

# **AINEOPINTOJEN HARJOITUSTYÖ: TIETOKANNAT**

Teemu Matvejeff

## Sisällys

AINEOPINTOJEN HARJOITUSTYÖ: TIETOKANNAT .....	1
1. JOHDANTO .....	3
2. YLEISKUVA JÄRJESTELMÄSTÄ .....	4
3. KÄYTTÖTAPAUKSET .....	4
4. JÄRJESTELMÄN TIETOSISÄLTÖ .....	5
5. RELAATIOTIETOKANTA .....	6
6. JÄRJESTELMÄN YLEISRAKENNE .....	7
7. JÄRJESTELMÄN KOMPONENTIT .....	8
7.1 MALLI-LUOKAT .....	8
7.2 SERVLETIT .....	8
7.3 JSP-TIEDOSTOT .....	9
7.4 TAG-TIEDOSTOT .....	9
7.5 HTML-TIEDOSTOT .....	10
7.6 CSS-TIEDOSTOT .....	10
8. KÄYTTÖLIITTYMÄ .....	10
9. ASENNUSTIEDOT .....	10
10. KÄYNNISTYS- JA KÄYTTÖOHJEET .....	10
11. TESTAUS, LÖYDETYT BUGIT, PUUTTEET & JATKOKEHITYSIDEAT .....	11
12. OMAT KOKEMUKSET .....	11
13. LIITTEET .....	<b>Virhe. Kirjanmerkkiä ei ole määritetty.</b>

## 1. JOHDANTO

Järjestelmän tarkoituksena on toimia elektronisena keittokirjana, johon on mahdollista tallentaa tietoa erilaisista resepteistä. Data pitää sisällään vähintään ruokalajin nimestä, valmistamiseen käytettävistä raaka-aineista (ns. ruokalaji-tili). Toinen tili sisältää lyhyen kuvauksen ruuasta ja sen valmistusohjeet. Sovellus mahdollistaa reseptien selailun, lisäämisen ja poistamisen tietokannasta, tietojen muokkaamisen ja tietojen selaamisen eli ns. CRUD-nelikossa määritetyt toiminnot.

Tietokantaa pääsy ei ole vapaata ja se vaatii kirjautumisen järjestelmään (yksityiskohdat myöhemmin). Haku toimii reseptin nimellä ja palauttaa jonkin tyyppisen virheilmoituksen jos ehdot täyttävää vaihtoehtoa ei löydy. Tämän lisäksi sovellus tarjoaa mahdollisuuden nähdä listan järjestelmässä olevista resepteistä.

Työn toteuttamiseen käytetään PostgreSQL-tietokantaa ja sovellus kirjoitetaan Java-ohjelmointikielellä.

Tietokannassa on eritasoisia käyttäjiä:

1. Tavallinen käyttäjä
2. Pääkäyttäjä (oikeudet mm. päivittää tietokantaa)

## 2. YLEISKUVA JÄRJESTELMÄSTÄ

Alla on esitetty ensimmäinen versio järjestelmän käyttötapauskaaviosta:

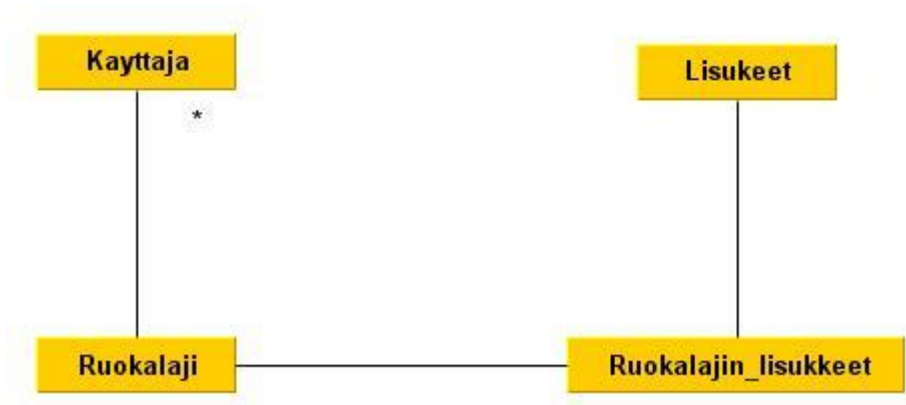


Järjestelmässä on kolmenlaisia käyttäjiä:

1. Peruskäyttäjällä (merkitty kaaviossa vain käyttäjänä) on oikeus selata tietokantaa ja suorittaa hakuja mutta ei mitään muuta.
2. Pääkäyttäjä eli ylläpitäjä huolehtii järjestelmän ylläpidosta ja pitää huolta että siihen laitettu informaatio on ajan tasalla. Perustoimintojen (eli hakujen) lisäksi pääkäyttäjä-oikeudella varustetut käyttäjät voivat poistaa tietoalkioita järjestelmästä, päivittää kyseisiä alkioita tai lisätä uusia alkioita järjestelmään.

## 3. KÄYTTÖTAPAUKSET

#### 4. JÄRJESTELMÄN TIETOSISÄLTÖ

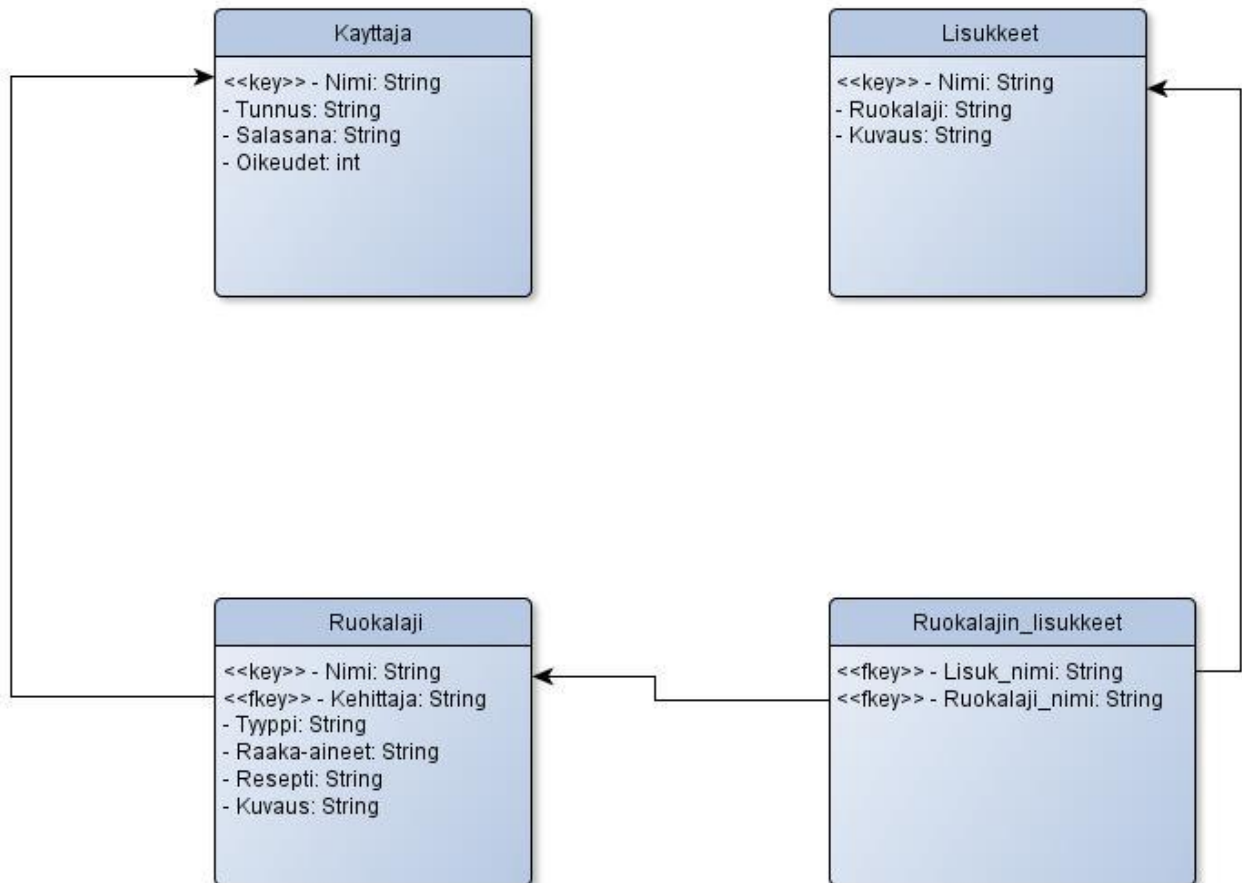


RUOKALAJIN-LISUKKEET	
ATTR	TYYPPI
<u>RUOKALAJI-NIMI</u>	CHA(80)
<u>LISUK_NIMI</u>	- " -

KARTTAJA	
ATTRIBUUTI	TYYPPI
NIMI	CHAR(80)
TUNNUS	CHAR(80)
SALASANA	CHAR(80)
OLKEUDET	INT
<del>FAKSO</del>	

RUOKALAJI	
ATTR.	TYYPPI
NIMI	CHAR(80)
KEHITTAJA	- " -
TYYPPI	- " -
RAAKKA-AINEET	- " -
RESEPTI	- " -
KUVAUS	CHAR(255)

LISUKKEET	
ATTR.	TYYPPI
NIMI	CHAR(80)
KUOKALAJI	CHAR(80)
KUVAUS	CHAR(255)



## 6 JÄRJESTELMÄN YLEISRAKENNE

Sovelluksen suunnittelussa on käytetty ns. MVC-mallia, jossa näkymät, mallit ja kontrollerit on jaoteltu omiin hakemistoihin (hakemistojen nimet on mallit,).

- **Mallitiedostot** kuten Haut.java sisältävät yleisiä metodeja, jotka ovat tärkeitä toiminnallisuuden toiminnan kannalta (tässä tapauksessa erilaisia hakumetodeja). Tämän lisäksi mallitiedostoja ovat tietoja välittävien olioiden luokkatiedostot.
- **Näkymätiedostot** ovat erilaiset jsp-päätteiset tiedostot. Näiden tiedostojen kautta määritellään ruudulla näkyvät verkkosivut ja niiden komponentit kuten laatikot, lomakkeet ja napit.
- **Kontrollerit**, joita servletit ovat, huolehtivat varsinaisesti sovelluksen toiminnasta. Ne hakevat näkymätiedostojen kautta lomakkeiden arvoja ja kutsuvat metodeja asianmukaisissa paikoissa mm. näyttämään uusia sivuja, testaamaan käyttöoikeuksia sekä suorittamaan hakuja.

## 7 JÄRJESTELMÄN KOMPONENTIT

### 7.1 MALLILUOKAT

**ServlettiIsa.java:** Ehkä sovelluksen toiminnan kannalta yleisesti tärkein luokka on ServlettiIsa, joka on periytetty HttpServlet-luokasta. ServlettiIsa toimii puolestaan ylikuokkana kaikille muille sovelluksen servleteille (jotka ovat servlettejä sen perusteella että ovat ServlettiIsa-luokan aliluokkia). ServlettiIsa tarjoaa useita palveluita kuten NaytaJSP-metodin, mikä mahdollistaa käyttäjän ohjaamisen uudelle, parametrina annetulle sivulle. Muita metodeja on mm. käyttäjän kirjautumisen tarkistaminen sekä virheviestien esittämiseen tarkoitettu metodi.

**Haut.java:** Tämä luokka huolehtii sovelluksen hakutoiminnoista. Sovellus ei ole servletti mutta se pitää sisällään hakuihin tarvittavat staattiset metodit, joita servletit kutsuvat. Metodit ottavat vastaan hakupyynnöt String- parametreina, jotka sitten asetetaan PreparedStatement-luokkaa käyttäen valmiiksi muotoiltuihin SQL-kyselyihin. Tämä lähestymistapa valittiin sillä se vähentää SQL-injektioiden vaaraa.

Hakumetodeja on mm. haku reseptin nimellä. Metodin tulokset kääritään varta vasten määritetyn olion sisään, joka välittää tulokset eteenpäin käsittelyä varten.

**Käyttäjä.java:** Luokka toimii templaattina, jonka pohjalta kullekin käyttäjälle luodaan tili (lopullinen versio työstä, ei toteutettu nyt). Tämän luokan olio pitää sisällään käyttäjän nimen ja muut tarvittavat tiedot ja määrittää hänen statuksensa.

**Istunto.java:** Istunto jotain, joka käynnistyy jokaisen sovellukseen kirjautumisen yhteydessä ja joka suljetaan uloskirjautuessa. Sovelluksen sivuja ei voi tarkastella ilman että istunto on päällä.

**Ruokalaji.java:** Tämä malliluokka pitää sisällään ruokalaji-kohtaiset tiedot eli nimen, tyypin (alkuruoka, pääruoka, jne.) sekä raaka-aineet.

**TietokantaYhteys:** Tämä luokka pitää sisällään metodit yhteyksien luomiseen ja sulkemiseen tarvittavat metodit.

### 7.2 SERVLETIT

Servletit sovelluksessa muodostavat yhdessä JSP-tiedostojen kanssa sovelluksen rungon. Servletit kutsuvat JSP-tiedostoja, tarvittavia (mm. kysely) metodeja ja välittävät tarpeen tullen parametreja JSP-tiedostoihin, missä niitä voidaan esittää luettavassa muodossa.

**Etusivu.java:** Etusivu on se sivu, jolle käyttäjä ohjataan ensimmäiseksi kirjaututtuaan sisään. Servletti tarkastaa, että onko käyttäjä kirjautunut sisään ja jos on niin näyttää etusivun. Muuten käyttäjä ohjataan kirjautumissivulle.

**Etusivu\_jalkeen:** Etusivusta on tehdyssä sovelluksessa olemassa kaksi versiota. Ensimmäinen versio on se, minne käyttäjä ohjataan kirjautumisen jälkeen (Etusivu) ja toinen otetaan käyttöön kun käyttäjä suorittaa kyseisellä sivulla jonkin reseptin haun sivulla. Kuten Etusivu-servletti niin Etusivu\_jalkeen myös tarkistaa käyttäjän kirjautumistilanteen.

**Kirjautuminen.java:** Tämän sivun kautta käyttäjä kirjautuu sovellukseen sisään.



**MuutoksetLisäykset:** Servletti, joka huolehtii uusien ruokien lisäämisestä tietokantaan.

**MuutoksetPoisto:** Servletti, joka huolehtii olemassa olevan ruokalajin poistamisesta tietokannasta.

**MuutoksetMuokaa:** Servletti, joka huolehtii olemassa olevan reseptin tietojen muokkaamisesta.

**Rekisteroityminen:** Servletti, jonka tarkoituksena on huolehtia uuden käyttäjätunnuksen lisäämiseen liittyvistä toiminnoista.

**UlosKirjautuminen:** Servletti, joka huolehtii olemassa olevan istunnon sulkemisesta. Servletti ohjaa tämän jälkeen käyttäjän etusivulle.

### 7.3 JSP-TIEDOSTOT

JSP-tiedostot (JSP on lyhenne sanoista Java Server Page) määrittelevät sivuston rakenteen kuten siinä olevat kentät, taustan värityksen ja muut vastaavat toiminnot. Kullakin Servletillä (poislukien ServlettiIlsa) on oma JSP tiedosto (jotkin servletit saattavat jakaa saman JSP-sivun), joka määrittelee sivun ulkoasun ja palikat kuten tekstikentät, otsikot ja painikkeet. Kullakin JSP-tiedostolla on vastaava nimi kuin Servletillä helpottamaan keskinäisten yhteyksien hahmottamista.

**Etusivu\_ennen:** Kun käyttäjä kirjautuu sovellukseen ensimmäisen kerran niin näyttää tämän tiedoston määrittelemän näkymän.

**Etusivu\_jälkeen:** Käyttäjä näkee tämän sivun näkymän suoritettuaan ensimmäisen toiminnon (tässä tapauksessa haun) sivulla.

**Kirjautuminen:** Sovellus vastaa kirjautumissivun näkymän esittämisestä.

**Lisukkeet:** Lisukkeiden lisäyksistä vastaavan sivun näkymä on määritelty tässä tiedostossa.

**Muutokset:** Sivu, jonka kautta tehdään ruokareseptien muutokset, määritetään tässä tiedostossa.

**Ohjeet:** Nimensä mukaisesti kyseessä on sivu, jossa näkyy sovelluksen käyttöohjeet

**index.jsp:** Ei kuulu projektiin, on automaattisesti generoitu tiedosto.

Jos aika/jaksaminen antaa myöten niin lopuksi sovellukseen tulee vielä Kayttajat

### 7.4 TAG-TIEDOSTOT

Sovelluksessa on käytössä vain yksi tag-tiedosto nimeltä **Pohja.tag**. Tag-tiedostot ovat palasia HTML-koodia, joka toistuu kaikissa sivuissa ja joka on eriytetty omaksi tiedostokseen, jossa kyseistä informaatiota on helppo päivittää ja hallinnoida. Periaate on hyvin paljon linjassa OOP-periaatteen kanssa, jossa toiminnot kapseloidaan ja joiden palveluita voidaan käyttää tietämättä tarkalleen mitä tapahtuu millä tavoin. Tässä sovelluksessa tag-tiedostoa käytetään yläpalkin esittämiseen (ja sitä kautta sivunavigaatioon) ja virheiden näyttämiseen.

## 7.5 HTML-TIEDOSTOT

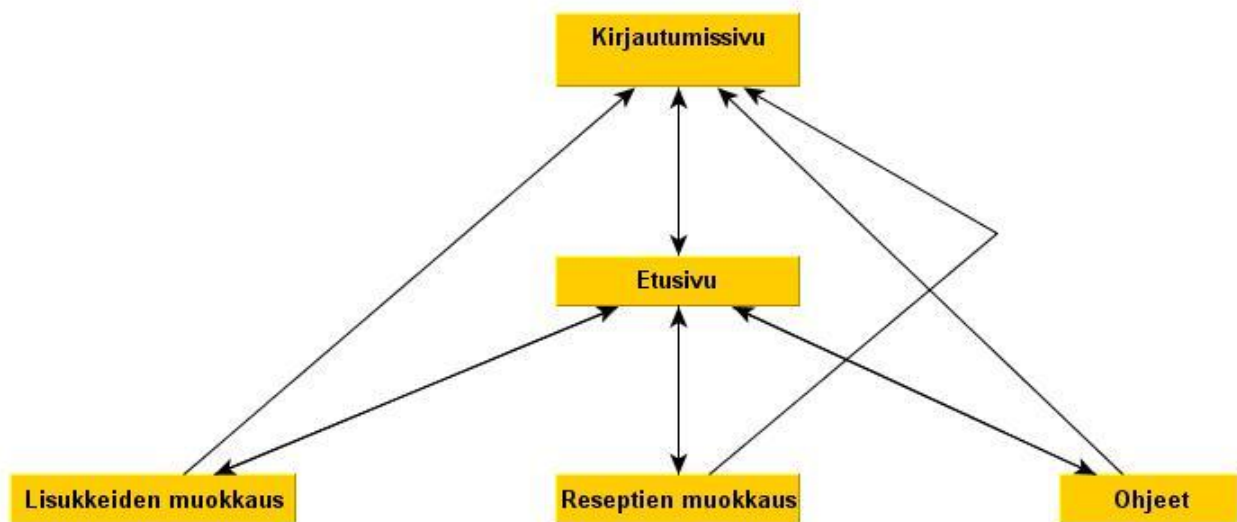
Sovelluksen HTML-tiedostot eivät ole missään aktiivisessa käytössä tässä vaiheessa. Sovelluksen kehittelyn alkuvaiheissa ne toimivat havainnollistavana runkona, jonka pohjalta rakennettiin JSP-tiedostot.

## 7.6 CSS-TIEDOSTOT

**main.css:** Tyylien hallinnointia varten luotu css-tiedosto, ei tällä hetkellä käytössä missään osassa sovellusta.

## 8 KÄYTTÖLIITTYMÄ

Sovelluksen käyttöliittymä on esitetty lyhykäisyydessään alla. Nuolen suunta kertoo mihin miltäkin sivulta pääsee.



## 9 ASENNUSTIEDOT

Sovelluksen asennustiedostona toimii ElektroninenKeittokirja.war-paketti, joka sijaitsee palvelimella. Sovelluksen avaamisesta ja lukemisesta huolehtii Tomcat-palvelin, jota on käytetty tässä sovellusprojektissa.

## 10 KÄYNNISTYS- JA KÄYTTÖOHJEET

Koska sovellus on verkkopohjainen, sen käyttäminen tapahtuu selaimen avulla kirjautumalla asianmukaisessa osoitteessa järjestelmään. Kirjautuminen sovellukseen tapahtuu osoitteessa

<http://t-teemateve.users.cs.helsinki.fi/ElektroninenKeittokirja>

Käyttäjätunnuksina voi käyttää:

Login: Aeon

Salasana: Kuusi

## **11 TESTAUS, LÖYDETYT BUGIT, PUUTTEET & JATKOKEHITYSIDEAT**

Sovelluksen testaaminen on jäänyt vajavaiseksi toteutettaessa sovelluksen toimintoja ja tästä johtuen sovelluksen bugisuudesta on paha sanoa muuta paitsi että sitä varmasti on. Yksi pidempään jatkunut huoli liittyy kuitenkin yhteysvarannon käyttöön ja sen tarkan syyn selvittäminen on osoittautunut hyvin hankalaksi. Syystä tai toisesta jotkin osat sovelluksesta eivät (ainakaan aina) sulje kaikkia yhteyksiä ulkopuoliselle palvelimelle, jossa on mm. PostgreSQL ja seurauksena on ollut ns. ”yhteyspoolin” täyttyminen mikä kaataa sovelluksen lokaalin version kunnes yhteydet on katkaistu ja sovellus käynnistetty uudelleen. Palvelimella olevassa versiossa ongelmaa ei ole havaittu enää kirjautumiseen liittyvän osuuden korjaamisen jälkeen mutta on odotettavaa, että pidemmän testausrupeaman jälkeen kyseinen ongelma saattaa nostaa päätään.

Edellä mainituista seikoista huolimatta sovellusta on pyritty testaamaan jotta mm. virheviestien toiminta on voitu todentaa. Testaukset on suoritettu täysin manuaalisesti (sovelluksen teossa ei ole käytetty mm. JUnit testausta) käymällä läpi eri käyttötapauksia kuten puuttuvia kenttiä ja muuta vastaavaa.

Mitä tulee jatkokehitysideoihin niin sovelluksen ulkoasussa olisi paljon toivomisen varaa. Sivut ovat hyvin minimalistisia ja niiden komponenttien asettelu pitäisi tehdä paremmin (esimerkkinä tästä on mm. lisukelistauksen laatikon liikkuminen etusivulla). Teksteissä ei ole käytetty tyyliä millään tavalla ja väripaletti on hyvin monotoninen ja persoonaton. Virheviestien esittäminen varsinkin on tehty aika huonosti tällä hetkellä (kissan korkuiset kirjaimet sivun ylälaudassa eivät ole hirveän nätit). Pienempi kirjasinkoko, parempi asettelu mm. ongelmakohdan kohdalle ja toinen värityys auttaisivat asiaa paljon.

Sisällön puolesta listaukset olisi mukavampi tehdä taulukoihin lukemisen helpottamiseksi (tästä voi taas käyttää esimerkkinä lisukelistaa etusivulla). Sovellus kaipaisi lisää toimintoja kuten mahdollisuuden käyttäjätietojen muokkaamiseen, salasanojen vaihtamiseen moderaattoreiden puolelta ja muuta vastaavaa.

## **12 OMAT KOKEMUKSET**

Oman kokemuksen mukaan sovellus on kohtalaisen vakaa tällä hetkellä. Se pystyy suoriutumaan perustoiminnoista kuten reseptien ja lisukkeiden lisäyksistä, uudelleen nimeämisistä sekä hauista ja se myös esittää halutut asiat luettavassa muodossa. Sovelluksen teko oli kuitenkin vaikeata monessakin mielessä vaikka aikaisempaa kokemusta vastaavanlaisen projekti läpiviemisestä oli edelliseltä periodilta. Syynä tähän oli että lähdin itse Java web-sovelluksen tekemiseen ensimmäisen periodin aikana täysin puhtaalta pöydältä ja asiat kuten servlettien ja JSP-tiedostojen yhteenpelaaminen, sessioitten hallinta ja muut vastaavat asiat olivat täysin tuntemattomia. Tämän

projekti läpivieminen on ollut hyvin opettavaista ja totta puhuen kaiken sen listaaminen mitä opin sitä tehdessä veisi aivan liian kauan aikaa. Todettakoon vain että nyt pystyn rakentamaan omia pieniä web-sovelluksia ja kykenen jatkamaan itseni sivistystä omaehtoisesti kun kurssi päättyy.