VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Ville-Waltteri Viklund

# WEB-SOVELLUKSEN SUUNNITTELU JA TOTEUTUS

Liiketalous ja matkailu

2010

VAASAN AMMATTIKORKEAKOULU

Tietojenkäsittelyn koulutusohjelma

TIIVISTELMÄ

| | |
|---|---|
| Tekijä | Ville Viklund |
| Opinnäytetyön nimi | Web-sovelluksen suunnittelu ja toteutus |
| Vuosi | 2010 |
| Kieli | suomi |
| Sivumäärä | 38 |
| Ohjaaja | Sirkka Hellman |

Tämä opinnäytetyö verkkosivulle liitettävän sovelluksen suunnittelun ja toteutuksen. Tavoitteena oli oppia lisää web-tekniikoista ja käytettävyydestä sekä toteuttaa hyvän käyttökokemuksen tarjoava lisäosa kalassa.net-verkkosivustolle.

Suurin osa aineistosta oli käytettävyyden osalta Nielsenin kirjat sekä verkkosivut. Web-tekniikoiden ja kielien materiaalina olen käyttänyt monia eri kirjoja sekä PHP:n dokumentaatiota.

Käytännön osuuden aikana opin uusia web-tekniikoita ja käytettävyyden soveltamista käytäntöön. Opinnäytetyöstä jäi kipinä opiskella lisää käytettävyyden teorioita.

Asiasanat PHP, Käytettävyys, UML

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Tietojenkäsittelyn koulutusohjelma

## ABSTRACT

| | |
|---|---|
| Author | Ville Viklund |
| Title | Designing and Implementing a Web Application |
| Year | 2010 |
| Language | Finnish |
| Pages | 38 |
| Name of Supervisor | Sirkka Hellman |

This thesis presents the planning and implementation of a web application for kalassa.net website. The thesis also covers the techniques and programming languages used in the application. The goal for this thesis was to learn more about usability and web techniques.

A vast majority of the used material in the usability field was from Jakob Nielsen's books and website. For web techniques and languages I used many different books and the documentation of PHP.

During the practical study I learned new web techniques and applying theories of usability to practice. The making of the thesis left a spark to study more theories about usability.

Keywords PHP, Usability, UML

**SISÄLLYS**

TIIVISTELMÄ ..... 2
ABSTRACT ..... 3
1 Johdanto ..... 3
1.1 Tausta ..... 3
1.2 Tavoitteet ja rajaukset ..... 3
1.3 Työn rakenne ..... 3
2 Käyttötapaukset ..... 4
2.1 Kokonaiskuva ..... 4
2.2 Kalastajan käyttötapaukset ..... 5
2.2.1 Sovelluksen selaaminen ..... 5
2.2.2 Saaliiden käsittely ..... 7
2.2.3 Kommenttien käsittely ..... 9
2.3 Hallintakäyttäjän käyttötapaukset ..... 11
2.3.1 Kalalajien käsittely ..... 11
2.3.2 Pyyntitapojen käsittely ..... 14
2.3.3 Kommenttien käsittely ..... 16
2.3.4 Saaliiden käsittely ..... 17
3 Tietokanta ..... 19
4 Tekniikat ja kielet ..... 20
4.1 PHP ..... 20
4.2 Javascript ..... 22
4.3 SQL ..... 23
4.4 AJAX ..... 23
4.5 Tietoturva ..... 24
4.5.1 Cross Site Scripting ..... 24
4.5.2 SQL-injektio ..... 25
5 Käytettävyys ..... 27
6 Käytettävyyden teorioiden soveltaminen ..... 28
6.1 Linkit ..... 28
6.2 Vasteaika ..... 29

6.3 Painonapit ..... 29
6.4 Virhetilanteet ..... 30
6.5 Miellyttävyys ..... 31
7 Käyttöliittymän suunnittelu ja toteutus ..... 32
7.1 Lisää kala -lomake ..... 32
7.2 Profiili ..... 33
7.3 Saaliin kommentit ..... 34
7.4 Näyttöala ..... 35
8 Loppusanat ..... 36
9 Lähdeluettelo ..... 37

# 1 Johdanto

Tämän opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa web-sovellus. Sovelluksen suunnittelun aloitin UML:n mukaisilla käyttötapauskaavioilla. UML on joukko graafisia merkintöjä, joita käytetään sovellusten suunnitteluun ja kuvaamiseen. (Fowler, 2004, 1). Sovelluksen toteutukseen käytin yleisimpiä web-ohjelmoinnissa käytettyjä kieliä, jotka esittelen tarkemmin myöhemmin.

## 1.1 Tausta

Opiskelen neljättä vuotta Vaasan ammattikorkeakoulussa tietojenkäsittelyn linjalla. Teen työkseni ja vapaa-ajalla www-sivustoja sekä erilaisia dynaamisia web-sovelluksia, joten tämä aihe oli luonnollinen valinta minulle.

Opinnäytetyö tehtiin kalastusaiheiselle kalassa.net-verkkosivustolle lisäosaksi. Sivustolla on tällä hetkellä noin 100 000 kävijää kuukaudessa. Tätä opinnäytetyötä varten olen lukenut runsaasti alan kirjallisuutta sekä etsinyt lisätietoa internetistä.

## 1.2 Tavoitteet ja rajaukset

Opinnäytetyön tavoitteena on oppia lisää käytettävyydestä ja kuinka se toteutetaan käytännössä. Lisäksi tavoitteena on myös tutustua eri web-tekniikoihin ja niiden käyttämiseen.

Opinnäytetyön lopullisena tavoitteena on oppia tekemään sivustoja, joista käyttäjä saa hyvän käyttökokemuksen.

## 1.3 Työn rakenne

Tässä johdanto luvussa olen käsitellyt työn taustaa, tavoitteita ja rajauksia sekä työn rakennetta.

Luvussa kaksi käsitellään sovelluksessa suunnitteluvaiheessa tehtyjä käyttötapauksia. Käyttötapauksia olen havainnollistanut kaavion ja sanallisin selityksin.

Luvussa kolme käsitellään sovelluksen tietokantaa.

Luvussa neljä käsitellään sovelluksessa käytettyjä tekniikoita sekä ohjelmointikieliä.

Luvussa viisi esitellään käytettävyyden eri teorioita.

Luvussa kuusi käsitellään käytettävyyden teorioiden soveltamista käytäntöön.

Luvussa seitsemän käsitellään käyttöliittymän suunnittelua ja toteutusta.

# 2 Käyttötapaukset

Käyttötapauksilla kuvataan toimijan ja järjestelmän vuorovaikutusta. Kuvaukset toteutetaan käyttäen kaavioita ja sanallisia selityksiä. Selityksessä voidaan käydä läpi muun muassa seuraavat; toimija, käytettävyysvaatimukset, tapahtumien kulku, esiehdot, poikkeukset ja virhetilanteet sekä lopputulos. Lisäksi käyttötapauksille voidaan antaa oma yksilöintitieto, sillä nimet saattavat muuttua.

Käyttötapauksissa keskitytään siihen, mitä järjestelmän tulisi tehdä eikä miten se sen tekee. (Conallen, 2003, 174)

## 2.1 Kokonaiskuva

Käyttötapaukset on jaettu kahdelle toimijalle, kalastajalle ja hallintakäyttäjälle.

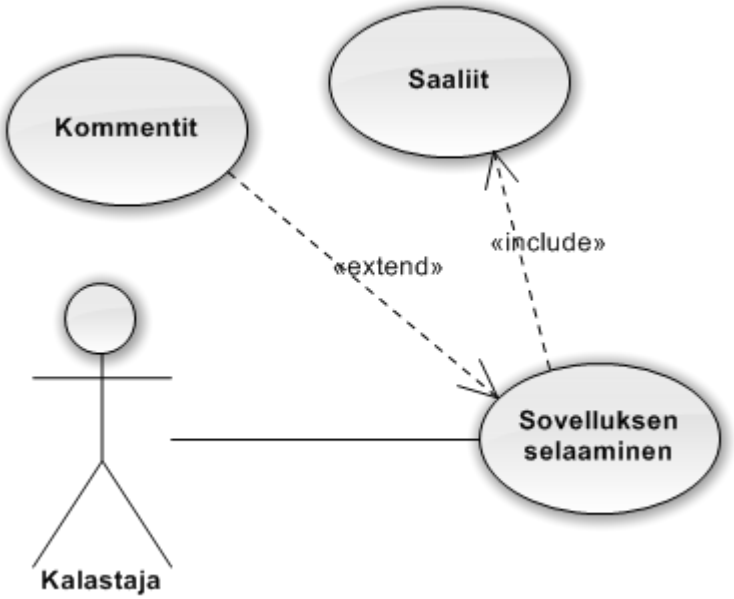
Kalastajan käyttötapauksiin kuuluvat sovelluksen selaus, kommenttien käsittely sekä saaliiden käsittely. Hallintakäyttäjän käyttötapauksiin kuuluvat kalalajien käsittely, kalojen käsittely, kommenttien käsittely ja pyyntitapojen käsittely.

**Kuva 1. Käyttötapausten kokonaiskuva.**

## 2.2 Kalastajan käyttötapaukset

Kalastajan käyttötapauksilla tarkoitetaan niitä tilanteita, joissa kalastaja käyttää sovellusta kuvien 2-4 mukaisesti.

### 2.2.1 Sovelluksen selaaminen




**Kuva 2. Sovelluksen selaaminen.**

**Käyttötapaus**: Saaliiden selaaminen

**Toimija**: Kalastaja

**Käytettävyysvaatimukset**: Toimija voi selata saaliita omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa joko profiili- tai tilastot-linkkiä, jotka listaavat järjestelmään lisätyt saaliit.

**Esiehdot**: Järjestelmään on lisätty saaliita.

**Poikkeukset/Virhetilanteet**: Tietokannasta haku ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Toimija selaa saaliita.

**Käyttötapaus**: Kommenttien selaaminen

**Toimija**: Kalastaja

**Käytettävyysvaatimukset**: Toimija voi selata kommentteja omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa saaliiden listauksessa kommentit-linkkiä, joka avaa kyseiselle saaliille lisätyt kommentit.

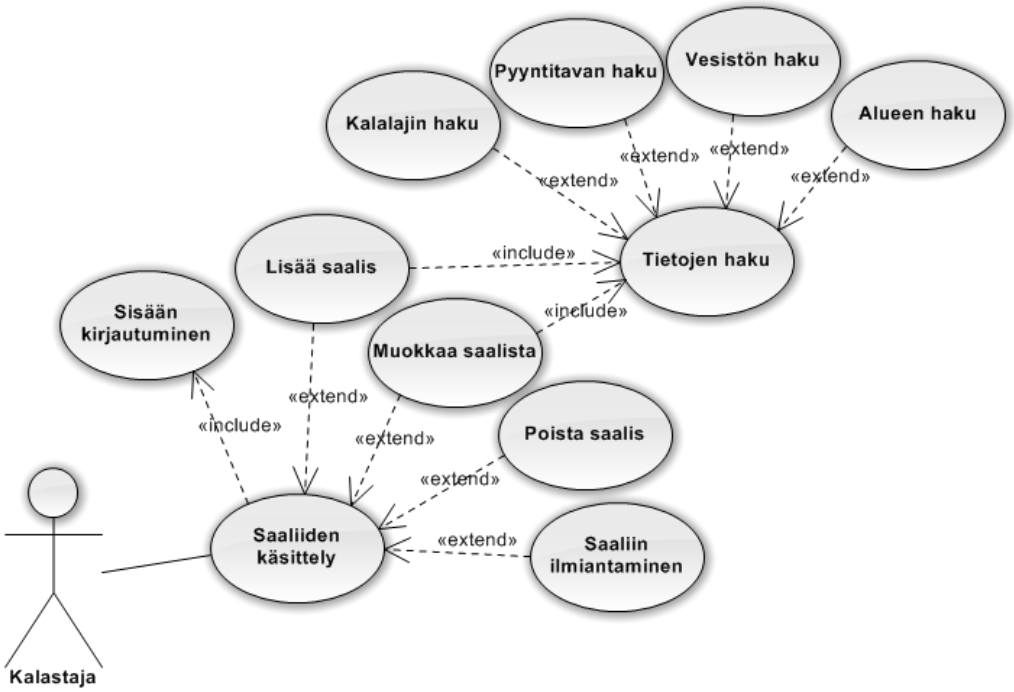
**Esiehdot**: Kommentteja on lisätty saaliille.

**Poikkeukset/Virhetilanteet**: Tietokannasta haku ei onnistu järjestelmän yhteysongelman vuoksi. Toimijalla on JavaScript kytketty pois päältä.

**Lopputulos**: Toimija selaa kommentteja.

### 2.2.2 Saaliiden käsittely

Kuvassa 3 esitellään kalastajan käyttötapauksista saaliiden käsittely. Kalastaja voi lisätä, muokata ja poistaa saaliita. Lisäksi kalasta voi myös ilmiantaa muiden lisäämiä saaliita.




**Kuva 3 Saaliiden käsittely**

**Käyttötapaus**: Lisää saalis

**Toimija**: Kalastaja

**Käytettävyysvaatimukset**: Toimija voi lisätä saaliin omalta koneeltaan.

**Tapahtumien kulku**: Toimija täyttää lisää kala -lomakkeella pakolliset kentät, jotka ovat kala, paino, pyyntitapa, vesistö, päivämäärä ja kuva. Lomakkeen täyttämisen jälkeen toimija painaa tallenna-nappia. Mikäli lisäys onnistuu, saalis lisätään välittömästi muiden kalastajien nähtäväksi.

**Esiehdot**: Toimija on kirjautunut järjestelmään.

**Poikkeukset/Virhetilanteet**: Paino-kenttään on syötetty kirjaimia tai pakollisia tietoja puuttuu. Tietokantaan lisäys ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Toimijan saalis on lisätty tietokantaan.

### **Käyttötapaus:** Muokkaa saalista

**Toimija:** Kalastaja

**Käytettävyysvaatimukset:** Toimija voi muokata lisäämään saalistaan omalta koneeltaan.

**Tapahtumien kulku:** Toimija siirtyy muokkauslomakkeelle painamalla halutun saaliin kohdalla muokkaus-ikonia. Lomakkeella toimija muokkaa haluamansa tiedot ja painaa tallenna-nappia.

**Esiehdot:** Toimija on kirjautunut järjestelmään. Toimija on lisännyt järjestelmään saaliita.

**Poikkeukset/Virhetilanteet:** Paino-kenttään on syötetty kirjaimia tai pakollisia tietoja puuttuu. Tietokantaan lisäys ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos:** Toimijan saaliin muokatut tiedot tallennetaan tietokantaan.

### **Käyttötapaus**: Poista saalis

**Toimija**: Kalastaja

**Käytettävyysvaatimukset**: Toimija voi poistaa lisäämänsä saaliin omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa saaliin poista-ikonia profiilistaan tai jostain muusta listauksesta. Ehtojen täyttyessä kala poistetaan välittömästi.

**Esiehdot**: Toimija on kirjautunut järjestelmään. Toimija on lisännyt järjestelmään saaliita.

**Poikkeukset/Virhetilanteet**: Tietokannasta poisto ei onnistu järjestelmän yhteys-ongelman vuoksi.

**Lopputulos**: Toimijan saalis on poistettu tietokannasta.

**Käyttötapaus**: Ilmianna saalis

**Toimija**: Kalastaja

**Käytettävyysvaatimukset**: Toimija voi ilmiantaa järjestelmään lisätyn saaliin omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa halutun saaliin kohdalla kalavale?-linkkiä, josta lähtee ilmoitus ylläpidolle mahdollisesta kalavaleesta. Järjestelmä näyttää toimijalle ponnahdusikkunana ilmoituksen ilmoituksen rekisteröinnistä.

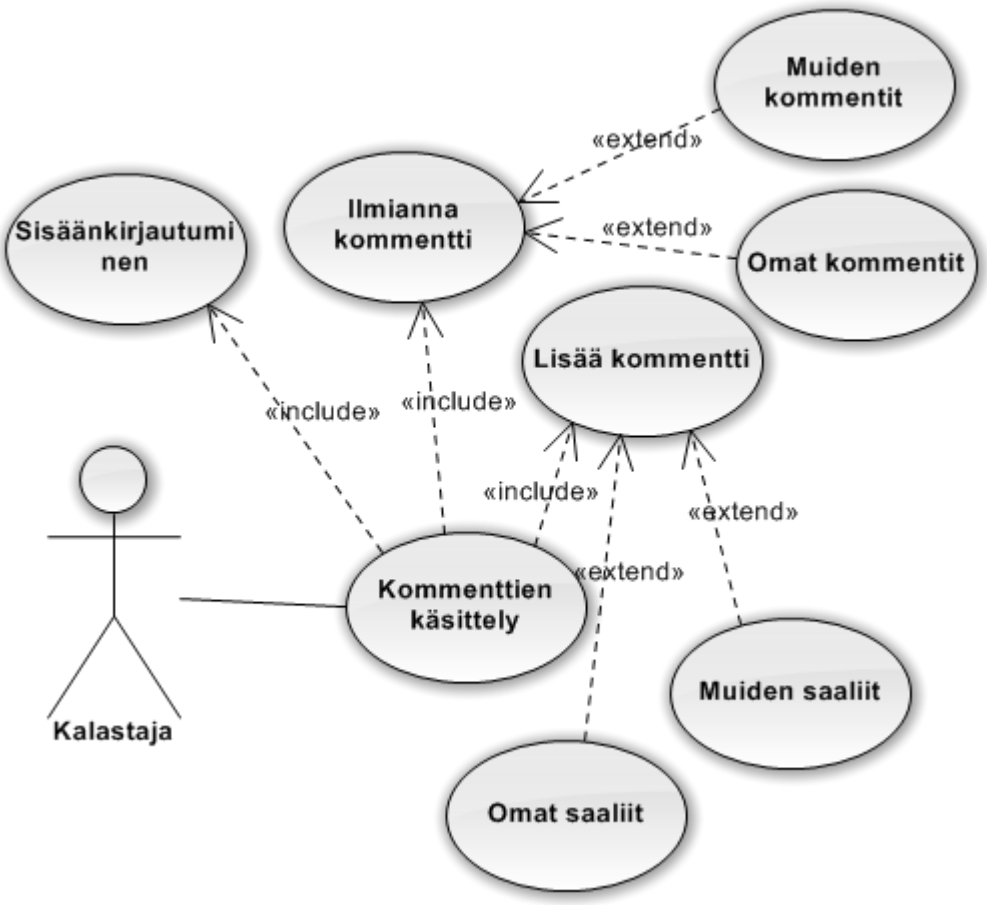
**Esiehdot**: Toimija on kirjautunut järjestelmään. Järjestelmään on lisätty kaloja.

**Poikkeukset/Virhetilanteet**: Tietokannasta lisäys ei onnistu järjestelmän yhteys-ongelman vuoksi.

**Lopputulos**: Saaliin tietoihin on lisätty epäilys kalavaleesta. Tämä ilmoitus näkyy vain ylläpidolle.

### 2.2.3 Kommenttien käsittely

Kuvassa 4 esitellään kalastajan käyttötapauksista kommenttien käsittely. Kalastaja voi lisätä kommentteja omille ja muiden lisäämiä saaliita. Lisäksi kalastaja voi myös ilmiantaa asiattomia kommentteja.




**Kuva 4. Kommenttien käsittely.**

**Käyttötapaus**: Lisää kommentti

**Toimija**: Kalastaja

**Käytettävyysvaatimukset**: Kalastaja voi lisätä saaliille kommentin omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa saaliin lisää kommentti-linkkiä, jonka jälkeen hän täyttää avautuvan lomakkeen ja painaa kommentoi-nappia. Mikäli kommentoija ja kommentti täyttävät vaatimukset, kommentti lisätään välittömästi muiden kalastajien nähtäväksi.

**Esiehdot**: Toimija on kirjautunut järjestelmään.

**Poikkeukset/Virhetilanteet**: Paino-kenttään on syötetty kirjaimia tai pakollisia tietoja puuttuu. Tietokantaan lisäys ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Toimijan saalis on lisätty tietokantaan.

**Käyttötapaus**: Ilmianna kommentti

**Toimija**: Kalastaja

**Käytettävyysvaatimukset**: Toimija voi ilmiantaa kommentin omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa ilmianna kommentti-linkkiä, jonka jälkeen kommentti poistuu näkyvistä, jos esiehdot täyttyvät.

**Esiehdot**: Toimija on kirjautunut järjestelmään. Järjestelmään on lisätty saaliita. Saaliille on lisätty kommentti.

**Poikkeukset/Virhetilanteet**: Tietokantaan lisäys ei onnistu järjestelmän yhteysongelman vuoksi.

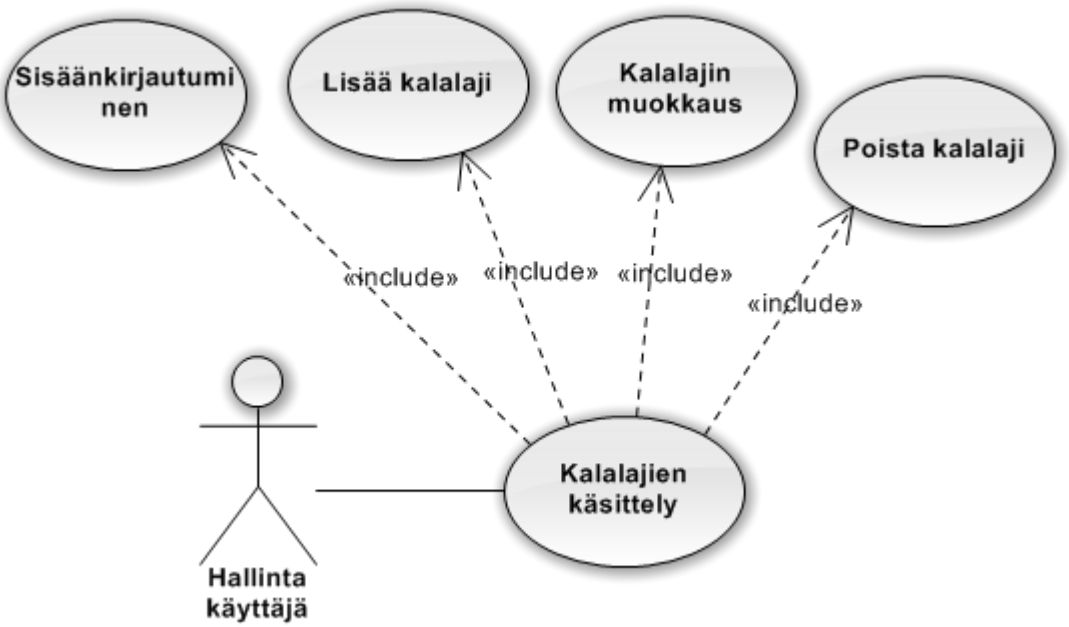
**Lopputulos**: Tieto ilmiannetusta kommentista lisätään tietokantaan ja ilmiannettu kommentti poistuu näkyvistä.

### 2.3 Hallintakäyttäjän käyttötapaukset

Hallintakäyttäjän käyttötapauksilla tarkoitetaan, niitä tilanteita, joissa hallintakäyttäjä käyttää sovellusta kuvien 5-8 mukaisesti.

#### 2.3.1 Kalalajien käsittely

Kuvassa 5 esitellään hallintakäyttäjän käyttötapauksista kalalajien käsittely. Hallintakäyttäjä voi lisätä, muokata ja poistaa kalalajeja.




**Kuva 5. Kalalajien käsittely.**

## **Käyttötapaus**: Lisää kalalaji

**Toimija**: Hallintakäyttäjä

**Käytettävyysvaatimukset**: Hallintakäyttäjä voi lisätä kalalajin omalta koneeltaan.

**Tapahtumien kulku**: Toimija täyttää lisää kalalaji-lomakkeella pakollisen kentän nimi. Täyttämisen jälkeen toimija painaa tallenna-nappia. Mikäli lisäys onnistuu, kalalaji on välittömästi käytettävissä sovelluksen käyttäjille.

**Esiehdot**: Hallintakäyttäjä on kirjautunut järjestelmään.

**Poikkeukset/Virhetilanteet**: Nimi-kenttään syötetään jotain muuta kuin kirjaimia. Tietokantaan lisäys ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Kalalaji on lisätty tietokantaan.

### **Käyttötapaus**: Muokkaa kalalajia

**Toimija**: Hallintakäyttäjä

**Käytettävyysvaatimukset**: Toimija voi muokata kalalajia omalta koneeltaan.

**Tapahtumien kulku**: Toimija siirtyy muokkauslomakkeelle painamalla halutun kalalajin kohdalla muokkaa-ikonia. Lomakkeella toimija muokkaa haluamansa tiedot ja painaa tallenna-nappia.

**Esiehdot**: Toimija on kirjautunut järjestelmään.

**Poikkeukset/Virhetilanteet**: Nimi-kenttään syötetään jotain muuta kuin kirjaimia. Tietokantaan lisäys ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Kalalajin muokatut tiedot ovat tallennettu tietokantaan.

### **Käyttötapaus**: Poista kalalaji

**Toimija**: Hallintakäyttäjä

**Käytettävyysvaatimukset**: Toimija voi poistaa kalalajin omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa kalalajien listauksessa halutun kalalajin kohdalla poista-ikonia. Esiehtojen täyttyessä kalalaji poistetaan tietokannasta.

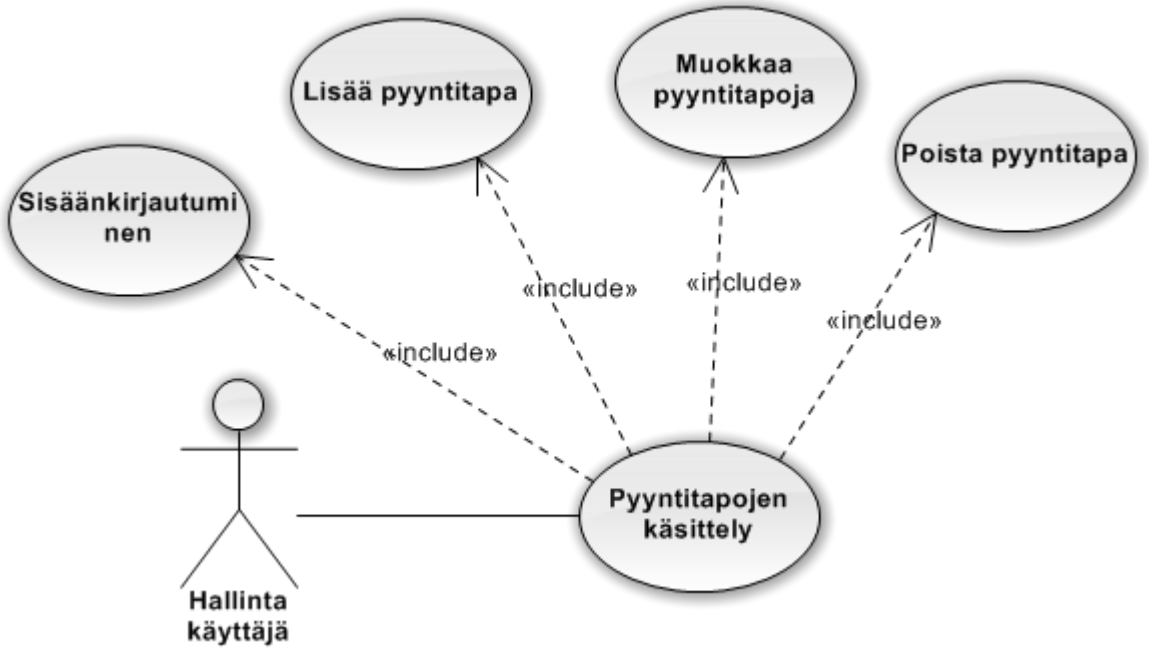
**Esiehdot**: Toimija on kirjautunut järjestelmään.

**Poikkeukset/Virhetilanteet**: Tietokannasta poisto ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Kalalaji on poistettu tietokannasta.

### 2.3.2 Pyyntitapojen käsittely

Kuvassa 6 esitellään hallintakäyttäjän käyttötapauksista pyyntitapojen käsittely. Hallintakäyttäjä voi lisätä, muokata ja poistaa pyyntitapoja.




**Kuva 6. Pyyntitapojen käsittely.**

**Käyttötapaus**: Lisää pyyntitapa

**Toimija**: Hallintakäyttäjä

**Käytettävyysvaatimukset**: Toimija voi lisätä pyyntitapoja omalta koneeltaan.

**Tapahtumien kulku**: Toimija täyttää lisää pyyntitapa-lomakkeen pakollisen kentän nimi, jonka jälkeen toimija painaa tallenna-nappia. Mikäli lisäys onnistuu, pyyntitapa on välittömästi käytettävissä sovelluksessa.

**Esiehdot**: Toimija on kirjautunut järjestelmään.

**Poikkeukset/Virhetilanteet**: Nimi-kenttään syötetään jotain muuta kuin kirjaimia. Tietokantaan lisäys ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Pyyntitapa on lisätty tietokantaan.

### **Käyttötapaus**: Muokkaa pyyntitapoja

**Toimija**: Hallintakäyttäjä

**Käytettävyysvaatimukset**: Toimija voi muokata pyyntitapoja omalta koneeltaan.

**Tapahtumien kulku**: Toimija siirtyy muokkaa pyyntitapa-lomakkeelle painamalla halutun pyyntitavan kohdalla muokkaa-ikonia. Lomakkeella toimija muokkaa haluamansa kentät ja painaa tallenna-nappia.

**Esiehdot**: Toimija on kirjautunut järjestelmään. Pyyntitapoja on lisätty järjestelmään.

**Poikkeukset/Virhetilanteet**: Nimi-kenttään syötetään jotain muuta kuin kirjaimia. Tietokantaan lisäys ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Pyyntitavan tiedot on muokattu tietokantaan.

### **Käyttötapaus**: Poista pyyntitapa

**Toimija**: Hallintakäyttäjä

**Käytettävyysvaatimukset**: Toimija voi poistaa pyyntitapoja omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa pyyntitapojen listauksessa halutun pyyntitavan kohdalla poista-ikonia. Esiehtojen täyttyessä pyyntitapa poistetaan tietokannasta.

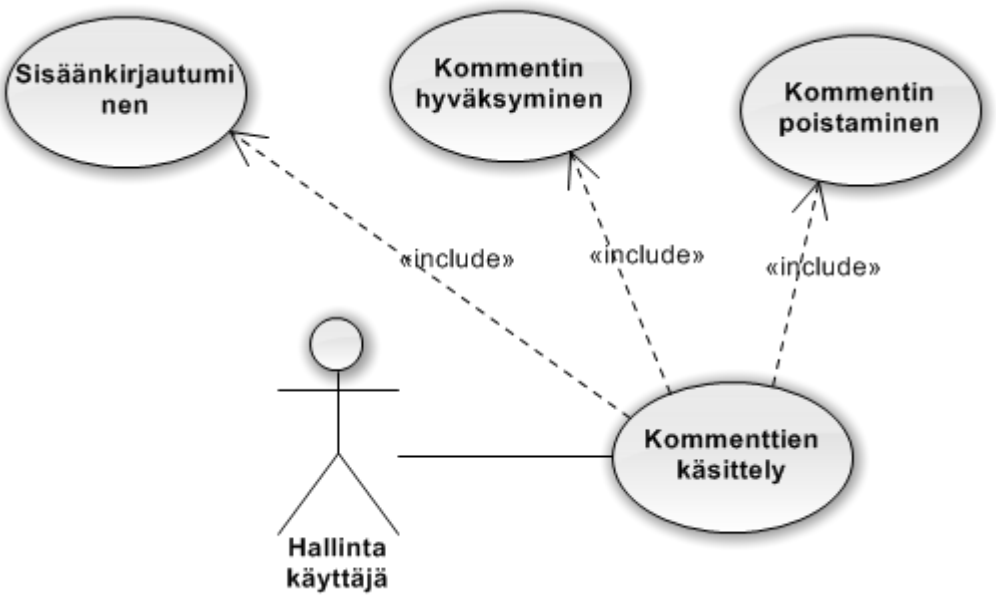
**Esiehdot**: Toimija on kirjautunut järjestelmään. Pyyntitapoja on lisätty järjestelmään.

**Poikkeukset/Virhetilanteet**: Nimi-kenttään syötetään jotain muuta kuin kirjaimia. Tietokantaan lisäys ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Pyyntitapa on poistettu tietokannasta.

### 2.3.3 Kommenttien käsittely

Kuvassa 7 esitellään hallintakäyttäjän käyttötapauksista kommenttien käsittely. Hallintakäyttäjä voi joko hyväksyä tai poistaa kommentin.




**Kuva 7. Kommenttien käsittely.**

**Käyttötapaus**: Hyväksy kommentti

**Toimija**: Hallintakäyttäjä

**Käytettävyysvaatimukset**: Toimija voi hyväksyä kommentteja omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa kommenttien listauksessa hyväksy-ikonia. Esiehtojen täyttyessä kommentti hyväksytään sovelluksessa näytettäväksi.

**Esiehdot**: Toimija on kirjautunut järjestelmään. Järjestelmään on lisätty saaliita. Järjestelmään on lisätty kommentteja. Kommentti on ilmiannettu.

**Poikkeukset/Virhetilanteet**: Tietokantaan lisäys ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Kommentti näkyy sovelluksessa.

**Käyttötapaus**: Poista kommentti

**Toimija**: Hallintakäyttäjä

**Käytettävyysvaatimukset**: Toimija voi poistaa kommentin omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa kommenttien listauksesta poista-ikonia. Esiehtojen täyttyessä kommentti poistetaan tietokannasta.

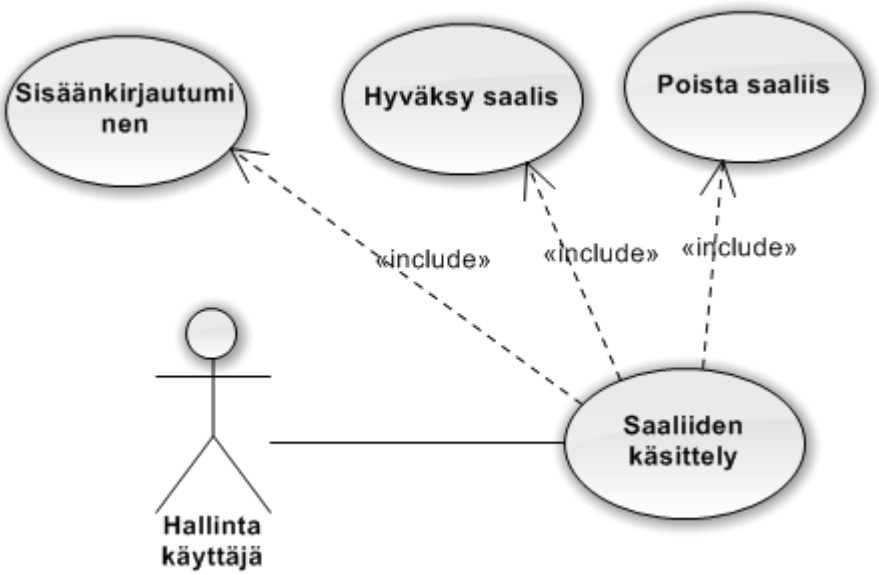
**Esiehdot**: Toimija on kirjautunut järjestelmään. Järjestelmään on lisätty saaliita. Järjestelmään on lisätty kommentteja. Kommentti on ilmiannettu.

**Poikkeukset/Virhetilanteet**: Tietokannasta poistaminen ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Kommentti on poistettu tietokannasta.

### 2.3.4 Saaliiden käsittely

Kuvassa 8 esitellään hallintakäyttäjän käyttötapauksista saaliiden käsittely. Hallintakäyttäjä voi joko hyväksyä tai poistaa saaliin.




**Kuva 8. Saaliiden käsittely.**

**Käyttötapaus**: Hyväksy saalis

**Toimija**:
Hallintakäyttäjä

**Käytettävyysvaatimukset**: Toimija voi hyväksyä saaliita omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa saaliiden listauksessa hyväksy-ikonia. Esiehtojen täyttyessä ilmiannettu saalis poistetaan sovelluksesta.

**Esiehdot**: Toimija on kirjautunut järjestelmään. Järjestelmään on lisätty saaliita. Saalis on ilmiannettu.

**Poikkeukset/Virhetilanteet**: Tietokannasta poisto ei onnistu järjestelmän yhteysongelman vuoksi.

**Lopputulos**: Saalis on poistettu sovelluksesta.

**Käyttötapaus**: Poista saalis

**Toimija**: Hallintakäyttäjä

**Käytettävyysvaatimukset**: Toimija voi poistaa saaliita omalta koneeltaan.

**Tapahtumien kulku**: Toimija painaa saaliiden listauksessa poista-ikonia. Esiehtojen täyttyessä ilmiannettu saalis hyväksytään sovelluksessa näytettäväksi.

**Esiehdot**: Toimija on kirjautunut järjestelmään. Järjestelmään on lisätty saaliita. Saalis on ilmiannettu

**Poikkeukset/Virhetilanteet**: Tietokantaan lisäys ei onnistu järjestelmän yhteysongelman vuoksi.

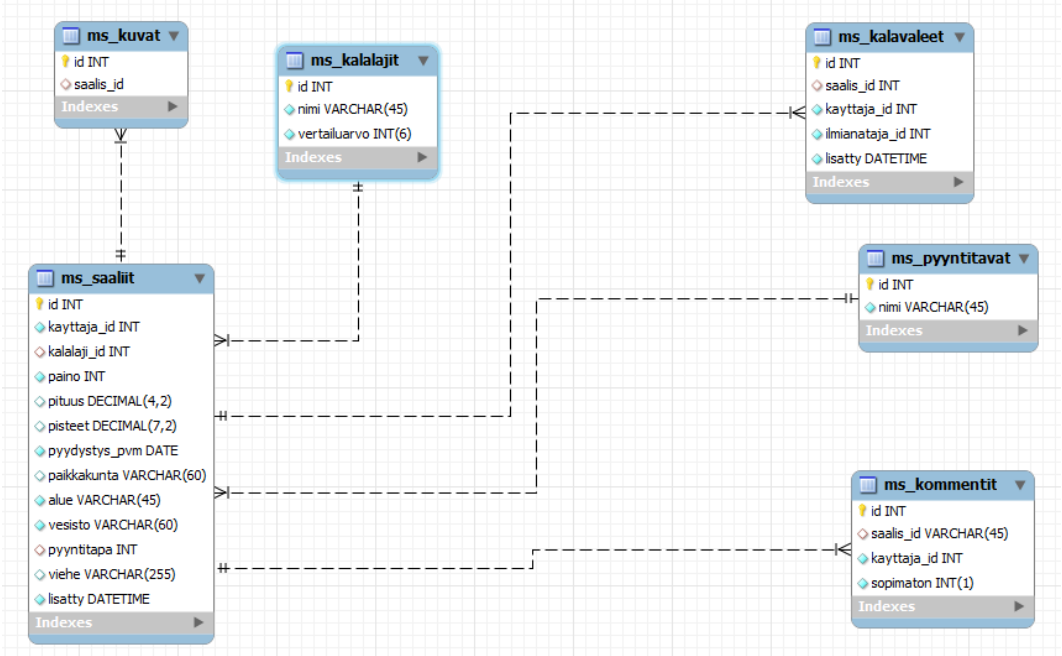
**Lopputulos**: Saalis näkyy sovelluksessa.

# 3 Tietokanta

Sovellus käyttää Linux-palvelimella toimivaa MySQL SQL -tietokannan hallintajärjestelmää.

Sovellus on liitetty kalassa.net-verkkosivustolle, joten se myös käyttää kalassa.net jäsentiedot sisältävää taulua osana järjestelmäänsä. Sovelluksen oma tietokanta sisältää kuusi taulua. Taulut on nimetty käyttäen ms_ etuliitettä, jotta ne on helpompi erottaa muista kalassa.net-tietokannan tauluista.

Tietokanta on toteutettu niin, että se voidaan liittää olemassa olevaan järjestelmään, joka sisältää jäsenrekisterin.




**Kuva 9. Tietokanta.**

Kuvassa 9 näytettävän tietokannan keskeisimpänä tauluna toimii ms_saaliit-taulu, johon tallennetaan nimensä mukaisesti saaliin tiedot. Yhtä saalista kohden voidaan liittää yksi pyyntitapa, yksi kuva, rajaton määrä kommentteja sekä kalavalepäilyksiä.

# 4 Tekniikat ja kielet

Sovelluksen toteutuksessa on käytetty yleisimpiä web-tekniikoita. Ulkoasu on toteutettu CSS:llä käyttäen div-taittoa. Ainoastaan kalassa.net kuva ja valikko on toteutettu taulukkorakenteella. Erikoisefektit toteutettiin jQuery JavaScript-kirjaston avulla.

## 4.1 PHP

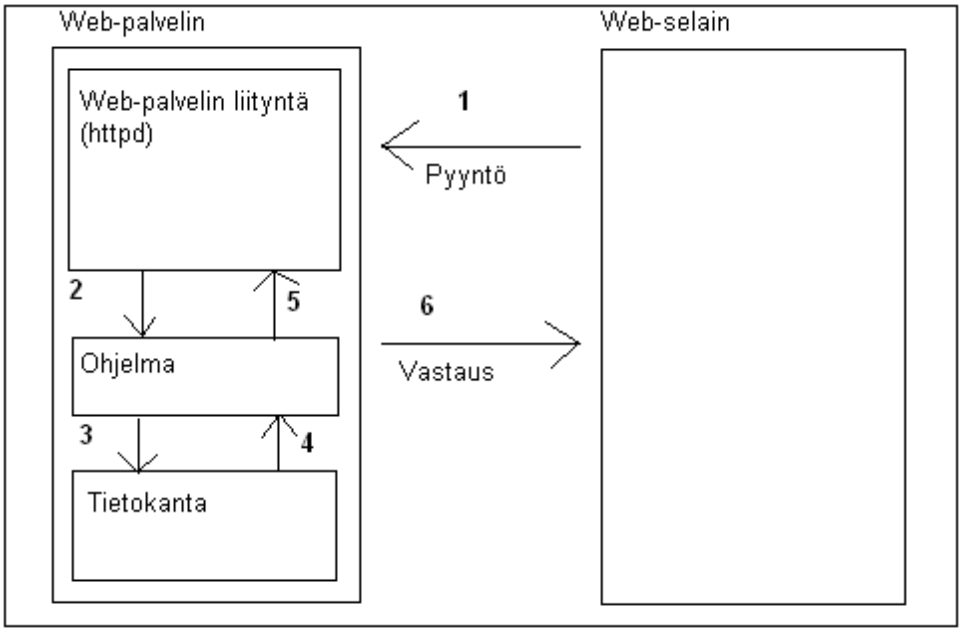
PHP (Hypertext Preprocessor) on HTML-sivustojen dynaamiseen kehitykseen soveltuva avoimen lähdekoodin komentosarjakieli. PHP-koodi suoritetaan palvelimella toisin kuin esimerkiksi JavaScipt. PHP:n syntaksi muistuttaa paljon C-, Java- ja Pearl-kieliä.

PHP:n ensimmäiset versiot kehitti Rasmus Lerdof vuonna 1995 nimellä PHP/FI (Personal Home Page/Forms Interpreter). Ensimmäinen julkaistu versio oli toteuttu C-ohjelmointikielellä. Vuonna 1997 julkaistu toinen versio asennettiin jopa 50 000 palvelimelle.

Vuonna 1998 julkaistu PHP3 oli kokonaan uudelleen kirjoitettu versio PHP/FI 2.0:sta, jonka kehittäjinä toimivat Andi Gutmans, Zeev Suranski sekä Rasmus Lerdof. Uudelleenkirjoituksen yhteydessä PHP3:sta muodostui pitkälti samanlainen kuin nykyisin käytettävistä PHP-versioista.

Toukokuussa 2000 julkaistu PHP4 sisälsi uuden ytimen, nimeltä Zend Engine. Uusi versio sisälsi monia huomattavia parannuksia ja PHP saavutti vielä suuremman suosion ympäri maailmaa. Viimeisin PHP:n versio eli PHP5 julkaistiin heinäkuussa 2004. Uusin versio sisälsi uuden ytimen Zend Engine 2.0 ja myös huomattavasti parannuksia PHP:n olio-ohjelmointiin.

PHP-koodi kirjoitetaan <?php ja ?> merkkien väliin ja PHP:ta sisältävät tiedostot tallennetaan .php-päätteellä. Palvelin tunnistaa tiedoston .php-päätteen ja lähettää sen PHP-tulkille esiprosessoitavaksi. Esiprosessointi sisältää PHP-koodin kääntämisen sekä sen suorittamisen. Tämän jälkeen tulkki palauttaa palvelimelle HTML-koodia, joka näytetään asiakkaalle. Vaikka PHP-koodi on suoritetussa tiedostossa, käyttäjä ei milloinkaan näe sitä. (php.net, 2010; php.net, 2010a) Tämä toiminto on havainnollistettu kuvassa 10.




**Kuva 10. PHP:n toimintaperiaate**

1. Web-selain lähettää pyynnön palvelimelle.
2. Palvelin tunnistaa .php-päätteen ja ohjaa sen käyttämään sille varattua ohjelmaa.
3. Suoritetaan mahdolliset tietokantapyynnöt.
4. Palautetaan tavarat tietokannasta.
5. Muodostetaan pyynnöstä HTML-dokumentti.
6. Välitetään vastaus web-selaimelle.

(Rantala, 2005, 4)

## 4.2 Javascript

JavaScript on komentosarjakieli, jonka avulla voidaan luoda www-sivustoja, jotka ovat dynaamisesti vuorovaikutteisia käyttäjän kanssa.

Ensimmäinen JavaScript versio JavaScript 1.0 julkaistiin vuonna 1995 Netscapen Navigator 2-selaimen mukana. Alkuperäiseltä nimeltään JavaScript oli LiveScript, mutta se vaihtui markkinoinnin myötä JavaScriptiksi, johtuen Netscapen yhteistyöstä Java-kielen kehittäjän kanssa. Viimeisin JavaScriptin versio on Mozilla-projektin kehittämä JavaScript 1.6. (Peltomäki & Nykänen, 2006, 90–92)

JavaScript liitetään tavallisesti script-elementtien sisään html-dokumentin otsikko-osaan (head), mutta se voidaan sijoittaa myös runko-osaan (body). JavaScript-koodin suorittaa selaimeen upotettu JavaScript-tulkki.

### 4.3 SQL

SQL (Structured Query Language) on esimerkiksi MySQL-tietokannoissa käytettävä kysyntäkieli.

SQL:n kehitettiin SEQUEL-tietokantakielen pohjalta IBM:n tutkimuskeskuksessa 1970-luvulla. SQL:n ensimmäinen kaupallinen tuote oli vuonna 1979 julkaistu Oracle. IBM julkaisi oman kaupallisen tuotteensa SQL/DS vuonna 1981. Tämän jälkeen kehitettiin paljon muita SQL-pohjaisia tuotteita, kuten SYBASE ja RDB. ANSI julkaisi SQL:n standardin vuonna 1986.

SQL:ää käytetään erityisellä käyttöliittymällä tai upotettuna ohjelmointikieleen esimerkiksi Pearl, Java ja PHP. (Hovi, 2004, 9–11)

### 4.4 AJAX

AJAX (Asynchnorous Javascript and XML) tarkoittaa tekniikkaa, jolla lähetetään ja palautetaan dataa www-selaimen ja www-palvelimen välillä. AJAXin tarkka määrittely on vaikeaa, silla se koostuu useista eri tekniikoista, mutta pohjautuu JavaScriptiin ja XML-dokumenttien rajapintaan.

AJAX-tekniikan erikoisuutena on se, että käyttäjä voi olla vuorovaikutuksessa web-sovelluksen kanssa lataamatta uudestaan koko sivua. HTTP-pyyntö lähetetään palvelimelle ja sen palauttama data näytetään käyttäjälle JavaScriptin avulla.

Tekemässäni sovelluksessa käytin AJAX-tekniikkaa käyttäjäpuolen kommenttien käsittelyyn. (Peltomäki & Nykänen, 2006, 297–297)

### 4.5 Tietoturva

Tietokantapohjaisissa web-sovelluksissa tietoturva on erittäin tärkeä ominaisuus. Yksinkertainen vieraskirjakin saattaa aiheuttaa suuren riskin koko tietokannalle, mikäli käyttäjän lähettämää tietoa ei käsitellä oikein.

#### 4.5.1 Cross Site Scripting

XSS eli Cross Site Scripting on web-lomakkeiden välityksellä lähetettävä haitallinen koodi. Yleisesti näitä hyökkäyksiä kohdistetaan vieraskirjoihin ja foorumeille. Hyökkäys voisi olla esimerkiksi seuraavanlainen:

```
<script=”text/javascript”>
        alert(”XSS”);
</script>
```

Hyökkääjä syöttää edellä olleen koodin lomakkeen input-kenttään, jonka välityksellä saadut tiedot tallennetaan tietokantaan. Seuraava käyttäjä, joka avaa sivun, missä kyseisen lomakkeen syötteet tulostetaan saa vastaansa varoitusikkunan. Varoitusikkuna ei itsessään ole haitallinen, mutta vastaavalla tekniikalla voidaan ohjata käyttäjä kokonaan toiselle sivulle tai muuttaa sivuston ulkoasua. (Wellington & Thomson, 2008, 365)

XSS voidaan ehkäistä heti lomakkeen tietojen vastaanottovaiheessa, käyttäen PHP:n htmlentities-funktiota. Htmlentities-funktio muuttaa kaikki HTML-tagit niitä vastaaviksi entiteeteiksi, ehkäisten haitallisen koodin suorituksen. (php.net, 2010b) Tekemässäni sovelluksessa suoritin XSS:n ehkäisyn tulostamalla kaiken käyttäjälle näkyvän tiedon erillisen funktion läpi:

```
function printData($data){
        echo htmlentities($data, ENT_QUOTES, ”UTF-8”);
}
```

### 4.5.2 SQL-injektio

SQL-injektio on yleinen web-sovelluksia vastaan kohdistettu hyökkäys, jossa hyökkääjä lähettää web-lomakkeen välityksellä haitallista SQL-koodia. Näitä hyökkäyksiä kohdistetaan esimerkiksi kirjautumislomakkeisiin. SQL-injektio onnistuu, jos lomakkeelta vastaanotettuja tietoja ei käsitellä tarpeen mukaisesti tai SQL-kysely on muodostettu huolimattomasti. SQL-injektiota voidaan yrittää esimerkiksi seuraavalla tavalla:

```
$query = "SELECT * FROM users WHERE user='{$_POST['username']}' AND
password='{$_POST['password']}'";
mysql_query($query);
```

Käyttäjän antamat tiedot voisivat olla esimerkiksi:

```
$_POST['username']='aidan';
$_POST['password']='"OR"="';
```

Tällä kyselyllä käyttäjä onnistuisi kirjautumaan sisään sovellukseen pelkästään oikealla käyttäjänimellä.

Sovelluksessa ehkäisin tämän ongelman suodattamalla kaiken käyttäjältä tulleen tiedon erillisen funktion läpi:

```
function cleanInput($data){
        if(get_magic_quotes_gpc()){
                $data = stripslashes($data);
                $data = mysql_real_escape_string($data);
        }else {
        $data = myql_real_escape_string($data);
        }
        return $data;
}
```

# 5 Käytettävyys

Käytettävyys on monialainen käsite ja sillä on monia määritelmiä. Web-puolella ehkä tunnetuin alan asiantuntija on Jakob Nielsen. Nielsenin mukaan käytettävyys on laadun määritelmä, joka ilmaisee, kuinka helppoa sovelluksen tai tuotteen käyttö on. Nielsenin määrittelemät käytettävyyden komponentit ovat:

- Opittavuus; kuinka helposti sovellusta oppii käyttämään.
- Tehokkuus; kuinka nopeasti oppinut käyttäjä voi suorittaa tehtäviä.
- Muistettavuus; kuinka hyvin sovellus jää käyttäjien mieleen
- Virheet; kuinka paljon käyttäjän tekevät virheitä.
- Tyytyväisyys; kuinka miellyttävä sovellusta on käyttää

(Nielsen, 2003)

Kansainvälinen standardoimisjärjestö ISO on myös luonut oman standardinsa käytettävyydestä, joka on ISO 9241–11. Tämän käytettävyysstandardin mukaan käytettävyyden komponentit ovat:

- Vaikuttavuus; kuinka hyvin ja ajallaan tehtävät saadaan suoritettua.
- Tehokkuus; kuinka paljon resursseja kuluu tavoitteen saavuttamiseen.
- Miellyttävyys; kuinka miellyttäväksi käyttäjä kokee tehtävän suorittamisen.

(ISO, 1998)

Näiden lisäksi hyvä käytettävyyden vaatimukset on myös määritelty seuraavasti:

- Ymmärrettävä; sovelluksen tai laitteen käyttö on helppo päätellä.
- Vaivaton; sovellus tai laite antaa suoriutua tehtävistä mahdollisimman helposti.

- Kattava; sovellus tai laite tarjoaa kaikki tiedot mitä tarvitaan tehtävään, mihin laite on tarkoitettu.

- Esteettisesti miellyttävä; sovellus tai tuote vaikuttaa laadukkaalta ja osaavalta.

(Wiio, 2004, 2–4)

# 6 Käytettävyyden teorioiden soveltaminen

## 6.1 Linkit

Tekstilinkit on eroteltava muusta sisällöstä käyttämällä leipätekstistä poikkeavaa väriä ja alleviivausta. Nämä tehokeinot varmistavat linkkien erottumisen tavallisen tekstin seasta, varsinkin jos käyttäjä on heikkonäköinen. Linkkien alleviivauksesta voidaan luopua, mikäli ulkoasusta käy selvästi ilmi, että kyseessä on esimerkiksi lista linkkejä tai navigaatio.

Linkkien värit vaihtelevat sen mukaan, onko käyttäjä jo napsauttanut vai ei. Napsauttamattomien linkkien väri tulee olla kirkas ja helposti huomattava. Napsautettujen linkkien väri tulee olla käytetyn näköinen, joka selvästi indikoi käyttäjälle, minkä osan sivustosta hän on jo nähnyt.

Linkkiteksti tarkoituksena on kertoa, mitä sivulla voi tehdä. Tästä johtuen on syytä välttää ”klikkaa tästä” ja vastaavia linkkitekstejä, koska ne eivät välitä käyttäjälle mitään hyödyllistä informaatiota.

Sovelluksessa olen käyttänyt linkeillä alleviivausta, oli linkkiä klikattu tai ei. Klikattujen linkkien värinä käytin kirkasta sinistä (#0000EE) ja klikkaamattoman linkin värinä kulunutta violettia (#551A8B). Näiden värien avulla käyttäjä näkee selvästi, esimerkiksi mitä saaliita hän on jo kommentoinut. (Nielsen, 2000, 53; 60–62; Nielsen, 2004)

## 6.2 Vasteaika

Vasteajalla tarkoitetaan sitä, kuinka nopeasti web-sivu latautuu käyttäjän selattavaksi. Nielsenin mukaan hyvä sivuston latausaika olisi 10 sekuntia, joka on nykyyhteyksillä lähes naurettavan pitkä. Sekunnin kymmenesosan vasteaika on syytä säilyä, mikäli käyttäjän halutaan tuntevan, että järjestelmä reagoi käyttäjään välittömästi. (Nielsen 2000, 41)

Saaliiden listauksessa on kuvista käytetty pienennettyä versiota, jonka avulla sivut latautuvat käyttäjälle nopeammin, vaikka saaliita olisi paljon. Lisäksi kommentit haetaan tietokannasta vasta kun käyttäjä painaa näytä kommentit-linkkiä ja silloinkin ladataan vain yhden saaliin kommentit. Näillä keinoilla saadaan sivuston selauksesta miellyttävää ja nopeaa. Lisäksi sovelluksen toteutuksessa on noudatettu Yahoo! Developer Networkin sivuston suorituskykyyn liittyviä suosituksia, joita on muun muassa CSS:n sijoitus ulkoiseen tyylitiedostoon ja JavaScriptin sijoittaminen sivun alaosaan. (Nielsen, 2000, 42)

## 6.3 Painonapit

Painonappien asettelu vaihtelee pääsääntöisesti käyttöjärjestelmittäin. Windows-käyttöjärjestelmissä ok-painike on ennen peruuta-painiketta. Macintosh-käyttöjärjestelmissä painikkeet ovat taas toisinpäin, peruuta-painike ennen ok-painiketta. Mikäli ok-painike asetetaan ensimmäiseksi, se on luonnollisessa paikassa vasemmalta oikealla lukeville, mutta jos ok-painike on viimeisenä, se on aseteltu samalla tavalla kuin edellinen ja seuraava-painike tyypillisesti ovat.

Sovelluksessa olen asetellut painikkeet siten, että lomakkeen hyväksymispainike on ensin ja peruuta jälkeen, koska valtaosa kohdeyleisöstä on Windows-käyttäjiä. Lisää kala-lomakkeella en käyttänyt tyhjennä/peruuta-painiketta ollenkaan, sillä tyhjennä-painikkeesta ei yleisesti ole käyttäjälle mitään hyötyä. Lomakkeen tyhjennyksen hyöty ei ole tarpeeksi suuri verrattuna mahdollisen käyttäjän virhepainallukseen, jolloin käyttäjä menettää kaikki täyttämänsä tiedot. Lisää kommentti-lomakkeella käytin sulje-painikketta, koska lomake tulee esiin linkkiä painamalla ja häviää näkyvistä sulje-painikkeesta. Tässä tapauksessa mahdollinen

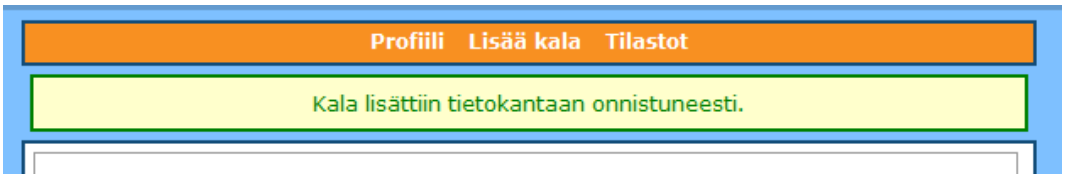
virhepainallus ei ole niin vakava, koska lomakkeessa on vain yksi kenttä ja lomakkeen piilottaminen tuo lisää näyttöalaa saaliiden listaukselle. (Nielsen, 2008; Nielsen, 2000)

## 6.4 Virhetilanteet

Mahdolliset virhetilanteet on syytä ehkäistä hyvällä ohjeistuksella ja selkeällä käyttöliittymällä. Virhetilanteisiin jouduttaessa niistä toipuminen ja käyttäjän ohjeistaminen on ensisijaisen tärkeää.

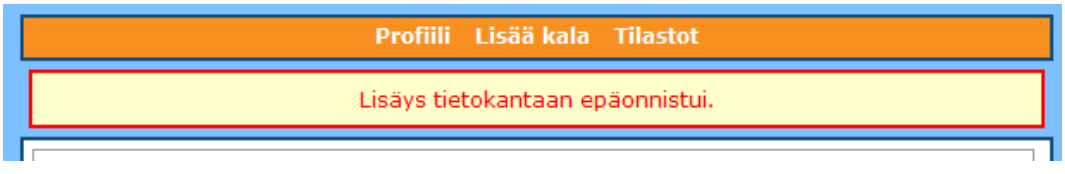
Sovelluksessa on lomakkeiden täytöissä ehkäisty virhetilanteita JavaScript-tarkastuksilla. Tarkastuksen avulla käyttäjälle ilmoitetaan ponnahdusikkunalla mahdolliset virheet lomakkeelle antamissaan tiedoissa, ennen kuin tiedot lähetetään palvelinpuolelle käsiteltäväksi. Näin käyttäjän ei tarvitse täyttää koko lomaketta uudelleen, vain korjata virheellisesti täyttämänsä kenttä. Lisää kala-lomakkeella on kahteen lomakkeen kenttää tehty JavaScriptillä tarkistus, joka sallii vain numeerisia arvoja syötettäväksi kenttiin.

Käyttäjälle annetaan myös palautetta tietokantakäsittelyiden yhteydessä. Esimerkiksi kun käyttäjä lisää saaliin tietokantaan, hän saa seuraavanlaisen ilmoituksen.




**Kuva 11. Viesti onnistuneesta tietokantaan lisäyksestä.**

Lisäyksen epäonnistuessa annetaan siitäkin kuvan 12 mukainen virheviesti.




**Kuva 12. Viesti epäonnistuneesta tietokantaan lisäyksestä.**

Vastaavanlaisia ilmoituksia annetaan kaikista käyttäjän suorittamista toiminnoista, jotka vaativat tietokannan käyttöä.

### 6.5 Miellyttävyys

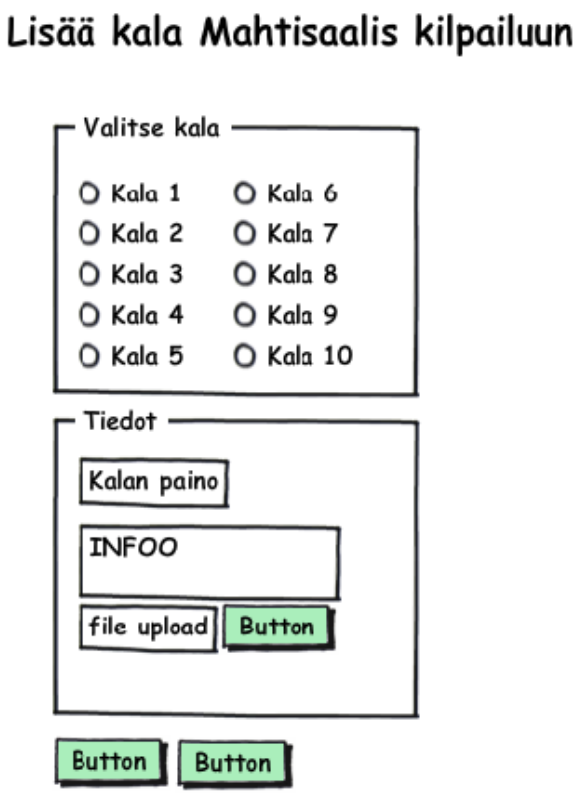
Sovellus on toteutettu käyttäen kalassa.net-sivuston värein. Lisäksi sovelluksen yläosa on toteutettu samalla tavalla kuin muuta kalassa.net-sivut, eli sivuilla on yhtenevät yläosan kuvat sekä linkkipalkki. Näin sivuston käyttäjä kokee olevansa vielä samalla sivustolla, jota on jo tottunut käyttämään.

# 7 Käyttöliittymän suunnittelu ja toteutus

## 7.1 Lisää kala -lomake

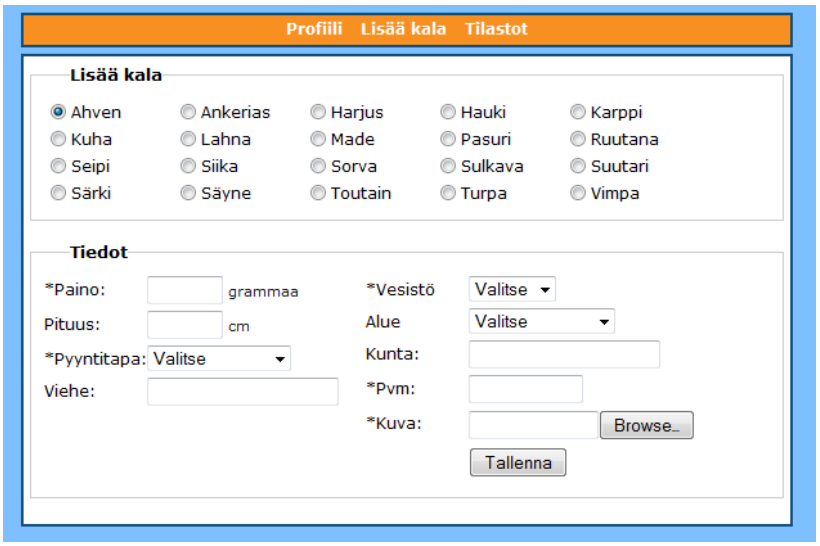
Lomake sisältää perustiedot ja kuvan lisättävää saalista varten. Pakolliset kentät ovat kalalaji, paino, pyyntitapa, vesistö, päivämäärä ja kuva.

Alkuperäisessä käyttöliittymähahmotelmassa pakollisiksi kentiksi suunniteltiin vain kalalaji, paino ja kuva. Lisäksi käyttäjällä on mahdollisuus kirjoittaa lisätietoa saaliistaan textarea-kenttään.




**Kuva 13. Lisää kala -lomakkeen hahmotelma**

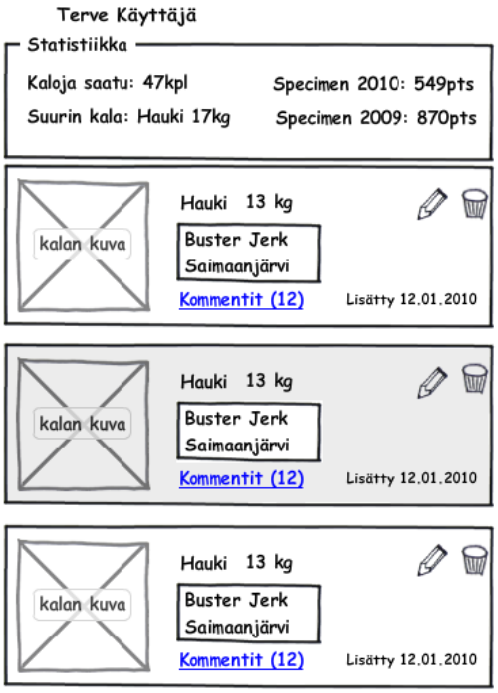
Toteutetusta lomakkeesta jätettiin tyhjennä/peruuta-painike kokonaan pois, koska sen hyöty käytettävyydelle ei ole yhtä suuri kuin mahdollinen haitta. Toteutuneeseen lomakkeeseen lisättiin myös kenttiä tulevien käyttäjien mielipiteiden perusteella. Lomakkeesta jätettiin myös otsikko kokonaan pois sen ilmiselvyyden vuoksi.




**Kuva 14. Toteutettu lisää kala -lomake.**

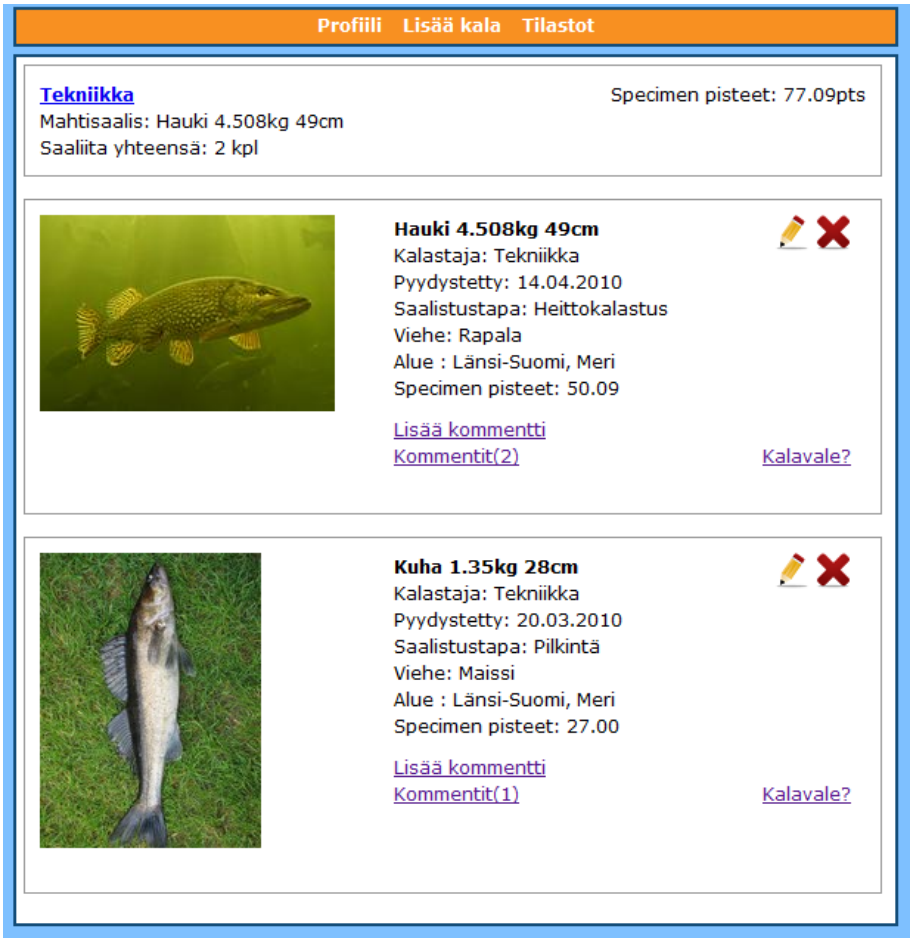
## 7.2 Profiili

Listaus toimii käyttäjän profiili-näkymänä, jossa listataan käyttäjänimi, suurin saalis, saaliiden lukumäärä ja specimen-pisteet. Lisäksi listataan myös kaikki käyttäjän saamat saaliit tietoineen.




**Kuva 15. Profiilinäkymän hahmotelma.**

Toteutuksessa noudatettiin hyvin pitkälti tehtyä suunnitelmaa. Kalan tietoja lisättiin enemmän, mutta muuten tiedot pysyivät samoina.




**Kuva 16. Toteutettu profiilinäkymä.**

## 7.3 Saaliin kommentit

Saaliin kommentit suunniteltiin avattavaksi AJAX-tekniikalla ilman sivunlatausta. Kommentit saadaan painamalla kommentit -linkkiä, joka sisältää myös suluissa kommenttien lukumäärän.

**Kuva 17. Kommenttinäkymän hahmotelma.**

Toteutuksessa noudatettiin alkuperäistä suunnitelmaa. Ilmoita asiaton kommentti-linkki lyhennettiin Asiaton kommentti?-linkiksi. Listaukseen loppuun lisättiin Piilota kommentit-linkki.

**Kuva 18. Toteutettu kommenttinäkymä.**

## 7.4 Näyttöala

Käyttöliittymän toteutuksessa on otettu huomioon myös pienempää resoluutiota käyttävät käyttäjät. Sovellus on toteutettu vain 600 kuvapistettä leveäksi, jotta so-

vellus olisi käytettävissä jopa 640x480 resoluutiolla. Kapea toteutus mahdollistaa myös myöhemmät elementtien lisäykset käyttöliittymään.

# 8 Loppusanat

Opinnäytetyön tekeminen oli aloittamisen vaikeuden jälkeen mukava ja opettavainen prosessi. Tutustuin tarkemmin käytettävyyteen ja sen toteuttamiseen sekä eri web-tekniikoihin. Projekti oli erittäin opettavainen ja tulevaisuuden kannalta varmasti erittäin hyödyllinen.

Aikataulullisesti projekti ei edennyt aivan suunnitelmien mukaisesti ja näin ollen opinnäytetyö valmistui yli kuukauden myöhässä. Suurin syy myöhästymiseen oli aloittamisen vaikeus, silla aloittamisen jälkeen työ valmistui odotettua nopeammin.

Lopputuloksena olen tyytyväinen tekemään työhöni, mutta käytännön osuuteen tekisin muutamia muutoksia jälkeenpäin oppimani perusteella.

# 9 Lähdeluettelo

Conallen, Jim 2003. Building Web Applications with UML 2nd edition.

Fowler, Martin 2004. UML Distilled 3rd edition.

Hovi, Ari 2004. SQL-opas.

ISO.9241. Ergonomic requirements for office work with visual display terminals.

Nielsen, Jakob 2000. Reset and Cancel Buttons [online]. [viitattu 07.04.2010]. Saatavilla www-muodossa <URL: http://www.useit.com/alertbox/20000416.html>

Nielsen, Jakob 2000. Www-suunnittelu

Nielsen, Jakob 2003. Usability 101: Introduction to Usability [online]. [viitattu 30.03.2010]. Saatavilla www-muodossa: <URL: http://www.useit.com/alertbox/20030825.html>.

Nielsen, Jakob 2004. Guidelines for Visualizing Links [online]. [viitattu 03.04.2010]. Saatavilla www-muodossa <URL: http://www.useit.com/alertbox/20040510.html >

Nielsen, Jakob 2008. Ok-Cancel or Cancel-Ok? [online]. [viitattu 03.04.2010]. Saatavilla www-muodossa: <URL: http://www.useit.com/alertbox/ok-cancel.html>

Peltomäki, Juha; Nykänen, Ossi 2006. Web-selainohjelmointi

PHP.net, 2010. History of PHP [online]. [viitattu 28.3.2010]. Saatavilla www-muodossa: <URL: http://fi2.php.net/manual/en/history.php.php>.

PHP.net, 2010a. What is PHP [online]. [viitattu 10.04.2010]. Saatavilla www-muodossa: <URL: http://fi.php.net/manual/en/intro-whatis.php>.

PHP.net, 2010b. htmlentities [online]. [viitattu 15.02.2010]. Saatavilla www-muodossa: <URL: http://php.net/manual/en/function.htmlentities.php>

Rantala, Ari 2005. Web-ohjelmointi.

Wellington, Luke; Thomson, Laura 2008. PHP & MySQL Web Development 4th edition.

Wiio, Antti 2004. Käyttäjäystävällisen sovelluksen suunnittelu.