



VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Sami Ranta

# ASIAKASRÄÄTÄLÖITY TARJOUS- LASKENTATYÖKALU

Escarmat Oy

Tekniikka

2025

## TIIVISTELMÄ

---

Tekijä	Sami Ranta
Opinnäytetyön nimi	Asiakasräätälöity tarjouslaskentatyökalu Escarmat Oy
Vuosi	2025
Kieli	suomi
Sivumäärä	42
Ohjaaja	Jani Ahvonen

Opinnäytetyö käsittelee asiakasräätälöidyn visuaalisen tarjouslaskentalisätyökalun toteutusta Escarmat Oy:lle.

Toimeksiantajan nykyinen tarjouslaskentajärjestelmä ei ole yhteensopiva erään asiakkaan tarjousten kanssa, minkä takia tämän asiakkaan tarjoukset on laskettu manuaalisesti taulukkolaskentapohjilla. Lisätyökalu tulisi toimimaan nykyisen tarjouslaskentatyökalun rinnalla. Kehitetyn työkalun avulla tarjouslaskenta automatisoituu ja tarjoukset voidaan lähettää asiakkaalle takaisin heidän haluamassaan lähdeformaattissa.

Sovellus on toteutettu CakePHP 2 -sovelluskehityksellä ja noudattaa täten MVC-arkkitehtuuria. Lisätyökalu hakee hintatiedot tietokannasta, laskee materiaalikustannukset, työkustannukset ja katteet ohjelmoidun logiikan perusteella.

Lopputuloksena syntyy käyttäjäystävällinen ja tehokas lisätyökalu, joka tehostaa tarjouslaskentaa. Sovellus mahdollistaa myös vanhojen tarjousten nopean hinnan tarkistuksen. Se on suunniteltu laajennettavaksi kattamaan myös muita asiakkuuksia tulevaisuudessa.

## ABSTRACT

---

Author	Sami Ranta
Title	Customer-tailored Quotation Calculation Tool
	Escarmat Oy
Year	2025
Language	Finnish
Pages	42
Name of Supervisor	Jani Ahvonen

The thesis deals with the implementation of a customer-tailored visual quote calculation tool for Escarmat Oy.

The company's current calculation system is not suitable for one specific customer's quotations, which is why these customers' quotes have been calculated manually using spreadsheet templates. The additional tool should work alongside the current quote calculation tool. With the help of the developed tool, quote calculation is automated, and quotes are sent back to the customer in the source format they prefer.

The application was implemented with the CakePHP 2 application framework and thus follows the MVC architecture. The additional tool extracts price information from the database, calculates material costs, labor costs and margins based on programmed logic.

The result is a user-friendly and efficient additional tool that helps quote calculation. The application also enables quick price checking of old quotes and is designed to be expanded to cover other customer accounts again.

---

Keywords quotation calculation, MVC, CakePHP 2, software development

# SISÄLLYS

TIIVISTELMÄ .....	2
ABSTRACT .....	3
1 JOHDANTO .....	7
1.1 ESCARMAT .....	7
1.2 Tausta ja aihevalinta .....	8
1.3 Työn tavoitteet.....	9
1.4 Työn rakenne.....	10
1.5 Tekoälyn käyttö tässä opinnäytetyössä.....	10
2 TEORIA JA TEKNIIKAT.....	11
2.1 Tarjouslaskenta ja sen merkitys .....	11
2.2 Aiemmat ratkaisut ja olemassa olevat työkalut .....	12
2.3 CakePHP 2 -sovelluskehys.....	13
2.4 MVC-arkkitehtuuri.....	14
2.5 Tietokantojen suunnittelu ja normalisointi.....	16
2.6 Tietoturva.....	17
3 KEHITYSPROSESSI JA TEKNINEN TOTEUTUS .....	19
3.1 Vaatimusmäärittely ja käyttäjävaatimukset.....	19
3.2 Teknologiset vaatimukset ja integraatio.....	20
3.3 Suunnittelu ja arkkitehtuuri.....	21
3.4 Tietokannan rakenne (MS SQL) .....	23
3.5 Käyttöliittymän suunnittelu.....	25
3.6 Ohjelmointi ja tietokannat .....	29
3.7 Tiedonkäsittely ja hinnanlaskentalogiikka.....	33
3.8 Hierarkkinen laskenta SQL-näkymällä .....	34
3.9 Ajax -käyttöliittymän responsiivisuus .....	35
4 TULOKSET JA ARVIOINTI .....	37
5 YHTEENVETO .....	38
5.1 Toteutus ja tutkimuskysymykset .....	38
5.2 Itsearviointi ja johtopäätökset .....	38
LÄHTEET.....	40

## KUVAT

Kuva 1. Tarjousten listausnäkyminen.....	25
Kuva 2. Uuden tarjouksen lisäysnäkyminen.....	26
Kuva 3. Yksittäisen tarjouksen näkyminen.....	27
Kuva 4. Materiaaliluettelon import-näkyminen.....	27
Kuva 5. Säättöarvojen näkyminen.....	28
Kuva 6. Erikoistuotteiden lisäysnäkyminen.....	28
Kuva 7. Perustuntimäärän hakulogiikkaa päämallista.....	30
Kuva 8. Pääohjainluokan metodeja.....	31
Kuva 9. admin_update_settings-ohjaintoiminnon hakutoimintoja.....	32
Kuva 10. Tarjoussivun HTML-rakennetta.....	33
Kuva 11. Tuontitoiminnon tiedostotyyppin ja kohde-ID:n tarkistus.....	34
Kuva 12. SQL-näkymän esimerkkitulokset hierarkkisesta laskennasta.....	35

## KUVIOT

Kuvio 1. MVC-arkkitehtuurin perusajatus.....	14
Kuvio 2. Tarjousprosessin vuokaavio.....	22
Kuvio 3. Tarjouslaskentatyökalun tietokantarakenne (ER-kaavio).....	24

## LYHENTEET

Ajax	engl. Asynchronous JavaScript and XML, tekniikka web-sovellusten vuorovaikutuksen parantamiseen
BOM	engl. Bill of Materials, materiaaliluettelo, tuoteluettelo
CRUD	engl. Create, Read, Update, Delete, tietokannan perustoiminnot
CSRF	engl. Cross-Site Request Forgery, verkkohyökkäystekniikka
CSS	engl. Cascading Style Sheets, verkkosivujen tyylimäärittely
FK	engl. Foreign Key, vierasavain

HMTL	engl. Hypertext Markup Language, verkkosivujen merkintäkieli
JSON	engl. JavaScript Object Notation, tiedonsiirtoformaatti
MVC	engl. Model-View-Controller, ohjelmistoarkkitehtuuri
ORM	engl. Object-Relational Mapping, olioiden muuntaminen relaatiotietokantaan sopiviksi
PHP	engl. Hypertext Preprocessor, ohjelmointikieli
PK	engl. Primary Key, perusavain
SQL	engl. Structured Query Language, relaatiotietokantojen kyselykieli
SQL-injektio	Tietoturva-aukko, jossa syötetään haitallista SQL-koodia
XLSX	Microsoft Excel -tiedostomuoto (Office open XML)

# 1 JOHDANTO

Työn toimeksiantaja on Escarmat Oy. Teollisuudessa keskeisimpiä tekijöitä ovat nykyään kilpailukyky ja asiakaslähtöisyys. Toimeksiantajan nykyinen tarjouslaskentajärjestelmä ei ole yhteensopiva tietyn asiakkaan materiaaliluetteloiden kanssa. Kyseisen asiakkaan materiaaliluetteloiden perusteella yrityksessä suoritetaan manuaalinen laskentaprosessi, joka aiheuttaa virhearvoja ja vaikeuttaa tarjousten analysointia. Tämänhetkinen laskentaprosessi on vaikea suorittaa ilman erikoisosaamista, minkä takia tarvittiin käyttäjäystävällinen ratkaisu.

Opinnäytetyön aiheena on ohjelmistokehitys. Kehityksen kohteena on asiakasräätälöity visuaalinen lisätyökalu olemassa olevan tarjouslaskentajärjestelmän rinnalle. Ratkaisu ei ole korvaamassa nykyistä järjestelmää kokonaan, vaan se toimii nykyisen tarjouslaskurin rinnalla toimeksiantajan sisäisessä verkossa. Projekti tehdään, koska yritys haluaa tehostaa toimintaansa, säästää aikaa ja resursseja sekä parantaa kilpailukykyään markkinoilla.

## 1.1 ESCARMAT

Escarmat Oy on vuonna 1994 perustettu Vaasan seudulla toimiva teollisuusyritys, joka on vakiinnuttanut asemansa sähkö-, automaatio ja ohjauskeskusten suunnittelussa ja valmistuksessa. Escarmat keskittyy myös energiatekniikan alihankintaan. Yrityksessä työskentelee yli 100 automaatioalan ammattilaista. Yrityksen liikevaihto on n. 18 miljoonaa euroa. Yrityksen toimintaan kuuluu korkealaatuiset komponentit, innovatiiviset teknologiat ja asiakaslähtöinen palvelu. Escarmatin tavoitteena on tarjota ratkaisuja, jotka parantavat asiakkaiden tuotannon tehokkuutta ja turvallisuutta. (Escarmat Oy, 2024.)

## 1.2 Tausta ja aihevalinta

Toimeksiantajan nykyinen asiakaskohtainen tarjouslaskenta suoritetaan manuaalisesti eli työntekijä analysoi materiaaliluetteloja ja siten toteuttaa valmiiseen Excel pohjaan tarjouslaskennan. Nykyinen taulukkolaskentaprosessi ei tue kaikkien käyttäjien sujuvaa työskentelyä, sillä manuaalinen laskenta on monimutkainen ja vaatii hyvää teknistä ymmärrystä. Tarjousten laatiminen tapahtuu valmiissa taulukkolaskentapohjissa, mikä tekee prosessista virheherkän ja kuormittavan. Pitkät ja yksityiskohtaiset materiaaliluettelot ovat vaikeaselkoisia etenkin sellaisille työntekijöille, joilla ei ole kokemusta laskennasta. Tiedon käsittelyssä saattaa tapahtua inhimillisiä virheitä ja koko laskentaprosessi voi tuntua epämiellyttävältä ja turhautavalta, mikä heikentää työn sujuvuutta.

Aihe on minulle tärkeä aidon työkokemuksen takia. Yritykselle uusi lisätyökalu parantaa työntekijöiden työskentelymukavuutta ja tarjoaa modernimman käyttökokemuksen. Uusi lisätyökalu tarjoaa selkeämmän ja käyttäjäystävällisemmän tavan suorittaa tarjouslaskentaprosessin. Työkalu tukee työntekijöitä eri osaamistasoilla ja tekee koko prosessista miellyttävämmän ja helposti ohjattavan.

### 1.3 Työn tavoitteet

Opinnäytetyön tavoitteena on kehittää visuaalinen asiakasräätälöity tarjouslaskentatyökalu olemassa olevan tarjouslaskentatyökalun rinnalle. Uusi työkalu on suunniteltu sellaisten asiakkaiden tarpeisiin, joiden materiaaliluettelot eivät ole yhteensopivia nykyjärjestelmän kanssa. Tavoitteena on sujuvoittaa tarjousprosessia, vähentää manuaalista työtä ja parantaa laskennan tarkkuutta.

Konkreettiset tavoitteet ovat:

- Toteuttaa selainpohjainen tarjouslaskentatyökalu, joka hyödyntää CakePHP 2 -kehystä ja toimii osana toimeksiantajan ympäristössä.
- Auttaa henkilöstöä syöttämään ja käsittelemään tarjouspyyntöjä ilman syvällistä teknistä osaamista.
- Sujuvoittaa tarjousten käsittelyä automatisoimalla manuaalisia vaiheita.
- Vähentää laskentavirheitä tarjoamalla laskentalogiikan, joka perustuu tietorakenteisiin.
- Tuottaa tarjous asiakkaalle asiakaslähtöisessä tiedostomuodossa, joka vastaa asiakkaan vaatimuksiin.

Tavoitteena on, että yritys voisi lisätyökalun käyttöönoton myötä siirtyä pois manuaalisesta laskentaprosessista kyseisen asiakkaan kohdalla. Ohjelmisto ei välttämättä täysin poista virheiden mahdollisuutta, mutta sen käyttö parantaa silti tarjouslaskennan toisteisuutta ja minimoi virheet. Selkeä rakenne ja helppo käyttöliittymä tukevat tehokasta työkentelytapaa ja helpottaa laskemista. Ohjelmisto on suunniteltu kattamaan muitakin asiakkuuksia mahdollisesti tulevaisuudessa ja uusia tarjouslaskentaan liittyviä määreitä on helppo lisätä.

## **1.4 Työn rakenne**

Työ koostuu viidestä pääluvusta. Johdannossa esitetään yritys, aiheen tausta ja tavoitteet sekä perustellaan aiheen valinta. Toinen luku käsittelee työn teoreettista viitekehystä ja käytettyjä teknologioita. Luvussa syvennyttään tarjouslaskennan merkitykseen ja prosessiin, tarkastellaan aiempia ratkaisuja ja esitellään CakePHP 2 -kehys sekä MVC-arkkitehtuuri. Kolmannessa luvussa käsitellään projektin toteutusvaiheita, kuten vaatimusten määrittelyä, järjestelmän suunnittelua ja ohjelmointia, tietokantarakenteita sekä käyttöliittymän suunnittelua. Lisäksi kolmannessa luvussa kuvataan tarkemmin, miten tietojen käsittely ja laskenta on toteutettu. Luvussa esitellään myös hierarkkinen laskentalogiikka, jonka avulla materiaaliluetteloiden rakenteet tulkitaan ja kokonaismäärät lasketaan.

Neljännessä luvussa esitetään opinnäytetyön tulokset sekä työkalun käyttöönotto ja testaus. Luvussa arvioidaan saavutettuja hyötyjä ja käsitellään mahdollisia jatkokehitysmahdollisuuksia. Luku antaa kokonaiskuvan työn tavoitteiden saavuttamisesta ja niiden vaikutuksista. Viidennessä luvussa esitetään opinnäytetyön yhteenveto. Kehitystyön aikana syntyneisiin tutkimuskysymyksiin vastataan ja arvioidaan työn onnistumista. Viidennessä luvussa käsitellään myös oppimiskokemuksia ja luku toimii päätöksenä koko opinnäytetyölle ja reflektoi työn merkitystä työnantajalle.

## **1.5 Tekoälyn käyttö tässä opinnäytetyössä**

Tässä työssä olen käyttänyt ChatGPT:tä ja Calude 3.7 sonnet -tekoälymalleja ideointiin ja tiedonhakuun. Tekoäly on auttanut muokkaamaan tekstiä helpommin ymmärrettäväksi. Työn tekniset ratkaisut ja sisältö ovat omaa työtäni.

## 2 TEORIA JA TEKNIIKAT

### 2.1 Tarjouslaskenta ja sen merkitys

Tarjouslaskenta on olennainen osa liiketoimintaa teollisuus- ja projektiluonteisessa valmistuksessa, jossa jokainen asiakasprojekti saattaa poiketa toisistaan. Tarkoituksena tarjouslaskennassa on määrittää toimitettavan tuotteen tai palvelun kustannukset ja asettaa sille kannattava kilpailukykyinen myyntihinta (Skycode, 2023). Tarkkuus ja nopeus tarjouslaskennan prosessissa voi ratkaista kaupan syntymisen ja varmistaa yrityksen taloudellisen menestyksen.

Tarjouslaskenta koostuu yleensä seuraavista osa-alueista:

- Materiaalikustannukset, jotka perustuvat tuotteiden osaluetteloihin ja toimittajahintoihin.
- Työkustannukset, jotka muodostuvat työn suorittamisesta ja tunti hinnoista.
- Yleiskustannukset ja katteet, joilla katetaan yrityksen toiminnan kuluja ja tavoitellaan voittoa.
- Erytyiskustannukset, kuten lisätyöt tai asiakkaan toivomukset.

Perinteinen tapa toteuttaa tarjouslaskenta taulukkolaskentaohjelmalla on joustava, mutta samalla erittäin altis virheille, se voi olla hyvin aikaa vievää, etenkin silloin, kun käsiteltävien materiaaliluettelojen (BOM) lista voi olla pituudeltaan monta sataa riviä. Tämän työn tapauksessa sarakkeita materiaaliluetteloissa on 16 luettavaa, esimerkiksi Level, Parent, Material, Qty ja muita tietoja.

Monet yritykset pyrkivät nykyään tehostamaan tarjouslaskentaa ohjelmistoratkaisujen avulla. Tarjouslaskennan laatu vaikuttaa suoraan asiakaskokemukseen. Nopeasti toimitettu, selkeä ja kilpailukykyinen tarjous antaa yrityksestä ammattimaisen kuvan. Tarjouslaskentaprosessi on

strategisesti tärkeä osa yrityksen liiketoimintaa erityisesti silloin kun tarjouksia tehdään säännöllisesti ja kilpailu on kovaa. Tehokas tarjouslaskenta edellyttää usein integraatiota yrityksen muihin järjestelmiin, kuten toiminnanohjausjärjestelmiin (ERP), jossa ylläpidetään ajantasaisia hintatietoja ja asiakaskohtaisia tietoja. (Iskanius ja muut 2009.)

## **2.2 Aiemmat ratkaisut ja olemassa olevat työkalut**

Tarjouslaskennan tueksi on olemassa monia erilaisia työkaluja, jotka vaihtelevat taulukkolaskentapohjista toiminnanohjausjärjestelmiin ja erityisiin tarjouslaskentaohjelmistoihin. Excel on yleisesti käytetty työkalu tarjouslaskentaan pienissä ja keskisuurissa yrityksissä helppokäyttöisyyden ja joustavuuden vuoksi. Monimutkaisemmissa projekteissa Excel-pohjaiset laskentapohjat ovat alttiita virheille ja vaikeasti ylläpidettäviä.

Toimeksiantajalla on jo käytössä heidän tarpeisiinsa räätälöity SkySeller Tarjouslaskenta, jonka avulla yritys voi laatia tarjouksia helposti, tarkasti ja nopeasti. Ohjelmisto hakee tarvittavat materiaalitiedot ja hinnat suoraan toiminnanohjausjärjestelmästä (ERP), joten komponenttien poimiminen tarjouksiin on tehokasta ja virheetöntä. Tarjousten hallinta helpottuu merkittävästi, kun vanhoja tarjouksia voidaan hyödyntää pohjana uusille, ja käytössä on aina ajantasaiset hinnat sekä asiakaskohtaiset alennusprosentit. Lisäksi ohjelma tallentaa kaikki tarjousversiot automaattisesti, mahdollistaen versioiden palauttamisen myöhemmin (Skycode 2021).

Sarjatuotannossa SkySellerin tarjoama sarjakokojen simulointi on tärkeää, koska se auttaa säilyttämään tarkan näkymän katteisiin myös suuremmissa volyymeissä. Ohjelmisto on integroitu suoraan toimeksiantajan sisäiseen järjestelmään ja sitä kehitetään jatkuvasti käyttäjien toiveiden mukaisesti.

SkySeller kattaa suurimman osan toimeksiantajan tarjouslaskennasta, mutta se ei ole yhteensopiva tämän tietyn asiakkaan lähettämien materiaaliluetteloiden kanssa. Kyseisen asiakkaan käyttämä tiedostoformaatti poikkeaa toimeksiantajan vakiokäytännöistä, mikä pakottaa käsittelemään nämä tarjoukset käsin. Tästä syntyi toimeksiantajan tarve erilliseen visuaaliseen lisätyökaluun, joka lähestulkoon kokonaan automatisoi laskentaprosessin kyseisissä tapauksissa ja toimii nykyjärjestelmän rinnalla. Tässä opinnäytetyössä kehitettävä työkalu vastaa juuri tähän tarpeeseen. Se hyödyntää CakePHP 2 -ohjelmistokehystä ja tarjoaa yksinkertaisemman ja asiakaskohtaisesti räätälöidyn käyttöliittymän tarjouslaskentaa varten.

### **2.3 CakePHP 2 -sovelluskehys**

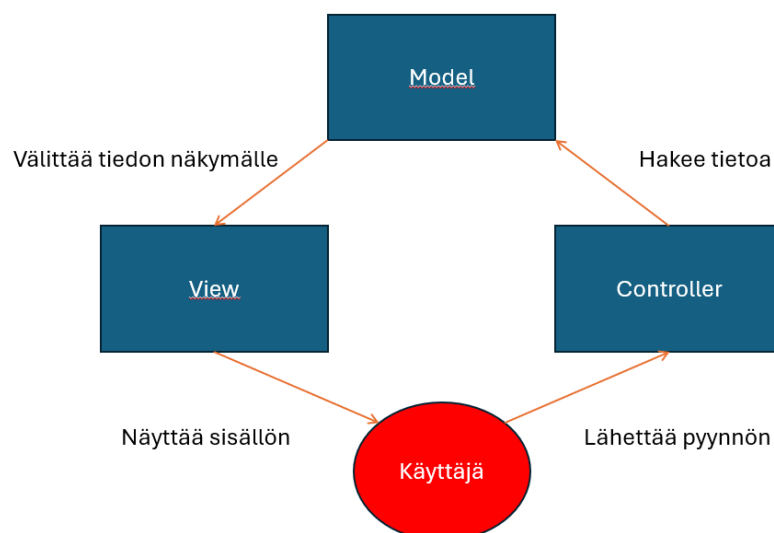
CakePHP 2 on avoimen lähdekoodin PHP-pohjainen sovelluskehys, joka noudattaa Model-View-Controller arkkitehtuuria. CakePHP 2 on suunniteltu nopeuttamaan ja helpottamaan web-sovellusten kehitystä. Kehys perustuu Model-View-Controller arkkitehtuuriin ja tarjoaa valmiita komponentteja, luokkia sekä rakenteita, joiden avulla kehittäjä voi keskittyä sovelluksen logiikkaan ilman, että tarvitsee rakentaa kaikkea alusta asti. (CakePHP, 2024a.)

CakePHP 2 -kehys tarjoaa useita hyödyllisiä ominaisuuksia, kuten tietokantataulujen käsittelyn automaattisesti (CRUD). Lomake- ja syöttökenttiä voidaan validoida, joka mahdollistaa esimerkiksi autentikoinnin sovelluksessa. Sovelluskehysten yleinen lähestymistapa perustuu konventioon, jossa kehittäjän tarvitsee kirjoittaa vähemmän koodia, kunhan noudattaa ohjelmointistandardeja ja nimeämissääntöjä. (CakePHP, 2024a.)

Toimeksiantaja käyttää CakePHP 2 -kehystä omassa sisäisessä järjestelmässään, jonka kautta omat sovellukset toimivat lähiverkossa. Opinäytetyössä kehitetty lisätyökalu rakennettiin samaan ympäristöön, jotta se olisi yhteensopiva nykyisen infrastruktuurin kanssa ja käyttöönotto olisi mahdollisimman sujuvaa. Kehitysympäristönä käytettiin XAMPP-palvelinalustaa, Notepad++-koodieditoria, CakePHP 2 -hakemistorakennetta ja konfiguraatitiedostoja. Notepad++ on ilmainen lähdekoodieditori (Notepad++, n.d.). Sovellusta kehitettiin ja testattiin erillisessä \dev-hakemistossa ennen sen siirtoa tuotannon puolelle.

## 2.4 MVC-arkkitehtuuri

CakePHP 2 perustuu MVC-arkkitehtuuriin (Model-View-Controller), joka on yleisesti käytetty suunnittelumalli web-sovellusten kehityksessä. Malli erottaa sovelluksen kolmeen erilliseen osaan: malliin, näkymään ja ohjaimen. Tämä erottelu edesauttaa koodin ylläpitoa, testausta ja uudelleenkäyttöä. (ks. Kuvio 1).



Kuvio 1. MVC-arkkitehtuurin perusajatus.

### MVC-arkkitehtuurin peruskomponentit:

1. Malli (Model): Mallien tehtävä on hallinnoida sovelluksien tietokantayhteyksiä, tiedon validointia ja käsittelyä. CakePHP 2 -mallit ovat luokkia, jotka vastaavat tietokantatauluja ja sisältävät toiminnallisuuden tiedon hallintaan.
2. Näkymä (View): Käyttöliittymän esittely ja ulkoasu. Näkymät ovat tiedostoja, jotka sisältävät HTML-koodia sekä PHP-koodia (PHP.net, 2024), joka näyttää tietoa loppukäyttäjälle. Näkymät ovat .ctp-päätteisiä tiedostoja, jotka sijaitsevat View-hakemistossa.
3. Ohjain (Controller): Välittäjä mallin ja näkymän välillä. Ohjaimet käsittelevät käyttäjän pyyntöjä, jotka sisältävät toimintoja eri URL-reittien käsittelyyn (Mozilla, n.d).

MVC-arkkitehtuurin käyttäminen CakePHP 2 -sovelluskehyksessä tuomaa etuja. Se tarjoaa koodille selkeän rakenteen, mikä helpottaa sovelluksen ymmärtämistä ja ylläpitoa (CakePHP, 2024b). Kehittäjät pääsevät työskentelemään samanaikaisesti mallin, ohjaimen ja näkymän parissa. Komponentteja voidaan uusiokäyttää eri osissa sovellusta. Sovellusta on erittäin helppo laajentaa komponenttien ollessa erillisiä. Testattavuus helpottuu, koska komponentit ovat erillisiä.

Tämän työn sovelluksessa on mallit tarjouksille, tarjousriveille, tarjousasetuksille ja muille tarvittaville tietokantarakenteille. Mallit hakevat tietoa, kuten tuotteiden hintoja. Näkymät esittävät käyttäjälle tarjouslaskennan vaiheita, kuten tarjousten listaus, säätöarvot, uuden tarjouksen lisääminen, avattu tarjous, tuonti, erikoistuotteet ja erikoistuoteryhmät. Ohjaimet käsittelevät mitä käyttäjä syöttää ohjelmassa, joka tekee ohjelmasta dynaamisen. Käyttäjä voi esimerkiksi tuoda uuden materiaaliuutteen Excel pohjasta. Ohjaimet ohjaavat tiedonkulkua mallien ja näkymien välillä.

## 2.5 Tietokantojen suunnittelu ja normalisointi

Hyvin suunniteltu tietokanta takaa sovelluksen tehokkuuden, luotettavuuden ja skaalautuvuuden. Normalisointi on keskeinen periaate reaali-tietokantojen suunnittelussa, jossa pyritään välttämään tietojen toisteisuutta ja varmistamaan tietojen eheys. Prosessissa tietokannan rakenne organisoidaan niin, että minimoidaan riippuvuusongelmat. Normalisointi tapahtuu useassa normalisointimuodossa, joista yleisimmät ovat ensimmäinen, toinen ja kolmas normaalimuoto. Normalisointimuotoja voidaan kuvailla lyhenteillä (NF), (1NF), (2NF), (3NF). (Microsoft Corporation, 2025.)

Tarjouslaskentalisätyökalun tietokantasuunnittelussa sovellettiin normalisointiperiaatteita seuraavasti. Ensimmäisessä normaalimuodossa varmistettiin, että kaikissa tauluissa on perusavain ja että kaikki sarakkeet sisältävät vain jakamattomia arvoja. Esimerkiksi materiaalitiedot normalisoitiin siten, että jokainen rivi kuvaa yhtä materiaalia, ja materiaalin ominaisuudet, kuten materiaalikoodi, kuvaus ja hinta, tallennetaan omiin sarakkeisiinsa. Toisessa normaalimuodossa varmistettiin, että ei-avainattribuutit ovat täysin riippuvaisia koko perusavaimesta. Tämä tarkoittaa sitä, että jos materiaalirivillä on yhdistelmäavain (material\_id, taso, järjestysnumero), niin kaikki rivin tiedot riippuvat tästä koko avaimesta, eivätkä vain osasta siitä. Kolmannessa normaalimuodossa varmistettiin, ettei kaikki ei-avainattribuutit riippuisi suoraan perusavaimesta, eivätkä toisen ei-avainattribuutin kautta (Elmasri & Navathe, 2016). Esimerkiksi materiaalirivin hinta ei riipu materiaalin kuvauksesta vaan suoraan materiaalikoodista.

Vaikka normalisointi saattaa vähentää tietojen toisteisuutta, joskus tiettyjä osia tietokannasta voi de-normalisoida suorituskyvyn parantamiseksi. Työssä tiettyjä usein tarvittavia laskettuja arvoja tallennetaan tauluihin, vaikka ne olisi voitu teoriassa laskea reaaliajassa olemassa olevista tiedoista. Tietokannan välisiä suhteita hallitaan CakePHP:n

ORM-mallin avulla, vaikka varsinaisia vierasavaimia ei ole määritelty tietokantatasolla. Esimerkiksi materiaaliriveillä on viiteavain materiaalitau-luun, jonka avulla yhteys säilyy materiaalien ja rivien välillä. Tämä me-netelmä mahdollistaa, että tarjouslaskurissa kaikki tiettyyn tarjoukseen liittyvät rivit voidaan hakea ja muutokset näkyvät kaikissa tarjoukseen liittyvissä riveissä vaikuttaen automaattisesti kustannuslaskentaan.

Tietokannan suorituskykyä optimoitiin indeksoinnilla. Tietyille hakueh-doille, kuten materiaalikoodeille ja moottorityypille, luotiin indeksit. In-deksointi nopeuttaa hakua merkittävästi, mikä on tärkeää suuremmissa tietokannoissa (Huhtala, 2014). Tulevaisuuden laajennustarpeet huomi-oiitiin tietokantasuunnittelussa. Taulujen rakenteet suunniteltiin niin, että ne tukevat uusien ominaisuuksien lisäämistä. Tietokantarakenne on joustava, jotta lisättäessä uusia ominaisuuksia ei tarvitse muuttaa koko tietokantarakennetta.

## **2.6 Tietoturva**

Tietoturva on keskeinen aihe web-sovelluksissa. Erityisesti liiketoimin-takriittisiä tietoja, kuten tarjouksiin liittyviä materiaalitietoja, hintatie-toja tai asiakastietoja täytyy käsitellä varoen. Tämä luku käsittelee tar-jouslaskentatyökalun tietoturvaa ja käyttöympäristöä.

CakePHP2 -sovelluskehys tarjoaa valmiita työkaluja yleisimpien tieto-turvauhkien torjumiseen, kuten CSRF-suojaa (Cross-Site Request For-gery) ja SQL-injektiosuojaa (OWASP, n.d.). Näitä ominaisuuksia hyö-dynnettiin lisätyökalun kehityksessä. Lisätyökalun käyttöympäristö on suunniteltu Escarmatin tietoturva vaatimusten mukaisesti. Lisätyökalu käyttää monitasoista käyttäjäautentikointia ja käyttäjävaltuutusta. So-vellus toimii yrityksen sisäverkossa, joka itsessään rajoittaa pääsyä so-vellukseen. (CakePHP, 2024a.)

Käyttäjien tunnistautuminen lisätyökalussa tapahtuu CakePHP Auth-komponentin avulla, joka hoitaa käyttäjän tunnistamisen kirjautumislo-makkeen avulla. Aktiivisuustarkistuksella varmistetaan, että vain aktii-viset käyttäjät voivat kirjautua sisään. Käyttäjärooleilla määritetään mitä toimintoja käyttäjä voi tehdä. Kaikilla käyttäjillä ei ole valtuuksia käydä muokkaamassa hintatietoja tai luoda uusia tarjouksia, ainoastaan valituilla. (CakePHP, 2024a.)

## 3 KEHITYSPROSESSI JA TEKNINEN TOTEUTUS

### 3.1 Vaatimusmäärittely ja käyttäjävaatimukset

Ohjelmistokehityksen ensimmäisessä vaiheessa eli vaatimusmäärittelyssä määritellään, miten sovelluksen kuuluisi toimia. Vaatimusmäärittely on erittäin tärkeä vaihe ohjelmistoprojektissa, sillä ilman selkeitä tavoitteita projektin aloittaminen on haasteellista. Käyttäjävaatimuksilla määritetään, kuinka loppukäyttäjien tulisi pystyä käyttämään sovelluksen eri toimintoja. Työn alkaessa vaatimusmäärittelyt olivat aika suppeat, mutta työn edetessä käyttäjävaatimukset lisääntyivät ja tarkentuivat. Työn aikana määriteltiin seuraavanlaisia käyttäjävaatimuksia pääpiirteittäin:

1. Perustietojen lisääminen: Käyttäjän tulee voida lisätä tarjouksesta tehtävän tuotteen perustietoja, joiden perusteella voidaan suorittaa myöhemmissä vaiheissa tarjouslaskenta. Perustiedot ovat kytköksissä säätöarvoihin, joista saadaan halutut arvot laskennan suorittamiseksi.
2. Tarjousten listaus: Perustietojen lisäämisen jälkeen tarjoukset siirtyvät etusivun listaukseen, josta voi nähdä sarakkeita, kuten työaika, tarjoushinta, sovittu hinta, erotus, ero % ja versio.
3. Säätöarvojen lisääminen ja muokkaus: Säätöarvoissa käyttäjän tulee voida hallita tarjousten laskentaperusteisiin liittyviä arvoja. Säätöarvoissa määritellään tarjousten tuotteiden määrittäviä arvoja ja yleisiä arvoja, kuten materiaalikustannusta, voittoa ja laskennallista kerrointa. Säätöarvojen tulee olla muokattavissa ja niitä pitää pystyä lisäämään tarpeen mukaan.
4. Tarjousten hakutoiminto: Käyttäjän tulee voida hakea tarjouksia sovelluksen etusivun listauksesta.
5. Tarjousten avaaminen ja poisto: Käyttäjän pitää voida avata tarjous ja poistaa tarjous.

6. Materiaaliluetteloiden tuonti: Yksittäisen tarjouksen avaamisen jälkeen käyttäjän tulee voida tuoda taulukkolaskentamateriaaliluettelo yksittäisen tarjouksen näkymään.
7. Hintojen päivitys ja laskenta: Taulukkolaskentamateriaaliluettelon tuonnin jälkeen käyttäjä voi yksittäisen tarjouksen näkymässä painaa painiketta, josta jokaiselle tuotteelle materiaaliluettelosta päivittyy hinta, jos sellainen tuotekoodi on olemassa erillisessä tietokannassa. Samaa painiketta painamalla käyttäjän pitää pystyä suorittamaan tarjouslaskenta.
8. Tarjousten revisiointi: Käyttäjän pitää pystyä revisioimaan tarjous yksittäisen tarjouksen näkymässä.
9. Tarjouksen vienti: Käyttäjän tulee voida viedä laskettu tarjous asiakkaan vaatimaan lähdeformaattiin eli .xlsx. -muotoon.

### **3.2 Teknologiset vaatimukset ja integraatio**

Tässä luvussa määritellään, mitä teknologioita ja alustoja sovelluksen kehityksessä ja käytössä tarvitaan. Sovelluskehityksenä käytetään CakePHP 2:a, koska koko järjestelmä on alun perin sen varassa. HTML, CSS, JQuery ja JavaScript toimivat käyttöliittymänä. Excel-tiedostojen lukemiseen hyödynnetään PhpSpreadsheet-kirjastoa tuonti- ja vientiprosesseissa (PhpSpreadsheet, 2024). Sovelluksen tulee toimia yleisimmillä selaimilla työpöytäkäytössä. Kehitysympäristössä käytettiin XAMPP-pakettia, joka sisältää Apache-web palvelimen, jota käytetään laajalti PHP-sovellusten ajamiseen eri käyttöjärjestelmissä (Apache Software Foundation, n.d.). Apache on asennettu suoraan Windows Serverille. Yleisesti ottaen teknisten ratkaisujen pitää olla geneerisiä ja joustavia tulevaisuutta varten. Sovelluksen tulee olla integroitunut Lemonsoft-toiminnanohjausjärjestelmään (ERP), josta haetaan ajantasaiset tuotehinnat ja muut relevantit tiedot (Lemonsoft Oyj, 2024). Sovelluksen tulee olla saatavilla ainoastaan yrityksen sisäisessä verkossa ja

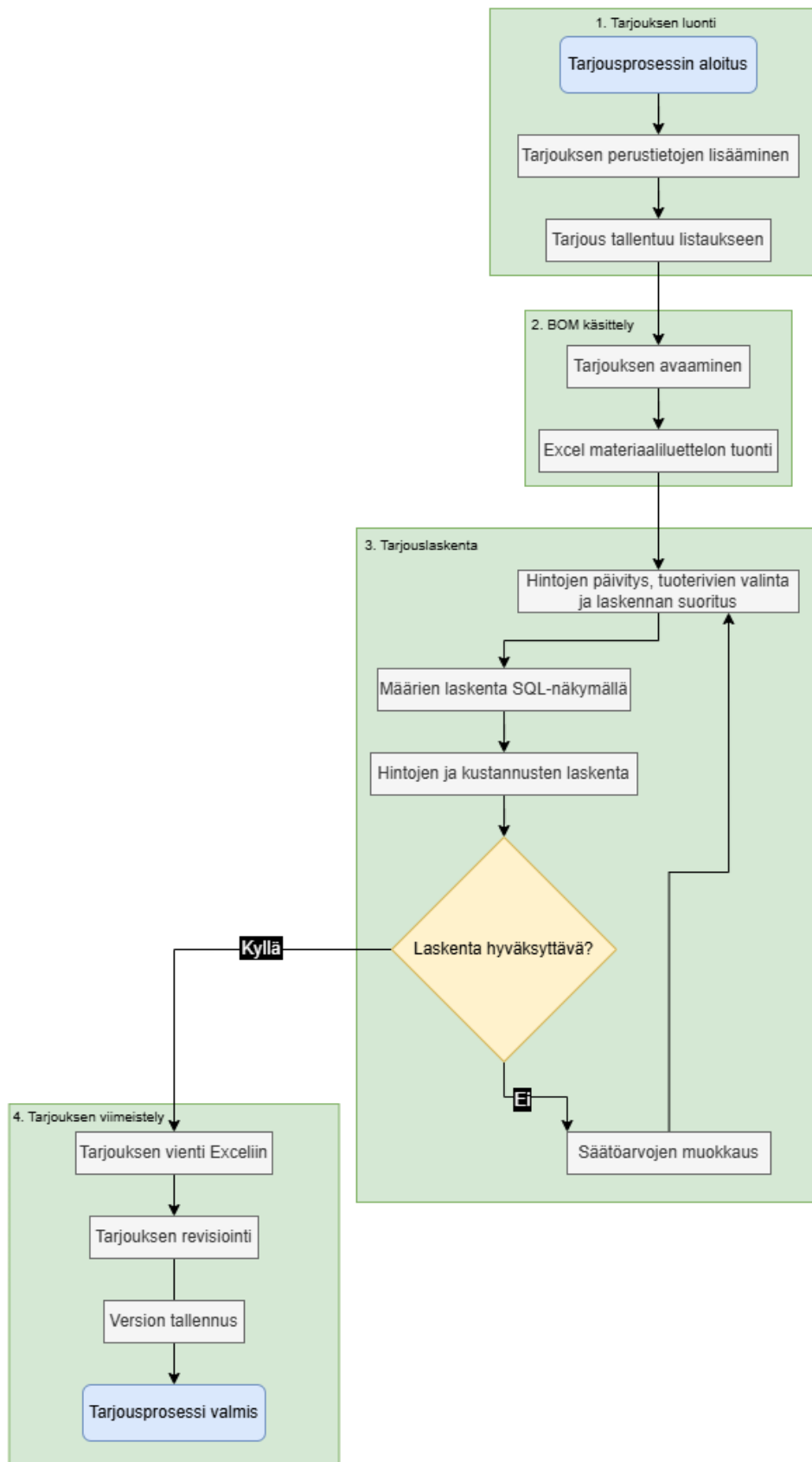
käyttäjätunnistuksena toimii olemassa oleva tunnistus. Vaatimusmäärittelyprosessi toteutettiin yhdessä Escarmatin henkilöstön kanssa. Koko prosessiin kuului aloituspalaveri ja viikkopalaverit.

### **3.3 Suunnittelu ja arkkitehtuuri**

Tarjouslaskentalisätyökalun suunnittelussa käytettiin CakePHP 2 MVC-arkkitehtuuria. Sovelluksessa on useita malleja (Models), jotka sisältävät liiketoimintalogiikkaa. Lisätyökalu sisältää päämallin, joka sisältää tietoja, kuten materiaalikoodoja, kuvaksia ja hintatietoja. Tarjouksen rivikohtainen malli kuvaa yksittäistä tuoteriviä, joka sisältää tietoja, kuten tason, vanhemman, tuotekoodin ja kokonaismäärän tietylle tuotteelle. Revisiointimalli hallinnoi tarjousten revisiohistoriaa ja sen avulla voidaan tarkastella vanhempia versioita. Tarjouksen materiaalien säätöarvoille on malli, joka sisältää laskentaan liittyviä arvoja, kuten työtunteja ja lisätarvikkeita. Katteen ja materiaalikulujen hallintaan on oma mallinsa. Laskentaan on vielä kytköksissä malli, joka hallinnoi tietyn tuoteryhmän määrittelyjä, jotka vaikuttavat tarjouslaskentaan (ks. Kuvio 2).

Näkymät (Views) on jaettu eri hakemistoihin selkeyden takia. Pääluokkahakemisto sisältää näkymiä tarjousten listaamisesta, muokkaamisesta ja uusien luomisesta. Rivikohtainen hakemisto sisältää näkymät tarjousten tuoterivien hallinnasta. Asetushakemisto sisältää näkymiä arvojen määrittelyyn ja hallintaan liittyen. Näkymät on toteutettu CakePHP. ctp-tiedostoina, jotka sisältävät HTML-koodia PHP-koodin lisäksi.

Sovelluksen päätoiminnallisuutta ohjaavat ohjaimet (Controller). Pääohjain hallinnoi tarjouksia kokonaisuutena. Pääohjaimen toimintoja ovat tarjousten listaaminen, luominen, muokkaaminen ja laskenta. Tuoterivien ohjain käsittelee yksittäisten tuotteiden rivien toiminnot. Asetusten ohjain hallinnoi tarjoukseen liittyvien materiaalien asetuksia, mikä vaikuttaa hinnoitteluun ja työkustannuksiin tarjouksissa. Erikoistuotteiden ohjain hallinnoi valikoitujen tuotteiden kappalemääriä, jotka vaikuttavat laskentaan.



Kuvio 2. Tarjousprosessin vuokaavio.

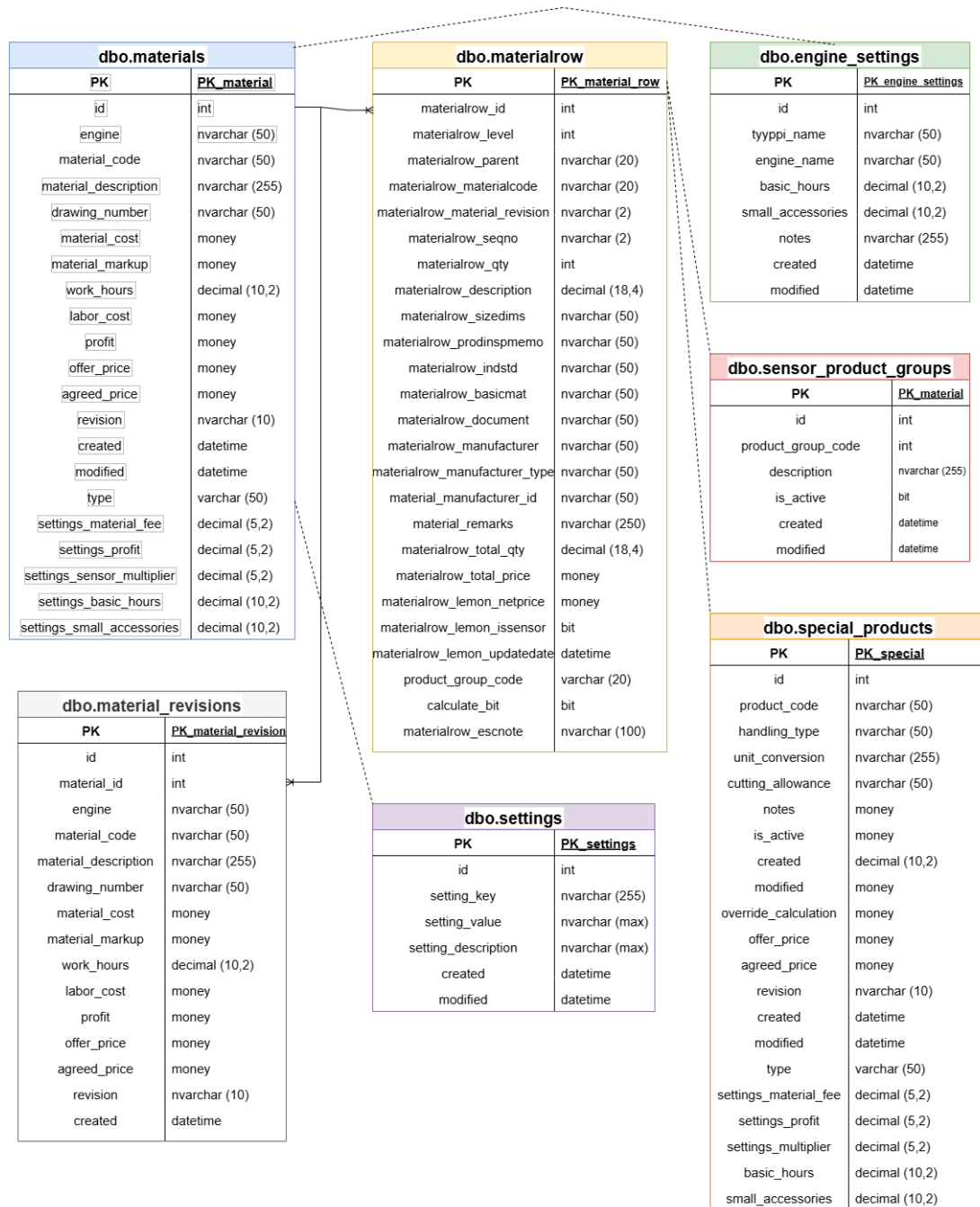
### 3.4 Tietokannan rakenne (MS SQL)

Sovelluksen tietokanta toteutettiin Microsoft SQL-serverillä (Microsoft Corporation, 2024). Tietokanta sisältää yhteensä seitsemän taulua. Taulut liittyvät toisiinsa loogisesti ja tukevat ohjelman eri toiminnallisuuksia. Tauluja ja tietueita lisätään ohjelmaan tarvittaessa. Päätaulu materials sisältää yleiset tarjouskohtaiset tiedot, kuten materiaalikoodin, kuvauksen, tyyppin, kustannustiedot ja laskennan tulokset. Jokaisella materiaalirivillä voi olla useita rivitason tietoja, jotka tallennetaan materialrow-tilaan. Tämä taulu sisältää järjestysnumeron, materiaalikoodin, määrän sekä lisätietoja tuotteesta. Rivit liittyvät materiaaliin material\_id:n avulla.

Versiohistoriaa varten on material-revision-tila, joka mahdollistaa tarjousten aiempien versioiden tarkastelun ilman muokkausmahdollisuutta. Laskennan parametreihin liittyviä asetuksia tallennetaan kahden tilaan. Engine\_settings-tilassa on moottorityyppikohtaisia asetuksia, kuten perustyötunnit ja lisätarvikkeet. Settings-tilaan tallennetaan lisätyökalun yleiset laskentaan liittyvät asetukset, kuten voitto-%.

Erikoistuotteiden käsittelyssä käytetään special\_products-tilaa, joka sisältää poikkeuksellisesti käsiteltäviä tuoterivejä. Special\_products-tilaan käyttäjä voi lisätä käyttöjärjestelmän kautta tuotteen ja asettaa sille tietyn määrän, jolloin lisätty määrä ja hinta muuttuu tuoduissa materiaaliluetteloissa SQL-proseduurin kautta. Sensoreihin liittyvä logiikka määritellään sensor\_product\_groups-tilassa.

Jokaisella taululla on ensisijainen avain (PK), joka varmistaa tietueiden yksilöllisyyden. Vierasavaimia (FK) ei ole määritelty tietokantatasolla vaan suhteet on toteutettu sovellustasolla CakePHP:n ORM-yhteyksien kautta. Tämä tuo joustavuutta tietokannan ja sovelluksen välisten toimintojen hallintaan ja vähentää tietokantarajoitteiden vaikutuksia kehitystyössä (ks. Kuvio 3).



Kuvio 3. Tarjouslaskentatyökalun tietokantarakenne (ER-kaavio).

### 3.5 Käyttöliittymän suunnittelu

Tarjouslaskentalisätyökalun pääkäyttäjät ovat myyntihenkilöstöä. Sovelluksen käyttäjiin olisi tarkoitus kuulua myös muuta yrityksen henkilöstöä, joka ei välttämättä ole teknisesti kovin pätevää. Tämän takia käyttöliittymässä keskitytään selkeyteen ja helppokäyttöisyyteen. Käyttöliittymä ohjaa käyttäjää tarjouksen toteutuksessa.

Käyttöliittymän päänäkymiin kuuluu neljä eri päänäkymää, joista ensimmäinen eli tarjousten listaus näkyy ensimmäisenä avatessa laskurin. Listausta on taulukkomuodossa, josta voi tarkastella olemassa olevia tarjouksia. Taulukon soluihin kuuluu tietoja tarjouksista, kuten perustietoja ja säätöarvojen perusteella laskettuja arvoja (ks. kuva 1).

Engine	Material	Material description	Drawing number	Työaika	Tarjoushinta	Sovittu hinta	Erotus	Ero %	Versio	Toiminnot
	DEMO001	Demo Material 1	DRW-001	50.00 h	2,044.42 €	0.00 €	-2,044.42 €	-100.00%	1.0	<a href="#">Näytä</a> <a href="#">Poista</a>
	DEMO002	Demo Material 2	DRW-002	5.89 h	244.12 €	0.00 €	-244.12 €	-100.00%	1.0	<a href="#">Näytä</a> <a href="#">Poista</a>
	DEMO003	Demo Material 3	DRW-003	.00 h	0.00 €	0.00 €	0.00 €	0.00%	1.0	<a href="#">Näytä</a> <a href="#">Poista</a>
	DEMO004	Demo Material 4	DRW-004	.00 h	0.00 €	0.00 €	0.00 €	0.00%	1.0	<a href="#">Näytä</a> <a href="#">Poista</a>
	DEMO005	Demo Material 5	DRW-005	.00 h	0.00 €	0.00 €	0.00 €	0.00%	1.0	<a href="#">Näytä</a> <a href="#">Poista</a>
	DEMO006	Demo Material 6	DRW-006	.00 h	0.00 €	0.00 €	0.00 €	0.00%	1.0	<a href="#">Näytä</a> <a href="#">Poista</a>
	DEMO007	Demo Material 7	DRW-007	.00 h	0.00 €	0.00 €	0.00 €	0.00%	1.0	<a href="#">Näytä</a> <a href="#">Poista</a>
	DEMO008	Demo Material 8	DRW-008	.00 h	0.00 €	0.00 €	0.00 €	0.00%	1.0	<a href="#">Näytä</a> <a href="#">Poista</a>
	DEMO009	Demo Material 9	DRW-009	.00 h	0.00 €	0.00 €	0.00 €	0.00%	1.0	<a href="#">Näytä</a> <a href="#">Poista</a>
	DEMO010	Demo Material 10	DRW-010	.00 h	0.00 €	0.00 €	0.00 €	0.00%	1.0	<a href="#">Näytä</a> <a href="#">Poista</a>

Kuva 1. Tarjousten listausnäkyminen.

Etusivulta pääsee joko lisäämään tarjouksen eli näkymä, josta voi syöttää tarjoukseen liittyviä perustietoja tai sitten säätöarvoihin, josta voi muokata järjestelmän tuoteasetuksia tai laskennallisia arvoja. Etusivulla on hakutoiminto (ks. kuva 2).

The screenshot shows the 'Escarmat' software interface. At the top, there is a navigation bar with the logo and menu items: 'Hallinta', 'Raportointi', and 'Työkalut'. Below this, the main heading is 'Tarjouslaskuri' (Quote Calculator), followed by the sub-heading 'Lisää uusi tarjous' (Add new quote). The form is titled 'Perustiedot' (Basic information) and contains the following fields:

- Type**: A dropdown menu with the placeholder text '-- Valitse --'.
- Engine**: A dropdown menu with the placeholder text '-- Valitse --'.
- Material**: A text input field.
- Material description**: A text input field.
- Drawing number**: A text input field.

At the bottom of the form, there are two buttons: 'Tallenna' (Save) and 'Peruuta' (Cancel).

Kuva 2. Uuden tarjouksen lisäysnäkömä.

Etusivulta pystyy avaamaan yksittäisen tarjouksen, jonka näkymässä päästään tarkastelemaan aiemmin syötettyjä perustietoja, kustannuslaskentaa, materiaaliluetteloja ja revisioita. Yksittäisen tarjouksen sivulta pääsee suorittamaan toimintoja, kuten tarjouksen tuominen, vienti, rakenteen vienti, erikoistuoeryhmien alilevelien nollaus, revisiointi, hintojen haku ja erikoistuoitteiden määrien korjaus (ks. kuvat 2 ja 3).

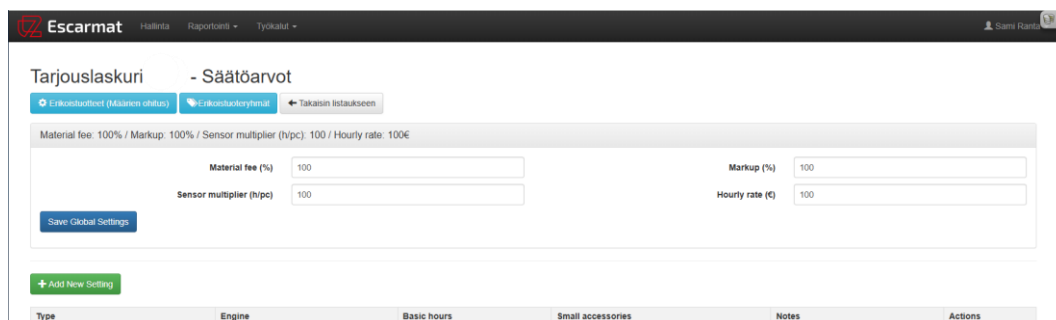
Level	Parent	Material	Rev	Seq	Qty	TotalQty	Description	Size/Dimensions	Net price	Total price	Calculate	Notes
0	Top Level	ITEM000001	-		1,00	1,00	MAIN ASSEMBLY		0,00 €	0,00 €	<input checked="" type="checkbox"/>	
1	ITEM000001-	ITEM000002	A	10	2,00	2,00	SUB ASSEMBLY		0,00 €	0,00 €	<input checked="" type="checkbox"/>	
1	ITEM000001-	ITEM000005	-	50	1,00	0,00	ELECTRICAL ASSEMBLY		0,00 €	0,00 €	<input type="checkbox"/>	
1	ITEM000001-	ITEM000010	-	90	3,00	0,00	SPECIAL PART		0,00 €	0,00 €	<input type="checkbox"/>	
2	ITEM000002/A	ITEM000003	B	20	3,00	0,00	COMPONENT	M8x25	0,00 €	0,00 €	<input type="checkbox"/>	
2	ITEM000002/A	ITEM000004	-	30	1,00	3,00	MECHANICAL PART		0,00 €	0,00 €	<input checked="" type="checkbox"/>	
2	ITEM000002/A	ITEM000005	C	40	4,00	12,00	SENSOR		0,00 €	0,00 €	<input checked="" type="checkbox"/>	
2	ITEM000006/-	ITEM000007	A	60	5,00	15,00	CABLE	2.5mm <sup>2</sup>	0,00 €	0,00 €	<input checked="" type="checkbox"/>	
2	ITEM000006/-	ITEM000008	-	70	2,00	6,00	CONNECTOR		0,00 €	0,00 €	<input checked="" type="checkbox"/>	
2	ITEM000006/-	ITEM000009	B	80	1,00	3,00	SENSOR		0,00 €	0,00 €	<input checked="" type="checkbox"/>	

Kuva 3. Yksittäisen tarjouksen näkymä.

Yksittäisen tarjouksen näkymästä pääsee Excel-tiedostojen tuontinäkymään, jossa on ohjeita tiedoston formaatista ja vedä-ja-pudota toiminnallisuus tiedoston lataamiseksi. Yksittäisen tarjouksen näkymästä on myös mahdollista viedä pelkkä rakenne eli tarjouksen materiaalikoodit, kuvaus, tuntimäärä ja tuotemäärä Excel-tiedostoksi (ks. kuva 4).

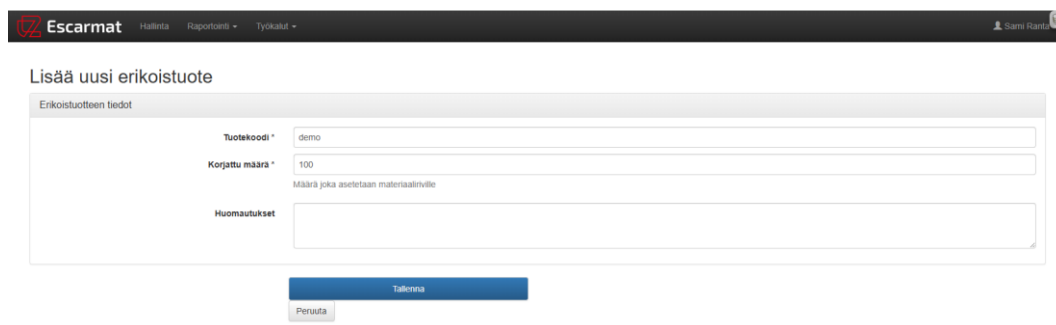
Kuva 4. Materiaaliluettelon import-näkymä.

Materiaalikate, voittomarginaali, sensorien kerroin ja tuntihinta voidaan muuttaa säätöarvot sivulta. Asetetut globaalit arvot vaikuttavat ainoastaan uusiin tarjouksiin eikä ne tule vaikuttamaan aikaisempiin tarjouksiin (ks. kuva 5).



Kuva 5. Säätöarvojen näkymä.

Säätöarvojen näkymästä pääsee käsittelemään erikoistuotteita, jossa käyttäjällä on mahdollisuus lisätä kokonaan uusi erikoistuote, jonka materiaalikoodin perusteella lisätyn tuoterivin määrät (Qty, TotalQty) korjautuvat halutulla määrällä. Erikoistuotteiden korjatut määrät vaikuttavat kustannuslaskentaan. Säätöarvot-sivulta voi myös hallita erikoistuoteryhmiä, joiden perusteella järjestelmä tunnistaa lisätyt tuoteryhmät ja lisättyjen tuoteryhmien alilevelit voidaan siten nollata SQL-proseduurin kautta (ks. kuva 6).



Kuva 6. Erikoistuotteiden lisäysnäky.

Käyttöliittymän suunnittelussa käytettiin Bootstrap-kehystä responsiivisen ulkoasun toteuttamiseksi. Bootstrap on yksi maailman suosituimmista avoimen lähdekoodin responsiivisten verkkosivujen ohjelmistokehyksistä (Bootstrap, n.d.). jQuery-kirjastoa käytettiin interaktiivisten

elementtien, kuten Ajax-päivitysten ja dynaamisten lomakkeiden toteuttamiseen. Käyttöliittymässä käyttäjän toiminnon onnistuminen tai epäonnistuminen palauttaa jonkun visuaalisen viestin.

### **3.6 Ohjelmointi ja tietokannat**

Lisätyökalun ohjelmointi aloitettiin mallien, näkymien ja ohjainten luomisella CakePHP 2 -kehiksen konventioiden mukaisesti. Ensimmäisenä vaiheena oli tietokantataulujen ja niihin vastaavien mallien luominen. Tämän opinnäytetyön koodissa ei esitetä asiakkaan tietoja, tarkkoja laskentakaavoja, liiketoimintasääntöjä, käyttäjätunnuksia, salasanoja, todellisia tuotteita, tietokantojen nimiä, palvelimien nimiä tai IP-osoitteita. Koodin ja datan esittämisessä hyödynnetään anonymisointia, yksinkertaistamista ja kommentteja tai kuvauksia (Devane, 2022). Koodi on muuten todellista CakePHP 2 -koodia.

Mallit sisältävät tarjousten käsittelyyn tarvittavat tietokantarakenteet sekä niihin liittyviä validointeja ja laskentatoimintoja. Päämallissa hallinnoidaan perustietoja, hinnoittelua ja kokonaiskustannuslaskentaa. Tukimalleissa käsitellään antureiden tunnistusta, moottorityyppikohtaisia oletusasetuksia ja tarjousten versiohistorian hallintaa (ks. kuva 7).

```
public function calculateBaseHours($material) {
    // tarkista onko perustunnit jo asetettu
    if (!empty($material['work_hours'])) {
        return floatval($material['work_hours']);
    }

    // tarkista jos saatavilla
    if (empty($material['type']) || empty($material['engine'])) {
        return 0;
    }

    $type = strtoupper($material['type']);
    $engine = strtoupper($material['engine']);

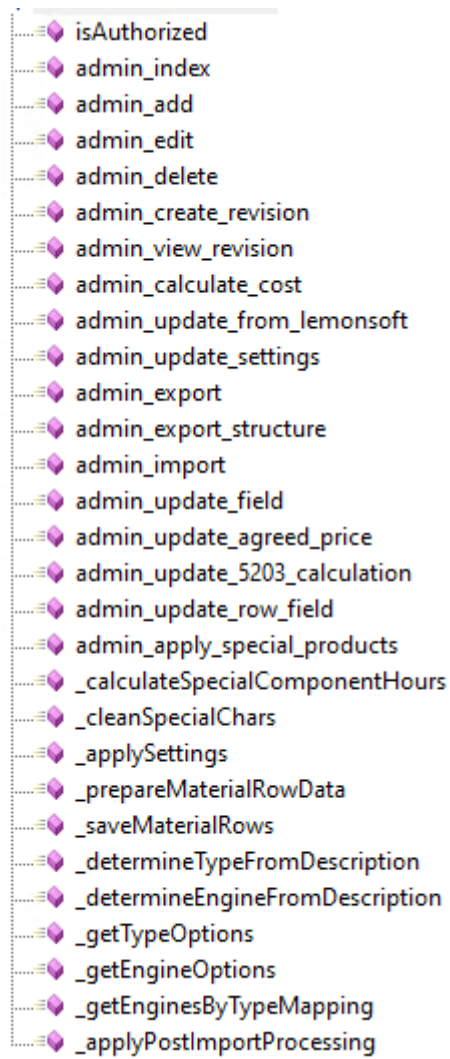
    // Haetaan asetukset tietokannasta
    if (!isset($this->EngineSetting)) {
        $this->EngineSetting = ClassRegistry::init('EngineSetting');
    }

    $settings = $this->EngineSetting->find('first', array(
        'conditions' => array(
            'tyyppi_name' => $type,
            'engine_name' => $engine
        )
    ));

    if (!empty($settings)) {
        // tietokannasta saadut tunnit
        $hours = $settings['EngineSetting']['basic_hours'];
    }
}
```

Kuva 7. Perustuntimäärän hakulogiikkaa päämallista.

Ohjaimet vastasivat eri toiminnoista, kuten materiaalien listauksesta ja uusien tarjousten luomisesta. Pääohjain sisältää metodeja kuten add, edit ja view (ks. kuvat 8 ja 9).



The image shows a list of 30 methods for the Admin class, each preceded by a small purple icon. The methods are:

- isAuthorized
- admin\_index
- admin\_add
- admin\_edit
- admin\_delete
- admin\_create\_revision
- admin\_view\_revision
- admin\_calculate\_cost
- admin\_update\_from\_lemonsoft
- admin\_update\_settings
- admin\_export
- admin\_export\_structure
- admin\_import
- admin\_update\_field
- admin\_update\_agreed\_price
- admin\_update\_5203\_calculation
- admin\_update\_row\_field
- admin\_apply\_special\_products
- \_calculateSpecialComponentHours
- \_cleanSpecialChars
- \_applySettings
- \_prepareMaterialRowData
- \_saveMaterialRows
- \_determineTypeFromDescription
- \_determineEngineFromDescription
- \_getTypeOptions
- \_getEngineOptions
- \_getEnginesByTypeMapping
- \_applyPostImportProcessing

Kuva 8. Pääohjainluokan metodeja.

```

public function admin_update_settings($id = null) {
    $this->autoRender = false;
    $this->response->type('json');

    if (!$id && isset($this->request->data['id'])) {
        $id = $this->request->data['id'];
    }

    if (!$id) {
        echo json_encode(['success' => false, 'message' => 'Invalid material ID']);
        return;
    }

    // Lataa materiaali nykyisellä tyyppillä ja moottorilla
    $material = $this->Material->findById($id);
    if (!$material) {
        echo json_encode(['success' => false, 'message' => 'Material not found']);
        return;
    }

    // Lataa moottoriaisetukset tälle tyyppi/moottori-yhdistelmälle
    $engineSettings = $this-> EngineSetting->find('first', array(
        'conditions' => array(
            'tyyppi_name' => $material['Material']['type'],
            'engine_name' => $material['Material']['engine']
        )
    ));

    if (empty($engineSettings)) {
        echo json_encode(['success' => false, 'message' => 'No engine settings found for this combination']);
        return;
    }

    // Päivitä materiaali oikeilla asetuksilla
    $this->Material->id = $id;
    $updateData = array(
        'settings_basic_hours' => $engineSettings['EngineSetting']['basic_hours'],
        'settings_small_accessories' => $engineSettings['EngineSetting']['small_accessories']
    );

    // Hae globaalit asetukset varmuusarvoilla
    if (!isset($this->Setting)) {
        $this->loadModel('Setting');
    }
}

```

Kuva 9. admin\_update\_settings-ohjaintoiminnon hakutoimintoja.

Näkymät toteutettiin CakePHP .ctp-tiedostoina, jotka sisältävät HTML-koodia ja PHP-logiikkaa. Bootstrap-kehystä ja jQuery-kirjastoa hyödynnettiin käyttöliittymän toteutuksessa. Tietokannan rakenne määriteltiin SQL-lausein, joita käytettiin kehitys- ja tuotantoympäristöissä. Tietokantayhteydet määriteltiin CakePHP-konfiguraatiotiedostoissa, jotka sisälsivät tietokantapalvelimen, käyttäjänimen, salasanan ja tietokannan nimen (ks. kuva 10).

```

<div class="Materials form">
  <h2>Tarjouslaskuri </h2>
  <h3>Lisää uusi tarjous</h3>

  <form action="<?php echo $this->Html->url(array('action' => 'add')); ?>" method="post">
    <div class="panel panel-default">
      <div class="panel-heading">
        <h4 class="panel-title">Perustiedot</h4>
      </div>

      <div class="panel-body">
        <!-- Type field -->
        <div class="field-container">
          <label class="field-label">Type</label>
          <div class="field-input">
            <select name="data[Material][type]" id="MaterialType">
              <option value="">-- Valitse --</option>
              <?php foreach($typeOptions as $key => $value): ?>
                <option value="<?php echo $key; ?>"><?php echo $value; ?></option>
              <?php endforeach; ?>
            </select>
          </div>
        </div>

        <!-- Engine field -->
        <div class="field-container">
          <label class="field-label">Engine</label>
          <div class="field-input">
            <select name="data[Material][engine]" id="MaterialEngine">
              <option value="">-- Valitse --</option>
              <?php foreach($engineOptions as $key => $value): ?>
                <option value="<?php echo $key; ?>"><?php echo $value; ?></option>
              <?php endforeach; ?>
            </select>
          </div>
        </div>
      </div>
    </div>
  </form>

```

Kuva 10. Tarjoussivun HTML-rakennetta.

### 3.7 Tiedonkäsittely ja hinnanlaskentalogiikka

Sovellus pystyy lukemaan asiakkaan Excel-tiedostoja, laskemaan tarjouskustannukset ja työkustannukset. Sovellus myös tuottaa tarjouksia asiakkaan formaatissa. Excel-tiedostojen tuonnissa käytettiin PhpSpreadsheet -kirjastoa ja tuontiprosessi sisälsi seuraavat vaiheet:

1. Tiedoston lataaminen: Käyttäjä lataa Excel-tiedoston sovellukseen.
2. Tiedoston validointi: Tarkistetaan, että tiedostolla on oikea formaatti ja se sisältää tarvittavat sarakkeet.
3. Tietojen lukeminen: Luetaan tuotetiedot Excel-tiedostosta.
4. Tietojen validointi: Varmistetaan tietojen muoto ja vaatimukset.
5. Tietojen tallennus: Tallennetaan tiedot tietokantaan.

```

public function admin_import() {
    // Check
    $currentMaterialId = null;
    $currentMaterial = null;

    if (!empty($this->request->query['material_id'])) {
        $currentMaterialId = $this->request->query['material_id'];
        $currentMaterial = $this->Material->findById($currentMaterialId);
    }

    if ($this->request->is('post')) {
        if (!empty($this->request->data['Material']['upload_file']['tmp_name'])) {
            $file = $this->request->data['Material']['upload_file'];

            // Hyväksytyt formaatit
            $allowedTypes = array(
                'application/vnd.ms-excel',
                'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',
                'application/octet-stream',
                'text/csv',
                'text/plain'
            );

            if (!in_array($file['type'], $allowedTypes)) {
                $this->Session->setFlash('File type not allowed. Please use Excel or CSV file.', 'default', array('class' => 'alert alert-danger'));
                return $this->redirect(empty($currentMaterialId) ? array('action' => 'edit', $currentMaterialId) : array('action' => 'index'));
            }

            // aseta kohde id
            $targetMaterialId = null;
            if (!empty($this->request->data['Material']['target_material_id'])) {
                $targetMaterialId = $this->request->data['Material']['target_material_id'];
            }
        }
    }
}

```

Kuva 11. Tuontitoiminnon tiedostotyyppin ja kohde-ID:n tarkistus.

Hinnanlaskentalogiikka sisälsi useita vaiheita. Jokaisen tuoterivin hintatiedot haettiin erillisen järjestelmän tietokannasta tuotekoodin perusteella. Työkustannukset lasketaan tuotetyypin, materiaalityypin ja erikoiskomponenttien määrän perusteella. Perustyötunnit määräytyivät tuote- ja materiaalityypin mukaan. Erikoiskomponenttien määrä vaikuttaa työtunteihin lisäämällä tietyn määrän tunteja per komponentti. Säättöarvoissa määritellään tuntihinta, joka kerrotaan työtunneilla. Kate lasketaan prosenttiosuutena kokonaiskustannuksista. Kateprosentti määritellään myös säättöarvoissa.

### 3.8 Hierarkkinen laskenta SQL-näkymällä

Tarjouslaskentasovelluksen keskeinen ominaisuus on hierarkkinen laskenta, joka toteutettiin SQL-näkymän avulla. Asiakkaan materiaaliluettelot ovat monitasoisia rakenteita, joissa alempien tasojen määrät vaikuttavat ylempien tasojen kokonaismääriin ja siten koko tarjouksen hinnanlaskentaan.

Hierarkkisessa laskennassa haasteena on määrittää jokaisen rivin todellinen kokonaismäärä (TotalQty). Tämä määrä lasketaan kertomalla materiaalirivin oma määrä sen alatasojen määrillä. Esimerkiksi jos tasolla 0 on yksi tuote, jonka alapuolella on tasolla 1 kolme tuotetta ja tasolla 2 kaksi tuotetta, tasolla 2 olevan tuotteen kokonaismäärä on 6.

Laskentaan kehitettiin SQL-näkymä, joka suorittaa hierarkkisen laskennan tehokkaasti. Näkymä käy jokaisen rivin läpi ja seuraa materiaalirakennetta tasoittain (lvl0qty-lvl5qty) selvittäen jokaisen tason määrän ja laskiessaan niiden perusteella lopullisen kokonaismäärän sarakkeeseen TotalQTY. Laskennassa otetaan huomioon ainoastaan rivit, joiden calculate\_bit on asetettu (ks. kuva 12).

	materialrow_id	material_id	material_level	materialrow_quantity	lvl0qty	lvl1qty	lvl2qty	lvl3qty	lvl4qty	lvl5qty	TotalQTY
1	1	123456	0	1.0000	1.0000	NULL	NULL	NULL	NULL	NULL	1.000000
2	2	123456	1	2.0000	NULL	2.0000	NULL	NULL	NULL	NULL	2.000000
3	3	123456	2	3.0000	NULL	NULL	3.0000	NULL	NULL	NULL	6.000000
4	4	546296	0	1.0000	1.0000	NULL	NULL	NULL	NULL	NULL	1.000000
5	5	546296	1	1.0000	NULL	1.0000	NULL	NULL	NULL	NULL	1.000000
6	6	546296	2	1.0000	NULL	NULL	1.0000	NULL	NULL	NULL	1.000000
7	7	546296	3	8.0000	NULL	NULL	NULL	8.0000	NULL	NULL	8.000000
8	8	546296	4	1.0000	NULL	NULL	NULL	NULL	1.0000	NULL	8.000000
9	9	546296	5	20.3500	NULL	NULL	NULL	NULL	NULL	20.3500	162.800000
10	10	546296	4	45.0000	NULL	NULL	NULL	NULL	45.0000	NULL	360.000000
11	11	546296	4	6.0000	NULL	NULL	NULL	NULL	6.0000	NULL	48.000000
12	12	546296	3	1.0000	NULL	NULL	NULL	1.0000	NULL	NULL	1.000000
13	13	546296	4	1.0900	NULL	NULL	NULL	NULL	1.0900	NULL	1.090000
14	14	546296	3	1.0000	NULL	NULL	NULL	1.0000	NULL	NULL	1.000000
15	15	546296	4	1.0900	NULL	NULL	NULL	NULL	1.0900	NULL	1.090000
16	16	546296	3	12.0000	NULL	NULL	NULL	12.0000	NULL	NULL	12.000000
17	17	546296	4	0.1300	NULL	NULL	NULL	NULL	0.1300	NULL	1.560000
18	18	546296	3	120.0000	NULL	NULL	NULL	120.0000	NULL	NULL	120.000000
19	19	546296	4	0.0900	NULL	NULL	NULL	NULL	0.0900	NULL	10.800000
20	20	546296	3	8.0000	NULL	NULL	NULL	8.0000	NULL	NULL	8.000000
21	21	546296	4	0.0500	NULL	NULL	NULL	NULL	0.0500	NULL	0.400000

Kuva 12. SQL-näkymän esimerkkitulokset hierarkkisesta laskennasta.

### 3.9 Ajax -käyttöliittymän responsiivisuus

Lisätyökalun käyttöliittymää ehostettiin Ajax-toiminnallisuudella, joka mahdollistaa tietojen päivittämisen ilman koko sivun uudelleenlatausta. Sovelluksesta tulee tällöin responsiivisempi ja nopeampi käyttää. Ajax-

toiminnot toteutettiin jQuery-kirjaston avulla, joka integroitiin osaksi CakePHP-näkymiä (jQuery Foundation, 2024).

Ajax-toimintoja sovelluksessa ovat:

1. Hintojen päivitys +kustannuslaskenta: Päivittää hinnat ja laskee kustannukset automaattisesti.
2. Kustannuslaskennan muokkaaminen: Tyypin tai moottorin valintalistojen muutokset, jossa muutetaan tarjoukseen asetettuja alkuperäisiä asetuksia.
3. Revisioiden luominen.
4. Engine-valikkojen dynaaminen päivitys: Moottorivalikko päivittyy valitun moottorityypin perusteella.
5. File upload drag & drop: Excel-tiedostojen vetäminen ja pudottaminen hakemistosta ohjelmaan.
6. Modal-ikkunoiden Ajax-sisältö: Moottorien säätöarvojen muokaus modal.
7. Tekstikenttien automaattinen kasvu: Notes-sarakkeen tekstikentät kasvaa sisällön mukaan.
8. Virhepalautteen näyttö: Kaikki Ajax-kutsut palauttavat onnistumis- tai virheviestin.
9. Loading-indikaattorit.
10. Automaattinen tallennus: Tallentaa muutokset automaattisesti, kun käyttäjä poistuu kentästä.

## 4 TULOKSET JA ARVIOINTI

Tarjouslaskentatyökalu on kehitetty käyttöön Escarmatille. Työkalun siirtoa tuotantoympäristöön ei ole vielä suoritettu. Sovellus siirretään kehitysympäristöstä tuotantokäyttöön loppuvalmistelujen jälkeen. Tulevia käyttäjiä varten lisätyökalulle laadittiin käyttöohjeet, joissa käydään läpi sovelluksen perustoimintoja: uuden tarjouksen luominen, materiaaliuuttelun tuominen Excelistä, laskennan suoritus, tarjouksen vienti Exceliin ja tarvittaessa muokkaaminen.

Lisätyökalua testattiin esimerkkitapauksilla kehitysprosessin aikana. Käyttäjät pitivät sovellusta selkeänä ja helppokäyttöisenä. Lisätyökalun todettiin tuottavan tarvittavat oikeat laskelmat. Reaaliaikainen palaute tarjousten kustannuksista ja katteesta koettiin hyödylliseksi. Käyttäjätestauksesta voitiin todeta, että sovellus tehostaa tarjousten laatimista ja se on käyttäjäystävällinen.

Kehitysvaiheessa lisätyökalulle suoritettiin yksikkötestausta kriittisille laskentafunktioille. Laskenta saatiin toimimaan kehityksen loppuvaiheessa. Laskenta tuottaa oikeat tulokset eri syötearvoilla ja tunnistaa tarvittavat komponentit, joiden avulla laskennasta saadaan oikeita arvoja materiaaliin, voittoon, kokonaishintaan, työkustannukseen, materiaalikatteeseen ja erotukseen. Käyttäjätestausta tehtiin valikoitujen henkilöiden kanssa. Käyttäjätestaajat kokeilivat sovellusta todellisilla esimerkkitapauksilla ja tulosten perusteella laskentalogiikka toimi oikein.

Tavoitteet sovelluksen osalta on saavutettu kehitysvaiheessa. Laskentaprosessi automatisointiin ja lopputuloksena saavutettiin käyttäjäystävällinen sovellus. Voidaan arvioida, että lisätyökalu tulee parantamaan käyttäjien kokemusta laskennasta ja tulevat käyttäjät sitä osaavat käyttää. Sovellusta voidaan mahdollisesti laajentaa tukemaan useampia asiakkaita tulevaisuudessa.

## 5 YHTEENVETO

### 5.1 Toteutus ja tutkimuskysymykset

Tässä opinnäytetyössä tavoitteena oli kehittää Escarmat Oy:lle asiakasräätälöity tarjouslaskentatyökalu. Työn taustalla oli tarve tehostaa tarjouslaskentaa, vähentää manuaalista laskentaa ja suunnitella sovellus, jota työntekijät vähäiselläkin teknisillä taidoilla osaisivat käyttää. Teoriaosuudessa perehdyttiin tarjouslaskennan periaatteisiin sekä ohjelmistokehityksen teknologioihin.

Opinnäytetyön tutkimuskysymyksiksi tarkentui kehityksen aikana:

1. Miten kustannuslaskenta toteutetaan?
2. Miten asiakas saa haluamassaan formaatissa tarjouksen takaisin?
3. Miten käyttöliittymästä saadaan käyttäjäystävällinen?

Toteutettu lisätyökalu vastaa kehityksen aikana ilmeneviin kysymyksiin. Lisätyökalu lukee asiakkaan Excel-muotoisen materiaaliluettelon, tulkitsee sen sisällön, hakee hintatiedot, suorittaa hierarkkisen laskennan ja halutut tiedot pystytään viemään Excel-muotoon takaisin tarjous muodossa. Tiedot voidaan myös viedä Excel-muotoon pelkkänä rakenteena, jossa näkyy ainoastaan työtunnit, materiaalikoodi, kuvaus ja määrä. Käyttöliittymä on selkeä ja se ohjaa käyttäjää tarjousprosessin läpi. Käyttäjäpalaute on ollut positiivista.

### 5.2 Itsearviointi ja johtopäätökset

Sovellus rakennettiin CakePHP 2 -sovelluskehityksen päälle, ja arkkitehtuuri noudatti MVC-mallia. Koodista tuli selkeää ja tietokantarakenne suunniteltiin huolellisesti. Ohjelma pystyy hakemaan ajantasaisia hintatietoja automaattisesti. Excel-tiedostojen tuonti ja vienti toimivat sujuvasti. Kokonaisuutena työ on erittäin onnistunut työmäärään nähden ja

lopputulos täyttää asetetut vaatimukset. Ohjelma on toimeksiantajalle hyödyllinen ja sitä voi tulevaisuudessa laajentaa, mikä mahdollistaa jatkokehityksen mahdollisille muille asiakkuuksille.

Työ tarjosi mahdollisuuden soveltaa ohjelmistokehityksen teoriaa ja omaa osaamista yrityksessä. Opin työn aikana käyttämään CakePHP-2 kehystä, soveltamaan MVC-arkkitehtuuria ja taitoni tietokantasuunnittelussa kehittyivät. Työ oli kokonaisuudessaan haastava toteuttaa yksin, mutta apua sai palavereissa tai kysyttäessä. CakePHP 2 -sovelluskehityksen käyttö on parantanut ymmärrystäni MVC-arkkitehtuurista ja käytöstä web-sovelluskehityksessä. Olen huomattavasti syventänyt osaamistani ohjelmistokehityksestä ja siitä, kuinka liiketoimintaympäristössä kehitystä voidaan soveltaa. Projektin lopputulos oli mielestäni onnistunut ja asiakaskohtainen tarjouslaskuri otetaan käyttöön kesäkuun 2025 aikana.

## LÄHTEET

- Apache Software Foundation. (n.d.). *Apache HTTP Server Project*. Noudettu 29.5.2025 osoitteesta <https://httpd.apache.org/>
- Bootstrap. (n.d.). *Introduction · Bootstrap v3.3*. Noudettu 26.5.2025 osoitteesta <https://getbootstrap.com/docs/3.3/>
- CakePHP. (2024a). *CakePHP 2.x Cookbook Documentation*. Noudettu 12.5.2025 osoitteesta <https://book.cakephp.org/2/en/index.html>
- CakePHP. (2024b). *Understanding Model-View-Controller*. Noudettu 2.6.2025 osoitteesta <https://book.cakephp.org/2/en/cakephp-overview/understanding-model-view-controller.html>
- Devane, H. (12.5.2022). *Data anonymization techniques: What they are and how to use them*. Immuta. Noudettu 29.5.2025 osoitteesta <https://www.immuta.com/blog/data-anonymization-techniques/>
- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems (7. painos)*. Pearson. Noudettu 29.5.2025 osoitteesta <http://ir.harambeeuniversity.edu.et/bitstream/handle/123456789/1810/Fundamentals%20of%20Data-base%20Sys-tems%20.pdf%20%28%20PDFDrive.com%20%29.pdf?sequence=1&isAllowed=y>
- Escarmat Oy. (2024). *Escarmat – Sähkö- ja automaatiokeskukset teollisuuteen*. Noudettu 26.5.2025 osoitteesta <https://www.escarmat.com/>
- Huhtala, A. (2014). *Tietokanta-arkkitehtuurin suunnittelu ja optimointi joukkoistamista hyödyntävään palveluun* [Diplomityö, Oulun yliopisto]. Oulun yliopiston julkaisuarkisto. <https://oulu-repo.oulu.fi/bitstream/handle/10024/39029/nbnfioulu-201404271313.pdf>

- Iskanius, P., Klaavu, M., & Myllyoja, J. (2009). *Tehokas tarjouslaskenta – prosessi, järjestelmät ja kehittämishaasteet (TOMI-raportti 2). Teknillinen korkeakoulu.* Noudettu 26.5.2025 osoitteesta [https://www.researchgate.net/publication/253340906\\_Toiminnanohjausjärjestelmien\\_vaatimusmaarittelyn\\_laatiminen\\_pk-yri-tysten\\_kayttoon\\_TOMI\\_raportti\\_2](https://www.researchgate.net/publication/253340906_Toiminnanohjausjärjestelmien_vaatimusmaarittelyn_laatiminen_pk-yri-tysten_kayttoon_TOMI_raportti_2)
- jQuery Foundation. (2024). *jQuery API Documentation.* Noudettu 29.5.2025 osoitteesta <https://api.jquery.com/>
- Lemonsoft Oyj. (2024). *Lemonsoft ERP - Toiminnanohjausjärjestelmä.* Noudettu 29.5.2025 osoitteesta <https://news.lemonsoft.fi/toiminnanohjausjarjestelma-erp>
- Microsoft Corporation. (2025). *Database normalization description.* Noudettu 2.6.2025 osoitteesta <https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description>
- Microsoft Corporation. (2024). *SQL Server documentation.* Noudettu 26.5.2025 osoitteesta <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver17>
- Mozilla. (2024). *HTTP request methods.* Noudettu 29.5.2025 osoitteesta <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Methods>
- Notepad++ (n.d). *What is Notepad++.* Noudettu 29.5.2025 osoitteesta <https://notepad-plus-plus.org/>
- OWASP. (n.d.). *OWASP Top Ten.* Noudettu 26.5.2025 osoitteesta <https://owasp.org/www-project-top-ten/>
- PhpSpreadsheet. (2024). *PhpSpreadsheet Documentation.* Noudettu 26.5.2025 osoitteesta <https://phpspreadsheet.readthedocs.io/en/latest/>

PHP.net. (2024). *PHP: Hypertext Preprocessor*. Noudettu 26.5.2025 osoitteesta <https://www.php.net/>

Skycode. (2021). *SkySeller Tarjouslaskenta*. Noudettu 26.5.2025 osoitteesta <https://skycode.fi/skyseller-tarjouslaskenta/>

Skycode. (2023). *Tarjouslaskenta*. Noudettu 29.5.2025 osoitteesta <https://skycode.fi/project/escarmat/>