TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Emil Fenenko 131030IAPB

AVATUD PANGANDUSE VARUMEHHANISMI SÜSTEEMI ARENDUS

Bakalaureusetöö

Juhendaja: Tarvo Treier
MSc

Tallinn 2020

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: Emil Fenenko

25.05.2020

Annotatsioon

Antud bakalaureusetöös käsitletakse pangale arendatud varumehhanismi lahendust avatud panganduse raames. Töös tutvustatakse lahenduse seadusandlusest ning tehnilistest piirangutest tulenevaid nõudeid ning selgitatakse rakenduse arendamisel tehtud valikuid nõuete kontekstis.

Töö tulemus on rakendusliides, mida kolmanda osapoole ettevõtjad saavad integreerida oma rakendustesse, mille tulemusena saavad pangakonto omanikud kasutada ettevõtte finantsteenuseid pangaväliselt. Liidest on praeguseks enda süsteemidesse integreerinud kaks finantsteenust pakkuvat ettevõtet, mille tulemusel sooritatakse antud liidese abil sadu makseid päevas.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 35 leheküljel, 5 peatükki, 1 joonist, 2 tabelit.

Abstract

The Development of Open Banking Fallback System

In order to adapt to every changing financial services industry, wherein the use of digital payment services is growing every year, the European Union adopted the revised Payment Service Directive. In essence, banks operating in the European Economic Area are mandated to open their banking services to third party providers, in the purisuit of more competition within the financial services marketplace.

The aim of this thesis is create a Fallback Mechanism, which third party providers can depend on to access the necessary financial services mandated by regulation. As the solution uses the underlying internetbanks to implement the services, the thesis covers all the technical bottlenecks associated with this solution, providing necessary context and reasoning to the architecture of the system.

The thesis covers the set of functional and nonfunctional requirements set out to the system provide a standard the application needs to adhere to. Additionally an overview of testing proceedures is given to verify the systems compliance with it’s requirements.

The final product servers as a microservice between third party providers and the customers data. The resulting interface provides all the tools necessary for third party providers to integrate it within their own systems. The application has demonstrated it’s reliability to provide the necessary services, while meeting the regulatory mandates set out by revised Payment Services Directive.

The thesis is in Estonian and contains 35 pages of text, 5 chapters, 1 figure, 2 tables.

Lühendite ja mõistete sõnastik

PSD2	*Revised Payment Services Directive*, muudetud Makseteenuste Direktiiv
TPP	*Third Party Provider*, kolmanda osapoole makseteenuste pakkuja
PSU	*Payment Services User*, makseteenuste kasutaja
REST	*Representational state transfer*, arhitektuuristiil, mida kasutatakse hajutatud süsteemides
API	*Application Programming Interface*, rakendusliides
OCSP	*Online Certificate Status Protocol*, protokoll, mida kasutatakse digitaalse sertifikaadi kehtetuks tunnistamise staatuse saamiseks
AISP	*Account Information Service Provider*, kontoteabe teenuse pakkuja
PISP	*Payment Initiation Service Provider*, makse algatamise teenuse pakkuja
PIISP	*Payment Instrument Issuer Service Provider*, kaardipõhiseid makseinstrumente väljastav makseteenuse pakkuja
ASPSP	*Account Servicing Payment Service Provider*, kontot haldav makseteenuse pakkuja
eIDAS	*electronic IDentification, Authentication and trust Services*, e-identimise ja e-tehingute jaoks vajalike usaldusteenuste kohta siseturul
QWAC	*Qualified Website Authentication Certificate*, kvalifitseeritud veebilehe autentimissertifikaat
QTSP	*Qualified Trust Service Provider*, reguleeritud organ, mis vastutab sertifikaatide väljastamise ja haldamise eest
HTTP	*Hypertext Transfer Protocol*, Hüperteksti edastusprotokoll
IBAN	*International Bank Account Number*, rahvusvaheline pangakonto number
SEPA	*Single Euro Payments Area*, Euroopa Liidu ühtne euromaksete piirkond
KPI	*Key Performance Indicator*, võtmenäitaja
HTML	*Hypertext Markup Language*, hüperteksti märgistuskeel

GDPR	*General Data Protection Regulation*, Isikuandmete kaitse üldmäärus
FinTech	*Financial Technology*, finantstehnoloogia

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Alates oktoobrist 2015 on Euroopa Liidus tegutsevatel pankadele seatud nõue avada oma siseseid panganduse andmeid ning teenuseid kolmanda osapoole finantsteenuseid pakkuvatele ettevõtetele, tagamaks konkurentsivõimekust finantsteenuste turul Euroopa Majanduspiirkonna siseselt.

Lõputöö eesmärgiks on arendada liides, mida kolmanda osapoole finantsteenuste pakkujad saavad kasutada makse ning kontoteabe teenusteks ühes Euroopa Liidus tegutsevas pangas.

Püstitatud eesmärgi täitmiseks peab tutvuma liidest puudutava regulatsiooniga ning pangaspetsiifiliste allsüsteemidega. See võimaldab rakenduse spetsifikatsioonide kirjeldamist, mille alusel koostatakse süsteemi arhitektuur ning tehakse tehnoloogilised valikud. Kontrollimaks, et lõpptoode täidab kavandatud eesmärki, kirjeldatakse töös ka ettenähtud testimisprotseduure.

Arendatava süsteemi regulatsioonist tulenevat vajadust ning sätestatuid määrusi tutvustab lähemalt teine peatükk, käsitledes seejuures võimalike lahendusi antud panga kontekstis.

Järgnev peatükk käsitleb sügavuti rakendusele kehtestatuid funktsionaalseid ning mittefunktsionaalseid nõudeid ning nendest tuleneva süsteemiloogika analüüsi.

Neljas peatükk fokuseerib rakenduse tehnilise poolele, tutvustades valitud tehnoloogiad ning süsteemi arhitektuuri. Lisaks antakse ülevaade rakenduse testimisest.

Töös käsitletud rakendus on arendatud finantslahendusi pakkuvas tarkvaraettevõttes. Projekti arenduses osales suures mahus üks projektijuht ning kaks arendajat, kusjuures töö autor on vastutav üle poole kirjutatud lähtekoodi eest. Lisaks osales autor projekti arhitektuuri välja töötamisel, oli pidevas kontaktis kliendiga tagamaks ärinõuete täitmist ning on vastutav süsteemi ülalpidamise eest.

2 Taust

Peatükis käsitletakse süsteemi vajalikkust praeguses Euroopa Majanduspiirkonna panganduse maastikul. Samuti tutvustatakse ning põhjendatakse antud töö varumehhanismi lahenduse olemust, vaadeldes seda läbi võimalike lahenduste konteksti.

2.1 Makseteenuste direktiiv

Oktoobris, aastal 2015, võttis Euroopa Liit vastu muudetud Maksuteenuste direktiivi (edaspidi PSD2 ehk *revised Payment Services Directive*), kohustades makseteenuse kontode haldajaid avama siseseid makseteenuseid riiklike pädevate asutuste poolt valideeritud kolmanda osapoole makseteenuste pakkujatele (edaspidi TPP ehk *Third Party Provider*).

PSD2 peamisteks eesmärkideks on:

- Aidata kaasa integreeritumale ja tõhusamale Euroopa makseturule
- Ühtlustada makseteenuste pakkujate konkurentsitingimusi, kaasates makseteenuseid pakkuvaid ettevõtteid
- Muuta maksed turvalisemaks
- Parandada Euroopa tarbijate ja ettevõtete kaitset [1]

TPP-d, on PSD2 alusel registreeritud, litsentseeritud ning reguleeritud EL-i tasandil entiteedid, kellele on tagatud juurdepääs makseteenuste kasutajate (edaspidi PSU ehk *Payment Services User*) volitusel oma maksekontodele. Selle tulemusel, on TPP-del võimalik koguda enda makseteenust kasutatavate PSU-de kontoteavet ning kasutada makseteenuseid.

Eelnevalt kirjeldatu võimaldamiseks, on kontot haldavad makseteenuse pakkujad kohustatud pakkuma spetsiaalset juurdepääsuliidest, mis võimaldab turvalist teabevahetust erinevaid teenuseid kasutatavate TPP-dega [2, lõik 20]. Tagamaks, et

spetsiaalset juurdepääsuliidest kasutavad TPP-d saavad jätkata oma teenuste osutamist juhul, kui liides ei ole kättesaadav või ei tööta nõuetekohaselt, tuleb makseteenuse kontode haldajatel ette näha varumehhanism. Varumehhanism võimaldab TPP-del kasutada kontot haldavate makseteenuse pakkujate kasutajapoolset liidest, mida kasutatakse omaenese PSU-de identifitseerimiseks ja teabe vahetamiseks. [2, lõik 24]

Kuigi spetsiaalne liides ning varumehhanism erinevad suuresti realisatsiooni kohapealt, kehtivad nendele ühine äriloogiline ning tehniline standard. Standardi poolt sätestatud nõudeid on palju, kuid olulisimateks punktideks on:

- Tugeva autentimise meetodi(te) (SCA) kasutus – liides rakendab meetmeid makseteenuste kasutaja autentimiseks, mis põhinevad kahe või enama turvaelemendi olemasolul, mis on liigitatud kategooriatesse järgmiselt: teadmine (miski, mida teab üksnes kasutaja), omamine (miski, mida omab üksnes kasutaja) ja olemus (miski, mis on kasutajale omane) [2, lõik 6]

- TPP tuvastamine – liides tagab turvalist info- ja suhtluskanalit, seejuures tuvastades ning volitades kolmanda osapoole makseteenuse pakkujaid

- Teenuste pakkumine – liides realiseerib töökindlaid makse- ning kontoteabe teenuseid

Käesolevas töös käsitletakse varumehhanismi süsteemi arendust ühele Euroopa Liidus opereerivale pangale, mille varumehhanismi definitsioonis kirjeldatud kasutajapoolseks liideseks on antud kontot haldava makseteenuse pakkuja internetipangad.

2.2 Võimalikud lahendused

Suurimaks murekohaks internetipankade TPP-dele kasutamiseks avamisel, on internetipankadesse sisseehitatud funktsionaalsus ning juurdepääs informatsioonile, millele TPP ei oma vastavat volitust. Selle alla kuulub näiteks laene, liisinguid ja pangakaarte puudutav funktsionaalsus ning kaardi omanike tundlik informatsioon, näiteks PAN numbrid.

Selleks, et täita eelnevalt kirjeldatud PSD2 rakendatud äriloogika ja tehnilise standardi olulisemaid nõudmisi, on varumehhanismi lahenduseks järgnevalt lahti seletatud kaks võimaliku varianti.

2.2.1 Ümberehitus

Modifitseerida internetipankasid selliselt, et internetipangas oleks võimalik tuvastada TPP-d ning eduka tuvastuse järel piirata informatsiooni ning funktsionaalsuse kättesaadavust, seejuures eemaldades kõik mitte SCA nõudmisele vastavad autentimise meetodid.

Arendust vajavateks komponentideks oleks kolm erinevat internetipanka, millest igale tuleks lisada TPP autentimine ja autoriseerimine, funktsionaalsuse piiramine, informatsiooni valikuline esitamine ja ainult SCA nõuetele vastavate PSU autentimise meetodite lubamine. Teenuste osutamiseks arendustööd pole vaja, kuna kõik vajalikud teenused on internetipankades juba olemas (nt makseteenus).

2.2.2 Vahendaja

Arendada internetipanga peale nn. vahendaja, mille ülesandeks on TPP tuvastamine, informatsiooni ja funktsionaalsuse filtreerimine ning teenuste rakendamine.

Antud lahenduse puhul on tegemist täiesti eraldiseisva komponendiga, mille puhul lisaks esimeses variandis kirjeldatud funktsionaalsuse arendamisele, peab juurde arendama teenuste osutamiseks vajaliku funktsionaalsust.

2.2.3 Lahenduse valimine

Peamiseks parameetriks lahenduse valimisel oli lahenduse arendamisele kuluv aeg.

Esimese variandi lahenduse teostuseks vajalik töö on ebaselge, kuna kõigi kolme internetipanga puhul on tegemist n.ö. päritud varaga ning antud lahenduse rakendamiseks oleks vaja teha muutusi internetipankade arhitektuuris. Kuna kasutusel olevad internetpangad on täiesti eraldiseisvad süsteemid, siis juba analüüsile kuluv aeg on liiga mahukas ja ebaotstarbekas.

Teise variandi puhul, kuna tegemist on sisuliselt nullist arendatud rakendusega. See võimaldab arendada süsteemi selliselt, et ühiselt kasutatavad funktsionaalsused on jagatud erinevate internetipankade süsteemide poolt. Seega jääks internetipanga

spetsiifiliseks vaid teenuste realiseerimine. Sellest tulenevalt, pidas klient otstarbekamaks edasi minna teise variandiga.

2.3 Valitud lahendus

Sisuliselt on lahendus rakendusliides, mis on ehitatud eksisteerivate internetipankade peale. Teenuste osutamiseks internetipankades võetakse kasutusele nn. veebiämblik, millega automatiseeritakse vajalikud kasutajavood.

Antud lähenemine on efektiivne, kuna PSD2 spetsifikatsioonist tulenevad nõuded on kohaldatavad kõikidele internetipankadele, mis võimaldab ühiselt kasutatavate komponentide realiseerimist. Ainsateks internetpanga spetsiifilisteks komponentideks jäävad veebiämblikud, tulenevalt internetipankade kasutajaliidestes esinevatest erinevustest.

3 Rakenduse analüüs

Eelnevalt kirjeldatud PSD2 nõudmiste alusel, peab rakendus olema võimeline pakkuma töökindlat teenust teenindatavatele TPP-dele. Seejuures kuulub rakenduse vastutuse alla ka teabevahetuse turvalisus. Käesolevas peatükis käsitletakse rakendusele esitatud nõudeid ning rakenduse kitsaskohtadest tulenevaid väljakutseid.

3.1 Funktsionaalsed nõuded

Varumehhanismi ärinõuded tulenevad suuresti eelnevalt kirjeldatud PSD2-e poolt sätestatud määrustega. Tagamaks ärinõuetest kinnipidamist, on järgnevalt tutvustatud funktsionaalsed nõudmised seatud selliselt, et vajalikud funktsionaalsused on süsteemis olemas, seejuures silmas pidades valitud lahenduse olemust ning sellega kaasnevaid piiranguid.

Varumehhanism peab olema võimeline pakkuma teenuseid kõikides olemasolevates internetpankades ning olema võimeline toetama ka tulevasi kanaleid, kui peaks selleks vajadus tekkima.

Süsteemisiseselt, tagamaks turvalisust teabevahetuse kanalis, peab eksisteerima protsess TPP-de tuvastamiseks ning autoriseerimiseks. Vahend, mida kasutatakse TPP-de identifitseerimise võimaldamiseks on PSD2 tehniliste standardite alusel kirjeldatud sertifikaat. Tagamaks terviklikku TPP valideerimise protsessi, peab sertifikaat läbima järgnevaid samme:

- TPP sertifikaadi formaalne valideerimine
- TPP OCSP kontroll
- TPP sertifikaatide valge nimekiri
- TPP volitamine
    1. AISP roll

2. PISP roll
3. PIISP roll
4. ASPSP roll

Suurimaks piiranguks antud süsteemis on internetpankade teenuste kättesaadavus. Nimelt on kõikide internetpankades olevate teenuste jaoks vajalik aktiivse seansi olemasolu, mille alusel toimub panga ja PSU vaheline kommunikatsioon. Rakendus, põhinedes internetipankadele, peab samuti võimaldama seansi loomise, haldamise ja katkestamise funktsionaalsust. Tagamaks turvalist teabevahetuse kanali loomist, peab rakendus võimaldama ainult SCA nõetele vastavate vahendite kasutust autentimise (seega seansi loomise) protsessis.

Internetipankades on PSU-del võimalus pärast autentimise protsessi valida, mis klienti nad soovivad esindada. Antud töö kontekstis, klient on entiteet, millega on võimalik kasutada rakenduse poolt realiseeritud teenuseid. Internetipanga siseselt jagunevad klientide tüübid kaheks:

- Eraklient
- Äriklient

Rakendus peab toetama mõlemat liiki klienti ning rakendama seejuures vajaliku funktsionaalsust esindatava kliendi valimiseks.

Kõik eelnevalt kaetud funktsionaalsed nõuded on eeldused PSD2 spetsifikatsioonis kirjeldatud teenuste kasutamisele. Antud teenuseid, mida rakendus realiseerima peab on kirjeldatud PSD2 tehnilise ja äriloogilise standardi poolt, nimelt makseteenus, kontoteabe teenus ning kaardipõhiste makseinstrumentide väljastamise teenus.

Kõikidest funktsionaalsetest nõuetest tulenevalt peavad rakenduses olema realiseeritud järgnevad REST liidesed:

1. PSU seansi loomine ja katkestamine
2. PSU poolt esindatava klientide identifitseerimine ning valimine
3. Kontoteabe teenus

4. Makseteenus

5. Kaardipõhiste makseinstrumentide väljastamisele suunatud teenus

3.2 TPP Tuvastamine

TPP tuvastamiseks on PSD2 tehnilise standardi alusel ette nähtud igale litsentseeritud TPP-le *eIDAS* regulatsiooni põhimõtetel väljastatud QWAC sertifikaat (*Qualified Website Authentication Certificate)*. Sisuliselt on tegemist SSL sertifikaadiga, millel on vaja privaatvõtit (eraldi fail, mida kasutatakse andmete krüpteerimiseks ning dekrüpteerimiseks), et seda kasutada.

Sertifikaati väljastab riiklikul tasandil opereeriv organ (QTSP ehk *Qualified Trust Service Provider*). Sertifikaadis on krüpteeritud andmed sertifikaadi, TPP ning QTSP kohta. Lisaks sisaldab sertifikaat PSD2-le spetsiifilist informatsiooni. Asjakohaseks teabeks TPP valideerimisel on järgnev:

- Väljastaja identifikaator
- TPP identifikaator
- Valideerimise võti
- Kehtivusaeg
- TPP rollid

Selleks, et süsteemile ligi saada, peab TPP lisama enda päringu päisesse kehtiva sertifikaadi ning privaatvõtme.

Rakendus peab olema võimeline tegema sertifikaadile nn. formaalse valideerimise. Siinkohal kontrollitakse, kas päises olev sertifikaat on õiges formaadis, on süntaktiliselt korrektne, sisaldab vajalike andmeid ning on valideeritud ja kehtiv.

Lisaks kindlustamaks sertifikaadi valiidsust kontrollib rakendus, ega sertifikaati pole tühistatud, mida tehakse OCSP (*Online Certificate Status Protocol*) abil. OSCP kontroll seisneb selles, et QTSP, kes on sertifikaadi väljastanud, hoiab nimekirja tühistatud sertifikaatidest, mille vastu võrreldakse päises olevat sertifikaati.

TPP-le litsentsi opereerimiseks ELi makseturul ning selleks vajaliku sertifikaati annavad välja erinevad organid. See tähendab, et juhul kui TPP-lt on eemaldatud opereerimise litsents, ei tühistata sellega kaasnev sertifikaat. Sellest tulenevalt on kontot haldavad makseteenuse pakkujad kohustatud haldama valget nimekirja süsteemi kasutatavatest TPP-dest. Süsteem peab kontrollima iga päringut tegeva TPP identiteeti ning kindlustama tema olemasolu valges nimekirjas.

Sertifikaadis on kirjeldatud TPP roll. Võimalikke rolle on kokku neli:

- AISP (*Account Information Service Provider*) – kontoteabe teenuse pakkuja, ehk TPP-l on volitatud juurdepääs PSU kontoteabele
- PISP (*Payment Initiation Service Provider)* – makse algatamise teenuse pakkuja, ehk TPP on volitatud sooritama makseid PSU maksekontodega
- PIISP (*Payment Instrument Issuer Service Providers*) – kaardipõhiseid makseinstrumente väljastav makseteenuse pakkuja, ehk TPP on volitatud saama vajaliku kontoteavet kaardipõhiste makseinstrumentide väljastamiseks
- ASPSP (*Account Servicing Payment Service Providers*) – kontot haldav makseteenuse pakkuja, ehk TPP on PSD2 raames volitatud kasutama kõiki eelnevalt kirjeldatud rollidele pakutud teenuseid.

Süsteemis peab TPP-l olema juurdepääs ainult teenustele, mida kirjeldab temale vastav roll.

3.3 Seansi haldus

Kõik teenused, mida kirjeldab PSD2 spetsifikatsioon, eeldab varumehhanismi lahenduses aktiivset internetipanga seanssi. Süsteemil peab olema realiseeritud seansi halduri nimeline komponent, mille põhiprotsessideks on seansi alustamine, jälgimine ja lõpetamine. Seansi haldamiseks peab süsteem realiseerima seansi alustamise ja lõpetamise liidest.

Seansi halduril on ülevaade süsteemis aktiivsetest ja mitteaktiivsetest seanssidest. Igal seansil on oma unikaalne identifikaator ning seansi staatus. Lisaks on iga seansiga seotud internetpangaga teabevahetus kanali instants. Kõik süsteemile suunatud päringud läbivad

seansi halduri, mille ülesanne on suunata päringud edasi õigele kanali instantsile või tagastada veateade mitteaktiivse seansi puhul. Kuna süsteemi seansid on tihedalt seotud internetpankade seanssidega, siis peab seansi halduril olema ülevaade internetpanga seansi staatusest.

Seanss kasutab identifikaatorina HTTP küpsiseid, mida haldur talletab ning kasutab päringute suunamise eesmärgil. Seansi alustamiseks tuleb teha vastav päring süsteemile. Päringu sisenditeks on:

- BIC (*Bank Identification Code*), mille alusel avatakse BIC-ile vastav internetipanga suhtluskanal
- SCA meetodi parameetrid, mida kasutatakse vastavas internetpangas PSU seansi alustamiseks.

Eeldusel, et internetipangas on PSU autentimise protsess edukalt läbitud, määratakse seanss aktiivseks ning kõik järgnevad päringud (vastava HTTP küpsisega) suunatakse seansiga seotud internetipanga ning süsteemi vahelise teabevahetus kanali poole.

Lisaks seansi alustamise päringule peab süsteemis olema seansi sulgemiseks vastav päring. Päringu saamisel, suunab seansi haldur päringu läbi teabevahetus kanali internetipangale, milles sulgetakse PSU ja internetipanga vaheline seanss. Seejärel liigitatakse seanss süsteemisiseselt mitteaktiivseks ning järgnevatel päringutel vastava HTTP küpsisega (v.a. uue seansi alustamise päringu puhul) tagastatakse veateade.

Teabevahetus kanalite avatuna hoidmine on süsteemis päris kallis protsess, seega peab seansi haldur olema võimeline kauakestvaid ning mitte kasutusel olevaid seansse liigitama mitteaktiivseteks ning sulgema seotud teabevahetuse kanaleid. Seansi haldur peab kontrollima perioodiliselt kõiki aktiivseid seansse ning eeldusel, et nendega pole toimunud teabevahetust konstandiga määratud perioodi jooksul, määrama neid mitteaktiivseteks.

Selleks, et TPP-del oleks arusaam nende poolt alustatud ning süsteemi poolt mitteaktiivseks liigitatud seanssidest, peab süsteem talletama mitteaktiivseid seansse. Seega, kui süsteem saab päringu, mis on suunatud süsteemselt mitteaktiivseks liigitatud seansile, on süsteem võimeline tagastama arusaadavat veateadet. Võimalikud seansside mitteaktiivseks liigitamse põhjusteks on:

- SESSION_NOT_OPENED – süsteem pole saanud päringu päises oleva HTTP küpsisega seansi alustamise päringut
- SESSION_CLOSED – süsteemile on tehtud päring antus seanss sulgeda
- SESSION_EXPIRED – süsteem on määranud seanssi mitteaktiivseks, kuna seotud teabevahetuse kanalit pole konstandiga määratud ajaperioodil kasutatud
- BROKEN_SESSION – süsteemis on toimunud viga, mis nõuab uue seansi alustamist.

3.4 PSU poolt esindatavate klientide valimine

Esindatava kliendi valimise protsess oluline vahepunkt seansi algatamise ning PSD2 poolt spetsifitseeritud teenuste kasutamise vahel. Süsteem peab realiseerima vajaliku funktsionaalsust, et PSU saaks osutada TPP-le klienti millega soovib kasutada kontoteabe ning makseteenuseid.

Seansi eduka alustamise protsessi käigus, peab süsteem tagastama nimekirja PSU klientidest, keda PSU esindada saab. Kliendi valimine toimub vastava päringuga, mille sisendiks on kliendi identifikaator. Eeldusel, et kliendi valimise protsess on edukalt läbitud, on kõik järgnevad seansi halduri poolt saadud päringutele vastavad toimingud sooritatud internetipangas valitud kliendiga.

Aktiivse seansi vältel peab olema võimalik süsteemisiseselt klienti valida mistahes arv kordi, et tagada süsteemi kasutamise käepärasust.

3.5 PSD2 spetsifikatsiooni realiseerivad liidesed

Selles peatükis kirjeldatakse PSD2 spetsifikatsioonist tulenevate teenuste lahendused varumehhanismi kontekstis.

3.5.1 Kontoteabe liides

PSU kontoteave on andmete kogum, mille abil on AISP-del võimalik läbida oma teenuste osutamiseks vajalike protsesse. Teenuste hulka, mida AISP-d pakkuda võivad on näiteks PSU laenuvõimekuse arvutamine. Kontoteabe alla kuuluvad PSU-le kuuluvad kontod,

kontode saldod ning kontoväljavõtted. Kõikidele vajalike andmetele juurdepääs on internetipankade poolt kasutatud teenustega realiseeritud ning süsteem peab võimaldama seda funktsionaalsust läbi rakendusliidese.

Selleks, et hoida teenuseid varumehhanismis minimaalsena, peab süsteemis realiseeritud olema kaks API-t:

- PSU kontode päring
- PSU kontode konto väljavõtte päring

Kontode päringuga tagastab süsteem kontode nimekirja koos AISP-dele asjakohase informatsiooniga. Asjakohase informatsiooni alla kuulub konto IBAN (*International Bank Account Number*), saldo, vabajääk jne. Selle alla ei kuulu kontoga seotud kaardi informatsioon, konto krediidimäär, konto omaniku nimi jne.

Kontode väljavõtte saamiseks internetipangas peavad konto IBAN ning väljavõtte periood olema spetsifitseeritud. Selle tulemusena on süsteemile suunatud konto väljavõtte päringu sisenditeks konto IBAN, perioodi algus- ning lõppkuupäev. Süsteem peab olema võimeline tegema päringut internetipanga siseselt ning lugema ja tagastama asjakohaseid andmeid.

3.5.2 Makseteenuste liides

PSD2 nõuete kohaselt, peab süsteemis olema ettenähtud kõik vajalikud ressursid, et PISP-id saaksid läbi rakenduse algatada makseid. Internetipankades on võimalik vormistada kolme liiki makseid:

- Pangasisene makse – makse ühe panga siseselt
- SEPA makse – Euro makse Euroopa Majanduspiirkonna siseselt
- Välismakse – makse väljaspool Euroopa Majanduspiirkonda

Süsteem peab toetama pangasiseste ning SEPA maksete vormistamist ning algatamist. Süsteemil puudub juurdepääs vajalikele andmetele, et kindlaks määrata maksetüüpi sisendite alusel, kuid internetipangal on vajalik võimekus olemas. Seega pangasiseseid

makseid tuleb samuti vormistada SEPA maksena ning maksetüübi määramine toimub internetipanga siseselt. Välismaksed on antud projekti skoobist väljas.

Maksete algatamine internetpankades on kahest sammust koosnev protsess: maksedokumendi koostamine ja selle allkirjastamine. Süsteemis makse algatamise vooks vajalikud API-d on:

- Makse algatamise päring
- Maksedokumendi allkirjastamise staatuse päring

Makse algatamise päringu sisenditeks on SEPA makse vormistamiseks vajalikud andmed (kreeditori ja deebitori IBAN, deebitori nimi, summa ning selgitus). Süsteem sisestab andmed internetipanka ning eeldusel, et vajalikud kontrollid on läbitud (nt. kontol on piisavalt vabu vahendeid) väljundina tagastama SCA meetodi kontrollkoodi, mille alusel allkirjastatakse koostatud dokument.

Tagamaks, et PISP oleks teadlik allkirjastamise staatusest, peab süsteem realiseerima vastavat API-t. Makse allkirjastamisel on kolm võimaliku staatust:

- Ootel – allkirjastamise protsess on alustatud ning PSU tegevuse ootel
- Korras – maksedokument on edukalt allkirjastatud (lõplik staatus)
- Ebaõnnestunud – maksedokument pole allkirjastatud (lõplik staatus)

Süsteemis ettenähtud kasutajavoog maksedokumendi allkirjastamisel on staatuse pärimine, kuni jõutakse lõpliku staatuseni.

Peale makse algatamist toimub pangasiseselt maksedokumendi protsessimine, mille tulemusel teostatakse või lükatakse makse tagasi. Selleks, et PISP saaks veenduda, et algatatud makse on panga poolt protsessitud, peab varumehhanismis olema API maksestaatuse saamiseks.

Kirjeldatud funktsionaalsus on internetipangas realiseeritud. Algatades makset genereeritakse internetipanga poolt maksele maksenumber, mis on SERIAL tüüpi, seega vaikimisi unikaalne. Süsteem realiseerib teenustSü tagastades makse identifikaatori

makse eduka algatamisel ning TPP-l on võimalik teha päringuid vastu maksestaatuse API-t.

Võimalikke maksestaatuseid on süsteemis kokku neli:

- SETTLED – makse on töödeldud
- WAITING FOR SETTLEMENT – makse on töötlemisel
- REJECTED – makse on tagasi lükatud
- UNCONFIRMED – makse dokument pole allkirjastatud

3.5.3 Kaardipõhiste makseinstrumentide väljastamisele suunatud teenus

PIISP, ehk makseinstrumente väljastav teenusepakkujale peab süsteemis olema suunatud üks API, milleks on PSU rahaliste vahendite olemasolu kontroll. Antud teenus leiab kasutust näiteks kaardimaksetel. PSU maksab kaardiga valitud kauba või teenuse eest ning PIISP saadab päringu süsteemi pihta, kontrollimaks kas PSU kontol on piisavalt vabu vahendeid, et tehing lõpetada.

Rahaliste vahendite olemasolu kontrolli sisenditeks on PSU konto IBAN, valuuta ning summa, mille vastu võrreldakse kontol olevaid vabu vahendeid. Antud API väljundiks on tõeväärtus.

Reaalsuses on antud teenuse kasutamine PIISP-de poolt ebaotstarbekas, kuna teenusele ligipääsu saamiseks on vaja luua internetipanga seanss, mis on kulukas protsess.

3.6 Olekukoodid ja veateated

Kõik ühendused süsteemiga kasutavad HTTP-d (*Hypertext Transfer Protocol*). Tagamaks, et TPP-d saavad varumehhanismi API-le tugineva rakenduse arendada, on rakenduses defineeritud kogum HTTP olekukoodidest. Olekukoodide ning nendega kaasas käivate sõnumite või veateadete abil saab TPP oma rakenduse raames kontrollida kasutajavooge. Süsteemis kasutusel olevad üldised HTTP koodid on kirjeldatud järgnevas tabelis.

Tabel 1 Süsteemis defineeritud HTTP olekukoodid

HTTP olekukood	**Sõnumi kood**	**Kirjeldus**
200	OK	Päring on õnnestunud
201	Accepted	Olekukood, mis viitab autentimise ning makse allkirjastamise protsessi alustamisele.
400	Bad request	Esines valideerimise viga. Olekukood katab valesti vormindatud süntaksi päringus ning vigaseid andmeid päringu sisendis .
401	Unauthorized	Tagastatud juhul failitees viidatud ressursi pole süsteemis defineeritud või TPP-l puudub vajalik volitus
403	Forbidden	Tagastatud kui päringu sisendis viidatud ressursile puudub PSU-l vastav volitus
405	Method not allowed	Tagastatakse kui HTTP päringu meetod ei ole toetatud etteantud API-le.
500	Internal server error	Süsteemisisene vea teke.

Sõnumitele ja veateadetele, mida rakendus päringu väljundis tagastab, on nõue, et need peavad olema eelnevalt defineeritud ning olema tähendusrikkad. Selline standard võimaldab TPP-l katta kõiki võimalike stsenaariumeid varumehhanismi integreerivas rakenduses.

3.7 Mittefunktsionaalsed nõuded

Süsteemile seatavad mittefunktsionaalsed nõuded on paika pandud, et tagada ja hinnata süsteemi jõudlust ning käideldavust. Lisaks kirjeldavad mittefunktsionaalsed nõuded vajalike komponente süsteemi toimingute hindamiseks.

3.7.1 Rakenduse jõudlus

Peamiseks nõudeks on 100 aktiivse seansi paralleelselt käsitlemine. See arv on proportsionaalne koormusega, mida kõik süsteemi kasutavad TPP-d genereerivad tavakasutusega. Kuna süsteem on sisuliselt varumehhanism, siis piirang on seatud eeldusega, et vähesed TPP-d investeerivad aega, et realiseerida omapoolset rakendust antud API-le.

Lisaks on seatud süsteemile nõue, et süsteem ei tohi lisada rohkem kui 300ms internetipanga reaktsiooniajale. Rakendus kasutab internetipangaga suhtlemiseks kasutajaliidest ning sellest tulenevalt on internetipangale päringute tegemine kulukas protsess. Selleks, et hoida kommunikatsiooni süsteemi ja internetipanga vahel minimaalsena peab süsteem rakendama kõiki võimalike meetmeid.

Üheks meetmeks on sisendi valideerimine. Kuigi internetipankades on vastav funktsionaalsus olemas, on sellele toetumine kulukas protsess ning seda tuleks hoida minimaalsena.

Samuti peab rakendus hoidma seansi siseselt kogutud andmeid vahemälus. See tagab, et korduvad päringud kindlale API-le ei pöördu alati teabevahetuse kanali poole. Seansi vältel toimub perioodiliselt vahemälu puhastus, et tagada andmete värskust.

3.7.2 Rakenduse toimingute jälgimine

Süsteemis peab paigas olema vajalik infrastruktuur, mis võimaldab rakendust jälgida ning vigade ilmnemisel pakkuda vajalike tööriistu tarkvara silumiseks. Lisaks peab rakendus võimaldama statistikate loomist rakenduse jõudluse kohta.

Rakenduse jälgimiseks peab süsteem realiseerima avatud maailmale suletud liidest. Antud liidese eesmärk on raporteerida süsteemi jõudlust puudutavaid KPI-sid (*Key Performance Indicator*), näiteks süsteemi poolt kasutatud ressursid, aktiivsete seansside arv, vigade statistika jne.

Koodi silumiseks on rakendusele seatud nõue pidada arvet oma tegevustest logide näol. Igas kirjes peab sisalduma hulk parameetreid, mille abil on võimalik identifitseerida vea tekke asjaolusid.

Süsteemi poolt realiseeritud teenused on sõltuvad välistest süsteemidest, nimelt internetipankadest. Selle tulemusena peab arendamise käigus olema võimalikult palju testandmeid, et ennetada vigade tekke kohti. Lisaks, peab süsteem võimaldama tarkvara silumise jaoks kuvatõmmiste ning internetipanga kasutajaliidesest tõmmatud HTML failide salvestamist vea tekkel. Nende abil, on võimalik identifitseerida ning ajapikku katta kõik võimalikud vea tekke stsenaariumid.

4 Rakenduse realisatsioon

Eelnevalt käsitletud nõuete alusel kirjeldatakse peatükis arhitektuuri ning tehnoloogilisi valikuid. Samuti tutvustatakse rakenduse testimise meetmeid, tagamaks rakenduse kasutuskõlbulikust.

4.1 Tehnoloogiad

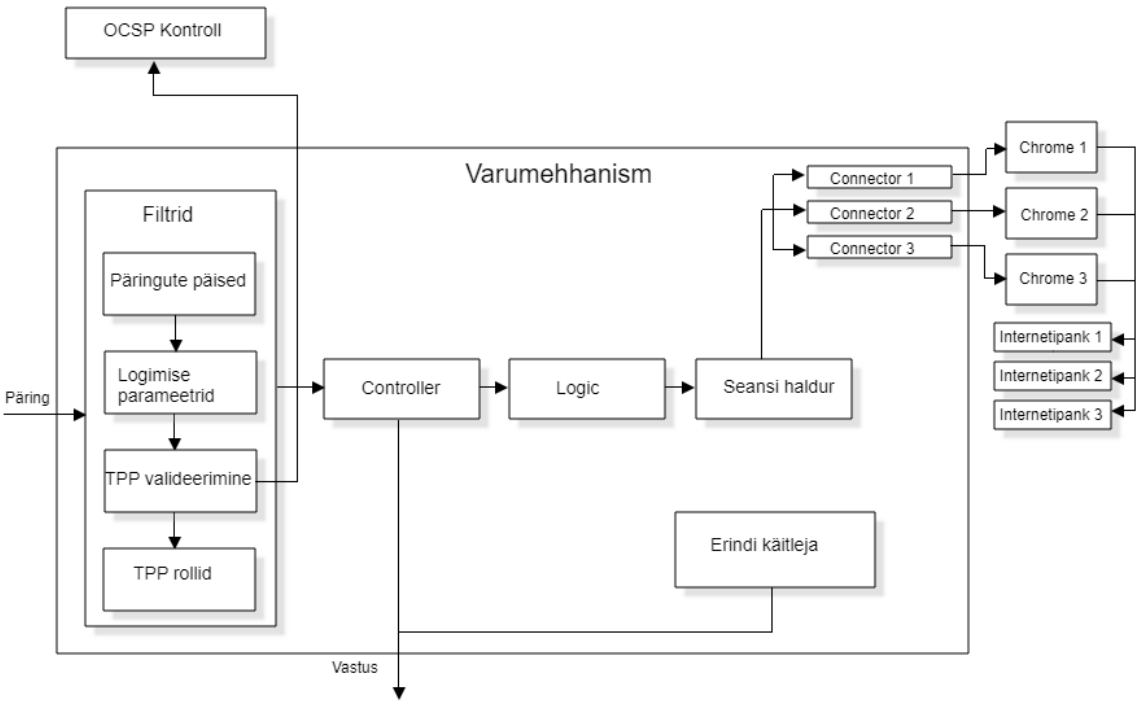
Rakendust on arendatud Java 12 baasil, kasutades *SpringBoot* raamistiku. Antud tehnoloogilised valikud on tehtud projekti arendusettevõtte ning töö autori pädevusi silmas pidades. Lisaks, pakub *SpringBoot* pakub sõltuvuste sisestamiseks ning rakenduse konfigureerimiseks kasutajasõbralikke vahendeid.

Rakendus põhineb REST API disainimustril ning suhtluskihiks on HTTP. Rakendus on *stateless*, millest tulenevalt puudub andmebaasi komponent. Kuna rakenduses käsitletakse tundlikuid isikuandmeid, siis andmete haldamine on vastuolus GDPR-iga (*General Data Protection Regulation,* Isikuandmete kaitse üldmäärus). Lisaks pole rakenduse põhiprotsessideks andmebaasi tarvis.

Internetipankade nn. veebiämbliku realiseerimiseks on kasutatud tarkvara testimisraamistiku *Selenium Webdriver*, mis kasutab andmete vahenduseks veebibrauserispetsiifilist draiverit, milleks antud lahenduses on *Google Chrome*-il põhinev draiver (*ChromeDriver*). *Seleniumi* kasutamiseks ei ole vaja eraldiseisvat serverit, veebidraiver käivitab brauseri lokaalselt ning täidab tegevusi selles brauseris [3]. See võimaldab arendajal visuaalse tagasiside saamist arendamise ajal. *Seleniumi*-i abil on võimalik väga hästi imiteerida inimlikku kasutajaliidese kasutust, mille võrra on *Selenium* aeglane võrreldes raamistikega nagu *HTMLUnit*, kuid kuna rakendus kasutab PSU-dele suunatud liideseid, mille reageerimiseajad sisenditele on informeeritud tavakasutajate kiirusest, on see omadus positiivne.

4.2 Arhitektuur

Süsteemi arhitektuur on paika pandud selliselt, et süsteemis oleks minimaalselt internetipanga spetsiifilisi komponente. Ainsateks internetipankade spetsiifilisteks komponentideks on nn. veebiämblikute realisatsioon, kuna kõigi kolme internetipanga kasutajaliidesed erinevad üksteisest sellisel määral, et ei võimalda ühist lahendust. Illustreerimaks süsteemisiseste komponentide rolli tarkvaras, on järgnevalt esitatud päringu elutsükkel rakenduse siseselt:



Joonis 1. Rakenduse arhitektuur päringu elutsükli näitel

Tabel 2. Komponendid koos nende rolliga rakenduses

Komponent	**Roll rakenduses**
Päringu päiste filter	Päringu päisest süsteemi toiminguteks vajaliku informatsiooni ekstraheerimine
Logimise parameetrite filter	Logi kirjetele parameetrite sätestamine
TPP valideerimise filter	TPP valideerimine (kirjeldatud pt 3.2)
TPP rolli filter	TPP volitamine (kirjeldatud pt 3.2)
OCSP kontroll	Sertifikaadi OSCP kontroll (kirjeldatud pt 3.2)
Controller	Päringu sisendi ekstraheerimine ning väljundi tagastamine
Logic	Sisendi valideerimine, seansihaldurilt saadud seaniga seotud connectori instantsile päringu edastamine
Seansi haldur	Seansi haldus (kirjeldatud pt 3.3)
Connector	Internetipanga spetsiifiline teabevahetus kanal. Teabevahetus *ChromeDriveri* vahendusel internetipankadega
Erindi käitleja	Süsteemi tööajal esinevate erindite käitlemine

4.3 Veebiämbliku murekohad

Veebiämblikute kasutamine informatsiooni lugemiseks ning toimingute tegemiseks on laialt levinud metoodika. Seda lihtsal põhjusel, et antud lähenemine hõlbustab ettevõtetel (eriti levinud *FinTech*-is, ehk finantstehnoloogia) võrdlemisi lihtsalt API-sid luua andmete vahetuseks, toetumata andmete halduritele, kuna kasutajapoolsed liidesed on tavaliselt avalikud.

Sellise lahendusega kaasnevad lahendusele omased probleemid:

- Veebilehtedest andmete lugemiseks peavad veebiämblikud alla laadima palju rohkem informatsiooni, kui neil vaja on (igal lehel terve HTML, konkreetsete andmeväljade asemel), mis oluliselt mõjutab süsteemi jõudlust.

- Veebilehtedel navigeeritakse justkui kasutaja, kuid oluliselt kiiremalt. Sellest tulenevalt on rakenduses *race condition*-id kergelt esinema (näiteks JavaScript mingi nupu taga pole ära laadinud enne sellele vajutamist).

- Muutused veebilehe HTML-is kannavad kaasas riski lõhkuda kogu rakenduse toimingut.

Nende probleemide leevendamine osutus rakenduse arendamisel kõige keerukamaks osaks, kuna kirjeldatud probleemide ennetamine tekitab vastuolulisi otsuseid. Veebilehtede veakindlateks tegemine nõuab ettenägelikust olukordadele, milleks puuduvad vastavad testandmed, näiteks ootamatud teated internetpankadelt või *race condition*-id, mis võivad esineda spetsiifilistest olukordades. Kuigi kasutusel oleva tehnoloogiaga on võimalik kirjutada loogikat, mis ennetab potentsiaalseid probleeme, teeb ta seda jõudluse arvelt. Sellest tulenevalt, peab arendamise jooksul leidma optimaalse tasakaalu jõudluse ja töökindluse vahel.

Väikeste muutustega veebilehtedel on hästi realiseeritud veebiämblikud toimetulekuks varustatud, kasutades hästi defineeritud elementide lokaatoreid ning eelnevalt defineerituid kasutajavooge. Sellegipoolest suuremad muutused kasutajavoogudes on alati riskikohaks veebiämblikuid rakendatavatel süsteemidel, eriti sellised mis nõuavad kliendilt ebaootuspärast sisendit (nt. internetipangas kontaktandmete uuendamine), et edasi liikuda.

Finantsandmete koondamise teenust pakkuv ettevõte *MX*, mis toetub suuresti veebiämblikutele, on probleemi lahendamiseks kasutusele võtnud nn. „*multi-sourced aggregation*“-i (mitmest allikast koondamine) [5]. Sisuliselt kasutab nende andmete agregeerimise teenus mitut alamteenust (tuntuimad pakkujad on *FiniCity*, *Yodlee* ja *Quovo*), mis realiseerivad veebiämblikuid kindlates finantsasutustes. Seega olukorras, kus üks andmeallikas ei ole saadaval või ei tule toime spetsiifilise päringuga, tõstetakse

ühendus järgmisele andmeallikale. Selline lähenemine väldib olukordi, kus klient peab ootama muutust andmeallika tarkvaras, et pakutud finantsteenust kasutada.

Erinevalt *MX*-i rakendusest, on varumehhanism ettevõtte sisene projekt, mis toetub ettevõtte sisestele allsüsteemidele. Seega opereeritakse eeldusel, et suuremad muutused internetipankade HTML-is või kasutajavoos raporteeritakse eelnevalt varumehhanismi toetavale meeskonnale, tagamaks muutuste kajastamist käsitletavas süsteemis.

4.4 Testimine

Rakenduse testimine jaguneb kaheks osaks, funktsionaalseks testimiseks ning koormustestimiseks.

4.4.1 Funktsionaalne testimine

Funktsionaalsel testimisele on kasutusele võetud ühiktestimine, suitsutestimine ning regressioonitestimine.

Tarkvara on, kus võimalik, kaetud ühiktestidega, et kindlustada erinevate süsteemi osade korrektset käitumist. Rakenduses on realiseeritud lihtne kasutajaliides, mis võimaldab teste käivitada ning nende tulemusi kuvada läbi brauseri genereeritud HTML-ina. Antud funktsionaalsus pakub mugavat testimise protsessi arendusprotsessi ajal.

Selleks, et testida rakenduse üldist funktsionaalsust on kasutusele võetud suitsu- ning regressioonitestimine. Antud rakendust ei saa täielikult automatiseeritud testidega katta, kuna kasutusel on süsteemivälised teenused. Selleks, et testida kõiki integratsioone erinevate teenustega on vaja läbida kogu rakenduse tegevusahelat. Süsteemi suitsuteste on automatiseeritud läbi *Postman*-i ning testimise raporti genereerimiseks on kasutusel *Newman*. *Postman* võimaldab luua API testide kollektsioone, mida käivitatakse *Newmani*-i abil läbi kasurea. *Newman* genereerib testimise raporti HTML failina, mida kasutatakse dokumentatsioonina igal tarkvaraversiooni kasutusele võtmisel. *Newman*-i ja *Postman*-i integratsiooni kohta saab täpsemalt lugeda *Postman*-i kodulehelt [4].

Regressioonitestimine on manuaalne protsess, mida viib läbi delegeeritud testimise meeskond. Meeskond teeb regressiooni teste enne igat tarkvara versiooni kasutusele võttu. Põhitegevusteks on vajalike testandmete kogumine, testimine ning testide

defektide arendajatele raporteerimine. Lisaks on tarkvara versiooni kasutusele võtul tarvis testimise meeskonnalt vastav regressiooni testide raport.

4.4.2 Koormustestimine

Koormustestidel on kaks põhieesmärki:

- Kindlaks teha, mitu paralleel seansi on süsteem võimeline käsitlema
- Kuidas seansside arv mõjutab süsteemi jõudlust

Seansside piirarv on vastavuses saadaval olevale vabale vahemälule, kuna see on proportsionaalselt süsteemi poolt enim kasutatud ressurss. Suur vahemälu kasutus tuleneb veebidraiveritest. Selleks, et määrata seansside piirarv, käivitatakse süsteemis perioodiliselt seansse, kuni serveri kokku jooksmiseni. Antud arv peab olema märgatavalt suurem kui mittefunktsionaalsetes nõuetes sätestatud miinimum, et süsteemil oleks tagatud varuressurss.

Lisaks tuleb koormustestidega silmas pidada, et süsteem piirkoormuse all, ei tohi kaotada jõudlust. Selleks, et seda testida, käivitatakse süsteemis järk-järgult seansse ning võrreldakse süsteemi reaktsiooniaegu erinevate koormate all.

5 Kokkuvõte

Lõputöö eesmärk oli arendada rakendusliides, mis pakub kolmanda osapoole finantsteenuse pakkujatele juurdepääsu panganduse andmetele ja teenustele. Valminud infosüsteem kasutab internetipankade funktsionaalsust teenuste pakkumiseks, rakendades seejuures Makseteenuse direktiivist tulenevaid nõudeid.

Rakendamise arendamisel oli vaja suurt tähelepanu pöörata rakenduse vastavust Makseteenuse direktiivi nõuetele, internetipankade projekti integreerimisele ning süsteemi jõudluse tagamisele. Ülesande suurimaks riskikohaks oli internetipankade kasutajaliidese abil andmete sõelumine, mis tuleneb kasutajaliidesest esinevatest anomaaliatest. Rakenduses realiseeritud koodi silumiseks ettenähtud funktsionaalsus aitab järk järgult rakenduse kasutatavust parandada, tagades süsteemi töökindluse paranemist.

Valminud liidese kasutajateks on registreerunud kuus kolmanda osapoole finantsteenuste pakkujat, millest kaks on integreerinud liidese enda rakendustesse, viies läbi sadu makseid igapäevaselt. Töö autor vastutab süsteemi ülalpidamise eest, jälgides süsteemi jõudlust ja tehes süsteemisiseseid täiendusi.

Kõik töös püstitatud eesmärgid said täidetud. Valminud süsteem on Makseteenuste direktiivile nõuetele vastav ning pakub töökindlaid makse- ning kontoteabe teenuseid, kinnitades, et antud tarkvara arhitektuur on sobilik varumehhanismi lahendus.

Kasutatud kirjandus

[1] The revised Payment Services Directive (PSD2) and the transition to stronger payments security – Euroopa Keskpank [WWW] https://www.ecb.europa.eu/paym/intro/mip-online/2018/html/1803_revisedpsd.en.html

[2] Komisjoni delegeeritud määrus (EL) 2018/389 – EUR-Lex [WWW] https://eur-lex.europa.eu/legal-content/ET/TXT/?uri=CELEX%3A32018R0389

[3] Selenium Webdriver – Vikipeedia [WWW] https://et.wikipedia.org/wiki/Selenium

[4] Command line integration with Newman – Postman [WWW] https://learning.postman.com/docs/postman/collection-runs/command-line-integration-with-newman/

[5] Account Aggregation and Your Reputation – MX [WWW] https://www.mx.com/resources/2015/3/5/are-you-risking-your-reputation-on-account-aggregation