VERKKOKAUPAN INTEGROINTI TOIMINNANOHJAUSJÄRJESTELMÄÄN

LAHDEN AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikka
Ohjelmistotekniikka
Opinnäytetyö
Kevät 2011
Leo Kallioinen

Lahden ammattikorkeakoulu
Tietotekniikka

KALLIOINEN, LEO: Verkkokaupan integrointi toiminnanohjausjärjestelmään

Ohjelmistotekniikan opinnäytetyö 35 sivua

Kevät 2011

TIIVISTELMÄ

Opinnäytetyössä rakennetaan PHP:n, AJAX:n ja SQL-tietokantojen avulla Udda Finland Oy:lle automaattisesti päivittyvä verkkokaupa, jonka sisältö generoidaan Uddan käyttämän toiminnanohjausjärjestelmän tietokannasta. Luotu verkkosivusto julkaistaan Louhi Networksin palvelimilla.

Asiakkaiden rekisteröityminen tallennetaan Louhen tietokantaan ja tili tulee aktivoida Uddan toimesta ennen kuin se on käytettävistä. Ilman rekisteröitymistä tuotteita ei voi tilata eikä niiden hintoja näytetä.

Verkkokauppaan luodaan erillinen hallintapaneeli, jonka kautta voidaan muokata sivulla näkyviä tuotekategorioita ja näiden nimiä. Kategoriaa klikattaessa sivustolle tulostetaan sivunumeroitu lista kategoriaan kuuluvista tuotteista, joita on varastossa tai on varastoon tulossa. Tuotteista tulostetaan myös tarkempi tuotekuvaus.

Jokaiselle tuotteelle haetaan automaattisesti tuotekuva Uddan FTP-serveriltä. Kuvista luodaan samalla myös GD2-kirjaston avulla thumbnail ja tarvittaessa pienennetään alkuperäistä kuvaa, jos se on liian suuri. Epäonnistuneet siirrot kirjoitetaan logiin.

Verkkokaupassa on ostoskori, jonka tuotteet tallennetaan sessioon PHP:n avulla. Ostoskorista asiakas voi siirtyä kassalle, jonka kentät täytetään Louhella sijaitsevasta asiakastietokannasta ja session ostoskorista. Tilausvahvistus lähetetään PEAR:n Mail PHP -laajennuksella sekä asiakkaan että Uddan sähköpostiin.

Opinnäytetyössä tutustutaan evästeiden turvallisuuteen ja siihen, kuinka suojaudutaan SQL-injektioilta. Työ onnistuttiin toteuttamaan määritellyin puitteiden mukaisesti. Udda Finland on tyytyväinen lopputulokseen, ja sivustoa tullaan mahdollisesti vielä jatkokehittämään.

Avainsanat: verkkokauppa, PHP, SQL, AJAX, GD2, jQuery

Lahti University of Applied Sciences
Degree Programme in information technology

KALLIOINEN, LEO: Integration of webshop to an enterprise solution planning.

Bachelor's Thesis in software engineering 35 pages

Spring 2011

ABSTRACT

The objective of the thesis was to build a webshop for Udda Finland Oy using PHP, AJAX and SQL databases. The developed webshop will be automatically updating its content, which is generated from the ERP database used by Udda. The website will be published on Louhi Networks' servers.

Guest registration is stored in Louhi's MySQL database and the account must be activated by Udda before it can be used. Without registration the products cannot be ordered and their prices are not displayed.

The online store has a separate control panel through which you can edit the product categories and their names that are listed on the page. Clicking the category on the webshop produces a page listing available products. Products can also be printed with more detailed description.

For each product a thumbnail image will be generated, with a GD library and, when needed, the original picture is downsized for better viewing. All images are downloaded from Udda's FTP server.

All items can be stored to the shopping cart of the webshop which is created with PHP sessions. From the shopping cart the customer can go to the checkout where the fields will be filled from Louhi's customer database and the shopping cart session. Order information is mailed to the customer and Udda Finland with PEAR's Mail PHP extension.

The thesis discusses the issue of the cookie security as well as protection from SQL injections. The work was successfully implemented within the defined parameters. Udda Finland is satisfied with the outcome and the site will possibly be developed even further.

Key words: webshop, PHP, SQL, AJAX, GD, jQuery

SISÄLLYS

LYHENNELUETTELO

AJAX	*Asynchronous JavaScript And XML.* Nimitys web-ohjelmointitavasta, joka luo sivustolle interaktiivisuutta ilman, että sivua tarvitsee päivittää.
jQuery	Avoimen lähdekoodin ja selainriippumaton JavaScript-kirjasto.
SHA-1	*Secure Hash Algorithm.* Kryptograafinen tiivistealgoritmi, jolla luodaan annetusta tekstistä 160 bittiä pitkä tiiviste.
SQL	*Structured Query Language.* Standardoitu kyselykieli, jolla haetaan, päivitetään ja muokataan relaatiotietokantojen tietoja.
PEAR	*PHP Extension and Application Repository.* Repositorio PHP-laajennuksille ja komponenteille.
PHP	*Hypertext Preprocessor.* Ohjelmointikieli, jota usein käytetään palvelinpuolella, kun luodaan dynaamisia internetsivuja.

1 JOHDANTO

Oy Udda Finland Ltd on suomalainen maahantuonti- ja agentuuritoimintaan keskittynyt perheyhtiö, joka on toiminut alalla jo kahden vuosikymmenen ajan. Udda Finlandin toimisto sijaitsee Lahdessa, jossa työskentelee kuusi ihmistä.

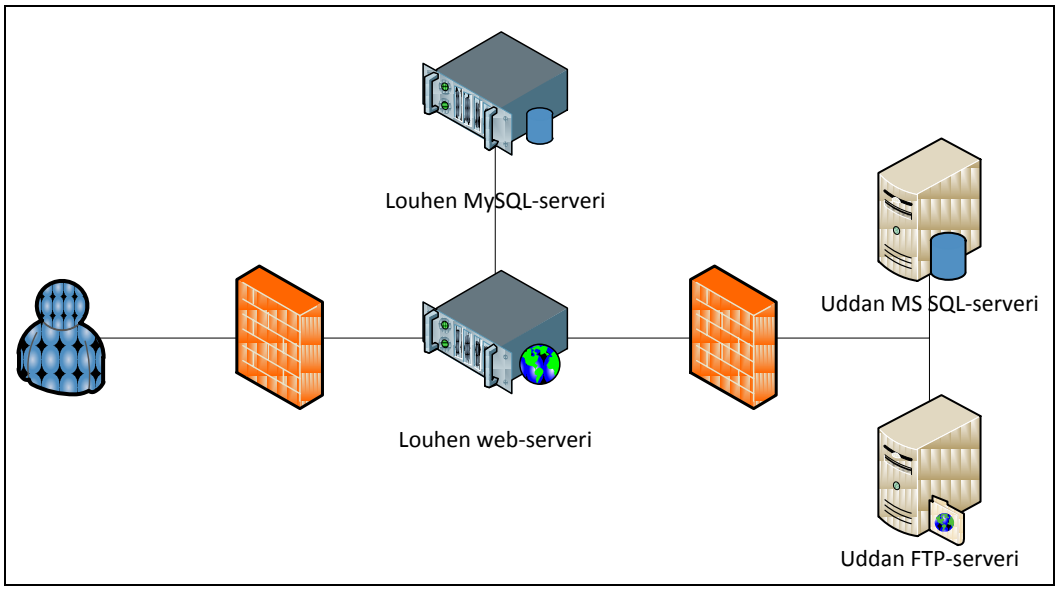
Tällä hetkellä Udda Finlandin klikkaa.fi:ltä tilaaman verkkosivuston manuaalinen päivittäminen on vaivalloinen kolmivaiheinen prosessi, joka käsittää ensiksi kolmannen osapuolen tuottamalla ohjelmalla tuotteiden tulostamisen Excel-tiedostoon, jota vielä tekstieditorilla joudutaan käsin muokkamaan ennen kuin sen voi käydä lähettämässä sivustolle. Tämän vuoksi Udda Finland päätyi hakemaan uutta ratkaisua verkkosivuille, joka hoitaisi tuotteiden päivityksen itse.

Tämän työn tavoitteena on luoda verkkokauppa, joka koostuu monista eri verkkokaupalle tyypillisistä komponeista, joiden avulla luodaan sulava kokonaisuus ja selvittää, mitä kyseisiltä komponenteilta vaaditaan ja miten ne voitaisiin toteuttaa annetuilla työkaluilla. Työssä ei syvennytä sivuston ulkonäöllisiin vaatimuksiin. Verkkosivun tulisi luoda sivuilla näkyvät tuotteet ja tuotekategoriat automaattisesti suoraan Uddan toimistossa sijaitsevalta serveriltä. Uddan tulee myös pystyä hallitsemaan, mitkä kategoriat ja tuotteet näkyvät verkkosivuilla milläkin nimillä, ja lisäämään jokaiselle yksittäiselle tuotteelle tarkempi tuoteseloste. Lisäksi rekisteröityvien asiakkaiden tilejä täytyy voida asettaa aktiivisiksi tai epäaktiiviseksi, vaikuttaen näiden kykyyn kirjautua sivuille. Vaikka tuotteita voi selata vapaasti kirjautumatta, niin hintatietoja ei haluta näyttää kaikille ulkopuolisille.

Työssä käytetään unix-pohjaista palvelinta verkkosivua varten ja käytetään webhotelli-palveluntarjoajan asentamia työkaluja ja kirjastoja, joihin lukeutuu PHP5, GD2-kirjasto ja MySQL. Työssä on myös käytössä Microsoft SQL Server.

2 SERVERIT

Työssä on käytössä kaksi eri serveriä, joista ensimmäinen sijaitsee Louhi Networks Oy:n tiloissa ja toinen sijaitsee Udda Finlandilla. Louhen SQL- ja web-serverit sijaitsevat samalla Louhen palvelimella. Myös Udda Finlandin FTP- ja MS SQL -serverit sijaitsevat samalla Udda Finlandin palvelimella. (Kuvio 1.)



KUVIO 1. Käytössä olevat serverit

2.1 Udda Finlandin toimiston serveri

Uddan toiminnanohjausjärjestelmä koostuu Windows Server 2008:lla käyvästä Visma Nova C/S Pro -nimisestä ohjelmistosta, jolla voidaan ohjata yrityksen taloushallintoa, henkilöstöhallintoa, tuotannonohjausta. Se on samalla materiaalinohjausjärjestelmä**.** Se sisältää myös sähköistä arkistointia, tiedonsiirtoa ja Internetkaupankäyntiä tukevat ominaisuudet. Ohjelmistokokonaisuus voi sisältää myös räätälöitäviä ohjelmia ja raportteja. (Visma Nova 2011.) Visma Nova C/S Pron yhteydessä serverille asentuu Micrsoft SQL Server, johon kaikki tieto tallennetaan.

Jotta Uddan SQL-severiin voitaisiin ottaa yhteyttä, niin Uddan reitittimeen luotiin sääntö, joka sallii ulkopuoliset yhteydet sekä kehityskoneelta että Louhen web-hotellista.

2.2 Louhen serveri

> *Louhi Networks Oy on Suomen johtava palvelinten ja palveluiden isännöintiin (Hosting) keskittynyt yritys. Palvelinkeskuksissa on ylläpidossa ja 24/7 valvonnassa yli 10.000 webhotellia ja satoja liiketoimintakriittisiä verkkokauppoja.* (Louhi 2011.)

Udda Finlandin valitsema web-hotellipalvelu on nimeltään Enterprise ja tällä hetkellä Louhen kattavin palvelinratkaisu. Työn kannalta olennaista on, että kyseinen palvelu tarjoaa Uddan käyttöön 5120 Mt levytilaa, PHP5:n ja SQL-tietokannan. Lisäksi Louhella on käytössä maksuton tekninen tuki, joka vastaa nopeasti palvelupyyntöihin.

3 VERKKOKAUPAN OSA-ALUEET

3.1 Hallinnointi

Kategorioita, tuotteita ja tuotteiden tietoja on hyvä päästä muokkaamaan. Tästä syystä verkkokaupalle on luotava hallinnointipaneeli. Hallinnointi voitaisiin upottaa sivuston sekaan niin, että muokattavat elementit näkyisivät vain oikeilla tunnuksilla kirjauduttaessa. Tämä saattaisi aiheuttaa paljon ylimääräistä koodia, ehdonalaisuuksia ja mahdollisesti tietoturvariskejä, jos sivuston HTML-koodin seassa tahattomasti näkyisi elementtejä, joiden kuuluisi olla piilossa. Näistä syistä päädyttiin ratkaisuun, jossa toteutetaan hallintapaneelit sivustosta irralleen ja jotka voi piilottaa salasanan taakse. Tämä lähestymistapa voi olla myös paljon eheämpää ja visuaalisesti ystävällisempi käyttäjäkokemuksen mielessä, sillä selain ei silloin täyty erilaisista painikkeista. Kategorioille ja asiakastileille olisi hyvä luoda omat hallintapaneelit.

Kategorioiden hallintapaneelista tulee olla mahdollista yksitellen valita, mitkä kategoriat tulevat näkyville, ja asettaa niille vaihtoehtoinen nimi, sillä jotkut Novassa käytetyt kategorioiden nimet saattavat olla ei-toivottuja kilpailijoiden silmille. Näkyvyysasetukset tullaan tallentamaan Louhen tietokantaan, jolloin pystytään vähentämään SQL-kyselyiden määrää Uddan serveriltä ja saadaan sivusto latautumaan nopeammin. Hallintapaneeli voi olla ulkonäöltään yksinkertainen, eikä sen tarvitse noudattaa muun sivuston teemaa, sillä paneelin tarkoituksena on toiminnallisuus ja sitä ei tulla käyttämään kuin Udda Finlandin työntekijöiden toimesta. Kategorioiden hallintapaneelin ulkonäkö on havainnoillistettu kuviossa 2.

Asiakastilien hallinta voi noudattaa samaa yksinkertaista kavaa sillä erotuksella, ettei tilien tietojen ei tarvitse olla muokattavaissa muuten kuin siltä osin, että onko tili aktiivinen vai suljettu. Asiakkaat voivat itse tarvittaessa päivittää yhteystietojaan.

Kategorioiden näkyvyys			
Koodi	Nimi	Vaihtoehtonimi	Näkyvillä
0			☐
9008	AEG	AEG	☑
3502	Allibert	Allibert	☑
4002	Bentom Muovit	Bentom Muovit	☑
9007	Bomann	Bomann	☑
9006	Clatronic	Clatronic	☑
9125	Disney astiat	Disney astiat	☑
9116	Eesti Sauna	Sauna	☑
9005	Erätuotteet	Erätuotteet	☐
9002	Erätuotteet	Erätuotteet	☐
3501	Heidrun	Heidrun	☑
9126	Inter	Inter	☑
7010	Jalkapallotarvikkeet	Jalkapallotarvikkeet	☐

KUVIO 2**.** Hallintapaneeli kategorioiden näkyvyyksille

3.2 Rekisteröityminen

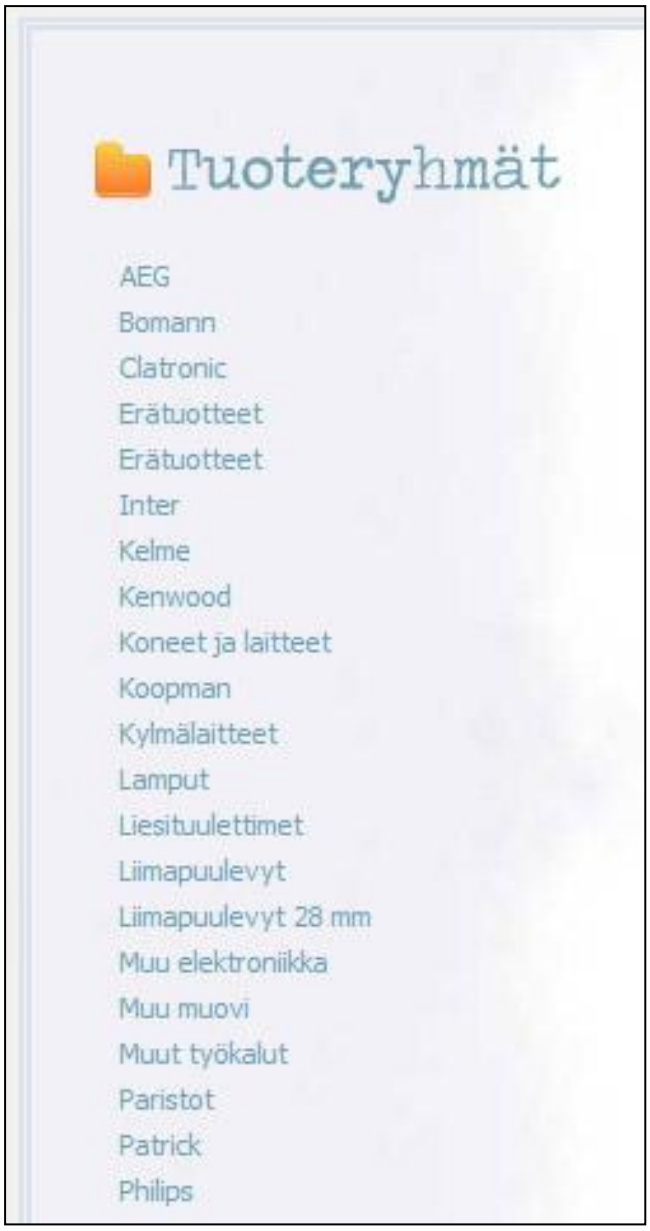
Verkkokauppoihin rekisteröityminen on useimmiten vapaaehtoista, mutta tässä tapauksessa siitä on haluttu pakollista. Jokainen voi halutessaan rekisteröityä asiakkaaksi, Uddan pyynnöstä jokainen tili tulee silti voida henkilökohtaisesti tarkastaa ennen aktivointia. Jokaisesta rekisteröitymisestä tulee lähettää Uddalle sähköposti, jossa on linkki, millä voidaan asettaa rekisteröityneen käyttäjän tili aktiiviseksi. Ennen aktivointia tiliä ei voi käyttää eivätkä tuotteiden hinnat ole näkyvillä. Asiakastietokantaa säilytetään Louhen serverillä.

Asiakkaalta otetaan talteen yhteyshenkilön nimi ja joko hänen tai yrityksen yhteystiedot. Asiakkaan tunnuksena toimii rekisteröintivaiheessa annettu sähköposti. Tietoturvan vuoksi salasanoja ei tallenneta tietokantaan selkokielisinä, vaan tunnukselle generoidaan tietokantaan tiivistesumma asiakkaan antamasta sähköpostiosoitteesta, henkilökohtaisesta salasanasta ja kolmesta eri suolasta. Suola on nimitys ennaltamääritetylle tekstille tai arvolle, joka ripotellaan muodostettavan kokonaisuuden väliin, jotta ulkopuolisten ihmisten olisi hankalempi selvittää tunnuksesta luotavaa tiivistesummaa. Mitä satunnaisempia suolat ovat, sitä vaikeampi tunnuksen arvaaminen on hyökkääjälle. Paremman tietoturvan vuoksi vähintään yhden suolan on hyvä olla dynaaminen, mikä voisi olla esimerkiksi käyttäjän rivinumero SQL-tietokannassa.

Lopuksi vielä tunnuksesta suoloineen luodaan PHP:llä sha1-tiivistesumma. PHP:llä on valittavissa tiiviste-algoritmeista metodit md5 ja sha1, joista päädyttiin sha1:een, koska se tuottaa vähemmän konflikteja. Tämä tarkoittaa sitä, ettei lopputulokseksi saada yhtä usein samaa tiivistesummaa useista eri sanoista . Kirjautumistapahtumassa tiivistesumma luodaan uudestaan samalla algoritmillä ja verrataan, vastaako se samaa salattua avainta, joka asiakkalle on asetettu asiakastietokantaan.

3.3 Tuotekategoriat

Sivuston vasempaan laitaan on listataan kaikki ne tuotekategoriat, jotka Udda valinnut kategorioiden hallintapaneelista näkyville. Kategoriat tulostuvat siis tällöin Louhen SQL-tietokannasta, jolloin sivulataus sujuu nopeammin. Kategoriat tulee tulostaa css-tyyliteltyinä HTML-listana ja niistä luodaan linkit, joiden mukaan sivuston keskustaan tulostetaan kyseisen kategoriaan kuuluvat tuotteet. Kuviossa 3 ilmenee lopullinen ulkonäkö.



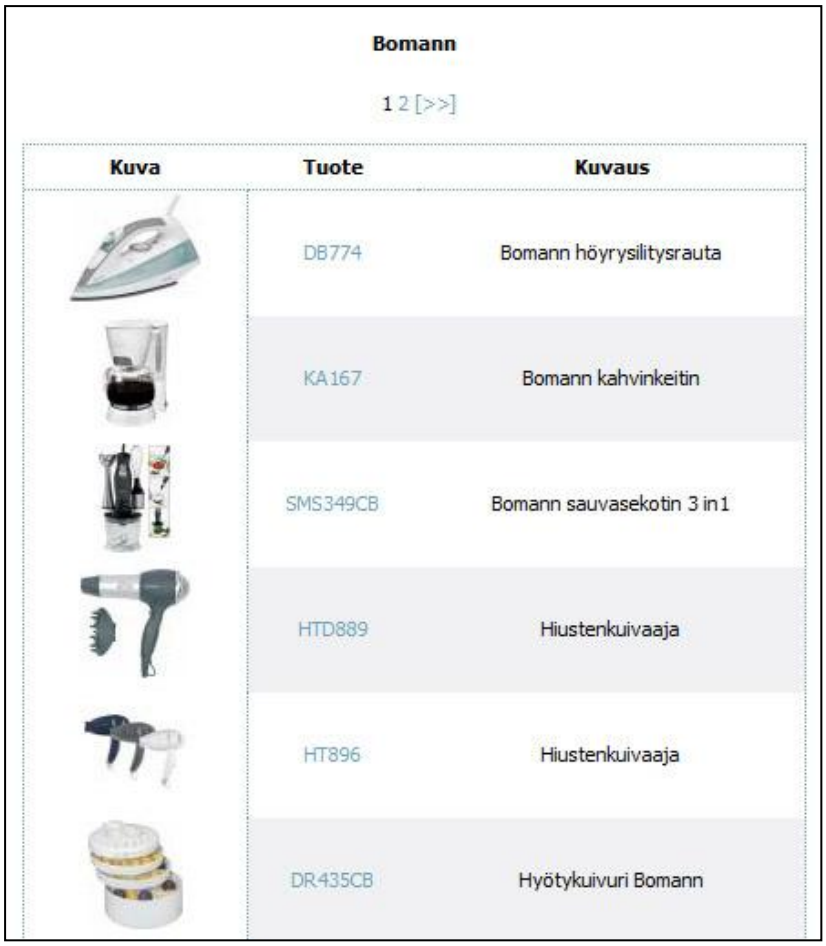
KUVIO 3**.** Näkyville asetetut tuoteryhmät

3.4 Tuotteet

3.4.1 Tuotelista

Tuotelistaan tulostetaan HTML-taulukkona kaikki kategoriaan kuuluvat tuotteet ja lyhyt kuvaus niistä. Tuotteilla tulee olla pieni thumbnail-kuva, jota klikkaamalla pääsee selaamaan tuotteen tarkempia tietoja ja lisäämään se ostoskoriin. Myös tuotekoodin tulee toimia linkkinä tuoteselosteeseen. Samalla kategorialla voi olla useita eri tuotteita, joten tuotteet on käyttömukavuuden vuoksi hyvä jakaa sivuihin.

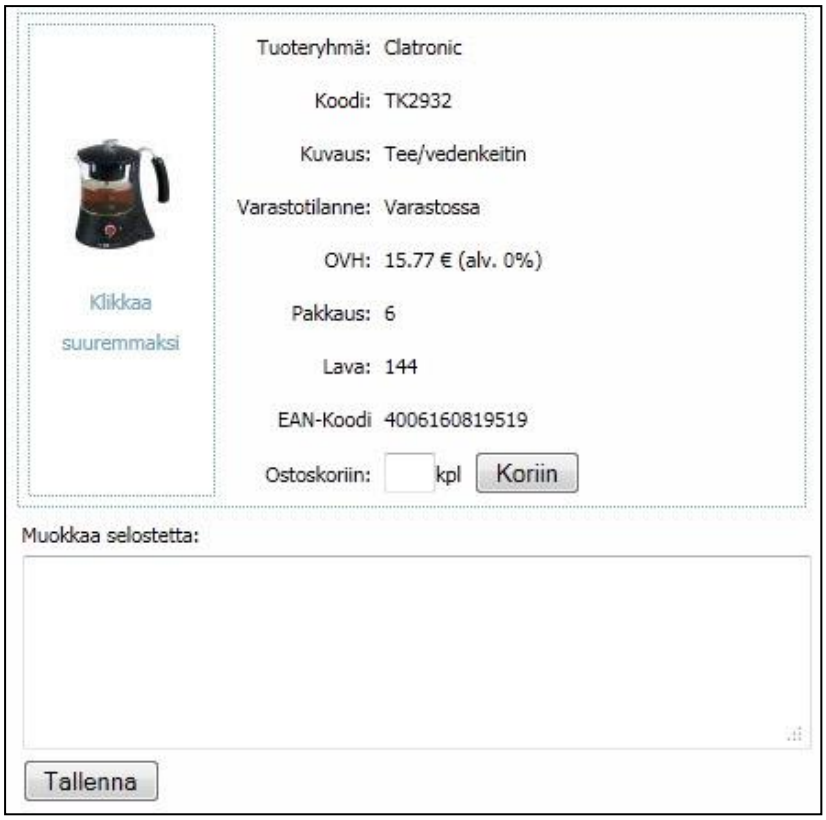
Jottei lista venyisi liian pitkäksi, niin sivua kohden näkyväksi valittiin 20 tuotetta. Sivulla näkyvien tuotteiden lukumäärää ei ole välttämätöntä olla selaajan muutettavissa. Tuotelistalla ei saa myöskään näkyä tuotteita, joita Udda Finland ei halua verkkokaupassa näkyvän, vaikka ne tietokannasta löytyvätkin, sillä ne saattavat olla jo myynnistä poistuneita. Helpoiten tämä olisi hoidettavissa siten, että Uddan henkilökunta lisää oman toiminnanohjausjärjestelmänsä kautta jokaiselle sellaiselle tuotteelle lipun tai parametrin, jonka he eivät halua näkyvän. Toteutettu lopputulos on nähtävillä kuviossa 4.



KUVIO 4. Tuotelista sivunumeroineen

3.4.2 Tuotekuvaus

Koska tuotteen saatavuustietojen kuuluu olla reaaliaikaisia, niin tarkempi tuotekuvaus haetaan suoraan Udda Finlandin varastotietokannasta. Tuotteen tarkemmat tiedot haetaan sivulatauksen yhteydessä ja näistä generoidaan HTML-taulu. Kyseessä oleva tuote saadaan osoitekentän *tuote*-muuttujasta. Jokaiselle yksittäiselle tuotteelle voidaan myös laatia pidempi tuoteseloste, joka tallennetaan Louhen tietokantaan. Tarkemman selosteen lisäämiseen täytyy kirjautua admin-tunnuksin. Toteutettu lopputulos on nähtävillä kuviossa 5.



KUVIO 5**.** Tuotekuvaus jonka seloste on muokattavissa

3.4.3 Tuotteiden kuvat

Uddan toiveesta tuotekuvien siirtämisen heidän serveriltään web-palvelimelle on tapahduttava automaattiseti. Lisäksi kuvaa klikattaessa kuvan tulee suurentua, minkä voi suorittaa jollain valmiilla javascript-functiolla. Kuvien päivittämiseen on olemassa kaksi eri lähestymistapaa, joista ensimmäisessä Louhen serveri hakee Uddan serveriltä kuvat, tai sitten toisessa Uddan serveri itse lähettää määräajoin Louhen serverille kuvat. Koska unix-serverinä Louhella on yksinkertaista suorittaa ajoitettuja skriptejä, niin ratkaisuksi valittiin ensimmäinen vaihtoehto.

Jotta kuvat olisivat noudettavissa Uddan serveriltä, täytyy siellä olla käynnissä vähintän joko HTTP- tai FTP-serveri. FTP-servereille löytyy monia ilmaisia vaihtoehtoja, joiden hallintaan on olemassa graafisia käyttöliittymiä. Graafinen käyttöliittymä tuo enemmän käyttäjäystävällisyyttä, mikäli Udda Finlandin työntekijöiden tarvitsee sen asetuksia joskus muokata. FTP-serverille voidaan asettaa käyttäjätilejä ja estää pääsy ei-toivotuilta vierailta, jolloin se on ideaalinen

käyttötarkoitukseen. Lisäksi käytettäessä FTP-kansiota Uddan henkilöstön ainoaksi tehtäväksi jää raahata halutut tuotekuvat heidän verkkolevyllä sijaitsevaan kansioon, joka on määritetty FTP-kansioksi.

3.5 Ostoskori

Ostoskorit ovat nykyajan verkkokaupoissa vakiinnuttaneet asemansa välttämättömiksi. Ostoskori on käytännössä taulukko, johon asiakas voi lisätä ja poistaa haluamiaan tuotteita aivan kuten tavallisessakin kaupassa asioidessa. Ostoskorista useimmiten näkee yhdellä silmäyksellä, montako kappaletta mitäkin tuotetta on ostamassa ja kuinka paljon tuotteet tulevat yhteensä maksamaan. Udda Finlandin aikaisemmalla sivustolla ei ollut olemassa ostoskoria, jolloin asiakkaat joutuivat itse kirjoittamaan ylös ostoksensa.

Ostoskorin luomiseen on olemassa useita eri lähestymistapoja, joista mainittavimmat ovat evästeet, sessiot ja tietokantataulut. Ostosten tallentaminen evästeisiin ja sessioihin toimivat samalla periaatteella, ja tietokantataulussa voi luoda jokaiselle asiakkaalle omakohtaisen taulun tai käyttää yhtä isoa yhteistä taulua kaikille käyttäjille. Käytettäessä yhteistä tietokantataulua kaikkien asiakkaiden kesken tauluun talennettaisiin jokaiselle riville asiakkaan tunnus, tuotteen koodi ja kappalemäärä. Rivejä tulisi niin paljon kuin asiakkaalla on tuotteja korissa ja ostoskorin sisältö haettaisiin asiakkaan tunnuksen perusteella.

Tietokantoja käytettäessä ostoskorin kanssa on etuna se, että voi ostoskoria on helpompi päivittää ilman sivun uudelleenlatausta. Työssä on valittu käytettäväksi PHP-sessiot, joihin voi tallentaa kaksiulotteisia tauluja. Sessiot ovat turvallinen vaihtoehto, sillä kaikki tieto tallentuu vain serverille, jolloin ei tarvitse olla huolissaan käyttäjän mahdollisista yrityksistä muokata evästeitään. PHP-skriptin suorittaminen on nopeaa ja ostoskorin sisältö säilyy istunnon ajan, jolloin erillisiä skriptejä ei tarvitse luoda tietokantojen siistimiseksi ostoskoreihin unohtuneista tuotteista.

Harmillisesti tämä lähestymistapa silti vaatii sivulatauksen, mutta asiakkaan näkökulmasta se hoituu silti nopeasti. Ostoskorin lopullinen ulkomuoto on näkyvillä kuviossa 6.

KUVIO 6**.** Ostoskori jossa on tuotteita

3.6 Kassa

Kassalla asiakkaalle tulee huomauttaa, mikäli ostosten määrä alittaa rahtivapausrajan joka on 1000 €. Tällöin on sovittava erikseen toimitusehdoista. Tuotteiden hinnat tulee näkyä kassalla pyöristettyinä, mutta kuitenkin laskettu tarkoista arvoista, jottei pyöristys tule heittämään kuin muutamalla sentillä. Ennen tilauksen vahvistusta asiakkaan tulee vielä voida muuttaa rekisteröintivaiheessa tallentamiaan yhteystietoja. Yhteystiedot esitäytetään hakemalla tiedot asiakastietokannasta.

Kun vahvistuslomake on täytetty ja hyväksytty, niin tilausvahvistus lähetetään sähköpostitse sekä asiakkaalle että Udda Finlandille.

3.7 Turvallisuus

3.7.1 Evästeet

Selaajan tietokoneelle tallennettaviin evästeisiin voidaan lisätä monia hyödyllisiä ja usein sivuston kannalta tarpeellisia tietoja, kuten kirjautumistiedon, jolloin palvelu tietää, kuka käyttäjä on kyseessä. Ennen evästeiden käyttöönottoa jokainen vierailu sivustolla oli serverin mielestä täysin uusi tapahtuma. Periaatteessa serveri unohti heti käyttäjän, joka siirtyi sivustolta hetkeksikin pois.

Nykyään evästeitä käytetään lukemattomiin eri tarkoituksiin. Niillä voidaan seurata, minkä linkkien kautta käyttäjä on selautunut eri sivustoille, muokata sivuston ulkoasua ja sisältöä käyttäjän mieltymyksien mukaan ja luoda statistiikkaa. Evästeitä virheellisesti saatetaan usein pitää selaajalle tietoturvariskinä, mutta evästeiden kautta ei voida selvittää mitä käyttäjän tietokone sisältää, eikä niitä voi käyttää viruksina. (Cookie Central 2011.)

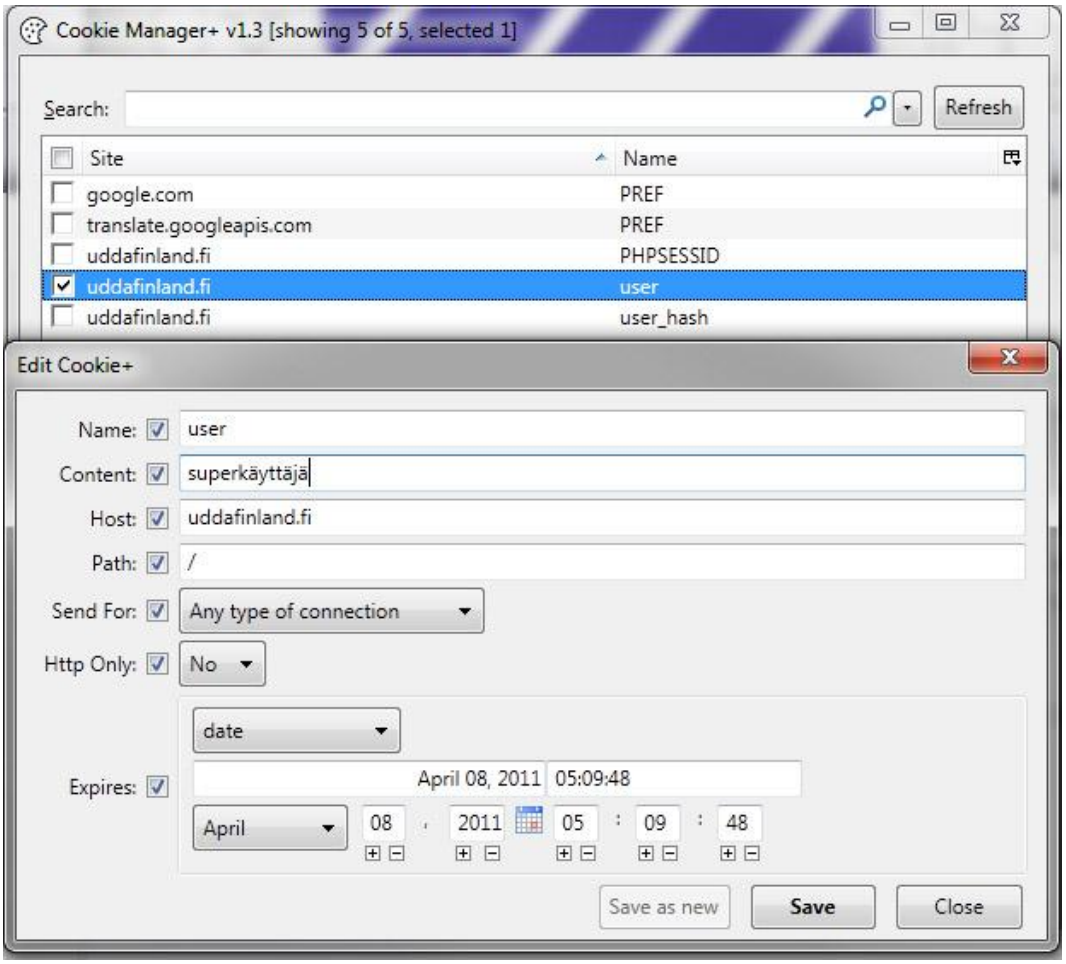
Sen sijaan web-palvelulle voi olla tietoturvariski, jos käyttäjä voi muokata evästeitään niin, että ne vaikka huijaavat serverin kuvittelemaan käyttäjällä olevan ylläpitäjätason tunnukset palveluun (kuvio 7). Tästä syystä evästeisiin tulee tallentaa käyttäjätunnisteen lisäksi myös jotain uniikkia, mitä on likipitäen mahdotonta kokeillen arvata.

Työssä tallennetaan evästeeseen käyttäjätunnisteen lisäksi sama tiivistesumma, joka luodaan rekisteröitymisen yhteydessä. Jokaisen sivulatauksen yhteydessä on aina tarkistettava, että käyttäjätunniste ja tiivistesumma vastaavat tietokannasta löytyviä vastineita. Muussa tapauksessa Evästeet palautetaan oletusasetuksiin, eikä käyttäjä pääse selaamaan sivustoa kirjautuneena tunnuksilla, joita oli koittanut muokata.

Evästeitä on kahta lajia, ja ne jaetaan persistiivisiin, eli pysyviin evästeisiin, ja sessioihin, eli istuntokohtaisiin evästeisiin. Istuntokohtaiset evästeet ovat

olemassa vain sen ajan, kun käyttäjä pysyy kyseistä evästettä käyttävällä sivustolla. Selauduttuaan pois eväste häviää. Persistiivinen eväste tallennetaan käyttäjän tietokoneelle, jossa se säilyy tiettyyn aikarajaan asti, mutta sen voi silti turvallisesti poistaa käyttäjän toimesta. Eväste lähetetään takaisin käyttäjän selaimesta vain sille sivustolle, joka evästeen on luonutkin.

Nykyään on myös kehitteillä evercookieita, eli niin kutsuttuja ikievästeitä, jotka pyrkivät säilymään tietokoneella käyttäjän poistoyrityksistä huolimatta. Ikievästeet tallentavat itsensä useisiin eri kohteisiin tietokoneella, ja mikäli huomaavat yhden niistä olevan kateissa, niin sen tilalle luodaan uusi kopio.



KUVIO 7**.** Evästeitä on mahdollista muokata manuaalisesti

Evästeiden käyttöä säädellään Sähköisen viestinnän tietosuojalain 7.3§:ssa seuraavasti (Finlex 2004):

> *Viestintäverkkojen avulla toteutettu evästeiden tai muiden palvelun käyttöä kuvaavien tietojen tallentaminen käyttäjän päätelaitteelle ja näiden tietojen käyttö on sallittua palvelun tarjoajalle, jos palvelun tarjoaja antaa käyttäjälle ymmärrettävät ja kattavat tiedot tallentamisen tai käytön tarkoituksesta. Samalla palvelun käyttäjälle on annettava mahdollisuus kieltää tässä momentissa tarkoitettu tallentaminen tai käyttö.*
>
> *Edellä 1 momentissa säädetty palvelun tarjoajan tietojen antamisvelvollisuus ja käyttäjän kielto-oikeus ei koske tietojen sellaista tallentamista tai käyttöä, jonka ainoana tarkoituksena on toteuttaa tai helpottaa viestin välittämistä viestintäverkoissa tai joka on välttämätöntä sellaisen palvelun tarjoamiseksi, jota tilaaja tai palvelun käyttäjä on nimenomaisesti pyytänyt.*
>
> *Edellä tässä pykälässä tarkoitettu tallentaminen ja käyttö on sallittua ainoastaan palvelun vaatimassa laajuudessa ja sillä ei saa rajoittaa yksityisyyden suojaa enempää kuin on välttämätöntä.*

3.7.2 SQL-injektiot

SQL-injektio on tekniikka, jolla pyritään syöttämään SQL-tietokantoja käyttävään sovellukseen tietoturva-aukkojen kautta ei-toivottuja kyselyitä. Injektioilla voidaan saada selville esimerkiksi arkaluontoisia käyttäjätietoja tai pahimmassa tapauksessa ottaa hallintaan koko tietokanta ja poistaa sen tauluja. Pääasiassa SQL-injektioilla hyökätään eri web-palveluihin, vaikka eivät hyökkäykset rajoitu pelkästään niihin.

Yleinen tapa käsitellä tietokantaa on käyttää muuttujia, kuten kuviossa 8 on havainnoillistettu.

```php
<?php
$user = "Leo";
$password = "Eekkeri!Maata";

$SQLobject->Query("SELECT age FROM users WHERE user = '$user'
                   AND password = '$password'");
?>
```

KUVIO 8. Tyypillinen SQL-kysely pseudokoodattuna

SQL-injektiossa hyökkääjä pyrkii syöttämään muuttujien avulla kyselyitä, joita ei pitäisi pystyä antamaan. Turvattomasti muodostetun kyselyn ja syötteiden tarkastuksen vuoksi vahingollinen lause saattaa silti päätyä mukaan (PHP 2011c).

Edelliseen esimerkkiin on helppo upottaa SQL-kyselyn kannalta haitallista tietoa sekä $user- että $password-muuttujien tilalle, mikäli niitä kysytään web-sovelluksessa vaikka kirjautumisen yhteydessä. Seuraavassa pseudokoodiesimerkissä on kuvailtu tyypillinen SQL-injektio (kuvio 9).

```php
<?php
$user = "Leo";
$password = "mitävain' OR '1'='1";

$SQLObject->Query("SELECT age FROM users WHERE user = '$user'
                   AND password = '$password'");
?>
```

KUVIO 9. SQL-injektio

Kuvion 9 esimerkissä saadaan kaivettua esiin käyttäjän ikä riippumatta siitä, mikä salasana käyttäjällä on tietokantaan tallennettuna. Tämä johtuu siitä, että tietokantarajapinta käsittelee kyselyn kokonaisuudessaan kuvion 10 mukaisesti, jolloin salasanan voi unohtaa, sillä 1 = 1 on tosi kaikesta huolimatta.

```
SELECT age FROM users WHERE user = 'Leo'
AND password = 'mitävain'
OR '1' = '1'";
```

KUVIO 10. Injektiolla murrettu SQL-kysely

SQL-injektioilta voi suojautua muutamin eri keinoin, joista käytetyimmät ovat tietokantarajapinnan valmistelufunktiot kyselyille ja muuttujien escapettaminen ennen syöttämistä kyselyyn (PHP 2011a). Tietojenkäsittelyssä escapettamisella tarkoitetaan niiden merkkien, joilla muuten olisi erityinen merkitys, muuttamista tavalliseksi tekstiksi lisäämällä niiden eteen escape-merkki. Escapetuksessa käytettävä merkki vaihtelee ohjelmointikielittäin, mutta useimmiten sitä kuvataan kenoviivalla.

Käytettäessä joko olemassa olevia tai itse luotuja tietokantarajapinnan funktioita voidaan varmistua siitä, että käytettävät muuttujat ovat varmasti haluttua tietotyyppiä. Silloin kysely ensin valmistellaan muuttujien kanssa, muuttujat muunnetaan tarvittavaan muotoon ja vasta sen jälkeen ajetaan kysely (kuvio 11).

```
1  PreparedStatement prep = conn.prepareStatement("SELECT age FROM users
2                                                  WHERE user = ?
3                                                  AND password = ?");
4  prep.setString(1, username);
5  prep.setString(2, password);
6  prep.executeQuery();
```

KUVIO 11**.** Parametrisoitu SQL-kysely Javan JDBC APIA käyttäen

Nopeampi, mutta virhealttiimpi tapa välttyä injektioilta on käyttää hyväksi erityismerkkien escapetus SQL:ssa. Tällöin muuttujassa korvataan jokainen erikoismerkki sellaiseksi, millä ei ole mitään erityistä tarkoitusta SQL-kyselyssä. Esimerkiksi käytettäessä MySQL:ää voidaan lisätä jokaisen muuttujassa esiintyvän heittomerkin eteen kenoviiva, joka on MySQL:n escape-merkki, jolloin heittomerkki muuttuu kyselyssä tavalliseksi tekstiksi. Nyt sen sijaan, että kuvion 9 injektio johtaisi kuvion 10 muotoiseen kyselyyn uudella ehdolla, niin tietokanta käsitteleekin salasanan tavallisena tekstinä eikä luo uutta ehtoa (kuvio 12). Tällöin kysely palauttaa käyttäjän tiedot vain siinä tapauksessa, jos tämän salasana todellakin on ”*mitävain’ OR ’1’=’1*”.

```
1   SELECT age FROM users
2   WHERE user = 'Leo'
3   AND password = 'mitävain\' OR \'1\' = \'1'";
```

KUVIO 12**.** Escapettu SQL-injektio

Erikoismerkit vaihtelevat vähän käytettyjen tietokantojen mukaan, mikä on otettava huomioon, mikäli käyttää useampaa eri tietokantaa samaan aikaan ja luo oman funktion muuttujien korjailemiseen.

4 TOTEUTUS

4.1 Hallinnointi

4.1.1 Kategoriat

Koska hallintapaneeleiden ei tarvitse olla kovin monimutkaisia ulkonääöltään tai toiminnaltaan, niin se on päädytty luomaan tavallisena HTML-tauluna, johon luodaan PHP silmukassa niin monta riviä kuin Uddan tietokannasta löytyy kategorioita (kuvio 15). Asetusten tallennuksena on käytetty apuna jQuery-kirjaston ajax-funktioita. Tallennus-nappia painettaessa jQuery serialisoi formin parametrit ja lähettää ne edelleen PHP-skriptille, jonka tehtävänä on tallentaa kaikki tiedot. Haasteena oli selvittää, kuinka PHP:llä pystyy vastaanottamaan joka kerta vaihteleva määrä muuttujia, joissa kerrotaan mahdollisesti kategorian uusi nimi ja onko se näkyvillä ja asetetaan nämä tiedot omille tunnistenumeroilleen tietokannassa.

Ajax-lähetyksessä lähetetään siis kaksi muuttujaa per kategoria. Kategorian nimi on oletuksena alkuperäinen ja tyhjää ei voi lähettää. Muuttujan nimi on kategorian tunnistusnumero, ja sen arvona lähetetään alkuperäinen tai uusi nimi. Kategorian näkyvyys lähetetään valintaruutu-arvoina, jolloin muuttujan nimenä käytäänkin taulua *v[]* ja siihen lisätään vain niiden kategorioiden tunnistenumerot, joiden halutaan näkyvän. POST-lähetykseen sisältyy vain ne valintaruudut, jotka ovat aktiivisia.

POST-kutsun vastaanottava skripti käy ensiksi foreach-silmukalla läpi kaikki vastaanotetut POST-muuttujat ja tallentaa ne uuteen tauluun. Aluksi skripti tarkistaa, onko kategoriaa jo ennestään Louhen tietokannassa. Mikäli kategoria löytyy, tyydytään siitä muuttamaan vain sen nimi, muussa tapauksessa tauluun luodaan uusi kategoria asetetulla nimellä. Tämän jälkeen skripti käy läpi kaikki POST:sta saadut *v*-taulun alkiot silmukan sisällä. Koska jokaisen alkion sisältönä on halutun tuotteen tunnistenumero, Louhen taulusta etsitään samalla tunnistenumerolla löytyvät kategoriat ja päivitetään niiden näkyvyyskenttä.

Kun uudet tiedot on onnistuneesti tallennettu, päivitetään hallinointisivu käyttämällä javascriptin location.reload(true)-komentoa, jolloin uudet arvot tulevat välittömästi näkyviin. Jottei käyttäjän tarvitsisi itse muistaa, mitkä kategoriat on aikaisemmin valittu näytettäviksi, niin PHP:n avulla nämä rivit voidaan esivalita yksinkertaisella ehtolauseella. Jotta taulukosta saataisiin visuaalisti selkeämmän näköinen ja esivalintaa korostaakseen, on jokaiselle riville päätetty lisätä taustaväri, joka vaihtuu sen mukaan, onko kategoria näkyvillä vai ei. Väri vaihdetaan käyttämällä PHP:n ehtolauseketta, joka vaihtaa silmukassa olevan taulukkorivin css-taustaväriä sen mukaan onko *$checked* muuttujan arvona mitään. (Kuvio 13, rivi 13)

Väreinä on käytetty silmää miellyttäviä, ei liian räikeitä web-turvallisia värejä. Heksanumeroiden sijaan on käytetty X11-värinimiä niiden paremman ymmärrettävyyden vuoksi. X11-värinimet toimivat myös useimmilla selaimilla (W3C 2011).

```
1  <?php
2  $result = mssql_query("SELECT DISTINCT koodi, kategoria
3                         FROM tuoteryhma ORDER BY kategoria");
4  while($row = mssql_fetch_array($result))
5  {
6    $koodi = mysql_real_escape_string($row['koodi']);
7    $result2 = mysql_query("SELECT nakyvilla, varanimi
8                            FROM categories WHERE koodi = '$koodi'");
9    $row2 = mysql_fetch_row($result2);
10
11   $checked = ($row2[0] == true) ? "CHECKED" : "";
12
13   echo "<tr style='background: ".(!empty($checked) == 1 ? 'LightGreen' : 'Salmon' )."'>\n";
14   echo "<td>".$row['koodi']."</td>";
15   echo "<td>".$row['kategoria']."</td>";
16   echo "<td><input name='".$row['koodi']."' value='$nimi' type='text' /></td>";
17   echo "<td><input name='v[]' value='".$row['koodi']."' type='checkbox' $checked/></td>";
18   echo "</tr>\n";
19 }
20 ?>
```

KUVIO 13**.** HTML-taulukon rivien luominen PHP-silmukassa

4.1.2 Asiakastilit

Asiakastilien hallintapaneeli on toteutettu lähes samaa ratkaisua käyttäen kuin kategorioidenkin hallintapaneeli. Erona on, että nyt jokaisen taulukon rivi sisältää oman HTML-formin ja oman tallennuspainikkeen sen sijaan, että käytettäisiin yhtä yhteistä tallennuspainiketta kaikille tileille.

Jokaisella formilla on tässä sama luokka, johon on liitetty jQuery-funktio. Erottaakseen, mitä formia on klikattu, voi jQueryssä käyttää muuttujaa $(this), jolloin funktio jättää tallentamatta muiden formien tiedot, vaikka näillä olisikin sama luokka.

Tallennettaessa jQuery lähettää AJAX:n avulla rivin tiedot PHP-skriptille, joka niin ikään tallentaa tiedot Louhen MySQL-tietokantaan ja tiedottaa onnistumisestaan json-olion sisällöllä. Tietojen päivittyessä pakotetaan sivun päivitys, jolloin uudet arvot ja värit ovat jälleen välittömästi näkyvillä. Päivityksen onnistumisesta kerrotaan käyttäjälle käyttäen javascriptin *alert()*-funktiota.

Tilien aktiivisuus on valittavissa HTML:n alasvetovalikolla. Tilin aktiivisuus on esivalittu näyttämään tilaa, joka tilillä on tietokannassa käyttäen samaa ehtolauseketta kuin kategorioiden näkyvyydyssäkin, mutta sovellettuna valintaruutujen sijaan <option>:iin (kuvio 14).

```php
<?php
echo "<select name='tila'>\n";
echo "<option value='1' ".($aktivoitu == 1 ? 'selected' : '' ).">Aktiivinen</option>\n";
echo "<option value='0' ".($aktivoitu == 0 ? 'selected' : '' ).">Suljettu</option>\n";
echo "</select>\n";
?>
```

KUVIO 14**.** Alasvetovalikon dynaaminen esivalinta

4.2 Rekisteröityminen

Rekisteröityminen suoritetaan tavallisella HTML-formilla, joka lähettää arvot PHP-tiedostolle. PHP-tiedosto tarkistaa ehtolausein, että vaaditut kentät eivät olleet tyhjiä, salasana on riittävän pitkä, salasana on kirjoitettu kahdesti samalla tavalla ja vielä sähköpostin oikeamuotoisuuden käyttämällä säännöllisiä lausekkeita, eli regular expressionilla (lyhyesti regexp). Säännöllisillä lausekkeilla voidaan tarkistaa, esiintyykö syötteessä jokin tietty merkki, sana, lause tai noudataako se tiettyä kaavaa.

Mikäli kaikki ovat kohdillaan, niin sähköpostiosoitteesta, annetutusta salasanasta ja kolmesta eri suolasta muodostetaan sha1-tiivistesumma. Yksi suola on dynaaminen, joka on käyttäjän rivinumero tietokantataulussa. Tiivistettä ei siis voi luoda ennen kuin rivinumero on selvillä, joten tunnus tallennetaan ensin tietokantaan satunnaisella tiivistesummalla, josta otetaan talteen paluuarvona rivinumero. Tämän jälkeen rivi päivitetään varsinaisella tiivistesummalla käyttäen saatua rivinumeroa.

4.3 Tuotekategoriat

Kaikki sivustolla esiintyvät tuotekategoriat noudetaan Louhen tietokannasta yksinkertaisella SQL-lauseella. Tällöin valitaan kaikki ne rivit, joilla on sarakkeessa ”nakyvilla” boolean arvo TRUE. Tulokseksi saadun taulun sisältö on sen jälkeen helppo tulostaa PHP:n while-silmukassa. Silmukassa sen kertainen kategorian nimi ja kategorian tunniste upotetaan HTML-koodiin järjestelemättömän listan <UL> ja linkin <LI> sisälle.

4.4 Tuotteet

4.4.1 Tuotelista

Tuotteet haetaan samalla tavalla kuin kategoriatkin ja upotetaan silmukassa HTML-tauluun, mutta tuotteet haetaan suoraan Udda Finlandin tietokannasta. Ttuotteilla on useita sarakkeita, jotka eivät ole missään käytössä, joten Udda Finlandin kanssa sovittiin, että ne tuotteet, joiden piirustusnumero-kentässä on luku 1, filtteröidään pois verkkokaupalla näkyvistä tuotteista.

Tuotteita saattaa olla joissakin kategorioissa paljon, joten kyselyn yhteydessä haetaan vain siivu oikeista osumista, eli kyselyt sivutetaan. Tuotteiden määräksi on valittu 20 kappaletta sivua kohden. Jotta saataisiin selville sivujen maksimimäärän, niin täytyy ensin hakea kaikki mahdolliset tulokset ja jakaa se kahdellakymmenellä. Tulokseen saattaa jäädä jakojäännös, joten tulos kasvatetaan seuraavaan kokonaislukuun PHP:n *ceil()*-funktiolla.

Tämän jälkeen tehdään uusi SQL-kysely, jolla haetaan vain ne tuotteet, jotka alkavat tietyltä etäisyydeltä. Etäisyys lasketaan kyseessä olevan sivunumeron avulla ja kysely katkaistaan 20 tuotteen jälkeen. Toisin kuin MySQL:ssä, ei MS SQL:ssä ole olemassa LIMIT-määrettä, jolla sivutus onnistuisi kivuttomasti, mutta sitä on mahdollista matkia käyttämällä sisäkkäisiä SELECT-käskyjä. Alkuperäisessä SELECTISSÄ voidaan valita kaikki, mitkä seuraava SELECT palauttaa laskevassa järjestyksessä. Näistä haetaan TOP 20 kolmannen SELECTIN tuloksista, jossa haetaan TOP <aloitusetäisyys> nousevassa järjestyksessä. Kuviossa 15 havainnoillistaan, kuinka MySQL:ssä ja MS SQL:ssä toteutetaan sivutus.

```
1  /* MySQL */
2  SELECT * FROM table LIMIT 10, 20;
3
4  /* MS SQL */
5  SELECT * FROM (SELECT TOP 20 value FROM (
6                 SELECT TOP 10 value FROM table
7                 ORDER BY value ASC) AS tmp_table
8                 ORDER BY value DESC) AS tmp_table
9                 ORDER BY value ASC;
```

KUVIO 15. Sivutus alken riviltä 10, valiten seuraavat 20

4.4.2 Tuoteseloste

Tuotteen tarkemmat tiedot haetaan suoraan Udda Finlandin tietokannasta ja lisätään HTML-tauluun. Koska tuotteille ei ole olemassa lyhyttä kuvausta pidempää tuoteselostetta, niin ylläpitäjän oikeuksilla tuotteille voi tallentaa selosteen Louhen tietokantaan.

Tiedot tallennetaan käyttäen jQueryn AJAX-funktioita, jolloin ylimääräisille sivulatauksille ei ole tarvetta. PHP:lla on annettu ehto, jolla estetään tallennus-formin tulostuminen, ellei evästeissä ole ylläpitäjän tunnus ja tälle oikea tiivistesumma.

4.4.3 Tuotekuvat

Tuotteiden kuvat haetaan luomalla skriptin kautta FTP-yhteys Louhen serveriltä Uddan serverille. Koska kuvien päivityksen ei tarvitse olla reaaliaikaista, niin riittää, että päivitys tapahtuu vaikka keskiyöllä, kun muu mahdollinen ruuhka on pienimmillään. Tämä onnistuu helposti unixin ajastetuilla tehtävillä, joita kutsutaan cronjobeiksi.

Cron on unix-pohjaisten käyttöjärjestelmien ajastinpalvelu. Cronille voi antaa suoritettavia tehtäviä crontabin kautta. Jokainen crontabissa määritelty rivi on

erillinen työ, jolle on asetettu suoritusaika. Cron-daemon tarkistaa minuutin välein crontabin sisällön ja käynnistää suoritettavat käskyt määrättyyn aikaan, elleivät ne ole jo suorituksessa. Jokaiselle saman käyttöjärjestelmän käyttäjälle on mahdollista lisätä oma crontab-tiedosto. Crontab on avattavissa komennolla ”*crontab -e*”. (Vixie 2011.)

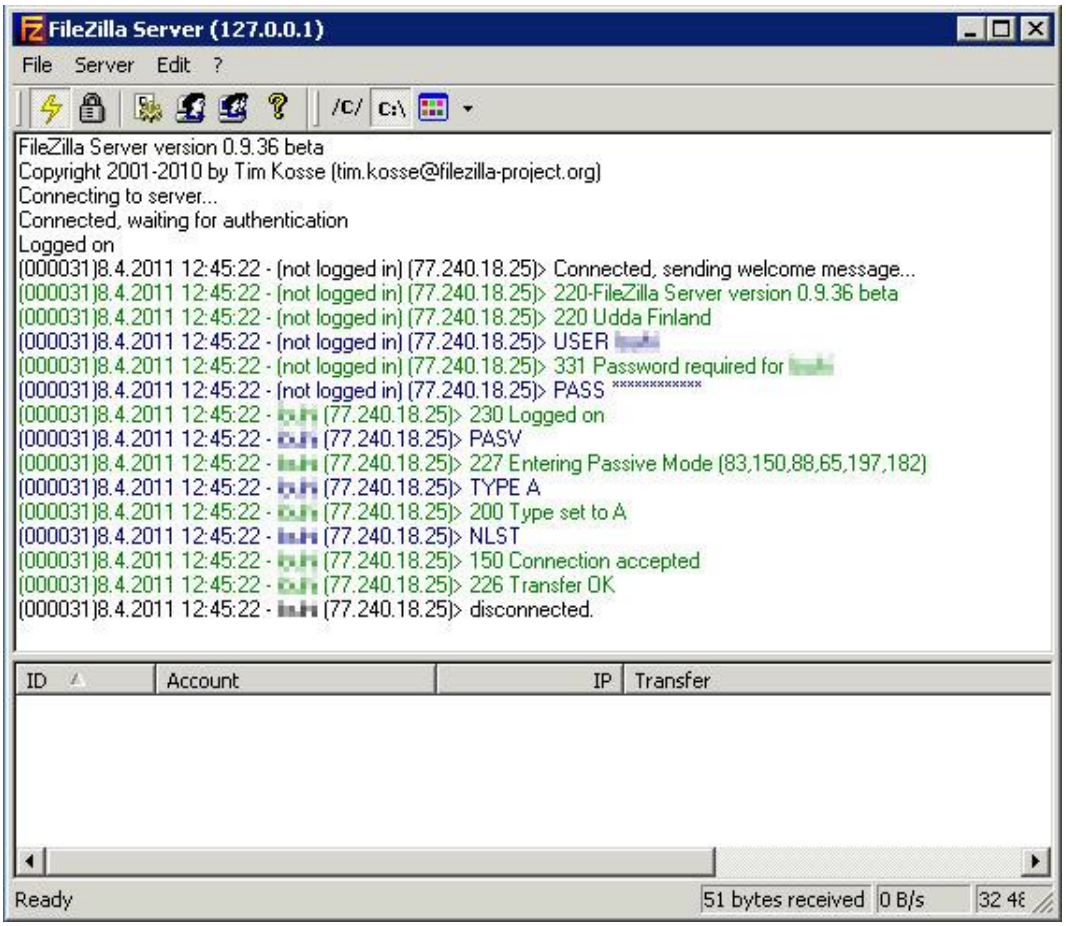
Kuvion 16 mukaisesti crontab rivien syntaksissa annetaan ensin välilyönnein eroteltuna 5 numeroa jotka edustavat suorituksen alkamisen ajankohtaa. Tämän jälkeen seuraa käyttäjän tunnus ja tarkka polku suoritettavaan ohjelmaan.

KUVIO 16**.** Esimerkki cronitetusta tehtävästä

Jotta FTP-yhteys voitaisiin luoda, on Uddan serverillä ensiksi oltava FTP-serveri käynnissä. Työssä käytettäväksi ohjelmaksi on valittu FileZilla Server, joka on ilmainen ja avoimen lähdekoodin FTP-serveriohjelma (FileZilla 2011).

FileZillan käyttöönotto on yksinkertaista. Asennuksen jälkeen luodaan käyttäjille tunnus ja salasana ja sidotaan nämä omiin hakemistoihinsa, joiden sisältöä he voivat selata. Tunnuksille voi asettaa rajoituksia sen mukaan, haluaako käyttäjille olevan mahdollista selata hakemistoa, lukea tiedostoja tai jopa muokata hakemiston sisältöä. Turvallisuuden vuoksi käytetään Louhelle annettavissa tunnuksissa vain tiedostojen lukuoikeutta. FileZilla on asetettu käynnistymään ja asettamaan FTP-palvelu päälle tietokoneen käynnistyksen yhteydessä. Kuviossa 17 on nähtävissä FileZillan käyttöliittymä.

Vaikka ei-toivottuja selaajia voi torjua jo reitittimen palomuurisäänöissä, niin on silti hyvä vielä itse FileZillassa varmistaa, että tunnukselle sallitaan pääsy vain tietystä IP-osoitteesta. Näin vältytään ylimääräisiltä murtautumisyrityksiltä, jotka ovat hyvin yleisiä verkossa julkisella IP:llä olevilla palveluilla.



KUVIO 17**.** FileZilla hyväksyy kirjautumisen

Kun FTP-yhteys on muodostettu, niin PHP-skripti käy läpi kaikki tuotekuvat, jotka ovat verkkokaupassa jo ennestään, minkä jälkeen skripti vertailee, onko FTP-kansiossa mitään uusia kuvia ladattavaksi. Mikäli uusia kuvia on löydetty, niin silmukan avulla ladataan jokainen tiedosto. Udda Finlandin henkilöstön kanssa on sovittu, että kuvat saisivat korkeimmillaan olla noin 4000x4000 pikselin suuruisia, koska Louhen serveri on rajoittanut käytettävää muistia 64 Mt:uun.

Varmuuden vuoksi skripti silti karsii ladattavien kuvien listasta pois kuvat, jotka saattaisivat varata lisää muistia ja siten käsittelyskriptin tahattoman katkeamisen kesken siirron. Näin käydessä epäonnistuminen kirjataan logiin.

Skriptissä kuvien muokkaamiseen käytetään GD2-kirjastoa, koska se on jo asennettu Louhen webhotelliin. Skripti muuttaa pakatut kuvat ensiksi pakkaamattomaan bitmap-muotoon, ennen kuin se pystyy käsittelemään niitä, minkä vuoksi se saattaa syödä melkoisen osan rajatusta muistista, varsinkin kun useimmat kuvat ovat värisyyvydeltään 24-bittisiä. Varattavan muistin määrän kilotavuina voidaan laskea kaavalla $\frac{X*Y*B}{8*1024}$, jossa X tarkoittaa kuvan leveyttä, Y korkeutta ja B värisyvyyttä (Bourke 2011). Bitmap-kuvien eri tiedostokokoja on kuvattu taulukossa 1.

TAULUKKO 1. Eri bitmap-kuvien koko kilotavuina

Resoluutio	Värisyvyys	Varattava muisti
640 x 480	8 bittiä	300 Kt
	16 bittiä	600 Kt
	24 bittiä	1 200 Kt
2048 x 1024	8 bittiä	2 048 Kt
	16 bittiä	4 096 Kt
	24 bittiä	6 144 Kt
4096 x 4096	8 bittiä	16 384 Kt
	16 bittiä	32 768 Kt
	24 bittiä	49 152 Kt

Jokaisella silmukan kierroksella ladatusta kuvasta luodaan ensiksi pienempi, korkeintaan 640 pikselin korkuinen kuva säilyttäen alkuperäisen kuvasuhteen ja tästä vielä erikseen 60 pikseliä korkea thumbnail-kuva. Kuvanmuokkauksen jälkeen kuvat siirretään omiin kansioihinsa Louhen serverillä ja ladattu väliaikainen kuva tuhotaan levytilaa varaamasta.

Kuvien siirto on ajoitettu tapahtuvan keskiyöksi, sillä Uddan internetyhteyden lähetyskaista on rajoitettu 1 Mbit/s:iin ja työaikoihin tapahtuva siirto voisi tuhkahduttaa muun internetliikenteen. Kun kaikki halutut kuvat on siirretty, niin yhteys suljetaan.

4.5 Ostoskori

Kun asiakas haluaa valita tuotteita ostettavaksi, on näitä valintoja varten luotu PHP:n ja sessioevästeiden avulla ostoskori, jonne kaikki sen kertaiset tuotteet voidaan tallentaa. Tuotteita ostoskoriin siirrettäessä lisätään evästeissä sijaitsevaan kaksiulotteiseen tauluun avaimeksi tuotetunniste ja sen arvoksi kappalemäärä. Tuotteiden hinnat haetaan SQL-kutsulla Uddan tietokannasta ja kertolaskujen vuoksi pyöristetään normaalisti PHP-funktiolla *round($luku, $desimaalitarkkuus)* (PHP 2011b).

Tuotteen lisäys ostoskoriin tapahtuu, kun käyttäjä on klikannut linkkiä, joka lataa sivun uudestaan ja luo osoiteriville uuden muuttujan ”*koriin*”, jonka arvona on joko artikkeleiden kappalemäärä kokonaislukuna tai tekstit ”*poista*” ja ”*tyhjennä*”, mikäli asiakas haluaa poistaa ostoskorista tuotteita tai tyhjentää korin kokonaan. Jokaiselle muuttujalle on oma ehtolohko. Jos muuttujan arvo on numeerinen, niin tiedetään, että asiakas haluaa lisätä korin tuotteita. Tällöin session kaksiulotteiseen tauluun voidaan yksinkertaisesti lisätä uusi tuotekoodi avaimena, jonka alkioksi annetaan kappalemäärä. (Kuvio 18, rivi 6.)

Mikäli muuttujan arvo on ”*poista*”, niin käytössä on omintakeinen ratkaisu käydä foreach-silmukassa vaihe vaiheelta kopioimassa jokainen ostoskoritaulun alkio väliaikaiseen tauluun, kunhan kyseessä ei ole poistettava tuote. Väliaikaiseen tauluun jää silloin jäljelle kaikki muut tuotteet, paitsi poistettava. Tämän jälkeen ostoskoritaulu voidaan ylikirjoittaa väliaikaisen taulukon sisällöllä.
(Kuvio 18, rivi 12.) Kun muuttujan arvona on ”*tyhjennä*” ja asiakas haluaa tyhjentää koko ostoskorin, niin koko taulu yksinkertaisesti poistetaan PHP:n *unset()*-funktiolla ja luodaan tyhjä taulu tilalle (Kuvio 18, rivi 24).

```php
<?php
if(isset($_GET['koriin']))
{
  $kpl = $_GET['koriin'];
  unset($_SESSION['ostoskori_tmp']);
  if (is_numeric($kpl))
  {
    $tuote = $_GET['tuote'];
    $_SESSION['ostoskori'][$tuote] = $kpl;
    header("Location: index.php?tuote=".$tuote."");
  }
  else if ($kpl == 'poista')
  {
    $id = $_GET['id'];
    foreach($_SESSION['ostoskori'] as $tuote => $kpl)
    {
      if ($tuote != $id)
      $_SESSION['ostoskori_tmp'][$tuote] = $kpl;
    }
    unset($_SESSION['ostoskori']);
    $_SESSION['ostoskori'] = $_SESSION['ostoskori_tmp'];
    header("Location: index.php?tuote=".$id."");
  }
  else if ($kpl == 'tyhjenna')
  {
    unset($_SESSION['ostoskori']);
  }
}
?>
```

KUVIO 18. Ostoskorin muokkauksen eri vaiheet

4.6 Kassa

Kassa toimii periaatteessa kuin laajempana versiona ostoskorista. Kassan sisältö tulostetaan samalla tavalla ostoskorisession tauluista. Tilaushetkellä ostoskorin sisältöä voi vielä muokata HTML-formilla, joka on kohdistettu sivuun itseensä. Formi lähettää POSTina ostoskorin sisällön ja sivulatauksen yhteydessä PHP päivittää ostoskorin sisällön ennen sen tulostamista sivulle.

Tilausvahvistuksesta lähetetään sähköpostimuistutus sekä asiakkaalle että Udda Finlandin henkilökunnalle. Sähköpostin lähetyksessä käytetään hyväksi PEAR:sta löytyvää Mail_Mime-laajennusta PHP:lle (PEAR 2011). Mail_Mime tarjoaa luokat, joilla voi luoda ja käsitellä MIME-viestejä ja lähettää sähköpostia, joka sisältää tekstiä, HTML:ää, HTML-kuvia ja liitetiedostoja.

Mikäli PEAR on jo asennettu unix-serverille, niin Mail_Mimen asentaminen tapahtuu konsolikomennolla ”pear install Mail_Mime”. Tämän jälkeen laajennus on käytettävissä, kun sen ensin sisällyttää lisäämällä rivin *include(Mail/mime.php);* PHP-tiedoston alkuun. Tämän jälkeen laajennuksesta voi luoda uuden olion, jonka sisälle tallennetaan stringinä viestin sisältö ja tauluna käytettävät headerit. Lopulta viesti lähetetään Mail-luokan *send*-metodilla. Onnistuessaan mail palauttaa TRUEN ja epäonnistuessa FALSEN. (Kuvio 19)

Kun sähköposti on onnistuneesti lähetetty, niin käyttäjä siirretään sivulle, jolla kiitetään tilauksesta.

```php
<?php
include('Mail.php');
include('Mail/mime.php');

$text     = 'Tavallista tekstiä';
$html     = '<html><body>HTML-tekstiä</body></html>';
$file     = '/home/leo/liitetiedosto.txt';
$crlf     = "\n";
$headers  = array(
              'From'    => 'leo@esimerkki.fi',
              'Subject' => 'Mime esimerkki'
              );

$mime = new Mail_mime($crlf);

$mime->setTXTBody($text);
$mime->setHTMLBody($html);
$mime->addAttachment($file, 'text/plain');

$body     = $mime->get();
$headers  = $mime->headers($headers);

$mail =& Mail::factory('mail');
$mail->send('matti@esimerkki.fi', $headers, $body);
?>
```

KUVIO 19**.** Mime-esimerkki

4.7 Evästeet

PHP:n avulla on yksinkertaista ja nopeaa hallita evästeitä. Koska evästeet lähetetään HTTP-otsikkotietojen mukana, täytyy evästeet luoda ennen kuin PHP luo mitään muita tulosteita. Tämä johtuu HTTP-protokollan rajoitteista. Siksi evästeet on hyvä luoda jo PHP-tiedoston ensimmäisillä riveillä.

Persistiivinen eväste luodaan PHP:n funktiolla *setcookie()*. Funktiolle voi antaa 7 parametria, mutta usein vain kolmea ensimmäistä tarvitaan. Kolmessa ensimmäisessä parametrissa määritellään järjestyksessä evästeen nimi, sen arvo ja ajankohta, jolloin eväste tuhoutuu. Persistiivisen evästeen sisältöön pääsee käsiksi globaalin *$_COOKIE*-taulun kautta. (Kuvio 20.)

```php
<?php
// Jos evästettä ei ole olemassa, niin luodaan
// uusi sisältäen tiedon "Mörökölli" joka säilyy tunnin.
if (!isset($_COOKIE["Pipari"]))
  setcookie("Pipari", "Mörökölli", time()+3600);

// Tulostetaan evästeen sisältö.
echo $_COOKIE["Pipari"];

// Tuhotaan eväste asettamalla aika menneisyyteen.
setcookie("Pipari", "", time()-3600);
?>
```

KUVIO 20**.** Persistiivisen evästeen käyttö

Istuntokohtaisia evästeitä luodessa istunto täytyy ensin luoda. Istunnolle on myös mahdollista antaa lyhyt, mutta kuvaava nimi yleensä niitä käyttäjiä varten, joilla on evästevaroitukset käytössä. Istunnon luomisen jälkeen istuntoevästeitä voidaan hallita suoraan globaalin *$_SESSION*-taulun avulla. (Kuvio 21.)

```php
<?php
// Nimetyn istunnon käynnistys.
session_name("nimettySessio");
session_start();

// Jos alkio "sivulataukset" jo olemassa,
// niin kasvatetaan sen arvoa. Muutoin luodaan alkio.
if(isset($_SESSION['sivulataukset']))
  $_SESSION['sivulataukset'] += 1;
else
  $_SESSION['sivulataukset'] = 1;

// Alkion arvon tulostus.
echo "Sivulataukset = ". $_SESSION['sivulataukset'];

// Istunnon tuhoaminen.
session_destroy();
?>
```

KUVIO 21**.** Nimetyn istunnon avulla lasketut sivulataukset

4.8 SQL-injektiot

Työssä on päädytty torjumaan SQL-injektiot käyttäen hyväksi escape-merkkejä. Tällöin injektiossa esiintyvät erikoismerkit saadaan käännettyä harmittomaksi tekstiksi. PHP:n funktioiden joukosta löytyy valmis funktio, mikä tekee juuri tämän MySQL-kyselyiden kanssa, joka helpottaa huomattavasti toteuttamista. Kyseinen funktio on *mysql_real_escape_string($string)* ja onnistuessaan se palauttaa escapetun stringin, epäonnistuessaan falsen (PHP 2011a).

Työssä käytetään myös Micrsoftin SQL-tietokantaa, jossa on käytössä eri escape-merkki kuin MySQL:ssä. PHP:stä ei löydy valmista käyttökelpoista funktioita tähän konversioon, joten täytyy luoda oma funktio, joka käytännössä imitoi MySQL:n vastaavaa funktiota.

Funkion luominen on käytännössä yksinkertaista. Tarvitsee vain selvittää, mitkä ovat MS SQL:ssä käytetyt erikoismerkit, ja asettaa niiden eteen MS SQL:n escapemerkki, joka tavallisesta käytännöstä poiketen on heittomerkki.

Mikäli funktioon syötetty lause on numeerinen tai tyhjä, niin muutosta ei tarvitse tehdä, sillä nämä ovat jo turvallisia lisukkeita syötteissä. (Kuvio 22)

```php
<?php
function ms_escape_string($data)
{
    if ( !isset($data) or empty($data) ) return '';
    if ( is_numeric($data) ) return $data;

    $non_displayables = array(
        '/%0[0-8bcef]/',    // url encoded 00-08, 11, 12, 14, 15
        '/%1[0-9a-f]/',     // url encoded 16-31
        '/[\x00-\x08]/',    // 00-08
        '/\x0b/',           // 11
        '/\x0c/',           // 12
        '/[\x0e-\x1f]/'     // 14-31
    );
    foreach ( $non_displayables as $regex )
        $data = preg_replace( $regex, '', $data );
    $data = str_replace("'", "''", $data );
    return $data;
}
?>
```

KUVIO 22**.** Microsft SQL:lle luotu alustusfunktio

Vaikka syötteet onkin ennalta alustettu, niin aina kannattaa varautua mahdollisiin virheisiin. Kaikkia mahdollisia vastakeinoja ei välttämättä tule aina ajateltua, joten turvallisuuden vuoksi on parasta rajoittaa myös itse tietokantaan kirjauduttavilla tunnuksien oikeuksiin.

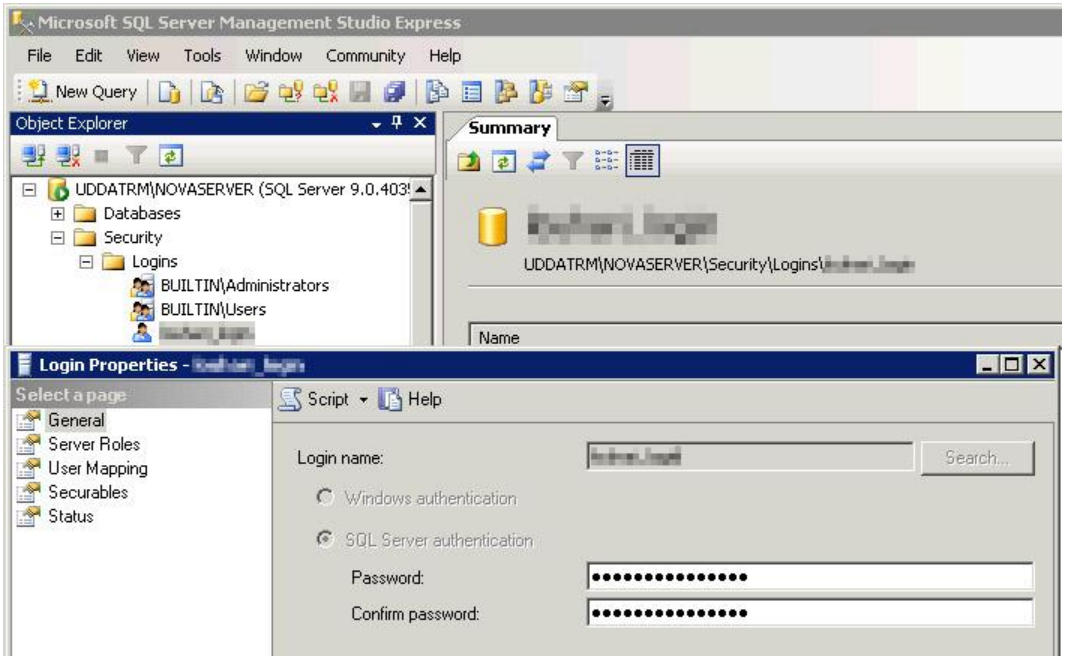
Mikäli yhteyden luonnissa käytettäisiin MS SQL:n oletustunnusta, jolla on ylläpitäjän käyttöoikeudet, niin hyökkääjä saattaisi hyvinkin luoda uusia käyttäjätiliä tai vaikka tuhota koko tietokannan. Verkkokaupan ei tarvitse asioidessaan muuta kuin lukuoikeuden tietokantaan, joten vaikka jos SQL-lauseiden sekaan saataisiinkin ujutettua vahingollinen syöte, niin siitä ei koituisi mitään lopullista vahinkoa tietojen katoamisen muodossa.

MS SQL -tietokantoihin on helppo lisätä uusia käyttäjä Microsoftin oman hallinnointityökalun, Microsoft SQL Server Management Expressin avulla (Microsoft MSDN 2011).

Ensiksi ohjelman käynnistyttyä avataan ”Security”-kansio, jonka alta löytyy ”Logins”-kansio, jonne kaikki kirjautumistunnukset on määritelty. Jokainen käyttäjä täytyy olla sidottu yhteen login-tunnukseen. Login-tunnuksella voi sitoa käyttäjät tiettyyn tietokantaan, jonka ulkopuolelle he eivät pääse selautumaan. (Kuvio 23.)

Tämän jälkeen luodaan vielä erikseen käyttäjä, joka sidotaan kyseiseen loginiin. Käyttäjä luodaan tietokannoittain, eli MSSMSE:ä käyttäen selaudutaan hakemistoon *Databases/Tietokanta/Security/Users* ja luodaan uusi käyttäjä. Tässä käyttäjälle voidaan tarkintaa eri käyttöroolit, eli tässä tapauksessa käytetään roolia, jolla on vain lukuoikeus.

Nyt kun Login ja käyttäjä on luotu rooleineen, niin tietokantaan voi yhdistää normaalisti PHP:n avulla juuri luodulla loginilla tarvitsematta pelätä, että sitä käytettäisiin väärin.



KUVIO 23**.** Login-tunnuksen luonti

5 YHTEENVETO

Opinnäytetyön tavoitteena oli luoda verkkokauppa, jonka tuotetiedot näkyisivät käyttäjälle reaaliaikaisena suoraan varastotietokannasta, ja tähän tavoitteeseen myös päästiin. Jokainen toteutettu komponentti sivustolla toimii sulavasti toistensa kesken mahdollistaen miellyttävän käyttäjäkokemuksen.

Udda Finland on tyytyväinen lopputulokseen, vaikkakin sivuston ulkoasua saatetaan vielä myöhemmin muutella muun näköiseksi, sillä työssä on keskitytty enemmän sivuston toimivuuden luomiseen kuin graafiseen ilmeeseen. Verkkokauppaa käytetään myös työkaluna Udda Finlandin Virossa toimivassa varastossa, sillä pienin muutoksin siitä voi nopeasti tarkistaa nykyisten ja tulevien tuotteiden saldomäärät.

Kokonaisuudessaan työ kattaa kaiken sen, mitä Udda Finland on siltä vaatinut julkaisua varten. Myöhemmin voi olla mahdollista, että sivustoa jatkokehitetään sen verran, että sitä voisi olla mahdollisuus muokata selaimen kautta.

Tätä verkkokauppaa luodessa tuli vastaan lukuisia eri tekniikoita sivuston interaktiivisuuden kehittämiseksi, joita voisi olla mielenkiintoista sulauttaa joko tähän tai tulevaisuudessa muihin web-projekteihin.

LÄHTEET

Bourke P. 2011. Beginners Guide To Bitmaps [viitattu 17.4.2011]. Saatavissa: http://paulbourke.net/dataformats/bitmaps/

Cookie Central. 2011. What are cookies? WWW-dokumentti [viitattu 7.4.2011]. Saatavissa: http://www.cookiecentral.com/cm002.htm

FileZilla. 2011. WWW-dokumentti [viitattu 8.4.2011]. Saatavissa: http://filezilla-project.org/

Finlex. 2004. Sähköisen viestinnän tietosuojalaki. WWW-dokumentti [viitattu 13.4.2011]. Saatavissa: http://www.finlex.fi/fi/laki/ajantasa/2004/20040516

Louhi. 2011. Louhen yritysesittely. WWW-dokumentti [Viitattu 10.2.2011]. Saatavissa: http://www.louhi.fi/yritysesittely

Microsoft MSDN. 2011. Using SQL Server Management Studio. WWW-dokumentti [viitattu 8.4.2011]. Saatavissa: http://msdn.microsoft.com/en-us/library/ms174173.aspx

Pear. 2011. Mail_Mime. WWW-dokumentti [viitattu 12.4.2011]. Saatavissa: http://pear.php.net/package/Mail_Mime/

Visma Nova. 2011. Visma Nova tuote-esite. PDF-dokumentti [Viitattu 10.2.2011]. Saatavissa: http://www.visma.fi

PHP. 2011a. Manual: mysql_real_escape_string. WWW-dokumentti [viitattu 8.4.2011]. Saatavissa: http://php.net/manual/en/function.mysql-real-escape-string.php

PHP. 2011b. Manual: Round. WWW-dokumentti [viitattu 7.4.2011]. Saatavissa: http://fi.php.net/manual/en/function.round.php

PHP. 2011c. Manual: SQL Injection. WWW-dokumentti [viitattu 7.4.2011]. Saatavissa: http://php.net/manual/en/security.database.sql-injection.php

Vixie P. 2011. manpagez. crontab(5). WWW-dokumentti [viitattu 8.4.2011]. Saatavissa: http://www.manpagez.com/man/5/crontab/

W3C. 2011. CSS Color Module Level 3. WWW-dokumentti [viitattu 7.4.2011]. Saatavissa: http://www.w3.org/TR/css3-color/#svg-color