Teemu Roivas, Julius Koljonen, Junjie Yin, Grigorij Semykin

Väliraportti - Varastosovellus

Metropolia ammattikorkeakoulu, Helsinki

13.10.2016

Helsinki Metropolia University of Applied Sciences

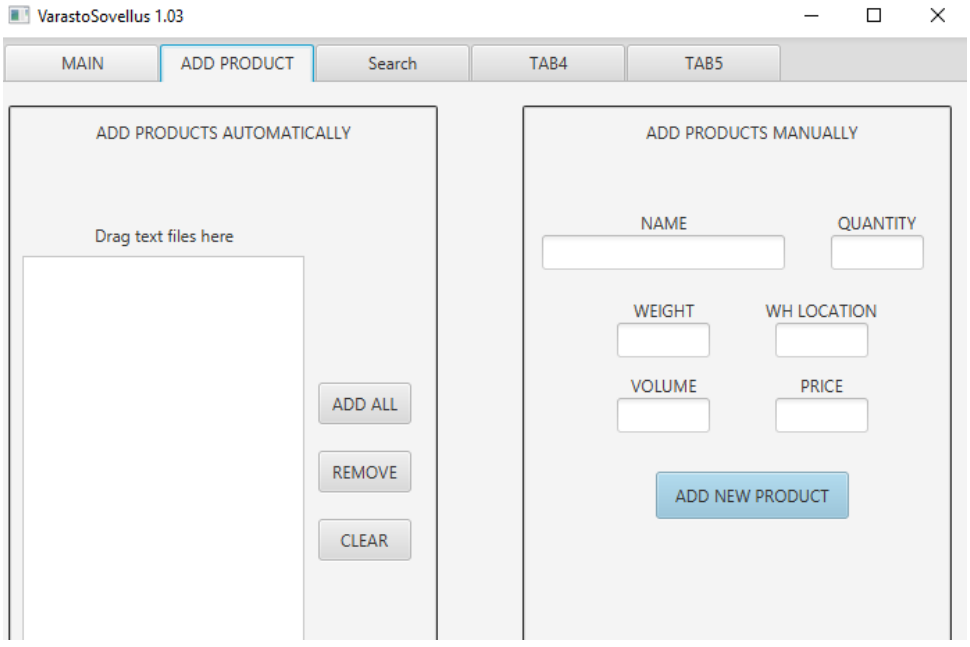
Sisältö

1 Kuvaus

Tavoitteena on edelleen tehdä varasto-ohjelma, jolla voidaan hallitta yrityksen varastoa, lähetyksiä ja tilauksia. Tavoitteena on myös, että varasto-ohjelmasta saisi graafista dataa varastotapahtumiin liittyen, esim. kuinka kauan tuote keskimäärin seisoo varastossa.

2 Toteutettu toiminnallisuus

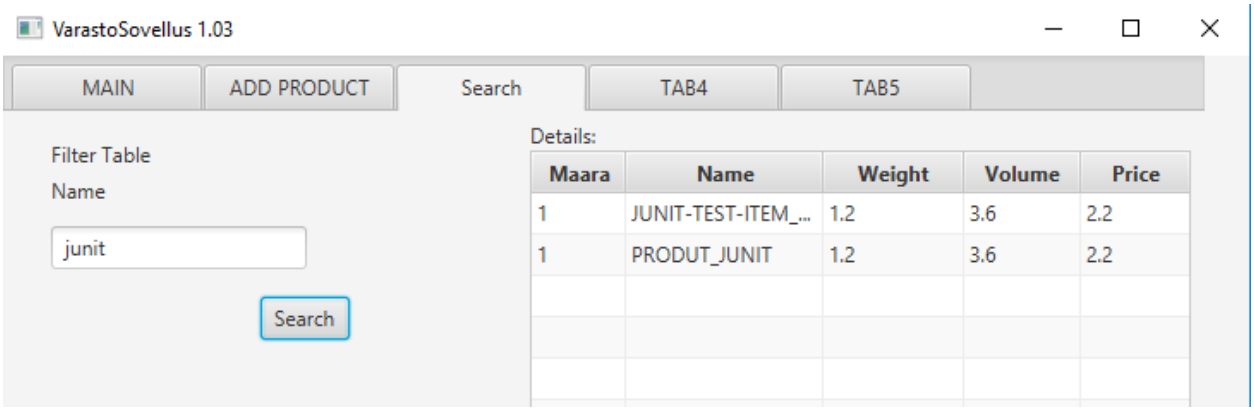
Tällä hetkellä varasto-ohjelmaan on toteutettu toimiva sisäänkirjautuminen. Käyttäjien lisääminen ja hallinnointi ei kuitenkaan vielä ole mahdollista loppukäyttäjälle.
Perustoiminnoista uuden tuotteen lisääminen (kuva 1.) varastoon on toteutettu. Tuotteen voi lisätä manuaalisesti syöttämällä tuotteen tiedot niihin kuuluviin tekstikenttiin. Useamman tuotteen lisääminen onnistuu tekstitiedoston kautta. Teksitiedosto tulee tiputtaa sille tehtyyn ”add products automatically” -kenttään. Tekstitiedostosta lisätessä tuotteen tietojen täytyy olla oikeassa järjestyksessä.



Kuva 1: Tuotteen lisääminen.

Tuotteille on toteutettu hakutoiminto, jolla voi etsiä yhden tai useamman tuotteen. Hakuehto on osa tuotteen nimestä, esim. hakuehdolla ”ruu” saadaan tulokseksi kaikki

”ruu” merkkijonon sisältävät tuotteet kuten ”ruuvi”. Hakutoiminto näyttää tuotteet taulukko muodossa (kuva 1.).



Kuva 2: Varasto-ohjelman hakutoiminto. Hakuehdoksi on annettu junit, joten ohjelma hakee tietokannasta kaikki junit merkkijonon sisältävät tuotteet tuotenimen perusteella.

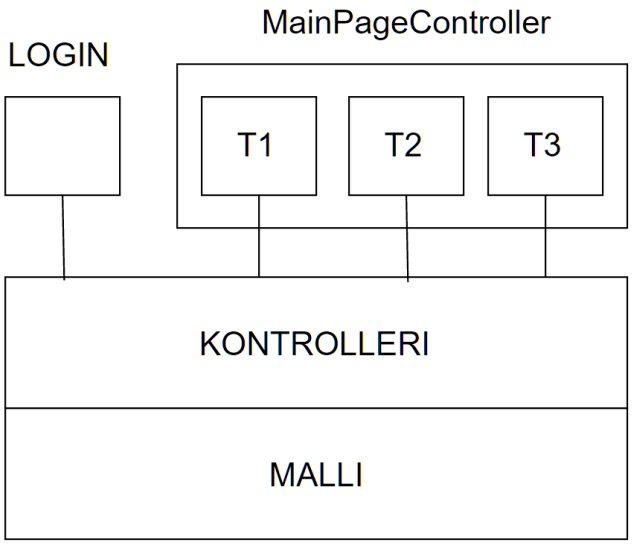
Tuotteiden muokkaaminen on toteutettu hakutoiminnon yhteydessä. Tuplaklikkaamalla solua saadaan solu aktivoitua. Solujen tietoja voi muokata haluamakseen. Päivitys tietokantaan tapahtuu vasta, kun päivitys on hyväksytty painamalla päivitä-nappia.

Projektiin on toteutettu MySQL-tietokanta, joka sijaitsee virtuaalipalvelimella. Jatkuva Integrointi toimii Jenksinsin kautta, joka kääntää ja suorittaa projektin ja kaikki siihen liittyvät junit testit. Projektissa on käytössä hajautettu versionhallinta Git. GitLab tarjoaa paikan ohjelmakehitysprojektille.

3 Arkkitehtuuri

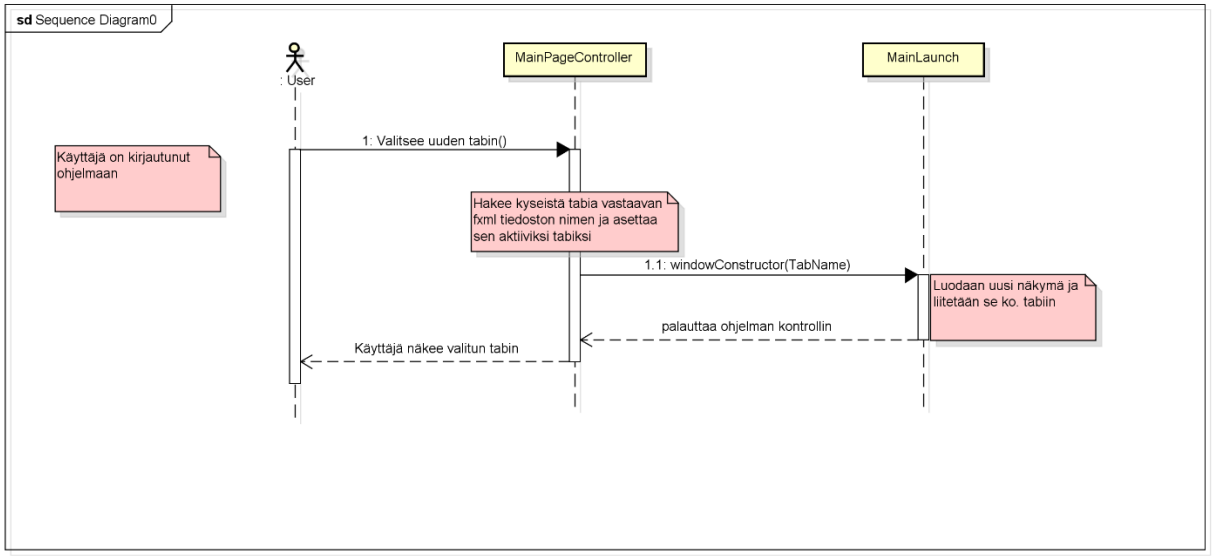
Ohjelman rakenne noudattaa MVC-rakennetta eli näytön toteutukset eivät ota kantaa tai vaikuta muuhun ohjelman osaan (kontrolleri ja malli) sekä päinvastoin. Ohjelma näyttö on toteutettu javafx:llä. Ohjelman näyttö osuus koostuu useista erillisistä javafx:än käyttämistä fxml-tiedostoista ja niitä vastaavista java näyttölogiikka olioista. Eli ts. Jokaisella javafx ikkunalla on oma java-luokka, joka vastaa ko. Javafx ikkunan logiikasta. Näytön logiikasta vastaavilla luokilla on yhteys ohjelman pääkontrolleriin, joka vastaa näytön ja mallin välisestä kommunikaatiosta (kuva 3.). Tämä yhteys luodaan aina kun

ohjelma joutuu luomaan uuden ikkunan. Jotta ko. Yhteyden luominen olisi helppoa ohjelmallisesti niin jokainen näytönlogiikan-luokka toteuttaa tietyn rajapinnan, jossa määritellään metodi pääkontrollerin liittämiselle.



Kuva 3: Karkea kuvaus ohjelman arkkitehtuurista

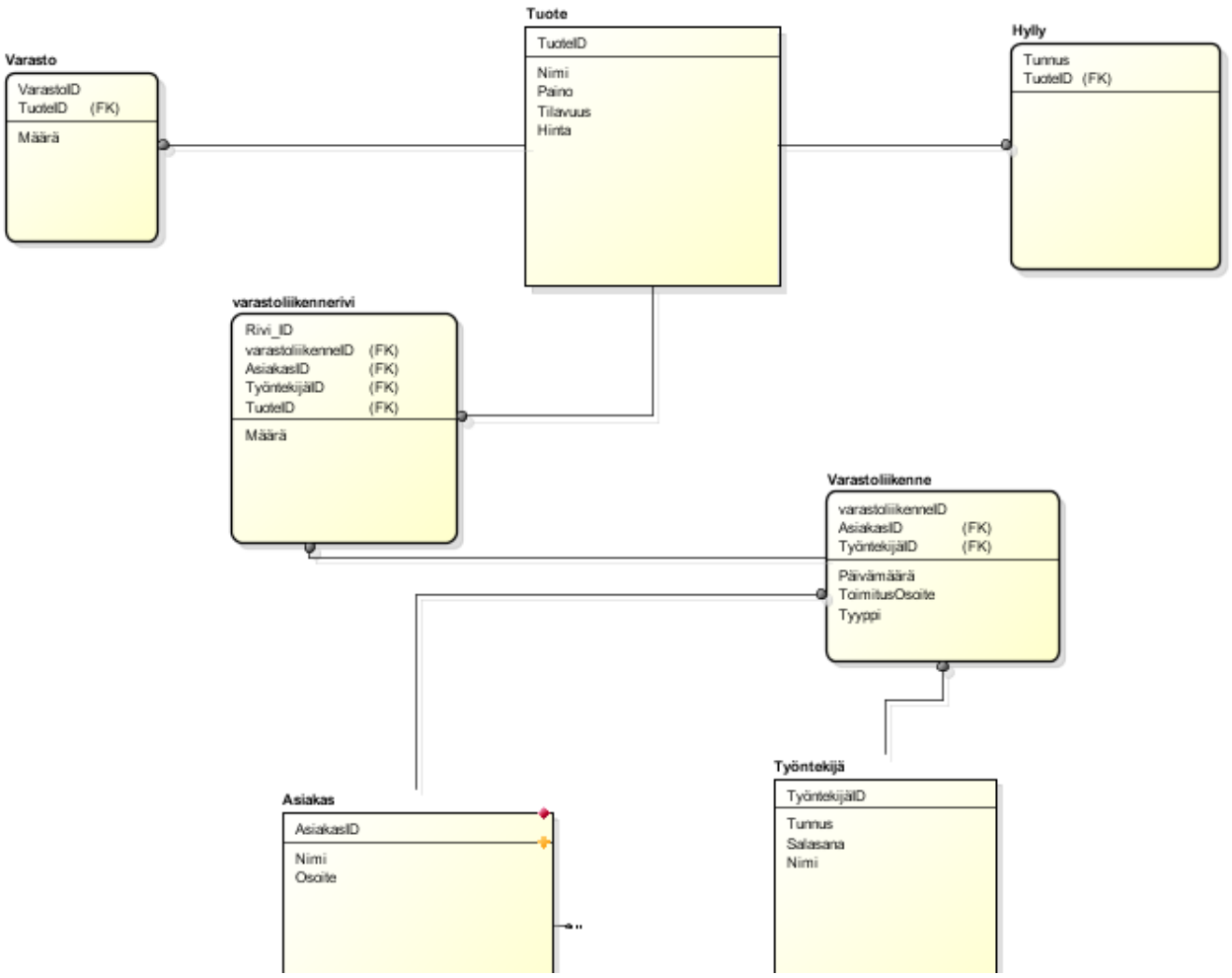
MainPageController-luokka vastaa eri näyttö luokkien vaihdosta pääikkunassa. Aina kun ohjelmassa vaihdetaan eri tabiin MainPageController kutsuu aina metodia, jolla pääikkunan päälle ladataan uusi näkymä. Lataus suoritetaan käyttämällä MainLaunchin staattista windowConstructor metodia, joka lataa uuden fxml-tiedoston uudeksi näkymäksi (kuva 4.). ko. Metodi ottaa vastaan annetun fxml-tiedoston nimen merkkijono(String) muodossa.(---miten loisin uuden näyttö kontrollin---)(Ohjelman pääsilmukka pääasiassa pyörii MainLaunchin ja MainPageControllerin välillä?)



Kuva 4: Sekvenssikaavio käyttäjän valitessa toinen välilehti

Ohjelman pääkontrolleri vastaa näytön ja mallin välisestä kommunikoinnista sekä pitää yllä kirjautuneen käyttäjän tietoja. Pääkontrollerin avulla voidaan pitää yllä istuntoa käyttäjän ollessa kirjautunut. Ohjelmassa pääkontrollerin ja mallin välillä vallitsee kompositio ja molemmat luokat luodaan ohjelman ajon aikana vain kerran.

Ohjelman malli vastaa tietokanta (Kuva 5.) yhteydestä ja tietokanta operaatioista. Malli ottaa vastaan kontrollerilta saadut argumentit ja liittää ne haluttuhin parametrisoituihin sql-kyselyihin ja suorittaa ne. Malli palauttaa metodista riippuen joko boolean arvoja, käyttäjän tietoja tai tuote-olioita.



Kuva 5: Tämän hetkinen tietokanta.