

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Pavel Botšarov 231114IAIB

Suurlao andmetöötlus EBM Group AS näitel

Bakalaureusetöö

Juhendaja: Vladimir Viies

Dotsent

Tallinn 2023

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: Pavel Botšarov

24.04.2023

Annotatsioon

Baasil EBM Group AS sisestruktuuri mudeli alusel seati sisse ülesanded infosüsteemi rakendamiseks. Selle lõputöö eesmärk oli luua veebipõhine PHP ja MySQL alusel süsteem. Välja toodud süsteem kasutatakse EBM Group ASi tehase õli arvestuse ja kontrolli jaoks.

Süsteem loodi tehase lepingutes märgitud erinevatesse pakenditesse valatava taimeõli koguse arvestamiseks. Loodud süsteemi saavad kasutada mitte ainult süsteemi tellijad, vaid ka kliendid ehk teised süsteemi kasutajatena registreeritud ettevõtted.

Töös kirjeldatakse kasutatavat andmebaasi, andmete töötlemise meetodit ja algorütmide ning edasist esitamist süsteemis. Süsteemi saab lisada kasutajaid ja määrata neile rolle. Süsteemis saab sisestada uusi kasutajaid ja määrata nende rollid, muuta lepinguid, genereerida aruandeid ja teha tellimusi.

Kogu MySQL andmebaas koosneb seitsmest tabelist. Lepingu ja tellimuste muutuste ajalugu salvestatakse eraldi tabelisse. Skriptipõhine PHP-serveripoolne keel oli kasutatud ühenduse loomiseks, päringute täitmiseks ja teabe töötlemiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 55 leheküljel, 6 peatükki, 15 joonist, 10 tabelit.

Abstract

Data Processing in a Large-Scale Warehouse: A Case Study of EBM Group AS

Office work of contemporary enterprises is impossible without Internet technologies. A system that adds, changes, and organizes data for optimized organization of office work according to the requirements using by the company is needed. Modern solutions should be introduced in order to automate the storage of the data and facilitate the account of all transactions taking place at the EBM Group AS enterprise. During the previous years, all transactions, orders and contracts in the factory were recorded in Excel or on paper, which caused errors.

The purpose of the present thesis is to create a web system based on the PHP language and a free MySQL database for the administration and control of vegetable oil accounting for the EBM Group AS plant. We set the tasks for the implemented information system based on the model of the internal structure of the EBM Group AS plant.

Chapter 1 considers the possibility of structuring information in the MySQL database for organizing office work in an enterprise. The process of connecting, querying and processing data using the PHP scripting language to process data and display information on the screen is described.

Chapter 2 presents the model and structure of the organization of the internal system. This model describes the goals which were set for creating the system. This includes the creation and editing of contracts, writing off and accounting for the amount of vegetable oil in certain packages, creating orders from customers in other countries, and recording all actions that occur with contracts in history. The roles (administrator, moderator, user, and viewer) that are assigned to each user in the system are represented. Depending on the role, the user can create and delete contracts, place orders and view the history of changes. Scenarios and conditions for performing any operation are outlined.

Chapter 3 depicts the model and structure of the database of seven MySQL tables in which all information is stored. The conditions and required fields are written for each table. Using the described PHP functions, the examples of connecting to a database, querying data and distributing them into an array variable for further processing are entered. The application describes four models of using the system, depending on the assigned role of the user. These models specify situations and conditions in which a user can create and modify a contract, make orders and confirm them.

The thesis is written in the Estonian language. It contains 55 pages of the text, which is divided into six chapters. Also, it includes fifteen figures and ten tables.

Lühendite ja mõistete sõnastik

Apache	Vaba lähtekoodiga veebiserveri tarkvara.
MySQL	Tasuta ja avatud lähtekoodiga andmebaasisüsteem.
PHP	Skriptimiskeel, veebilehtede arendamiseks ja programmeerimiseks.
LAMP	"Linux, Apache, MySQL ja PHP"
XAMPP	Lokaalserver (Apache, MySQL, PHP)

Sisukord

1	Sissejuhatus.....	12
2	Andmebaas ja nende kasutamine	13
2.1	Relatsiooniline andmebaasisüsteem MySQL.....	13
2.2	PHP keel MySQL andmebaaside töötlemiseks.....	14
2.3	PHP ja MySQLi suhtlus	15
3	Terviksüsteemi EBM Grupp AS üldvaade	17
3.1	Organisatsiooni eesmärgid	17
3.1.1	Infosüsteemi eesmärgid	17
3.1.2	Põhiobjektid	17
3.1.3	Lausendid.....	18
3.2	Põhiprotsessid.....	18
3.2.1	Tegutsejad.....	19
3.2.2	Asukohad	19
3.3	Terviksüsteemi tükeldus.....	19
3.3.1	Pädevusalad.....	19
3.3.2	Funktsionaalsed allsüsteemid.....	19
3.3.3	Registrid.....	20
3.4	Tellimuse funktsionaalne allsüsteem	20
3.4.1	Eesmärgid	20
3.4.2	Allsüsteemi kasutavad pädevusalad.....	20
3.5	Allsüsteemi poolt kasutavad registrid	20
3.6	Allsüsteemi põhiprotsessi tegevusdiagramm	21
3.7	Allsüsteemi kasutusjuhtude eskiismudel.....	22
3.8	Täiendavad nõudmised.....	24

3.9	Tellimuste register.....	24
3.9.1	Eesmärgid	24
3.9.2	Registrit kasutavad pädevusalad	24
3.9.3	Registrit teenindavad funktsionaalsed allsüsteemid	25
3.9.4	Infovajadused	25
3.9.5	Registri kontseptuaalne eskiismudel.....	25
4	EBM Grupp AS-le pakutav süsteemi tehniline arhitektuur	26
4.1	Tellimuse/mahakandmise funktsionaalne allsüsteem	26
4.1.1	Kasutusjuhtude mudel.....	26
4.2	Tellimuse/mahakandmise register	29
4.2.1	Kontseptuaalne andmemudel	29
4.2.1.1	Olemi-suhte diagramm.....	29
4.2.1.2	Olemitüüpide definitsioonid.....	29
4.2.1.3	Atribuutide definitsioonid	30
4.2.2	Andmebaasioperatsioonide lepingud	31
4.2.3	Registri põhiobjekti seisundidiagramm	34
4.3	CRUD matriks	35
5	EBM Grupp AS pakutud süsteemi loogiline disain.....	36
5.1	Tellimuse ja makandmise funktsionaalne allsüsteem	36
5.1.1	Realiseeritavad töökohad	36
5.1.2	Reaalsete kasutusjuhtude kirjeldused	36
5.2	Laiendatud tellimuste registri loogiline disain	39
5.2.1	Loogiline andmebaasi diagramm.....	39
5.2.2	Tabelite detailsed kirjeldused	39
6	Süsteemi tehniline arhitektuur ja realisatsioon	43
6.1	Esmane prototüüp.....	43
6.1.1	Nõudmised tarkvarale ja ristvarale	43

6.1.2	Programmi uus versioon.	43
6.2	Töötav süsteem.....	43
6.2.1	Andmebaasi ja rakenduse server.....	44
6.2.2	Halduri, tellijate töökohad	44
6.3	Realisatsioon MySQLis ja SQL laused	44
6.3.1	Andmebaasi loomine	44
6.3.2	Tabelid	44
6.3.3	Arvujada generaatorid ja unikaalseid väärtusi.....	47
6.3.4	Täiendavate testandmete lisamine	47
6.4	Andmetöötlus PHP abil.....	48
6.4.1	Funktsioon mysqli connect	48
6.4.2	Funktsioon mysqli query and fetch array.....	49
	Kokkuvõtte.....	50
	Kasutatud kirjandus	51
	Lisa 1 - "Kasutusjuhtude kirjeldused"	53
	Kasutusjuht 1	53
	Kasutusjuht 2	53
	Kasutusjuht 3	54
	Kasutusjuht 4	55

Jooniste loetelu

Joonis 1. Klient, veebiserver ja PHP suhtlus.	14
Joonis 2. Klient, PHP ja MySQL suhtlus.....	15
Joonis 3. Tellimuse allsüsteemi põhiprotsessi tegevusdiagramm.....	21
Joonis 4. Tellimuse funktsionaalse allsüsteemi kasutusjuhtude eskiismudel.	22
Joonis 5. Registri kontseptuaalne eskiismudel.....	25
Joonis 6. Olemi-suhte diagramm	29
Joonis 7. Tellimuse seisundidiagramm.....	34
Joonis 8. Põhitabel lepingutega.....	36
Joonis 9. Uue lepingu koostamise veebiaken.	36
Joonis 10. Tellimuse koostamine või õli mahakandmise veebiaken	37
Joonis 11. Tellimuse muutmine veebiaken.....	37
Joonis 12. Tellimuste vaatamiseks veebiaken halduri jaoks.....	38
Joonis 13. Tellimuste vaatamiseks veebiaken kliendi jaoks.....	38
Joonis 14. Tellimuse registri loogilise disaini andmebaasi diagramm.	39
Joonis 15. Programmi infosüsteemi tehniline arhitektuur	43

Tabelite loetelu

Tabel 1. Nõudmise kirjeldused.	24
Tabel 2. Olemitüüpide definitsioonid	29
Tabel 3. Atribuutide definitsioonid.....	30
Tabel 4. Tabeli nimi: user	39
Tabel 5. Tabeli nimi: contract.....	40
Tabel 6. Tabeli nimi: virtual_number	41
Tabel 7. Tabeli nimi: type_of_package	41
Tabel 8. Tabeli nimi: contract_history.....	41
Tabel 9. Tabeli nimi: type_of_package_history	42
Tabel 10. Tabeli nimi: type_of_package_h	42

1 Sissejuhatus

Kaasaegsete ettevõtete asjaajamine pole võimalik ilma internetitehnoloogiata. Hea asjaajamise korraldamiseks on vaja süsteemi, mis lisab, muudab ja korraldab andmeid vastavalt ettevõtte nõudmistele.

EBM Group AS-i ettevõttes toimuvate tehingute haldamise automatiseerimiseks ja lihtsustamiseks on vajalikud kaasaegsed lahendused.

Varem kandsid tehased kõik tehingud, tellimused ja lepingud Exceli programmi või paberile. Kui kiired kirjed tehti paberil, siis mõned tellimuste andmed kadusid ja pidevalt tekkisid vead. Excelil pole piisavalt funktsionaalsust, mida ettevõtte vajab. Kõik see ei toiminud reaalselt ega vastanud kaasaegsetele asjaajamise meetoditele.

Valmislahenduste otsimine, mis rahuldaks seatud ülesandeid, ei andnud tulemusi ja otsustati luua oma süsteem.

Kõige tõhusam lahendus asjaajamiseks on veebisüsteemi loomine, mis pöördub andmebaasi poole, kus hoitakse ja töödeldakse kõiki tellimusi ja lepinguid.

Süsteemi loomiseks nullist vajas autor kolme kuu tööd, kuna programmil on piisavalt erinevaid toiminguid lepingute ja tellimuste osas. Esimene programmi versioon võeti vastu 2009. aasta septembris. Veebisüsteem loodi PHP skriptikeele ja tasuta MySQL andmebaasi põhjal.

Järk-järgult laienes süsteemi funktsionaalsus seoses programmeerimiskeele PHP evolutsiooniga ja EBM Group AS-i ettevõtte tekkivate uute vajaduste ning ülesannetega, kuna ettevõtte tegevusring laienes ja töömaht suurenes.

Loodud lõpp-produkt võimaldab kasutajatel süsteemi registreeruda, jälgida kõiki muudatusi lepingutes ning tellimustes, jälgida lepingute muutmise ajalugu ja hoida teavet andmebaasis, millele saab juurdepääsu Interneti kaudu.

2 Andmebaas ja nende kasutamine

Andmebaase kasutatakse info salvestamiseks ja välja kuvamiseks. Andmebaas on arvutisüsteemis salvestatud andmete organiseeritud kogum, millele on lihtne juurde pääseda, mida saab hallata ja värskendada. See koosneb tabelitest, mis on omavahel seotud ja esindavad erinevaid andme- või objektikomplekte. Andmebaase kasutavad tavaliselt ettevõtted, ülikoolid, haiglad ja muud organisatsioonid, mis nõuavad kiiret salvestusruumi ja kiiret juurdepääsu suurele andmemahule [1].

Andmebaase on mitut tüüpe, sealhulgas failipõhised, relatsioonilised, NoSQL- ja graafandmebaasid. NoSQL kasutab paindlikke andmemudeleid struktureerimata või poolstruktureeritud andmete jaoks. Graafandmebaasid keskenduvad seostele üksuste vahel, mis võib olla kasulik sotsiaalmeedias või soovitude mehhanismides [2]. Relatsioonilised andmebaasid kasutavad tabelite ja veergude salvestamiseks ja andmete korraldamiseks [3]. Relatsioonilisi andmebaase võib soovitada mis tahes teabe vajaduste rahuldamiseks, kus andmeelemendid on omavahel seotud ja neid tuleb juhtida vastavalt terviklikkuse eeskirjadele turvaliselt ja usaldusväärset, üks selline andmebaas on MySQL, mida kasutatakse allpool toodud programmis [4].

2.1 Relatsiooniline andmebaasisüsteem MySQL

MySQL on üks populaarsemaid avatud lähtekoodiga suur relatsiooniline andmebaasisüsteem, mis võimaldab kasutajatel tõhusalt andmeid hoida ja haldada. See on laialdaselt kasutusel veebisaitide, rakenduste ja tarkvaralahenduste taga ning on saadaval erinevate platvormide jaoks, sealhulgas Windows, Linux ja MacOS. MySQL toetab erinevaid programmeerimis- ja serverplatvorme ning pakub ulatuslike SQL (Structured Query Language) funktsioonide komplekti. Andmebaasi haldamiseks kasutatakse tavaliselt MySQL MariaDB, mis pakub kasutajasõbralikku graafilist kasutajaliidest erinevate SQL päringute koostamiseks ning andmete importimiseks ja eksportimiseks. See on usaldusväärne ja kiire ning seda on lihtne juurutada igasugusele IT-keskkonnale. Paljud maailma suurimad ja kiiremini kasvavad organisatsioonid, sealhulgas Facebook, Twitter, Booking.com ja Verizon, usaldavad MySQL-i, et säästa aega ja raha oma suurte veebisaitide, ärikriitiliste süsteemide ja pakk-tarkvara jaoks. [5] MySQL pole mitte ainult maailma populaarseim avatud lähtekoodiga andmebaas, vaid on

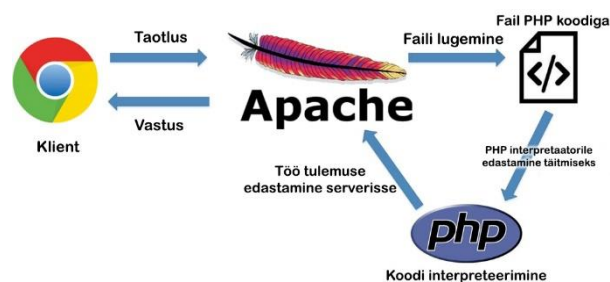
saanud ka eelistatud andmebaasiks uue põlvkonna rakenduste jaoks, mis on loodud LAMP (Linux, Apache, MySQL, PHP/Perl/Python) [6] virnas.

2.2 PHP keel MySQL andmebaaside töötlemiseks

PHP (inglisekeelne lühend sõnadest "Hypertext Preprocessor") on serveripõhine skriptikeel, mida kasutatakse peamiselt dünaamiliste veebilehtede ja rakenduste loomiseks. See töötab serveril ja genereerib HTML-koodi, mis saadetakse kliendi veebibrauserile. PHP-l on võimsad võimalused andmebaaside, failide, stringide ja massiividega töötamiseks ning mugavaks ja kiireks arendamiseks on ka palju teeke ja raamistikke. See on saadaval paigaldamiseks enamikul veebiserveritel ja toetatud enamikul operatsioonisüsteemidel [7].

See on väga paindlik keel, mis võimaldab kogunud kätes luua igasuguse keerukusega kõrge koormusega veebirakendusi. Facebook, VKontakte, Badoo, Wikipedia - kõik need gigandid kasutavad PHP-d [8]. PHP-is kirjutatud programm töö ajal ei kuvata standardset akent, mida oleme harjunud nägema. Enamikel juhtudel töötab see programm serveril "nähtamatult". See tähendab, et me ei saa seda töötamise ajal juhtida.

Fakt on see, et see programm peab vastu võtma kasutaja päringu, tegema mingeid toiminguid, andma tulemuse kasutajale ja seejärel lõpetama. Mida kiiremini see toimib, seda parem. PHP programmid erinevad selles osas "tavalistest" programmides, mis püsivad pidevalt mälus, ootavad kasutaja toiminguid ja reageerivad nendele reaajas.



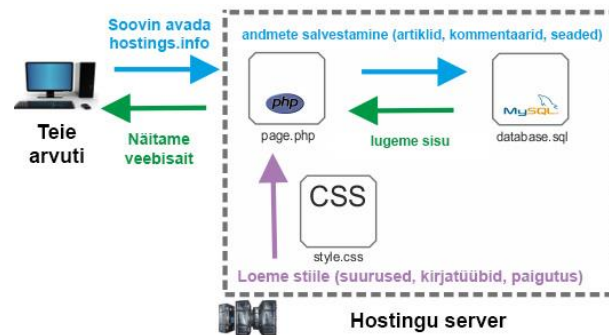
Joonis 1. Klient, veebiserver ja PHP suhtlus.

PHP ei tööta iseenesest. Kasutajaga töötamiseks peab see olema konfigureeritud koos veebiserveriga, näiteks Nginx või Apache. Lihtsamalt öeldes saadab kasutaja HTTP-protokolliga kaudu päringu veebiserverilt. Veebiserver määrab, mis tüüpi failile ligi pääseti ja kui tegemist on .php laiendiga failiga, siis veebiserver saadab päringu, et see PHP keeletõlgi poolt töödeldaks, skript loetakse, teisendatakse baitkoodiks, käivitatakse ja tagastatakse vastus

veebiserverile. Veebiserver saadab genereeritud tulemuse kasutajale. See tulemus on tavaliselt dünaamiliselt loodud HTML-leht [9].

2.3 PHP ja MySQLi suhtlus

Dünaamiliste saitide arendamiseks saab linkida MySQL ja PHP. Kõik, mida vajame, on loodud andmebaas [10].



Joonis 2. Klient, PHP ja MySQL suhtlus.

Näiteks loodud andmebaas phpMyAdmin kaudu. ([vt 6.3.1](#))

Teabe salvestamiseks tuleb luua ka tabelid. ([vt 6.3.2](#))

Vaikimisi toetab PHP MySQL-andmebaasiga tööd ning selle sisseehitatud funktsioonid võimaldavad selle andmebaasiga lihtsalt ja tõhusalt töötada. Konkreetselt saate teostada mis tahes päringud, andmeid lugeda ja kirjutada ning vigu töödelda. Olenemata sellest, kui keerulised või lihtsad on teie stsenaariumid, koosnevad need sisuliselt järgmistest sammudest (sammud MySQL-andmebaasiga töötamisel):

- Ühendage soovitud andmebaasiga, kasutades teadaolevaid andmebaasi nime, kasutajanime ja parooli.
- Veenduge, et ühendus oli edukas, vigu pole ja andmebaasi server on saadaval.
- Koostage õige SQL päring (näiteks andmete lugemiseks või kirjutamiseks).
- Saada päring ja veendu, et andmed on saadud.
- Töötle saadud andmed MySQL andmebaasist.
- Kasutage saadud andmeid oma eesmärkidel.

Ühenduse loomiseks peate MySQL-i andmetele juurdepääsemiseks seadistama ühenduse PHP-skriptis, kasutades PHP funktsioone [11]. ([vt 6.4.1](#))

Järgmiseks peate looma SQL-päringu, et hankida andmebaasist andmed ja valmistada need ette lehel kuvamiseks. Päringute saatmiseks ja andmete toomiseks saab kasutada selliseid käsked nagu `mysqli_query ()` ja `mysqli_fetch_array ()`. ([vt 6.4.2](#))

Seejärel saab massiivi vastuvõetud andmeid kasutada ja lehel kuvada [12].

3 Terviksüsteemi EBM Grupp AS üldvaade

EBM GRUPP AS ettevõtte asutati erainvestorite poolt 2000 aastal ja sellest sai Tallinna meresadama infrastruktuuri osa. Tootmisettevõtte ehitati 250 m kaugusele kaidest kasutades kaasaegseid tehnoloogilisi lahendusi. Edasiste ettevõtte arengutes kaasatakse samuti kõige uuemat tehnoloogiat ja tootmiseadmeid erinevate õlitüüpide töötlemiseks ja uute toodete loomiseks. Terminalil on 9 õlimahutit roostevaba terasest kogu mahutavusega 3500 tonni. Olemasolev meresadama infrastruktuur annab ettevõttele suurepärased võimalused erinevate toiduõlide vastuvõtmiseks ja ümberlaadimiseks. Samuti kasutatakse laadimisel spetsiaalseid tankereid, merekonteinereid ja raudteed, mille haruliin on ühendatav endiste ENSV Liidu maade raudtee liinidega. See tagab kauba tarne igasse regiooni ja turgudele. Kvaliteetse teenuse osutamiseks meie klientidele pakume nii uksest-ukseni teenust kui ka erinevate logistika ssegmentidena.

Olulisemaks ettevõtte eeliseks on töö kõikide toiduõli sortidega, kaasarvates haruldaste liikidega ning eksootiliste toiduõlidega, nagu näiteks viinamarjaseemneõli ja linaseemneõli. Paindlikus nii suurte kui ka väiksete tellimuste täitmisel.

3.1 Organisatsiooni eesmärgid

EBM Grupp AS eesmärkideks on taimeõlide ümberlaadimine läbi terminali, ladustamine ja villimine, mille maht on keskmiselt 100 tonni kuus. Nende toimingute käigus on vaja arvestada erinevat tüüpi õlide ja erinevat tüüpi pakendite arvuga, millesse õlid valatakse.

3.1.1 Infosüsteemi eesmärgid

- Saada ülevaade lepingus sisalduva üldise ja jääkõliga õli kogusest;
- Saada ülevaade õli ostu-müügi alg- ja lõpphinnast;
- Saada ülevaade klientidest erinevate riikide;
- Võimaldada tellimuste tegemist elektroonilisel kujul;
- Saada ülevaade tellimustest;
- Saada ülevaade mahakandmise ajaloost ja lepingu muudatustest.

3.1.2 Põhiobjektid

- Administraator;

- Direktor;
- Moderaator (e. Haldur);
- Kasutaja (e. Tellija);
- Vaatleja;
- Leping;
- Tellimus.

3.1.3 Lausendid

- Administraator on isik;
- Direktor on isik;
- Moderaator (e. lepingu haldur) on isik. Tehases on lepingu juhataja ja asetäitja;
- Kasutaja (e. tellija) on isik. Teises firmas, kes koostab tellimust;
- Vaatleja on isik. Teises firmas, kes vaatab lepingu seisukord;
- Administraator kinnitab kasutajate registreerimist;
- Administraator kontrollib programmi tööd;
- Direktor saab koostada lepinguid ja koguseid maha kanda;
- Moderaator kinnitab kasutaja registreerimist;
- Moderaator koostab lepingud, kirjutab koguse maha;
- Moderaator kinnitab tellimusi;
- Kasutaja (või klient) teeb programmis tellimuse;
- Kasutaja teavitab moderaatorit tellimusest programmi teel;
- Vaatleja saab taotleda programmi kasutajalt tellimust.

3.2 Põhiprotsessid

- Lepingu loomine;
- Lepingu koguse mahakandmine;
- Lepingu kustutamine;
- Kliendi registreerimine;
- Tellimuse koostamine;
- Tellimuse kinnitamine.

3.2.1 Tegutsejad

- Administraator;
- Direktor;
- Haldur;
- Klient;
- Vaatleja.

3.2.2 Asukohad

- Administraator töötab oma arvutiga interneti kaudu;
- Direktor töötab tehases oma kabinetis oma arvuti taga;
- Haldur töötab tehases oma kabinetis oma arvuti taga;
- Klient ja vaatleja töötavad omas firmas kabinetis oma arvuti taga.

3.3 Terviksüsteemi tükeldus

Tegemist on lao andmetöötlus infosüsteemiga.

3.3.1 Pädevusalad

Organisatsiooni sisesed pädevusalad:

- Direktori pädevusala;
- Halduri pädevusala.

Organisatsiooni välised pädevusalad:

- Kliendi pädevusala.

3.3.2 Funktsionaalsed allsüsteemid

Sisulised allsüsteemid:

- Kliendi funktsionaalne allsüsteem;
- Tellimuse funktsionaalne allsüsteem.

Administratiivsed allsüsteemid:

- Haldurite funktsionaalne allsüsteem.

3.3.3 Registrid

Sisulised registrid:

- Isikute register;
- Koguse register;
- Pakendite tüübi register;
- Ajaloo register.

Administratiivsed registrid:

- Kliendi register;
- Lepingute register.

Järgnevas töös vaatleme täpsemalt tellimuse funktsionaalset allsüsteemi ja tema poolt kasutatavaid registreid.

3.4 Tellimuse funktsionaalne allsüsteem

3.4.1 Eesmärgid

- Võimaldada tellijatel koostada tellimusi;
- Võimaldada saada ülevaade lepingutest;
- Võimaldada automaatselt kontrollida, et tellimuses on valitud minimaalselt üks pakk;
- Võimaldada kinnitada tellimusi.

3.4.2 Allsüsteemi kasutavad pädevusalad

- Halduri pädevusala;
- Kliendi pädevusala.

3.5 Allsüsteemi poolt kasutavad registrid

Allsüsteem teenindab tellimuste registrit.

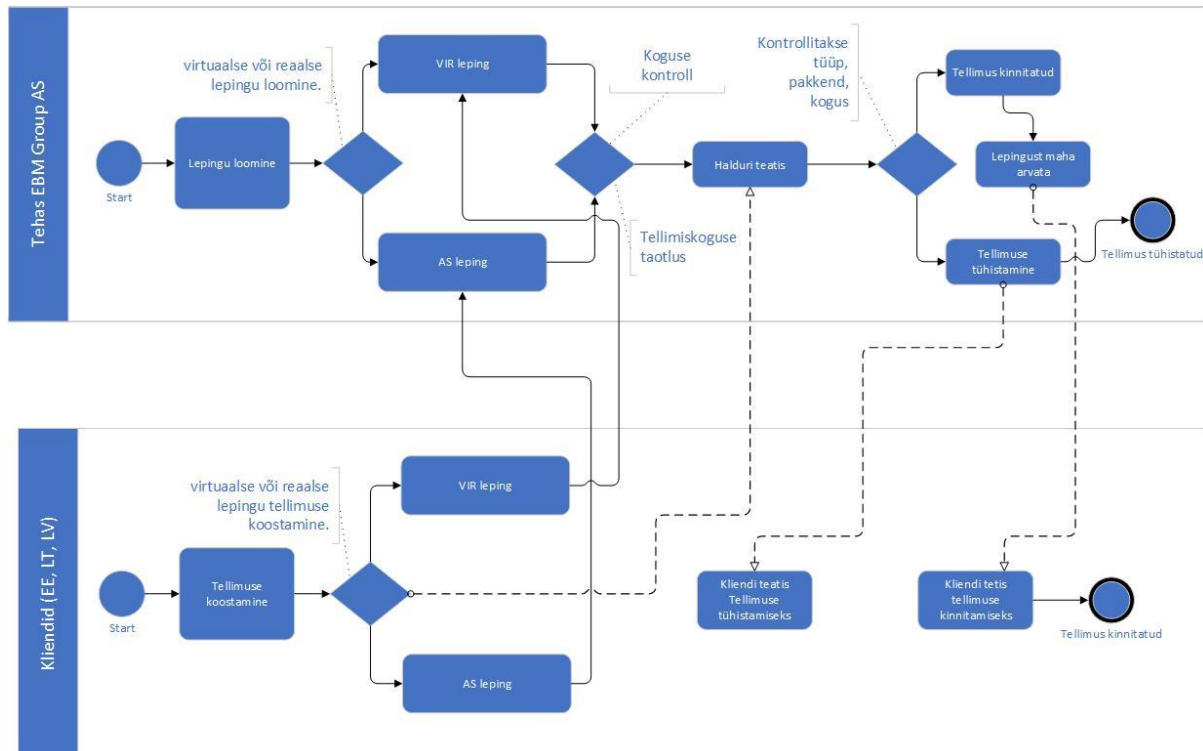
Allsüsteem kasutab:

- Isikute register;
- Lepingute register;

- Tellimuse register;
- Pakendite tüübi register;
- Ajaloo register.

3.6 Allüsteemi põhiprotsessi tegevusdiagramm

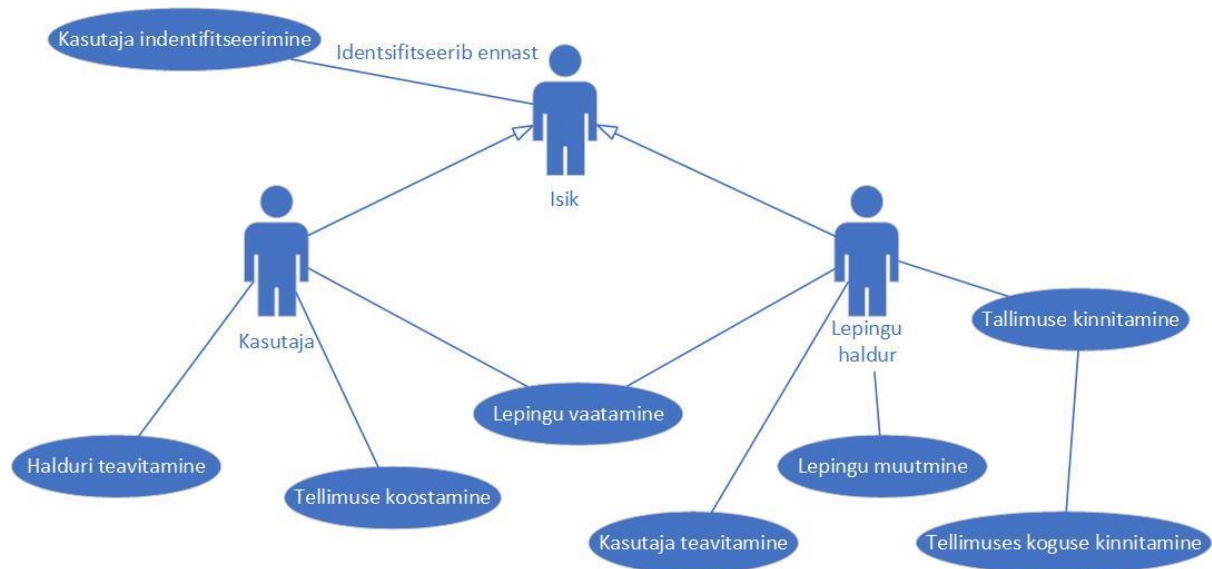
Seal esitatakse põhiprotsessi tegevusdiagramm, kus näidatakse lepingu loomine, tellimuse koostamine ja halduri ja kliendi teavitamine.



Joonis 3. Tellimuse allüsteemi põhiprotsessi tegevusdiagramm.

3.7 Allsüsteemi kasutusjuhtude eskiismudel

Seal esitatakse kasutusjuhtude eskiismudel, kus näidatakse kes ja mida saaks süsteemis teha.



Joonis 4. Tellimuse funktsionaalse allsüsteemi kasutusjuhtude eskiismudel.

Järgnevalt esitatakse kõrgtaseme formaadis kasutusjuhtude teksti kirjeldused.

Kasutusjuht: Kasutaja identifitseerimine.

Tegutsejad: Haldur, Kasutaja (isikud).

Kirjeldus: Isik identifitseerib ennast. Selleks sisestab ta kasutajanime, parooli. Süsteem autendib isikut, st. kontrollib isiku väidetavat identiteeti. Kui isik on identifitseeritud, siis lubatakse isik süsteemi siseneda, vastasel juhul mitte.

Kasutusjuht: Tellimuse koostamine tehases.

Tegutsejad: Haldur.

Kirjeldus: Lepingujuht teeb tehasele tellimuse. Tellimuses kuvatakse õli tüüp, kogus ja pakend. Villimistsehh võtab selle tellimuse vastu ja täidab selle nõutavatesse pakenditesse.

Kasutusjuht: Tellimuse koostamine.

Tegutsejad: Kasutaja (Klient).

Kirjeldus: Kasutaja tahab tellida õli konkreetses pakendis. Klient esitab tellimuse, valides õlisorti, pakendi ja koguse. Saadab kinnituse programmis lepingute juhile.

Kasutusjuht: Tellimuse kinnitamine.

Tegutsejad: Haldur.

Kirjeldus: Kui kliendi tellimuses on kõik andmed õiged ja õli kogus olemas, siis tellimus võetakse vastu ja saadetakse edasi villimisse.

Kasutusjuht: Tellimuse ja lepingu vaatamine.

Tegutsejad: Haldur, Kasutaja.

Kirjeldus: Kui haldur või kasutaja soovib tellimuse andmeid vaadata, kuvab süsteem kõik tellimused. Kasutaja saab vaadata kogu lepingu maha kirjutamise ajalugu.

Kasutusjuht: Tellimuse ja lepingu muutmine.

Tegutsejad: Haldur.

Kirjeldus: Haldur saab muuta andmeid nii tellimuses kui ka lepingus.

Kasutusjuht: Tellimuse kustutamine.

Tegutsejad: Haldur.

Kirjeldus: Kui tellimuses on täidetud ebaõiged andmed (kogus, pakend), võib haldur tellimuse kustutada.

Kasutusjuht: Lepingu muutmine.

Tegutsejad: Haldur.

Kirjeldus: Haldur saab lepingu kustutada, kui see on aegunud või õli kogus lepingus on null. Tavaliselt leping arhiveeritakse ja seda aktiivsetes lepingutes ei kuvata.

3.8 Täiendavad nõudmised

Tabel 1. Nõudmise kirjeldused.

Tüüp	Nõudmise kirjeldus
Tarkvara	Süsteem peab andmete hoidmiseks kasutama andmebaasisüsteemi abil loodud andmebaasi. Kõik kasutajad saavad kasutada erinevaid operatsioonisüsteeme, kus on juurdepääs Internetile läbi veebibrauseri.
Turvalisus	Tellimuse allsüsteem teenindab tellimuse registrit. Tellimuse registri turvaklass võiks olla: K1T1S3 K1 – töökindlus – 90% (lubatud summaarne seisak nädalas ~ ööpäev); lubatav nõutava reaktsiooniaja kasv tippkoormusel – tunnid (1÷10); T1 – info allikas, selle muutmise ja hävitamise fakt peavad olema tuvastatavad; info õigsuse, täielikkuse ja ajakohasuse kontroll erijuhtudel ja vastavalt vajadusele; S3 – ülisalajane info: info kasutamine on lubatud ainult teatud kindlatele kasutajatele, juurdepääs teabele on lubatav juurdepääsu taotleva isiku õigustatud huvi korral.
Keel	Süsteemi kasutajaliides on inglise keeles ja kasutusjuhend on vene keeles.
Töökiirus	Päringu tegemisel ja andmete muudatuse salvestamine süsteemi poolt ei tohi võtta aega rohkem kui 10 sekundit.
Kasutajaliides	Kasutajate kasutajaliides peab olema veebipõhine.
Arendusvahendid	Arendusvahendina tuleks kasutada organisatsioonile hangitud Microsoft Visio.
Varukoopiad	On vaja teha varukoopiad iga päev.

3.9 Tellimuste register

3.9.1 Eesmärgid

Säilitada informatsiooni tellimustest, et oleks tagatud koguse mahakandmise funktsionaalses allsüsteemis defineeritud eesmärkide täitmine.

3.9.2 Registrit kasutavad pädevusalad

Registri haldamine on programmi ülesanne. Registrit kasutavad nii programmi haldurit kui ka kasutajad.

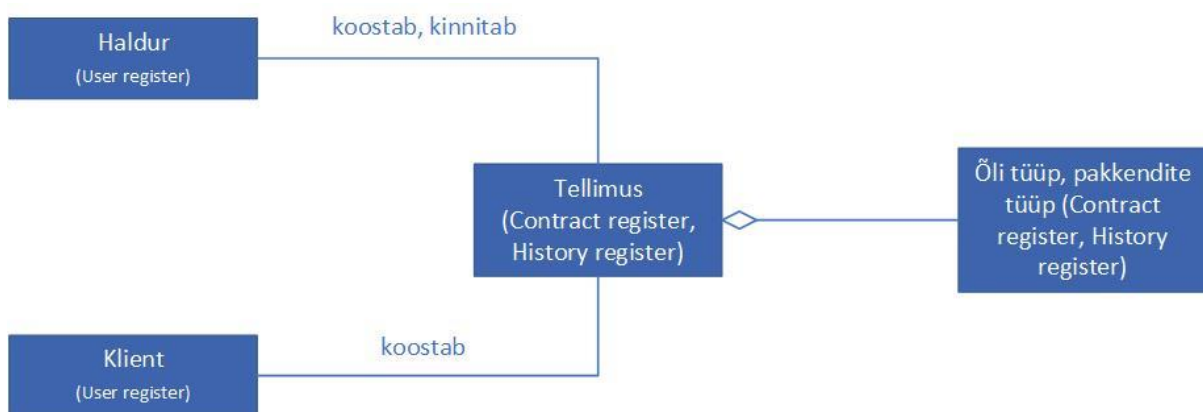
3.9.3 Registrit teenindavad funktsionaalsed allsüsteemid

Tellimuste registri andmeid kasutab ja uuendab programmi funktsionaalne allsüsteem.

3.9.4 Infovajadused

- Nimekiri tellijatest, kes olid teinud tellimusi;
- Tellimuses olevate pakkide nimekiri;
- Lepingust maha kantud õli kogus;
- Lepingus kõigi tegevuste salvestamine ja kuvamine.

3.9.5 Registri kontseptuaalne eskiismudel



Joonis 5. Registri kontseptuaalne eskiismudel.

4 EBM Grupp AS-le pakutav süsteemi tehniline arhitektuur

4.1 Tellimuse/mahakandmise funktsionaalne allsüsteem

4.1.1 Kasutusjuhtude mudel

Kasutusjuht: Kasutaja identifitseerimine

Primaarne kasutaja: Administraator, moderaator, kasutaja (isikud)

Osapooled ja nende huvid:

- Administraator, moderaator, kasutaja: Soovivad siseneda süsteemi ja teha tegevusi neile antud volituste piires.

Käivitatav sündmus: Isik soovib süsteemi siseneda.

Eeltingimused: Isik on juba registreeritud.

Järeltingimused: On tehtud kindlaks, kas isikul on õigus süsteemi siseneda või mitte. Isik on autenditud ja talle on antud võimalus kasutada süsteemi talle antud volituste piires.

Stsenaarium (tüüpiline sündmuste järjestus):

1. Isik soovib siseneda süsteemi.
2. **Süsteem** palub isikul ennast identifitseerida.
3. Isik identifitseerib ennast (sisestades kasutajanime, parooli).
4. **Süsteem** kontrollib, kas esitatud volitustõendiga (antud juhul parooliga) isiku andmed on süsteemis olemas või mitte ([vt 4.2.2 OP1.0](#)).
5. **Süsteem** annab isikule volituse süsteemi kasutada.

Isik võib üritada süsteemi siseneda kuni kolm korda.

Laiendused (või alternatiivne sündmuste käik)

4a. Kui süsteem ei leia esitatud volitustõendiga isiku siis, ei saa isik õigust süsteemi kasutada.

1. Süsteem kuvab subjektile teate, et taolist kasutajat ei leidu ja ei anna talle õigust süsteemi kasutada.

Kasutusjuht: Lepingu koostamine

Primaarne tegutseja: Moderaator (e. Haldur)

Osapooled ja nende huvid:

- **Moderaator:** Soovib kiiresti ja mugavalt enda lepingu koostada.

Eeltingimused: Moderaator on identifitseeritud.

Järeltingimused: Alustanud lepingu koostamine.

Stsenaarium (tüüpiline sündmuste järjestus):

1. **Moderaator** alustab lepingu koostamist ([vt 4.2.2 OP2.0](#)).
2. **Moderaator** otsustab mis leping tuleb rakendada, virtuaalne või reaalne.
3. **Moderaator** sisestab:
 - 3.1. Märgib, milline riik (tellijajaoks) näeb lepingut;
 - 3.2. Lepingu algus- ja lõppkuupäeva;
 - 3.3. Riigi, kuhu tarnitakse;
 - 3.4. Õli tüübi;
 - 3.5. Õli koguse lepingus;
 - 3.6. Pakendi tüübi ja ostuhinna.
4. **Süsteem** kontrollib sisestatud andmed ja salvestab uus leping.
5. **Süsteem** näitab üldises lepingute nimekirjas uut lepingut.

Kasutusjuht: Lepingu mahakandmine

Primaarne tegutseja: Moderaator (e. Haldur)

Osapooled ja nende huvid:

- Moderaator: Soovib kiiresti ja mugavalt enda lepingust teha mahakandmist.

Eeltingimused: Moderaator on identifitseeritud.

Järelingimused: Alustanud lepingu mahakandmine.

Stsenaarium (tüüpiline sündmuste järjestus):

1. Moderaator valib virtuaalne või reaalne leping.
2. Moderaator alustab lepingu mahakandmist ([vt 4.2.2 OP2.1](#)).
3. Moderaator sisestab:
 - 3.1. Konto lepingust mahaarvamise summa;
 - 3.2. Kommentaar - pakendi tüüp ja kogus;
 - 3.3. Konto ülekandmine lepingult teisele lepingule (kui vajalik);
 - 3.4. Konto mahaarvamise kuupäev.

4. **Süsteem** kontrollib sisestatud andmed ja salvestab lepingu muudatusi.

5. **Süsteem** näitab üldises lepingute nimekirjas muudetud lepingut.

Kasutusjuht: Lepingu kustutamine

Primaarne tegutseja: Moderaator

Osapooled ja nende huvid:

- Moderaator: Soovib kiiresti ja mugavalt enda lepingut kustutada, kui tekkinud viga.

Eeltingimused: Moderaator on identifitseeritud ja leping on sisestatud.

Järelingimused: Valitud reaalne või virtuaalne leping.

Stsenaarium (tüüpiline sündmuste järjestus):

1. Moderaator kustutab leping ([vt 4.2.2 OP3.1](#)).
2. **Süsteem** kustutab leping.

Kasutusjuht: Tellimuse koostamine kasutajate poolt

Primaarne tegutseja: Kasutaja (e. Tellija)

Osapooled ja nende huvid:

- Kasutaja: Soovib kiiresti ja mugavalt enda tellimuse planeerida ja koostada. Soovib kinnitust selle kohta, et tema tellimus on kinnitatud.

Eeltingimused: Kasutaja on identifitseeritud.

Järelingimused: Tellimus on koostatud ja sellesse on valitud lepingut, pakendi tüüp ja kogus.

Stsenaarium (tüüpiline sündmuste järjestus):

1. Kasutaja alustab tellimuse koostamist.
2. Kasutaja otsib lepingu number.
3. Kasutaja sisestab pakendi tüüp ja kogus ([vt 4.2.2 OP2.2](#)).
4. **Süsteem** kontrollib kogus, salvestab pakendi tüüp ja kogus tellimusesse.
5. **Süsteem** saadab teavituse e-posti teel moderaatorile ja annab teada programmis, et tuli uus tellimus.

Kasutusjuht: Tellimuse kinnitamine

Primaarne kasutaja: Moderaator (e. Haldur)

Osapooled ja nende huvid:

- Moderaator: Kinnitada saadetud tellimusi.

Käivitav sündmus: Programm on saanud pool tellimuse kogus ja lepingu number.

Eeltingimused: Moderaator on identifitseeritud. Tellimus on koostatud ja saadetud.

Järeltingimused:

Stsenaarium (tüüpiline sündmuste järjestus):

1. Kui kõik pakendi tüüpe mis on määratud tellimuses ja kogus on olemas laos, moderaator muudab tellimuse seisundi “kinnitatuks” ([vt 4.2.2 OP2.3](#)).

Laiendused (või alternatiivne sündmuste käik)

1. Kui mitte kõik pakendi tüüpe mis on määratud tellimuses on olemas laos, moderaator muudab tellimuse seisundi “mittekinnitatuks” ([vt 4.2.2 OP2.4](#)).

Kasutusjuht: Tellimuse vaatamine

Primaarne kasutaja: Kasutaja, Moderaator

Osapooled ja nende huvid:

- Kasutaja soovib näha kõigi enda esitatud tellimused ja nende seisundid.
- Moderaator soovib näha, kõigi kasutajate esitatud tellimused ja nende seisundit

Käivitatav sündmus: kasutaja või moderaator soovib vaadata informatsiooni tellimustest.

Eeltingimused: Kasutaja või moderaator on identifitseeritud.

Järeltingimused: Kasutaja või moderaator on saanud informatsiooni tellimustest.

Stsenaarium (tüüpiline sündmuste järjestus):

1. Moderaator või kasutaja avaldab soovi vaadata tellimuse andmeid.
2. **Süsteem** kuvab nimekirja koostatud tellimustest kus on näha tellimuse seisund, tellimuse kogus, pakendi tüüp, lepingu number ja tellimuse kuupäev.
3. Kasutaja või moderaator valib nimekirjast tellimuse.
4. **Süsteem** kuvab tellimuse andmed. Sealhulgas kuvatakse tellimuse komponente ([vt 4.2.2 OP4.1](#)).

Laiendused (või alternatiivne sündmuste käik)

2a. Moderaator näeb ka kasutajate nimesid, kes on tellimust teinud.

Kasutusjuht: Tellimuse muutmine

Primaarne tegutseja: Moderaator

Osapooled ja nende huvid:

- Moderaator: Soovib kiiresti ja mugavalt enda tellimuse muuta. Kui tellimuses kõik andmed on õiged, siis kinnitab see tellimus, kui pole, siis muudab pakendi tüüp või kogus.

Eeltingimused: Moderaator on identifitseeritud ja kliendi tellimus on „mittekinnitatud”.

Järeltingimused: Tellimus on koostatud ja selles on valitud lepingu number, pakendi tüüp ja kogus.

Stsenaarium (tüüpiline sündmuste järjestus):

1. Moderaator alustab tellimuse muutmist ([vt 4 2.2 OP2.3](#)).
2. Moderaator kontrollib tellimuses sisestatud andmed.
3. Moderaator muudab tellimuses andmed, kui on vaja.
4. **Süsteem** salvestab muudatused tellimusesse.
5. **Süsteem** arvutab tellimuse kogus ja salvestab muudatusi.

Kasutusjuht: Tellimuse kustutamine

Primaarne tegutseja: Kasutaja

Osapooled ja nende huvid:

- Kasutaja: Soovib kiiresti ja mugavalt enda tellimuse kustutada.

Eeltingimused: Kasutaja on identifitseeritud ja kliendi tellimus on „Canceled”.

Järeltingimused: Tellimus on koostatud ja selles on vale valitud lepingu number pakendi tüüp või kogus mille pole laos.

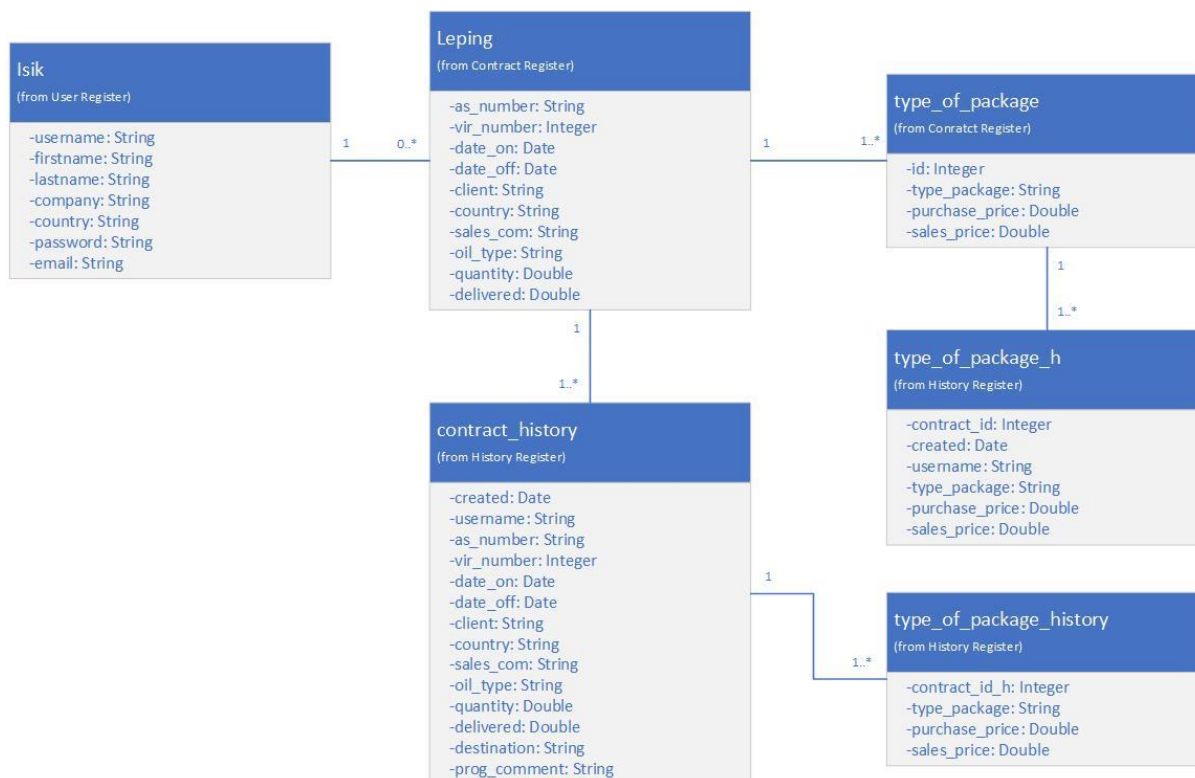
Stsenaarium (tüüpiline sündmuste järjestus):

1. Kasutaja kustutab tellimust.
2. Süsteem kustutab tellimust ([vt 4.2.2 OP4.2](#)).

4.2 Tellimuse/mahakandmise register

4.2.1 Kontseptuaalne andmemudel

4.2.1.1 Olemi-suhte diagramm



Joonis 6. Olemi-suhte diagramm

4.2.1.2 Olemitüüpide definitsioonid

Tabel 2. Olemitüüpide definitsioonid

Olemitüübi nimi	Kuuluvus registrisse	Definitsioon
Isik	Isikute register	Mistahes süsteemiga seotud füüsiline isik. (nt. Administraator, Haldur, Klient)

Leping	Lepingute register	Reaalne või virtuaalne leping konkreetse õli koguse arvestamiseks.
Pakendi tüüp	Pakendite tüüpi register	Pakendi tüüp, mis on seotud konkreetse lepinguga.
Tellimus	Tellimuse register	Klient teeb tehasesse tellimuse, kus kuvatakse lepingu number, pakendi tüüp ja kogus.
Lepingu ajaloo	Ajaloo register	Pärast koguse muutmist lepingus fikseeritakse muudatuse ajalugu.
Pakendi tüüpi ajaloo	Ajaloo register	Pärast lepingus koguse muutmist fikseeritakse pakendiliigi ajalugu.

4.2.1.3 Atribuutide definitsioonid

Tabel 3. Atribuutide definitsioonid

Olemitüübi nimi	Atribuudi nimi	Atribuudi definitsioon	Näiteväärtus
Isik	username	Isikule antud unikaalne nimi, mida kasutatakse tema süsteemi poolseks autentimiseks.	MarkS
Isik	firstname	Isikute eesnimi.	Mark
Isik	lastname	Isikute perekonnanimi.	Sepp
Isik	company	Firma, kus töötab isik	nt. EBM Group AS
Isik	country	Riik, kus töötab isik.	EE / LV / LT
Isik	email	e-post mis kasutab isik	mark.s@hotmail.ee
Isik	password	Isikule antud unikaalne salasõna,	*****

		mida kasutatakse tema süsteemi poolseks autentimiseks.	
Leping, Tellimus, Lepingu ajaloo	number	Lepingu number, reaalne või virtuaalne	AS93291, EBM-7
Leping	contract date	Lepingu algus ja lõpp aeg	10-10-2022 - 30-04-2023
Leping	client	Ettevõtte nimi, kellele õli tarnitakse	FREE, RIMI
Leping	country	Riik, mille jaoks on leping tehtud	EE, LV, LT
Leping	sales comment	Lepinguga seotud ettevõtte lühinimetus	CF, BA, EG, FREE
Leping, Tellimus, Lepingu ajaloo	oil type	Lepinguga seotud õli liik	Soya, Raps, Sun, etc.
Leping, Tellimus, Lepingu ajaloo, Pakendi tüüpi register	type of package	Lepinguga seotud pakendite liik	1L, 0.9L, 3L, etc.
Leping	purchase price	Pakendi ostuhind	1.7480
Leping	delivered	Kui palju õli on tarnitud	74 540.00
Leping, Tellimus, Lepingu ajalugu	prog comment	Märge lepinguga tehtud toimingust.	Created, Delivered, Changed, Ordered, Canceled.

4.2.2 Andmebaasioperatsioonide lepingud

OP1.0 Kasutaja identifitseerimine (username)

Eeltingimused:

Programmi kasutaja tahab identifitseerida ennast

Järeltingimused:

Programmi kasutaja on juba programmi veebilehel

Kasutus kasutusjuhtude poolt: Kasutaja identifitseerimine

OP2.0 Lepingu koostamine (number)

Eeltingimused:

Haldur tahab lepingu koostada.

Haldur on identifitseeritud süsteemis moderaatori rollis ja on avatud veebiaken "Create".

Järelingimused:

Süsteem salvestab uus leping ja ajaloos kirjeldab lepingu number ja "prog comment" "Created".

Kasutus kasutusjuhtude poolt: Lepingu koostamine

OP2.1 Lepingu mahakandmist (number)

Eeltingimused:

Leping on koostatud ja sisestatud;

Haldur valis lepingu number ja avas aken "Delivery";

Tuleb sisestada mahakandmise kogus, kommentaar ja "Inv. Number".

Järelingimused:

Lepingust sisestatud kogus on maha arvatud;

Lepingu ajaloos kirjeldatakse lahtris "prog comment" „Delivered”.

Kasutus kasutusjuhtude poolt: Lepingu mahakandmine (Tellimuse allsüsteem).

OP2.2 Tellimuse koostamine (number)

Eeltingimused:

Leping on süsteemis;

Kasutaja valis lepingu number ja avas aken "Delivery";

Tuleb sisestada mahakandmise kogus, kommentaar ja "Inv. Number".

Järelingimused:

Lepingust sisestatud kogus on maha arvatud;

Lepingu ajaloos kirjeldatakse lahtris "prog comment" „Ordered”;

Saadetakse teavitus moderaatori poole, et tehtud tellimus.

Kasutus kasutusjuhtude poolt: Tellimuse koostamine kasutajate poolt (Tellimuse allsüsteem).

OP2.3 Tellimuse seisundi muutmine kinnitatuks (tellimuse id, number)

Eeltingimused:

Tellimus on koostatud ja saadetud teavitus halduri poole

Järelingimused:

Tellimuse ajaloos seisund muudetakse kinnitatuks "Delivered"

Kasutus kasutusjuhtude poolt: Tellimuse kinnitamine (Tellimuse allsüsteem).

OP2.4 Tellimuse seisundi muutmine mitte kinnitatuks (tellimuse id, number)

Eeltingimused:

Tellimus on koostatud ja saadetud teavitus halduri poole;

Tellimuse staatus "Ordered".

Järelingimused:

Tellimuse ajaloos seisund muudetakse kinnitatuks "Canceled";

Haldur lisas kommentaar, miks pole kinnitatud;

Kogus, mis oli maha arvatud, lükatakse tagasi lepingusse.

Kasutus kasutusjuhtude poolt: Tellimuse kinnitamine (Tellimuse allsüsteem).

OP2.5 Tellimuse kommentaari muutmine (tellimuse id, tekst)

Eeltingimused:

Tellimus on mittekinnitatud.

Järelingimused:

Tellimuse kommentaarisse on kirjutatud mittekinnitamise põhjus.

Kasutus kasutusjuhtude poolt: Tellimuse kustutamine (Tellimuse allsüsteem).

OP3.1 Lepingu kustutamine (number)

Eeltingimused:

Leping on koostatud ja sisestatud;

Haldur on identifitseeritud süsteemis moderaatori rollis;

Haldur avas veebiaken "Change/Delete".

Järelingimused:

Leping on kustutatud koos tema komponentidega.

Kasutus kasutusjuhtude poolt: Tellimuse kustutamine (Tellimuse allsüsteem).

OP4.1 Tellimuse vaatamine (tellimuse id)

Eeltingimused:

Tellimus on koostatud, muutmisel või tema seisund on „mittekinnitatud”.

Järelingimused:

Tellimus on koostatud koos tema komponentidega.

Kasutus kasutusjuhtude poolt: Tellimuse vaatamine (Tellimuse allsüsteem).

OP4.2 Tellimuse kustutamine (tellimuse id)

Eeltingimused:

Tellimus on „mittekinnitatud”.

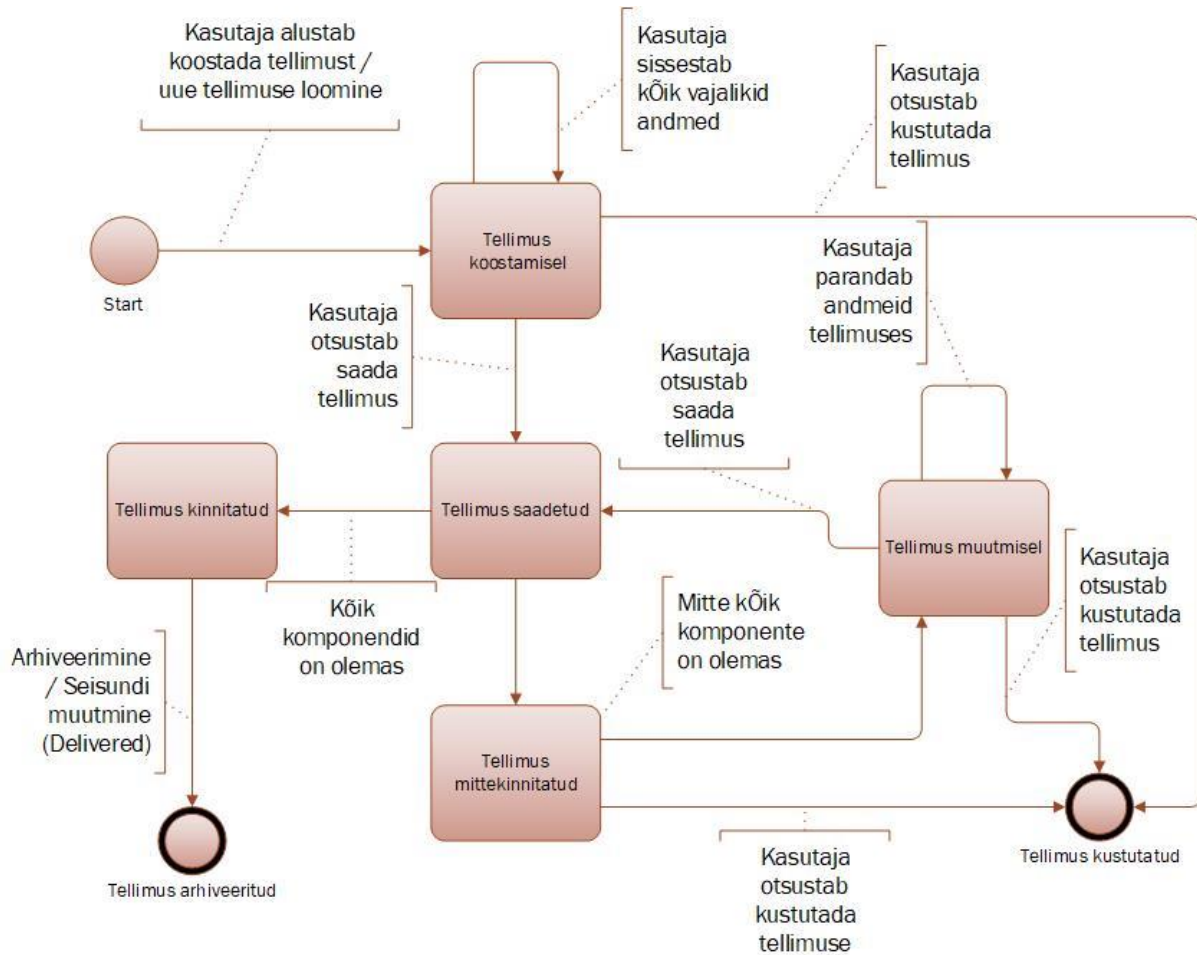
Järelingimused:

Tellimus on kustutatud ajaloost koos tema komponentidega.

Kasutus kasutusjuhtude poolt: Tellimuse kustutamine (Tellimuse allsüsteem).

4.2.3 Registri põhiobjekti seisundidiagramm

Registri põhiobjekt on „Tellimus”



Joonis 7. Tellimuse seisundidiagramm.

4.3 CRUD maatriks

See CRUD maatriks esitatakse olemitüüpide ja kasutusjuhtude täpsusega.

Veergudeks on kasutusjuhud ja ridadeks olemitüübid.

	1	2	3	4	5	6	7	Kokkuvõte
Isik	R	R	R	R	R	R	R	R
Haldur	R	RU	RU	R	R	RU	RU D	RUD
Kasutaja	R	R	RU	R	CR	R	RD	CRUD
Leping				R	R	R	R	R
Pakendi tüüp				R	R	R	R	R
Saadavus				R	R	R	R	R
Tellimus		RU	RU	R	CRU	RU	RD	CRUD
Tellimuse kogus				R	CRUD	CRD	RD	CRUD
Lepingu ajaloo		R	R	R	R	R	R	R
Pakendi tüüpi ajaloo				R	R	R	R	R

1. Kasutaja identifitseerimine;
2. Tellimuse kinnitamine;
3. Isikute teavitamine;
4. Tellimuse vaatamine;
5. Tellimuse koostamine;
6. Tellimuse muutmine;
7. Tellimuse kustutamine.

Kollasel taustal on esitatud olemitüübid, mis kuuluvad vaadeldava allsüsteemi teenindatavasse registrisse. Iga sellise olemitüübi kohta peab leiduma vähemalt üks kasutusjuhtum, mille käigus vastavaid andmeid luuakse.

5 EBM Grupp AS pakutud süsteemi loogiline disain

5.1 Tellimuse ja makandmise funktsionaalne allsüsteem

5.1.1 Realiseeritavad töökohad

- Halduri töökoht
- Kasutaja töökoht

5.1.2 Reaalsete kasutusjuhtude kirjeldused

Järgnevalt esitatakse allsüsteemi kõige olulisema kasutusjuhu kirjeldus reaalse kasutusjuhu kirjelduse täpsusega.

Siin on põhitabel (vt Joonis 8.) lepingutega, kus kuvatakse kõik vajalikud andmed lepingu kohta. Sorteerida saab ka 4 parameetri järgi, teada saada mahakantud õli koguhulk ja jääkkogus. Uue lepingu loomise protsessi kirjeldatakse allpool. ([vt Lisa 1. Kasutusjuht 1.](#))

User: Admin | Exit | Administration | Instruction | Print | **Orders (1)** **C**

AS No.	Virt. No.	Contract Date	Client	Country	Sales Com.	Oil Type	Quantity	T. of P.	Purchase Price	Sales Price	Delta	Delivered	Rem. Q-ty
AS220166	EE LV LT	07-04-2022 - 30-04-2022	FREE	EE	CF	Sun	30'200.00	1L	1.9256			29'810.00	390.00
AS220283	EE LV LT	01-11-2022 - 31-01-2023	FREE	LV	EG	Raps	150'000.00	1L SL_PET	1.8042 1.7742			141'825.00	8'175.00
AS220284	EE LV LT	01-11-2022 - 31-01-2023	FREE	LV	EG	Raps	150'000.00	1L SL_PET	1.7398 1.7098			38'600.00	111'400.00
AS220301	EE LV LT	01-11-2022 - 31-12-2022	FREE	LT	BA	Raps	18'000.00	1L	1.7582			9'825.00	8'175.00
AS220303	EE LV LT	01-01-2023 - 31-01-2023	FREE	LT	BA	Raps	10'000.00	1L	1.6984			0.00	10'000.00
AS220305	EE LV LT	01-02-2023 - 30-04-2023	FREE	LT	BA	Raps	42'000.00	1L	1.6272			0.00	42'000.00
AS220306	EE LV LT	01-02-2023 - 30-04-2023	IBC	LT	BA	Raps	48.00	IBC	1.6356			20.00	28.00

Joonis 8. Põhitabel lepingutega.

Uue lepingu koostamise veebiaken. ([vt Lisa 1. Kasutusjuht 1.](#))

Input Table - Google Chrome

Who can view:	EE <input checked="" type="checkbox"/> A	LV <input type="checkbox"/>	LT <input type="checkbox"/>				
Contract Number:	<input type="text"/>	Contract Date:	1 <input type="text"/> 3 <input type="text"/> 2023 <input type="text"/> D <input type="text"/> M <input type="text"/> Y <input type="text"/>	Client:	<input type="text"/>	Country:	<input type="text"/>
Sales Comment:	<input type="text"/>	Oil Type:	<input type="text"/>	Quantity:	<input type="text"/> .00	C Reset	
T. of P.:	<input type="text"/>	Pur. Price:	<input type="text"/>	Sale Price:	<input type="text"/>	B + Enter	

Joonis 9. Uue lepingu koostamise veebiaken.

Selleks, et kasutaja saaks tellimuse esitada või moderaator teatud koguse õli maha kirjutaks, tuleb avada aken “Delivery”. ([vt Lisa 1. Kasutusjuht 2.](#))

Joonis 10. Tellimuse koostamine või õli mahakandmise veebiaken

Lepingu täielik muutmine, nulli viimine või kustutamine koos ajalooga. ([vt Lisa 1. Kasutusjuht 3.](#))

Joonis 11. Tellimuse muutmine veebiaken.

Tellimuste lehel halduri jaoks , kus saab lisa kommentaar, tühista või vastu võta tellimust.. (vt [Lisa 1. Kasutusjuht 4.](#))

User: **Admin** (administrator)

Order id	Order user	Contract Number	Date and Time	Destination	Contract Quantity	Ordered/Delivered Quantity	User Comment	Prog Comment	Cancel order	Agree
28289	Margit	AS220166	2023-04-28 12:36:35	12347	30200.00	60.00	A na CF: 1 pal x 1L	Ordered	Cancel (28289)	Agree
28266	Admin	187	2023-03-27 21:22:18	0002	3444.00	200.00	na CF: 1 ibc x 200kg	Delivered	B	C
28265	Admin	187	2023-03-27 21:21:05	0001	3444.00	300.00	LV 2p 1L	Delivered		
28258	Tanja	AS230059	2023-03-22 14:56:45	18707	155000.00	4000.00	na CF: 5 pal x 5L PET	Delivered		
28257	Slava	AS220283	2023-03-21 19:05:21	load22/03	150000.00	13325.00	7p 5L pet 3p 5L pet e900 5p 5L 1p 1L	Delivered		
28256	Slava	AS230011	2023-03-21 19:02:16	load22/04	30000.00	5650.00	5p 5L pet 2p 1L	Delivered		

Joonis 12. Tellimuste vaatamiseks veebiaken halduri jaoks.

Tellimuste lehel kliendi jaoks, kus saab vaadata millised tellimused on tellitud, tühistatud või vastu võetud. (vt [Lisa 1. Kasutusjuht 4.](#))

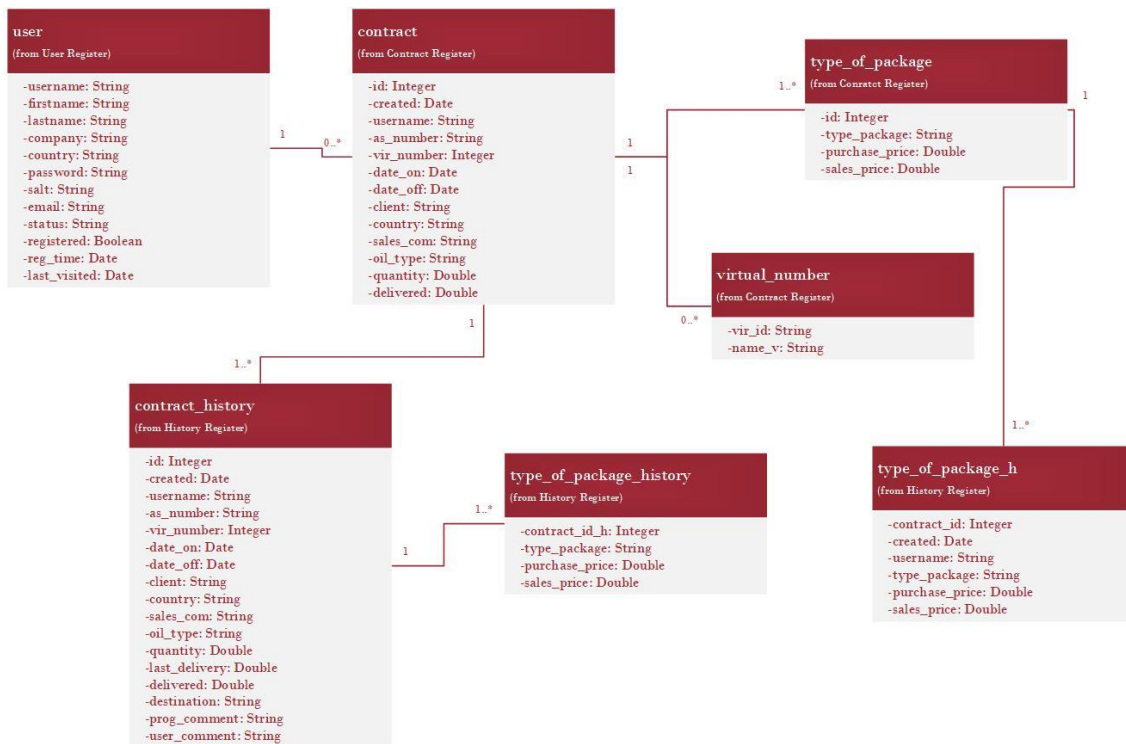
User: **Margit** (user)

Contract Number	Date and Time	Destination	Contract Quantity	Ordered/Delivered Quantity	User Comment	Prog Comment
AS220166	2023-04-28 12:36:49	12346	30200.00	50.00	na CF: 1 pal x 3L (some mistake) (Canceled)	A Delete
AS220166	2023-04-28 12:36:35	12347	30200.00	60.00	na CF: 1 pal x 1L	B Ordered
AS230059	2023-03-03 09:35:43	18625	125000.00	5880.00	5L raps OiliO - 1 pallet, 5L Toidu oiliO - 1 pallet, 20L raps OiliO- 3 pallets, 5L PET raps - 2 pallets, 5L Fry ovilo Pet - 1 pallet	C Delivered
AS230013	2023-03-03 09:34:22	18625	35.00	4.00	920kg IBC raps - 4 pc	Delivered
AS230059	2023-03-02 14:23:46	18619	125000.00	10625.00	5L Fry ovilo - 5 pallets, 5LK PET raps - 5 pallets, 5L Toidu OiliO - 1 pallet, 1L Toidu OiliO - 1 pallet	Delivered

Joonis 13. Tellimuste vaatamiseks veebiaken kliendi jaoks.

5.2 Laiendatud tellimuste registri loogiline disain

5.2.1 Loogiline andmebaasi diagramm



Joonis 14. Tellimuse registri loogilise disaini andmebaasi diagramm.

5.2.2 Tabelite detailed kirjeldused

Tabeli nimi: **user**

Tabel 4. Tabeli nimi: user

Veeru nimi	Tüüp	Pikkus	Võimalikud väärtused	Vaikeväärtus	Kohustuslik
username	varchar	20			jah
firstname	varchar	20			
lastname	varchar	20			
company	varchar	20			
country	varchar	20			jah
password	varchar	32	Parool MD5 šifrina		
salt	varchar	10	10-kohaline juhuslik väärtus		

			registreerimisel		
email	varchar	32	peab sisaldama "@"		
status	varchar	15			jah
registered	tinyint	1			
reg_time	datetime				
last_visited	datetime				

Unique Key (email)

Tabeli nimi: **contract**

Tabel 5. Tabeli nimi: contract

Veeru nimi	Tüüp	Pikkus	Võimalikud väärtused	Vaikeväärtus	Kohustuslik
id	int	11			jah
created	datetime				
username	varchar	20			
as_number	varchar	20	Peab algama tähega "AS"		
vir_number	int	10			
date_on	date				jah
date_off	date				jah
client	varchar	20			
country	varchar	20			
sales_com	varchar	20			
oil_type	varchar	20			
quantity	double	(16,2)			
delivered	double	(16,2)		0.00	jah
EEview	varchar	(1)		1	
LVview	varchar	(1)		0	
LTview	varchar	(1)		0	

Primary Key (id)

Unique Key (as_number)

Unique Key (vir_number)

Tabeli nimi: **virtual_number**

Tabel 6. Tabeli nimi: virtual_number

Veeru nimi	Tüüp	Pikkus	Võimalikud väärtused	Vaikeväärtus	Kohustuslik
name_v	varchar	10		virtual	
vir_id	int	10			jah

Tabeli nimi: **type_of_package**

Tabel 7. Tabeli nimi: type_of_package

Veeru nimi	Tüüp	Pikkus	Võimalikud väärtused	Vaikeväärtus	Kohustuslik
contract_id	int	10			jah
type_package	varchar	10			jah
purchase_price	double	(16,4)	Nõutav 4 kohta pärast koma	0.0000	jah
sales_price	double	(16,4)	Nõutav 4 kohta pärast koma	0.0000	jah

Tabeli nimi: **contract_history**

Tabel 8. Tabeli nimi: contract_history

Veeru nimi	Tüüp	Pikkus	Võimalikud väärtused	Vaikeväärtus	Kohustuslik
id_h	int	10			jah
created	varchar	10			
username	varchar	20			
delivery_date	double	(16,4)			jah
as_number	varchar	20	Peab algama tähega "AS"		
vir_number	int	10			
date_on	date				jah
date_off	date				jah
client	varchar	20			
country	varchar	20			
sales_com	varchar	20			
oil_type	varchar	20			

quantity	double	(16,2)			jah
last_delivery	double	(16,2)			
delivered	double	(16,2)			
destination	varchar	30			
prog_comment	varchar	15			
user_comment	text				

Primary Key (id_h)

Tabeli nimi: **type_of_package_history**

Tabel 9. Tabeli nimi: type_of_package_history

Veeru nimi	Tüüp	Pikkus	Võimalikud väärtused	Vaikeväärtus	Kohustuslik
contract_id_h	int	10			jah
type_package	varchar	10			jah
purchase_price	double	(16,4)	Nõutav 4 kohta pärast koma	0.0000	jah
sales_price	double	(16,4)	Nõutav 4 kohta pärast koma	0.0000	jah

Tabeli nimi: **type_of_package_h**

Tabel 10. Tabeli nimi: type_of_package_h

Veeru nimi	Tüüp	Pikkus	Võimalikud väärtused	Vaikeväärtus	Kohustuslik
contract_id	int	10			jah
created	datetime				
username	varchar	20			
type_package	varchar	10			jah
purchase_price	double	(16,4)	Nõutav 4 kohta pärast koma	0.0000	jah
sales_price	double	(16,4)	Nõutav 4 kohta pärast koma	0.0000	jah

6 Süsteemi tehniline arhitektuur ja realisatsioon

6.1 Esmane prototüüp

Infosüsteemi arendamise esimese iteratsiooni käigus luuakse andmebaasi ja rakenduse prototüüplahendus, kasutades andmebaasisüsteemi MySQL, HTTP server Apache ja PHP 5.3 (30 June 2009). See antakse testimiseks kolm testkasutajat ja komponendi andmed.

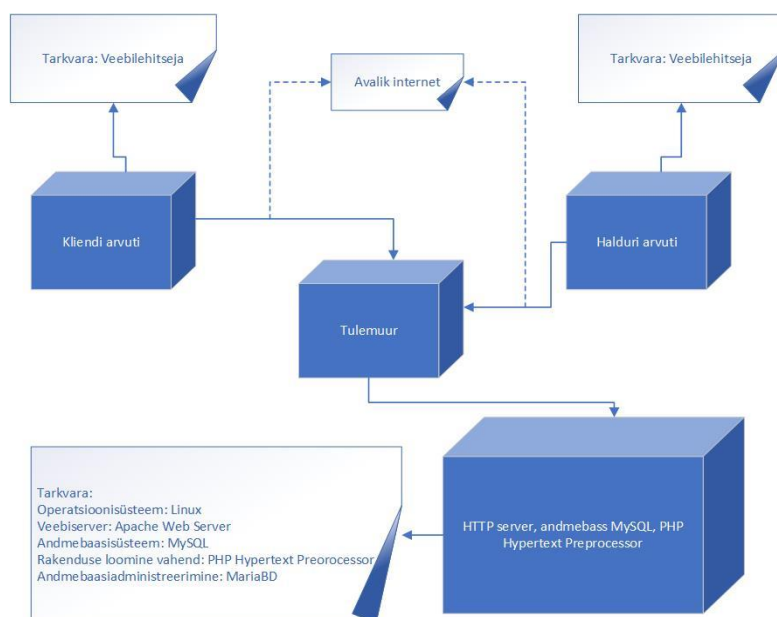
6.1.1 Nõudmised tarkvarale ja ristvarale

Andmebaas paigutatakse lokaal arvutis asuvale failiserverile, nii et mitu testkasutajat saab seda korraga kasutada.

6.1.2 Programmi uus versioon.

Hetkel, 2023, viimistletakse programmi uut versiooni PHP 8.2-le. Ülemineku ajal on autoril palju vigu, mis vajavad parandamist ja vahetamist uute funktsioonide vastu, et pääseda ligi MySQL andmebaasile. Kõik muudatused tehakse XAMPP HTTP-serveri kohalikus masinas. Järgnevalt on toodud näited uue versiooni MySQL-i funktsioonidest ja andmetötlusest.

6.2 Töötav süsteem



Joonis 15. Programmi infosüsteemi tehniline arhitektuur

6.2.1 Andmebaasi ja rakenduse server

- Operatsioonisüsteem: Linux;
- Veebiserver: Apache Web Server;
- Andmebaasisüsteem: PostgreSQL;
- Rakenduse loomise vahend: PHP Hypertext Preprocessor;
- Andmebaasi administreerimine: MariaBD.

6.2.2 Halduri, tellijate töökohad

Töökohad on realiseeritud PHP abil. Selle kasutamiseks peab arvutis olema vörgubrauser.

6.3 Realisatsioon MySQLis ja SQL laused

Kogu programmi info salvestamine on realiseeritud MySQL andmebaasi baasil. Tabelite loomine ja andmete vaatamine on realiseeritud allpool. Mõned nimed või andmed asendatakse väärtusega "x".

6.3.1 Andmebaasi loomine

```
CREATE DATABASE la01_98xxxxx;  
DROP DATABASE la01_98xxxxx;
```

6.3.2 Tabelid

- Tabeli loomine, kuhu kasutaja andmed salvestatakse.

```
CREATE TABLE `user` (  
  `username` varchar(20) DEFAULT NULL,  
  `firstname` varchar(20) DEFAULT NULL,  
  `lastname` varchar(20) DEFAULT NULL,  
  `company` varchar(20) DEFAULT NULL,  
  `country` varchar(20) DEFAULT NULL,  
  `salt` varchar(10) DEFAULT NULL,  
  `password` varchar(32) DEFAULT NULL,  
  `email` varchar(32) DEFAULT NULL,  
  `status` varchar(15) DEFAULT NULL,  
  `registered` tinyint(1) DEFAULT NULL,  
  `reg_time` datetime DEFAULT NULL,  
  `last_visited` datetime DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

- Tabeli loomine, kuhu salvestatakse loodud lepingud.

```
CREATE TABLE `contract` (
  `id` int(11) NOT NULL,
  `created` datetime DEFAULT NULL,
  `username` varchar(20) NOT NULL,
  `as_number` varchar(20) DEFAULT NULL,
  `vir_number` int(10) UNSIGNED DEFAULT NULL,
  `date_on` date NOT NULL,
  `date_off` date NOT NULL,
  `client` varchar(20) NOT NULL,
  `country` varchar(20) NOT NULL,
  `sales_com` varchar(20) NOT NULL,
  `oil_type` varchar(20) NOT NULL,
  `quantity` double(16,2) UNSIGNED NOT NULL,
  `delivered` double(16,2) UNSIGNED NOT NULL DEFAULT 0.00,
  `EEview` varchar(1) DEFAULT '1',
  `LVview` varchar(1) DEFAULT '0',
  `LTview` varchar(1) DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

- Tabeli loomine, virtuaalsete lepingute nummerdamiseks.

```
CREATE TABLE `virtual_number` (
  `name_v` varchar(10) DEFAULT NULL,
  `vir_id` int(10) UNSIGNED NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

- Tabeli loomine pakendiliikide jaoks lepingute alla.

```
CREATE TABLE `type_of_package` (
  `contract_id` int(10) UNSIGNED NOT NULL,
  `type_package` varchar(20) NOT NULL,
  `purchase_price` double(16,4) UNSIGNED NOT NULL,
  `sales_price` double(16,4) UNSIGNED NOT NULL DEFAULT 0.0000
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

- Ajaloo tabeli koostamine, mis fikseerib kõik lepingu muudatused (lepingu loomine, muutmine, koguse korrigeerimine ja õlikoguse mahakandmine).

```
CREATE TABLE `contract_history` (
  `id_h` int(11) NOT NULL,
  `created` datetime DEFAULT NULL,
  `username` varchar(20) DEFAULT NULL,
  `delivery_date` date NOT NULL,
  `as_number` varchar(20) DEFAULT NULL,
  `vir_number` int(10) UNSIGNED DEFAULT NULL,
  `date_on` date NOT NULL,
  `date_off` date NOT NULL,
  `client` varchar(20) DEFAULT NULL,
  `country` varchar(20) DEFAULT NULL,
  `sales_com` varchar(20) DEFAULT NULL,
  `oil_type` varchar(20) DEFAULT NULL,
  `quantity` double(16,2) UNSIGNED NOT NULL,
  `last_delivery` double(16,2) UNSIGNED DEFAULT NULL,
  `delivered` double(16,2) UNSIGNED DEFAULT NULL,
  `destination` varchar(30) DEFAULT NULL,
  `prog_comment` varchar(15) DEFAULT NULL,
  `user_comment` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

- Ajaloo tabeli koostamine (seotud tabeliga contract_history), mis fikseerib lepingus olevate pakitüüpide muutused ning ostu-müügihindade muutused.

```
CREATE TABLE `type_of_package_history` (
  `contract_id_h` int(10) UNSIGNED NOT NULL,
  `type_package` varchar(20) NOT NULL,
  `purchase_price` double(16,4) UNSIGNED DEFAULT NULL,
  `sales_price` double(16,4) UNSIGNED DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

- Ajaloo tabeli loomine (seotud tabeliga `type_of_package`), mis salvestab kõik lepingus sisalduvad muudatused paketi tüübi järgi.

```
CREATE TABLE `type_of_package_h` (
  `contract_id` int(10) UNSIGNED NOT NULL,
  `created` datetime DEFAULT NULL,
  `username` varchar(20) DEFAULT NULL,
  `type_package` varchar(20) NOT NULL,
  `purchase_price` double(16,4) UNSIGNED NOT NULL,
  `sales_price` double(16,4) UNSIGNED NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

6.3.3 Arvujada generaatorid ja unikaalseid väärtusi

```
ALTER TABLE `user`
  ADD UNIQUE KEY `email` (`email`);
```

```
ALTER TABLE `contract`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `as_number` (`as_number`),
  ADD UNIQUE KEY `vir_number` (`vir_number`);
```

```
ALTER TABLE `contract_history`
  ADD PRIMARY KEY (`id_h`);
```

```
ALTER TABLE `contract`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `contract_history`
  MODIFY `id_h` int(11) NOT NULL AUTO_INCREMENT;
```

6.3.4 Täiendavate testandmete lisamine

```
INSERT INTO `user` (`username`, `firstname`, `lastname`, `company`,
  `country`, `salt`, `password`, `email`, `status`, `registered`,
  `reg_time`, `last_visited`) VALUES
  ('Admin', 'Pavel', 'B.', 'PaShen', 'Estonia', 'ympkoyxpxz',
  '7766951df593136d08bc4d2255a803f8', 'pashen@hot.ee', 'administrator', 1,
  NULL, '2023-01-01 14:03:06'),
  ('Test1', 'Test1', 'Testov1', 'EBM GRUPP AS', 'Estonia', 'zafujbtnig',
  '3efd098e7bd17023bddcca8800976b07', 'test1@hot.ee', 'moderator', 1, '2009-
  01-01 11:46:18', '2023-01-01 22:33:26');
```

```
('Test2', 'Test2', 'Testov2', 'Claire Foods OÜ', 'Estonia', 'zudhvdnkt',
'177db69e82a5e06985b45cf5f07a4070', 'test2@hot.ee', 'user', 1, '2009-01-01
11:51:39', '2023-01-01 10:36:33');
```

```
INSERT INTO `virtual_number` (`name_v`, `vir_id`) VALUES
('virtual', 1);
```

```
INSERT INTO `contract` (`id`, `created`, `username`, `as_number`,
`vir_number`, `date_on`, `date_off`, `client`, `country`, `sales_com`,
`oil_type`, `quantity`, `delivered`, `EEview`, `LVview`, `LTview`) VALUES
(1, '2009-09-18 16:50:55', 'Test1', 'AS90804', NULL, '2009-03-01', '2009-
08-31', 'ETK', 'EE', 'CF', 'Raps', 60600.00, 60600.00, '1', '1', '1'),
(2, '2009-09-18 16:51:56', 'Test1', 'AS90963', NULL, '2009-02-01', '2009-
07-31', 'RIMI', 'EE', 'CF', 'Soya', 17977.00, 17977.00, '1', '1', '1'),
(3, '2009-09-19 13:06:07', 'Tets1', NULL, 1, '2009-11-02', '2009-12-30',
'FREE', 'EE', 'CF', 'Raps', 197800.00, 197800.00, '1', '1', '1');
```

```
INSERT INTO `type_of_package` (`contract_id`, `type_package`,
`purchase_price`, `sales_price`) VALUES
(1, '1L', 0.8160, 0.0000),
(2, '3L', 0.7200, 0.0000),
(2, '5L', 0.7130, 0.0000),
(3, '1L', 0.7134, 0.0000),
(3, '0.9L', 0.6541, 0.0000),
(3, '5L', 0.7134, 0.0000),
(3, '20L', 0.7134, 0.0000);
```

6.4 Andmetöötlus PHP abil

Allpool on näidatud mõned peamised PHP-funktsioonid MySQL-i andmete haldamiseks.

Kõigepealt peame looma ühenduse andmebaasiga.

6.4.1 Funktsioon `mysqli connect`

Faili alguses luuakse ühendus MySQL-i andmebaasiga, kutsudes välja funktsiooni:

```
function OpenCon(){
    $dbhost = "mysqlhost56";
    $dbuser = "la01_98xxxxx";
    $dbpass = "xxxxxxx";
    $db = "la01_98xxxxx";
```



```
$conn = new mysqli($dbhost, $dbuser, $dbpass, $db) or die("Connect
failed: %s\n". $conn -> error);
return $conn;
}
```

Faili lõpus, pärast kõiki andmebaasi pöördumist, suletakse ühendus funktsiooni kutsega:

```
function CloseCon($conn){
    $conn -> close();
}
```

6.4.2 Funktsioon mysqli query and fetch array

Pärast edukat ühenduse loomist andmebaasiga peaks tegema päringu ja kirjutada massiivi, et seda programmi lehel kuvada.

Päringu jaoks kasutame funktsiooni mysqli_query:

```
$contract_query = mysqli_query($conn, "SELECT * FROM contract WHERE
as_number = '$CONTRACT_NAME'");
```

Päringu teisendamiseks ja andmete massiivi kirjutamiseks peaks kasutama funktsiooni mysqli_fetch_array:

```
$contract_array = mysqli_fetch_array($contract_query);
```

Massiivis olevaid vastuvõetud andmeid saab kasutada ja paigutada tabeli kujul programmi lehele. ([vt Joonis 8](#))

Kokkuvõtte

Antud töö eesmärk oli luua PHP-keele ja tasuta MySQL andmebaasi põhine veebisüsteem EBM Group AS tehase laos oleva õli koguse haldamiseks ja kontrollimiseks.

Loodud süsteem kirjeldab lepingute korraldamise, õli koguste tellimise ja kustutamise protsessi lähtudes konkreetsest lepingust.

Süsteem koosneb moodulitest: lepingu loomine, muutmine ja kustutamine ning koguse tellimine lepingut järgides.

Selles süsteemis registreeritakse kasutajad antud rolliõiguste põhjal (administraator, moderaator, kasutaja, vaataja). Rolliõiguste põhjal saab kasutaja sõltuvalt antud rollist või kasutaja tasemest vaadata, luua või muuta olemasolevaid lepinguid ja teha tellimusi.

Andmebaasis kasutati seitset tabelit. Andmebaasi tabelid on kirjeldatud.

Loodud süsteemi funktsionaalsust pidevalt arendati ja täiendati, kuna:

- PHP-keel arenes ja see uuendati versioonilt 5 kuni 8,2-ni;
- Tellija poolt esitati uusi nõudmisi süsteemi lõppvariandiga seotud funktsionaalsusele.

Viimase aasta jooksul on moderaatori ja administraatori moodulit täiendatud nii, et neil on võimalik kõiki tellimusi kinnitada või tühistada. Viimases versioonis on lisandunud eraldi avatava veebiakna funktsioon, kus on võimalik kõiki tellimusi näha ja tühistada. Samuti on lisandunud võimalus tühistatud tellimused kustutada.

Kasutatud kirjandus

- [1] „What Is a Database?“, Oracle, [Võrgumaterjal]. Available: <https://www.oracle.com/ee/database/what-is-database/>. [Kasutatud 12 12 2022].
- [2] eFusion, „11 типов современных баз данных“, Proglib, 08 01 2020. [Võrgumaterjal]. Available: <https://proglib.io/p/11-tipov-sovremennyh-baz-dannyh-kratkie-opisaniya-shemy-i-primery-bd-2020-01-07>. [Kasutatud 15 12 2022].
- [3] „What is a Relational Database?“, Oracle, [Võrgumaterjal]. Available: <https://www.oracle.com/ee/database/what-is-a-relational-database/>. [Kasutatud 20 12 2022].
- [4] „Краткий обзор реляционных систем управления базами данных“, 8host, 11 09 2016. [Võrgumaterjal]. Available: <https://www.8host.com/blog/kratkij-obzor-relyacionnyx-sistem-upravleniya-bazami-dannyx/>. [Kasutatud 21 12 2022].
- [5] „Why MySQL?“, MySQL, [Võrgumaterjal]. Available: <https://www.mysql.com/why-mysql/>. [Kasutatud 10 01 2023].
- [6] „Why MySQL?“, Web Archive, [Võrgumaterjal]. Available: <https://web.archive.org/web/20070515140809/http://www.mysql.com/why-mysql/>. [Kasutatud 10 01 2023].
- [7] „PHP – Mis on PHP?“, Metshein, [Võrgumaterjal]. Available: <https://www.metshein.com/unit/php-mis-php/>. [Kasutatud 17 01 2023].
- [8] I. Stukov, „Кто такой разработчик на PHP“, Skillbox Media, 07 09 2022. [Võrgumaterjal]. Available: <https://skillbox.ru/media/code/kto-takoy-razrabotchik-na-php-tehnologii-navyki-zarplaty-vakansii/>. [Kasutatud 20 01 2023].

- [9] A. Ivaškevičs, „Как работает PHP,“ PHP Zone, 14 01 2017. [Võrgumaterjal]. Available: <https://php.zone/kurs-php-dlya-nachinayushih/kak-rabotaet-php>. [Kasutatud 20 01 2023].
- [10] D. Sokol, „Базы данных,“ hostings.info, [Võrgumaterjal]. Available: <https://ru.hostings.info/schools/bazy-dannyh.html>. [Kasutatud 05 02 2023].
- [11] „Взаимодействие PHP и MySQL. Подключение к базе данных MySQL.,“ Napli, [Võrgumaterjal]. Available: <https://napli.ru/php-mysql-connecting-to-mysql-database/>. [Kasutatud 20 02 2023].
- [12] „Взаимодействие PHP и MySQL,“ Add PHP, [Võrgumaterjal]. Available: https://addphp.ru/materials/mysql/1_4.php. [Kasutatud 05 02 2023].

Lisa 1 - “Kasutusjuhtude kirjeldused”

Kasutusjuht 1

Kasutusjuht: Lepingu koostamine (roll: moderaator)

Tüüp: Reaalne kasutusjuht

Stsenaarium (tüüpiline sündmuste järjestus):

1. Haldur siseneb süsteemisse moderaatori rollis.
2. Haldur peab olema süsteemis, ja süsteem avab lepingute nimekiri (**Joonis 8**) koos lepingute nimega (**J. 8 väli A**) ja nupuga „Create“ (**J. 8 väli B**).
3. Haldur tahab koostada uus leping (vajutab nupule “Create”).
4. Haldur sisestab vajalikud andmeid uues aknas “Create”: (**Joonis 9**)
 - a. Valib mis riigis kasutaja näeb see tellimus (EE, LV, LT);
 - b. Kui koostatakse reaalne leping, siis tuleb sisestada lepingu kood algusega “AS” (nt. AS123456); (**J. 9 väli A**)
 - i. Kui koostatakse virtuaalne leping, siis lahtris “Contract number” pole vaja midagi kirjutada, ja automaatselt süsteem genereerib järgmine virtuaalne lepingu number “EBM-#” (n.t EBM-9);
 - c. Lepingu algus ja lõpp aeg (lõpp aeg ei tohi olla vähem kui algus aeg)
 - d. Mille kliendi jaoks see leping mõeldud “Client”;
 - e. Kuhu toimetatakse tellimused “Country”;
 - f. Mis firmaga see leping seotud “Sales Comment”;
 - g. Tüübi õli sellega lepingu seotud “Oil Type”;
 - h. Õli kogus lepingus “Quantity”;
 - i. Pakendi tüüp “T. of P.”;
 - i. Kui lepinguga seotud mõned pakendi tüüpe, siis saab lisada veel uus pakendi tüüp vajutades nupule “+”;
 - j. Ostuhind selle pakile “Pur. Price”;
5. Haldur vajutab nupule “Enter”; (**J. 9 väli B**)
6. Süsteem kontrollib, kas kõik andmed on sisestatud õigesti, siis programm kirjeldab kõik andmed MySQL tabelitesse.
 - a. Kui süsteem leidis viga, siis teatab haldurit konkreetsest veast.
 - b. Süsteem jätab kõik sisestatud andmed lahdrides.
 - c. See tegevus kirjeldatakse lepingu ajaloos.
7. Kui haldur otsustab katkestada koostada tellimust, siis vajutab nupule “Reset”; (**J. 9 väli C**)
 - a. Programm kustutab sisestatud andmed lahtrist ära.
 - b. Tuleb sulgeda aken “Create”.

Kasutusjuht 2

Kasutusjuht: Tellimuse koostamine / mahakandmine lepingust (roll: moderaator, user)

Tüüp: Reaalne kasutusjuht

Stsenaarium (tüüpiline sündmuste järjestus):

1. Haldur / klient siseneb süsteemisse.
2. Haldur / klient peab olema süsteemis, ja temale määratud roll (moderaator / user)
 - a. Süsteem avab lepingute nimekiri (**Joonis 8**) koos lepingute nimega.
3. Haldur / klient valib nimekirjast leping kust tahab teha mahakandmist / tellimus teha.
 - a. Vajutab kaks korda reaalne või virtuaalne lepingu nimele. (**J. 8 väli A**)

- b. Programm avab veebiaken “Delivery” (**Joonis 10.**)
- c. Programm kuvab õli jääkkogus lepingus (**J. 10 väli A**) ja vajalikud lahtrid sisestamiseks.
- 4. Haldur / klient sisestab:
 - a. Kogus, kui palju on vaja maha kanda / tellida;
 - b. Kommentaar, kuhu (mis firmale) see tellimus tuleb ja mis pakend (nt. CL: 1 pal x 1L);
 - c. Mahakandmise kood (Inv. Number);
 - d. Mahakandmise kuupäev (vaikimisi: tänane kuupäev)
- 5. Kui kõik andmed sisestatud haldur / klient vajutab nupule “Enter” (**J. 10 väli B**)
 - a. Süsteem kontrollib, kas kõik andmed on õiged.
 - b. Kui vajutab haldur (rollis moderaator), siis kohe sisestatud kogus maha kantud lepingust ja kirjeldatakse see tegevus ajaloos kirja pandud (staatus: Delivered).
 - c. Kui vajutab kasutaja (rollis user), siis kohe sisestatud kogus maha kantud lepingust ja kirjeldatakse see tegevus ajaloos kirja pandud (staatus: Ordered). See tellimus kuvab süsteemis ja näeb see tellimus haldur (rollis moderaator) (**J. 8 väli C**)
 - d. Kasutaja näeb number kui palju tehtud uut tellimust. (**J. 8 väli C**)
- 6. Haldur / klient otsustab katkestada teha mahakandmist / tellimust, siis vajutab nupule “Cancel”. (**J. 10 väli C**)
 - a. Program ei salvesta midagi ja sulgub veebiaken “Delivery”.

Kasutusjuht 3

Kasutusjuht: Lepingu muutmine / kustutamine (ainult administraator või moderaator rollis)

Tüüp: Reaalne kasutusjuht

Stsenaarium (tüüpiline sündmuste järjestus):

1. Haldur siseneb süsteemisse moderaatori rollis.
2. Haldur peab olema süsteemis, ja süsteem avab lepingute nimekiri (**Joonis 8**) koos lepingute nimega.
3. Haldur valib nimekirjast leping kus tahab teha väike parandus või teha mahakandmist teisele lepingule.
 - a. Vajutab kaks korda reaalne või virtuaalne lepingu nimele.
 - b. Programm avab veebiaken “Delivery” (**Joonis 10**)
 - c. Kui haldur paneb linnuke “Minus Delivered” (**J. 10 väli E**), saab teha parandus, kui mingi viga oli mahakandmist lepingust, lisab ka kommentaar.
 - d. Saab teha mahakandmine kogus teisele lepingule.
 - i. Reaalsest lepingust reaalsele lepingule peale saab (kus on samad õli tüüp);
 - ii. Virtuaalselt lepingust virtuaalsele peale saab;
 - iii. Virtuaalselt lepingust reaalsele lepingule peale saab;
 - iv. Reaalsest lepingust virtuaalsele peale ei saa;
4. Haldur otsustab parandada kogu leping.
 - a. Haldur peab olema veebiaknal “Delivery”. (**Joonis 10**)
 - b. Vajutab nupule “Change/Delete” (**J. 10 väli D**)
 - c. Programm avab veebiaken “Change Contract” (**Joonis 11**)
 - i. Parandab andmed lepingus ja vajutab “Enter” (**J. 11 väli A**)
 - ii. Kui tahab teha leping nulliks, siis valib linnuke “...balanced” ja vajutab nupule “Balance” (**J. 11 väli B**)

- iii. Kui tahab lepingut kustutada, siis valib linnuke "... deleted" ja vajutab nupule "Delete"; (**J. 11 väli C**)
5. Süsteem kontrollib kõik andmed ja muudab või kustutab leping.

Kasutusjuht 4

Kasutusjuht: Tellimuse kinnitamine / vaatamine (roll: moderaator, user)

Tüüp: Reaalne kasutusjuht

Stsenaarium (tüüpiline sündmuste järjestus):

Haldur / klient siseneb süsteemisse.

Haldur / klient peab olema süsteemis, ja temale määratud roll (moderaator / user)

Haldur tahab vaadata uued tellimused;

- a. Haldur vajutab lingile "Orders (#)", kus sulgudes uue tellimuse kogus;
- b. Programm avab veebiaken "Orders" (**Joonis 12**), kus nimekirjas saab tellimus kinnitada "Agree" (**J. 12 väli C**) või mitte kinnitada "Canceled".
- c. Kui moderaator märkab tellimuses viga, lisab ta kommentaari tellimusele (**J. 12 väli A**) ja vajutab "Cancel (#)" nuppu (**J. 12 väli B**), misjärel programm salvestab uus kommentaar. Õli kogus, mis oli varem maha arvatud, tagastatakse tagasi lepingusse.

Klient tahab vaadata oma tellimused;

- d. Klient vajutab lingile "Ordered (#)", kus sulgudes tellitud tellimuse kogus;
- e. Programm avab veebiaken "Ordered" (**Joonis 13**), kus nimekirjas saab vaadata oma mitte kinnitatud-, tellitud- (**J. 13 väli B**) ja viimased kinnitatud tellimused (**J. 13 väli C**);
- f. Kui tellimuse mitte kinnitatud, saab see tellimus kustutada (**J. 13 väli A**).
- g. Süsteem kontrollib kõik andmed ja kirjeldab sündmus ajaloos.