

Rasmus Julin, Sami Sikkilä, Hannu Korhonen, Vadim Zubchenko

# Ohjelmistotuotantoprojekti 2

Loppuraportti

Metropolia Ammattikorkeakoulu

Ohjelmistotuotanto / Tieto- ja viestintätekniikka

Ohjelmistotuotantoprojekti 2

9.5.2019

**Sisällys**

1	Johdanto	1
1.1	Lähtötiedot	1
1.2	Projektin vaatimukset	1
2	Teknologiat ja työskentelytavat	3
2.1	Ohjelmia- ja ohjelmistoja	3
3	Arkkitehtuuri	4
3.1	MVC	4
3.2	Testaus	5
4	Ohjelma	6
4.1	Toiminta	6
4.2	Vaatimusten toteutus	9
5	Yhteenveto	10

## 1 Johdanto

Projektin tavoitteena oli luoda varastohallintatyökalu pienelle kivijalka-tietokonekaupalle. Sovellus tulee yrityksen sisäiseen käyttöön ja suuntautuu myynnin ja varastonhallinnan tueksi, joten sen on tarkoitus olla graafisesti yksinkertainen ja mahdollisimman suoraviivainen.

### 1.1 Lähtötiedot

Sidosryhmä koostuu myyntihenkilöistä, varastomiehistä, esimiehestä ja ylläpitäjästä. Jokaiselle käyttäjäryhmälle luodaan heille suunnatut käyttöliittymät, eivätkä he pääse muiden käyttäjien näkymään. Tästä poikkeuksena ylläpitäjä, jolla on käyttöoikeus koko sovellukseen. Sovellus koostuu myyntisivusta, varastosivusta, tilaussivusta ja taloussivusta. Myyntisivulla myyntihenkilö voi luoda tilauksen saatavilla olevista osista ja tietokonepaketeista ja täyttää tilaukseen asiakastiedot. Varastosivulla varastohenkilöstö voi hallita varastoa ja lisätä tai poistaa sieltä tuotteita. Tilaussivulla kaikki työntekijät näkevät parhaillaan olevat tilaukset ja taloussivulla esimies voi tarkkailla yrityksen taloustilannetta.

### 1.2 Projektin vaatimukset

Ohjelmistotuotantoprojektiin määriteltiin alkuvaatimukset projektin ohjaajien toimesta. Alla listattuna ohjelmistotuotantoprojektin 1 ja 2 vaatimukset.

#### Ohjelmistotuotantoprojekti 1

- Projektin aloitus ja sovelluksen rungon luonti
- Hibernate toimimaan
- Metropolian etäpalvelimelle yhteys
- Maven projekti
- Jenkins ja Agilefant toimimaan

#### Ohjelmistotuotantoprojekti 2

- Uusi asiakas: компьютер компания. (Venäläinen tietokonekauppa)
- Uusia toiminnallisia vaatimuksia:

- Otetaan huomioon uudelleenkäytettävyyden näkökulmia toteutuksessa:
  - erilaisten komponenttien koko ja paino, sijainti varastossa
  - hakumahdollisuudet
  - varaston rahallisen arvon hallinta
  - laitekokoonpanojen ehdottaminen/räätälöinti asiakkaille
- Käyttöliittymän lokalisointi
- Käyttöliittymän muokkaus käytettävämmäksi
  - mitä: konkretisoikaa itse tarkemmiksi vaatimuksiksi
- Ei-toiminnallisia vaatimuksia
  - Otetaan huomioon uudelleenkäytettävyyden näkökulmia toteutuksessa:
    - ohjelmoidaan rajapintoja vasten
    - hyödynnetään koostamista periyttämisen sijaan
    - hyödynnetään luontimalleja olioita luotaessa
    - hyödynnetään rakenne- ja käyttäytymismalleja tarkoituksenmukaisesti
  - Refaktoroidaan koodia:
    - poistetaan mahdolliset ”bad smells in code”

## 2 Teknologiat ja työskentelytavat

Projekti suoritettiin käyttämällä Scrum-työskentelytapaa ja tätä tehostavaa Agilefant-sovellusta. Agilefant-sovellukseen määriteltiin käyttäjätarinat ja käyttötilanteet, joiden avulla sovelluksen etenemistä ja työryhmän käyttämiä työtunteja voitiin tarkkailla ja hallita. Agilefantin käyttäjätarinoista saimme valmiiksi 28/30. Kuvassa 1 esitetään Agilefantin käyttäjätarinoita.

Kuva 1. Agilefantin käyttäjätarinat.

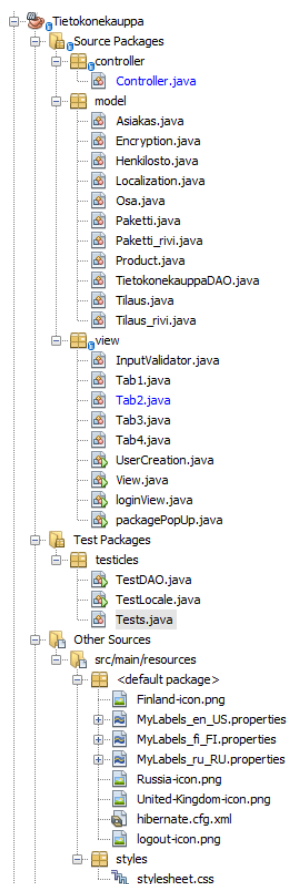
## 2.1 Ohjelmia- ja ohjelmistoja

Sovellus luotiin Apache NetBeans-ohjelmalla käyttäen Maven- työkalua. Ohjelmointi ja sovelluksen luominen suoritettiin käyttäen seuraavia ohjelmointikieliä, relaatiotietokantajärjestelmiä ja sovelluskehyksiä: Java, Hibernate, SQL, MariaDB, CSS ja XML. Aputyökaluina käytettiin Jenkins- ja SonarQube palveluita ja versionhallinnassa käytettiin Githubia. Tietokanta on Metropolian Educloud SSH-palvelimella, jota käytettiin Putty-ohjelman avulla.

### 3 Arkkitehtuuri

#### 3.1 MVC

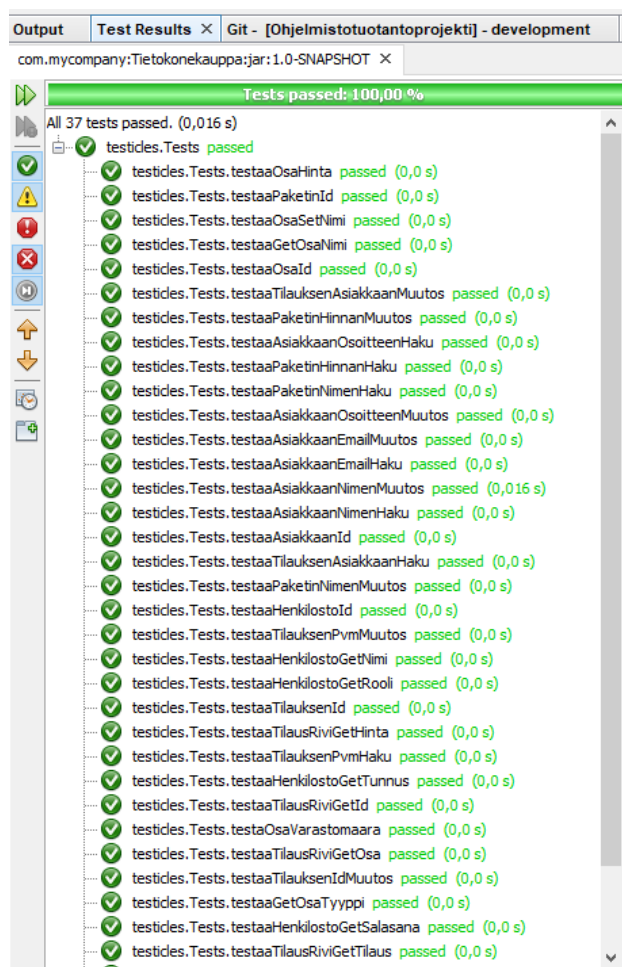
Varastonhallintasovellus on arkkitehtuuriltaan MVC-mallin mukainen. Käyttöliittymän näkymä koostuu yhdeksästä eri tiedostosta: View.java, LoginView.java, Tab1.java, Tab2.java, Tab3.java, Tab4.java, UserCreation.java, InputValidatori.java ja PackagePopUp.java. Malli-osio koostuu tiedostoista Asiakas.java, Henkilosto.java, Osa.java, Paketti.java, Paketti\_rivi.java, Product.java, TietokonekauppaDAO.java, Tilaus.java ja Tilaus\_rivi.java, Encryption.java ja Localization.java. Kontrolleri-osiossa on vain controller.java-tiedosto. Tyylit, lokalisaatiot ja hibernate määritellään Other Sources-kansiossa.



Kuva 2. Sovelluksen kansiorakenne.

### 3.2 Testaus

Sovelluksen testaus suoritettiin käyttämällä JUnit 5 ja TestFX testiympäristöjä. Testejä muodostui yhteensä 52, mutta osa jouduttiin poistamaan Jenkins ja SonarQube ongelmien takia. TestFX testit toimivat ohjelmassa erittäin mallikkaasti, mutta nämä kaikki jouduttiin karsimaan edellä mainittujen syiden takia. Kuvassa 3 esitetään sovelluksen testien tuloksia. Testeissä testataan sovelluksen tärkeimpiä toimintoja, joiden avulla sovelluksesta löydettiin haavoittuvaisuuksia ja virheitä.



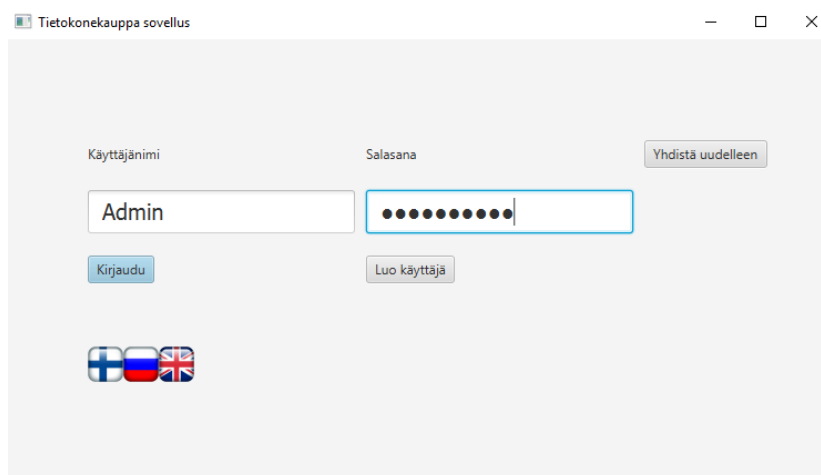
Kuva 3. Sovelluksen testien tuloksia.

## 4 Ohjelma

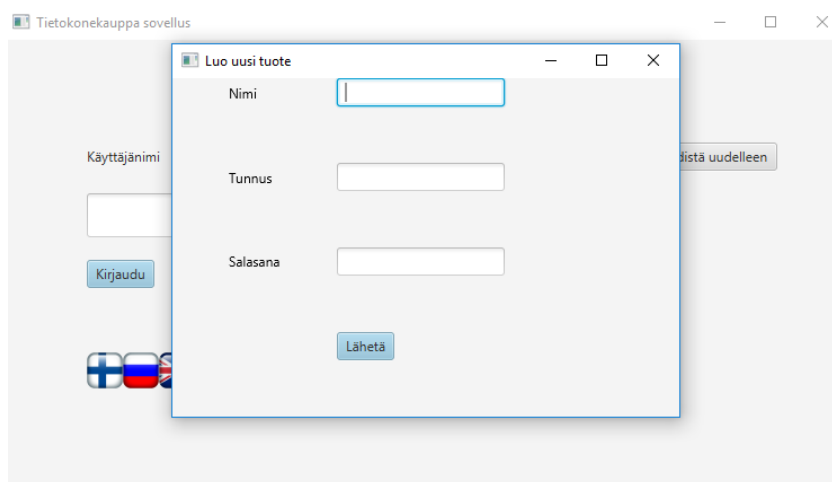
### 4.1 Toiminta

Sovellus käynnistyy LoginView.java tiedostosta, jossa käyttäjä kirjautuu tunnuksella ja salasanalla sovellukseen. Käyttäjä voi myös luoda uudet käyttäjätunnukset, jolloin salasana salataan tietokantaan käyttäen Base64- kryptausta. Kirjautumisen yhteydessä käyttäjä voi myös valita sovelluksen kielen, mutta sen voi vaihtaa myös myöhemmin sovellusta käyttäessä.

Kirjautumisen jälkeen aukeaa käyttäjälle suunnattu sivu, eikä hän pääse muiden käyttäjien sivuille. Tästä poikkeuksena sovelluksen ylläpitäjä, jolla on oikeudet kaikille sivuille. Kuvassa 4 näkyy sovelluksen kirjautumisnäkymä.



Kuva 4. Sovelluksen kirjautumisnäkymä.



Kuva 5. Uuden käyttäjän luontinäkymä.



Sovelluksen myyntisivulla eli Tab1.java tiedostossa käyttäjä voi tarkastella varastossa olevia tuotteita, jotka haetaan taulukkoon tietokannasta. Tuotteita ovat paketit ja osat, joista voi luoda tilauksen, jolloin kyseisen tuotteen kokonaismäärä varastossa pienenee tilaukseen varatulla määrällä. Käyttäjä voi vaihtaa taulukon näkymää painamalla radiopainikkeita, jolloin taulukossa näytetään osat tai paketit valinnan mukaan. Taulukon syötettä voi myös suodattaa hakukentällä, jolloin taulukossa näytetään haun mukaiset tuotteet. Haku on toteutettu FilterList- oliolla, jolloin haku on dynaaminen ja nopea. Varastotaulukosta voi valita haluamansa tuotteet ja näiden määrät, joista voi tehdä tilauksen. Tilauksen luonti vaatii, että tilaukseen on valittu tuotteita ja asiakaskentät on täytetty. Sovelluksessa on virheilmoitukset, jotka kertovat käyttäjän virhekäytöksestä. Taulukon ei voi lisätä tyhjiä rivejä eikä ilman valintaa voi riviä myöskään poistaa. Näistä tulee myös käyttäjälle virheilmoitus. Myyntisivu esitetty kuvassa 6.

The screenshot displays a web application interface for purchasing. At the top, there's a header with 'Tervetuloa Admin' and a date 'Käytössä 9.5.2019 12:01:03'. Below the header, there are tabs for 'Myynti', 'Varasto', 'Tilaukset', and 'Tilauksien tiedot'. The 'Myynti' tab is active. On the left, there's a search bar labeled 'Tuote' and a list of products with columns 'Tuote', 'Nimi', 'Määrä', and 'Hinta'. The products listed are: Intel i3-3300 (10, 90.0), Intel i3-5300K (8, 100.0), Intel i5-5600K (9, 150.0), Intel i5-5300 (9, 120.0), Intel i7-5700K (7, 200.0), Asus K-150 (10, 60.99), Asus K-170 (9, 90.99), and Asus P-150 (9, 70.99). On the right, there's a cart table with columns 'Tuote', 'Määrä', and 'Hinta'. It contains one item: Intel i5-5600K (1, 150.0). Below the cart, there's a 'HINTA YHTEENSÄ' field showing 150.0. At the bottom, there are input fields for 'Nimi', 'Postiosoite', and 'email', and a 'Luo tilaus' button.

Kuva 6. Sovelluksen myyntisivu, jossa on yksi tuote valittu tilaukseen.

Varastosivulla eli tiedostoissa Tab2.java ja PackagePopUp.java käyttäjä voi valita haluamansa tuotteet vasemmalta, jolloin taulukkoon tulee tietokannasta saatavilla olevat tuotteet ja näiden tiedot, kuten kuvitteellinen varastopaikka. Sivulla voi myös luoda uusia tuotteita, eli osia ja paketteja. Paketti luodaan kasaamalla osista paketti, jolloin ohjelma laskee osien hinnan ja lisää tähän prosentuaalisen katteen. Paketin luonti ja varastosivu esitetty kuvassa 7.

Varastosivu Admin

Kirjautumisaika 9.5.2019 12:01:03

Myynti Varasto Tilaukset Taloustiedot

Hyöly	Tuote	Hinta(kpl)	Määrä	HUOM
A3	NVIDIA GTX 660	45.99	10	Näytönohjain
A3	NVIDIA GTX 760	65.99	10	Näytönohjain
A3	NVIDIA GTX 960	85.99	9	Näytönohjain
A3	NVIDIA GTX 1080	285.99	10	Näytönohjain
A3	NVIDIA GTX 1080Ti	345.99	10	Näytönohjain
A3	AMD R			Näytönohjain
A3	AMD R			Näytönohjain
A3	AMD R			Näytönohjain
A3	AMD R			Näytönohjain
A3	AMD R			Näytönohjain

Luo uusi paketti

Paketti: Testipaketti

Paketti nimi: 10

Lisää osa

Osta: Processor - Osta: Intel i3-3300K - 100.0

Osta: Emolevy - Osta: MSI B75 - 45.99

Osta: RAM - Osta: MSI 12Gb - 129.99

Osta: Power Supply - Osta: MSI 1200W - 118.95

Osta hinta: 418.95

Paketin hinta: 427.33

Tallenna

Luo tuote Luo paketti

Kuva 7. Varastosivu ja paketin luonti.

Tilaukset sivulla tiedostoissa Tab3.java listataan taulukkoon kaikki tehdyt tilaukset. Tilauksista näkee tilauksen ID:n, asiakastiedot, tilauspäivämäärän ja tilauksen yhteishinnan. Valitsemalla taulukosta yhden tilauksen, tulee alempaan taulukkoon näkyviin tilauksen tilausrivit eli tilatut tuotteet. Valitun tilauksen voi myös poistaa halutessaan. Tilaukset välilehti esitetty kuvassa 8.

Tilaukset Admin

Kirjautumisaika 9.5.2019 13:28:59

Myynti Varasto Tilaukset Taloustiedot

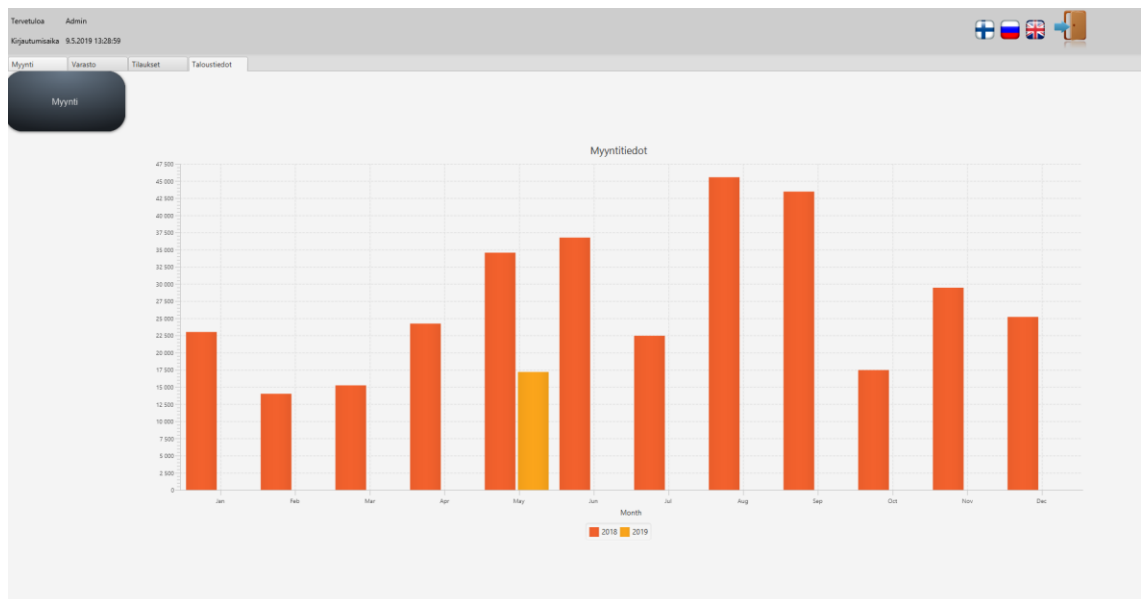
ID	Asiakas	Tilauspvm	Hinta
92	Jarmonpalvelut Oy / Jarmo Pekkala, jarmo@jarmonpalvelut.fi, Kokinkuja 12A, 00210 Helsinki	2019-05-06 11:39:37.0	2084.87
96	Sami Sikkilä, sami.sikkila@kontula.fi, Kontulankatu 1, 00120 Helsinki	2019-05-06 11:41:32.0	838.24
99	Pekka Koskinen Oy / Pekka Koskinen, pekka.koskinen@pekkakoskinen.fi, Naantalinkatu 1, 01200 Naantali	2019-05-06 11:42:58.0	6327.56
106	Mikko Karjalainen, mikko.karjalainen@gmail.com, Puistokatu 12, 02100 Espoo	2019-05-06 11:48:28.0	1654.9
118	Kari Kekkonen Ky, kari.kekkonen@hotmail.com, Linturinne 2J, 01360 Vantaa	2019-05-06 11:53:38.0	1238.24
122	Sami Samppanen, sami@sami.fi, Saminkuja 3A, 02020 Helsinki	2019-05-06 14:27:49.0	2020.58
128	Testcustomer, jarmo@jarmo.fi, jkjakfadj kjkj	2019-05-06 15:38:13.0	1618.35
133	Testikauppa, testi@osoite.fi, Testiosoite	2019-05-08 12:46:04.0	1398.35

Tuote	Paketti	Määrä	Hinta
Heavy work		3	1884.87
Intel i5-5600K		1	150.0
Asus P-150		1	70.99
NVIDIA GTX 960		1	85.99
Fractal 750W		3	121.99

Poista tilaus

Kuva 8. Tilaukset.

Taloussivulla tiedostossa Tab4.java rakennetaan graafi myydyistä tilauksista. Graafi päivitetty automaattisesti jokaisen tilauksen luonnin yhteydessä. Tämä esitetty kuvassa 9.



Kuva 9. Taloussivu.

Sovelluksen NavBar luodaan tiedostossa View.java, jolloin se näkyy jatkuvasti muiden välilehtien päällä kuten halusimme. NavBar kertoo kirjautuneen käyttäjän, kirjautumisaika ja siitä voi vaihtaa sovelluksen kieltä tai kirjautua ulos. Uloskirjautuminen ohjaa käyttäjän takaisin kirjautumissivulle, jossa käyttäjä voi kirjautua takaisin tai sulkea sovelluksen.

## 4.2 Vaatimusten toteutus

Ohjelmistotuotantoprojekti 1 kaikki vaatimukset täyttyivät projektissa. Ohjelmistotuotantoprojekti 2 vaatimuksista kaikkea ei aivan saatu suoritettua. Tämä koskee lähinnä ei-toiminnallisia vaatimuksia, koska osa näistä olisi vaatinut koko sovelluksen uudelleen rakentamista. Myös sovelluksen ulkoasu jäi vajaaksi tavoitteestamme ajanpuutteen takia.

## 5 Yhteenveto

Sovellus valmistui ajallaan toimintakuntoon ja pääsimme suurilta osin tavoitteisiin. Sovellus on kokonaisvaltainen varastohallinnan työkalu, joka täyttää ohjelmistotuotantoprojekti 1 aikana asetetut ryhmän sisäiset tavoitteet ja ylittää nämä roimasti toiminnallisuudellaan. Ei toiminnalliset vaatimukset olisivat täyttyneet, jos Suunnittelumallit kurssi olisi pidetty Ohjelmistotuotantoprojekti 1 kanssa samaan aikaan. Päällekkäisyys ohjelmistotuotantoprojekti 2 kanssa tuotti hankaluuksia projektille, sillä sovellusta oli jo rakennettu ja suunniteltu liian pitkälle ennen kuin vaatimukset annettiin. Sovellusta silti onnistuttiin refaktoroimaan ja muutama suunnittelumalli saatiin sisällytettyä sovellukseen, kuten singleton, state ja observer. Lisääjalla sovelluksen ulkoasua olisi voinut hioa käyttäen hyödyksi suunnitteluohjeistoja, jotka opimme Käyttäjäkeskeinen suunnittelu -kurssilla.