

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Artur Toikka IABB179154

**INFOSÜSTEEM TEENUSTE PAKKIJATELE JA KLIENTIDELE  
ILUTEENUSTE VALDKONNA NÄITEL**

Bakalaureusetöö

Juhendaja: Mart Roost  
Infotehnoloogia  
magister

Tallinn 2020

## **Autorideklaratsioon**

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

Autor: Artur Toikka

18.05.2020

## **Annotatsioon**

Käesolev töö on tehtud bakalaureuse lõputöö raames. Töö hõlmab üldiselt kahte valdkonda, milleks on ärianalüüs ja tarkvara arendus. Töö keskendub iluteenuste turu uurimisele broneerimissüsteemide vaatenurgast. Eesmärk on tuvastada olukord turul broneerimissüsteemidega ning leida puudused ja leida võimalikult efektiivne lahendus probleemile. Lahendus probleemile on tarkvaraline ja sisaldab endas süsteemi nõuete paika panemist lähtuvalt uuritud probleemi leitud tulemustest, süsteemi kavandamist, arhitektuuri otsuste tegemist, andmebaasi kavandamist, programmeerimistööriistade valimist ja töötava prototüübi valmistamist. Töö tulemusel valmis toimiv broneerimissüsteemi prototüüp, mis on antud töös kirjeldatu ja dokumenteeritud.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 27 leheküljel, 9 peatükki, 15 joonist.

## **Abstract**

### **Information System for service providers and clients on the example of beauty services**

This work has been done as a bachelor's thesis. The work generally covers two areas, which are business analysis and software development. The work focuses on the study of the beauty services market from the perspective of reservation systems. The aim is to identify the situation in the market for reservation systems and to find shortcomings and find the most effective solution to the problem. The solution to the problem should be software system. Reservation systems were analyzed mainly from Estonia, US and UK. Based on the results of the analysis were made decisions about the design of the prototype that was later built. Reservation system making includes setting the system requirements based on the results of the researched problem, designing the system, making architectural decisions, designing the database, choosing programming tools and making a minimal valuable product as a prototype. Tools that were used to create the reservation system were mainly from Microsoft. For the server language was used C# and .NET Framework Core, then for database was used Entity Framework Core and user interface was built using the Blazor framework. As a result of the work, a working prototype of the reservation system was created, which is described and well documented with database schema, screenshots from user interface and my explanations. This work also this work contains the link to the prototype code in the extras at the end pages.

The thesis is in Estonian and contains 27 pages of text, 9 chapters, 15 figures.

## Lühendite ja mõistete sõnastik

MVP	Minimum viable product, minimaalne kasulik toode
DRY	Don't repeat yourself, ära korda ennast
Clean Code	Puhas kood, (puhta koodi printsiip)
DAL	Data access layer, andmeedastus kiht
Application layer (Core)	Rakenduse kiht
UI	User interface, kasutajaliides
CRUD	Create, read, update, delete – lisa, loe, uuenda, kustuta
IDE	Integrated development environment, integreeritud arenduse keskkond

## Sisukord

Autorideklaratsioon .....	2
Annotatsioon.....	3
Abstract Information System for service providers and clients on the example of beauty services .....	4
Lühendite ja mõistete sõnastik .....	5
Sisukord.....	6
Jooniste loetelu .....	8
1 Sissejuhatus .....	9
2 Olemasolevad lahendused .....	10
2.1 Välismaal .....	10
2.2 Eestis.....	11
3 Mida kavatsen teha teisiti, kui olemasolevad .....	12
4 Metoodika.....	12
5 Analüüs.....	13
5.1 Probleemi valideerimine.....	13
5.2 Lahendus probleemile.....	13
5.3 Potentsiaalsed kliendid .....	14
5.4 Potentsiaalsed probleemid .....	15
5.5 Mida saab teenusepakkuja? .....	15
5.6 Mida saab teenuse kasutaja?.....	16
5.7 Sissetulekud .....	16
5.8 Väljaminekud.....	16
5.9 Konkurents.....	17
5.10 Järeldused .....	17
6 Ärianalüüs.....	18
6.1 Väärtusvahetuste skeem .....	18
6.2 Äriprotsessid.....	19
6.3 Äritransaktsioonid .....	19
6.4 Kasutusjuhud .....	20

6.5 Domeenimudel.....	22
7 Prototüüpimine .....	23
7.1 Lahenduse disain .....	23
7.2 Andmebaasi mudel .....	24
7.3 Äriloogika tarkvaraline realiseerimine .....	25
7.3.1 Kasutajate rollid ja õigused .....	25
7.3.2 Valdkonnad ja teenused.....	25
7.3.3 Otsing .....	26
7.3.4 Broneerimine .....	26
7.3.5 Kalender .....	26
7.4 Kasutajaliides.....	27
7.4.1 Valdkondade lisamine .....	27
7.4.2 Teenuste lisamine .....	28
7.4.3 Profiili andmete haldamine.....	28
7.4.4 Teenuste lisamine profiilile .....	29
7.4.5 Otsing .....	30
7.4.6 Broneerimine .....	30
7.4.7 Kliendi broneeringud.....	31
7.4.8 Broneeringud teenusepakkujale.....	32
7.5 Suhtlus andmebaasiga.....	32
8 Analüüs ja järeldused.....	32
8.1 Tulemused .....	32
8.2 Alternatiivid.....	33
8.3 Piirangud.....	33
8.4 Võrreldes teistega .....	33
8.5 Äriline põhjendatus.....	34
8.6 Edasised tööd.....	34
9 Kokkuvõte .....	35
Kasutatud kirjandus .....	36
Lisa 1 – Koodi meetrikad .....	37
Lisa 2 – Link koodile.....	37

## Jooniste loetelu

Joonis 1. Väärtusvahetuste mudel .....	18
Joonis 2. Funktsionaalsed ärieesmärgid ehk äriprotsesside struktuur .....	19
Joonis 3. Äritransaktsioonide mudel .....	19
Joonis 4. Kasutusjuhtude mudel .....	20
Joonis 5. Domeenimudel .....	22
Joonis 6. Andmebaasi mudel .....	24
Joonis 7. Valdkondade lisamine .....	27
Joonis 8. Teenuse lisamine .....	28
Joonis 9. Profiili haldamine .....	28
Joonis 10. Profiili teenuste list.....	29
Joonis 11. Uue teenuse lisamine.....	29
Joonis 12. Otsing .....	30
Joonis 13. Broneerimine .....	30
Joonis 14. Kliendi broneeringud.....	31
Joonis 15. Broneeringud teenusepakkujale .....	32



# 1 Sissejuhatus

Antud töö eesmärk on valideerida ja teha valmis minimaalne kasutatav tarkvaraline mudel (MVP). Infosüsteem teenuste pakkujatele ja klientidele iluteenuste valdkonna näitel.

Hetkel on turu probleem selles, et iluteenuste maailmas on palju eraettevõtjaid, kes kasutavad klientide leidmiseks sotsiaalvõrgustikke ning sealseid kommuune, see teeb suhteliselt ebamugavaks ülevaate saamise oma kliendibaasi üle ning graafiku haldamiseks tuleb samuti kasutada veel eraldi tööriistasid.

Samas olukorras on ka teenuste kasutajad, kes peavad erinevate teenusepakkujatega suhtlema erinevate kanalite vahendusel.

Soov on teha süsteem, mis ühendaks omavahel teenuste pakkujad ning kliendid ja võimaldaks mugavalt koostada töögraafikuid ning pidada arvet statistika üle. Samuti ideaalis võiks süsteemis olla realiseeritud ka arveldamise võimalus, selleks et ka raamatupidamine oleks samas kohas ja lihtsal kujul kättesaadav.

Visiooni realiseerimiseks on tarvis valmis teha portaal, mille kaudu spetsialistid saavad pakkuda oma teenuseid klientidele, kes omakorda selle sama süsteemi kaudu ka saavad kerge vaevaga leida enda jaoks sobiva teenusepakkuja ning seal ka kohe broneerida teenuse osutamise aeg.

Portaalis peab olema võimalus otsida teenusepakkujaid tegutsemise sfääri järgi ja pakutavate teenuste järgi. Samuti otsingu tulemusi sorteerida hinna järgi ning näha teiste klientide tagasisidesid teenusepakkujale.

Teenuse osutajal peab olema oma profiil, kus on loetletud tema poolt pakutavad teenused, ülevaade tema töögraafikust ja vabadest aegadest, mida klient saab broneerida ja teiste teenust kasutanud klientide tagasisided teenuse kohta.

Tulemuseks peaks olema valideeritud äriidee kontseptsioon ja detailanalüüsi mudelid ning minimaalne kasutatav prototüüp kirjeldatud süsteemist.

Eesmärgini jõudmiseks on kavas analüüsida ja valideerida äriideed Lean Startups metoodika järgi. Prototüübi valmistamiseks kavatsen kasutada Microsofti .NET platvormi, C# programmeerimiskeelt, Entity Framework Core'i ning muid tööriistaid, mida läheb realiseerimisel tarvis.

## 2 Olemasolevad lahendused

On mitmeid olemasolevaid lahendusi, kuid enamasti nad on realiseeritud kas ühe konkreetse teenusepakkuja poolt enda teenuste pakkumiseks või lihtsalt keskendudes ühele konkreetsele erialale näiteks juuksurid või maniküürisalongid. Samuti suureks puuduseks peale süsteemide killustatuse on see, et isegi kui süsteem on kasutatav erinevate teenusepakkujate poolt, siis on see orienteeritud välismaa turule, ning Eestis ei ole nad kasutusel või ka ei ole kasutatavad. Võimalik, et mingisuguste seaduslike piirangute tõttu või lihtsalt pole piisavalt ressursi, et tulla Eestimaa turule.

### 2.1 Välismaal

Inglismaa moeajakirjas Bazaar on toodud list nende sõnul parimatest äppidest iluturul [1]. Enamik neist ongi orienteeritud vaid Inglismaa siseturule ja isegi spetsiifiliselt Londonile. Mis veel paistab kohe silma, on see fakt, et suurem enamus neist positioneerib end *high-end* teenusepakkujatenäna, mis ilmselgelt on eemaletõukav keskklassi kasutajate jaoks.

Mõned äpid võimaldavad süsteemi siseseid makseid realiseerida, kuid siis on puuduseks see, et on piiratud valik teenuseid ning jällegi selgelt *high-end* klientidele orienteeritud. Nagu näiteks LeSalon Beauty äpp [2], mis töötab vaid Londonis, selle lähistel, Manchesteris ja Prantsusmaal.

Samuti tundub, et äpid võtavad protsente teenusepakkujate tehingutelt, ning äpi Urban [3] kohta on kirjas, et see võimaldab teenuse osutajatel pidada oma isiklikku karjääri

andes võimaluse võtta 72% tellimustest endale. Ehk Urban on teenuse osutamise võrgustik, mitte platvorm iseseisvate teenuse osutajate jaoks.

ValuezAppz [4] on platvorm, mis võimaldab lihtsalt teha enda äri jaoks äppi kasutades lihtsat *drag and drop* tehnoloogiat ning seejärel publitseerida seda Windows'i, Google, ja Apple äppide poodides. Realiseeritud on juba andmete kogumine, erinevad lojaalsusprogrammide integreerimise võimalused. Puuduseks on loomulikult see, et teenus on tasuline, oma äpi reklaamimisega peab tegelema ise ning jällegi sellise realiseerimise tagajärjel tekib killustatus ja ebamugavus kliendi jaoks. Samuti seda kõike peab ka haldama ja uuendama ning kui kliendil tekib probleem äpi kasutamisega, siis peab äpi loomise teenuse kasutaja ise sellega tegelema ning tõenäoliselt kontakteeruma äpi teenuse osutajaga, mis toob kaasa teatava keerukuse, kuna probleemist probleemi lahendajani lülide arv kasvab ning lahenduse ooteaeg sellevõrra pikeneb.

Veel üks äpp jällegi Inglismaalt on Ruby, päris hästi realiseeritud minimalistlik ja minimaalselt klikke nõudev broneerimissüsteem, kuid kohe avalehel broneeringu akna kohal on kirjas „Five star beauty services direct to your door“, mis kohe vihjab, et tegemist on pigem kallima teenusega. Samuti äpp on orienteeritud vaid Londonile.

<https://ruby.com/>

## 2.2 Eestis

Postimehes nägin 2019 aasta artiklit Fleetl äp [5]i kohta, mis peaks viima mugavalt kokku iluprofessionaalid ja kliendid, kuid link, mis viis äpi kodulehele ei töötanud.

BookLux [6] on Eesti turule orienteeritud äpp, mis pakub lahendusi erinevate sfääride ettevõtetele broneerimissüsteeme.

Veel üks Eesti turule orienteeritud äpp on Finde [7], kuid see on tehtud salongide jaoks.

IluBooking [8] nagu ka Finde on tehtud Eesti turu jaoks, kuid pakub oma teenuseid üksnes salongidele.

### **3 Mida kavatsen teha teisiti, kui olemasolevad**

Minu nägemus valdkonnast paljudes punktides langeb kokku olemasolevate lahenduste poliitikaga, kuid minu soov ei ole luua salongide võrku ega monopoliseerida iluteenuste turgu, vaid luua eraldiseisev platvorm võimaldamaks eraisikutest teenusepakkujatel pakkuda oma teenuseid eraisikutest klientidele eeskätt iluteenuste valdkonnas ning ideaalis laiendada sfääride haaret, et ka muudel aladel teenuseid pakkuvad spetsialistid saaksid oma teenuseid samal platvormil pakkuda. Võimalusel sooviksin säilitada spetsialistide seas orgaanilist konkurentsi, sekkumata kasutaja teenusepakkuja otsinguprotsessi pakkudes konkreetset teenusepakkujat, vaid lastes kliendil teha oma valik lähtudes spetsialistide profiilidest ja neile jäetud tagasisidest teiste klientide poolt. Samuti minu soov on realiseerida toode nii, et maksmine toimuks süsteemi siseselt võimaldades pidada puhast raamatupidamist teenusepakkuja jaoks, et seda hõlbustada soovin realiseerida süsteemi eest tasumise teenuse pakkuja jaoks fikseerituks ajaperioodi jooksul ega võtaks protsente sissetulekult. Minu ootuste kohaselt peaks selline meede elimineerima kiusatuse arveldada klientidega sularahas ning hoiaks ära ka võimalikud küsimused maksuameti poolt spetsialistile, kes saaks iga hetk kätte väljavõtte oma tehingutest, mille pealt oleks näha kõik tema sissetulekud.

### **4 Metoodika**

Lahendust hakkan realiseerima toetudes Lean Startup metoodikale [9] selleks, et teekond eesmärgini oleks maksimaalselt kiire ja otsene.

Lean startup on metoodika äride ja toodete arendamiseks, mille eesmärk on arenduse tsükleid lühendada ja kiiresti kindlaks teha, kas pakutud ärimudel on rakendatav. Seda saavutatakse juhindudes ärihüpoteesidest eksperimenteerimisel, korduvalt toote versioone välja lastes ja valideeritud õppimisega.

## **5 Analüüs**

Analüüsid keskendun ennekõike olemasoleva probleemi valideerimisele, potentsiaalsete klientide huvi olemasolu lahenduse vastu hindamisele ja antud valdkonna turul olemasolevate lahenduste uurimisele.

### **5.1 Probleemi valideerimine**

Tänapäeva iluteenuste turg on üsna mitmekesine ning inimesed seisavad tihti probleemi ees, kuidas leida õige teenusepakkuja sobivas kohas, sobival ajal ja sobiva hinna ning kvaliteedi suhtega. Kuigi paljudel inimestel on kujunenud ajaga välja oma personaalne iluteeninidaja, siis vaatamata sellele esinevad siiski olukorrad, kui teenusepakkuja, kellega klient on harjunud, on puhkusel, haige või tal pole pakkuda sobivat aega. Samuti tuleb elus tihti ette olukordi, mil mõnda teenust on vaja just täna, kohe ja praegu. Kuid kelle juurde siis minna või kust leida omale sobiv teenusepakkuja? Siinkohal ilmneb probleem, et kliendid ei tea kust otsida omale sobivat teenusepakkujat. Sotsiaalmeediast võib küll leida gruppe pealkirjadega “Ilutegijate pakkumised klientidele” või “Eesti iluteeninidajate pakkumised ja kontaktid” ning veel paljugi muid sarnaseid, kuid nendest foorumitest sobiva teenusepakkuja leidmine on ajakulukas, keeruline ja riskantne. Tihti on ka iluteenuste pakkujate probleemiks korraliku broneerimissüsteemi väljatöötamine ning seetõttu seisavad kliendid endale sobiliku aja broneerimise probleemi ees. Võib juhtuda, et iluteeninidaja ei vasta kõneledele, sõnumile vastuse ootamine võtab kaua aega või ei leia klient üldse viisi, kuidas teenusepakkujaga ühendust saada. Lisaks võib teenusepakkujal puududa efektiivne kanal oma potentsiaalsete klientideni jõudmiseks. Teenusepakkujate suureks probleemiks on ka see, et aja varasemalt broneerinud kliendid ei ilmu kohale, seega kaotab teenusepakkuja ajas ja ka raha jääb saamata.

### **5.2 Lahendus probleemile**

Selleks, et lahendada antud probleemi, tuleb välja töötada broneerimissüsteem, kuhu oleks koondunud kõik Eesti iluteenuste pakkujad. Nendeks on juuksurid, küünetehnikud, massöörid, depilatsiooni tegijad, augustajad, ripsmetehnikud, tatoveerijad, kulmutehnikud, jumestajad ja kosmeetikud. Antud süsteemi kasu teenusepakkujate jaoks seisneb selles, et langeb ära vajadus välja mõelda süsteemi oma

graafiku ja klientide dokumenteerimiseks. Väheneb ka vajadus klientidega suhtlemiseks kirja või telefoni teel eesmärgiga aega kirja panna. Teenuseosutaja ei pea enam liigset vaeva nägema klientide lisamisega oma graafikusse, tema ülesandeks jääb vaid kinnitada broneering. Samuti ei jää teenusepakkujad ilma klientide mitte-kohale ilmumise tasust. Lihtsamaks muutub ka klientide elu, kui broneerimissüsteemi on koondunud kokku suurem hulk hetkel turul tegutsevaid iluteenuste pakkujaid. Süsteemi kaudu saab klient kiiresti ja lihtsalt üles leida iluteenindajate vabad ajad, võrrelda teenuseid asukoha, hinna, sobiva aja ning muude lisateenuste järgi. Lisaks on võimalik näha pilte tehtud tööd, leida iluteenindaja kontakt või broneerida sobilik aeg - seda kõike vaid paari klikiga.

### **5.3 Potentsiaalsed kliendid**

Kuna antud lahendus on keskendunud iluteenuste maailmale, siis kõige tõenäolisemateks klientideks süsteemile oleksid:

- Juuksurid
- Küünetehnikud
- Massöörid
- Depilatsiooni tegijad
- Augustajad
- Ripsmetehnikud
- Tatoveerijad
- Kulmutehnikud
- Jumestajad
- Kosmeetikud
- Iluteenuste kasutajad

Tõenäoliselt on broneerimissüsteem tänu oma olemusele laiendatav ka muudele valdkondadele, kuna ei sisalda otseselt mingisugust spetsiaalselt valdkonnaspetsiifilist loogikat.

## **5.4 Potentsiaalsed probleemid**

Asjaolud, mis võivad olla potentsiaalsete klientide jaoks takistuseks ehitatava süsteemi kasutamiseks on näiteks:

- Teenuse pakkujal on väljaaretatud oma broneerimissüsteem ja välja kujunenud kliendibaas.
- Teenuse pakkuja ei ole vastuvõtlik uuendustele ja on kinni vanades traditsioonides
- Teenuse pakkuja ei kasuta arvutit ega nutitelefoni, seega pole tal võimalik oma profiili süsteemi luua.
- Teenuse pakkuja ei tea midagi süsteemi olemasolust.

## **5.5 Mida saab teenusepakkuja?**

Kasu, mille saab teenusepakkuja antud süsteemist on tõenäoliselt piisavalt suur, et kaaluda süsteemi kasutamist.

- Lihtne broneerimissüsteem broneeringute haldamiseks
- Väheneb telefoni teel broneeringute tegemine
- Väheneb sõnumivestlus kliendile sobiva aja leidmiseks
- Väheneb ummistunud postkast peale puhkust või haiguslehel viibimist
- Tagatis juhul kui klient ei ilmu kirjapandud ajal kohale

## 5.6 Mida saab teenuse kasutaja?

Nagu teenusepakkujate mured, nii ka klientide mitmed mured saavad antud süsteemi kasutamise abil vähendatud:

- Tekib mugav võimalus leida suuremat hulka turul tegutsevaid iluteenuste pakkujaid ühest kohast
- Aja kokkuhoid, kuna sobiva teenuse broneerimine on vaid paari kliki kaugusel
- Võimalus leida iluteenuse pakkujat asukoha järgi
- Võimalus võrrelda teenusepakkujaid hinna järgi
- Võimalus võrrelda teenusepakkujaid teenuse osutamise aja järgi
- Võimalus näha teenuse pakkuja kõiki vabu aegu ühest kohast
- Võimalus näha kõigi teenuse pakkujate reitinguid ja klientide tagasisidet
- Võimalus tühistada juba broneeritud aeg ilma isikliku suhtlusega

## 5.7 Sissetulekud

Loomulikult süsteemi on vaja üleval pidada, edasi arendada ning vajaduselt ka toetada. See kõik loomulikult ei ole tasuta, seega isegi lihtsalt eksisteerimiseks peab süsteem tooma sissetulekut. Siin on mõned tõenäolised sissetulekuallikad süsteemi jaoks:

- Teenuse pakkuja maksab fikseeritud tasu kuus broneerimissüsteemi kasutamise eest
- Reklaam süsteemis näiteks Google Ads
- Muu reklaamiklient, kes soovib oma reklaami broneerimissüsteemis näidata

## 5.8 Väljaminekud

Süsteemi ühekordsest arendusest ei piisa, seda on vaja pidevalt üleval hoida ning selleks on paratamatult vaja teha kulutusi.



- Süsteemi serverite eest on vaja maksta
- Arendustöö kulud
- Võimalikud lisakulud teenuste integreerimise eest
- Domeeninime kulud

## **5.9 Konkurents**

Otsene konkreetne konkurent täpselt samasuguse tootega Eesti turul puudub, pigem on konkurentsi pakkuvad Facebooki kommuunid ja Instagram, platvormid, mille kaudu paljud eraettevõtjad ja nende kliendid on harjunud töötama.

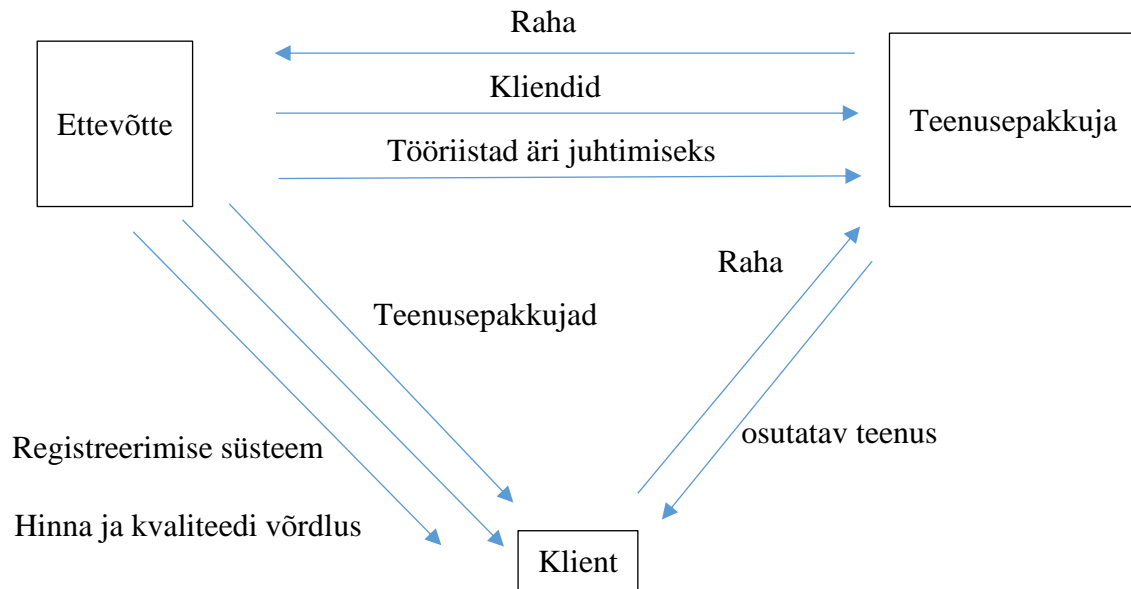
## **5.10 Järeldused**

Antud broneerimissüsteem lihtsustab märkimisväärselt nii teenuste kasutajate kui ka teenuste pakkujate elu. Teenuste pakkujatel on lihtsam hallata klientide broneeringuid ühest kohast, mis hoiab märkimisväärselt kokku aega klientide kõnede ja sõnumitele vastamiseks ja arvelt. Ka langeb ära probleem raha kättesaamisega klientidelt, kes broneeritud ajaks etteatamata kohale ei ilmu. Samuti pole enam vaja muretseda puhkuse või haiguse ajal klientide kirjadele ja kõnedele vastamise pärast.

Teenuste kasutajatel on võimalus leida suur hulk iluteenuste pakkujaid ühest kohast. Tekib võimalus võrrelda iluteenindajaid hinna, asukoha ja graafiku alusel. Sobivat aega iluteenindaja juurde on võimalik leida kiiremini ja ei pea muretsema enam vastuseta jäänud kõnede ja sõnumite pärast.

## 6 Ärianalüüs

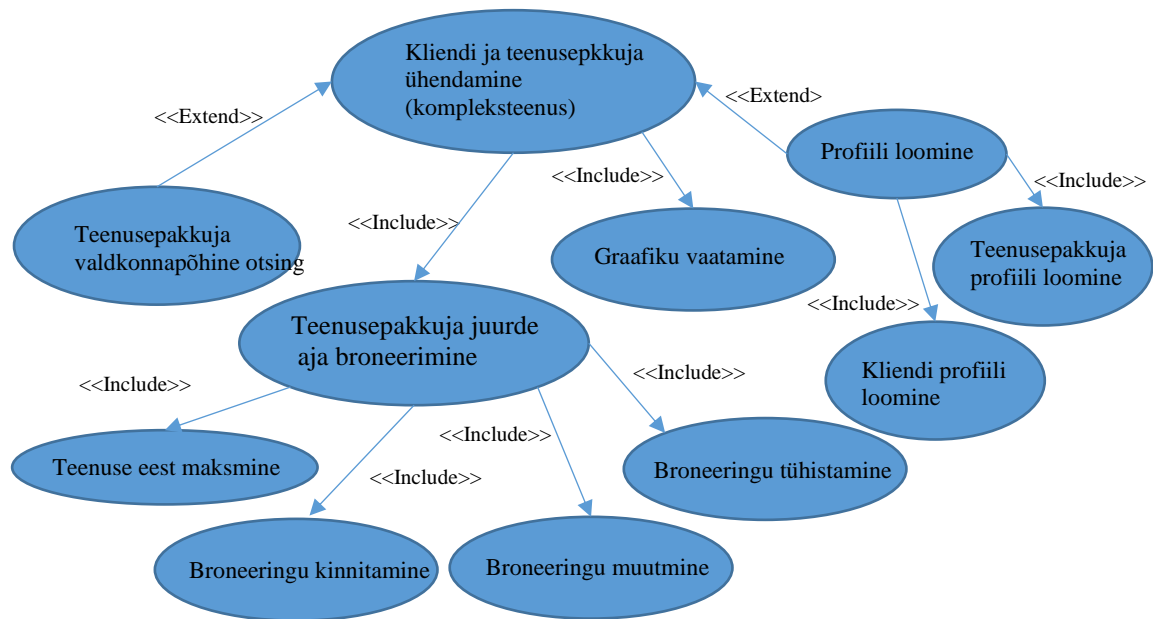
### 6.1 Väärtusvahetuste skeem



Joonis 1. Väärtusvahetuste mudel

Antud skeemil (Joonis 1) on näha kuidas osapoolte poolt pakutavad väärtused liiguvad osapoolte vahel, ja kõik toovad kõigile selles mudelis mingisugust kasu. Võib tunduda justkui ettevõtte ei saaks kliendilt otse mittemingisugust kasu, kuid just tänu kliendi olemasolule kasutab teenusepakkuja ettevõtte teenuseid ja vastupidi. Teenusepakkuja jaoks on peamine väärtus, mida ta ettevõtte käest saab kliendibaasi olemasolu.

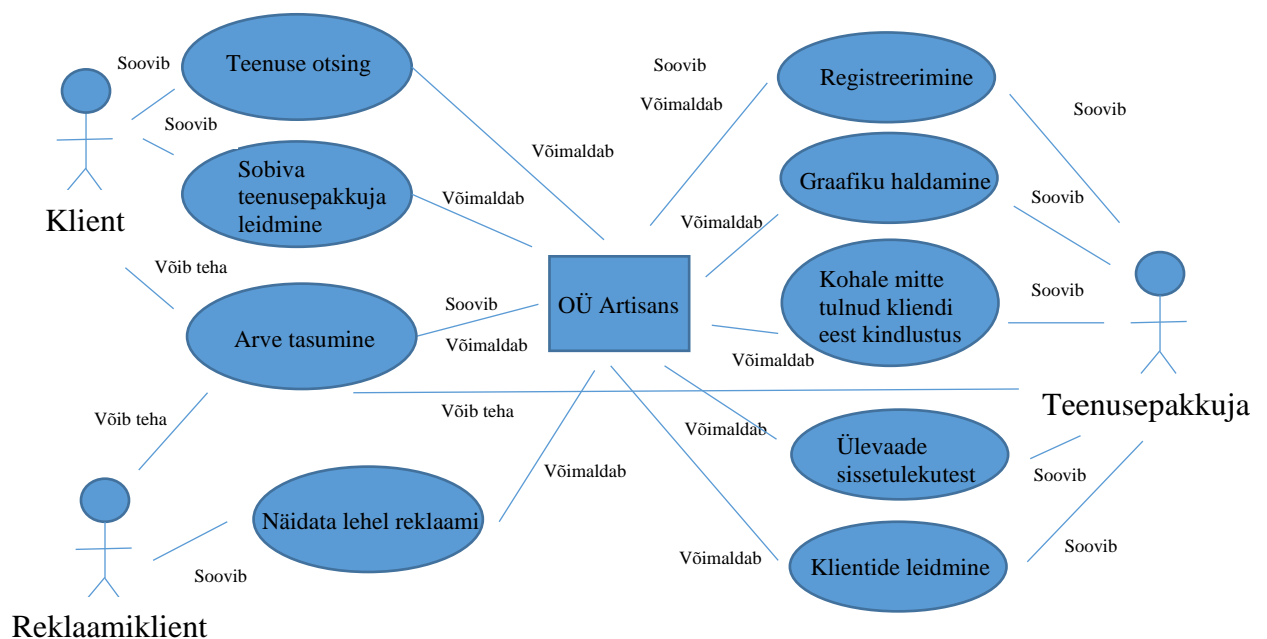
## 6.2 Äriprotsessid



Joonis 2. Funktsionaalsed ärieesmärgid ehk äriprotsesside struktuur

Antud skeemilt (Joonis 2) on näha seosed äriprotsesside vahel. Välja joonistuvad põhiprotsessid ja neid toetavad protsessid.

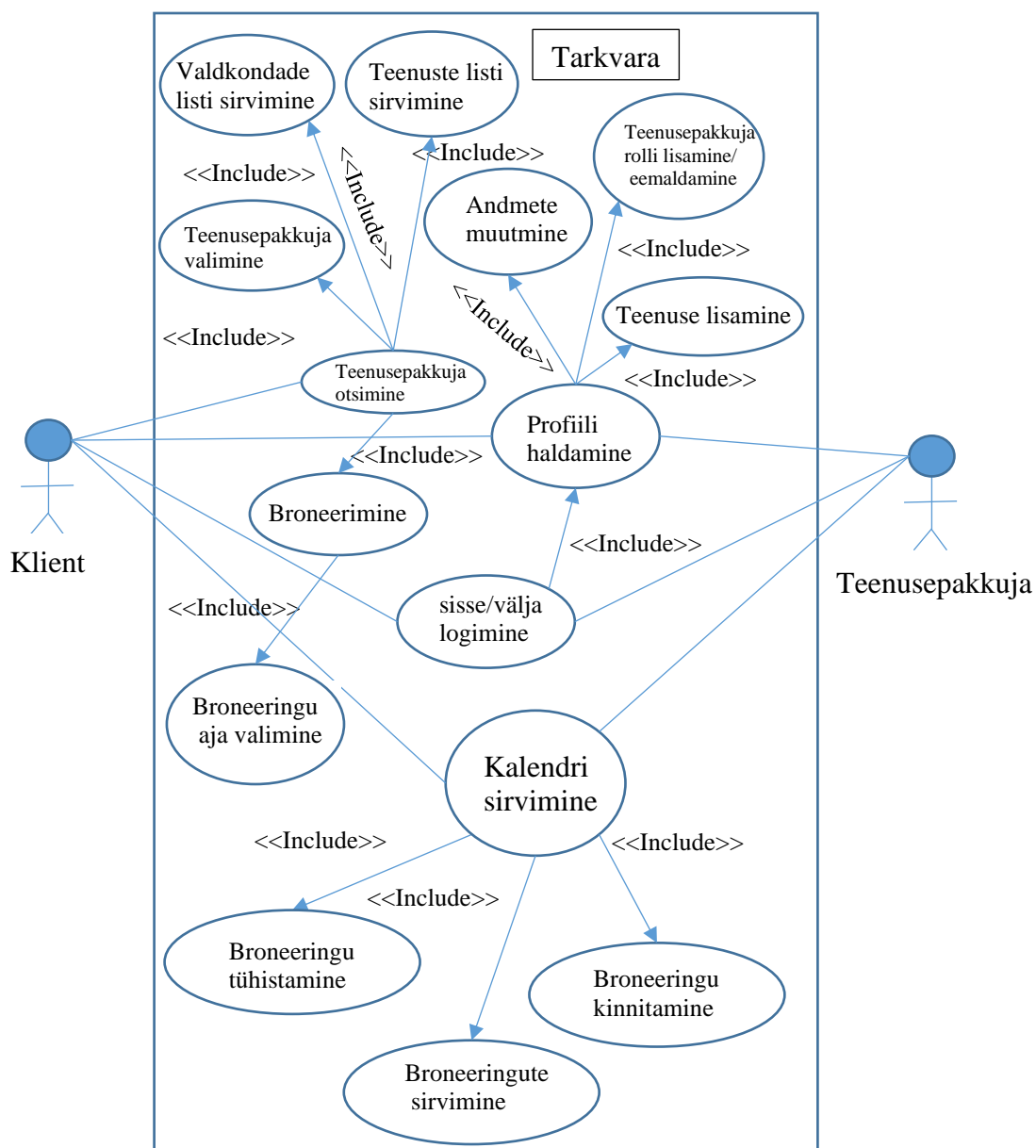
## 6.3 Äritransaktsioonid



Joonis 3. Äritransaktsioonide mudel

Äritransaktsioonide mudelilt (Joonis 3) on näha osapoolte eesmärgid ning seosed, mille kaudu need eesmärgid saavad täidetud süsteemi abil.

## 6.4 Kasutusjuhud



Joonis 4. Kasutusjuhtude mudel

Süsteemil on loodud kahe osapoole huvide rahulamiseks – kliendi ja teenusepakkuja. Kasutusjuhtude mudel (Joonis 4) kirjeldab võimalikke kasutusjuhtusid [10]. Osapooled saavad kokku süsteemis, kus klient soovib leida endale vajaliku teenuse pakkujat ja

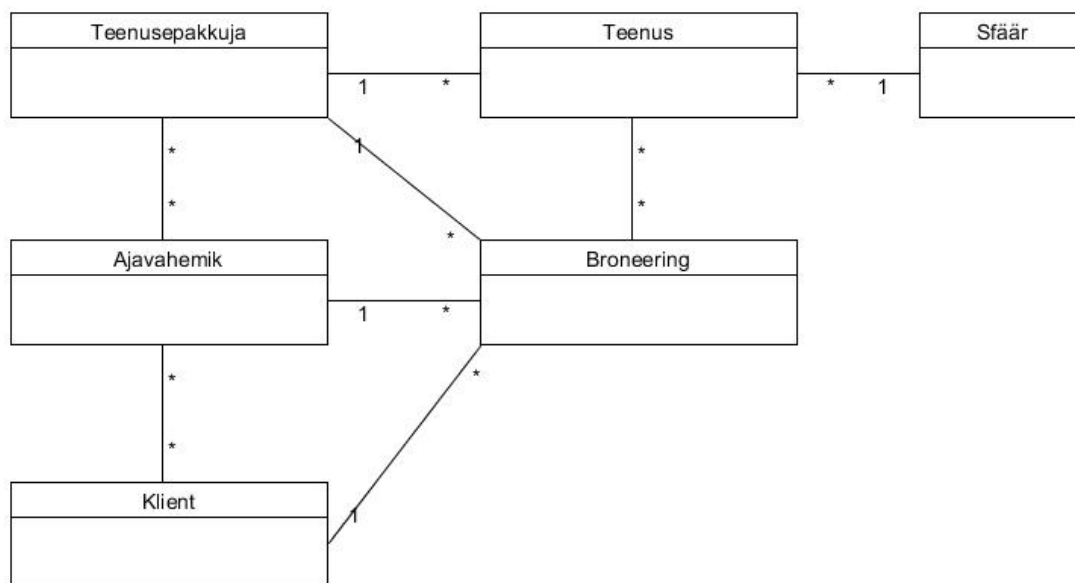
teenusepakkujad pakuvad enda teenuseid klientidele. Põhiline suhtlusvahend osapoolte vahel on broneerimissüsteem, mille abil saavad kliendid broneerida aega teenusepakkuja juurde.

- Inimene avab veebirakenduse, rakendus teeb kindlaks, kas inimene on sisse loginud ja vastavalt sellele pakub sisselogimist või kasutajaks registreerumist. Registreerumisel tekib andmebaasi uus kirje kasutajaandmetega.
- Sisse loginud kasutaja muudab oma profiili avalikke andmeid kasutajaliidese kaudu, tarkvara salvestab muudatused andmebaasi.
- Kasutaja vajutab nuppu, et tavalisest profiilist saaks teenusekasutaja profiil. Andmebaasis tekib kasutajale teenusepakkuja roll.
- Teenusepakkuja rollis kasutaja lisab oma profiilile osutatavaid teenuseid teenuste listist. Andmebaasi tekivad kirjed, mis seovad tema profiili ja pakutavaid teenuseid.
- Teenusepakkuja rollis kasutaja määrab osutatava teenuse hinna, hinnangulise teenuse osutamise aja ja lisab omapoolse kommentaari. Andmebaasi tekivad kirjed vastava teenuse juurde.
- Klient otsib teenusepakkujat teenusele otsingu lehel, selleks valib valdkonna ja teenuse. Tarkvara tagastab nimekirja otsitava teenuse pakkujatest.
- Klient olles välja valinud sobiva teenusepakkuja vajutab broneerimise nuppu, süsteem saadab ta broneerimise lehele, kus ta saab valida teenuse osutamise kuupäeva, lisada veel soovitud teenuseid ja broneeringut kinnitada. Kinnitamisel loob süsteem andmebaasi uue broneeringu kirje koos kõikide broneeringu andmetega.
- Klient avab kalendri ja näeb sealt kõiki oma broneeringuid. Süsteem leiab andmebaasist kõik broneeringud, kus kliendiks on antud kasutaja ja tagastab tulemused.

- Teenusepakkuja avab kalendri ja näeb kõiki temale tehtud broneeringuid. Süsteem leiab andmebaasist kõik broneeringud, kus teenusepakkujaks on antud klient ja tagastab need kasutajale.

Kasutajaliides saadab käskluseid tarkvarale, mis on serveril, see omakorda teeb päringuid andmebaasist ning tagastab andmed kasutajaliidesele, mis kuvab need kasutajale.

## 6.5 Domeenimudel



Joonis 5. Domeenimudel

Antud doomenimudel (Joonis 5) kirjeldab süsteemi sisulist olemust maksimaalselt abstraktsel tasemel, andmaks ülevaadet kõikidest seotud valdkondadest, mis on vajalikud ehitatava süsteemi toimimiseks.

Teenusepakkujal on profiil.

Teenusepakkuja lisab pakutavaid teenuseid.

Kliendil on profiil.

Klient otsib soovitud teenuse pakkujat.

Klient broneerib teenusepakkuja juurde aja.

Teenusepakkuja kinnitab või lükkab broneeringu tagasi.

Klient sirvib oma broneeringuid kalendrist.

Teenusepakkuja näeb temale broneeritud aegu ja teenuseid kalendrist.

## **7 Prototüüpimine**

Lean Stratup metoodika kohaselt tegin valmis MVP, et seda saaks kohe testida ja oleks näha, kas idee toimib. Lahendust tegin minimaalselt ja robustselt, kuid töökindlalt ja efektiivselt.

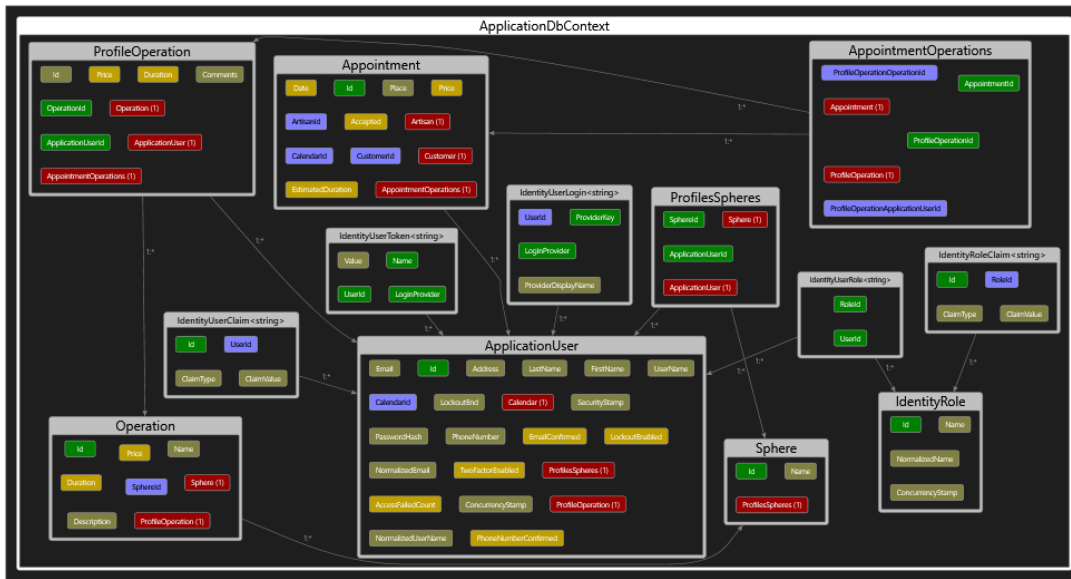
### **7.1 Lahenduse disain**

Lahendus on tehtud .NET [11] platvormil kasutades ASP.NET Core Framework'i, Entity Framework Core'i [12] andmetabelite kirjeldamiseks ja andmebaasiga suhtlemiseks ning Blazor [13] raamistikku kasutajaliidese tegemiseks.

Süsteem on tehtud toetudes kihilise arhitektuuri printsiipidele ja jagatud kolmeks kihiks: UI, DAL ja Core. UI kihis on kasutajaliides, DAL kihis on andmebaasi objektide mudelid ja rakenduse andmebaasi mudel. Core sisaldab põhiliselt kasutajaliidese ja andmebaasi omavahelise suhtluse loogikat, ehk andmeedastus funktsioone, meetodeid kirjete lisamiseks, kustutamiseks, muutmiseks ja andmete pärimiseks, sisuliselt vastutab CRUD operatsioonide toimimise eest.

Programmeerimisel oli kasutatud Microsoft Visual Studio IDE'd, C# programmeerimiskeelt, HTML'i [14] ja CSS'i [15].

## 7.2 Andmebaasi mudel



Joonis 6. Andmebaasi mudel

(Joonis 6) pealt on näha lahenduse andmebaasi diagramm. Diagrammil on välja toodud kõik andmebaasi objektid, koos nende omavaheliste seostega. Andmebaas on relatsiooniline [16] ja toetab süsteemi objekt-orienteeritud disaini [17].

Andmebaasis on kõikidel objektidel oma unikaalne identifikaator Id

Andmebaasi tabel nimega ApplicationUser sisaldab kasutaja profiili andmeid, mis on vajalikud rakendusse sisse logimiseks, isiklike andmeid nagu nimi, aadress, telefoninumber, seoseid teiste tabelitega nagu kasutaja rollide tabel, ja seoseid teenuste tabeliga, kus säilitatakse seoseid teenustega, juhul kui tegemist on teenusepakkuja profiiliga.

Tabelis IdentityRole on kasutajate rollid näiteks administraator, kasutaja või teenusepakkuja.

Tabelis IdentityUserRole on mitu-mitmele seosed kasutajate ja nende rollide vahel.

ProfileOperation tabelis on mitu-mitmele seosed teenusepakkuja profiili vahel ja pakutavate teenuste vahel, sealos on seostele ka lisatud hind, hinnanguline teenuse kestvusaeg (minutites) ja teenusepakkuja poolne täpsustav kommentaar teenusele.



Seosed on samuti ka eraldi objektid oma unikaalse Id-ga, et nendest saaks moodustada listi teenusepakkuja teenustest.

Sphere tabelis on kõik võimalikud valdkonnad.

Operation tabelis on kõik tüüpteenused, millest teenusepakkujad saavad endale lisada unikaalseid pakutavaid teenuseid oma hinna, kestvuse ja kommentaaridega, kõik teenused kuuluvad ka mingisugusesse valdkonda, see on määratud mitu-ühele seosega valdkondadega, seda on vaja otsingu teostamiseks.

Appointment tabel on broneeringute hoidmiseks. Nagu igal objektil on ka broneeringul oma unikaalne Id. Samuti igal broneeringul on ka teenusepakkuja ja kliendi (broneerija) Id-d. Broneeringul broneeringu aeg, koht, hinnanguline kestvusaeg (minutites), hind ja seos tabeliga AppointmentOperations, kus on mitu-mitmele seosed kõikide teenustega, mis on seotud antud broneeringuga.

## **7.3 Äriloogika tarkvaraline realiseerimine**

### **7.3.1 Kasutajate rollid ja õigused**

Kasutajad jagunevad enamjaolt kaheks - teenusepakkujateks ja klientideks. On olemas ka administraatorid [18]. Kasutajate profiilidel saab olla kolme tüüpi rolli kasutaja, teenusepakkuja ja administraator. Kasutaja rollis profiilil on kõige vähem õiguseid. Ta saab muuta oma profiili andmeid (nime, aadressi, telefoninumbrit, profiilipilti jne), samuti saab ta otsingust otsida teenusepakkujaid ning broneerida nende juurde aegasid ning hiljem oma broneeringuid läbi kalendri hallata. Kasutajal on ka võimalus ise endal aktiveerida teenusepakkuja roll. Teenusepakkuja roll võimaldab lisada oma profiilile osutatavaid teenuseid, mis teeb võimalikuks teenusepakkuja leidmist ning tema juurde aja broneerimist klientide jaoks. Samuti saab teenusepakkuja hallata talle tehtud broneeringuid. Kinnitada neid või tagasi lükata. Administraatori rolliga kasutaja saab lisada süsteemi profiile, hallata olemasolevate profiilide andmeid, lisada ja kustutada kirjeid sfääride ja teenuste nimekirjast.

### **7.3.2 Valdkonnad ja teenused**

Et hoida süsteemi organiseerituna, ja et erinevate valdkondade teenused ei oleks segamini listides, on nad grupeeritud tegevusvaldkondade alusel nagu näiteks teenused,

mis puudutavad juuksurite valdkonda on leitavad juuksurite sfäärist, tatoveerijate teenused tatoveeringute sfäärist jne. Igal sfääril on nimekiri selle sfääri raames pakutavatest tüüpteenustest. Nende tüüpteenuste alusel lisavad teenusepakkujad oma profiilidele pakutavaid teenuseid, millele märgivad kõik individuaalselt hinna, mille eest nad antud teenust osutavad, umbkaudse aja, millega nad seda teenust osutavad ja soovi korral ka omapoolsed kommentaarid osutatava teenuse kohta. Nende teenuste põhjal, mille teenusepakkujad oma profiilidele lisavad on nad hiljem klientide jaoks leitavad otsingu kaudu.

### **7.3.3 Otsing**

Teenusepakkuja otsing on tehtud maksimaalselt mugavaks, lihtsaks ja minimaalselt klikke nõudvaks. Kõik mida on näha otsingu lehel on kaks avanevat nimekirja, millest üks on list valdkondadest ning teine teenustest, mida antud valdkonna raames pakutakse. Kui muutub valitud valdkond, siis uueneb osutavate teenuste list ning kui on valitud ka soovitud teenus, siis tekib nimekiri teenusepakkujatest, kes soovitud teenust pakuvad. Profiilide tulemuste tabelist on kohe ka näha teenusepakkuja soovitud hind, umbkaudne teenuseosutamise aeg ning kõrval on nupp, mis viib broneerimise lehele.

### **7.3.4 Broneerimine**

Broneerimisel peab kasutaja samuti tegema minimaalse arvu klikke, ta peab ainult valima aja, mille ta soovib teenusepakkuja juurde broneerida, kõik ülejäänud väljad on juba tarkvara poolt täidetud märkamatu kasutaja jaoks. Broneerija saab lisada broneeringusse veel teenuseid, mida teenusepakkuja pakub. Teenusepakkuja teenuseid saab valida teenuste listist broneeringu juurest. Broneeringust saab ka valitud teenuseid ühe vajutusega eemaldada. Broneeringu juures on ka kohe näha lõplik hind ja oletatav teenuse pakkumise aeg. Hinna ja oletatava aja väärtused muutuvad broneeringusse teenuste lisamisel ja nende eemaldamisel sealt.

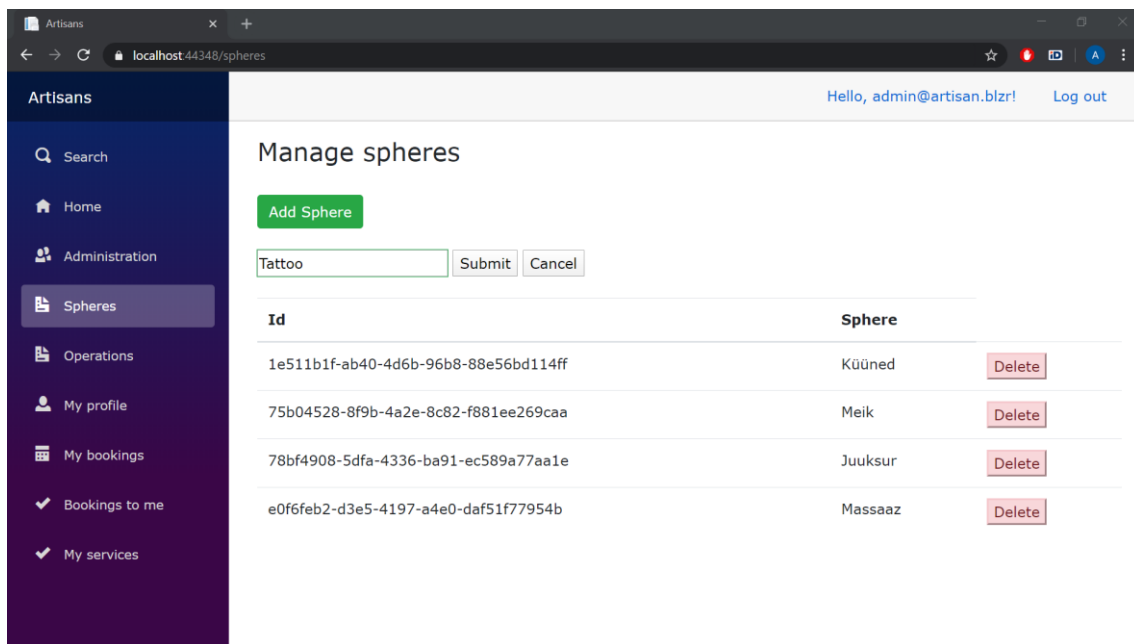
### **7.3.5 Kalender**

Kalender on selles süsteemis olemas vaid kasutajaliideses. Kui kasutaja avab kalendri, siis näeb ta listi oma broneeringutest, kus on kliendi Id on tema profiili Id. Sama moodi ka teenusepakkujate kalendriga, kus teenusepakkuja näeb listi broneeringutest, kus teenusepakkuja Id on tema profiili Id.

## 7.4 Kasutajaliides

Kasutajaliides on tehtud Blazor raamistiku abil ning koosneb komponentidest. Komponente on võimalik taaskasutada, mis läheb hästi kokku DRY printsiibiga elimineerides vajaduse oma koodi korrata. Hea näide sellest on valdkonna ja teenuse valimise rippmenüüde komponent, mis on integreeritud otsingu lehele ja teenusepakkuja teenuste lisamise lehele olles *child component* vastavalt otsingu või teenuste haldamise *parent component*'idele.

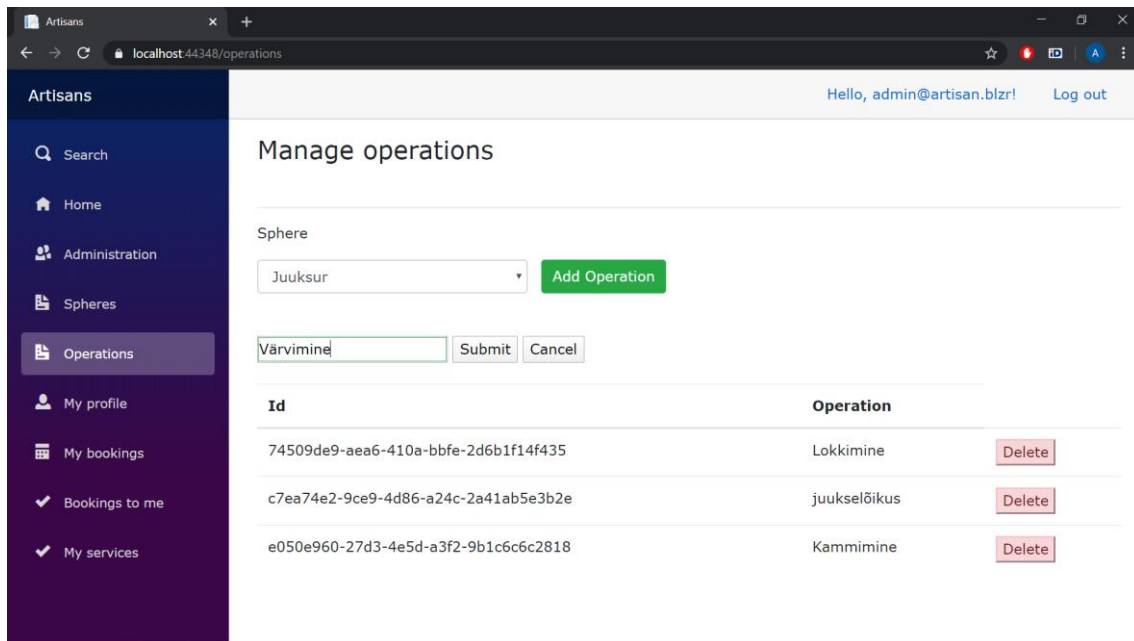
### 7.4.1 Valdkondade lisamine



Joonis 7. Valdkondade lisamine

(Joonis 7) pealt on näha administraatori vaates valdkondade lisamise liides.

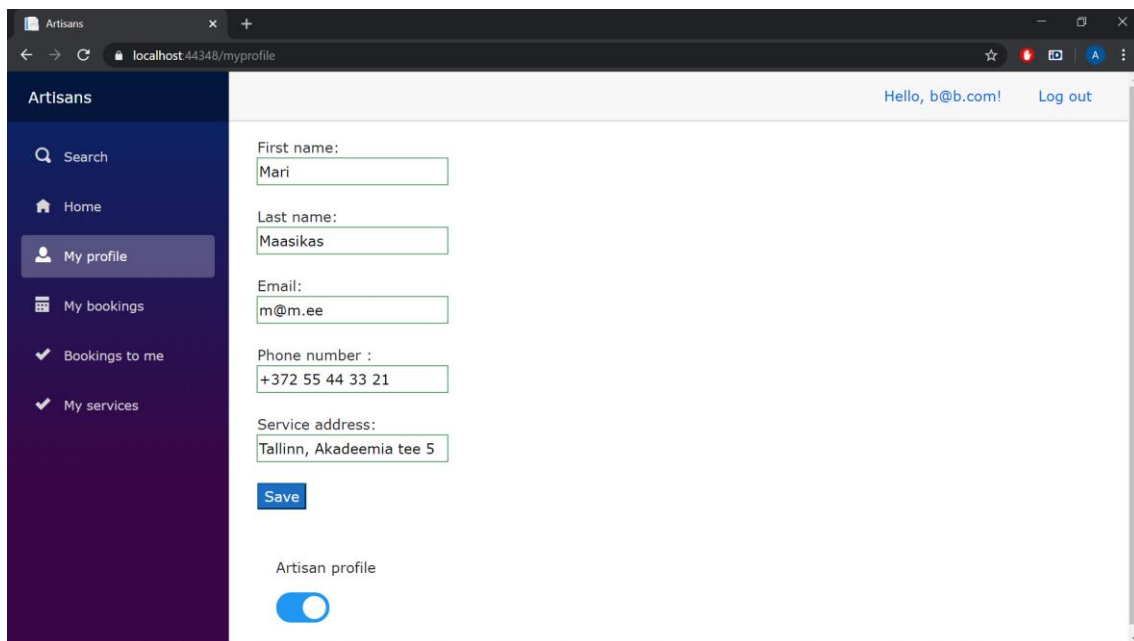
## 7.4.2 Teenuste lisamine



Joonis 8. Teenuse lisamine

(Joonis 8) peal on näha valdkondade alla teenuste lisamise liides administraatori vaates.

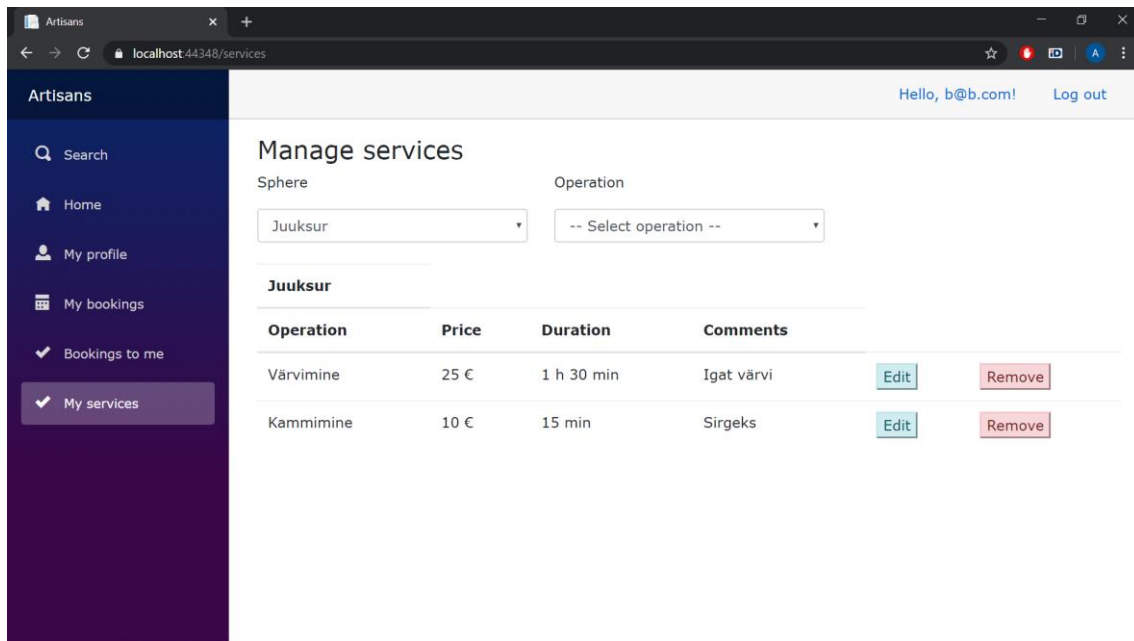
## 7.4.3 Profiili andmete haldamine



Joonis 9. Profiili haldamine

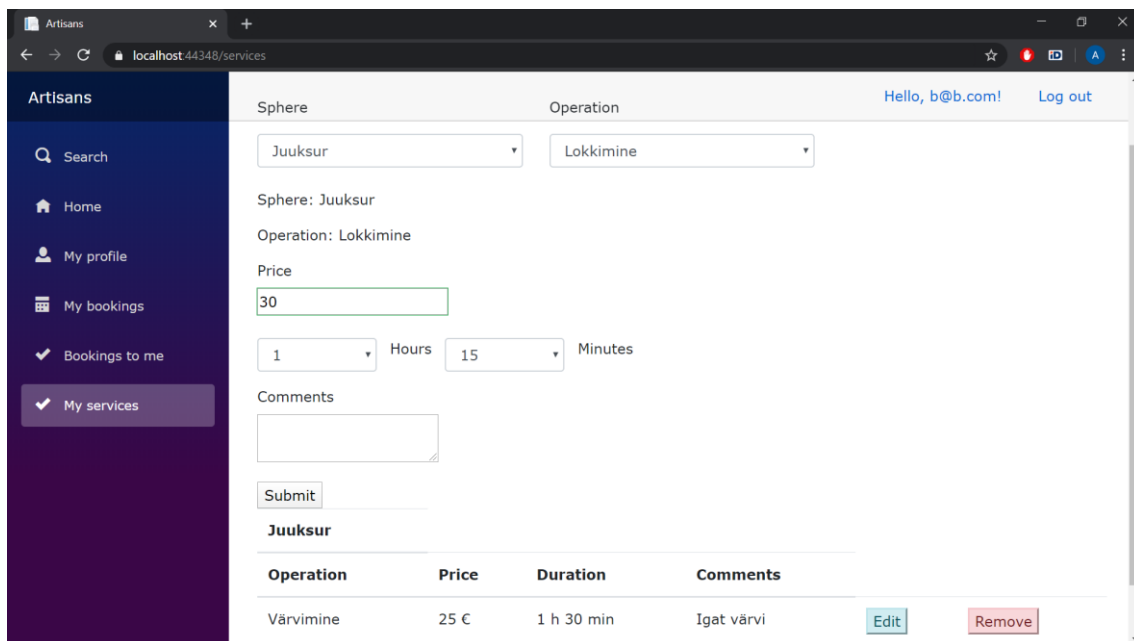
(Joonis 9) peal on näha profiili andmete haldamine ja teenusepakkuja profiili aktiveerimine.

## 7.4.4 Teenuste lisamine profiilile



Joonis 10. Profiili teenuste list

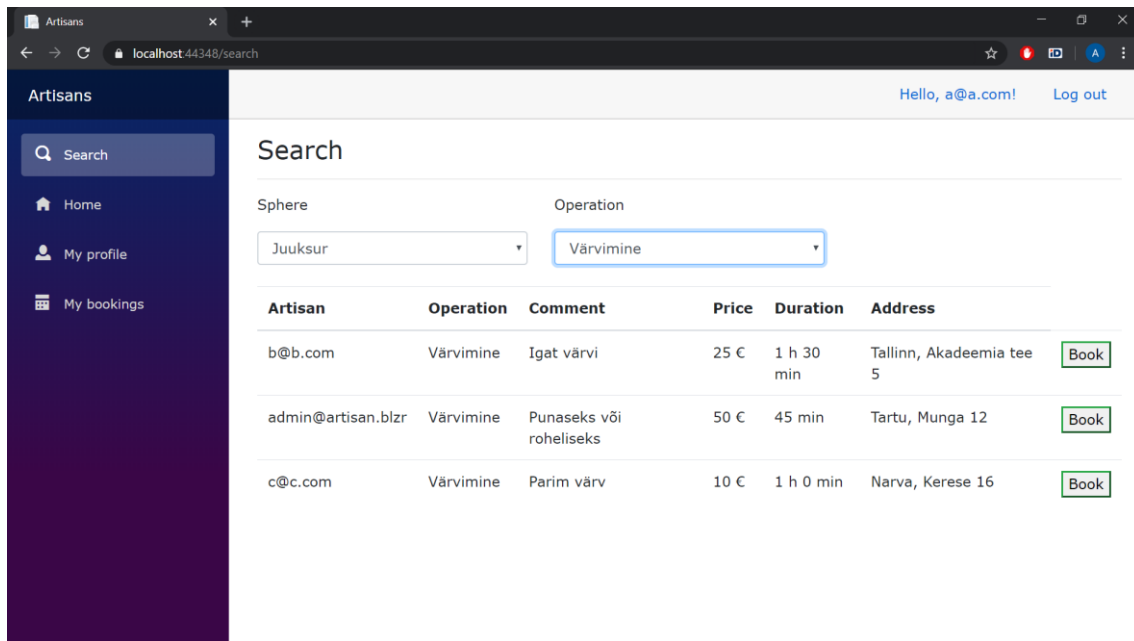
(Joonis 10) pealt on näha teenusepakkuja rolli vaates listi profiiliga seotud teenustest, mis on teenusepakkuja enda lisatud. Siin saab lisada uusi teenuseid või muuta olemasolevaid.



Joonis 11. Uue teenuse lisamine

(Joonis 11) on näha profiilile uue teenuse lisamise protsess. Olemasoleva teenuse muutmine näeb välja analoogselt.

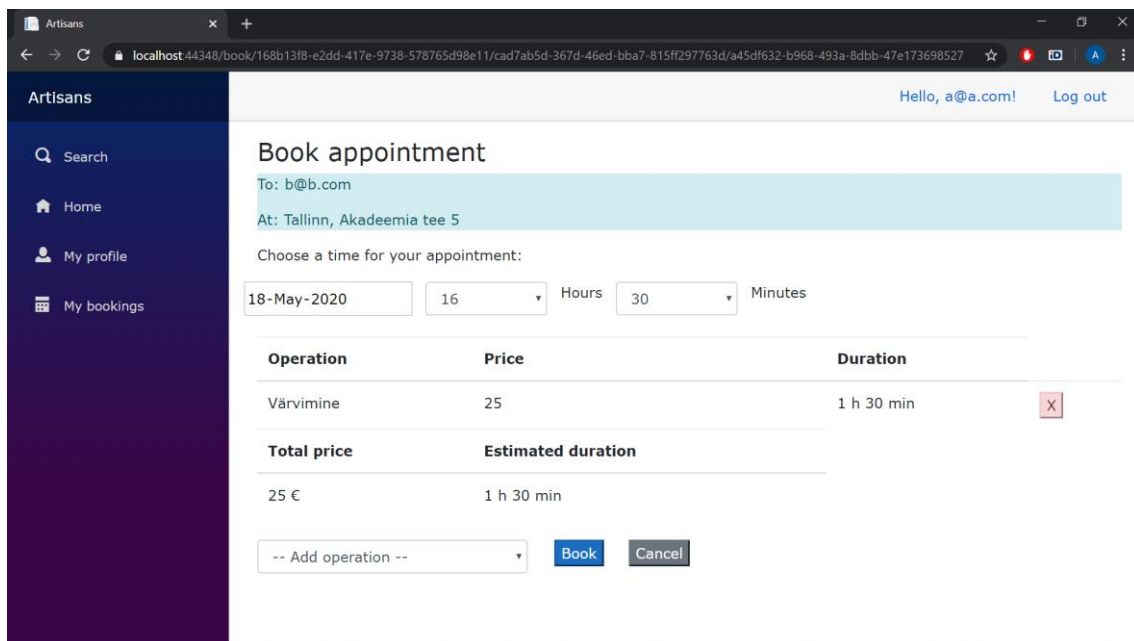
## 7.4.5 Otsing



Joonis 12. Otsing

(Joonis 12) peal on näha otsingu leht tavakasutaja vaates, koos otsingutulemustega. *Book* nupule vajutades avaneb broneerimisleht.

## 7.4.6 Broneerimine

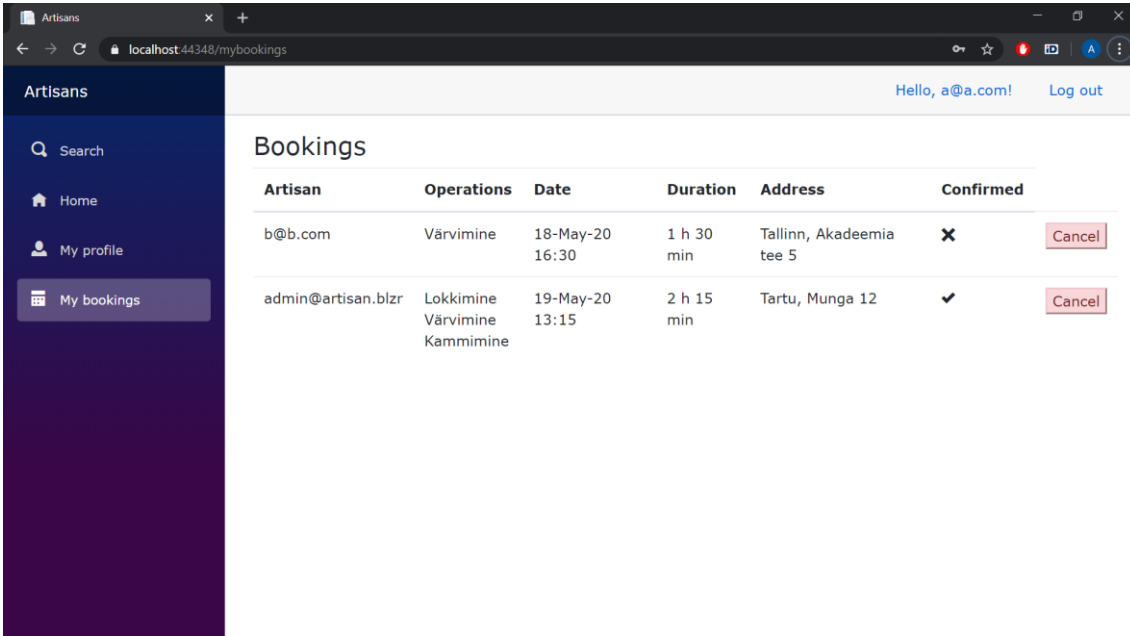


Joonis 13. Broneerimine

(Joonis 13) peal on näha broneerimise menüü. Siit saab klient valida aja, lisada veel juurde teenuseid, mida teenusepakkuja pakub. Kui valitud aeg ei sobi näiteks on minevikus või kattub teenusepakkuja muude broneeringutega, siis ei lase süsteem

kliendil broneeringut teha ja näitab missugused ajad on antud päeval kinni arvestades juba olemasolevate broneeringute algusaegasid ja kestvusi ning antud broneeringu algusaega ja kestvust.

#### 7.4.7 Kliendi broneeringud

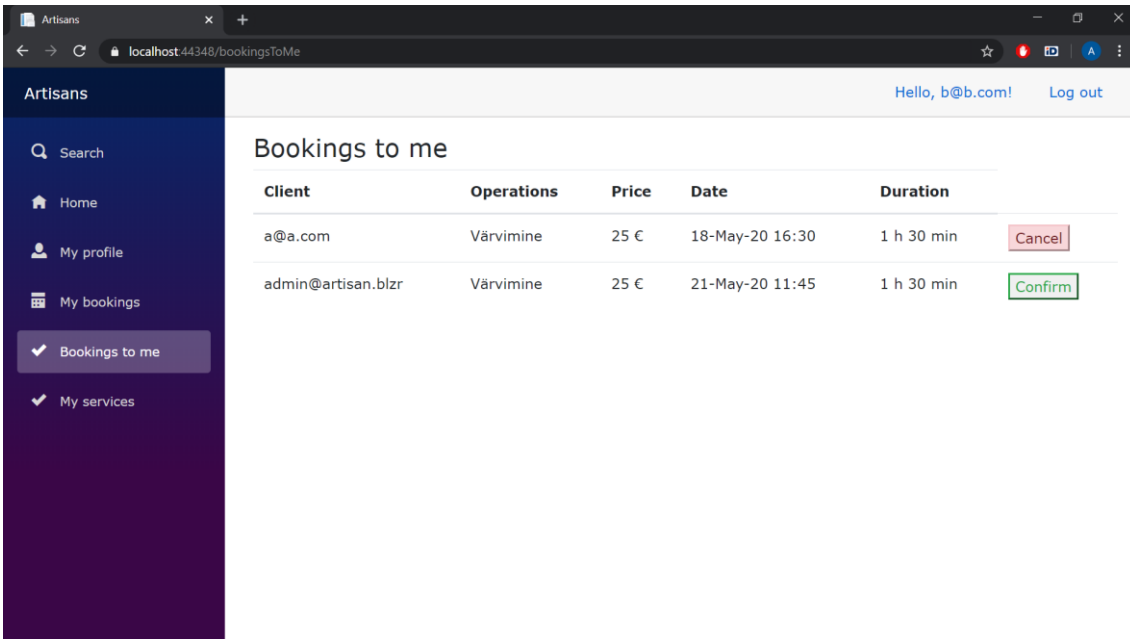


Artisan	Operations	Date	Duration	Address	Confirmed	
b@b.com	Värvimine	18-May-20 16:30	1 h 30 min	Tallinn, Akadeemia tee 5	✘	Cancel
admin@artisan.blzr	Lokkimine Värvimine Kammimine	19-May-20 13:15	2 h 15 min	Tartu, Munga 12	✔	Cancel

Joonis 14. Kliendi broneeringud

(Joonis 14) peal on näha kliendi vaates list kliendi broneeringutest. Näha on teenusepakkujat, teenuste nimekirja, kuupäeva, kestvust, aadressi ja staatust, kas teenusepakkuja on broneeringu kinnitanud.

## 7.4.8 Broneeringud teenusepakkujale



Client	Operations	Price	Date	Duration	
a@a.com	Värvimine	25 €	18-May-20 16:30	1 h 30 min	Cancel
admin@artisan.blzr	Värvimine	25 €	21-May-20 11:45	1 h 30 min	Confirm

Joonis 15. Broneeringud teenusepakkujale

(Joonis 15) pealt on näha teenusepakkuja vaates broneeringud talle. Teenusepakkuja saab kinnitada ja tühistada broneerimisi.

## 7.5 Suhtlus andmebaasiga

Kasutajaliides suhtleb andmebaasiga Core kihi kaudu, kus asuvad *Service*'id, mis teevad päringuid andmebaasi ja tagastavad saadud väärtused või kirjutavad andmebaasi neile edastatud väärtuseid. Andmebaasiga suhtlevad meetodid toimivad kõik asünkroonselt [19] selleks, et vältida *thread*'ide blokeerumist ja et kõik operatsioonid toimiksid lõpuks kiiremini.

## 8 Analüüs ja järeldused

### 8.1 Tulemused

Turu uurimise ja lahenduse prototüüpimise tagajärjel valmis MVP broneerimissüsteemist teenusepakkujate ja klientide jaoks. Süsteemis saavad nii teenusepakkuja kui ka klient teha endale profiilid. Teenusepakkuja saab märkida



missuguseid teenuseid ta osutab, lisada nendele teenustele hinna, umbkaudse teenuse osutamise aja ja oma kommentaari teenuse kohta. Klient saab otsingu abil leida teenusepakkuja, kes osutaks soovitud teenust ning broneerida tema juurde aja. Samuti saavad kasutajad hallata oma profiilide avalikke andmeid, ja broneeringuid.

Prototüüp on just selliste võimalustega, kuna need on püstitatud probleemi lahendamiseks põhilised ja minimaalselt vajalikud.

## **8.2 Alternatiivid**

On ka muid võimalikke lahendusi probleemile, näiteks pakkuda mõnele olemasolevale broneerimissüsteemile võimalust tulla Eesti turule ja kohandada nende lahendus vastavalt rakenduse klientide vajadusele.

Samuti võimalikuks variandiks oleks lahenduse integreerimine olemasolevasse süsteemi paljude kasutajatega nagu näiteks integratsioon Facebook'i äpp'ina, selle asemel, et teha eraldiseisvat veebirakendust.

Ise arendamise asemel saaks arendusteenust sisse osta nende käest, kellel on juba kogemus olemas, see tõstaks lahenduse hinda, kuid tõenäoliselt ka kvaliteeti ja kiirust.

Lahenduse tegemisel oleks võinud keskenduda konkreetsele valdkonnale ja lisada kohe ka valdkonna väga spetsiifiliseid nüansse rakendusse ja miks mitte ka muud funktsionaalsust peale teenuste broneerimise.

## **8.3 Piirangud**

Hetkel ei ole reaalselt kasutajate kogemust antud rakendusega, mille põhjal saaks teha mingisuguseid konkreetseid järeldusi.

## **8.4 Võrreldes teistega**

Võrreldes olemasolevate lahendustega on peamine erinevus see, et olemasolevad lahendused on juba väljas ja inimestele kättesaadavad, aga antud lahendus on veel prototüübi staadiumis. Küll aga olemasolevad süsteemid ei lahenda otseselt individuaalse teenusepakkuja ja kliendi probleemi, kuna suurem enamus olemasolevaid

lahendusi on, kas mõne ettevõtte enda teenuste broneerimiseks või suunatud salongidele, mitte eraisikutele. Samuti süsteemid, mis on töö alguses kirjeldatud on mahukamad ja eesmärgini jõudmiseks nõuavad märksa rohkem klikke. Kõige lähemal klikkide arvu poole pealt on Inglismaalt pärit lahendus Ruby, kuid selles süsteemis on ka päris palju nuppe ja linke, mis ei ole broneerimise probleemi lahendamiseks kuidagi seotud.

## **8.5 Äriline põhjendus**

Kuna Eestis on suur osakaal oskustöölisi, siis on sellisel süsteemil suhteliselt kindel turg riigi siseturul. Suurimad konkurendid hetkel on sotsiaalvõrgustikud, mis ei ole otseselt mõeldud teenusepakkujate otsimiseks ja aegade broneerimiseks. Pakutav süsteem teeks paljude teenusepakkujate ja klientide elud lihtsamaks.

## **8.6 Edasised tööd**

Antud MPV on vaid minimaalsete võimalustega töötav lahendus, mis on vaid alguspunktiks reaalsele tootele ja mis tõestab päris toote vajalikkust ja võimalikkust. Järgmine samm oleks veel mõningate pisiasjade lisamine ja esimestele reaalsele kasutajatele kätte andmine iluteenuste sfääris, põhjalik kasutajaliidese esteetilise disaini läbitöötamine, et see ei oleks kasutajate jaoks eemaletõukav. Seejärel edasine arendus lähtuvalt kogemusest, mis tekib peale rakenduse kasutajate kätte andmist. Hiljem järjestikune sfäärade lisamine süsteemi, et ka näiteks torumeeste visiite saaks broneerida läbi rakenduse. Kuna hetkel ei ole pääsu nutitelefonidest, siis ilmselt peaks lahenduse realiseerima ka mobiilide jaoks äppide kujul tagamaks paremat kasutajakogemust klientidele.

## 9 Kokkuvõte

Kui saavad kokku erinevad inimesed, siis neil on alati midagi huvitavat üksteise jaoks varrukas peidus. Tuli välja, et Eestis pole eraettevõtjatest iluteenusepakkujatel mugavat viisi oma klientide broneeringute haldamiseks ja klientidel omakorda puudus mugav viis aja broneerimiseks teenusepakkuja juurde. Eesmärk oli seda probleemi lahendada lihtsa broneerimissüsteemiga, mis võimaldab saada mugavalt ülevaadet broneeringutest. Alustasin turul olemasolevate lahenduste uurimisest ja analüüsimisest. Probleemi lahendamiseks tegin broneerimissüsteemi veebirakenduse kujul, kus kliendid saavad kerge vaevaga teenusepakkuja juurde aega broneerida. Antud töö jõudis oma eesmärgini ehk sai valmis minimaalne kasutatav produkt broneerimissüsteemi kujul teenusepakkujate ja nende klientide jaoks. Rakendus pole veel küll kasutajatele kättesaadav, kuid selleni pole palju jäänud. Prototüüp on väga minimalistlik ja maksimaalselt suunatud probleemi lahendamisele. Selle pealt on suurepäraselt näha, et tavalise väikse probleemi lahendamise jaoks ei ole tarvis keerulist süsteemi loendamatu arvu nuppude ja seadetega. Tundub, et pole põhjuseid antud prototüüpi mitte edasi arendada.

## Kasutatud kirjandus

- [1 [Võrgumaterjal]. Available: <https://www.harpersbazaar.com/uk/beauty/make-up-nails/news/g23362/best-beauty-apps/>.]
- [2 [Võrgumaterjal]. Available: <https://apps.apple.com/gb/app/id946475548>.]
- [3 [Võrgumaterjal]. Available: [https://urban.co/en/gb/?siteid=54264&source=webgains&wgexpiry=1593618250&wgu=272225\\_54264\\_15858422509987\\_9578057462](https://urban.co/en/gb/?siteid=54264&source=webgains&wgexpiry=1593618250&wgu=272225_54264_15858422509987_9578057462).]
- [4 [Võrgumaterjal]. Available: <https://www.valueappz.com/>.]
- [5 [Võrgumaterjal]. Available: <https://sobranna.postimees.ee/6715874/uus-app-viib-kliendid-ja-iluprofessionaalid-kiiresti-ja-mugavalt-kokku>.]
- [6 [Võrgumaterjal]. Available: <https://www.booklux.com/>.]
- [7 [Võrgumaterjal]. Available: <https://finde.ee/>.]
- [8 [Võrgumaterjal]. Available: <https://www.ilubooking.ee/>.]
- [9 E. . Ries, The lean startup: how today's entrepreneurs use continuous innovation to create radically successful businesses, toim., kd. , , : Crown Publishing, 2011, pp. [103](http://books.google.com/books?id=tvfyz-4JILwC&pg=PA103)].
- [1 G. Fournier, Essential Software Testing: A Use-Case Approach, CRC Press, 2008. 0]
- [1 „.NET Framework 4.5,“ , . [Võrgumaterjal]. Available: <https://www.microsoft.com/en-us/download/details.aspx?id=30653>. [Kasutatud 15 5 2020].]
- [1 J. . Lerman, Programming Entity Framework, toim., kd. , , : O'Reilly Media, , p. 912. 2]
- [1 M. . James, „Blazor .NET In The Browser,“ , . [Võrgumaterjal]. Available: <https://www.i-programmer.info/news/89-net/11539-blazor-net-in-the-browser.html>. [Kasutatud 15 5 2020].]
- [1 „HTML5 – Hypertext Markup Language – 5.0,“ , . [Võrgumaterjal]. Available: <https://www.w3.org/2014/10/html5-rec.html.en>. [Kasutatud 15 5 2020].]
- [1 „Cascading Style Sheets: CSS Browsers,“ , . [Võrgumaterjal]. Available: <http://www.w3.org/Style/CSS/#browsers>. [Kasutatud 15 5 2020].]
- [1 G. . Wiederhold, Database Design, toim., kd. , , : , 1977, p. . 6]
- [1 G. . Booch, Object-Oriented Analysis and Design with Applications, toim., kd. , , : 7] Addison Wesley, , p. .

- [1] K. W. M. Siu, „Users' Creative Responses and Designers' Roles,“ *Design Issues*, kd. 8] 19, nr 2, pp. 64-73, 2003.
- [1] G. . Paperin ja E. L. Eilebrecht, „Async wrapper handling execution of asynchronous 9] operations for synchronous and asynchronous routines,“ , 2011. [Võrgumaterjal]. Available: <http://freepatentsonline.com/8756604.html>. [Kasutatud 15 5 2020].
- [2] L. S. Sterling, *The Art of Agent-Oriented Modeling*, London: The MIT Press, 2009. 0]
- [2] „Visual C# (MSDN),“ , . [Võrgumaterjal]. Available: 1] <https://msdn.microsoft.com/en-us/library/kx37x362.aspx>. [Kasutatud 15 5 2020].
- [2] M. . Fowler, *Patterns of Enterprise Application Architecture*, toim., kd. , , : Addison-2] Wesley, 2002, p. .

## Lisa 1 – Koodi meetrikad

Hierarchy	Maintainability Index	Cyclomatic Compl.	Depth of Inheritance	Class Coupling	Lines of Source code	Lines of Executable co...
Artisans (Debug)	72	478	4	340	9,602	1,652
Core (Debug)	86	106	1	87	558	187
DAL (Debug)	84	112	6	159	1,519	420

## Lisa 2 – Link koodile

<https://github.com/ArturToikka/Artisans>