

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Karl-Erik Kõivupuu 185234IABB

Raul Vilippus 190511IABB

Nikita Kudryashov 185253IABB

Aleksandr Levitsev 185264IABB

INTERNETI RENDIPORTAALI EDASIARNEDAMINE

Bakalaureusetöö

Juhendajad: Karl-Erik Karu MSc

Gunnar Piho PhD

Tallinn 2021

Autorideklaratsioon

Kinnitame, et oleme 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.

Autorid: Karl-Erik Kõivupuu, Raul Vilippus, Nikita Kudryashov, Aleksandr Levitsev

18.05.2021

Annotatsioon

Antud töö käsitleb tudengite initsiatiivil koostatud projekti, mille eesmärgiks on inimestele pakkuda platvormi, mis võimaldaks kasutajal/kliendil kas omale/endale rentida või välja rentida soovitud esemeid. Inimesed eelistavad tihti peale esemete rentimist, esemete ostmise üle. Kindlasti mitte iga eseme osas, kuid esemed mida kasutatakse vaid paar korda aastas või on hädaolukorras vaja. Lisaks on inimesed järjest rohkem informeeritud tänapäeva ühiskonna mõjudest keskkonnale ning tahetakse anda enda panust, et situatsiooni parandada. Esemete rentimine pakub inimestele just seda võimalust, kuna väheneb ületarbimine ning ebavajalike esemete ostmine, mis otseselt aitab säästa keskkonda. Projekti aluseks võeti juba valmisolev kuid minimaalse funktsionaalsusega mall, mida arendati edasi vastavalt vajadustele ning meeskonna liikmete visioonile. Projekti raames järgiti erinevaid arendusmeetodikaid, mis aitasid kaasa arendustööle. Järgida nii SCRUM raamistikku kuid rakendati ka suures hulgas paarisprogrammeerimist.

Kuna aluseks olev mall oli väga algeline – kohati puudulik -, siis projekti vältel lisandus palju uut funktsionaalsust. Antud töö raames muudeti veebiplatvorm tunduvalt kasutajasõbralikumaks, lisati arenenud kalendrisüsteem ning broneerimisvõimalus pakutavate toodete puhul. Teise hulgas lisati ka erinevad? maksevõimalused ning kaheastmeline autentimine kasutajaturvalisuse jaoks. Sellest tulenevalt valmis rendi platvorm Rentif, mis võimaldab turvaliselt ja aega kokkuhoidvalt rentida inimeselt-inimesele esemeid.

Abstract

This work is related to a project initiated by students, the aim of which is to provide people a platform that would allow them to either rent or lease the desired items. People often prefer to rent things rather than buy. Of course, not for every item, but for items that are used only a few times a year or are needed in an emergency. People are increasingly aware of the impact of modern society on the environment and want to help improve the situation. Renting items gives people this opportunity, as there is no overconsumption and purchase of unnecessary items, which directly contributes to the preservation of the environment. The project was based on a template with minimal functionality, which was further developed according to the needs and vision of the team members. Within the project, various development methodologies were followed, which contributed to the development work. We have followed the SCRUM framework and also a lot of pair programming was done.

As the initial template was very incomplete - a lot of new functionality was added during the project. As part of this work, the web platform was made much more user-friendly, an advanced calendar system and a booking option for the offered products were added. Among other things, various payment options and two-step authentication for user security. As a result, rental platform Rentif has been completed, which allows you to rent items from person to person in a safe and time-saving manner.

Lühendite ja mõistete sõnastik

Back-end	Loogika ja andmete salvestamine
Front-end	Kuvand, mida näeb kasutaja
RAD	Rapid application development
PHP	Hypertext Preprocessor
HTTP	Hypertext Transfer Protocol
HTML	HyperText Markup Language.
XAMPP	X (any of four different operating systems), Apache, MySQL, PHP and Perl
MVC	Model-View-Controller
IT	Infotehnoloogia
Perl	Kõrgel tasemel tõlgendatud dünaamiline programmeerimiskeel.
Twig	PHP programmeerimiskeelega kirjutatud avatud lähtekoodiga kompileeriv mallimootor.
Blade	Lihtne, kuid samas võimas mallimootor, mis on varustatud Laraveliga
CSS	Cascading Style Sheets
XML	eXtensible Markup Language
DDoS	Distributed Denial-of-Service
Cloudflare	Cloudflare on Ameerika ettevõtte, mis pakub CDN-teenuseid, DDoS-kaitset, turvalist ressursile juurdepääsu ja DNS-servereid.
KKK	Korduma Kippuvad Küsimused
API	Application Programming Interface
MIT	Massachusetts Institute of Technology
Bootstrap	Tööriistakomplekt veebilehtede ja veebirakenduste loomiseks
Kissflow	Pilvepõhine töövoo automatiseerimise tarkvara.

FuelPHP	PHP-s kirjutatud avatud lähtekoodiga veebirakenduste raamistik
HMVC	Hierarchical model–view–controller.
Pull request	Ülevaataamistaotlus
Deploy	Juurutada
Landing page	Maandumislehekülg
Country code	Telefoninumbrate eesliited
Two-factor authentication	Kaheastemeline autentifitseerimine
Tags	Sildid
Grid	Ruudustik
List	Nimekiri
JS	JavaScript
Vue.js	Vaba JavaScripti raamistik, millega luuakse kasutajaliideseid
Cache	Arvutites osa, mis hoiustab andmeid nende kiireks uuesti kasutamiseks
Composer	PHP programmeerimiskeele jaoks rakenduse tasemel pakihaldur, mis pakub tööriistu sõltuvuste haldamiseks PHP rakenduses
Waterfall	Üks enimteatud ja vanimaid tarkvaraarenduse meetodeid, mis on oma nimetuse saanud järjestikuse veekoskede analoogiast – vesi langeb ühelt astmelt järgmisse ega pöördu tagasi.
AJAX	Asynchronous Javascript and XML

Sisukord

1 Sissejuhatus.....	11
2 Metoodika	13
2.1 Objekti detailne ülevaade	13
2.2 Tööriistade kirjeldus.....	13
2.3 Tööprotsessi kirjeldus	15
3 Töö tulemused.....	17
3.1 Rakendus	17
3.1.1 Avaleht.....	17
3.1.2 Registreerimine ja sisselogimine	19
3.1.3 Kuulutuse lisamine	23
3.1.4 Kuulutuste vaatamine ja sorteerimine	30
3.1.5 Tavakasutaja paneel.....	31
3.1.6 Lehekülje haldaja paneel	37
3.1.7 Kuulutuse broneerimine ja maksmine	40
3.1.8 Kontakteerumine	44
3.1.9 Lisainformatsioonid.....	45
3.2 Arhitektuur	46
3.3 Disain	48
3.4 Kood.....	48
3.4.3 Andmebaas	49
3.4.4 XAMPP	49
3.5 Testid.....	49
4 Analüüs ja järeldused.....	50
4.1 Tehnilise teostuse analüüs.....	50
4.1.1 Nõuded.....	50
4.1.2 Arhitektuur.....	50
4.1.3 Disain.....	51
4.1.4 Kood	52

4.2 Projekti teostamise põhjendus	53
4.3 Logid	55
4.4 Hinnang projekti teostamise protsessi kohta	58
4.4.1 Projekti juhtimise ja teostamise protsess	58
4.4.2 Hinnang projektile	58
4.4.3 Hinnang protsessile.....	59
4.5 Meeskondlik hinnang	59
5 Kokkuvõte.....	60
Kasutatud kirjandus	61
Lisa 1 - Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	62
Lisa 2 – Enda panuse kirjeldus ja analüüs – Raul Vilippus	63
Lisa 3 – Enda panuse kirjeldus ja analüüs – Karl-Erik Kõivupuu.....	66
Lisa 4 – Enda panuse kirjeldus ja analüüs – Nikita Kudryashov.....	67
Lisa 5 – Enda panuse kirjeldus ja analüüs – Aleksander Levitsev	68

Jooniste loetelu

Joonis 1. Rendiportaali landingpage	18
Joonis 2. Rendiportaali kasutaja registreerimine	20
Joonis 3. Kasutaja telefoni numbri verifitseerimine	21
Joonis 4. Rendiportaali sisselogimine.....	22
Joonis 5. Kasutaja parooli ununemine	23
Joonis 6. Kuulutuse lisamise esimene vaade	24
Joonis 7. Kuulutuse lisamise teine vaade.....	25
Joonis 8. Kuulutuse lisamise kolmas vaade.....	26
Joonis 9. Kuulutuse lisamise neljas vaade	27
Joonis 10. Kuulutuse lisamise viies vaade.....	28
Joonis 11. Kuulutuse lisamise kuues vaade	29
Joonis 12. Kuulutuste sorteerimine.....	30
Joonis 13. Kuulutuste filtreerimine.....	31
Joonis 14. Kasutaja rippmenüü.....	32
Joonis 15. Kasutajapaneel.....	33
Joonis 16. Kasutaja paroolivahetamine.....	34
Joonis 17. Kasutaja poolt tehtud broneeringud.....	34
Joonis 18. Kasutaja poolt salvestatud kuulutused.....	35
Joonis 19. Kasutaja kuulutused.....	35
Joonis 20. Kasutajale sissetulevad broneeringud.....	36
Joonis 21. Detailne ülevaade broneeringust.....	37
Joonis 22. Administraatoripaneel.....	38
Joonis 23. Kategooriate lisamine	39
Joonis 24. Lehekülje disaini vahetamine	40
Joonis 25. Kuulutus.....	41
Joonis 26. Kalender.....	42

Joonis 27. Ajavalimine.....	42
Joonis 28. Maksekorraldus.....	43
Joonis 29. Kontakt	44
Joonis 30. KKK.....	45
Joonis 31. Abi	45
Joonis 32. Kasutustingimused.....	46
Joonis 33. Privaatsuspoliitika.....	46
Joonis 34. Projekti struktuur	47
Joonis 35. Mallprojekti kõik koodiread	55
Joonis 36. Lõpp-projekti kõik koodiread	56
Joonis 37. Meeskonnaliikmete panustatud aeg	57
Joonis 38. Eneseanalüüs, GitHub commit'id	63

1 Sissejuhatus

Ületarbimine on tarbimise vorm, mis ületab ökosüsteemi loomulikku taastumisvõimet ning mille tulemus on ökoloogiline defitsiit. Maailmas ostetakse ja tarbitakse mitmesuguseid tooteid mõtlemata, kas neid ikka vaja on, mis asjadest hiljem edasi. Rahvaarvu tõusuga suureneb tootmine, ühes sellega kasvab materiaalsuse tähtsus ning see omakorda loob inimestele üha suuremaid ja paremaid võimalusi lubada asju, mida tegelikult vaja ei lähe [1]. Antud platvorm on loodud eelkõige inimeste elu lihtsustamiseks ning rahaliseks kokkuhoiuks. Ühiskond seisab silmitsi hetkel tõsiste probleemidega ning üks neist on ületarbimine – nii toidu, maavarade kuid ka esemete. Inimesed ostavad pidevalt suurtes kogustes uusi esemeid, mida kasutatakse tihti eseme eluea jooksul vaid paar korda. Luues inimeselt-inimesele rendi platvormi aitame ühiskonnal säästa nii loodust, kui ka raha, kuna uue eseme ostmise asemel on inimesel võimalik antud ese rentida ning see hiljem tagastada, mis välistab ebavajaliku ostu ning rahuldab mõlemat osapoolt.

RendiPuhul on tegemist uuelaadse rendi platformiga, mis võimaldab kasutajal rentida pea piiramatu valikuga tooteid. Teisisõnu, inimestel on võimalik rentida pea ükskõik mida senikaua kuni antud toode leidub pakutavate toodete seas. Antud idee sündis pärast Eesti rendiportaalides surfamist, kus oli selgelt näha, et olemasolevad rendi platvormid on kas piiratud valikuga või vähe kasutatud ning soovitud eseme leidmine ja selle rentimine oli pea võimatu. Kuna puudus laialdasem inimeselt inimesele rendi platvorm ning selle vajadust oli tunda ka omal nahal, siis otsustati alustada tööd uuelaadse rendi platvormi arendusega.

Projekt valmis kahes osas. Esimeses osas tuli meeskonnal leida sobiv mall, mille alusel oleks võimalik antud projekti kallal tööd alustada. Malli leidmine oli eriti oluline, kuna projekti otsustati kirjutada PHP keeles, millega polnud meeskonnal varasemalt erilist kokkupuudet olnud. Sobiva malli leidmisele järgnes vajamineva funktsionaalsuse kirjapanek ning peale seda hakkas pihta arendustöö. Esimese etappi käigus valmis elementaarne funktsionaalsus, et oleks võimalik antud projekti kallal võimalikult palju eksperimenteerida ning leida kõige sobivamad lahendused. Kuna projekti esmane valmimistähtaeg oli planeeritud suve alguseks, siis oli kõik esimese etapi jooksul

tehtud arendustöö eriti tähtis, et kogu seda tööd oleks võimalik teises etapis põhjalikult testida ning edasi arendada. Projekti teises osas oli põhirõhk peamiselt juba kasutajaliidese arendamis

2 Metoodika

Järgnevates peatükkides saab täpsema ülevaate objektist, tööriistadest ning tööprotsessist. Räägitakse mis tööriistu kasutati ning mis põhjusel. Lisaks milline nägi välja tööprotsess, mida tehti tööprotsessi käigus.

2.1 Objekti detailne ülevaade

Käesoleva projekti idee tuleneb nii isiklikust vajadusest, äriideest, kui ka tahtest anda oma panus looduse säilitamiseks. Hetkel Eesti veebimaastikul ringi vaadates ei torca silma ükski laialdaselt kasutatud rendi platvorm, kust oleks inimestel võimalik rentida erinevaid eluks vajaminevaid esemeid. Sageli on inimesed sunnitud soetama uusi esemeid, mida tihtipeale kasutatakse vaid paar korda aastas või üldse paar korda eseme eluea jooksul. Sellest tulenevalt suureneb ületarbimine ning sellega kaasnevad ühiskondlikud probleemid.

Projekti raames valmis veebiplatvorm, mille abil on inimestel võimalik rentida või välja rentida soovitud esemeid. Kasutaja on kohustatud antud veebilehel ennast registreerima, sisestades oma telefoni numbri, mida kontrollitakse two-factor authentication'i abil, emaili aadressi, nime, riigi ja aadressi. Kasutajal on võimalik ka veebilehele ennast mitte registreerida, kuid see välistab võimaluse veebilehel enda soovitud esemeid kas välja rentida või teiste kasutajate esemeid rentida. Registreerimisprotsess on tehtud võimalikult turvaliseks, et teha kogemus kasutaja jaoks võimalikult turvaliseks ning vähendada pettuste arvu. Veebilehe avamisel on kasutajal võimalik näha eeltoodud kuulutusi või siis enda soovil otsingut kasutades välja otsida sobivad kuulutused. Veebilehele on integreeritud ka kaardisüsteem, kus kasutaja saab näha postituse teinud kasutaja asukohta.

2.2 Tööriistade kirjeldus

Antud projekti arendamiseks kasutati peamiselt PHP programmeerimiskeelt. PHP (PHP: Hypertext Preprocessor) on skriptimiskeel, mida kasutatakse peamiselt serveripoolsetes

lahendustes dünaamiliste veebilehtede loomisel. PHP toetab objektorienteeritud programmeerimist, andmebaasidega suhtlemist ning moodulite kaudu veel paljusid protokolle ja rakendusi. Avatud lähtekood ja võimalus lihtsalt siduda PHP koodi HTML-iga on teinud PHP-st väga populaarse vahendi dünaamiliste veebilehtede loomisel. Levinud on kooslus mida hüütakse rakenduste esitähete järgi LAMP (Linux, Apache, MySQL, PHP) (Vikipeedia, 2006). Kuigi meeskonnal puudus varasemalt kokkupuude PHP-ga otsustasime siiski PHP abil antud projekti läbi viia. Paindlikkus võimaldab PHP-d tõhusalt kombineerida teiste programmeerimiskeeltega nii, et tarkvaratoode saaks kasutada iga konkreetse funktsiooni jaoks kõige tõhusamat tehnoloogiat. Selline paindlikkus hõlbustab oluliselt arendusprotsessi, muutes selle kiiremaks ja odavamaks. Arenduskeskkonnaks kasutati PHPstormi ja XAMPPi. Projekti lokaalseks jooksumiseks kasutati Apache HTTP serverit ning andmebaasina kasutati MariaDB andmebaasi. Mõlemad nii Apache, kui MariaDB kasutati läbi XAMPP arenduskeskkonna.

Web framework'ina (veebiraamistik) kasutati Laraveli. Laraveli komplektis on MVC arhitektuurimuster ning kuna kogemused MVC'ga on varasemalt olemas, tundus laravel kõige loogilisem valik antud projekti jaoks. Laraveli kasuks otsustati just sellepärast, et tegemist on ühe enim kasutatud veebiraamistikuga ning selle täiustatud funktsioonid ja arendustööriistad, hõlbustavad veebirakenduste kiiret arendamist ning kuna antud projekti jaoks polnud aega just kõige rohkem, tuli leida vastav keel ja raamistik, mis sobiks ajaliselt.

Front-End'i kirjutamiseks kasutati Twig'i, Blade'i ja CSS'i keeli. Twig on PHP programmeerimiskeele jaoks loodud mallikeel (template language), mis kompileerib optimeeritud PHP koodi. Seda kasutatakse tihti peale peamiselt HTML-i väljastamiseks, kuid seda saab kasutada ka mis tahes muu tekstipõhise vormingu väljastamiseks [2].

Blade on lihtne, kuid samas võimas mallimootor (templating engine), mis kuulub Laraveli juurde. Erinevalt mõnest PHP mallimootorist ei piira Blade teid mallides lihtsa PHP-koodi kasutamist. Tegelikult kompileeritakse kõik Blade'i mallid tavaliseks PHP-koodiks ja paigutatakse vahemällu, kuni neid muudetakse, mis tähendab, et Blade lisab rakendusele sisuliselt null üldkulusid. Blade mallifailid kasutavad faililaiendit .blade.php ja neid hoitakse tavaliselt resources/views kataloogis [3].

CSS on keel, millega kirjeldatakse veebilehtede esitlust, sealhulgas värve, paigutust ja fonte. See võimaldab esitlust kohandada erinevat tüüpi seadmetele, näiteks suurtele ekraanidele, väikestele ekraanidele või printeritele. CSS on HTML-ist sõltumatu ja seda saab kasutada mis tahes XML-põhise märgistuskeelega.

Kogu projekti haldus toimus läbi Githubi vahenduse. GitHub on koodihaldus platvorm arendajatele, mis võimaldab meeskondadel töötada koos samade projektide kallal, olles eri paigus. Projekti käigus rakendati *pull request* metoodikat, kus mingi *issue* lõpetamisel tuli teha oma koodimuudatustest pull request oma branchist master branchi. Pärast pull requesti tegemist, vaatasid kõik meeskonnaliikmed muudatused üle, andsid tagasisidet ning kas andsid oma nõusoleku antud muudatuste masterisse lükkamiseks või saatsid pull requesti tagasi nõudes parandusi.

Meeskonna vaheliseks valitud kommunikatsioonimeetodiks valiti Microsoft Teams ja Discord, kus toimusid igapäevased kohtumised, suhtlus juhendajaga ning erinevad arutelud arenduse kohta.

Kiire rakenduste väljatöötamine (RAD) on arendusmudel, mis seab esikohale kiire prototüübi loomise ja kiire tagasiside pikaajaliste arendus- ja testimistsüklite asemel. Kiire rakenduste arendamise abil saavad arendajad kiiresti tarkvara mitmekordistada ja värskendada, ilma et oleks vaja iga kord arenduskava nullist alustada. RAD on arengumudel, mis tekkis siis, kui arendajad said aru traditsioonilisest juga arengumudelist ei olnud eriti tõhus. Waterfall mudeli peamine viga on see, et kui programm on testimisfaasis, on tarkvara põhifunktsioonide ja funktsioonide muutmine keeruline. See annab teile sisuliselt tarkvara, mis võib teie arenevale nõudele vastata või mitte [4].

2.3 Tööprotsessi kirjeldus

Tööprotsess algas esialgse kohtumisega meeskonnaliikmete vahel, mis toimus Teamsi vahendusel. Esimesel kohtumisel arutati läbi projekti eesmärk ning ajaline skoop. Kirjutati ning pakuti välja esimesi *issue*'sid, mille kallal hakati kogu meeskonnaga esialgselt töötama. Kuna puudusid varasemad kokkupuuted PHP programmeerimiskeelega, otsustati esimene nädal panna suurt rõhku eneseharimisele antud programmeerimiskeeles, et valmistuda projekti läbiviimiseks.

Ettevalmistuste hulka kuulusid näiteks erinevate internetipõhiste kursuste läbi vaatamine. Seejärel seadistati töökeskkond ning alustati *issue*'de kallal töötamisega.

Projekti arendamisel otsustati kasutada SCRUM metoodikat. SCRUM on agiilse tarkvara arendamise raamistik, mis aitab meeskondadel kursis olla iga meeskonnaliikme tehtud töö osas. Kuna kogu projekt viidi läbi kaugõppe vormis, oli antud raamistik eriti tähtis, kuna puudus füüsiline kokkupuude meeskonnaliikmete vahel. Kuigi tööd ei jaotatud otseselt sprintideks nagu muidu kohane on, jaotati siiski välja individuaalsed ülesanded ning pandi paika mingisugune ajaline periood mille jooksul peaks antud ülesanne tehtud olema. Meeskond kogunes vähemalt kord nädalas Teamsi keskkonnas, et saada ülevaadet iga liikme progressist, probleemidest ning edasisest plaanist. Töötati vähemalt 5 päeva nädalas, 6-10 tundi päevas.

Projekti läbiviimine oli jaotatud kahte etappi. Esimeses etapis pandi suurt rõhku funktsionaalsusele ning ideede ellu viimisele. Teises etapis pandi rohkem rõhku projekti visuaalsele poolele, et oleks võimalik projekti *deploy*'da. Iga nädal vaadati olemasolevad *issue*'d üle ning arutati meeskonnasiseselt, kes mingi *issue* saab. Alati vaadati meeskonnaliikme tugevusi ja nõrkusi ning *issue*'d määrati vastavalt. Seda just sellepärast, et vähendada ebavajalikku ajakulu. Juhul, kui keegi meeskonnaliikmetest sattus probleemi otsa, prooviti algselt antud probleem ise lahendada, kuid juhul kui sellega hakkama ei saadud, sai kogu meeskond Teamsi vahendusel kokku ning prooviti probleem koos lahendada. See tagas pideva töövoogu ning vältis olukordi, kus keegi jääb mingi teatud probleemi taha kinni ning ei saa tööga jätkata.

Iga teatud ajaperioodi järel saadi meeskonnaga Teamsi vahendusel kokku ning vaadati üle kogu perioodi jooksul tehtud töö. Selle eesmärgiks oli konstruktiivselt tagasi vaadata tehtud tööle ning anda meeskonnaliikmetele tagasisidet, mis läks hästi, mis halvasti, mida saaks paremini teha, et parandada edasiminevat tööd. Iga meeskonna liige tõi omalt poolt välja positiivsed ja negatiivsed arengukäigud, mis talle silma olid jäänud ning arutati, kuidas neid lahendada.

3 Töö tulemused

Projekti arendati meeskonna poolt väljastatud nõute põhiselt. Nõuded on välja toodud järgmises peatükis. Tegemist on iseseisva projektiga, kus meeskonnal puudusid ettevõtte pooled juhendajad. Rentif rendi platvorm arendati olemasoleva malli põhjal, kus meeskond arendas juurde töötava prototüübi jaoks kõik puudulikud lisafunktsioonid. Järgnevates alampeatükkides räägitakse rakendusest, arhitektuurist, disainis, koodist ning testimisest lähemalt.

3.1 Rakendus

Rendiplatvormi sisse logides avaneb kasutajal võimalus lisada kuulutusi ning rentida teiste kasutajate kuulutusi. Kasutajal on võimalus rendiplatvormi kasutada ka ilma sisse logimata, kuid sellisel juhul on kasutajal piiratud õigused ning puudub võimalus esemeid rentida. Tegemist on rendiplatvormiga kus toimub kasutajate vaheliste tehingute haldamine, millega on seotud ka inimeste isiklikud andmed. Selle tõttu oli meeskond sunnitud suurt rõhku panema turvalisusele. Kasutajad on registreerimisel sunnitud kinnitama telefoni numbri ja emaili aadressi, et vähendada libakasutajate arvu. Lisaks implementeeriti veebilehe turvalisuseks OVH anti-DDoS protection ning Cloudflare, et kaitsta veebilehte teenusetõkestusrünnakute eest.

3.1.1 Avaleht

Minnes veebilehele avaneb vaade meie *landing page*'le

General Transport Real estate Children's products Entertainment and hobbies
Sporting goods Electronics Services Fashion, style and beauty Home and Garden
Work Animals All for business



Haagis

€ 15,00 per day

Tallinn, EE
Admin

Kaamera

€ 10,00 per day

Tallinn, EE
Admin

Korter

€ 25,00 per day

Tallinn, EE
Admin

Arvuti

€ 5,00 per day

Tallinn, EE
Admin

BMW x5

€ 40,00 per day

Tallinn, EE
Admin

Lehepuhur

€ 5,00 per day

Tallinn, EE
Admin

Haagis

€ 15,00 per day

Tallinn, EE
Admin

My First Listing

£15,00

London, UK
Admin[VIEW ALL](#)

Joonis 1. Rendiportaali landingpage


Lehekülje *landing page*'l on kasutajal võimalik otsida tooteid läbi otsingumootori sisestades märksõnu, toodete nimesid või asukoha põhiselt. Lisaks on võimalik otsida tooteid kategooria põhiselt. Kategooriad on välja pakutud lehekülje poolt ning asetsevad otsingulahtri all. Et lehekülj


ei tunduks liiga tühjavõitu ja annaks kasutajale aimu, mida siit leida, pakub lehekülge kasutajate poolt lisatud kuulutusi. Kõikide kuulutuste vaatamiseks on lisatud nupp “View all”, mis viib vaatele kust on võimalik otsida kõiki lisatud kuulutusi.

3.1.2 Registreerimine ja sisselogimine

Valides veebilehe ülevalt paremast nurgast “Register”, avaneb külastajal registreerimisvorm, kus on võimalik antud veebilehele ennast kasutajaks registreerida. Ilma kasutajaks registreerimata ei ole inimesel võimalik postitada kuulutusi ega rentida teistelt inimestelt esemeid, samas on võimalik veebilehte kasutada ka ilma registreerimata, kuid siis on ainult võimalik kuulutusi vaadata. “Register” nupule vajutamisel avaneb kasutajal registreerimisvorm.

rentif

 Login with Facebook

 Login with Google

Name

e.g. Jack Sparrow

Email address

e.g. something@example.com

Phone number

 5123 4567

Password

Something secure

Confirm your password

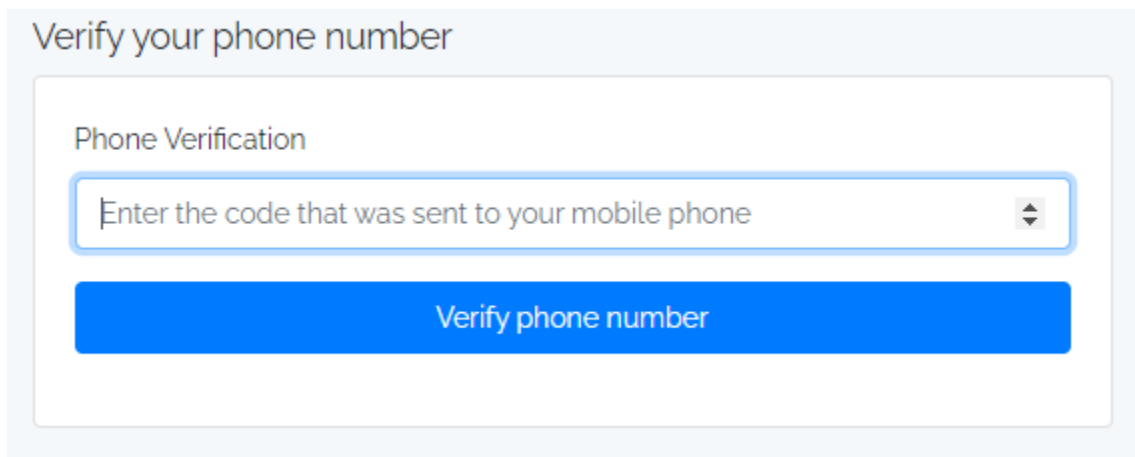
Sign Up

By signing up, you agree to our [Terms and Conditions](#).
Learn how we collect, use and share your data in our [Data Policy](#) and how we use cookies and similar technology in our [Cookies Policy](#).

Already have an account? [Log in](#)

Joonis 2. Rendiportaali kasutaja registreerimine

Et antud veebilehel kasutajaks saada peab inimene sisestama enda täisnime, emaili aadressi, telefoni numbrini ning määrama kasutajale parooli. Antud vormi on implementeeritud ka kõikide riikide *country code*'id. Telefoni numbrini lahtri kõrval asetseb lippude dropdown list, kust kasutaja saab soovi korral leida soovitud riigi ning *country code*'i. Implementeeritud on veel lisaks kasutaja asukoha põhine riigi koodi määramine, kus vaadatakse kasutaja IP aadressi lehele sisenemisel ning määratakse selle põhised vastav riigi kood, et kasutaja ei oleks sunnitud käsitsi riigi koodi antud listist välja otsima. Parooli valimisel on nõudeks valida parool, mis sisaldab vähemalt ühte suurt tähte, vähemalt ühte numbrit ning on vähemalt 9 karakterit pikk. Soovitud parooli valimisel palutakse kasutajal parool kinnitada. Juhul kui kõik lahtrid on õigselt täidetud ning kasutaja vajutab “Sign Up” nuppu, suunatakse kasutaja järgmisele lehele, kus kasutaja on sunnitud kinnitama nii oma telefoni numbrini, kui ka emaili aadressi. Veebilehele on implementeeritud *two-factor authentication*.



Verify your phone number

Phone Verification

Enter the code that was sent to your mobile phone

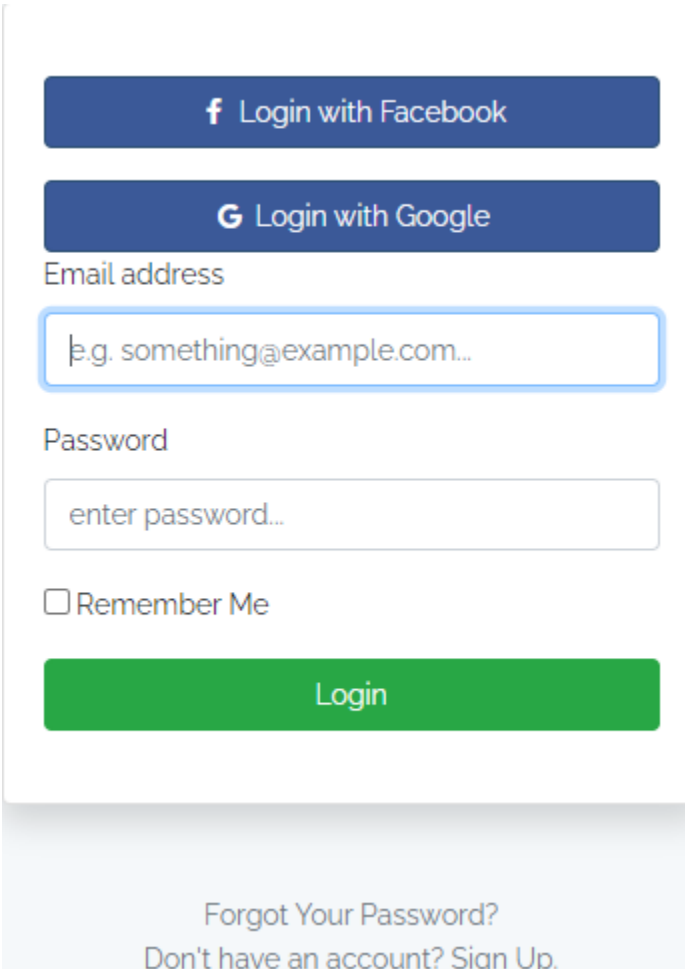
Verify phone number

Joonis 3. Kasutaja telefoni numbrini verifitseerimine

Kasutajale saadetakse tema poolt sisestatud telefoni numbrile kood, smsi vahendusel, mille kasutaja peab sisestama pildi peal näha olevasse lahtrisse. Kasutajale saadatud kood aegub 5 minuti jooksul. Juhul, kui kasutaja sisestab vale koodi on tal endiselt võimalik sisestada koodi uuesti, kuid juhul kui õiget koodi ei sisestata on registreerimisprotsess läbi kukkunud ning kasutaja peab protsessiga uuesti alustama. Juhul kui kasutaja sisestab õige koodi suunatakse ta järgmisele lehele, kus palutakse kasutajal kinnitada enda emaili aadress. Juhul kui kasutaja on kinnitanud

enda emaili aadressi suunatakse ta tagasi avalehele. Kasutajal on nüüd võimalik hakata kas soovitud esemeid välja rentima või mingilt teiselt kasutajalt esemeid rentima.

Inimesel on võimalik veebilehele registreeruda ka Facebooki'i või Google kaudu. Täpselt sama kehtib ka sisse logimisel. Juhul, kui kasutaja on ennast edukalt registreerunud, on tal võimalik ennast veebilehele sisse logida vajutades üleval paremal nurgas asetsevale "Sign In" nupule. Nupule vajutades avaneb järgnev vaade.

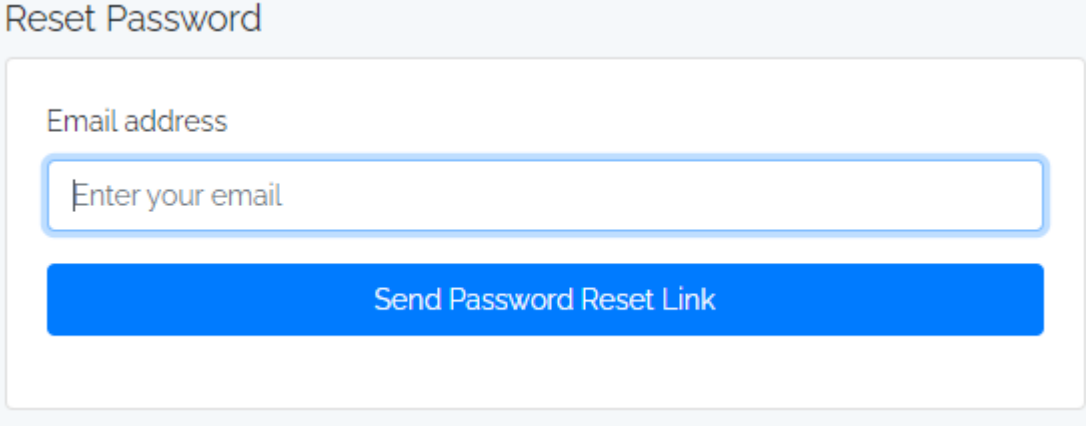


The image shows a login interface. At the top, there are two blue buttons: "Login with Facebook" (with a white 'f' icon) and "Login with Google" (with a white 'G' icon). Below these is a text label "Email address" followed by a white input field with a light blue border containing the placeholder text "e.g. something@example.com...". Underneath is a text label "Password" followed by a white input field with a light gray border containing the placeholder text "enter password...". Below the password field is a checkbox labeled "Remember Me". At the bottom of the form is a large green button labeled "Login". Below the main form area, there is a light gray box containing the text "Forgot Your Password?" and "Don't have an account? Sign Up." in a smaller font.

Joonis 4. Rendiportaali sisselogimine

Kasutajal on nüüd võimalik sisse logida, kas Facebook'i, Google või registreerumisel sisestatud andmete põhjal. Kasutajal on võimalik täita kast "Remember me", mis jätab kasutaja andmed

meelde ning logib kasutaja automaatselt veebilehele sisse. Juhul, kui kasutaja on unustanud parooli on tal võimalik vajutada “Sign up” vormi all asetsevale “Forgot your password?” tekstile, mis suunab ta järgmisele lehele, kus palutakse sisestada registreeritud emaili aadress.



Reset Password

Email address

Enter your email

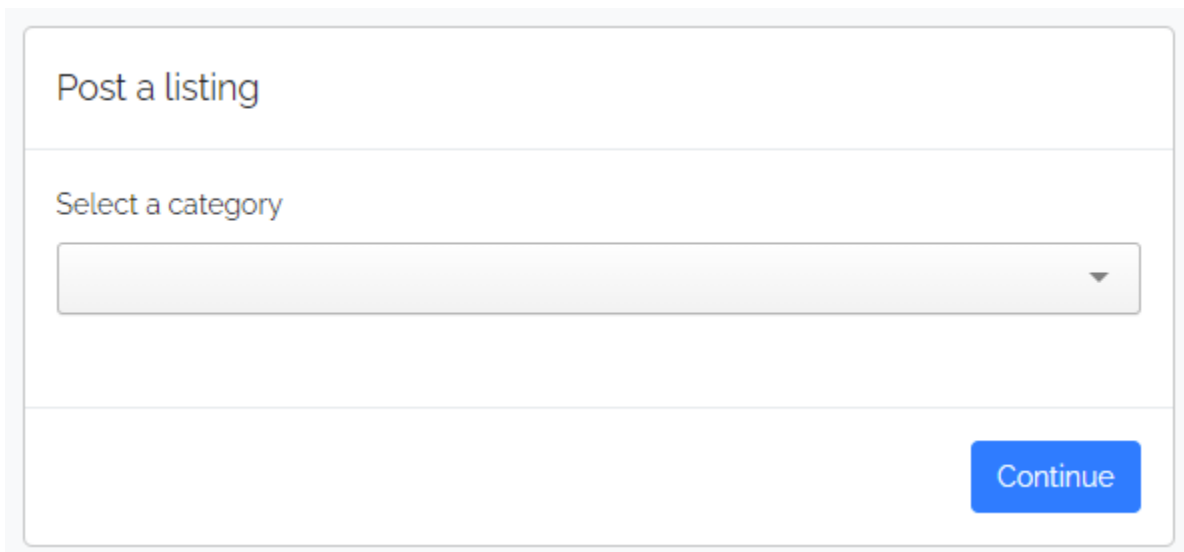
Send Password Reset Link

Joonis 5. Kasutaja parooli ununemine

Pärast emaili sisestamist saadetakse kasutajale emailile kiri, kus tal on võimalik oma parooli muuta.

3.1.3 Kuulutuse lisamine

Valides lehekülje ülevalt “Post listing”, avaneb vaade, kus on kasutajal võimalik alustada oma kuulutuse lisamist. Algselt on vaja kasutajal valida kategooria nimistikust kuulutusele sobilik kategooria.



Post a listing

Select a category

Continue

Joonis 6. Kuulutuse lisamise esimene vaade

Kategooria valimisel avaneb kasutajale järgmine vaade, mis nõuab kasutajalt toote nime ja lühitutvustust.

Post a listing

Category: General × Listing type: Rent an item ×

Title

Description

Normal ▾ **B** *I* U 🔗 ☰ ☷ ↵

You can upload images and set your own pricing in the next step.

Create & continue

Joonis 7. Kuulutuse lisamise teine vaade

Kui kasutaja on täitnud olemasolevad lahtrid on võimalik liikuda edasi viimasele vaatele. Eelnevale vaatele lisandub kaks lahtrit: tags ning toote lõpp kuupäev, millest kumbki ei ole kohustuslik väli. Lisaks tuleb kasutajal määrata toote asukoht, pildid, rendi hind, maksimaalne ning minimaalne rendiperiood, kättesaadavus ning vajaduse korral saab lisada lisandväärtusi tootele.

Editing Listing

Lehepuhur [\(view\)](#) unpublished

Listing type: Rent an item **Category:** General

Listing Details

Enter the title, description and location of your listing.

Images

Upload at least one image to make your listing stand out.

Pricing

Set-up pricing and availability for your listing.

Listing details

Title

Description

Normal **B** *I* U

Rentida lehepuhur, mis töötab nii aku kui ka elektri pealt. Puhur on äsja ostetud ning töökindel. Aku peal töötab kuskil 2-3 tundi ning võrreldes teiste puhuritega on ta vaikne. Lisaküsimuste korral võib kirjutada sõnumi.

Tags

Listing ends at (Leave empty for no end date)

Joonis 8. Kuulutuse lisamise kolmas vaade

Tags lahter annab võimaluse toodet läbi otsingumootori teistel kasutajatel paremini leida.

Location

10137 Tallinn, Estonia

City

Tallinn

Country

EE

Joonis 9. Kuulutuse lisamise neljas vaade

Toote asukoha määramiseks kasutame kasutaja IP aadressi, vajadusel kui antud teenus ei tööta on kasutajal võimalus kirjutada käsitsi täpne aadress ning nihutada kaardil olevat nupukest, mis täidab automaatselt lahtrid asukohapõhiselt.

Pricing information

Price in EUR

per day

Stock

Minimum rent period

▼

Maximum rent period (days)

Pricing details

Period	Price	Income
3 hour	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px 10px; display: inline-block;">3.50</div>	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px 10px; display: inline-block;">2.8</div>
6 hour	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px 10px; display: inline-block;">4.00</div>	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px 10px; display: inline-block;">3.2</div>
12 hour	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px 10px; display: inline-block;">4.50</div>	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px 10px; display: inline-block;">3.6</div>
1 week	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px 10px; display: inline-block;">23.00</div>	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px 10px; display: inline-block;">18.4</div>
1 month	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px 10px; display: inline-block;">75.00</div>	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px 10px; display: inline-block;">60</div>

[Change up to your preferences](#)

Joonis 10. Kuulutuse lisamise viies vaade

Toote hinna määramiseks oleme kasutanud võimalikult kasutajasõbralike lahendusi. Selleks tuleb kasutajal panna päevane rendihind, kui renditavat toodet on rohkem kui üks, saab määrata sobiva koguse ning tuleb määrata minimaalne ja maksimaalne rendiperiood. Kasutaja poolt määratud hinna põhjal arvutatakse automaatselt lehekülje poolt loodud optimaalsete kordajate abil perioodi hinnakiri, mida on võimalik kasutajal optimeerida vastavalt oma soovidele. Kasutajale on nähtav palju tulu on võimalik teenida valitud perioodide jooksul hinna kõrval olevas „Income“ lahtris.

Teenitav tulu saadakse hinna ja rendiportaali teenustasu lahutamise tulemusel. Viimaks on kasutajal vaja määrata, mis nädalapäevadel on võimalik toodet välja rentida.

Available dates & time

Availability Configure availability by weekdays

Available 24/7
Configure availability by weekdays

Weekday	Available	From	Untill
Monday	Yes	08:00	23:30
Tuesday	Yes	08:00	23:30
Wednesday	Yes	08:00	23:30
Thursday	Yes	08:00	23:30
Friday	Yes	08:00	23:30
Saturday	Yes	08:00	23:30
Sunday	Yes	08:00	23:30

Additional options & services

Name	Price	Max Qty	
e.g. Gift wrap	0.00		Delete

+ Add

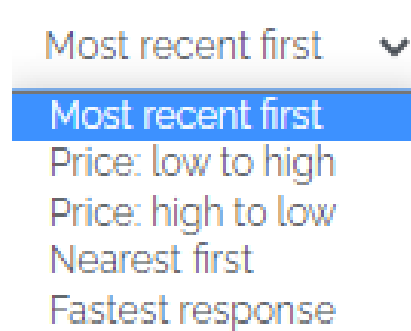
Publish

Joonis 11. Kuulutuse lisamise kuues vaade

Kasutajal on võimalik ise määrata, mis kellaegadel ning nädalapäevadel soovitud toodet välja rentida. Samuti lisandväärtuste puhul on võimalik lisada lahtreid ning määrata hind ja kogus.

3.1.4 Kuulutuste vaatamine ja sorteerimine

Valides avaleheküljel ülevalt “Browse” või lehekülje keskelt “View all” avaneb vaade kõikidest kasutajate poolt lisatud kuulutustest. Kuulutuste vaatamiseks on kasutajatele antud erinevaid võimalusi - näiteks vaadata kuulutusi *grid* vaates, *list* vaates ning kaardi vaates. Lisatud on ka võimalus kasutada *grid* ning *list* vaadet koos kaardi vaatega, mis annab kasutajale parema ülevaate kuulutustest.



Joonis 12. Kuulutuste sorteerimine

Kasutajal on võimalik kuulutusi sorteerida erinevate filtrite põhjal. Kasutajal on võimalik filtreerida kuulutusi, kuulutuse uudsuse, hinna, asukoha ja reageerimis kiiruse järgi. Lisaks saab kuulutusi sorteerida kategooria, distantssi ning hinnavahemiku alusel.

Distance

Anywhere ▾

Categories

- Transport
- Real Estate
- Children's Products
- Entertainment And Hobbies
- Sporting Goods
- Electronics
- Services
- Fashion, Style And Beauty
- Home And Garden
- Work
- Animals
- All For Business
- General

Price

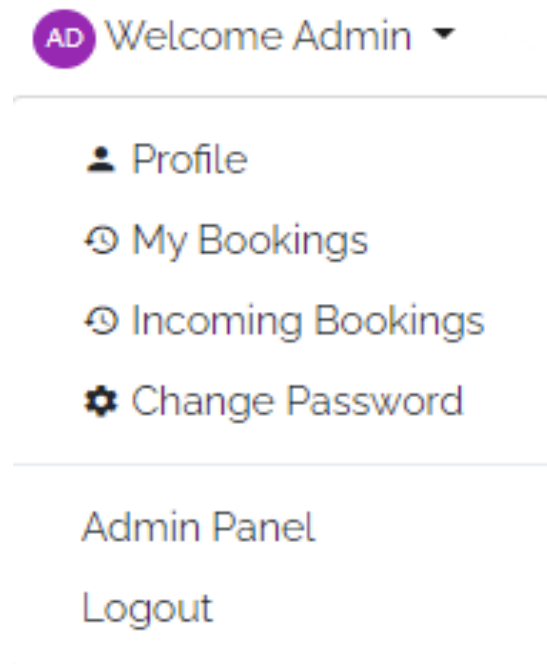
to

[Update results](#)

Joonis 13. Kuulutuste filtreerimine

3.1.5 Tavakasutaja paneel

Tavakasutaja paneel asetseb lehekülje üleval paremas nurgas kasutaja rippmenüü “Profile” küljes.



Joonis 14. Kasutaja rippmenüü

Rippmenüüs on võimalik kasutaja paneeli tähtsamatele toimingutele kiiresti ligipääseda otseteede kaudu. Vajutades “Profile” peale saame avada kasutaja paneeli vaate.

Menu

Edit Profile

Change Password

My Bookings

Favorites

Selling

Listings

Incoming Bookings

My Profile

Display Name: Admin

Username: admin
This will be used as your profile page URL.

Bio

Avatar

Choose File No file chosen

Your location

Enter your address.

City: Tallinn

Region: Harjumaa

Country: Estonia

Private information

Email: admin@example.com

Phone: +44

Gender: Not specified

Save profile

Joonis 15. Kasutajapaneel

Paneeli vasakul pool on nimekiri võimalikest tehtavatest toimingutest. “Edit profile” võimaldab vajadusel kasutajal peale kasutaja tegemist teha toiminguid, näiteks täpsustades enda asukohta, sugu, profiili pilti ning lühikirjeldust.

Change password

Email admin@example.com

Password

Confirm password

Save password

Joonis 16. Kasutaja paroolivahetamine

“Change Password” annab kasutajale võimalus muuta oma parooli, juhul kui parool on meelest läinud. Parooli muutmisel tuleb järgida parooli nõudeid, mis on 3.1.2 peatükis välja toodud.

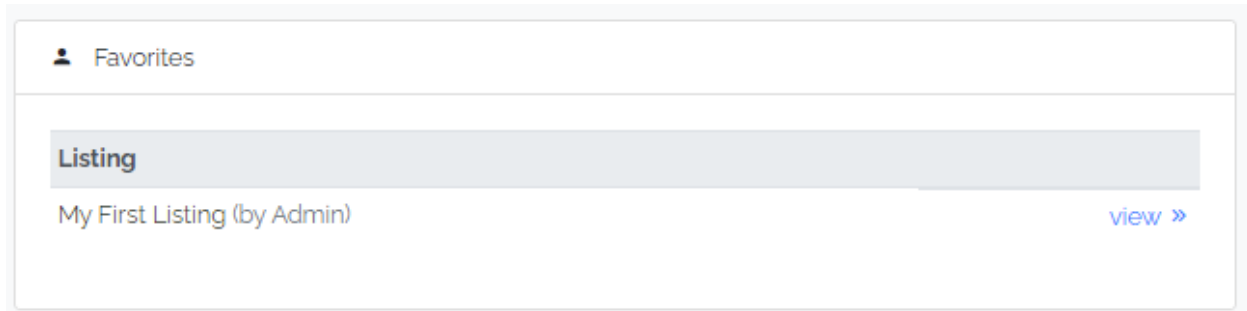
My Bookings

Listing	Booking Number	Amount	Date Ordered	Status	Start Date	End Date
---------	----------------	--------	--------------	--------	------------	----------

Joonis 17. Kasutaja poolt tehtud broneeringud

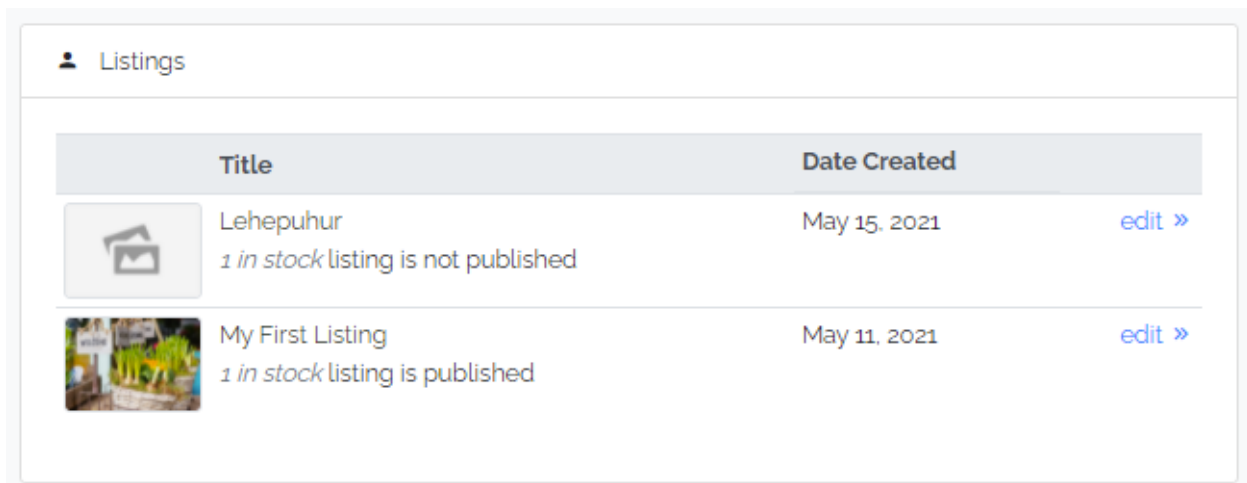
“My Bookings” vaade näitab kasutaja poolt tehtud broneeringuid. “Listing” näitab, mis kuulutusega on tegemist, kuulutuse tiitli näol. “Booking Number” on unikaalne number igale tehtud broneeringule, mille kaudu on võimalik lihtsasti klienditoel leida probleemi korral kasutaja tehtud broneering. “Amount” on renditava eseme hind. “Date Ordered” annab kasutajale ülevaate, mis kuupäeval on mingi broneering tehtud. “Status” näitab broneeringu olekut. Broneeringu tegemisel on olek alati “Open”, mis tähendab, et kasutaja broneeringut ei ole veel vastu võetud ega tagasi lükatud. Broneeringu vastuvõtmisel muutub olek “accepted” ja tagasilükkamise korral

“declined”. “Start Date” ja “End Date” näitavad tehtud broneeringu rentimise algus -ja lõppkuupäeva.



Joonis 18. Kasutaja poolt salvestatud kuulutused

“Favorites” vaates on võimalik kasutajal näha salvestatud kuulutusi. Salvestatud kuulutused kuvatakse kuulutuse tiitli ja omaniku kujul. Vajutades otsetee “View” peale, avaneb salvestatud kuulutus eraldi leheküljel.



Joonis 19. Kasutaja kuulutused

“Listings” vaade kuvab kasutaja poolt tehtud kuulutusi, mis võimaldab lihtsasti ligipääseda ning hallata kuulutusi. “Title” on kuulutuse nimi, mis sisaldab lisainformatsiooni sellest, kas kuulutus on tehtud avalikuks, või ei. “Date Created” näitab mis kuupäeval antud kuulutus on tehtud. “Edit”

on vaate otsetee kuulutuse redigeerimiseks, kus on võimalik kasutajal teha kõiki muudatusi kuulutusega seoses.




Listing	Buyer	Booking number	Amount	Date Booked	Status
	Admin	VDMW3	40.20 GBP	Mar 19, 2021	accepted View »

Joonis 20. Kasutajale sissetulevad broneeringud

“Incoming Bookings” vaates esinevad kõik teiste kasutajate poolt tehtud broneeringud. “Listing” annab teada, mis kuulutusega tegemist on. “Buyer” on broneeriija kasutajanimi ning “Booking number” on unikaalne number antud broneeringu jaoks, mille kaudu on võimalik probleemide korral klienditoel lihtsasti kuulutus ülesse leida. “Amount” näitab broneeringust saadavat tulu, “Date Ordered” kuupäeva millal broneering tehtud ning “Status”, kas broneering on vastu võtmata, tühistatud või aktsepteeritud. “View” avab detailse vaate broneeringust.

[« Back to bookings](#)

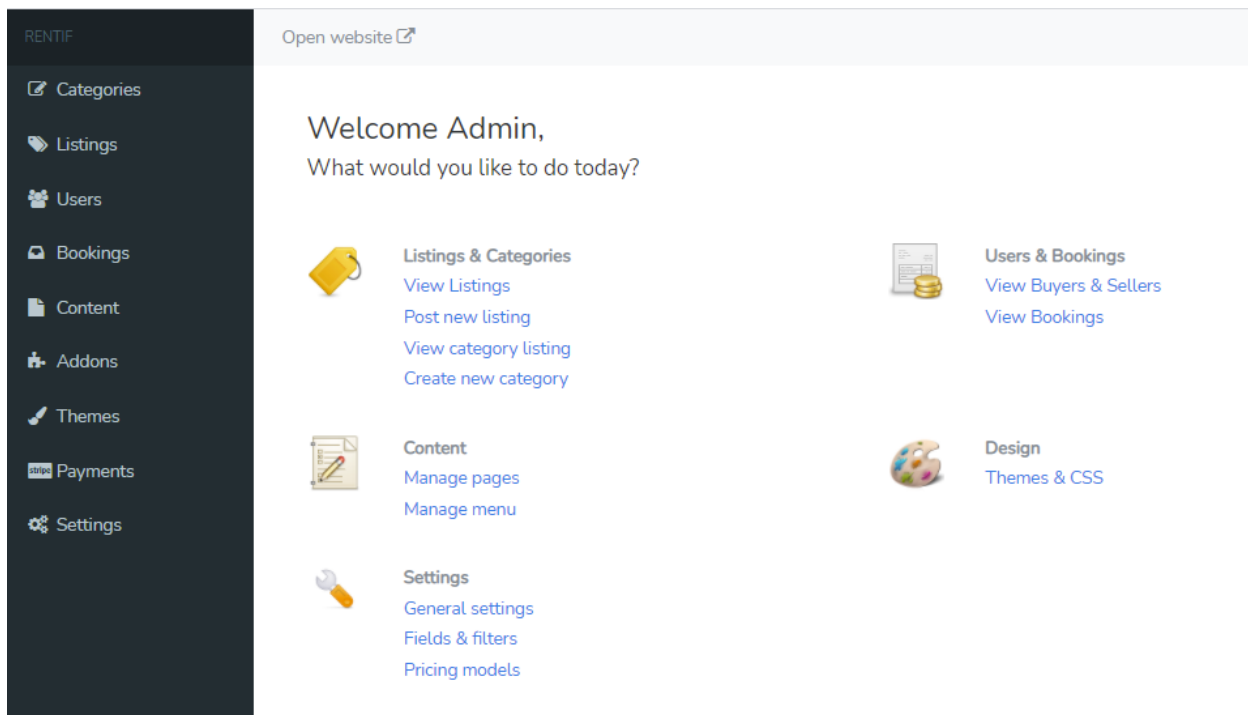
 Booking #7	accepted
Booking date	Fri, Mar 19, 2021 11:11 AM
Listing	My First Listing
User choices	Option 1: sfdsfdsfd Start day: Mar 20, 2021 End day: Mar 22, 2021
Total paid	£40.20
Marketplace Fee	£9.20
Status	accepted
Accepted at	Mar 19, 2021

Joonis 21. Detailne ülevaade broneeringust

Vaates on näha kasutaja poolt tehtud valikud ning valitud rendi periood.

3.1.6 Lehekülje haldaja paneel

Lehekülje haldaja paneelile ligi pääsemiseks peab sisse logitud olema admini kasutajaga. Kui admini kasutajaga on sisse logitud tekib ülevale paremasse nurka mutrivõtme märk, selle peale vajutades suunatakse administraator lehekülje haldaja paneeli.



Joonis 22. Administraatoripaneel

Lehekülje haldaja paneelil (edaspidi admin paneel) on administraatoril võimalik:

- hallata kogu lehekülge
- kustutada postitusi
- lisada postitusi
- vaadata postitusi
- vaadata kategooriaid
- lisada uusi kategooriaid
- jälgida tellimusi
- jälgida ostjaid ja müüjaid
- lisada erinevaid teemasid ning neid muuta

- lisada maksemeetodeid ning neid eemaldada

Näiteks “Create new category” nupule vajutamisel avaneb uus lehekülg, kus administraator saab lehele lisada uue kategooria.

[< back](#)

Adding new category

Name

Order

Parent category

--SELECT--

Seller listing type

- Post an announcement
- Sell something
- Rent room
- List your service
- Rent an item
- Request

Submit

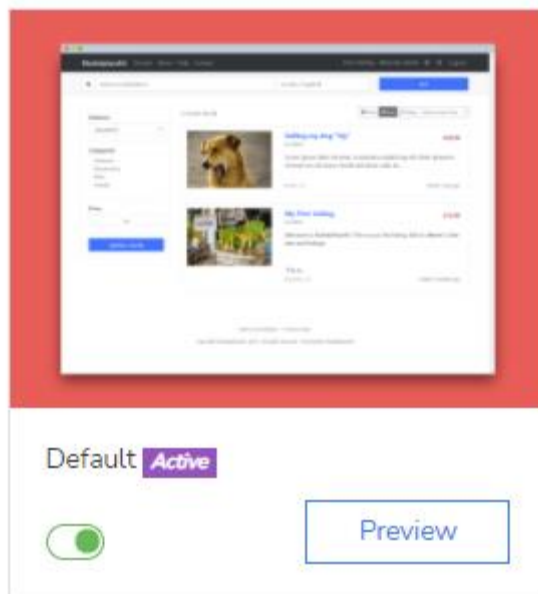
Joonis 23. Kategooriate lisamine

Juhul kui administraator vajutab näiteks “Themes and CSS” nupule avaneb teemade list. Teemasid on võimalik juurde lisada vajutades “Upload Zip” nupule üleval paremal nurgas. Teemasid on võimalik vaadata eelvaates ning lihtsasti kas aktiveerida või eemaldada vajutades vasakul all asetsevale nupule.

Lehekülje haldamine toimub läbi admin paneeli, kus lehe administraator saab jälgida kasutajaid ning implementeerida uuendusi.

Themes

[Installed Themes](#) [Upload Zip](#)



Joonis 24. Lehekülje disaini vahetamine

3.1.7 Kuulutuse broneerimine ja maksmine

Kuulutuse leheküljel on võimalik broneerida toodet.

Home / General

Lehepuhur

Published Edit listing Add to spotlight Disable listing

Admin



Successful transactions: 1000123000



VERY HIGH
ACCEPTANCE RATE



1 day 20 hours
AVERAGE RESPONSE TIME



Overview

Save to favorites

Rentida lehepuhur, mis töötab nii aku kui ka elektri pealt. Puhur on äsja ostetud ning töökindel. Aku peal töötab kuskil 2-3 tundi ning võrreldes teiste puhuritega on ta vaikne. Lisaküsimuste korral võib kirjutada sõnumi.

5.00€/day

3.50€

3 hour

4.00€

6 hour

4.50€

12 hour

23.00€

1 week

75.00€

1 month

Select a date range

17.05.2021 to 17.05.2021

Rent start

08:00

Return

11:00

Minimum rent period: 3 hour

Maximum rent period: 1 month

Availability

Monday	08:00	21:00
Tuesday	08:00	21:00
Wednesday	08:00	21:00
Thursday	08:00	21:00
Friday	08:00	21:00
Saturday	unavailable	
Sunday	unavailable	

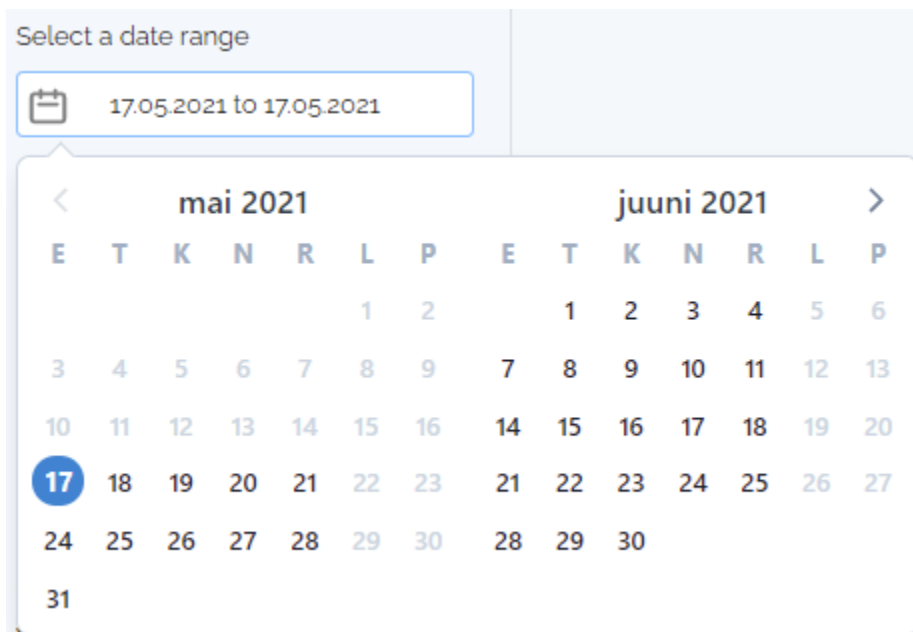
5.00 x 1 day € 5.00

Total € 5.00

Book now >

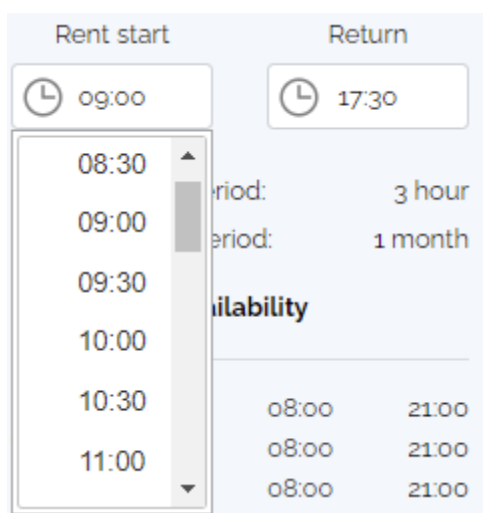
Joonis 25. Kuulutus

Esmalt on näha paremal pool toote hinnakirja ning toote kättesaadavust. Kui kasutajal on teada, mis päevadeks ta soovib rentida antud toodet, siis vajutades "Select a date range" kalendri lahtrisse avaneb kalendri vaade.



Joonis 26. Kalender

Kalendri vaates on võimalik valida rendiperiood vajutades ühe korra soovitud kuupäevale ja teist korda vajutades samale või järgmisele kuupäevale olenevalt rendiperioodi pikkusest. Järgmisena on vaja valida rendi algus -ja lõppkellaeg.



Joonis 27. Ajavalimine

Sobiva kellaaja valimiseks tuleb klõpsata lahtri peale ning avaneb rippmenüü vabadest kellaegadest. Kui kasutaja on täitnud kõik eelnevad toimingud on võimalik alustada broneeringu täide viimisega. Vajutades nupule “Book now” avaneb uus vaade maksekorraldusest.


rentif

[← Back](#)

Secure checkout

Address

Order summary




Lehepuhur
Admin


Start day	17.05.2021 09:00
End day	20.05.2021 17:30

5.00 x 4 days	€ 20.00
Total	€ 20.00

Shipping address is the same as my billing address

Payment methods





Place order

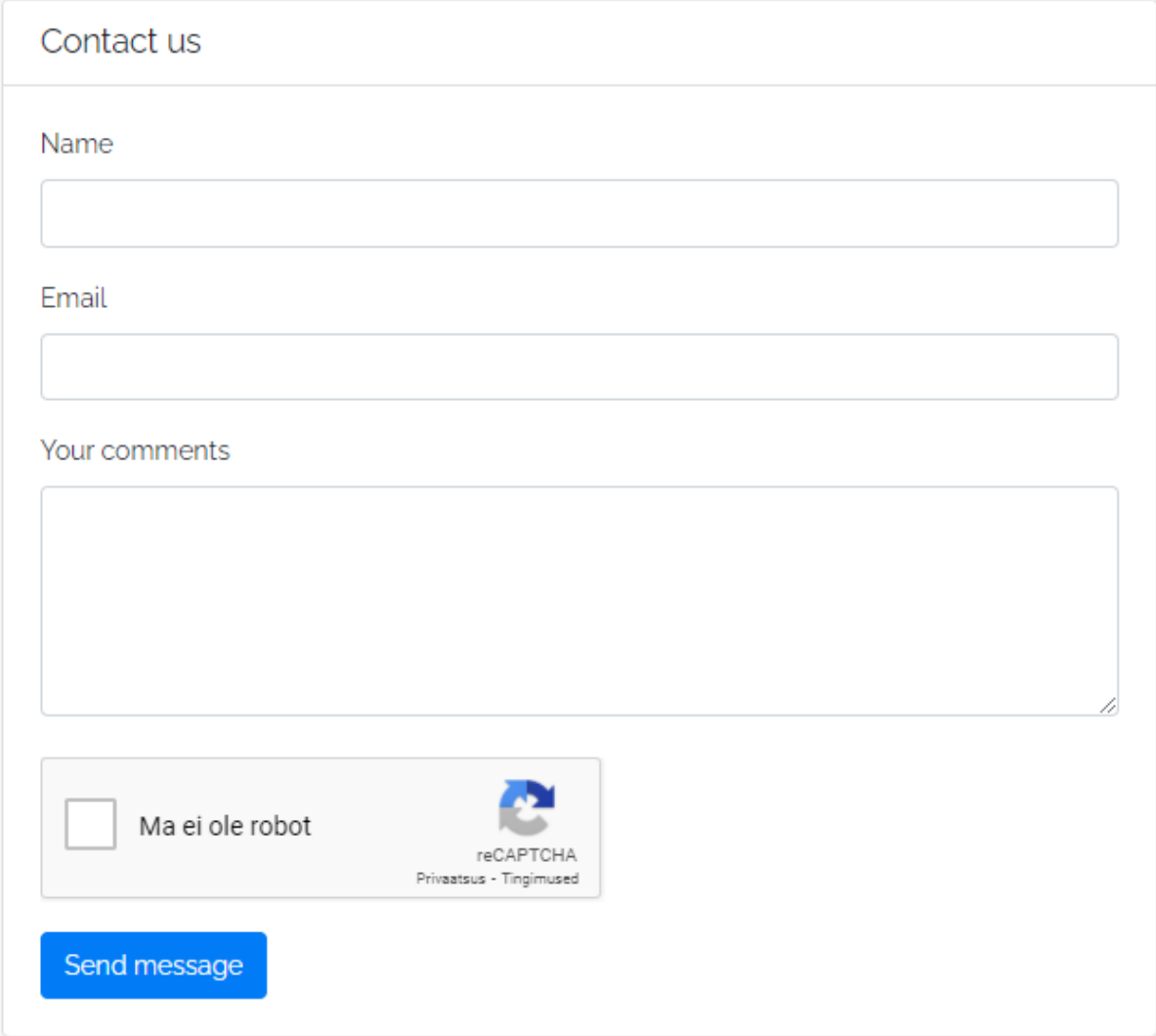
By clicking "Place Order" I approve the User Terms and confirm I have read the Privacy Notice. I agree to the terms & conditions of this Merchant.

Joonis 28. Maksekorraldus

Makse tegemiseks on vajalik täita kõik lahtrid milleks on “Full Name”, “Address”, “City”, “State”, “Post code/Zip” ning “Country”. Samuti on võimalik märkida linnukesega, kas kasutaja soovib toote saada samale aadressile, mis on märgitud maksevormis. Järgmiseks tuleb valida endale sobiv maksemeetod. Hetkel on võimalikeks makse meetoditeks SEB ning PayPal.

3.1.8 Kontakteerumine

Valides lehekülje ülevalt “Contact”, avaneb vaade, kus on võimalik kasutajal pöörduda oma murede/probleemidega klienditoe poole. Selleks tuleb täita olemasolevad lahtrid milleks on kasutaja nimi, e-maili aadress ja probleemi lühitutvustus. Enne kirja teele saatmist on vaja tõestada, et tegemist on inimesega, mis aitab vältida “spam” rünnakuid veebilehe vastu.



Contact us

Name

Email

Your comments

Ma ei ole robot

reCAPTCHA
Privateatus - Tingimused

Send message

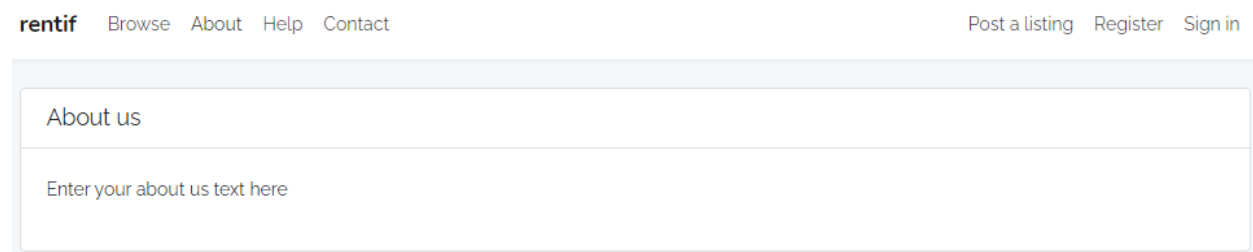
Joonis 29. Kontakt

Pärast kasutaja poolt kirja saatmist, saab klienditugi teatise. Sellest hetkest on võimalik klienditoe

alustada kliendi abistamisega ning vajadusel küsida lisainformatsiooni. Kogu informatsiooni vahetus sellest hetkest saab olema kliendi määratud e-maili aadressi ning meiepoolse klienditoe e-maili aadressi vahel.

3.1.9 Lisainformatsioonid

Leheküljel on võimalik leida erinevaid informatsiooni platvorme. Informatsiooni platvormideks on KKK, kasutustingimused ning privaatsuspoliitika. Platvormi kohta kuuluvat informatsiooni on võimalik leida “About” nupule vajutades, mis asetseb lehekülje üleval vasakus nurgas.



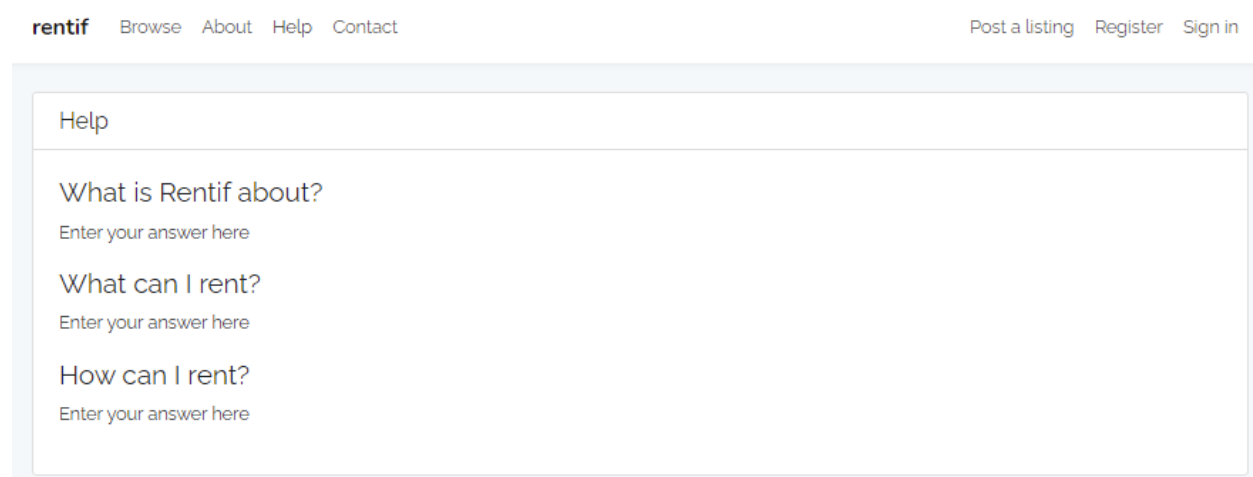
rentif Browse About Help Contact Post a listing Register Sign in

About us

Enter your about us text here

Joonis 30. KKK

KKK kohta saab lugeda lehekülje ülevalt “Help” vaates.



rentif Browse About Help Contact Post a listing Register Sign in

Help

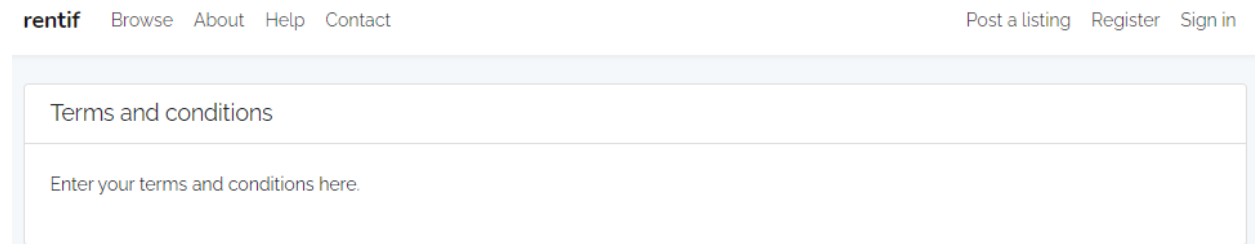
What is Rentif about?
Enter your answer here

What can I rent?
Enter your answer here

How can I rent?
Enter your answer here

Joonis 31. Abi

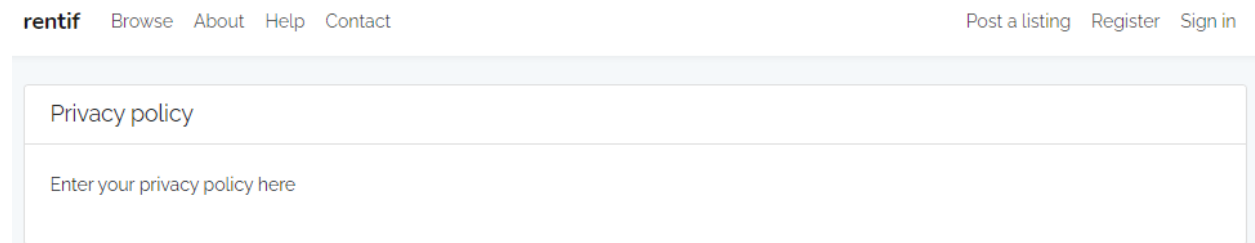
Kasutustingimuste informatsioon on leitav veebilehe alumises osas “Terms of Conditions” vaates.



The screenshot shows the top navigation bar of the 'rentif' application. On the left, there is a logo 'rentif' followed by links for 'Browse', 'About', 'Help', and 'Contact'. On the right, there are links for 'Post a listing', 'Register', and 'Sign in'. Below the navigation bar is a form titled 'Terms and conditions'. The form has a header section with the text 'Terms and conditions' and a main input area with the placeholder text 'Enter your terms and conditions here.'

Joonis 32. Kasutustingimused

Ning privaatsuspoliitikast saab lugeda samuti veebilehe alumises osas leitavast “Privacy Policy” vaatest.

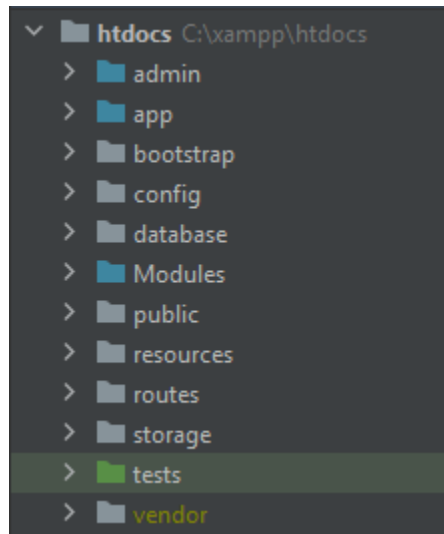


The screenshot shows the top navigation bar of the 'rentif' application, identical to the previous one. Below it is a form titled 'Privacy policy'. The form has a header section with the text 'Privacy policy' and a main input area with the placeholder text 'Enter your privacy policy here.'

Joonis 33. Privaatsuspoliitika

3.2 Arhitektuur

Projekti struktuur on ehitatud Laravel raamistiku baasil. Laravel on tasuta avatud lähtekoodiga veebiraamistik, mis on välja töötatud MVC arhitektuurimudeli abil. Laravel vabastatakse MIT-i litsentsi alusel. Raamistik tegeleb ise struktuuri ülesseadmisega - muutes projekti arendamist kiireks ja mugavaks.



Joonis 34. Projekti struktuur

- admin - Administraatori paneeli kõik vajalikud failid: kontrollid, vaated ja vormid.
- app - Veebilehe tuum, kus asuvad kõik kontrollid, API-d ja vidinad.
- bootstrap - Bootstrap raamistiku failid, kus on deklareeritud kõik selle teed ja sõltuvused.
- config - Veebilehe kõik konfiguratsiooni failid ja API-de võtmed.
- database - Kõik andmebaasi migratsioonid, mis on vaja veebilehe jätkusuutlikuks tööks.
- Modules - Administraatori paneeli kaudu installitud moodulid.
- public - Kompileeritud veebileht, mis näidatakse tavakasutajale. Selles kaustas on kõik css, pildid ja javascript liidesed.
- resources - Kõik tavakasutaja vaated, imporditud paketid (*JS*, *Vue.js*) ja tõlked.
- routes - Kõik viited veebilehe kontrollidele: mis tüüpi ja mis andmeid kontrollid nõuavad.
- storage - Kõik raamistiku genereeritud *cache* failid ja logid.
- tests - Kõik projekti testid.

- vendor - *Composer*’i installeeritud paketid.

3.3 Disain

Disaini osas lähtuti suuresti juba olemasolevatest analoogsetest rendi platvormidest ning kasutusele võetud malli disainist. Kuna puudus varasem kogemus sellise veebiportaali arendamisel, otsustati näpunäiteid ja soovitusi võtta analoogsetelt rendi ja ostumüügi platvormidelt. Disaini loomisel lähtuti kasutajasõbralikkuse põhimõtetest, et vältida olukordi, kus disain muutub liiga arusaamatuks ning raskesti kasutatavaks. Palju informatsiooni ning tagasisidet saadi just pereliikmetelt ja sõpradelt, kelle seas koostati küsitlusi ning uuringuid disaini osas (mis on visuaalselt rahuldav).

Kasutajaliidese puhul püüti disain teha võimalikult kasutajasõbralikuks. Kuna antud platvormi hakkavad kasutama tavalised inimesed kellel ilmselt puutuvad varasemad kokkupuuted IT-maailmaga, siis oli eriti tähtis, et kasutajaliides oleks võimalikult arusaadav ning lihtsalt navigeeritav. Kasutati erinevaid piltide ja värvide kombinatsioone, et tähelepanu tähtsamatele kohtadele lehel suunata. Avalehele on paigutatud võimalikult vähe funktsionaalsusi ja informatsiooni, et inimesel tekiks selge ülevaade lehel toimuvast. Lehe keskel asetseb otsingukast, kus inimesel on koheselt võimalik otsida soovitud toodet. Kogu lehe vältel on kasutatud kindlaid värvi kooslusi, et leht ei muutuks liiga kirjuks ning et kasutajal oleks võimalik lehega harjuda. Kogu projekti vältel on lähtutud kasutajasõbralikkusest.

3.4 Kood

Rakenduse valmimiseks kirjutatud kood jaotub kaheks - back-end ja front-end. Front-end on kirjutatud Twig ning Blade arenduskeeletes, back-end PHP arenduskeeles. Andmebaasi puhul kasutati MySQL teenust ning veebilehe kuvamiseks localhost’is Apache teenust. Apache ja MySQL haldamisel/käivitamisel kasutati XAMPP rakendust. Koodi kirjutamisel lähtuti puhta koodi põhimõtetest. Näiteks meetodite ja klasside nimetamisel kasutati tähenduslike nimesid, mis olid hästi arusaadavad ning hääldatavad, klasside nimetamisel kasutati nimisõnasid ning meetodite nimetamisel tegusõnasid. Meetodid ja funktsioonid olid fokusseeritud ainult ühele kindlale probleemile. Välditi koodi kordusi.

3.4.3 Andmebaas

Andmebaasina kasutati MySQL'i. MySQL on relatsiooniline andmebaaside haldussüsteem, mis põhineb SQL - struktureeritud päringukeelel. Rakendust kasutatakse mitmesugustel eesmärkidel, sealhulgas andmeladustamine, e-kaubandus ja logirakendused. MySQL-i kõige tavalisem kasutamine on siiski veebiandmebaasi jaoks. Selle abil saab salvestada kõike, alates ühest teabekogust kuni kogu veebipoe jaoks saadaolevate toodete nimekirjani. Koos sellise skriptikeelega nagu PHP või Perl (mõlemat pakutakse meie hostikontodel) on võimalik luua veebisait, mis reaalajas suhtlevad MySQL-i andmebaasiga, et kuvada veebisaidi kasutajale kiiresti kategoriseeritud ja otsitavat teavet [5].

3.4.4 XAMPP

XAMPP on Apache Friends'i välja töötatud tasuta ja avatud lähtekoodiga platvormidevaheline veebiserveri lahenduste virnapakett, mis koosneb peamiselt Apache HTTP serverist, MariaDB andmebaasist ja PHP ja Perli programmeerimiskeeltesse kirjutatud skriptide tõlkidest. Kuna enamused tegelikke veebiserveri juurutusi kasutavad samu komponente kui XAMPP, muudab see võimalikuks ülemineku kohalikult testserverilt reaalserverisse.

3.5 Testid

Kuna projekti arendamisel rakendati RAD meetodikat, ei pandud rõhku testide kirjutamisele. Testide kirjutamine ei kuulu RAD'i juurde ning testide kirjutamise asemel keskendutakse rakenduse kiirele arendamisele ning välja laskmisele. Kuna projekti arendamiseks oli aega vähe, otsustati RAD'i kasuks ning testide kirjutamine jäeti järgmistesse tsüklitesse.

Juhul, kui projekti arendamisega jätkatakse ka pärast projekti esitamist lõputööna, on meeskonnal plaanis kirjutada valmis projekti puhul vajalikud testid.

4 Analüüs ja järeldused

Antud peatükis analüüsitakse senimaani tehtud tööd ning tehakse sellest järeldused. Analüüsitakse nõudeid, disaini ning koodi.

4.1 Tehnilise teostuse analüüs

4.1.1 Nõuded

Nõuete kirjutamisel lähtuti eelkõige projekti põhilisest visioonist. Antud rendi platvormi kasutavatel inimestel oleks võimalik turvaliselt ja kiirelt kas rentida esemeid või neid välja rentida. Kuigi muud sarnased veebilehed juba eksisteerivad (rentist, osta, okidoki, soov24), ei pakuta just kõige suuremat valikut ning tihti ollakse fokuseeritud ainult ühele kindlale valdkonnale (näiteks Autod). Klientidel peab olema võimalus kõik vajalikud esemed rentida ühelt veebiplatvormilt. Selliste veebiplatvormide puhul on üks olulisemaid aspekte turvalisus. Kuna inimesed rendivad oma personaalseid esemeid, siis peab kogu protsess olema võimalikult lihtne ja turvaline, et vältida kelmuste juhtumeid. Just seda meeles pidades otsustati implementeerida *Two-factor authentication*, kus enne kasutajaks saamist, tuleb sisestada enda e-mail, telefoni number, nimi ning aadress. Et välistada juhtumeid, kus inimesed sisestavad võltsitud andmeid otsustati kasutada *Two-factor authentication*'i, kus kasutaja peab kinnitama enda e-maili aadressi ning telefoni numbri. Esialgu saadetakse kasutajale kiri sisestatud e-maili aadressile, kus palutakse kasutajal kinnitada oma e-maili aadress, pärast e-maili aadressi edukat kinnitamist saadetakse kasutajale kood sms'i vahendusel kasutaja poolt sisestatud telefoni numbrile, mille kasutaja peab sisestama veebilehele.

4.1.2 Arhitektuur

Projekt on MVC arhitektuuri baasil ülesse ehitatud. Projektis kasutasime antud arhitektuuri sellepärast, et kasutatud mall oli tehtud Laravel raamistiku baasil, samuti oli arhitektuuri struktuur varasemalt tuttav. Olemasoleva arhitektuuri muutmiseks ei näinud me mõjuvat põhjust, kuna projekt on loogiliselt ülesse ehitatud ja kõik elemendid on vastavalt omas kohas. Kasutatud arhitektuuri struktuuri on välja toodud Laravel dokumentatsioonis, kus miski element peaks asuma.

Kasutatud raamistiku eelis on see, et arendaja ei pea kogu projekti ise algusest konfigureerima, parandavad koodi ülevaadet ja loetavust. Laraveli vaikimis rakenduse struktuur on mõeldud pakkuma suurepärasest lähtepunkti nii suurtele kui ka väikestele rakendustele, võimaldades rakenduse korraldamist paindlikuks. Laravel ei kehtesta peaaegu mingeid piiranguid selle kohta, kus mõni konkreetne klass asub - seni, kuni arendaja saab klassi automaatselt edasi anda.

Alternatiivselt oleks võinud kasutada mingit teist arhitektuuri, nagu näiteks FuelPHP, mis on samuti tuntud framework PHP jaoks. FuelPHP on paindlik, täiskoguga PHP raamistik, mis ilmus esmakordselt 2011. aastal. Lisaks MVC kujundusmusteri toetamisele on sellel ka oma versioon, mida nimetatakse HMVC. Erinevalt MVC-st ei pea HMVC-ga sisu paljudel lehtedel kuvamiseks dubleerima. Seetõttu kulutab see vähem aega ja mälu [6]. Antud projekti jaoks siiski otsustati Laraveli kasuks, mis töötab MVC baasil. Seda just sellepärast, et esialgselt kasutusele võetud mall oli kirjutatud laraveli baasil ning ei soovitud kasutusele võtta mingit teist frameworki.

4.1.3 Disain

Rakenduse disain oli suuresti mõjutatud kasutusele võetud mallist. Üleüldine disaini struktuur tulenes ees olevast mallist, mida muudeti ning arendati kogu projekti vältel enda vajaduste kohaselt. Rakenduse disaini protsessi olid kaasatud ka meeskonna pereliikmed ning tuttavad, et oleks võimalik rakenduse disaini neutraalsest vaatepunktist vaadata ning saada rohkem informatsiooni tänapäeva inimeste soovidest ja vajadustest veebilehtede disaini osas. Kuna puudusid otsesed nõuded, oli meeskonnal võimalik implementeerida disain ise. Suurt rõhku pandi kalendri disainile, mille kaudu said kliendid määrata rendi perioodi pikkust ja rentimiseks vabu olevaid aegu.

Kasutajaliidese raamistikuna kasutati Bootstrap'i. Bootstrap (ka Twitter Bootstrap) on tasuta tööriistakomplekt veebilehtede ja veebirakenduste loomiseks. Sisaldab JavaScripti pluginaid, HTML- ja CSS-kujundusmalle tüpograafia, vormide, nuppude, navigatsiooni, teadete ja teiste veebiliideste komponentide jaoks. Bootstrap kasutab uuemat CSS-i ja HTML-i, seetõttu on vaja olla väga tähelepanelik kasutades vana veebibrauserit (näiteks aegunud IE brauser) [7].

Bootstrap on teisel kohal kõige tähtsamatest projektidest GitHubis. Tal on rohkem kui 123,000 tähte [7].

Kogu projekti vältel kasutati just kõige rohkem Bootstrapi CSS-kujundusmalle, olid olemas ka eraldiseisvad CSS failid, kuid enamus disainist implementeeriti CSS-kujundusmallide abil. Bootstrap raamistiku kasuks otsustati just sellepärast, et esialgne mall sisaldas juba kujundusmalle ning nende eemaldamine ning kasutajaliidese poole täielik ümbertegemine oleks võtnud liiga palju ajalisi ressursse, mida antud lõputöö tegemisel kahjuks võimalik ei olnud. Otsustati tutvuda lähemalt Bootstrapi metoodikaga, mis osutus väga mugavaks ning kasulikuks raamistikuks. Selle tulemusena oli kogu meeskond ühel arvamusel ning otsustati antud raamistikuga jätkata.

4.1.4 Kood

Kogu projekti vältel järgiti raamatu Clean Code põhimõtteid. Seda ei tehtud alati tahtlikult, kuid tänu koolis õpitule, olid antud raamatul põhinevad põhimõtted juba sisse harjutatud ning tulid enamus olukordades naturaalselt. Kõik muutujate nimed on kirjeldavad ja üheselt mõistetavad. Kasutusel olevad nimed on hääldatavad ja arusaadavad. Ei kasutatud ühe või kahe tähelisi nimesid. Klasside nimetused on UpperCamelCase ning meetodid, funktsioonid ja muutujad on lowerCamelCase formaadis. Funktsioonid on mahult väiksed ning teevad ainult ühte kindlat ülesannet. Funktsioonide nimed on arusaadavad ning kirjeldavad funktsiooni sisu. Teevad ainult ühte kindlat ülesannet, puuduvad kõrvalmõjud.

Konfigureeritavad andmeid hoitakse kõrgel tasemel. Eelista polümorfismi kui if / else või switch / case. Eralda mitme keermeline (multi-threading) kood. Vältida liigset seadistatavust. Järgida Demeteri seadust. Klass peab teadma ainult oma otseseid sõltuvusi. Loogelised avavad sulud on kirjutatud ridadel asuvate koodijuppide lõppu ning reavahetus toimub pärast avavat loogelist sulgu. Iga sulgev loogeline sulg on uuel real ehk reavahetus toimub enne sulgevat loogelist sulgu. Kasutada pärimist, et vältida koodi duplikatsiooni ning üleliigsete klasside moodustumist. Iga klass ja kontrolleri vastutab/sooritab ainult ühte ülesannet. Kõik nimed peavad viitama klassi/kontrolleri/protseduuri tehtavale tööle, et vältida segadust.

Kommentaaride kirjutamisel püüti lähtuda Clean Code põhimõtetest. Püüdke end alati koodiga selgitada. Ära kirjuta üleliigseid kommentaare. Ära lisa üleliigset müra, see tähendab, et vältida ebavajaliku ning iseenesest mõistetava teksti kirjutamist. Ära kasuta sulgude sulgemiskommentaare. Ära kommenteeri välja koodi, mitte kasutusel oleva koodi tuleks alati eemaldada. Kasuta kavatsuse selgitusena. Kirjuta kommentaare koodi selgitamiseks, et juhul kui

antud kood peaks kunagi sattuma teise arendaja kätte, on tal võimalik olemasolevate kommentaaride abil koodist aru saada. Kasuta tagajärgede hoiatamiseks ehk hoiata mis võib juhtuda, kui näiteks eemaldad selle rea koodist.

Koodi Githubi commitimisel ning pushimisel tuli commitide selgitusse märkida antud commiti eesmärk ja tehtud töö, et hilisemalt oleks võimalik probleemide korral üles leida problemaatiline commit.

4.2 Projekti teostamise põhjendus

Projekti teostamisel lähtusid meeskonna liikmed erinevatest õpetustest veebis ning samuti kasutati raamatute abi. Meeskonna jaoks kohased raamatud saadi O'Reilly andmebaasist ning erinevad õpetused leiti internetist (ntks. stackoverflow). Raamatute läbi töötamise mõte oli saada kinnitust, et valitud metoodika, tööriistad ning arenduspoliitika on sobilikud.

Arhitektuuri loomisel kasutati erinevaid andmefaile, milles olid andmed jaotatud relatsioonilise andmemudeli kohaselt, tänu millele sai erinevaid andmefaile vajaduse korral kergesti ühendada.

Front-end koodi kirjutamisel lähtuti objektorienteeritud paradigmast. Klassid defineerivad objektide struktuuri ja elemendid ning rakenduse jooksutamisel luuakse vajalikud objektid. PHP on serveripoolne skriptikeel, mida kasutatakse peamiselt veebiarenduseks, kuid mida kasutatakse ka üldotstarbelise programmeerimiskeelena. Objektorienteeritud programmeerimine (PHP OOP) on PHP5-le lisatud teatud tüüpi programmeerimiskeele põhimõte, mis aitab luua keerukaid ja korduvkasutatavaid veebirakendusi [8]. Mõned objektorienteeritud mõisted php's on näiteks objekt, mis on klassi määratletud andmestruktuuri üksikeksemplar. Määratakse klassi üks kord ja seejärel saab teha palju sellele kuuluvaid objekte. Objektid on tuntud ka kui eksemplarid.

PHP (rekursiivne akronüüm PHP-le: Hypertext Preprocessor) on laialt kasutatav avatud lähtekoodiga üldotstarbeline skriptikeel, mis sobib eriti veebiarenduseks ja mida saab HTML-i sisse põimida. PHP-d eristab näiteks kliendipoolsest JavaScriptist see, et kood käivitatakse serveris, genereerides HTML, mis seejärel kliendile saadetakse. Klient saab selle skripti käitamise tulemused, kuid ei tea, mis on selle aluseks olev kood. Võib isegi oma veebiserveri konfigurida

nii, et see töötaks kõiki HTML-faile PHP-ga, ja siis pole tegelikult mingit võimalust, et kasutajad saaksid öelda, mis teil on.

Kuigi PHP erineb suuresti näiteks pythonist ning ei ole algselt mõeldud, kui OOP keelena, siis tänu uuendustele on jõudnud sobilike OOP keelte hulka ka PHP. Seda just järgnevate põhjuste pärast: Objektid on laiendatavad, objektid on asendatavad, objekte on võimalik testida, objektid on hooldatavad ning objektid tekitavad tabatavaid (ja taastatavaid) vigu.

Koodi kirjutamisel lähtuti Robert C. Martini raamatust "Clean Code" ning internetist leitud näpunäiteid jälgides. Clean Code raamatu põhjal rakendati erinevaid soovitusi, mis olid raamatust leitavad. Koodi kirjutamisel järgiti rangelt puhta koodi põhimõtteid. Toimus pidev koodi puhastamine vanast ebafunktsionaalsest koodist ning eemaldati ebavajalikud kommentaarid. Meetodite ja klasside nimetamisel kasutati tähenduslike nimesid, mis olid hästi arusaadavad ning hääldatavad, klasside nimetamisel kasutati nimisõnaseid ning meetodite nimetamisel tegusõnaseid. Meetodid ja funktsioonid olid fookuseeritud ainult ühele kindlale probleemile. Vältiti koodi kordusi, kus kood esines mitmes erinevas klassis/meetodis mitu korda. Klasside nimetused on UpperCamelCase ning meetodid, funktsioonid ja muutujad on lowerCamelCase formaadis. Tihtipeale unustati meeskonna siseselt eeltoodud reegleid järgida, kuid tänu Clean Code raamatule suudeti töö järje peale saada ning jooksvalt implementeerida antud reeglid.

Projekti vältel lähtuti RAD (Rapid application development) põhimõtetest. Kiire rakenduste väljatöötamine (RAD) on kiire tarkvaraarenduse meetodika, mis seab esikohale prototüübi kiire väljalaskmise ja kordamise. Erinevalt *waterfall* meetodist rõhutab RAD tarkvara ja kasutajate tagasiside kasutamist range planeerimise ja nõuete registreerimise asemel [9]. RAD eelisteks on: Suurem paindlikkus ja kohanemisvõime, kuna arendajad saavad arendusprotsessi käigus kiiresti kohandusi teha. Kiired kordused, mis vähendavad arendusaega ja kiirendavad edastamist. Koodi korduvkasutamise soodustamine, mis tähendab vähem käsitsi kodeerimist, vähem ruumi vigadele ja lühemat testiaega. Üllatusi on vähem, kuna erinevalt *Waterfall* meetodist sisaldab RAD integreerimisi tarkvaraarenduse alguses [9]. Tänu RAD põhimõtete järgmisele suudeti antud veebiplatvorm õigeaegselt valmis saada.

Projekti alguses puudus meeskonnal selge arusaam ja siht, kuidas antud rendi platvormi teha. Kuna puudus kogemus nii kasutusele võetud programmeerimiskeeles ning üleüldine kogemus

veebirakenduste tegemisel, lähtuti suuresti internetist leitud õpetustest ning raamatutest. Kindlasti ei aidanud hetkel maailmas valitsev kriisiolukord, mis oli põhjustatud Covid-19 poolt. Sellest tulenevalt ei olnud meeskonnal võimalik füüsiliselt kohtuda ning koos tööd teha, mis kohe kindlasti raskendas noore meeskonna tööd. Lahenduseks võeti iganädalased (tihtipeale igahommikused) kohtumised, kus kuulati üksteise muresid ning pakuti abi erinevate probleemide osas. Lisaks oli tegemist hea viisiga, kuidas kõik informeeritud hoida. Siin kohal tuli suuresti kasuks SCRUM raamistiku järgmine

4.3 Logid

Mall projekt

Extensions	Count	Size SUM	Size Min	Size Max	Size AVG	Lines	Lines Min	Lines Max	Lines AVG	Lines CODE
config	1	914	914	914	914	23	23	23	23	23
css	5	1210793	4599	846619	242158	19920	6	10947	3984	17108
example	1	941	941	941	941	53	53	53	53	43
gitattributes	1	111	111	111	111	5	5	5	5	5
gitignore	15	490	9	181	32	53	1	16	3	50
gitkeep	17	0	0	0	0	0	0	0	0	0
htaccess	1	593	593	593	593	21	21	21	21	14
ico	1	0	0	0	0	0	0	0	0	0
js	6	413255	626	328883	68875	2490	1	2257	415	1921
json	11	980052	69	602219	89095	19826	1	14433	1802	19825
lock	2	574091	209529	364562	287045	16207	5996	10211	8103	15272
md	2	6918	2116	4802	3459	227	60	167	113	126
php	682	1535375	12	47387	2251	39705	1	670	58	25000
scss	5	10222	140	7111	2044	557	8	404	111	387
svg	1	2679	2679	2679	2679	1	1	1	1	1
twig	86	290191	155	18378	3374	6447	5	370	74	5253
txt	3	36074	24	35150	12024	698	2	674	232	572
vue	1	565	565	565	565	23	23	23	23	21
xml	1	1088	1088	1088	1088	32	32	32	32	31
									Kokku	85652

Joonis 35. Mallprojekti kõik koodiredad

Lõpp-projekt

Extensions	Count	Size SUM	Size Min	Size Max	Size AVG	Lines	Lines MIN	Lines Max	Lines AVG	Lines CODE
config	1	937	937	937	937	23	23	23	23	23
css	5	1257452	4888	855001	251490	20739	6	11765	4147	17755
env	1	1680	1680	1680	1680	69	69	69	69	55
example	1	1518	1518	1518	1518	69	69	69	69	55
gitignore	16	383	10	111	23	42	1	8	2	42
gitkeep	17	0	0	0	0	0	0	0	0	0
htaccess	1	614	614	614	614	21	21	21	21	14
ico	1	103303	103303	103303	103303	77	77	77	77	75
js	6	415839	646	328883	69306	2490	1	2257	415	1922
json	10	384862	72	274587	38486	5448	1	5113	544	5447
lock	1	548818	548818	548818	548818	14583	14583	14583	14583	14583
log	1	28546799	28546799	28546799	28546799	158681	158681	158681	158681	11750
md	1	4776	4776	4776	4776	164	164	164	164	108
phar	1	2205196	2205196	2205196	2205196	102755	102755	102755	102755	63982
php	947	4743327	13	60193	5008	106350	1	1733	112	77388
scss	5	10870	148	7608	2174	562	8	409	112	392
svg	5	34486	2679	15305	6897	5	1	1	1	5
tmp	13	396028	12031	53771	30463	7753	235	1041	596	6056
twig	89	368614	203	42857	4141	7775	5	831	87	6461
txt	2	940	26	914	470	24	2	22	12	19
vue	1	588	588	588	588	23	23	23	23	21
webmanifest	2	964	471	493	482	38	19	19	19	38
xml	1	1120	1120	1120	1120	32	32	32	32	31
yml	1	216	216	216	216	5	5	5	5	5
									Kokku	206227
									koodiridade vahe	
									120575	

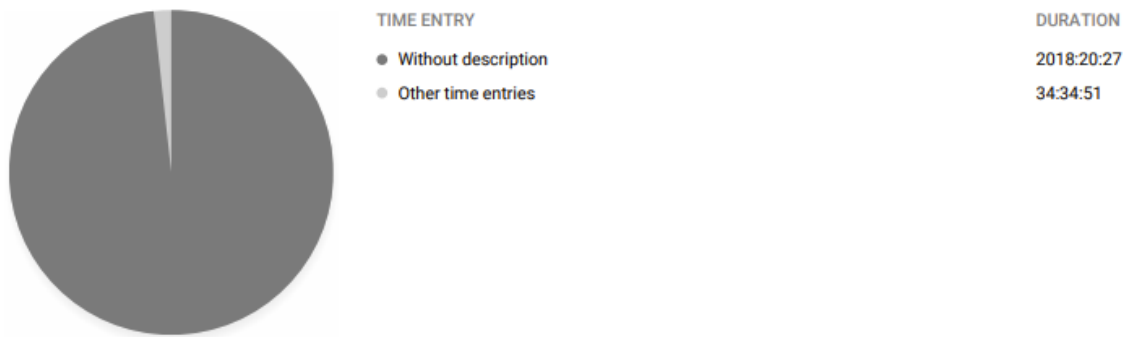
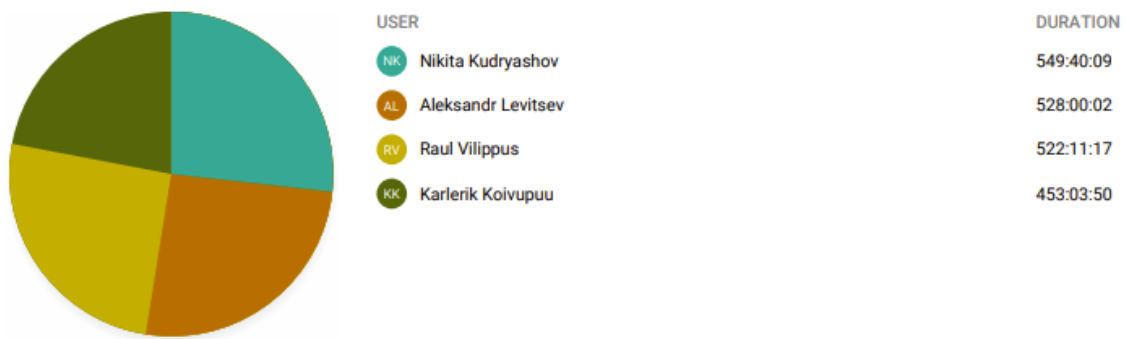
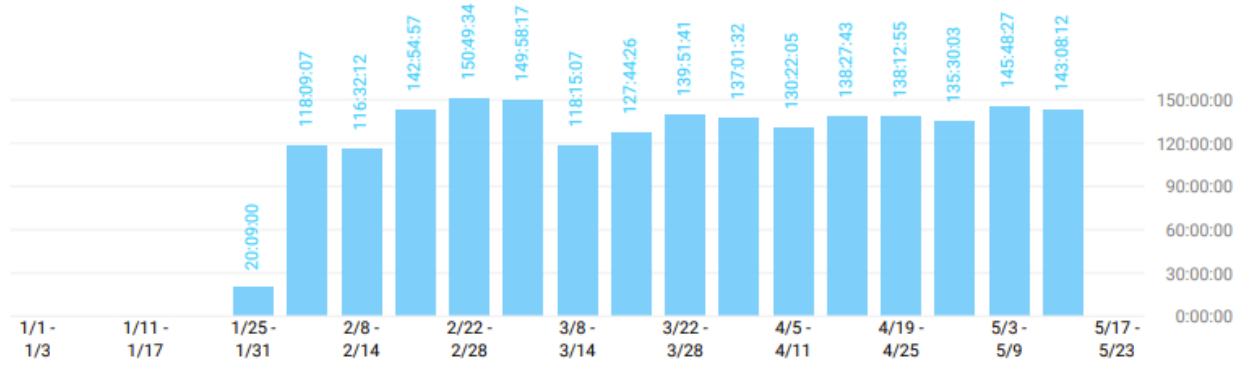
Joonis 36. Lõpp-projekti kõik koodiread

Summary Report



01/01/2021 – 05/18/2021

TOTAL HOURS: 2052:55:18



Joonis 37. Meeskonnaliikmete panustatud aeg

4.4 Hinnang projekti teostamise protsessi kohta

4.4.1 Projekti juhtimise ja teostamise protsess

Lõputöö arendustöö toimus 16 nädala jooksul. Lõputöö tegemisega alustati 25. Jaanuaril 2021 ning lõpetati 30. Mail 2021. Ette antud ajaperioodi vältel tehti tööd kolm kuni viis päeva nädalas. Päevad jagunesid varieeruvalt, kuid alati toimus töö vähemalt kolmel päeval nädalas. Kõik tööpäevad kestsid ligikaudu 6-10 tundi, sellest tulenevalt töötasid meeskonnaliikmed nädalas nelja peale kokku ligikaudu 135 tundi. 16 nädala peale teeb see kokku ligikaudu 2053 tundi tööd. Meeskonna siseselt toimusid esimeste nädalate jooksul iga hommikused kohtumised Teamsi vahendusel, kus arutati liikmetel ees seisvaid raskusi ning edusid. Arendustöö lõpu poole toimusid kohtumised vähemalt kord nädalas, et kõik meeskonnaliikmed oleksid informeeritud arendusprotsessist. Pidevalt arendati ka ennast individuaalsel tasandil. Pandi suurt rõhku enese arendamisele PHP programmeerimiskeeles ning sellega kaasnevate keelte arengus.

4.4.2 Hinnang projektile

Projekti vältel oli takistusi suhteliselt vähe. Kindlasti võib välja tuua mõned raskendavad asjaolud nagu näiteks Koroonaviirus. Koroonaviiruse tõttu ei olnud meeskonnaliikmed suutelised füüsiliselt koos töötama, mis algaja meeskonna puhul raskendab kohe kindlasti tööprotsessi. Oldi sunnitud kogu tööprotsessi korraldama virtuaalselt Teamsi vahendusel ning sellest tulenevalt oli perioode, kus meeskonnaliikmete motivatsioon sai kannatada. Algselt sujus arendustöö natuke aeglasemalt, kuna puudusid varasemad kokkupuuted antud programmeerimiskeelega ning sellega kaasnevate protsesside osas.

Üleüldiselt saab öelda, et kogu projekt sujus edukalt ning meeskonna liikmetel probleeme üksteisega ei esinenud. Tagasiside meeskonna liikmete poolt üksteisele oli üldiselt positiivne ning tööd tehti kohusetundlikult. Arendustööga küll ei jõutud sinna maani, mis esialgselt oli soovitud ajalise nappuse tõttu, kuid arendustööga plaanitakse jätkata ka peale Lõputöö kaitsmist. Meeskonnaliikmete omavaheline suhtlus oli hea ning probleemide korral pakuti üksteisele abi ning lahendati probleemid ühtselt.

4.4.3 Hinnang protsessile

Projekti ülikooli poolsed juhendajad olid Tallinna Tehnikaülikooli Karl-Erik Karu ja Gunnar Piho. Kogu projekti vältel oli kommunikatsioon meeskonna liikmete ja ülikooli poolse juhendaja vahel väga hea. Probleemide või küsimuste korral oli juhendaja tagasiside alati konstruktiivne, usaldusväärne ning kiire. Kordagi ei tekkinud meeskonnaliikmetel tunnet, et juhendaja poole ei ole võimalik pöörduda.

Töökorraldus oli paindlik ja arusaadav. Kuna tegemist oli Lõputööga ja viimase semestriga, siis oli projekti jaoks aega rohkem kui tavaliselt. Ülikooli poolne tunniplaan oli hästi paika pandud, mis jättis palju ruumi arendutööks projekti osas. Kõikidel meeskonnaliikmetel olid olemas võimalused teha tööd kodust, mis aitas eriti kaasa arendustööle hetkeses globaalses kriisiolukorras. Kuna tunniplaan Ülikoolis oli hästi läbi mõeldud, andis see võimaluse meeskonnal projekti kallal võimalikult paindlikult ning efektiivselt töötada, puudus stress, kus meeskonnaliikmed oleks tundnud, et projekti kallal töötamise ja kooli poolsete ainete kallal samaegselt töötamine ei ole võimalik.

Meeskonnaliikmete omavaheline läbisaamine oli väga positiivne kogu projekti vältel. Kuna meeskonna liikmed on ka varasemalt koos projekte läbi viinud, olid kõik liikmed üksteisega kohanenud ning tuttavad ning see välistas projekti jooksul võimalikud probleemid. Juba projekti alguses olid kõigi jaoks selged liikmete tugevused ja nõrkused ning ülesandeid määrati vastavalt sellele, et välistada võimalike probleeme tulevikus. Meeskonnaliikmetel oli võimalik valida ülesandeid, mida nad tundsid, et on võimelised lahendama.

4.5 Meeskondlik hinnang

Skaalal “-2” kuni “+2” hinnati meeskondlikult kõikide meeskonnaliikmete panust ja osalemist tööprotsessis hindegaga “0” ehk “panustas samaväärselt”. Etteheiteid üksteise suunal ei esinenud.

5 Kokkuvõte

Lõputööna teostati mugav ja kompaktne veebiplatvorm Rentif, kus inimestel on võimalik rentida esemeid inimeselt inimesele, tehes seda turvaliselt. Projekti meeskonda kuulusid neli äriinfotehnoloogia kolmanda kursuse tudengit Karl-Erik Kõivupuu, Raul Vilippus, Nikita Kudryashov ning Aleksandr Levitsev. Vaadates erinevaid veebis asetsevaid rendi teemalisi platvorme ei jäänud silma ükski platvorm, kus inimestel oleks võimalik rentida ükskõik milliseid esemeid suurima olemasoleva valikuga.

Projekti arendamiseks kasutati juba olemasolevat malli, mille peale hakati soovitud rakendust kirjutama. Projekti malli ja lõppversiooni koodiridade vaheks osutus 120000 koodirida. Projekti arendati PHP programmeerimiskeeles ning raamistikuks oli Laravel. Meeskond töötas projekti kallal jaanuarist juuni alguseni .

Kasutatud kirjandus

- [1] „Vikipeedia,“ 16 Mai 2021. [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/%C3%9Cletarbimine>.
- [2] „Riptutorial,“ 13 Mai 2021. [Võrgumaterjal]. Available: <https://riptutorial.com/twig>.
- [3] „Laravel Docs,“ 10 Mai 2021. [Võrgumaterjal]. Available: <https://laravel.com/docs/8.x/blade>.
- [4] „Kissflow,“ 17 Mai 2021. [Võrgumaterjal]. Available: <https://kissflow.com/low-code/rad/rapid-application-development/>.
- [5] „123Reg,“ 7 Mai 2021. [Võrgumaterjal]. Available: <https://www.123-reg.co.uk/support/servers/what-is-mysql-and-why-do-i-need-it/>.
- [6] „Raygun,“ 4 Mai 2021. [Võrgumaterjal]. Available: <https://raygun.com/blog/top-php-frameworks/>.
- [7] „Vikipeedia,“ 23 Aprill 2021. [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/Bootstrap>.
- [8] „Greycampus,“ 22 Aprill 2021. [Võrgumaterjal]. Available: <https://www.greycampus.com/codelabs/php/OOPS>.
- [9] „Capterra,“ 6 Detsember 2019. [Võrgumaterjal]. Available: <https://blog.capterra.com/what-is-rapid-application-development/>.

Lisa 1 - Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Raul Vilippus, Karl-Erik Kõivupuu, Nikita Kudryashov, Aleksandr Levitsev

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

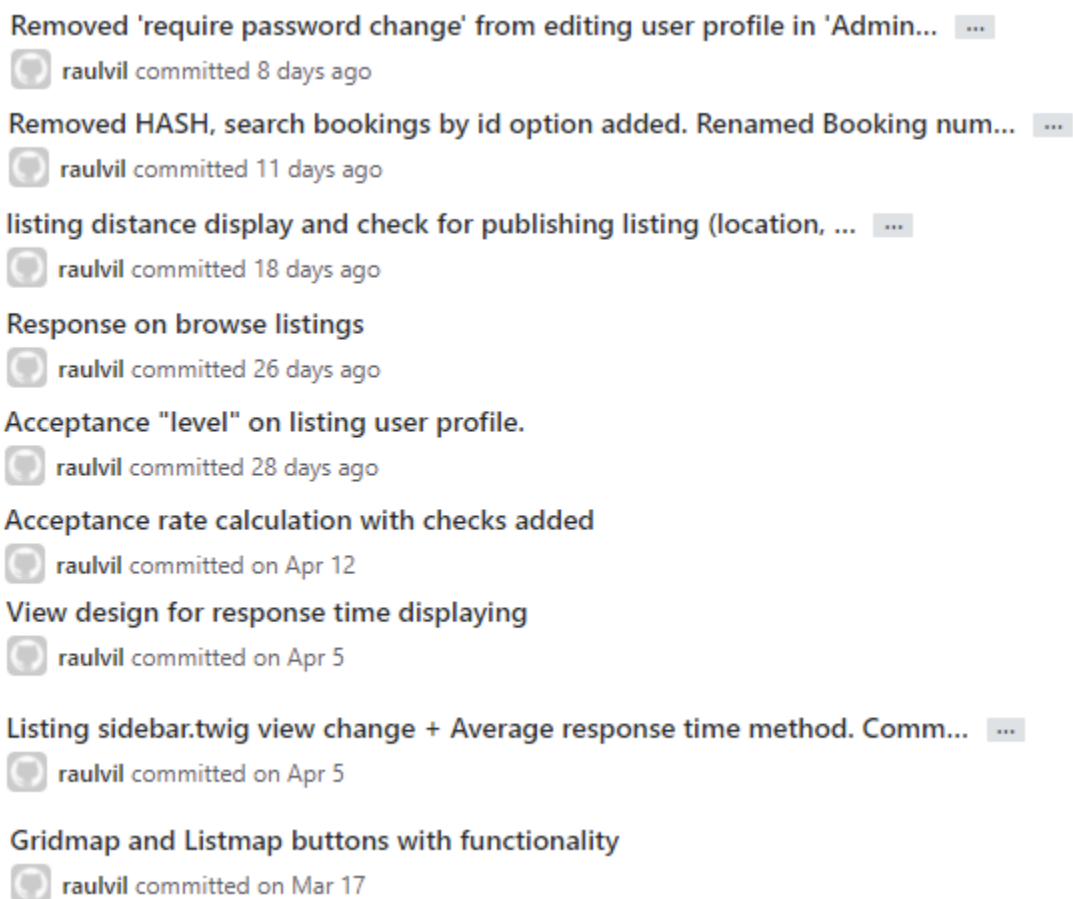
18.05.2021

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

Lisa 2 – Enda panuse kirjeldus ja analüüs – Raul Vilippus

Järgmisestest lõikudes kirjutan enda panusest ning eneseanalüüsist tehtud meeskondlikus projektis. Tehtud commit'ide autoriks ei pruugi olla ainult mina, kuna kasutasime tööprotsessi kiirendamiseks paarisprogrammeerimist, mistõttu on ka võimalik leida teiste panuseid minu commitidest ning minu panust teiste commit'idest.

Lisan mõned tähtsamad commit'id mis projekti käigus tehtud sai. Commit'ide logi on võetud GitHub'ist.



Joonis 38. Eneseanalüüs, GitHub commit'id

Kõikidel meeskonnaliikmetel oli võimalus valida oma oskuste ja võimekuste põhjal Issue'sid.

Mina valisin endale tööd, mida ma teadsin, et olen võimaline edukalt sooritama. Aegajalt tuli küsida meeskonna abi, kui ideed otsa said. Ühed kõige olulisemad tegevused olid kasutaja profiilile lisainformatsiooni andmete loomine ning seejärel nende kuvamine. Meie eesmärgiks on klientidele anda võimalus näha, kui kiiresti vastab kuulutuse postitaja klientidele ning kui tihti võtab postitaja broneeringu vastu. Selleks tuli andmebaasis olevate andmetega teha arvutused ning kuvada need klientidele. Üheks teiseks oluliseimaks tööks oli toodete vaatesse funktsiooni lisamine. Algselt oli kasutajal võimalik vaadata tooteid kas *grid*, *list* või *map* vaates. Minu lisatud funktsioon lubas kasutajal avada *map*’i koos *grid*’i või *list*’iga, et anda kasutajale parem ülevaade kuulutustest ning toodete asukohast.

Esialgu kõige suuremaks väljakutseks sai näidisprojekti töölesaamine. Nimelt oli tegemist suhteliselt vana projektiga (projekti oli viimati uuendatud 2019. aastal) ning kõik vajalikud uuendused olid tegemata, mis põhjustasid erinevaid turvariske. Esimesed kolm nädalat läks aega, et hakata vaikselt arusaama kuidas miski töötab, mis kaustades klassid on jne. Järgmine väljakutse oli PHP. Nimelt pole ma varasemalt kokku puutunud PHP keelega ning teha lõputöö projekt võõras keeles oli üks suur eneseületus. Tuli palju iseseisvalt õppida ning abiks oli teiste meeskonnaliikmete tugi. Mida aeg edasi seda enesekindlamaks muutusin ja koodi kirjutamine muutus aina ladiusamaks. Samuti mainiks ka seda, et kuna tegemist oli iseseisva meeskondliku projektiga, polnud probleemide tekkimisel pöörduda kellegi poole tehniliste küsimuste korral.

Kõige lihtsamaks osutus distantsilt koodi arendamine. Kuna Covid-19 tõttu oli kool ja muud asutused suletud, olime sunnitud töötama kodustes tingimustes. Esialgu plaaniti võtta kontor, kus saaksid kõik kohapeal käia tööd tegemas, aga mida aeg edasi, seda vähem nägime me vajadust omada oma kontorit.

Kõige rohkem arenesin koodi kirjutamise, meeskonna juhtimise ning iseseisvalt koodi kirjutamise osas. Varasematel aastatel oleme projektitöodes suuremat rõhku pannud paarisprogrammeerimisele, kuna vajalikud teadmised olid puudlikud. Iseseisvalt tööd tehes oled sunnitud läbitöötama erinevaid materjale ning igakord kui probleemist jagu sain, tundus see kui suure võiduna.

Varasemate kogemuste ning teadmiste põhjal olin ma teadlik, et meeskonnaga töödates peab arvestama teiste oskuste ning võimekustega. Mistõttu sai tööjaotused tehtud varasemate aastate kogemuste põhjal.

Lisa 3 – Enda panuse kirjeldus ja analüüs – Karl-Erik Kõivupuu

Ilmselt ühtedeks tähtsaimateks ülesanneteks projekti vältel oli minu jaoks turvameetmete implementeerimine. Nimelt two-factor authentication implementeerimine, kus kasutaja oli sunnitud kinnitama registreerimisel oma e-maili ja telefoninumbri. Tegelesin ka SEO (search engine optimization), mis oli minu jaoks eriline väljakutse, kuna ei olnud varasemalt millestki sellisest midagi otseselt kuulnud. Suur panus sai veel näiteks antud avalehel asuva search bar'i tegemisel, kategooriate lisamisel, *dropdown search bar*'i implementeerimisel, inimese *input*'i põhiste *suggestion*'ite loomine otsingu kasutamisel, kus otsingu puhul olid ühendatud kaks andmebaasi (esemed ja kategooriad), et kasutaja otsingu puhul saaks koheselt näidata ka mis kategoorias antud toode asub. Need olid ühed suurimad ettevõtmised projekti vältel.

Suurimateks väljakutseteks projekti teostamise vältel oli minu jaoks kindlasti täiesti uues programmeerimiskeeles projekti teostamine (PHP). Puudus varasem kokkupuude arenduskeskkonnaga (PHPstorm), millega harjumine võttis omajagu aega. Veebi arendus oli kindlasti raskem, kui ma arvasin. Projekti vältel kasutatud programmid olid minu jaoks täiesti uued ning seetõttu olin sunnitud keskenduma, mitte ainult uue keele õppimisele vaid ka sellega kaasnevate programmide õppimisele. Kuigi mida aeg edasi läks ning mida rohkem PHP-ga tutvusin, tundsin, et tegemist on väga hea keelega, mille selgeks õppimine võttis tunduvalt vähem aega, kui näiteks JavaScripti või Pythoni puhul. Projekti arenduskäik oli kindlasti minu jaoks üks meeldivamaid kogemusi. Kogu protsess oli väga lihtne, andmebaasi haldamisest, kuni prototüübi testimiseni veebi keskkonnas. Rohkem oleks võinud ilmselt keskenduda PHP õppimisele, kuigi seda sai piisavalt tehtud, ei tunne endiselt, et oleksin piisavalt pädev PHP-ga toimetamisel. Ettevalmistuse koha pealt oli kindlasti abi eelnevatest projektidest koolis, kus õpetati meile näiteks meeskonnatöö olulisust ja clean code'i. Üleüldiselt julgeks öelda, et varasemad õpingud valmistasid mind mingil määral antud projekti jaoks ette. Kindlasti tuli abiks ka varasemalt läbitud koodi kirjutamise kursused internetis.

Lisa 4 – Enda panuse kirjeldus ja analüüs – Nikita Kudryashov

Rentif meeskonnaprojekti raames olid suurimateks väljakutseteks tõenäoliselt uus programmeerimis keel, uued tehnoloogiad ja keskkond, kus toimus arendusprotsess (PHPStorm). Meil ei olnud ettevõtet ega mentoreid, kes oleksid meie tööprotsessi organiseerinud või arendamise käigus näpuga järele käinud ja aidanud - pidime kõikide väljakutsetega ise hakkama saama. Tänu eelnevatele kogemustele ei tekkinud meil sellega raskusi, tööprotsess oli algusest peale hästi organiseeritud ning oli seatud kindel sihtmärk, kuhu projekti raames tahame jõuda.

Projekti tegi keeruliseks asjaolu, et varasemalt ei olnud taoliste lahendustega kokku puutunud, pidime õppima kuidas luua turvalist süsteemi, kuidas teha veebileht nähtavaks (veebimajutus, SEO) ning kuidas kogu keskkond peab funktsioneerima. Kuna tegemist on isealgatatud projektiga, oli motivatsiooni rohkem kui tavaliselt ning töö sujus väga produktiivselt. Eelkõige arvan, et projektiga ühele järjele jõuda, aitasid suuremal määral varasemad programmeerimis oskused, eelnev töökogemus ja ülikoolis omandatud teadmised. Kõige arendavam oli minu jaoks iseõppimine, kus leidsime lahendusi kasutusjuhtumitele, millega pole varasemalt keegi kokkupuutunud. Läbi selle protsessi õppis väga palju uut ja sai paremini aru, mida tuleks muuta, et platvormi oleks mugav ja lihtne kasutada. Samuti aitas see saada parema ettekujutluse sellest, kuidas meie rakendus peaks lõppude lõpuks töötama ja väljanägema.

Minu peamised eesmärgid olid organiseerida tööd, planeerida ülesandeid, luua kasutusjuhtumeid ja aidata meeskonnakaaslasi programmeerimise kohapealt. Olen tegelenud programmeerimisega juba pikemat aega enne ülikooli sisseastumist, seega arvan, et olen piisavalt kompetentne teisi nõustama ja suundama. Enamus tööst toimus paarisprogrammeerimisel, et kõik meeskonnaliikmed areneksid projekti vältel samal kiirusel.

Enesehinnanguna tunnen kindlasti, et omandasin uusi ja kasulikke teadmisi. Omandasin väga häid organiseerimis- ja juhtimisoskusi. Õppisin probleemidega paremini toime tulema ja vaatamata kõigele olen enda arvates kõigi väljakutsetega hästi hakkama saanud.

Lisa 5 – Enda panuse kirjeldus ja analüüs – Aleksander Levitsev

Projekti rakendamise käigus õppisin palju uusi tehnoloogiaid, mida ma varem ei tundnud, ja sain kasulikke kogemusi programmeerimises. Sealhulgas JavaScript, Vue.js, AJAX ning Twig. Kõige rohkem pidin ennast arendama PHP-s, milles on peamiselt meie projekt kirjutatud.

Esimene ühine väljakutse oli vana projekti pakettide värskendamine, kuna projekti värskendati viimati 2019. aastal. Selle ignoreerimine võib põhjustada probleeme veebilehe turvalisuse ja toimivusega. See protsess võttis kaua aega, sest pakettide konflikte oli palju, millega seoses pidime otsima alternatiive ja tegema koodis muudatusi. Tänu sellele saime projekti ülesehitusega kiiresti tuttavaks ja uute muudatuste lisamine projekti oli lihtsam.

Kõige keerulisem ülesanne oli broneerimiskalendri installimine ja programmeerimine, mis on väga oluline element, kuna sellest sõltub kliendi valiku mugavus. Kalendri koostamisel juhindusime asjaolust, et ka inimene, kes arvutit ei kasuta, saab kuupäevi valida. Kalender ei võimalda kliendil broneeritud kuupäevi valida ja hüppab üle ka kuupäevadest, mil klient ei saa toodet rentida / tagastada, ja valib automaatselt esimese saadaoleva kuupäeva. Järgmine suur muudatus minu poolt oli hinnasüsteemi lisamine, mis laadib AJAXi kaudu kasutajale soovitatud hinnad, mida saab soovi korral muuta. Lisatud on ka üürileandja võime teatud päevad keelata ja ajaliselt piirata. Töötasin võrdselt nii loogika kui ka disaini kallal. Ning ka palju muid koodiparandusi.

Võrreldes algse malliga on meie projekt palju paremaks muutunud. Mõne lahenduse kavandamisel küsisin tihti oma sõpradelt ja tuttavatest, kas seda lahendust on mugav kasutada, mõnikord pidin selle isegi ümber tegema, sest olen veendunud, et see mis on mugav minule – ei ole sama mugav kõigile teistele. Kuna me viime selle projekti ise läbi, mitte kellegi teise jaoks, oli meil täielik tegevusvabadus ja me ise seadsime tegevuse prioriteedid. Sellega seoses oli suurenenud motivatsioon töötada ja tuleb ka märkida, et protsess oli hästi korraldatud.

Kahjuks koroonaviiruse piirangute tõttu ei olnud võimalik füüsiliselt kohtuda ja kogu töövoog toimus veebis. Kuid me ei andnud alla ja hoidsime ühendust ning tegime tööd peaaegu iga päev. Võin kindlalt öelda, et selle projekti elluviimisel omandasin palju kasulikke oskusi ja kui võrrelda neid juba õpitud praktikaga, siis nüüd oli palju rohkem õpitud. Hakkasin probleemide lahendamisel kannatlikumaks, sest mõnikord ei õnnestunud probleeme lahendada nädala või kauemgi. Isegi pärast lõputöö valmimist jätkame projektiga, sest näeme selles perspektiivi.