

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Siiri Kliimson 204312IAAM

ISIKLIKE FINANTSTEGEVUSTE JÄLGIMISE RAKENDUSE KAVANDAMINE

Magistritöö lühikokkuvõte

Juhendaja: Nadežda Furs
MBA, doktorant

Tallinn 2022

1 Eesmärgid

Rahatarkus ehk finantskirjaoskus on oskuste, teadmiste, hoiakute ja käitumise kogum, mis on vajalik arukate otsuste tegemiseks rahaasjades ja oma pere majandusliku heaolu tagamiseks [1]. Kuigi üle 70% elanikest omab rahatarkuse baastadmisi, siis ei rakenda suur osa elanikest oma teadmisi (seda teeb vaid 58,8%). Samuti on lausa 28% neid, kellel ei jää peaaegu kunagi raha kuu lõpus üle ja 31% elanikest ei ole palgast säästnud.

Ligikaudu kolmandik elanikest tunneb pidevat muret oma arvete tasumise pärast, kuid majandusliku kindlustunde saavutamine on vaid seitsme protsendi elanike üks rahaline eesmärk [2]. Samast uuringust selgub ka, et suur osa elanikest ei planeeri piisavalt oma rahaasju ning eelarvet koostab vaid 43% elanikkonnast ning 63% peredest peab oma kulude üle arvet. Kas see ka praktikas nii on, pole võimalik kindlaks teha, kuid nagu uuringus välja tuuakse, võib nii kõrge protsent viidata teadlikkusele.

Kuna inimeste eelarvestamine ning rahaasjadest ülevaate omamise protsent on madal, siis väheneb ka võimekus teadlikult ning maksimaalselt säästa. See toob kaasa kas reservi puudumise või liiga väikese hädaabifondi, mis omakorda vähendab kindlustunnet tuleviku suhtes.

Magistritöö autor on osa arendusmeeskonnast, kes on seadnud endale eesmärgiks luua Eesti elanikele mugav ning lihtne tööriist, mille abil on võimalik parandada ülevaadet enda rahaasjadest ning sellega ka tõsta oma rahatarkust. Näiteks on isiklike finantsasjade juures oluline seada eesmäärke, teha kindlaks hetkeolukord, koostada bilanss omatavate varade ja kohustuste kohta, vaadata süstemaatiliselt rahavoogude aruannet ning koostada (pere) eelarvet [3].

Magistritöö eesmärk on kaardistada kuluarvestuse rakendus koht konkurentsiturul ja määrata ära põhilised välised mõjurid, mille abil hiljem panna paika strateegiad. Samuti koostada loodava rakenduse minimaalse jätkusuutliku toote (edaspidi MVP (*minimum viable product*)) analüüs, mis võimaldaks inimestel jälgida nii oma rahavooge kui ka

koostada eelarvet, mille abil saavad kasutajad tekitada endale mänguruumi igakuistes kulutustes, samal ajal soovitud ulatuses säästes.

Tarkvara peaks olema kasutajale võimalikult mugav ning intuiitiivne, et kasutajad saaksid seda kasutada võimalikult eesmärgipäraselt. Rakendus peab olema kasutatav mitme kasutaja poolt korraga, kui soovitakse näiteks leibkonna kulutusi koos jälgida. Samuti saab oma kasutajakonto külge lisada arvelduskonto, mille tehingud sünkroniseeritakse kasutaja rahavoogude hulka läbi panga API-de (*Application Programming Interface*). Et minimeerida aja kulu tehingute kategoriseerimisel, saab kasutaja vastavalt tehinguga kaasas olevale infole konfigureerida, kas tegu on kulu või tuluga ning millisesse kategooriasse tehing määratakse (näiteks toit, auto, üür jmt). Lisaks saab kasutaja koostada eelarvet, et määrata piirid kulutusteks ja tulu ja säästude puhul eesmärkideks, mida soovitakse saavutada.

2 Töös kasutatavad metoodikad

Autor kasutab loodava rakenduse kavandamisel iteratiivset lähenemist, kus arendamisele lähenetakse tsüklikena, kus iga tsükkel on kui mini-projekt analüüsi, disaini, programmeerimise ja testimisega [4]. Iteratsiooni lõppedes ei pruugi valminud arendus alati minna kasutajale kohe kasutamiseks, samas ei ole üks iteratsioon ainult analüüsiks või testimiseks, vaid koosneb kõikidest funktsionaalsuste loomiseks vajalikest sammudest. Lisaks uue funktsionaalsuse arendamisele parandatakse arenduses tekkinud vigu. Loodava rakenduse esimene iteratsioon on MVP koostamine (käesolev töö). Seejärel hakatakse arendama, analüüsima ja testima jooksvalt ning iga iteratsioon lõppeb mingi täiendava funktsionaalsuse välja laskmisega (kuigi see ei pruugi kohe rakendusse jõuda).

MVP lähenemine, mida autor kasutab, on arenduse lähenemine uue toote turule toomiseks [5]. MVP on versioon uuest tootest, mis võimaldab pakkuda tarbitavat toodet väikeste iteratsioonidega. Kasutajale pakutakse uusi funktsionaalsusi ja edasiarendusi osadena. Seeläbi saab kasutaja tarbida järjest arenevat toodet ning arendusmeeskond saab süvenenult pühenduda ühele väiksemale tükile korraga.

Autor kasutab ka disain-mõtlemise tehnikaid, et mõista paremini (tulevasi) kasutajaid ning paremini aru saada nende probleemidest [6]. Sellise lähenemisega soovib autor sügavamalt mõista kasutajaid ning kiiresti töö käigus õppida ning selle arvelt rakendust parandada ja edasi arendada (*fail early on and often*).

Järgnevates alapeatükkides tutvustab autor metoodikaid, mida kasutab loodava rakenduse analüüsi koostamiseks.

2.1 Ärianalüüsi metoodikad

Käesolevas alapeatükis toob autor välja metoodikad, mida ta kasutab loodava rakenduse ärianalüüsi koostamiseks.

2.1.1 PESTLE

PESTLE on metoodika analüüsimaks ettevõtte peamisi poliitilisi, majanduslikke, sotsiaalseid, tehnoloogilisi, juriidilisi ning keskkonna võtmetegureid, mis mõjutavad ettevõtet väljastpoolt [7] ning välja joonistuvate mõjurite alusel soovib autor määratleda ettevõtte strateegiad.

Poliitilised tegurid määravad, kuidas valitsus võib mõjutada majandust või kindlat valdkonda, majanduslike tegurite all tuuakse välja need, mis mõjutavad ettevõtet otseselt ning pikaajaliselt [8]. Sotsiaalsete tegurite all tuuakse välja millised on näiteks kultuurilised normid või hoiakud ühiskonnas [9].

Tehnoloogiliste mõjude all on innovatsioon ning tehnoloogiate areng, juriidiliste teguritena vaadatakse, kuidas on võimaldatud või piiratud juurdepääs andmetele ning näiteks milline on maksusüsteem ja keskkonna teguritena tuuakse välja, kuidas ümbritsev keskkond võib ettevõtet mõjutada [10]. Näiteks energia tarbimine või erinevad keskkonda mõjutavad seadused.

2.1.2 Porteri viie konkurentsijõu mudel

Konkurentsijõu mudeli abil saab määrata ettevõtte tegevusvaldkonna konkurentsiolukorra [11]. Porteri viie konkurentsijõu mudel aitab välja selgitada tegevusvaldkonna olukorra ning analüüsida ettevõtte kohta seal. Läbi viidava analüüsi abil soovib autor kaardistada loodava rakenduse koha turul ning määrata ära konkurentsijõulisuse.

2.1.3 Riskide analüüs

Et paremini määrata ära tootega seonduvad riskid, kasutab autor riskide analüüsiks soojuskaardi metoodikat. Riskide soojuskaardi analüüs võimaldab riskide määramisel kasutada visuaalset tööriista, et määratleda riski juhtumise tõenäosust ja mõju ettevõttele [12].

loodav lahendus pakub. Autor soovib isikute abil paremini mõista, kes on tulevased potentsiaalsed kliendid, tuvastada nende rakenduste kasutusharjumused ning seeläbi kaardistada nende (tõenäolised) vajadused ja soovid [6].

2.1.6 Ärisõnastik

Äriliselt olulised terminid kirjeldab autor lahti ärisõnastikus. Äriterminid on nimisõnad, mida organisatsioon kasutab kirjeldamiseks osapoolt, asukohta, konkreetset või abstraktset asja [15]. Iga toodav kirjeldus peab selgelt ja üheselt määratlema, milliseid objekte kirjeldatakse ja milliseid mitte. Ärisõnastik võimaldab autoril süstematiseeritult välja tuua rakenduses olulised mõisted ning need lahti kirjeldada, et kõigil osapooltel oleks ühine arusaam, mida termineid kasutades mõeldakse.

2.2 Süsteemianalüüsi meetodikad

Järgnevatel alapeatükkides kirjeldab autor meetodikaid, mida ta kasutab loodava rakenduse süsteemianalüüsi koostamiseks.

2.2.1 Funktsionaalne tükeldamine (*Functional Decomposition*)

Nõuete kogumisel (ja kasutusmallide mudeli loomisel) kasutab autor funktsionaalset tükeldamist. Selline viis aitab hallata keerukust ning vähendada ebaselgust läbi protsesside, süsteemide ja funktsionaalsuste tükeldamise [16]. Seeläbi on võimalik iga osa analüüsida eraldi. Funktsionaalsuse tükeldamisele lähenetakse vaadates süsteemi kui funktsionaalsuste koostööd.

2.2.2 Kasutusmallide mudelid

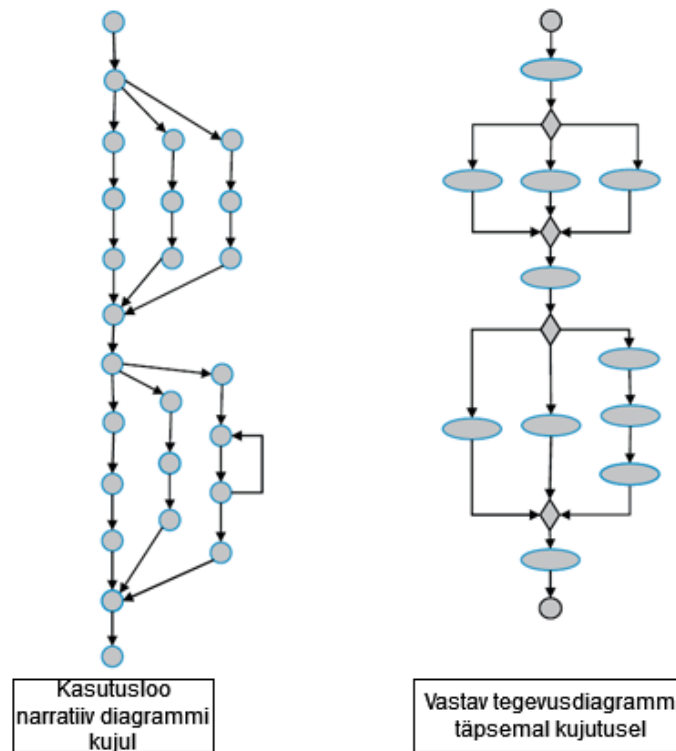
Kasutusmallide mudelid (*Use Case Diagram*) kirjeldavad, kuidas inimene või süsteem modelleeritava lahendusega suhtlevad, et saavutada eesmärki [16]. Kasutusmallid (edaspidi UC) kirjeldavad tegevusi kasutaja (*actor*) ja loodav lahenduse vahel.

Kasutusmallid kirjeldab autor järgides vaba formaati (*casual format*), mida kasutatakse mitteformaalse kirjeldusena ning kus ühe kasutusmalli erinevad stsenaariumid kirjeldatakse ühe lõiguga [17]. Autor ei kasuta kirjelduse täisversiooni (*fully dressed*), kuna kasutab täpsema voo kirjeldamiseks tegevusdiagramme (*Activity Diagram*).

Antud metoodikat kasutab autor, et kaardistada kõiki kasutaja tegevusi rakendusega funktsionaalsuste järgi, mille kaudu on lihtsam tarkvara arendama hakata.

2.2.3 Tegevusdiagrammid

Kasutusmallide kirjeldamise narratiivid on olulised funktsionaalsete nõuete kaardistamiseks [18]. Kasutusmallide kirjeldused on head, et kirjeldada süsteemi käitumist väliste päringute puhul, kuid nende põhjal on keerulisem paika panna, kas kasutusmall on terviklik või mitte. Kasutusmallide mudeli selgitamiseks kasutab autor tegevusdiagramme, kuna need pakuvad lisaks tekstilisele kirjeldusele ka visuaalset, mida on teatud olukordades lihtsam jälgida, kui suurt hulka teksti läbi lugeda.



Joonis 2. Protsessi loogika läbi kasutusmallide ning tegevusdiagrammide (autori tõlgitud) [18]

Tegevusdiagrammide eelis BPMN-i (*Business Process Model and Notation*) ees on fokuseeritus tarkvara lahenduste kirjeldamisele ning seda on lihtne implementeerida ka programmi koodis [19]. Tegevusdiagrammide kasutamine võimaldab kaardistada detailset sammude kirjeldust ning visuaalselt kuvada selget töövoogu.

2.2.4 Olemi-suhte diagramm

Olemi-suhte diagramm (edaspidi ERD (*Entity Relationship Diagram*)) visualiseerib kuidas olemid, objektid või kontseptid üksteisega suhtlevad [20]. ERD-d kasutatakse nii andmebaasi disainimiseks, andmebaasi veaotsinguks või ka näiteks äriprotsesside ümber tegemiseks. Käesolevas töös kasutab autor ERD füüsilist andmemudelit andmebaasi disainimiseks, mis on aluseks rakenduse pakutavate funktsionaalsuste realiseerimiseks. Füüsiline andmemudel võimaldab imiteerida kõige täpsemalt andmebaasi struktuuri ning kirjeldab ära ka kõik atribuudid koos vastavate piirangute ja andmetüüpidega [21].

Andmemudeli koostab autor, et rakenduse arendajal oleks arendama hakates visuaalne mudel mille järgi andmebaas kokku panna, ning et olulised komponendid ei jää välja toomata (sh. kõiki funktsionaalsusi oleks võimalik täita).

2.2.5 Seisundidiagramm

Loodav rakendus kasutab mõnede objektide puhul seisundeid ning nende visualiseerimiseks kasutab autor seisundidiagrammi. Seisundidiagrammi kirjeldab erinevaid komponendi seisundeid süsteemis terve tema elukaare jooksul [22]. Visuaalne pilt kasutusel olevatest seisunditest aitab saada tervikpildi ning näitab visuaalselt, kuidas toimub üleminek ühelt seisundilt teisele.

2.2.6 Kasutusmallide punktid (*Use Case Points*)

Kasutusmallide punktid (edaspidi UCP) on meetoodika, et hinnata ühe kasutusmalli arendustöö suurust [23]. UCP arvutamiseks läbitakse järgmised sammud [24]:

1. Arvutatakse korrigeerimata kasutusmalli keerukus (edaspidi UUCP)

1.1. Klassifitseeritakse kasutusmalli keerukus (Tabel 1). Autor määrab keerukuse kasutusmallide protsessi kirjeldavate tegevusdiagrammide sammude arvu põhjal.

Kasutusmalli keerukus	Operatsioonide arv	Kaal
Lihtne UC (NSUC)	≤ 3	5
Keskmine UC (NAUC)	4-7	10

Keerukas UC (NCUC)	>7	15
--------------------	----	----

Tabel 1. Kasutusmalli klassifitseerimine (autori tõlgitud) [24]

Sammudeks loetakse kõik tegevused, kuna need tuleb olenemata protsessi kulgemisest valmis arendada. Küll aga kordusele minevaid tegevusi ei arvestata omaette operatsioonina, kuna need tsüklid võivad käia määramata arv kordi. Kui tegu on paralleelsete operatsioonidega, siis võetakse mõlemad eraldi operatsioonidena arvesse. Kui protsess viitab teisele kasutusmallile, siis võetakse see arvesse ühe operatsioonina, kuna vajalikud funktsionaalsused saab üle tuua teise kasutusmalli jaoks loodud arendustega.

1.2. Arvutatakse korrigeerimata kasutusmalli kaal (edaspidi UUCW) järgneva valemi abil: $UUCW = 5 * NSUC + 10 * NAUC + 15 * NCUC$

1.3. Klassifitseeritakse teguri (*actor*) keerukus (Tabel 2)

Teguri keerukus	Tüüp	Kaal
Lihtne tegur (NSA)	Väline süsteem, mis suhtleb arendatava rakendusega üle API	1
Keskmine tegur (NAA)	Väline süsteem, mis suhtleb arendatava rakendusega läbi kindla protokolliga	2
Keerukas tegur (NCA)	Kasutaja, kes suhtleb arendatava rakendusega läbi kasutajaliidese	3

Tabel 2. Tegurite klassifitseerimine (autori tõlgitud) [24]

1.4. Arvutatakse korrigeerimata teguri kaal (edaspidi UAW) järgneva valemi abil: $UAW = 1 * NSA + 2 * NAA + 3 * NCA$

1.5. Arvutatakse UUCP järgmise valemiga: $UUCP = UUCW + UAW$

2. Korrigeeritakse UUCP tehnilise keerukusega

2.1. Võetakse arvesse tehniliste faktorite kaalud (edaspidi TW) (Tabel 3)

Faktor	Kirjeldus	Kaal
T1	Hajussüsteem	2,0
T2	Vastuse aeg või läbilaskevõime	1,0
T3	Lõppkasutaja efektiivsus	1,0

T4	Keerukad sisemised protsessid	1,0
T5	Kood peab olema taaskasutatav	1,0
T6	Lihtne allalaadimine	0,5
T7	Lihtne kasutamine	0,5
T8	Teisaldatavus	2,0
T9	Lihtne muutmine	1,0
T10	Samaaegsus	1,0
T11	Lisa turvalisuse objektiivid	1,0
T12	Otsene kolmanda osapoole ligipääsetavus	1,0
T13	Kasutaja koolituse vajadus	1,0

Tabel 3. Tehniliste faktorite kaalud (autori tõlgitud) [24]

2.2. Kõigile faktoritele määratakse olulisus (edaspidi TRV) nullist viieni (null – pole oluline, viis – on väga oluline)

2.3. Arvutatakse faktori mõju (edaspidi TF) järgmise valemi kaudu: $TF = TW * TRV$ ning seejärel täielik tehniline faktor (edaspidi TFactor), mis on summa TF-dest.

2.4. Arvutatakse tehnilise keerukuse faktor (edaspidi TCF) järgmise valemi kaudu: $TCF = 0,6 + (0,01 * TFactor)$

3. Kohandatakse keskkonna keerukust

3.1. Võetakse arvesse keskkonna faktoreid, millel on vastavad kaalud (edaspidi EW) (Tabel 4)

Faktor	Kirjeldus	Kaal
F1	Tuttav projekti mudel	1,5
F2	Rakenduse kogemus	0,5
F3	Objektorienteerituse kogemus	1,0
F4	Peaanalüütiku võimekus	0,5
F5	Motivatsioon	1,0
F6	Stabiilsed nõuded	2,0

F7	Osalise tööajaga meeskond	-1,0
F8	Keeruline programmeerimiskeel	-1,0

Tabel 4. Keskkonna faktorite kaalud (autori tõlgitud) [24]

3.2. Kõigile faktoritele määratakse olulisus (edaspidi ERV) nullist viieni (null – pole oluline, viis – on väga oluline)

3.3. Arvutatakse faktori mõju (edaspidi EF) järgmise valemi kaudu: $EF = EW * ERV$ ning seejärel täielik keskkonna faktor (edaspidi EFactor), mis on summa EF-dest.

3.4. Arvutatakse keskkonna keerukuse faktor (edaspidi ECF) järgmise valemi kaudu: $ECF = 1,4 + (-0,03 * EFactor)$

4. Arvutatakse korrigeeritud kasutusmallide punktid: $UCP = UUCP * TCF * ECF$

UCP-d kasutab autor, et planeerida realistlikult aega rakenduse avaldamiseks ning samuti kaardistada, millises järjekorras (võimalusel) tuleks kasutusmalli põhiseid arendusi tegema hakata. Arendajaga on kokku lepitud, et alustatakse funktsionaalsusest, mis on oluline kõigi komponentide toimimiseks ning seejärel võetakse esmalt arendusplaani lihtsamad tööd ning võimalusel autor tükeldab suuremaid kasutusmalle väiksemateks, et neid oleks võimalik iteratiivsel põhimõttel arendada (meeskonnas hakatakse rakendama kahenädalasi sprinte ning üks arendustöö ei tohiks tavaliselt ületada 2 päeva arendustöö mahtu).

2.3 Arhitektuur ja prototüüpimine

Järgnevides alapeatükkides kirjeldab autor komponentdiagrammi meetodikat ning prototüüpimise meetodikaid, mida ta loodava rakenduse arhitektuuri ja prototüübi koostamisel järgib.

2.3.1 Komponentdiagramm

Komponent defineerib oma käitumise selle põhjal, mida teised liidesed temalt küsivad ja mida talle annavad [17]. Komponentide modelleerimine rõhutab, et liidesed on olulised ning süsteemi osad on modulaarsed, eraldiseisvad ning asendatavad.

Komponendipõhine disain on mõeldud näitlikustamiseks eraldiseisvaid tarkvara osi, liideseid ja seoseid, mida on võimalik eraldi arendada [25].

Autor loob komponentdiagrammi, et struktureerida kasutusmalle ning muuta lihtsamaks arenduste järjekord. Samuti aitab komponentdiagramm väljast tulevatel inimestel mõista, millised on suuremad tarkvara tükid, mida arendatakse.

2.3.2 Prototüüpimine

Prototüüpimine on oluline osa disainimõtleemisest ja kasutaja kogemuse disainist [26]. Prototüüp võimaldab ideid kiiresti testida ja edasi arendada. Neli põhilist prototüüpimisest saadavat tulemust on prototüüp ise, ehk idee esitus, täpsus (madal, keskmine või kõrge, autor antud töö raames koostab kõrge täpsusega prototüübi) [27]. Samuti saab luua interaktiivsust ning seeläbi loodud näidist kasutajate peal testida ning lõpuks evolutsioon, mille käigus prototüüp areneb ja saavutab oma lõpliku kuju. Peamisteks kõrge täpsusega prototüüpide eelisteks on paindlikkus, realistlikkus ning kasutajate peal testimise kiirus. Samas kitsaskohtadeks on õppimiskõver, mis tuleb läbida enne prototüüpima hakkamist ning suurenenud ajakulu.

Autor loob kasutusmallide põhjal kõrge täpsusega klikitava makett (*mock-up*) prototüübi, et läbi viia hiljem ka kasutajatestimist ning pakkuda arendajale võimalikult täpset malli, mille põhjal arendama hakata.

2.4 Implementatsiooni plaan

Implementatsiooni plaan on strateegiline plaan, mis aitab saavutada meeskonna seatud eesmärgid [28]. See on dokument, mis paneb paika samm-sammulised tegevused, et seatud eesmärk saavutada.

Autor kasutab implementatsiooni plaani, et määratleda projekti ajaline raamistik, defineerida eesmärgid ning anda meeskonnale ühine allikas, mille alusel jaotatakse vastutust, kes, mille eest, millal, kuidas ja miks vastutab. Samuti soovib autor määrata järgmised sammud selliselt, et tehtavaid töid hakataks tegema vastavalt prioriteetidele. Implementatsiooni plaani pannakse sammud, mis järgnevad tehtud rakenduse kavandamisele.

Järeldused

Töös kasutatud metoodikate abil koostas autor ülevaate rakendust puudutavatest välismõjuritest ning konkurentsijõust. Autor koostas maatriksi, et tuvastada erinevusi juba olemasolevate rakenduste ja loodava lahenduse vahel. Samuti koostas ta ärianalüüsi käigus eelnevale tugineva ärimudeli, millele baseerusid persoonad kliendi paremaks tundmaõppimiseks. Ärianalüüsis koostas autor veel MVP kirjelduse, üldised ärinõuded ja -sõnastiku ning kaardistas peamised rakendust ning ettevõtet mõjutavad riskid.

Süsteemianalüüsi faasis kaardistas autor kasutusmallid, mille kirjeldas lahti kirjelduste ja tegevusdiagrammide abil. Kõigile kasutusmallidele määras autor ka keerukuse ning seeläbi arvutas rakenduse üldise keerukuse ning potentsiaalse arenduse ajakulu. Lisaks koostas autor komponentdiagrammi rakenduse omavahel suhtlevate ent eraldiseisvate komponentide kaardistamiseks ning koostas andmemudeli.

Lisaks visualiseeris autor läbi prototüüpimise osad kasutajaliidese vaated, mille abil saada ülevaadet, kas ning kuidas on kasutusmallid realiseeritavad. Samuti annab see nii analüütikule kui ka arendajale ühise arusaama sellest, kuidas peaks kasutajaliides välja nägema.

Autor koostas implementatsiooni plaani, kus tõi välja järgmise kahe aasta ülesanded, mis tuleb kindlasti täita. Ning viimasena toodi välja mõned võimalikud edasiarendused, mida rakendusele peale MVP lansseerimist juurde saab lisada.

Magistritöö tulemi põhjal saab meeskond hakata oma ala spetsialiste kaasama, kelle tööle on võimalik loodud dokumentatsiooni alusel sisendit andma selle kohta, mida peab olema rakendus võimeline tegema ning rakenduse arendaja saab loodud dokumentatsiooni alusel alustada programmeerimist.

Kasutatud kirjandus

- [1] Rahandusministeerium, „RAHATARK EESTI - Eesti elanike rahatarkuse edendamise strateegia aastateks 2021-2030,“ Tallinn, 2021.
- [2] Turu-uuringute AS, „Eesti elanike finantskirjaoskuse ehk rahatarkuse uuring,“ Turu-uuringute AS, 2019.
- [3] T. T. T. V. Kristjan Liivamägi, Rahaedu põhimõtted, Argo, 2021.
- [4] C. Larman, Agile and Iterative Development: A Manager's Guide, Addison-Wesley Professional, 2003.
- [5] Agile Alliance, [Võrgumaterjal]. Available: [https://www.agilealliance.org/glossary/mvp/#q=~\(infinite~false~filters~\(tags~\(mvp\)\)~searchTerm~'~sort~false~sortDirection~'asc~page~1\)](https://www.agilealliance.org/glossary/mvp/#q=~(infinite~false~filters~(tags~(mvp))~searchTerm~'~sort~false~sortDirection~'asc~page~1)). [Kasutatud 09 11 2021].
- [6] P. L. L. Michael Lewrick, The Design Thinking Toolbox, München: John Wiley & Sons, 2020.
- [7] M. Morrison, „CIPD,“ 10 11 2020. [Võrgumaterjal]. Available: <https://www.cipd.co.uk/knowledge/strategy/organisational-development/pestle-analysis-factsheet#gref>. [Kasutatud 30 10 2021].
- [8] PESTLEanalysis.com, „PESTLE ANALYSIS,“ [Võrgumaterjal]. Available: <https://pestleanalysis.com/what-is-pestle-analysis/>. [Kasutatud 30 10 2021].
- [9] GroupMap, GroupMap, [Võrgumaterjal]. Available: <https://www.groupmap.com/map-templates/pestle-analysis/>. [Kasutatud 30 10 2021].
- [10] R. Pathak, „analyticssteps,“ 06 12 2020. [Võrgumaterjal]. Available: <https://www.analyticssteps.com/blogs/what-pestle-analysis>. [Kasutatud 30 10 2021].
- [11] R. Pavlov, „Poliitika ja strateegia,“ %1 *Organisatsiooni käsiraamat*, EAS, 2004.
- [12] Balbix, [Võrgumaterjal]. Available: <https://www.balbix.com/insights/cyber-risk-heat-map/>. [Kasutatud 12 12 2021].
- [13] S. Gierej, „The Framework of Business Model in the Context of Industrial Internet of Things,“ *Procedia Engineering*, kd. 182, p. 786, 2017.
- [14] K. A. K. Tomasz Miaskiewicz, „Personas and user-centered design:How can personas benefit productdesign processes?,“ *Design studies*, kd. 32, nr 5, pp. 417-430, 2011.
- [15] G. Witt, Writing Effective Business Rules, Morgan Kaufmann, 2012.
- [16] International Institute of Business Alanlysis, BABOK v3, Toronto: International Institute of Business Analysis, 2016.
- [17] C. Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Pearson, 2004.
- [18] S. X. S. Narasimha Bolloju, „Benefits of supplementing use case narratives with

- activity diagrams—An,” *The Journal of Systems and Software*, kd. 85, pp. 2182-2191, 2012.
- [19] P. Vliegthard, „Inveskills,” Inveskills, 19 02 2021. [Võrgumaterjal]. Available: <https://www.inveskills.com/bpmn/bpmn-vs-uml/>. [Kasutatud 24 10 2012].
- [20] Lucidchart, „Lucidchart,” [Võrgumaterjal]. Available: <https://www.lucidchart.com/pages/er-diagrams>. [Kasutatud 24 10 2021].
- [21] D. Taylor, „Guru99,” Guru99, 7 10 2021. [Võrgumaterjal]. Available: <https://www.guru99.com/data-modelling-conceptual-logical.html>. [Kasutatud 24 10 2021].
- [22] tutorialspoint, „tutorialspoint,” [Võrgumaterjal]. Available: https://www.tutorialspoint.com/uml/uml_statechart_diagram.htm. [Kasutatud 24 10 2021].
- [23] R. S. Z. E. R. Andy Effendi, „Adjustment Factor for Use Case Point Software Effort Estimation,” *Procedia computer science*, kd. 157, pp. 691-698, 2019.
- [24] tutorialspoint, „tutorialspoint,” [Võrgumaterjal]. Available: https://www.tutorialspoint.com/estimation_techniques/estimation_techniques_use_case_points.htm. [Kasutatud 30 10 2021].
- [25] IEEE, SWEBOK - Guide to the Software Engineering Body of Knowledge V3.0.
- [26] T. Y. S. Rikke Friis Dam, „Interaction Design Foundation,” 2020. [Võrgumaterjal]. Available: <https://www.interaction-design.org/literature/article/design-thinking-get-started-with-prototyping>. [Kasutatud 31 10 2021].
- [27] UXPin, „Studio by UXPin,” [Võrgumaterjal]. Available: <https://www.uxpin.com/studio/blog/what-is-a-prototype-a-guide-to-functional-ux/>. [Kasutatud 31 10 2021].
- [28] Team Asana, „Asana,” 04 10 2021. [Võrgumaterjal]. Available: <https://asana.com/resources/implementation-plan>. [Kasutatud 20 12 2021].
- [29] L. Noonan, „Do budgeting apps really work?,” *Financial Times*, 07 10 2019. [Võrgumaterjal]. Available: <https://www.ft.com/content/6dff9670-cf26-11e9-b018-ca4456540ea6>. [Kasutatud 13 03 2021].

Lisa 1 - Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Siiri Kliimson

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Isiklike finantstegevuste jälgimise rakenduse kavandamine“, mille juhendaja on Nadežda Furs
 - 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.
4. Magistritööle kehtib juurdepääsupiirang kuni 1. jaanuar 2027.

04.01.2022

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 loomingu 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.

