

# Painontarkkailu

---

## Tietokantasovellus -harjoitustyö

Tietojenkäsittelytieteen laitos  
Syksy 2011  
Hannu Päiveröinen  
[hannu.paiveroinen@helsinki.fi](mailto:hannu.paiveroinen@helsinki.fi)  
014023633

## Sisältö

Suunnitteludokumentti .....	1
1. JOHDANTO.....	1
2. YLEISKUVA JÄRJESTELMÄSTÄ.....	1
3. KÄYTTÖTAPAUKSET.....	2
4. JÄRJESTELMÄN TIETOSISÄLTÖ.....	3
5. KÄYTTÖLIITTYMÄN HAHMOTELMA.....	4
6. RELAATIOTIETOKANTAKAAVIO .....	5
Toteutusdokumentti .....	6
1. JOHDANTO.....	6
2. OHJELMISTON YLEISRAKENNE .....	6
3. JÄRJESTELMÄN KOMPONENTIT .....	7
4. SQL CREATE TABLE-LAUSEET .....	10
5. ASENNUSTIEDOT.....	11
6. KÄYNNISTYS- / KÄYTTÖOHJE.....	11
7. LIITTEET .....	11

# Suunnitteludokumentti

## 1. JOHDANTO

- **Järjestelmän tarkoitus**

Tehtävänä on laatia painontarkkailua tukeva järjestelmä. Järjestelmän avulla käyttäjä voi laskea nauttimansa kalorimäärän ja päivittäin tarvitsemansa kalorit. Tietokantaan sisältyy myös tietoja eri aktiviteettien energiankulutuksesta. Järjestelmää voivat käyttää sekä satunnaisasiakkaat että rekisteröityneet asiakkaat.

Ruoat, raaka-aineet ja aktiviteetit on haun helpottamiseksi kyettävä luokittelemaan erilaisiin ryhmiin. Tietty kohde voi kuulua useaan ryhmään. Järjestelmässä tullaan kiinnittämään erityistä huomiota järjestelmän käytettävyyteen.

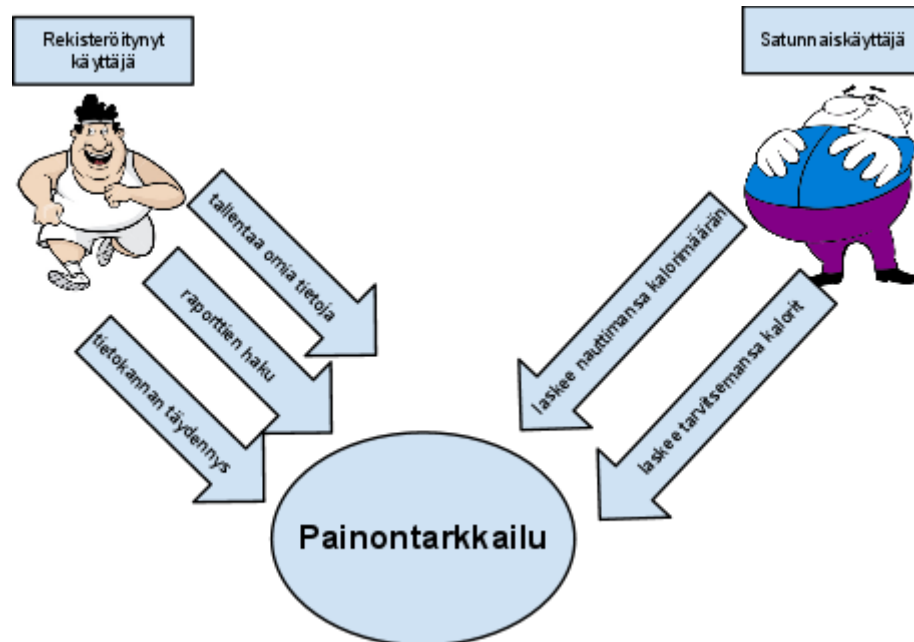
- **Toimintaympäristö**

Toimintaympäristönä web.

- **Toteutusympäristö**

Järjestelmä toteutetaan käyttämällä NetBeanssia, kielenä Java.

## 2. YLEISKUVA JÄRJESTELMÄSTÄ



Kuva:1: Sidosryhmäkaavio

## **Käyttäjäryhmät:**

Määritellään sidosryhmäkaaviossa esiintyvät käyttäjät:

- *Satunnaiskäyttäjällä tarkoitetaan ketä tahansa, joka Internetin välityksellä tulee lukemaan Painontarkkailu-sivuja. Satunnaiskäyttäjä voi käyttää sekä nautittujen että kulutettujen kalorien laskentaosia.*
- *Rekisteröitynyt käyttäjä omistaa samat oikeudet kuin satunnaiskäyttäjä. Lisäksi rekisteröitynyt käyttäjä voi tallentaa päiväkirjatyypistä tietoja omasta energiansaannistaan ja -kulutuksestaan sekä saada näihin liittyviä raportteja. Rekisteröitynyt käyttäjä voi myös täydentää tietokantaa lisäämällä sinne uusia ruokia, raaka-aineita tai aktiviteetteja*

## **3. KÄYTTÖTAPAUKSET**

### **Satunnaiskäyttäjän käyttötapaukset:**

*Sivuston infon lukeminen*

- *Kuka tahansa voi käydä lukemassa palvelun yleistietoja*

*Laskurit*

- *Käyttäjällä on mahdollisuus käyttää usean tyyppisiä laskureita*
- *Esimerkki: Kuinka paljon on käyttäjän päivittäinen kulutus?*

*Rekisteröityminen asiakkaaksi*

- *Rekisteröitymisessä käyttäjä ilmoittautuu palvelun käyttäjäksi päästen näin palvelun muiden asiakaspalveluiden käyttäjäksi. Satunnaiskäyttäjästä tulee rekisteröitynyt käyttäjä. Hän antaa henkilötietonsa ja saa asiakastunnuksen, joka on avain asiakkaille tarkoitettuihin palveluihin. Prototyypissä asiakastunnus aktivoituu heti.*

### **Rekisteröityneen käyttäjän käyttötapaukset:**

*Satunnaiskäyttäjän oikeudet*

- *Rekisteröityneellä käyttäjällä on samat käyttöoikeudet kuin satunnaiskäyttäjällä*

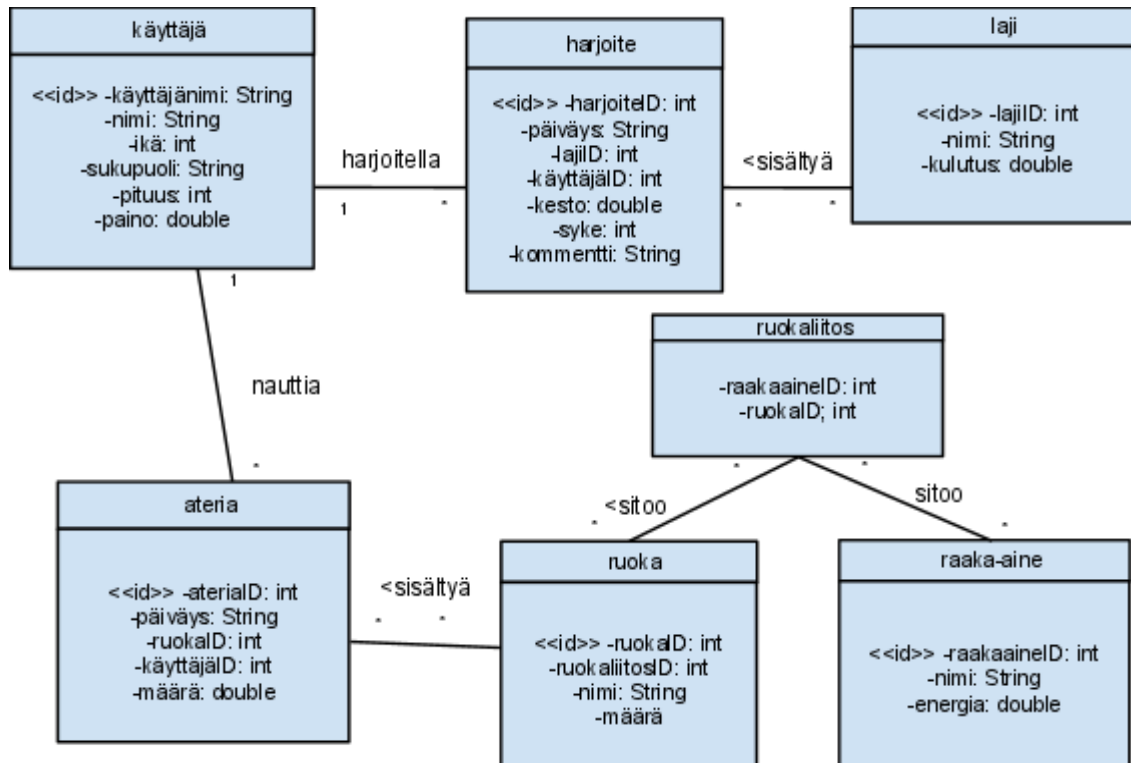
*Tietokannan päivitys*

- *Käyttäjä voi täyttää lomakkeen, jolla päivitetään tietokantaan uusi ruoka, raaka-aine tai aktiviteetti*

*Raportin haku*

- *Käyttäjä voi pyytää tilastollisia raportteja kulutuksestaan*

## 4. JÄRJESTELMÄN TIETOSISÄLTÖ

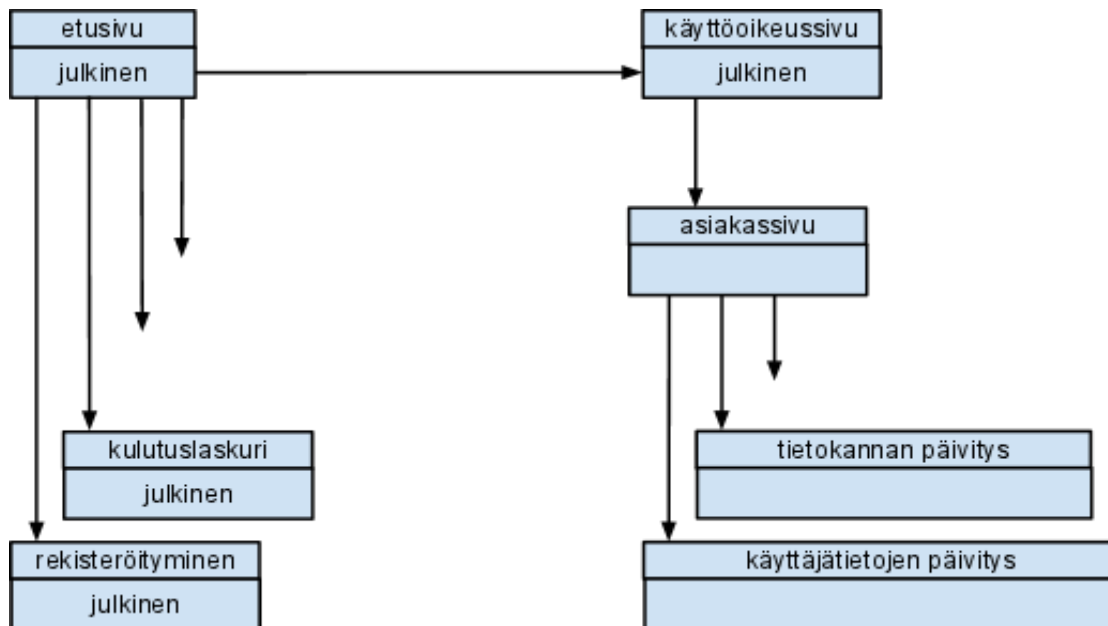


Kuva 2: **Käsitekaavio**

Tietokohde: käyttäjä		
Attribuutti	Arvojoukko	Kuvailu
käyttäjänimi	String	Käyttäjä luo yksilöllisen käyttäjänimen, joka toimii attribuutin avaimena
nimi	String	Käyttäjän oma nimi
ikä	int	Ikä kokonaislukuina
sukupuoli	String	Nainen / Mies
pituus	int	Pituus senttimetreinä
paino	double	Paino 100 gramman tarkkuudella

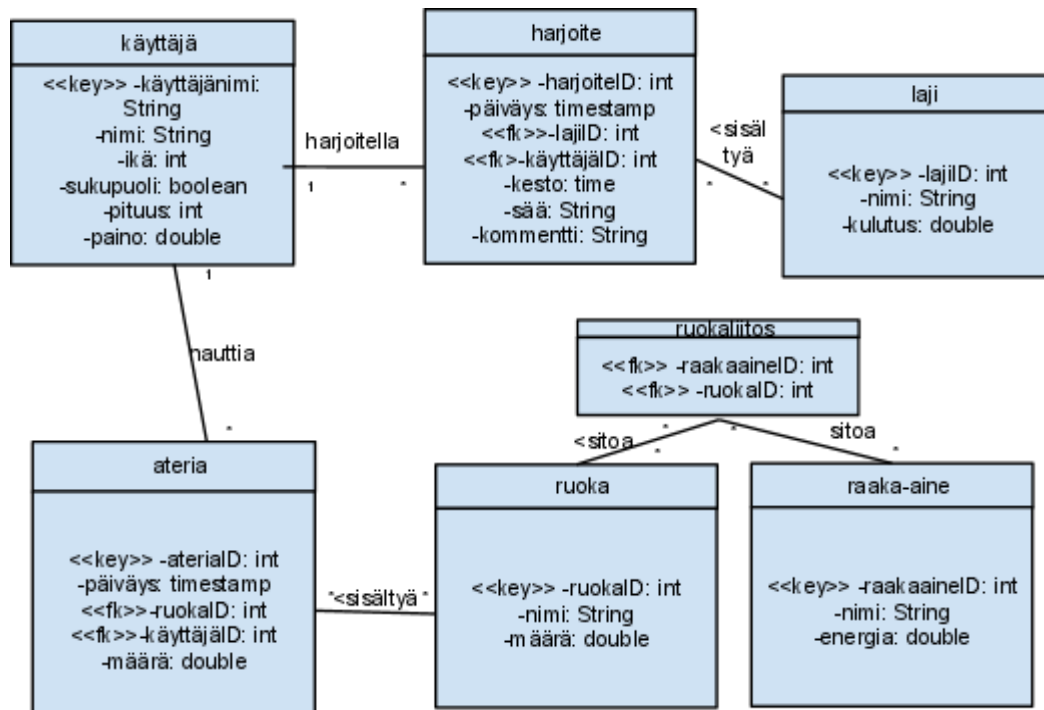
Käyttäjätiedot luodaan rekisteröinnin yhteydessä. Tietoja tullaan päivittämään ajan myötä. Käyttäjä identifoidaan henkilökohtaisella käyttäjänimellä.

## 5. KÄYTTÖLIITTYMÄN HAHMOTELMA



Kuva 3: **Käyttöliittymän sivujen väliset yhteydet**

## 6. RELAATIOTIETOKANTAKAAVIO



Kuva 4: Relaatietietokantakaavio kuvana

# Toteutusdokumentti

## 1. JOHDANTO

- **Järjestelmän tarkoitus**

Tehtävänä on laatia painontarkkailua tukeva järjestelmä. Järjestelmän avulla käyttäjä voi laskea nauttimansa kalorimäärän ja kuluttamansa kalorit.

Tietokantaan sisältyy myös tietoja eri aktiviteettien energiankulutuksesta.

Järjestelmää voivat käyttää sekä satunnaisasiakkaat että rekisteröityneet asiakkaat.

Järjestelmässä tullaan kiinnittämään erityistä huomiota järjestelmän käytettävyyteen.

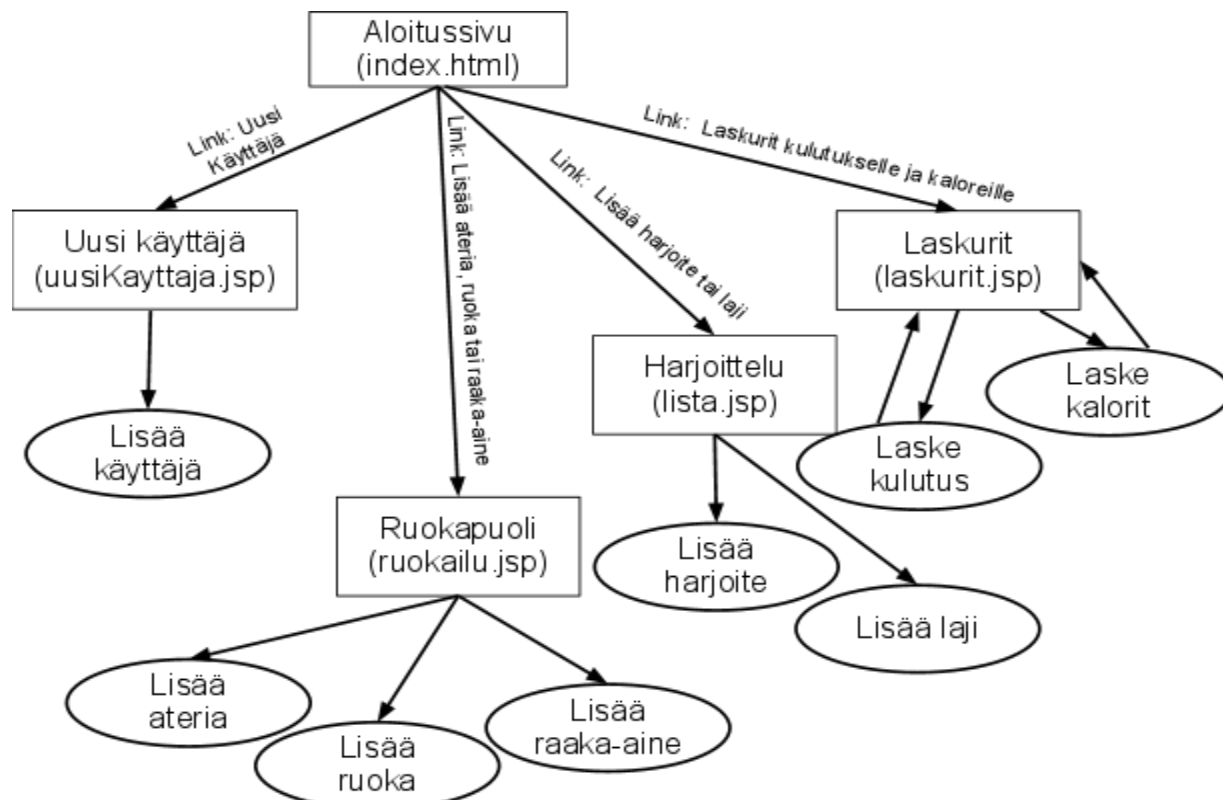
- **Toimintaympäristö**

Toimintaympäristönä web.

- **Toteutusympäristö**

Järjestelmä toteutetaan käyttämällä NetBeansia, kielenä Java.

## 2. OHJELMISTON YLEISRAKENNE



- Kuva 5: Rakennekaavio



### 3. JÄRJESTELMÄN KOMPONENTIT

#### **aloitussivu**

Staattinen html-sivu, tiedosto: index.xhtml.

Sivulla on lyhyt yleisesite järjestelmästä sekä linkit, joiden avulla pääsee käsiksi järjestelmän tarjoamiin palveluihin:

[Uusi Käyttäjä](#)

johtaa uuden asiakkaan tietoja kyselevälle rekisteröintisivulle (uusiKayttaja.jsp)

[Lisää harjoite tai laji](#)

johtaa uusia harjoitteita ja lajeja kyselevälle harjoitesivulle (lista.jsp)

[Lisää ateria, ruoka tai raaka-aine](#)

johtaa uusia aterioita, ruokia ja raaka-aineita kyselevälle ruokailusivulle (Ruokailu.jsp)

[Laskurit kulutukselle ja kaloreille](#)

johtaa kulutuksen ja kaloreiden laskemiseen sisältävien lomakkeiden kyselysivulle (Laskurit.jsp)

#### **käyttäjän lisääminen**

Staattinen html-sivu, tiedosto: uusiKayttaja.jsp.

Sivulla on lomake, jolla lisätään uusi käyttäjä. Lomakkeen tarkistaa lisaaKayttajaServlet.

[Uusi käyttäjä](#)

käyttäjänimi

nimi

ikä: annetaan kokonaislukuna

sukupuoli : valitaan radiobuttonilla

pituus : annetaan kokonaislukuna

paino: annetaan reaalilukuna

#### **ruokapuoli**

Staattinen html-sivu, tiedosto: ruokailu.jsp.

Sivulla on kolme lomaketta. Lomakkeiden oikeellisuus tarkistetaan erillisillä servleteilla:

[Uusi ateria](#)

Käyttäjä voi lisätä uuden aterian lomakkeella. Lomakkeen tarkistaa lisaaAteriaServlet:

käyttäjä : Haetaan pudotusvalikosta

ruoka: Haetaan pudotusvalikosta

päiväys: Esitäytetty käyttöpäivämäärällä. Voidaan muuttaa.

määrä: Kuinka monta annosta nautitaa. Tulee olla reaaliluku.

[Uusi ruoka](#)

Käyttäjä voi lisätä uuden ruoan lomakkeella. Lomakkeen tarkistaa lisaaRuokaServlet

nimi: Ruoan nimi.

määrä: Kuinka paljon raaka-ainetta tulee ruokaan. Tulee olla reaaliluku.

raaka-aine: Valitaan listalta raaka-aine.

#### Uusi raaka-aine

Käyttäjä voi lisätä uuden raaka-aineen lomakkeella. Lomakkeen tarkistaa lisääRaakaaineServlet

nimi: Raaka-aineen nimi.

energia: Raaka-aineen energiamäärä. Tulee olla reaaliluku.

### **liikuntapuoli**

Staattinen html-sivu, tiedosto: lista.jsp.

Sivulla on kolme lomaketta. Lomakkeiden oikeellisuus tarkistetaan erillisillä servleteilla:

#### Uusi harjoite

Käyttäjä voi lisätä uuden harjoitteen lomakkeella. Lomakkeen tarkistaa lisääHarjoiteServlet

käyttäjä : Haetaan pudotusvalikosta

laji: Haetaan pudotusvalikosta

kesto: Harjoitteen kesto. Annetaan reaalilukuna.

päiväys: Esitäytetty käyttöpäivämäärällä. Voidaan muuttaa.

keskimääräinen syke: Harjoitteen syke. Annetaan kokonaislukuna.

kommentti: Vapaa kommentointi

#### Uusi laji

Käyttäjä voi lisätä uuden lajin lomakkeella. Lomakkeen tarkistaa lisääLajiServlet

nimi : Lajin nimi

kulutus: Lajin kulutus. Annetaan reaalilukuna.

### **laskurit**

Staattinen html-sivu, tiedosto: laskurit.jsp.

Sivulla on kaksi lomaketta. Lomakkeiden oikeellisuus tarkistetaan erillisillä servleteilla ja servelit ovat vastuussa laskureiden toiminnasta:

#### Laske kulutus

Käyttäjä voi laskea harjoitteen kulutuksen. Lomakkeen tarkistaa lisääKulutusServlet

käyttäjä : Haetaan pudotusvalikosta, jonka jälkeen käyttäjän tiedot voi päivittää lomakkeelle

ikä: käyttäjän ikä: ilmoitetaan kokonaisluvulla.

sukuoli: valitaan radiobuttonista

pituus: käyttäjän pituus. Ilmoitetaan kokonaisluvulla

paino: käyttäjän paino. Ilmoitetaan reaaliluvulla.

laji: Haetaan pudotusvalikosta

kesto: Harjoitteen kesto. Annetaan reaalilukuna.

keskimääräinen syke: Harjoitteen syke. Annetaan kokonaislukuna.

kommentti: Vapaa kommentointi

#### Laske kalorit

Käyttäjä voi laskea aterian kalorimäärän. Lomakkeen tarkistaa lisääKaloritServlet

ruoka : Haetaan pudotusvalikosta

määrä: Annosten määrä. Ilmoitetaan reaaliluvulla.

### **Laske kulutus servletti**

Servletti tarkistaa Laskurit-sivulla kulutuslomakkeelta annetut tiedot ja niiden perusteella laskee ja palauttaa kulutuksen.

### **Laske kalorit servletti**

Servletti tarkistaa Laskurit-sivulla kalorilomakkeelta annetut tiedot ja niiden perusteella laskee ja palauttaa kalorit.

### **Laskurit servletti**

Servletti antaa Laskurit-sivulle arvoja.

### **LisaaAteria servletti**

Servletti tekee ruokailu-sivuston Lisää Ateria-lomakkeen arvoille oikeellisuustarkistuksen ja lisää aterian tietokantaan.

### **LisaaHarjoite servletti**

Servletti tekee lista-sivuston Lisää Harjoite-lomakkeen arvoille oikeellisuustarkistuksen ja lisää harjoitteen tietokantaan.

### **LisaaKayttaja servletti**

Servletti tekee uusikayttaja-sivuston Lisää Käyttäjä-lomakkeen arvoille oikeellisuustarkistuksen ja lisää käyttäjän tietokantaan.

### **LisaaLaji servletti**

Servletti tekee lista-sivuston Lisää Laji-lomakkeen arvoille oikeellisuustarkistuksen ja lisää lajin tietokantaan.

### **LisaaRaakaaine servletti**

Servletti tekee ruokailu-sivuston Lisää Raaka-aine-lomakkeen arvoille oikeellisuustarkistuksen ja lisää raaka-aineen tietokantaan.

### **LisaaRuoka servletti**

Servletti tekee ruokailu-sivuston Lisää Ruoka-lomakkeen arvoille oikeellisuustarkistuksen ja lisää ruoan tietokantaan.

### **Lista servletti**

Servletti antaa Lista-sivulle arvoja.

### **PaivitaKaytaja servletti**

Servletti päivittää laskurit sivulla olevan laskekulutus-lomakkeen arvot.

### **Ruokailu servletti**

Servletti antaa Ruokailu-sivulle arvoja.

### **UusiKayttaja servletti**

Servletti antaa uusiKayttaja-sivulle arvoja.

## 4. SQL CREATE TABLE-LAUSEET

```
CREATE TABLE Ateria (  
    AterialD long NOT NULL,  
    date varchar(50) NOT NULL,  
    määrä numeric(20) NOT NULL,  
    PRIMARY KEY (AterialD),  
    FOREIGN KEY (Kayttaja) REFERENCES Kayttaja(KayttajaID),  
    FOREIGN KEY (Ruoka) REFERENCES Ruoka(RuokaID)  
);
```

```
CREATE TABLE Harjoite (  
    HarjoiteID long NOT NULL,  
    kestoMinuuteissa numeric(20) NOT NULL,  
    kommentti varchar(100) NOT NULL,  
    syke numeric(20) NOT NULL,  
    PRIMARY KEY (HarjoiteID),  
    FOREIGN KEY (Kayttaja) REFERENCES Kayttaja(KayttajaID),  
    FOREIGN KEY (Laji) REFERENCES Laji(LajiID)  
);
```

```
CREATE TABLE Kayttaja (  
    KayttajaID long NOT NULL,  
    kayttajanimi varchar(50) NOT NULL,  
    nimi varchar(50) NOT NULL,  
    ikä numeric(20) NOT NULL,  
    sukupuoli varchar(10) NOT NULL,  
    pituus numeric(30) NOT NULL,  
    paino numeric(20) NOT NULL,  
    PRIMARY KEY (AterialD),  
);
```

```
CREATE TABLE Laji (  
    LajiID long NOT NULL,  
    nimi varchar(50) NOT NULL,  
    kulutus numeric(20) NOT NULL,  
    PRIMARY KEY (AterialD),  
);
```

```
CREATE TABLE Raakaaine (  
    RaakaaineID long NOT NULL,  
    nimi varchar(50) NOT NULL,  
    energia numeric(20) NOT NULL,  
    PRIMARY KEY (AterialD),
```

);

```
CREATE TABLE Ruoka(  
    RuokaID long NOT NULL,  
    nimi varchar(50) NOT NULL,  
    maara numeric(20) NOT NULL,  
    PRIMARY KEY (RuokaID),  
);
```

```
CREATE TABLE Ruokaliitos (  
    RuokaliitosID long NOT NULL,  
    PRIMARY KEY (RuokaliitosID),  
    FOREIGN KEY (Raakaaine) REFERENCES Raakaaine(RaakaaineID),  
    FOREIGN KEY (Ruoka) REFERENCES Ruoka(RuokaID)  
);
```

## 5. ASENNUSTIEDOT

- Projekti on kansiohierarkiassa. Esimerkiksi netbeanssilla voi aukaista projektin, luoda tietokantayhteyden ja serverin.

## 6. KÄYNNISTYS- / KÄYTTÖOHJE

- Käynnistys suoritetaan netbeanssilla suoraan projektista.

## 7. LIITTEET

- Tehtäväkuvaus löytyy osoitteesta:  
<http://www.cs.helsinki.fi/u/laine/infoht/aiheet/ik98hl5.html>
- Java documentin pikakuvake on päähakemiston alla: Painontarkkailu-javadoc