

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Kärt Raidmaa 176485

# **AJA MÄRKIMISE SÜSTEEMI TIMETRACKER ANALÜÜS JA DISAIN**

Magistritöö

Juhendaja : Paul Leis

PhD

Kaasjuhendaja: Margus Püüa

Dipl. Insener

Tallinn 2018

## **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: Kärt Raidmaa

14.05.2019

## **Annotatsioon**

Käesoleva magistritöö „Aja märkimise süsteemi Timetracker analüüs ja disain“ eesmärgiks on Eesti tarkvaraarendusettevõtte poolt loodud olemasoleva tööaja arvestamise süsteemi hetkeseisu analüüsida, kaardistada kasutajate vajadused ning leida ettevõttele kõige kasumlikum viis nii süsteemi parandada, samuti uurida sobitavust teistele ettevõtetele. Antud töös vaadeldakse täpsemalt tänast süsteemi, tööaja arvestamise süsteemi, kasutajate vajadusi tööaja märkimisel ja arvestamisel, kasutatud metoodikat, paranduskohti ning võimalike süsteemi lahendeid.

Antud magistritöö teema sai valitud ettevõtte huve silmas pidades ning olles küsimuse ees, kas jätta antud süsteem ettevõttesiseseks kasutuseks või liikuda suunas, mis võimaldaks tulevikus lahendust pakkuda ka teistele ettevõtetele. Antud vaatest lähtuvalt tuli vajadus mõista turgu, aja märkimise süsteemi kasutajakogemust, olemasoleva süsteemi seisust, paranduskohti ja uusi väärtust loovaid funktsionaalsusi, et pakkuda välja uue süsteemi edasised võimalikud tegevused ning disain.

Analüüsima ning leidmaks uusi võimalikke ja ettevõttele kasulikke süsteemi parandusi ja lahendusi, viidi läbi nii üldine tööaja arvestamise turu-uuring kui ka kasutajakogemuse uuring tänaste kasutajatega, vastutavate osapoolte ja võimalike uute kasutajatega.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 45 leheküljel, 7 peatükki, 10 joonist, 2 tabelit.

## **Abstract**

### **Analysis and Design of a Work Time Tracking System Timetracker**

The main purpose of the thesis „Analysis and design of a work time tracking system Timetracker“ is to analyse the state of today's system, map out the needs of users and to find the best solution for the new and shared system, keeping in mind the company's interests.

To design the new and improved system an analysis of today's application, competitive time tracking market analysis, user experience research and overview of possible usability fixes and architectural solutions was done in the context of this thesis.

In this day and age where „time is money“ and businesses track more and more work and hours done by employees, solutions need to give extra value to the end users and meet their needs. In competitive market basic solutions tend not to attract customers.

Despite the highly competitive marketplace there are still companies who have not found the time tracking solution that would fit them. Systems out there are either too big and complex, too expensive or do not fit the requirements of a client. Reasons mentioned are why many development companies tend to develop their own solutions. One such company is the software company which time tracking system Timetracker is under view in this thesis.

To realize the state of today's system, the needs of the users, both today's users and possible new users, to analyse the possible solutions for reengineering or fixing the application, is the reason why this thesis was written. It is to give an overview for the company in order to be the basis for the development and support of new and improved system that could be shared between different companies and not be in use only in the company who developed Timetracker.

The thesis is in Estonian and contains 45 pages of text, 7 chapters, 10 figures, 2 tables.

## Lühendite ja mõistete sõnastik

<i>Admin</i>	Isik, kellel on süsteemis laiemad tegevusõigused. Isik, kes saab hallata süsteemi.
Angular 1 ehk AngularJS	JavaScript skriptimise keelel põhinev vabavaraline front-end raamistik.
<i>Back-end</i>	Äriloogika ja andmete suhtluse kiht rakenduses
<i>Design thinking</i>	Disaini lähenemine, mis pakub probleemide lahendamisele lahenduse põhist lähenemist.
<i>Drag and drop</i>	Tõmbamise teel ekraanil konteksti liigutamine või paiknemise muutmine vastavalt vajadusele
<i>Front-end</i>	Kasutajaliides, mis pakub suhtlust kasutaja ja tagatausta vahel
Füüsiline masin	Riistvaral põhinev seadeldis, tagamaks vajalik arvutiga töötamiseks ja süsteemi jooksmiseks.
Kasutajakogemus ehk UX ( <i>user experience</i> )	Inimese ja arvuti vaheline suhtlus. Subjektiiivne isiku tunnetus ja arvamus süsteemist.
Kontekstuaalne uuring ehk <i>contextual research</i>	Kasutajakeskse disaini uuringute meetod, kus vaadeldakse kasutaja käitumist temale omases keskkonnas.
<i>Legacy</i>	Käibel mitte olev, pärand vanast
Microsoft Azure AD ehk konto	Pilvepõhine ligipääsu ja tuvastamise haldusteenus
<i>Multi-tenant</i>	Arhitektuur, kus üks rakenduse eksemplar on kasutusel mitme ettevõtte vahel.
<i>Native</i> rakendus	Tarkvara programm, mis on loodud toimimaks kindlal platvormil või seades, kasutades seadepõhist tarkvara ja riistvara.
Nginx	Veebi server, mis parandab turvalisust ning parandab rakenduse kättesaadavust veebis.
Prototüüp	Varajane näidis või mudel võimalikust lahenduse disainist.
<i>Single-tenant</i>	Arhitektuur, kus üks rakenduse eksemplar on korraga kasutuses vaid ühe ettevõtte jaoks.
Spring Framework	Java platvormi jaoks mõeldud rakenduse raamistik.
<i>Tag</i>	Märksõna, mis iseloomustab ja on seotud süsteemi sisestatud tööga
Timetracker	Antud töös käsitletav tööaja märkimise ja arvestamise süsteem

*Typescript*

Vabavaraline programmeerimiskeel, mis võimaldab luua JavaScripti rakendusi.

*User*

Isik, kellel on süsteemi kasutamise tavaõigused.

## Sisukord

1 Sissejuhatus .....	11
2 Timetrackeri ülevaade .....	13
2.1 Praegune lahendus .....	13
2.1.1 Rollid .....	14
2.1.2 Reeglid ja nõuded .....	14
2.1.3 Tegevused .....	15
2.1.4 Süsteemi ülevaade .....	21
2.2 Timetrackeri hetkeseis .....	22
2.2.1 Raamistikud .....	23
2.2.2 Süsteemi funktsionaalsused ja kasutatavus .....	25
3 Ülevaade aja märkimise süsteemidest .....	26
3.1 Meetodid .....	26
3.2 Aja märkimise süsteemid Eesti turul .....	29
3.3 Turu kokkuvõte.....	31
4 Süsteemi disaini meetodika .....	32
4.1 Avastamine .....	33
4.2 Defineerimine .....	33
4.3 Kujutlemine .....	34
5 Aja märkimise süsteemi kasutajakogemuse analüüs .....	35
5.1 Kasutajakogemuse uuring.....	35
5.1.1 Plaan .....	37
5.1.2 Tulemid.....	38
5.2 Kasutus ja kogemus .....	40
6 Võimalike lahenduste analüüs .....	42
6.1 Visioon ja eesmärgid .....	42
6.2 Reeglid ja nõuded .....	44
6.3 Süsteemi lahendid.....	47
6.3.1 Raamistiku uuendus.....	47
6.3.2 Jagatav süsteem .....	48

6.4 Disain.....	50
6.4.1 Tegevused ja süsteemi disain .....	50
6.4.2 Kasutajaliides.....	54
7 Kokkuvõte .....	55



## Jooniste loetelu

Joonis 1. Timetrackeri kasutuslugude diagramm. ....	16
Joonis 2. Timetracker süsteemi infrastruktuur. ....	21
Joonis 3. Timetrackeri kontseptuaalne klassiskeem. ....	22
Joonis 4. Disainimõtlemise protsessi sammud. ....	32
Joonis 5. Kasutajakogemuse uuringu protsess (Erin Sanders). ....	36
Joonis 6. <i>Single-Tenant</i> ja <i>Multi-Tenant</i> arhitektuur. ....	49
Joonis 7. Admin kasutaja kasutusmalliskeem. ....	51
Joonis 8. User kasutaja kasutusmalliskeem. ....	52
Joonis 9. Timetracker 2.0 klassiskeem. ....	53
Joonis 10. Timetracker 2.0 kalendri vaate prototüüp. ....	54

## **Tabelite loetelu**

Tabel 1. Aja märkimise meetodite võrdlus.....	28
Tabel 2. Timetracker süsteemi IT SWOT. ....	43

# 1 Sissejuhatus

Ajalooliselt on kasutatud aja märkimist kui kõige lihtsamat meetodit mõõtmaks tööd ning arvutamaks töötasusid. Juba iidse Egiptuses seati Hammurapi seaduse<sup>1</sup> abil miinimum koormused töötajatele. Osasid arvestati päevades, kuudes ja aastates; osadele tasuti tehtud töö pealt. Tasud olenesid ka töö tüübist: arsti tasu oli kõrgem kui tavatöötaja. [1]

Ajaga ja tehnoloogiate arenguga on tööaja märkimise viisid ning vahendid arenenud märkimisväärselt ning muutunud paljudes eluvaldkondades nii lahutamatuks kui ka vajalikeks osadeks palkade arvestuses ja klientidega arveldamises. Benjamin Franklini lause „aeg on raha“ kirjeldab ideaalselt töötaja ja tööandja vahelist suhet. See pole aga alati nii olnud: detailse tööaja märkimise ajalugu algab 20. sajandi alguses, kui tutvustati paberikandjal ajaraamatuid Reginald Heber Smith poolt. Esimesena võeti ajaraamatud kasutusse advokaatide seas. Täna kasutatakse ajamärkimist ehituses, arhitektuuris, õigusteaduses, infotehnoloogia teenustes ja mitmetes teistes valdkondades. Paberil ajaraamatud on asendatud elektrooniliste süsteemidega, mis automatiseerivad aja jälgimist, lihtsustavad arveldamist ning omavad ka mitmeid lisafunktsionaalsusi, mis võimaldavad ettevõtete juhtidel koguda olulist äri seotud andmestikku [1].

Ajaarvestamise süsteeme eri meetoditega ja eri valdkondadele tuleb aina enam juurde. Ajaarvestamise nõudluse kasvu tõttu on tegemist pidevalt kasvava turuga. 14. mai 2019 anti Euroopa kohtu poolt välja pressiteade, milles toodi välja kohtuhagi, kus kirjeldatakse vajalikkust aja märkimise süsteemid teha tööandjatele kohustuslikuks, mõõtmaks töötajate tehtud töötunde [2].

Täna on aja arvestamise süsteemid juba paljudes ettevõtetes muutunud lahutamatuks ja kohustuslikeks osadeks. Üheks selliseks ettevõtteks on ka antud magistritöös vaadeldav tarkvaraarendusettevõtte, kes on loonud 2013. aastal oma süsteemi. Tegemist on aja arvestamiseks, märkimiseks ja jälgimiseks mõeldud süsteemiga, mis täna võimaldab

---

<sup>1</sup> Kuningas Hammurapi poolt paika seatud Vana-Babüloonia seadustik, mille eesmärgiks oli saavutada üldine heaolu ja valitseda õiglaselt.

kasutamist vaid ettevõttes endas. Tulenevalt ettevõtte huvidest antud süsteem teha kättesaadavaks ka teistele ettevõtetele, tuleb mõista kas ja kuidas tuleb ümber disainida tänane süsteem.

Lähtuvalt ettevõtte soovidest ning ajaarvestamise vajalikkuse kasvamisest, on antud töö eesmärgiks luua ülevaade kasutuses olevast süsteemist, selle seisust ning pakkuda välja parandusettepanekud ja võimalikud disaini lahendid uue jagatava süsteemi tarvis. Töötamaks välja võimalikud süsteemi disaini lahendid, on töös kirjeldatud olemasolevat süsteemi, vaadeldud turul olevaid süsteeme ja uuritud nii töötajate kui ka võimalike klientide kasutajakogemust.

Töö koosneb viiest osast. Töö esimeses osas (peatükk 2) analüüsitakse olemasolevat süsteemi, antakse ülevaade tegevustes ning süsteemi puudustest nii tehnoloogia kui ka kasutuskogemuse vaatest.

Teises osas (peatükk 3) vaadeldakse täpsemalt tööaja märkimise süsteeme, nende vajadust, meetodeid ning Eesti turul olevaid lahendusi.

Kolmandas töö osas (peatükk **Error! Reference source not found.**) kirjeldatakse töös valitud uue süsteemi analüüsil aluseks võetud meetodikat. Tuuakse välja üldine töös kasutatud protsessi kirjeldus ning töös läbi viidud sammud.

Neljandas osas (peatükk 5) tuuakse välja kasutajakogemuse analüüsi tulemused süsteemi kasutajakogemuse uuringust. Sellest lähtuvalt analüüsitakse olemasolevat süsteemi ning tuuakse välja kaardistus paranduskohtadest ning kasutajate üldistest vajadustest.

Viiendas osas (peatükk 6) analüüsitakse töös käsitletava tööaja arvestamise süsteemi uue parandatud versiooni võimalikke lahendusi, tulenevalt kasutajakogemuse uuringus välja tulnud ettevõtte soovidest ja kasutajate vajadustest lisaks turu uuringule ja olemasoleva süsteemi analüüsile.

## 2 Timetrackeri ülevaade

Timetracker on tööaja märkimise süsteem Eesti tarkvaraarendusettevõtte töötajatele, mis on loodud 2013. aastal. Ettevõtte tegevjuhi sõnul ei olnud sel ajal turul sobivat tööaja märkimise vahendit, mis oleks olnud kergesti kasutatav ning kalendrisse märkimise võimekusega. Turul olevad lahendused olid kohmakad ning kallite kasutuslitsentsidega. Ettevõtte tegevjuhi arvates oli alustaval tarkvaraarendusettevõttel nii odavam kui ka lihtsam, arvestades olemasolevat inimressurssi ja kompetentsi, ise luua endale sobiv tööaja märkimise lahendus.

Loodud süsteem võimaldab ettevõtte töötajatel kalendri vaates oma tööaja kasutuse märkimist, koormuse jälgimist ning ajakulu mõõtmist. Ettevõttel võimaldab antud lahendus jälgida töötajate koormuse täituvust, arveldada klientide projektidele kulunud töötunde ja vaadelda sisemistele töödele kulunud tundide kogust.

Antud peatükis vaadeldakse täpsemalt tänast lahendust – reegleid, nõudeid, tegevusi, süsteemi, kasutuses olevaid tehnoloogiaid, puudusi ja üldist kasutatavust.

### 2.1 Praegune lahendus

Timetracker süsteemi viimane versiooni uuendus tehti süsteemi arendaja sõnul aastal 2016. Kuna tegemist on süsteemiga, mida kasutatakse vaid ettevõtte siseselt ning täidab täna oma peamisi eesmärke – töötundide märkimine ja arvestamine – pole ka rõhku pandud süsteemi arendusse. Peale 2016. aastat on arendajate sõnul tehtud süsteemi vaid väiksemaid parandusi funktsionaalsusi piiravate vigade korral.

2013. aastal loodud süsteem on vananevate ehk *legacy* tehnoloogiate peal nagu Spring Framework 4 *back-end* poolel ja Angular 1 *front-end* poolel. Süsteemi sisenemiseks on kasutuses Microsoft Azure, mis võimaldab sisenemist vaid vastava ettevõtte Azure kasutajakonto korral. Spring Framework puhul on tegemist raamistikuga, mille tugi lõpetatakse 2020 aastal ja Angular 1 ehk AngularJS tugi lõpetati 2018 aastal. Samuti on

tegemist tehnoloogiatega, mis ei võimalda süsteemi kergesti kasutusse andmist teistele, mitte ettevõttega seotud isikutele või ettevõtetele.

Järgnevalt vaadeldakse täpsemalt süsteemis kasutatavat sõnastikku, tegutsejaid, tegevusi ning reegleid. Tegemist on süsteemiga millel puudub dokumentatsioon ning kõik tänase süsteemi sisend tulenev igapäevasest otsesest aja märkimise süsteemi kasutusest.

### 2.1.1 Rollid

Timetracker süsteemis saab kasutaja olla kahes eri rollis. Rollid jagunevad järgmiselt:

- Admin
- User

**Admin** kasutaja on laiemate õigustega isik ehk tiimijuht või juht, kes saab lisaks üldistele tegevustele anda või eemaldada õigusi kasutajatelt, lisada projekte ja kliente või neid sulgeda, alustada või sulgeda ajamärkimise perioode kalendris, võtta välja raporteid teiste kasutajate, kõikide projektide ja klientide tööde kohta.

**User** tüüpi kasutaja ehk tavakasutaja ehk töötaja on isik, kes on piiratud õigustega kasutaja. Tegemist on kasutajaga, kes saab kasutada kõiki süsteemi põhifunktsionaalsusi märkimaks oma tööaega või oma tehtud tööde tundide ülevaate saamiseks.

### 2.1.2 Reeglid ja nõuded

Timetrackeri süsteemil puudub ettevõtte arendajate ja tegevjuhi sõnul dokumentatsioon, kus oleks toodud välja reeglid ja nõuded. Ettevõtte poolse tööaja märkimise süsteemi kunagise projektijuhi sõnul toimus reeglite ja nõuete suusõnaline edasi andmine ning ei pandud rõhku dokumentatsiooni loomisele.

Järgnevad reeglid ja nõuded on välja toodud vastavalt eksisteerivale lahendusele, selle analüüsist ning kasutusest kujunenud arusaamast ettevõtte töötajate poolt.

Süsteemis saab olla isik kas admin või user rollis. Süsteemi saab kasutada vaid isik, kellel on ettevõtte Microsoft Azure konto. Süsteemi sisenemine toimub Azure konto meiliaadressiga ja parooliga. Isik saab süsteemis igal ajal muuta aja märkimise süsteemi sisenemiseks kasutatavat parooli, kuid mis on teatud ajaintervalli tagant kohustuslik muuta.

Admin tüüpi isiku puhul on tegemist laiendus süsteemi õigusega isikutega , kes saab hallata süsteemis õiguseid, rolle, positsioone, kliente, projekte, töid, koormusi ja vabu päevi.

User ja admin tüüpi isikud saavad märkida süsteemis tööaegsid kalendrisse, panna tööaegsid salvestama, planeerida kalendrisse oma töid, vaadata ja planeerida teiste töötajate planeeritud töid, võtta välja ülevaatlikke raporteid, vaadata tööde ülevaadet, vaadata kulunud tunde ning tundide seisu päevas, nädalas ja kuus, lisada ja muuta töid, sulgeda töid, märkida töödele hinnanguid, vaadata töödele kulunud tunde, märkida töödele kommentaare, lisada teavitusi tuletamiseks meelde kalendri täitmist ja logida välja süsteemist.

Admin tüüpi isik saab võtta välja raporteid nii enda kui teiste töötajate kohta ning kõikide projektide kohta. User tüüpi isik saab võtta raporteid välja vaid enda tööde kohta ja projektide kohta, millele õigused on antud admin poolt.

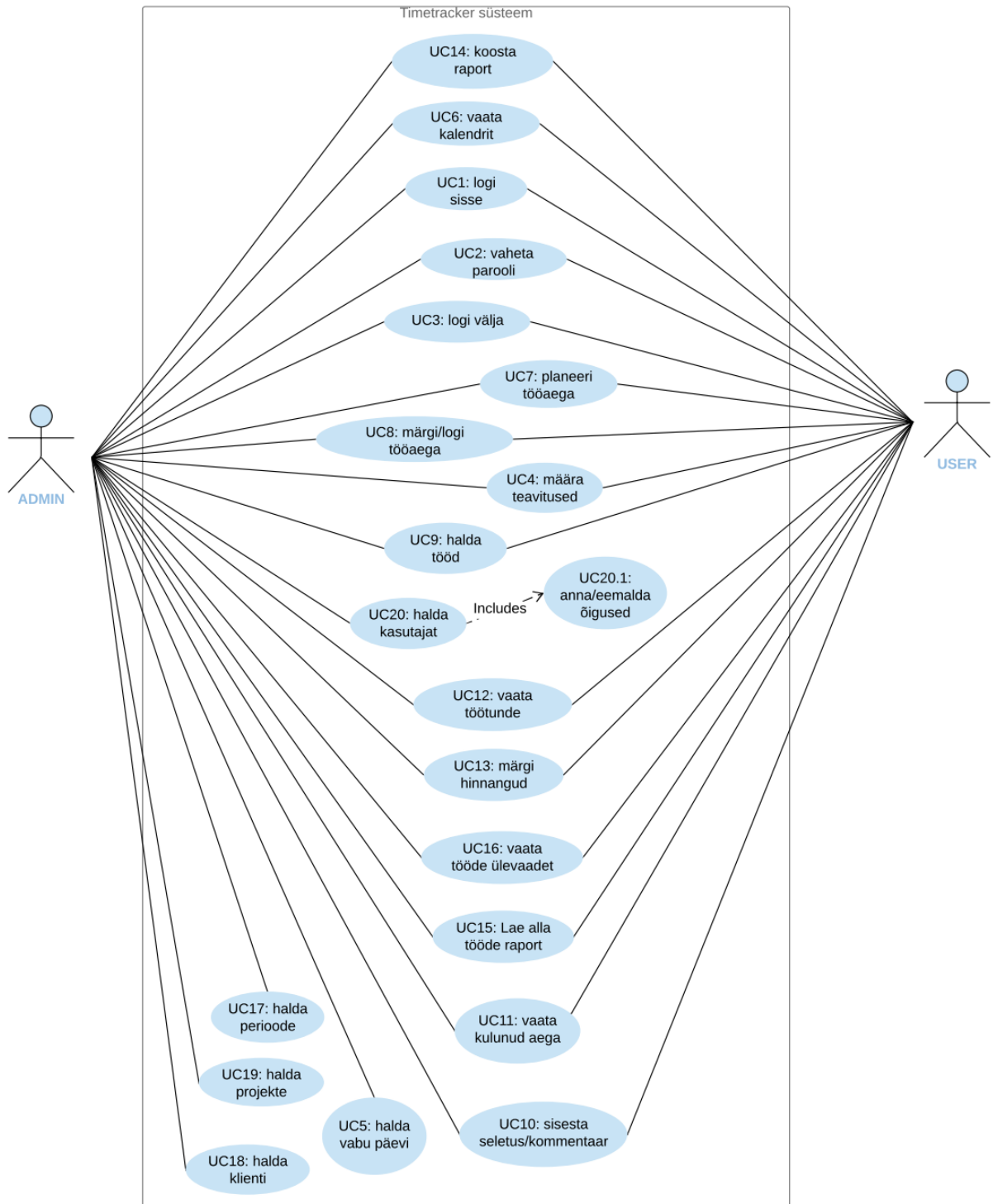
Süsteemi puhul on tegemist veebirakendusega, mis ei kohane vastavalt nutiseadmete ekraani suurustele ehk ei skaleeru. Kasutava ekraani suurus peab olema suurem kui 1060x600 pikslit, et kasutajaliides jääks korrektseks ja kasutatavaks.

Süsteem kuvab vastavalt isiku koormuseks määratud välja tavatunnid päevas, nädalas ja kuus. Vastavalt kasutaja sisestatud tundidele ja määratud koormusele näidatakse töötundide täituvust süsteemis.

Süsteemi veebilehte esilehe avanemine, süsteemi sisenemine, välja logimine ja muud funktsionaalsused annavad vastuse 3-6 sekundi jooksul.

### **2.1.3 Tegevused**

Aja märkimise süsteem võimaldab suurel määral erinevaid tööaja märkimiseks ja ülevaate saamiseks vajalikke tegevusi. Läbi kasutuslugude diagrammi (Joonis 1) ja kasutuslugude täpsemate kirjelduste nii admin kui kasutaja rollis, süsteemi igapäevasest kasutamisest tulenedes, on ära kirjeldatud süsteemis tehtavad võimalikud tegevused.



Joonis 1. Timetrackeri kasutuslugude diagramm (autori loodud).

Kasutuslugude kirjeldused on välja toodud mitteametlikul viisil ehk lühikeste tekstiliste lugudena süsteemi kasutamisest teatud kasutaja vaatest, saavutamaks eesmärki [3]. Valitud on antud töö osas mitteametlik viis andmaks üldist ülevaadet süsteemi



tegevustest. Lisas 1 on näitena toodud välja raporti koostamise (UC14) *fully dressed* ehk detailsema struktuuriga kasutajaloo kirjeldus.

#### **Logi sisse (UC1):**

User või admin, kes omab tarkvaraarendusettevõtte isiklikku Microsoft Azure kontot saab logida süsteemi sisse oma Azure konto meiliaadressi ja parooliga.

#### **Vaheta parooli (UC2):**

Isik, vaatamata rollile, saavad muuta oma salasõna, millega süsteemis sisse logivad. Muutmaks salasõna ehk parooli tuleb sisestada vana parool, uus parool ning uue parooli kinnitamiseks veelkord uus parool. Peale uue parooli sisestamist salvestatakse süsteemi kasutajale uus parool.

#### **Logi välja (UC3):**

Isikud saavad aja märkimise süsteemist välja logida vajutades kasutaja alt avanevas menüüs lingile „Logi välja“. Kasutaja logitakse süsteemist välja ning talle kuvatakse süsteemi sisenemise vaadet.

#### **Määra teavitused (UC4):**

Isikul on võimalus määrata omale teavitused, tuletamaks meelde tööde täitmist nii igapäevaselt, nädala kaupa kui ka kuu kaupa. Valides teavituse viisi, saab vastavalt teavituse perioodi valikule kasutaja vastavasisulise meili tuletamaks meelde kalendri täitmist.

#### **Halda vabu päevi (UC5):**

Admin tüüpi kasutaja saab sisestada kui sulgeda vabu päevi. Uue vaba päeva saab sisestada süsteemi valides vaba päeva sisestamise vormil kalendrist vastava päev. Sisestatud vabasid päevasid kuvatakse vormi all. Kustutamiseks admin vajutab päeva taga olevale sulgemise nupule, mille järel süsteem muudab päeva tundide arvestuse.

#### **Vaata kalendrit (UC6):**

Isikud saavad sisse logides ülevaate jooksva nädala kalendri vaatest. Isik saab vaadata kalendrit nädalaste perioodidena, töönädala kaup või terve nädala kaupa. Kalendris näeb

isiku märgitud tehtud töid, nende nimetuse ja pikkusega koos. Isik näeb kalendris planeeritud töid ja tehtud töid nii enda vaates kui ka teiste isikute vaates.

#### **Planeeri tööaega (UC7):**

Isiku enda tööaja või isiku poolt teiste isikute tööaja perioodi planeerimine kalendri vaates. Tegemist on funktsionaalsusega, mis annab nii ülevaate isiku planeeritud töödest kui ka võimaldab planeerida kindla isiku tööaega. Tööaeg ei logita tehtud töötundidesse kuni isik ise ei kinnita ära planeeringus olevat tööd. Kinnitamise järel salvestatakse planeeritud töö isiku tehtud tööde kalendrisse ja arvestatakse tehtud töötundide arvestusse juurde.

#### **Märgi/logi tööaega (UC8):**

Süsteemi kasutajad märgivad aega süsteemi kolmel eri viisil, kas *drag and drop* viisil tööd tirides kalendrisse õigele päevale ja õigesse ajahetke, õige kulunud ajalise pikkusega, tööaja salvestamisel jooksva töö salvestamise lõpetamise järgselt või planeeringu vaates töö logimine oma tööde kalendrisse, valides töö peale vajutades avanevast modaalist „Logi aeg“. Peale töö märkimist lisatakse tehtud töö tunnid isiku töötundide arvestusse juurde.

#### **Halda tööd (UC9):**

Töid saavad sisestada, muuta ja sulgeda kõik isikud klientide ja projektide alla, millele õigused on antud. Uue töö sisestamiseks tuleb isikul valida „Loo uus töö“, peale mida avaneb vorm. Isikul tuleb lisada töö nimi/kirjeldus (kohustuslik väli), valida projektide loetelust süsteemi sisestatud ja õigused antud projekt, mille alla töö läheb ning töö tüüp (arendus, hooldus, muu või viga). Peale töö sisestamist salvestatakse töö süsteemi. Süsteemi sisestatud töö nime, projekti, tüüpi saab muuta senikaua kuni töö on süsteemis. Töö mitte kasutuse korral saab töö sulgeda, mille järel süsteemis muudetakse töö mitteaktiivseks.

#### **Sisesta seletus/kommentaari (UC10):**

Kalendrisse märgitud tehtud tööde alla on isikutel võimalik sisestada kommentaare, tehtud töö kirjeldusi. Kommentaari sisestamiseks tuleb vajutada kalendrisse märgitud kindlale tööle, mille järel avaneb modaal. Hüpinkaknas on teksti lahter sisestamiseks töö

kirjeldus. Kommentaari salvestamiseks süsteemi tuleb peale kirjelduse sisestamist vajutada hüpikaknas „Salvesta“ nupule. Kommenteere kuvatakse tööle vajutades avanevas hüpikaknas, raportites kui ka tööde ülevaates.

#### **Vaata kulunud aega (UC11):**

Isikutel on võimalik vaadata kindla töö alla kulunud tunde. Kulunud tundide vaatamiseks tuleb valida soovitud töö vasakul kuvatavast tööde loetelust või otsingu abil otsida soovitud töö ning seejärel vajutada töö muutmise peale. Avaneb töö muutmise vorm, kus kuvatakse kulunud aega kokku. Vaatamaks täpsemalt kulunud aegasid, saab kasutaja valida „Näita kõiki“, mille järel avaneb ajakulu täpsem ülevaade. Täpsemas ülevaates on vastavalt rollidele eraldi analüüsi, testimise, programmeerimise ja kogu kulunud aja ülevaade tundides.

#### **Vaata töötunde (UC12):**

Isikud saavad süsteemis vaadata ülevaadet nii päevas, nädalas kui kuus ette nähtud tunde ning tehtud töötunde. Töötundide ülevaade antakse kasutajale kalendri vaates. Kasutajale kuvatakse temale vastava koormusega tavatunnid päevas, nädalas ja kuus, isiku tehtud tunnid ning tehtud ületunnid või alatunnid päevas, nädalas ja kuus.

#### **Märgi hinnangud (UC13):**

Kasutajad, vaatamata rollile, saavad süsteemi sisestatud töödele määrata hinnanguid. Eraldi saab sisestada analüüsi, arenduse, testimise, puhvri ja/või töö täieliku hinnangu. Süsteemi sisestatud hinnangute järel hakatakse kuvama töö all hinnangu täituvust vastavalt kulunud ajale. Sisestatud rollipõhised hinnangud arvutatakse süsteemis kokku kui koguhinnanguna. Juhul kui hinnanguid sisestatud pole, ei kuvata koguhinnangut ning ei kuvata täituvuse piiri.

#### **Koosta raport (UC14):**

Admin tüüpi isikud saab luua raporteid enda, teiste või kõigi kasutajate kohta. Admin tüüpi isikud saab luua raporteid kõikide klientide ja projektide kohta. User tüüpi isikud saab luua raporteid vaid enda kohta, endaga seotud klientide ja projektide kohta, millele õigused on antud. Raportite genereerimisel saab valida ajaperioodi, mille kohta soovitakse raport genereerida, raporti tüüp (kasutaja järgi grupeeritud, finants raport,

kasutaja ja töö järgi grupeeritud või kuu, kasutaja ja projekti järgi grupeeritud), kliendi, projekti, töö ja kasutaja. Kõik valikud on valikulised (ei ole kohustuslikud) ja vastavalt valikutele pannakse süsteemi poolt kokku raport. Raportis kuvatakse info tabelina.

#### **Lae alla tööde raport (UC15):**

Peale raporti genereerimist on võimalik süsteemi poolt kokku pandud raport alla tõmmata kui CSV formaadis vajutades lingile „Lae alla CSV“.

#### **Vaata tööde ülevaadet (UC16):**

Kasutajad saavad luua süsteemis tööde ülevaated sisestades töö nimetuse ning valides filtritest süsteemis oleva projekti, kliendi, töö tüübi või staatuse. Isik ei pea valima ühtegi filtri valikut või võib valida ühe kuni mitu valikut. Mitte sisestades nime või valides filtreid, kuvatakse kasutajale kõikide süsteemis olevate tööde loetelu. Tööde ülevaates kuvatakse kasutajale kõik sisestatud nimele ja filtri valikutele vastavate süsteemis olevate tööde nimetused, projekti, kliendi, tüübi, staatuse ja arveldamise viis (sisemine või mitte sisemine).

#### **Halda perioodi (UC17):**

Admin tüüpi isikud saavad avada ja sulgeda teatud ajaperioode kalendris tööaegade märkimiseks. Sulgemisel teatud periood, ei võimalda süsteemi enam kasutajatel antud ajaperioodi töötunde juurde lisada. Vajadusel saab admin juba sulgenud ajaperioode taasavada.

#### **Halda klienti (UC18):**

Admin tüüpi isik saab sisestada uusi kliente süsteemi ja sulgeda süsteemi lisatud kliente. Kliendile määratakse süsteemis sisestamisel nimi ja arveldamise tüüp ehk kas tegemist sisemise kuluga või arveldatava kliendiga. Sulgemise järel muudetakse süsteemis klient mitteaktiivseks. Vajadusel saab admin tüüpi isiku juba mitteaktiivseks muudetud kliendi taas aktiveerida.

#### **Halda projekte (UC19):**

Admin tüüpi kasutaja saab sisestada olemasolevate klientide alla uusi projekte ja süsteemi sisestatud projekte sulgeda. Süsteemi sisestatakse kliendi alla projekti nimi, projektile

määratakse värv ning ajaperioodi pikkus, mille kaupa arveldus ja ajamärkimine kinni pannakse süsteemis. Sulgemise järel määratakse projekt süsteemis mitteaktiivseks. Vajadusel saab admin juba sulgetud projekti taasavada süsteemis.

### **Halda kasutajat (UC20):**

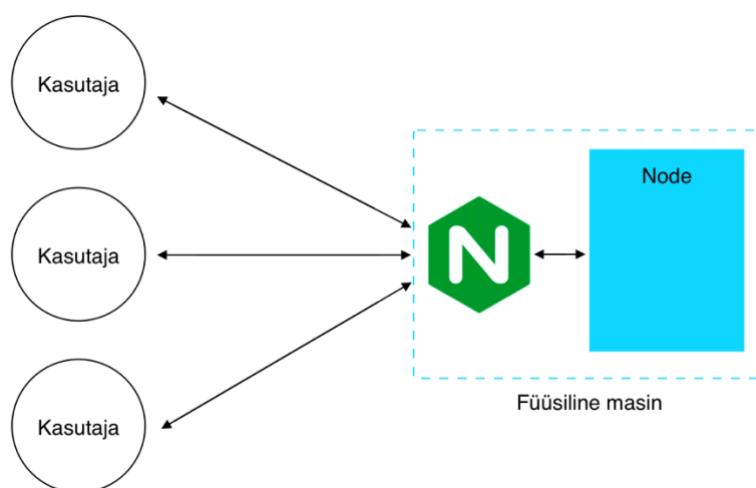
Admin tüüpi isik saab süsteemis hallata süsteemi sisestatud kasutajaid. Halduses saab muuta isiku õiguseid, positsiooni, rolle klientide projektides, koormusi ja seisu (aktiivne või mitteaktiivne).

### **Anna/eemalda õigused (UC20.1):**

Admin tüüpi kasutaja saab kasutajate halduses muuta isiku õiguseid. Admin tüüpi kasutaja saab anda või eemaldada admin õigused, ja ligipääse projektide töödele.

### **2.1.4 Süsteemi ülevaade**

Timetracker rakenduse puhul on tegemist *in house* süsteemiga ehk süsteemiga, kus andmeid hoiustatakse ettevõtte sisemises füüsilises masinas ning kasutajad pääsevad süsteemile ligi oma arvutitest. Kasutajatele on jagatud rakendus ja jagatud andmebaas, mis kõik on füüsilises masinas ning füüsilises masinas olev Nginx serverib *front-end*'i ja suunab päringud serveris olevasse privaatsesse *back-end*'i (Joonis 2).

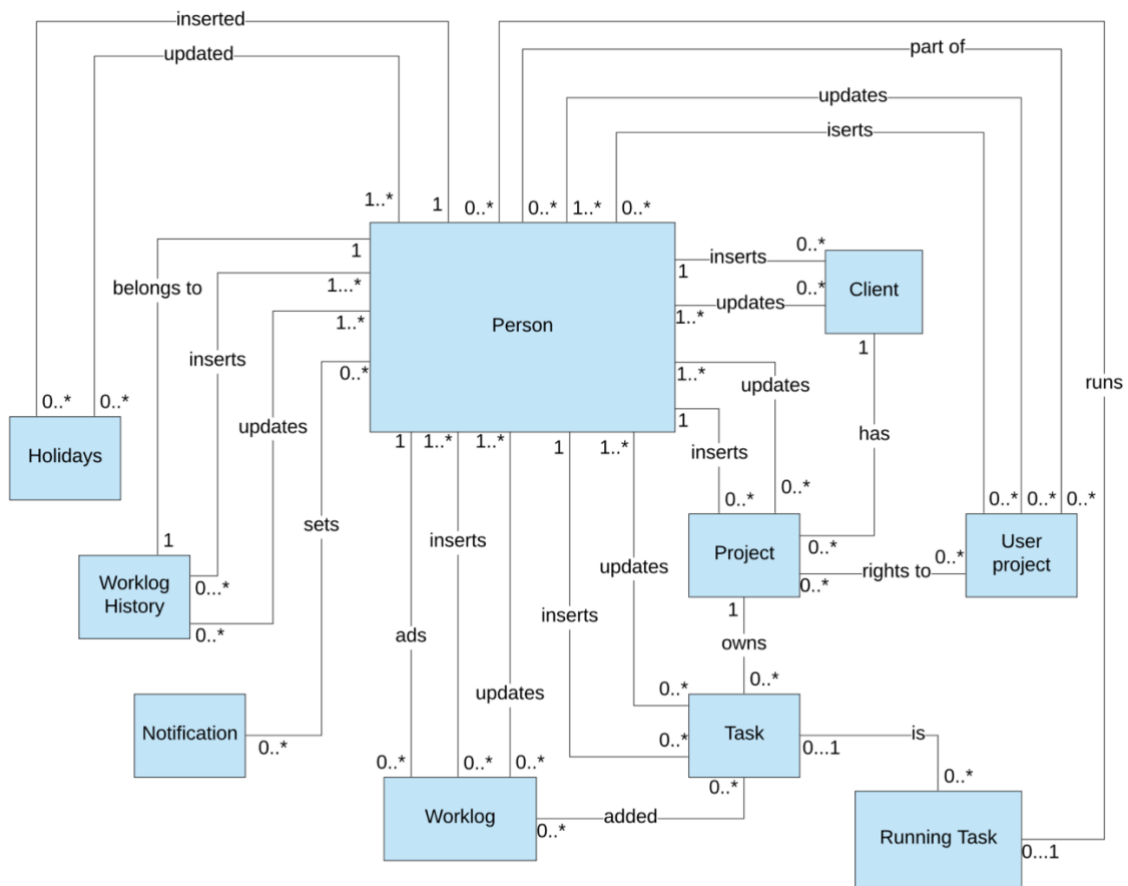


Joonis 2. Timetracker süsteemi infrastruktuur (autori loodud).

Timetracker aja märkimise süsteem kasutab PostgreSQL andmebaasi, mis on tänase lahenduse vaates süsteemi arendajate arvates väga korralik ja võimaldab ka tuleviku lahenduste vaates jätkata kasutamist. Tegemist on PostgreSQL puhul võimsa avatud

lähtekoodiga objekt-orienteeritud andmebaasi süsteemiga, mida on arendatud aktiivselt üle 30 aasta ning mida kasutatakse ülemaailmselt. PostgreSQL on teeninud ära oma tugeva maine usaldusväärsuses ja töövõimes [4].

Olemasoleva süsteemi objektid ja nendevahelised seosed on toodud välja üldise klassiskeemiga, andmaks ülevaate süsteemi elementidest ja suhtlusest (*class diagramm*, Joonis 3). Süsteemi tänase täpse PostgreSQL andmebaasi ERD diagrammi, kus on välja toodud kogu kogutav andmestik, leiab lisast 2.



Joonis 3. Timetrackeri kontseptuaalne klassiskeem (autori loodud).

## 2.2 Timetrackeri hetkeseis

Timetracker aja märkimise süsteem kasutab täna platvormi Spring Framework 4 ja Angular 1 (Angular JS), mida võib tänasega lugeda vanadeks ehk *legacy* tehnoloogiateks. Tänapäevase seisuga veel Spring Framework kodulehe järgi toetatakse *back-end* tehnoloogia Spring Framework 4 versiooni, kuid seda vaid 2020 aastani. Angular 1 puhul on tegemist

aga *front-end* tehnoloogiaga, mille võime lugeda kui raamistikuks, mida ei toetata enam. 2018 aastal algas Angular 1 kolme aastane ülemineku periood, mille ajal toetatakse vaid 1.7 versiooni ning suuremate turva murede korral ka 1.2 versiooni [5]. Tulenevalt platvormide mitte uuendamisest ja hooldamisest võivad tuleneda erinevad mured nagu turvaaugud, kiirus ja olemasolevate tegevuste ebakorrektnete töötamine.

Lisaks vananenud raamistikele on kasutuses oleva aja märkimise süsteemi miinusteks mitmete vajalike ja abistavate funktsionaalsuste ebaselgus ning olemasolu puudumine. Tänapäevased funktsionaalsused täidavad oma igapäevast ülesannet ettevõtte siseselt, märkimaks tööaega ja arveldamiseks tunde, kuid nõuab endiselt palju lisatööd, mõtlemist ning kasutuse osas juhendamist. Tänapäevase süsteemi lahendus võimaldab kasutust ainult ettevõttes endas ning süsteemi arendusele rõhku ei ole pandud viimased kaks aastat. Sellest tulenevalt on tegemist süsteemiga, millele on kogunenud tehnoloogiline võlg ning puudujäägid.

Mõistmaks tänast seisust ja mida tuleks tulevikus silmas pidada, on vaadeldud süsteemi hetkel kasutatavate tehnoloogiate ning funktsionaalsuste puuduseid.

### **2.2.1 Raamistikud**

Angular 1 ehk Angular JS puhul on tegemist avatud lähtekoodiga veebiraamistikuga, mis on loodud Google poolt 2009 aastal ja ametlikult välja andud 2010 aastal. Aastaid hiljem, 2014. aastal mõistsid AngularJS arendajad, et tegemist on liiga komplekse ja muresid täis raamistikuga ning otsus võeti vastu täielikult Angular JS ümber kirjutada. 2016 aastal anti välja Angular JS ümber kirjutatud versioon Angular 2. Uue raamistiku versiooni muudatused olid vastavuses kaasaegse veebi kasvavate nõuetega ja niivõrd radikaalsed, et ei võimaldanud enam lihtsat uuendamist Angular JS pealt Angular 2 peale. Alates Angular 2 välja andmisest 2016 on jäänud tuumalt toode samaks, kuid antud on välja mitmeid uuemaid parandustega versioone. Tänapäevaseks on väljas Angular 7 ja plaaniliselt 2020 aastal tuleb välja uus versioon, Angular 8 [6, 7].

Seoses AngularJS üleminekuperioodiga ning seisuga, kus enam ühtegi uut täiendust ei tehta ja parandatakse vaid kindlatele kriteeriumitele vastavaid muresid kuni 2020 aasta lõpuni, on rakendused olukorra ees, kus tuleb mõelda ümberkirjutamisele. AngularJS populaarsuse tõttu on tänapäevaseks tuhandeid rakendused samas situatsioonis nagu ka aja

märkimise süsteem Timetracker. Aeg on minna üle uutele raamistikele, mis tähendab aga ajakulu kui ka rahalist väljaminekut, kuid mis on suuremas vaates kasulikud [8, 7].

AngularJS pealt üle minek uuematele ja kaasaegsematele *front-end* raamistikele täna on mõistlik mitte ainult põhjusel, et AngularJS enam ei arendata ega toetata Google poolt vaid ka järgnevatel põhjustel:

- Vähem andmete kasutust ja kiirem käivitusaeg lisaks muudatuste kiiremale tuvastamisele ning kasutuslugude kaupa optimeerimisele.
- Angular 1 keerukas dokumentatsioon võrreldes uutega
- Komponentide keerukuse kasvamisel koodi loetavus, hallatavus ja käitlevus muutuvad valurikkaks.
- Uuemad raamistikud lubavad luua *native* rakendusi, mida Angular 1 ehk AngularJS ei võimalda.
- AngularJS ja Angular 2 on kinni raamistiku põhistes sõltuvuste vähendamises, kontrollerites, mallides. Esineb nii õppimiskõver kui ka rakenduste ehitamisel piirangud.
- AngularJS ei võimalda Typescript kasutamist [8].

Spring Framework puhul on tegemist palju kasutatava *back-end* raamistikuga Java jaoks, mis aga nõuab palju trafaretset ehk standardseid konfiguratsioone rakenduste loomisel. Arvestades tänapäeva tehnoloogia arengut ja kasvavate lahenduste nõudlusi, tuleks Spring puhul mõelda üle minekut Spring Boot peale. Spring Boot korral on tegemist Spring laiendusega, mis lahendab ära nii konfiguratsioonide kui ka paigaldamise automatiseerimise. Lisaks võimaldab Spring Boot kergemalt süsteemi uute elementide rakendamist [7]. Ajamärkimissüsteemi arendajate sõnul on Spring framework puhul hästi struktureeritud raamistikuga Java jaoks, kuid mis ei käi kaasas enam teatud kaasaja nõuetega. Timetrackeri puhul tuleks mõelda sõltuvuste uuendamisele ning tehnilisest võlast lahti saamisele liikudes üle Spring Boot 2 peale, tehes kood loetavamaks.



### **2.2.2 Süsteemi funktsionaalsused ja kasutatavus**

Vastavalt Timetrackeri aja märkimise süsteemi salvestatud andmestike pealt võib järeldada, et täna kasutajad ei kasuta taimeri funktsionaalsust tööaja logimiseks. Vähem kui veerand kasutajatest kasutavad hinnangute andmise funktsionaalsust ning enamus täidetud ajalogidest on täidetud 7 päeva jooksul (95%) sama päeva asemel (60%). Lisaks kasutatakse tööaja planeerimist nii enda kui teiste jaoks vähem kui 5% kasutajate poolt.

Väikestest näitudest tulenevalt võib eeldada süsteemi puhul, et tegemist on kasutajale mitte mugavate ja ebaselgete funktsionaalsustega. Eeldada ei saa, et tegemist funktsionaalsustega, mida vaja ei ole. Süsteemi kasutusest tulenevalt võib järeldada, et taimeri, hinnangute andmine ja vaatamine, täitmise meeldetuletused ning planeerimine vajavad rohkem tähelepanu ning analüüsi, muutmaks kasutajatele väärtust andvateks tegevusteks.

Lisaks ülal mainitud funktsionaalsustele on suurimaks kasutatavuse mureks tänases kaasaegses ja kiire tehnoloogia arenguga maailmas süsteemi nutiseadmetes ebamugav kasutamine tulenevalt rakenduse ekraani suurusele vastavalt kohanemise muredest. See piirab süsteemi kasutuskohti.

Tänaused puudused ei ole süsteemi kasutamist piiravad aspektid, kuid tegemist on süsteemiga, mis vajab kasutatavuse täpsemat analüüsi ning vajaduste täpsemat kaardistamist, likvideerimaks kõikvõimalikud probleemsed kohad ning pakkuda kasutajatele lisaväärtust andvat süsteemi.

### 3 Ülevaade aja märkimise süsteemidest

Otsides Google otsingumootori kaudu tööaja märkimise variante ja lahendusi, kuvatakse välja suur hulk erinevaid valikuid [1]. Turule on tekkinud hulganisti eri võimalustega ja eri valdkondadele mõeldud tarkvarasid tööle kulunud aja jälgimiseks. Auna suureneva nõudluse ja eri vajaduste tõttu kasutavad tarkvarad erisuguseid lahendusi, meetodeid ning omavad erinevaid funktsionaalsusi. Vaatamata suurele valikule eri süsteemidele, on suurim väljakutse leida ettevõttel või inimesel sobiv lahendus, sest iga inimene ja ettevõtte mõõdab erinevaid parameetreid [9].

Mõistmaks ajaarvestamise turgu, konkurentsi ja turu võimalusi, vaadeldakse ning võrreldakse järgmistes alampeatükkides erinevaid aja märkimise meetodeid ja ning turul olevaid süsteeme.

#### 3.1 Meetodid

Aja märkimiseks on tänapäeval kujunenud välja väga palju eri meetodeid tänu eri valdkondadele, kes kasutavad aja märkimist ühel või teisel viisil. Ennem kindla süsteemi valikut, tuleb mõista vajadusi ning leida sobivaim meetod töötundide arvestamiseks [1]. Tänu suurele valikule meetoditele leiab iga ettevõtte ja isik endale just sobiva.

Kõige algelisemate ja tänaseni kasutuses olevate ajamärkimise viisidena võib välja tuua Exceli, vihikud ja suusõnalise viisi. Suusõnalise korral küsitakse tööga seotud isikutelt eeldatav ajakulu ning seejärel arveldatakse vastavalt kokku arvatatud arvatava ajakulu järgi. Vihiku ehk paberile aja märkimise korral töötaja mõõdab oma tööaega kella või muu seadmega ning hiljem kirjutab ülesse. Exceli korral saab märkida vastavalt vajadusele kulunud aegasid projekti või kindla tööülesande alla, vaja on vaid teha vastavad veerud nagu näiteks kuupäev, tegevus, algusaeg ja lõppaeg [1, 10].

Antud tööaja märkimise algelistel lahendustel ei ole võimekusi, mida omavad aga spetsiaalsed ajamärkimiseks mõeldud tarkvarad [1]. Tegemist on endiselt väga kohmakate ja aeganõudvate viisidega.

Üheks vanimaks mooduseks, mis endiselt kasutuses, kuid kasutatavate tehnoloogiate poolelt muutunud, on *clock in/clock out* süsteem ehk tööle saabumise/töölt lahkumise registreerimise süsteem. Kunagiste paberkaartide augustamiste asemel on kasutusse võetud erinevad võimalused nagu helistamine tasuta numbrile, kasutades arvutit või spetsiaalset mobiilirakendust [1].

Sarnaselt eelnevale meetodile on kasutuses ka kaardisüsteemid, kus tõmmatakse tööle tulles ja töölt lahkudes spetsiaalsest aparaadist läbi oma kaart, kuid mida loetakse siiski eraldi meetodiks [1].

Kokku töötatud tundide arvestamiseks kasutatakse lihtsaid taimeriga süsteeme. Aja jälgimiseks peab töötaja vaid nuppu vajutama süsteemis, mis alustab töötundide maha salvestamist. Veidi keerukamad taimeriga süsteemid võimaldavad lisaks eristada eri tüüpi töid, mille alla tööaega salvestatakse [1, 10].

Ideaalseim meetod välitööde puhul aja jälgimiseks on asukohapõhised süsteemid [1]. GPS-jälgimise puhul saab töötaja läbi nutiseadmetes olevate rakenduste märkida oma tööaega ise koos asukohaga. *Geofence* ehk objekti raadiuse määramise tehnoloogia korral saab haldur määrata raadiuse ja seadistusest sõltuvalt sinna alasse sisenedes hakkab tööaeg automaatselt käima töötajal või saadetakse meeldetuletus tööaja märkimiseks [11]. Lisaks asukohapõhisele ajamärkimisele võimaldavad osad ka jälgida läbitud teekonna pikkusi, mis sobivad hästi transpordiga tegelevatele ettevõtetele [1].

Töötajate täielikult automatiseeritud tööaja jälgimiseks on taustal toimivad monitooringul põhinevad süsteemid. Need süsteemid jälgivad töötaja tegevust ekraanil. Sellised lahendused sobivad kontoritöötajatele, kes töötavad ekraanidega. Süsteem automaatselt salvestab maha aja, mis on kindlate tööülesannete peale kulutatud. Lisategevustena võimaldab antud süsteem maha salvestada klaviatuuri tegevusi ning teha ekraanist pilte [1].

Täpsemaks ja edasijõudnumaks meetodiks kui paberkujul ja Excel tabelitena tööaegade märkimiseks on manuaalse salvestamisega süsteemid nagu Timetrackeri. Kasutaja saab ise kindla tööülesande alla aega märkida, valides sobiva ajahulga või intervalli. Sisse ehitatud teavituste võimekus meenutab töötajatele tööaja märkimisest. Antud lahendused võimaldavad lisaks tööaja märkimisele ka jälgida ületunde, alatunde, makstud puhkuseid ja äraolekuid [1, 10].

Kasutades projekti halduseks mõeldud süsteeme nagu Trello, saab lihtsalt jälgida igale tööülesandele kulutatud tunde. Paljud projekti halduseks mõeldud süsteemid ning aja jälgimiseks mõeldud süsteemid on disainitud töötamiseks koos. Lisaks võimaldavad projekti haldamise keskkonnad jälgida kui kaua on mingi kindel tööülesanne seisnud mingis staadiumis [10].

Valides sobivaimat süsteemi tuleb leida kõigepealt sobivaim meetod, vaadelda selle tugevusi ja nõrkusi (Tabel 1) ning kaardistades ära ettevõtte vajadused tööaja arvestamisel [1].

Tabel 1. Aja märkimise meetodite võrdlus.

Aja märkimise meetod	Tugevused	Nõrkused
Suusõnaline	Kiire ja lihtne.	Ebatäpne, puudub ülevaade kulunud ajast
Paberkandjal	Kasutatav kõikjal, tasuta, teadvustatakse endale millele aega kulutatakse	Oht kaotada või unustada täita, ülevaate andmiseks tuleb teha paberist pilt või skaneerida süsteemi.
Excel	Tasuta ning lihtne täita. Vajalik on vaid kell ning Excel või Google Sheets.	Usaldusel põhinev, võib esineda ebatäpsust. Tabeli tegemine ja käsitsi sisestamine võib muutuda ajakulukaks.
<i>Clock in/clock out, kaardi süsteemid</i>	Lihtne kasutada, kerge jälgida kohalolu ja töötunde kokku.	Ei võimalda tööde kaupa aega märkida, puudub ülevaade tehtud töödest, ei võimalda kaugtööd.
Taimeriga süsteemid	Ülevaade tegevuste kaupa, automaatne aja arvestus.	Vigane aja salvestamine, unustatakse sulgeda või vahetada tööde vahel.
GPS ja <i>Geofence</i>	Täpne info millal ja kus töötaja on.	Mikrojuhtimise ja jälgimise tunne, seaduslikud murekohad 24/7 jälgimises.
Monitooring	Detailne informatsioon töötaja tegevusest, tuvastatakse mitte lubatud tegevused.	Tekitab töötajates ebamugavustunnet, jälgimise tunnetus tugev.
Manuaalsed süsteemid	Ei tekita kontrollimise tunnetust, annab ülevaate ajakulust ja tehtud töödest.	Unustatakse täita, ajakulukas, vajab palju usaldust tööandja ja töötaja vahel.
Aja jälgimine projekti juhtimises	Ülevaade kindlatele tööülesannetele kulunud aegadest, ülevaade ülesannete seisudest.	Vajab ühtset arusaama ajamärkimise vajadusest, mitme süsteemi kasutamine, ajakulukas.

## 3.2 Aja märkimise süsteemid Eesti turul

Aja märkimise süsteemid on ülemaailmselt muutunud lahutamatuks osadeks paljudel ettevõtetel, seda ka Eestis. Turg on täis kõikvõimalikke eri süsteeme, eri meetoditega ja eri suunitlustega [9]. Uurides Eesti turgu, tuleb ette tosikond eri süsteeme. Leidub suuri ja väikseid süsteeme, mis on mõeldud erinevat tüüpi ettevõtetele, ainult juristidele ja advokaatidele, ainult ehitajatele, süsteemid projekti halduseks, müügihalduseks või raamatupidamiseks. Eksisteerivad süsteemid omavad eri tüüpi võimekusi ja meetodeid märkimaks tööaega. Mõned uuringul ette tulnud süsteemid Eesti turul on Toggl, Scoro, Wemply, Begin, Erply, Trello, Jira, Directo, Lawtime, Hours ja Remato.

Toggl on globaalne ettevõtte, mis on oma lahendusega 2019. aastal 20 parima ajamärkimise süsteemi seas [12]. Tegemist on süsteemiga, mis on mõeldud nii tiimidele, vabakutselistele kui ka agentuuridele produktiivsuse tõstmiseks ning mis toimib kõikjal ja kõikides seadmetes. Süsteem omab nii lihtsamaid kui ka keerukamaid võimekusi, mis on seadistatavad vastavalt kliendi vajadustele. Toggl võimaldab tiimi haldust, projekti haldust, erinevat tüüpi raportite loomist, aja märkimist nii salvestama pannes taimeriga kui automaatset jälgimist ning on integreeritav rohkem kui 100 erineva tööriistaga. Toggl võimaldab uutel kasutajatel liituda tasuta prooviversiooniga, veendumaks süsteemi sobivuses. Süsteemi kliendina tuleb peale proovi versiooni tasuma kasutuse eest vastavalt valitud pakatile. Pakutakse stardipaketti hinnaga 9 dollarit kuus ühe kasutaja kohta, *premium* ehk tulusamat paketti 18 dollarit kuus kasutaja kohta ning suurettevõtetele eri hinnaga pakette [13].

Scoro puhul on tegemist süsteemiga, mis on mõeldud toomaks kokku kõik firma juhtimisega seotud tegevused ühte kohta, seehulgas ka tööaja märkimine ja arvestamine. Tegemist on suure ja paljude eri funktsionaalsustega süsteemiga, mis on integreeritav eri kalendritega, Dropboxiga, Mailchimpiga või teatud raamatupidamissüsteemidega. Süsteem on kasutatav kõikjal ja kõikides seadmetes ning võimaldab suuremal määral kohandamist vastavalt kliendi vajadustele. Vastavalt vajadustele leiab klient ka omale sobivaima pakutava paketi, hinnad algavad Scoro puhul alates 19€ kasutaja kohta kuus. Lisaks kasutusele pakub ettevõtte ja juurutamise pakette ettevõtetele, mis algavad 799€ ja on kuni 2999€. Scoro süsteemis toimub tööaja ja muude tegevuste märkimine käsitsi kalendrisse. Lisaks tööaegadele saab märkida isiklikke tegevusi ja puhkusi, mida

seadistades isiklikuks, ei kuvata aruandlusesse. Kalender võimaldab planeerida kogu kasutaja aega [14].

Begin, Lawtime, Remato ja Erply puhul on tegemist lahendustega, mis on kindlate kasutajagruppide suunitlustega. Begin ja Remato on mõeldud tootmis-ja ehitusvaldkonna ettevõtetele, Erply jaemüügi ettevõtete halduseks koos aja arvestamise võimalusega ning Lawtime advokaadi büroodele. Vaadeldes täpsemalt Lawtime süsteemi, on tegemist stopperil põhineva ajamärgimise süsteemiga, kus leidub ka kalendri vaade tehtud tööde vaatamiseks. Tegemist on aga süsteemiga, mis piirab kasutust oma nõuetega süsteemidele, kus seda kasutatakse. Süsteemi toimimiseks on vaja Windows operatsioonisüsteemi, vähemalt 50MB vaba kõvakettamahtu ja minimaalsest 512MB, soovitatavalt 2GB operatiivmälu. Integratsioonide jaoks on soovitatav Microsoft Office 2007, 2010 või 2013 [15]. Begin ja Remato puhul aga tegemist süsteemidega, mis kasutavad aja käsitsi ja stopperiga märgimise abistamises ka GPS positsioneerimisi. Lisaks aja märgimisele võimaldavad projekte hallata ning Remato puhul ka tööriistaid jälgida ja virtuaalset inventuure vaadata [16, 17].

Trello ja Jira puhul on tegemist peamiselt projekti haldamise ja planeerimise süsteemidega kui ajamärgimise süsteemidega. Trello puhul tegemist tasuta süsteemiga, Jira puhul süsteemiga, mille hinnad on alates 7 dollarit kuus vastavalt kasutajate arvule. Süsteemide puhul ei ole tegemist otse tööaegade logimiseks mõeldud süsteemidega, kuid võimalus on süsteemis tööks kulunud aegasid arvestada. Aegade arvestamisel aitavad kaasa mitmed erinevad ajamärgimise süsteemide integratsioonid [18, 19].

Wemply ja Hours on tegemist kõigile mõeldud tööaja arvestamise programmidega. Hours põhineb tööajale peamiselt, Wemply võimaldab lisaks tööajale jälgida ka puhkuse arvestust ning personali arvestust, süsteemis on võimalik funktsionaalsusi vastavalt vajadusele seadistada. Kokkuvõttes on tegemist suhteliselt väikeste ja aja arvestamisele keskenduvate süsteemidega võrreldes teiste vaadeldud süsteemidega [20, 21].

Directo on äritarkvara, milles on ühendatud erinevad ettevõtte juhtimiseks vajalikud osad nagu raamatupidamine, müük, logistika, projektijuhtimine, töökorraldus, personali ja palga arvestus. Ühe osana on süsteemist ka tööaja arvestamine, kuid peamiselt on süsteem mõeldud finantsarvestusteks. Ettevõtte tugevusteks on pikk tegevusaeg ja tugev maine Eesti turul ning personaalse lähenemise pakkumine. Süsteemi hinnad algavad 149 eurost,

ning on kasvava vastavalt üheaegsete kasutajate arvule, liidestustele, laiendustele, lisateenustele ja aktiveerimistele [22].

Vaatamata suurele valikule erinevatele süsteemidele turul, on paljudel ettevõtetel loodud oma sisemised süsteemid nagu ka antud magistritöös uuritav aja märkimise süsteem Timetracker. Peamiselt leidub ise loodud ja ettevõtte siseseid süsteeme tarkvaraarendusega tegelevates ettevõtetes. Üheks selliseks ettevõtteks on ka Tieto. Ettevõtte töötajate sõnul on tegemist täpselt ettevõtte vajadustele vastava lahendusega, mis on liidestatud kõikide vajalike süsteemidega ning mis annab ülevaate töötundidest, projektidest, alamprojektidest ja töötundide täituvusest.

### **3.3 Turu kokkuvõte**

Aja märkimise turg on vaid kasvav, nii ettevõtete vajaduste tõttu kui ka võimalike riiklike nõuete järgi. Näiteks Euroopa kohtu hagi Deutsche panga osas hispaanias, kus nõutakse aja märkimise süsteemide kohustuslikuks tegemist [2].

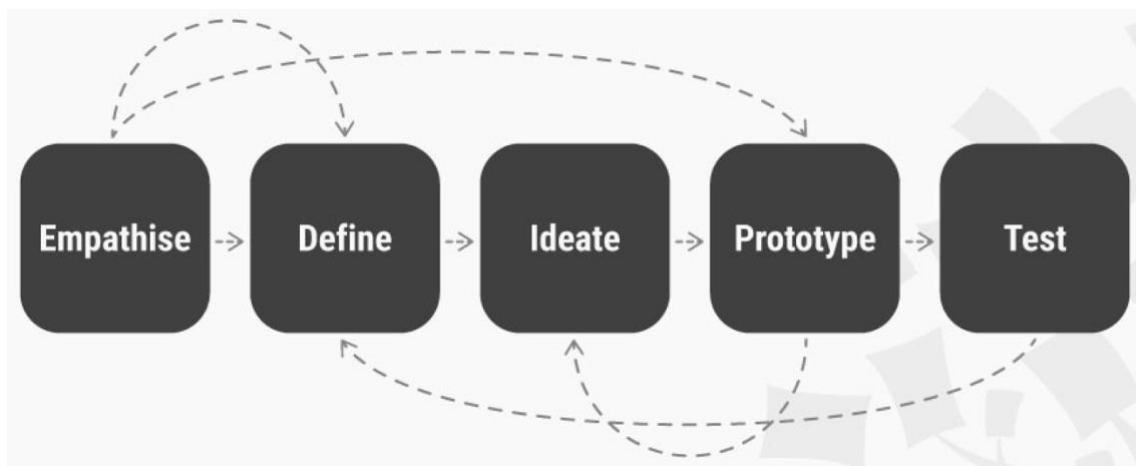
Antud töös vaadeldud aja arvestamise süsteemid on vaid mõned täna turul leiduvatest süsteemidest, kuid katavad suurt osa Eesti turul kasutatavaid. Toggl, Trello, Jira ja Directo puhul on tegemist märkimisväärsede tegijatega, kelle pakutavatest võimalustest tulenevalt võib öelda, et aja märkimise võimekused katavad vajadusi, kuid kelle kõrval on turul veel ruumi uutele süsteemidele.

Turu suurimaks puuduseks on süsteemide vaatlusel põhinedes lihtsad, vaid aja ja maksumuse arvestamiseks mõeldud süsteemid, mis võimaldavad nii aja märkimist kalendrisse kui ka salvestamist, luua ja hallata raporteid, sisestada ja vaadelda hinnangut, nende täituvusi, teiste osapoolte poolset panust töödessa ning töötaja üldist tööaja täituvust. Lisaks sellele puuduvad turult vabavaralised süsteemid, mis võimaldaksid isikute või ettevõtete soovi korral personaliseerimist ning vajalike lisavõimekuste realiseerimist.

## 4 Süsteemi disaini meetodika

Protsesse disainimaks tooteid ja teenuseid vastavalt kasutajate reaalsele vajadusele on mitmeid. Täna maailmas täis akronüüme ja sõnakõlke kasutatakse mitmeid eri sõnu kirjeldamiseks ühtesid ja samasid tegevusi. Hägunenud on piirid ning arusaamad eri disaini protsessidest. Peamised kasutuses olevad ja mainitavad protsessid ning meetodikad on kasutajakeskne disain, disainimõtlemine, süsteemi disain, süsteemne mõtlemine, agiilne UX, interaktsiooni disain, kasutajakogemuse disain ja kasutajaliidese disain. Tegemist ei ole samatähenduslike protsessidega, kuid suurel määral on hägunenud jooned nende eesmärkide ja sammude vahel [23].

Antud magistritöös võeti aluseks disainimõtlemise meetodika. Disainimõtlemine ehk *design thinking* on meetodika, mis pakub probleemide mõistmisele ja parandamisele lahendusepõhist lähenemist [24]. Tegemist on kasutajakesksel, koostööl ja tsükli kordustel põhineva lähenemisega probleemide kaardistamisele ja lahendamisele. Protsess koosneb viiest sammust -avastamine, defineerimine, kujutlemine, prototüüpimine ja testimine (Joonis 4). Disainimõtlemises toetutakse tugevalt empaatialle, ideede genereerimisele ja eksperimentidele, loomaks lahendusi, mis vastavad inimeste vajadustele ja soovidele. Lahenduste vajadustele vastamise kõrvalt veendutakse samas tehnoloogilises lahenduses ja strateegia mõistlikkuses [25].



Joonis 4. Disainimõtlemise protsessi sammud [24].



Aja märkimise süsteemi Timetracker puhul on tegemist süsteemi ümberprojekteerimisega (*reengineering*), lahendades olemasoleva süsteemi mured ning võimaldades süsteemi kasutada rohke kui ühel ettevõttel [26]. Silmas pidades kasutaja vajadustele vastamist ning probleemile lahenduse otsimist, on süsteemi ümber disainimiseks valitud just disainimõtleme meetoodika. Suurel määral erinevate protsesside ja meetoodikate olemasolu tähendab ka alati mitte projekti jaoks vale protsessi ja meetoodikate valikut, vaid pigem lähtutakse omale sobivaimatest variantidest. Ei ole valet varianti ja lähtuda tuleb omale sobivaimast. Aja märkimise süsteemi analüüsis, vajades lahenduspõhist lähenemist probleemide likvideerimisele, valiti disaini mõtleme meetoodika, kasutades sellest osalisi samme [27]. Tööst jäeti välja prototüüpimise ehk realiseerimise ning testimise sammud põhjusel, et tegemist on võimalike lahenduste kirjeldava tööga ning mitte süsteemi realiseerimise tööga.

Järgnevatel alampeatükkides antakse ülevaade disainimõtleme sammudest, mida antud aja märkimise süsteemi uue lahenduse analüüsimiseks kasutati.

#### **4.1 Avastamine**

Disainimõtleme esimeseks sammuks on avastamise samm (*discover/empathize*). Esimese sammu eesmärgiks on luua emotsionaalne side murega, mida minnakse lahendama. Selleks uuritakse kasutajakogemust läbi vaatluste, suhtluse ning tunnetuse, inimeste enda keskkonnas. Läbi uuringu luuakse põhjalikum ja personaalne arusaam seotud teemast. Peamiseks avastamise eesmärgiks on jätta kõrvale isiklikud eelarvamused maailmast, et saada arusaama kasutajatest ja nende vajadustest. Empaatiat loomine lahenduse analüüsi algstaadiumis on oluline [24].

Aja märkimise süsteemi tänaste probleemide mõistmiseks ning töötamiseks välja lahendusi, seati avastamise sammus paika eesmärgid, visioon, reeglid, nõuded. Lisaks viidi läbi kasutajakogemuse uuringud, mõistmaks kasutajate vajadusi ja murekohti.

#### **4.2 Defineerimine**

Defineerimise sammus pannakse kokku informatsioon, mis koguti eelnevas, avastamise sammus. Analüüsitakse uuringu tulemusi ja defineeritakse põhiprobleem. Murekohtade illustreerimiseks kirjeldatakse probleemid kasutuslugudena [24].

Probleemi korralik defineerimine abistab koguda ideid tegevuste, featuuride ja muude elementide jaoks. Probleemile lahenduse otsimine läbi küsimuste küsimise aitab kaasa ideede loomisele ning keskenduda selgelt disainile. Ilma püstituseta töötatakse teadmatuses lahenduse poole [24, 28].

Korraliku probleemi püstitus defineerimise sammus peab olema sõnastatud kindla kasutaja vaatest (kasutuslood), nende vajadustest tulenevalt. Püstitus peaks lähtuma inimestest, keda proovitakse aidata, mitte toetudes otseselt tehnoloogiatele, rahalistele väljaminekutele ja muudele täpsustustele. Samas peab olema probleem selline, mis oleks lahendatav [28].

Timetrackeri süsteemi lahendi väljatöötamiseks loodi defineerimise sammus uued kasutuslood, analüüsi uuringuid ja kaardistati valukohad.

### **4.3 Kujutlemine**

Idee faasis ehk kujutlemise sammus alustatakse ideede loomist, mis lahendaksid probleeme. Avastamise sammus luuakse arusaam kasutajate vajadustest, defineerimise sammus analüüsitakse tähelepanekuid ning nende tulemusena on jõutud probleemi püstituseni. Kindla taustaga saab alustada uute lahenduste väljatöötamist ja vaadelda erinevaid võimalusi. Idee loomiseks on sadu erinevaid tehnikaid nagu näiteks ajurünnak ehk *brainstorming*, joonistamine, piltlood, rollimäng, visualiseerimine [24, 29].

Lisaks lahenduse visuaalsele disainile vaadeldakse kujutlemise sammus ka süsteemi disaini. Tulenevalt üldisest kujutluse loomisest läbi visuaalsete prototüüpide, vaadatakse antud magistritöös idee faasis ka uue süsteemi disaini ehk info arhitektuuri.

## **5 Aja märkimise süsteemi kasutajakogemuse analüüs**

Loomaks head kasutajakogemust, on vaja teada kasutajate tegelikke vajadusi ja tegevusi. Kasutajakogemuse puhul vaadeldakse lõppkasutaja suhtlust nii ettevõttega kui ka selle teenuste ja toodetega [30]. Ilma mõistmata kasutajaid, mida nad vajavad ning soovivad, on peaaegu võimatu pakkuda toodet või teenust, mis pakuks väärtust mitte ainult ettevõttele vaid ka kasutajale endale. Tänapäevases kiire arenguga ja konkurentsirohkes keskkonnas ei saa enam jätta tahaplaanile kasutajate soove ja vajadusi. Aina enam pannakse rõhku kasutajakogemusele äri eesmärkide ja soovide kõrvalt. Kasutajakogemuse tuumaks on kindlustada pakutava lahenduse väärtus kasutajate silmis. Selleks, et pakkuda mõtestatud ja väärtust andvat kasutajakogemust, peab olema edastatav info Peter Morville sõnul kasulik, kasutatav, haldusväärne, leitav, ligipääsetav ning usaldusväärne [31].

Parimaks viisiks mõista kasutaja käitumismustreid, vajadusi, soove ja motivatsioone, on läbi kasutajate ja kasutajakogemuse uuringute. Nii uute lahenduste realiseerimise kui olemasolevate lahenduste puhul, on parim viis mõistmiseks kasutada kasutajakogemuse ja kasutaja uuringu meetodeid nagu vaatlused, intervjuud ja kõik muud tagasiside meetodid [23]. Tulenevalt sellest, olemasolevate turul olevate aja märkimise süsteemide kui ka Timetrackeri analüüsist endast pandi kokku uuringuks täpsem plaan, eesmärgiga leida nõrkused, tugevused, vajadused ning eelised. Kasutuskogemuse uuring võeti aluseks analüüsima üldist kogemust, kaardistamaks ära valukohti, mida tuleks parandada ja kinnitada või tagasi lükata Timetrackeri võimalikku kohta turul olevate lahenduste kõrval.

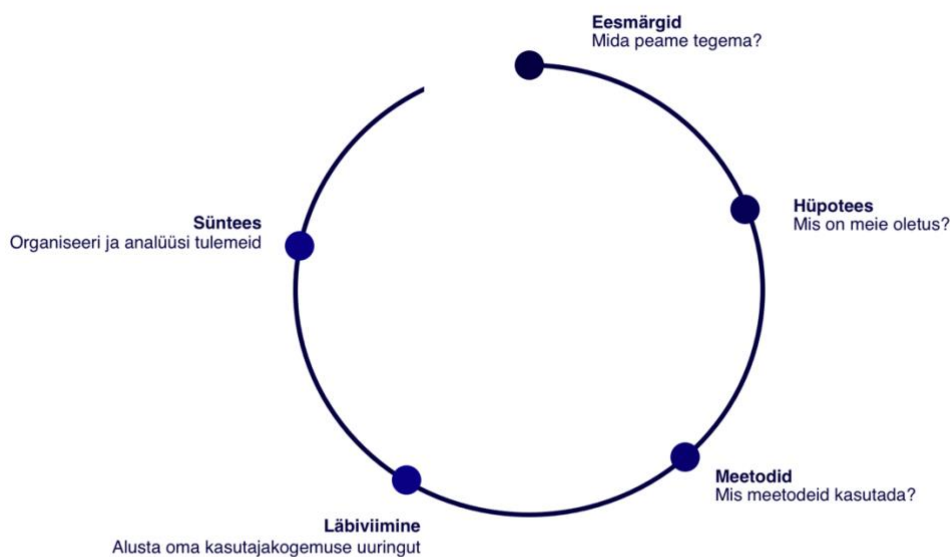
Järgnevatel alampeatükkides vaadeldakse täpsemalt kasutajakogemuse uuringut, plaani, uuringu tulemustest lähtuvat süsteemi kogemuse parandusideid ja võimalikku kasutust.

### **5.1 Kasutajakogemuse uuring**

Aina kasvava globaalse turu konkurentsi juures oodatakse mõõdetavaid tulemusi. Ettevõtted tahavad kindlust, et äriideed mõistetakse üheselt ning lahendus toob investeringut sisse. Uurimismeetodite nagu kasutajakogemuse uuringu sissetoomine

disaini protsessi aitab kaasa sihtgrupi defineerimisele, toetada kontseptsiooni ja mõõta efektiivsust. Uuringutel põhinev lähenemine võib aidata defineerida viisi, kuidas probleeme lahendatakse ning illustreerida lahenduste väärtust [32].

Viimaks läbi usaldusväärset ja pidavat kasutajakogemuse uuringut, tuleb läbida kindlad sammud (Joonis 5). Ennem kui saab alustada oma uuringuga ning analüüsida tulemeid, tuleb seada paika uuringu eesmärgid, oletuslik hüpotees ning meetodid, millega läbi viia uuring [33].



Joonis 5. Kasutajakogemuse uuringu protsess (Erin Sanders).

Aja märkimise süsteemi Timetracker turu võimaluste, olemasolevate ning võimalikke uute kasutajate mõistmiseks viidi läbi kasutajakogemuse uuringud nii ettevõtte töötajatega kui inimestega, kes oma töös märgivad ülesse töötunde ja arveldavad tunnitasu järgi tööd. Eesmärgiks oli uuringuga ära kaardistada olemasolevate kasutajate murekohad ning uute võimalike kasutajate ootused, soovid ja hetkel kasutatavate vahendite puudused. Uuringutesse mindi hüpoteesiga, et tööaja arvestamise süsteemid on tänapäeval paljudes valdkondades olulised ning vaatamata suurele valikule ei ole kõik leidnud endale parimat lahendit. Lisaks seati oletus, et vaatamata paljudele eri lahendustele on võimalik leida endiselt eelis teiste süsteemide ees.

### 5.1.1 Plaan

Paremaks uuringu läbiviimiseks seati paika plaan, mille järgi uuring läbi viia. Plaani täpse ühe lehelise ülesehituse leiab lisast 3.

Olemaks kindel, et tegevused põhinevad kasutajakogemuse uuringute tulemustel, peab olema plaan selge, kiire, vastuvõetav ja koostööl põhinev. Uuringu plaan ei tohi olla pikem kui üks lehekülg. Pikad plaanid või plaani puudus ei tööta inimeste puhul [34].

Kasutajakogemuse uuringu plaani hüved:

- Üles kirjutatud plaan aitab tuvastada mis toimib ja mis ei toimi inimeste jaoks, samuti mis küsimustele otsivad vastuseid.
- Plaan kindlustab kõikide osapoolte kaasamist läbiviimisse ja tulemustesse. Kui pole plaani, siis on oht, et vastutavad osapooled ei tunne seotust.
- Üles kirjutamine aitab, kirjas tunduvad asjad teisiti kui oma peas olevad mõtted [34].

Omades plaani on lihtsam leida ühine keel ja luua seotuse tunne kõikide vastutavate osapooltega. Plaanid on üldjuhul endale, juhtidele, tootomanikele ja arendajatele. Tänu plaanile saab kõikidele osapooltele välja tuua eesmärgid, küsimused ja ajakava. Lisaks annab see ülevaate ka uuringu üldisest suurusel, kulust, kui ka miks seda uuringut üldiselt vaja on [34].

Ühe lehelise uuringu kokkuvõtte peaks koosnema pealkirjast, autorist ja seotud osapooltest, kuupäevast, taustast, eesmärgist, uuringu küsimustest, metodoloogiast, osalejate kirjeldusest, ajakavast [34].

Vastavalt sellele sai Timetrackeri jaoks kokku pandud uuringu plaan. Plaaniga seati paika uuringu meetodid nagu intervjuud ja vaatlused, osalejate grupid, üldine kava, küsimused ja taust. Plaan võeti aluseks uurinu läbiviimisel erinevate osapooltega, saavutamaks seatud eesmärki. Läbiviidud kava alusel koguti informatsioon panemaks kokku süsteemi kasutajakogemuse üldine arusaam ehk tulemid.

### 5.1.2 Tulemid

Vastutavate osapooltega läbiviidud vestlustest tulenevalt võib järeldada, et tänase seisuga on Timetrackeri arendusele pandav rõhk minimaalne. Vaatamata aga sellele ei ole jäetud tahaplaanile mõtet ja soovi süsteemi parandada ning võimalusel teha kättesaadavaks nii teistele eraldi seisvatele üksustele kui ka firmadele. Ettevõtte on valmis eraldama ressursse nii inimeste, aja kui muu vaates, eeldusel, et on tehtud kaardistus parandustest, lisavajadustest, võimalustest ja üldine pilt süsteemi võimalikust lahendusest. Vaatamata paljudele sarnastele süsteemidele turul usub ettevõtte, et on võimalik pakkuda oma süsteemiga piisavat konkurentsi ning täita teatud kasutajaskonna vajadused. Peamiselt usutakse, et loodud tööaja arvestamise süsteem oleks pigem suunatud sel juhul vabakutselistele ja juura valdkonnas tegutsevatele kui IT ettevõtetele.

Kontekstuaalse uuringu käigus vaadeldi igapäevast töötajate süsteemi kasutust, mille järel uuriti vestluse käigus täpsemalt süsteemi kasutuse kohta. Kokkuvõttes vaadeldi ja intervjueriti 6. töötajat. Vastavalt vaatlusele ja intervjuudele sai järeldada igapäeva kasutajate vaatest järgmisi positiivsed, negatiivseid, parandust vajavaid või uusi vajalikke aspekte:

- Süsteemi on lihtne kasutada – kalendri vaates *drag and drop* meetodil tööde märkimine.
- Puhas disain, ei ole midagi üleliigset, kuid kõik vajalik funktsionaalsus olemas.
- Kiire ülevaade normtundidest ja tundide täituvusest kuus, nädalas ja päevas, kuid kohati segadusttekitav. Arvestatakse jooksev päev tööpäeva alguses miinusesse koheselt.
- Viimati vaadatud ja märgitud tööd kohe nähtaval, teiste otsimine ebamugav. Tuleb ise teada, mis tööd süsteemi sisestatud ja mis ei ole. Otsimine vaid nime järgi. Võimaldada grupeerida *tag*'ide abil töid, mis abistaks ka kergemalt otsida vajalikke töid.
- Parem ülevaade töö hinnangust, hinnang eraldi välja tuua kalendri vaates. Eraldi märkega tööd, kui kulunud aeg suurem hinnangust.

- Salvestamine funktsionaalsus arusaadavamaks teha ja uue alustamisel eelneva automaatne peatamine. Jooksev töö nähtavamaks ja meeldetuletused.
- Raportite genereerimine ebaselge ja piiratud. Soov saada parem ülevaade tööd, kulunud aegadest, arveldusest ja muust tööga seotud informatsioonist. Raporti valikud/filter selgem ja lihtsam, lisaks võimalus välja võtta diagrammidena ülevaateid (kulunud tunnid kuus ja nädalas, kõige rohkem aega kulunud projekt, arveldus kuu või nädala kaupa jne).
- Jira ja Outlookiga sünkroniseerimine lihtsustaks hinnangute lisamist ja kalendri täitmist.
- Puudub lihtne ja kiire ülevaade, kes töö alla aega märkinud.
- Soov kiiresti kalendri vaates saada ülevaade mis tüüpi tööga tegu.
- Soov, et oleks kasutatav ka nutiseadmes.
- Planeerimise funktsionaalsus ebaselge ja peidus.

Potentsiaalsete kasutajatega viidi läbi neli intervjuud. Intervjueeritavad olid IT sektorist, õigusteaduse valdkonnast ja vabakutselised töötajad. Intervjuude tulemina võib järeldada, et aja märkimise süsteemid on olulised osad tööaja märkimises nii IT, juura erialadel kui ka vabakutseliste seas. Intervjueeritavad tõid praegu kasutatavate vahenditena välja eri süsteeme nagu Jira, Toggl, MS Dynamics AX ja Directo. Neljast intervjueeritavast üks oli rahul praegu kasutuses oleva süsteemiga ning ei tundnud huvi uue süsteemi vastu. Teised intervjueeritavad tõid välja erinevaid puuduseid praegustes süsteemides ning nimetasid vajalikke funktsionaalsusi, millest puudust tunnevad. Järgnevalt on toodud välja aspektid, mida intervjueeritavad sooviksid süsteemis näha ning mis paneks neid tänase süsteemi kasutuse osas ümber mõtlema:

- Võimalus salvestama panna jooksvat tööd, märkimaks võimalikult täpset kulunud tööaega.
- Kiire ja lihtne ülevaade kulunud tundidest ning töö kirjeldustest.
- Süsteemi sidumine Outlooki, Exceli ning Jiraga.

- Raportite lihtne ja mugav loomine ning muutmine ennem eksportimist Excelisse.
- Kalendrisse märgitud kohtumiste ja aegade logimine kui tööna
- Klientide sisestamine nime ja koodi järgi, kliendi peale kulunud aja kokkuvõtlik ülevaade.
- Aja märkimise süsteemi muutmine täpselt endale sobivaks.

## 5.2 Kasutus ja kogemus

Vastavalt kasutajakogemuse uuringutele võib järeldada, et vaatamata suurele kogusele erinevatele tööaja arvestamise süsteemidele turul, leidub veel endiselt isikuid ja ettevõtteid, kes pole leidnud endale sobivat lahendust. Otsitakse võimalikult lihtsat ja mugavat viisi oma tööaja märkimiseks ning ülevaate saamiseks, kuid mis samal ajal omaks endas keerukamaid funktsionaalsusi vajalike raportite loomiseks, integreerimiseks teiste süsteemidega, töö kirjelduste, tundide ja hinnangute vaatamiseks, projektide ja tööde otsimiseks.

Vaadeldes kasutajaskonda, peaks panema antud süsteemi puhul rohkem rõhku ettevõtte enda töötajate kõrvalt õigusteaduse valdkonnale, vabakutselistele kui ka ehitajatele. Tegemist on gruppidega, kelle seas on tänaste tööaja märkimise lahenduste osas kõige enam rahulolu ning kes otsivad võimalikult lihtsat süsteemi, mis omaks nendele kõiki vajalikke funktsionaalsusi. IT valdkonnas või muudes suundades võib vaadata Timetrackeri eelisena ja müügiartiklina lahenduse enda taga peituvat tugevat tarkvaraarendusettevõtet, kes on võimeline pakkuma personaalseid lisafunktsionaalsusi.

Tarkvaraarendusettevõtte, kelle loodud ja kus Timetracker kasutuses igapäevaselt, ei ole loobumas ega otsimas endale uut süsteemi tööaja märkimiseks ning arveldamiseks. Vaatamata vähesele tähelepanule alates aastast 2016 süsteemi arenduse vaatest, on ettevõtte valmis panema rõhku süsteemi parandamise peale ning tehtud uuringute ja ettepanekute põhjal looma uut ja paremat lahendust nii ettevõtte töötajatele kui kolmandatele isikutele.



Vastavalt ettevõtte töötajatele, kes süsteemi igapäevaselt kasutavad ning võimalikele uutele kasutajatele tuleks arvestada süsteemi parandamisel või uue süsteemi loomisel järgmisi muudatuste ettepanekuid ning uusi funktsionaalsusi:

- Rakendus kasutatavaks nutiseadmetes.
- Visuaalne uuendus – selgem kalendri vaade, raportite vaade, menüü, otsingud.
- Töö tüüpide eristus kalendris.
- Tööde logimine selgemaks ja arusaadavamaks.
- Tööde logimise lõpetamine automaatselt uue alustamisel, kinnituse küsimine logimiseks.
- Logitud töötundide ülevaade kasutajate kaupa.
- Genereeritud raportite muutmine enne eksportimist.
- Hinnangu ületamisel kalendri vaatesse eraldi mäрге tööst, nähtavamaks tegemine hinnangu ületamine.
- Märgitud tundide ülevaate lihtsustamine.
- Töödele tag'ide lisamine, tööde otsingu täiendamine ja muutmine tag'ide järgi tööde otsing.
- Integreerimine teiste süsteemidega või erilahenduste võimalus.
- Tööde kirjeldus selgem, eelvaade kalendris.
- Süsteemi sisemised teavitused lisada erinevalt meilile saadetavatest teavitustest.
- Säilitada ja rõhutada nii logimist kui ka *drag and drop* tööde märkimist kalendri vaatesse – tugev pluss mõlema olemasolu turul olevate lahenduste kõrval.

Välja toodud parandusettepanekud ja uued funktsionaalsused parandaksid märgatavalt olemasoleva aja märkimise süsteemi kasutajakogemust ning looksid väärtust mitte ainult tarkvaraarendusettevõtte enda töötajatele vaid ka teistele isikutele ja ettevõtetele.

## **6 Võimalike lahenduste analüüs**

Tulenevalt olemasolevast süsteemist, turu ja kasutajakogemuse uuringust ning lähteideest aja märkimise süsteemi teha kättesaadavaks ka teistele ettevõtetele, vaadeldakse järgnevates alampeatükkides eesmärgi, uusi nõudeid, võimalikke süsteemide lahendusi ning disaini ettepanekuid. Magistritöös kasutatud meetodite abiga kogutud informatsioon ja sisend võimaldab pakkuda välja võimalikke uusi lahendusi ja funktsionaalsusi, mis realiseerimise järgselt võiks luua lisaväärtust nii ettevõttele kui ka ettevõttega mitte seotud isikutele.

Antud peatükis keskendutakse tööaja arvestamise süsteemi võimalikele realiseerimise variantidele ning nende kirjeldamisele, vaadeldes mis on tarkvaraarendusettevõttele kõige kasumlikum. Eesmärgiks on anda ülevaade üldisest süsteemi visioonist ning soovidest kui ka võimalikust uuendatud süsteemi disainist.

### **6.1 Visioon ja eesmärgid**

Uuendatud aja märkimise süsteemi Timetracker (edaspidi Timetracker 2.0) visiooniks on võimaldada kõikjal ja kõigil soovijatel märkida oma tööaega, luua raporteid arvelduseks ning saada ülevaade nii enda kui teiste isikute töötundidest.

Eesmärgiks on Timetracker viia võimalikult lihtsalt ja väheste ressurssidega üle uuematele tehnoloogiatele, vähendades nii vana tehnoloogia kasutamist. Lisaks on eesmärk leida lahend, mis oleks parim nii ettevõttele endale kui mitte ettevõttega seotud kasutajatele.

Soovides laiendada ettevõtte sisesest kasutusest kaugemale, seatakse eesmärgiks tagada süsteemi arenduse jaoks nii pidevat aja- ja inimressurssi kui ka tugisüsteemi. Mõistmaks süsteemi tootena pakkumises tugevusi, nõrkusi, ohtusid, kasvu ja arenemise võimalusi, koostati IT SWOT analüüs (Tabel 2). SWOT analüüsi eesmärgiks on vähendada nõrkuste mõju ning suurendada samal ajal tugevusi. Lisaks annab SWOT analüüs ülevaatliku pildi

selles, kuidas positsioneeritakse end konkurentide suhtes, millised on tulevikuvõimalused ja ning milliste ohtude eest tuleks end kaitsta [35].

Tabel 2. Timetracker süsteemi IT SWOT (autori loodud).

Tugevused	Nõrkused
<ul style="list-style-type: none"> <li>• Personaalne lähenemine, kasutajatugi</li> <li>• Kogenud tarkvaraarendusettevõtte loodud tarkvara</li> <li>• Täpsus</li> <li>• Salvestamine ja käsitsi sisestamine</li> <li>• Aja ülevaade/analüüs - alatunnid, ületunnid, enesearendamise tunnid, tööpuudus, ülekoormus jne</li> </ul>	<ul style="list-style-type: none"> <li>• Tehnoloogia ja platvorm - tehnoloogia kiire areng</li> <li>• Piiratud funktsionaalsus - aja märkimine ja raporteerimine</li> <li>• Võimalikud tõrked keskkonnale ligipääsemisel</li> <li>• Toote müügikogemuse puudumine</li> </ul>
Võimalused	Ohud
<ul style="list-style-type: none"> <li>• Erilahendused</li> <li>• Teiste süsteemide integreerimine - Google Calendar, Jira, Outlook jne</li> <li>• Kasutatavad nutiseadmetes</li> <li>• Aina rohkem tunnitaset arvestust pidavad ettevõtted</li> <li>• Pilvelahendus</li> </ul>	<ul style="list-style-type: none"> <li>• Andmete konfidentsiaalsus ja säilimine</li> <li>• Muutuv ärikeskkond</li> <li>• Muutuvad vajadused</li> <li>• Muutuvad, vananevad tehnoloogiad</li> <li>• Suur konkurents</li> <li>• Süsteemi hallatavus</li> </ul>

Ettevõtte vaatest on tugevusteks erialane inimeste kompetents, tugev maine erinevate süsteemide arenduses, üle kuue aasta pikkune kogemus tarkvara arenduses ning võimekas tiim, kes pakuks nii tuge kui ka personaalsete lahenduste loomist. Süsteemi tugevusteks aja märkimise osas on *start-stop* ja käsitsi sisestamise võimekus. Kasutaja saab sisestada minuti täpsusega tööaega või 15 minutilise täpsusega tirides ja kohendades tööd kalendri õiges ajahetkes. Lisaks tööaja märkimisele ja arvestamisele on süsteemi tugevuseks ülevaatlik pilt kasutaja tundidest päeva, nädala ja kuu kaupa – täpne ülevaade tehtud tundidest, tegemata tundidest ja tööaja katvusest.

Nõrkustena võib välja tuua ja arvesse võtta kiiret tehnoloogia arengut, millega raske kaasas käia, piiratud funktsionaalsus aja märkimise ja arvestamise näol, võimalikud tõrked süsteemis ning tarkvaraarendusettevõttena toote müügi kogemuse puudumine.

Ohtudena tuleb silmas pidada süsteemides andmete hoiustamist, nende haldamist ja turvalisuse tagamist ning ka süsteemi tehnoloogiate vananemist. Lisaks sellele tuleb silmas pidada ka keskkonna pidevat muutumist, vajaduste muutumist ning suurt ajaarvestamise süsteemide konkurentsi.

Süsteemi muutmisel ja pakkumisel, tuleb kaaluda kergete integratsioonide võimaldamist teiste süsteemidega ning ettevõtetele erilahenduste pakkumist. Lisaks integratsioonidele ja erilahendustele tuleb luua rakendus, mis oleks pilvepõhine ning kasutatav kõikjal ja kõikides seadmetes. Vaatamata suurele konkurentsile turul, on tänu kasvavale ajaarvestuse vajadusele uutel lahendustel samuti head võimalused.

## **6.2 Reeglid ja nõuded**

Tulenevalt olemasoleva süsteemi puuduste analüüsist, tarkvaraarendusettevõtte soovidest uuele süsteemile ja kasutajakogemuse uuringu analüüsist pandi kokku uute reeglite ja nõuete kogum. Antud alampeatükis käsitletakse süsteemi nii parandavaid kui ka uusi reegleid ja nõudeid kasutajalugudena. Neid käsitletakse agiilse projektijuhtimise vaatest, loomaks arusaama vajalikest uutest funktsionaalsustest, nende suurusest ning paranduskohtadest, mida kaaluda Timetracker 2.0 arenduses [36]. Tegemist on analüüsi põhjal tehtavate muudatusettepanekutega süsteemile ning sellest tulenevalt võivad need olla muutuvad.

### **Reeglid:**

1. Süsteemil on kliendid. Klientide alla kuulub mitmeid isikuid. Iga kasutaja ehk isik on seotud vähemalt ühe kliendiga. Igal kliendil ning kliendiga seotud kasutajatel on eraldi ligipääs kliendi süsteemi.
2. Süsteemis saab kasutaja olla admin või user rollis. Admin rollis isik haldab süsteemis kliendi alla kuuluvaid isikuid. Admin saab lisada ja kustutada kliendi all olevaid isikuid ehk süsteemi kasutajaid.
3. Klient saab valida sisse logimise viisina kas süsteemi enda konto või Azure AD. Ettevõtte kontoga seotakse kasutajad läbi meiliaadressi ja privaatse ettevõtte (kliendi) identifikaatori.

4. Kliendi poolt sisestatakse uus kliendiga seotud isik läbi isikuga seotud meiliaadressi. Isiku sisestamisel saadetakse meiliaadressile kinnituskiri vaikimisi genereeritud parooliga, mida isik saab süsteemis muuta.
5. Iga sisestatud tööga süsteemis on seotud null kuni mitu *tag*'i ehk märksõna. Märksõnad võimaldavad kasutajal süsteemis grupeerida ja otsida olevaid töid, nende informatsiooni ning tehtud tunde ajaperioodidel.
6. Süsteem eristab sisestatud tööde tüüpe ikoonide abil. Kliendile kuvatakse tööde juures tüübile vastav ikoon. Süsteemi saab admin tüüpi isik sisestada sobivaid tüüpe ning valida nendele vastavaid ikoone.
7. Kasutajad saavad kasutada tööaja märkimise süsteemi kõikjal ja kõikides seadmetes.
8. Süsteem peab olema kasutajale ligipääsetav 98% ajast, 2% ajast võib korrastustöödeks ja haldamiseks olla süsteem kasutajatele maas.
9. Kasutajale kuvatakse vastavalt valitud ajaperioodile temale vastavalt süsteemi sisestatud töötunnid ja nende vastavus perioodis ette nähtud töötundidele. korraga genereerida ülevaate ühe valitud kindla perioodi kohta. Ülevaate genereerimisel tuleb sisestada algusaeg ja lõppaeg.
10. Kasutajad näevad märgitud tööde detailsemas vaates kaasalööjate nimekirja ehk nimekirja isikutest, kes on antud töö alla märkinud aega. Lisaks isikute loetelule kuvatakse ka märgitud tundide arvu isiku kohta.
11. Tööd, millele on hinnang sisestatud ning märgitud tunnid ületavad hinnangu, eristatakse teistest töödest märgitud tööde seas. Juhul kui hinnangut ei ole sisestatud või sisestatud hinnang ei ole ületatud, ei eristata märgitud tööd.
12. Töö salvestamisel kuvatakse ekraanile eraldi sektsioon töö salvestuse jooksmisest. Korraga saab salvestama panna ühe töö. Uue töö märkimisel kalendrisse või salvestamise alustamisel küsitakse jooksva töö lõpetamise kohta isikult lõpetamise kinnitust. Kinnitamise järel salvestatakse tehtud töö vastavalt salvestatud perioodile kalendrisse. Katkestamisel vastav töö ei salvestata tehtud töödesse.

**Kasutajalood:**

- Kliendina soovin, et saaksin kasutada süsteemi endale sobivaimas kohas ja sobivaimas nutiseadmes, et saaksin süsteemi kiiresti märkida tööaegasid ja võtta välja ülevaateid.
- Kliendina soovin, et süsteem oleks turvaline ning sellele ei saaks ligi kolmandad isikud, et hoida andmete lekkimist ja konfidentsiaalsust.
- Kliendina soovin, et saaksin valida oma ettevõttele sobivaima süsteemi sisenemise viisi, et tagada kiire ja mugav kasutajate autentimise.
- Kliendina soovin, et saaksin sisestada ja muuta oma süsteemi kasutavaid isikuid, et tagada vajalike ja kindlate inimeste olemasolu süsteemis.
- Kasutajana soovin, et töid eristataks tüüpide järgi kalendris, saamaks kiiret ülevaadet tehtud töö laadist.
- Kliendina soovin, et saaksin sisestada tööde tüüpe ja et tüübid vastaksid vajadustele.
- Kasutajana soovin, et saaksin luua kokkuvõtteid oma töötundidest vastavalt valitud perioodile, saamaks ülevaateid süsteemi sisestatud tundidest pikemates perioodides kui kuus, nädalas või päevas.
- Kasutajana soovin näha salvestamisel jooksvat tööd ja aega, et salvestamine oleks silme ees ja ei unustaks.
- Kasutajana soovin, et salvestatav töö lõpetataks ning küsitaks kinnitust juhul, kui sisestan uue töö või alustan uue salvestamist, et töö ei jääks jooksva ega kustuks.
- Kasutajana soovin lisada töödele külge märksõnu, et grupeerida töid ning otsida kiiremini töid, mis vastavad teatud märksõnale.
- Kasutajana soovin, et hinnangu sisestamisel ning selle ületamisel, kuvataks töö juures eraldi välja märke seisust ja ületamisest, et saaksin märkimisel ülevaate hinnangu täitumisest.

- Kasutajana soovin saada ülevaate töö alla aega märkinud isikutest ning nende tundidest, et olla kursis kes sama teemaga tegelevad.

## 6.3 Süsteemi lahendid

Antud töös tehtud süsteemi analüüsi järgselt võime järeldada, et Timetracker süsteemi puhul on tegemist aja märkimise süsteemiga, mida täna kasutatakse vaid ühes kindlas ettevõttes. Tegemist on lahendusega, mis ei ole mõeldud jagamiseks teiste osapooltega või pakkumask kui tootena ning on täna vanade tehnoloogiate peal. Antud vaatest on aeg mõelda süsteemi uuendamise peale või ümber tegemisele. Tulenevalt sellest on süsteemi puhul erinevad lahendid, kas vabaneda tehnoloogilisest võlast ja kasutada edasi ettevõtte siseselt või luua süsteemist toode ehk teistega jagatav lahendus (Timetracker 2.0).

### 6.3.1 Raamistiku uuendus

Raamistiku uuendamisel tuleks *front-end* puhul silmas pidada võimalike uute raamistike olemasolu ning kasutuselevõttu. Valikuid on palju, tänapäeval on *front-end* maailma areng äärmiselt kiire. Selleks ajaks kui raamistik on valitud ning rakendus loodud võib valitud raamistik juba olla vananenud. Samas on aga *front-end* maastikul raamistikest soovituslik kasutada uusimaid ja silmapaistvamaid lahendusi ja vaadata, kas tasuvad end ära [37]. Ent kõige selle juures, tuleb kõikide võimaluste seast valida olukorrale vastavalt kõige mõistlikum raamistik. AngularJS puhul on tegemist raamistikuga, millest loobumisel on kõik olukorra ees, kus peab lahenduse nullist ümber kirjutama mõnele uuele raamistikule [38]. Antud süsteemi vaates oleks lahenduseks samm-sammu haaval liikumine uusima Angular versiooni peale (Angular 7). Tegemist on üleminekuga, mille kohta leidub mitmeid kirjutisi ning on korralik ülevaade ja kirjeldus Angulari enda veebilehele, kuidas teha ümberkirjutamine võimalikult valutuks ja lihtsaks [38]. Lisaks mitmetele juhistele ja abistavatele kirjutistele võimaldab AngularJS pealt üleminek Angular versiooni peale lihtsalt kopeerida olemasolev disain üle uuele.

Spring Framework puhul on tegemist ühe populaarseima raamistikuga Java jaoks ning soovides vabaneda tehnoloogia võlast ja lihtsustada arhitektuuri, tuleks vaadata Spring Framework laiendust ehk Spring Booti. See võimaldab kiiremat ja kergemat Spring rakenduste arendamist [39]. Spring Boot lihtsustab suurel määral süsteemi arhitektuuri, lisades kihi aitamiseks seadistuste ja paigalduste automatiseerimisel ning lihtsamalt uusi

võimekusi rakendada [40]. Peamiselt on Spring Framework pealt üle minekuga Spring Boot raamistikule seadistustega. Üle minekul peaks olema minimaalset rakenduse koodile või muudele erikomponentidele [39].

### 6.3.2 Jagatav süsteem

Pidades silmas süsteemi jagamise vajadust ja soovi, tuleks vaadata erinevaid eksisteerivaid võimalusi. Mõningate variantidena saab vaadelda pilvelahendusi või *in-house* ehk majasiseseid võimalusi. Mõlemad võivad parandada ärilisi vajadusi, kuid tuleb vaadelda mis toimib lahenduse vaatest ja on kõige kasulikum [41].

Majasisese serveri puhul on tegemist traditsioonilisema majutusviisiga, kus kasutatakse füüsilist masinat organisatsiooni vajaduste täitmiseks. Kogu kontroll on ettevõtte enda käes ja ei pea lootma turvalisuse osas pilveteenusele. Lisaks võimaldab majasisese serveri kasutus uuendada server just vajadustele vastavaks ning jäävad ära kuutasud. Suurte puudustena majasisesel on aga märgatavate investeeringute tegemine riistvarasse ning tarkvara litsentside uuendamisse. Lisaks on vajalik süsteemi haldamise oskus ja teadmised [42, 43].

Pilveteenuse puhul võimaldatakse indiviidil ligipääsu personaalsele, äri ja ettevõtte andmestikule läbi interneti. Pilves olev server hoiab ära majasisesed IT lisakulud serveri halduse pealt. Samuti võimaldab pilveteenus äri muutuvatele nõuetele vastavalt pakkuda sobivaimaid lahendusi. Pilveteenuse miinusteks on kindla internetiühenduse olemasolu, hoidmaks ühendust serveriga, igakuised kulud ning ka delikaatsete andmestikke hoiustamise kindlus [42, 43].

Võrreldes pilvelahendusi ja majasiseseid lahendusi, tuleb silmas pidada rahalisi väljaminekuid ning jõudluse küsimusi. Pilve suurima hüvena arvestamiseks selle kasutuselevõttu on ära jääv serverite hankimise ning hooldamise kulu. Pilve puhul, äri vajaduste suurenemisel saab märgatavalt kergemalt ja kokkuhoidvamalt suurendada serveri jõudlust, ei pea rentima või ostma uut serverit. Samas peab kaalutlema ka füüsilise serveri hüvesid, milleks on kogu kontrolli omamine [42, 43].

Vaatamata kogu kontrolli omamisele ning vastavalt soovidele serveri seadistamist, tuleks kaaluda Timetracker 2.0 puhul pilveteenuseid põhjustel, et soov on võimaldada süsteemi



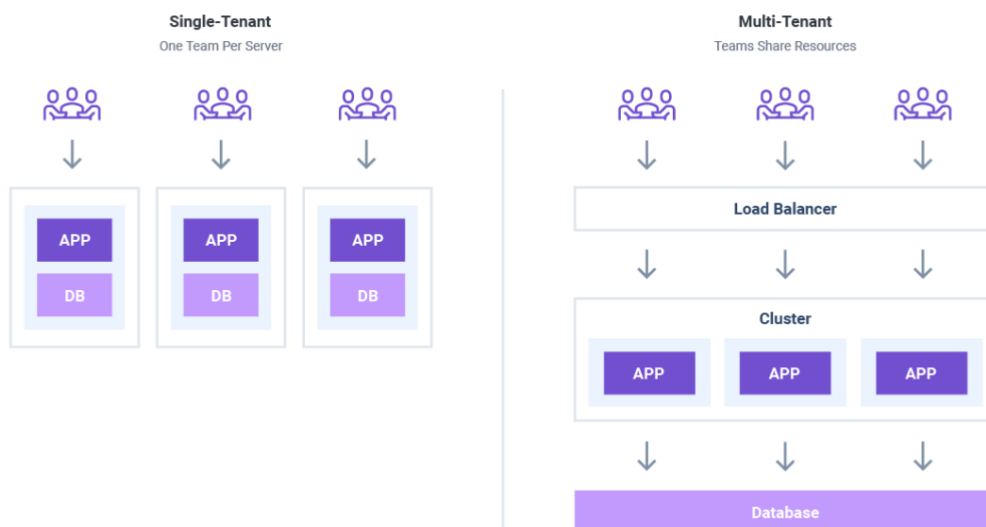
kasutust korraga mitmetele ettevõtetele. See tõstab koormust süsteemile ning majasisese serveriga võivad tekkida jõudluse mured ning vajadused suurendada serveriparki.

Olles valinud välja pilve peale üle mineku, tuleb lahenduse osas analüüsida võimalikke sobivaid süsteemi arhitektuure, kas minna *single-tenant* teed või *multi-tenant* teed.

Vähendamaks maksumust, mis kaasneb rakenduse võimaldamisega mitmetele kasutajate kogumikule, on läinud paljud lahendused üle *multi-tenant* süsteemidele *single-tenant* asemel. Kui *single-tenant* puhul on vaja kindlat ressursi täitmaks iga organisatsiooni vajadusi, siis *multi-tenant* puhul katab rakendus mitmete kasutajaskondade vajadused läbi ühe süsteemi eksemplari (Joonis 6) [44].

*Multi-tenant* puhul on tegemist tsentraalselt hallatava süsteemiga, kus rakendusel on ühine koodipõhi, andmebaas ning jooksutatakse ühtset eksemplari (*instance*) mitmete kasutajate gruppide vahel, suutes jooksutamise ajal igale kasutajaskonnale muuta süsteemi grupi kohandustele vastavaks. Süsteemis on iga grupiga seotud andmestik konfidentsiaalne ning andmestik on andmebaasis seotud kindla grupi kasutajaskonnaga [45, 44, 46].

*Single-tenant* arhitektuuris on igal kliendil oma iseseisev andmebaas ja süsteemi eksemplar. Vastavalt ülesehitusele ei ole sellise lahenduse puhul võimalik jagada ühtegi rakenduse eksemplari klientide vahel [46].



Joonis 6. *Single-Tenant* ja *Multi-Tenant* arhitektuur [46].

Kaaludes erinevaid arhitektuurilisi variante, tuleb silmas pidada äri soove ja vajadusi. *Multi-tenant* ja *single-tenant* puhul on suurimad kaalutavad faktorid ressursi küsimus, andmebaaside haldamine ja rakenduse jagatavus. Investeeringud on märkimisväärselt väiksemad *multi-tenant* puhul kui *single-tenant* lahenduses, uuendused ühele süsteemile mitme aseme tuues uuenduste maksumused alla, uute võimekuste ja täienduste pakkumine korraga ühele mitme asemel. Need on vaid mõned olulisemad aspektid, mis mõjutavad valikut [46].

Vaadeldes aja märkimise süsteemile sobivaimat, siis peab silmas pidama, et *single-tenant* puhul on kõikide gruppide puhul lahendused vaadeldavad kui eraldi tükid, mis tähendab ka nii mitmekordset tööd uuenduste, täienduste, paranduste ja toe osas. Märgatavalt tulusam pikas perspektiivis on võtta kasutusele *multi-tenant* arhitektuur, kuid millega kaasnevad omad nõrkused. Soovil pakkuda suuremaid erisusi süsteemis vastavalt kasutajate grupile ehk ettevõttele, tuleb teha siiski muudatus kõigile. Teatud seadistustega on võimalik eristada kasutajaskondade kaupa võimaldatavaid tegevusi, kuid võib tähendada ebavajalikku võimekust teistele kasutajatele. Lisaks on oht alati, et süsteem ei erista ühe kasutajaskonna andmestikku teise omast ja võivad esineda valeandmestikud või ligipääsud valele eksemplarile rakendusest.

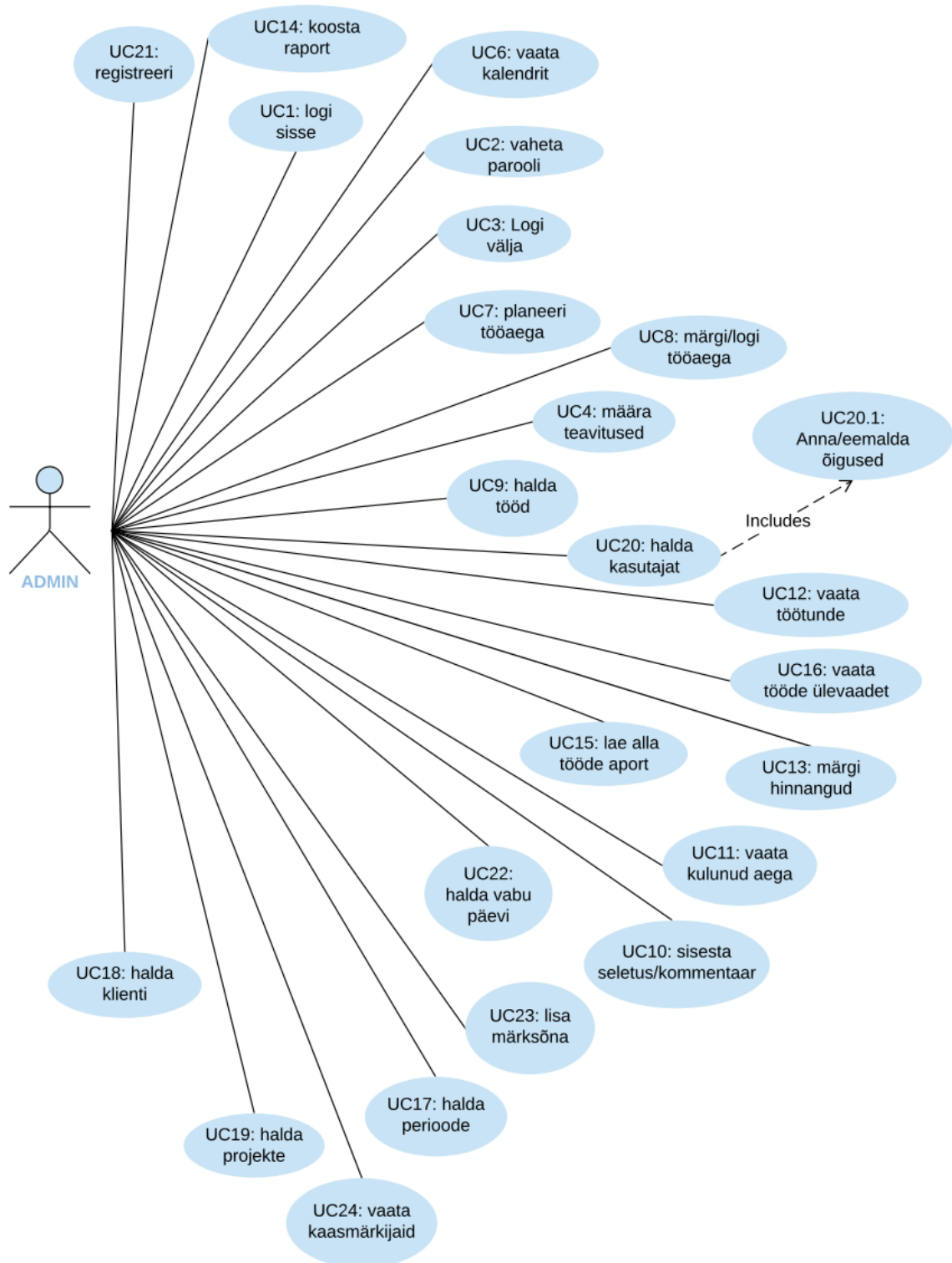
## 6.4 Disain

Aja märkimise süsteemi uuenduse vaatest ning soovist vabaneda tehnoloogilisest võlast, tuleks lähtuda raamistike välja vahetamisest samm-samm haaval uute vastu. Ümberkirjutamisel tuleks aga arvestada ka juba kaardistatud ning antud töös välja toodud paranduskohti ning uusi funktsionaalsusi. Analüüsist tulenevalt võib järeldada, et säärane lähenemine hoiab ära mitmekordse töö, tulenevalt sellest, et suurel määral on *front-end* raamistiku välja vahetamisel tegemist täieliku ümber kirjutamisega ning *back-end* puhul suuremat sorti migreerimisega.

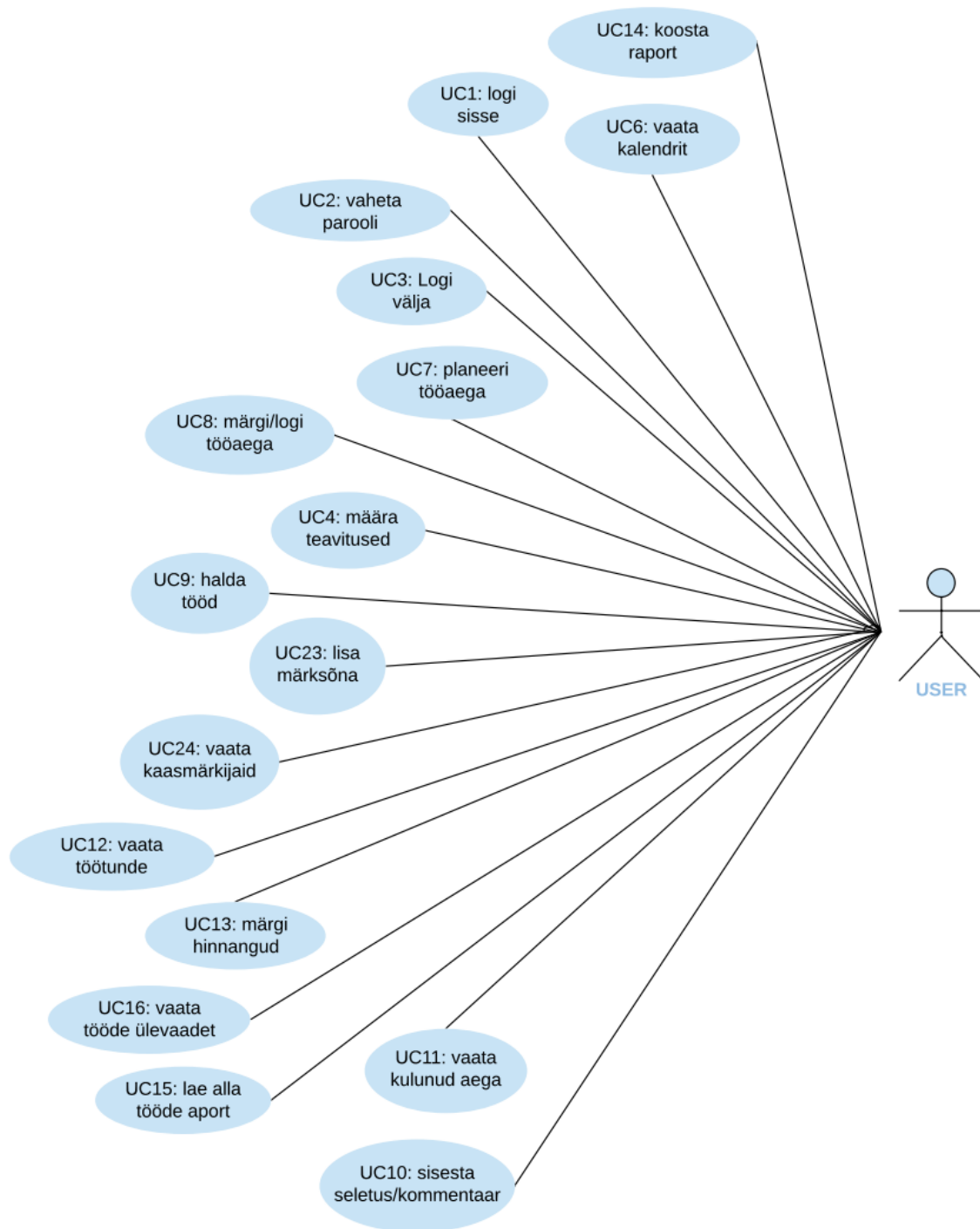
### 6.4.1 Tegevused ja süsteemi disain

Tulenevalt olemasoleva süsteemi analüüsist, ettevõtte töötajate, vastutavate osapoolte kui potentsiaalsete kasutajate vajadustest ja soovidest, tuleb üle vaadata olemasoleva süsteemi tegevused ja kogutav andmestik. Tulenevalt uutest pakutavatest võimekustest ja parandusettepanekutest, on toodud välja uue, parandatud rakenduse (Timetracker 2.0)

kasutusmalliskeemid ehk *use case diagram* nii admin rolli vaatest (Joonis 7) kui user rolli vaatest (Joonis 8).



Joonis 7. Admin kasutaja kasutusmalliskeem (autori loodud).



Joonis 8. User kasutaja kasutusmalliskeem (autori loodud).

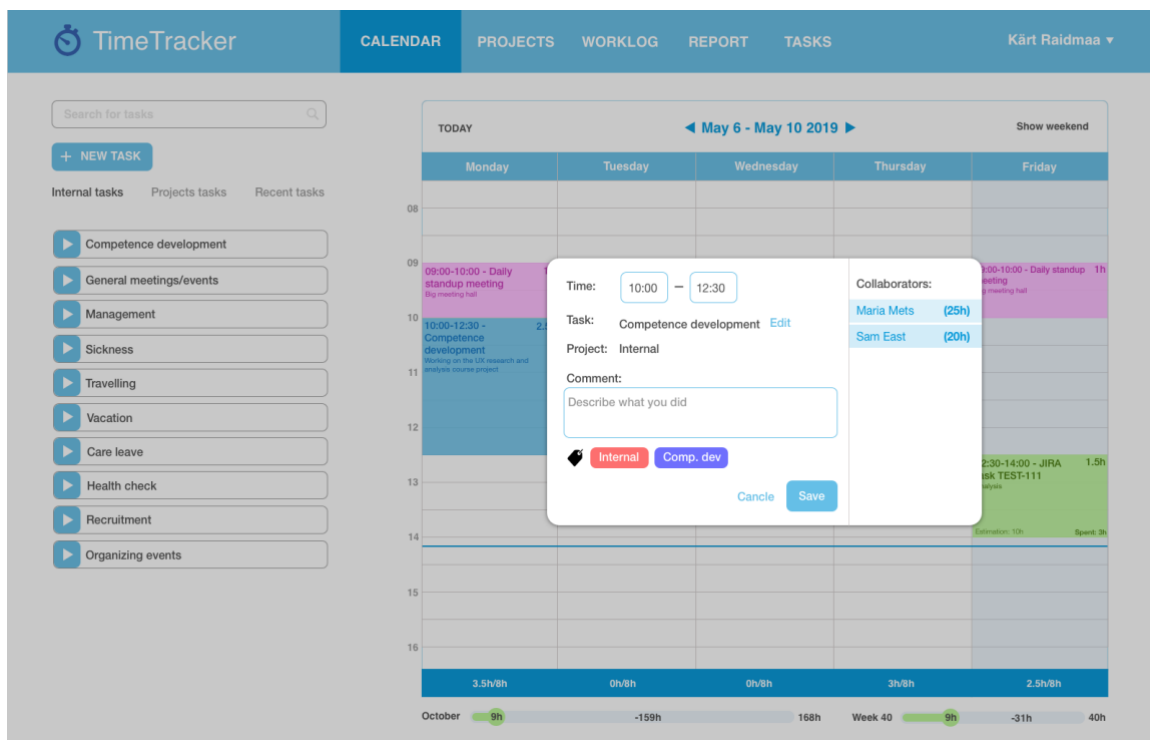
Vastavalt kasutusmalliskeemile saab paika seada süsteemi uue klassiskeemi (Joonis 9), loomaks täpsemat ülevaadet süsteemi disainist. Tulenevalt analüüsist ja ideest pakkuda lahendust teistele ettevõttele tuleb tänast andmestikku täiendada kliendi andmestikuga, autentimise (*login*) andmestikuga ning märksõnade andmestikuga.



## 6.4.2 Kasutajaliides

Süsteemi uuendamisel, pidades silmas rakenduse võimaldamist teistele osapooltele ning kasutust nutiseadmetes, tuleb üle vaadata tänane lahenduse kujunduslik pool ehk kasutajaliidese väljanägemine. Üldise ilme muutmise kõrvalt tuleb silmas pidada ka kasutajakogemuse analüüsist välja tulnud valukohti. Vastavalt analüüsile, paranduste kaardistamisele ja vajadustele, on loodud uue süsteemi võimalikud uued visuaalsed portotüübid.

Tulenevalt kasutajakogemuse uuringute vaatluste käigus välja tulnud murekohast menüü leidmisel ja mõistmisel, soovist näha kaasa löönud isikute nimekirja töö all ning siduda töid märksõnadega, sai muudetud üldist rakenduse väljanägemist ning kalendri vaate võimekust (Joonis 10).



Joonis 10. Timetracker 2.0 kalendri vaate prototüüp (autori loodud).

## 7 Kokkuvõte

Antud magistritöö sai kirjutatud lähtuvalt tööaja arvestamise vajalikkuse kasvust ja aja märkimise süsteemi Timetracker arendanud tarkvaraarendusettevõtte soovist teha süsteem kättesaadavaks teistele ettevõtetele.

Uue süsteemi lahenduste väljapakkumiseks anti kõigepealt ülevaade olemasolevast süsteemist, selle puudustest, aja märkimise süsteemidest turul ja Timetrackeri turu eelistest.

Süsteem on tänaseks vananevatel platvormidel (AngularJS ja Spring Framework 4), mida tuleks süsteemi edasiseks turvaliseks kasutamiseks ja korras hoidmiseks viima üle uutele raamistikele nagu Spring Boot 2 ja Angular 7. Vaadates funktsionaalsusi, siis täidab rakendus oma põhieesmärke, kuid vajab tuge ja kasutajate vajaduste üle vaatamist.

Soovides teha süsteemist toode, tuleb kaardistada murekohad ja leida võimalused pakkumaks kasutajatele lisaväärtust. Peamise turu eelisenä Timetrackeri puhul on lahenduse taga olev tugev tarkvaraarendusettevõte, võimalus pakkuda personaalseid lahendusi ning lihtne ja kitsa funktsionaalsusega rakendus erinevate suurte ja keerukate lahenduste kõrval.

Mõistmaks nii olemasolevate kui ka potentsiaalsete kasutajate vajadusi, viidi läbi töö raames kasutajakogemuse uuring ning analüüsiti tulemeid. Tulemite analüüsist võis järeldada, et tegemist on süsteemiga, mis võiks täita paljude kasutajate vajadusi, kuid vajab funktsionaalsuste parandamist ning uute kaardistatud soovide sisse toomist.

Kasutajakogemuse uuringu tulemused olid aluseks süsteemi probleemide defineerimisel ning uue lahenduse kirjeldamisel. Uue võimaliku lahendusena on *in-house* lahenduse asemel pilveteenus, mis võimaldaks kerget haldust, omaks väikeseid kulusid ning mille puhul oleks kerge hallata kõikide võimalike klientide andmestikke ja funktsionaalsusi.

Vastavalt läbi viidud ülevaadetele, vaatlustele, meetodikatele ja uuringutele, kirjeldati võimalikud süsteemi lahendused ja disain. Kokkuvõttes tuldi antud magistritöös järgmiste tulemusteni:

- Olemasoleva süsteemi analüüs – sobib täna vaid ettevõttesiseseks kasutuseks, kuid turvalisuse ja jõudluse hoidmiseks tuleks vahetada vananevad raamistikud uute vastu.
- Jagatava süsteemi lahendusettepanekud – kaaluda tuleb *in-house* lahenduse kõrval *multi-tenant* ja *single-tenant* pilvelahendusi.
- Süsteemi pakutav disain – uued funktsionaalsused, reeglid, andmestik ja visuaalne prototüüp.



## Kasutatud kirjandus

- [1] actiTIME, "The Complete Guide to Time Tracking," Aprill 2019. [Online]. Available: <https://www.actitime.com/time-tracking/time-tracking-software-essay>. [Accessed 24 Aprill 2019].
- [2] Court of Justice of the European Union, "The Court of Justice of the European Union," 14 Mai 2019. [Online]. Available: <https://curia.europa.eu/jcms/upload/docs/application/pdf/2019-05/cp190061en.pdf>. [Accessed 14 Mai 2019].
- [3] C. Larman, "Craig Larman," 16 September 2004. [Online]. Available: [http://www.craigarman.com/wiki/downloads/applying\\_uml/larman-ch6-applying-evolutionary-use-cases.pdf](http://www.craigarman.com/wiki/downloads/applying_uml/larman-ch6-applying-evolutionary-use-cases.pdf). [Accessed 13 Mai 2019].
- [4] S. Al-ma'adeed, "A New Web-Based Project and Employee Time Tracking System for Project Management," in *European, Mediterranean & Middle Eastern Conference on Information Systems 2010*, Abu Dhabi, 2010.
- [5] "AngularJS: Miscellaneous: Version Support Status," Google, 2018. [Online]. Available: <https://docs.angularjs.org/misc/version-support-status>. [Accessed 05 Mai 2019].
- [6] AltexSoft, "Pros and Cons of Angular Development - AltexSoft," 07 August 2018. [Online]. Available: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/>. [Accessed 06 Mai 2019].
- [7] Google, "AngularJS API Docs," [Online]. Available: <https://docs.angularjs.org/api>. [Accessed 06 Mai 2019].
- [8] K. Gawron, "Long Term Support for AngularJS: What Are Viable Alternatives for Your App?," 02 Juuli 2018. [Online]. Available: <https://www.monterail.com/blog/long-term-support-angularjs-viable-alternatives>. [Accessed 07 Mai 2019].
- [9] R. Prii, "Tööaja arvestus, mis töötab | Lihtsam, kiirem, efektiivsem | begin.ee," Begin OÜ, 19 Märts 2019. [Online]. Available: <https://begin.ee/blog/tooaja-arvestus/>. [Accessed 10 Mai 2019].
- [10] Netsoft Holdings, LLC, "A Complete Guide to Time Tracking | Hubstaff," Netsoft Holdings, LLC, [Online]. Available: [https://hubstaff.com/time\\_tracking](https://hubstaff.com/time_tracking). [Accessed 24 Aprill 2019].

- [11] R. Prii, "Mis on Geofence ja miks on see muutumas uueks trendiks?," Begin OÜ, 04 September 2018. [Online]. Available: <https://begin.ee/blog/geofence/>. [Accessed 24 Aprill 2019].
- [12] "Best Time Tracking Software Apps in 2019 | G2 Crowd," G2 Crowd, Inc., [Online]. Available: [https://www.g2.com/categories/time-tracking#highest\\_rated](https://www.g2.com/categories/time-tracking#highest_rated). [Accessed 11 Mai 2019].
- [13] "Toggl," Toggl OÜ, [Online]. Available: <https://toggl.com>. [Accessed 11 Mai 2019].
- [14] "Scoro," Scoro Software 2019, [Online]. Available: <https://www.scoro.ee>. [Accessed 11 Mai 2019].
- [15] "LawTime," Van Zoig OÜ, [Online]. Available: <http://www.lawtime.ee>. [Accessed 11 Mai 2019].
- [16] "Begin," Begin OÜ, [Online]. Available: <https://begin.ee/>. [Accessed 11 Mai 2019].
- [17] "Remato," Remato Eesti OÜ, [Online]. Available: <https://remato.ee/>. [Accessed 11 Mai 2019].
- [18] "Trello," Atlassian, [Online]. Available: <https://trello.com/>. [Accessed 11 Mai 2019].
- [19] "Jira Software," Atlassian, [Online]. Available: <https://www.atlassian.com/software/jira>. [Accessed 11 Mai 2019].
- [20] "Wemply," Wemply OÜ, [Online]. Available: <https://wemply.com/>. [Accessed 11 Mai 2019].
- [21] "Hours tööaja arvestuse tarkvara," Palvor OY, [Online]. Available: <https://hours.ee/>. [Accessed 11 Mai 2019].
- [22] "Directo.ee - Directo," Directo OÜ, [Online]. Available: <http://www.directo.ee/>. [Accessed 11 Mai 2019].
- [23] D. Rebelo, "Difference between Design thinking and Service Design - Medium," 8 Oktoober 2015. [Online]. Available: <https://medium.com/@diogorebelo/difference-between-design-thinking-and-service-design-35e33044d413>. [Accessed 10 Mai 2019].
- [24] R. Dam and T. Siang, "5 Stages in the Design Thinking Process | Interaction Design Foundation," Interaction Design Foundation, Aprill 2019. [Online]. Available: <https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process>. [Accessed 14 Mai 2019].
- [25] F. Simonds, "Human Centred Design vs Design Thinking vs Service Design vs UX .... What do they all mean?," 8 Juuni 2016. [Online]. Available: <https://www.linkedin.com/pulse/human-centred-design-vs-thinking-service-ux-what-do-all-simonds/>. [Accessed 10 Mai 2019].

- [26] J. Satzinger, R. Jackson and S. Burd, "Systems Analysis & Design in a Changing World, Fifth Edition," Cengage Learning, 2009.
- [27] R. Dam and T. Siang, "What is Design Thinking and Why Is It So Popular? | Interaction Design foundation," Aprill 2019. [Online]. Available: <https://www.interaction-design.org/literature/article/what-is-design-thinking-and-why-is-it-so-popular>. [Accessed 10 Mai 2019].
- [28] R. Dam and T. Siang, "Stage 2 in the Design Thinking Process: Define the Problem and Interpret the Results | Interaction Design Foundation," Interaction Design Foundation, Mai 2019. [Online]. Available: <https://www.interaction-design.org/literature/article/stage-2-in-the-design-thinking-process-define-the-problem-and-interpret-the-results>. [Accessed 14 Mai 2019].
- [29] Martin, "18 Best Idea Generation Techniques - Cleverism," Cleverism, 14 Mai 2015. [Online]. Available: <https://www.cleverism.com/18-best-idea-generation-techniques/>. [Accessed 14 Mai 2019].
- [30] D. Norman and J. Norman, "The Definition of User Experience (UX) - Nielsen Norman Group," [Online]. Available: <https://www.nngroup.com/articles/definition-user-experience/>. [Accessed 08 Mai 2019].
- [31] "User Experience Basics | Usability.gov," Usability.gov, [Online]. Available: <https://www.usability.gov/what-and-why/user-experience.html>. [Accessed 08 Mai 2019].
- [32] J. Visocky O'Grady and K. Visocky O'Grady, A designer's research manual: succeed in design by knowing your client and what they really need., Beverly, Massachusetts: Rockport Publishers, Inc., 2006.
- [33] R. Elmansy, "How to Conduct a Successful UX Research for Your Design Project," Designorate, 18 Aprill 2016. [Online]. Available: <https://www.designorate.com/ux-research-for-design-project/>. [Accessed 05 Mai 2019].
- [34] T. Sharon, "The UX Research Plan That Stakeholders Love — Smashing Magazine," 26 Jaanuar 2012. [Online]. Available: <https://www.smashingmagazine.com/2012/01/ux-research-plan-stakeholders-love/>. [Accessed 26 Aprill 2019].
- [35] "SWOT analüüs ja põhimõtted selle koostamiseks | Leanway," Leanway, [Online]. Available: <https://leanway.ee/swot-analuus/>. [Accessed 11 Mai 2019].
- [36] "User Stories: An Agile Introduction - Agile Modeling," Ambysoft Inc., [Online]. Available: <http://www.agilemodeling.com/artifacts/userStory.htm>. [Accessed 12 Mai 2019].

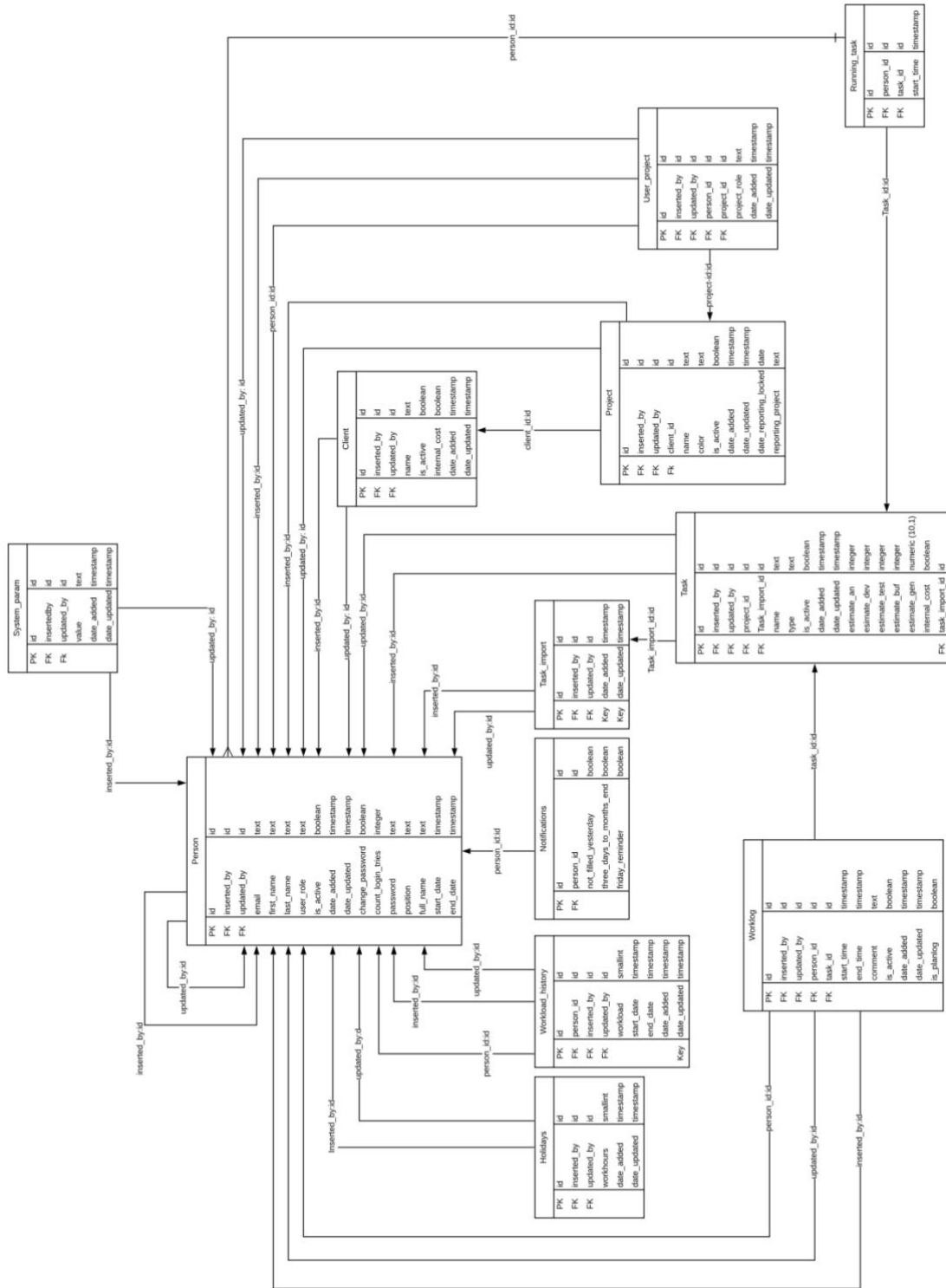
- [37] A. Tiwari, "Migrating Angular 1.x apps to React—The Hybrid Way! - Medium," 15 November 2017. [Online]. Available: <https://medium.com/walmartlabs/migrating-angular-1-x-apps-to-react-the-hybrid-way-3267ccf33755>. [Accessed 12 Mai 2019].
- [38] "Angular - Upgrading from AngularJS to Angular," Google, [Online]. Available: <https://angular.io/guide/upgrade>. [Accessed 12 Mai 2019].
- [39] Umesh, "Migrating from Spring to Spring Boot | Java Development Journal," 30 Märts 2018. [Online]. Available: <https://www.javadevjournal.com/spring-boot/migrating-from-spring-to-spring-boot/>. [Accessed 12 Mai 2019].
- [40] L. Hazlewood, "How We Migrated Our Backend to Spring Boot in 3 Weeks - Stormpath," 23 Mai 2016. [Online]. Available: <https://stormpath.com/blog/spring-boot-migration>. [Accessed 08 Mai 2019].
- [41] A. Lesser, "The Cloud Vs. In-House Infrastructure: Deciding Which Is Best For Your Organization - Forbes," Forbes Technology Council, 25 Juuli 2017. [Online]. Available: <https://www.forbes.com/sites/forbestechcouncil/2017/07/25/the-cloud-vs-in-house-infrastructure-deciding-which-is-best-for-your-organization/>. [Accessed 12 Mai 2019].
- [42] E. Thompson, "In-House Server vs Cloud - What's Best for Business? | Business.org," 21 Veebruar 2014. [Online]. Available: <https://www.business.org/it/data-storage/house-server-vs-cloud-server/>. [Accessed 12 Mai 2019].
- [43] "Cloud vs Dedicated - Which Is Cheaper At Scale? - Technected," 16 November 2016. [Online]. Available: <https://www.technected.com/cloud-vs-dedicated-which-is-cheaper-at-scale>. [Accessed 12 Mai 2019].
- [44] Salesforce.com, Inc., "The Force.com Multitenant Architecture. Understanding the Design of Salesforce.com's Internet Application Development Platform," Salesforce.com, San Francisco, 2008.
- [45] A. Barcelona, "The Benefits of Multi-tenant CMS Solutions - dotCMS," 27 Märts 2019. [Online]. Available: <https://dotcms.com/blog/post/multi-tenant-cms-provides-the-key-to-international-growth>. [Accessed 12 Mai 2019].
- [46] J. Wilson, "5 Reasons to Choose Multi-Tenant SaaS Architecture for Your Application - Medium," Medium, 03 Jaanuar 2019. [Online]. Available: <https://medium.com/swlh/5-reasons-to-choose-multi-tenant-saas-architecture-for-your-application-c823241660dc>. [Accessed 12 Mai 2019].
- [47] M. Knight, "Business Glossary Basics - DATAVERSITY," Dataversity Education, LLC, 10 Jaanuar 2017. [Online]. Available: <https://www.dataversity.net/business-glossary-basics/>. [Accessed 06 Mai 2019].

- [48] "PostgreSQL: The World's Most Advanced Open Source Relational Database," PostgreSQL Global Development Group, [Online]. Available: <https://www.postgresql.org/>. [Accessed 08 Mai 2019].
- [49] R.-J. van Oeveren, "The difference between design thinking and service design | Koos Service Design.," Koos, 17 September 2018. [Online]. Available: <https://www.koos servicedesign.com/blog/difference-between-design-thinking-service-design/>. [Accessed 10 Mai 2019].

## Lisa 1 – Raporti koostamise (UC14) kasutajaloo detailsem kirjeldus *fully dressed* variandis

Kasutuslugu	Kõikidele töödele kulunud tundide kohta raporti genereermine
Number	14
Nimi	Koosta raport
Tegutseja	User/admin
Eeltingimused	Kasutaja (user või admin) on sisse loginud ja raportite lehel
Järeltingimused	Kasutaja on (user või admin) saanud ülevaate tVäödest ja nendele kulunud tundidest
Põhistsenaarium	<p>Kasutaja on aja märkimise süsteemis</p> <p>Kasutaja valib menüüst “Raportid”</p> <p>Kasutaja valib ajavahemiku sisestades alguskuupäeva ja lõppkuupäeva</p> <p>Kasutaja ei täida ühtegi teist lahtrit (tahab ülevaadet kõikidest töödest, tegutsejatest, kirjeldustest, töö tegemise perioodist)</p> <p>Kasutaja vajutab nupule “Genereeri raport”</p> <p>Kasutajale luuakse ülevaade</p>
Alternatiivsed stsenaariumid	<ol style="list-style-type: none"> <li>1. Kasutaja valib lisaks perioodile filtritest ühe või mitu valikut (klient, projekt, töö, kasutaja) <ol style="list-style-type: none"> <li>a. Kasutajale kuvatakse info raportis vastavalt filtri valikutele</li> </ol> </li> </ol>
Nõuded	Kasutajal peab olema interneti ühendus ja süsteemi ligipääs
Märkused	Raportid genereeritakse kasutajale vastavalt talle antud õigustele ja ligipääsudele. Admin puhul on tegemist kasutajaga, kellel on õigused kogu informatsioonile. User puhul on tegemist kasutajaga, kes pääseb ligi vaid , projektide ja klientide tööde informatsioonile, millele tal õigused. User kasutaja ei näe informatsiooni teiste kasutajate kohta.

# Lisa 2 – Timetracker olemasoleva süsteemi andmebaasi ERD



Olemasoleva süsteemi andmebaasi ERD.

# Lisa 3 – Ajamärkimissüsteemi kasutatavuse uuringu plaan

UX RESEARCH PLAN

Last Updated On:  
03.10.2018

## TIMETRACKER - time tracking system study

### Author and Stakeholders

**Author:** Kärt Raidmaa, Analyst

**Stakeholders:** CEO, Business Analyst, UX analyst

### Background

At company X we use Timetracker application to mark down the work hours.

Its a application developed by the company and it gives the user a chance to mark the exact time for that exact task in hand. By that it gives the user an overview of the time spent on a certain task or job. It shows the user the hours needed in a month and a week and also the progress of how many hours of it is already done.

The user can search for jobs or tasks already entered into the system and also add new tasks or jobs that are being worked on.

USer can also add the jobs and tasks under certain projects and take out reports of times spent on project or just a certain task.

Until now it has only been in use by the people working in the company. Now the interest is to make it into a service that can be shared with other companies alike who need an application that would give them a chance to mark down work hours as accurately as possible and get a better overview of hours spent.

### Research Questions

What operating system do you use?

What kind of application do you use for marking down your work hours?

How do you report your work hours?

Do you mark down the exact job or task you work on?

How do you get an overview of the hours spent on a certain job?

How important it is to get an accurate overview of the hours spent on an exact job or project?

What would you want an ideal time tracking application to have (that you missing right now)?

### Methodology

**Stakeholders Interviews** - Since the main goal is to get Timetracker into a state where it is an actual product that can be sold to others we need to start with interviewing the stakeholders. Get to know the needs and goals of stakeholders by making it into a product. What do the stakeholders want?

**Contextual Inquiry** - As it is an application that is in use daily in the company itself it gives the opportunity to view the users using it first-hand. By observing the users using it daily and then by what was seen have a 1-on-1 interview with the user. It would give a better understanding of the application overall, what are the positive sides, what are the negative sides, what should be changed and what do they like the most about the system.

**User Interviews/Focus Group** - Finally a focus group, possible users interviews should be held out. To get a better understanding what people use in other workfields to mark down their work hours and if there is a need for another time tracking application. What are the needs of an user and what are the expectations? What would make them choose the new application instead of the one in use now?

### Participants

Company X employees

Office and business people between the age 24-60.

People who need to mark down work hours.

People who need to mark down work hours by certain jobs or tasks.

Companies and people who need to get an accurate overview of the hours spent on an exact project or job.

IT companies, law offices, hospitals, marketing companies etc

### Schedule

Week 1: Interviews with the stakeholders, observations and 1-on-1 interviews with the Timetracker application users.

Week 2: User Interviews with possible users.

Week 3: Analysis of the interview results and put a report together, go over it with the stakeholders and come to a conclusion.

Ask yourself: What is the outcome? Is there a place for that application? Are there enough potential users? What needs to be changed? How to stand out from all the other competitors?

Kasutajakogemuse uuringu plaan.