



Olli Kolkki, Juuso Lahtinen, Lera Holopainen, Severi Reivinen

Varastonhallinta -ohjelmisto

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknologian tutkinto-ohjelma

Toteutusdokumentti

10.10.2021

Sisällys

1	Johdanto	1
2	Tuotteen vaatimukset	1
3	Käyttäjäroolit ja käyttötapaukset	2
4	Käsitteet, määritelmät ja ohjelmiston tietomalli	3
4.1	Tekniset käsitteet	3
4.2	Kehitysympäristö	4
4.3	Sovelluksen tietomalli	4
5	Ohjelmiston rakenne	7
6	Ohjelmiston toiminta	10
7	Kehitysprosessi ja kehitysvaiheen tekniikat	13
7.1	Ensitoimet tilauksien hallintaan	13
7.2	Verkkokaupan varasto ja tuotteet	14
7.3	Käyttöliittymäsuunnittelu ja toteutus	14
8	Käyttöohje	15
9	Yhteenveto	15

1 Johdanto

Toteutusdokumentin tarkoitus on kertoa ohjelman toiminnasta ja toteutuksesta, sekä antaa ohjeet siihen, miten ohjelmaa käytetään. Aluksi dokumentissa esitellään käyttäjäroolit, käyttötapaukset ja lähtökohdat, jonka jälkeen käydään läpi varastonhallintaohjelmaa ja erityisesti sitä, mitä hyötyjä se tuo ja miten se tulee rakentaa yrityksien käyttöön sopivaksi. Dokumentissa kuvataan myös projektin kehitysprosessi, käytetyt ohjelmistot ja tekniikat. Dokumentin lopussa on ohjelman käyttöohje.

Varastonhallintasovellus on tarkoitettu helpottamaan verkkokauppapohjaisia yrityksiä. Sovelluksen avulla yritykset pystyvät hallitsemaan verkkokauppaan tulleita tilauksia sekä verkkokaupassa olevia tuotteita.

2 Tuotteen vaatimukset

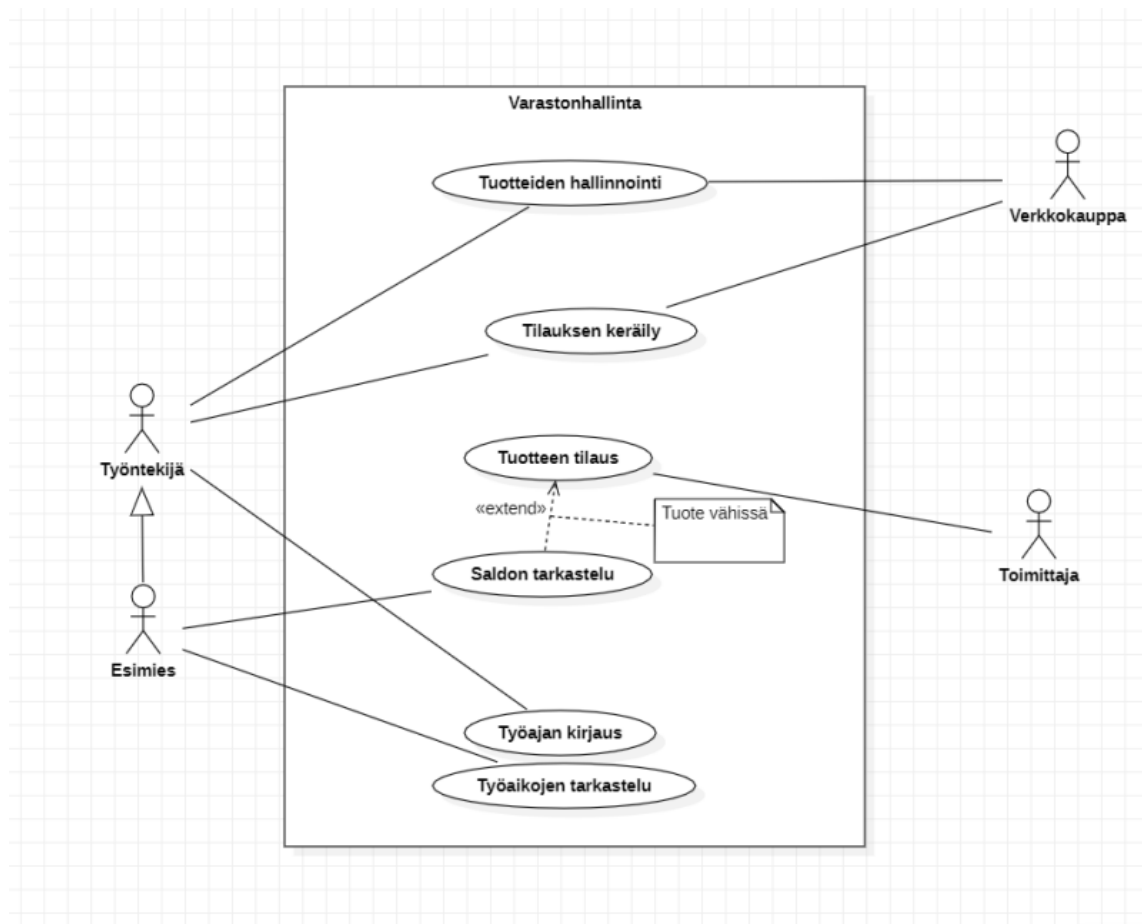
Sovelluksen avulla verkkokauppaan pystytään luomaan uusia tuotteita, muokkaamaan vanhoja tuotteita sekä tarkastelemaan ja käsittelemään tilauksia. Ohjelman asu on yksinkertainen, mutta ohjelma pyrkii tarjoamaan kaikki välttämättömät ominaisuudet verkkokaupan hallintaan.

Sovellus tarjoaa mahdollisuuden muokata kaikkien luotujen tuotteiden tietoja, kuten hintaa, kuvausta ja varastopaikkaa. Tuotteita voi myös hakea erilaisten parametrien mukaan.

Tilausten hallinta -ikkunan avulla voi tarkastella verkkokauppaan saapuneita tilauksia sekä jo käsitellyt tilaukset. Jokaista tilausta voidaan tarkastella erikseen, jolloin saadaan yksityiskohtaisempia tietoja tilauksesta, kuten tilauksen sisältämät tuotteet ja tilaajan tiedot. Tilaus voidaan merkitä käsitellyksi, jolloin tilauksen sisältämien tuotteiden määrä vähennetään varaston saldosta.

Sovelluksen avulla pystyy hallitsemaan suurta tietomäärää varaston tarpeen mukaan. Mahdollisista virhetilanteista ilmoitetaan käyttäjälle yksityiskohtaisesti.

3 Käyttäjäroolit ja käyttötapaukset



Kuva 1 Varastohallintasovelluksen käyttötapauskaavio

Yllä olevassa kaaviossa nähdään, että sovelluksella on pääasiassa neljä erilaista käyttäjää (aktoria).

Varastossa työskentelevä työntekijä käyttää järjestelmää tuotteiden sekä tilausten hallinnointiin. Tulevaisuudessa sovellukseen lisätään myös toiminto, jotta työntekijä voi kirjata työaikansa.

Varaston esimies tekee samoja asioita mitä työntekijä tekee sovelluksella, mutta lisäksi esimies voi korjata tuotteiden saldoja ja tarkastella työntekijöiden aikoja. Tätä roolia ei ole vielä ohjelmoitu sovellukseen.

Verkkokauppa lähettää varastolle tilauksia, joka käsitellään varaston järjestelmän sisällä.

Toimittaja, joka on tietolähde tuotteille.

Aiemmin mainitussa kaaviossa nähdään myös eri käyttötapauksia. Työntekijän käyttötapauksiin kuuluu tuotteiden hallinnointi, tilausten keräily sekä työajan kirjaaminen. Esimies tarkastelee saldoa, ja tilaa tuotetta tarvittaessa lisää, jos tuote on vähissä tai loppunut. Esimies voi myös tarkastella työaikoja.

Aktoreilla, jotka eivät sinänsä näy ohjelmistossa (verkkokauppa ja toimittaja), on myös käyttötapaukset sovelluksessa. Verkkokauppa on ns. sovelluksen tilaaja, eli käyttää sovellusta oman verkkokauppansa tuotteiden hallinnoinnissa ja tilauksen keräilyssä. Toimittaja taas toimittaa tuotteita varastoon.

4 Käsitteet, määritelmät ja ohjelmiston tietomalli

Projekti toteutettiin perinteisen ohjelmistokehityksen vaiheiden mukaisesti (määrittely, suunnittelu, toteutus, testaus ja käyttöönotto) siinä määrin kuin se katsottiin hyödylliseksi projektille.

4.1 Tekniset käsitteet

Tekniset vaatimukset varastohallintasovellukselle asetettiin seuraavasti: ohjelmiston tulisi olla helppokäyttöinen, sovelluksessa tulee käyttää hyväksi todettuja teknologioita - Java -ohjelmointikieli ja PostgreSQL objekti-relaatiotietokanta. Sovelluksessa hyödynnetään eri olio-ohjelmoinnin suunnittelumalleja ja MVC-arkkitehtuuria.

4.2 Kehitysympäristö

Eclipse

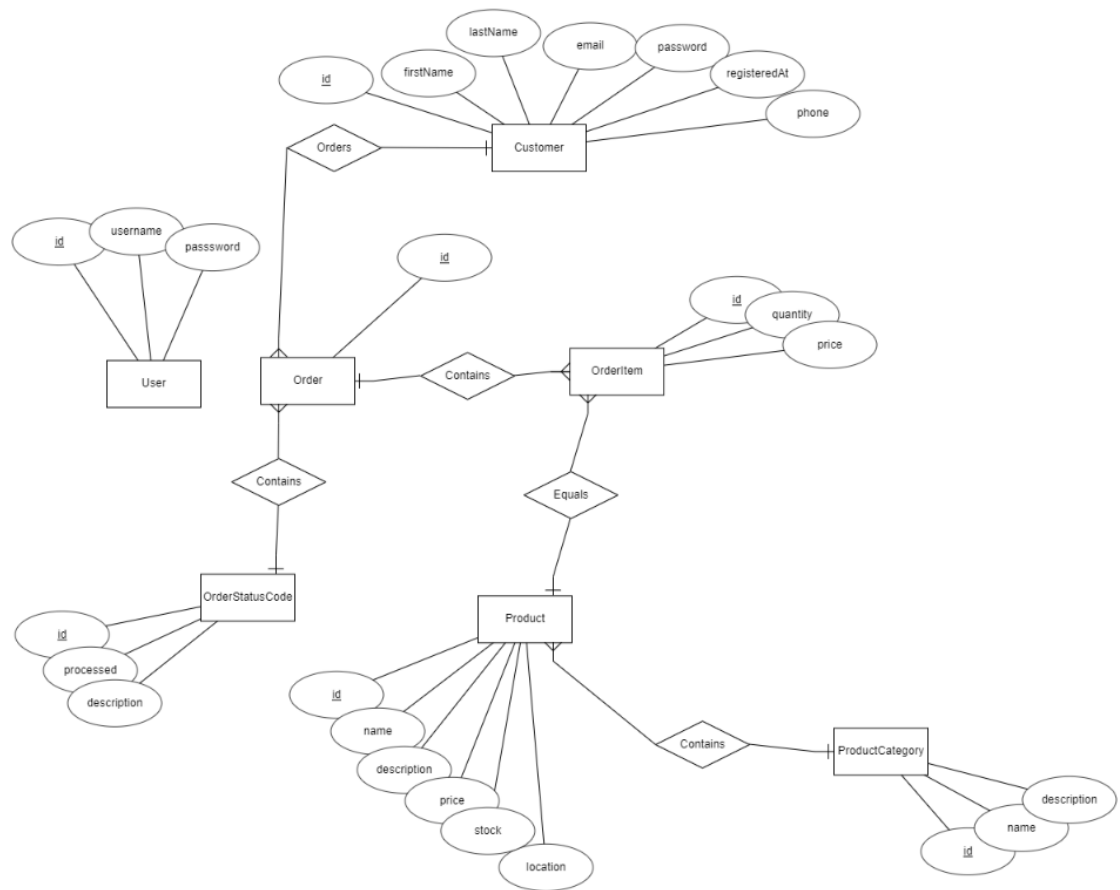
Ohjelmointiympäristönä käytettiin Eclipse, joka helpottaa ja nopeuttaa Java -sovellusten kehitystä. Eclipse:ssa on lukuisia toimintoja, jotka auttavat monissa eri tapauksissa.

Maven

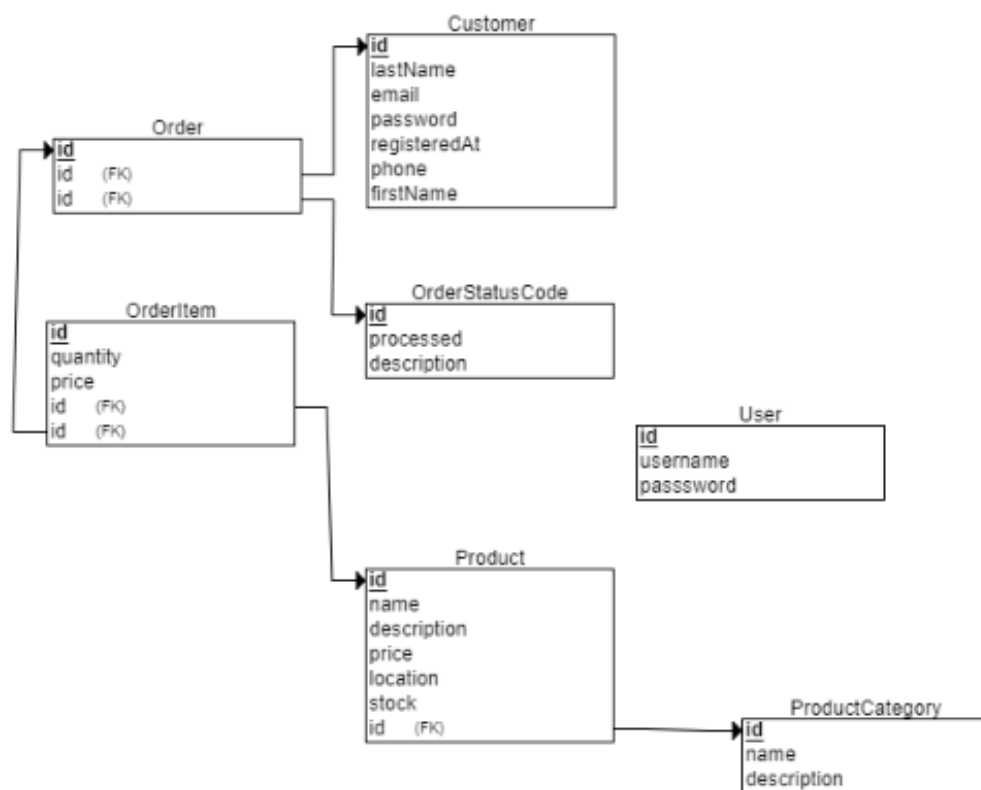
Maven on Apache kehittämä Java-sovellusten rakentamiseen ja ylläpitämiseen helpottava työkalu. Maven helpottaa ylläpitämään sovelluksen riippuvuuksia. Maven-projektiin liittyy POM-tiedosto, joka hallitsee rakennetta sekä kertoo tietoja projektista. POM:iin lisätään projektin perustiedot, riippuvuudet ja resurssit.

4.3 Sovelluksen tietomalli

Järjestelmästä voi nähdä mitä tuotteita varastossa on. Tuotteeseen on liitetty tietoja, kuten varastosaldo ja hyllypaikka. Järjestelmään kirjataan tilauksen kulku tuotteiden vastaanotosta, käsittelyyn ja toimitukseen asiakkaalle.

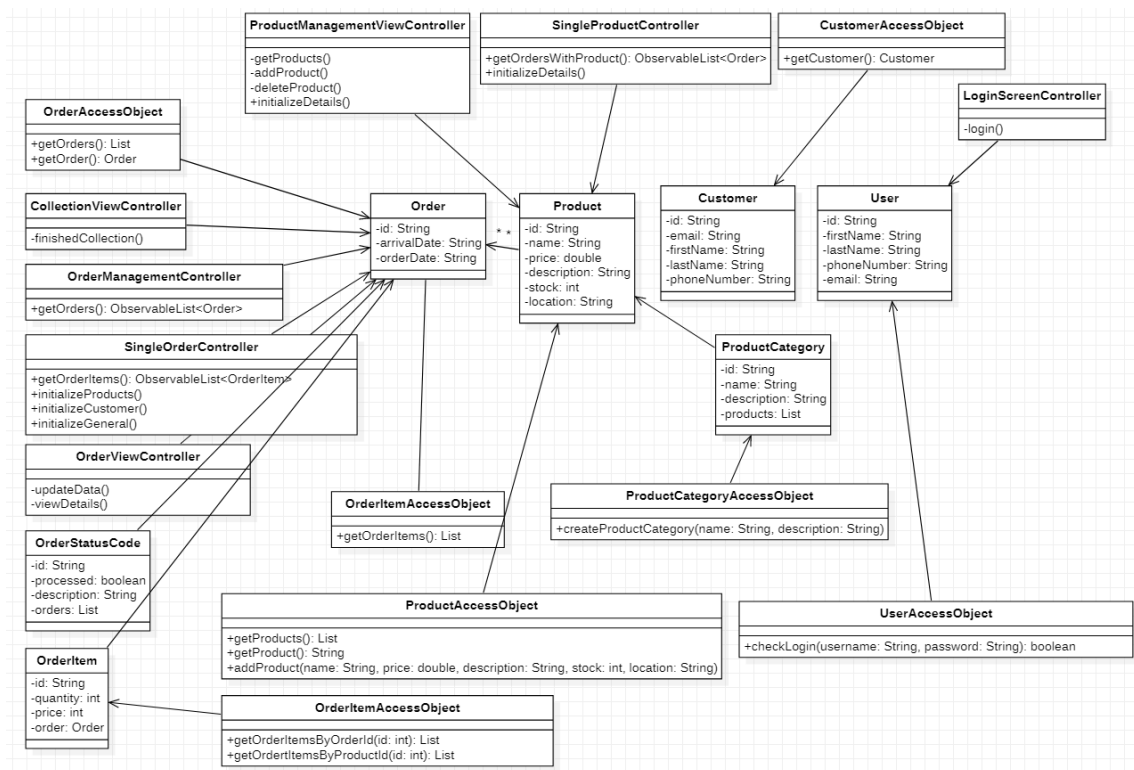


Kuva 2 Sovelluksen tietomalli (ER-malli)



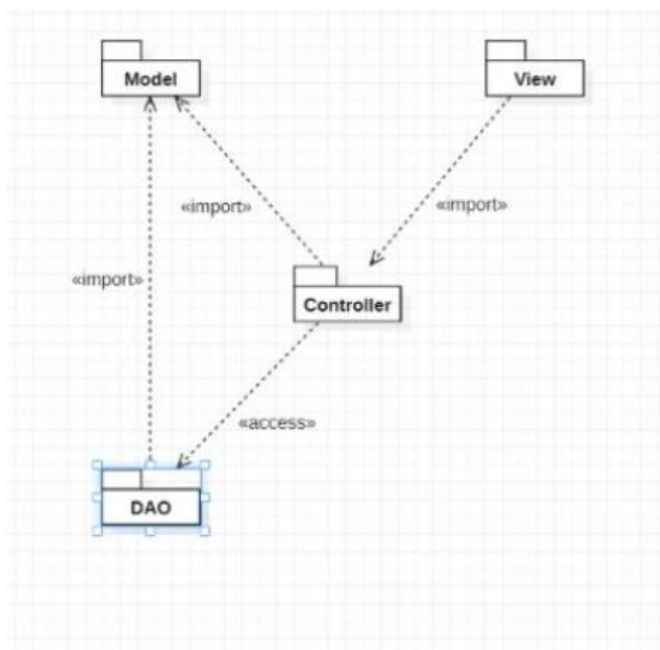
Kuva 3 Sovelluksen tietomalli (relaatiotietokantakaavio)

5 Ohjelmiston rakenne



Kuva 4 Varastohallintasovelluksen luokkakaavio

Sovellus on suunniteltu noudattamaan MVC-mallia (View, Model, Controller). Näkymät, mallit sekä näkymien kontrollerit on jaettu erilleen toisistaan omiin pakkauksiinsa. Näin sovelluksen rakenne pysyy siistinä ja ymmärrettävyys on helpompaa.



Kuva 5 Varastohallintasovelluksen pakkauskaavio

Pakkauskaavio sisältää neljä pakkausta, joihin on asetettu näihin kuuluvat luokat.

Model – pakkauksen sisältö

- **Product** on tuote, joita verkkokauppa sisältää. Tuotteen instanssimuuttujat sisältävät konkreettiset tiedot tuotteesta kuten nimen, hinnan ja lisätiedot. Lisäksi malli sisältää tiedon varastopaikasta sekä varastosaldosta.
- **Customer** on verkkokaupan asiakas, joka tekee tilauksia verkkokaupasta. Asiakas sisältää välttämättömät tiedot tilauksen tehneestä henkilöstä, kuten nimen, sähköpostiosoitteen sekä puhelinnumeron.
- **OrderItem** on tilauksen sisältämä tuote. OrderItem sisältää viitteen Product-luokkaan ja saa sitä kautta tiedon tilatusta tuotteesta. Erona on kuitenkin se, että OrderItemin ominaisuuksiin kuuluu myös tieto tilatun tuotteen määrästä, yhteishinnasta sekä viite tilaukseen, johon kyseinen OrderItem kuuluu.

- **Order** on asiakkaan tekemä tilaus verkkokaupasta. Tilaus koostuu aiemmin mainituista **OrderItem**teistä. Order sisältää myös tiedon tilauksen tehneestä asiakkaasta, tilauspäivämäärän sekä tiedon siitä, onko tilaus käsitelty varastolla.
- **OrderStatusCode** on malli, jota tilaus käyttää määrittääkseen tiedon tilauksen tilasta. Tällä hetkellä mahdolliset tilat ovat käsitelty ja käsittelemätön, mutta tulevaisuudessa näitä tiloja voisi olla lisää.
- **ProductCategory** mahdollistaa tuotteiden (**Product**) kategorioimisen varastolla. Malli sisältää kategorian nimen sekä lisätietoa kategoriasta. Jokaisen varastossa olevan tuotteen voi kategoroida omaan kategoriansa. Esimerkiksi housut voitaisiin asettaa alaosa – kategoriaan ja paidat ja hupparit yläosa – kategoriaan. Ominaisuus tullaan kehittämään myöhemmin.
- **User** on varastonhallintasovelluksen käyttäjä. Tällä hetkellä malli sisältää vain käyttäjätunnuksen sekä salasanan, mutta tulevaisuudessa tämä malli sisältää laajempaa tietoa sovelluksen käyttäjästä. Lisäksi käyttäjälle tullaan tulevaisuudessa asettamaan rooli (esim. työntekijä, esimies), joka määrittää sovelluksen käyttäjän käyttöoikeudet.

DAO – pakkauksen sisältö

- **OrderAccessObject** mahdollistaa tietokantakutsut liittyen tilausluokkaan (**Order**).
- **OrderItemAccessObject** mahdollistaa tietokantakutsut liittyen tilauksen tuotteen luokkaan (**OrderItem**).
- **ProductAccessObject** mahdollistaa tietokantakutsut liittyen verkkokaupan tuote luokkaan (**Product**).

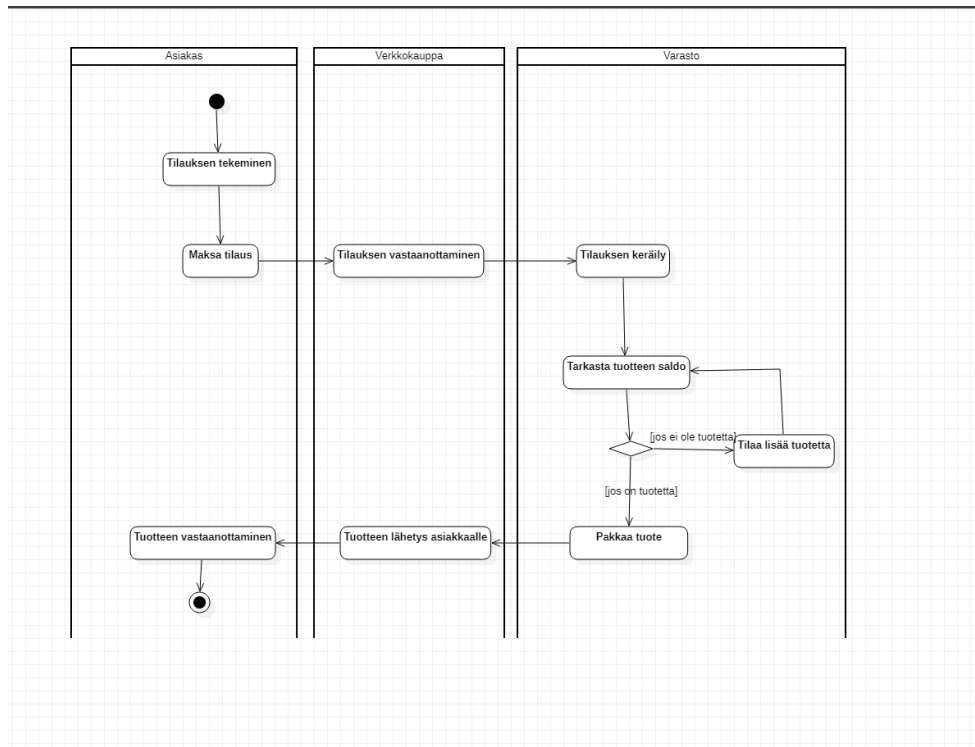
- **ProductCategoryAccessObject** mahdollistaa tietokantakutsut liittyen tuotteen kategorioimiseen (**ProductCategory**).
- **UserAccessObject** mahdollistaa tietokantakutsut sovelluksen käyttäjiin liittyvään luokkaan (**User**).

Controller – pakkaus sisältää sovelluksen kontrollerit. MVC – mallissa näkymät on eritelty sovelluksen toiminnallisuudesta ja näkymiä voidaan hallita kontrolleiden avulla. Jokaiselle näkymälle on luotu oma kontrollerinsa, jonka avulla näkymässä tapahtuvat toiminnallisuutta vaativat operaatiot määritellään.

View – pakkaus sisältää sovelluksen näkymät. Näkymä tarkoittaa käytännössä sovelluksen käyttäjän näkemää ikkunaa sovellusta käyttäessä. Esimerkiksi sovellukseen kirjautuminen on oma ikkunansa, mistä siirrytään seuraavaan käyttäjän syöttäessä oikeat tunnistautumistiedot.

6 Ohjelmiston toiminta

Ohjelmisto helpottaa varaston toimintaa toimimalla tuotteiden ja tilausten läpikulun kirjanpitolvälineenä.



Kuva 6. Varastonhallintasovelluksen aktiviteettikaavio

Tilanne alkaa asiakkaan tekemästä tilauksesta. Asiakas-uimaradalla on asiakkaan tarpeelliset aktiviteetit tuotteen saamiseksi. Verkkokauppa-uimaradalla vastaanotetaan ja lähetetään tilaus, joka toteutuu varasto-uimaradalla olevissa aktiviteettikohdissa. Seuraavassa listassa on kuvattu ohjelmistomme toiminnan kannalta aktiviteettikaavion olennaisimmat osat.

- **Tilauksen tekeminen** Asiakas tekee tilauksen verkkokaupan nettisivulla.
- **Tilauksen vastaanottaminen** Verkkokauppa saa tilauksen, ja lähettää keräilypyynnön varastoon. Tässä vaiheessa tilaus saapuu varastonhallintaohjelmaan.
- **Tilauksen keräily** Työntekijä tarkastaa tilattujen tuotteiden tiedot järjestelmästä ja keräilee ne.
- **Tarkasta tuotteen saldo** Työntekijä varmistaa, että tuotteen saldo on riittävä, ja tilaa tuotetta tarvittaessa lisää.

- **Tuotteiden pakkaus ja lähetys** Tässä vaiheessa tilaus lähetetään varastolta ja merkitään ohjelmaan käsitellyksi

General
Recipient
Products

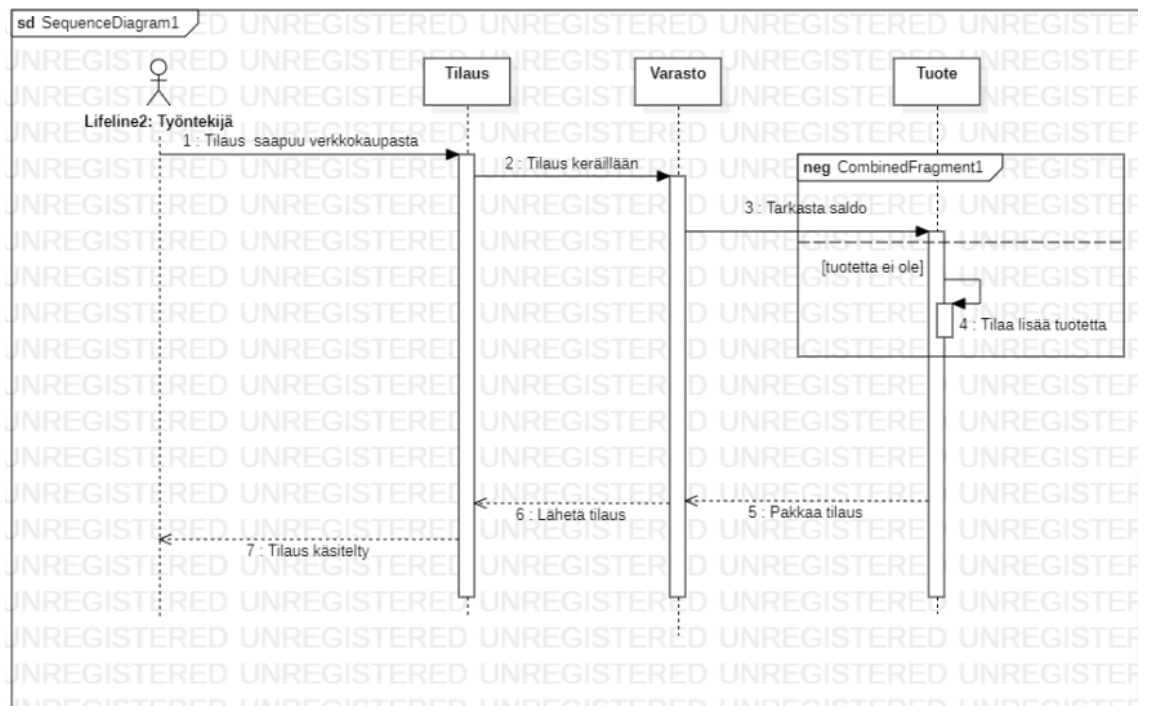
Back

☐ Order is ready

Process order

Id	Name	Quantity	Picking	Current Stock	Total Price	Location
f7d5cc0c-6593-4cf...	White Design T-Shirt	1		3	79.99	5C13
9ec5350d-4856-4...	Red Pyjama	2		10	39.98	C8784D
d9722d6c-0987-4a...	Black T-Shirt	1		12	5.99	5C12
4e0f2f6e-d461-4c...	Levi's Beans	1		10	89.99	C8784F
f0f3dbc1-8a90-46...	Black Ripped Jeans	1		5	49.99	5C14

Kuva 7. Tilausten keräily- ja käsittelynäkymä



Kuva 8. Varastohallintasovelluksen sekvenssikaavio

Sekvenssikaavion käyttötapauksessa tarkastellaan myös tilauksen käsittelyä työntekijän näkökulmasta. Tapaus lähtee liikkeelle verkkokaupasta saadusta tilauksesta, jonka työntekijät keräilevät. Jos tuotteiden saldo ei riitä, tuotteita tilataan lisää. Jos saldo on riittävä ja keräily onnistuu, niin tuotteet pakataan ja tilaus lähetetään. Tilaus merkitään lopuksi käsitellyksi.

Tilausten hallintänäkymässä työntekijä voi tarkastella tilausten tietoja, kuten lähetysosoitetta, tai tilauksella olevia tuotteita. Keräilijöitä varten näkymässä on muokattava numerokenttä, johon keräilijä merkitsee kerätyn määrän. Keräilijä voi merkitä tilauksen valmiiksi, jolloin sen tila muuttuu tietokannassa.

Tilausten hallinnan lisäksi sovelluksessa voi hallinnoida varaston tuotteita. Käyttäjä voi lisätä, poistaa ja muokata niitä. Esimerkiksi tuotteen saldoa voi vähentää, tai varastopaikkaa muuttaa.

7 Kehitysprosessi ja kehitysvaiheen tekniikat

Projekti aloitettiin kartoittamalla mitä toimintoja varastohallintajärjestelmän ensimmäisen versioon tarvitaan. Sen lisäksi kartoitettiin, mitä tietoja tallennetaan tietokantaan. Varastohallintajärjestelmältä vaadittiin seuraavat toiminnot:

- Tuotteiden lisäys, muokkaus, poisto
- Käyttäjäystävällinen ulkoasu
- Tilauksien hallinta

7.1 Ensitoimet tilauksien hallintaan

Sovelluksen ensimmäisessä versiossa huomio kohdistettiin verkkokaupan tuotteiden käsittelyyn ja tilauksien hallintaan. Toimivuuden takaamiseksi taustalle kehitettiin mock-tyylinen verkkokauppa, jossa verkkokaupan asiakas pystyy

selaamaan tuotteita, lisäämään näitä ostoskoriin ja lopulta lähettämään tilauksen. Varastohallintasoftware vastaa jälkitoimista, joita tilauksen käsittely vaatii.

7.2 Verkkokaupan varasto ja tuotteet

Verkkokaupan tuotteiden vaihtuessa jatkuvasti on sovelluksessa pyritty yksinkertaistamaan näiden hallintaa. Sovelluksen käyttäjällä on mahdollisuus lisätä uusia tuotteita, muokata jo varastossa olevia sekä poistaa tuotteita. Näitä muutoksia tehdessä varaston tuotevalikoima sekä verkkokaupan sisältö päivittyvät reaaliajassa.

7.3 Käyttöliittymäsuunnittelu ja toteutus

Käyttöliittymän suunnittelussa pyrittiin ottamaan huomioon visuaaliset seikat, yksinkertaisuus sekä toimintojen käytettävyys. Ainoat vaatimukset käyttöliittymälle olivat käyttäjäystävällinen ulkoasu sekä helppo opittavuus. Piirrettiin alustavia kuvia eri näkymistä Figma:ssa. Näkymät toteutettiin Java-FX hyödyntäen SceneBuilder -ohjelmassa.

Järjestelmä oli tarkoitus toteuttaa itsenäisenä sovelluksena. Sovelluksen ohjelmointikieleksi valittiin Java.

Sovelluksen toteuttamiseen on käytetty Scrum -ketterää menetelmää, ja sprinttien suunnittelun apuna projektinhallintasoftware Nektionia. Nektioniin on kirjattu sovelluksen ominaisuuksia ns. käyttäjätarinoina ja näitä ominaisuuksia on toteutettu sprinttien aikana. Kaikkia ominaisuuksia ei ole vielä tässä vaiheessa toteutettu.

Ohjelmointiympäristönä on käytetty Eclipseä, josta luotu Maven-projekti. Projektissa on käytetty eri liitännäisiä: JavaFX (SceneBuilder), JUnit, PostgreSQL ja Hibernate. Versionhallintana käytettiin GitLabia, ja tulevaisuutta varten asennettiin jatkuvan kehityksen työkalu Jenkins.

Sovelluksen testaukseen käytettiin JUnit-kirjastoa. Testejä tehtiin tietokannan apuluokille, joissa testataan, ettei tietokantaan voi syöttää virheellistä tietoa tai duplikaatteja.

8 Käyttöohje

Sovelluksen avatessa kirjaudutaan sisään ennalta määritetyillä tunnuksilla. Alkuvalikossa on navigaationäppäimet tilausten- ja tuotteidenhallintaan. Tilausten hallintaikkunassa voi valita tilauksen, jota haluaa tarkastella. Valinnan jälkeen välilehdillä on vaihtoehtona merkitä tilaus käsitellyksi ja tarkastella sen tietoja. Tuotteiden hallintaikkunassa on vastaavat toiminnallisuudet, joiden lisäksi käyttäjällä on mahdollisuus lisätä ja muokata tuotteita.

9 Yhteenveto

Tämän dokumentin tarkoitus on avata sovelluksen tilaajalle sovelluksen teknistä toteutusta sekä ohjeistaa sovelluksen käytössä. Teknisen toteutuksen avaamiseksi on hyödynnetty sovelluksen kehityksessä käytettyjä kaavioita. Kaaviot on myös selitetty sanallisesti, jotta dokumentin lukija ymmärtää kehityksen perusidean.

Dokumentissa esitellään sovelluksen tärkeimmät toiminnallisuudet, eli tilausten ja tuotteiden hallinta. Sovellus auttaa tilausten toimitusketjussa tarjoamalla näkymät keräilyyn ja tilauksen käsittelyyn. Lisäksi sovellus toimii apuvälineenä varaston yleiseen kirjanpitoon.

Lopputuloksena saatiin toimiva varastonjärjestelmän ensimmäinen versio. Järjestelmästä pyrittiin saamaan helposti laajennettava. Kehitysideoita sovellukselle on ainakin esimiesroolin toteutus ja työaikaseuranta. Esimies voisi luoda järjestelmään uusia käyttäjiä, käytännössä siis tekisi uuden käyttäjän aina uudelle varastotyöntekijälle. Esimies pystyisi myös tarkastelemaan työntekijöiden työaikoja.