

# **Suunnitteludokumentti**

Biocafe

Helsinki 7.9.2006

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

**Kurssi**

581260 Ohjelmistotuotantoprojekti (9 + 1 op, 6 ov)

**Projektiryhmä**

Sami Laiti  
Mari Vierelä  
Sampsa Lappalainen  
Jaakko Nyman  
Teemu Kemppainen

**Asiakkaat**

Harri Laine  
Petri Kutvonen

**Johtoryhmä**

Juha Taina  
Kimmo Simola

**Kotisivu**

<http://www.cs.helsinki.fi/group/biocafe>

**Versiohistoria**

Versio	Päiväys	Tehdyt muutokset
0.9	17.8.2006	Ensimmäinen LaTeX-versio
0.91	24.8.2006	Päivityksiä tietoihin
0.95	28.8.2006	Kuvat kuntoon
0.96	3.9.2006	Päivityksiä
1.0	5.9.2006	Lopullinen versio

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
1.1	Dokumentin rakenne . . . . .	1
1.2	Sanasto . . . . .	1
<b>2</b>	<b>Järjestelmäarkkitehtuuri</b>	<b>3</b>
2.1	Arkkitehtuurikaavio . . . . .	3
2.2	Tiedon kulku osajärjestelmien välillä . . . . .	4
<b>3</b>	<b>Tietokanta</b>	<b>5</b>
3.1	Tietokantakaavio . . . . .	5
3.2	Tietokannan taulut . . . . .	5
<b>4</b>	<b>Tietokantakirjasto</b>	<b>8</b>
4.1	Luokkakaavio . . . . .	8
4.2	Palveluiden tarkempi määrittely . . . . .	8
<b>5</b>	<b>Viestinvälitysoliot</b>	<b>9</b>
5.1	Luokkakaavio . . . . .	9
5.2	Luokkien esittelyt . . . . .	9
5.2.1	DBUser . . . . .	9
5.2.2	DBProduct . . . . .	10
5.2.3	DBBasicProduct extends DBProduct . . . . .	10
5.2.4	DBDeliverable . . . . .	10
<b>6</b>	<b>Pääsovellus</b>	<b>11</b>
6.1	Yleiset apuluokat . . . . .	11
6.1.1	Luokkakaavio . . . . .	11
6.1.2	User extends DBUser implements Comparable . . . . .	12
6.1.3	BasicProduct extends DBBasicProduct . . . . .	12
6.1.4	Deliverable extends DBDeliverable . . . . .	12
6.1.5	FingerPrint . . . . .	12
6.1.6	FingerPrintSet . . . . .	12
6.1.7	SupplementSet . . . . .	12

6.1.8	Utilities . . . . .	12
6.2	Rajapintaluokka . . . . .	13
6.2.1	Extractor . . . . .	13
6.3	Hallintänäkymän käyttämät luokat . . . . .	13
6.3.1	Käyttöliittymän komponenttikuvaus . . . . .	13
6.3.2	Luokkakaavio . . . . .	14
6.3.3	MainFrame extends JFrame . . . . .	14
6.3.4	UserView extends AbstractTableModel . . . . .	15
6.3.5	BasicProductView extends AbstractTableModel . . . . .	15
6.3.6	DeliverableView extends AbstractTableModel . . . . .	16
6.3.7	ButtonRenderer extends JButton implements TableCellRenderer .	16
6.3.8	ButtonEditor extends AbstractCellEditor implements TableCellEditor, ActionListener . . . . .	16
6.3.9	RegisterFingersDialog extends JDialog Implements: FingerCallback, ImageCallBack, StatusCallBack, Extractor . . . . .	16
6.3.10	SupplementDialog extends JDialog . . . . .	16
6.3.11	TableSorter extends AbstractTableModel . . . . .	16
6.3.12	Dummy . . . . .	17
6.4	Kahvinäkymän käyttämät luokat . . . . .	17
6.4.1	Käyttöliittymän komponenttikuvaus . . . . .	17
6.4.2	Luokkakaavio . . . . .	18
6.4.3	CoffeeDialog extends JDialog implements Extractor . . . . .	18
6.4.4	CoffeeDeliverableView implements AbstractTableModel . . . . .	18
6.4.5	PortionSelector extends JPanel . . . . .	19

## Liitteet

### 1 Tietokannan alustuslauseet

# 1 Johdanto

Tässä suunnitteludokumentissa on kuvattu biometriseen tunnistukseen perustuvan kahvikassajärjestelmän sisäinen rakenne. Järjestelmä toteutetaan Helsingin yliopiston tietojenkäsittelytieteen laitoksen henkilökunnan kahvihuoneeseen. Dokumentissa on tarkoitus kuvata järjestelmän sisäinen rakenne sillä tarkkuudella, että sen toteutus on dokumentin avulla suoraviivaista. Dokumentti on järjestelmän toteutusvaiheen jälkeen päivitetty kuvaamaan toteutettua järjestelmää, sillä järjestelmän rakenne on suunnittelun aikana muuttunut jonkin verran.

## 1.1 Dokumentin rakenne

Dokumentissa on aluksi kuvattu järjestelmän yleinen arkkitehtuuri sekä osajärjestelmien välillä kulkevat tiedot korkealla tasolla. Tämän jälkeen on kuvattu järjestelmän käyttämän tietokannan looginen rakenne sekä sen täydellinen tietosisältö. Tietokannan jälkeen dokumentissa on kuvattu tietokantakirjasto, joka on järjestelmää varten toteutettu kirjastoluokka tarvittaville tietokantapalveluille. Tietokantakirjasto ja järjestelmän pääsovellus kommunikoivat keskenään viestinvälitysolioilla, jotka on kuvattu kirjastoa seuraavassa luvussa. Tietokantaan liittyvien alilukujen jälkeen on kuvattu järjestelmän pääsovellus. Sovelluksesta on dokumentissa esitetty sen kaksi eri näkymää sekä näkymien käyttämät apuluokat. Kummastakin näkymästä sekä sovelluksen apuluokista on kustakin oma alilukunsa. Sovelluksen ja näkymien kuvaaminen on tehty luokkatasolla, joistain luokista on esitelty muutamia esimerkkipalveluita. Tarkat palvelukuvaukset sekä -määritteet löytyvät liitteenä olevasta sovellusohjelmointirajapinnan kuvauksesta (API).

## 1.2 Sanasto

Ohessa on esitetty dokumentissa käytettyjen termien määritelmät.

**DBLib** Tietokantakirjasto jota sovellus käyttää kommunikointiin tietokannan kanssa

**Hallintanäkymä** Pääsovelluksen hallintanäkymä, jonka avulla järjestelmän tuote- sekä käyttäjätietoja voidaan muokata

**Huomautusraja** Raja tuoteveloissa, jonka jälkeen käyttäjä saa tiedon ylivelkaantumisesta

**ID** Käyttäjän yksilöllinen tunniste tietokannassa, josta lasketaan *PIN-koodi*

**Järjestelmä** Järjestelmä on yleissana koko toteutettavalle ohjelmistolle

**Kahvinäkymä** Kahvihuoneen näkymä, joka on tarkoitettu vain kahvihuoneessa käytettäväksi

**Käyttäjä** Järjestelmän käyttäjäksi rekisteröitynyt henkilö

**PIN-koodi** Käyttäjän ID:stä laskettu numerosarja, jolla käyttäjä voi kirjata tuotekäyttäjään

**Perustuote** Jokin järjestelmän kirjanpituotteista, esimerkiksi kahvi ja sokeri ovat perustuotteita

**Pääsovellus** Järjestelmän pääsovellus, johon on kaksi erillistä näkymää ja joka hoitaa kommunikoinnin tietokannan kanssa

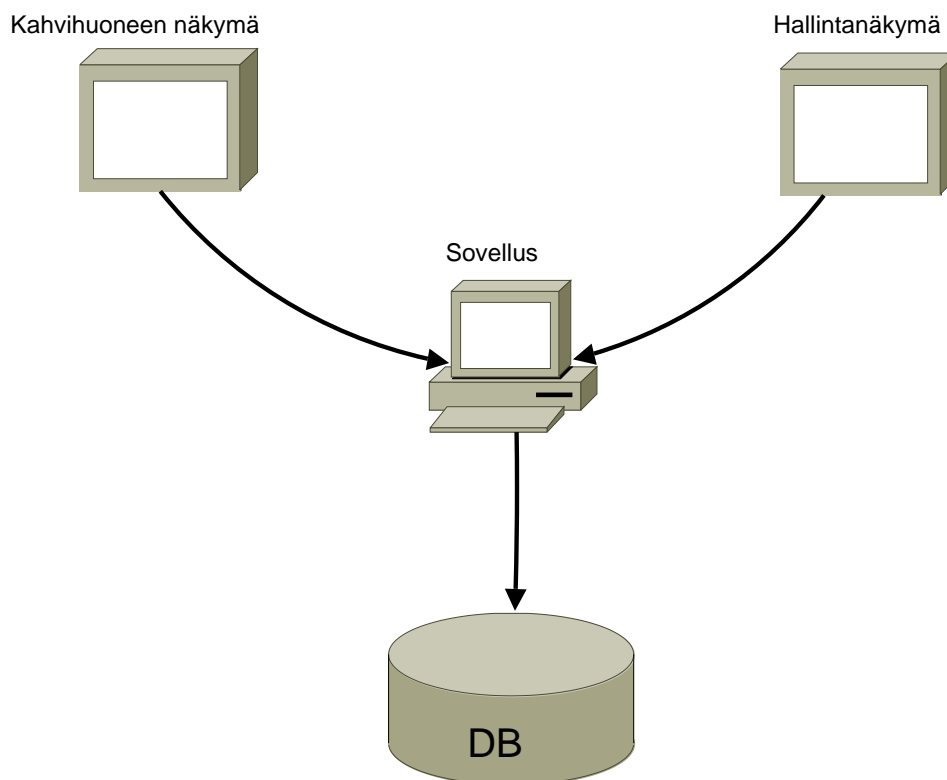
**Tuontituote** Jokin järjestelmän mahdollisista tuontituotteista, esimerkiksi kahvipaketti ja suodatinpakkaus ovat tuontituotteita

**Viestinvälitysolio** Päätietokannan ja sovellusten välillä liikkuvat DB-alkuiset oliot, joilla tietokannan tietosisältöä luetaan ja muutetaan

## 2 Järjestelmäarkkitehtuuri

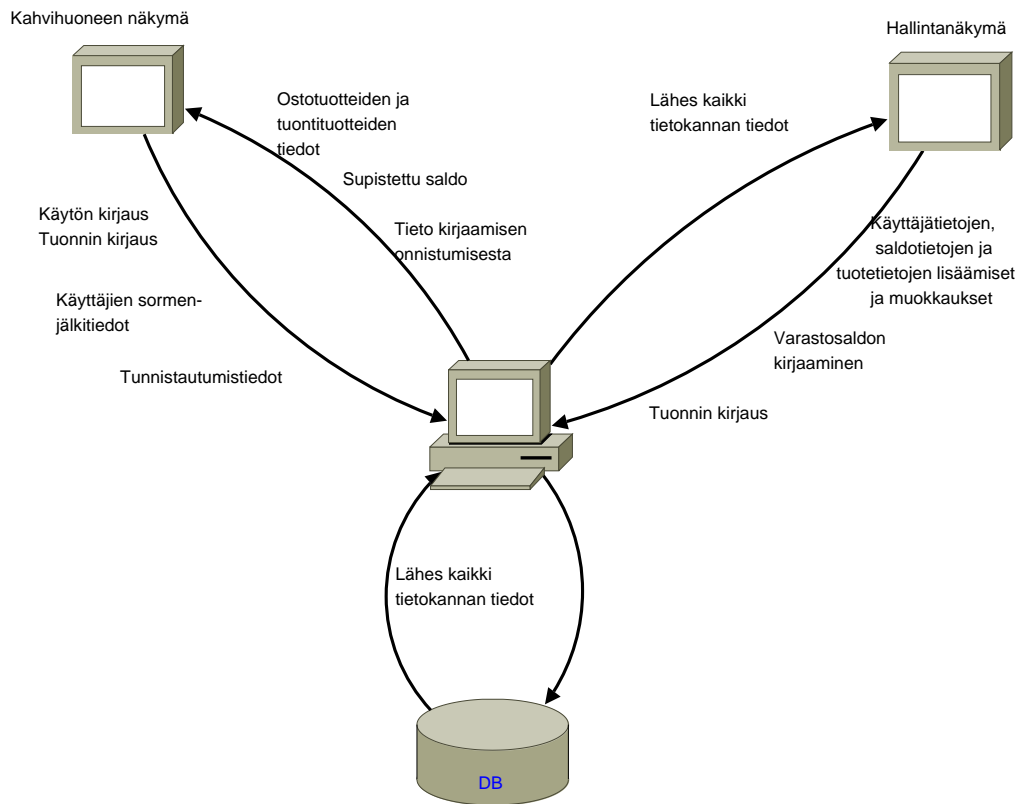
Järjestelmä koostuu pääsovelluksesta sekä tietokannasta. Sovellukseen on kaksi erillistä näkymää, kahvihuoneen näkymä ja hallintanäkymä. Kahvihuoneen näkymässä näkyy yksinkertainen käyttöliittymä tuotteiden käytön sekä tuonnin kirjaamiseen. Hallintanäkymän avulla saadaan järjestelmään lisättyä uusia käyttäjiä sekä tuotteita. Näkymällä saadaan myös muutettua olemassa olevien käyttäjien sekä tuotteiden tietoja. Päätietokantaan talletetaan kaikki tarvittava tieto järjestelmän käyttäjistä ja tuotteista sekä tarvittavat tiedot tuotteiden kulutuksen seurantaan.

### 2.1 Arkkitehtuurikaavio



Kuva 1: Järjestelmän arkkitehtuuri

Kuvassa 1 on kuvattu järjestelmän arkkitehtuuri korkealla tasolla. Täysin erillisiä komponentteja ovat käytännössä vain sovellus sekä tietokanta. Näkymät ovat vain sovelluksen erilaisia käyttöliittymiä, joista soveltuvampi avataan tarpeen mukaan. Kuvassa nuolet kuvaavat palvelupyynnöitä.



Kuva 2: Tiedon kulku osajärjestelmien välillä

## 2.2 Tiedon kulku osajärjestelmien välillä

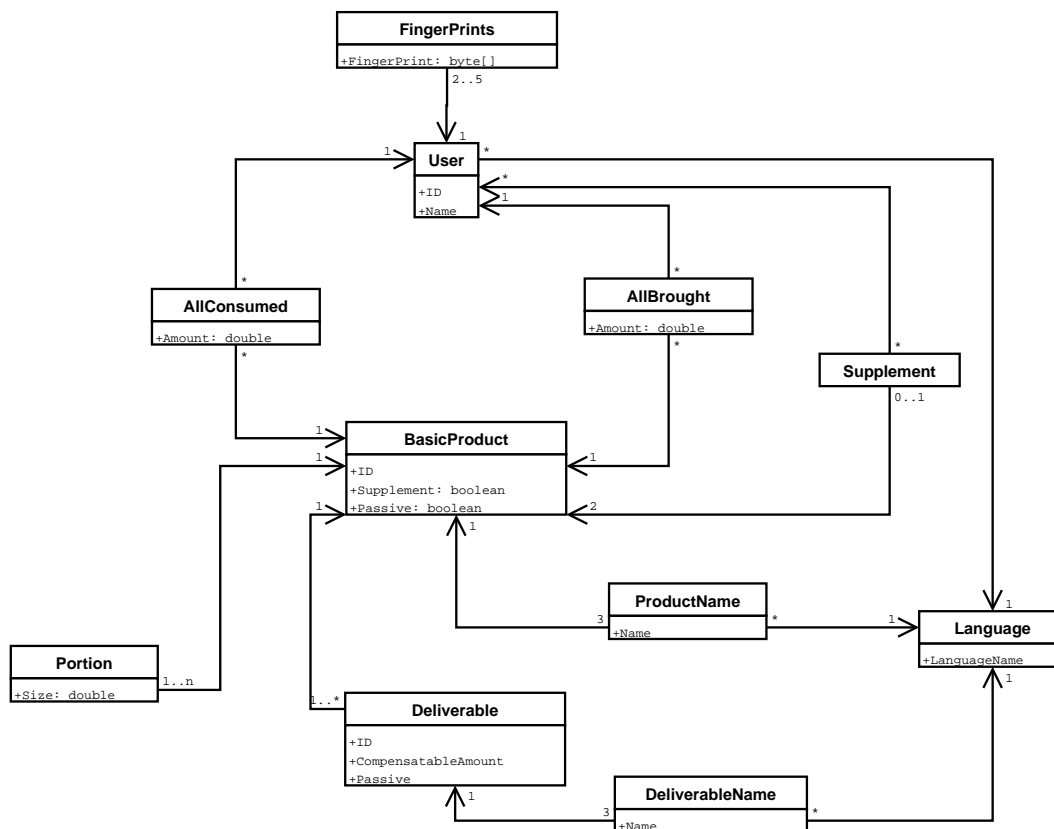
Kuvassa 2 on esitelty eri tiedot mitä järjestelmässä kulkee. Kuvassa nuolet kuvaavat suuntaa johon tieto kulkee. Supistettu saldo kuvaa saldotietoa, jonka käyttäjä saa palautteena onnistuneesta kirjauksesta kahvihuonenäkymässä.



### 3 Tietokanta

Järjestelmän tietokantana toimii tietojenkäsittelytieteen laitoksen yleisellä tietokantapalvelimella pyörivä Oracle-tietokanta. Tässä aliluvussa on kuvattu tietokannan tietosisältö ja sen looginen rakenne. Tietokannan luontiin tarvittavat SQL-luontilauseet ovat dokumentin liitteenä.

#### 3.1 Tietokantakaavio



Kuva 3: Tietokantakaavio

Oheisessa kaaviossa (Kuva 3) on kuvattu tietokannan looginen rakenne sekä sen taulujen väliset riippuvuudet. Taulujen attribuutit eivät ole täysin samat, kuin mitä lopullisessa tuotteessa.

#### 3.2 Tietokannan taulut

Seuraavaksi on esitelty tietokannan eri taulut. Selvityksenä on taulun käyttötarkoitus sekä sen attribuuttien tulkinta. Attribuuttien nimeäminen ei ole täysin yhtenevä liitteenä olevien SQL-luontilauseiden kanssa, mutta nimen tulkinta selvinnee asiayhteydestä.

## Language

---

Taulu järjestelmän eri kielivalintoja varten.

languageName kielen nimi

## Users

---

Taulu käyttäjän perustietoja varten.

user\_ID Käyttäjän yksilöllinen ID

name Käyttäjän nimi

passive Tieto siitä, onko käyttäjä poistettu

languageName Käyttäjän käyttämä kieli

## FingerPrints

---

Taulu käyttäjän sormenjälkiä varten.

fingerPrint Sormenjälki tavumuodossa

userId Käyttäjä jonka sormenjälki on kyseessä

## BasicProduct

---

Taulu järjestelmän perustuotteita varten.

productId Perustuotteen yksilöllinen ID

supplement Tieto siitä, onko tuote lisuke

passive Tieto siitä, onko tuote passivoitu

## Portion

---

Taulu järjestelmän juotavien tuotteiden annoskokoja varten.

sizeOf Annoksen koko

productId Tuote, jota annostieto koskee

## Supplement

---

Taulu järjestelmän käyttäjien lisukepreferenssejä varten.

userId Käyttäjä jota lisukevalinta koskee

productId Tuote jonka kanssa lisuketta käytetään

supplementId Lisuke jota käytetään

## Deliverable

---

Taulu järjestelmän tuontituotteita varten.

deliverableId Tuontituotteen yksilöllinen ID

compensatableAmount Hyvitettävä määrä

productId Hyvitettävä tuote

Passive Tieto siitä, onko tuontituote passivoitu

**ProductName**

---

Taulu järjestelmän perustuotteiden nimiä varten.

name	Nimi
productId	Tuote jonka nimi on kyseessä
languageName	Kieli jolla nimi on esitetty

**DeliverableName**

---

Taulu järjestelmän tuontituotteiden nimiä varten.

name	Nimi
deliverableId	Tuote jonka nimi on kyseessä
languageName	Kieli jolla nimi on esitetty

**AllConsumed**

---

Taulu järjestelmän käyttäjien kulutuksen seurantaan.

amount	Se määrä, paljonko käyttäjä on kuluttanut tuotetta
userId	Käyttäjä, jota tieto koskee
productId	Tuote, jota tieto koskee

**AllBrought**

---

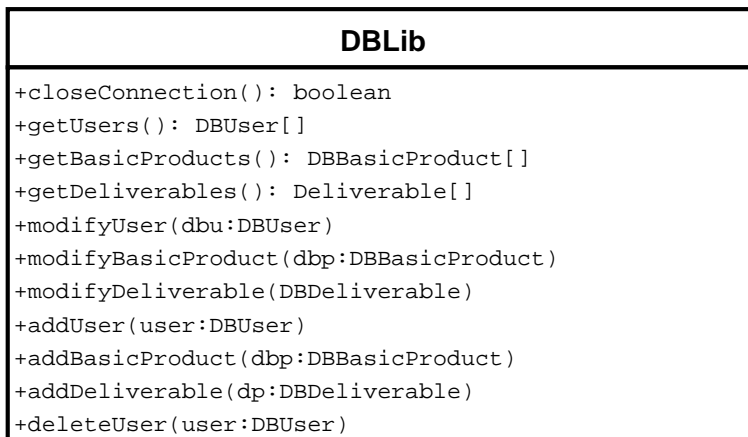
Taulu järjestelmän käyttäjien tuontien seurantaan.

amount	Se määrä, paljonko käyttäjä on tuonut tuotetta
userId	Käyttäjä, jota tieto koskee
productId	Tuote, jota tieto koskee

## 4 Tietokantakirjasto

Tietokantakirjasto on kirjasto operaatioita joita käytetään tiedon siirtämiseen tietokannasta sovelluksille. Kirjastosta luodaan sitä käyttävälle sovellukselle ilmentymä, jonka jälkeen tietokannalta voi kirjaston palveluita hyväksikäyttämällä pyytää sovelluksien tarvitsemia tietojoukkoja. Kirjasto mahdollistaa myös tietojen päivittämiseen kirjastoon. Mikäli jokin kirjaston operaatio ei onnistu, heittää kirjasto poikkeuksen `DBException`. Tiedot tietokannasta liikkuvat DB-alkuisina olioina, jotka tarjoavat operaatiot omien tietojensa käsittelyyn.

### 4.1 Luokkakaavio



Kuva 4: Tietokantakirjaston palvelut

Kaaviossa 4 on kuvattu tietokantakirjaston palvelut. Kirjaston välittämät oliot ovat viestinvälitysolioita, jotka esitellään seuraavassa luvussa.

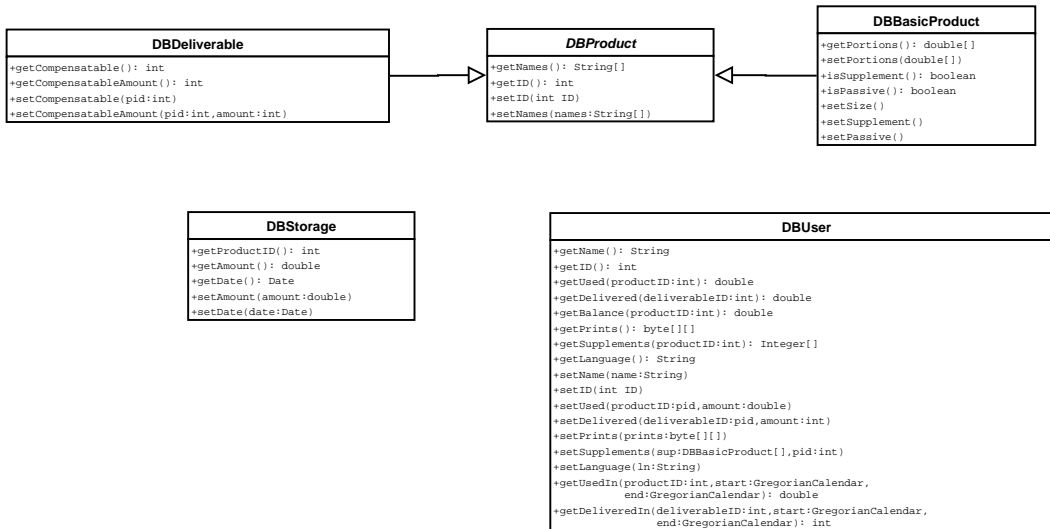
### 4.2 Palveluiden tarkempi määrittely

Kaikki operaatiot heittävät `DBException`-poikkeuksen mikäli tietokantaoperaatiossa tapahtuu virhe. Exception sisältää virheilmoitukset luonnollisella kielellä suomeksi, ruotsiksi ja englanniksi. Palvelut on kuvattu yksityiskohtaisesti liitteenä olevassa sovellusohjelmoijan rajapintakuvauksessa.

## 5 Viestinvälitysoliot

Viestinvälitysoliot ovat olioita, joihin ”pakataan” tietokannan tietosisältö DBLibin toimesta. Näistä olioista periytetään itse sovelluksen käyttämät niin sanotut ”oikeat” oliot, kuten User ja BasicProduct. Idea on, että viestinvälitysoliot sisältävät lähinnä vain akessoreita sovelluksen tarvitsemiin tietoihin, kun taas ”oikeat” oliot täydentävät palveluita sovelluksen tarvitsemilla palveluilla.

### 5.1 Luokkakaavio



Kuva 5: Tietokantakirjaston ja sovelluksen väliset viestinvälitysoliot

Kaaviossa 5 on esitelty tietokantakirjastosta sovelluksille kulkevat viestinvälitysoliot. Viestinvälitysolioiden kentissä kulkee koko tietokannan tietosisältö.

**DBProduct** on kaikkien tuotteiden abstrakti ylliluokka. **DBBasicProduct** ja **DBDeliverable**-luokat kapseloivat tuotteisiin liittyvät tiedot, **DBUser** taas sisältää kaikki käyttäjiin liittyvät tiedot. **MetaData** on koko järjestelmää koskevaa tietoa, esimerkiksi perustuotteiden huomautusraja on tällaista tietoa.

### 5.2 Luokkien esittelyt

Seuraavaksi on esitelty tarkemmin eri viestinvälitysolioiden käyttötarkoitukset sekä tietosisällöt luokkakohtaisesti.

#### 5.2.1 DBUser

Luokka järjestelmän käyttäjän tietojen kapselointiin. Sisältää yksilöivät tiedot käyttäjästä sekä käyttäjän osalta perustuotteiden tuonti-, kulutus- ja lisuketiedot.

### **5.2.2 DBProduct**

Abstrakti ylliluokka järjestelmän tuotteille. Sisältää tuotteille yhteiset ominaisuudet, esimerkiksi tuotteen yksilöivän tunnusteen sekä tiedon siitä onko tuote mahdollisesti passiivitu.

### **5.2.3 DBBasicProduct extends DBProduct**

Luokka järjestelmän perustuotteille. Sisältää tiedon siitä, onko tuote lisuke sekä juotavien tuotteiden osalta osaa kertoa eri annoskoot.

### **5.2.4 DBDeliverable**

Luokka järjestelmän tuotaville tuotteille. Sisältää tiedot tuotteesta sekä siitä, mitä perustuotetta tämän tuotteen tuominen hyvittää ja kuinka paljon.

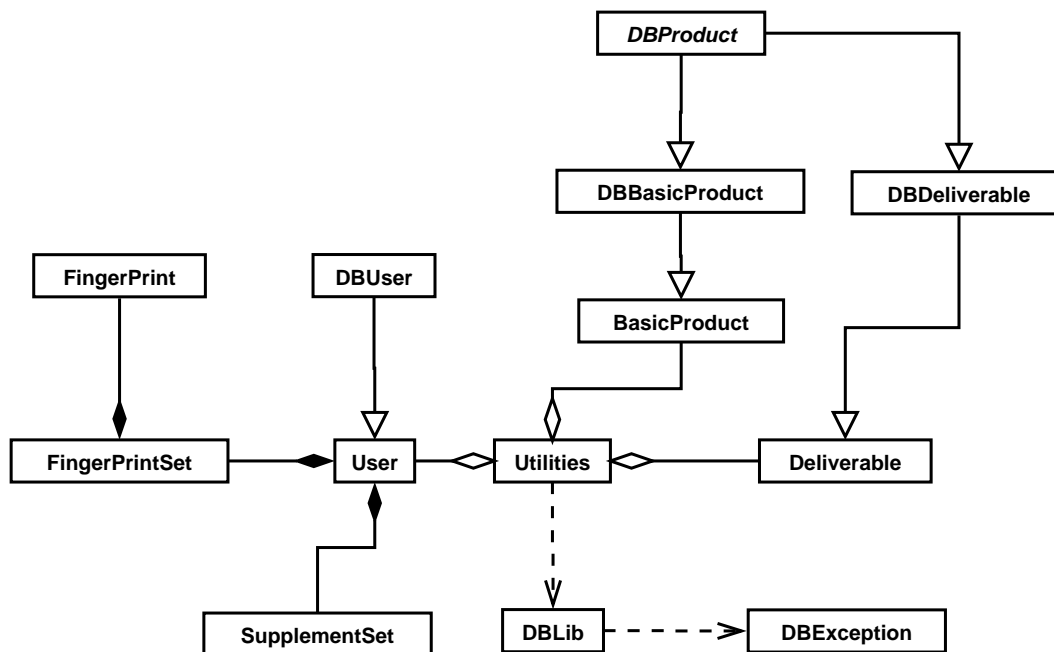
## 6 Pääsovellus

Seuraavaksi on esitelty järjestelmän pääsovelluksen käyttämät luokat sekä esitelty niiden toimintaa. Aluksi on esitelty sovelluksen käyttämät yleiset apuluokat ja tämän jälkeen sovelluksen käyttöliittymän eri näkymät. Päänäkymistä on kerrottu pääpiirteittäin, mitä erilaisia valmiita GUI-komponentteja näkymän rakentamiseen on käytetty. Myös kaikki sovelluksen luokat on esitelty, jaoteltuna sen suhteen, missä näkymässä niitä enimmäkseen käytetään. Luokkien riippuvuudet sekä käyttösuhteet menevät jonkin verran ristiin, mutta tätä on koitettu selvittää useammalla luokkakaavioilla. Jokaisesta luokasta on kirjoitettu myös lyhyehkö esittely luokan toiminnasta sekä käyttötarkoituksesta. Tarkat palvelumäärittelyt löytyvät liitteenä olevasta sovellusohjelmoijan rajapintakuvauksesta.

### 6.1 Yleiset apuluokat

Yleiset apuluokat ovat sovelluksen käyttämiä luokkia, joissa tiedon käsittely tapahtuu. Sovelluksella on omat olionsa käyttäjiä ja tuotteita varten, samoin esimerkiksi käyttäjien sormenjäljet on kapseloitu omaksi luokakseen. Myös koko sovelluksen eniten käytetty luokka Utilities kuuluu apuluokkiin.

#### 6.1.1 Luokkakaavio



Kuva 6: Sovelluksen apuluokat

Kaaviossa 6 on esitelty sovelluksen käyttämien apuluokkien sisältyvyudet ja riippuvuudet. Siinä on myös tietokantaan liittyvä kirjasto sekä viestinvälitysoliot esiteltyinä, sillä

ne liittyvät tiiviisti apuluokkien toimintaan. Käytännössä tapahtumien kulku menee seuraavasti: Utilities-luokka pyytää DBLib-luokalta tarvitsemansa tuote- sekä käyttäjätiedot jotka tulevat viestinvälitysoliona. Utilities muuntaa viestinvälitysoliot sovelluksen sisäiseksi olioksi – esimerkiksi ”User” – ja tallentaa ne omaan tietorakenteisiinsa.

### **6.1.2 User extends DBUser implements Comparable**

Luokka on tarkoitettu yksittäisen käyttäjän tietojen kapselointiin. Luokka täydentää DBUser-luokan palvelujoukkoa muutamalla sovelluskohtaisella lisäpalvelulla.

### **6.1.3 BasicProduct extends DBBasicProduct**

Tämä luokka taas on tarkoitettu yksittäisen perustuotteen tietojen kapselointiin. Luokka laajentaa DBBasicProduct-luokan palveluita muutamalla sovelluskohtaisella palvelulla.

### **6.1.4 Deliverable extends DBDeliverable**

Yksittäisen tuontituotteen tietojen kapselointiin. Tämäkin luokka laajentaa DBDeliverable-luokan palveluita.

### **6.1.5 FingerPrint**

Luokka yksittäisen käyttäjän yksittäisen sormenjäljen kapselointiin. Tarjoaa sormenjälkien vertailuun tarvittavan equals-metodin, joka käyttää hyväkseen sormenjälkitunnistimen kirjastoluokan tunnistusrutiinia.

### **6.1.6 FingerPrintSet**

Luokka yksittäisen käyttäjän eri sormenjälkien säilyttämiseen ja kapselointiin.

### **6.1.7 SupplementSet**

Luokka yksittäisen käyttäjän lisukepreferenssien säilyttämiseen ja kapselointiin.

### **6.1.8 Utilities**

Utilities on sovelluksen käyttämä staattinen apuluokka, joka tarjoaa palveluita lähinnä hallintanäkymän MainFrame- sekä View-luokille. Utilities-luokalla on tietorakenteissaan tallessa tietokannan tietosisältö ja suurin osa sen palveluista liittyy tietokannan sisällön tutkimiseen ja muokkaamiseen. Konkreettinen tietokantayhteys on hoidettu luokan



DBLib ilmentymällä. Luokalla on myös palveluita sovelluksen monikielisyyden mahdollistamiseen. Esimerkiksi palvelu `getSelectedLanguage` palauttaa sovelluksessa sillä hetkellä käytössä olevan kielivalinnan. `getString`-metodi taas antaa sovelluksen valitulla kielellä olevan merkkijonoesityksen tietystä avain-merkkijonosta. Jos sovelluksessa valittuna oleva kieli on esimerkiksi suomi, palauttaa `getString("new_basicproductname")` arvon "Uusi tuote". Mikäli valittu kieli on englanti, palauttaa metodi arvon "New product".

## 6.2 Rajapintaluokka

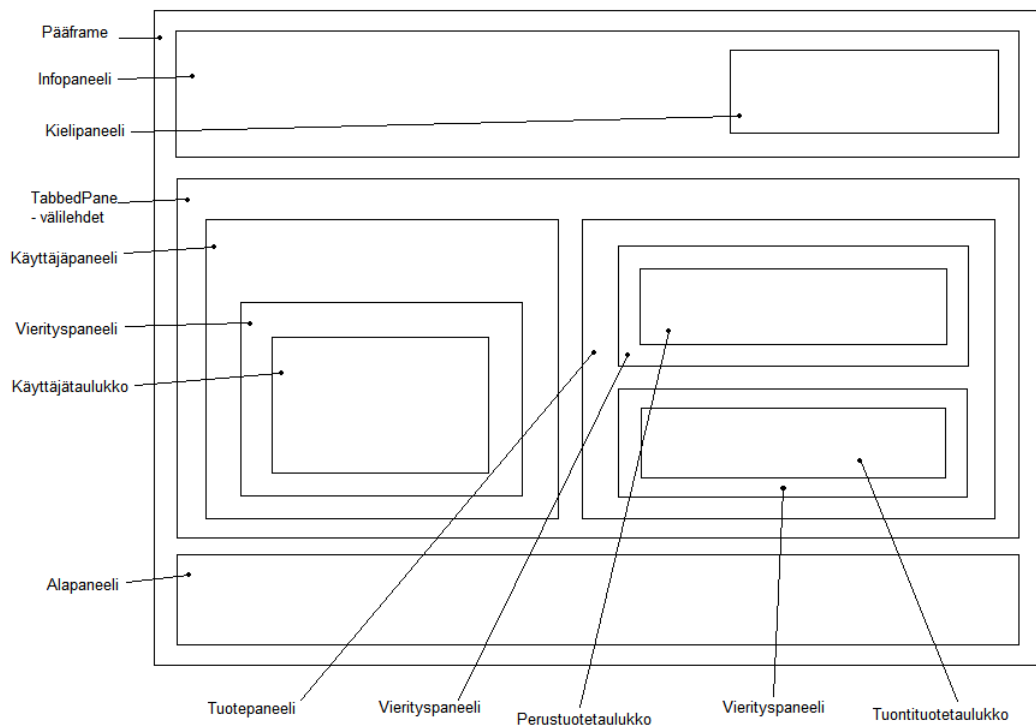
### 6.2.1 Extractor

Extractor on rajapintaluokka, jonka toteuttavat kaikki sormenjälkiä tarvitsevat dialogit. Eli käytännössä luokat `RegisterFingersDialog` sekä `CoffeeDialog`.

## 6.3 Hallintanäkymän käyttämät luokat

Seuraavaksi on esitelty hallintanäkymässä käytettävät luokat yksityiskohtaisemmin.

### 6.3.1 Käyttöliittymän komponenttikuvaukset



Kuva 7: Hallintanäkymän komponentit

Kuvassa 7 on esitetty hallintänäkymässä käytettyjä Swing-komponentteja, lähinnä näkymän sisältöelementit. Näkymän yläosassa on näyttötilaa mahdollisille virheilmoituksille sekä valintapainikkeet sovelluksen käyttökielen vaihtamiseen. Tämän tilan alla on käyttöliittymän pääosa, joka koostuu kahdesta erillisestä välilehdestä. Toinen välilehti on tarkoitettu käyttäjien hallinointia varten ja toinen välilehti tuotteiden hallinointia varten. Käyttäjä-välilehti on aktiivisena kun sovellus käynnistetään.

Käyttäjä-välilehti on käytännössä vain yksi suuri taulukko. Taulukon rivit kuvastavat eri käyttäjiä ja sarakkeet käyttäjien eri attribuutteja. Käyttäjään suoraan liittyviä sarakkeita ovat käyttäjän nimi, ID-tunniste, sormenjäljet ja käyttäjän käyttämä kieli. Yhdessä sarakkeessa on valintaruutu, jonka valitsemalla käyttäjä poistetaan järjestelmästä tallennuksen yhteydessä. Seuraavina sarakkeina on tiedot käyttäjän eri tuotteiden saldosta sekä yksilöidyt tiedot tuoduista ja kulutetuista tuotteista. Viimeisinä sarakkeina taulukossa on käyttäjän lisukepreferenssit, tuotteen tuomiseen liittyvät kentät sekä painike käyttäjätietojen tallentamista varten. PIN-saraketta sekä saldoa lukuunottamatta kaikki käyttäjän attribuutit ovat muokattavissa.

Tuotteet-välilehti taas koostuu kahdesta taulukosta joista toisella hallitaan perustuotteita ja toisella tuontituotteita. Näissäkin taulukoissa rivit kuvastavat tuotteita ja sarakkeet näiden attribuutteja. Näissä taulukoissa samoja sarakkeita ovat tuotteen suomen-, ruotsin- ja englanninkieliset nimet sekä tieto siitä, onko tuote passivoitu. Perustuotetaulukossa on tämän lisäksi tieto siitä, onko tuote lisuke sekä tuotteen eri annoskoot. Lisukkeilla on vain yksi annoskoko. Tuontituotetaulukossa on yhteisten sarakkeiden lisäksi sarake, joka kertoo perustuotteen minkä kulutusta tuontituotteen tuonti hyvittää. Hyvittävä määrä määritellään taulukon seuraavassa sarakkeessa. Taulukossa on myös sarake tuontituotteen pakkauksen koolle (esimerkiksi 500g tai 100kpl).

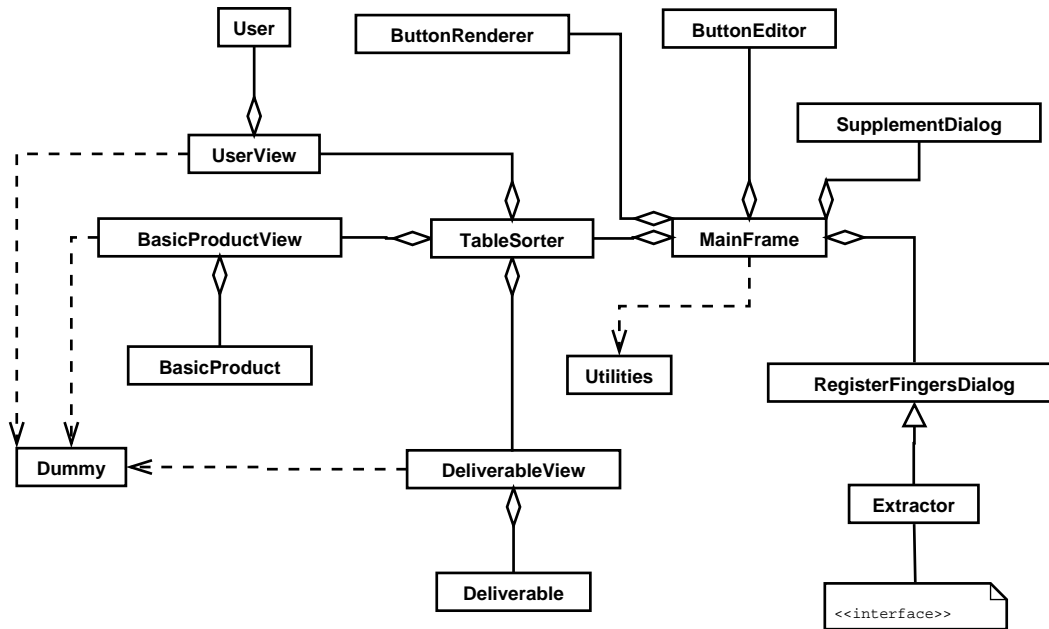
Kaikkien kolmen pääosassa olevan taulukon viimeinen rivi on tyhjä. Täyttämällä viimeisen rivin tiedot ja painamalla tallennuspainiketta voidaan järjestelmään lisätä uusia käyttäjiä ja tuotteita. Näkymän alaosassa on painikkeet kahvisaldoraportin kirjoittamiseen sekä kahvihuoneen näkymän avaamiseen.

### 6.3.2 Luokkakaavio

Kuvassa 8 on esitelty hallintänäkymään tiivissä yhteydessä olevat luokat. View-luokat ovat Javan taulukkokomponentin, *JTable*n, vaatimia tietomalleja taulukon tietosisällöstä. Luokat toteuttavat *AbstractTableModel*-rajapinnan, joka määrittelee palvelut joita taulukon tietomallin tulee toteuttaa. Eräs esimerkki tällaisesta palvelusta on *getValueAt*, joka saa parametrinaan solun koordinaatit taulukossa ja palauttaa kyseisissä koordinaateissa sijaitsevan olion kutsujalle. *TableSorter* on apumalli, joka pitää käyttäjälle näkyvän taulukon järjestyksessä.

### 6.3.3 MainFrame extends JFrame

Sovelluksen pääohjelmaluokka. Hoitaa sovelluksen käyttöliittymän komponenttien alustamisen ja piirtämisen sekä tarvittavien tapahtumakuuntelijoiden asettamisen. Hoitaa myös



Kuva 8: Hallintanäkymään tiiviisti liittyvät luokat

sovelluksen tietosisällön päivittämisen. Päivitys tapahtuu aina kun sovelluksen käyttökiel-  
tä vaihdetaan tai tietokantaan tallennetaan muutoksia.

Luokka käyttää Utilities-luokan palveluita tietomallien tietosisällön päivittämiseen. Main-  
Framen ”refreshTables”-metodi päivittää View-olioiden tietosisällön näiden palveluiden  
avulla. Luokka käyttää Utilities-luokalta myös monikielisyteen liittyviä palveluita.

#### 6.3.4 UserView extends AbstractTableModel

Käyttäjähallintaan tarvittavan taulukon tietomalli. Mallin keskeinen tietosisältö koostuu  
User-tilistä jonka jokainen alkio kuvaa käyttäjätilin käyttäjätietojen yhtä riviä.  
User-tilin kentät taas kuvaavat taulukon sarakkeita. Mallin getValueAt-metodi toimii si-  
ten, että se valitsee rivinumeron perusteella User-tilin alkion ja kutsuu sarakenu-  
meron perusteella tälle tilille haluttua aksessoria. Eli jos kutsuja haluaa viidennen  
käyttäjän nimitiedon (kolmas sarake), kutsuu hän palvelua getValueAt(4,2). UserView pa-  
lauttaa tällöin arvon users[4].getName(). Tässä on oletettu, että mallin User-tili-  
luokkaan viittaava muuttuja on *users*.

#### 6.3.5 BasicProductView extends AbstractTableModel

Perustuotteiden hallintaan tarvittavan taulukon tietomalli. Tietosisältö ja palvelut ovat sa-  
mankaltaiset kuin UserView:ssäkin. Taulukon rivit koostuvat BasicProduct-tilioista joita  
malli säilyttää BasicProduct-tililuokassa.

### 6.3.6 DeliverableView extends AbstractTableModel

Tuontituotteiden hallintaan tarvittavan taulukon tietomalli. Tietosisältö ja palvelut ovat samankaltaiset kuin muissakin sovelluksen malleissa. Taulukon rivit koostuvat Deliverable-olioista joita malli säilyttää Deliverable-tilauksessa.

### 6.3.7 ButtonRenderer extends JButton implements TableCellRenderer

ButtonRenderer on JTable-olion tietyn solun arvojen "piirtäjä", joka on kustomoitu piirtämään solu JButton-komponentin muodossa. ButtonRenderer esittää FingerPrintSet-arvot sormenjälkien lukumääränä painikkeen tekstissä. SupplementSet- ja Dummy-arvoja piirtävä ButtonRenderer käyttää konstruktorissa annettua String-oliota painikkeen tekstinä.

### 6.3.8 ButtonEditor extends AbstractCellEditor implements TableCellEditor, ActionListener

ButtonEditor on JTable-olion tietyn solun arvojen muokkaaja. Olio piirtää solun sinä aikana, kun arvoa muokataan. Lisäksi solu tarjoaa käyttäjälle tavan itse muokkauksen toteuttamiseen. Käytännössä solu piirretään kuten ButtonRenderinkin tekee, ja arvojen muokaus FingerPrintSet- ja SupplementSet-arvojen kohdalla tapahtuu niitä palauttavien dialogien avulla. ButtonEditor siis huolehtii solun piirtämisestä, kun dialogi on esillä. Dummy-arvojen editori ei anna käyttäjän editoida mitään, vaan palauttaa suoraan uuden Dummy-arvon. Käyttäjälle tämä näkyy normaalina painikkeen painamisena.

### 6.3.9 RegisterFingersDialog extends JDialog Implements: FingerCallBack, ImageCallBack, StatusCallBack, Extractor

Käyttäjän sormenjälkien rekisteröinnissä käytettävä popup-dialogi. Pyytää käyttäjältä vähintään 3 sormenjälkeä ja enintään 5 sormenjälkeä. Rajapinnat jotka dialogi toteuttaa, ovat Griaulen valmiin kaluston määrittelemät sormenjälkien lukemisessa tarvittavat rajapinnat. Extractor-rajapinta on rajapinta, jonka jokainen sormenjälkeä vastaanottava dialogi toteuttaa. Dialogi kutsutaan esiin ButtonEditorin toimesta, kun sen tapahtumakuuntelija huomaa, että sormenjälkinappia on painettu.

### 6.3.10 SupplementDialog extends JDialog

Käyttäjän lisuketietojen rekisteröintiin käytettävä popup-ikkuna.

### 6.3.11 TableSorter extends AbstractTableModel

Luokka on lähes suoraan saatu Javan ohjeistuksesta. TableSorteria voi käyttää JTable:ssa ns. "välimallina", se ei itsessään sisällä tietomallin tietoja, mutta se pitää niiden järjestyksestä kirjaa. Eli kun JTable haluaa jonkun arvon tietomalliltaan, kutsuu se TableSorterin

getValueAt-metodia, joka taas kuvaa pyydytyt koordinaatit oikean tietomallin koordinaateiksi. Kuvaukseen se käyttää omia tietorakenteitaan, joihin on siis tallennettu taulun järjestys.

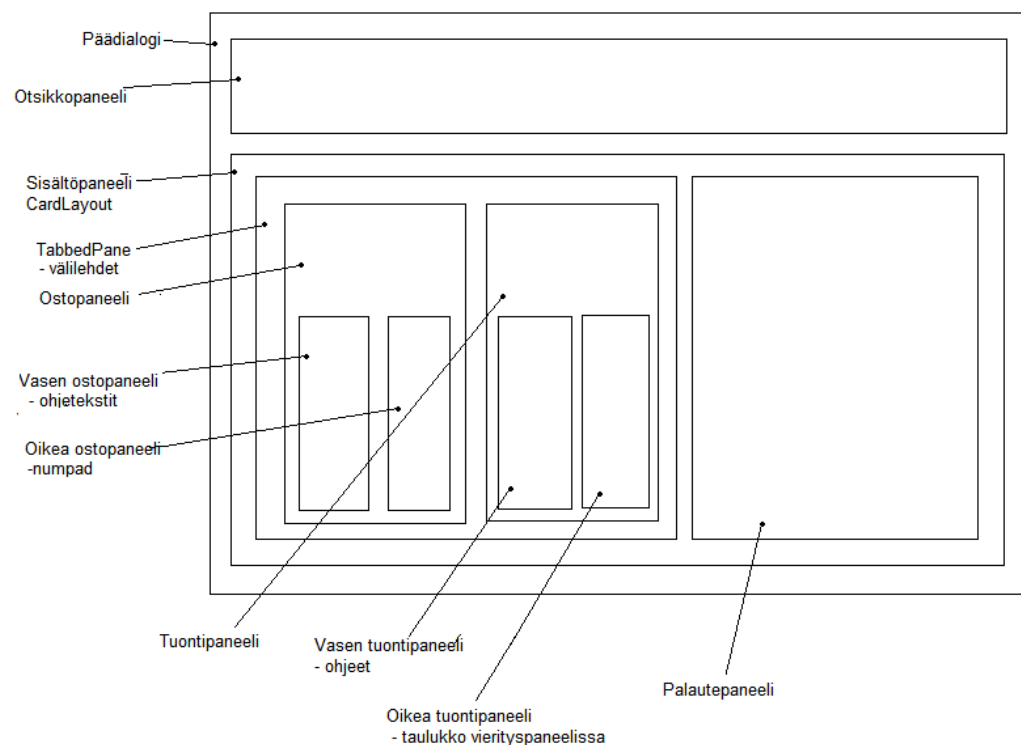
### 6.3.12 Dummy

Dummy on käytännössä vain tyhjä apuluokka, jota taulukon tietomallit käyttävät sen havaitsemiseen, että taulukon viimeisessä sarakkeessa olevaa tallenna-painiketta painetaan. Luokkaan on tallennettu myös tieto siitä, onko kyseistä taulukon riviä muokattu.

## 6.4 Kahvinäkymän käyttämät luokat

Seuraavaksi on esitelty pelkästään kahvinäkymän käyttämät luokat sekä niiden käyttötarkoitus yksityiskohtaisemmin.

### 6.4.1 Käyttöliittymän komponenttikuvaukset



Kuva 9: Kahvinäkymän komponentit

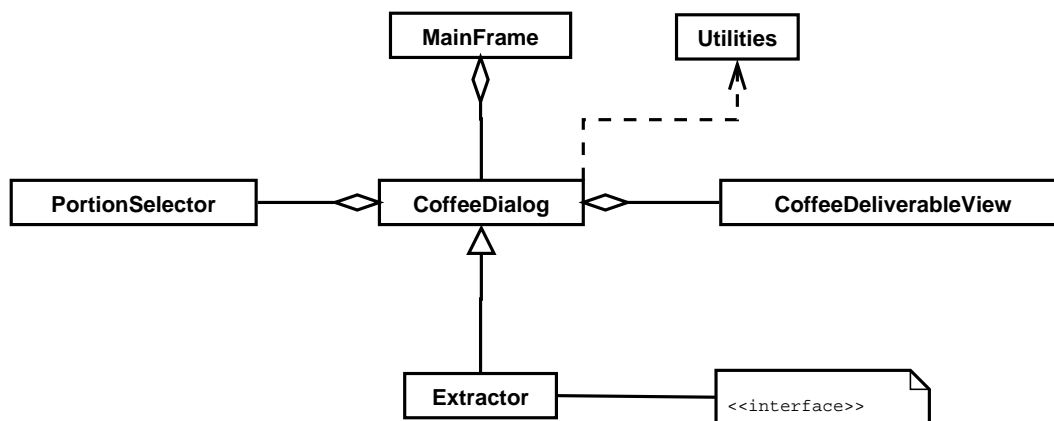
Kahvinäkymän GUI-komponentit on esitetty kuvassa 9.

Näkymän yläosassa on näkymän otsikkopaneeli. Otsikkopaneelilla ei sinänsä ole mitään informatiivista sisältöä. Paneelin alapuolella oleva sisältöpaneeli koostuu käytännössä kahdesta näkymästä, joiden vuorottaminen on hoidettu *CardLayout*-asettelunhallinnalla. Toinen näkymistä on näkymän perustila ja toinen on palautetila.

Perustila koostuu kahdesta välilehdestä, joista toinen on tarkoitettu tuotteiden ostoa varten ja toinen tuotteiden tuontia varten. Ostovälilehdessä vasemmalla puolella on välilehden ohjetekstit ja oikealla puolella on ohjeellinen numpad-näppäimistö. Numpad on oma olionsa, joka on periytetty *JPanel*-komponentista. Tuontivälilehti koostuu myös vasemmalla puolella olevista ohjeteksteistä, mutta tämän välilehden vasemmalla puolella on *JTable*-komponentti, jossa on listattu mahdolliset tuotavat tuotteet.

Palautetilassa on vain muutama *JLabel*-komponentti, joihin täytetään onnistuneen kirjauksen jälkeen palautetekstit käyttäjälle. Näkymä siirtyy palautetilasta perustilaan automaattisesti viiden sekunnin kuluttua palautteen näyttämisestä, sekä minkä tahansa näppäimen painalluksella.

#### 6.4.2 Luokkakaavio



Kuva 10: Kahvinäkymän käyttämät luokat

Kuvassa 10 on esitelty kahvinäkymän pääasiallisesti tarvitsemat luokat.

#### 6.4.3 CoffeeDialog extends JDialog implements Extractor

Luokka rakentaa kahvihuonenäkymän tarvitsevan GUI:n. Se käyttää sormenjälkientunnistusta sekä monikielisyyspalveluita Utilities-luokalta.

#### 6.4.4 CoffeeDeliverableView implements AbstractTableModel

Tietomalli tuotavien tuotteiden taulukkoa varten. Käytännössä täysin analoginen hallintanäkymän vastaavan kanssa.

#### 6.4.5 PortionSelector extends JPanel

PortionSelector on JPanelista periytetty oma komponentti, joka esittää numeronäppäimistöä. PortionSelector sisältää kaksi näppäinryhmää, joissa molemmissa on aina yksi näppäin valittuna (JToggleButton ja ButtonGroup), muut painikkeet ovat JButtonin kaltaisia. Ensimmäisen ryhmän muodostavat kolme kielinäppäintä ja toisen numeronäppäimet. Numeronäppäimiä käytetään valitsemaan ostettava juoma. PortionSelector tarvitsee tiedot juomavaihtoehtoista sopivaa alustusta varten.

## **Liite 1. Tietokannan alustuslauseet**

Tietokannan alustuslauseet liittyvät liitteenä olevasta tiedostosta tietokanta.sql.