

Yhteenvetoraportti

Kuvaus

Ohjelman käyttötarkoitus on hallita pienen yrityksen tuotetietoa ja kirjata ylös yrityksen myyntihistoria. Ohjelmalla on kolme eri tason käyttäjäryhmää, viewer voi selata tuotekuvauksia ja saldoja, user voi kirjata uusia tilauksia ja tehdä korjauksia varastosaldoon, admin voi disabloida tuotteita ja luoda uusia käyttäjiä sekä tuotteita järjestelmään. Alkuperäiseen käyttöliittymä suunnitelmaan on tullut muutamia muutoksia, info-välilehden korvaa tilaushistoria-välilehti johon jää rekisteri kaikista tehdyistä tilauksista. Erillistä saavutus-välilehteä ei tule vaan tuotteiden saldoa voidaan kasvattaa tarvittaessa tuotehallinnan kautta.

Toteutuneet ominaisuudet

Sisäänkirjautuminen – Ohjelmaan pääsee kirjautumaan Admin käyttäjällä luoduilla käyttäjänimi ja salasana yhdistelmällä.

Käyttäjähallinta – Ohjelmassa voidaan luoda kolme eri tasoista käyttäjää. Admin:in, user:in tai viewer:in. Käyttäjien oikeuksia ja tietoja voidaan muokata ohjelman kautta tai tarvittaessa poistaa käyttäjä kokonaan.

Roolikohtainen näkymä – Käyttäjillä näkyy kirjautumisen jälkeen vain välilehdet joihin kyseisellä roolilla on oikeuksia.

Lokalisointi - Kirjautumisvalikon yhteyteen on lisätty kielenvaihto. Valittavana kielinä on Englanti ja Suomi. Kielen vaihtaminen muuttaa käyttöliittymän tekstielementit valitulle kielelle.

Tuotehallinta - Käyttäjät joilla on sopiva rooli, pystyvät lisäämään, poistamaan ja muokkaamaan tuotteiden tietoja.

Ei toteutuneet ominaisuudet

Tuotehistoria - Tuotehistoriaan olisi jäänyt rekisteritiedot kaikista tehdyistä tilauksista.

Graafit - Graafien avulla olisi voinut visualisoida varaston menekkiä.

Arkkitehtuuri

Arkkitehtuurin toteutuksessa pyrittiin käyttämään olio-ohjelmoinnin periaatteita käyttämällä periytymistä ja abstrakteja yläluokkia, koodin duplikaation estämiseksi. Eri tietokanta-luokkien kontrollerit toteutettiin perimällä geneerisen yläluokan ja antamalla sille tyyppi-muuttujan, joten yhteisiä ominaisuuksia ei tarvinnut toteuttaa erikseen kaikille luokille.

Koodista pyrittiin tekemään modulaarista jakamalla käyttöliittymän eri komponentit erillisiin FXML-tiedostoihin ja controller-luokkiin. Tämän ansiosta käyttöliittymää on helppo muokata jatkossa, vaihtamalla vain yksittäistä komponenttia.

Ohjelmassa pyrittiin käyttämään erilaisia suunnittelumalleja (Singleton, factory, dependency injection) koodin selkeyttämiseksi.