**Suunnittelukuvasto**

Websovellus

Fillariverstas

Ihmeidentekijät

Versio 0.8 Luonnos

Tekijä Ari Lintukorpi 13.5.2010

Jan Nyman

Jori Sivonen

Jouko Rahikainen

Anssi Metsäranta

Tarkastanut Anssi Metsäranta

Hyväksynyt

MUUTOSHISTORIA

|  |  |  |
| --- | --- | --- |
| versionumero | pvm | muutos |
| 0.1 | 5.4.2010 | Luonnos |
| 0.2 | 9.4.2010 | Lisätty materiaalia |
| 0.3 | 15.4.2010 | Lisätty kaavioita ja sisällysluettelo |
| 0.4 | 16.4.2010 | Lisätty kaavioita ja korjattu puutteita |
| 0.5 | 19.4.2010 | Lisätty paketit ja korjattu katselmoinnissa havaittuja virheitä |
| 0.6 | 30.4.2010 | Lisätty paketteja |
| 0.7 | 5.5.2010 | Lisätty paketteihin lisää tietoa |
| 0.8 | 13.5.2010 | Lisätty lisää paketteja ja tietoa |

Sisällysluettelo

[1 Johdanto 4](#_Toc261344092)

[2 Käyttöliittymät 4](#_Toc261344093)

[3 Oliomalli 8](#_Toc261344094)

[4 Relaatiotietokanta 10](#_Toc261344095)

[4.1 Relaatiokaavat 10](#_Toc261344096)

[4.2 Relaatioiden luontilauseet 13](#_Toc261344097)

[5.1 Kirjautuminen osakäyttötapaus 17](#_Toc261344098)

[5.2 Pyörän suunnittelu osakäyttötapaus 18](#_Toc261344099)

[5.3 Merkin valinta osakäyttötapaus 18](#_Toc261344100)

[6 Viestiyhteyskaaviot 19](#_Toc261344101)

[6.1 Kirjautuminen viestiyhteistyökaavio 19](#_Toc261344102)

[6.2 Suunnittelu viestiyhteistyökaavio 22](#_Toc261344103)

[6.3 Osan valinta viestiyhteistyökaavio 27](#_Toc261344104)

[7 Paketit 31](#_Toc261344105)

[7.1 Kohdeluokat-paketti 32](#_Toc261344106)

[7.2 Servletit-paketti 40](#_Toc261344107)

[7.3 Tietokanta-paketti 43](#_Toc261344108)

# 1 Johdanto

Tässä dokumentissa on kuvattu Fillariverstas Oy:lle toteuttavan Websovelluksen suunnitelma. Sovellus mahdollistaa polkupyörien suunnittelun ja tarjouspyynnön lähettämisen. Tämä dokumentti sisältää käyttöliittymäsuunnitelman ja ohjelmiston rakenteen kuvauksen. Lisäksi dokumentissa kuvataan yhteistyö- ja viestintäyhteyskaaviot ja tietokannan rakenne. Dokumentti on kirjoitettu määrityskuvaston pohjalta.

Ensimmäisen iteraation aikana suunnitellaan ja toteutetaan asiakkaiden kirjautuminen järjestelmään ja rungon valinta. Toisen iteraation aikana suunnitellaan ja toteutetaan polkupyörän suunnittelu ja tarjouspyynnön lähettäminen.

# 2 Käyttöliittymät

Järjestelmän käyttöliittymä toteutetaan Internetselainpohjaisena. Käyttöliittymät koodataan xhtml-ilmaisukieltä käyttäen ja JSP-tekniikkaa (JavaServer Pages) ja JSTL (JavaServer Pages Standard Tag Library) standardi tagikirjastoa noudattaen. Käyttöliittymätiedostot talletetaan jsp-tarkenteilla.

Toteutettavassa kokonaisuudessa on neljä näkymää:

* kirjautumis-näyttö, kirjautuminen.jsp ja
* pyörän suunnittelu, suunnittelu-näyttö, suunnittelu.jsp.
* suunnitelty pyörä, suunniteltuPyora.jsp
* tarjouspyyntö lähetetty, kiitos.jsp

Kun käyttäjä käynnistää ohjelman, käyttöliittymän näkymä näyttää kuvan 2.1 Kirjautuminen – kirjautuminen.jsp mukaiselta.



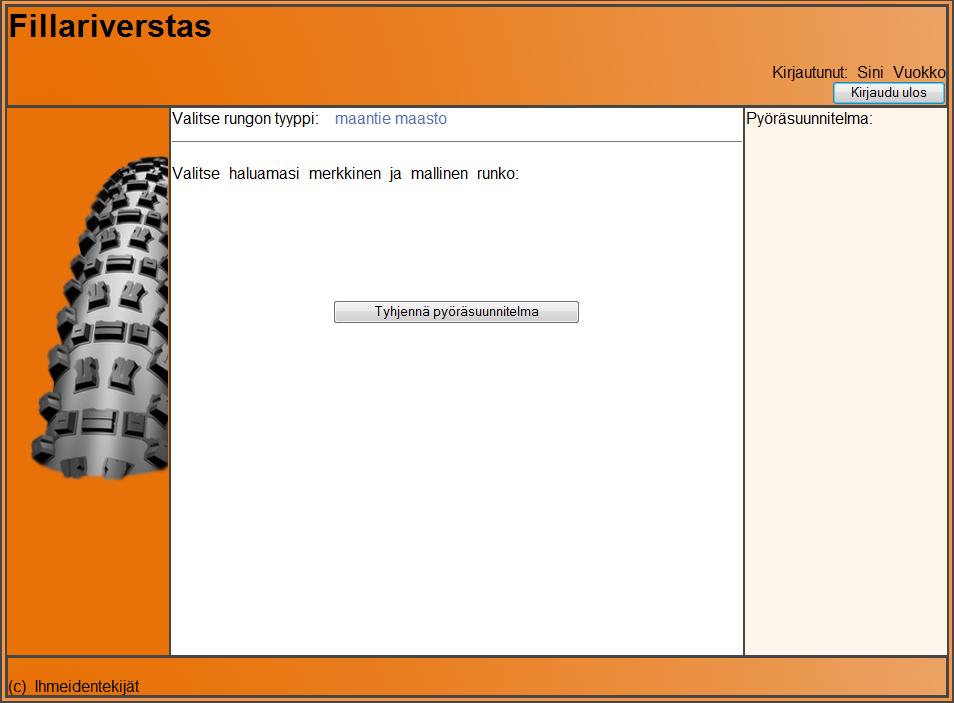
Kuva 2.1. Kirjautuminen – kirjautuminen.jsp

Käyttäjä kirjoittaa kirjautuminen-näytön kenttiin sähköpostiosoitteensa, joka toimii käyttäjätunnuksena ja salasanan. Kun asiakas painaa kirjaudu-painiketta järjestelmä tarkistaa onko käyttäjätunnusta olemassa ja onko salasana syötetty oikein. Jos ehdot täyttyivät, asiakkaan kirjautuminen onnistuu.

Jos käyttäjätunnusta ei löydy tietokannasta, järjestelmä kirjoittaa kirjautumis-näytölle tekstin: ”Väärä käyttäjätunnus tai salasana!”.

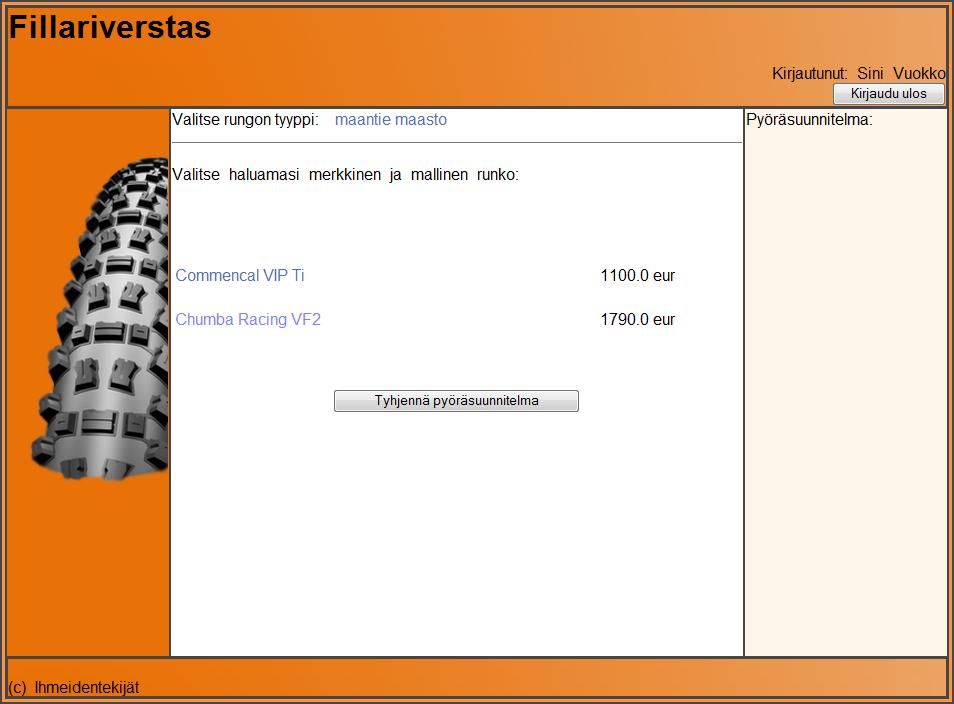
Jos tietokannan käsittelyssä tapahtui virhe, järjestelmä palauttaa käyttäjän kirjautumis-näytölle.

Kirjautumisen tarkempi toiminta on kerrottu kohdassa 5.1 Kirjautuminen osakäyttötapaus.



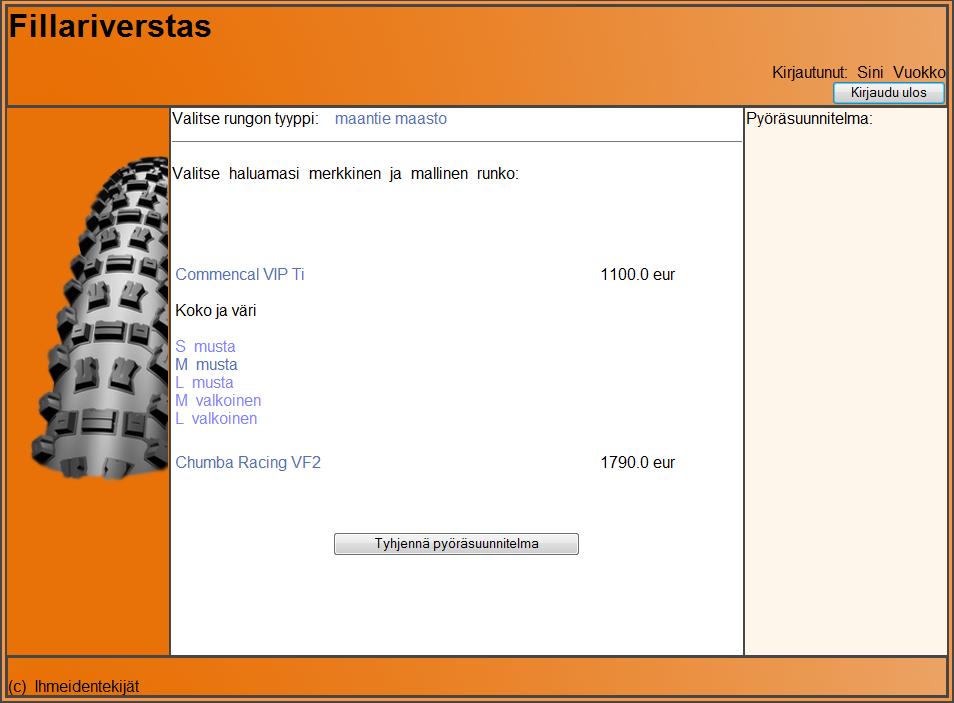
Kuva 2.2. Suunnittelu, osan tyypin valinta – suunnittelu.jsp

Tyypin valinnan tarkempi kuvaus on kerrottu kohdassa 5.2 Pyörän suunnittelu osakäyttötapaus.

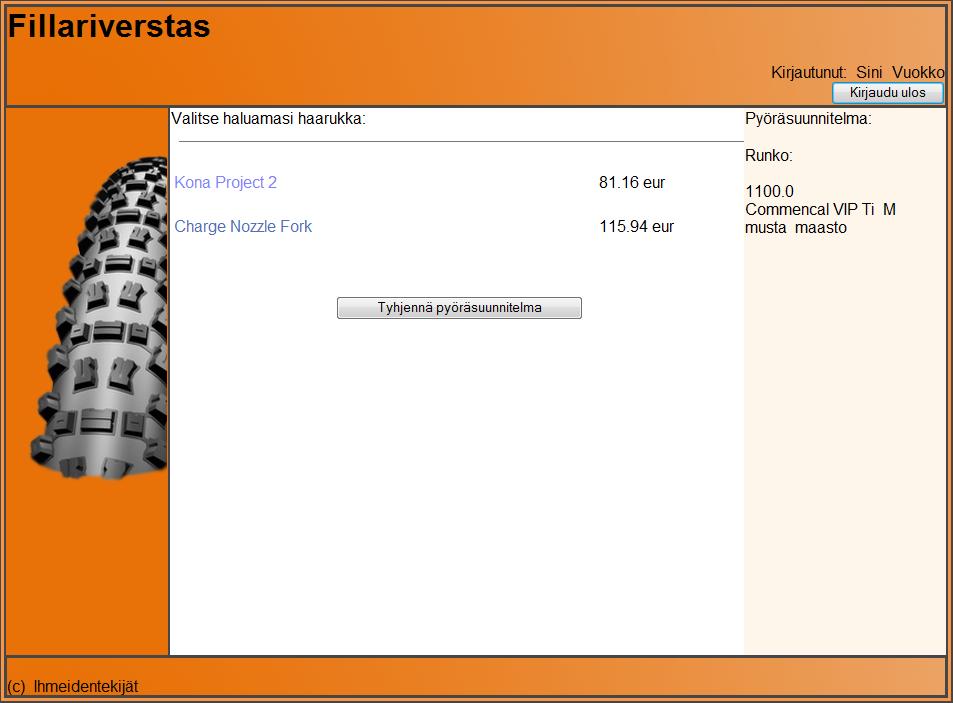


Kuva 2.3. Suunnittelu, osan tyyppi valittu – suunnittelu.jsp

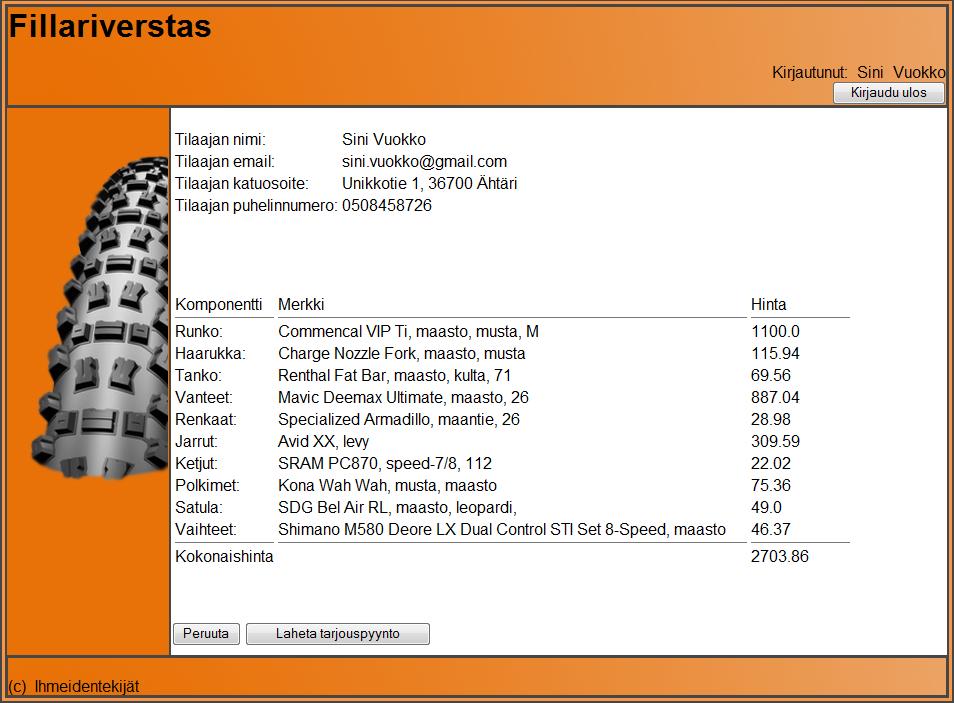
Merkin valinnan tarkempi kuvaus on kerrottu kohdassa 5.3 Merkin valinta osakäyttötapaus.



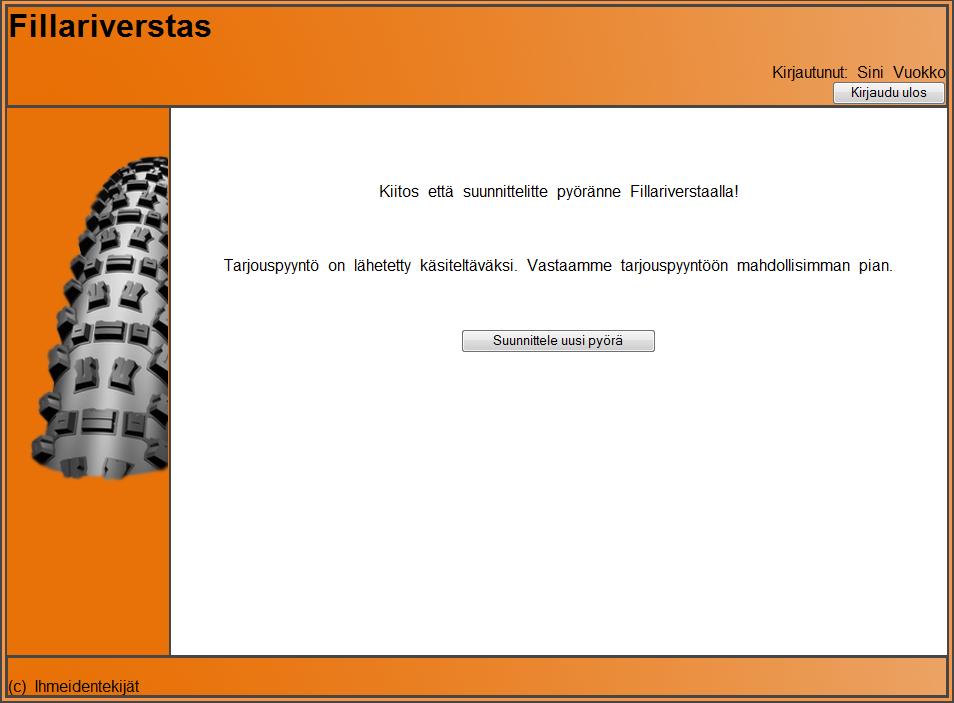
Kuva 2.4. Suunnittelu, osan valinta – suunnittelu.jsp



Kuva 2.5. Suunnittelu, osa valittu – suunnittelu.jsp



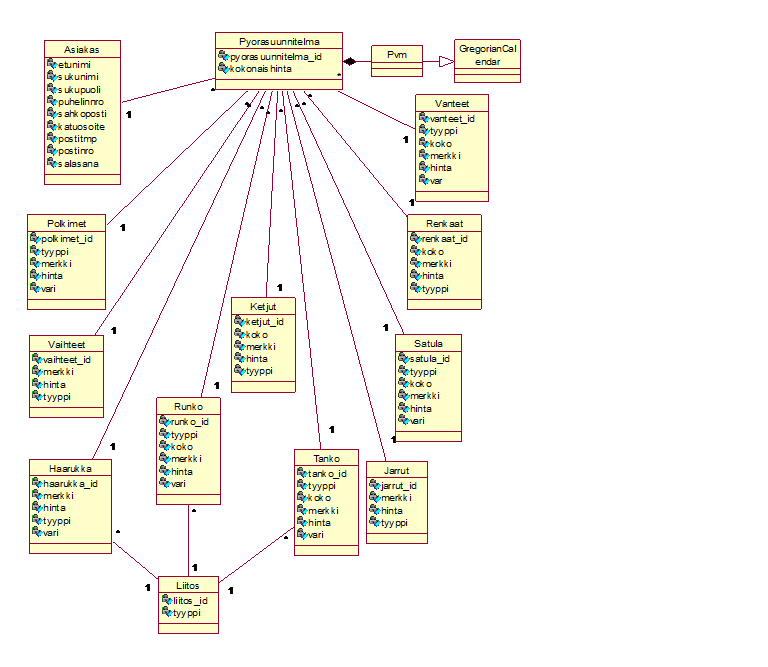
Kuva 2.6. Suunnittelu, tarjouspyynnön lähetys – suunniteltuPyora.jsp



Kuva 2.7. Suunnittelu, tarjouspyyntö lähetetty – kiitos.jsp

# 3 Oliomalli

Tietojärjestelmän luokkakaavio on kuvattu kuvassa 3.1 Luokkakaavio. Kirjautumisen toimintaa ohjaa Java-servletti nimeltään Kirjautuminen.java. Kirjautuminen toimii käyttöliittymien (kirjautuminen.jsp ja suunnittelu.jsp) ja liiketoimintaluokkien välillä. Se huolehtii ohjelman toiminta­logiikasta ja toiminnasta. Se tutkii käyttäjän tekemät valinnat ja ohjaa sovelluksen eri toimintavaihto­eh­toi­hin. Se muodostaa sovelluksen toiminnassa tarvittavat liiketoimintaoliot käyttäjän antamien syötteiden perusteella.

 Kuva 3.1. Luokkakaavio

# 4 Relaatiotietokanta

Relaatiotietokannan relaatiokaavio on kuvattu kuvassa Kuva 4.1 Tietokannan relaatiokaavio. Pyöräsuunnitelma-relaation ja muiden relaatioiden välillä on yhteyssuhde 

Kuva 4.1. Tietokannan relaatiokaavio.

## 4.1 Relaatiokaavat

Relaatiokaavat on kuvattu kuvassa 4.2 Relaatiokaavat. Relaation pääavain on alleviivattu ja viiteavain on kursivoitu.

ASIAKAS(

sahkoposti : VARCHAR(50) NOT NULL,

salasana : VARCHAR(20) NOT NULL,

etunimi : VARCHAR(20) NOT NULL,

sukunimi : VARCHAR(30) NOT NULL,

sukupuoli : VARCHAR(20) NOT NULL,

puhelinnumero : VARCHAR(20) NOT NULL,

katuosoite : VARCHAR(20) NOT NULL,

postitmp : VARCHAR(20) NOT NULL,

postinro : VARCHAR(20) NOT NULL,

)

PYORASUUNNITELMA(

pyorasuunnitelma\_id : INTEGER NOT NULL,

kokonaishinta : DECIMAL(8,2) NOT NULL,

pvm : DATE NOT NULL,

*asiakas\_id* : VARCHAR(20) NOT NULL,

*polkimet\_id* : INTEGER NOT NULL,

*vaihteet\_id* : INTEGER NOT NULL,

*haarukka\_id* : INTEGER NOT NULL,

*runko\_id* : INTEGER NOT NULL,

*ketjut\_id* : INTEGER NOT NULL,

*tanko\_id* : INTEGER NOT NULL,

*jarrut\_id* : INTEGER NOT NULL,

*satula\_id* : INTEGER NOT NULL,

*renkaat\_id* : INTEGER NOT NULL,

*vanteet\_id* : INTEGER NOT NULL,

)

VANTEET(

vanteet\_id : INTEGER NOT NULL,

tyyppi : VARCHAR(30) NOT NULL,

koko : INTEGER NOT NULL,

merkki : VARCHAR(100) NOT NULL,

hinta : DECIMAL(8,2) NOT NULL,

)

RENKAAT(

renkaat\_id : INTEGER NOT NULL,

tyyppi : VARCHAR(30) NOT NULL,

koko : INTEGER NOT NULL,

merkki : VARCHAR(100) NOT NULL,

hinta : DECIMAL(8,2) NOT NULL,

)

SATULA(

satula\_id : INTEGER NOT NULL,

vari : VARCHAR(20) NOT NULL,

tyyppi : VARCHAR(30) NOT NULL,

koko : VARCHAR(1) NOT NULL,

merkki : VARCHAR(100) NOT NULL,

hinta : DECIMAL(8,2) NOT NULL,

)

JARRUT(

jarrut\_id : INTEGER NOT NULL,

tyyppi : VARCHAR(30) NOT NULL,

merkki : VARCHAR(100) NOT NULL,

hinta : DECIMAL(8,2) NOT NULL,

)

KETJUT(

ketjut\_id : INTEGER NOT NULL,

tyyppi : VARCHAR(30) NOT NULL,

lenkit\_lkm : INTEGER NOT NULL,

merkki : VARCHAR(100) NOT NULL,

hinta : DECIMAL(8,2) NOT NULL,

)

TANKO(

tanko\_id :INTEGER NOT NULL,

vari : VARCHAR(20) NOT NULL,

tyyppi : VARCHAR(30) NOT NULL,

koko : INTEGER NOT NULL,

merkki : VARCHAR(100) NOT NULL,

hinta : DECIMAL(8,2) NOT NULL,

*liitos\_id* : INTEGER NOT NULL,

)

RUNKO(

runko\_id : INTEGER NOT NULL,

vari : VARCHAR(20) NOT NULL,

tyyppi : VARCHAR(30) NOT NULL,

koko : VARCHAR(1) NOT NULL,

merkki : VARCHAR(100) NOT NULL,

hinta : DECIMAL(8,2) NOT NULL,

*liitos\_id* : INTEGER NOT NULL,

)

HAARUKKA(

haarukka\_id : INTEGER NOT NULL,

vari : VARCHAR(20) NOT NULL,

tyyppi : VARCHAR(30) NOT NULL,

merkki : VARCHAR(100) NOT NULL,

hinta : DECIMAL(8,2) NOT NULL,

*liitos\_id* : INTEGER NOT NULL,

)

LIITOKSET(

liitos\_id : INTEGER NOT NULL,

tyyppi : VARCHAR(10) NOT NULL,

)

VAIHTEET(

vaihteet\_id : INTEGER NOT NULL,

tyyppi : VARCHAR(30) NOT NULL,

merkki : VARCHAR(100) NOT NULL,

hinta : DECIMAL(8,2) NOT NULL,

)

POLKIMET(

polkimet\_id : INTEGER NOT NULL,

vari : VARCHAR(20) NOT NULL,

tyyppi : VARCHAR(30) NOT NULL,

merkki : VARCHAR(100) NOT NULL,

hinta : DECIMAL(8,2) NOT NULL,

)

Kuva 4.2 relaatiokaavat

## 4.2 Relaatioiden luontilauseet

Pyöräsuunnitelman relaatiokannan luontilauseet on kuvattu kuvassa 4.3. Relaatioiden luontilauseet.

CREATE TABLE Asiakas (

sahkoposti VARCHAR(50) NOT NULL,

salasana VARCHAR(20) NOT NULL,

etunimi VARCHAR(20) NOT NULL,

sukunimi VARCHAR(30) NOT NULL,

sukupuoli VARCHAR(20) NOT NULL,

puhelinnumero VARCHAR(20) NOT NULL,

katuosoite VARCHAR(20) NOT NULL,

postitmp VARCHAR(20) NOT NULL,

postinro VARCHAR(20) NOT NULL,

PRIMARY KEY (sahkoposti)

);

CREATE TABLE Liitokset (

liitos\_id INTEGER NOT NULL,

tyyppi VARCHAR(10) NOT NULL,

PRIMARY KEY (liitos\_id)

);

CREATE TABLE Haarukka (

haarukka\_id INTEGER NOT NULL,

vari VARCHAR(20) NOT NULL,

tyyppi VARCHAR(30) NOT NULL,

merkki VARCHAR(100) NOT NULL,

hinta DECIMAL(8,2) NOT NULL,

liitos\_id INTEGER NOT NULL,

PRIMARY KEY (haarukka\_id),

FOREIGN KEY (liitos\_id) REFERENCES Liitokset(liitos\_id)

);

CREATE TABLE Runko(

runko\_id INTEGER NOT NULL,

vari VARCHAR(20) NOT NULL,

tyyppi VARCHAR(30) NOT NULL,

koko VARCHAR(1) NOT NULL,

merkki VARCHAR(100) NOT NULL,

hinta DECIMAL(8,2) NOT NULL,

liitos\_id INTEGER NOT NULL,

PRIMARY KEY (runko\_id),

FOREIGN KEY (liitos\_id) REFERENCES Liitokset(liitos\_id)

);

CREATE TABLE Tanko(

tanko\_id INTEGER NOT NULL,

vari VARCHAR(20) NOT NULL,

tyyppi VARCHAR(30) NOT NULL,

koko INTEGER NOT NULL,

merkki VARCHAR(100) NOT NULL,

hinta DECIMAL(8,2) NOT NULL,

liitos\_id INTEGER NOT NULL,

PRIMARY KEY (tanko\_id),

FOREIGN KEY (liitos\_id) REFERENCES Liitokset(liitos\_id)

);

CREATE TABLE Polkimet (

polkimet\_id INTEGER NOT NULL,

vari VARCHAR(20) NOT NULL,

tyyppi VARCHAR(30) NOT NULL,

merkki VARCHAR(100) NOT NULL,

hinta DECIMAL(8,2) NOT NULL,

PRIMARY KEY (polkimet\_id)

);

CREATE TABLE Vaihteet(

vaihteet\_id INTEGER NOT NULL,

tyyppi VARCHAR(30) NOT NULL,

merkki VARCHAR(100) NOT NULL,

hinta DECIMAL(8,2) NOT NULL,

PRIMARY KEY (vaihteet\_id)

);

CREATE TABLE Ketjut(

ketjut\_id INTEGER NOT NULL,

tyyppi VARCHAR(30) NOT NULL,

lenkit\_lkm INTEGER NOT NULL,

merkki VARCHAR(100) NOT NULL,

hinta DECIMAL(8,2) NOT NULL,

PRIMARY KEY (ketjut\_id)

);

CREATE TABLE Jarrut(

jarrut\_id INTEGER NOT NULL,

tyyppi VARCHAR(30) NOT NULL,

merkki VARCHAR(100) NOT NULL,

hinta DECIMAL(8,2) NOT NULL,

PRIMARY KEY (jarrut\_id)

);

CREATE TABLE Satula(

satula\_id INTEGER NOT NULL,

vari VARCHAR(20) NOT NULL,

tyyppi VARCHAR(30) NOT NULL,

koko VARCHAR(1) NOT NULL,

merkki VARCHAR(100) NOT NULL,

hinta DECIMAL(8,2) NOT NULL,

PRIMARY KEY (satula\_id)

);

CREATE TABLE Renkaat(

renkaat\_id INTEGER NOT NULL,

tyyppi VARCHAR(30) NOT NULL,

koko INTEGER NOT NULL,

merkki VARCHAR(100) NOT NULL,

hinta DECIMAL(8,2) NOT NULL,

PRIMARY KEY (renkaat\_id)

);

CREATE TABLE Vanteet(

vanteet\_id INTEGER NOT NULL,

tyyppi VARCHAR(30) NOT NULL,

koko INTEGER NOT NULL,

merkki VARCHAR(100) NOT NULL,

hinta DECIMAL(8,2) NOT NULL,

PRIMARY KEY (vanteet\_id)

);

CREATE TABLE Pyorasuunnitelma (

pyorasuunnitelma\_id INTEGER NOT NULL,

kokonaishinta DECIMAL(8,2) NOT NULL,

pvm DATE NOT NULL,

asiakas\_id VARCHAR(50) NOT NULL,

polkimet\_id INTEGER NOT NULL,

vaihteet\_id INTEGER NOT NULL,

haarukka\_id INTEGER NOT NULL,

runko\_id INTEGER NOT NULL,

ketjut\_id INTEGER NOT NULL,

tanko\_id INTEGER NOT NULL,

jarrut\_id INTEGER NOT NULL,

satula\_id INTEGER NOT NULL,

renkaat\_id INTEGER NOT NULL,

vanteet\_id INTEGER NOT NULL,

PRIMARY KEY (pyorasuunnitelma\_id),

FOREIGN KEY (asiakas\_id) REFERENCES Asiakas(sahkoposti),

FOREIGN KEY (polkimet\_id) REFERENCES Polkimet(polkimet\_id),

FOREIGN KEY (vaihteet\_id) REFERENCES Vaihteet(vaihteet\_id),

FOREIGN KEY (haarukka\_id) REFERENCES Haarukka(haarukka\_id),

FOREIGN KEY (runko\_id) REFERENCES Runko(runko\_id),

FOREIGN KEY (ketjut\_id) REFERENCES Ketjut(ketjut\_id),

FOREIGN KEY (tanko\_id) REFERENCES Tanko(tanko\_id),

FOREIGN KEY (jarrut\_id) REFERENCES Jarrut(jarrut\_id),

FOREIGN KEY (satula\_id) REFERENCES Satula(satula\_id),

FOREIGN KEY (renkaat\_id) REFERENCES Renkaat(renkaat\_id),

FOREIGN KEY (vanteet\_id) REFERENCES Vanteet(vanteet\_id)

);

CREATE SEQUENCE seq\_suunnitelma MINVALUE 1 INCREMENT BY 1;

Kuva 4.3. Relaatioiden luontilauseet

5 Järjestelmän arkkitehtuurimalli

Järjestelmän rakenne käyttäjän ja pääservletin välillä on kuvattu JSP Model 2:n mukaisessa MVC-mallissa kuvassa 5.1 Järjestelmän rakenne.

Kuva 5.1. Järjestelmän rakenne.

Järjestelmässä käytetään 3-tasoista mallia, joka ilmenee kuvassa 5.2. 3-tasomalli.

#### JSP:t

#### Tieto-kanta-

#### luokat

#### Liike-toiminta-

#### luokat

#### Servlet-luokka

Kuva 5.2. 3-tasomalli

Yhteistyökaavioissa ei näy container-ohjelmistoa, vaan kaavioissa käyttöliittymä kutsuu suoraan pääservletin doGet- ja doPost-metodia.

Yhteistyökaavioilla kuvataan järjestelmän käyttötapaukset ja käyttötapausten osakäyttötapaukset. Niihin on liitetty metodin kutsun paluuarvo katkoviivoin UML-standardista poiketen.

## 5.1 Kirjautuminen osakäyttötapaus

Kun järjestelmä käynnistetään, avautuu kirjautuminen.jsp-ikkuna. Tähän ikkunaan käyttäjä kirjoittaa käyttäjätunnuksensa, joka on asiakkaan sähköpostiosoite ja tämän lisäksi salasanan. Tämän jälkeen hän painaa Kirjaudu-painiketta. Kun asiakkaan tiedot on haettu onnistuneesti, tietokannasta haetaan vielä runkojen tyypit ja siirrytään suunnittelu.jsp-näyttöön.

Tapahtumat, jotka seuraavat Kirjaudu-nappulan painalluksen jälkeen, on kuvattu kuvassa 5.2 Kirjautuminen yhteistyökaavio.



Kuva 5.2. Kirjautuminen yhteistyökaavio

## 5.2 Pyörän suunnittelu osakäyttötapaus

Näytölle tulee aktiiviset linkit, joista käyttäjä valitsee haluamansa rungon tyypin. Kun käyttäjä on valinnut tyypin, alle listautuu hakukriteeriä vastaavat osat. Osista näytetään vain merkki ja hinta. Tapahtuma on kuvattu yhteistyökaavion avulla kuvassa, 5.3 Pyörän suunnittelu yhteistyökaavio.



Kuva 5.3. Pyörän suunnittelu yhteistyökaavio

## 5.3 Merkin valinta osakäyttötapaus

Kun käyttäjä valitsee rungon merkin ja painaa pyörän merkissä olevaa linkkiä, alle avautuu aktiiviset linkit jossa on tarkemmat tiedot osista (esim. väri, koko). Tapahtuma on kuvattu yhteistyökaavion avulla kuvassa, 5.3 Merkin valinta osakäyttötapaus.

Kuva 5.3 Merkin valinta osakäyttötapaus

# 6 Viestiyhteyskaaviot

Viestiyhteyskaavioissa kuvataan samat käyttötapaukset ja osa­käyttö­ta­pauk­set kuin yhteistyökaavioissakin. Kaavioissa näkyy käyttöliittymän (jsp-tiedoston) ja pääservletin välinen kommunikointi siten, että käyttöliittymä kutsuu pääservletin doPost- tai doGet-metodia. Todellisuudessa käyttöliittymä ei suoraan kutsu pääservletin metodeja vaan kutsun tekee Container-sovellus (ks. kuva 5.1. Järjestelmän rakenne.). Container-sovellus ei siis näy kaavioissa. Lisäksi kaavioissa doGet- ja doPost-metodien palauttama arvo kuvataan metodien paluuarvona. Todellisuudessa paluutieto palautetaan doGet- ja doPost-metodien parametrin HttpServerRequest-olion attribuutin avulla.

Lisäksi kun pääservletti antaa käskyn aktivoida toisen käyttöliittymän, tämä kuvataan viestiyhteyskaavioissa metodin ”aktivoi”-kutsulla. Todellisuudessa pääservletti välittää tiedon RequestDispatcher-olion metodin forward avulla.

## 6.1 Kirjautuminen viestiyhteistyökaavio

Kirjautuminen järjestelmässä on yksi käyttötapaus, kirjautuminen, joka muodostuu yhdestä osakäyttötapauksesta:

1. Kirjautuminen, jolloin käyttäjä kirjoittaa sähköpostiosoitteen ja salasanan, jonka jälkeen painaa Kirjaudu-painiketta.

Tapahtuma on kuvattu viestiyhteistyökaavion avulla kuvissa, 6.1.1 kirjautuminen viestiyhteyskaavio 1, 6.1.2 kirjautuminen viestiyhteyskaavio 2, 6.1.3 kirjautuminen viestiyhteyskaavio 3.



Kuva 6.1.1 kirjautuminen viestiyhteyskaavio 1



Kuva 6.1.2 kirjautuminen viestiyhteyskaavio 2

Kuva 6.1.3 kirjautuminen viestiyhteyskaavio 3

## 6.2 Suunnittelu viestiyhteistyökaavio

Suunnittelujärjestelmässä on yksi käyttötapaus, pyörän suunnittelu, joka muodostuu useasta osakäyttötapauksesta:

* 1. Ensiksi valitaan ylhäällä olevasta linkistä osan tyyppi, jonka jälkeen alle listautuu hakukriteeriä vastaavat osat. Osista näytetään listauksessa malli ja hinta.

Tapahtuma on kuvattu viestiyhteistyökaavion avulla kuvissa, 6.2.1, 6.2.2 suunnittelu viestiyhteyskaavio.



Kuva 6.2.1 suunnittelu viestiyhteyskaavio 1

Kuva 6.2.2 suunnittelu viestiyhteyskaavio 2

## 6.3 Osan valinta viestiyhteistyökaavio

Suunnittelujärjestelmässä on yksi käyttötapaus, pyörän suunnittelu, joka muodostuu useasta osakäyttötapauksesta:

* 1. Käyttäjä painaa haluamansa osan nimeä ja alle avautuu lista, jossa voi valita osan värin, koon yms. Haluttu osa siirtyy suunnitelmaan linkkiä painamalla. Kaikki osat valittuaan näytölle listautuu yhteenveto suunnitelmasta ja käyttäjä painaa Lähetä tarjouspyyntö-painiketta jolloin tarjouspyyntö menee käsittelyyn.

Tapahtuma on kuvattu viestiyhteistyökaavion avulla kuvissa: 6.3.1, 6.3.2, 6.3.3 ja 6.3.4 osan valinta viestiyhteyskaavio. Kuvat 6.3.1 ja 6.3.2 kuvaavat tilannetta, jossa valitaan pyörän ensimmäinen osa. Kuvat 6.3.3 ja 6.3.4 kuvaavat tilannetta, jossa valitaan pyörän seuraava osa. Seuraavat osat valitaan samalla periaatteella kuin näissä kahdessa jälkimmäisessä kuvassa.

Kuva 6.3.1 osan valinta viestiyhteyskaavio 1

Kuva 6.3.2 osan valinta viestiyhteyskaavio 2

Kuva 6.3.3 osan valinta viestiyhteyskaavio 3

Kuva 6.3.4 osan valinta viestiyhteyskaavio 4

## 6.4 Tarjouspyynnön lähetys viestiyhteyskaavio

Suunnittelujärjestelmässä on yksi käyttötapaus, pyörän suunnittelu, joka muodostuu useasta osakäyttötapauksesta:

Käyttäjä painaa ”Lähetä tarjouspyyntö” painiketta. Järjestelmä lisää pyöräsuunnitelmaan tietokantaan ja siirtää käyttäjän Kiitos.jsp näkymään.

Tapahtuma on kuvattu viestiyhteistyökaavion avulla kuvissa: 6.4.1 ja 6.4.2 osan valinta viestiyhteyskaavio.

Kuva 6.4.1 Tarjouspyynnön lähetys viestiyhteyskaavio 1



Kuva 6.4.2 Tarjouspyynnön lähetys viestiyhteyskaavio 2

## 6.5 Tarjouspyynnön peruutus viestiyhteyskaavio

Tarjouspyynnön peruutus toimii täsmälleen samalla tavalla kahdessa eri käyttötapauksessa, Suunnittelu.jsp ja Kiitos.jsp.

Käyttäjä painaa ”Tyhjennä pyöräsuunnitelma”, ”peruuta” tai ”suunnittele uusi pyörä” painiketta. Järjestelmä tyhjentää pyöräsuunnitelman ja siirtää käyttäjän Suunnittelu.jsp näkymään ensimmäisen komponentin valintaan.

Tapahtuma on kuvattu viestiyhteistyökaavion avulla kuvassa: 6.5 Tarjouspyynnön peruutus viestiyhteyskaavio.



Kuva 6.5 Tarjouspyynnön peruutus viestiyhteyskaavio

## 6.6 Ulos kirjautuminen viestiyhteyskaavio

Ulos kirjautuminen toimii täsmälleen samalla tavalla kahdessa eri käyttötapauksessa, Suunnittelu.jsp ja Kiitos.jsp.

Käyttäjä painaa ”Kirjaudu Ulos” painiketta. Järjestelmä kirjaa käyttäjän ulos ja siirtää hänet Kirjautuminen.jsp näkymään.

Tapahtuma on kuvattu viestiyhteistyökaavion avulla kuvassa: 6.6 Ulos kirjautuminen viestiyhteyskaavio.



Kuva 6.6 Ulos kirjautuminen viestiyhteyskaavio

# 7 Paketit

Websovelluksen luokat jaetaan kolmeen pakettiin: kohdeluokat, servletit ja tietokanta.



Kuva 7.1 Kohdeluokat-paketti

## 7.1 Kohdeluokat-paketti

Kohdeluokat-pakettiin kuuluvat AsiakasOhjaus-, RunkoOhjaus-, PyoraSuunnitelma-, Asiakas-, Runko-, Liitos-, Pvm-, Haarukka-, Jarrut-, Ketjut-, Polkimet-, Renkaat-, Satula-, Tanko-, Vaihteet- ja Vanteet-luokka.

AsiakasOhjaus-luokka Toimii FillariIhmeet- ja TietokantaOhjaus-luokan välissä.

Attribuutit ArrayList<HashMap <String, String>> lista;

Muodostimet public AsiakasOhjaus() - oletusmuodostin

Metodit public Asiakas hae (String ktunnus, String salas) throws SQLException

* hakee käyttäjän antaman hakukriteerin perusteella asiakkaiden tiedot tietokannasta

RunkoOhjaus-luokka Kuvaa käsiteltävää tietoa pyörän tyypin valinta ja haku.

Attribuutit ArrayList<HashMap <String, String>> lista;

ArrayList<String>tyypit;

String tyyppi;

String merkki;

Muodostimet public RunkoOhjaus() - oletusmuodostin

Metodit public ArrayList<Runko> haeRungot(String tyyppi) throws SQLException

public ArrayList<Runko> haeValitutRungot(String merkki) throws SQLException

public Runko haeValittuRunko(int runkoId) throws SQLException

public ArrayList<Haarukka> haeHaarukat(Liitos liitos) throws SQLException

public ArrayList<String> haeHaarukkaTyypit() throws SQLException

public ArrayList<Haarukka> haeValitutHaarukat(String merkki) throws SQLException

public Haarukka haeValittuHaarukka(int haarukkaId) throws SQLException

PyoraSuunnitelma-luokka Kuvaa käsiteltävää tietoa pyörän suunnittelu.

Attribuutit private int pyorasuunnitelma\_id;

private Double kokonaishinta;

private String pvm;

Muodostimet public PyoraSuunnitelma() - oletusmuodostin

Metodit -

Asiakas-luokka kuvaa käsiteltävää tietoa asiakas.

Attribuutit private String sahkoposti;

private String salasana;

private String etunimi;

private String sukunimi;

private String sukupuoli;

private String puhelinnumero;

private String katuosoite;

private String postitmp;

private String postinro;

Muodostimet public Asiakas (String sposti, String etunimi, String sukunimi, String sex, String puh, String katuos, String postitmp, String postinro)

Metodit public void setSahkoposti (String sahkoposti)

public void setSalasana(String salasana)

public void setEtunimi(String etunimi)

public void setSukunimi(String sukunimi)

public void setSukupuoli(String sukupuoli)

public void setPuhelinnumero(String puhelinnumero)

public void setKatuosoite(String katuosoite)

public void setPostitmp(String postitmp)

public void setPostinro(String postinro)

public String getSahkoposti()

public String getSalasana()

public String getEtunimi()

public String getSukunimi()

public String getSukupuoli()

public String getPuhelinnumero()

public String getKatuosoite()

public String getPostitmp()

public String getPostinro()

public String toString()

public static boolean tarkastaSahkoposti(String posti)

public static boolean tarkastaSalasana(String sana)

Runko-luokka Kuvaa käsitettä pyörän runko.

Attribuutit private int runko\_id;

private String vari;

private String tyyppi;

private String koko;

private String merkki;

private double hinta;

private Liitos liitos;

Muodostimet public Runko (int runko\_id, String vari, String tyyppi, String koko, String merkki, double hinta)

public Runko (int runko\_id, String vari, String tyyppi, String koko, String merkki, double hinta, Liitos lii)

Metodit public void setRunko\_id(int runko\_id)

public void setVari(String vari)

public void setTyyppi(String tyyppi)

public void setKoko(String koko)

public void setMerkki(String merkki)

public void setHinta(double hinta)

public void setLiitos\_id(int liitos\_id)

public int getRunko\_id()

public String getVari()

public String getTyyppi()

public String getKoko()

public String getMerkki()

public double getHinta()

public int getLiitos\_id()

* palautetaan kutsujalle ko. attribuutin arvo

public String toString()

Liitos-luokka Kuvaa pyörän rungon, tangon ja haarukan liitostyyppiä.

Attribuutit private int liitos\_id;

private String tyyppi;

Muodostimet public Liitos (int liitos\_id, String tyyppi)

Metodit public void setLiitos\_id(int liitos\_id)

public void setTyyppi(String tyyppi)

public int getLiitos\_id()

public String getTyyppi()

* palautetaan kutsujalle ko. attribuutin arvo

public String toString()

Haarukka-luokka Kuvaa käsitettä pyörän haarukan valinta.

Attribuutit private int haarukka\_id;

private String vari;

private String tyyppi;

private String merkki;

private double hinta;

private int liitos\_id;

private Liitos liitos;

Muodostimet public Haarukka(String merkki, double hinta)

public Haarukka (int haarukka\_id, String vari, String tyyppi, String merkki, double hinta)

public Haarukka (int haarukka\_id, String vari, String tyyppi, String merkki,double hinta, Liitos lii)

Metodit public void setHaarukka\_id(int haarukka\_id

public void setVari(String vari)

public void setTyyppi(String tyyppi)

public void setKoko(int koko)

public void setMerkki(String merkki)

public void setHinta(double hinta)

public void setLiitos\_id(int liitos\_id)

public int getKoko()

public String getTyyppi()

public String getVari()

public int getHaarukka\_id()

public String getMerkki()

public double getHinta()

public int getLiitos\_id()

Jarrut-luokka Kuvaa käsitettä pyörän jarrujen valinta.

Attribuutit private int jarrut\_id;

private String tyyppi;

private String merkki;

private double hinta;

Muodostimet public Jarrut (int jarrut\_id, String tyyppi, String merkki, double hinta)

Metodit public void setJarrut\_id(int jarrut\_id

public void setTyyppi(String tyyppi)

public void setMerkki(String merkki)

public void setHinta(double hinta)

public int getJarrut\_id

public String getTyyppi

public String getMerkki()

public double getHinta()

Ketjut-luokka Kuvaa käsitettä pyörän ketjujen valinta.

Attribuutit private int ketjut\_id;

private String tyyppi;

private int lenkit\_lkm;

private String merkki;

private double hinta;

Muodostimet public Ketjut (ketjut\_id, String tyyppi, lenkit\_lkm, String merkki, double hinta)

Metodit public void setKetjut\_id(int ketjut\_id)

public void setTyyppi(String tyyppi)

public void setLenkit\_lkm(int lenkit\_lkm)

public void setMerkki(String merkki)

public void setHinta(double hinta)

public int getKetjut\_id()

public String getTyyppi()

public int getLenkit\_lkm()

public String getMerkki()

public double getHinta()

Polkimet-luokka Kuvaa käsitettä pyörän polkimien valinta.

Attribuutit private int polkimet\_id;

private String vari;

private String tyyppi;

private String merkki;

private double hinta;

Muodostimet public Polkimet (polkimet\_id, String vari, String tyyppi, String merkki, double hinta)

Metodit public void setPolkimet\_id(int polkimet\_id)

public void setVari(String vari)

public void setTyyppi(String tyyppi)

public void setMerkki(String merkki)

public void setHinta(double hinta)

public int getPolkimet\_id()

public String getVari()

public String getTyyppi()

public String getMerkki()

public double getHinta()

Renkaat-luokka Kuvaa käsitettä pyörän renkaiden valinta.

Attribuutit private int renkaat\_id;

private String tyyppi;

private int koko;

private String merkki;

private double hinta;

Muodostimet public Renkaat (renkaat\_id, String tyyppi, int koko, String merkki, double hinta)

Metodit public void setRenkaat\_id(int renkaat\_id)

public void setTyyppi(String tyyppi)

public void setKoko(int koko)

public void setMerkki(String merkki)

public void setHinta(double hinta)

public int getRenkaat\_id()

public String getTyyppi()

public int getKoko()

public String getMerkki()

public double getHinta()

Satula-luokka Kuvaa käsitettä pyörän satulan valinta.

Attribuutit private int satula\_id;

private String vari;

private String tyyppi;

private String koko;

private String merkki;

private double hinta;

Muodostimet public Satula (int satula\_id, String vari, String tyyppi, String koko, String merkki, double hinta)

Metodit public void setSatula\_id(int satula\_id)

public void setTyyppi(String tyyppi)

public void setKoko(int koko)

public void setMerkki(String merkki)

public void setHinta(double hinta)

public int getSatula\_id()

public String getTyyppi()

public int getKoko()

public String getMerkki()

public double getHinta()

Tanko-luokka Kuvaa käsitettä pyörän tangon valinta.

Attribuutit private int tanko\_id;

private String vari;

private String tyyppi;

private int koko;

private String merkki;

private double hinta;

private int liitos\_id;

private Liitos liitos;

Muodostimet public Tanko(String merkki, double hinta)

public Tanko (int tanko\_id, String vari, String tyyppi, String koko, String merkki, double hinta, int liitos\_id)

public Tanko (int tanko\_id, String vari, String tyyppi, int koko, String merkki,double hinta, Liitos lii)

Metodit public void setTanko\_id(int tanko\_id)

public void setVari(String vari)

public void setTyyppi(String tyyppi)

public void setKoko(int koko)

public void setMerkki(String merkki)

public void setHinta(double hinta)

public void setLiitos\_id(int liitos\_id)

public int getTanko\_id()

public String getVari()

public String getTyyppi()

public int getKoko()

public String getMerkki()

public double getHinta()

public int getLiitos\_id()

Vaihteet-luokka Kuvaa käsitettä pyörän vaihteiden valinta.

Attribuutit private int vaihteet\_id;

private String tyyppi;

private String merkki;

private double hinta;

Muodostimet public Vaihteet (vaihteet\_id, double hinta, String merkki, String tyyppi)

Metodit public void setVaihteet\_id(int vaihteet\_id)

public void setTyyppi(String tyyppi)

public void setMerkki(String merkki)

public void setHinta(double hinta)

public int getVaihteet\_id()

public String getTyyppi()

public String getMerkki()

public double getHinta()

Vanteet-luokka Kuvaa käsitettä pyörän vanteiden valinta.

Attribuutit private int vanteet\_id;

private String tyyppi;

private int koko;

private String merkki;

private double hinta;

Muodostimet public Vanteet (vanteet\_id, String tyyppi, int koko, String merkki,

double hinta)

Metodit public void setVanteet\_id(int vanteet\_id)

public void setTyyppi(String tyyppi)

public void setKoko(int koko)

public void setMerkki(String merkki)

public void setHinta(double hinta)

public int getVanteet\_id()

public String getTyyppi()

public int getKoko()

public String getMerkki()

public double getHinta()

Pvm-luokka Kuvaa käsitettä päivämäärä.

Attribuutit int pp;

int kk;

int vv;

Muodostimet public Pvm( int pp, int kk, int vv)

Metodit

public void setPaiva(int pp)

public void setKuukausi(int kk)

public void setVuosi(int vv)

public void setPvm(int pp, int kk, int vv)

public int getPaiva()

public int getKuukausi()

public int getVuosi()

public void lisaaPaiva(int arvo)

public void lisaaKuukausi(int arvo)

public void lisaaVuosi(int arvo)

public String toString()



## 7.2 Servletit-paketti

Sisältää järjestelmän pääservletin eli käyttöliittymää ohjaavan FillariIhmeet-luokan. FillariIhmeet-luokka toimii käyttäjälle näytettävien jsp-sivujen sekä liiketoimintaa ohjaavan AsiakasOhjaus-luokan välillä.

FillariIhmeet-luokka Sisältää ohjelman logiikan, on järjestelmän pääservletti ja toimii käyttöliittymien ja Kohdeluokkien välillä.

Muodostimet public FillariIhmeet() - oletusmuodostin

Metodit protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void merkkiValittu(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void tyyppiValittu(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void runkoValittu(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void haarukkaTyyppiValittu(HttpServletRequest request, HttpServletResponse response, Runko runko) throws ServletException, IOException

protected void haarukkaMerkkiValittu(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void haarukkaValittu(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void tankoTyyppiValittu(HttpServletRequest request, HttpServletResponse response, Runko runko) throws ServletException, IOException

protected void tankoTyyppiValittu(HttpServletRequest request, HttpServletResponse response, Runko runko) throws ServletException, IOException

protected void tankoMerkkiValittu(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void tankoValittu(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void vanneTyyppiValittu(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void vanneMerkkiValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void vanneValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void RengasTyyppiValittu(HttpServletRequest request,

HttpServletResponse response, Vanteet vanne)

protected void rengasMerkkiValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void RenkaatValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void JarruTyyppiValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void JarrutMerkkiValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void JarrutValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void KetjutTyyppiValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void ketjuMerkkiValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void KetjutValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void PolkimetTyyppiValittu(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void polkimetMerkkiValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void PolkimetValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void satulaTyyppiValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void satulaMerkkiValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void satulaValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void vaihdeTyyppiValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void vaihdeMerkkiValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void vaihdeValittu(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

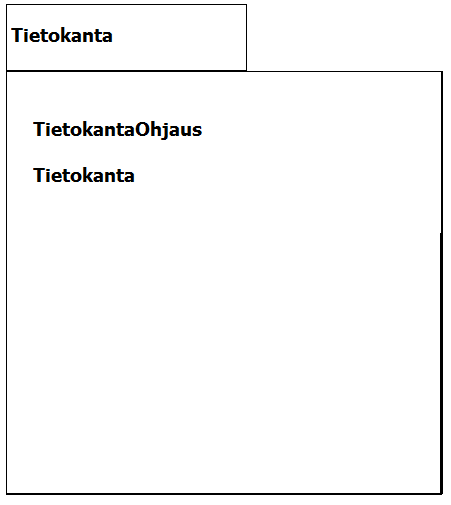
protected void lahetaPyorasuunnitelma(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

protected void kirjauduUlos(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void kirjaudu(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException



## 7.3 Tietokanta-paketti

Sisältää TietokantaOhjaus- ja Tietokanta-luokat.

TietokantaOhjaus-luokka Toimii tietokanta-paketin rajapintana.

Attribuutit private Tietokanta tietokanta;

Muodostimet public TietokantaOhjaus() - oletusmuodostin

Metodit public ArrayList <HashMap<String,String>> haeAsiakas (String sposti, String ssana) throws SQLException

public ArrayList<HashMap<String, String>> haeTyypit() throws SQLException

public ArrayList<HashMap<String, String>> haeRungot(String tyyppi)

throws SQLException

public ArrayList<HashMap<String, String>> haeValitutRungot(String merkki)

throws SQLException

public ArrayList<HashMap<String, String>> haeValittuRunko(int runko\_id)

throws SQLException

public ArrayList<HashMap<String, String>> haeHaarukkaTyypit() throws SQLException

public ArrayList<HashMap<String, String>> haeHaarukat(Liitos liitos)

throws SQLException

public ArrayList<HashMap<String, String>> haeValitutHaarukat(String merkki) throws SQLException

public ArrayList<HashMap<String, String>> haeValittuHaarukka(int haarukka\_id) throws SQLException

public ArrayList<HashMap<String, String>> haeTankoTyypit() throws SQLException

public ArrayList<HashMap<String, String>> haeTangot(Liitos liitos)

throws SQLException

public ArrayList<HashMap<String, String>> haeValitutTangot(String merkki)

throws SQLException

public ArrayList<HashMap<String, String>> haeValittuTanko(int tanko\_id)

throws SQLException

public ArrayList<HashMap<String, String>> haeVanneTyypit() throws SQLException

public ArrayList<HashMap<String, String>> haeVanteet(String tyyppi)

throws SQLException

public ArrayList<HashMap<String, String>> haeValitutVanteet(String merkki)

throws SQLException

public ArrayList<HashMap<String, String>> haeValittuVanne(int vanteet\_id)

throws SQLException

public ArrayList<HashMap<String, String>> haeRengasTyypit() throws SQLException

public ArrayList<HashMap<String, String>> haeRenkaat(int vannekoko)

throws SQLException

public ArrayList<HashMap<String, String>> haeValitutRenkaat(String merkki)

throws SQLException

public ArrayList<HashMap<String, String>> haeValittuRengas(int renkaat\_id)

throws SQLException

public ArrayList<HashMap<String, String>> haeJarruTyypit() throws SQLException

public ArrayList<HashMap<String, String>> haeJarrut(String tyyppi)

throws SQLException

public ArrayList<HashMap<String, String>> haeValitutJarrut(String merkki)

throws SQLException

public ArrayList<HashMap<String, String>> haeValittuJarru(int jarrut\_id)

throws SQLException

public ArrayList<HashMap<String, String>> haeKetjuTyypit() throws SQLException

public ArrayList<HashMap<String, String>> haeKetjut(String tyyppi)

throws SQLException

public ArrayList<HashMap<String, String>> haeValitutKetjut(String merkki)

throws SQLException

public ArrayList<HashMap<String, String>> haeValittuKetju(int ketjut\_id)

throws SQLException

public ArrayList<HashMap<String, String>> haePolkimetTyypit() throws SQLException

public ArrayList<HashMap<String, String>> haePolkimet(String tyyppi)

throws SQLException

public ArrayList<HashMap<String, String>> haeValitutPolkimet(String merkki) throws SQLException

public ArrayList<HashMap<String, String>> haeValittuPoljin(int polkimet\_id)

throws SQLException

public ArrayList<HashMap<String, String>> haeSatulaTyypit() throws SQLException

public ArrayList<HashMap<String, String>> haeSatulat(String tyyppi)

throws SQLException

public ArrayList<HashMap<String, String>> haeValitutSatulat(String merkki)

throws SQLException

public ArrayList<HashMap<String, String>> haeValittuSatula(int satula\_id)

throws SQLException

public ArrayList<HashMap<String, String>> haeVaihteetTyypit() throws SQLException

public ArrayList<HashMap<String, String>> haeVaihteet(String tyyppi)

throws SQLException

public ArrayList<HashMap<String, String>> haeValitutVaihteet(String merkki)

throws SQLException

public ArrayList<HashMap<String, String>> haeValittuVaihde(int vaihteet\_id)

throws SQLException

public boolean talletaPyorasuunnitelma(PyoraSuunnitelma ps) throws SQLException

Tietokanta-luokka kuvaa fyysistä tietokantaa. Sen avulla saadaan yhteys todelliseen fyysiseen tietokantaa ja voidaan suorittaa kyselyjä ja päivityksiä em. kantaa.

Attribuutit private String sqlstate;

private boolean yhteysOnAuki = false;

private boolean transaktioKesken;

private Connection tietokantayhteys;

Metodit public boolean yhdista() throws SQLException

- ottaa yhteyden fyysiseen tietokantaan

public boolean katkaise() throws SQLException

- katkaisee yhteyden fyysiseen tietokantaan

public ArrayList<HashMap> suoritaKysely(String sql) throws SQLException

- suorittaa parametrissa saamansa sql-lauseen tietokannassa ja palauttaa tuloksen ArrayList <HashMap>-oliossa.

public int suoritaPaivitys(String sql) throws SQLException

- suorittaa parametrissa saamansa sql-päivityslauseen ja palauttaa arvonaan päivitettyjen rivien lukumäärän.