

SuperlaskuttajaWebApp dokumentaatio

Aku Silvennoinen 1.3.2015

1. Johdanto

Laskutusohjelma web-sovelluksena. Toteutetaan sovellus, jolla voidaan laskuttaa asiakkaita. Ohjelmaan tulee syöttää laskuttajan ja asiakkaiden tiedot. Kun edelliset tiedot on syötetty voidaan sovellukseen lisätä asiakaskohtaisia suoritteita. Ohjelmalla voidaan laskuttaja-, asiakas -ja suoritettietojen pohjalta tehdä laskuja. Laskut voidaan viedä pdf-muotoon lähetettäväksi sähköpostilla tai muuten. Ohjelma huolehtii mm. laskujen laskujen numeroinnista kirjanpitoa varten ja viitteiden automaattisesta luomisesta asiakasnumeron ja asiakaskohtaisen laskunumeron perusteella. Lisäksi sovellus säilyttää vanhat laskut muuttumattomina: jos muokkaat asiakkaan tietoja, tietojen muutokset eivät vaikuta jo muodostettujen laskujen tietoihin.

Sovelluksen tavoitteena on helpottaa pienen mittakaavan laskutusta, jossa laskutukset koostuvat suurimmilta osin immateriaalisista suoritteista.

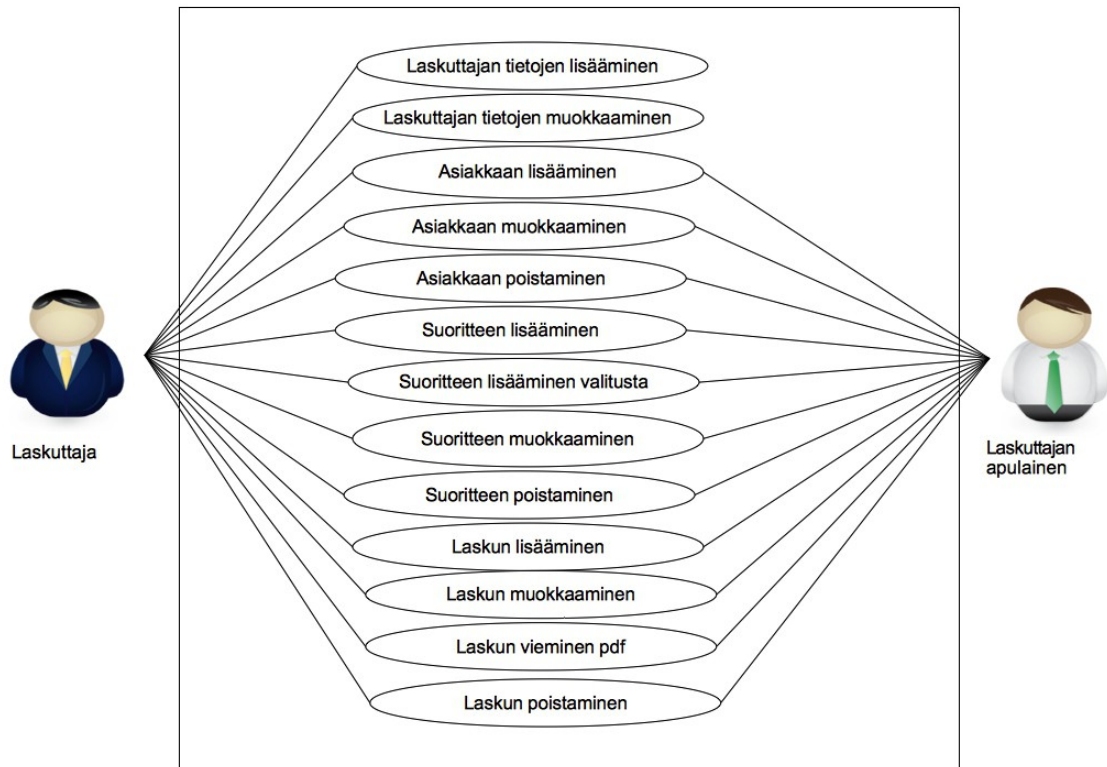
Työ toteutetaan Tomcat-ympäristössä.

Web-sovelluksen alustajärjestelmän tulee tukea Javaa.

Sovellus edellyttää Postgresql-tietokannan käyttöä.

2. Yleiskuva järjestelmästä

Käyttötapauskaavio



Kuva 1: Käyttötapauskaavio

Käyttäjärühmät

Laskuttaja

Laskuttajalla on oikeus tehdä mitä vain applikaatiossa.

Laskuttajan apulainen

Laskuttajan apulaisella on kaikki muut oikeudet, paitsi laskuttajan tietojen lisääminen tai muokkaaminen.

Käyttötapauskuvaukset

Laskuttaja tietojen lisääminen

Voidaan lisätä sovellukseen laskuttajan tiedot. Sovellukseen voidaan lisätä vain yhden laskuttajan tiedot.

Laskuttajan tietojen muokkaaminen

Voidaan muokata laskuttajan tietoja. Jos laskuja on olemassa ennen muokkaamista, niin tietojen muokkaaminen ei vaikuta olemassa olevien laskujen tietoihin.

Asiakkaan lisääminen

Voidaan lisätä asiakas.

Asiakkaan muokkaaminen

Voidaan muokata asiakasta. Jos asiakas esim. muuttaa toiselle paikkakunnalle tämän avulla on helppo päivittää osoitetiedot ajantasalle.

Asiakkaan poistaminen

Voidaan poistaa asiakkaan tiedot sovelluksesta. Onnistuu vain jos poistettavalla asiakkaalla ei ole suoritteita järjestelmässä (eli ei myöskään laskuja).

Suoritteen lisääminen

Voidaan lisätä sovellukseen suorite. Esim. 2h it-konsultointia tai 5kpl tietokoneita.

Suoritteen lisääminen valitusta

Voidaan lisätä sovellukseen suorite jo järjestelmässä olevaa suoritetta pohjana käyttäen. Tarkoitettu suoritteiden lisäämisen nopeuttamiseen.

Suoritteen muokkaaminen

Voidaan muokata suoritetta.

Suoritteen poistaminen

Voidaan poistaa suorite. Onnistuu vain jos suorite ei ole millään laskulla.

Laskun lisääminen

Voidaan lisätä sovellukseen lasku. Valitaan tilaaja, vastaanottaja, laskutettavat suoritteet. Syötetään myös muut tarvittavat tiedot, kuten maksuaika.

Laskun muokkaaminen

Tällä voidaan muokata jo olemassa olevaa laskua.

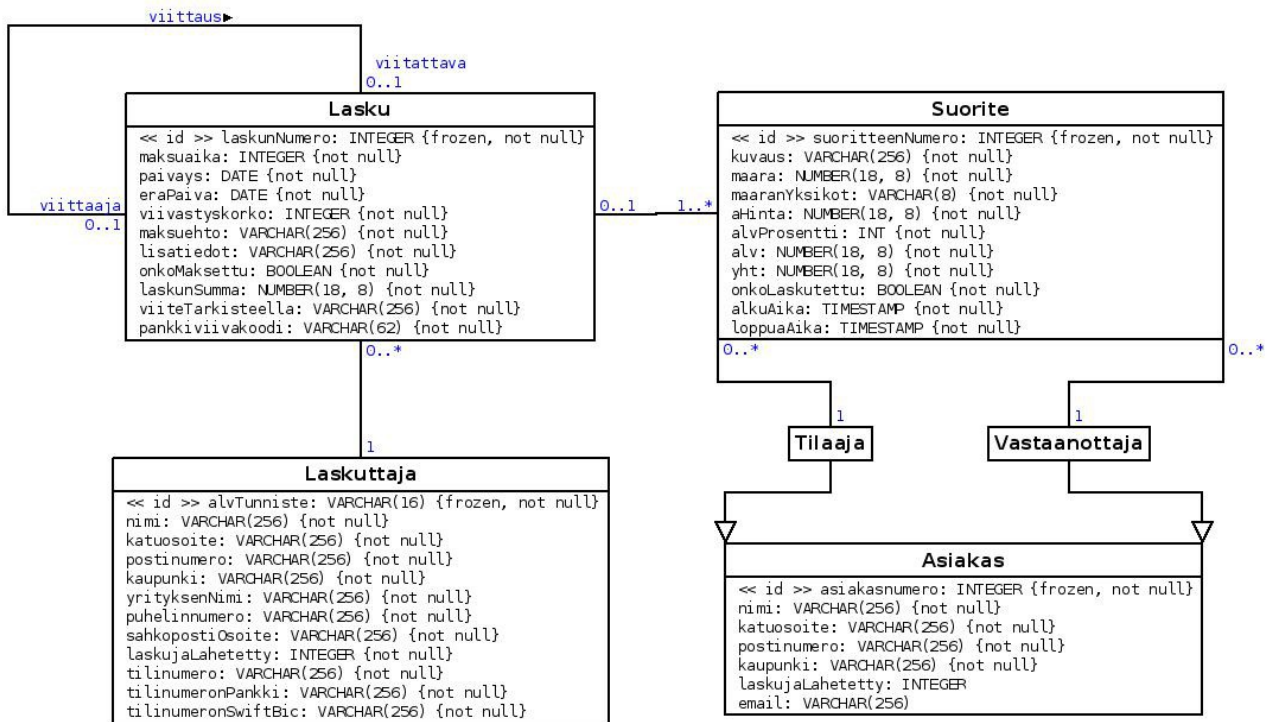
Laskun vieminen pdf

Voidaan viedä valittu lasku pdf-muotoon sähköpostilla lähettämistä varten tai vaikkapa tulostamista varten.

Laskun poistaminen

Jos lasku on jotenkin virheellinen, niin laskun voi poistaa tästä.

3. Järjestelmän tietosisältö



Kuva 2: Käsittemalli

Tietokohde: Laskuttaja

Attribuutti	Arvojoukko	Kuvailu
alvTunniste	Merkkijono, max 16 merkkiä	Yrityksen alvtunniste ks. esim. http://www.vero.fi/fi-FI/Syventavat_veroohjeet/Arvonlisaverotus/EU_maiden_arvonlisaverotunnisteet(14426)
nimi	Merkkijono, max 256 merkkiä	Laskuttajan nimi
katuosoite	Merkkijono, max 256 merkkiä	
postinumero	Merkkijono, max 256 merkkiä	
kaupunki	Merkkijono, max 256 merkkiä	
yrityksenNimi	Merkkijono, max 256 merkkiä	
puhelinnumero	Merkkijono, max 256 merkkiä	
sahkopostiOsoite	Merkkijono, max 256 merkkiä	
laskujaLahetetty	Kokonaisluku ≥ 0	Tätä attribuuttia tarvitaan mm. uusien laskujen muodostamiseen (laskunumero).
tilinumero	Merkkijono, max 256 merkkiä	
tilinumeronPankki	Merkkijono, max 256 merkkiä	
tilinumeronSwiftBic	Merkkijono, max 256 merkkiä	

Laskuttajaan voi liittyä monia laskuja. Tietokannassa on vain yksi laskuttaja.

Tietokohde: Lasku

Attribuutti	Arvojoukko	Kuvailu
laskunNumero	Kokonaisluku > 0	
viittausAiempaanLaskuun	Kokonaisluku > 0	Joskus laskulla voidaan korjata aiempaa laskua. Tällöin tulee viitata tuohon aiempaan laskuun.
maksuaika	Kokonaisluku ≥ 0	
paivays	Päiväys	
eraPaiva	Päiväys	Laskun eräpäivä \geq laskun päiväys
viivastyskorko	$0 \leq$ Kokonaisluku ≤ 100	
maksuehto	Merkkijono, max 256 merkkiä	
lisatiedot	Merkkijono, max 256 merkkiä	
onkoMaksettu	Totuusarvo	
laskunSumma	Desimaaliluku	
viiteTarkisteella	Merkkijono, max 256 merkkiä	Tähän löytyy dokumentaatio: http://www.fkl.fi/teemasivut/sepa/tekninen_dokumentatio/Dokumentit/kotimaisen_vitteen_rake_nneohje.pdf
pankkiviivakoodi	Merkkijono, max 62 merkkiä	Tähän löytyy dokumentaatio: http://www.fkl.fi/teemasivut/sepa/tekninen_dokumentatio/Dokumentit/Pankkiviivakoodi-opas.pdf

Laskuun liittyy aina yksi laskuttaja. Laskuun liittyy vähintään yksi suorite.

Tietokohde: Suorite

Attribuutti	Arvojoukko	Kuvailu
suoritteNumero	Kokonaisluku ≥ 0	
kuvaus	Merkkijono, max 256 merkkiä	
maara	Desimaaliluku	
maaranYksikot	Merkkijono, max 8 merkkiä	
aHinta	Desimaaliluku	
alvProsentti	$0 \leq$ Kokonaisluku ≤ 100	
alv	Desimaaliluku	
yht	Desimaaliluku	
onkoLaskutettu	Totuusarvo	
alkuAika	Päiväys kellonajalla	Alkuajan ja loppuajan perusteella voidaan laskea tarkkaan laskutettava aika.
loppuAika	Päiväys kellonajalla	Alkuajan ja loppuajan perusteella voidaan laskea tarkkaan laskutettava aika.

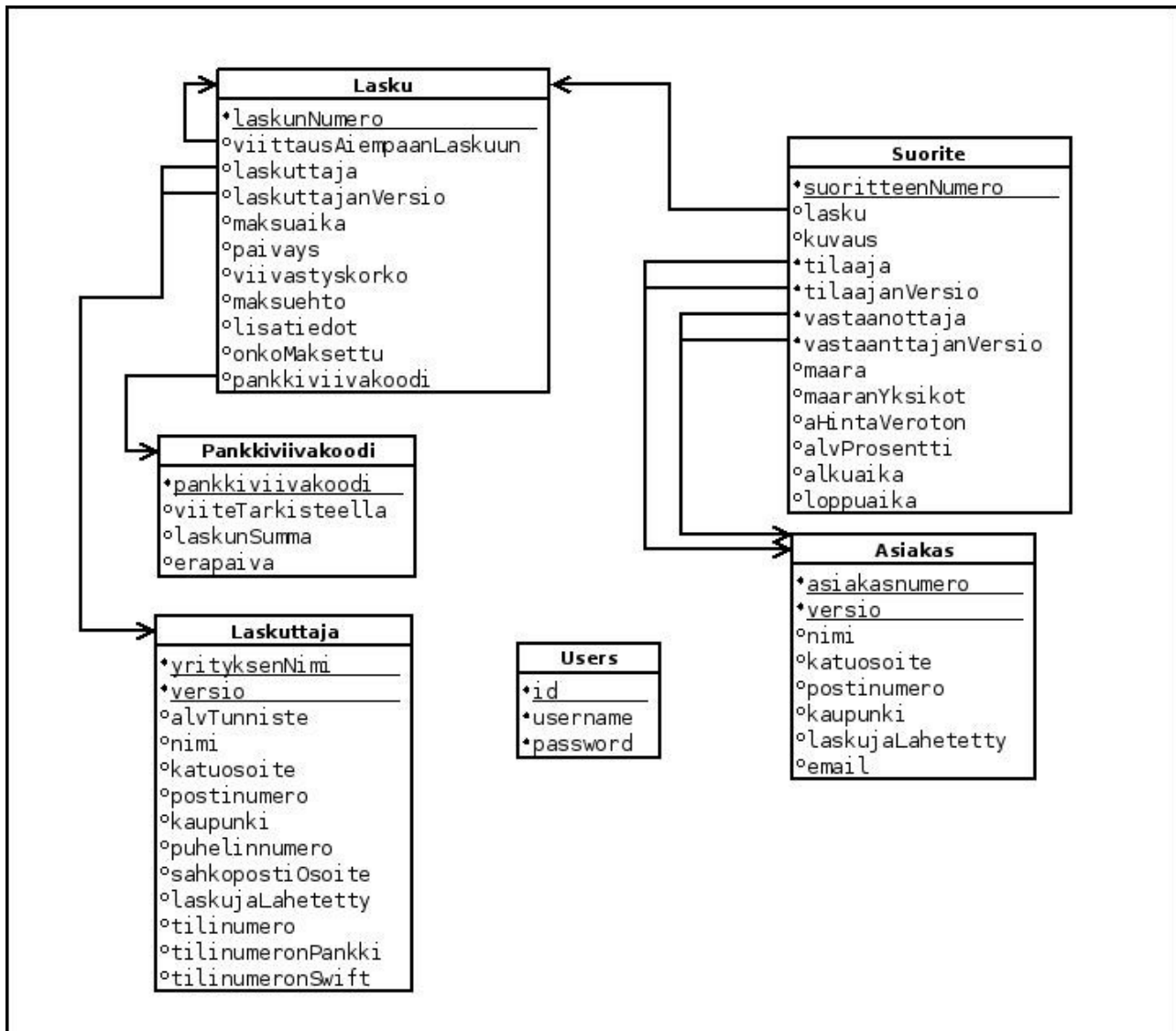
Suoritteeseen liittyy enintään yksi lasku. Välttämättä ei yhtään laskua. Suoritteeseen liittyy yksi tai kaksi asiakasta. Jos tilaaja on sama kuin vastaanottaja, niin suoritteeseen liittyy vain yksi asiakas. Jos taas tilaaja ja vastaanottaja ovat eri, niin tällöin suoritteeseen liittyy kaksi asiakasta.

Tietokohde: Asiakas

Attribuutti	Arvojoukko	Kuvailu
asiakasnumero	Kokonaisluku ≥ 100	Asiakasnumeron perusteella muodostetaan viitenumeroita, jonka vuoksi asiakasnumeron tulee olla vähintään kolmenumeroinen ja siinä ei saa olla etunollia.
nimi	Merkkijono, max 256 merkkiä	
katuosoite	Merkkijono, max 256 merkkiä	
postinumero	Merkkijono, max 256 merkkiä	
kaupunki	Merkkijono, max 256 merkkiä	
laskujaLahetetty	Kokonaisluku ≥ 0	
email	Merkkijono, max 256 merkkiä	

Asiakkaaseen voi liittyä laskuja. Ei välttämättä liity yhtään.

4. Relaatiotietokantakaavio



Kuva 3: Relaatiotietokantakaavio

5. Järjestelmän yleisrakenne

Tietokantasovellusta tehdessä on noudatettu MVC-mallia.

Kontrollerit löytyvät polusta /src/java/Superlaskuttaja/Controllers. Kontrollereiden nimeäminen noudattaa konventiota, jossa ensin ilmoitetaan kontrolleria koskeva malli, jonka jälkeen on merkkijono "Servlet". Esim. AsiakasServlet.java.

Mallit sijaitsevat polussa /src/java/Superlaskuttaja/Models. Mallien nimeäminen noudattaa konventiota, jossa mallin nimi kirjoitetaan isolla alkukirjaimella. Esim. Asiakas.java.

Mallihakemistossa on myös järjestelmän tarvitsemia apu – ja yleisluokkia:

- DBConnection.java – Tämä luokka huolehtii tietokantayhteydestä Postgresql – tietokantaan
- MerkkiJaMerkkijonoTarkistin.java – Tämä luokka tarjoaa erilaisia merkkijonojen tarkistusmetodeja
- Tilinumero – Tämä luokka tarjoaa metodit mm. tilinumeron validiuden tarkistamiseen

- UnivClass – Tämä luokka tarjoaa järjestelmän, erityisesti kontrollereiden usein tarvitsemia metodeja

Näkymät löytyvät omista hakemistoistaan hakemistosta web. Esim. malliin Asiakas liittyvät näkymät löytyvät kansioista asiakkaat. Näkymien nimennässä on noudatettu konventiota, jossa näkymät nimetään nimillä index.jsp, show.jsp, edit.jsp ja new.jsp. Laskuttaja-malliin liittyvät näkymät löytyvät hakemistosta laskuttaja, koska järjestelmässä on tarkoitus olla vain yksi laskuttaja kerrallaan. Kirjautumiseen liittyvät näkymät löytyvät hakemistosta login.

Järjestelmän sivujen template löytyy hakemistosta /web/WEB-INF/tags.

Reititysasetukset löytyvät tiedostosta /web/WEB-INF/web.xml.

Järjestelmä on toteutettu siten, että jokaista mallia vastaa vain yksi kontrolleri (servlet). Tämä on toteutettu siten, että reititysasetuksiin (web.xml) on määritelty esim. polkuun `AsiakasServletCreate` mentäessä `init-param` (initialization parameter) `univParam`, jonka arvo on `create`.

`AsiakasServlet.java` sitten lukee tuon parametrin `univParam` arvon ja valitsee oikean lohkon kontrollerista suoritettavaksi.

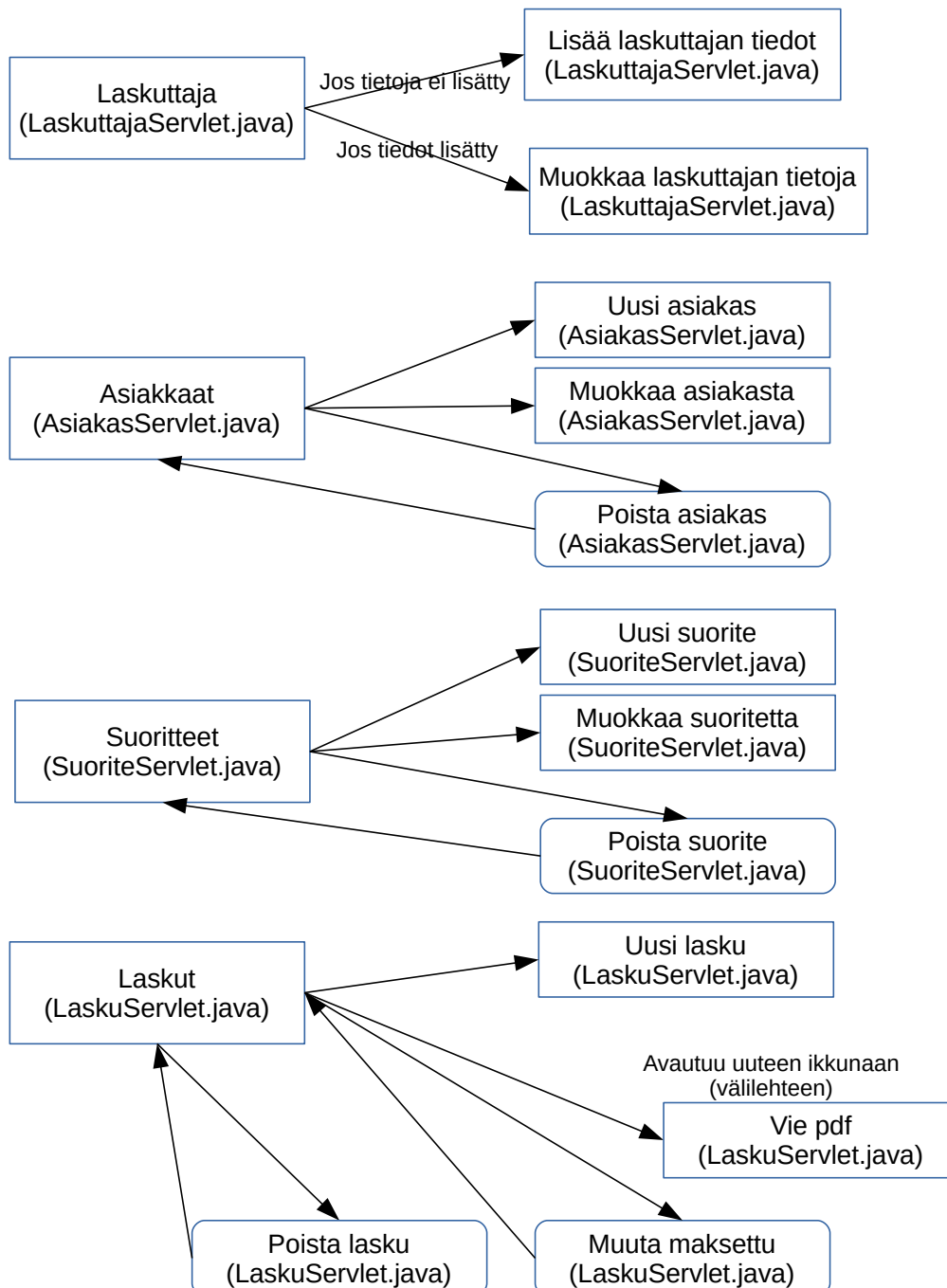
Käytetyt apukirjastot sijaitsevat hakemistossa /web/WEB-INF/lib.

Järjestelmässä on käytetty javascriptiä sivulla, jossa lisätään lasku. Tarkemmin ottaen sivulla käytetään jqueryä. Sivulle tuodaan dataa json-muodossa laskukontrollerilta.

6. Käyttöliittymä ja järjestelmän komponentit

Kirjautumista vaativat sivut

Sivustolla on navigaatiopalkki, johon sisältyy kohteet Laskuttaja, Asiakkaat, Suoritteet ja Laskut. Lisäksi kirjautumislinkit.



Kuva 4: Käyttöliittymäkaavio

7. Asennustiedot

Sovelluksen nykyinen asennus

Sovellus on nyt asennettu tomcat 6 palvelimelle users.cs.helsinki.fi osoitteessa. Sovelluksen käytössä oleva postgresql-tietokanta sijaitsee myös kyseisessä osoitteessa. Sovellus on asennettu lataamalla .war-tiedosto asentajan home-hakemiston polkuun /tomcat/webapps. Asentajan käyttöön saama tomcat 6 – palvelin on konfiguroitu niin, että se asentaa kyseiseen hakemistoon viedyt .war-tiedostot automaattisesti näytettäväksi internetiin.

Sovelluksella on käytössään osoitteessa users.cs.helsinki.fi toimiva postgresql-tietokanta, joka on automaattisesti sitä pyydettyäessä nimetty asentajan käyttäjätunnuksen mukaan. Sovellus käyttää tietokantaa ajurin org.postgresql.Driver avulla. Sovelluksen modeleissa on luokka DBConnection, joka tarjoaa sovellukselle tietokantayhteyden. Ajurin org.postgresql.Driver tarvitsemat kirjautumistiedot sekä tietokantapalvelimen osoite on konfiguroitu tiedostoon META-INF/context.xml. Sovelluksen käytössä oleva tietokanta on alustettu käyttöön create table – lauseilla, jotka löytyvät dokumentaatiosta. Tietokannan alustamisessa on käytetty postgresql:n komentorivityökalua psql. Esim. tietokantataulujen luominen on tehty komennolla `psql < create-tables.sql`.

Sovelluksen asentaminen

Oletetaan, että käytössä on tomcat 6 tai uudempi. Lisäksi oletetaan, että käytössä on PostgreSQL 8.4.22 tai uudempi, josta on käytössä tietokanta, johon on kaikki oikeudet.

Ensin tietokanta

Pystytään sovelluksen tarvitsemat taulut komennolla `psql < create-tables.sql`. Tiedoston create-tables.sql löydät dokumentoinnista. Kun taulut ovat onnistuneesti luotu, niin muokkaa tiedostoon META-INF/context.xml tietokantasi osoite, käyttäjänimesi ja salasanasasi.

Sitten itse sovellus

Suosittelavinta on kopioida sovelluksen dist-hakemistossa oleva .war-tiedosto käytössä olevan tomcat-palvelimen hakemistoon webapps. Oletuksena tomcatissa on ainakin versiossa 7 käytössä autodeploy, joten sovelluksen pitäisi tulla näkyviin samaiseen webapps-hakemistoon vähän ajan kuluttua. Sen pitäisi myös näkyä nettiin. Jos Ongelmia ilmenee, niin tarkista logit hakemistosta tomcat/logs/.

Tarvitset vielä vähintään yhden käyttäjän

Sovelluksen kaikki osiot vaativat kirjautumisen. Luo käyttäjätunnus jo aiemmin mainitulla psql-työkalulla komennolla: `INSERT INTO Users (id, username, password) VALUES (4, 'käyttäjätunnus', 'salasana');`. Huomaa, että id ei saa olla varattuna, koska se on taulun avain.

Kun edellisen vaiheet on tehty, niin sovellus on käytössäsi.

8. Käynnistys. / käyttöohje

Harjoitustyö on osoitteessa

<http://t-akusilve.users.cs.helsinki.fi/SuperlaskuttajaWebApp/>

Käyttäjätunnus: Elmeri

Salasana: iremlE

Toimintajärjestyksestä

Laskuja ei voi luoda, jos ei ole yhtään laskuttamatonta suoritetta.

Suoritteita ei voi luoda, jos ei ole yhtään asiakasta.

Asiakkaita ei voi luoda, jos laskuttajan tietoja ei ole asetettu.

9. Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

Sovellusta on testattu luomalla tietokohteita ja kokeilemalla, että meneekö kaikki kuten on suunniteltu. Osa logiikan luokista on tosin testattu aiemmin huolellisesti yksikkötesteillä. Aiemmin testattu luokka esim. Pankkiviivakoodi. Kriittisimmille luokille on olemassa testit osoitteessa <https://github.com/Augustus58/Virtuaaliviivakoodigeneraattori/tree/master/Superlaskuttaja/src/test/java/superlaskuttaja/logiikka>, mutta testejä ei ole tuotu tähän projektiin.

Eräs tunnettu bugi on, että modelin Suorite metodi setaHintaVeroton ei suostu toimimaan luokalla BigDecimal, kuten pitäisi. Tarkemmin ottaen luokan BigDecimal metodi .divide ei toimi jostain syystä halutulla tavalla. Ongelma on ratkaistu käyttämällä tietotyyppiä Double, mutta se ei ole hyvä asia, koska lukujen tarkkuudet kärsivät ja voi tulla kertaantuvia pyöristysvirheitä! Kyseinen metodi on osoitteessa

<https://github.com/Augustus58/SuperlaskuttajaWebApp/blob/59a043891aa8e98f93bb49eae1c9e7ff85098482/src/java/Superlaskuttaja/Models/Suorite.java> rivillä 501.

Jatkokehitysideoista vahvin on yhteys pankkiin, jolla automaattisesti tarkistettaisiin onko laskut maksettu ja ne merkittäisiin maksetuiksi tai annettaisiin ilmoitus eräpäiväylityksistä jne.

10. Omat kokemukset

Helppoa oli kun oli tiettyjä materiaaleja valmiina tähän projektiin käytettäväksi. Itse tietokantakyselyiden tekeminen oli sujuvaa ja helppoa, koska aiheen parissa on tullut viime aikoina työskenneltä paljon. Aluksi vaikeaa oli javascript/jquery, koska en ole paljon kys. tekniikoita ennen käyttänyt. Sain web.xml:ää muokkaamalla aikaan mystisen bugin, jonka vuoksi esim. omia jquery-kirjastoja ei saanut käyttöön. Tiedostot eivät löytäneet, koska reititykset eivät olleet niiden löytymisen kannalta suotuisia. Myös tomcat:in merkistöongelmien kanssa on ollut työskarkaa. Opin paljon javalla toteutetuista MVC:tä noudattavista web-sovelluksista, postgresql:n käytöstä, tomcat:ista, javascriptista ja jquerystä.

11. Muu dokumentaatio

Muu dokumentaatio löytyy projektin dokumentointi-hakemistosta.