

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Diana Vaher 202031IADB

Panga ja sularahavedaja vahelise sõnumivahetuse arendus

Bakalaureusetöö

Juhendaja: Kaido Kikkas
PhD

Kaasjuhendaja: Ants Hindpere
BSc

Tallinn 2023

Autorideklaratsioon

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

Autor: Diana Vaher

13.05.2023

Annotatsioon

Käesoleva töö eesmärgiks on lahendada pangapoolne osa sularahavedaja ja panga vahelisest infovahetusest. Sularahavedaja infosüsteemis on vastav funktsionaalsus juba välja arendatud.

Töös analüüsitakse kirjandust, nõudeid ja plaanitava arenduse töövoogusid ning juba olemasolevat pangarakenduse funktsionaalsust. Analüüsi põhjal pakutakse lahenduseks rakendusliides, mis võtab vastu XML laiendatavas märgistuskeeles koostatud sõnumeid. Kliendi poolt kinnitatud lahendus võetakse arendusse ning töö tulemusena luuakse rakendusliides, mis võtab vastu sularahavedaja poolt saadetud sõnumeid ning saadab vastu kättesaamiskinnitusi. Vastuvõetavate sõnumite info kontrollitakse, töödeldakse ning salvestatakse panga andmebaasi. Arendus toimub agiilse meetodi järgi, pidevas koostöös panga esindajatega.

Lisaks rakendusliidesele luuakse käesoleva töö käigus kasutajaliidese vormid, mis võimaldavad panga töötajatel vajadusel ka läbi kasutajaliidese andmeid sisestada ning töödelda.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 31 leheküljel, 6 peatükki, 13 joonist, 3 tabelit.

Abstract

Development of the Messaging Functionality Between the Bank and the Cash Courier Service

The aim of this thesis is to enhance the functionality of the banking software by enabling the automated processing of messages sent by a cash courier service provider. The cash courier service provider already has a solution in use for exchanging information with banks, so the current thesis needs to solve the functionality for the bank's side.

The thesis analyzes literature, software requirements, planned workflows for messaging and the existing functionality of the banking software. As a result of the analysis, the optimal solution is to use the application programming interface (API) on the bank's side, that would receive and validate XML messages sent by the cash courier service, parse the data from the message and process it. After processing the confirmation is sent back to the courier.

In addition to the API, new forms for the user interface are added to the banking application user interface. These forms visualize the data of the received messages and enable the processing of the cash collection and delivery information manually, in case of such need.

Development is done in an agile way, using Scrum development framework. The development team is in communication with the bank via the product owner through all phases of the development.

The thesis is in Estonian and contains 31 pages of text, 6 chapters, 13 figures, 3 tables.

Lühendite ja mõistete sõnastik

Agiilne meetod	Tarkvara arenduses kasutatav projektijuhtimise liik, rõhuga võimel kohanduda muutustega ning neile reageerida [1]
Aktiva	Bilansi vasakul poolel kajastatavad ettevõtte varad
API	<i>Application Programming Interface</i> , rakendusliides [2]
BSD litsents	Vabavaraliste litsentside perekond, lühend tuleneb nimetusest <i>Berkeley Software Distribution</i>
Deebet	Aktivakontol jääki suurendav ja passivakontol jääki vähendav muudatus [3]
DSM	<i>Dependency Structure Matrix</i> , meetod, millega projektides sõltuvusi kaardistatakse
DTD	<i>Document Type Definition</i> , märgistuskeeles kasutatav dokumendi tüübi määrang korrekse struktuuri tuvastamiseks
FTP	<i>File Transfer Protocol</i> , klient-server arhitektuuriga failiedastusprotokoll
FTPS	<i>FTP Secure</i> , transpordikihi turbeprotokoll [4] kasutatav FTP laiendus [5]
FURPS	Nõuete klassifikatsioonitehnika, mis loob viis nõuete klassi: funktsionaalsus (<i>functionality</i>), kasutatavus (<i>usability</i>), usaldatavus (<i>reliability</i>), jõudlus (<i>performance</i>), toetatavus (<i>supportability</i>)
HTTP	<i>Hypertext Transfer Protocol</i> , hüperteksti edastusprotokoll, klient-server arhitektuuriga andmevahetusprotokoll, mida kasutatakse Internetis dokumentide vahetamiseks [6]
HTTPS	<i>Hypertext Transfer Protocol Secure</i> , hüperteksti edastusprotokoll üle turvasoklite kihi [6]
Kreedid	Aktivakontol jääki vähendav ja passivakontol jääki suurendav muudatus [3]
MoSCoW	Nõuete prioritseerimise tehnika, mis loob neli prioriteediklassi: peab (<i>must</i>), peaks (<i>should</i>), võiks (<i>could</i>) ja ei tee (<i>won't</i>)

Passiva	Bilansi paremal poolel kajastatavad ettevõtte kohustused ja omakapital
PL/SQL	Struktureeritud päringukeele (SQL) protseduuriline laiendus
POST	HTTP meetod andmete muutmiseks või uuendamiseks serveris
REST	<i>Representational State Transfer</i> , tarkvara arhitektuuri laad
Riskifilter	Spetsiifiline tehingute kontroll pangasüsteemis, kus on defineeritud riskantsete tehingute parameetrid (näiteks summa, millest suuremad tehingud on vaja käsitsi üle kontrollida)
Scrum	Agiilne projektijuhtimise raamistik interaktiivsete ja inkrementaalsete projektide haldamiseks [7]
SFTP	Turvaline failiedastusprotokoll, võimaldab failiedastust ja failide haldamist eemal asuvates arvutites [6]
SGML	<i>Standard Generalized Markup Language</i> , üldistatud märgistuskeelte standard
SOAP	<i>Simple Object Access Protocol</i> , lihtne objektipöördusprotokoll, mida kasutatakse struktureeritud andmete edastamiseks [6]
SSH	<i>Secure Shell</i> , turvakest ehk krüptograafilise võrguprotokoll, võimaldab turvalist krüpteeritud kaugpöördumist [5]
Sularaha transiidikonto	Transportimisel olevat sularaha kogust kajastav konto
TLS	<i>Transport Layer Security</i> , transpordikihi turbeprotokoll [5]
UUID	<i>Universally Unique Identifier</i> , 128-bitine objektimärgis arvutisüsteemides [8]
Vabavara	Vaba ning avatud lähtekoodiga tarkvara, mille kasutajal on tarkvara muutmine ning levitamine lubatud [6]
Välisvõti	Relatsioonilise andmebaasi veerg või veergude grupp, mis viitab sama või mõne teise tabeli unikaalsele võtmele, tavaliselt primaarvõtmele [9]
Äriloogiline validatsioon	Panga äriloogikal põhinev andmete kontroll
XML	<i>Extensible markup language</i> , laiendatav märgistuskeel [6]
XSD	<i>XML Schema definition</i> , XML skeemi definitsioon [6]

Sisukord

1 Sissejuhatus	10
1.1 Ülesande püstitus	10
1.2 Lähtetingimused	11
2 Metoodika	13
3 Probleemi analüüs ja lahenduse valik	16
3.1 Sularahaveo korraldus	16
3.2 Ülevaade olemasolevast pangarakendusest	17
3.3 Analüüs ja lahenduse valik	18
3.3.1 Liidestamine	18
3.3.2 Võimalikud probleemid sõnumivahetuses	22
3.3.3 XML teegi valik	23
3.3.4 Andmete valideerimine ja töövoogude ülevaade	25
4 Lahenduse realiseerimine	30
4.1 Muudatused andmebaasis	30
4.2 Muudatused kasutajaliideses	31
4.3 Lahenduse testimine	37
4.4 Hinnang loodud lahendusele	38
5 Tagasiside ja edasised plaanid	40
6 Kokkuvõte	41
Kasutatud kirjandus	42
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	46
Lisa 2 – Ärinõuetel põhinev tegevusplaan	47
Lisa 3 – Muudatusnõuete omavahelised sõltuvused	49
Lisa 4 – Muudatusnõuete omavahelised sõltuvused maatriksina	51

Jooniste loetelu

Joonis 1. SFTP serveri ja kliendi lihtsustatud andmevahetuse skeem aktiivses ja passiivses režiimis (autori joonis allika [38] põhjal).....	20
Joonis 2. Riskifiltri kontrolli lahendus rakendusliidese korral.....	21
Joonis 3. XML failide lugemise kiiruste võrdlus [52].....	24
Joonis 4. Sularaha tarne töövoog.....	26
Joonis 5. Sularaha korje töövoog.....	27
Joonis 6. Sõnumi salvestamine kasutajaliideses.....	32
Joonis 7. Sõnumi kinnitamine kasutajaliideses.....	32
Joonis 8. Sõnumite olekud andmete töötlemisel.....	33
Joonis 9 Kohandatav filter.....	34
Joonis 10. Sõnumite nimekirja vaade.....	34
Joonis 11. Algne olekute komponendi kuvamise kujundus.....	35
Joonis 12. „Kinnitatud“ olekus sõnumi vaade.....	36
Joonis 13. Sularahaveo riskifiltri vaade.....	37

Tabelite loetelu

Tabel 1. Kategooriate prioritseerimine.....	14
Tabel 2. DTD ja XSD võrdlus [53].	24
Tabel 3. Päringute keskmine töötlemiskiirus enne ja pärast indekseerimist.	38

1 Sissejuhatus

Automatiseeritud andmevahetus ja andmete automaatne töötlemine ning salvestamine lihtsustab ja kiirendab ettevõtete vahelist koostööd. Samuti aitab automatiseeritus vältida inimfaktorist tulenevaid vigu andmete sisestamisel.

Käesoleva töö eesmärgiks on lahendada pangapoolne osa sularahavedaja ja panga vahelisest infovahetusest. Plaanitav funktsionaalsus võimaldab pangal klientidele sularahaveo teenuseid pakkuda, kasutades kolmandat osapoolt turvatud veoteenuste jaoks. Selliseid teenuseid kasutavad põhiliselt kaubandusettevõtted, kes soovivad kaubandustegevuses saadud sularaha turvaliselt pankka saata. Samuti tellitakse sularahaveoga kaupluste kassadesse vahetusraha.

Töö käigus luuakse pangapoolne funktsionaalsus sularahavedaja poolt saadetud sõnumite vastuvõtmiseks, info lugemiseks ja töötlemiseks ning andmebaasi salvestamiseks. Samuti peab rakendus saatma töödeldud sõnumite kohta sularahavedajale kättesaamiskinnitused ning probleemide korral spetsiifilised veateated.

Lisalahendusena luuakse pangarakenduse kasutajaliidesesse vormid, mis võimaldavad kasutajal sularaha tehingute infot sisestada ning töötlemisse saata.

1.1 Ülesande püstitus

Käesoleva töö käigus lahendatakse panga ja sularahavedaja vahelise infovahetuse pangapoolne osa, mille põhilised funktsionaalsused on järgnevalt loetletud.

- Pangarakendus võtab saabunud sõnumid vastu, kontrollib ja töötleb saadud info ning saadab kättesaamiskinnituse. Vigase sõnumi või tõrke korral saadetakse kättesaamiskinnitus koos veakoodiga.
- Pangarakenduse kasutajaliideses saab sularahahoidlaid seadistada ja panga pearaamatuga siduda. Sularahahoidla vaates kuvatakse kasutajale hoidlas olevat rahasummat.

- Pangarakenduse kasutajaliideses on võimalik klientidele sularahaveo lepinguid lisada ja olemasolevaid lepinguid lõpetada.
- Pangarakenduse kasutajaliideses kuvatakse saabunud sõnumite info ning on võimalik lisada ja kinnitada uusi sõnumeid.

Skoopi kuuluvad sõnumivahetuse töövoog, sularahaveo lepingute ja sularahahoidlate info haldamisvõimaluste lisamine kasutajaliidesesse, sularaha tehingute riskifiltri kontroll, muudatused andmebaasi tabelites ning veakontrolli läbinud sõnumite info põhjal kannete tegemine.

Töö skoobist jääb välja sularaha tehingute info lisamine raportitesse, mis on edaspidi plaanis, kuid töö valmimise hetkeks ei ole arendusülesannete koostamiseks veel piisavalt detailset infot. Samuti ei puudutata sõnumivahetuse turbeküsimusi, sest liidestuse välised turbevalikud ei ole töö autori otsustada.

1.2 Lähtetingimused

Pank on käesoleva töö alguseks sularahaveo koostööpartneri juba välja valinud. Sularahavedaja pakub liidestamiseks välja valiku kolme juba nende poolt kasutusel oleva lahenduse vahel, mis on järgnevad:

1. Airea Messenger sõnumivahetusprogramm [10],
2. FTP serveril põhinev sõnumivahetus,
3. rakendusliides, kuhu sõnumeid saadetakse.

Sularahavedaja kasutab infovahetuseks kindlat XML vormingut. Vastussõnumi struktuur on sama nii eduka kättesaamise ja töötlemise puhul kui ka viga edastava vastuse korral. Vastuses oodatakse saadetud sõnumis sisaldunud identifitseerivat koodi, et oleks tuvastatav, millise tehingu kohta vastus saadeti. Vea korral lisatakse vastusesse ka veakood. Kasutusel olevate veakoodide nimekiri pakutakse välja sularahavedaja poolt. Pank määrab omalt poolt, mida sõnumi saabumisel kontrollida, näiteks konto- ja äriregistri numbrite olemasolu pangarakenduse andmebaasis ning riskifiltri kontroll.

Pangarakenduse senine graafilise kasutajaliidese kujundus annab ette kasutajaliidesesse lisatavate vormide malli ning nõude sobituda ülejäänud kujundusega.

Pangandussektori eripära tõttu on sularahakannetega seotud automatiseeritud tegevuste toimumise aeg tööpäevadega piiratud ning toimingud peavad jääma kindlasse ajavahemikku. Seega peab loodavas arenduses saama seadistada töötlemiseks lubatavate kellaegade vahemikku ning vaid lubatud vahemikus saabunud sõnumeid tohib töödelda. Lubatud ajavahemiku kontrollimisel tuleb jälgida riiklikke pühasid ja puhkepäevi – ka siis ei tohi sõnumeid automaatselt töödelda.

2 Metoodika

Käesoleva töö tegevused planeeritakse järgnevalt:

- nõuete täpsustamine iteratiivsel meetodil [11], [12],
- nõuete kategooriatesse struktureerimine,
- nõuete analüüs ja modelleerimine,
- töövoogude skeemide valideerimine,
- sõnumivahetusviisi valik,
- nõuete prioritseerimine sõltuvuste struktuuri maatriksi meetodi järgi [13],
- arendustööd.

Arenduse kontseptsioon ja skoop on enne töö algust pangaga kokku lepitud, käesolev töö jätkab nõuete väljaselgitamisega iteratiivsel meetodil. Iteratiivse meetodi puhul jagatakse arenduse lõppeesmärk väiksemateks osadeks ehk iteratsioonideks ja vahetult enne iteratsiooniga alustamist täpsustatakse arenduseks vajalikud detailid selle konkreetse iteratsiooni jaoks. Iteratiivne lähenemine valitakse eeldusel, et nõuded täpsustuvad, kui osa funktsionaalsusest on juba valminud ja tellijal tekib oma soovidest detailsem nägemus [14, lk 44]. Iteratiivne meetod ei eelda, et kogu info on alguses teada. [11]

Kuna nõuded on töö alguses arendamiseks liiga ebamäärased ja neid on keeruline täpsemalt sõnastada, tuleb nõuete täpsustamiseks teha lisatööd. Olemasoleva info põhjal luuakse üldistav nimekiri plaanitavatest arendustööde punktidest, mis struktureeritakse nõuetele sarnaselt funktsionaalseteks ning mittefunktsionaalseteks töödeks ning jagatakse kategooriatesse, mis koos omavaheliste sõltuvustega on tabelis 1.

Tabel 1. Kategooriate prioritseerimine.

Nr	Kategooria	Sõltuvad kategooriad
1	Arenduse eeltööd	2, 3, 4, 5
2	Loodava funktsionaalsuse töövood	3, 4, 5
3	Kasutajaliides	4, 5
4	Seaded	5
5	Raportid	

Funktsionaalseteks kategoriseeritakse rakenduse tööd ja vajaminevat infot puudutavad teemad, mittefunktsionaalseteks kujundusega seotud teemad [15, Alajaot. 1–3]. Üldise tegevusplaani koostamine aitab käesoleva töö autoril olemasolevat infot süstematiseerida ja saada sellest parem ülevaade. Struktureeritud tegevusplaani punktid on lisas 2. [15, Ptk 1]

Järgnevalt luuakse planeeritavatest töövoogudest skeemid [16] (joonised 4 – 8), mis valideeritakse ja täpsustatakse koos panga esindajaga. Töövoogude skeemid aitavad panga esindajale paremini pakutavat lahendust selgitada ja pangapoolseid nõudeid täpsustada. Peale töövoogude täpsustusi valitakse sobiv liidestusviis sõnumite vahetamiseks, millest täpsemalt kirjutatakse peatükis 3.3.1.

Kui liidestusviis on valitud, alustatakse muudatusnõuete koostamisega. Muudatusnõuete sõnastamisel peetakse oluliseks, et iga arendusse võetav muudatusnõue oleks piisavalt eraldiseisev, et võimaldada selle täitmise kontrollimist ja testimist [15, Alajaot. 1–2]. Üks muudatusnõue koosneb ühest kuni viiest tegevusplaani punktist. Muudatusnõuete prioritseerimiseks otsustatakse valida meetod, kus välised kasutajad on vähem kaasatud. Valik tuleneb nõudest jälgida panga kasutajaliidese muudatustes olemasolevat kujundust ja rakenduse üldist loogikat, samuti panga kui tellija eripärast, kes pigem ootab arenduspartneripoolset ärianalüüsi. Seetõttu välistatakse võimaluste hindamise meetod (*Opportunity Scoring*) [17], Kano mudel [18] ja kliendikeskne tootekavandamine (*Quality Function Development*) [19]. Aeganõudvamad meetodid nagu AHP (*Analytical Hierarchy Process*) [20] ei ole käesoleva suurusega skoobi puhul põhjendatud. Välja jäetakse ka nominaalset skaalat kasutavad meetodid, mis grupeerivad nõuded, kuid grupisisiselt prioriteete ei erista (MoSCoW, FURPS) [21, lk 2].

Erinevate äriliste nõuete funktsionaalsused on paljudel juhtudel omavahel seotud (sõltuvuses). Kuna nõuete omavaheliste sõltuvuste ignoreerimine muudab arenduse planeerimise ja tähtaegade määramise keerulisemaks [22], otsustatakse kasutada DSM (*Dependency Structure Matrix*) meetodit [13], [23], mis on abiks sõltuvuste süstematiseerimisel ning vajalike sisendite ja väljundite kaardistamisel. Muudatusnõuete omavahelised sõltuvused kaardistatakse ja leitud sõltuvuste põhjal koostatakse maatriks, mis on parema ülevaate andmiseks toodud kahel erineval viisil lisades 3 ja 4.

Nõuete prioritseerimise teeb raskemaks asjaolu, et kõik nõuded ei ole arenduse alguses teada või selgub arenduse käigus, et mõne nõude osas on oluline info puudu. Seega võib öelda, et järgmises prioritseerimise etapis määrab arenduseks vajamineva info olemasolu nõuete arendusse võtmise järjekorra.

Nõuete järjekord lisas 3 kajastab tinglikult arendustööde järjekorda, sest nõudeid lisandub töö käigus jooksvalt ning juba alustatud nõuded jäävad kohati pikemalt ootele, et täpsustuksid kõik vajaminevad detailid. Seega ei ole ühtset viisi, kuidas dokumenteerida nõuete lõplikku järjekorda. Lisas 3 olevatesse tabelitesse koondatakse kõik nõuded, sealhulgas arenduse jooksul lisandunud nõuded, mis on märgitud tärniga.

Muudatusnõuete detailset sisu käesolevas töös välja ei tooda, kuid üldpildi selgitamiseks antakse nõuetele sisu kirjeldavad pealkirjad. Muudatusnõuded sisaldavad lisaks põhifunktsionaalsusele üksuste teste ning kasutajaliidese teste, samuti muudatusi ja lisasid dokumentatsioonis ning pangarakenduse „abi“ lehekülgedel.

Peale prioritseerimist alustatakse arendusega. Arendus viiakse läbi agiilselt, järgides Scrum projektijuhtimise raamistiku põhimõtteid [7]. Arendustööde teostamiseks kasutatakse kahenädalasi sprinte (kokkuleppeline arendustööde planeerimises kasutatav ühik) [24].

3 Probleemi analüüs ja lahenduse valik

Loodava sõnumivahetuse variantide valik on piiratud sularahavedaja poolt pakutud võimalustega, lisaks sellele otsusele analüüsitakse töös ka teisi loodava funktsionaalsuse aspekte, sealhulgas kasutajaliidese kujundust. Järgnevates alampeatükkides on ülevaade sularahaveo korraldusest, loodavatest töövoogudest ning olemasolevast pangarakendusest. Ülevaatele järgneb võimalike lahendusvariante analüüs ja lahenduse valik.

3.1 Sularahaveo korraldus

Turvatud sularahavedu pakuvate ettevõtete teenuseid kasutavad näiteks sularahaga kauplevad kaubandusettevõtteid, kuna sularaha pankade transportimine vajab kõrgendatud turvameetmeid ning ohutuse ja mugavuse tõttu on mõistlik kasutada professionaalset teenust. Sama kehtib pangast vahetusraha tellimisel.

Sel põhjusel eelistavad sularahaga kauplevad ettevõtted koostööpartneritena pankasid, mis pakuvad sularahaveo teenuseid. Sularahaveo teenuste pakumine on pankade jaoks võimalus suurendada konkurentsivõimet ja parandada positsiooni turul. Pankadel omakorda on sularahavedajaga mugav töötada, kuna pank ei pea ise sularahaga tegelema, piisab vaid informatsioonist sularaha liikumiste kohta. Plaanitav sularahaveo teenus koosneb kahest vastassuunalisest töövoost: sularaha korjest ning sularaha tarnest.

Sularaha korje

- Klient tellib sularahavedajalt sularaha korje.
- Sularahavedaja teostab korje (toob sularaha kliendi juurest ära).
- Sularaha kontrollitakse ja loetakse sularahavedaja poolt üle.
- Sularahavedaja saadab korje infoga sõnumi pangale.
- Pank võtab sõnumi vastu ja valideerib sõnumi sisu.
- Kui sõnum ei valideeru või korjele on piiranguid, saadab pangarakendus vastava veateate.

- Eduka valideerimise järel tehakse pangasüsteemis vastavad kanded.
- Pank saadab kannete tegemise kohta sularahavedajale kinnituse.

Sularaha tarne

- Klient tellib sularahavedajalt sularaha tarne.
- Sularahavedaja edastab pangale info plaanitava veo kohta.
- Pank võtab sõnumi vastu ja valideerib sõnumi sisu.
- Kui sõnum ei valideeru, kontrol pole piisavalt vahendeid või tarnele on piiranguid, saadab pangarakendus vastava veateate.
- Eduka valideerimise järel tehakse pangasüsteemis vastavad kanded.
- Pank saadab kannete tegemise kohta sularahavedajale kinnituse.
- Sularahavedaja toimetab sularaha kliendi juurde.

3.2 Ülevaade olemasolevast pangarakendusest

Pangarakendus on kirjutatud Pythoni ja Javascripti programmeerimiskeeltes, Pythoni raamistikest on kasutusel Bottle [25]. XML failide töötlemiseks on kasutusel vabavaraline BSD litsentsiga [26], [27] lxml teek. Lxml aitab lihtsustada kohandatud protsesse XML vormingus failide lugemisel, sidudes C programmeerimiskeelt kasutavad teegid libxml2 [28] ning libxslt [29] Pythoni programmeerimiskeelega. Lxml toetab DTD (*Document Type Definition*) ja XSD (*XML Schema Definition*) baasil XML failide valideerimist. [30]

XML vormingus andmevahetus väliste infosüsteemidega on pangarakenduses enne käesoleva töö algust lahendatud põhiliselt rakendusliideste kaudu, mis järgivad erineval määral REST arhitektuuri. REST arhitektuuri iseloomustavad kuus printsiipi, millest tuntumad on klient-server arhitektuur ning olekuta loogika, mis tähendab, et iga päring peab sisaldama kogu infot vastuse saamiseks. [31]

Ühel juhul on kasutusel SOAP standard [32]. SOAP eeliseks on WS-Security turvalaiendus [33], mis turvab sõnumi terviklust, konfidentsiaalsust ning sõnumi kaupa autentimist. Turvamiseks kasutatakse XML digiallkirjastamist ja krüpteerimist ning sõnumitele lisatud turvamärke. WS-Security toimimine on sõltumatu kasutatavast transpordi protokollist. SOAP standardi järgi koostatud sõnumites on lisaks saadetavatele

andmetele ka päis ja kogu sõnum on omakorda SOAP formaadi spetsiifilises ümbrikus [34].

3.3 Analüüs ja lahenduse valik

Planeeritav funktsionaalsus koosneb kahest töövoost, osa töövoogude toimingutest toimub sularahavedaja poolel ning osa panga poolel. Tänu korje ja tarne töövoogude sarnasusele ning infot edastavate sõnumite samale vormingule, saab andmete töötlemise panga poolel paljudes punktides ühendada. Töövoogude erinevus seisneb automaattöötlemise ajalises nihkes raha transpordi ja teavitussõnumi saatmise vahel, lisakontrollides sularaha tarne puhul ning riskifiltri märkega sõnumite käsitlemises. Käesoleva töö raames arendatakse töövoogude pangapoolne osa.

3.3.1 Liidestamine

Liidestusviis tuleb sularahavedajaga kokku leppida, sularahavedajal on vastavad lahendused juba välja töötatud ja nad ei soovi uusi lahendusi juurde lisada. Pangal on võimalik valida sularahavedaja poolt juba kasutusel olevate lahendusvariantide vahel.

Sularahavedaja pakub sõnumivahetuseks välja järgnevad võimalused:

1. Aurea Messenger sõnumivahetusprogramm [10],
2. FTP serveril põhinev sõnumivahetus,
3. rakendusliides.

Esimene valik on spetsiaalne sõnumivahetuseks mõeldud programm, mis oleks liidestamiseks kõige mugavam, kuid kuna programm on tasuline, siis ei soovi pank seda kasutada, et mitte tõsta halduskulusid.

Teine pakutud lahendusvariant on sõnumite vahetus FTP failiedastusprotokolli kaudu, kasutades sularahavedaja poolt hallatavat serverit. FTP protokoll ei ole krüpteerimata andmevahetuse tõttu turvaline, samuti võimaldab protokoll jõuründeid (*brute force attack*), portide ülevõtmist ning paroolide varastamist [35]. Seega tuleb FTP serveri variandi valimisel turvalisuse tõstmiseks kasutada lisameetmeid [36]. Arvestades vajadust mingitel juhtudel viivitusega vastata ning sama kande kohta korduvalt infot

vahetada, sobiks FTP serveri kasutamine plaanitava töövoos jaoks paremini kui rakendusliides.

Kolmas variant on pangapoolne rakendusliides, mis võtab vastu sularahavedaja poolt saadetud sõnumeid ning vastab kättesaamiskinnitusustega. Rakendusliidese puhul tuleb kasutada HTTPS andmete edastusprotokolli, mis krüpteerib edastatava info [37]. Rakendusliidese põhiliseks puuduseks on võimetus teatud juhtudel vastusega oodata. Samuti ei saa päring – vastus süsteemis saata rohkem aega võtva oleku muudatuse kohta uuendatud infot, sest suhtluse algatab päringu saatja.

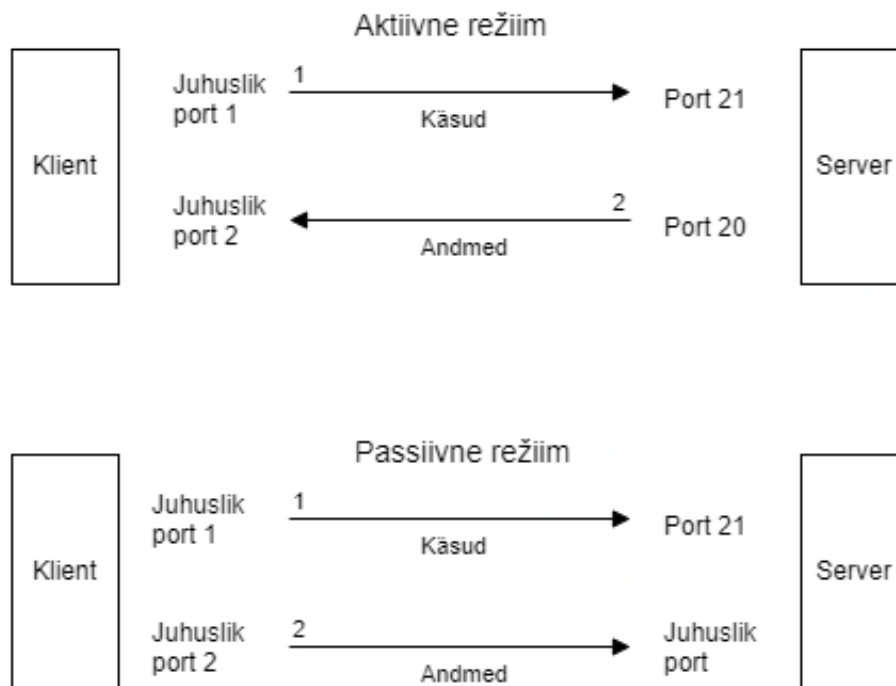
Järgnevalt antakse ülevaade kahest erinevast failivahetusprotokolli kasutavast serveri variandist, mida lahendustena kaalutakse.

FTPS (transpordikihi turbeprotokoll) on TLS krüpteerimisprotokolli kasutav FTP laiendus [4], [38]. TLS kasutamine lisab andmevahetusele krüpteeringu, tervikluse kontrolli ning autentimise andmeid vahetavate poolte tuvastamiseks [39]. FTPS laiendab FTP-d ja on kasutatav ka juhul, kui protokolli vahetamine SFTP kasuks ei ole mingil põhjusel võimalik.

FTPS protokoll saab kasutada aktiivses ja passiivses režiimis. Režiimide lihtsustatud skeem on joonisel 1 [40], kus on noolega näidatud esmase ühenduse suund. Aktiivses režiimis algatab klient käskude vahetuseks ühenduse serveri 21. pordiga, server ühendub omakorda läbi porti 20 tagasi kliendi andmevahetuseks avatud porti. Aktiivset režiimi üldiselt ei kasutata, kuna kliendipoolse juhuslikult valitud kuulava porti kaudu infovahetuse lubamiseks tuleks tulemüüris lubada sisenev liiklus laia pordivahemiku jaoks.

Passiivses režiimis ühendub klient juhuslikult pordist serveri porti 21 ning server annab kliendile teada juhusliku porti numbri, mille server on andmevahetuseks avanud. [41]

Passiivse režiimi korral valib server juhusliku porti üsna suure pordivahemiku seast ning annab kliendile teavituse sinna ühenduda. [42]



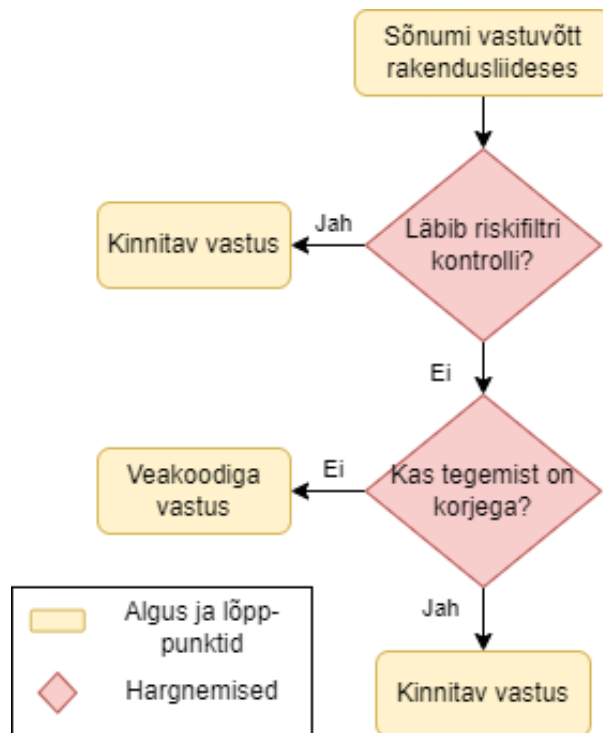
Joonis 1. SFTP serveri ja kliendi lihtsustatud andmevahetuse skeem aktiivses ja passiivses režiimis (autori joonis allika [38] põhjal).

Passiivse seadistuse puhul on turvalisuse probleem serveripoolses tulemüüri seadistuses, kuigi serveril on loogilisem omada suuremat avatud portide vahemikku kui kliendil [41]. Seetõttu soovitatakse kirjanduses üldiselt FTPS asemel eelistada järgnevalt kirjeldatud SFTP lahendust. [43], [44]

SFTP on turbeprotokoll, mis krüpteerib edastatud failid ning sisselogimisandmed SSH protokolliga abil. Kui FTPS korral on andmete vahetuseks vaja kahte porti, millest üks on juhuslikult valitud avalik port [41], siis SFTP kasutab andmete edastamiseks ühte mittejuhuslikku porti. Mittejuhusliku porti kasutamine teeb ka tulemüüri seadistamise lihtsamaks [45]. [46]

Algselt otsustatakse kasutada SFTP põhise sõnumivahetust, sest see võimaldab mõlemasuunalist sõnumite saatmist. Sõnumite vahetuse protsess on paindlikum, kuna ei ole panga poolt piiratud ainult sõnumi lugemise ning kinnituste saatmisega nagu rakendusliidese puhul. Paindlikkus on vajalik riskifiltri märkega sõnumite puhul, sest pangatöötaja peab riskifiltri märkega sõnumi kinnitama või tühistama ja sellist viivitust vastuse saatmisel rakendusliides ei võimalda.

Nõudeid täpsustades selgub võimalus sõnumivahetuse loogika rakendusliidese jaoks sobivaks muuta, tingimusel et riskifiltri tabamuse korral on kohati vajalik panga- ja sularahavedaja töötajate omavaheline suhtlus. Kui riskifiltri märke korral saadetakse sularaha korje puhul kinnitav vastus ja sularaha tarne puhul veakoodiga vastus, oleks võimalik lahendusena kasutada ka rakendusliidest. Joonisel 2 on lihtsustatud töövoog rakendusliidese kaudu saabunud sõnumite riskifiltri kontrollist.



Joonis 2. Riskifiltri kontrolli lahendus rakendusliidese korral.

Kui automaatselt tagasilükatud sularaha tarne sõnumit on hoolimata aktiivsest riskifiltrist siiski vaja töödelda, sisestatakse ning kinnitatakse sõnumi info pangatöötaja poolt läbi rakendusliidese. Sularaha korje puhul on panga ja sularahavedaja töötajate vaheline suhtlus vajalik vaid siis, kui riskifiltri märke saanud sõnumit tühistada. Sel juhul tühistab pangatöötaja riskifiltri märkega sularaha korje sõnumi läbi kasutajaliidese ning edasine tegevus korjega saabunud sularaha osas lepatakse sularahavedajaga kokku. Muudel juhtudel piisab päring-vastus infovahetusest, kuna kogu vajaminev info on iga kord sõnumis ja vastuses olemas. Sularahavedaja ning panga esindajate hinnangul peaks riskifiltri märkega sõnumeid saabuma väga harva, seega ei pea pank vajalikuks riskifiltri tabamusega sõnumite täielikku käsitlemist rakendusliidese funktsionaalsusega katta.

Sõnumivahetusmeetodi muudatus rakendusliidese kasuks ei tekita olulist ajakadu, kuna arendus on muudatuse tegemise ajal üsna algusjärgus ja vaid väike osa loodud koodi tuleb ära kustutada. Ajasäästu võib põhjustada ka rakendusliidese lihtsam arenduse loogika, mistõttu detailide täpsustamine panga ning sularahavedajaga peaks võtma vähem aega.

3.3.2 Võimalikud probleemid sõnumivahetuses

Rakendusliides võtab vastu POST päringuga [47] saadetud sõnumeid, valideerib sõnumite sisu, teeb vajalikud muudatused ja kanded ning salvestab saadud info panga andmebaasi. Sõnumite automaatsel töötlemisel on oht, et kui saadetakse mitu korda sama sõnum, siis topelt saadetud sõnum ka töödeldakse korduvalt. Probleemi vältimiseks otsustatakse sõnumite vastuvõtul jälgida sularahavedajapoolset unikaalset identifikaatorit, juba esinenud identifikaatoriga sõnum lükatakse veakoodiga tagasi ning sõnumi sisu ei salvestata.

Unikaalset identifikaatorit omavad, kuid mittevalideeruvad sõnumid salvestatakse andmebaasi ilma rahalisi kandeid tegemata, päringu vastusele lisatakse sel juhul ka vastav veakood. Päringu saatja saab vastussõnumis veakoodi järgi viite probleemile, korrigeerib vigase koha ja saadab uue unikaalse identifikaatoriga märgistatud ning parandatud sõnumi. Iga kande kohta on olemas oma unikaalne kande number, mida lubatakse vigasena korduvalt vastu võtta, kuid kui sõnum läbib kontrollid ja töötamise edukalt, unikaalset numbrit rohkem korrata ei tohi.

Pank defineerib sularaha tehingute jaoks lubatud ajavahemikud, ning rakendusliidese kaudu saabunud sõnumitel põhinevaid sularahakandeid saab teha vaid selles ajaaknas. Siit tekib küsimus, kuidas lahendada väljaspool lubatavat ajavahemikku saadetud sõnumite töötlemine. Lihtne oleks sellised sõnumid tagasi lükata, kuid kui sõnum on muus osas korrektne, tekitaks see teisele osapoolle põhjendamatult lisatööd. Keerukuselt järgmine variant oleks saabunud sõnumid hoida saabumise järjekorras rakenduse mälus, mis ei ole hea lahendus, sest näiteks elektrikatkestuse korral oleksid selliselt hoitud andmed kustunud. Kindlam variant on salvestada lubatud ajavahemiku välisel ajal vastuvõetud sõnumid tabelisse ja lisada olekut märkiv väli, mille järgi rakendus töötlemata sõnumid üles leiab. Peale edukat töötlemist uuendab rakendus sõnumi oleku.

Sularahavedaja on valmis panga poolt sõnumite vastuvõtuks lubatavate ajavahemike järgimiseks lisama oma rakendusse järjekorral põhineva funktsionaalsuse, kus saab

pankadele lubatud ajavahemikud seadistada ning väljaspool lubatud ajavahemikku loodud sõnumeid lubatud ajavahemiku saabumiseni järjekorras hoida. Käesolevas töös piirdatakse pangale planeeritavas funktsionaalsuses vaid lubatud ajavahemiku väliselt saabunud sõnumite tagasilükkamisega, kuna eeldatakse, et sularahavedaja lahendab väljaspool lubatud aegu loodud sõnumite saatmise probleemi ning vaid erijuhtudel saadetakse mõni sõnum väljaspool kokkulepitud ajaakent.

Juba arenduse analüüsi etapil tekib vajadus spetsiifilisemate veakoodide järgi ja vastav ettepanek saadetakse sularahavedaja arendajatele. Varuvariandina analüüsitakse juhtu, kui sularahavedaja ei nõustu uusi veakoode lisama. Sel puhul saaks uusi veakoode kasutada pangasiseselt, et tagada täpsem vea defineerimine, kuid välja saadetakse veakood läbiks tõlkefaasi, kus spetsiifilisem veakood tõlgitakse üldisemaks, sularahavedaja poolt defineeritud veakoodiks. Samas jääks panga andmebaasi detailsem info juhtunud vea kohta, mida panga töötajatele kuvada. Veakoodide tõlkimist ei realiseerita, sest sularahavedaja annab oma nõusoleku uute veakoodide lisamiseks.

3.3.3 XML teegi valik

XML failide lugemiseks tuleb valida teek. Seni on panga infosüsteemis olnud kasutused lxml teek, mis võimaldab DTD ja XSD valideerimist. Kaalutakse ka xmlschema teeki [48], mis võimaldab samuti XSD valideerimist ning XML andmete lugemist ja on kasutatav MIT litsentsi alusel [49]. Xmlschema vajab töötamiseks elementpath pakki [50] ning pakub samu funktsionaalsusi mis lxml, seega on mõistlik valida juba kasutusel olev teek ja mitte lisada panga infosüsteemi uut sõltuvust.

Kolmas uuritud võimalus XML failide lugemiseks on Pythoni teek minidom, minimeeritud variant dokumendi objektimudelist, mõeldud kui väike ja lihtne lahendus sõnede ja XML failide lugemiseks. Minidom ei sisalda kaitset mitteusaldusväärsete andmete lugemisel, mis võivad sisaldada pahavara. [51]

Minidom ei sobi XSD valideerimisvõimaluse puudumise ja turvalisuse tõttu, ning jääb lxml-ile alla ka failide töötlemise kiiruse poolest. Näitena on joonisel 3 jõudlusandmed lxml kodulehelt [52], kus võrreldakse 274 KB suuruse XML faili töötlemist, mis sisaldab Shakespeare'i „Hamleti“ teksti. Minidom kiirus on 0.066 sekundit, samas lxml saab hakkama 0.004 sekundiga.

```

xml.etree.ElementTree.parse done in 0.006 seconds
xml.etree.cElementTree.parse done in 0.007 seconds
xml.etree.cElementTree.XMLParser.feed(): 6636 nodes read in
0.006 seconds
lxml.etree.parse done in 0.004 seconds
drop_whitespace.parse done in 0.004 seconds
lxml.etree.XMLParser.feed(): 6636 nodes read in 0.004 seconds
minidom tree read in 0.066 seconds

```

Joonis 3. XML failide lugemise kiiruste võrdlus [52].

Kuna lxml võimaldab DTD ja XSD järgi valideerimist, tuleb järgnevalt otsustada kumba neist kasutada. DTD variant pole nii range kui XSD ja omab XSD-ga võrreldes mõningaid piiranguid. Tabelis 2 on nende omavaheline võrdlus [53].

Tabel 2. DTD ja XSD võrdlus [53].

Nr	DTD	XSD
1	DTD tähendab dokumendi tüübi definitsiooni.	XSD tähendab XML skeemi definitsiooni.
2	DTD on tuletatud üldistatud märgistuskeelte (SGML) süntaksist.	XSD on kirjutatud XML formaadis.
3	DTD ei toeta andmetüüpe.	XSD toetab elementide ja nende omaduste andmetüüpe.
4	DTD ei toeta nimeruumi.	XSD toetab nimeruumi.
5	DTD ei määra alaelementide järjekorda.	XSD määrab alaelementide järjekorra.
6	DTD ei ole laiendatav.	XSD on laiendatav.
7	DTD ei ole kergesti õpitav.	XSD on kergesti õpitav, kuna see ei ole omaette keel.
8	DTD kontrollib vähem XML struktuuri.	XSD kontrollib täpsemalt XML struktuuri.

Tabelis toodud aspektid on mõneti tinglikud, näiteks DTD-le saab erinevate tehnikatega tekitada laiendatavust ning elementide järjekorra kontrolli. Seega oleks käesolevas töös võimalik kasutada mõlemat varianti, eeldusel et laiendatavus ja järjekord on

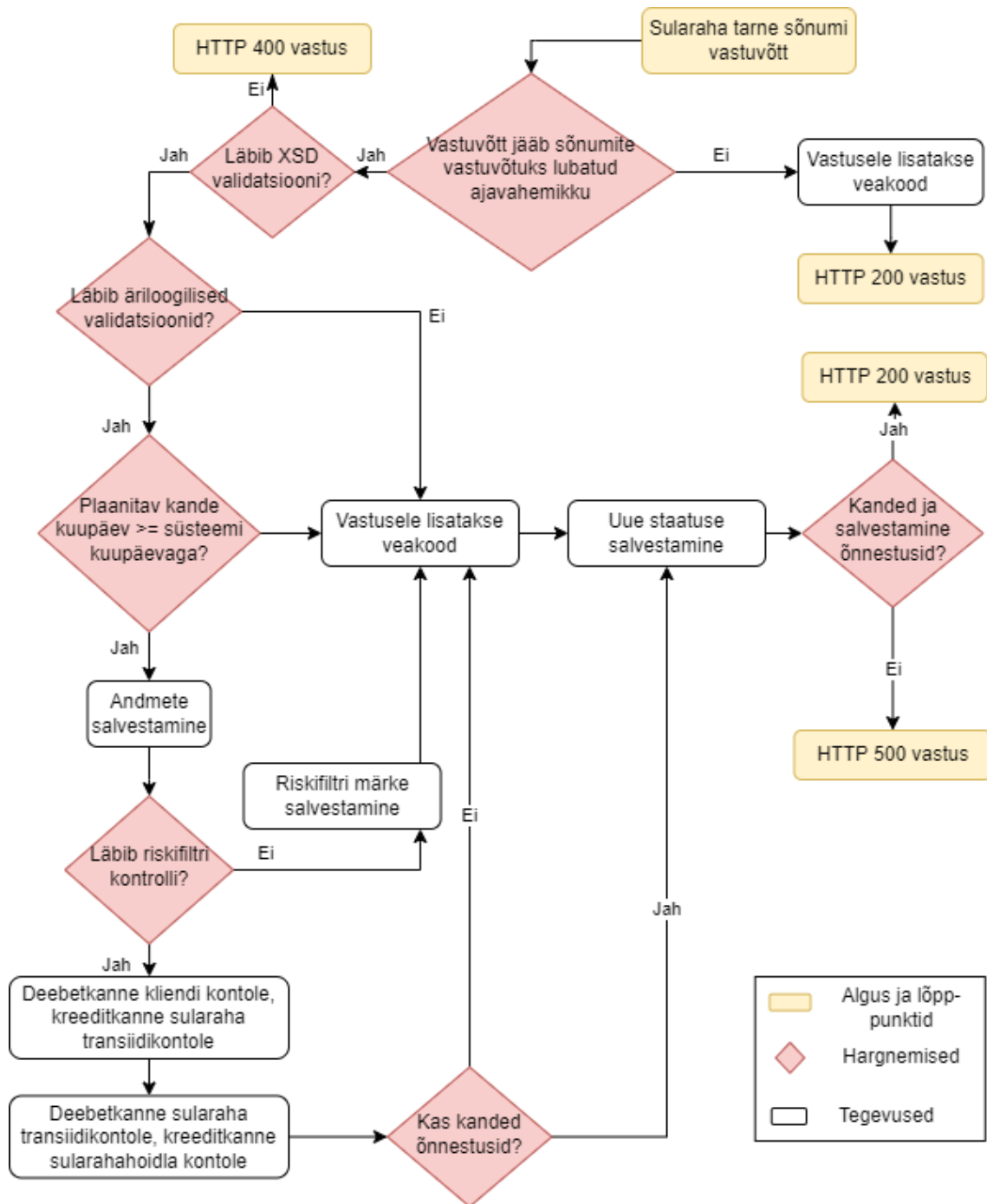
kontrollimisel olulised. Kuna panga infosüsteemis on seni kasutusel vaid XSD, otsustatakse XSD kasuks, et olulise põhjusega mitte suurendada erinevate lahendusviiside arvu.

3.3.4 Andmete valideerimine ja töövoogude ülevaade

Sularahavedajal sõnumi struktuuri valideerimiseks XSD-d pakkuda ei ole, seega tuleb fail näidiseandmete järgi luua. XSD kontroll tagab elementide järjekorra, elementide olemasolu ja andmete valideerimise. Näiteks juhul kui mingi sõne ei jää lubatud pikkusvahemikku või sõnet sisaldama pidanud element puudub, ei tehta järgnevaid ärioloogikal põhinevaid kontrole, vaid saadetakse HTTP 400 veakoodiga vastus [54]. Vastus sisaldab tekstilist viidet mittevalideerunud väljale ning põhjusele, miks väli ei valideerunud.

XSD kontrollis valideerunud andmete töötlemist jätkatakse ärioloogikal põhinevate kontrollidega, näiteks sõnumis viidatud äriregistri number ja pangakonto number peavad kuuluma samale ettevõttele. Oluline on ka kanneteks piisava vaba kontojäägi olemasolu ning kehtiv sularahaveo leping.

Alternatiivne variant oleks ärioloogilistes kontrollides ka andmete olemasolu ning lubatud andmevahemikke jälgida. Planeeritavas töövoos on võimalik teha mitu kontrollpunkti, seetõttu eelistatakse formaadi ja struktuuri kontroll teha kõige alguses, et vigaseid andmeid ei oleks vaja enam edasi töödelda. Joonistel 4 ja 5 on mõlema sõnumitüübi töövood. Tänu eraldi sammuna teostatavale XSD valideerimisele on teises valideerimise etapis vaja kontrollida põhiliselt ärioloogikat. Eeldatakse, et XSD validatsiooni edukalt läbinud sõnumites on kohustuslikud andmed olemas.



Joonis 4. Sularaha tarne töövoog.

kontrollide järjekord on vastavalt rakenduse spetsiifikale prioritseeritud, kuid infovahetuse mõttes ei ole vahet, mis järjekorras sularahavedaja veateated saab.

Arenduse alguses kaalutakse andmebaasi vahetabeli loomist, mis suudaks salvestada ühe sõnumi kohta palju veakoode, kuid sellest loobutakse, sest sularahavedaja andmetel on vigased sõnumid pigem erand kui reegel ning mitme veakoodi kuvamise kasutegur pole piisav, et nende salvestamiseks jaoks eraldi tabel teha. Samuti ei ole võimalik vastuses mitut veakoodi tagastada, sest sularahavedaja poolt ette antud vastuse struktuur seda ei võimalda.

Järgnevalt tuuakse ärioloogiliste kontrollide näiteid. Esimene kontroll on saabunud sõnumi UUID objektimärgise unikaalsuse kontrollimine. Andmebaasi tabelisse lisatud unikaalsuse piirangu tõttu [55] ei saaks korduvat UUID identifikaatorit sisaldavat sõnumit salvestada, seega on vaja esimesena kontrollida UUID unikaalsus ning duplikaadi korral saabunud sõnumi sisu logisse lisada ja tagastada korduvale UUID-le vastav veakood.

Teiseks kontrolliks planeeritakse sularahahoidla koodi olemasolu seadetes defineeritud sularahahoidla koodide hulgas ning hoidla aktiveeritud olekut, sest ilma seadistatud sularahahoidlateta ei ole pangasüsteemis kannete jaoks vajalikke kontosid loodud. Lisaks planeeritakse graafilisse kasutajaliidesesse hoiatused seadistamata sularaha kontode korral, mis peaks pangatöötajaid sularahakontode seadistamisele suunama.

Edasi kontrollitakse kliendi ja konto olemasolu andmebaasisüsteemis, kliendi konto aktiveeritust, kasutamisiiranguid ning kliendi äriregistri numbri ja konto omaniku vahelisi seoseid. Sularaha tarne puhul on oluline ka vaba kontojääk.

Enne kannete tegemist tehakse saabunud sõnumile riskifiltri kontroll. Riskifilter tähendab erinevaid eeldefineeritud reegleid, mille vastu tulevase kande infot kontrollitakse. Kui saabunud sõnumi info sobitub ühe või mitme riskifiltri definitsiooniga, filtri ja sõnumi identifikaatorid salvestatakse ning pangatöötajatele saadetakse teavitus riskifiltri märkega sõnumi kontrollimiseks. Riskifiltri väljundi jaoks planeeritakse eraldi vahetabel, mis sisaldab riskifiltri ja sõnumi identifikaatoreid. Eraldi tabel võimaldab mitu-mitmele seoseid ning kui riskifiltri identifikaatorit ei defineerita andmebaasis välisvõtmena, on riskifiltrit sõltumatult tema eelnevatest seostest võimalik kustutada.

Riskifiltri märke saanud sõnumi töötlemine on sularaha tarne ja korje puhul erinev. Sularaha korje puhul tehakse kanne sularahahoidlast transiidikontole ka riskifiltri märke korral, sularaha tarne puhul ei tehta riskifiltri märkega sõnumi töötlemisel kumbagi kannet. Et hilisemat riskifiltri käsitsi töötlemist lihtsustada, otsustatakse riskifiltri märke korral sõnumi vaates kuvada nupud, mis võimaldavad sõnumi riskifiltri märkest vabastada või sõnumi töötlemine tagasi lükata. Vabastamise korral tehakse kanded lõpuni, tagasilükkamise puhul taastatakse algsed kontode seisud ja sularaha korje puhul kantakse raha hoidla kontole tagasi. Et võimaldada tehtud kande tuvastamist ja andmete otsimist raha tagasi kandmiseks, lisatakse siinkohal sõnumite tabelisse uus väli kannete dokumendi numbriga salvestamiseks.

Peale kannete edukat toimumist uuendatakse eelnevalt andmebaasi salvestatud sõnumi olekut ning saadetakse päringule vastus.

4 Lahenduse realiseerimine

Sõnumivahetusviisi valikul lähtutakse panga äriloogika spetsiifikast, turvalisusest ning haldamise lihtsusest. Esialgu plaanitakse SFTP põhise lahendust, kuid töö käigus nõudeid täpsustades selgub võimalus kasutada ka rakendusliidese varianti. Lahenduse vahetus võimaldab lihtsustada arenduse loogikat ja seega otsustatakse rakendusliidese kasuks.

Lahendus realiseeritakse autori poolt eelnevalt kirjeldatud skoobi ulatuses, mis hõlmab arendustööde planeerimist, arenduse eeltöid, rakendusliidese funktsionaalsust, andmete valideerimist ja salvestamist, kasutajaliidese muudatusi ning suurt osa kasutajaliidese teste. Raportite muudatused ja uued raportid skoopi ei kuulu.

4.1 Muudatused andmebaasis

Kõigepealt analüüsitakse, millised andmebaasi muudatused on planeeritava funktsionaalsuse jaoks vajalikud. Kuna kõiki nõudeid ei ole alguses teada, luuakse osa allpool nimetatud tabelitest edaspidi, arenduse käigus.

Uute andmebaasi tabelite nimekiri on järgnev:

- sularahaveo sõnum,
- sularahaveo sõnumi ajalugu,
- sularahahoidla,
- sularahaveo leping,
- sularahaveo lepingu ajalugu,
- sularahaveo riskifiltri märgete tabel.

Et mõista, milliseid välju on vastuvõetavate sõnumite tabelisse vaja luua, selekteeritakse saabuvast sõnumist panga töövoogude jaoks olulised väljad. Tabelisse salvestatakse kliendiga, teostatava tehinguga ning sõnumi identifitseerimisega seotud info. Sõnumi olekut ning algupära märkivad väljad lisatakse juurde, et teha vahet käsitsi sisestatud ning

rakendusliidese kaudu saabunud infol. Nende väljade olekute järgi plaanitakse seadistada lubatavad tegevused kasutajaliideses.

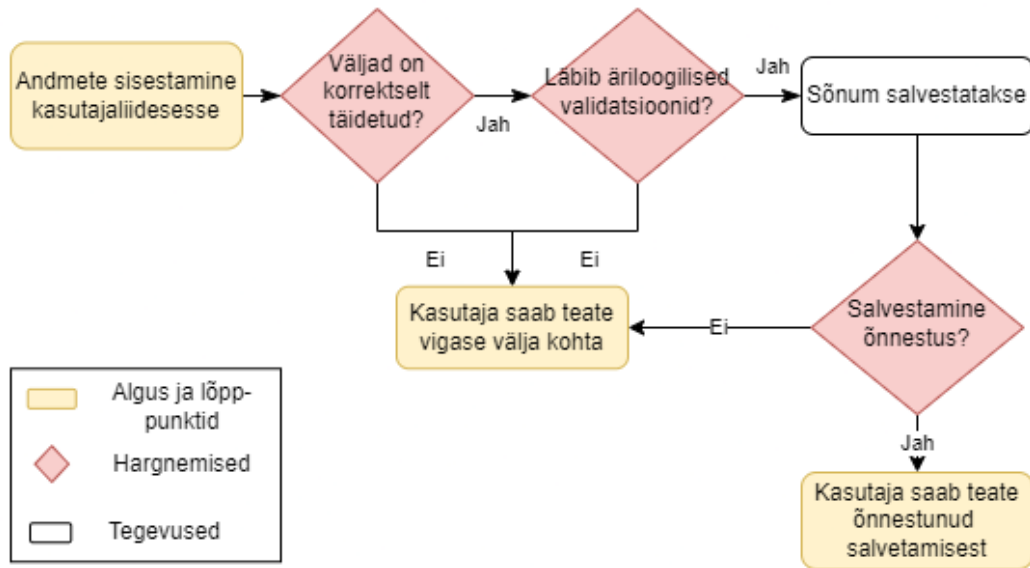
Kuna sularaha korje ning tarne sõnumitel on sama struktuur, saab need salvestada ühte tabelisse ning sõnumitüüpe on võimalik eristada tüüpi iseloomustava välja alusel. XSD kontrolli läbinud, kuid järgnevates kontrollides mittevalideerunud sõnumite kandeale lisatakse juurde veakood. Veakoodi salvestamiseks luuakse samuti eraldi väli.

Pangarakenduses on kasutusel Oracle andmebaasisüsteem [56], mis võimaldab lisaks päringutele luua PL/SQL protseduure ja funktsioone [57], [58]. Eelistatult kasutatakse andmebaasidega seotud tegevusteks just protseduure või funktsioone, mis aitavad Pythoni koodi lühemana hoida, samuti on nii lihtsam koodi organiseerida ja andmebaasipäringud ühte kohta koondada. [59]

4.2 Muudatused kasutajaliideses

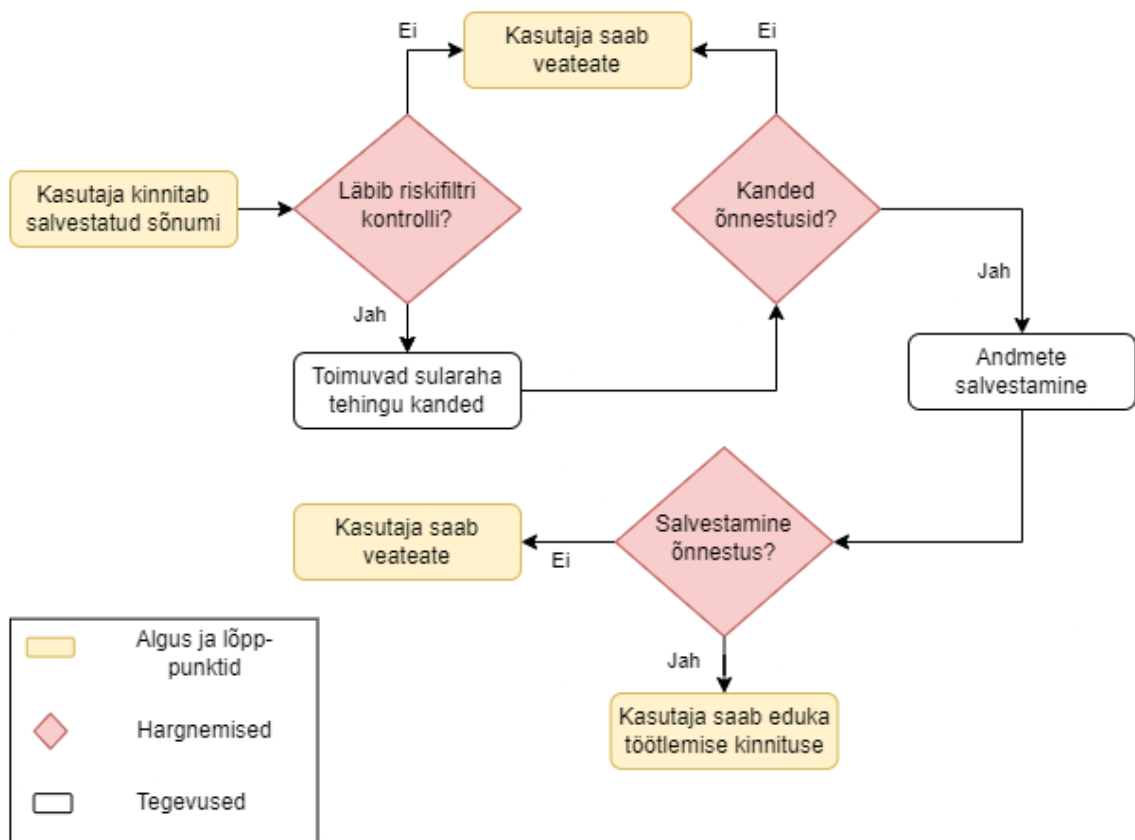
Peale sularahaveo lepingute tabeli loomist lisatakse kasutajaliidesesse sularahaveo lepingute alustamise ja lõpetamise võimalus. Kuna ühel kliendil võib olla mitu kontot, seotakse leping konkreetse kontoga, mitte kliendiga. Seejärel lisatakse rakendusliidese funktsionaalsus sularahavedaja sõnumite vastuvõtmiseks, mille töövoog on ülevahtlikult joonistel 4 ja 5.

Käsitsi sõnumite lisamise töövoog on üsna sarnane rakendusliidese töövooga, kuid kasutajaliideses ei ole võimalik salvestada vigaseid ja mittevalideeruvaid sõnumeid. Mittevalideeruvate andmete korral saab kasutaja veateate ja vigane väli märgitakse selgitava teavitusega. Sõnumi salvestamine läbi kasutajaliidese on joonisel 6.



Joonis 6. Sõnumi salvestamine kasutajaliideses.

Korrektse sõnumi salvestamise järel veel kandeid ei toimu, selleks on olemas eraldi nupp kinnitamiseks. Kinnitamine hõlmab ka kannete tegemist ning on joonisel 7.

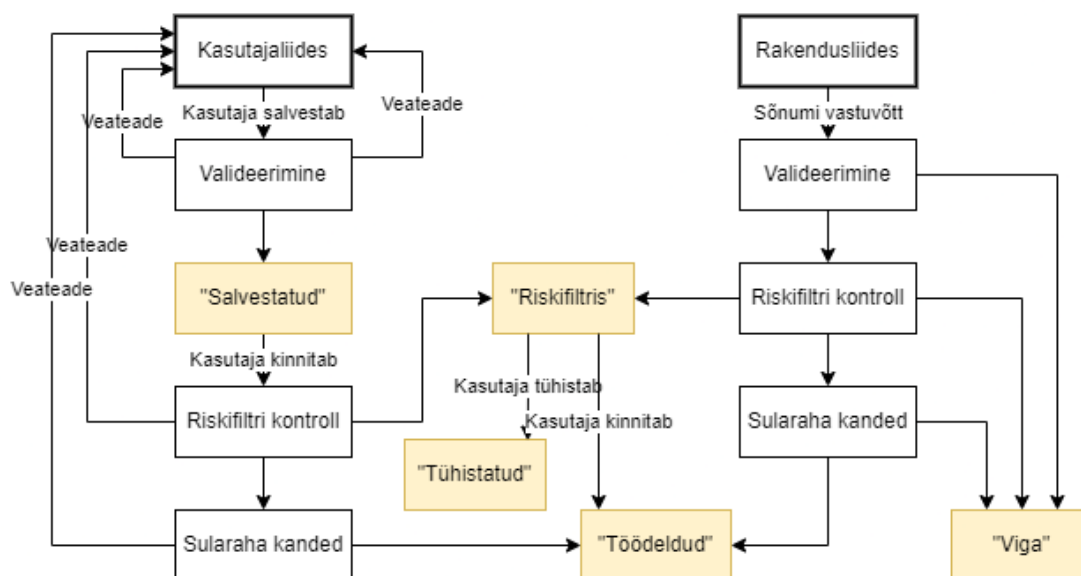


Joonis 7. Sõnumi kinnitamine kasutajaliideses.

Kinnitamise alternatiivina saab salvestatud sõnumit tühistada, mis jätab sõnumi andmed alles, kuid rohkem muuta ei luba. Kinnitamine ja tühistamine on eraldi sammud ja vastavad nupud on aktiivsed vaid salvestatud sõnumi korral.

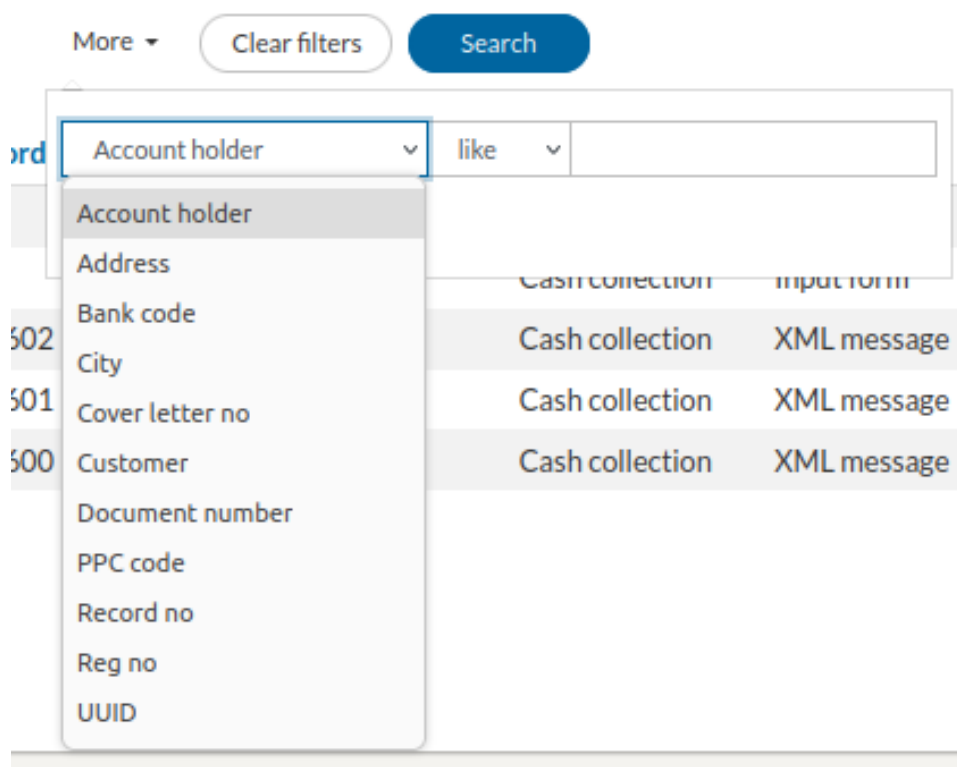
Töödeldud sõnumeid muuta ega tühistada ei saa. Kõik sõnumid, mis on kord juba salvestatud, jäävad alles.

Joonisel 8 on kollasega märgitud võimalikud sõnumi olekud kasutajaliidese ja rakendusliidese töövoogude korral.



Joonis 8. Sõnumite olekud andmete töötlemisel.

Paralleelselt rakendusliidese loomisega lisatakse kasutajaliidesele sõnumite vaated. Kasutajaliidese sõnumite nimekirja vaates on seadistatav veergude valik. Kuna sõnumil on rohkem välju, kui nimekirja vaates oleks ekraanile mõistlik kuvada, tehakse nimekirja vaikekuvasse eelvalik ja jäetakse ülejäänud veergude kuvamine kasutaja otsustada. Soovi korral saab kasutaja enda suva järgi veergusid vaatesse lisada või neid sealt eemaldada. Vaatesse lisatakse kuus otsingufiltrit ning kohandatav lisafilter (joonis 9). Lisafilter koondab enda alla vähemolulised otsinguvõimalused ja võtab vaid ühe filtri jagu ruumi.



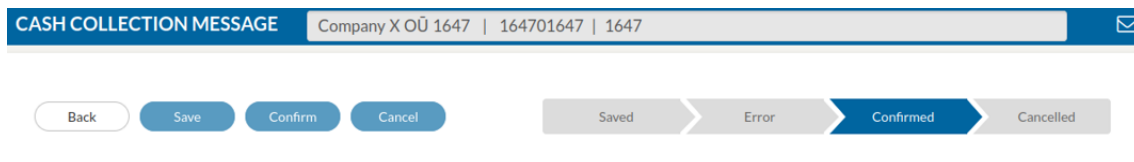
Joonis 9 Kohandatav filter.

Nimekirja vaate otsinguvõimaluste lisamisel analüüsitakse seniseid nimekirjade vaateid ning uuritakse panga esindaja arvamust sagedamini kasutatavate otsingute osas. Valminud nimekirja vaade koos otsinguvõimalustega on joonisel 10.

ID	Document number	Registration number	Account	Record number	Amount	Currency	Type	Origin	Status	Returned status
2	4357071	10011010011	LT619990199000001010	002	230.00	EUR	Cash delivery	Input form	Confirmed	–
1		10011010011	LT619990199000001010	001	13.00	EUR	Cash collection	Input form	Saved	–

Joonis 10. Sõnumite nimekirja vaade.

Algse nõude kohaselt peab sõnumi vaade jälgima üldist rakenduse kujundust, mis näeb ette olekute komponendi kuvamist sõnumi vaate üleval paremal nurgas (joonis 11).



Joonis 11. Algne olekute komponendi kuvamise kujundus.

Plaanitakse kasutada nelja erinevat sõnumi olekut, mida olekute komponent kuvab horisontaalselt üksteise järel. Samas ei ole olekute järgnevus töövoogudes kujundusega kooskõlas, sest mitmes kohas esineb hargnemisi ja kõik olekud ei ole iga kord võimalikud.

Et muuta olekute kuvamine loogilisemaks, määratletakse kõik võimalikud töövoole olekute jadad ning kuvatakse kasutajale ainult selle konkreetse töövoole variandi olekud. Kui konkreetne olek ei ole tühistatud või vigane, eelistatakse edukat töövoogu (salvestatud sõnumis kuvatakse „Salvestatud“ → „Kinnitatud“).

Määratletud olekute jadad on järgnevad:

- „Salvestatud“ → „Kinnitatud“,
- „Salvestatud“ → „Tühistatud“,
- „Kinnitatud“,
- „Viga“.

Kahjuks ei ole ka see variant sobiv, sest rakendusliidese kaudu saabunud sõnumite töötlemisel kuvatakse vaid ühte olekukomponenti, mis värvilt kattub rakenduse kasutajaliidese muude nuppude värviga ja on üksinda kuvatult kasutaja jaoks segadusse ajav [60].

Lisaks tekib arenduse käigus vajadus viienda oleku järele ning selles etapis otsustatakse kuvada oleku info eraldi väljal ning loobutakse olekute komponendi kasutamisest ühe sõnumi vaates. Joonisel 12 on näha lõplik sõnumi kujundus ilma olekute komponendita, oleku info on kuvatud väljal „*Status*“.

CASH COLLECTION MESSAGE		Company c465d503-9 OÜ 33885453661 12437		✉ ⓘ ⏻	
<div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> Back Save Confirm Cancel </div>					
Message information:					
Message ID	2	Status	Confirmed		
UUID	{22723522-6F1F-44CB-C18B-333132300}	Entry date	10.03.2023		
PPC code	FS1	Sending time	06.02.2023 09:18:22		
Record no	123600	Cover letter date	23.03.2023		
Cover letter no	663782	Origin	XML message		
Bank code		Risk filters			
Transaction information:					
Reg no	33885453661	Customer	Company c465d503-9 OÜ		
Account		Account holder	Company c465d503-9 OÜ		
City	City05	Currency	EUR		
Address	Address05	Amount	20.00		
Returned status	Successfully processed	Type	Cash collection		

Joonis 12. „Kinnitatud“ olekus sõnumi vaade.

Kasutajaliidese töövoos võimaldatakse sisestatud andmete salvestamine ka enne kinnitamist. Kinnitamise jaoks luuakse eraldi samm. Käsitsi andmete sisestamisel tekib inimlikke vigu ning seetõttu antakse kasutajale võimalus andmeid enne kannete tegemist kontrollida ja parandada. Muudatuste tegemist ei piirata kasutajaliidese sisestatud ja salvestatud, kuid töötlemata sõnumi korral.

Järgmine kasutajaliidese muudatus puudutab sularaha tehingute riskifiltri funktsionaalsust. Pangarakenduses on riskifilter maksete analüüsimiseks juba kasutusel ning suures osas saab seda taaskasutada. Koodi struktuuri mõttes oleks selgem teha sularaha riskifilter täiesti eraldi funktsionaalsusena, kuid see hõlmaks kogu senise riskifiltri koodi ümber kirjutamist ja eraldi vaadete loomist, seepärast otsustati aja kokkuhoidmiseks taaskasutada kattuvat osa riskifiltri funktsionaalsusest ja maksete riskifiltri vaatest.

Sularahaveo sõnumite riskifiltri paigutus menüüs oleks loogilisem sularahale eraldatud menüüs, kuid see asukoht tähendaks pangatöötajatele rohkem klikke, sest eri tüüpi riskifiltreid haldavad samad töötajad. Seetõttu lisatakse olemasolevasse maksete riskifiltri vaatesse riskifiltri tüübi valik ning sularaha riskifiltri lisamisel on vaja vaid filtri tüüp ära muuta.

Sularaha riskifiltri funktsionaalsuses on võimalik kasutada vaid mingit osa maksetes kasutatavatest väljadest, mittevajalikud väljad peidetakse sularaha filtri tüübi valimisel. Joonisel 13 on sularahaveo riskifiltri vaade.

The screenshot shows a web interface for configuring a risk filter. At the top, there's a blue header with the text 'RISK FILTER' and three icons: an envelope, a question mark, and a power button. Below the header, there are two buttons: 'Back' and 'Save'. The main area contains several form elements: 'Filter type' with radio buttons for 'Payment' and 'Cash collection' (the latter is selected); an 'ID' text input; a 'Description' text input; 'Customers' with a dropdown arrow, a search input containing 'Find customer', and a close button; 'Accounts' with a dropdown arrow; 'Incoming/Outgoing' with a dropdown arrow; 'Channel' with a dropdown menu showing 'CASH COLLECTION'; and 'Limit amount' with a dropdown menu. On the right side, there are date pickers for 'Start date' (set to 29.03.2023) and 'End date', and a 'Last change by' text input.

Joonis 13. Sularahaveo riskifiltri vaade.

Uue filtri loomisel saab filtri tüüpi vahetada raadionupuga “Filter type“. Eeldusel, et sularaha jaoks kasutatavat filtrit läheb harvem vaja, jäetakse uue filtri loomisel vaikimisi valituks maksete filter.

4.3 Lahenduse testimine

Arendatavaid muudatusnõudeid testib autor arenduse käigus ja iga arendusülesande lõpetamisel. Arendus hõlmab kasutajaliidese testide ja üksuste testide lisamist pangarakendusele, samuti tuleb peale iga muudatusnõude arendust olemasolevad testid üle kontrollida ning vajadusel katki läinud testid parandada.

Arendusülesandeid testitakse ka töögrupi testijate poolt. Testimisel kasutatakse näiteks halli kasti meetodit [61], funktsionaalset testimist [62], spetsifikatsioonidel põhinevat testimist ning regressioonitestimist [63]. Testijatelt saadakse mitmel juhul olulist tagasisidet ka rakenduse loogika ja kasutusmugavuse kohta.

Arenduse funktsionaalsusi plaanitakse testida sularahavedajapoolse testkeskkonna abil, mis saadab välja kokkulepitud vormingus sõnumeid. Erinevatel põhjustel testkeskkondade vahelise testimise algus venib ning arenduse jooksul kasutatakse

rakendusliidesesse sõnumite saatmiseks ja vastuste kontrollimiseks Postmani tarkvara [64].

Rakendusliidese töökiiruse testimisel mõõdetakse 99 päringu keskmist kiirust. Kui sularahaveo sõnumite tabelis eelnevalt kandeid sisestatud ei ole, on keskmine päringu töötlemise kiirus 21.9 ms, mis arvutatakse Apache veebiserveri logifaili põhjal. Kui baasi on lisatud 200000 kannet, mõõdetakse töötlemise kiiruseks 42.6 ms. Seejärel tehakse sularaha sõnumite töötlemiseks kasutatavate päringute analüüs ja lisatakse tabelitesse indeksid päringute kiirendamiseks [65]. Indeksid lisatakse tabelite veergudele, mille järgi päringud sõnumite töötlemisel kandeid otsivad [66]. Peale indeksite lisamist muutub ühe sõnumi töötlemine 200000 salvestatud sõnumi korral keskmiselt 14.4 ms võrra kiiremaks. Indeksite kasutamine muudab 100 kandega andmebaasi päringud keskmiselt 3.9 ms võrra aeglasemaks, mis tuleneb indeksi lisamisest tingitud aeglustumisest. Aeglustumine on põhjustatud indeksite uuendamisvajadusest muudetavates tabelites [67]. Samas indeksite lisamisest saadav kiiruse kasv kaalub suurema hulga kannete puhul aeglustumise üles. Mõõdetud kiirused on tabelis 3.

Tabel 3. Päringute keskmine töötlemiskiirus enne ja pärast indekseerimist.

99 päringu keskmine kiirus	100 kandega andmebaas	200000 kandega andmebaas
Enne indeksite lisamist	21.947 ms	42.638 ms
Peale indeksite lisamist	25.841 ms	28.331 ms

Arenduse valmimisel teeb pank omakorda vastuvõtutestimise, eelkäidu keskkonnas kontrollitakse sularahaveo funktsionaalsust ja liidestuse tööd ning seejärel võetakse uus rakendus kasutusele.

4.4 Hinnang loodud lahendusele

Lahenduse ülesehituse osas tuleb vastavalt olemasoleva rakenduse funktsionaalsusele ning kasutajate töö spetsiifikale teha mõningaid mööndusi.

Koodi struktuuri mõttes oleks loogilisem luua sularaha riskifiltri jaoks eraldi tabel ning lisada sularaha tehingute riskifilter sularaha alammenüüsse. Selle vastu räägib aga asjaolu, et samad kasutajad haldavad kõiki riskifiltreid ning neil oleks mugavam töötada

samas vaates kõikide seadistatavate riskifiltritega. Otsustatakse kasutajate mugavuse kasuks ning luuakse riskifiltri lahenduse jaoks eraldi vaheleht juba olemasolevasse maksete riskifiltri vaatesse. Uue filtri vaate loomisel jälgitakse varem kasutuses olnud riskifiltri loogikat, et uue filtrivariandi kasutamine oleks intuitiivne ja ei tekitaks kasutajates segadust.

XSD valideerimismeetodi miinuseks jääb vigase vastuse ebaspetsiifilisus, sest vastuses tagastatakse HTTP veakood 400, koos kehas oleva XSD validatsiooni käigus genereeritud veateatega, kuid veateadet sularahavedaja poolt kinnitatud veakoodiks ei tõlgita. Vastussõnumite saaja vaatepunktist oleks informatiivsem sõnumi info XSD kontrollimisel lisada ka kodeering tekstilisest vastusest veakoodiks. Ajapuuduse tõttu piirduakse lahenduses XSD valideerimisel saadava veateate vastusesse lisamisega, kuid ei välistata veakoodideks tõlkimise muudatust tulevikus, kui sularahavedaja ja pank seda peaks soovima. Käesolevas arenduses ei peeta muudatust vajalikuks - pank eeldab, et peale testimist on vigaste sõnumite saabumine vaid väga harv erandjuht. Samuti puudub piisav info vastussõnumi veakoodide täpsuse vajaduse kohta sularahavedaja poolelt. Alles arenduse viimases etapis on võimalik sularahavedaja testühenduse kaudu rakendusliidese toimimist katsetada, mis seab arenduse muudatustele ka ajalised piirangud.

Muudes aspektides jääb töö autor loodud lahendusega rahule – kliendi nõuete ja ajapiirangu raames luuakse parim võimalik lahendus.

5 Tagasiside ja edasised plaanid

Käesoleva töö skoobist jääb välja sularaha liikumiste kajastamine raportites. Raportite loomine on lähiajal plaanis, sest sularahaveo info kajastamine raportites on pangale kohustuslik, kuid töö valmimise hetkel ei ole raportite arendamiseks veel piisavalt infot.

Loodud pangarakendus arendatakse spetsiifiliselt konkreetse maa panga tarbeks, vastavalt kohaliku sularahavedaja infosüsteemile ning nende juba välja kujunenud töövoogudele. Edaspidi on plaan on lisada sularaha käsitlemise võimekus sama panga teiste riikide esindustele, mis võib tähendada riigiti erineva struktuuriga sõnumite kasutamiste vajadust.

Riskifiltri märkega sõnumite täielik haldamine läbi kasutajaliidese jääb käesolevas töös lahendamata. Katmata on järgmised juhud:

1. Sularaha tarne sõnum saab riskifiltri märke ja lükatakse tagasi. Kui on toimunud selgitav suhtlus sularahavedajaga ja sõnum otsustatakse siiski töödelda, siis saab sõnumi läbi kasutajaliidese sisestada ja kinnitada. Enne seda on siiski vaja panga ja sularahavedaja töötajate vahelist suhtlust.
2. Sularaha korje sõnum saab riskifiltri märke ja tühistatakse kasutaja poolt kasutajaliidese kaudu, kuid tühistamise kohta sularahavedaja läbi liidese infot ei saa. Tühistamise info on vaja sularaha vedajale edastada.

Esimese juhu lahendamiseks tuleb kasutajaliidessesse sisestatava sõnumi info vastu võtta näiteks e-posti teel. Teisel juhul tuleb sularahavedajat tühistamisest teavitada. Mõlemal juhul on vaja panga ning sularahavedaja poolset lisakommunikatsiooni, kuid eeldatakse, et riskifiltri tabamused sularahaveo osas on pigem harv erand ja seepärast ei planeerita probleemi lahendamiseks eraldi funktsionaalsust.

Nimetatud juhtude jaoks parima lahenduse leidmine nõuaks rohkem väljakujunenud koostööd panga ja sularahavedaja vahel. Kui riskifiltri tabamuste hulk peaks olema oodatust sagedasem, plaanitakse vastav lahendus edaspidi.

6 Kokkuvõte

Käesoleva töö raames lisati panga infosüsteemi rakendusliides, mis võimaldas pangal sularahavedaja poolt saadetud sõnumeid vastu võtta. Rakendusliidest eelistati kolme võimaliku lahendusvariandi hulgast pangapoolse arenduse lihtsama loogika tõttu, mis vajab arendustööde tegemiseks vähem kolmepoolseid täpsustusi. Samuti oli rakendusliidese arendamine võrreldes FTP serveril põhineva sõnumite vahetuse lahendusega hinnanguliselt vähem ajakulukas.

Skoopi kuulusid nõuete analüüs, struktureerimine, modelleerimine ja prioritseerimine, muudatused andmebaasis, sobiva lahenduse valik ning arendustööd. Töös ei lahendatud XSD kontrolli vastuseks saadetavate veateadete tõlkimist veakoodideks, samuti jäid skoobist välja riskifiltri mäkkega sularaha korje sõnumi tagasilükkamise info saatmine läbi rakenduse ja riskifiltri mäkkega sularaha tarne sõnumi automaatne töötlemine.

Rakendusliidese arendus hõlmas sõnumite vastuvõtmist, sõnumi struktuuri ning andmetüüpide kontrolli, äriloogilisi kontrole, logisid, riskifiltri kontrolli, saabunud info andmebaasi salvestamist ning kannete tegemist. Peale kannete toimumist saadeti päringule kinnitav vastus. Kontrollid sisaldasid veakoodi lisamist koostatavasse vastusesse, samuti pangatöötajate teavitamist riskifiltri mäkke korral.

Sularaha tehingute info kuvamiseks loodi pangatöötajate jaoks kasutajaliidesesse sõnumite vaade, mis võimaldab sõnumite otsingut ja sorteerimist ning kuvab kasutajale sõnumi info. Kasutajal on võimalik sõnumeid lisada, salvestada ja kinnitada. Kasutajaliidesesse lisati ka sularahalepingute ja sularahakassade haldus.

Pank jäi loodud lahenduse loogikaga rahule ja lähiajal lisatakse valminud funktsionaalsus käidueelsesesse keskkonda vastuvõtutestimiseks.

Kasutatud kirjandus

- [1] „What is Agile Software Development?“, *Agile Alliance* |, 29. juuni 2015. <https://www.agilealliance.org/agile101/> (vaadatud 25. märts 2023).
- [2] „E-ITS Seletav sõnaraamat“, *Rakendusliidese mõiste*. Vaadatud: 5. aprill 2023. [Online]. Available at: <https://eits.ria.ee/et/seletav-sonaraamat/r?id=f9622ae6190ea35b420ea100f1ecaf54>
- [3] NetSuite.com, „Accounting 101: Debits and Credits“, *Oracle NetSuite*, 6. mai 2022. <https://www.netsuite.com/portal/resource/articles/accounting/debits-credits.shtml> (vaadatud 5. aprill 2023).
- [4] „What is TLS & How Does it Work? | ISOC Internet Society“, *Internet Society*. <https://www.internetsociety.org/deploy360/tls/basics/> (vaadatud 11. aprill 2023).
- [5] „AKIT - Andmekaitse ja infoturbe leksikon“, *SSH*. Vaadatud: 24. märts 2023. [Online]. Available at: https://akit.cyber.ee/term/751-ssh#t_751
- [6] „e-Teatmik: IT ja sidetehnika seletav sõnaraamat“. <http://vallaste.ee/index.asp> (vaadatud 13. aprill 2023).
- [7] „What is Scrum?“, *Scrum.org*. <https://www.scrum.org/learning-series/what-is-scrum> (vaadatud 25. märts 2023).
- [8] „AKIT - Andmekaitse ja infoturbe leksikon“, *UUID*. Vaadatud: 7. aprill 2023. [Online]. Available at: <https://akit.cyber.ee/term/7458-uuid>
- [9] „Understanding Relationship Mappings“, *Working with Foreign keys*. https://docs.oracle.com/cd/B14099_19/web.1012/b15900/relatn002.htm (vaadatud 10. aprill 2023).
- [10] „Aurea Messenger“, *Aurea Software*, 30. juuli 2020. <https://www.aurea.com/library/messenger/> (vaadatud 30. aprill 2023).
- [11] C. Larman, *Agile and Iterative Development: A Manager's Guide*, Addison-Wesley Professional, 2004.
- [12] „Iterative is not Necessarily Incremental“, *Scrum.org*. <https://www.scrum.org/resources/blog/iterative-not-necessarily-incremental> (vaadatud 11. mai 2023).
- [13] „Introduction to DSM“, *The Design Structure Matrix (DSM)*, 25. aprill 2019. <https://dsmweb.org/introduction-to-dsm/> (vaadatud 19. märts 2023).
- [14] International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge*, 3rd edition. Toronto: International Institute of Business Analysis, 2014.
- [15] *Guide To The Software Engineering Body Of Knowledge Swebok R Book Pdf*. Vaadatud: 8. aprill 2023. [Online]. Available at: <https://youbookinc.com/pdf/guide-to-the-software-engineering-body-of-knowledge-swebok-r>
- [16] M. Laguna ja J. Marklund, „Flowcharts“, *Business Process Modeling, Simulation and Design*, Milton, UNITED KINGDOM: CRC Press LLC, 2018, lk 118–121. Vaadatud: 16. aprill 2023. [Online]. Available at: <http://ebookcentral.proquest.com/lib/tuee/detail.action?docID=5606290>
- [17] „Opportunity Scoring“. <https://www.productplan.com/glossary/opportunity-scoring/> (vaadatud 13. aprill 2023).

- [18] „Kano Model“. <https://www.productplan.com/glossary/kano-model/> (vaadatud 13. aprill 2023).
- [19] „Quality Function Deployment (QFD)“. <https://www.productplan.com/glossary/quality-function-deployment/> (vaadatud 20. november 2022).
- [20] N. Mead, „Requirements Prioritization Case Study Using AHP“, lk 19.
- [21] I. Olaronke, R. Ikono, ja I. Gambo, „An Appraisal of Software Requirement Prioritization Techniques“, kd 1, apr 2018, doi: 10.9734/AJRCOS/2018/40763.
- [22] F. Hujainah, R. Bakar, ja M. Abdulgaber, „Investigation of Requirements Interdependencies in Existing Techniques of Requirements Prioritization“, *Tehnicki Vjesnik*, kd 26, lk 1186–1190, juuli 2019, doi: 10.17559/TV-20171129125407.
- [23] B. Viswanathan, „Best Dependency Structure Matrix (DSM) Guide for 2023“, *Project-Management.com*, 4. mai 2021. <https://project-management.com/dependency-structure-matrix-dsm/> (vaadatud 19. märts 2023).
- [24] „What is a Sprint?“, *Scrum.org*. <https://www.scrum.org/resources/what-is-a-sprint-in-scrum> (vaadatud 5. mai 2023).
- [25] „Bottle: Python Web Framework — Bottle 0.13-dev documentation“. <https://bottlepy.org/docs/dev/> (vaadatud 13. märts 2023).
- [26] „The 3-Clause BSD License | Open Source Initiative“. <https://opensource.org/licenses/BSD-3-Clause> (vaadatud 18. oktoober 2022).
- [27] „Open Source Licenses Explained“, *Mend*. <https://www.mend.io/resources/blog/open-source-licenses-explained/> (vaadatud 14. aprill 2023).
- [28] „Home · Wiki · GNOME / libxml2 · GitLab“, *GitLab*. <https://gitlab.gnome.org/GNOME/libxml2/-/wikis/home> (vaadatud 10. aprill 2023).
- [29] „Home · Wiki · GNOME / libxslt · GitLab“, *GitLab*. <https://gitlab.gnome.org/GNOME/libxslt/-/wikis/home> (vaadatud 10. aprill 2023).
- [30] „lxml - Processing XML and HTML with Python“. <https://lxml.de/> (vaadatud 18. aprill 2023).
- [31] „What is REST“, *REST API Tutorial*. <https://restfulapi.net/> (vaadatud 13. aprill 2023).
- [32] „IBM Documentation - SOAP“, 13. september 2022. <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/en/cics-ts/5.4?topic=services-soap-architecture-message-format> (vaadatud 14. aprill 2023).
- [33] „IBM Documentation - WS-Security“, 15. september 2022. <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/en/app-connect/11.0.0?topic=security-ws> (vaadatud 10. aprill 2023).
- [34] A. Walker, „SOAP Web Services Tutorial: What is SOAP Protocol? EXAMPLE“, 16. jaanuar 2020. <https://www.guru99.com/soap-simple-object-access-protocol.html> (vaadatud 12. aprill 2023).
- [35] C. Team, „FTP wasn't designed to be secure - how to mitigate the risks“, *Cerberus FTP Server*, 8. veebruar 2019. <https://www.cerberusftp.com/how-secure-is-ftp-how-you-can-mitigate-the-risks-of-using-file-transfer-protocol/> (vaadatud 2. aprill 2023).
- [36] „IBM Documentation“, 14. aprill 2021. <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us->

- south.containers.appdomain.cloud/docs/en/i/7.2?topic=options-ftp-security (vaadatud 14. aprill 2023).
- [37] „HTTP vs HTTPS - javatpoint“, *www.javatpoint.com*.
<https://www.javatpoint.com/http-vs-https> (vaadatud 7. aprill 2023).
- [38] M. A. ja N. Ikram, „Transport Layer Security (TLS)--A Network Security Protocol for E-commerce“, *Technocrat PNEC Research Journal*, jaan 2004.
- [39] „What is Transport Layer Security? | TLS protocol“, *Cloudflare*.
<https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/> (vaadatud 15. aprill 2023).
- [40] drthomasshinderme, „How FTP port requests challenge firewall security“, *TechRepublic*, 4. märts 2002. <https://www.techrepublic.com/article/how-ftp-port-requests-challenge-firewall-security/> (vaadatud 7. aprill 2023).
- [41] J. C. Villanueva, „Active vs. Passive FTP Simplified: Understanding FTP Ports | JSCAPE“. <https://www.jscape.com/blog/active-v-s-passive-ftp-simplified> (vaadatud 8. aprill 2023).
- [42] „FTPS Port number, risks and alternative explained“, *SFTP To Go: Managed SFTP/FTPS Cloud Storage as a Service*, 8. detsember 2021.
<https://sftptogo.com/blog/ftps-port-what-port-numbers-are-used-in-file-transfer/> (vaadatud 24. märts 2023).
- [43] „SFTP vs. FTPS: Understanding the 8 Key Differences“, *Spiceworks*.
<https://www.spiceworks.com/tech/networking/articles/sftp-vs-ftps/> (vaadatud 24. märts 2023).
- [44] „SFTP vs. FTPS: The Key Differences | GoAnywhere MFT“. <https://www.goanywhere.com/blog/sftp-vs-ftps-the-key-differences> (vaadatud 24. märts 2023).
- [45] „How to Migrate from FTP / FTPS to SFTP - Thorn Technologies“, 29. jaanuar 2020. <https://thorntech.com/migrate-ftp-to-sftp/> (vaadatud 6. november 2022).
- [46] „What is SFTP? | Comprehensive Guide to SFTP with its Advantages“, *EDUCBA*, 17. oktoober 2019. <https://www.educba.com/what-is-sftp/> (vaadatud 8. aprill 2023).
- [47] „What is HTTP POST Request Method?“, *ReqBin*, 15. september 2022.
<https://reqbin.com/Article/HttpPost> (vaadatud 5. aprill 2023).
- [48] D. Brunato, „xmlschema: An XML Schema validator and decoder“. Vaadatud: 6. aprill 2023. [OS Independent]. Available at:
<https://github.com/sissaschool/xmlschema>
- [49] „The MIT License“, *Open Source Initiative*, 31. oktoober 2006.
<https://opensource.org/license/mit/> (vaadatud 26. märts 2023).
- [50] „elementpath“. Scuola Internazionale Superiore di Studi Avanzati, 11. oktoober 2022. Vaadatud: 6. november 2022. [Online]. Available at:
<https://github.com/sissaschool/elementpath>
- [51] „xml.dom.minidom — Minimal DOM implementation — Python 3.11.0 documentation“. <https://docs.python.org/3/library/xml.dom.minidom.html> (vaadatud 13. aprill 2023).
- [52] „Benchmarks and Speed“. <https://lxml.de/performance.html> (vaadatud 6. november 2022).
- [53] „DTD vs XSD - javatpoint“, *www.javatpoint.com*.
<https://www.javatpoint.com/dtd-vs-xsd> (vaadatud 6. november 2022).
- [54] „Hypertext Transfer Protocol (HTTP) Status Code Registry“. <https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml#http-status-codes-1> (vaadatud 19. märts 2023).

- [55] „Oracle / PLSQL: Unique Constraints“. <https://www.techonthenet.com/oracle/unique.php> (vaadatud 19. märts 2023).
- [56] „Oracle | Cloud Applications and Cloud Platform“. <https://www.oracle.com/> (vaadatud 20. märts 2023).
- [57] „What is PL/SQL“, *Oracle Tutorial*. <https://www.oracletutorial.com/plsql-tutorial/what-is-plsql/> (vaadatud 3. veebruar 2023).
- [58] „Database PL/SQL Language Reference“. <https://docs.oracle.com/database/121/LNPLS/packages.htm#LNPLS00901> (vaadatud 10. mai 2023).
- [59] „Database PL/SQL Language Reference“. https://docs.oracle.com/cd/E11882_01/appdev.112/e25519/overview.htm#LNPLS134 (vaadatud 3. veebruar 2023).
- [60] W. Grant, „13 Make Interactive Elements Obvious and Discoverable“, *101 UX Principles*, Second Edition. Packt, 2022.
- [61] „Gray Box Testing | Software Testing“, *GeeksforGeeks*, 25. aprill 2019. <https://www.geeksforgeeks.org/gray-box-testing-software-testing/> (vaadatud 9. mai 2023).
- [62] T. Hamilton, „What is Functional Testing? Types & Examples“, 4. aprill 2020. <https://www.guru99.com/functional-testing.html> (vaadatud 9. mai 2023).
- [63] „What Is Regression Testing? Definition, Tools, Examples“, *katalon.com*. <https://katalon.com/resources-center/blog/regression-testing> (vaadatud 9. mai 2023).
- [64] „Postman API Platform | Sign Up for Free“. <https://www.postman.com/> (vaadatud 3. veebruar 2023).
- [65] „Database Concepts“. https://docs.oracle.com/cd/E11882_01/server.112/e40540/indexiot.htm#CNCPT811 (vaadatud 17. aprill 2023).
- [66] „7 Performance Tuning“. https://docs.oracle.com/cd/B12037_01/text.101/b10729/aoptim.htm#i1006756 (vaadatud 17. aprill 2023).
- [67] „The Power of Indexing“. http://www.dba-oracle.com/t_indexing_power.htm (vaadatud 17. aprill 2023).

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Diana Vaher

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Panga ja sularahavedaja vahelise sõnumivahetuse arendus“, mille juhendaja on Kaido Kikkas.
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

13.05.2023

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Ärinõuetel põhinev tegevusplaan

	Arenduse eeltööd - 1	F/MF
1	Sularaha kannete sõnumite tabel	F
2	Sularaha kannete sõnumite tabeli ajalugu	F
3	Sularahahoidlate tabel	F
4	Sularahahoidlate ajaloo tabel	F
5	Sularaha lepingute tabel	F
6	Sularaha lepingute ajaloo tabel	F
7	Sularaha riskifiltri väljundtabel	F
8	Sularaha transaktsioonide koodid transaktsiooni koodide tabelisse	F

	Loodava funktsionaalsuse töövood - 2	F/MF
9	Sõnumi vastuvõtmise loogika pangarakenduses	F
10	Saabunud sõnumi formaadi valideerimine	F
11	Saabunud sõnumi äriiline valideerimine	F
12	Sõnumi vastused ja veakoodid	F
13	Valideerunud sõnumite korral kannete tegemine	F
14	Sularaha tehingute riskifiltri funktsionaalsus	F
15	Sõnumite tagasilükkamine, mis ei ole saabunud lubatud ajavahemikus	F

	Kasutajaliides - 3	F/MF
16	Sularaha lepingut tähistav märkeruut kontodetailide vaates	F
17	Sõnumite nimekirja vaade	F
18	Detailne sõnumi vaade	F
19	Sularahahoidlate nimekirja vaade	F
20	Detailne sularahahoidla vaade	F
21	Uus alammenüü sularaha sõnumite ja hoidlate jaoks	F
22	Riskifiltri lisamine kasutajaliidesesse	F
23	Kasutajaliides peab jälgima juba olemasoleva rakenduse kujundust	MF
24	Kasutajaliides peab jälgima juba olemasoleva rakenduse struktuuri	MF
25	Kasutajaliideses peab saama ka käsitsi teha kõiki automatiseeritud operatsioone	F

	Seaded - 4	F/MF
26	Sularaha tehingute jaoks lubatud valuutade seadistus	F
27	Sularaha lepingute lisamise ja lõpetamise funktsionaalsus	F
28	Klassifikaator sularaha transaktsioonide ajavahemiku määramiseks	F
29	Summapõhise teenustasu lisamine tehingule	F
30	Sularaha tehingute kuupõhise hooldustasu lisamine	F

	Raportid - 5	F/MF
31	Sularaha tehingute info aruanded koostavasse päringumootorisse	F
32	Lisada Forma-F-3 Leedu regulatsioonile vastav raport	F
33	Lisada sularahatehingute info F607 alamraporti Ila alla	F
34	Sularaha tehingute info seadistamine prinditavale lepingule	F

Lisa 3 – Muudatusnõuete omavahelised sõltuvused

Nr	Kategooria	Sõltuvad nõuded	Sõltub nõuetest
1	Sularaha tehingute jaoks lubatud ajavahemiku klassifikaator	4, 20	
2	Sularahaveo lepingute andmete ja lepingu muudatuste ajaloo tabelid	6, 14, 20	
3	Sularaha liikumiste infot salvestav tabel koos ajaloo tabeliga	4, 7, 8, 14, 17, 18, 19, 20, 27	
4	Sõnumeid vastuvõttev rakendusliides	5, 6, 7, 8, 14, 15, 16, 17, 18, 19, 20, 21, 22, 27	1, 3
5	Saabunud sõnumi andmete vormi ja pikkuse kontroll	6, 14, 20, 27	4
6	Saabunud sõnumi andmete ärioloogiline kontroll	7, 8, 14	2, 4, 5, 9
7	Saabunud sõnumite nimekirja vaade koos otsingufiltritega	8, 14	3, 4, 6, 8
8	Sularaha sõnumite detailne vaade koos nuppude ja seotud funktsionaalsusega	7, 9, 14, 17, 18	3, 4, 6, 7
9	Andmebaasi tabel sularahahoidlate info jaoks	6, 10, 11, 12, 14	8
10*	Pearaamatu struktuuri seadistamine sularaha liikumiste kajastamiseks	12	9, 11
11*	Sularahahoidlate info nimekirja vaade koos otsingufiltritega	10	9, 12, 13
12*	Sularahahoidlate detailne vaade koos kontode loomisega pearaamatusse	11	9, 10
13*	Eraldi alammenüü lisamine rakendusse sularaha puudutava info jaoks	7, 11	
14	Sularaha korje ja tarne töövood	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27	2, 3, 4, 5, 6, 7, 8, 9, 15

Nr	Kategooria	Sõltuvad nõuded	Sõltub nõuetest
15	Uute sularaha kannete koodide loomine ja olemasolevasse rakendusse seadistamine	14, 22, 23, 24, 25, 26	4
16	Sularaha tehingute riskifiltri lisamine maksete riskifiltri kasutajaliidesesse	17, 18, 19	4, 14
17*	Sularaha tehingute riskifiltri funktsionaalsuse loomine	18, 19	3, 4, 8, 14, 16
18	Sularaha riskifiltri info lisamine sularaha sõnumite nimekirja vaatesse ja detailide vaatesse.	19	3, 4, 8, 14, 16, 17
19*	Teavitussõnumi saatmine riskifiltri märke tekkimise puhul		3, 4, 14
20	Saabumise aja kontroll ning väljaspool lubatud aega saabunud sõnumite tagasilükkamine		1, 3, 4, 5, 14
21	Kuupõhise sularahatehingute hooldustasu seadistus		2, 4, 14
22	Sularaha tehingu summa põhise teenustasu seadistus		4, 14, 15
23	Sularaha tehingute info seadistamine prinditavale lepingule		14, 15
24	Sularaha tehingute info seadistamine F607 Ila raportisse		14, 15
25	Lisada Forma-F-3 regulatsioonile vastav raport		14, 15
26	Sularaha tehingute info lisamine aruandeid koostavasse päringumootorisse		14, 15
27*	Aadressiväljade lisamine saabuva sõnumi töödeldava info hulka		3, 4, 5, 6, 14

*Nõue lisandus arendustööde käigus.

Lisa 4 – Muudatusnõuete omavahelised sõltuvused maatriksina

Nr	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1				x																x							
2						x							x								x						
3			x				x	x					x				x	x	x	x							x
4				x	x	x	x						x	x	x	x	x	x	x	x	x	x					x
5						x								x							x						x
6							x	x						x													x
7								x						x													
8							x		x					x			x	x									
9						x				x	x	x		x													
10												x															
11											x																
12												x															
13							x					x															
14																x	x	x	x	x	x	x	x	x	x	x	x
15															x							x	x	x	x	x	
16																	x	x	x								
17																		x	x								
18																			x								
19																											
20																											
21																											
22																											
23																											
24																											
25																											
26																											
27																											

Muudatusnõuete maatriksit saab lugeda kahel viisil. Piki ridu suunaga vasakult paremale lugedes on näha, millistele nõuetele selle rea pealkirja numbriga nõue sisendi annab (nõude numbrid on ridade ja veergude numbriteks). Ülalt alla piki veerge lugedes on näha, millistelt nõuetelt selle veeru pealkirja numbriga nõue sisendit vajab. Näiteks nõue number neli vajab sisendit nõuetelt üks ja kolm.