LAHDEN AMMATTIKORKEAKOULU
Lahti University of Applied Sciences

VARASTON KIRJANPIDON TIETOJÄRJESTELMÄ

Keraamisen laatan tilaustenkäsittely

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka
Opinnäytetyö
Syksy 2012
Viktor Alekseev

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

ALEKSEEV, VIKTOR:

Varaston kirjanpidon tietojärjestelmä
Keraamisen laatan tilaustenkäsittely

Ohjelmistotekniikan opinnäytetyö, 60 sivua

Syksy 2012

TIIVISTELMÄ

Tässä opinnäytetyössä keskitytään web-pohjaisen tietojärjestelmän suunnitteluun ja toteuttamiseen. Opinnäytetyön tavoitteena oli toteuttaa web-sovelluksen muodossa tietojärjestelmä, joka olisi tarkoitettu keraamisen laatan kirjanpitoon ja tilaustenkäsittelyyn varastossa. Kirjanpidolla tarkoitetaan tässä sitä, että järjestelmässä olisi tietoa asiakkaista ja heidän tekemistä tilauksista. Toisin sanoen, tietojärjestelmän olisi tarkoitus helpottaa varastotyötä yrityksen varastossa. Opinnäytetyö tehdään organisaatiolle, joka toimii rakennussuunnittelun alalla.

Opinnäytetyön teoriaosuudessa pyritään selvittämään, mikä on tietojärjestelmän merkitys ja tarkoitus. Teoriaosuudessa pohditaan tiedonhallintaa ja selvitetään tietojärjestelmän kehittämisen keinot, joilla saavutetaan tehokkuus järjestelmien kehitystyössä. Lisäksi teoriaosuudessa mainitaan järjestelmien ongelmakohdat.

Tietojärjestelmän toimintaympäristö on tärkeä asia tietojärjestelmän kehitystyössä, sillä se voi asettaa omat vaatimuksensa tulevalle järjestelmälle. Teoriaosuudessa pohditaan sitä, miten varasto toimintaympäristönä vaikuttaa kehitettävään tietojärjestelmään ja sen toiminnallisuuteen.

Teoriaosuudessa käsitellään myös web-sivujen kehittämisen tekniikoita, joita käytettiin tietojärjestelmän toteutuksessa. Kyseessä on pääasiassa PHP-ohjelmointikielen sekä MySQL-tietokannan käyttö. Tämän opinnäytetyön tapauksessa ei ole keskeistä eroa tietojärjestelmän ja web-sivujen toteutuksen välillä, sillä työssä tehdään web-sovellus. Sovelluksen tarkoitus on tietojen käsittelyn avulla helpottaa toimintaa, minkä takia sitä sanotaan tietojärjestelmäksi.

Toteutusosassa selvitetään toteuttavan sivuston tavoite, toiminnallisuus sekä sivutaan ohjelmointia käytetyillä ohjelmointitekniikoilla. Toteutusosassa kerrotaan, mitä voidaan tehdä tietojärjestelmän avulla. Myöhemmin, tulososassa arvioidaan saatu sivusto käyttömukavuuden ja puuttuvien asioiden näkökulmasta.

Opinnäytetyön tuloksena saadaan tietojärjestelmä, joka tarjoaa perustoimintoja tilausten kirjanpitoa varten. Vaativaa käyttöä varten tietojärjestelmä tarvitsisi jatkokehitystä. Tämän takia web-sovelluksen toiminnallisuuteen kohdistuu eniten huomiota, kun web-sovellusta aletaan jatkokehittää.

Avainsanat: www-sivut, tietojärjestelmä, tiedonhallintajärjestelmä

Lahti University of Applied Sciences
Degree Programme in Information Technology

ALEKSEEV, VIKTOR:

Information System for Warehouse Accounting
Ceramic Tile's order processing

Bachelor's Thesis in Software Engineering, 60 pages

Autumn 2012

ABSTRACT

This Bachelor's Thesis is the last part of Software Engineering studies of the degree programme in Information Technology. The subject of the thesis is the designing and creation of an information system which is intended primarily for ceramic tile order processing and warehouse accounting. The warehouse accounting in this system contains information about clients and their orders. The system is made in the form of Internet pages, so the subject can also be classified as the creation of the Web application which has the specific purpose to make the work in the warehouse easier. The main objective of the practical part of this thesis is the developing of this information system, or in other words the Web application.

The theoretical part of the thesis contains information about what information systems are, the principles of Web developing, and the programming technologies which were used in this work. These technologies are primarily MySQL-database and PHP-programming language. The theoretical part also deals with the operation of a warehouse and the importance of information.

The information system was developed for an organization which works in the field of architectural designing and civil engineering. The first functioning version of the information system, which provides basic functions, was developed during the practical part of the Bachelor's Thesis. The result of the practical part is the Web-based information system which can be used for basic ceramic tile order processing and warehouse accounting. For more demanding uses, it should be developed further.

On the level of functionality, the developed Web application is not multifunctional; the level of functionality of the first version can be described by the designer as average. A more detailed estimate of the result, possible problems and missing functions are given in the conclusion part of the thesis. The practical part of the thesis also contains information about the functionality of the developed Web site and the Web technologies which were used during the development of this Web application.

Key words: Web pages, information system, information management system

SISÄLLYS

1 JOHDANTO

Nykyään yrityksen toiminnassa tuotetaan yhä enemmän tekstejä, asiakirjoja ja dokumentteja. Yleensä kaikkien näiden tietojen hallinta voi tuntua vaikealta tai aikaavievältä. Tietojen hallintaan, käsittelyyn tai ylläpitämiseen on olemassa monia erilaisia ohjelmistoratkaisuja. Ratkaisujen tarkoitus, toimintaperiatteet ja soveltuvuus eri tarpeisiin voivat vaihdella kuitenkin huomattavasti. (Anttila 2001, 1 - 7.)

Karjalan tasavallan pääkaupingissa Petroiskoissa sijaitsee SeverStroiProekt-organisaatio, jolle kehitetään keraamisen laatan tilaustenkäsittelyyn tarkoitettu tietojärjestelmä. Organisaation toimiala keskittyy rakennustöihin ja käsittää siihen liityvät suunnittelu- ja insinöörityöt. Yritys tarjoaa laajat insinööripalvelut, kuten geotekniset tutkimukset, arkkitehtuuriratkaisut, suunnitteludokumentaatiot, insinööriverkostot ja rakennuspalvelut. (SeverStroiProekt 2012.)

Yrityksellä on käytössä varasto, jossa on keraamisen laatan varastotehtäviä. Aikaisemmin varastossa ei käytetty web-sovellusta tai tietojärjestelmää laatan tilaustenkäsittelyyn, sillä myynti ei ole yrityksen päätoimiala eikä keraaminen laatta ole tärkein tavara verrattuna muihin materiaaleihin, joiden kirjanpido hoidetaan ohjelmilla.

Tämän opinnäytetyön tarkoituksena on suunnitella ja toteuttaa tietojärjestelmä web-sovelluksena, jolla pystytään suorittamaan joukko kirjapidon tehtäviä organisaation varastossa. Kehitettävä web-sovellus erikoistuu varastossa olevan keraamisen laatan tilaustenkäsittelyyn. Tavoitteena on luoda sovellus, joka sisältää tietoja asiakkaista ja heidän tilauksista järkevässä muodossa sekä tiedon siitä, missä tilassa asiakkaan tilaama tuote on. Lisäksi sovelluksessa toteutetaan toiminnallisuus, jolla pystytään erottamaan hyväkuntoinen tuote hylkytavarasta.

Työn tutkimusongelma on toiminnallisuuden kehityksen lisäksi pyrkimys luoda web-sovellus, jota olisi helppo käyttää. Kyseisen sovelluksen käytännöllinen merkitys yleisesti on yllä kuvattujen toimintojen suorittamisen lisäksi parantaa yrityk-

sen varaston toiminnan tehokkuutta nostamalla käytettävissä olevan tiedon käsittelyä.

Kehitettävää web-sovellusta voidaan kutsua tietojärjestelmäksi, koska sovelluksen päämerkitys on tietojenkäsittelyn parantaminen siinä paikassa, missä tätä web-sovellusta käytetään. Opinnäytetyön työosuudessa käsitellään kehitetyn tietojärjestelmän toimintaperiatteet ja web-teknologiat, joiden avulla kyseinen web-sovellus tehtiin. Lisäksi kuvataan tietokantaa, käyttöliittymää ja web-sovelluksen lähdekoodin osia. Opinnäytetyön teoriaosuudessa pohditaan tietojärjestelmän merkitystä, varastoinnin ongelmakohtia, tietojen hallintaa ja käytettyjen web-tekniikoiden toimintaperiaatteita.

Opinnäytetyö rajataan niin, että työssä käsitellään sovelluksen ensimmäistä versiota, jossa on tehty toiminnallisuus tilaustenkäsittelyä ja kirjanpitoa varten. Toisin sanoen, mahdolliset epäonnistuneet tai puutteelliset asiat siirtyvät jatkokehitykseen, josta kerrotaan yhteenveto-osiossa, missä myös pohditaan sitä, kuinka hyvin päästiin tavoitteeseen. Tietojärjestelmä toteutetaan eri kielellä kuin suomi. Tässä opinnäytetyössä esitettyjen lähdekoodin osien ja käyttöliittymän ikkunoiden vieraskieliset sanat on korvattu suomenkielisillä vastineilla.

2 TIEDON MERKITYS JA TIEDONHALLINTA

Tämän opinnäytetyön työosuudessa kehitettävän tietojärjestelmän päätarkoitus on tietojen käsittely ja säilyttäminen. Nämä asiat ovat tärkeitä, koska nykyorganisaatioissa tieto on yksi tärkeämpiä ja samalla heikommin hyödynnettyjä resursseja. Jotta yritykset pystyisivät käyttämään tietoja mahdollisimman tehokkaasti, yritysten tiedonhallinta vaatii muutoksia lähes koko ajan. (Kaario & Peltola 2008, takakansi.)

2.1 Tiedon merkitys

Tiedon käsite on hyvin monimuotoinen ja moniselitteinen. Yrityksiin liittyen tietoa voidaan pitää tuottavuuden tekijänä ja tiedonhallintaa organisaation toiminnan tukena. Tietoa voidaan sanoa yrityksen tärkeäksi voimavaraksi. Tiedon avulla voidaan yrityksen toimintaa kehittää ja tehostaa. Joskus tietoa voidaan pitää kauppatavarana, jolle voidaan usein antaa rahallinen arvo. (Kaario & Peltola 2008, 4.)

Riippuen siitä, kuinka tulkitaan tiedon käsitettä, myös tiedonhallintaa voidaan määritellä eri asioiksi sen mukaan, kuinka ymmärretään tiedon käsitettä. Vähintään seuraavat tulkinnat ovat mahdollisia: tiedonhallinta voidaan käsittää tietokantojen hallinnaksi, liiketoimintatiedon hallinnaksi tai organisaatioon liittyvän tiedon hallinnaksi. (Kaario & Peltola 2008, 3 - 6.)

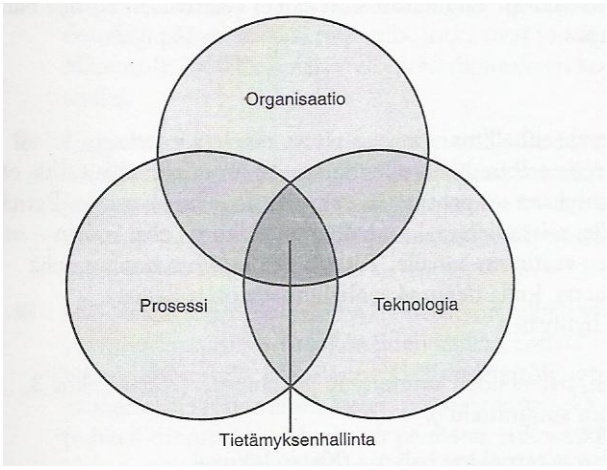
Opinnäytetyön tietojärjestelmän on tarkoitus sisältää tietoja varastossa olevista tilauksista, mikä on yhteydessä yrityksen toimintaan, muun muassa siihen, jolla yritys tienaa rahansa. Kehitettävän tietojärjestelmän rooli riippuu siitä, kuinka paljon siihen luotetaan ja sitä käytetään yrityksen toiminnassa, sekä siitä, kuinka paljon tietojärjestelmää kehitetään eteenpäin. Tämä tarkoittaa sitä, että tulevaisuudessa tietojärjestelmä voi sisältää paljon enemmän arvokasta tietoa, joka liittyy yrityksen toimintaan. Lisäksi voidaan mainita, että kehitettävä tietojärjestelmä erikoistuu varastossa olevan tavaran käsittelyyn. Tämän perusteella voidaan tehdä päätös, että tämän työn tapauksessa tiedonhallinta voidaan tulkita liiketoimintatiedon hallinnaksi.

Yrityksen toiminnan tehokkuuden kannalta on ymmärrettävä tiedonhallinnan merkitys. On tärkeää tunnistaa organisaation toiminnan kannalta tärkein tieto ja sen elinkaari. Lisäksi tiedon siirtyessä mahdollisimman tehokkaasti tietolähteestä tiedon saajalle tehostuu myös yrityksen suorituskyky itsestään. Tiedon muotoja on erilaisia, mutta juuri tekstimuotoisen tiedon hallinta korostuu tietotekniikan kasvaessa, koska tiedon kuvaaminen on melkein aina tekstimuotoisen sisällön hallintaa. (Kaario & Peltola 2008, 8.)

Globaalisesti puhuen, tänä päivänä tiedosta on tullut yrityksille entistä tärkeämpää pääomaa ja nykypäiväistä yhteiskuntaa voidaan sanoa tietoyhteiskunnaksi. Tietoja on tänään paljon enemmän kuin koskaan aikaisemmin, ja tiedon määrä jatkaa kasvuaan kiihtyvällä vauhdilla. Hyvin usein tiedonhankinnassa käytetään yhä enemmän erilaisia tietojärjestelmiä. (Koskela, Koskinen & Lankinen 2007, 64.)

2.2 Tietämyksenhallinta

Tietämyksenhallinta on keino, jolla saadaan liiketoimintaa koskevia tietoja yhteen eri lähteistä, muun muassa työntekijöilta, tietokannoista, paperilta, verkkopalveluista. Tietämyksenhallinnan avulla tarvittava ja oikea tieto saadaan oikeille ihmisille oikeaan aikaan, mikä antaa näille ihmisille välineet käsitellä ja analysoida nämä tiedot. Tieto myös muokataan siellä, missä se sijaitsee. Tämä puolestaan nopeuttaa ja tehostaa toimintaa muun muassa organisaatioissa. Tietämyksenhallinnassa on kyseessä informaation jakelu vähintään työntekijöiden ja liiketoimintaprosessien kesken. Tällä tavalla saadaan ihmiset ja liiketoiminta menestymään (Honeycutt 2001, xiii). Kuviossa 1 on esitetty tärkeimmät asiat, joita yritykset käsittelevät tietämyksenhallintaratkaisun toteutuksessa.



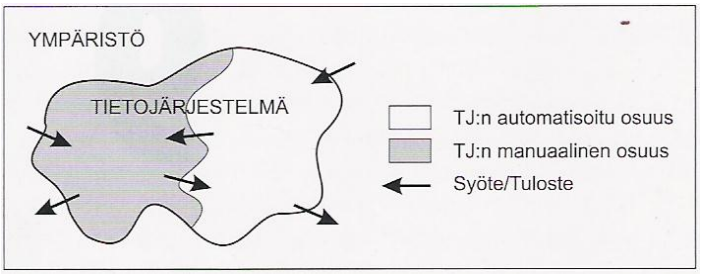
KUVIO 1. Tietämyksenhallinnan osa-alueet (Honeycutt 2001, xv)

Kuvio 1 osoittaa, että tietämyksenhallinnalla on keskeinen rooli organisaation, sen prosessien ja teknologian välillä. Parantamalla yhtä niistä voidaan tehostaa tietämyksenhallintaa. Toisaalta, tietämyksenhallintaan kuuluvat sekä liiketoimintaprosessit että teknologia, jolla voidaan hoitaa liiketoimintaprosesseja.

2.3 Tietojärjestelmät

Tietojärjestelmä on toimintaa palveleva, toiminnan toteuttava tai toimintaa helpottava kokonaisuus tai järjestelmä, joka voi vähintään koostua tiedoista, laitteista, ohjelmista sekä näitä käyttävistä ihmisistä. Tietojärjestelmän tarkoitus on siis helpottaa toimintaa tietojen käsittelyn avulla. Tietojärjestelmän käsite on osittain päällekkäinen ohjelmistojen kanssa, mutta kuitenkin sitä suurempi. Tietojärjestelmän käsite voi kattaa sellaiset organisaation asiat, kuten käyttäjäroolit, käyttöoikeudet ja vastuualueet. (Paananen 2005, 338.)

Tietojenkäsittely tietojärjestelmässä voi olla manuaalista, jolloin ihmiset hoitavat tietojenkäsittelyä, tai automaattista, jolloin tietoja voi käsitellä tietotekniikka ja sen ohjelmat. Nykyään tietojärjestelmissä on yleensä sekä manuaalisia että automaattisia osia. Tällöin molemmilla osilla on rajapinnat siihen ympäristöön, jota järjestelmä palvelee (KUVIO 2). Rajapinta tässä tapauksessa määrittelee sen, mitä syötteitä järjestelmä voi vastaanottaa ympäristöstään ja mitä tulosteita järjestelmä antaa. (Paananen 2005, 338.)



KUVIO 2. Tietojärjestelmän ja ympäristön vuorovaikutus (Paananen 2005, 338)

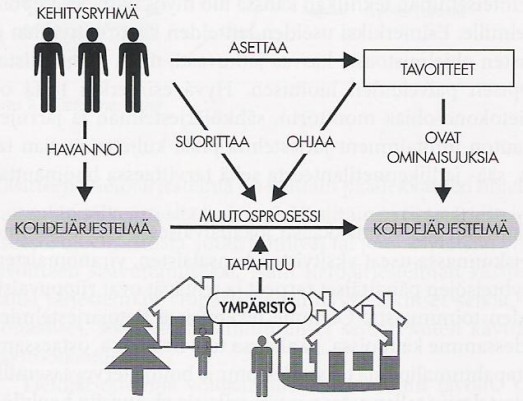
Kuviosta 2 nähdään, että ympäristöllä on tärkeä rooli tietojärjestelmän toiminnassa. Toimintaympäristö paitsi vastaanottaa järjestelmän tulosteet, myös määrittää tietojärjestelmän syötteet, jotka voidaan käsitellä järjestelmässä sekä manuaalisesti että automaattisesti.

Tietojärjestelmät voidaan luokitella monella eri tavalla, mutta tämän opinnäytetyön tapauksessa kannattaa mainita tapahtumankäsittelytietojärjestelmät. Näillä tietojärjestelmillä suoritetaan perinteisiä kaupallisia tehtäviä, kuten osto- ja myyntitilausten käsittelyä, varaston hallintaa, palkanlaskentaa tai vaikka opintosuoritusrekisterin ylläpitoa. Lisäksi automaattisen tietojenkäsittelyn laajentuminen uusille sovellusaloille tuo mukanaan uudentyyppisiä tietojärjestelmiä. (Paananen 2005, 339.)

2.4 Tietojärjestelmien ongelmallisuus

Tietojärjestelmän kehittämisessä kehitystyön päätavoite on joko uuden tietojenkäsittelytoimenpiteen mahdollistaminen tai jo olemassa olevien toimintojen tehostaminen. Keskeistä kehitystyössä on ymmärtää kehitettävän toiminnan inhimilliset yhteydet. Toiminnan kehittämisen täytyy kohdistua joko ihmisiin, toimintoihin tai tekniikkaan, koska tietojenkäsittely perustuu ihmisen tekniikan avulla suorittamiin toimenpiteisiin. Vaikka nykyisissä tietojärjestelmien kehityshankkeissa kehitetäänkin yleensä tekniikkaa, tietojärjestelmien kehitys vaikuttaa kaikkiin kolmeen edellä mainittuun kohteeseen: tekniikan lisäksi järjestelmää käyttäviin ihmisiin ja tietojenkäsittelytoimintoihin. Tietojärjestelmän kehittämistä (KUVIO 3) sanotaan

systeemityöksi. Systeemityö on järjestelmän kehittäjien tietyssä toimintaympäristössä tekemä kohdejärjestelmän muutosprosessi, joka suoritetaan asetettujen tavoitteiden mukaisesti. (Paananen 2005, 340 - 341.)



KUVIO 3. Tietojärjestelmän kehitys (Paananen 2012, 340)

Kuvio 3 osoittaa, että tietojärjestelmän kehitysryhmän on kehitystyössä selvitettävä tavoitteet ja omat toimenpiteet. Lisäksi kehitysryhmän on otettava huomioon myös järjestelmän toimintaympäristö.

Tietojärjestelmien suunnittelu ja kehitys nykyaikana on vaikea ja haastava työ. Perinteiset syyt tähän ovat järjestelmien suuri koko ja monimutkaisuus, järjestelmän abstrakti luonne, integrointi ja erilaiset toimintaympäristöt sekä kustannus- ja aikapaineet. Lisäksi tietojärjestelmien kehitystä häiritsee joukko muita toistuvia ongelmia. Yksi niistä on se, että ohjelmistoalalla hankkeiden epäonnistuminen on yleinen asia: merkittävä osa hankkeista ei valmistu koskaan tai ylittää niille varatut aika- tai taloudelliset resurssit. Toiseksi ongelmaksi tietojärjestelmien kehityksessä voidaan todeta virheet ja puutteet, joihin asiakas tai käyttäjä törmää. Tietojärjestelmät sisältävät usein paljon virheitä vielä toimitusvaiheessa. Lisäksi usein valmis tietojärjestelmä ei vastaa asiakasvaatimuksia tai järjestelmää ei voida ottaa käyttöön jostain syystä. Vielä yhdeksi ongelmaksi tietojärjestelmien kehittämisessä voidaan katsoa tietojärjestelmän ylläpito. Suurin osa järjestelmien kehittämistyöstä on vanhojen järjestelmien ylläpitoa, korjausta ja jatkokehitystä. Korjaukset järjestelmään ovat sitä kalliimpia, mitä aikaisemmassa kehitystyön vaiheessa teh-

työ virhettä korjataan, koska myös myöhemmät vaiheet joudutaan usein uusimaan. (Paananen 2005, 341.)

Tietojärjestelmän kehitystyössä pitää huolehtia myös sen helppokäyttöisyydestä. Järjestelmän huono käytettävyys on yksi suurimmista esteistä, joka voi häiritä tietämyksenhallintaa. Vaikeasti käytettävä järjestelmä aiheuttaa epäonnistumisen sen käytössä. Tietämyksenhallintaratkaisun, mukaan lukien tietojärjestelmän, täytyy tarjota ihmisille helppo pääsy tarvittaviin tietoihin, silloin kun tätä tarvitaan. Muuten ihmiset eivät käytä mitään järjestelmää. (Honeycutt 2001, 23.)

Tietojärjestelmän kehityksessä ei voida huomata kaikkia järjestelmään liittyviä tekijöitä. Jopa asiakkaan asettamat vaatimukset, joiden pohjalta tietojärjestelmä kehitetään, voivat nousta ongelmallisiksi silloin, kun ne ovat ristiriidassa keskenään tai tulevat esiin, kun järjestelmän kehitys on jo alkanut. Mitä myöhemmin uusi vaatimus nousee esiin, sitä vaikeammaksi sen toteutus tulee, sillä se voi johtaa koko järjestelmän tai sen osien uudelleen suunnitteluun. (Paananen 2005, 341.)

2.5 Tietojärjestelmän kehitys

Tietojärjestelmän kehittäminen on työtä, jossa tehtäväkokonaisuudet seuraavat toisiaan. Toisin sanoen, tietojärjestelmä toteutetaan yleensä vaiheittain. Yleensä edellisen vaiheen tulos on seuraavan vaiheen syöte. Ensimmäinen vaihe on esitutkimus, jossa selvitetään, miksi ja mitä varten tietojärjestelmä pitää kehittää ja mikä on sen tarkoitus ja tavoite. Toisessa vaiheessa, järjestelmäanalyysissä, selvitetään, mitä järjestelmän tulee tehdä ja mikä on sen toiminnallisuus. Tässä vaiheessa analysoidaan tietojärjestelmän käsittelemät tiedot, yhteykset ympäristöön sekä muun muassa käyttäjät. Järjestelmäanalyysin jälkeen tulee suunnitteluvaihe, joka jaetaan kahteen osaan: arkkitehtuuri- ja moduulisuunnitteluun. Arkkitehtuurisuunnittelussa järjestelmä jaetaan pieniin osiin, eli moduuleihin, joille määritellään rajapinnat. Moduulisuunnittelussa jokaiselle moduulille suunnitellaan rakenne. Tietojärjestelmän toteutusvaiheessa yllä mainitut moduulit toteutetaan ohjelmointikielellä ja yhdistetään toimivaksi järjestelmäksi. Suunnittelijan riittävä tietämys järjestelmien toteuttamisesta ja toteuttajan ammattitaito ovat edellytyksiä järjes-

telmän kehitystyön hyvälle tulokselle. Muita kehitysvaiheita ovat testausvaihe ja sitä seuraava käyttöönottovaihe. (Paananen 2005, 345.)

Tietojärjestelmien kehityksessä voidaan käyttää niin sanottuja CASE-välineitä. CASE (englanniksi Computer Aided Software Engineering) tarkoittaa tietokoneavusteista systeemityötä ja CASE-välineet ovat tässä systeemityössä käytettäviä työkaluohjelmistoja. Nämä työkaluohjelmistot tukevat muun muassa tietojärjestelmän kehitystyön määrittely- ja suunnitteluvaiheita ja liittävät erilaiset menetelmät ja välineet yhdeksi kehitysympäristöksi. Yksi esimerkki CASE-välineistä on kaavioiden piirtoon tarkoitettu editori tai jokin lähdekoodin editori. (Paananen 2005, 352.)

Jotta monimutkaisen tietojärjestelmän suunnittelu olisi helpompaa, sen kehityksessä voidaan käyttää kuvaus- tai mallinnustapoja, joiden avulla voidaan keskittyä järjestelmän yksityiskohtiin. Tällöin järjestelmä voidaan mallintaa eri tarkkuustasoilla ja eri näkökulmista projektin eri vaihessa (Häkkinen 2012, 2 - 3). Kuvaus- tai mallinnustapa muodostaa kielen, jolla voidaan kuvailla jotakin asiaa. Esimerkkejä näistä kielistä ovat luokkakaaviot tai tietovirtakaavio (Paananen 2005, 350). Tietovuokaaviossa voidaan kuvata järjestelmän tai sen osien tehtävää tehdä lähtötiedoista tulostietoja. Tietovuokaaviossa siis kuvataan, että järjestelmään tai sen osiin tulee ulkopuolista tietoa ja että se tuottaa tietoa ulkopuolelle. Ulkopuolena voi olla esimerkiksi käyttäjä tai joku muu olio. Osiinjako tietovuokaaviossa antaa tarkemman mallin järjestelmästä, yhdessä osat suorittavat kokonaisjärjestelmän tehtävän. Yleiskuvan järjestelmästä antaa järjestelmän ylimmän tason malli. Ylemmällä tasolla järjestelmä on yhtenäinen prosessi. (Laine 2008, 2 - 6.)

3 VARASTOTYÖN YKSITYISKOHDAT

Tässä luvussa selvitetään opinnäytetyössä toteutetun tietojärjestelmän toimintaympäristöä. Kuten aikaisemmin saatiin selville, tietojärjestelmä on suorassa vuorovaikutuksessa ympäristönsä kanssa, mikä vaikuttaa tietojärjestelmän tarkoitukseen ja toiminnallisuuteen. Opinnäytetyössä tehty web-sovellus on tarkoitettu käyttöön pääasiassa varastossa. Seuraavaksi selvitetään, mihin varastotyöskentelyssä voidaan kiinnittää erityistä huomiota ja mihin varastonhoidon osa-alueisiin toteuttama tietojärjestelmä liittyy.

3.1 Varastonhoitajan rooli ja varaston tyyppi

Varastonhoitajan tavallisimpiin tehtäviin kuuluu varmistaa kirjanpidon avulla, etteivät varastossa olevat tilaukset tai tuotteet vanhene tai vahingoitu. Lisäksi varastonhoitajan on osattava suorittaa hyvin tavaran vastaanottoa ja säilytystä koskevat toimenpiteet, jotta varaston toiminta olisi mahdollisimman tehokasta. (Hokkanen & Virtanen 2012, 10 - 15.)

Tavaran tai tuotteen vastaanotto, säilytys sekä luovuttaminen asiakkaalle ovat kaikkien varastojen perustoimintoja. Varaston tyyppi voi kuitenkin vaikuttaa tietojärjestelmän toiminnallisuuteen, riippumatta siitä, kuinka monipuolinen toiminnallisuus tietojärjestelmällä on. Varastojen tyyppejä on erilaisia: valmistuotevarasto, keskusvarasto, aluevarasto, jakeluvarasto, terminaali, manuaalivarasto, läpivirtausvarasto. (Hokkanen & Virtanen 2012, 20 - 23.)

3.2 Tuotteen käsittely varastossa

Tuotteiden saapuessa varastoon kiinnitetään huomiota tuotteiden määrään (kappalemääriin) ja kuntoon sekä varmistetaan, että saapuneiden tuotteiden laatu ja määrä vastaavat lähetyslistassa ilmoitettuja. Lisäksi tuotteiden hyllytykseen liittyvät seuraavat asiat: tietojen tallennus järjestelmään, hyllypaikkojen haku järjestelmästä, tuotteen tunnistus hyllytyksen avulla, tuotteiden säilytyskunnon toteaminen,

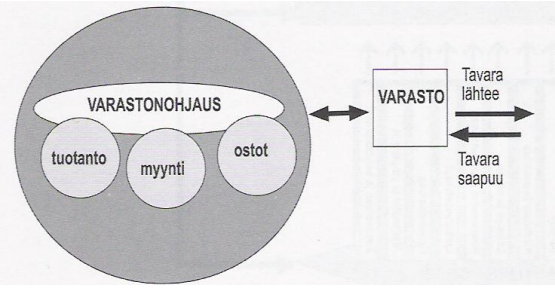
tuotteiden uudelleen sijoittelu, yleinen hyllytyksen tarkastus, hyllytyypit. (Hokkanen & Virtanen 2012, 33.)

Joskus tulee tarve säilyttää tuotteita puolivalmiina (esimerkiksi ilman pakkausta), jolloin näitä tuotteita voidaan muokata asiakaskohtaiseksi, muun muassa muuttaa pakkausmateriaalia. Tuotteen ”puolivalmiuden” takia tuote voidaan joskus säilyttää jossain muualla kuin varsinaisessa varastossa. Lisäksi tällainen tuote saattaa olla kirjanpidossa, mutta voi myös puuttua siitä. Tällaisten tuotteiden säilyttäminen varastossa voi tuoda haasteita tuotteen löytämiselle sekä muutenkin kirjanpidon suorittamiselle. (Hokkanen & Virtanen 2012, 19.)

3.3 Varastotoimintojen ohjaus

Koska tulevaisuudessa varastoinnin merkitys korostuu entistä enemmän (nykyaikana muihin yrityksen toimintoihin kohdistetaan enemmän huomiota kuin varastointiin), varastoinnin tehokkuuden ja tuottavuuden kehittämisessä joudutaan panostamaan erilaisiin tieto- ja ohjausjärjestelmiin ja tietovarastoihin. Varastoinnin merkitystä lisää ajatus siitä, että tilaukset lähetetään yleensä yrityksen varastosta, mikä tarkoittaa, että tällä tavalla varastolla on keskeinen rooli yrityksen ja asiakkaan välillä. Varaston keskeisiin tehtäviin kuuluvatkin asiakastilausten käsittely kustannustehokkaasti ja ajankohtaisesti sekä toimitusten määrän ja laadun oikeellisuuden varmistaminen. (Hokkanen & Virtanen 2012, 71 - 72.)

Varastonohjaus (KUVIO 4) on toimintaa, joka pyrkii tasapainottamaan kustannusten, varastotoiminnan ja laadun niin, että toiminta antaa parhaan mahdollisen hyödyn asiakkaille ja yritykselle. Varastonohjauksen tavoitteita ovat muun muassa kustannusten pienentäminen ja yleinen korkean palvelutason saavuttaminen. Varastonohjauksella pyritään yhdistelemään toimivaksi kokonaisuudeksi sellaisia asioita, kuten materiaalivirrat, tiedot, palvelutaso, ihmiset ja laitteet. (Hokkanen & Virtanen 2012, 72.)



KUVIO 4. Varastonohjaus ja sen yhteys tavaran käsittelyyn (Hokkanen & Virtanen 2012, 72)

Kuvio 4 osoittaa, että varastonohjaus koskee organisaation myyntiä ja ostoja. Varastossa oleva tietojärjestelmä on puolestaan osa varastonohjausta, sillä järjestelmän toiminta kohdistuu varaston operaatiohin. Monipuolisella tietojärjestelmän toiminnallisuudella voidaan varastonohjauksessa saavuttaa parempaa tehokkuutta, mikä vaikuttaa myyntiin ja ostoihin.

Koska nykypäivän varastoissa tuotteiden nimikkeiden määrä lisääntyy koko ajan, moni yritys on kohdistanut huomiota erilaisiin toiminnanohjausjärjestelmiin, joihin voi sisältyä myös varastonhallintamoduuli. Sen toiminta perustuu kokonaisuuksien hallintaan ja vaatii järjestelmän käyttäjältä tiettyä varastonhallinnan tietotaitoa, koska järjestelmä toimii juuri niin kuin käyttäjä saa sen toimimaan. (Hokkanen & Virtanen 2012, 72.)

3.4 Varastokirjanpito ja tietojärjestelmät

Varastokirjanpito kuuluu varaston ohjaustietoihin. Varastokirjanpidolla voidaan ylläpitää tuotteen perus- ja lisätietojen lisäksi tehokkaampaa varastonvalvontaa. Kirjanpidon avulla tuote voidaan hakea nopeasti millä tahansa tiedolla. Varastokirjanpitoon voidaan kirjata paljon asioita: ottoja, siirtoja, tavaran kokonaismääriä, hintoja, tilausten käsittelyjä. Kirjanpidon avulla voidaan saada useita erilaisia raportteja. (Hokkanen & Virtanen 2012, 73.)

Varastossa suoritetaan monipuolisia tehtäviä, ja tämä vaatii asiallisia apuvälineitä. Varaston toiminta perustuu käytettävissä olevan tiedon määrään. Tieto saapuvista

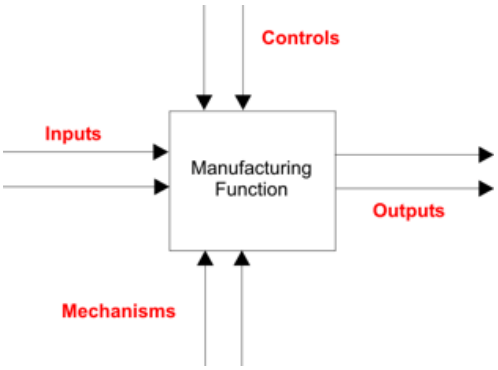
tuotteista tai tilauksista on tärkeä osa varaston jokapäiväistä toimintaa. Joskus hoidetaan kirjanpitoa tai saldojen hallintaa ja joskus puolestaan lasketaan tarvittavaa tilantarvetta. Varaston tietojärjestelmillä on niin laajat tekniset ja toiminnalliset ominaisuudet, että tarkka tietojärjestelmän kuvaus on työläs prosessi. (Hokkanen & Virtanen 2012, 122.)

4 WEB-TEKNIIKAT JA SUUNNITTELUVÄLINEET

Opinnäytetyön tietojärjestelmä toteutetaan web-sovelluksen muodossa. Toteutuksessa käytetään pääasiassa PHP-ohjelmointikieltä sekä MySQL-tietokantaa, jotka yhdessä vastaavat sovelluksen toiminnallisuuden toteutuksesta. Tässä luvussa kerrotaan, mitä etuja on olemassa, jos käyttää juuri näitä tekniikoita. Muilla kielillä, kuten javascript-skriptikielellä ja html-kuvauskielellä, on pieni rooli tämän työn web-sovelluksen toteutuksessa. Javascript-kieltä kuitenkin käytetään sivuston ja käyttäjän väliseen dynaamiseen vuorovaikutukseen.

4.1 IDEF0

IDEF0 on väline järjestelmän kokonaistoiminnan ja sen erillisten toimintojen kuvaamiseen. IDEF0 auttaa analysoimaan järjestelmää eri näkökulmistä ja erityisesti toiminnallisesta näkökulmasta. IDEF0 auttaa suunnittelijaa myös tunnistamaan, mitä toimintoja järjestelmässä tulee olemaan, mitä tarvitaan niiden tekemiseen ja toimiiko järjestelmä oikein vai väärin. Lisäksi IDEF0-kuvaustekniikan avulla järjestelmä voidaan jakaa alijärjestelmiin (osiin) ja kuvata jokaisen alijärjestelmän toiminta. IDEF0-mallin luominen on usein ensimmäinen askel pyrkimyksessä suunnitella jokin järjestelmä. IDEF0-konseptissa (KUVIO 5) laatikot ja nuolet ovat asioiden esitystapa. Laatikko kuvaa toimintoa tai tehtävää, laatikkoon tuleva tai siitä lähtevä nuoli kuvaa rajapintaa. Monen samanaikaisen toiminnon esittämiseen käytetään muita laatikkoja, joita yhdistetään nuolilla. Tällä toimenpiteellä osoitetaan, kuinka järjestelmän operaatiot (toiminnot) saadaan aikaan ja hallitaan. (IDEF 2012.)



KUVIO 5. IDEF0-menetelmän syntaksi (IDEF 2012)

IDEF0-konseptin periatteet ovat seuraavia: diagrammit perustuvat laatikoihin ja nuoliin, elementtien merkitys voidaan kuvata tekstillä, sallitaan enintään kuusi alitoimintoa yhtä laatikkoa kohti. IDEF0-menetelmä mahdollistaa järjestelmällisen ja tarkan esityksen toiminnoista sekä niiden suhteista. Menetelmällä voidaan tehostaa järjestelmän toimintojen mallinnusta. Toimintoja voidaan kuvailla syötteillä (inputs), tarkistuksilla (controls), ulostuloilla (outputs) ja mekanismilla (mechanism). (IDEF 2012.)

4.2 Projektin web-tekniikoiden edut

PHP-ohjelmointikielellä on nykyään merkittävä asema verkkopalvelujen toteutuksessa, erityisesti yhdessä MySQL-tietokannan kanssa käytettynä. PHP-kieli tarjoaa hyvän pohjan nykyaikaisten palvelujen toteuttamiselle. Silloin kun tietokantoja käytetään tietojen tallennukseen ja hakuun, PHP on se kieli, jonka avulla tietoja viedään tietokantaan, tallennetaan tauluihin ja haetaan sieltä erilaisissa tilanteissa. PHP-kieli soveltuu nykyään niin yksinkertaisten web-sovellusten kuin erilaisten monimutkaisten tietojärjestelmien toteutuskieleksi (Heinisuo & Rauta 2007, 9). Useiden projektien kehitysaika PHP-ohjelmointikielen käytön tapauksessa on selvästi lyhyempi kuin muita ohjelmointikieliä käytettäessä. PHP-kieleen olennaisesti liittyy tietokantojen käyttämisen helppous. Liittyen tietokannan ulkopuolisen toiminnallisuuden toteutukseen, PHP-ohjelmointikielessä on satoja sisäisiä funktioita, jotka voidaan kutsua antamalla funktion nimi ja tarvittaessa parametrina lisätiedot. (Zandstra 2001, 21, 86, 212.)

MySQL-tietokanta kehitettiin alun perin käytettäväksi verkkosovelluksissa, jotka asettavat tietokannoille seuraavia vaatimuksia: nopeus, laajennettavuus ja ylläpidon helppous. MySQL-tietokanta on relaatiotietokanta, mikä tarkoittaa, että kannassa on joukko yhteen liitettyjä tai suhteessa toisiinsa olevia tauluja, jotka koostuvat sarakkeista ja riveistä. MySQL-tietokanta varastoi kaikki tietokannat, taulut, sarakkeet ja rivit, niiden sisältämät tiedot ja käsittelee ne yhtenä kokonaisuutena. MySQL-tietokannassa tietokannat ja taulut luodaan, muokataan ja poistetaan erilaisten kyselyjen avulla. Niillä myös poimitaan tietoja tietokannasta. (Meloni 2003, 7 - 11.)

MySQL-tietokanta on nopea ja ei niin monimutkainen kuin muut tietokannat, esimerkiksi Oracle. MySQL-tietokanta on yhteensopiva lähes kaikkien ohjelmointikielten kanssa. Tietokannan etuihin kuuluvat myös hinta (yleensä tämä tietokanta on ilmainen) ja helppo siirrettävyys käyttöjärjestelmien välillä. MySQL-tietokantaa käytetään niin sähköisessä kaupassa kuin pelkän sisällön hallintaan. (Meloni 2003, 11 - 15.)

PHP-skriptikielen ja MySQL-tietokannan integraatiomahdollisuudet ovat tehokkaat. MySQL-tietokantaan liittyen, PHP-ohjelmointikieli tarjoaa kymmeniä funktioita, joilla voidaan käsitellä tietokannassa olevaa tietoa hakemalla dataa tietokannasta ja jatkokäyttämällä sitä omiin tarpeisiin (Gilmore 2005, 618). PHP-ohjelmasta MySQL-tietokantaa ei käsitellä suoraan, vaan ohjelma ottaa yhteyttä MySQL-tietokantapalvelimeen lähettämällä palvelimelle komentoja, jotka on kirjoitettu SQL-kielellä. Palvelin lähettää takaisin ohjelmalle vastaukseksi tietokannassa olevaa tietoa. (Heinisuo & Rauta 2007, 40.)

Tietojärjestelmän kannalta voidaan ottaa huomioon PHP-ohjelmointikielen tietojen välitysmenetelmät. Tämä kieli tarjoaa joukon ennaltamääritettyjä muuttujia, joihin päästään mistä tahansa lähdekoodin osasta. Näitä muuttujia PHP-kielessä sanotaan superglobaaleiksi muuttujiksi. Nämä muuttujat antavat kehittäjälle suuren määrän ympäristökohtaista tietoa. Esimerkiksi nämä muuttujat voivat sisältää ja palauttaa tietoa senhetkisestä käyttäjästä, hänen istunnostaan ja käyttöympäris-

töstään. Esimerkkejä näistä muuttujista ovat $_GET, $_POST, $_SESSION. Kaksi ensimmäistä sisältävät tietoja, jotka liittyvät vastaavasti kaikkiin GET- ja POST-metodeilla välitettyihin parametreihin. $_GET- ja $_POST-superglobaalit muuttujat ovat oletuksena ainoat tavat päästä muuttujiin, jotka on välitetty vastaavasti GET- ja POST-metodien avulla. $_SESSION-superglobaali muuttuja sisältää puolestaan kaikkiin istuntomuuttujiin liittyvää tietoa. Esimerkiksi, kun rekisteröidään istuntotiedot, niihin voidaan viitata koko Web-sovelluksesta, ilman että tietoa välitettäisiin GET- ja POST-metodien avulla. (Gilmore 2005, 63 - 68.)

5 TIETOJÄRJESTELMÄ WEB-SOVELLUKSENA

Tässä osiossa kuvataan opinnäytetyön tuloksena tehtyä web-sovellusta, jota voidaan sanoa tietojärjestelmäksi sen tarkoituksen perusteella. Työosuudessa selvitetään sovelluksen merkitys, toiminnallinen kuvaus sekä rakenteellinen kuvaus.

5.1 Yleistä

Tietojärjestelmä toteutetaan organisaation varastoa varten. Sovellus on alustavasti tarkoitettu keraamista laattaa koskevien tietojen ylläpitoon, mutta sovellusta voidaan tarvittaessa käyttää muiden materiaalien tilaustenkäsittelyyn. Tietojärjestelmän tehtävä on palvella varastoon saapuvien asiakkaiden tilausten kirjanpitoa. Itse tilaus tapahtuu yrityksen konttorissa, joka puolestaan ostaa tilatut tavarat hankkijalta tai tuottajalta (maahantuojalta). Hankkijalta tilatut tuotteet toimitetaan yrityksen varastoon, josta ne ovat asiakkaan noudettavissa.

Tietojärjestelmän asiakasvaatimuksiin kuuluvat laatan ostajan tilauksen tekohetkellä syntyvien tietojen ylläpitäminen varaston tietojärjestelmässä. Tiedot ovat ostajan tiedot sekä hänen tekemänsä tilauksen tiedot. Tietojen ylläpitäminen järjestelmässä helpottaa varaston toimintaa: tuotteiden luovuttamista asiakkaille, varastopaikkojen tehokasta käyttöä ja tuotteiden seuraamista yleisesti. Toinen toiminto, hylkytavaran esittäminen, ei ollut alustavasti asiakasvaatimuksena, mutta myöhemmin todettiin, että tietojärjestelmässä voitaisiin lisäksi merkitä saapuneen tuotteen laatu, sillä varastoon voi saapua hylkytavara. Tällä tavalla tuotteen kunnon merkitseminen nousi toiseksi asiakasvaatimukseksi. Tämän toiminnon toteuttamisen jälkeen henkilö, jonka rooli tietojärjestelmässä on johtaja, voi katsoa varastossa olevia hylkytuotteita myös kuvien avulla ja ottaa tarvittaessa yhteyttä tuotteen tuottajaan.

Tiedot ostajasta ja hänen tilauksesta voidaan syöttää järjestelmään, kun asiakkaan tilaama tuote saapuu varastoon. Tiedot saadaan konttorilta tai saapuvan tavaran mukana. Tiedot voidaan syöttää myös, kun tilaus ei vielä saapunut varastoon. Tällöin tuotteen varastopaikka järjestelmässä merkitään myöhemmin, kun tuote saa-

puu varastoon. Ensimmäinen vaihtoehto on parempi, koska tässä tapauksessa järjestelmään ei syötetä turhia tietoja tilauksista, joilla ei ole paikkaa varastossa.

5.2 Varasto tietojärjestelmän ympäristönä

Tietojärjestelmää on tarkoitus käyttää varastossa. Itse sovelluksen toimintaa ei kuitenkaan sidottu juuri varastoon, vaan sitä voidaan käyttää varastoa vastaavissa paikoissa ja siinä, missä on tarve tietojen ylläpitämiseen ja tilaustenkäsittelyyn. Liittyen varaston toimintaan, teoriaosuudessa tuotiin esille joukko asioita, jotka vaikuttavat tietojärjestelmän toiminnallisuuden toteutukseen.

Varastonhoitajaa koskevat huomautukset liittyvät tämän opinnäytetyön tietojärjestelmään, koska sen toteutuksessa pyritään alusta asti toteuttamaan toimintoja, joiden tulee pystyä kertomaan varastonhoitajalle tuotteen laadusta ja siitä, onko tuote varastossa vai ei. Näillä toiminnoilla varastotyöntekijä kykenee varmistamaan laadun ja säilytysajan, vaikka suora tuotteen tilan esittäminen ei ole toteutettu järjestelmässä. Kuitenkin pyrkimys toteuttaa tavaran paikan kirjaaminen tietojärjestelmään mahdollistaa näkemään, että varastosta löytyy tuote, jonka säilytysaika voitaisiin tarkistaa. Lisäksi järjestelmän toteutuksessa pyritään helppokäyttöiseen käyttöliittymään, joka mahdollistaa helpon tietojen syöttämisen ja ylläpitämisen, mikä johtaa varaston tehokkaaseen toimintaan.

Teoriaosuudessa mainittujen asioiden perusteella voidaan todeta, että hakutoiminnon toteutus tietojärjestelmässä on hyödyllinen. Monipuolisen hakutoiminnon avulla voidaan tunnistaa tietyllä paikalla oleva tuote, hoitaa hyllytystä ja tuotteiden hakua. Toiminnallisella tietojärjestelmällä voidaan tarkistaa, kuinka varastoon saapunut tuote vastaa lähetyslistassa ilmoitettua. Tämä tapahtuu siinä tapauksessa, jos tiedot tuotteesta on jo syötetty tietojärjestelmään ja tuote saapuu myöhemmin. Puolivalmiiden tuotteiden takia olisi hyvä toteuttaa mahdollisuus syöttää tuotteen paikka eri tavoilla.

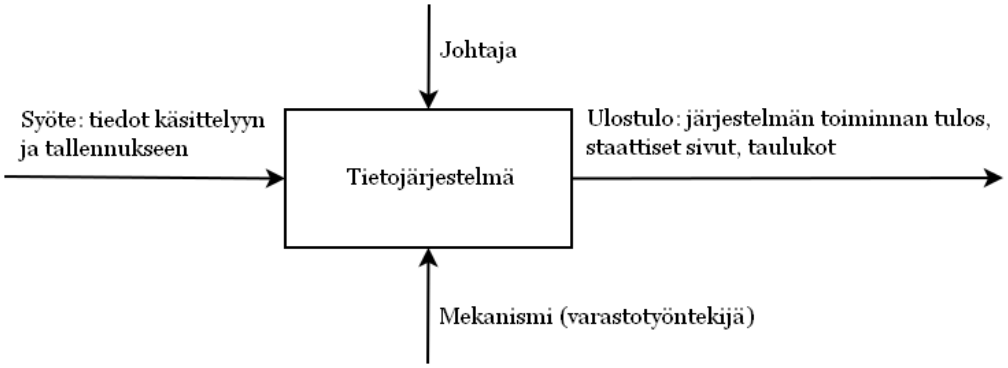
Kuten todettiin teoriaosuudessa, varaston tyyppi voi vaikuttaa tietojärjestelmän toiminnallisuuteen. Aikaisemman varaston kuvauksen perusteella voidaan päättää,

että tietojärjestelmän tilaajalla on käytössä jakeluvarasto, josta asiakkaat noutavat tilauksensa. Tämän takia toteutetussa tietojärjestelmässä varastossa oleva tilaus voidaan merkitä vastaanotetuksi yhdellä napin painamisella, jolloin järjestelmässä vapautuu myös tilauksen varastopaikka.

Vaikka tietojärjestelmä pyritään toteuttamaan helppokäyttöisenä, järjestelmä toimii juuri niin kuin käyttäjä saa sen toimimaan. Esimerkiksi järjestelmän käyttäjän antamien tietojen oikeellisuus on kokonaan käyttäjän vastuulla. Järjestelmään syötetyt väärät tiedot johtavat virheelliseen tietojenkäsittelyyn. Lisäksi järjestelmän käyttö väärällä tavalla voi aiheuttaa virheitä järjestelmän toiminnoissa.

5.3 Tietojärjestelmän arkkitehtuuri

CASE-välineisiin kuuluvan Dia-diagrammieditorin avulla tehtiin seuraava (KUVIO 6) järjestelmän kokonaistoiminnan yleiskuva, joka perustuu IDEF0-menetelmään sekä tietovuokaavioon.



KUVIO 6. Tietojärjestelmän toiminta

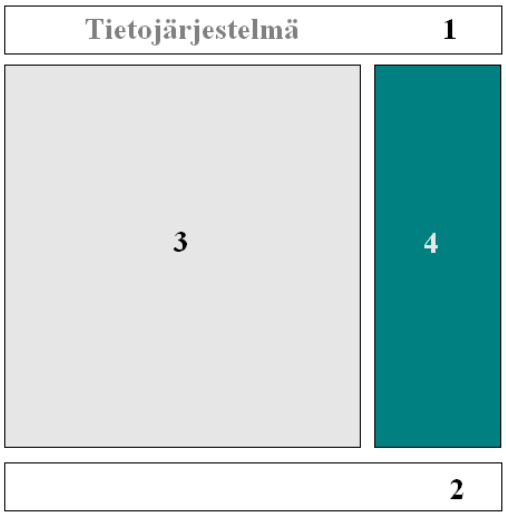
Tietojärjestelmän syötteenä ovat tiedot, jotka syötetään tietokantaan: asiakas- ja yritystiedot, tilauksen tiedot ja paikka varastossa. Tietojärjestelmän toiminnan tulos on tiedot asiakkaista ja tilauksista taulukkomuodossa.

Tietojärjestelmän toteutuksessa käytettiin vaiheittaista lähestymistapaa, joka on kuitenkin suppeampi kuin teoriaosuudessa mainittu, mutta muistuttaa sitä. En-

simmäiseksi jouduttiin selvittämään, miksi järjestelmä toteutetaan ja mitä tehtäviä se tulee suorittamaan. Tämän jälkeen asetettiin asiakasvaatimukset, joiden perusteella aloitettiin toiminnallisuuden toteutus. Tietojärjestelmän suunnitteluvaiheessa pyrittiin selvittämään ohjelmamoduulit ja niiden yhteistyö (rajapinnat), sillä tietojärjestelmä koostuu monesta ohjelmamoduulista. Toteutusvaiheessa käytettiin teoriaosuudessa mainittuja tekniikoita tietojärjestelmän toteutukseen.

Tietojärjestelmä on toteutettu HTML-kuvauskielen, PHP-palvelimen ohjelmointikielen ja Javascript-käyttäjän skriptikielen avulla. Tietokantana toimii MySQL-tietokanta. Kehitysympäristönä käytettiin pääasiassa PHP Expert editor -ohjelmaa.

Järjestelmän jokainen ladattava sivu (KUVIO 7) on jaettu kahteen osaan, jotka ladataan ja tulkitaan selaimessa staattiseksi HMTL-sivuksi. Ensimmäinen osa on sisältö, joka luodaan automaattisesti (käyttäjän kyselyllä) tai palvelimen vastauksen tuloksena. Toinen osa on sivupohjat, jotka määrittelevät järjestelmän yleistyylin. Päävalikko liitetään sivulle automaattisesti, riipuen siitä, mikä rooli on kyseessä.



KUVIO 7. Tietojärjestelmän käyttöliittymän rakenne

Kuviossa 7 on esitetty käyttöliittymän rakenne sekä yleinen väritys. Numeroilla 1 ja 2 on merkitty sivupohjat, jotka ovat aina samannäköiset riippumatta sisällöstä ja

muodostavat kokonaisuuden ohjelmamoduulien kanssa. Numerolla 4 on merkitty paikka päävalikolle, joka on erilainen jokaisella järjestelmän roolilla. Järjestelmän eri rooleilla on omat toiminnot. Päävalikko muodostetaan roolien päämoduuleissa. Kentässä numero 3 ladataan se toiminto, joka valittiin päävalikosta. Lisäksi kentässä 3 järjestelmän käyttäjät suorittavat toimintansa.

Järjestelmän jokaisella roolilla on oma päämoduuli, jossa on toteutettu joukko pakollisia perustoimintoja. Kirjautumattomalle käyttäjälle tämä päämoduuli on index.php-tiedosto, varastotyöntekijälle klad.php-tiedosto, johtajalle dir.php-tiedosto, ylläpitäjälle admin.php-tiedosto. Kaikki lisätoiminnot on toteutettu erikseen ja ne liitetään päämoduuliin tarvittaessa (käyttäjän kutsulla).

Käyttötapaukset ja niiden oikeudet on toteutettu järjestelmässä sisäänkirjautumisen avulla. Onnistuneella sisäänkirjautumisella istunto pysyy voimassa koko työn ajan. Kaikki skriptien perusmuuttujat alustetaan vastaavien moduulien sisällä. Pääsy tietokantaan ja tauluihin on rajoitettu salasanalla. Tiedot tietokannasta pysyvät erillisessä config.php-tiedostossa, joka sijaitsee palvelimella. Näihin tietoihin ole pääsyä selaimeista. Uloskirjautuminen järjestelmästä lopettaa istunnon, minkä takia ei ole mahdollista kirjautua sisään ilman tunnuksen ja salasanan uutta syöttämistä.

5.4 Tietokanta

Tietokantaan pääsy ja sen kehitys tapahtui phpMyAdmin-ohjelmiston avulla. Tietojärjestelmässä käytetään MySQL-tietokantaa, jonka versio on 5.1 ja joka on asennettu Unix-palvelimelle. Tietojärjestelmässä käytetään yhtä tietokantaa, jossa on neljä toisistaan riippumatonta taulua.

Ensimmäinen taulu, jonka nimi tietokannassa on plitka_users, on tarkoitettu tietojärjestelmän käyttäjien tietojen tallentamiseen. Tässä taulussa on seuraavat kentät: käyttäjän nimi, tunnus, salasana järjestelmään pääsyä varten, käyttäjän työtietokanta (vastuualue) sekä käyttäjän asema tietojärjestelmässä. Lisäksi taulussa on id-

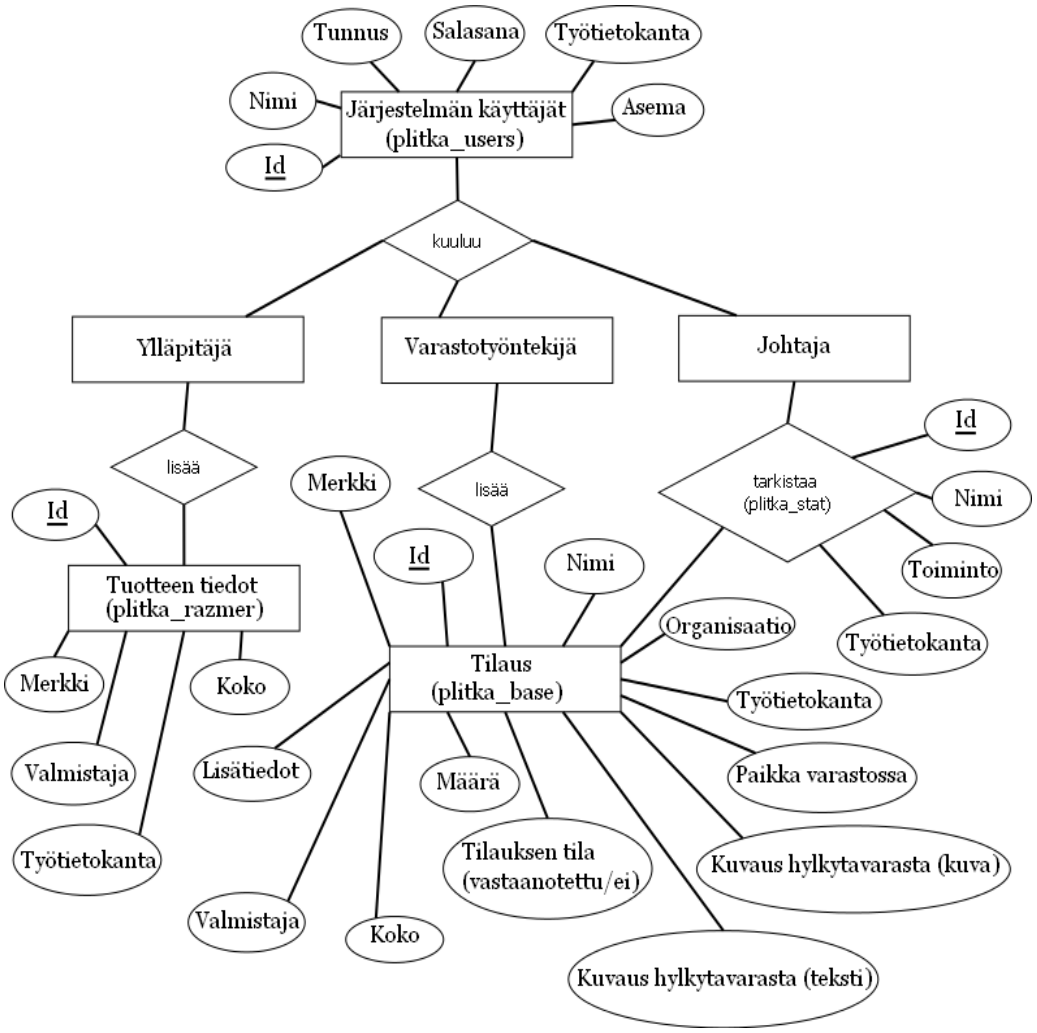
kenttä, jolla on auto-increment-ominaisuus. Tämän kentän avulla jokaiselle tietueelle saadaan oma tunnusnumero.

Toinen tietokannan taulu tietojärjestelmässä on nimeltään plitka_razmer. Tähän tauluun tallennetaan tietoja uuden tuotteen (keraamisen laatan) ominaisuuksista. Käyttäen näitä tietoja varastotyöntekijä voi lisätä uudet tilaukset tietojärjestelmään. Tuotteen ominaisuudet ja niitä vastaavat kentät ovat tuotteen valmistaja, merkki, koko ja työntekijän työtietokanta, jonka avulla tuotteet jaetaan tuotteen tyypin (tai muun ominaisuuden) mukaan omiin ryhmiinsä. Lisäksi tässä taulussa on id-kenttä, jolla on auto-increment-ominaisuus.

Kolmas tietokannan taulu, jonka nimi tietokannassa on plitka_base, sisältää eniten kenttiä. Taulu sisältää seuraavat kentät: asiakkaan nimi, asiakkaan lisätiedot, asiakkaan tilaaman tuotteen valmistaja, merkki, tuotteen koko, lukumäärä, varastoon saapuneen tuotteen paikka, tilauksen lisänneen työntekijän työtietokanta, tilauksen tila, sanallinen kuvaus hylkytavarasta, kuvallinen kuvaus hylkytavarasta ja asiakkaan yritys. Lisäksi taulussa on id-kenttä, jolla on auto-increment-ominaisuus.

Tietokannan neljäs taulu, jonka nimi järjestelmän tietokannassa on plitka_stat, kuvaa varastotyöntekijöiden aktiivisuutta ja sisältää tietoja varastotyöntekijöiden toiminnoista. Taulussa on yhteensä 4 kenttää: toimenpiteen suorittaneen varastotyöntekijän nimi, varastotyöntekijän toiminto ja hänen työtietokanta. Työntekijän toimintona voi olla varastossa olevan tuotteen merkitseminen asiakkaan vastaanottamaksi tai uuden tilauksen lisääminen tietojärjestelmään. Taulun neljäntenä kentänä on id-kenttä, jolla on auto-increment-ominaisuus.

Seuraava kuvio (KUVIO 8) on tehty ER-kaavion pohjalta ja havainnollistaa tietokannan käyttöä tietojärjestelmässä.



KUVIO 8. Tietokannan rooli tietojärjestelmässä

Kuviossa 8 on esitetty tietojärjestelmän tietokannan taulut ja niiden rooli järjelmän toiminnassa. Kuviosta nähdään taulujen nimet sekä taulujen kentät, jotka kuvaavat, mitä tietoa tallennetaan tietokantaan.

5.5 Järjestelmän toiminnallinen kuvaus

Web-sovelluksen käyttöön tarvitaan käyttäjätunnus ja salasana. Niiden syöttämisen jälkeen tapahtuu sisäänkirjautuminen: jos käyttäjän antamat tunnus ja salasana ovat olemassa tietokannassa, käyttäjän sisäänkirjautuminen onnistuu ja hän saa pääsyn niihin toimintoihin, joihin käyttäjä on oikeutettu roolinsa mukaan. Kaikkiaan tietojärjestelmässä on kolme roolia: ylläpitäjä, varastotyöntekijä ja johtaja. Sisäänkirjautumisen moduuli on kokonaan toteutettu etusivun ohjelmamoduulissa.

Tietoturvasyistä sisäänkirjautumisen yhteydessä käytetään istuntoja: superglobaali-tyyppisessä taulukossa pysyvät käyttäjäkohtaiset parametrit, jotka käyttäjä saa sisäänkirjautuessaan. Tällä tavalla tietokoneeseen ei synny evästeitä ja ikkunan sulkemisen jälkeen tarvitaan uusi sisäänkirjautuminen. Istunnot käynnistetään komennolla session_start (KUVIO 9), joka sijaitsee aina ohjelmamoduulin ensimäisellä rivillä.

```
<?
session_start();
include("config.php");
$_session['logON']='no';
```

KUVIO 9. Istunnon aloitus

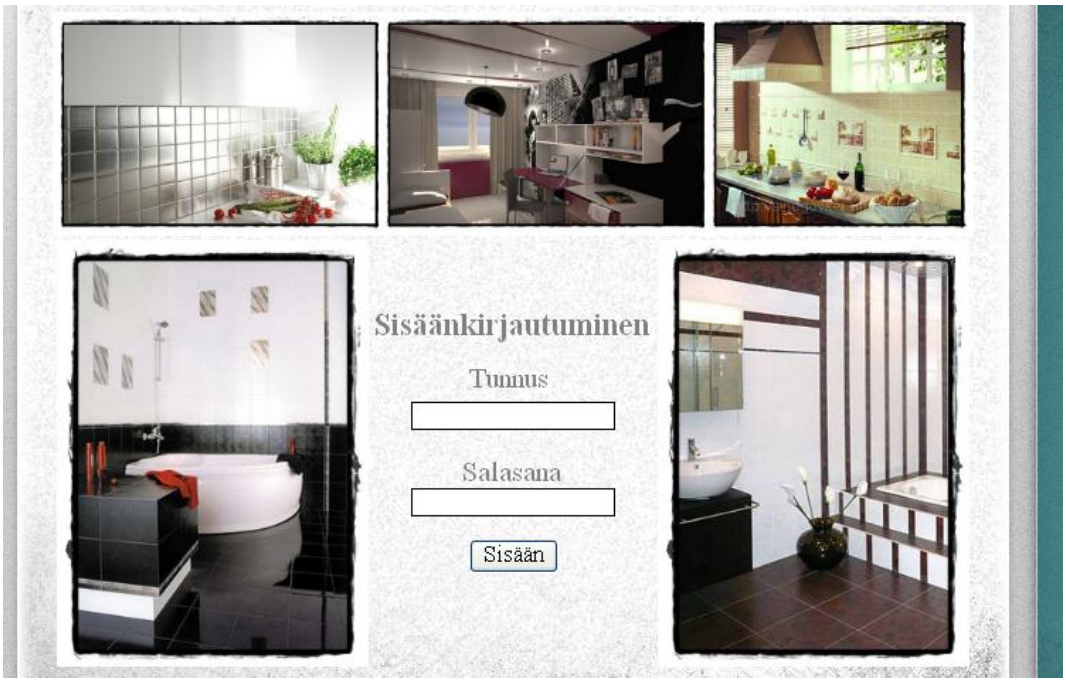
Superglobaalin taulukon $_SESSION['logON']-muuttujaan tallennetaan tiedot käyttäjän sisäänkirjautumisen tilasta: parametri ”no” on oletusarvo ja tarkoittaa sitä, että sisäänkirjautumistietoja ei vielä ole annettu tai ne annettiin väärin. Parametrin arvo ”yes” puolestaan kertoo, että sisäänkirjautuminen onnistui. Komennot, jotka tarvitaan yhteyden muodostamiseen tietokantaan, on sijoitettu erilliseen config.php-tiedostoon, joka liitetään ohjelmamoduuliin include-komennon avulla.

Etusivun lataamisessa sisäänkirjautumisen moduuli muodostaa ja tulostaa lomakkeen (KUVIO 10), joka on tarkoitettu tunnuksen ja salasanan syöttämiseen.

```
<form action='' name=forma method=post>
<p align=center>Tunnus<br><input type=text name=login style=\"border: solid 1px;\"></p>
<p align=center>Salasana<br><input type=password name=pass style=\"border: solid 1px;\"></p>
<center><input type=submit value='Sisään' size=10></center>
</form>
```

KUVIO 10. Sisäänkirjautumisen lomake

Käyttöliittymässä kirjautumisen kenttä, joka sijaitsee etusivulla, on seuraavan näköinen (KUVIO 11):



KUVIO 11. Sisäänkirjautumisen kenttä etusivulla

Kuviosta 11 näkyy, että etusivun koristamiseen käytetään tyylimäärityksiä sekä aiheeseen sopivia kuvia. Kuvat ladataan etusivulla ja tyylimääritykset toteutetaan yleensä erillisissä tiedostoissa css-tekniikan ja html-kuvauskielen div-elementtien avulla.

Lomakkeen tiedot välitetään palvelimelle POST-metodilla etusivun ohjelmamoduulille. Tunnuksen kenttä on HTML-tagin text-tyyppiä, salasanan kenttä on puolestaan password-tyyppiä, mikä mahdollistaa salasanan merkkien salassapidon salasanan kirjoitushetkellä. Käyttäjän antamien tietojen lähetys käsittelijälle (KUVIO 12) tapahtuu ”Sisään”-napin painamisen jälkeen.

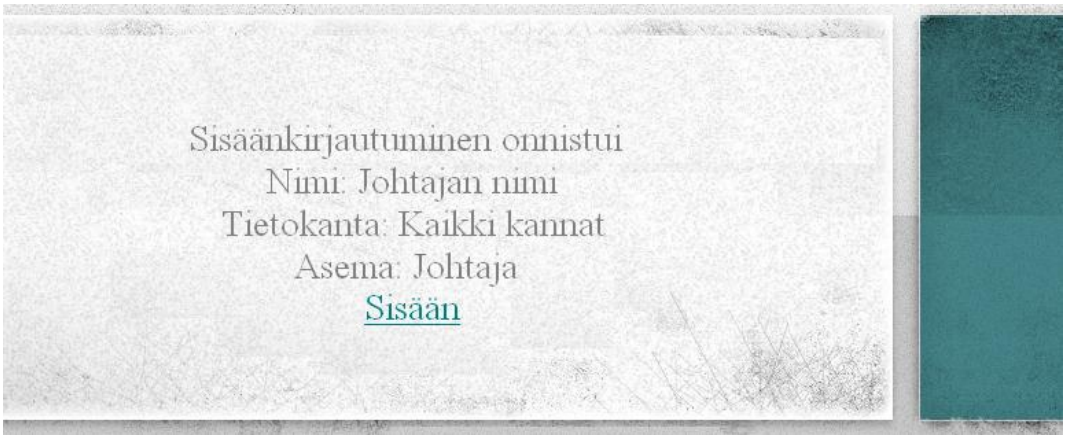
```
if (isset($_POST['login']))
{
$log=$_POST['login'];
$pass=$_POST['pass'];
echo '<center><h2>';
$result = mysql_query("SELECT * FROM `plitka_users` where
(`login`='$log')AND(`pass`='$pass')");
while ( $postrow[] = mysql_fetch_array($result));
if (count($postrow)>1)
    {echo 'Kirjautuminen onnistui <br>';
     $_SESSION['logON']='yes';
     $_SESSION['fio']=$postrow[0][fio];
     $_SESSION['base']=$postrow[0][base];
     $_SESSION['state']=$postrow[0][state];
     echo 'Nimi: '.$_SESSION['fio'].'<br> Tietokanta: '.$_SESSION['base'].'<br> Asema:
     '.$_SESSION['state'];
     if (($postrow[0][state]=='admin')or($postrow[0][state]=='Yllapitaja'))
     {echo '<br><a href=admin.php>Sisään</a>';}
     else if ($postrow[0][state]=='Varastotyontekija')
     {echo '<br><a href=klad.php>Sisään</a>';}
     else if ($postrow[0][state]=='Johtaja'){echo '<br><a href=dir.php>Sisään</a>';}
    }
    else {echo '<h1 align=center>Virhe sisäänkirjautumisessa.<br>
    <a href=index.php>Siirry</a> pois</h1>';}
echo '</h2></center>';
}
```

KUVIO 12. Lomakkeen käsittelijä

Muuttujat $log ja $pass sisältävät käyttäjän antamia tietoja, joiden perusteella muodostetaan SQL-kysely tietokantaan. Jos kysytty ”tunnus-salasana”-pari on olemassa tietokannassa (komento count($postrow) palauttaa löydettyjen tietueiden lukumäärän), käyttäjälle ilmoitetaan onnistuneesta sisäänkirjautumisesta, muuten tulee ehdotus yrittää uudelleen.

Seuraavat muuttujat saavat arvon automaattisesti: $_ SESSION['logON'], $_SESSION['fio'] – työntekijän nimi, $_SESSION['base'] – käytössä oleva työtietokanta, $_SESSION['state'] – työntekijän asema (virka). Riippuen asemasta, käyttäjä siirtyy seuraaviin ohjelmamoduuleihin: admin.php (ylläpitäjä), klad.php (varastotyöntekijä) ja dir.php (johtaja).

Onnistuneen kirjautumisen jälkeen järjestelmän käyttäjä näkee ikkunan (KUVIO 13), jossa tulostetaan hänen tiedot sekä linkki, jolla siirrytään eteenpäin.



KUVIO 13. Aloitusnäkymä johtajan roolilla kirjautumisen jälkeen

Kuviossa 13 on esitetty käyttäjän, jonka rooli järjestelmässä on johtaja, aloitusnäkymä. Ikkunassa mainitaan käyttäjän nimi, pääsy tietokantaan sekä asema (rooli). Järjestelmän johtajalla on pääsy kaikkiin tietokantoihin. Tietokannalla tässä tapauksessa tarkoitetaan varastotyöntekijän työtietokantaa, eli varastotyöntekijän vastuualuetta, joka määräytyy keraamisen laatan eri tyyppien mukaan. Esimerkkejä työtietokannoista ovat seinälaatat, lattialaatat tai kaikki työtietokannat. Ylläpitäjä voi muokata työtietokantojen listaa tekstitiedoston avulla, joka sisältää pelkästään työtietokantojen nimet. Ylläpitäjä määrittää varastotyöntekijälle työtietokannan. Johtajalla ja ylläpitäjällä on aina oikeus kaikkiin tietokantoihin, koska nämä roolit ovat johtoasemassa eikä niitä ole sidottu yhteen työtietokantaan.

5.5.1 Ylläpitäjä

Tietojärjestelmässä on olemassa ylläpitäjän rooli. Ylläpitäjän tehtävät järjestelmän ensimmäisessä versiossa ovat seuraavat: varastotyöntekijän vastuualueiden (työtietokantojen) luominen ja poistaminen, järjestelmän käyttäjän luominen ja hänen roolinsa määrittäminen, käyttäjien poistaminen, keraamisen laatan valmistajan, merkin ja koon lisääminen järjestelmään. Kuten aikaisemmin on mainittu, ylläpitäjä tarvitsee tunnuksen ja salasanan, jotta voi kirjautua järjestelmään. Ylläpitäjien lukumäärä tietojärjestelmässä ei ole rajoitettu, mikä mahdollistaa ylläpitäjien vuorotyötä.

Ylläpitäjän perusmoduuli sijaitsee admin.php-tiedostossa, muut isot toiminnot (käyttäjälistan muokkaus, uusien merkkien lisääminen jne) on sijoitettu erillisiin tiedostoihih, jotka liitetään (KUVIO 14) perusmoduuliin riippuen kyselyistä.

```
$st='0';
$st= $_GET["st"];
$st= preg_replace("|[^0-9]|i", "", $st);
if ($st=='') {$st='4';}
$s= 'a_st_'.$st.'.php';
if(file_exists($s)) {include($s);}
```

KUVIO 14. Lisämoduulien liittäminen

Muuttujassa $st pysyy liitettävän toiminnon numero, oletusarvona numero on nolla. Tiedot liittämiseen välitetään palvelimelle GET-metodilla, mikä voi vaatia lisää turvallisuustoimenpiteitä. Funktio preg_replace, jonka parametrit ovat ”|[^0-9]|i”, poistaa mahdolliset tarpeettomat symbolit kyselystä. Liitettävän tiedoston nimi koostuu ”a_st_”-alkuosasta ja toiminnon numerosta, joka välittyy muuttujassa $st. Tarvittava moduuli ladataan komennolla include. Kuten aikaisemmin mainittiin, moduulit ladataan kuvion 7 kenttään numero 3. Tällöin kentän sisältö vaihtuu riippuen käyttäjän valitsemasta toiminnosta.

Ylläpitäjällä on oikeus muokata käyttäjälistaa: lisätä käyttäjät ja määrittää heille roolit ja oikeudet omiin vastuualueisiin (työtietokantoihin) tai poistaa käyttäjät järjestelmästä. Käyttäjälista, käyttäjien nimet (fio-kenttä), tiedot sisäänkirjautumista varten (login- ja pass-kentät), oikeus pääsyyn tietokantaan (base-kenttä) ja asema (state-kenttä) pysyvät MySQL-tietokannan plitka_users-taulussa. Tietojen lisääminen tauluun tapahtuu SQL-kyselyllä (KUVIO 15).

```
$c1=$_POST['kyr'];
$c2=$_POST['st'];
$a1=$_POST['fio'];
$a2=$_POST['login'];
$a3=$_POST['pass'];
$sql = "INSERT INTO `plitka_users` (`id`,`fio`,`login`,`pass`,`base`,`state`)
VALUES (NULL, '$a1', '$a2','$a3','$c1','$c2')";
```

KUVIO 15. SQL-kysely tiedon lisäämiseen

Muuttujat $c1 ja $c2 sisältävät tietoja käyttäjän työtietokannasta (esimerkiksi seinälaatat tai lattialaatat) ja roolista (ylläpitäjä, varastotyöntekijä ja johtaja). Työtietokantojen lista on dynaaminen, ylläpitäjä voi muokata sitä käyttöliittymässä. Roolien lista on staattinen eli järjestelmässä on pysyvästi kolme roolia. Johtajaa ja ylläpitäjää varten on tehty automaattinen oikeuksien asettaminen (KUVIO 16) siinä tapauksessa, jos käyttäjää lisättäessä ylläpitäjä valitsee hänelle väärän asema-vastuualue-parin. Esimerkiksi ainoastaan varastotyöntekijälle voidaan määrittää työtietokanta, johtajalle ja ylläpitäjälle sitä ei voi tehdä, sillä heillä on aina pääsy kaikkiin työtietokantoihin. Automaattisessa asettamisessa johtaja ja ylläpitäjä siis oikeutetetaan kaikkien työtietokantojen käyttöön, jos heille yritetään valita varastotyöntekijän työtietokanta.

```
if (($c2=="Johtaja")OR($c2=="Yllapitaja"))
{if (($c1<>"Kaikki kannat")AND($c1!='')) {$c1="Kaikki kannat";echo "<font color=blue>
<h3 align=center>Käyttäjä sai käyttöoikeuden kantaan \"Kaikki kannat\"</h3></font>";}
}
```

KUVIO 16. Käyttäjän oikeuksien tarkistus uuden käyttäjän lisäämisessä

Uuden käyttäjän lisääminen sekä hänen oikeuksien tarkistus järjestelmän käyttöliittymässä on tehty seuraavan näköiseksi (KUVIO 17):

KUVIO 17. Uuden käyttäjän lisääminen järjestelmään

Kuviossa 17 näkyy lomake, jolla lisätään uudet käyttäjät järjestelmään. Asemaltaan käyttäjä voi olla ylläpitäjä, varastotyöntekijä tai johtaja. Työtietokanta (vastuualue) voi olla esimerkiksi lattialaatat, seinälaatat tai kaikki työtietokannat. Tämä tarkoittaa, että varastotyöntekijä voi lisätä tilaukset järjestelmään vain omassa työtietokannassa. Yllä oleva ilmoitus, että käyttäjä sai oikeudet kaikkiin tietokantoihin, tarkoittaa, että johtaja tai ylläpitäjä ei voi olla sidottu yhteen työtietokantaan, vaan hän saa oikeuden kaikkiin tietokantoihin, jos hänelle valitaan väärä asema.

Ylläpitäjällä on näkyvissä lista järjestelmän käyttäjistä. Käyttäjälistan avulla voidaan tarvittaessa poistaa käyttäjät. Käyttäjälistan tulostamisessa jokaisella rivillä on valintalaatikko, jolla on vastaavan käyttäjätunnuksen numeroarvo (KUVIO 18).

```
echo "<td><input name='item[]' type='checkbox' value=".$postrow[$i][id]."></td></tr>";
```

KUVIO 18. Käyttäjälistan rivi

Jokaisen valintalaatikon nimi muodostuu item[]-taulukkona. Kun ylläpitäjä valitsee käyttäjät, jotka halutaan poistaa, tiedot lähetetään POST-metodilla käsittelijälle (KUVIO 19) toiseen ohjelmamoduuliin.

```
$type = $_POST['item'];
if(!empty($type))
{
   $query = "(" ;
   foreach($type as $val){ $query.= $val.",";}
   $query = substr($query, 0, -1). ")" ;
   $query = "DELETE FROM plitka_users WHERE id IN ".$query;
}
```

KUVIO 19. Käyttäjätunnuksen poistaminen järjestelmästä.

Jos poistamiseen tarkoitetut tunnukset on valittu valintalaatikon avulla, $type-muuttujalla ei tule olemaan tyhjää arvoa ja skripti muodostaa $query-rivin, joka sisältää listan poistettavista muodossa ”(3,5,6,7...)”. Komento subst($query, 0, -1). ”)” poistaa viimeisen pilkun ja sen paikalle tulee sulkeva sulku. Tällä tavalla muuttujassa $query muodostetaan kysely tietojen poistamiseen plitka_users-taulusta.

Käyttöliittymässä käyttäjien lista on tehty seuraavan näköiseksi (KUVIO 20):

KUVIO 20. Käyttäjien lista käyttöliittymässä

Käyttäjien listassa tulostetaan tietojärjestelmän kaikki käyttäjät, heidän nimet, vastuualueet ja asemat. Käyttäjä poistetaan järjestelmästä valitsemalla hänet valintalaatikon avulla ja painamalla poista-nappia, joka sijaitsee käyttäjälistan alla.

Toinen ylläpitäjän toiminto lisätä tietojärjestelmään uusi keraamisen laatan valmistaja, merkki ja koko, joita myöhemmin varastotyöntekijä käyttää hyväkseen, on sijoitettu a_st_3.php-ohjelmamoduuliin. Kaikkien kolmen ominaisuuden lisääminen on tehty yhdellä sivulla. Kun halutaan lisätä esimerkiksi uusi tuotteen koko, sitä varten pudotusvalikoissa muodostetaan listat valmistajista ja merkkeistä (KUVIO 21):

```
$result = mysql_query("SELECT `marka` FROM `plitka_razmer` group by `marka`");
while ( $postrow[] = mysql_fetch_array($result)) ;
$posts = count($postrow)-1; $st='';
for($i = 0; $i < $posts; $i++) {$st.='<option>'.$postrow[$i][marka].'</option>';}

unset($postrow);
unset($result);
$result = mysql_query("SELECT `proizv` FROM `plitka_razmer` group by `proizv`");
while ( $postrow[] = mysql_fetch_array($result)) ;
$posts = count($postrow)-1; $sm='';
for($i = 0; $i < $posts; $i++) {$sm.='<option>'.$postrow[$i][proizv].'</option>';}
```

KUVIO 21. Valmistajien ja merkkien listojen muodostus

Muuttuja $result sisältää tietokannan palvelimen vastauksen kyselyyn ja taulukossa $postrow muodostetaan järjestetty ja riveille jaettu vastauksen sisältö. Komento unset() poistaa näiden muuttujien arvot, minkä takia niitä voidaan käyttää toistuvasti. FOR-silmukan avulla muodostetaan laatan valmistajien ja merkkien listat.

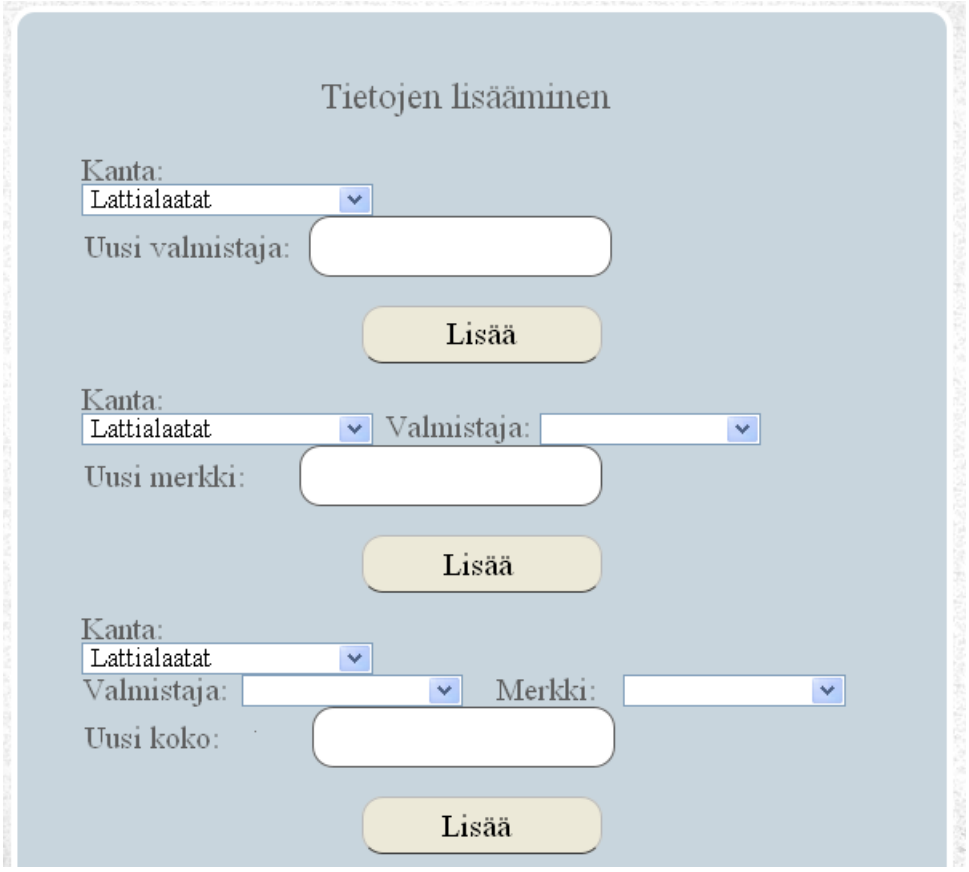
Kaikki tiedot laatasta, sen valmistajista ja koosta pysyvät tietokannan plitka_razmer-taulussa. Uusien tietojen lisäämisessä muodostetaan SQL-kysely tietojen lisäämiseen (KUVIO 22):

```
$c1=$_POST['proizv'];
$c2=$_POST['marka'];
$c3=$_POST['razmer'];
$a1=$_POST['base_r'];

$sql = "INSERT INTO `plitka_razmer` (`id`,`marka`,`proizv`,`razmer`,`base`)
VALUES (NULL, '$c1', '$c2','$c3','$a1')";
```

KUVIO 22. SQL-kysely laatan tietojen lisäämiseen tauluun

Käyttöliittymässä uuden valmistajan, merkin ja koon lisääminen on seuraavan näköinen (KUVIO 23):



KUVIO 23. Valmistajan ja merkin lisääminen

Kuviossa 23 näkyy tuotteen valmistajan, merkin ja koon lisäämisen lomake. Nämä tiedot järjestelmään syöttää ylläpitäjä. Tietojen lisäämisen jälkeen ne ovat käytössä varastotyöntekijällä, joka puolestaan lisää järjestelmään uudet tilaukset. Kuviossa 23 esitetty lomake on helppokäyttöinen. Kun halutaan lisätä uusi valmistaja, ensin valitaan sille kanta (tyyppi), joka voi olla esimerkiksi lattialaatat.

Valmistaja saadaan lisättyä tietojärjestelmään, kun kirjoitetaan valmistajan nimi vastaavaan kenttään ja painetaan Lisää-nappia. Uusi valmistaja on heti näkyvissä, kun halutaan valita sille uusi merkki vastaavassa kentässä.

5.5.2 Varastotyöntekijä

Sisäänkirjautumisen jälkeen tietojärjestelmän varastotyöntekijällä on seuraava toiminnallisuus: tilausten listan tarkistus, uuden tilauksen lisääminen (yritys- tai yksityisasiakkaat), hakutoiminto. Lisäksi varastotyöntekijä voi päivittää jo olemassa olevaa tilausta. Päivitys tässä tarkoittaa, että voidaan vahnojen tietojen perusteella lisätä järjestelmään uusi tilaus. Jokainen varastotyöntekijä toimii vain omassa yhdessä työtietokannassaan (nämä tietokannat ovat esimerkiksi lattia-, seinä- tai muut laatat). Tiedot muista työtietokannoista eivät ole saatavilla. Ylläpitäjä määrittää työtietokannan varastotyöntekijälle. Yhtä ja samaa työtietokantaa varten voidaan luoda monta varastotyöntekijää.

Kaikki varastotyöntekijän perustoiminnot on toteutettu klad.php-ohjelmamoduulissa (KUVIO 24):

```
session_start();
include ('config.php'); error_reporting(0);
include("top.html");

echo'<html><head><link type="text/css" rel="stylesheet" href="style2.css" />
</head><body style="padding-left:5px;">';

echo 'Varastotyontekijä: <font color=blue><b>'.$_SESSION[fio].'</b>
</font>, tyotietokanta: <font color=blue>'.$_SESSION[base].'</font><br>';
```

KUVIO 24. Varastotyöntekijän parametrien määrittely

Kuviossa 24 varastotyöntekijän päämoduuliin liitetään istuntojen superglobaalityyppinen taulukko, ladataan tietokannan asetukset ja staattinen sivupohja top.html, joka on näkyvän sivun ylemmän osan pohja. Käyttäjätietojen ja työtietokannan nimen tulostamiseen varastotyöntekijän sivulla käytetään arvoja istuntojen taulukosta: $_SESSION[fio]-nimi ja $_SESSION[base]-työtietokanta. Funktio

error_reporting() vastaa virheiden tietojen ja PHP-tulkin varoitusten tulostamisesta. Tässä tapauksessa funkion parametri on 0, eli funktio ei tulosta mitään.

Varastotyöntekijän ja muiden roolien toiminnot ovat saatavilla päävalikosta, jossa toimintojen lisäksi on tehty uloskirjautumisen nappi. Varastotyöntekijän toimintoihin kuuluu uuden tilauksen lisääminen järjestelmään. Asiakkaana voi olla yksityis- tai yritysasiakas. Tilauksen lisäämisen toiminto löytyy päävalikosta. Yksityisasiakkaan tilauksen lisääminen tapahtuu add.php-ohjelmamoduulissa ja yrityksen tilauksen lisääminen add_org.php-ohjelmamoduulissa. Yleinen moduulin rakenne voidaan esittää seuraavan kaavion (KUVIO 25) mukaan:

KUVIO 25. Tilauksen lisäämisen järjestys

Kun valitaan päävalikosta toiminto lisätä uusi tilaus, saadaan näkyville vastaava sivu, jossa on HTML-lomake (KUVIO 26) . Tässä lomakkeessa on kentät, jotka on pakko täyttää (asiakkaan nimi, lisätiedot, laatan määrä ja sen paikka varastossa).

```
<form action='klad.php' method=post>
<h3 align=center>Tilauksen lisääminen</h3>
<table width=300>
<tr><td>Nimi: </td><td><input type=text name=fio></td></tr>
<tr><td>Lisätiedot: </td><td><input type=text name=pasp></td></tr>
<tr><td>Määrä: </td><td><input type=text name=kol></td></tr>
<tr><td>Paikka varastossa: </td><td><input type=text name=razmeshenie></td></tr>
<tr><td><input type=submit name=add_kol value=Tallenna></td></tr>
```

KUVIO 26. Tilauksen lisäämisen lomake

Lomakkeen käsittelijänä toimii varastotyöntekijän klad.php-perusmoduuli. Lomakkeen tiedot välitetään palvelimelle POST-metodilla. Kaikki kentät ovat text-tyyppisiä, eikä kentän pituutta ole rajoitettu.

Laatan merkkien tulostus näkyville riippuu valmistajasta ja laatan koko riippuu merkistä. Tämän takia lomakkeessa on alussa ainoastaan yksi valmistajien lista pudotusvalikossa (KUVIO 34). Riippuen valmistajan valinnasta, javascript-skriptikielen avulla pudotusvalikoissa luodaan merkkien ja koon listat (KUVIO 27).

```
<tr><td>
Valmistaja: </td><td>
<select name=proizv id=proizv onchange=\"set_proizv(this);\">
<option selected>Valitse</option>".$proizv."</select><br>
<span  id=\"marka\"></span>
<span  id=\"razmer\"></span>
</td></tr>
```

KUVIO 27. Valmistajien lista pudotusvalikossa ja javascript-käsittelijä

Valmistajien listan tunnus (id) on proizv ja javascript-tapahtuman käsittelijän tunnus on set_proizv(). -tagit eivät alustavasti ole näkyvissä, niissä luodaan myöhemmin merkkien ja koon listat. PHP-kielen muuttuja $proizv sisältää täyden listan valmistajista muodossa <option>Valmistaja1</option>, <option>Valmistaja2</option> ja niin eteenpäin.

Valmistajien listan vastaanotto tapahtuu sivun lataamisessa vastaavan SQL-kyselyn avulla (KUVIO 28):

```
$proizv='';//Valmistajien lista tietokannasta
$a1=$_SESSION['base'];
$sql = "SELECT * FROM `plitka_razmer` where `base`=\"$a1\" group by `proizv`";
$result = mysql_query($sql);
while ( $postrow[] = mysql_fetch_array($result));
$posts=count($postrow)-1;
for($i = 0; $i < $posts; $i++)
{$proizv=$proizv."<option>".$postrow[$i][proizv]."</option>"; }
```

KUVIO 28. Laatan valmistajien tulostus tietokannasta pudotusvalikossa

Muuttuja $a1 sisältää käytössä olevan työtietokannan nimen, $postrow[]-taulukko sisältää tietokannan palvelimen vastauksen ja FOR-silmukan avulla muodostetaan $proizv-muuttuja.

Tapahtuman käsittelijäfunktio (KUVIO 29) set_proizv() saa käyttäjän pudotusvalikossa valitseman valmistajan (el.options[el.selectedIndex].value) ja muodostaa merkkien listan, jossa on puolestaan set_razm()-käsittelijäfunktio.

```
function set_proizv(el)
{
    var obj=el.options[el.selectedIndex].value;
    var opt='<select name=pr id=pr onchange=\"set_razm(this);\">
    <option selected>Valitse</option>';
```

KUVIO 29. Onchange-tapahtuman käsittelijäfunktio

Väliaikaiseen b[]-taulukkoon siirtyy konkreettiselle valmistajalle kuuluva merkkien lista (KUVIO 30):

```
for (var k=0; k<b.length; k++)
    if (b[k]!='') {opt=opt+'<option>'+b[k]+'</option>';}
opt=opt+'</select>';
document.getElementById(\"proizv\").innerHTML = opt;
document.getElementById(\"razmer\").innerHTML = '';
```

KUVIO 30. Laatan merkkien listan muodostus ja tulostus

FOR-silmukan avulla muodostetaan pudotusvalikossa merkkien lista (opt-muuttuja) muodossa <option>Merkki1</option> ja niin eteenpäin. Valmiin merkkien listan rivit kirjoitetaan span-tagien väliin .innerHTML-metodilla. Samalla tavalla muodostetaan ja täytetään pudotusvalikossa koon lista.

Uuden tilauksen lisääminen järjestelmään tehdään varastotyöntekijän ohjelmamoduulin avulla tilauksen lisäämisen sivulla. Välitettävät tiedot tarkistetaan niin, että lomakkeen kenttiin syötetyn tekstin pituuden pitää olla yli 2 merkkiä (KUVIO 31). Virheen tapahtuessa käyttäjä saa vastaavan varoituksen.

```
if ((strlen($a2)<3)OR(strlen($a3)<3)OR(strlen($a12)<3))
{echo '<center>Kaikki kentät eivat ole täytetty</center>';}
```

KUVIO 31. Syötettyjen tietojen tarkistus

Tilauksen lisäämisessä tuotteelle voidaan antaa paikka varastossa. Kuitenkin ennen uusien tietojen tallentamista järjestelmään skripti tarkistaa, voiko annettu paikka olla jo varattu, sillä samalla paikalla ei voi olla samanaikaisesti 2 eri tilausta. Jos lomakkeessa tuotteelle annettiin paikka, joka on jo varattu, tulostetaan vastaava viesti, että annettu paikka on varattu (KUVIO 32).

```
$sql = "SELECT * FROM `plitka_base` where `razmeshenie`=\"$a12\"";
$result = mysql_query($sql);unset($po);
while ( $po[] = mysql_fetch_array($result));
if ($po[0][razmeshenie]!='')
{$err_razm= '<center><a href=klad.php?s='.$po[0][id].'>Paikka on jo
varattu. </a></center>';}
```

KUVIO 32. Paikan tarkistus uuden tilauksen lisäämisessä

Muuttujassa $a12 pysyy käyttäjän antama ehdotus paikaksi varastossa. Samalla muodostetaan SQL-kysely, jolla tarkistetaan, onko käyttäjän ehdottama paikka jo varattu. Jos on, tulostetaan vastaava viesti ja linkki sivulle, josta näkyvät paikassa jo olevan tilauksen tiedot.

Kun uusi tilaus lisätään tietojärjestelmään, muodostetaan seuraava SQL-kysely (KUVIO 33).

```
$sql = "INSERT INTO `plitka_base`
(`fio`,`pasp`, `razmer`,`kol`,
`marka`,`proizv`,`razmeshenie`,`base`,`def`,`def_op`)
VALUES
('$a2','$a3', '$a10','$a11','$a8','$a9','$a12','$a1','$a13','$a14')";
```

KUVIO 33. Tietojen lisääminen tietokantaan

Tässä SQL-kyselyssä muuttujilla on seuraavat merkitykset (TAULUKKO 1):

TAULUKKO 1. Muuttujien merkitykset uuden tilauksen lisäämisessä

$a1=$_SESSION[‘base’]	Työtietokannan nimi
$a2=$_POST[‘fio’]	Tilaajan nimi
$a3=$_POST[‘pasp’]	Tilaajan lisätiedot
$a8=$_POST[‘marka’]	Valitun tuotteen merkki
$a9=$_POST[‘pr’]	Tuotteen valmistaja
$a10=$_POST[‘razmer’]	Valittu tuotteen koko
$a11=$_POST[‘kol’]	Tuotteen määrä
$a12=$_POST[‘razmeshenie’]	Paikka varastossa
$a13=$_POST[‘foto’]	Kuvaus hylkytavarasta (kuva) (jos olemassa)
$a14=$_POST[‘def_op’]	Kuvaus hylkytavarasta (teksti) (jos olemassa)

Edellä mainitut lähdekoodin osat, jotka liittyvät tilauksen lisäämiseen, ovat käytössä tilauksen lisäämisen sivulla, jota varastotyöntekijä kutsuu tarvittaessa päävalikosta. Kyseinen sivu käyttöliittymässä on seuraavan näköinen (KUVIO 34):

KUVIO 34. Uuden tilauksen lisäämisen lomake

Kuviossa 34 esitetyn lomakkeen avulla voidaan lisätä tietojärjestelmään varastoon saapunut asiakkaan tilaus. Merkin ja koon pudotusvalikot tulevat javascript-skriptikielen avulla näkyviin automaattisesti, kun valitaan valmistaja. Tämä

prosessi on kuvattu edellä. Lisäksi, kun yritetään tallentaa syötetyt tiedot järjestelmään, tapahtuu varaston paikan ja annetun tekstin pituuden tarkistus. Kun tilaus tallennetaan järjestelmään, tapahtuu kuviossa 33 esitetty SQL-kysely. Jos varastoon saapunut tuote on hylkytavara, vastaavassa kohdassa tilauksen lisäämisen sivulla voidaan valita kyllä-vaihtoehto, jolloin näkyviin saadaan kentät, joilla voidaan kuvata hylkytavaraa.

Varastotyöntekijällä ja johtajalla on käytössä sivu, jossa on tilauskohtainen tieto tilauksesta. Tämä sivu tulee näkyviin, kun varastotyöntekijä (tai johtaja) valitsee hiirellä tilausten listalta asiakkaan nimen. Tilauskohtaisen sivun avulla johtaja voi pelkästään katsoa asiakkaan ja tuotteen tietoja, mutta varastotyöntekijä pystyy merkitsemään tilauksen vastaanotetuksi tai päivittämään tilauksen tietojärjestelmässä jo olevien asiakastietojen perusteella, mikä nopeuttaa tilauksen syöttämistä järjestelmään. Käyttöliittymässä tilausten lista varastotyöntekijällä ja johtajalla on tehty seuraavan taulukon ja sen sarakkeiden muodossa (TAULUKKO 2 ja sivu 47):

TAULUKKO 2. Tilausten listan yleinen esitysmuoto

#	Nimi	Merkki	Valmistaja	Koko	Määrä	Paikka	Hylky

Nimi-sarakkeessa on listattu käyttäjän nimet linkkien muodossa. Linkeillä päästään tilauskohtaisille sivuille. Tilauksen numero saadaan tietokannasta. Jokainen järjestelmään tallennettu tilaus saa oman numeron, joka myös tulostetaan tilausten listalla. Varastotyöntekijän käytössä olevan tilauskohtaisen sivun katsomisen ohjelmamoduuli on toteutettu show.php-tiedostossa, joka tarvittaessa ladataan varastotyöntekijän perusmoduulissa. Tiedot tilauksen numerosta välitetään palvelimelle GET-metodilla ja linkki on seuraavan näköinen: klad.php?s=133. Saadut parametrit käsitellään moduulissa (KUVIO 35):

```
//GET-kysely tilauskohtaisen sivun katsomiseen
$s='';
$s= $_GET["s"];
$_SESSION['s']=$s;
if ($s!='') {include('show.php');$st='no';}
```

KUVIO 35. GET-kyselyn tietojen käsittely

Moduulissa show.php tapahtuu yhteyden muodostus tietokantaan ja tarvittavien tietojen valitseminen tilauksesta, jonka numero on $a1 (KUVIO 36):

```
$a1=$_SESSION['s'];
$sql = "SELECT * FROM `plitka_base` where `id`=\"$a1\"";
```

KUVIO 36. Tietojen saaminen kannasta tilauksen numeron perusteella

Saadut tiedot siirtyvät $po[]-taulukkoon ja niitä käytetään tilauskohtaisen sivun vastaavien kenttien täyttämiseen (KUVIO 37):

```
<tr><td width=50>Asiakkaan nimi: </td>
<td><input type=text name=fio value=\"".$po[0][fio]."\">
```

KUVIO 37. Tietojen tulostus tilauskohtaisella sivulla

Varastotyöntekijällä tilauskohtaisen sivun huomautettaviin ominaisuuksiin kuuluvat toiminnot, joilla voidaan päivittää tilaus, merkitä tilaus vastaanotetuksi, lisätä tilaukseen kuvaus hylkytavarasta tai katsoa hylkytavaran kuvia. Nämä toiminnot saadaan aikaan vastaavien nappien avulla. Kun asiakas tulee hakemaan tilauksensa, varastotyöntekijä näkee tilauskohtaisella sivulla mainitut asiakastiedot. Tämän jälkeen varastotyöntekijä voi merkitä tilauksen vastaanotetuksi painamalla ”vastaanotto”-nappia, joka sijaitsee tilauskohtaisella sivulla. Napin painamisen jälkeen tietojärjestelmästä ja sen tietokannasta poistuu luovutetun tuotteen paikka (KUVIO 38). Tiedot tilauksesta (asiakkaan nimi, lisätiedot, valmistaja) jäävät kuitenkin järjestelmään. Myöhemmin niitä voidaan ehkä käyttää kirjanpitoa varten. Lisäksi vanhojen tietojen perusteella voidaan syöttää nopeasti uusi tilaus järjestelmään. Tämä tapahtuu, kun tilauskohtaisella sivulla valitaan uuden tilauksen ominaisuudet ja painetaan ”tallenna”-nappia.

```
$c1=$_POST['z_id'];
$sql = "UPDATE `plitka_base` SET  razmeshenie='' WHERE id='$c1'";
if (mysql_query($sql, $db)) {}else {echo mysql_error();}
```

KUVIO 38. Kyselyn käsittely, kun tuote merkitään vastaanotetuksi

Kuvion 38 tapauksessa muodostetaan SQL-kysely taulun tietojen päivittämiseen rivillä, jolla tilauksen numero on $c1-muuttujassa.

Tilauskohtaisella sivulla sijaitsee päivitysnappi (KUVIOT 39 ja 42), jonka avulla voidaan lisätä uusi tilaus järjestelmään ja joka käyttää javascript-käsittelijää valmistajien, merkkien ja koon pudovalikoissa olevien listojen muodostamiseen:

```
<a class=button onclick=\"show();\">Paivita</a><br>
```

KUVIO 39. Päivitysnappi tilauskohtaisella sivulla

Tilauskohtaisen sivun Katso-nappi avaa paneelin (KUVIO 40 ja sivu 52), jossa on mahdollisia valokuvia hylkytavarasta, jos hylky olemassa:

```
document.getElementById(\"def_w\").innerHTML = '".$po[0][def]."';
```

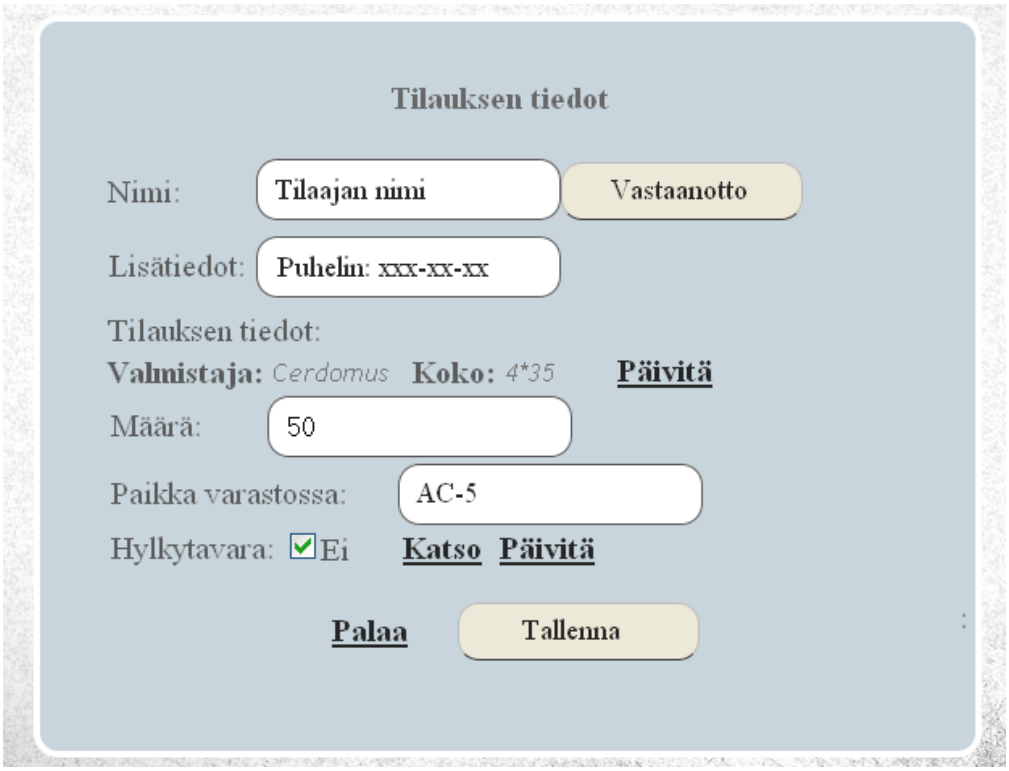
KUVIO 40. Kuvapaneelin avaaminen

Tilauskohtaisen sivun Palaa-napilla varastotyöntekijä siirtyy omalle aloitussivulle ilman muutoksia (KUVIO 41). Jos kyseessä on johtaja, hän siirtyy johtajan ohjelmamoduuliin, joka on dir.php-tiedosto. Jos kyseessä on varastotyöntekijä, hän siirtyy varastotyöntekijän ohjelmamoduuliin, joka on klad.php-tiedosto.

```
if ($_SESSION["state"]=='Johtaja') {$s="dir.php";}
else {$s="klad.php";}
<a  class=button href=".$s."> Palaa </a>
```

KUVIO 41. Paluu tilauskohtaiselta sivulta ilman muutoksia

Edellä kuvattu tilauskohtainen sivu käyttöliittymässä on seuraavan näköinen (KUVIO 42):



KUVIO 42. Tilauskohtainen sivu käyttöliittymässä

Kuviossa 42 esitetty sivu on varastotyöntekijän käytössä. Kun varastotyöntekijä valitsee asiakkaan omalta tilausten listalta, hän saa tämän sivun näkyviin. Asiakkaan ja tilauksen tiedot ovat tässä tapauksessa jo valmiiksi täytettyjä kenttiä. Tämän ikkunan avulla varastotyötekijä voi merkitä tilauksen vastaanotetuksi, syöttää uuden tilauksen järjestelmään tai esimerkiksi nähdä asiakkaan yhteystiedot. Uuden tilauksen lisääminen tapahtuu ylemmän Päivitä-napin avulla, tietojen syöttämisellä ja niiden tallentamisella. Vastaavasti tilauksen merkitseminen vastaanotetuksi tapahtuu kuvassa olevalla Vastaanotto-napilla.

Joskus varastoon voi saapua hylkytavara. Tämä koskee myös keraamista laattaa. Jotta tämä voitaisiin esittää tietojärjestelmässä, päävalikon kautta tapahtuvan tilauksen lisäämisen sivulla (ja myös tilauskohtaisella sivulla) on olemassa kenttä hylkytavaran kuvaamiseen. Kentän avulla hylkytavara voidaan kuvata sekä sanallisesti että kuvallisesti. Oletusarvona tilauksen lisäämisen sivulla tuote ei ole hylkytavara, ja kenttä hylkytavaran kuvaamiseen on piilossa. Tällöin lomakkeessa (sivu 40) näkyvät ainoastaan kaksi radionappia, jotka vastaavasti tarkoittavat, että kyseessä on joko hylkytavara tai ei. Näille napeille on määritelty seuraavat onchange-tapahtumat ja vastaavat käsittelijät (KUVIO 43):

```
<input type=radio name=def onchange=\"set_defekt(this);\" value='kylla'>Kyllä
<input type=radio name=def onchange=\"hide_defekt(this);\" value='ei'>Ei
```

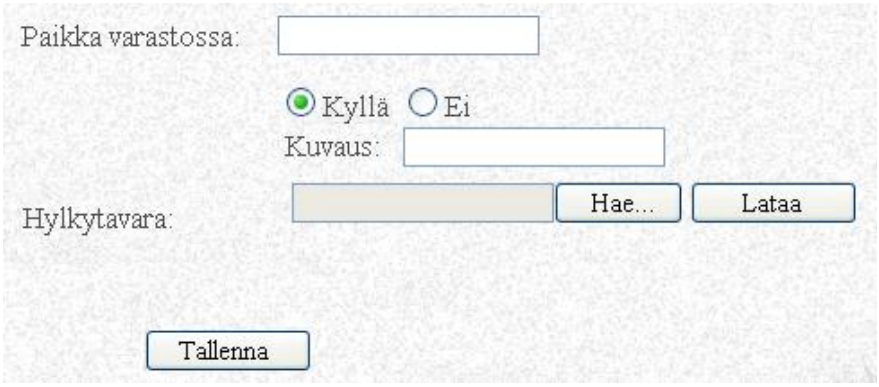
KUVIO 43. Javascript-käsittelijät hylkytavara-kentän tulostukseen

Kun valitaan Kyllä-elementti, innerHTML-metodin avulla sivulle liitetään koodi, jolla tulee näkyviin tekstikenttä hylkytavaran sanalliseen kuvaukseen sekä kehys, jolla ladataan palvelimelle tarvittava määrä kuvia (KUVIO 44):

```
function set_defekt(el)
{
    document.getElementById(\"def_op\").innerHTML = 'Kuvaus: <input
    type=text name=def_opisanie><br><iframe src=upload.php
    frameborder=0 width=400 height=120></iframe>';
}
```

KUVIO 44. Javascript-käsittelijä hylkytavara-kenttää varten

Hylkytavaran kuvaamiseen tarkoitetun kentän, joka sijaitsee tilauksen lisäämisen sivulla, näkymä käyttöliittymässä on esitetty seuraavassa kuvassa (KUVIO 45):



KUVIO 45. Lisäkenttä hylkytavaran kuvaamiseen

Kuviossa 45 olevalla lomakkeella voidaan kirjata tietojärjestelmään tiedot hylkytavarasta. Sanallinen kuvaus voidaan kirjoittaa Kuvaus-kenttään. Kuvallisessa kuvauksessa varastotyöntekijä tekee kameralla kuvan hylkytavarasta, hakee kuvan tietokoneelta ja lataa sen. Tilaus kirjataan järjestelmään, kun painetaan Tallenna-nappia.

Tiedostojen (kuvien) lataamisen ohjelmamoduuli on toteutettu erillisessä upload.php-tiedostosta. Palvelimen puolella luodaan myös oma upload-kansio, johon tallennetaan järjestelmään ladatut kuvat hylkytavarasta. Kansiolla on oikeudet tiedostojen lukemiseen, kirjoittamiseen ja suoritukseen. Kun ladataan uusi kuva palvelimelle, sille muodostetaan satunnaisten lukujen avulla uusi nimi. Jos nimi on jo käytössä, toimenpide toistetaan (KUVIO 46).

```
$uploaddir="upload/";
do $key=mt_rand(100000,99999999);
while (file_exists("$uploaddir/$key.jpg"));
$name_file=$key.'.jpg';
$uploadfile = $uploaddir.$name_file;
```

KUVIO 46. Ladatun kuvan satunnaisen nimen muodostus

Kansio, johon tallennetaan kuvat, määritellään $uploaddir-muuttujassa. Uuden nimen omaavan kuvan osoite sijaitsee puolestaan $uploadfile-muuttujassa. Superglobaali-tyyppisessä $_SESSION-taulukossa kaikkia kuvia varten muodostetaan seuraavan mallin (KUVIO 47) mukainen HTML-rivi:

```
$_SESSION['foto'].='<img src='.$uploadfile.'><br>';
```

KUVIO 47. Ladatun kuvan nimen tallentaminen istuntomuuttujaan

Tällä tavalla saadut tiedot lisätään plitka_base-tauluun ja sijoitetaan kahteen sarakkeeseen: DEF-tauluun, jossa on ladattujen kuvien lista, ja DEF_OP-tauluun, jossa on hylkytavaran sanallinen kuvaus tekstimuodossa.

Varastotyöntekijällä tilausten listassa on yksinkertainen visuaalinen merkki siitä, onko tuote hylkytavara vai ei. Tilausten listalla olevassa hylky-sarakkeessa punaisella värillä merkitään tilaukset, jotka ovat varastoon saapuneita hylkytuotteita. Jos tuote on kunnossa, se merkitään vihreällä värillä. Tämä tapahtuu tilausten listan muodostamisessa (KUVIO 48).

```
<td>Hylkytavara</td>
if ($postrow[$i][def]==''){echo "<td align=center><font color=green>Ei</font></td>";}
else {echo "<td align=center><font color=red>Kyllä</font></td>";}
```

KUVIO 48. Hylkytuotteen merkitseminen tilausten listalla

Taulukossa $postrow pysyvät tiedot kaikista tilauksista. Kuvion 48 koodissa tarkistetaan hylkytavaran kenttä tietokannasta. Jos se on tyhjä, tilausten listalla olevan Hylky-kentän teksti on vihreää. Jos tietokannan kenttä ei ole tyhjä, Hylky-kentässä oleva teksti on punaista. Käyttöliittymässä hylkytavaran näkymä kaikkien tilausten listalla on seuraavan näköinen (KUVIO 49):

Varastotyöntekijä: Työntekijä 1 työtietokanta Lattialaatat

#	Nimi	Merkki	Valmistaja	Koko	Määrä	Paikka	Hylky
137	Asiakas 1	BARCELONA	Cerdomus	4*35	50	AB-5	ei
133	Asiakas 3	MICRA	Cristal et Bronze	30*30	45	AC-7	ei
132	Asiakas 4	KUBIC	Cristal et Bronze	3*30	12	BA-2	ei
131	Asiakas 2	K-2	Porndor	30*30	12	BA-4	kyllä

KUVIO 49: Varastotyöntekijän tilausten lista käyttöliittymässä

Kuviossa 49 on esitetty yleinen näkymä varastotyöntekijän tilausten listasta. Tiedot tilauksista esitetään taulukon muodossa, mikä on kätevä esitystapa. Kuvassa näkyy myös aikaisemmin mainittu hylkytavaran merkitseminen.

Teoriaosuudessa todettiin, että hakutoiminto järjestelmässä olisi hyödyllinen, erityisesti jos tietojen määrä järjestelmässä tulee kasvamaan koko ajan. Tietojärjestelmässä on toteutettu hakutoiminto, joka on käytössä varastotyöntekijällä. Hakutoiminto voidaan kutsua kahdella eri tavalla. Päävalikosta voidaan kutsua täysi hakukenttä, jossa on 3 kenttää tietojen syöttämiseen. Sarakekohtainen hakukenttä, jossa on 1 kenttä tietojen syöttämiseen, voidaan puolestaan kutsua painamalla sarakkeen otsikkoa. Kaikissa tapauksissa haku on mahdollinen asiakkaan nimen, tuotteen merkin ja paikan mukaan. Hakukenttään pitää kirjoittaa hakuehdon koko nimi. Kaikissa tapauksissa hakutoiminnon näkyviin tulo ja pois näkyvistä meno on toteutettu javascript-skriptikielen avulla.

Sarakekohtaisessa haussa tilausten listan jokaisella otsikolla, jossa haku on mahdollinen (nimi, merkki ja paikka), on oma tunnus ja käsittelijä, joka käynnistyy onclick-tapahtuman avulla (KUVIO 50).

```
<td><span onclick=\"show_fio();\">Nimi</span>
<span id=f_fio style=\"display:none\">
<form action='' method=post><input type=text name=find_fio>
<input type=submit name=f_fio value=Etsi>
<a class=\"button\" onclick=\"hide_fio();\">
<font color=red>X</font></a>
</form></span></td>
```

KUVIO 50. Hakuotsikot tilausten listassa

Esiin tuleva sarakekohtainen hakulomake sijaitsee span-tagien välillä, missä hakulomakkeen näkyvyyden oletusarvo on asetettu ei-arvoksi. Tällä tavalla ilman otsikon painamista hiirellä hakulomake ei ole sivulla näkyvissä. Lomakkeen käsittelijä on sama ohjelmamoduuli, josta se kutsuttiin. Tiedot pavelimelle välitetään POST-metodin avulla. Lomakkeessa on vain kolme toiminnallista

elementtiä: tekstikenttä tietojen syöttämiseen, etsi-nappi sekä x-merkki, joka sulkee hakulomakeen. Tämä x-merkki toimii linkkinä javascript-käsittelijään, joka sulkee hakulomakeen, eli tekee sen näkymättömäksi (KUVIO 51).

```
function hide_fio()
{document.getElementById(\"f_fio\").style.display=\"none\";}
```

KUVIO 51. Hakulomakkeen sulkemisen käsittelijä

Haku on saatavissa myös päävalikosta. Kun siellä painetaan hakutoimintoa, käynnistyy onclick-tapahtuma ja show_f()-käsittelijä, joka asettaa hakulomakkeen näkyvyyden arvoon ”kyllä” (KUVIO 52). Tässä tapauksessa varastotyötekijällä on käytössä kaikki hakukentät samanaikaisesti. Kentät voidaan myös täyttää samanaikaisesti. Myös etsi-nappi ja x-merkki, joka sulkee hakulomakkeen, on käytössä.

```
document.getElementById("f_gen").style.display="block";
```

KUVIO 52. Hakulomakkeen kutsu

Kokonainen hakukenttä, joka kutsutaan päävalikosta, sekä sarakekohtainen hakukenttä, joka koskee yhtä saraketta, ovat käyttöliittymässä seuraavan näköisiä (KUVIO 53):

KUVIO 53. Hakukentät käyttöliittymässä

Kuviossa 53 ylemmällä näkyy kokonainen hakutoiminto, joka tulee esiin, kun se valitaan päävalikosta. Nimi-kentässä oleva haku on sarakekohtainen haku, joka tulee esiin, kun painetaan hiirellä Nimi-kenttää tilausten listalla.

Kun hakutiedot syötetään kenttiin, tilausten listan ohjelmamoduuli muodostaa kyselyn tietokannan palvelimelle (KUVIO 54):

```
$c1=$_SESSION['base'];
$c2='';
if (isset($_POST['f_fio'])) {$c2='AND fio=\''.$_POST['find_fio']."'";}
if (isset($_POST['f_marka'])) {$c2='AND marka=\''.$_POST['f_marka']."'";}
if (isset($_POST['f_razm'])) {$c2='AND razmeshenie=\''.$_POST['find_razm']."'";}
if (isset($_POST['f_top_gen']))
 {$c2='';
if ($_POST['find_fio']!=''){$c2.='AND fio=\''.$_POST['find_fio']."'";}
if ($_POST['f_marka']!=''){$c2.='AND marka=\''.$_POST['f_marka']."'";}
if ($_POST['find_razm']!=''){$c2.='AND
razmeshenie=\''.$_POST['find_razm']."'";}}
$sql="SELECT * FROM `plitka_base` where base='$c1' $c2 AND org=''
order by `id` desc";
$result = mysql_query($sql,$db);
```

KUVIO 54. Hakukyselyn muodostus

Edellä mainitussa koodissa $c2-muuttuja tarvitaan kyselyn muodostamiseen seuraavan mallin mukaan: fio=’nimi’ AND marka=’merkki’ AND razmeshenie=’paikka’. Jos haku tapahtuu yhdessä sarakkeessa, c2-muuttujan arvo kirjoitetaan uudelleen joka kerta. Näin kysely tulee sisältämään aina yhden parametrin. Jos haku kutsuttiin päävalikosta, jossa on usean hakuehdon haku, c2-muuttujan arvo päivittyy ja lisää uuden parametrin tietokannan palvelimen SQL-kyselyyn.

5.5.3 Johtaja

Onnistuneen sisäänkirjautumisen jälkeen johtajalla on päävalikossa kaksi toimintoa: tilausten listan tarkistus kaikista työtietokannoista ja päiväkirja. Näillä toiminnoilla johtaja voi seurata muutoksia varaston toiminnassa. Päiväkirja on tarkoitettu tilaustenkäsittelyn valvontaan ja antaa johtajalle kuvan siitä, mitä operaatioita varastossa tapahtuu. Operaatiot ovat esimerkiksi uuden tilauksen lisääminen

järjestelmään tai tilauksen luovuttaminen asiakkaalle. Nämä operaatiot ja niiden merkitseminen tietojärjestelmään ovat varastotyöntekijöiden vastuulla. Päiväkirjan tapauksessa staattiset tiedot kirjoitetaan tietokannan erilliseen tauluun, joka on nimeltään plitka_stat. Taulussa on seuraavat kentät (TAULUKKO 3):

TAULUKKO 3. Johtajan päiväkirjan sisältö

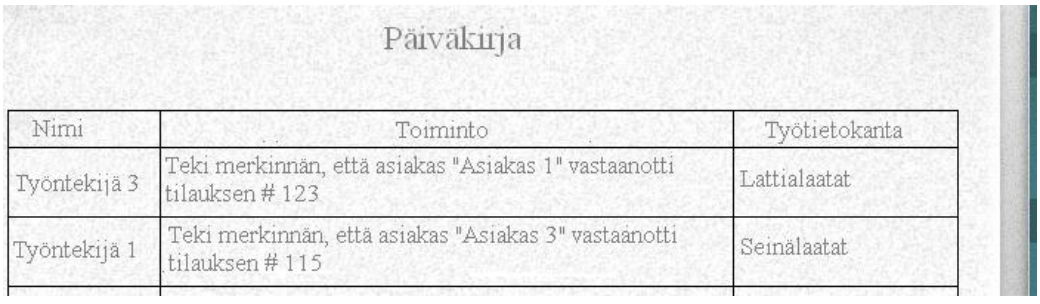
FIO	ACTION	BASE
Tehtävän suorittaneen varastotyöntekijän nimi	Toiminnon kuvaus	Varastotyöntekijän työtietokanta

Tietojen lisääminen tähän tauluun tapahtuu SQL-kyselyn avulla (KUVIO 55):

```
$b1=$_POST[fio];
$b4=$_SESSION[fio];
$b2=$_SESSION[base];
$b3='teki merkinnän, että '.$b1.' vastaanotti tilauksen #'.$c1;
$sql = "INSERT INTO `plitka_stat`
(`fio`,`action`, `base`) VALUES ('$b4','$b3','$b2')";
```

KUVIO 55. Tietojen lisääminen tietokantaan

Käyttöliittymässä päiväkirja on seuraavan näköinen (KUVIO 56):



KUVIO 56. Johtajan päiväkirja käyttöliittymässä

Kuviossa 56 näkyy päiväkirja, joka on johtajan käytössä. Nimi-kenttään kirjoitetaan varastotyöntekijän nimi. Asiakkaalla ja numerolla tässä kuvassa merkitään

puolestaan asiakkaan nimeä. Kuvassa näkyvät toiminnot kirjataan päiväkirjaan, kun varastotyöntekijä merkitsee tuotteen vastaanotetuksi. Lisäksi päiväkirjaan kirjoitetaan tieto tilauksen lisäämisestä: kun järjestelmään syötetään uusi tilaus, päiväkirjaan tulee lause, että uusi tilaus on syötetty järjestelmään.

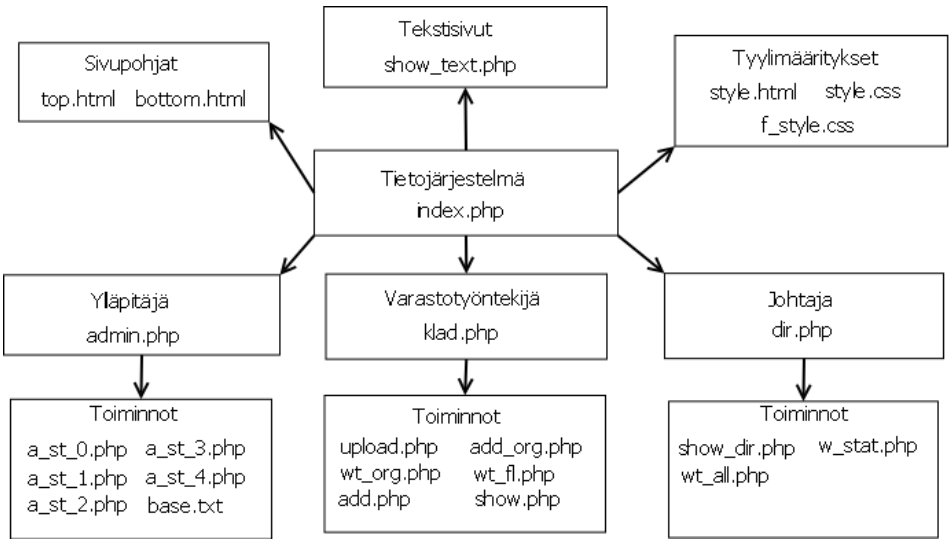
Johtajalla on käytössä lista kaikista tietojärjestelmässä olevista tilauksista. Lisäksi johtaja pystyy näkemään tilauskohtaiset sivut. Tämän tilauskohtaisen sivun ohjelmamoduuli on toteutettu johtajalla erillisessä show_dir.php-tiedostossa, joka ladataan tarvittaessa johtajan dir.php-perusmoduulissa. Johtajalla on käytössä samanlainen kuin varastotyöntekijällä, mutta toiminnallisuudeltaan paljon suppeampi tilauskohtainen sivu. Sivulla on ainoastaan kaksi nappia, joilla voidaan katsoa kuvat hylkytavarasta ja palata takaisin tilausten listaan (KUVIO 57):

KUVIO 57. Hylkytavaran kuvaus tilauskohtaisella sivulla

Kuviossa 57 on esitetty tilauskohtaisella sivulla oleva sanallinen ja kuvallinen kuvaus hylkytavarasta. Tämä toiminnallisuus on toteutettu varastotyöntekijän ja johtajan rooleilla.

5.6 Ohjelmamoduulit

Tietojärjestelmän projekti koostuu ohjelmamoduuleista, sivupohjista sekä aputiedostoista. Lisäksi järjestelmä käyttää joukkoa kansioita. Kansio images sisältää 15 graafista tiedostoa, joita käytetään sekä etu- että sisäsivujen muotoiluun. Kansio upload sisältää kuvia, joita varastotyöntekijä lataa hylkytavaran kuvaamista varten. Tällä kansiolla pitää olla oikeudet tiedostojen lukemiseen, kirjoittamiseen ja suorittamiseen. Seuraavassa kuvassa (KUVIO 58) on esitetty tietojärjestelmän ohjelmamoduulit ja niiden kuuluminen järjestelmän rakenteeseen.



KUVIO 58. Järjestelmän moduulit

Järjestelmän päätiedosto on index.php. Tämä tiedosto muodostaa etusivun, tarkistaa kirjautumisen yhteydessä mainitut tunnukset ja salasanat. Tiedosto sisältää linkkejä staattisille sivuille.

Ohjelmamoduuli a_st_0.php sisältää skriptin työtietokantojen muokkaamiseen. Moduuli liitetään ainoastaan ylläpitäjällä, ja se antaa mahdollisuuden lisätä, muokata ja poistaa työtietokannat. Moduulissa avataan tekstitiedosto, joka sisältää työtietokantojen listan, jota voidaan muokata. Tietojen käsittely selaimessa tapahtuu textarea-tagien välillä. Tekstitiedosto base.txt on tekstimuotoisen tietokannan tiedosto. Se sisältää ylläpitäjän luomat työtietokannat. Myös niiden muokkaus ja

poisto tapahtuu tässä tekstitiedostossa. Tekstitiedosto on yhteydessä muihin ylläpitäjän moduuleihin.

Ohjelmamoduuli a_st_1.php on tarkoitettu uuden käyttäjän lisäämiseen järjestelmään. Moduuli liitetään ainoastaan ylläpitäjällä, ja se antaa mahdollisuuden lisätä uudet käyttäjät, määritellä oikeudet (roolit) ja työtietokannan, asettaa uudelle käyttäjälle salasanan ja tunnuksen. Ohjelmamoduuli a_st_2.php sisältää skriptin, jolla tulostetaan kaikki järjestelmän käyttäjät näkyviin. Moduuli liitetään vain ylläpitäjällä, ja se antaa mahdollisuuden listata kaikki käyttäjät sekä poistaa ne järjestelmästä. On mahdollista valita enemmän kuin yksi käyttäjä poistamiseen. Tiedot käyttäjistä saadaan tietokannasta.

Ohjelmamoduuli a_st_3.php sisältää skriptin, jolla lisätään tietokantaan uudet valmistajat, merkit ja koko. Moduuli on käytettävissä vain ylläpitäjällä, ja sillä voidaan lisätä uudet valmistajat, merkit ja koot järjestelmään. Ohjelmamoduuli a_st_4.php sisältää puolestaan ”tervehdystekstin”, joka tulee sisäänkirjautumisen jälkeen ja ohjaa ylläpitäjää valitsemaan tarvittavan toiminnon päävalikosta. Kyseinen tiedosto on käytettävissä vain ylläpitäjällä ja sillä on informatiivinen merkitys.

Ohjelmamoduulissa add.php on toiminnallisuus yksityisasiakkaan tilauksen lisäämiseen tietojärjestelmään. Moduuli on saatavissa vain varastotyöntekijällä. Moduuli mahdollistaa tietojen syöttämisen, muodostaa lomakkeet ja javascript-koodin avulla valmistajien ja merkkien listat pudotusvalikoissa. Tämä moduuli välittää tiedot POST-metodilla klad.php-varastotyöntekijän perusmoduuliin. Ohjelmamoduuli add_org.php sisältää puolestaan skriptin, joka on tarkoitettu yritysasiakkaan tilauksen lisäämiseen tietojärjestelmään. Moduuli on käytössä vain varastotyöntekijällä. Moduuli suorittaa samat toimenpiteet kuin add.php-moduuli, mutta koskee yritysasiakkaita.

Ohjelmamoduuli admin.php on perusmoduuli ylläpitäjän toiminnallisuudessa. Moduuli käsittelee käyttäjän kyselyjä, lataa muut moduulit tarvittaessa, toimii niiden kanssa yhteistyössä. Varastotyöntekijän perusmoduuli on klad.php-tiedosto.

Se sisältää GET-kyselyjen käsittelijän, varastotyöntekijän perustoiminnot, helpottaa työtä tietokannan kanssa, liittää muut moduulit. Ohjelmamoduuli dir.php on johtajan perusmoduuli tietojärjestelmässä. Tiedosto mahdollistaa tilausten listan ja päiväkirjan katsomisen.

Tiedosto bottom.html on yksi staattisten sivujen sivupohjista. Se sisältää vähän tekstiä. Tiedoston top.html avulla puolestaan määritellään otsikoiden paikka, sivun rakenne, tekstiosat, kuvien sijoitus ja koko, käyttäjän valikko. Tiedosto style.html on aputiedosto, jossa on tyylimääritykset lomakkeita ja kenttiä varten. Kaikkien moduulien tyylistä vastaa style.css-tiedosto, jossa on tyylimäärityksiä. Toisessa f_style.css-tyylitiedostossa on myös erilaisissa moduuleissa käytettäviä tyylimäärityksiä.

Ohjelmamoduuli config.php on tiedosto, joka sisältää tietokannan asetukset, tunnuksen ja salasanan, jolla päästään tietokannan palvelimelle. Tiedostossa muodostetaan yhteys tietokantaan, mikä helpottaa muiden moduulien kehitystä ja käyttöä.

Tiedosto show.php on tilauskohtaisen sivun tiedosto. Tämä moduuli muodostaa staattisen HTML-sivun, jossa on tilaustiedot, käyttää hyväkseen tietokannasta saatuja tietoja ja lisää joukon nappeja. Kyseinen moduuli on varastotyöntekijän käytössä. Johtajalla puolestaan on melko samanlainen moduuli, joka on show_dir.php-tiedosto. Sen ero verrattuna varastotyöntekijän moduuliin on hyvin vähäinen toiminnallisuus, jolla voidaan ainoastaan katsoa tilauskohtaisia sivuja.

Tiedosto show_text.php on tekstisivujen katsomiseen tarkoitettu tiedosto. Kyseisten tekstisivujen sisältö ei ole toteutettu, mutta myöhemmin sivut voivat sisältää esimerkiksi aputietoja tai kuvauksen tietojärjestelmästä. Nämä tekstisivut ovat saatavissa etusivulla, ja niiden nimet ovat seuraavat: st_1.html, st_2.html, st_3.html, st_4.html.

Tiedosto upload.php sisältää moduulin, jolla ladataan kuvat palvelimelle. Oletusarvona tiedosto käyttää upload-kansiota. Tämä moduuli on käytössä varasto-

työntekijällä. Moduuli ladataan tilauksen lisäämisen sivulla, lomakkeessa kehyksessä, kun lisätään uusi tilaus järjestelmään.

Ohjelmamoduulissa wt_all.php on moduuli tilausten listan katsomiseen. Tämä moduuli on käytössä ainoastaan johtajalla. Lisäksi johtaja käyttää myös wt_stat.php-moduulia, jolla toteutetaan päiväkirjan toiminnallisuus. Varastotyöntekijän käyttämät wt_fl.php- ja wt_org.php-moduulit on tarkoitettu tilausten listan katsomiseen liittyen vastaavasti yksityis- ja yritysasiakkaisiin.

6 YHTEENVETO

Opinnäytetyön tavoite toteuttaa tietojärjestelmä web-sovelluksena varastoon tulevien tuotteiden tilaustenkäsittelyyn ja niitä koskevien tietojen kirjanpitoon toteutui onnistuneesti, sillä työssä on pyritty noudattamaan sille asetettuja asiakasvaatimuksia. Asiakasvaatimuksena oli toteuttaa sovellus, jonka avulla varaston toiminnassa tarvittavien asiakas- ja tilaustietojen kirjanpito olisi mahdollista. Toteutettu tietojärjestelmä erikoistuu juuri näiden tietojen käsittelyyn ja mahdollistaa sekä asiakas- että tilaustietojen ylläpidon. Tietojärjestelmän ensimmäinen versio tarjoaa perustoimintoja, joilla hoidetaan varastossa olevien tilausten tulo- ja menotoimenpiteitä.

Suunniteltaessa tietojärjestelmää varastoon pitää analysoida varaston toiminta ja tehtävät, jotka tietojärjestelmän on tarkoitus suorittaa. Teoriaosuudessa todettiin, että varastoja on erilaisia, ja jokin pieni tietojärjestelmä ei olisi tuonut hyötyä ison keskusvaraston tai läpivirtausvaraston toimintaan. Sen sijaan pienessä jakeluvarastossa tietojärjestelmää, jolla on yksinkertainen toiminnallisuus rekisteröidä tilaukset ja merkitä ne asiakkaan vastaanotetuksi, on hyödyllisempi käyttää muun muassa helppokäyttöisyytensä takia. Lisäksi isokokoinen varasto tarvitsisi isomman toiminnallisuuden lisäksi erikoistuneisuutta myös muihin materiaaleihin.

Tässä opinnäytetyössä tehtiin helppokäyttöinen web-sovellus, joka vastaa asiakasvaatimuksia. Toteutettu sovellus voisi helpottaa varastotyöntekijöiden työtehtäviä suorittaa tavallisia kirjanpidon toimenpiteitä: nähdä asiakkaat, heidän tilaukset ja niiden ominaisuudet, kuten tuotteen määrä, sen koko ja sijoitus varastossa. Toteutettu tietojärjestelmä palvelee teoriaosuudessa mainittuja varastojen perustoimintoja: tuotteen vastaanottoa, säilytystä sekä luovuttamista asiakkaalle. Sovelluksen on tarkoitus helpottaa näitä asioita, joten tämän perusteella tehtyä sovellusta voidaan sanoa tietojärjestelmäksi.

Jos sovelluksen kehitys aloitettaisiin uudestaan, järjestelmässä voitaisiin toteuttaa parempi toimintojen jako eri järjestelmän roolien keskellä ja esimerkiksi varastotyöntekijä voisi saada enemmän toimintoja. Lisäksi kehityksessä voitaisiin keskit-

työ lisätoimintoihin, joilla voitaisiin parantaa sovelluksen käyttömukavuutta ja tietokannan parempaan suunnitteluun, jotta tietokannan käyttö olisi tehokasta.

Tietojärjestelmän nykyversion jatkokehityksessä pyritään toteutettamaan väärien ja tarpeettomien tietojen poistoon tai päivitykseen tarkoitetut toiminnot. Lisäksi jatkokehityksessä otetaan huomioon seuraavat asiakasvaatimukset, joiden pohjalta suunnitellaan uutta toiminnallisuutta. Jatkokehityksessä voidaan parantaa sovelluksen käyttämää tietokantaa ja toteuttaa sille normalisointi, joka voisi estää päällekkäisen ja turhan tietojen syöttämisen ja muuttamisen.

Tällä hetkellä toteutettu tietojärjestelmä odottaa siirtymistä organisaation palvelimelle, josta sitä voidaan käyttää varaston toiminnassa. Järjestelmän kehittäjä ei voi taata järjestelmän täydellistä toiminnallisuutta teoriaosuudessa mainittujen syiden takia.

LÄHTEET

Anttila, J. 2001. Dokumenttien hallinta. Helsinki: Oy Edita Ab.

Gilmore, W.G. 2005. PHP & MySQL - Tehokas hallinta. Jyväskylä: Readme.fi.

Heinisuo, R. & Rauta, I. 2007. PHP ja MySQL. Tietokantaiset verkkopalvelut. 4. painos. Helsinki: Talentum Media Oy.

Hokkanen, S. & Virtanen, S. 2012. Varastonhoitajan käsikirja. Tallinna: Sho Business Development Oy.

Honeycutt, J. 2001. Tietämyksenhallinta. Helsinki: Edita Oyj.

Häkkinen, A. 2012. Mallinnus UML-katsaus [viitattu 16.10.2012]. Saatavissa: http://users.metropolia.fi/~hakka/Ohma/Ohma-02.pdf

IDEF. 2012. IDEF0 - Function Modeling Method [viitattu 17.10.2012]. Saatavissa: http://www.idef.com/IDEF0.htm

Kaario, K. & Peltola, T. 2008. Tiedonhallinta: Avain tietotyön tuottavuuteen. Porvoo: WS Bookwell.

Koskela, L., Koskinen, J. & Lankinen, P. 2007. Viestintä verkostoissa ja innovaatioissa. Juva: WS Bookwell Oy.

Laine, H. 2008. Ohjelmistojen mallintaminen. Tietovuokaaviot [viitattu 16.10.2012]. Saatavissa: www.cs.helsinki.fi/u/laine/malli/s08/pdf/dfd_c.pdf

Meloni, J. 2003. MySQL Trainer Kit. Helsinki: Edita Publishing Oy.

Paananen, J. 2005. Tietotekniikan peruskirja. Porvoo: WS Bookwell.

SeverStroiProekt. 2012. Palvelut [viitattu 4.10.2012]. Saatavissa: http://www.karelproekt.ru/ru/content/uslugi

Zandstra, M. 2001. PHP Trainer Kit. Helsinki: Oy Edita Ab.