambarį	Mokama didelė kaina už kiekvieną licenciją – kiek robotų, tiek ir licencijų
Mokymai ir sertifikavimas	Mokami mokymai, yra trys mokami sertifikatai	Teikiami nemokami mokymai, yra du nemokami sertifikatai	Mokami mokymai, yra 4+ skirtingų sričių sertifikatai

\* kaina derinama kiekvienam interesantui individualiai.

Įrankių apžvalgoje matome, kad didžiausią pranašumą pagal reikalaujamų žinių kiekį turi *UiPath*, nes reikia tik grafinio dizainerio išmanymo tam, kad būtų galima pradėti robotizuoti procesus naudojant šį įrankį. Kiti įrankiai reikalauja bent minimalių programavimo žinių. Visi įrankiai turi pakartotinio procesų ar objektų panaudojamumo galimybę, o tai reiškia, kad kuriant procesus nėra būtinas kūrimas nuo nulio – galima panaudoti jau sukurtas dalis iš kitų projektų. Vidinės talpos nėra ribojamos naudojant *Automation Anywhere* ir *Blue Prism*. Šie įrankiai to nekontroliuoja ir leidžia turėti tiek robotų, kiek telpa duomenų bazėje. Tačiau *UiPath* robotų skaičių riboja ir leidžia turėti ne daugiau kaip 250 robotų. Nors tai yra itin didelis skaičius, kurį net kai kurios didelės įmonės negeba pasiekti. Didžiausią tikslumą demonstruoja *Blue Prism*, tačiau nuo šio įrankio pagal šį kriterijų nedaug atsilieka *Automation Anywhere*, kuris turi pakankamai didelį tikslumą visose terpėse. *UiPath* tiksliausias *Citrix* aplinkų robotizavime, tačiau kitose terpėse tikslumas vidutinis. Taigi, jeigu yra poreikis robotizuoti procesą, kuris yra *Citrix* aplinkoje, ir tikslumas yra prioritetas, tai geriausias pasirinkimas būtų *UiPath*, o visais kitais atvejais kiti įrankiai. Robotai dažniausiai būna dviejų tipų – vidinių ir išorinių ofisų. *Automation Anywhere* ir *UiPath* siūlo robotizavimo galimybes abiejų tipų ofisams, o su *Blue Prism* galima kurti tik išorinių ofisų robotus. Taigi, renkantis *Blue Prism* būtina atkreipti į tai dėmesį ir pasidomėti, kokie procesai bus

automatizuojami. Vykdyto greičio srityje yra itin aktuali lygiagreto procesų vykdyto galimybė. *Blue Prism* šią galimybę turi ir ji labai padeda pagerinti procesų vykdyto laiką. *Automation Anywhere* šią galimybę stipriai riboja, o *UiPath* neriboja, tačiau ir nesuteikia didelės naudos – naudoti yra nepatikima. Įrašymo funkciją turi visi, išskyrus *Blue Prism*, įrankiai. Ši funkcija labai padeda procesų kūrime. Naudojama architektūra yra kliento – serverio. Vienintelis *UiPath* naudoja žiniatinkliu paremtą architektūrą. Būtent dėl šios priežasties *UiPath* galima naudoti tiek specialioje aplikacijoje, tiek naršyklėje, tiek mobiliajame įrenginyje, kai *Automation Anywhere* ir *Blue Prism* galima naudoti tik specialioje aplikacijoje. Kūrimo procese išsiskiria tik *Automation Anywhere*, kuriame procesai yra kuriami rašant scenarijus, o tai reikalauja papildomų programavimo žinių. *Blue Prism* ir *UiPath* įrankiuose procesai kuriami naudojant grafinį dizainerį ir galima visiškai išvengti programavimo. Skiriasi ir visų įrankių pamatinės technologijos, tačiau vyrauja ir vienas panašumas – *Microsoft (C#)*. Patikimumas visuose įrankiuose skirtingas – *Blue Prism* aukščiausias, *Automation Anywhere* šiek tiek mažesnis ir *UiPath* mažiausias. Mažiausios kainos lyderis – *UiPath*, tačiau visuose įrankiuose kaina derinama kiekvienam interesantui individualiai. *UiPath* taip pat siūlo ir nemokamus mokymus bei sertifikavimą, kai *Automation Anywhere* ir *Blue Prism* atveju šie dalykai yra mokami.

Įvertinus visus kriterijus, buvo nuspręsta rinkti *Blue Prism* įrankį, nes šis turi pranašiausią architektūrą, yra saugiausias ir ganėtinai paprastas naudoti (nereikalauja didelių programavimo žinių). Sukurtas produktas planuojamas naudoti dideliu mastu, todėl programos vykdyto stabilumas, greitis bei kokybė yra taip pat itin svarbūs kriterijai, kurie parodo geriausias rezultatus *Blue Prism* įrankio pusėje.

### 2.3. Egzistuojančių rinkoje programų sistemų savybių kiekybinis ir kokybinis palyginimas

Žemiau yra pateikiamas egzistuojančių rinkoje programų sistemų savybių kiekybinis ir kokybinis palyginimas (žr. 2.2 lentelė).

2.2 lentelė Programų sistemų palyginimas

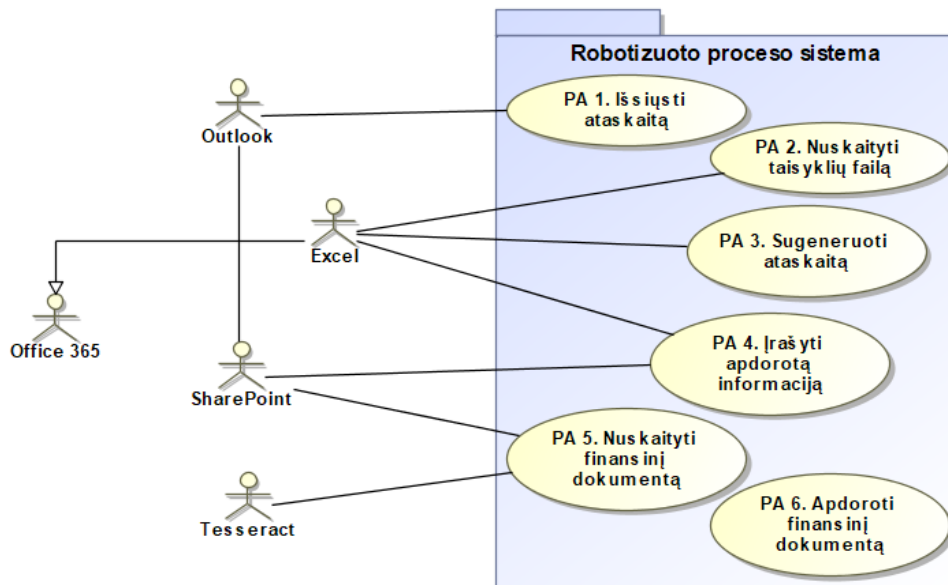
Kriterijus/Programa	Nanonets	Conexiom	Klippa	Logikia	FastFour
Patogumas	aukštas	žemas	vidutinis	vidutinis	vidutinis
Greitis	didelis	mažas	mažas	vidutinis	didelis
Populiarumas	50+	100+	500+	100+	100+
Kaina	Nemokamas, pro versija 500\$/mėn	Pradinė kaina 1\$/metus	Nemokama demo versija, 180-1800\$/mėn	Nepateikiama	Diegimo mokestis 1500\$, 0.3\$ už apdorotą dokumentą
Prieinamumas	Tik Naršyklėje	Tik Naršyklėje	Naršyklėje ir išmaniuose įrenginiuose	Naršyklėje ir išmaniuose įrenginiuose	Tik Naršyklėje
Pagalba	24/7	24/7	24/7	Tik darbo valandomis	Tik el. laiškais
Mokymai	Virtualūs	Individualūs	Individualūs/virtualūs	Nėra	Kelios virtualios sesijos
Dokumentacija	Yra	Yra	Yra	Yra	Yra

Pateikti sprendimai, egzistuojantys rinkoje, yra ganėtinai panašūs – visi automatizuoja dokumentų (daugiausiai finansinių) apdorojimą, visi naudoja OCR technologiją ir teigia, kad yra naudojami konvoliuciniai neuroniniai tinklai, tačiau šio fakto patikrinti negalima, nes prieiga prie įrankių išėties tekstų negalima. Nors iš pirmo žvilgsnio šie įrankiai gali pasirodyti vienodi – atliekantys tą pačią funkciją – tačiau atskirai visi turi savų pranašumų ir savų minusų. Naudotojai yra linkę vertinti pirmąjį išpūdį, todėl įrankių patogumas yra vienas iš svarbiausių aspektų. Šiuo aspektu „Nanonets“ yra lyderis, kuris suteikia itin patogią ir kiekvienam naudotojui suprantamą naudotojo sąsają. Naudotojo sąsajos intuityvumu atsilieka „Klippa“, „Logikia“ ir „FastFour“, tačiau naudotojai teigia, kad sąsaja yra pakankamai intuityvi ir suprantama – yra kur tobulėti, tačiau dirbti eina. Prasčiausią naudotojo sąsają turi „Conexiom“. Naudotojai teigia, kad šio įrankio sąsaja yra sunkiai suprantama ir, pradėjus dirbti su šiuo įrankiu, kyla daugybė supratimo problemų, kurių patys vartotojai dažnai negeba išspręsti. Greitis yra svarbus, tačiau dažnai ne kritinis kriterijus. Problemos dėl greičio turi įtakos verslui tada, kai turi būti apdoroti itin dideli kiekiai dokumentų. Tokiu atveju greitis gali tapti itin didele kliūtimi įrankio naudojimui. Didžiausią dokumentų apdorojimo greitį turi „Nanonets“ ir „FastFour“, vidutiniškai greitai dokumentai yra apdorojami „Logikia“ įrankiu, o lėtai – „Conexiom“ ir „Klippa“ įrankiais. Populiarumo kriterijus nurodo naudotojų pasitikėjimą įrankiu. Galima teigti, kad kuo daugiau asmenų ar įmonių naudoja įrankį, tuo patrauklesnis ir efektyvesnis įrankis atrodo rinkoje. Populiariausias tarp tiriamų įrankių su daugiau kaip penkiais šimtais klientų yra „Klippa“, o visi kiti įrankiai populiarumu yra ganėtinai panašūs (turi apytiksliai daugiau nei šimtą klientų), tačiau „Nanonets“ populiarumas mažiausias (šiek tiek daugiau nei penkiasdešimt klientų). Taip yra todėl, kad šis įrankis yra ganėtinai naujas rinkoje ir dar nėra pritraukęs daug klientų. Naudotojams svarbus bei dažnai kritinis kriterijus yra kaina. Visų įrankių kainos skiriasi ir yra matomi dveji mokėjimo planai. Vieni įrankiai yra perkami taikant vienkartinį mėnesio ar metų planus, kiti (šio tyrimo atveju tik „FastFour“) taiko vienkartinį diegimo mokestį, o vėliau apmokestina kiekvieną apdorotą dokumentą. „Nanonets“ įrankis yra vienintelis, kuris leidžia pabandyti įrankį nemokamai. Kiti įrankiai to padaryti neleidžia arba reikia tiesiogiai kontaktuoti su atsakingais asmenimis, kad šie suteiktų demonstracinę versiją. Įrankių kainos yra skirtingai apskaičiuojamos nuo pasirinkto plano, apdorojamų dokumentų kiekio ir pardavimų logikos, todėl sunku nustatyti, kuris įrankis yra pigiausias, o kuris brangiausias. Aišku tik tai, kad „Nanonets“ vieninteliai, kurie leidžia nemokamai ir nedideliais kiekiais dokumentų išbandyti jų sukurtą produktą. Prieinamumas yra mažiau svarbus kriterijus, nes šiuolaikiniame versle darbas dažniausiai vyksta kompiuteriu, tačiau tai pasako apie įrankio infrastruktūrą ir pasiekiamumą. Visi nagrinėjami įrankiai yra pasiekiami naršyklėje. „Klippa“ ir „Logikia“ taip pat yra pasiekiami ir išmaniuosiuose įrenginiuose. Tai suteikia nedidelį konkurencinį pranašumą prieš kitus įrankius. Dauguma nagrinėjamų įrankių („Nanonets“, „Conexiom“, „Klippa“) turi labai gerą ir išvystytą pagalbos sektorių, kuris veikia kiekvieną dieną visą parą. Prasčiausią pagalbą klientams teikia „FastFour“, kuriuos galima pasiekti tik elektroniniais laiškais. „Logikia“ teikia įvairiapusę pagalbą, tačiau tik darbo valandomis. Tai yra kanadiečių sukurtas įrankis, todėl naudotojams Europoje ar Azijoje gali kilti bėdų bandant sulaukti pagalbos (dėl laiko skirtumų). Mokymai yra labai svarbus kriterijus naudotojams, kurie dar tik pradeda naudotis įrankiu. Visi nagrinėjami įrankiai, išskyrus „Logikia“, suteikia mokymų galimybę individualiai arba virtualiai. Visi nagrinėjami įrankiai turi prieinamą naudojimosi bei veikimo dokumentaciją.

### 3. PROJEK TINĖ DALIS

#### 3.1. Panaudojimo atvejų vaizdas

Žemiau yra pateikiama panaudojimo atvejų diagrama (žr. 3.1 pav.). Kiekvienas panaudos atvejis (žr. 3.1 lentelė - 3.6 lentelė) yra detalizuojamas 3.1.1. skyriuje.



3.1 pav. Projekto panaudojimo atvejų diagrama (PAM)Panaudojimo atvejų sąrašas

#### 3.1.1. Panaudojimo atvejų sąrašas

3.1 lentelė PA 1. Išsiųsti ataskaitą detalioji specifikacija

<b>PA.</b> Išsiųsti ataskaitą.		
<b>Tikslas.</b> Išsiųsti atliktų darbų ataskaitą suinteresuotiems asmenims.		
<b>Aprašymas.</b> Ši ataskaita yra skirta sekti, kaip buvo vykdomas procesas. Trumpai tariant, tai ataskaita, kuri nurodo kaip procesas buvo įvykdytas, o ne ką procesas atliko. Ataskaita yra išsiunčiama suinteresuotiems asmenims. Jeigu procesas baigė darbą su sisteminė klaida, tai yra siunčiamas klaidos pranešimas ir aprašymas už procesą atsakingiems asmenims.		
<b>Prieš sąlyga:</b>	<i>Outlook</i> programa aktyvi.	
<b>Aktorius:</b>	Robotizuoto proceso sistema ir <i>Outlook</i> programa.	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>	-
	<b>Apimami PA</b>	-
	<b>Specializuojami PA</b>	-
<b>Po sąlyga:</b>	Sistema sėkmingai išsiunčia ataskaitą.	
<b>Pagrindinis scenarijus:</b>	<ol style="list-style-type: none"> <li>1. Sistema sugeneruoja elektroninio laiško temą ir testą.</li> <li>2. Sistema prideda gavėjus ir anksčiau sugeneruotą ataskaitą.</li> <li>3. Sistema aktyvuoja siuntimo veiksmą, <i>Outlook</i> programa išsiunčia laišką su ataskaitos priedu gavėjams.</li> </ol>	
<b>Alternatyvus scenarijus:</b>	1. Sisteminės klaidos atveju, sistema sugeneruoja	

	<p>klaidos temą ir pranešimą (tekstą).</p> <ol style="list-style-type: none"> <li>2. Sistema prideda klaidos pranešimo gavėjus.</li> <li>3. Sistema aktyvuoja siuntimo veiksmą, <i>Outlook</i> programa išsiunčia laišką.</li> </ol>
--	--

### 3.2 lentelė PA 2. Nuskaityti taisyklių failą detalioji specifikacija

<b>PA.</b> Nuskaityti taisyklių failą.		
<b>Tikslas.</b> Taisyklių faile esančią informaciją perkelti į programos kolekcijos kintamąjį.		
<b>Aprašymas.</b> Nuskaityti failą, dažnai duomenys būna netvarkingi (dėl žmogiškųjų klaidų faktoriaus atsiranda praleistų eilučių, nenumatytų simbolių). Programinės įrangos robotas turi taisyklingai sutvarkyti duomenis, t.y. pašalinti tuščias eilutes, nereikalingus simbolius, suklasifikuoti taisykles bei parengti taisyklių kolekciją, kuria naudojantis bus tikrinami finansiniai failai.		
<b>Prieš sąlyga:</b>	<i>Excel</i> programa aktyvi.	
<b>Aktorius:</b>	Robotizuoto proceso sistema ir <i>Excel</i> programa.	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>	-
	<b>Apimami PA</b>	-
	<b>Specializuojami PA</b>	-
<b>Po sąlyga:</b>	Sistema sėkmingai nuskaityti taisyklių failą.	
<b>Pagrindinis scenarijus:</b>	<ol style="list-style-type: none"> <li>1. Sistema inicijuoja <i>.xlsx</i> formato taisyklių failo, patalpinto lokaliai kompiuteryje, skaitymą.</li> <li>2. <i>Excel</i> programa grąžina faile esančią informaciją.</li> <li>3. Sistema pašalina tuščias eilutes, nereikalingus simbolius, tarpus.</li> <li>4. Sistema suklasifikuoja taisykles ir įrašo į sistemos taisyklių rinkinį.</li> </ol>	
<b>Alternatyvus scenarijus:</b>	<ol style="list-style-type: none"> <li>2. <i>Excel</i> programa nėra aktyvi. Visas programos vykdymas stabdomas.</li> </ol>	

### 3.3 lentelė PA 3. Sugeneruoti ataskaitą detalioji specifikacija

<b>PA.</b> Sugeneruoti ataskaitą.		
<b>Tikslas.</b> Sugeneruoti atliktų darbų žurnalą ir išsaugoti faile.		
<b>Aprašymas.</b> Ši ataskaita yra skirta sekti, kaip buvo vykdomas procesas. Trumpai tariant, tai ataskaita, kuri nurodo kaip procesas buvo įvykdytas, o ne ką procesas atliko. Ataskaita taip pat yra įrašoma ir išsaugoma <i>.xlsx</i> formato faile.		
<b>Prieš sąlyga:</b>	<i>Excel</i> programa aktyvi.	
<b>Aktorius:</b>	Robotizuoto proceso sistema ir <i>Excel</i> programa.	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>	-
	<b>Apimami PA</b>	-
	<b>Specializuojami PA</b>	-
<b>Po sąlyga:</b>	Sistema sėkmingai sugeneruoja ir įrašo ataskaitą.	
<b>Pagrindinis scenarijus:</b>	<ol style="list-style-type: none"> <li>1. Sistema apdoroja programos vykdymo žurnalą.</li> <li>2. Sistema suklasifikuoja atliktus darbus pagal baigties statusą.</li> <li>3. Sistema sugeneruoja lentelę, su visai duomenimis, ir įrašo į sistemos kolekciją.</li> <li>4. Sistema inicijuoja kolekcijos įrašymą į <i>Excel</i></li> </ol>	

	programos failą. 5. <i>Excel</i> programa grąžina įrašymo statusą.
<b>Alternatyvus scenarijus:</b>	5. <i>Excel</i> programa nėra aktyvi. Visas programos vykdymas stabdomas.

### 3.4 lentelė PA 4. Įrašyti apdorotą informaciją detalioji specifikacija

<b>PA.</b> Įrašyti apdorotą informaciją.		
<b>Tikslas.</b> Įrašyti proceso vykdymo rezultatus į failą.		
<b>Aprašymas.</b> Funkcija išsiskiria tuo, kad, prieš įrašant į failą, programinės įrangos robotas turi pertvarkyti kolekciją taip, kad ji būtų suprantama fiziniam asmeniui. Šis failas yra programos išvestis, todėl privalo būti tvarkingas ir suprantamas. Pertvarkymas vyksta sudedant suprantamus stulpelių pavadinimus, sunumeruojant eilutes, sulygiuojant tekstą ir pan. Realiai atliekamos visos funkcijos su failo redagavimu tam, kad būtų padidintas skaitomumas.		
<b>Prieš sąlyga:</b>	<i>Excel</i> programa aktyvi.	
<b>Aktorius:</b>	Robotizuoto proceso sistema, <i>Excel</i> programa ir <i>SharePoint</i> repozitorija.	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>	-
	<b>Apimami PA</b>	-
	<b>Specializuojami PA</b>	-
<b>Po sąlyga:</b>	Sistema sėkmingai įrašo informaciją.	
<b>Pagrindinis scenarijus:</b>	<ol style="list-style-type: none"> <li>1. Sistema apdoroja atvejo vykdymo rezultatus.</li> <li>2. Sistema pašalina nereikalingas eilutes, perteklinius tarpus, nelogiškus duomenis.</li> <li>3. Sistema inicijuoja kolekcijos įrašymą į <i>Excel</i> programos failą.</li> <li>4. <i>Excel</i> programa grąžina įrašymo statusą.</li> <li>5. Sistema inicijuoja <i>.xlsx</i> failo įrašymą į subendrintą <i>SharePoint</i> repozitoriją.</li> <li>6. Rezultatai įrašomi į subendrintą repozitoriją.</li> </ol>	
<b>Alternatyvus scenarijus:</b>	4. <i>Excel</i> programa nėra aktyvi. Visas programos vykdymas stabdomas.	
<b>Alternatyvus scenarijus:</b>	6. Nėra ryšio su <i>SharePoint</i> repozitorija. Įrašymo veiksmas praleidžiamas.	

### 3.5 lentelė PA 5. Nuskaityti finansinį dokumentą detalioji specifikacija

<b>PA.</b> Nuskaityti finansinį dokumentą		
<b>Tikslas.</b> Nuskaityti finansiniame dokumente esančią informaciją.		
<b>Aprašymas.</b> Programinės įrangos robotas, naudojantis <i>Tesseract-OCR</i> varikliu, turi nuskaityti finansiniame dokumente, kuris yra patalpintas <i>SharePoint</i> repozitorijoje, esančią informaciją (visą tekstą) vėlesniam apdorojimui.		
<b>Prieš sąlyga:</b>	<i>SharePoint</i> repozitorija pasiekiamas.	
<b>Aktorius:</b>	Robotizuoto proceso sistema, <i>Tesseract</i> variklis ir <i>SharePoint</i> repozitorija.	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>	-
	<b>Apimami PA</b>	-
	<b>Specializuojami PA</b>	-

<b>Po sąlyga:</b>	Sistema sėkmingai nuskaityto finansinį dokumentą.
<b>Pagrindinis scenarijus:</b>	<ol style="list-style-type: none"> <li>1. Sistema inicijuoja <i>.pdf</i> formato failo nuskaitymą iš <i>SharePoint</i> repozitorijos.</li> <li>2. <i>SharePoint</i> repozitorija grąžina failą.</li> <li>3. Sistema perduoda failą <i>Tesseract-OCR</i> varikliui.</li> <li>4. <i>Tesseract-OCR</i> variklis grąžina <i>.pdf</i> faile esantį tekstą.</li> </ol>
<b>Alternatyvus scenarijus:</b>	2. Nėra ryšio su <i>SharePoint</i> repozitorija. Visas programos vykdymas stabdomas.

3.6 lentelė PA 6. Apdoroti finansinį dokumentą detalioji specifikacija

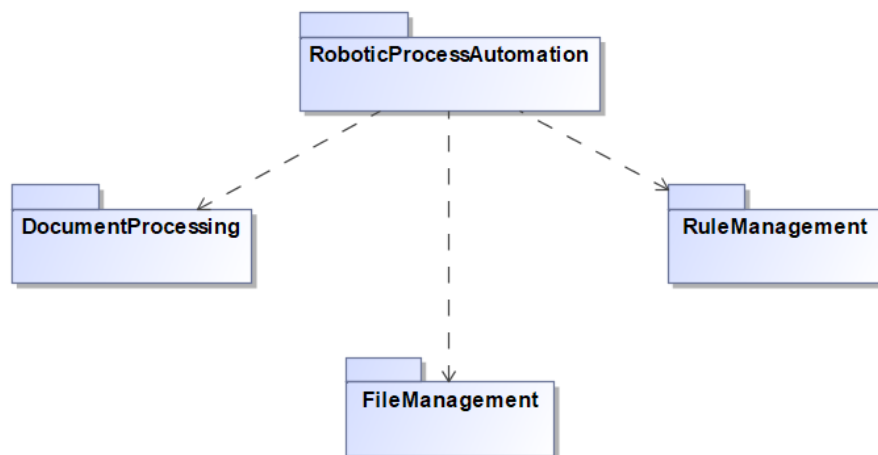
<b>PA. Apdoroti finansinį dokumentą</b>		
<b>Tikslas.</b> Rasti reikiamą informaciją naudojant finansinio dokumento tekstą ir taisykles.		
<b>Aprašymas.</b> Programinės įrangos robotas, turint teisingą taisyklių kolekciją bei nuskaitytą finansinį dokumentą, gali ieškoti specifinės informacijos. Surasti duomenys išsaugomi atskiroje kolekcijoje, kuri vėliau bus įrašoma į failą kaip programos išvestis (rezultatas). paieška finansiniame dokumente vyksta naudojant <i>Regex</i> technologiją. Pilnas finansinis dokumentas būna tarsi vientisas tekstas. Naudojant <i>Regex</i> ir turimas taisykles, programinės įrangos robotas gali greitai rasti informaciją tekstiname kintamajame (finansiniame dokumente).		
<b>Prieš sąlyga:</b>	Turimas finansinio dokumento tekstas ir nuskaitytos taisyklės.	
<b>Aktorius:</b>	Robotizuoto proceso sistema.	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>	-
	<b>Apimami PA</b>	-
	<b>Specializuojami PA</b>	-
<b>Po sąlyga:</b>	Sistema sėkmingai apdoroja finansinį dokumentą.	
<b>Pagrindinis scenarijus:</b>	<ol style="list-style-type: none"> <li>1. Sistema filtruoja dokumento tekstą tol, kol visi filtrai būna įvykdyti.</li> <li>2. Sistema įrašo išfiltruotus rezultatus į atskirą rezultatų kolekciją,</li> </ol>	
<b>Alternatyvus scenarijus:</b>	-	

### 3.2. Sistemos statinis vaizdas

#### 3.2.1. Apžvalga

Kuriamą įrankį galima išskaidyti į keturis atskirus paketus pagal kiekvieno paketo atliekamos veiklos prasmę (žr. 3.2 pav.). Detaliau šie paketi yra analizuojami 3.2.2. skyriuje.

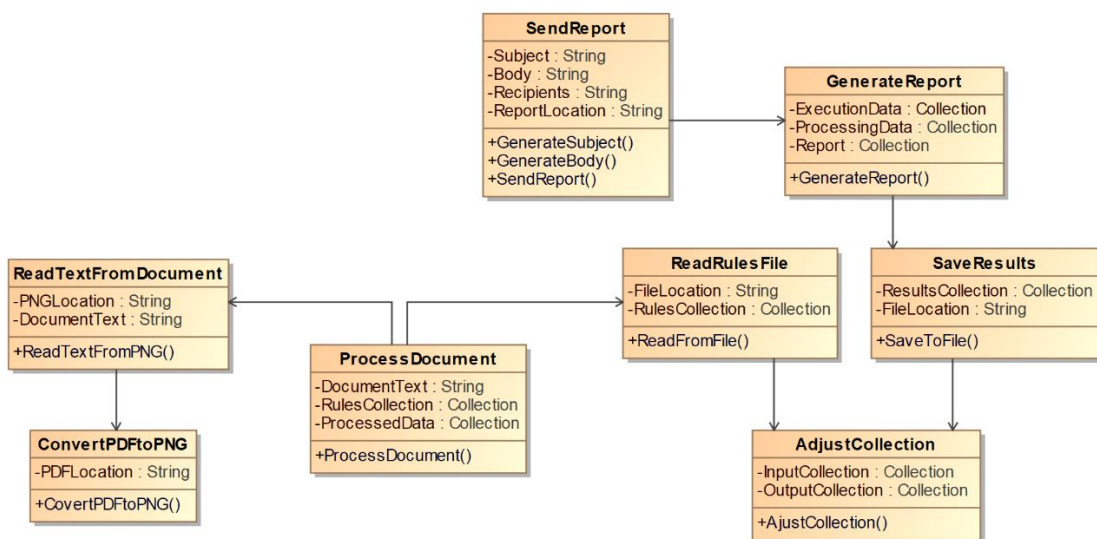




3.2 pav. Sistemos paketų diagrama

### 3.2.2. Paketų detalizavimas

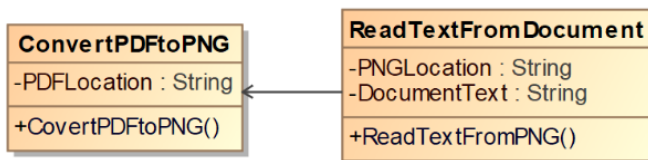
Žemiau esančioje diagramoje (žr. 3.3 pav.) galime matyti, kaip visų paketų klasės jungiasi tarpusavyje. Tai projekto klasių diagrama, kurioje galima matyti pagrindines klases, jų tarpusavio ryšius bei struktūrą. Savaimė suprantama, kad projekte yra daugiau klasių, tačiau diagramoje nurodytos klasės išlieka kaip esminės, tokios be kurių neitų sukurti projekto.



3.3 pav. Projekto klasių diagrama

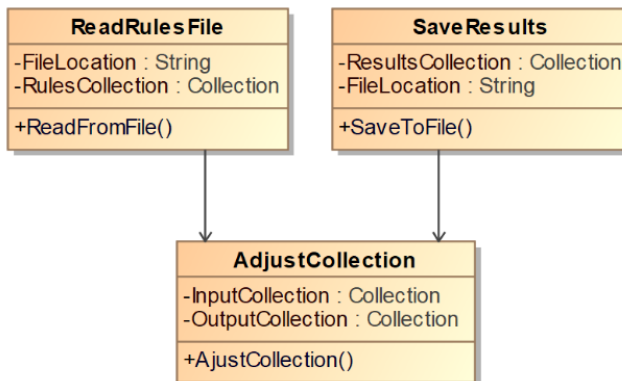
Pakete *DocumentProcessing* yra atliekami visi veiksmai, susiję su dirbtinio intelekto taikymu. Tai teksto skaitymo iš *.pdf* formato dokumento veiksmai. Tekstas iš šių dokumentų nuskaitymas dirbtinio intelekto pagalba, todėl visi veiksmai, susiję su dirbtiniu intelektu, yra talpinami šiame pakete.

Žemiau yra pateikiama šio paketo klasių diagrama (žr. 3.4 pav.).



3.4 pav. *DocumentProcessing* paketo klasių diagrama

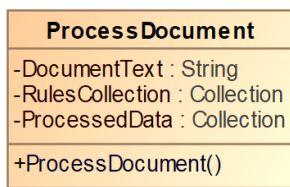
Pakete *FileManagement* yra atliekami visi veiksmai, susiję su failų valdymu. Tai susideda iš įvesties failų skaitymo (taisyklių failas) ir išvesties failų sudarymo bei įrašymo (darbų ataskaita, rezultatų failas). Žemiau yra pateikiama šio paketo klasių diagrama (žr. 3.5 pav.).



3.5 pav. *FileManagement* paketo klasių diagrama

Pakete *RuleManagement* yra atliekami visi veiksmai, susiję su dokumento apdorojimu naudojant taisyklių faile aprašytas taisykles. Tai vieta, kurioje, turint visus įvesties duomenis, gaunami išvesties duomenys, rezultatas.

Žemiau yra pateikiama šio paketo klasių diagrama (žr. 3.6 pav.).

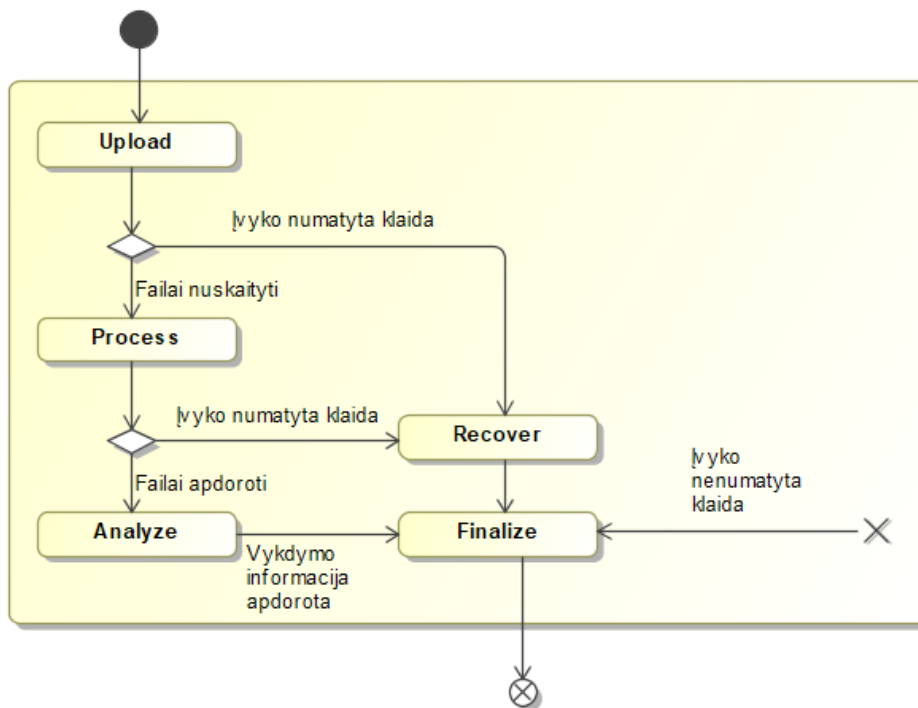


3.6 pav. *RuleManagement* paketo klasių diagrama

### 3.3. Sistemos dinaminis vaizdas

#### 3.3.1. Būsenų diagrama

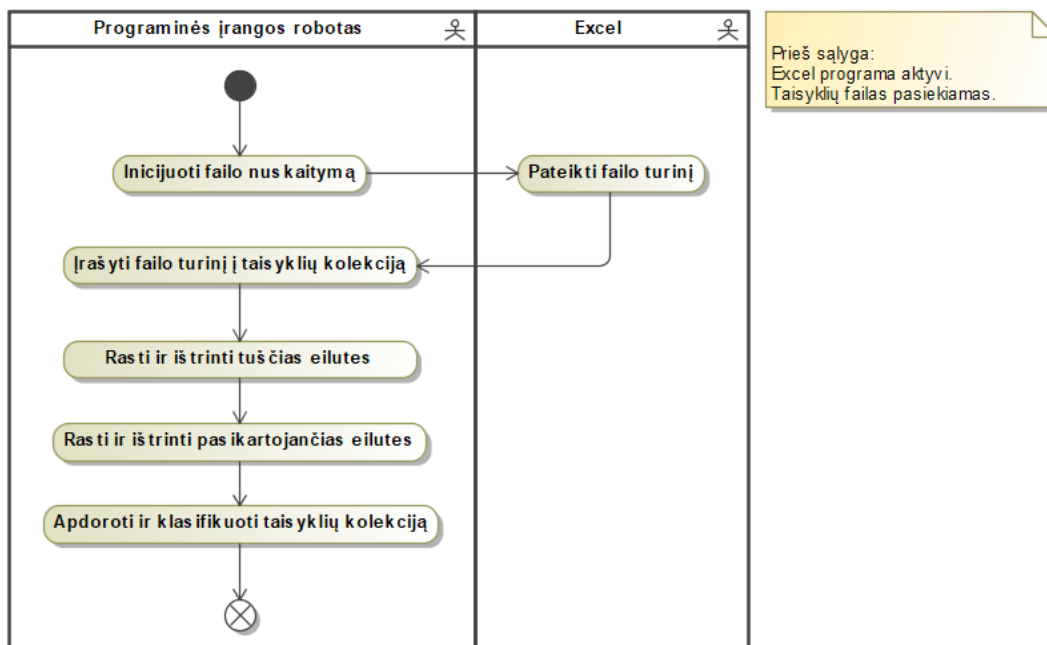
Žemiau yra pateikiama sistemos būsenų diagrama (žr. 3.7 pav.). Galima matyti, kad sistema pradėjusi darbą būna pradinėje „įkėlimo“ (angl. upload) fazėje. Tuo metu programa užkrauna į sistemą įvesties failus (taisyklių failą bei finansinius dokumentus). Kai failai yra sėkmingai užkrauti, sistema pereina į sekančią būseną – vykdymo (angl. process). Šioje fazėje yra apdorojami visi finansiniai dokumentai pagal taisyklių faile nurodytas taisykles. Kai failai yra sėkmingai apdoroti, sistema pereina į sekančią būseną – analizavimo (angl. analyze). Šioje fazėje yra analizuojamas programos veikimas – kiek dokumentų buvo apdorota, ką pavyko rasti, kas nebuvo surasta ir pan. Galiausiai seka užbaigimo būseną (angl. finalize), kurioje yra generuojami rezultatų failai bei veiklos ataskaita, kuri yra siunčiama suinteresuotiems asmenims, ir programa baigia darbą. Įvykus numatytai klaidai programa pereina į atsistatymo būseną (angl. recover), kurioje bandoma atstatyti įprastinį programos vykdymą ir tęsti darbus. Įvykus nenumatytai klaidai programa pereina į užbaigimo būseną, kurioje išsiunčia pranešimą su klaidos tipu bei detalėmis už procesą atsakingiems asmenims ir programa baigia darbą.



3.7 pav. Projekto būsenų diagrama

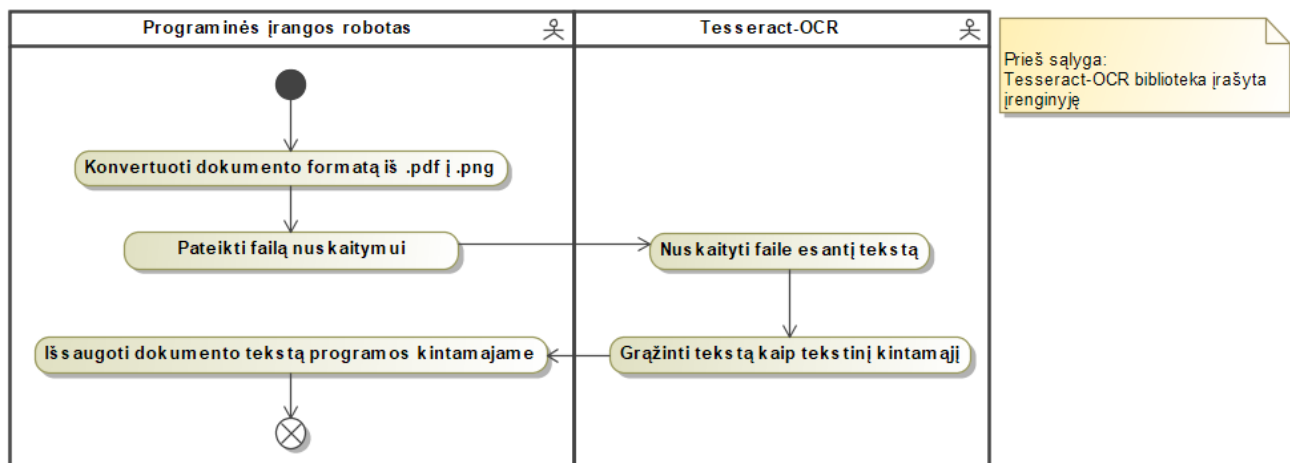
### 3.3.2. Veiklos diagramos

Žemiau yra pateikiama taisyklių failo nuskaitymo panaudos atvejo veiklos diagrama (žr. 3.8 pav.). Čia matome, kad programinės įrangos robotas arba sistema inicijuoja failo nuskaitymą. *Excel* programa gražina failo turinį. Galiausiai, sistema sutvarko taisykles ir paruošia naudojimui, t.y. įrašo failo turinį į taisyklių kolekciją, randa ir ištrina tuščias eilutes, pasikartojančias eilutes bei apdoroja ir klasifikuoja taisyklių kolekciją. *Excel* programa privalo būti aktyvi ir taisyklių failas privalo būti pasiekiamas tam, kad ši veikla galėtų būti vykdoma.



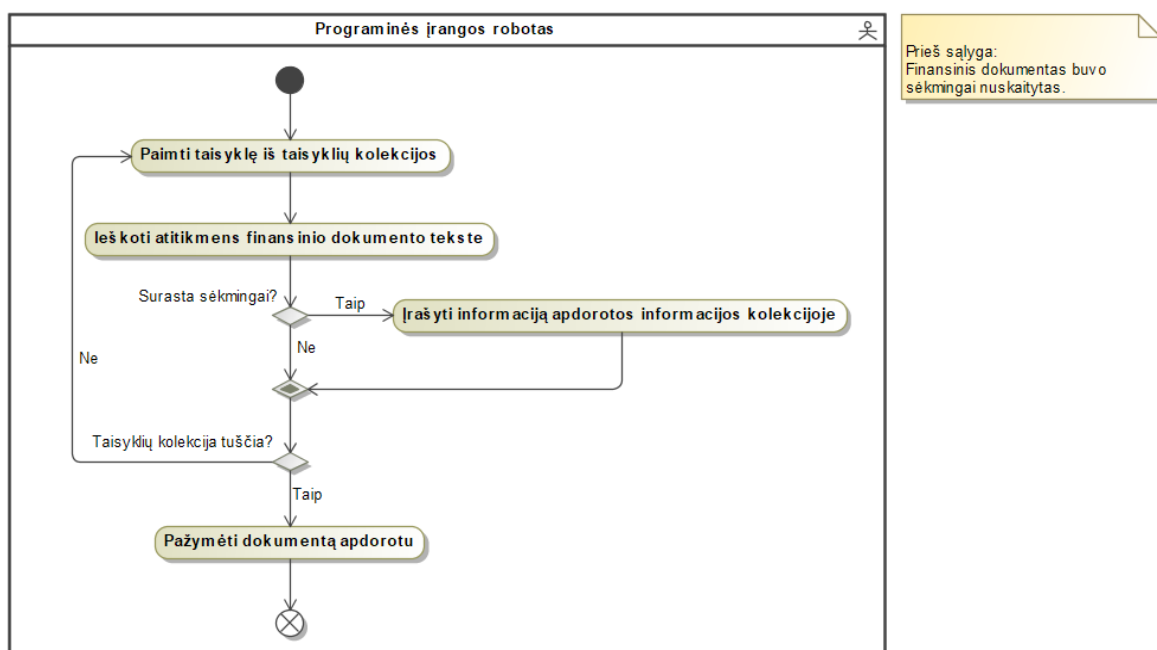
3.8 pav. Taisyklių failo nuskaitymo veiklos diagrama

Žemiau yra pateikiama finansinio dokumento nuskaitymo panaudos atvejo veiklos diagrama (žr. 3.9 pav.). Čia matome, kad programinės įrangos robotas arba sistema konvertuoja dokumento formatą iš *.pdf* į *.png* ir pateikia failą nuskaitymui. *Tesseract-OCR* variklis nuskaityto faile esantį tekstą ir grąžina jį kaip tekstinį kintamąjį. Sistema išsaugo dokumento tekstą programos kintamajame (vidinėje atmintyje). *Tesseract-OCR* variklis – biblioteka privalo būti įrašytas įrenginyje.



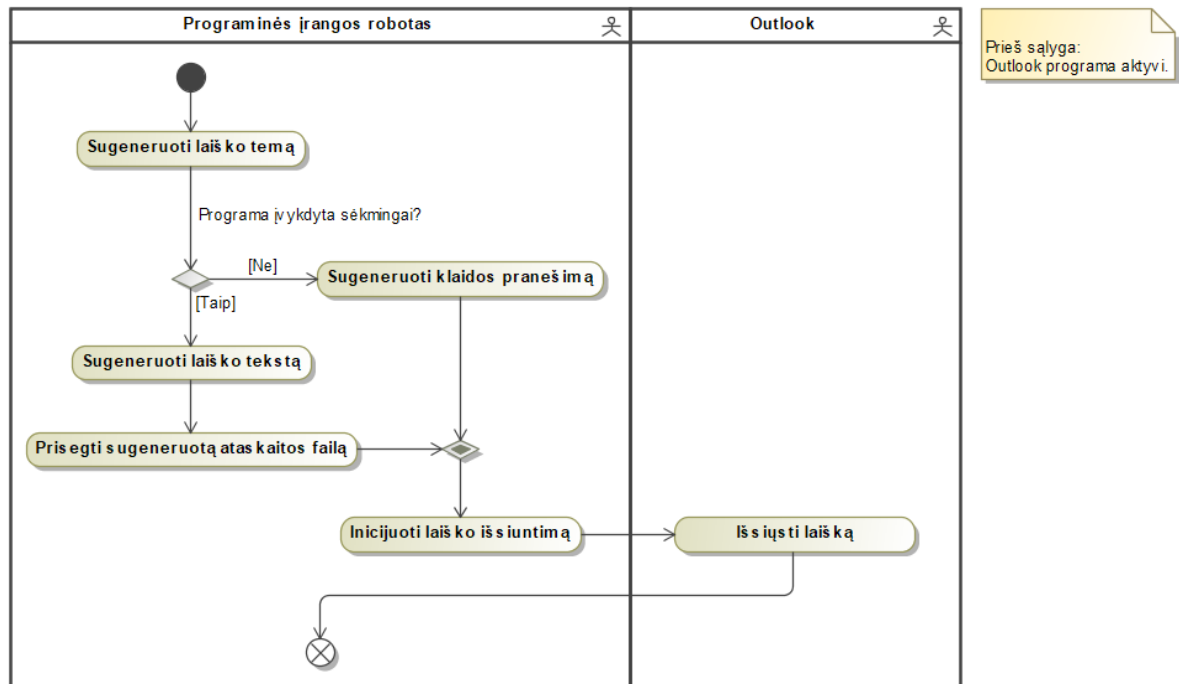
3.9 pav. Finansinio dokumento nuskaitymo veiklos diagrama

Žemiau yra pateikiama finansinio dokumento apdorojimo panaudos atvejo veiklos diagrama (žr. 3.10 pav.). Čia matome, kad programinės įrangos robotas arba sistema paima taisyklę iš taisyklių kolekcijos ir ieško atitikmens finansinio dokumento tekste. Jeigu atitikmuo buvo surastas sėkmingai, tai surasta informacija yra įrašoma į apdorotos informacijos kolekciją. Toliau sistema ima taisykles vieną po kitos ir ieško atitikmenų finansiniame dokumente kol visos taisyklės būna paimitos. Galiausiai, finansinis dokumentas yra pažymimas apdorotu. Finansinis dokumentas privalo būti sėkmingai nuskaitytas prieš vykdant šią veiklą.



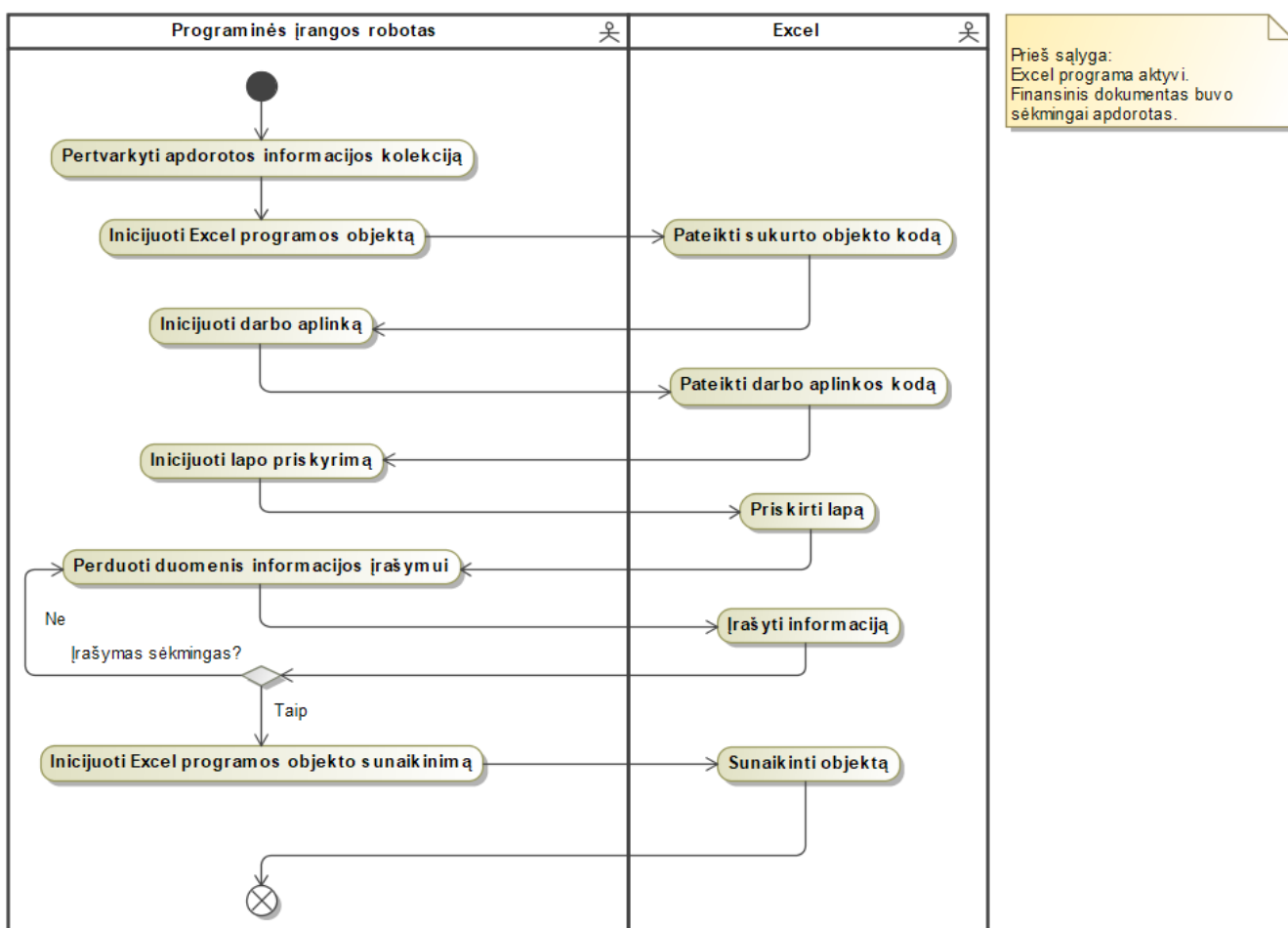
3.10 pav. Finansinio dokumento apdorojimo veiklos diagrama

Žemiau yra pateikiama ataskaitos išsiuntimo panaudos atvejo veiklos diagrama (žr. 3.11 pav.). Čia matome, kad programinės įrangos robotas arba sistema sugeneruoja laiško temą. Jeigu programa buvo įvykdyta sėkmingai, tai yra sugeneruojamas laiško tekstas ir prisegamas sugeneruotos ataskaitos failas. Jeigu programa nebuvo įvykdyta sėkmingai, tai yra sugeneruojamas klaidos pranešimas. Galiausiai, sistema inicijuoja laiško išsiuntimą. Outlook programa išsiunčia laišką. Outlook programa privalo būti aktyvi tam, kad ši veikla galėtų būti vykdoma.



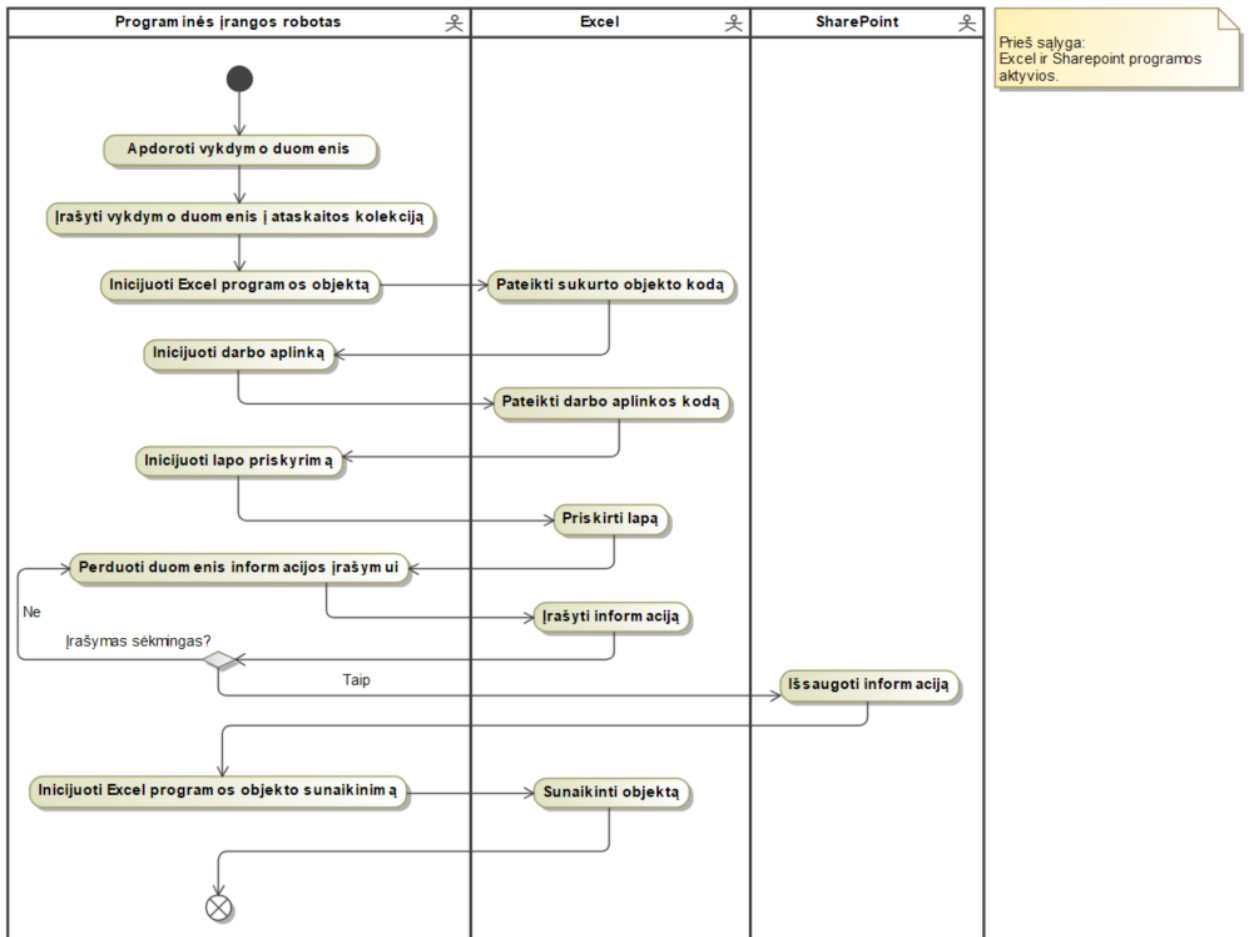
3.11 pav. Ataskaitos išsiuntimo veiklos diagrama

Žemiau yra pateikiama apdorotos informacijos įrašymo panaudos atvejo veiklos diagrama (žr. 3.12 pav.). Čia matome, kad programinės įrangos robotas arba sistema pertvarko apdorotos informacijos kolekciją bei inicijuoja Excel programos objektą. Excel programa grąžina sukurto objekto kodą. Sistema inicijuoja darbo aplinką. Excel programa grąžina darbo aplinkos kodą. Sistema inicijuoja lapo priskyrimą. Excel programa priskiria lapą. Sistema perduoda informaciją įrašymui. Excel programa įrašo informaciją ir grąžina įrašymo statusą. Jeigu įrašyti nepavyko, tai sistema bando kartoti įrašymą dar kartą. Galiausiai, programa inicijuoja Excel programos objekto sunaikinimą. Excel programa sunaikina objektą. Excel programa privalo būti aktyvi ir finansinis dokumentas privalo būti sėkmingai apdorotas tam, kad ši veikla galėtų būti vykdoma.



3.12 pav. Apdorotos informacijos įrašymo veiklos diagrama

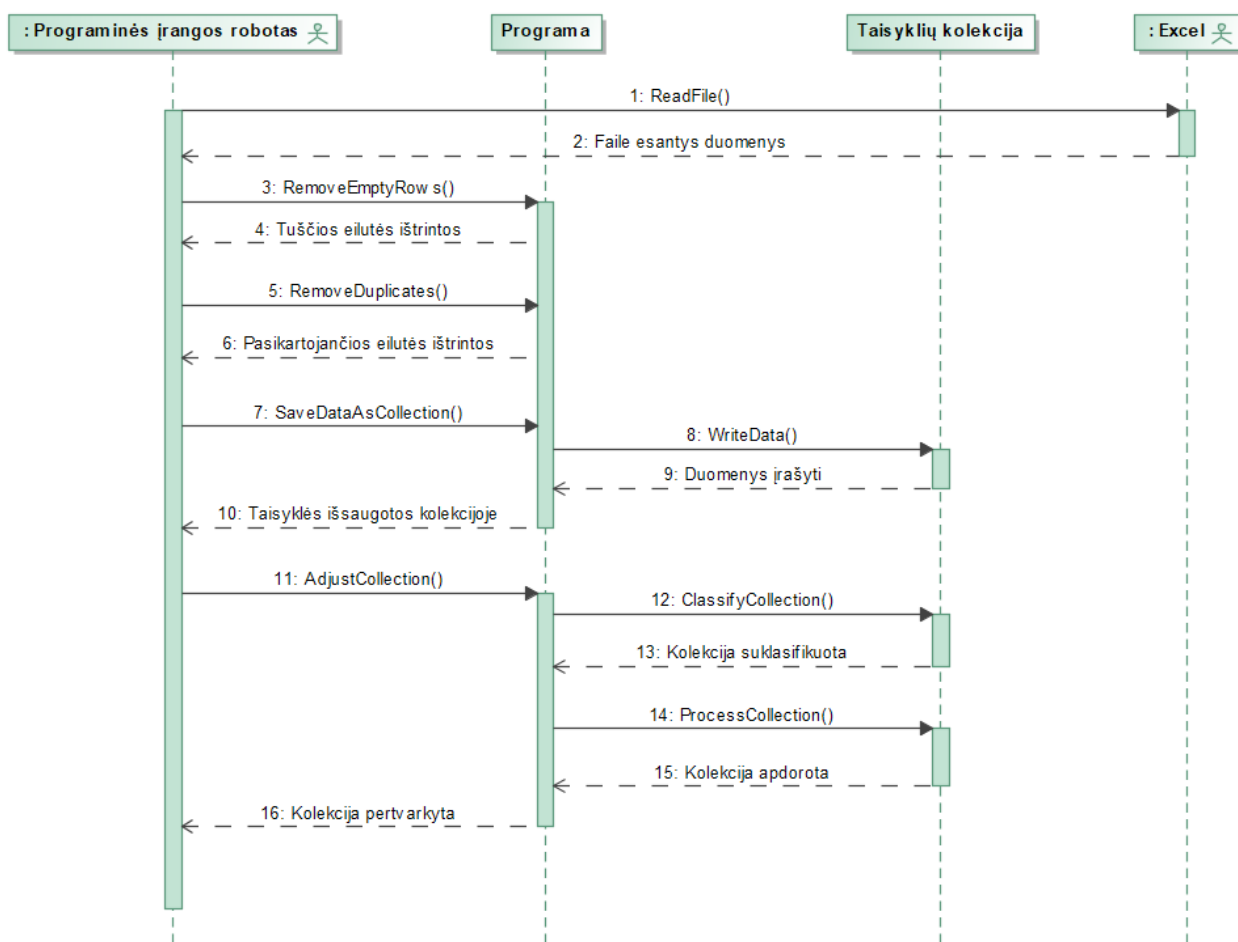
Žemiau yra pateikiama ataskaitos sugeneravimo panaudos atvejo veiklos diagrama (žr. 3.13 pav.). Čia matome, kad programinės įrangos robotas arba sistema apdoroja vykdymo duomenis, įrašo vykdymo duomenis į ataskaitos kolekciją ir inicijuoja *Excel* programos objekto sukūrimą. *Excel* programa grąžina sukurto objekto kodą. Sistema inicijuoja darbo aplinką. *Excel* programa grąžina darbo aplinkos kodą. Sistema inicijuoja lapo priskyrimą. *Excel* programa priskiria lapą. Sistema perduoda duomenis informacijos įrašymui. *Excel* programa įrašo informaciją ir grąžina įrašymo statusą. Jeigu įrašyti nepavyko, tai sistema bando kartoti įrašymą dar kartą. Jeigu duomenys buvo įrašyti sėkmingai, tai failas yra išsaugomas *SharePoint* repozitorijoje. Galiausiai, programa inicijuoja *Excel* programos objekto sunaikinimą. *Excel* programa sunaikina objektą. *Excel* ir *SharePoint* programos privalo būti aktyvios tam, kad ši veikla galėtų būti vykdoma.



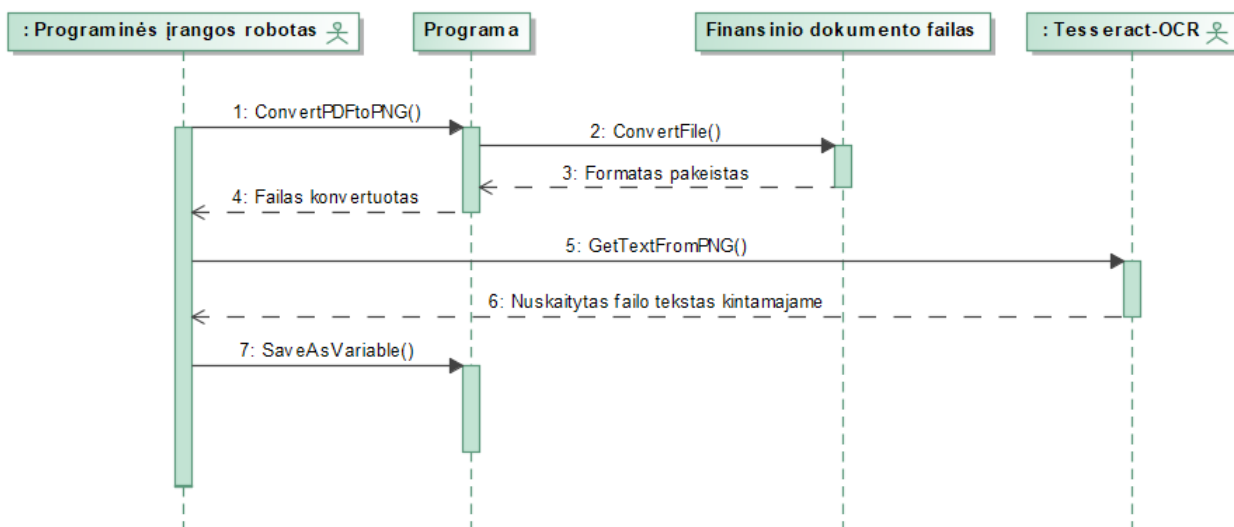
3.13 pav. Ataskaitos sugeneravimo veiklos diagrama

### 3.3.3. Sekų diagramos

Žemiau yra pateikiamos panaudos atvejų sekų diagramos (žr. 3.14 pav. - 3.19 pav.). Sekų diagramos detalizuoja, papildoma skyriuje 3.3.2. pateiktą informaciją.

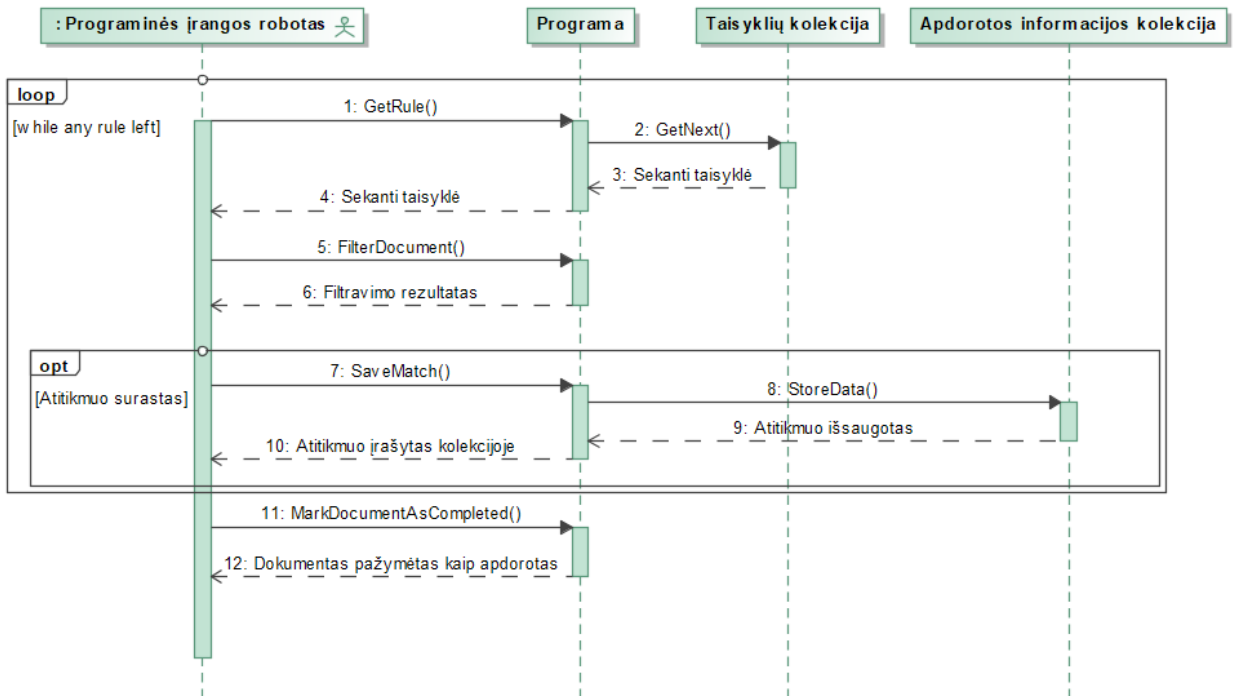


3.14 pav. Taisyklių failo nuskaitymo sekų diagrama

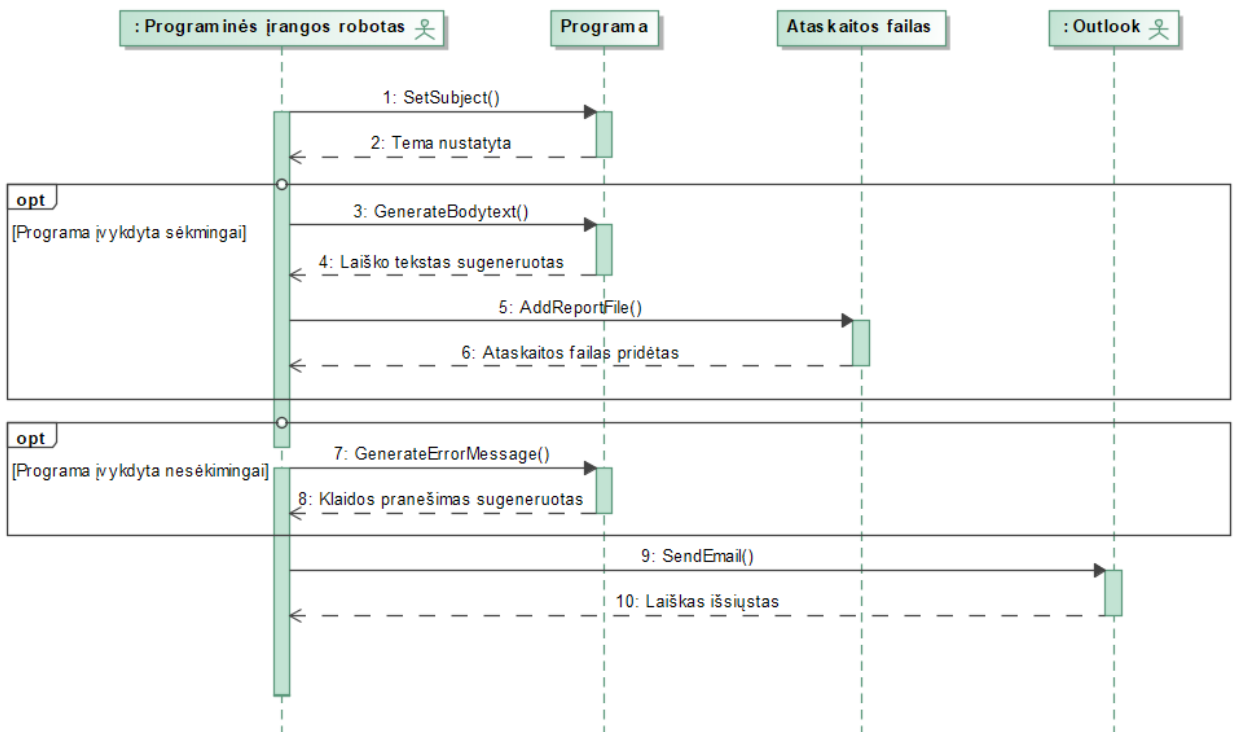


3.15 pav. Finansinio dokumento nuskaitymo sekų diagrama

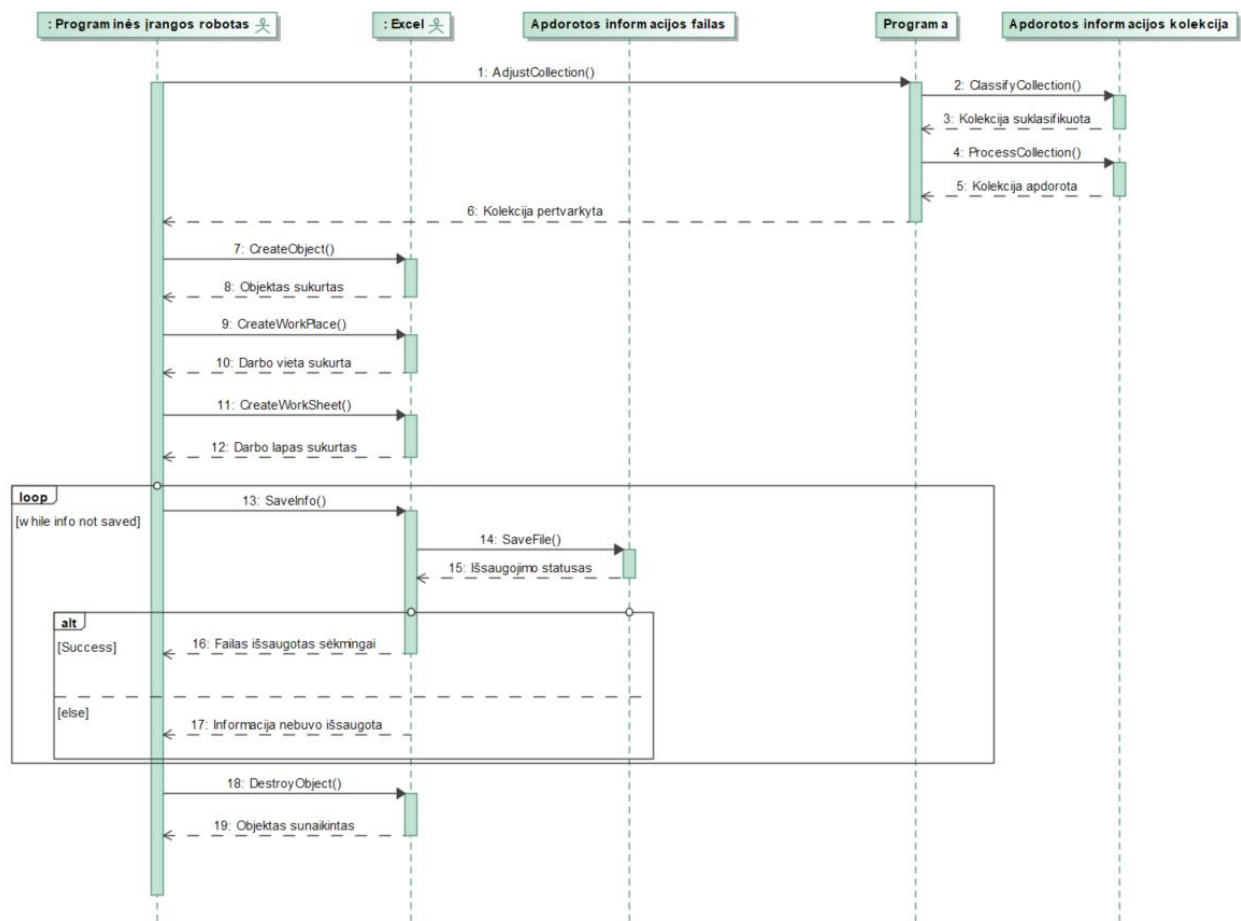




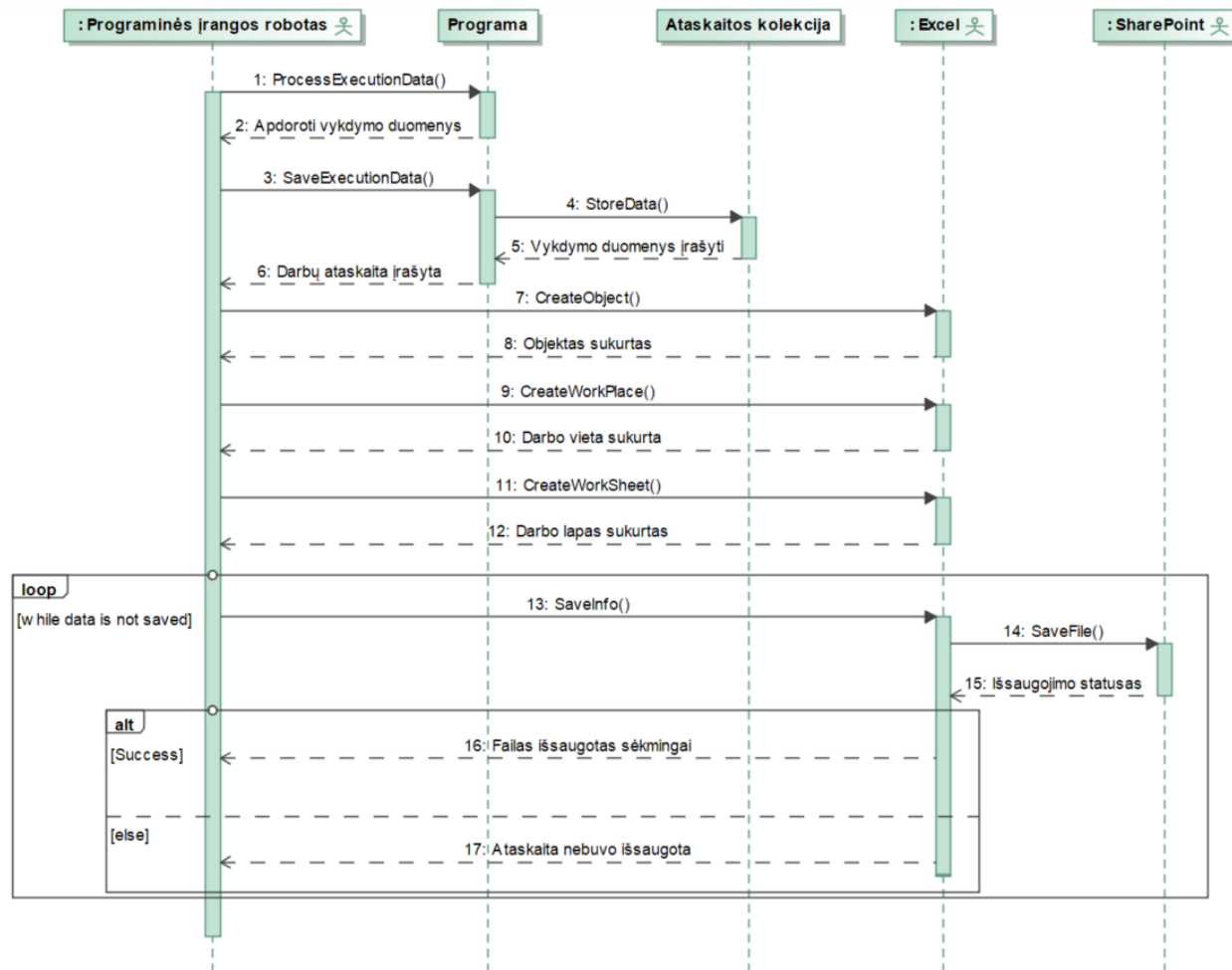
3.16 pav. Finansinio dokumento apdorojimo sekų diagrama



3.17 pav. Ataskaitos išsiuntimo sekų diagrama



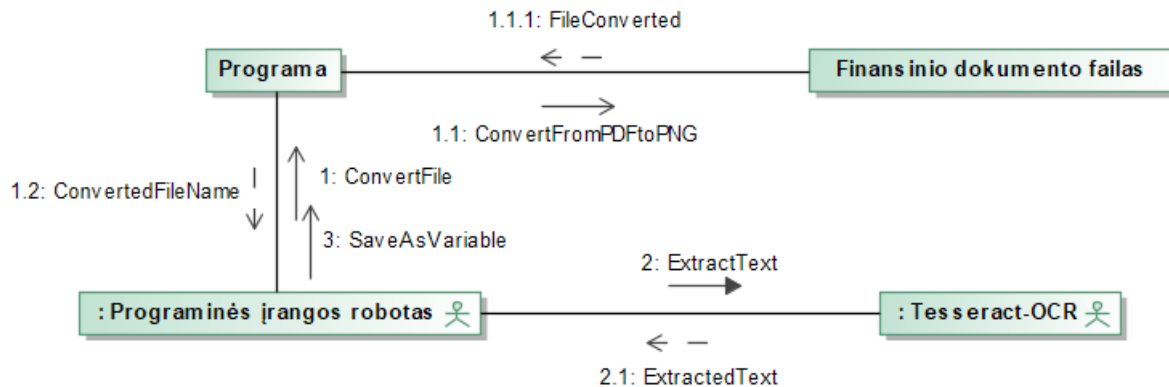
3.18 pav. Apdorotos informacijos įrašymo sekų diagrama



3.19 pav. Ataskaitos sugeneravimo sekų diagrama

### 3.3.4. Bendradarbiavimo diagrama

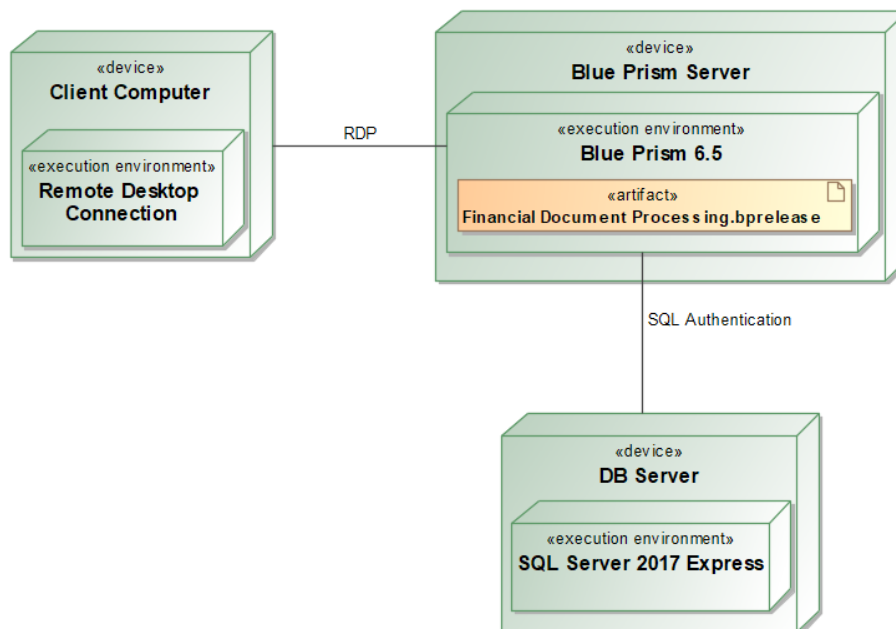
Tam, kad būtų geriau suprasta finansinio dokumento nuskaitymo sekų diagrama (žr. 3.15 pav.), buvo sudaryta to pačio panaudos atvejo bendradarbiavimo diagrama (žr. 3.20 pav.). Čia galime matyti, kaip programos objektai komunikuoja tarpusavyje tam, kad būtų gaunamas finansinio dokumento faile esantis tekstas.



3.20 pav. Finansinio dokumento nuskaitymo bendradarbiavimo diagrama

### 3.4. Išdėstymo vaizdas

Žemiau yra pateikiama proceso diegimo diagrama (žr. 3.21 pav.). Ši diagrama parodo, kad procesas yra įrašytas į *Blue Prism* serverį, kuris yra pasiekiamas naudojant *RDP* protokolą per nutolusį darbalaukį. *Blue Prism* serveris tiesiogiai kontaktuoja su *SQL Server* duomenų baze per *SQL Authentication* ryšį. Ši diagrama nurodo viską, ko reikia iš techninės pusės tam, kad procesas galėtų būti vykdomas.

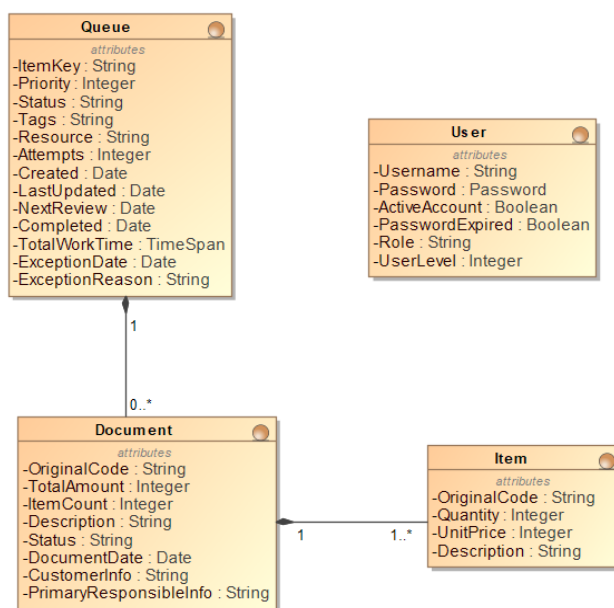


3.21 pav. Proceso diegimo diagrama

Programinė įranga yra naudojama *Windows 10* operacinėje sistemoje, kurioje yra įdiegta *Blue Prism v7.0* programinė įranga. Vienintelis skirtumas tik tas, kad kuriant buvo naudojama *Blue Prism* kūrimo aplinka, o naudojant programinį robotą po sukūrimo yra naudojama produktyvioji *Blue Prism* aplinka. Realios įtakos diegimo aplinkai šis skirtumas nesudaro.

### 3.5. Duomenų vaizdas

Šiame skyrelyje yra vaizduojama dalykinės srities duomenų bazės klasių diagrama (žr. 3.22 pav.). RPA projektams būdingos nedidelės klasių diagramos, nes jose yra saugomi tik rezultatai bei tarpiniai rezultatai. Klasių diagramą visada sudaro eilė (angl. *queue*). Šioje eilėje yra saugoma pagrindinė informacija apie kiekvieną iš atvejų (šio projekto atveju, tai informacija apie kiekvieną finansinį dokumentą). Tai tokia bendrinė informacija kaip statusas, vykdymo laikas, klaidos priežastis, žymos ir t.t. Tačiau suprantama, kad, apdorojus finansinį dokumentą, surasta informacija turi būti išsaugoma. Šiam tikslui yra naudojama dokumento (angl. *document*) klasė. Šioje klasėje yra saugoma visa informacija, reikalinga teisingam dokumento apdorojimui. Tai tokia informacija kaip dokumento numeris, kliento informacija, pardavėjo informacija, dokumento data, prekių kiekis dokumente ir t.t. Dokumentas gali turėti neapibrėžtą kiekį prekių ar paslaugų. Ši informacija saugoma atskiroje elemento (angl. *item*) klasėje. Tai tokia informacija kaip prekės ar paslaugos unikalus kodas, aprašymas, kiekis bei vieneto kaina. Kad projektas galėtų būti vykdomas vartotojas nėra reikalingas, bet tam, kad būtų sukurta eilė, vartotojas privalomas. Taigi, šioje klasių diagramoje yra atvaizduota vartotojo (angl. *user*) klasė. Ji neturi ryšio su kitomis klasėmis, tačiau yra reikalinga pradinėje projekto fazėje. Vartotojas yra reikalingas prižiūrint programinės įrangos roboto veiklą. Vartotojas turi tokią pagrindinę informaciją kaip vartotojo vardas, slaptažodis, rolė, lygis bei slaptažodžio galiojimo ir paskyros aktyvumo vėliavėles. Turint tokią duomenų bazės struktūrą, programinės įrangos robotas gali pilnai funkcionuoti ir saugoti procese apdorotą informaciją.



3.22 pav. Dalykinės srities duomenų bazės klasių diagrama

## 4. TYRIMO DALIS

### 4.1. Kokybės vertinimas

Šiame skyriuje pateikiama projekto kokybės vertinimo proceso ir procedūrų aprašymai, vertinimo rezultatai, išvados. Nustatyta ar programų sistema atitinka standartus bei reikalavimų specifikaciją, ar išpildyti užsakovo lūkesčiai.

#### 4.1.1. Atlikto darbo kokybės analizės tikslai

##### **Aptikti klaidas funkcionavime, logikoje, realizacijoje.**

Klaidoms aptikti buvo naudojamas tas pats įrankis kaip ir programų sistemos realizacijai – *Blue Prism*. Kiekviena funkcija buvo detalai tikrinama pagal iš anksto apibrėžtą scenarijų. Atlikus programos vykdymą su įvairiais pradiniais duomenimis ir scenarijais buvo pastebėta, kad programa dirba taip kaip numatyta scenarijuose ir jokių esminių, kritinių ar didelių klaidų programos funkcijose, logikoje ar pačioje realizacijoje nerasta. Buvo surastos kelios kosmetinės klaidos programos atskirų paketų, kintamųjų pavadinimuose, tačiau tokios nereikšmingos klaidos toliau nėra detalizuojamos kaip realios klaidos. Jos buvo iš karto ištaisytos.

##### **Patikrinti ar programų sistema atitinka reikalavimų specifikaciją.**

Programos atitikimas reikalavimų specifikacijai buvo tikrinamas kartu su užsakovais. Buvo detalai einama per programos kodą žingsnis po žingsnio (angl. *debug*) kartu tikrinant kiekvieno funkcionalumo reikalavimų specifikaciją. Po patikrinimo buvo nustatyta, kad programa pilnai atitinka anksčiau aprašytą reikalavimų specifikaciją.

##### **Įsitikinti, ar programų sistema sukurta pagal standartus.**

Programų sistema buvo kuriama pagal oficialius *Blue Prism* pateikiamus standartus. Standartai yra pateikiami oficialioje *Blue Prism* svetainėje adresu:

<https://portal.blueprism.com/documents/standard> (reikalinga aktyvi anketa tam, kad standartai būtų prieinami). Svetainėje yra pateikiama daugybė įvairiausių standartų kūrimo, projektavimo, palaikymo ir kitose srityse. Visi šie standartai yra taikomi šiame projekte. Didelis dėmesys buvo skiriamas saugumo standartams, kurie yra aprašomi šiame dokumente:

<https://www.blueprism.com/uploads/resources/white-papers/Blue-Prism-Security-Guide-2020.pdf> (dokumentas yra prieinamas visiems). Atlikus programų sistemos peržiūrą buvo nustatyta, kad kuriamas projektas pilnai atitinka naujausius *Blue Prism* standartus ir gerąsias praktikas.

#### 4.1.2. Kokybės vertinimo procesas

##### **Peržiūros**

Realizavus įrankį projekto vykdytojų komanda (projekto vadovas – užsakovas ir projekto vykdytojas) atlieka apžvalgą, kurios tikslas yra surinkti informaciją apie projektą, kas jame pavyko gerai ir kas nepasisėkė, kad tai būtų galima panaudoti tolesniuose projektuose ar tolesnėse šios finansinių dokumentų automatinio turinio nuskaitymo programų sistemos vystymo stadijose. Ši apžvalga paprastai yra atliekama projekto pabaigoje. Jei projektas didesnis apžvalga gali būti atliekama ir po kiekvienos svarbesnės dalies pabaigimo. Šio projekto atveju peržiūros buvo atliekamos kas pusmetį (kiekvieno semestro pabaigoje) gruodžio, birželio mėnesiais. Viso projekto metu reikalavimai nesikeitė. Buvo aptarti ir įgyvendinti skirtingi su dirbtiniu intelektu ir jo panaudojimu susiję sprendimai. Galiausiai pasirinktas *Tesseract*

dirbtinio intelekto simbolių atpažinimo įrankis. Kitų patobulinimų ar pakeitimų projekto metu nebuvo fiksuojama ar įgyvendinta.

### **Rolės ir atsakomybės**

Vilius Kerutis – projekto vykdytojas, programuotojas, testuotojas, analitikas.

Agnė Paulauskaitė – Tarasevičienė – projekto vadovė, užsakovė, analitikė.

Abu asmenys atsakingi už analitinę dalį – techninius sprendimus, kurie turi būti naudojami kuriamoje programų sistemoje, pvz.: RPA įrankio pasirinkimas ar dirbtinio intelekto skirtingų bibliotekų, kurios dirba su optiniu simbolių atpažinimu (angl. OCR), analizė.

Vilius Kerutis bendrai yra atsakingas už programų sistemos realizaciją, atitikimą reikalavimų specifikacijai, o Agnė Paulauskaitė – Tarasevičienė atsakinga už sklandų projekto vykdymą, grafikų laikymąsi, reikiamų duomenų pateikimą vykdytojui.

### **Apklaustos anketa**

Projekto vadovė kaip projekto atstovė iš užsakovų pusės užpildė apklaustos anketą (žr. **1 priedas**). Apklaustos anketos rezultatai teigiami. Programų sistema įvertinta aukščiausiais balais. Pastebėtas tik vienas dalykas, kurį derėtų keisti ar tobulinti – dirbtinio intelekto integracija RPA programinės įrangos kūrimo aplinkoje. Deja, šiuo metu nėra žinoma jokia kita šios integracijos alternatyva – programa naudoja komandinę eilutę sąsajai su dirbtinio intelekto bibliotekomis sukurti. Sąsaja yra ganėtinai stabili, tačiau nepilnai atitinka gerąsias praktikas. Kadangi nėra kitos alternatyvos problemai spręsti, tai šis sprendimas bus paliktas tol, kol atsiras galimybė, metodas, įrankis ar kt. geriau realizuoti sąsają tarp RPA aplinkos ir dirbtinio intelekto bibliotekų.

#### **4.1.3. Vertinimo rezultatai**

Žemiau pateiktoje kokybės vertinimo rezultatų lentelėje (žr. 4.1 lentelė) matome, kad kokybei įvertinti buvo naudojama dešimt kriterijų. Kiekvienas kriterijus turi aprašymą, įvertinimą ir komentarą, kodėl būtent toks įvertinimas buvo skiriamas. Kiekvienas kriterijus yra įvertintas dešimtbalėje sistemoje, kur 10 yra maksimalus įvertinimas. Maksimalų įvertinimą gavo funkcionalumas, efektyvumas, patikimumas, saugumas, palaikomumas ir išplečiamumas. Šios savybės pilnai atitinka apibrėžtus reikalavimus, yra įgyvendintos taikant naujausius principus bei standartus ir sistema pagal šiuos kriterijus dirba būtent taip, kaip buvo galima tikėtis projekto pradžioje. Šiek tiek mažesnę įvertinimą (9) gavo panaudojamumas, nes yra manoma, kad yra galimybė sistema naudotis dar paprasčiau, taigi yra palikta vietos tobulėjimui. Dar mažesnę įvertinimą gavo suderinamumas (8), nes programa šiuo metu yra suderinama su kitomis sistemomis per komandinę eilutę, o tai nėra pats lengviausias ir geriausias būdas suderinti skirtingas sistemas. Bus bandoma ieškoti kitų – geresnių, lengvesnių – būdų suderinti sistemą su kitomis sistemomis, tačiau šiuo metu yra suprantama, kad tokių būdų naudojant *Blue Prism* programinę įrangą nėra. Taigi, tai yra programos vieta, kurią galbūt bus galima patobulinti ateityje. Mažiausią įvertinimą gavo pernešamumas (7) ir tikslumas (6). Sukurtas produktas nėra itin lengvai pernešamas į kitą terpę, nes yra reikalinga specialios *Blue Prism* programinės įrangos instaliacija, kartu ir duomenų bazės konfigūravimas, naudojamų bibliotekų (*Tesseract*, *ImageMagick*) įrašymas, jų sukonfigūravimas naujoje terpėje – sisteminių kelių nustatymas sistemos viduje. Taigi, pernešamumas reikalauja nemažai darbo ir žinių tam, kad sistema galėtų sėkmingai funkcionuoti kitoje terpėje. Sukurtos programos tikslumas šiuo metu nėra itin patikimas ir geras. Tačiau to ir buvo tikimasi, nes yra numatyta,

kad programos tikslumas bus tobulinamas sekančiame etape atliekant tyrimus, išbandant skirtingas metodikas, praktikas ar kitus įvairius būdus ir jų kombinacijas. Taigi, šiuo momentu programa pilnai atlieka tiek, kiek buvo tikimasi projekto pradžioje.

Bendras sukurtos programų sistemos įvertinimas yra 9. Matome, kad yra vietos tobulėjimui, tačiau įvertinimas yra itin aukštas. Sukurta programa toliau bus tobulinama ir tikimasi, kad galiausiai sistemos įvertinimas turėtų pakilti iki 9.5 balų. Sistema pilnai atitinka iškeltus reikalavimus ir yra tinkama tolimesniam naudojimui.

**4.1 lentelė** Kokybės vertinimo rezultatai

Savybė	Aprašymas	Įvertinimas	Komentaras
<b>Funktionalumas</b>	Ar pakankamas funkcionalumas?	10	Visos numatytos funkcijos yra įgyvendintos, jų pakanka užduočiai atlikti.
<b>Efektyvumas</b>	Ar sistema efektyviai naudoja resursus?	10	Programa dirba foniniu (angl. <i>background</i> ) režimu, taigi resursų naudojama itin nedaug.
<b>Tikslumas</b>	Ar sistema gauna tikslus rezultatus?	6	Tikslumas bus toliau tobulinamas taikant įvairius metodus ir praktikas.
<b>Suderinamumas</b>	Pastangos, reikalingos sistemai suderinti su kita sistema.	8	Sąsajos kuriamos bendradarbiaujant komandinei eilutei. Tai nėra geriausias sprendimas, bet kitų variantų šiuo metu nėra.
<b>Panaudojamumas</b>	Sistemos paprastumas.	9	Sistema yra itin paprasta, ją lengva naudoti ir suprasti. Veikia <i>DaaS</i> (angl. <i>Document as a Service</i> ) principu.
<b>Patikimumas</b>	Sistemos gebėjimas tęsti darbą po sisteminių klaidų.	10	Sistema sėkmingai tęsia darbą po sisteminių klaidų nebent jos kartojasi.
<b>Saugumas</b>	Naudojamų duomenų apsauga.	10	Sistema pilnai atitinka <i>Blue Prism</i> nustatytus saugumo standartus.
<b>Palaikomumas</b>	Ar lengvai palaikoma sistema?	10	Sistemos palaikymas nereikalauja jokių papildomų laiko resursų, veikia stabiliai.
<b>Pernešamumas</b>	Pastangos, reikalingos produkto perkėlimui iš vienos aplinkos į kitą.	7	Reikalinga instaliuoti speciali programinė įranga ir papildomos bibliotekos, sisteminių kelių konfigūravimas yra gan sudėtingas.
<b>Išplečiamumas</b>	Ar lengva sistemoje integruoti naujas funkcijas?	10	Sistema sukurta taikant <i>Blue Prism</i> nustatytus kūrimo principus, taigi naujas funkcijas integruoti yra paprasta.

### Realio darbo kaina

Projektas buvo vykdomas ne komerciniais tikslais, todėl už tai nebuvo atlyginama jokių piniginių įvertinimų ir nėra pateikiama atliktų darbų kaina.

### Darbo našumas

Planuoti darbai buvo atlikti pagal numatytą tvarkaraštį.

Numatytas biudžetas taip pat nebuvo viršytas.

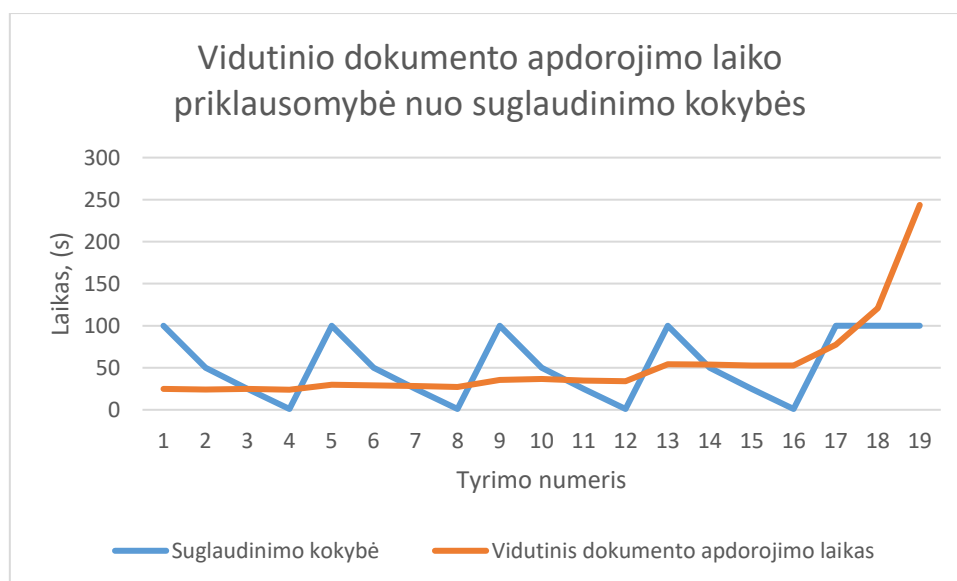


## 4.2. Sistemos parametrų tyrimas

Sistemos atpažįstamų laukų tikslumą įtakoja tam tikri sistemos parametrai, kurie gali būti valdomi. Šio tyrimo metu buvo bandoma išsiaiškinti, kurie parametrai, bei kokios reikšmės labiausiai įtakoja teisingą laukų atpažinimą.

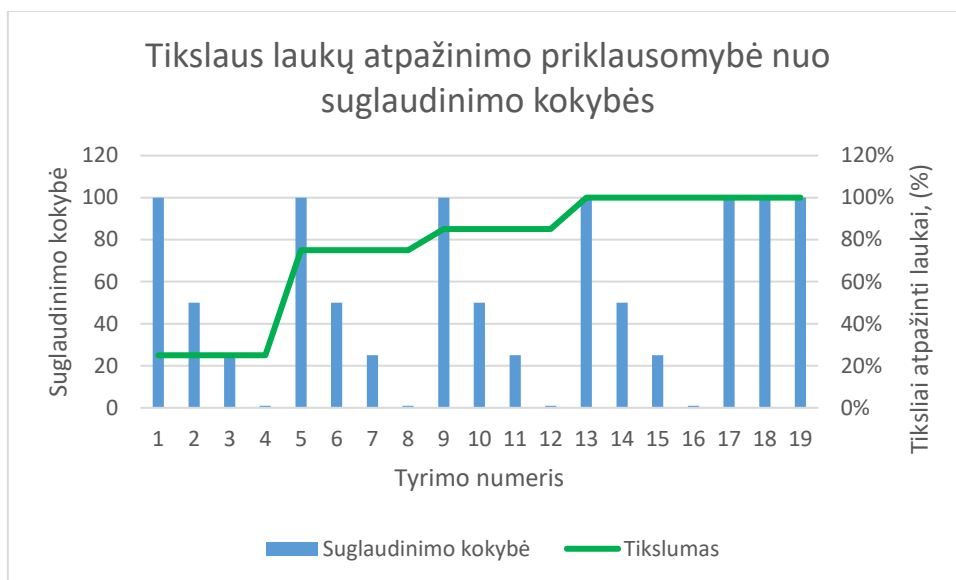
Vienas iš tokių parametrų yra suglaudavimo kokybė. Tai parametras, kurį galime rinktis, kai finansinis dokumentas yra konvertuojamas iš PDF formato į PNG formatą. Programinės įrangos roboto naudojamas *ImageMagick* įrankis leidžia rinktis suglaudavimo laipsnį. Pagal tą laipsnį yra išskiriama kokybė – kuo mažesnis suglaudavimo laipsnis, tuo aukštesnę paveiklo kokybę išsaugome. Parametras pasirenkamas skaičiumi nuo 1 iki 100, kur 100 reiškia aukščiausią išsaugojamą kokybę (mažiausią suglaudimą), o 1 reškia žemiausią paveiklo kokybę, tačiau didžiausią suglaudimą. Šio tyrimo metu buvo siekiama išsiaiškinti, ar suglaudavimo kokybė įtakoja atpažįstamų laukų tikslumą, dokumento vidutinį apdorojimo laiką, ir, jeigu įtakoja, taip kaip stipriai bei kokia optimaliausia suglaudavimo reikšmė turėtų būti pasirenkama tolimesniam programos naudojimui.

Žemiau apteiktoje diagramoje (žr. 4.1 pav.) matome, kad suglaudavimo kokybės laipsnis visiškai neįtakoja vidutinio dokumento apdorojimo laiko. Kiekvieno tyrimo metu buvo pasirenkamas vis kitas suglaudavimo kokybės laipsnis, tačiau jis visada svyravo tarp keturių reikšmių (1, 25, 50, 100). Dokumentų kiekis kiekvieno tyrimo metu buvo tas pats, o ir dokumentai kiekvieno tyrimo metu buvo tokie patys. Taigi, galime teigti, kad finansinio dokumento suglaudavimo laipsnis neturi įtakos vidutiniam dokumento apdorojimo laikui. Akivaizdu, kad įtakos turi kiti parametrai.



4.1 pav. Vidutinio dokumento apdorojimo laiko priklausomybė nuo suglaudavimo kokybės

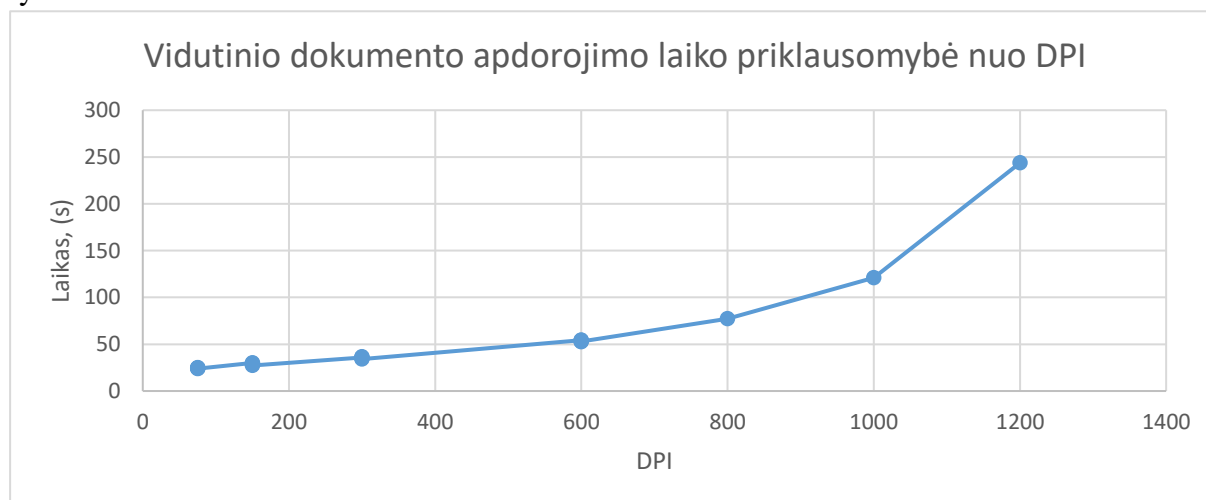
Žemiau pateiktoje diagramoje (žr. 4.2 pav.) matome, kad suglaudavimo laipsnis taip pat neturi įtakos vidutiniam dokumento apdorojimo laikui. Kiekvieno tyrimo metu, kaip ir aukščiau aprašyto tyrimo metu, buvo pasirenkamas vis kitas suglaudavimo kokybės laipsnis, tačiau jis visada svyravo tarp keturių reikšmių (1, 25, 50, 100). Dokumentų kiekis kiekvieno tyrimo metu buvo tas pats, o ir dokumentai kiekvieno tyrimo metu buvo tokie patys. Taigi, dar kartą galime teigti, kad finansinio dokumento suglaudavimo laipsnis neturi įtakos kuriamos sistemos kokybei – šiuo atveju neturi įtakos tiksliam laukų atpažinimui.



**4.2 pav.** Tikslaus laukų atpažinimo priklausomybė nuo suglaudinimo kokybės

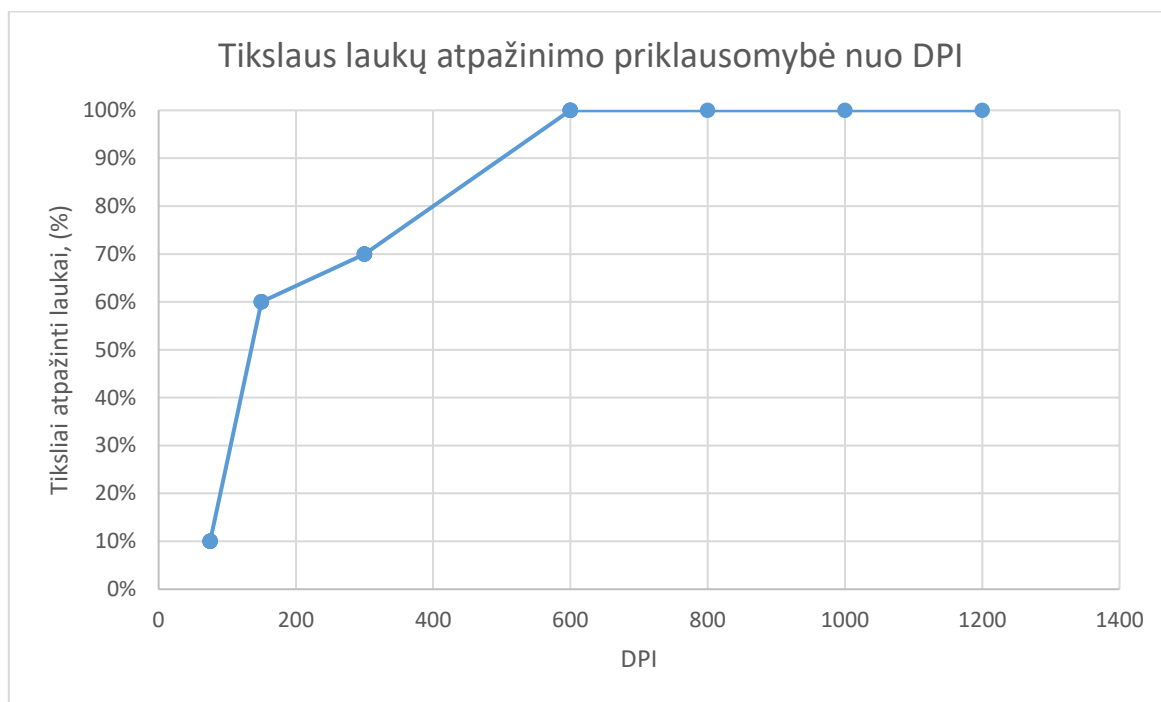
Kitas parametras, kurį sukurtoje sistemoje galima laisvai modifikuoti – DPI (angl. *Dots Per Inch*) arba taškai colyje. Tai parametras, kurį galime rinktis, kai finansinis dokumentas yra konvertuojamas iš PDF formato į PNG formatą. Programinės įrangos roboto naudojamas *ImageMagick* įrankis leidžia rinktis taškų kiekį colyje. Kuo daugiau taškų yra colyje, tuo geresnį paveikslą bei jo kokybę gauname. Akivaizdu, kad kuo daugiau taškų bus colyje, tuo geresnę paveikslo kokybę išgausime ir tuo didesnę laukų atpažinimo tikslumą pasieksime. Tačiau kyla klausimas, kiek taškų colyje yra optimalu tam, kad finansiniame dokumente esanti informacija būtų visada tiksliai surandama, o ir dokumento vidutinis apdorojimo laikas neišaugtų eksponentiškai? Šio tyrimo tikslas yra atsakyti į šį klausimą ir surasti optimaliausią taškų colyje vertę.

Žemiau pateiktoje diagramoje (žr. 4.3 pav.) matome vidutinio finansinio dokumento apdorojimo laiko priklausomybę nuo DPI. Galime pastebėti, kad didinant taškų skaičių colyje vertę kas 200 taškų vidutinis dokumento apdorojimo laikas pradeda augti eksponentiškai. Optimaliausia reikšmė sunaudojamo laiko atžvilgių būtų tarp 400 ir 600 šimtų taškų colyje. Tam, kad būtų surasta tiksli reikalingų taškų colyje reikšmė, reikia įvertinti laukų atpažinimo tikslumą kiekviename iš žemiau pavaizduotų taškų. Tai yra atliekama žemiau pateiktame tyrime.



**4.3 pav.** Vidutinio dokumento apdorojimo laiko priklausomybė nuo DPI

Žemiau pateiktoje diagramoje (žr. 4.4 pav.) matome tikslaus laukų atpažinimo priklausomybę nuo DPI. Galime pastebėti, kad, esant itin mažam DPI, finansinio dokumento laukai yra itin prastai atpažįstami. Tai lemia itin prasta paveikslo kokybė, nes tą paveikslą naudoja *Tesseract-OCR* įrankis ir, matomai, jis negali identifikuoti teksto tiksliai. Tačiau, didinant DPI reikšmę, tikslaus laukų atpažinimo procentas didėja ženkliai. Matome, kad pasiekus 600 DPI visi laukai yra atpažįstami teisingai. Taigi, galima teigti, kad esant 600 DPI ir didesnei reikšmei visi finansinio dokumento laukai, kurie turėtų būti atpažinti, bus atpažinti.

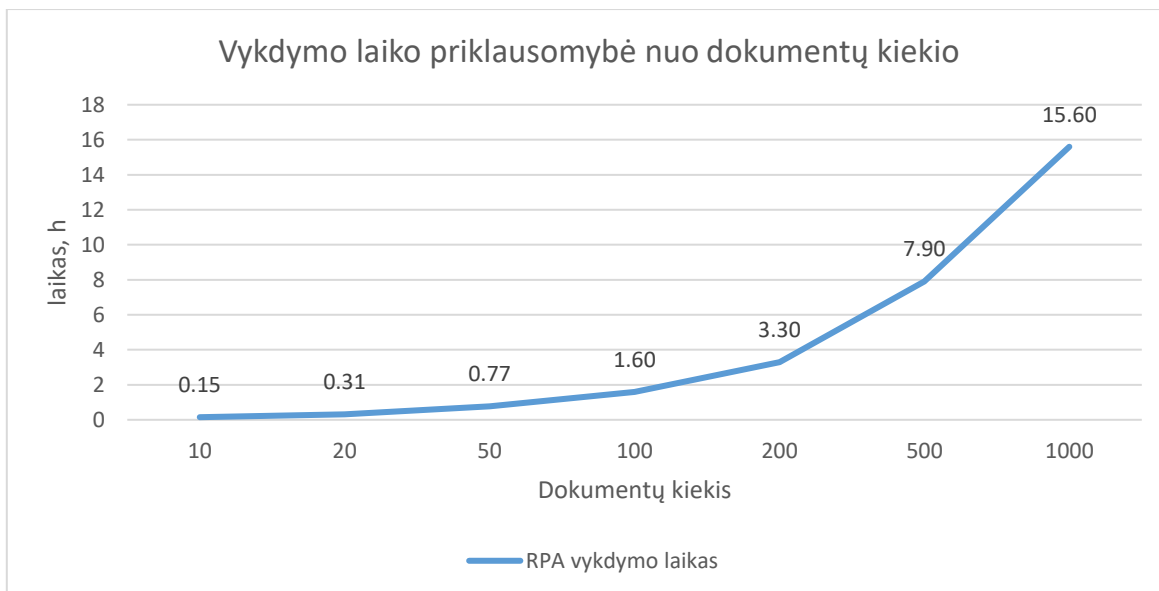


**4.4 pav.** Tikslaus laukų atpažinimo priklausomybė nuo DPI

Įvertinus visus aukščiau pateiktus tyrimus galime teigti, kad suglaudavimo kokybė visiškai neturi įtakos nei dokumento vidutiniam apdorojimo laikui, nei tiksliam laukų atpažinimui. Šis parametras gali visada būti paliktas su 100 verte (aukščiausia kokybė, mažiausias suglaudavimas). Tačiau DPI turi didelę įtaką tiek finansinio dokumento vidutiniam apdorojimo laikui, tiek tiksliam laukų atpažinimui, todėl šio parametro teisingas vertės parinkimas yra itin svarbus, netgi kritinis. Pirmasis DPI tyrimas parodė, kad optimaliausia DPI reikšmė svyruoja tarp 400 – 600 šimtų taškų colyje. Tačiau antrasis tyrimas parodė, kad DPI reikšmė privalo būti bent 600 taškų colyje tam, kad visi laukai būtų atpažinti teisingai. Taigi, apjungus šių dviejų tyrimų rezultatus pastebime, kad optimaliausia DPI vertė yra 600 taškai colyje. Tolimesniuose tyrimuose bus naudojamas suglaudavimo kokybės parametras su 100-o verte ir DPI parametras su 600 taškų colyje verte.

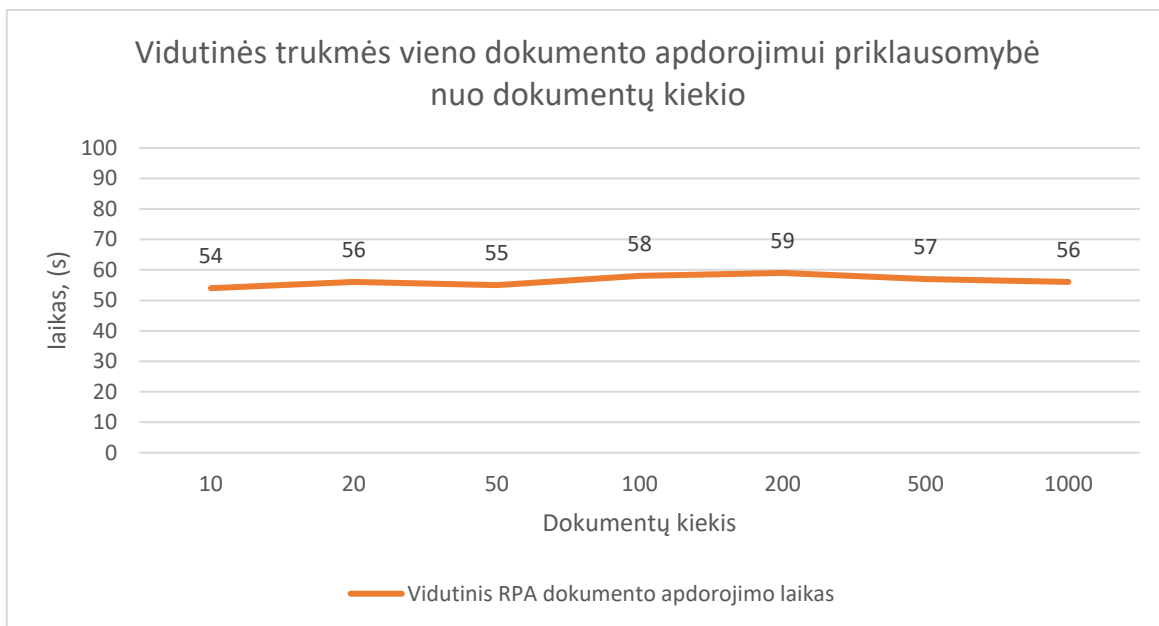
### 4.3. Greitaveikos tyrimas

Šiame tyrime buvo siekiama nustatyti, kiek laiko trunka finansinių dokumentų apdorojimas. Tyrime naudojami stabiliausi ir patikimiausi parametrai tam, kad būtų išgautas maksimalus atpažįstamų laukų tikslumas per trumpiausią laiką. Kaip matome diagramoje pateiktoje žemiau (žr. 4.5 pav.), programinės įrangos robotas gali apdoroti 1000 dokumentų per mažiau nei šešiolika valandų, o tai reiškia, kad per parą robotas gali apdoroti per 1500 dokumentų.



**4.5 pav.** Vykdymo laiko priklausomybė nuo dokumentų kiekio

Labai svarbus sistemos parametras yra stabilumas. Todėl buvo tiriama, ar vidutinis vieno finansinio dokumento apdorojimo laikas ženkliai nekinta apdorojant didelius kiekius dokumentų. Kaip matome diagramoje pateiktoje žemiau (žr. 4.6 pav.), vidutinis vieno dokumento apdorojimo laikas siekia nuo 54-ių iki 59-ių sekundžių.



**4.6 pav.** Vidutinės trukmės vieno dokumento apdorojimui priklausomybė nuo dokumentų kiekio

Žinant tai, kad apdorojamų dokumentų dydis yra labai įvairus (nuo vieno lapo iki keliolikos lapų), tai tam tikra kelių sekundžių paklaida yra visiškai suprantama ir reali. Nedidelį dokumentą programinės įrangos robotas gali apdoroti greičiau nei per pusę minutės, tačiau keliolikos lapų dokumentas gali būti apdorojamas tik per kelias minutes. Dažniausiai finansinius dokumentus, sąskaitas faktūras sudaro vos keli lapai (<5), todėl keli itin dideli dokumentai su keliolika lapų labai nepadidina vidutinio vieno finansinio dokumento apdorojimo laiko.

Taigi, galime teigti, kad robotas dirba itin stabiliai, gali apdoroti pusantro tūkstančio finansinių dokumentų per parą su mažesniu nei vienos minutės vidutiniu dokumento apdorojimo laiku.

Žemiau pateiktoje lentelėje (žr. 4.2 lentelė) galime matyti visų tyrimų apibendrintą rezultatų lentelę. Šioje lentelėje yra pateikiamas tyrimo numeris – jų iš viso buvo devyniolika. Taip pat yra pateikiamas DPI arba taškų skaičius colyje. Ši reikšmė yra derinama su suglaudavimo kokybės reikšmėmis, o tai reiškia, kad kiekviena DPI reikšmė turi būti iširta su kiekviena pasirinkta suglaudavimo kokybės reikšme. Taip pat yra pateikiamas vidutinis finansinio dokumento apdorojimo laikas sekundėmis kiekvieno tyrimo metu. Verta pabrėžti tai, kad kiekvieno tyrimo metu buvo naudojama ta pati imtis dokumentų tam, kad kiekvienas tyrimas turėtų identišką pradinę imtį, nes svarbiausias tyrimo tikslas – patikrinti programinės įrangos roboto išvestį, t.y. veiklos laiką bei laukų aptikimo tikslumą. Galiausiai, lentelėje yra matomi du tikslumo stulpeliai. Pirmasis mums parodo kiekį dokumentų, kurių visi laukai buvo surasti, pavyzdžiui, jeigu dokumente yra ieškomi penki laukai, tačiau buvo surasti tik keturi laukai, tai šis dokumentas šioje skiltyje gaus 0%. Jeigu buvo surasti visi ieškomi laukai, tai dokumentas gauna 100%. Visų dokumentų rezultatai yra apjungiami ir skaičiuojamas vidurkis, pavyzdžiui, jeigu turėjome dešimt dokumentų, bet visi dokumento laukai buvo surasti tik viename iš jų, tai šiame stulpelyje rašoma vertė yra 10% (1-4 tyrimai). Antrasis tikslumo stulpelis mums parodo kiekį visų laukų, kurie buvo surasti, pavyzdžiui, jeigu turime dešimt dokumentų ir kiekviename iš jų reikia surasti po penkis laukus, tai yra procentaliai skaičiuojama kiek bendrai laukų programa teisingai surado iš 50 galimų laukų (dešimt dokumentų po penkis laukus). Būtent dėl šio tikslumo stulpelių skirtumo pastebime, kad pilnai apdoroti dokumentą yra sunkiau negu bendrai ieškoti reikiamų laukų.

**4.2 lentelė** Visų tyrimų apibendrinta rezultatų lentelė

Tyrimo Nr.	DPI	Suglaudavimo kokybė	Vidutinis dokumento apdorojimo laikas, s	Tikslumas (visas dokumentas)	Tikslumas (laukai)
1	75	100	24.7	10%	25%
2	75	50	24.1	10%	25%
3	75	25	24.8	10%	25%
4	75	1	23.9	10%	25%
5	150	100	30	60%	75%
6	150	50	29.2	60%	75%
7	150	25	28.4	60%	75%
8	150	1	27.2	60%	75%
9	300	100	35.7	70%	85%
10	300	50	36.6	70%	85%
11	300	25	34.9	70%	85%
12	300	1	34.1	70%	85%
13	600	100	54.3	100%	100%
14	600	50	53.8	100%	100%
15	600	25	52.7	100%	100%
16	600	1	52.8	100%	100%
17	800	100	77.3	100%	100%
18	1000	100	120.9	100%	100%
19	1200	100	243.7	100%	100%

## 5. EKSPERIMENTINĖ DALIS

### 5.1. Testavimo rezultatai ir išvados

Sistemos testavimas davė teigiamus rezultatus. Programa pagal sukurtus testus veikė taip kaip buvo numatyta specifikacijoje. Buvo paruoštas taisyklių failas, kurio fragmentą galime matyti paveikslėlyje žemiau (žr. 5.1 pav.). Šiame taisyklių faile yra aprašomos kiekvieno lauko atskiros taisyklės skirtinguose skirtukuose (angl. *sheets*). Taisyklių užrašymo sintaksė atitinka reguliariąją išraišką, kuri dažniausiai naudojama paieškos modeliuose. Programinės įrangos robotas pritaiko kiekvieną taisyklę vieną po kitos su nuskaitytu dokumento tekstu ir ieško atitikmens. Jeigu atitikmenų buvo daugiau negu vienas, tai, būtent datos paieškos dokumente atveju, yra naudojama mažiausią pradžios indeksą turinti data, t.y. aukščiausiai dokumente esanti data. Taip yra daroma todėl, kad data (nepriklausomai nuo dokumento struktūros) yra dažniausiai rašoma dokumento pradžioje. Atvirkštinis variantas yra su galutine suma. Galutinės sumos paieškos algoritmas veikia lygiai taip pat, tačiau jeigu buvo surastas daugiau nei vienas atitikmuo, tai yra naudojama didžiausią pradžios indeksą turinti galutinė kaina, nes yra priimta (nepriklausomai nuo dokumento struktūros), kad galutinė suma yra dažniausiai rašoma dokumento pabaigoje. Panašiu principu yra ieškomi ir kiti laukai.

	A	B
1	<b>Rule</b>	<b>Example</b>
2	<code>\d{2}\.\s\d{2}\.\s\d{4}</code>	13. 01. 2021
3	<code>\d{2}\.\d{2}\.\d{4}</code>	13.01.2021
4	<code>\d{4}\.\s\d{2}\.\s\d{2}</code>	2021. 01. 13
5	<code>\d{4}\.\d{2}\.\d{2}</code>	2021.01.13
6	<code>\d{2}\-\s\d{2}\-\s\d{4}</code>	13- 01- 2021
7	<code>\d{2}\-\d{2}\-\d{4}</code>	13-01-2021
8	<code>\d{4}\-\s\d{2}\-\s\d{2}</code>	2021- 01- 13
9	<code>\d{4}\-\d{2}\-\d{2}</code>	2021-01-13
10	<code>\d{2}\s\-\s\d{2}\s\-\s\d{4}</code>	13 - 01 - 2021
11	<code>\d{4}\s\-\s\d{2}\s\-\s\d{2}</code>	2021 - 01 - 13
12	<code>\d{2}\V\d{2}\V\d{4}</code>	13/01/2021
13	<code>\d{4}\V\d{2}\V\d{2}</code>	2021/01/13
14	<code>\d{2}\s\V\s\d{2}\s\V\s\d{4}</code>	13 / 01 / 2021
15	<code>\d{4}\s\V\s\d{2}\s\V\s\d{2}</code>	2021 / 01 / 13
16	<code>\d{2}\V\d{2}\V\d{2}</code>	13/01/21
17	<code>\d{1}\V\d{2}\V\d{2}</code>	3/01/21
18	<code>\d{2}\V\d{2}\V\d{1}</code>	21/01/3
19	<code>\d{2}\-\d{2}\-\d{2}</code>	13-01-21
20	<code>\d{1}\-\d{2}\-\d{2}</code>	3-01-21
21	<code>\d{2}\-\d{2}\-\d{1}</code>	21-01-3
22	<code>\d{2}\.\d{2}\.\d{2}</code>	13.01.21
23	<code>\d{1}\.\d{2}\.\d{2}</code>	3.01.21
24	<code>\d{2}\.\d{2}\.\d{1}</code>	21.01.3
25	<code>\d{2}\s\V\s\d{2}\s\V\s\d{2}</code>	13 / 01 / 21

5.1 pav. Taisyklių failo datos nustatymo pavyzdys

Atliktas kliento priėmimo testavimas (angl. BAT – *Business Acceptance Testing*) parodė (žr. 5.2 pav.), kad sukurta sistema atitinka kliento reikalavimus ir veikia taip, kaip klientai viską įsivaizdavo pradžioje. Čia galime matyti atlikto testavimo rezultatus. Šie rezultatai buvo rodomi užsakovams, kurie patvirtino, kad informacija, kurią pateikė programinės įrangos robotas, yra teisinga ir programa gražina tokius rezultatus, kokių buvo laukiama. Tačiau, buvo pastebėta ir klaidų – neteisinga data ar nurodyta bloga galutinė kaina. Reikalavimų specifikacijoje buvo apibrėžta, kad programa privalo teisingai apdoroti bent 80 procentų finansinių dokumentų. Teisingai buvo apdorota apie 83 procentai visų testavime naudotų finansinių dokumentų, o tai reiškia, kad reikalavimų specifikacijoje numatyta riba buvo pasiekta ir programa pilnai atitinka anksčiau nustatytus užsakovų reikalavimus.

	A	B	C	D	E	F	G	H	I	J	K
1	Number	Status	Worktime	Finished At	Seller	Buyer	Document Number	Document Date	Total Price	Link	Exception Reason
2	1	Completed	0.00:00:34	30/10/2021 15:29:18	-	-	-	2021.01.13	8.7500	https://ktuedu.sharepoint.com/sites/2021R_Maste	
3	2	Completed	0.00:00:34	30/10/2021 16:00:26	-	-	-	2021 sausio 18	346,54	https://ktuedu.sharepoint.com/sites/2021R_Maste	
4	3	Completed	0.00:00:37	30/10/2021 15:27:05	-	-	-	11.01.2021	4 273,50	https://ktuedu.sharepoint.com/sites/2021R_Maste	
5	4	Completed	0.00:00:37	30/10/2021 15:32:45	-	-	-	2021.01.21	536.7600	https://ktuedu.sharepoint.com/sites/2021R_Maste	
6	5	Completed	0.00:01:03	30/10/2021 15:45:36	-	-	-	05.01.2021	8,960	https://ktuedu.sharepoint.com/sites/2021R_Maste	
7	6	Completed	0.00:01:15	30/10/2021 15:31:13	-	-	-	08.12.2020	305,70	https://ktuedu.sharepoint.com/sites/2021R_Maste	
8	7	Completed	0.00:00:34	30/10/2021 15:54:31	-	-	-	26.06.2020	5 327,00	https://ktuedu.sharepoint.com/sites/2021R_Maste	
9	8	Completed	0.00:00:38	30/10/2021 15:47:09	-	-	-	18/01/21	1,410.00	https://ktuedu.sharepoint.com/sites/2021R_Maste	
10	9	Completed	0.00:00:33	30/10/2021 15:38:19	-	-	-	2838 PVM SASKA	1 834.24	https://ktuedu.sharepoint.com/sites/2021R_Maste	
11	10	Completed	0.00:00:30	30/10/2021 15:55:01	-	-	-	22.01.2021	3 927,60	https://ktuedu.sharepoint.com/sites/2021R_Maste	
12	11	Completed	0.00:01:05	30/10/2021 15:53:22	-	-	-	22-01-21	306.7500	https://ktuedu.sharepoint.com/sites/2021R_Maste	
13	12	Exception	0.00:00:00	30/10/2021 15:19:30	-	-	-	-	-	https://ktuedu.sharepoint.com/sites/2021R_Maste	File uploaded into SharePoint is not .pdf format !
14	13	Completed	0.00:00:58	30/10/2021 15:48:07	-	-	-	05.10.2020	1.227,49	https://ktuedu.sharepoint.com/sites/2021R_Maste	
15	14	Completed	0.00:00:43	30/10/2021 15:55:44	-	-	-	2020 m. sausio 22	20 505,40	https://ktuedu.sharepoint.com/sites/2021R_Maste	
16	15	Exception	0.00:00:00	30/10/2021 15:19:27	-	-	-	-	-	https://ktuedu.sharepoint.com/sites/2021R_Maste	File uploaded into SharePoint is not .pdf format !
17	16	Completed	0.00:01:18	30/10/2021 15:43:30	-	-	-	15.01.2021	1,00000	https://ktuedu.sharepoint.com/sites/2021R_Maste	
18	17	Completed	0.00:00:47	30/10/2021 15:28:44	-	-	-	13. 01. 2021	4 299,00	https://ktuedu.sharepoint.com/sites/2021R_Maste	
19	18	Exception	0.00:00:00	30/10/2021 15:51:41	-	-	-	-	-	https://ktuedu.sharepoint.com/sites/2021R_Maste	Conversion from PDF to PNG failed, file not found !
20	19	Completed	0.00:01:01	30/10/2021 15:24:43	-	-	-	13.01.2021	8,23	https://ktuedu.sharepoint.com/sites/2021R_Maste	
21	20	Completed	0.00:00:59	30/10/2021 15:34:27	-	-	-	22.01.2021	249,74	https://ktuedu.sharepoint.com/sites/2021R_Maste	
22	21	Completed	0.00:01:51	30/10/2021 15:37:46	-	-	-	76 45 10	3.370,58	https://ktuedu.sharepoint.com/sites/2021R_Maste	
23	22	Completed	0.00:00:36	30/10/2021 15:52:17	-	-	-	2021-02-10	72,45	https://ktuedu.sharepoint.com/sites/2021R_Maste	
24	23	Completed	0.00:00:37	30/10/2021 15:49:20	-	-	-	2021 m. sausio 20	915,00	https://ktuedu.sharepoint.com/sites/2021R_Maste	
25	24	Completed	0.00:00:52	30/10/2021 15:27:57	-	-	-	12.01.2021	1.481,40	https://ktuedu.sharepoint.com/sites/2021R_Maste	
26	25	Completed	0.00:02:05	30/10/2021 15:42:12	-	-	-	15.01.2021	922.0111	https://ktuedu.sharepoint.com/sites/2021R_Maste	
27	26	Completed	0.00:00:47	30/10/2021 15:26:28	-	-	-	20.01.2021	185.000	https://ktuedu.sharepoint.com/sites/2021R_Maste	
28	27	Completed	0.00:00:40	30/10/2021 15:29:58	-	-	-	2021/01/20	1.052,25	https://ktuedu.sharepoint.com/sites/2021R_Maste	
29	28	Completed	0.00:00:43	30/10/2021 15:35:55	-	-	-	19/01/21	850,50	https://ktuedu.sharepoint.com/sites/2021R_Maste	
30	29	Completed	0.00:00:45	30/10/2021 15:35:12	-	-	-	2021.03.12	7,000	https://ktuedu.sharepoint.com/sites/2021R_Maste	
31	30	Completed	0.00:00:37	30/10/2021 15:59:16	-	-	-	13.01.2021	26,37	https://ktuedu.sharepoint.com/sites/2021R_Maste	
32	31	Completed	0.00:02:21	30/10/2021 15:51:41	-	-	-	19.01.2021	1.190,46	https://ktuedu.sharepoint.com/sites/2021R_Maste	
33	32	Exception	0.00:00:00	30/10/2021 15:19:30	-	-	-	-	-	https://ktuedu.sharepoint.com/sites/2021R_Maste	File uploaded into SharePoint is not .pdf format !
34	33	Completed	0.00:00:35	30/10/2021 15:58:04	-	-	-	2021.01.21	6.8200	https://ktuedu.sharepoint.com/sites/2021R_Maste	
35	34	Exception	0.00:00:00	30/10/2021 15:19:30	-	-	-	-	-	https://ktuedu.sharepoint.com/sites/2021R_Maste	File uploaded into SharePoint is not .pdf format !
36	35	Completed	0.00:00:36	30/10/2021 15:48:43	-	-	-	11.01.2021	134,87	https://ktuedu.sharepoint.com/sites/2021R_Maste	
37	36	Completed	0.00:00:36	30/10/2021 15:59:52	-	-	-	21.01.21	152,70	https://ktuedu.sharepoint.com/sites/2021R_Maste	
38	37	Completed	0.00:00:55	30/10/2021 15:32:08	-	-	-	21.01.2021	348,00	https://ktuedu.sharepoint.com/sites/2021R_Maste	
39	38	Completed	0.00:00:33	30/10/2021 15:39:33	-	-	-	2020 gruodzio 18	2 414,97	https://ktuedu.sharepoint.com/sites/2021R_Maste	
40	39	Exception	0.00:00:00	30/10/2021 15:19:30	-	-	-	-	-	https://ktuedu.sharepoint.com/sites/2021R_Maste	File uploaded into SharePoint is not .pdf format !
41	40	Completed	0.00:00:35	30/10/2021 15:53:57	-	-	-	2020.12.17	1 103.00	https://ktuedu.sharepoint.com/sites/2021R_Maste	
42	41	Completed	0.00:01:03	30/10/2021 15:44:33	-	-	-	12.01.2021	0	https://ktuedu.sharepoint.com/sites/2021R_Maste	
43	42	Completed	0.00:00:33	30/10/2021 15:56:53	-	-	-	2021-01-19	84,07	https://ktuedu.sharepoint.com/sites/2021R_Maste	
44	43	Completed	0.00:00:58	30/10/2021 15:25:41	-	-	-	13.01.2021	430,00	https://ktuedu.sharepoint.com/sites/2021R_Maste	
45	44	Completed	0.00:00:34	30/10/2021 15:40:07	-	-	-	2021 m. sausis 22	3.52987	https://ktuedu.sharepoint.com/sites/2021R_Maste	
46	45	Completed	0.00:00:35	30/10/2021 15:58:39	-	-	-	2021 m. sausio 20	3,50000	https://ktuedu.sharepoint.com/sites/2021R_Maste	
47	46	Completed	0.00:00:43	30/10/2021 15:33:28	-	-	-	28 A, Kaunas , 2021	950.000	https://ktuedu.sharepoint.com/sites/2021R_Maste	
48	47	Completed	0.00:00:55	30/10/2021 15:46:31	-	-	-	18.11.2020	4 881,00	https://ktuedu.sharepoint.com/sites/2021R_Maste	
49	48	Completed	0.00:00:36	30/10/2021 15:56:20	-	-	-	96 ct.Data 2021	4.614,96	https://ktuedu.sharepoint.com/sites/2021R_Maste	
50	49	Exception	0.00:00:00	30/10/2021 15:56:54	-	-	-	-	-	https://ktuedu.sharepoint.com/sites/2021R_Maste	Conversion from PDF to PNG failed, file not found !
51	50	Exception	0.00:00:33	30/10/2021 15:23:42	-	-	-	-	-	https://ktuedu.sharepoint.com/sites/2021R_Maste	Blank PDF file provided !
52	51	Exception	0.00:00:00	30/10/2021 15:44:33	-	-	-	-	-	https://ktuedu.sharepoint.com/sites/2021R_Maste	Conversion from PDF to PNG failed, file not found !
53	52	Completed	0.00:00:35	30/10/2021 15:57:29	-	-	-	2021.01.11	1.5700	https://ktuedu.sharepoint.com/sites/2021R_Maste	
54	53	Completed	0.00:00:41	30/10/2021 15:39:00	-	-	-	2021-01-20	4,5342	https://ktuedu.sharepoint.com/sites/2021R_Maste	

5.2 pav. Testavimo rezultatų pavyzdys



## 5.2. Laukų atpažinimo tikslumas

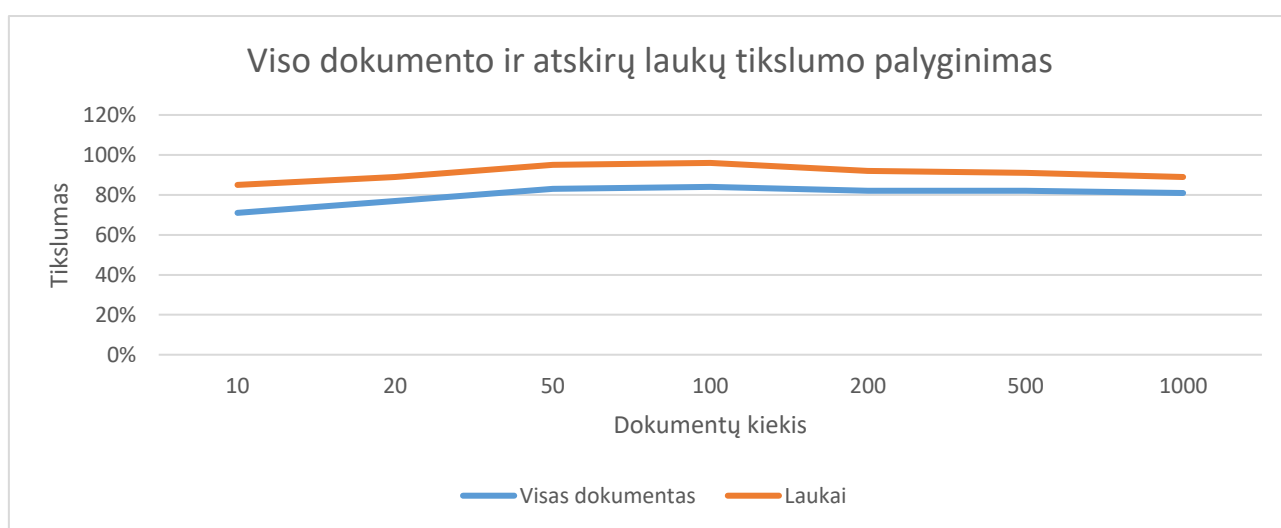
Šiame skyriuje yra aprašomas programinės įrangos roboto laukų atpažinimo tikslumo eksperimentas. Eksperimente buvo naudojami tūkstantis skirtingų dokumentų, tačiau didžiąją dalį jų buvo galima suskirstyti į kategorijas pagal dokumentą sukūrusią pusę (pardavėją), nes yra ganėtinai sunku rasti tūkstantį skirtingų įmonių finansinių dokumentų. Taigi, tu, tam tikrų kategorijų pagal pardavėją, finansiniai dokumentai yra identiškos struktūros, bet, žinoma, paremti skirtingais duomenimis.

Žemiau pateiktoje lentelėje (žr. 5.1 lentelė) matome finansinių dokumentų laukų atpažinimo tikslumą priklausomai nuo dokumentų kiekio. Tikslumo stulpelių skirtumas yra paaiškintas ankstesniame skyriuje šalia visų tyrimų apibendrintos rezultatų lentelės (žr. 4.2 lentelė). Matome, kad esant nedideliame dokumentų kiekiui tikslumas, kai dokumente yra randami visi laukai, yra ganėtinai nedidelis ir nesiekia reikalavimuose užsibrėžtos bent 80% ribos, tačiau taip yra todėl, nes imant mažą dokumentų kiekį niekada nežinoma kokie dokumentai papuls – gali būti, kad tikslumas sieks vos kelis procentus (jeigu papuls itin nepalankūs dokumentai, kuriuose programinės įrangos robotas negeba surasti visų reikalaujamų laukų), gali būti, kad tikslumas sieks 100% (jeigu papuls itin paprasti dokumentai, kurių reikalaujamus laukus surasti yra ganėtinai lengva).

5.1 lentelė Finansinių dokumentų tikslumas priklausomai nuo dokumentų kiekio

Dokumentų kiekis	Tikslumas (visas dokumentas)	Tikslumas (laukai)
10	71%	85%
20	77%	89%
50	83%	95%
100	84%	96%
200	82%	92%
500	82%	91%
1000	81%	89%

Kaip ir buvo galima tikėtis, atskirų laukų suradimo tikslumas yra didesnis nei viso dokumento laukų suradimo tikslumas (žr. 5.3 pav.). Taip yra todėl, kad surasti visus reikiamus laukus dokumente yra daug sunkiau nei surasti pavienius reikiamus laukus ir vėliau apskaičiuoti kiek bendrai laukų buvo surasta iš visų reikiamų surasti.



5.3 pav. Viso dokumento ir atskirų laukų tikslumo palyginimas



### 5.3. Apdorojimo laikų palyginimas

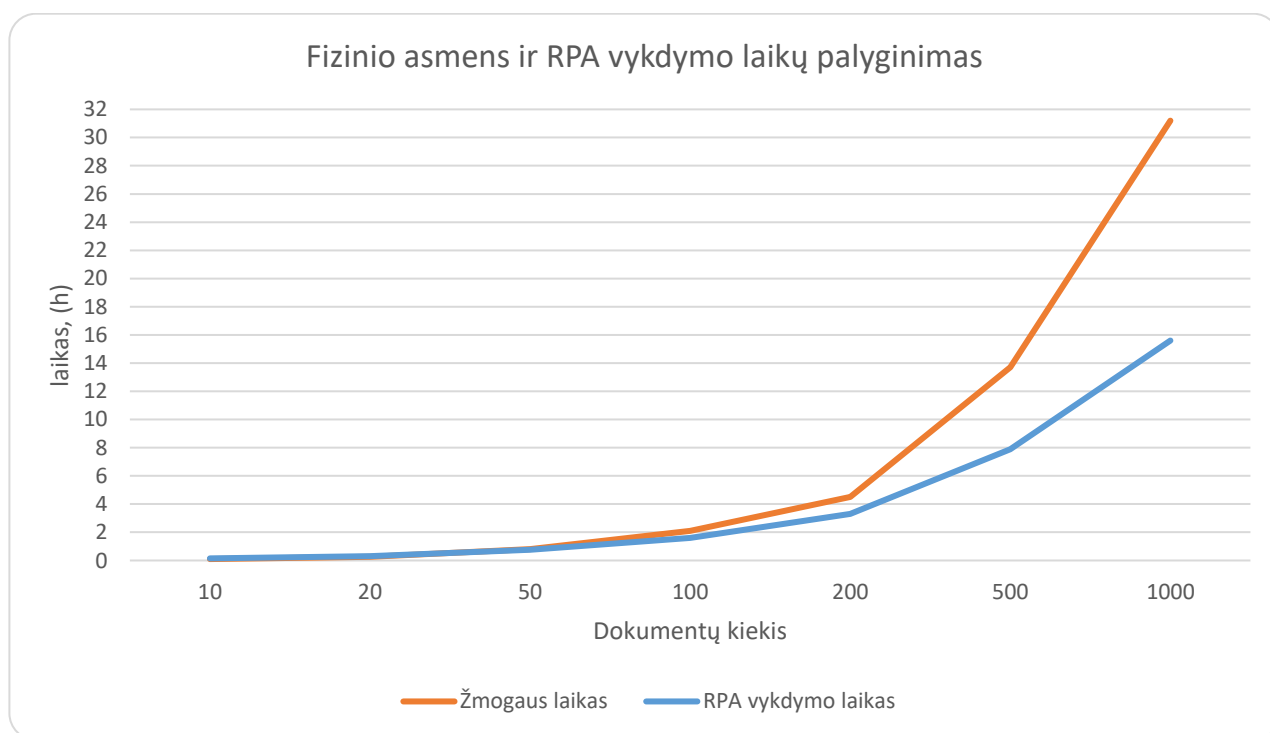
Šiame skyriuje yra aprašomas programinės įrangos roboto ir fizinio asmens darbo trukmės palyginimas laiko atžvilgiu. Pagrindinis eksperimento tikslas išsiaiškinti, ar programa gali dirbti kelis kartus greičiau nei fizinis asmuo. Programų sistemos specifikacijoje yra paminėtas reikalavimas, kad sistema privalo atlikti tą patį darbą bent du kartus greičiau nei fizinis asmuo.

Žemiau pateiktoje lentelėje (žr. 5.2 lentelė) galime pastebėti, kad fizinis asmuo pradžioje darbo gali apdoroti finansinius dokumentus netgi greičiau nei programinės įrangos robotas, tačiau fizinis asmuo dirba greičiau tik su itin nedideliais dokumentų kiekiais. Kai dokumentų kiekis tampa didesnis nei 50, programinės įrangos robotas pradeda dirbti greičiau už fizinį asmenį.

5.2 lentelė Fizinio asmens ir RPA apdorojimo laikų palyginimas

Dokumentų kiekis	Fizinis asmuo (h)	RPA (h)
10	0.10	0.15
20	0.25	0.31
50	0.80	0.77
100	2.10	1.60
200	4.50	3.30
500	13.70	7.90
1000	31.20	15.60

Tai galime vizualiai pastebėti ir žemiau esančioje diagramoje (žr. 5.4 pav.). Matome, kad didėjant dokumentų kiekiui fizinio asmens darbo laikas ženkliai padidėja lyginant su programinės įrangos roboto darbo laiku.



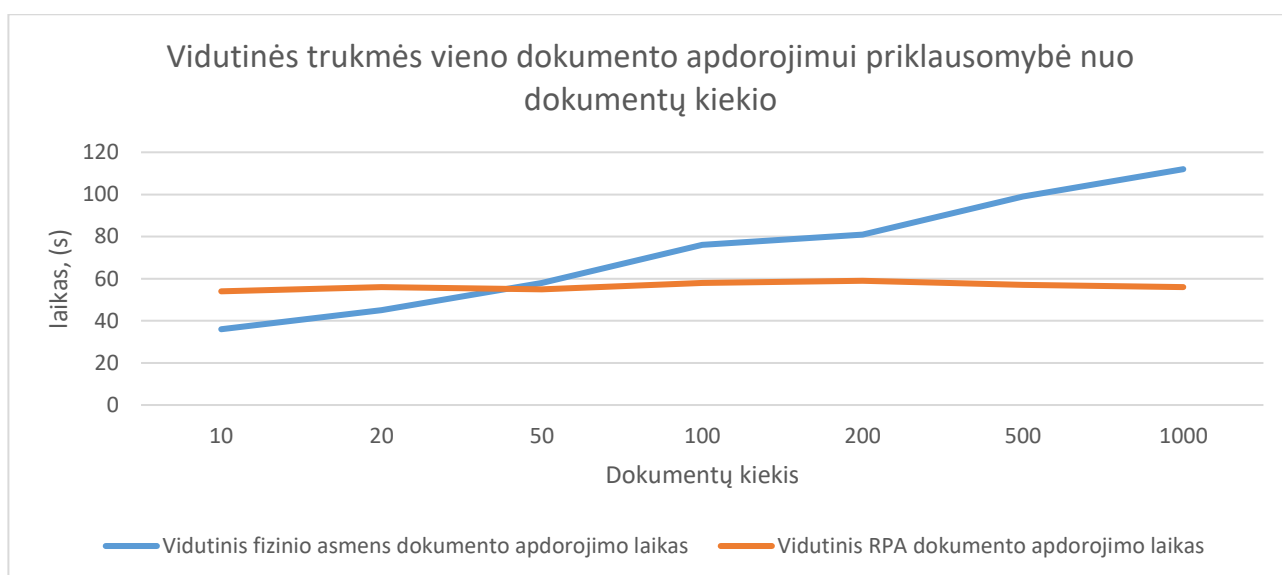
5.4 pav. Fizinio asmens ir RPA vykdymo laikų palyginimas

Šį reiškinį galime detaliau apžvelgti jeigu lyginsime ne visą darbo laiką, bet vidutinį vieno dokumento apdorojimo laiką. Žemiau pateiktoje lentelėje (žr. 5.3 lentelė) matome, kad augant finansinių dokumentų kiekiui fizinio asmens vidutinis dokumento apdorojimo laikas taip pat ženkliai auga, o programinės įrangos finansinio dokumento apdorojimo laikas išlieka ganėtinai stabilus.

### 5.3 lentelė Fizinio asmens ir RPA vidutinio dokumento apdorojimo laiko palyginimas

Dokumentų kiekis	Vidutinis fizinio asmens dokumento apdorojimo laikas, s	Vidutinis RPA dokumento apdorojimo laikas, s
10	36	54
20	45	56
50	58	55
100	76	58
200	81	59
500	99	57
1000	112	56

Tai galime vizualiai pastebėti ir žemiau esančioje diagramoje (žr. 5.5 pav.). Čia itin aiškiai matyti, kad fizinis asmuo iki 50-ies dokumentų ribos finansinius dokumentus apdoroja greičiau negu programinės įrangos robotas, tačiau perkopus 50-ies finansinių dokumentų ribą fizinis asmuo pradeda apdoroti finansinius dokumentus lėčiau negu programa. Negalime teigti, kad programa pradeda dirbti greičiau, nes ji dirba visada tolygiai, tačiau galime teigti, kad fizinio asmens darbo tempas sulėtėja ir kuo daugiau dokumentų, tuo lėčiau fizinis asmuo pradeda dirbti.



5.5 pav. Vidutinės trukmės vieno dokumento apdorojimui priklausomybė nuo dokumentų kiekio

Šis eksperimentas įrodo, kad programinės įrangos robotas eliminuoja žmogiškuosius faktorius tokius kaip: stresas, nuovargis ar įtampa. Fizinis asmuo negali visada dirbti tokiu pačiu tempu kaip programa. Verta paminėti ir tai, kad tokį darbą (tūkstančio finansinių dokumentų apdorojimą) atlikti žmogui kainuotų 3-4 darbo dienas, nes žmogus, tai ne robotas, jis negali dirbti be pertraukų, poilsio laiko ar visą parą be sustojimo. Teoriškai tokį darbą pradėjus pirmadienį tikėtina, kad fizinis asmuo šį darbą pabaigs ketvirtadienį (geriausiu atveju trečiadienį), o programinės įrangos robotas pabaigs tą patį pirmadienį, nes gali dirbti be sustojimo, be pertraukų ar poilsio laiko.

Taigi, nors ir matome, kad galiausiai robotas apdoros finansinius dokumentus bent dvigubai greičiau nei fizinis asmuo, bet įvertinus darbo dienos trukmę galime teigti, kad programinės įrangos robotas atliks darbą bent 3 ar keturis kartus greičiau nei fizinis asmuo. Esant mažesniai dokumentų kiekiui fizinis asmuo gali dirbti greičiau nei programa, tačiau šis robotas buvo kuriamas didelių dokumentų kiekių apdorojimui. Eksperimentas įrodo, kad programinės įrangos robotas tikrai gali apdoroti finansinius dokumentus bent du kartus greičiau nei fizinis asmuo, kai yra kalbama apie dideles duomenų imtis.

## 6. IŠVADOS

1. Atliktos literatūros ir specializuotų RPA produktų analizės metu buvo nustatyta, kad šiuo metu nėra pateikto sprendimo (sistemos, platformos), kuris nebūtų priklausomas nuo dokumento struktūros, robotizuotam, t.y. neįtraukiant kitų įrankių ir fizinių asmenų finansinių dokumentų turinio nuskaitymui.
2. Įvertinus automatizavimo galimybes, buvo sukurtas programinės įrangos robotas, skirtas robotizuoti finansinių dokumentų turinio nuskaitymo procesą, leidžiant palengvinti fizinių asmenų darbą panaikinant rutinines užduotis.
3. Sukurta ir sėkmingai realizuota robotizuoto proceso sistemos ir dirbtinio intelekto sąsaja, kuri leido sklandžiai apjungti RPA ir dirbtinio intelekto technologijas.
4. Atliktas sėkmingas sukurtos sistemos verifikavimas ir validavimas įtraukiant testinius scenarijus leido pagrįsti sistemos efektyvumą. Lyginant žmogaus ir roboto užduočių įvykdymo rezultatus laiko atžvilgiu, buvo pastebėta, kad roboto pranašumas tiesiogiai didėja didėjant finansinių dokumentų apdorojimui kiekiui. Finansinių dokumentų kiekį didinant nuo 500 iki 1000 roboto greitis išauga nuo 2 iki 3.9 karto lyginant su žmogaus įvykdymo greičiu.

Pasiekti baigiamojo darbo tyrimo rezultatai buvo publikuoti. Publikacijos pavadinimas: *“Intelligent Invoice Documents Processing Employing RPA Technologies”*, 2022.

## 7. LITERATŪRA

- [1] M. Rouse, “OCR - optical character recognition,” 04 2019. [Tinkle]. Prieiga internete: <https://searchcontentmanagement.techtarget.com/definition/OCR-optical-character-recognition>. [Kreiptasi 27 balandžio 2022].
- [2] A. Solonskyi, “The Manifest,” 27 01 2020. [Tinkle]. Prieiga internete: <https://themanifest.com/it-services/resources/how-business-benefit-optical-character-recognition>. [Kreiptasi 27 balandžio 2022].
- [3] C. Woodford, “Explain That Stuff,” 05 2018. [Tinkle]. Prieiga internete: <https://www.explainthatstuff.com/how-ocr-works.html>. [Kreiptasi 27 balandžio 2022].
- [4] M. Cheriet, N. Khama, C. Suen and C. L. Liu, Character Recognition Systems: A Guide for Students and Practitioners, Wiley, 2017.
- [5] T. A. Nartker, G. Nagy and S. V. Rice, Optical Character Recognition: An Illustrated Guide to the Frontier, Springer Science & Business Media, 2012.
- [6] H. F. Schantz, The history of OCR, optical character recognition, Recognition Technologies Users Association, 1982.
- [7] S. Impedovo, Fundamentals in Handwriting Recognition, Springer Science & Business Media, 2012.
- [8] T. Plötz and G. A. Fink, Markov Models for Handwriting Recognition, Springer Science & Business Media, 2012.
- [9] “Nanonets,” NanoNet Technologies Inc., 2020. [Tinkle]. Prieiga internete: <https://nanonets.com/>. [Kreiptasi 27 balandžio 2022].
- [10] “Conexiom,” Ecmarket Inc., 2020. [Tinkle]. Prieiga internete: <https://conexiom.com/>. [Kreiptasi 27 balandžio 2022].
- [11] “Klippa,” Klippa App B.V. - KVK, 2020. [Tinkle]. Prieiga internete: <https://www.klippa.com/>. [Kreiptasi 27 balandžio 2022].
- [12] “Logikia,” Logikia, 2017. [Tinkle]. Prieiga internete: <https://www.logikia.ca/>. [Kreiptasi 27 balandžio 2022].
- [13] “Fast Four,” Fast Four, 2018. [Tinkle]. Prieiga internete: <https://www.fastfour.com/en/>. [Kreiptasi 27 balandžio 2022].
- [14] “Tesseract OCR,” 2020. [Tinkle]. Prieiga internete: <https://tesseract-ocr.github.io/>.
- [15] M. M. Andreas, “Simple OCR with Tesseract,” Medium, 12 07 2020. [Tinkle]. Prieiga internete: <https://towardsdatascience.com/simple-ocr-with-tesseract-a4341e4564b6>. [Kreiptasi 27 balandžio 2022].
- [16] P. Siano, C. Cecati, H. Yu and J. Kolbusz, Real Time Operation of Smart Grids via FCN Networks and Optimal Power Flow, IEEE Transactions on Industrial Informatics, 2012.

- [17] R. Gandhi, “R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms,” Medium, 2018. [Tinkle]. Prieiga internete: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>. [Kreiptasi 27 balandžio 2022].
- [18] X. Fu, E. Ch'ng, U. Aickelin and S. See, CRNN: A Joint Neural Network for Redundancy Detection, 2017 IEEE International Conference on Smart Computing (SMARTCOMP), 2017.
- [19] J. Brownlee, “A Gentle Introduction to Long Short-Term Memory Networks by the Experts,” Machine Learning Mastery Pty. Ltd., 2017. [Tinkle]. Prieiga internete: <https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/>. [Kreiptasi 27 balandžio 2022].
- [20] A. Kang and H. Atul, “Optical Character Recognition Pipeline: Text Recognition,” The AI Learner, 2018. [Tinkle]. Prieiga internete: <https://theailearner.com/tag/attention-ocr/>. [Kreiptasi 27 balandžio 2022].
- [21] L. Gong, M. Thota, M. Yu, W. Duan, M. Swainson, X. Ye and S. Kollias, “A novel unified deep neural networks methodology for use by date recognition in retail food package image,” Springer Nature Switzerland AG., 2020. [Tinkle]. Prieiga internete: <https://link.springer.com/article/10.1007/s11760-020-01764-7>. [Kreiptasi 27 balandžio 2022].
- [22] C. D. Costa, “Top Programming Languages for Data Science in 2020,” Towards Data Science, 24 08 2020. [Tinkle]. Prieiga internete: <https://towardsdatascience.com/top-programming-languages-for-data-science-in-2020-3425d756e2a7>. [Kreiptasi 27 balandžio 2022].
- [23] “Tesseract OCR,” Github, 2020. [Tinkle]. Prieiga internete: <https://tesseract-ocr.github.io/>. [Kreiptasi 27 balandžio 2022].
- [24] “Keras,” Keras, 2020. [Tinkle]. Prieiga internete: <https://keras.io/>. [Kreiptasi 27 balandžio 2022].
- [25] “Theano,” LISA lab., 2017. [Tinkle]. Prieiga internete: <http://deeplearning.net/software/theano/tutorial/>. [Kreiptasi 27 balandžio 2022].
- [26] “Torch,” Torch, 2020. [Tinkle]. Prieiga internete: <http://torch.ch/>. [Kreiptasi 27 balandžio 2022].
- [27] “Infer.NET,” Microsoft, 2020. [Tinkle]. Prieiga internete: <https://dotnet.github.io/infer/>. [Kreiptasi 27 balandžio 2022].
- [28] “Blue Prism: Robotic Process Automation – RPA.,” [Tinkle]. Prieiga internete: Prieiga internete: <https://www.blueprism.com/>. [Kreiptasi 27 balandžio 2022].
- [29] “UiPath: Robotic Process Automation.,” [Tinkle]. Prieiga internete: Prieiga internete: [www.uipath.com](http://www.uipath.com). [Kreiptasi 27 balandžio 2022].
- [30] “Automation Anywhere: Global RPA Solutions.,” [Tinkle]. Prieiga internete: Prieiga internete: <https://www.automationanywhere.com/>. [Kreiptasi 27 balandžio 2022].
- [31] ““Blue Prism vs Automation Anywhere vs UiPath”.,” [Tinkle]. Prieiga internete: Prieiga internete: <https://www.rpatraining.co.in/blue-prism-vs-automation-anywhere-vs-uipath/>. [Kreiptasi 27 balandžio 2022].

## 8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

### Santrumpos:

1. RPA – (angl. *Robotic process automation*) – Robotinė procesų automatika (RPA) yra verslo procesų automatizavimo technologijos forma, pagrįsta metaforiniais programinės įrangos robotais.
2. OCR – (angl. *Optical character recognition*) – Optinis simbolių atpažinimas arba optinių simbolių skaitytuvas (OCR) yra elektroninis ar mechaninis spausdinto, ranka rašyto ar atspausdinto teksto vaizdų pavertimas skaitmeniniu tekstu.
3. CNN – (angl. *Convolutional neural network*) – Konvoliutinis neuroninis tinklas (KNN) yra dirbtinio neuroninio tinklo rūšis, dažniausiai naudojama vaizdų atpažinimui, objektų radimui ir identifikavimui nuotraukose ir pan.
4. BP – Blue Prism.
5. PĮ – programinė įranga.
6. PDD (angl. *Process Design Document*) – proceso projektavimo dokumentas, skirtas aprašyti proceso žingsnius bei vaizdiškai juos atvaizduoti.
7. UAT (angl. *User Acceptance Testing*) – paskutinė programinės įrangos testavimo fazė, kai programinės įrangos funkcionalumas lyginamas su sistemos specifikacija.
8. BAT (angl. *Business Acceptance Test*) – paskutinė programinės įrangos testavimo fazė, kai programinės įrangos funkcionalumas lyginamas su užsakovų reikalavimais.
9. RAM (angl. *Random Access Memory*) – kompiuterio operatyvioji atmintis.
10. UML (angl. *Unified Modelling Language*) – grafinė sistemų projektavimo ir specifikacijų kūrimo kalba, skirta projektuoti objektines programų sistemas.
11. RDP (angl. *Remote Desktop Protocol*) – „Microsoft“ tinklo protokolas, naudojamas ryšiui su nutolusiu kompiuteriu užmegzti.
12. PDF (angl. *Portable Document Format*) – atviro standarto formatas, skirtas elektroniniam dvimačiam dokumentui atvaizduoti.
13. PNG (angl. *Portable Network Graphics*) – bitų masyvo formatas, kuris suglaudinas „be nuostolių“.
14. TXT – tai tekstinio failo, kuriame tekstas išdėstytas eilučių pavidalu, plėtinys.

### Terminai:

1. Mašininis mokymasis – (angl. *machine learning*) – dirbtinio intelekto metodų klasė, kuriai būdingas ne tiesioginis problemos sprendimas, o mokymasis, kaip pritaikyti daugelio panašių problemų sprendimus. Apima metodų kūrimą, mokinančių kompiuterius „mąstyti“. Tai yra programų kūrimo būdas, kai sukurta sistema prisitaiko prie duomenų („apsimoko“).
2. Gilusis mokymasis – (angl. *deep learning*) – yra mašininio mokymosi atšaka, kur kiekvienas sluoksnis išmoksta vis abstraktesnius duomenų bruožus. Jis vadinamas giliu, nes turi daugiau nei kelis sujungtus neuronų sluoksnius. Modernus gilusis mokymasis dažniausiai turi tarp dešimties ir kelių šimtų sluoksnių, kurie išmoksta bruožus tiesiai iš duomenų.
3. Sprendimų medžiai – (angl. *decision trees*) – metodas naudoja sprendimų medį kaip prognozavimo modelį, kuris sujungia pastabas apie objektą (pavaizduojama šakomis) su išvadomis apie elemento siektina verte (vaizduojama lapais). Tai yra vienas iš prognozavimo modeliavimo metodų, naudojamų statistikoje, duomenų išgavime (angl. *data-mining*) ir mašiniame mokymesi (angl. *machine learning*).

## Priedai

### 1. priedas. Užpildyta apklausos anketa

Projekto informacija

Projekto Pavadinimas: Finansinių dokumentų automatinis turinio nuskaitymas naudojant dirbtinio intelekto ir RPA technologijas

Klientas: Agnė Paulauskaitė – Tarasevičienė

Pradžios data: 2020 – 09 - 01

Pabaigos data: 2022 – 02 - 01

Projekto tipas: Programinė įranga

Kliento telefonas: +37064100913

**Įvertinkite žemiau pateiktas projekto charakteristikas skalėje nuo 1 iki 5.**

**1 yra žemiausias įvertinimas, o 5 aukščiausias. Jei charakteristikas jums netinka pažymėkite „X“.**

- 5 programinės įrangos palaikymo adekvatumas.
- 4 techninės įrangos palaikymo adekvatumas.
- 4 komandos patirtis dirbti su naudojama technologija.
- 3 komandos patirtis dalykinėje srityje.
- 4 komandos patirtis su projekto procesu ir metodais.
- 5 technologijos tinkamumas projektui.
- 5 konfigūracijų valdymo efektyvumas.
- 4 kokybės užtikrinimo efektyvumas.
- 5 vartotojo reikalavimų aiškumas.
- X reikalavimų pasikeitimų kiekis.

**Įvertinkite projekto sėkmę skalėje. 5 - labai patenkintas, 4 - patenkintas 3 -neutralus, 2-nepatenkintas, 1-labai nepatenkintas.**

- 5 komandos pasitenkinimas projekto rezultatu.
- 5 komandos pasitenkinimas projekto vykdymu.
- 4 vadovybės pasitenkinimas projekto rezultatu.
- 5 vadovybės pasitenkinimas projekto vykdymu.
- 4 kliento pasitenkinimas projekto rezultatu.
- 5 kliento pasitenkinimas projekto vykdymu.
- 4 projekto pasiekimas planuotų tikslų.
- 5 projekto atitikimas planuotų tvarkaraščių.
- 5 projekto pasiekimas planuotų funkcionalumo tikslų.

**Pažymėkite projekto apžvalgų tipus kurie vyko:**

Taip reguliarios projekto apžvalgos (kaip dažnai: kartą per mėnesį)

Ne reguliarūs komandos progreso susitikimai (kaip dažnai: \_\_\_\_\_)

Ne gairių peržiūros, gairės: \_\_\_\_\_

### **Pažymėkite sukurtus produktus:**

Taip Projekto paraiška

Taip Reikalavimai arba funkcinė specifikacija

Taip Architektūros aprašymas

Ne Infrastruktūros projektas

Taip Produkto projektas

Taip Kodas

Taip Testiniai atvejai

Taip Kita. (Identifikuokite: Projektavimo metodologijos ir technologijų analizė, Naudotojo dokumentacija)

### **Pažymėkite vykusias technines peržiūras.**

Taip Paraiškos arba darbo apimties peržiūra

Taip Reikalavimų ar funkcinės specifikacijos peržiūra

Taip Architektūros peržiūra

Taip Projekto peržiūra

Taip Kodo peržiūra

Taip Testavimo atvejų peržiūra

Taip Kita. (Identifikuokite: Projektavimo metodologijos ir technologijų analizės, Naudotojo dokumentacijos peržiūros)

### **Identifikuokite tris dalykus, kurie buvo atlikti teisingai ir turėtų būti tęsiami toliau.**

1. PDF konvertavimas į PNG naudojant ImageMagick

2. PNG konvertavimas į TXT naudojant Tesseract

3. Informacijos suradimas tekste pagal apibrėžtas taisykles

### **Identifikuokite tris dalykus, kurie buvo atlikti blogai ir turėtų jų atlikimas būti pagerintas.**

1. Dirbtinio intelekto integracija RPA programinės įrangos kūrimo aplinkoje

2. Nėra

3. Nėra

### **Į kokias neįprastas aplinkos įtakas (kurios įtakojo ar galėjo įtakoti projektą) reiktų atsižvelgti analizuojant projekto eigą?**

Reiktų atsižvelgti į naujausias vaizdų atpažinimo technologijas ir dirbtinio intelekto sprendimų priėmimo metodologijas.

### **Kaip vertinate projekto rizikos valdymą?**

Rizikos valdymas atliktas puikiai. Buvo numatytos dešimt rizikų, jų tikimybės



bei galimas poveikis. Aprašyta, kaip suvaldyti rizikas joms pasitvirtinus.

**Kurias pagrindines rizikas identifikuotas projekto pradžioje jums sekėsi sėkmingai valdyti?**

Naujų reikalavimų iš užsakovų pusės atsiradimas, per dideli užsakovų lūkesčiai kuriamai sistemai, žmogiškųjų išteklių sumažėjimas, programos neatitikimas pagal apibrėžtą specifikaciją, versijų pokyčio problemos, techninės įrangos gedimai, pokyčiai naudojamose trečiųjų šalių bibliotekose, kenkėjiškų programų padaroma žala.

**Kurių rizikų jums nesisekė suvaldyti projekto metu, kodėl?**

Atsiras panaši arba tokia pati programinė įranga rinkoje, kitokios programinės įrangos, sprendžiančios problemą geriau, atsiradimas. Šios rizikos visiškai nepriklauso nei nuo kūrėjų, nei nuo užsakovų, todėl jų pasitvirtinimo suvaldyti neįmanoma. Buvo paruoštas veiksmų planas joms pasitvirtinus.

**Kokios neidentifikuotos rizikos tapo problemomis? Kaip jų būtų buvę galima išvengti?**

Neidentifikuotų rizikų, kurios vėliau tapo problema, projekto eigoje nepasitaikė.

## 2. priedas. Straipsnis

# Intelligent Invoice Documents Processing Employing RPA Technologies

Vilius Kerutis<sup>1</sup> and Dalia Calneryte<sup>1</sup>[0000-0003-4185-0397]

<sup>1</sup> Faculty of Informatics, Kaunas University of Technology, Studentų st. 50, Kaunas, Lithuania  
vilius.kerutis@ktu.lt, dalia.calneryte@ktu.lt

**Abstract.** The applications of Robotic Process Automation (RPA) are many and growing every year, covering banking and financial operations, insurance functions, auditing processes, logistics planning services and more, but automating invoice processing is still more challenging. However, different structure, the variety of keywords, the abundance of types of information makes the detection and retrieval of the information very complicated. Therefore, the currently solutions currently work well with predefined document structures. The aim of this study is to develop a solution based on RPA, Optical character recognition (OCR) technologies and deep learning methods for automated invoice processing without being bound to a specific document structure. The study focuses on five key fields of invoices that are most important to identify and read. The results showed that almost 83% of all invoices used in the experiments were processed correctly. In terms of the results for the detection of individual field information, the best results were found for the "date" and "total amount" fields, with 93.21% and 87.81% respectively, but the detection of the "seller" and "buyer" fields is complicated and requires extensive additional research. Experiments evaluating the document processing time of human the developed robot showed that the performance of human document processing decreases with the volume of documents processed, while the RPA time is almost constant. RPA is 1.76 times more efficient with 500 documents and 2 times faster with 1000 documents.

**Keywords:** robotic process automation, deep learning, optical character recognition.

## 1 Introduction

The automated reading of the content of financial documents using AI and software robotics technologies is a hot topic in today's business world. Robotic Process Automation (RPA) is a new technology utilizing software robots for automation of repetitive back-office processes. It can be used in a wide range of applications industries and business functions [1], [2], [3]. The main objectives of RPA are: (1) to eliminate the need for humans to work on repetitive tasks; (2) to speed up the whole process; (3) eliminate human error; and (4) maximize the robot's accuracy. The applications for RPA are numerous and growing every year, but some of the most common RPA tasks

include banks and financial operations [4],[5] insurance functions [6], auditing [7], tour enquiries [8] and logistics and planning services [9] and invoices [10]. The aim of such a service, targeted at financial documents, is to automatically scan and identify all the relevant fields of a document, regardless of the structure of the financial document. Currently, the solutions offered on the market are still dependent on the structure of the document, i.e. if a financial document is fed with a different structure than the one with which the solution has been developed, the effectiveness of the target information recognition decreases significantly [11]. If the structure of the financial document is different, the model no longer ensures correct recognition of the fields. In most cases, only a few standard fields are recognized, which are very easy to find, such as the final amount of the order, which is usually written at the bottom of the document, below the “total amount” of the order. Therefore, it makes sense to develop a solution that is able to accurately identify the information fields specified, regardless of the structure of the financial document. It is appropriate to integrate advanced artificial intelligence technologies together with RPA technologies [12], [13]. The potential customers of the system being developed are medium- to large-sized companies that sell goods or services and issue invoices. The most critical problem for such companies is the time spent on processing financial documents. The product could contribute to solving this problem by enabling the staff responsible for processing invoices to engage in other creative or intellectual activities with higher added value. This would increase the productivity and efficiency of the staff and their competence.

The most popular and commonly used solution in the business world for processing financial documents is Optical character recognition (OCR) [14]. OCR technology converts an image into text, thus extracting the necessary contextual information [15]. With the text, you can search, filter the text, or simply recognize what information is being presented, reuse that information elsewhere, etc. Identification itself can be complicated by the variety of fonts and writing styles that exist in the digital space alone, and if there are also recognizable images of handwritten text, the processing of the image and the identification of the text is even more complex.

Two methods are used to recognize as many different images as possible and convert them into text:

1. Pattern recognition. The principle is simple – the OCR logic is passed a variety of patterns for each known character. When the model receives a new image for character recognition, it applies the patterns until a match is found and the image is converted to text.
2. Feature detection. This method is much more complex than the first, but is much more widely used. It follows the rules of line drawing, for example, if there are two lines intersecting at one point, and below them there is another line crossing the previous two lines at two points, then this can be said to be the "A" symbol. This method is closely related to neural networks, and it is this principle that is used for symbol recognition.

Recognition of handwritten text in an image is performed using only feature detection [16], [17]. However, very high identification accuracies should not be expected, as handwritten text recognition is sometimes a difficult task even for humans.

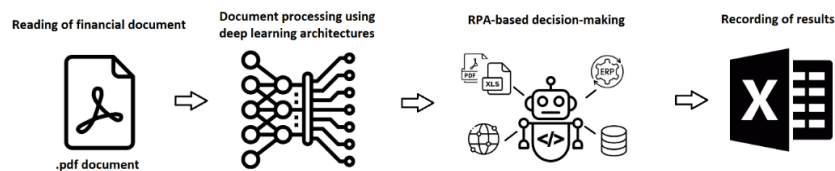
Hence, the basic principle of such OCR-based solutions is very similar: a document is fed to the system, all the fields that should be identified are marked in the document, the OCR technology scans the marked areas and the marked document is fed to the recognition model for training. Real new documents are then transmitted, and the model must be able to identify the information provided.

Tesseract is an open OCR engine available under the Apache 2.0 license [18]. This engine can be used directly or (for software development purposes) by using an API to extract printed text from images [19]. For the Tesseract engine, the image is fed as an input, which is transferred onto an adaptive grid, and finally the image is converted into binary type. The contours are used to try to find the lines and words of text. Once the text lines and words are found, the character contours are combined into complete words. These words pass two levels of word recognition, and the extracted text is output as the result of the Tesseract engine.

In the context of practical solutions, the commercial products currently on the market are quite similar - all of them automate the processing of documents (mainly financial ones), all of them use OCR technology and deep learning algorithms (Conexiom [20], Klippa [21], Logikia [22], FastFour [23], and others). While these tools may appear to be the same - performing the same function - individually they all have their own advantages and disadvantages. In terms of user interface usability, price, speed of operation, Nanonets is the leader, specializing mainly in financial documents, providing a very user-friendly and understandable user interface for every user, free trials and running quite fast compared to other solutions [24]. Experiments on 20 documents with slightly different structures showed a pattern field recognition accuracy of 50%.

## 2 Materials and Methods

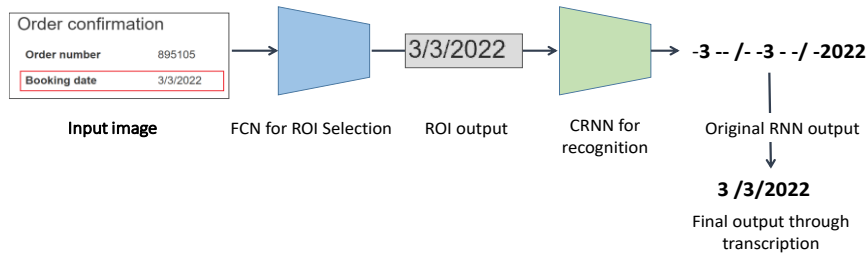
The idea of the developed project that would meet the customer's needs is presented in **Fig. 1**. The financial document is scanned using Adobe Acrobat Reader. The convolutional neural network then is used to find the fields required for proper document processing. Subsequently, the found fields and their information are passed to a software robot that performs rule-based decisions, that is, decides if the found fields meet the expected ones. Finally, after checking all the fields, the found information is saved in an MS Excel .xlsx file, which is intended to be stored in a SharePoint aggregated environment (this is not essential and can be adjusted, so it is not included in the layout of the project idea). The document is successfully flagged, and the process is repeated while there are financial documents to be processed.



**Fig. 1.** Product idea layout at a high level.

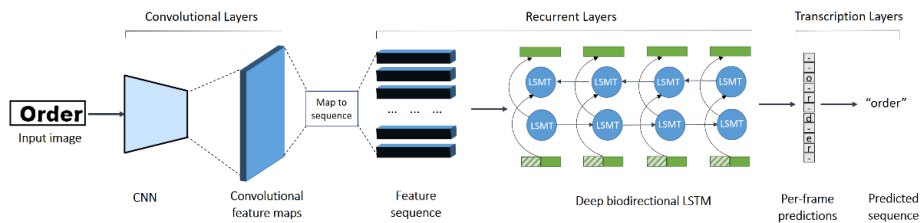
The processing of visual information is based on convolutional neural networks. The target information recognition algorithm consists of two steps:

1. Target object detection and identification. Typically, OCR-based commercial decisions retrieve all the information in a financial document, but this system uses CNN to search for and identify only target objects that are required by the accounting system. FCN [25], R-CNN [26] convolutional neural network architectures were used to identify and segment the target objects. Their positions are then identified and the distances between all objects are calculated using certain metrics. Fully convolutional network (FCN) is responsible for finding ROI (expiration date) by keywords ("total sum", "order number", "booking data", etc.). It acts as a filter to recognize the location of the image containing the keywords, so that the recognition task is performed on that particular small fragment of the image rather than the entire region of the image to reduce computational time [27]. In addition, limiting to only the detected expiration date locations enables to avoid recognition errors caused by interference with other text in various parts of the image (see **Fig. 2**).



**Fig. 2.** Sequence for text recognition process.

2. Text recognition. CRNN (Convolutional Recurrent Neural Networks) [28], with LSTM (Long Short-Term Memory architecture) [29] and Attention-OCR [30] were used for text recognition. According to scientific research, use of these architectures enable to achieve high-precision results in identifying text and recognize textual information in different formats. The CRNN model uses a convolutional neural network (CNN) to extract visual features that are transformed and delivered to a long short-term memory network (LSTM). The LSTM output is then associated with the location of the character labels in the dense layer (see **Fig. 3**).



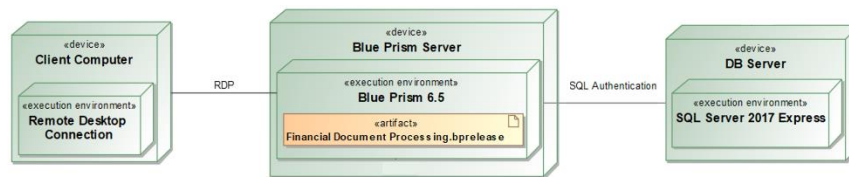
**Fig. 3.** Application of deep learning architectures in the system.

Decision trees are used for final processing of the information found. Decision trees are used to solve classification problems. The number of classification classes is defined in advance and corresponds to the number of fields in the accounting system.

## 2.1 Employed Technologies

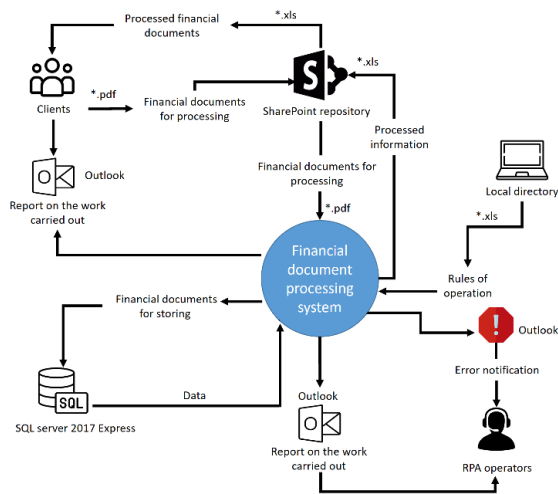
The problem is solved in two stages. In the first stage, the image (.pdf files or photos with text) is processed. The text is derived from the image using artificial intelligence and other methods. In the second stage, for the extracted text, RPA will make a rules-based decision to decide whether what was expected has been found and selected. Artificial intelligence solutions have been implemented using the Python programming language, as it is accepted to be the most convenient and capable language when working with a wide variety of data in different formats. The Tensorflow [31] software development platform, which is specifically designed to implement such projects, has also been used.

RPA technologies are used in the second stage. There are many robotic tools and more and more new features are emerging every year (e.g. artificial intelligence), but three leaders stand out: Blue Prism [32], UiPath [33], and Automation Anywhere [34] Blue Prism technology was chosen for this study.



**Fig. 4.** Deployment diagram.

The deployment diagram is given in **Fig. 4**. The diagram shows that the process is stored on the Blue Prism server, which is accessed using RDP protocol via remote desktop. Blue Prism server directly communicates with SQL Server database via SQP Authentication connection. This diagram shows everything that is needed from a technical point of view for the process to execute.



**Fig. 5.** Project context diagram.

The developed software is designed for operating in Windows 10 with Blue Prism v6.10 software installed. The only difference is that for the development the Blue Prism development environment is used, and after the software robot is developed, the Blue Prism production environment is used. The project context diagram is given in **Fig. 5**.

## 2.2 Developed Algorithm

The pseudo-code for the system operation algorithm that describes the entire sequence of the processes is provided below.

### Pseudocode: the algorithm for the system

```

Initiliazze Blue Prism PDF_list
Initialize String textFromPDF = StringEmpty()
Loop PDF from PDF_list
    PowerShell instance = new PowerShell()
    instance.ChangeDirectory(PDF_list.getDirectory())
    instance.ConvertPDF2PNG(PDF.getFilePath()) //Usage of ImageMagick
    instance.Invoke()
    if error then //Any possible error
        instance.Close()
        break
    if noFileError then //File was deleted or interrupted
        instance.TagItem("business exception")
        instance.ChangeStatus("Exception")
        instance.Close()
        break
    Environment.Sleep(10)
    Integer PNG_index = 0
    Loop PNG from PNG_list
        instance.ConvertPNG2TXT(PNG.getFilePath() + index++)
        instance.Invoke()

```

```

if noFileError then
    instance.ConvertPNG2TXT(PNG.getFilePath()) //Tesseract-OCR
    textFromPDF = FileManagement.Read(TXT)
    break
if error then
    instance.Close()
    break
instance.Close()
Loop TXT as TXT_list
    textFromPDF = textFromPDF + FileManagement.Read(TXT)
    FileManagement.Delete(TXT.getDilePath())
GarbageCollector.DeleteAllFiles(PDF_list.getDirectory())

```

Before converting .pdf files to .png and later to .txt file formats, .pdf are already loaded from the SharePoint repository to the local environment. The developed algorithm, which is given above, specifies how .pdf files are transformed into a text format variable which can be fully interpreted by a software robot.

First, a list is received from the RPA software (Blue Prism) about queuing pending processing files. The files in .pdf format are taken from the list one after the other and processed. Power Shell is used to work with files. Thus, a Power Shell object is created and the working directory (the one in which all .pdf files are downloaded) is changed. After changing the directory, it is possible to start the converting from .pdf to .png. A combination of Power Shell and ImageMagick is used for this. ImageMagick is a tool that converts .pdf files to .png format. Power Shell is used to activate the ImageMagick tool. ImageMagick creates as many .png files as there were .pdf pages, so later converting from .pdf to .txt requires iteration through all .png files. If an error occurs in this step, the job is completed, the Power Shell object is closed, and it is resumed from the next .pdf file.

The program then iterates through all the .png files and tries to convert them to .txt format. A Tesseract-OCR tool is used for this. It is activated by the use of Power Shell. If the .pdf file had only one page, then the .png file was created without an index at the end. During the first try the error occurs and the software tries to convert the file without an index at the end and the file is converted to .txt format and immediately stored into the final variable. After converting all .png files to .txt format, all .txt files are read one after the other into the final variable and deleted. If an error occurs in this step, the job is completed, the Power Shell object closes, and continues from the next .pdf file. Finally, all remaining files in the directory are deleted and the job is completed. The algorithm is easy to understand with an example. If we have one .pdf file that contains 5 pages, then 5 .png files will be created, which will later be converted to 5 .txt files. A total of 9 different files if the .pdf file holds 5 pages. 5 .txt files are merged into one text variable and eventually all files are deleted.

### 3 Experiments

250 different types of invoices were used to train the system. Testing was done with 50 new invoices from different suppliers and different structures. In the experiments, five information fields were searched for: "Seller", "Buyer", "Invoice No", "Date" and "Total amount". An example of a prepared date rule file is shown in **Fig. 6**. This rules

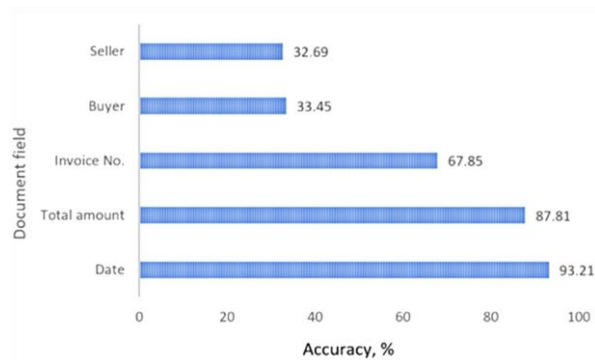


file describes the individual rules for each field on different sheets. The rule syntax corresponds to the regular expression most used in search models. The software robot applies each rule one after the other to the analyzed text of the document and looks for a match. If there is more than one match, in case of date search of the document, the match is the date with the lowest start index, that is the one at the top of the document. This is because the date (regardless of the structure of the document) is usually written at the beginning of the document. The reverse case is with the final amount. The final amount search algorithm works in the same way, but if more than one match is found, the final price with the highest start index is used because it is usual (regardless of the document structure) that the final amount is usually written at the end of the document. Other fields are searched on a similar principle.

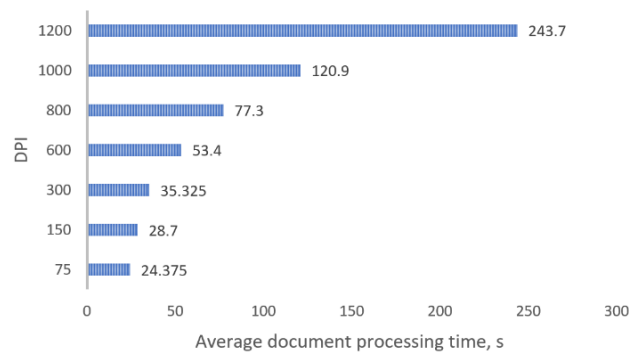
Rule	Example
<code>\d(2)\.\s\d(2)\.\s\d(4)</code>	13.01.2021
<code>\d(2)\.\d(2)\.\d(4)</code>	13.01.2021
<code>\d(4)\.\s\d(2)\.\s\d(2)</code>	2021.01.13
<code>\d(4)\.\d(2)\.\d(2)</code>	2021.01.13
<code>\d(2)\-\s\d(2)\-\s\d(4)</code>	13-01-2021
<code>\d(2)\-\d(2)\-\d(4)</code>	13-01-2021
<code>\d(4)\-\s\d(2)\-\s\d(2)</code>	2021-01-13
<code>\d(4)\-\d(2)\-\d(2)</code>	2021-01-13
<code>\d(2)\s-\s\d(2)\s-\s\d(4)</code>	13 - 01 - 2021
<code>\d(4)\s-\s\d(2)\s-\s\d(2)</code>	2021 - 01 - 13
<code>\d(2)\s\d(2)\s\d(4)</code>	13/01/2021
<code>\d(4)\s\d(2)\s\d(2)</code>	2021/01/13
<code>\d(2)\s\s\d(2)\s\s\d(4)</code>	13 / 01 / 2021
<code>\d(4)\s\s\d(2)\s\s\d(2)</code>	2021 / 01 / 13
<code>\d(2)\s\d(2)\s\d(2)</code>	13/01/21
<code>\d(1)\s\d(2)\s\d(2)</code>	3/01/21
<code>\d(2)\s\d(2)\s\d(1)</code>	21/01/3
<code>\d(2)\-\d(2)\-\d(2)</code>	13-01-21
<code>\d(1)\-\d(2)\-\d(2)</code>	3-01-21
<code>\d(2)\-\d(2)\-\d(1)</code>	21-01-3
<code>\d(2)\.\d(2)\.\d(2)</code>	13.01.21
<code>\d(1)\.\d(2)\.\d(2)</code>	3.01.21
<code>\d(2)\.\d(2)\.\d(1)</code>	21.01.3
<code>\d(2)\s\s\d(2)\s\s\d(2)</code>	13 / 01 / 21

Fig. 6. The example of file for date parsing rules.

Business Acceptance Testing (BAT) showed that the developed system meets the customer's requirements and works as customers imagined at the beginning. The requirements specification specified that the program must process at least 80 percent of the financial documents correctly. About 83 percent of all financial documents used in the test were processed correctly. When evaluating the results of information detection of individual fields, the best results were detected for “date” and “total amount”, 93.21% and 87.81%, respectively (see Fig. 7). The worst performers were “Seller” and “Buyer”, with accuracy of just 32.69% and 33.45%, respectively. Such low accuracy results are because the data of the seller and the buyer are arranged in different places, in different formats (text, information integrated in company logos, headers, etc.), there is no keyword identifying who the buyer is and who the seller is, etc.



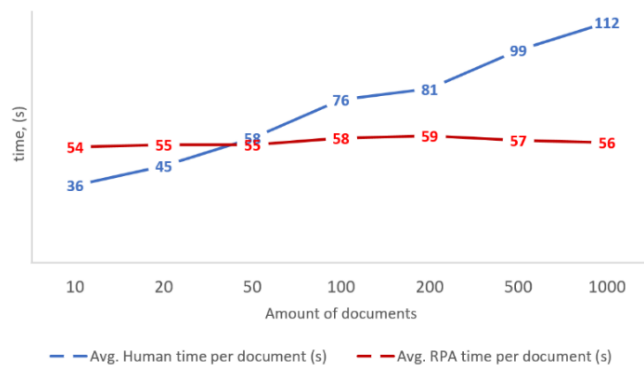
**Fig. 7.** Accuracy results for identification of document information fields.



**Fig. 8.** Dependence of the average processing time of a financial document on the DPI value.

The results of experiments related to the influence of document quality on recognition accuracy are provided in **Fig. 8**. The results show how the processing time of a document varies depending on the DPI value ranging from 75 to 1200. Experiments were also performed for different document quality  $q \in \{1, 25, 50, 100\}$  for DPI values, but this did not affect the results.

To evaluate the processing time of human documents and the performance of the developed robot, an experiment was performed in one company. The results of the experiment are presented in **Fig. 9**. The efficiency of human document processing decreases with increasing volume of processed documents, while the RPA time is almost constant, which works with an average of 56.28 s and a standard deviation of 1.66 when experimenting with a volume of 10 to 1000 documents. As can be seen from the graph, the number of documents does not affect the RPA processing time. Although a person works a little faster with a small number of documents, processing 500 documents using RPA is 1.76 times more efficient and processing 1000 documents using RPA is twice as efficient.



**Fig. 9.** Evaluating document processing efficiency: human document processing time vs. RPA robot processing time.

## 4 Discussion

The product under development is relatively small, but there are two risks involved that may affect the project's timing and budget:

1. Emergence of new requirements on the part of customers. Often, when developing RPA projects, there is a possibility of new requirements, which has an actual impact on the project plan over time. In the case of this project, this probability is relatively low and does not exceed 5%. Knowing this risk, the project plan has already been developed with the risks involved, so the actual emergence of requirements will not affect the project plan and deadlines.
2. Excessive customer expectations for the system being developed. Often, when developing RPA products, customers think that the developed system will fully automate the physical work, but the input of customers during the life of the product is also required. During this project, it is important that, at least during the enhanced maintenance period, customers check the program results file and try to find logical errors so that they are corrected and do not recur in the future. The likelihood of this risk is also low, at only about 5%, as clients will use the results file for daily review and are likely to find any logical errors (if any). The consequence of this risk is a poorly designed system that would be abandoned in the long run, which would undermine the reputation and confidence in the RPA methodology. It is planned that if the process correctly processes at least 80% of all financial documents, the project will be considered a success and expectations will be met.

## 5 Conclusions

In this study, a solution for automated invoice reading was developed incorporating RPA, optical character recognition (OCR) and deep learning technologies. The aim of this study is to investigate the feasibility of advanced technologies to process invoices in completely different formats, with different types of information. The whole process

has been automated and prepared for commercial applications using Blue Prism's RPA software. Different types of invoices were used to train the system, focusing on the detection and identification of five information fields: "Seller", "Buyer", "Invoice No.", "Date" and "Total". However, different structure, the variety of keywords, the abundance of types of information (words, numbers, tables, logo images) make the detection and retrieval of the information very complicated. In the experiments evaluating the results of the detection of information in the individual fields, the best results were "date" and "total", 93.21% and 87.81%, respectively. The worst performers were the Seller and the Buyer, with an accuracy of only 32.69% and 33.45% respectively. Such poor results can be explained by the fact that not all invoices contain the data of the seller or buyer, they are often arranged in different places, in different formats (text, information integrated in company logos, headers, etc.), there is no keyword identifying what is the buyer and who the seller is and so on. Experiments evaluating the processing time of human documents and the performance of the developed RPA based solution have shown that the processing efficiency of human documents decreases with increasing number of processed documents, while the RPA time is almost constant (with average value of 56.28 seconds and standard deviation from 1.66). Robotized process is 1.76 times more efficient than human at 500 documents and 2 times more efficient at 1,000 documents.

As the experiments have shown, more detailed research and task-specific algorithms are needed to identify as much targeted invoice information as possible, including contextual information analysis, synonyms between words, and of course big and representative database.

**Acknowledgments:** We thank the company "Helso" UAB for providing a set of representative invoices and for contributing to the entire automation process and valuable insights and support. We also thank dr. Agne Paulauskaite-Taraseviciene for support, useful suggestions and mentoring process.

## References

1. Capizzi, G., Lo Sciuto, G., Napoli, C., Shikler, R., & Woźniak, M.: Optimizing the organic solar cell manufacturing process by means of AFM measurements and neural networks. *Energies*, 11(5), 1221 (2018).
2. Połap, D., Woźniak, M., Napoli, C., & Tramontana, E.: Real-time cloud-based game management system via cuckoo search algorithm. *International Journal of Electronics and Telecommunications*, 61, 333-338 (2015).
3. Kokina, J., & Blanchette, S.: Early evidence of digital labor in accounting: Innovation with Robotic Process Automation. *International Journal of Accounting Information Systems*, 35, 100431 (2019).
4. Vedder, R., Welters, M., Mints, A., Ramkhelawan, A.: *Robotic process automation in the finance function of the future* (2016).
5. IBSintelligence, <https://ibsintelligence.com/whitepaper/ai-rpa-in-banking-operations>, last accessed 2022/04/01.
6. Lamberton, C., Brigo, D., & Hoy, D.: Impact of Robotics, RPA and AI on the insurance industry: challenges and opportunities. *Journal of Financial Perspectives*, 4(1) (2017).

7. Huang, F., & Vasarhelyi, M. A.: Applying robotic process automation (RPA) in auditing: A framework. *International Journal of Accounting Information Systems*, 35, 100433 (2019).
8. RPA's Impact on the Travel Industry, <https://www.uipath.com/blog/industry-solutions/rpas-impact-on-the-travel-industry>, last accessed 2022/04/01
9. Karabegović, I., Karabegović, E., Mahmić, M., & Husak, E.: The application of service robots for logistics in manufacturing processes. *Advances in Production Engineering & Management*, 10(4) (2015).
10. Sahu, S., Salwekar, S., Pandit, A., & Patil, M. Invoice Processing Using Robotic Process Automation. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, p. 216–223 (2020).
11. Polak, P., Nelischer, C., Guo, H., & Robertson, D. C.: "Intelligent" finance and treasury management: what we can expect. *AI & SOCIETY*, 35(3), 715-726 (2020).
12. Ling, X., Gao, M., & Wang, D. Intelligent document processing based on RPA and machine learning. In *2020 Chinese Automation Congress (CAC)* (pp. 1349-1353). IEEE (2020).
13. Herm, L. V., Janiesch, C., Reijers, H. A., & Seubert, F.: From symbolic RPA to intelligent RPA: challenges for developing and operating intelligent software robots. In *International Conference on Business Process Management* (pp. 289-305). Springer, Cham (2021).
14. Memon, J., Sami, M., Khan, R. A., & Uddin, M.: Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). *IEEE Access*, 8, 142642-142668 (2020).
15. Nagy, G., Nartker, T. A., & Rice, S. V. Optical character recognition: An illustrated guide to the frontier. In *Document recognition and retrieval VII* (Vol. 3967, pp. 58-69). SPIE (1999).
16. Impedovo, S. (Ed.): *Fundamentals in handwriting recognition* (Vol. 124). Springer Science & Business Media (2012).
17. Plötz, T., & Fink, G. A.: Markov models for offline handwriting recognition: a survey. *International Journal on Document Analysis and Recognition (IJ DAR)*, 12(4), 269-298 (2009).
18. Tesseract documentation, Tesseract OCR, Github, <https://tesseract-ocr.github.io/>, last accessed 2022/04/01.
19. Robby, G. A., Tandra, A., Susanto, I., Harefa, J., & Chowanda, A.: Implementation of optical character recognition using tesseract with the javanese script target in android application. *Procedia Computer Science*, 157, 499-505 (2019).
20. Conexiom, Ecmart Inc., A platform for trade document automation. <https://conexiom.com/>, last accessed 2022/04/01.
21. Klippa, Klippa App B.V. - KVK. Automated document processing solutions. <https://www.klippa.com/>, last accessed 2022/04/01.
22. Logikia, Your solution for a smart, paperless office, [www.logikia.ca/](http://www.logikia.ca/), last accessed 2022/04/01.
23. Fast Four, <https://www.fastfour.com/en/>, last accessed 2022/04/01.
24. Nanonets, NanoNet Technologies Inc., <https://nanonets.com/>, last accessed 2022/04/01.
25. Shelhamer, E., Long, J., & Darrell, T.: Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4), 640-651 (2016).
26. Girshick, R., Donahue, J., Darrell, T., & Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587) (2014).
27. Gong, L., Thota, M., Yu, M., Duan, W., Swainson, M., Ye, X., & Kollias, S.: A novel unified deep neural networks methodology for use by date recognition in retail food package image. *Signal, Image and Video Processing*, 15(3), 449-457 (2021).

28. Fu, X., Ch'ng, E., Aickelin, U., & See, S.: CRNN: a joint neural network for redundancy detection. In 2017 IEEE international conference on smart computing (SMARTCOMP) (pp. 1-8). IEEE (2017)
29. Brownlee, J. A Gentle Introduction to Long Short-Term Memory Networks by the Experts. Machine Learning Mastery, 19 (2017).
30. Optical Character Recognition Pipeline: Text Recognition, <https://theailearner.com/tag/attention-ocr/>, last accessed 2022/04/01.
31. Tensorflow. An end-to-end open source machine learning platform, <https://www.tensorflow.org/>, last accessed 2022/04/01.
32. Blue Prism: Robotic Process Automation RPA, Blue Prism Limited, <https://www.blueprism.com/>, last accessed 2022/04/01.
33. UiPath: Robotic Process Automation RPA, UiPath, <https://www.uipath.com/>, last accessed 2022/04/01.
34. Automation Anywhere: Enterprise RPA Platform, Automation Anywhere, Inc., <https://www.automationanywhere.com/>, last accessed 2022/04/01.