



Anna Raevskaia, Antti Taponen, Samu Luoma, Nicklas Sundell,
Lassi Piispanen

Handle that POS-kassajärjestelmä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknologian tutkinto-ohjelma

Tekninen dokumentti

5.5.2022

Sisällys

1	Sovelluksen visio	1
1.1	Sovelluksemme "Handle that POS"	1
1.2	Sovelluksen hinta-arvio	1
1.3	Riskit	1
1.4	Edut	2
1.5	Henkilöstötarve	2
1.6	Virstanpylväät	2
2	Sovelluksen yleiskuvaus	2
2.1	Kehittäjät	2
2.2	Maven riippuvuudet ja lisäosat	3
3	Sovelluksen suunnitteluratkaisut	4
4	Testit	5
5	Operaatioiden kuvaus	6
5.1	Ongelman sattuessa	7
6	Vaatimusten kuvaus	8
6.1	Käyttötapaukset	8
6.2	Käyttäjätarinat	8
6.3	Laadulliset vaatimukset	9
7	Järjestelmädokumentaatio	10
7.1	MVC - Model-View-Controller (Malli-Näkymä-Ohjain)	10
7.2	Käyttöoikeudet	11
7.3	Näkökulmia jatkokehitykseen	11
7.3.1	Asynkroninen tietokannan päivitys	12
7.3.2	Kuittivaihtoehtoja	12
7.3.3	Bonusasiakkaista enemmän tietoja	13
7.3.4	Tuotteiden palautus	13
7.3.5	Mobiililaitetuki	13
7.3.6	Mobiiliviivakoodinluku	13
	Lähteet	14

1 Sovelluksen visio

1.1 Sovelluksemme "Handle that POS"

- ylläpitää maksutapahtumia.
- on yksinkertainen, työkalun ei kuulu olla taakka työntekijälle.
- sisältää helppokäyttötoimintoja kokeneille käyttäjille, muokattavuus tehostaa käyttöä.
- on yhteensopiva kosketusnäyttöjen kanssa, suurin osa kassajärjestelmistä toimii kosketusnäyttöjen avulla.

1.2 Sovelluksen hinta-arvio

- Kehitys tapahtuu opiskelijaryhmän harjoitustyönä, eli ilmaisena.
- Sovellus on tarkoitus julkaista ilmaistuotteena (support as a service, SaaS), asiakas voi kuitenkin tilata muokkauspalveluja, asennuspalveluja ja teknistä tukea sovelluksen kehittäjiltä. Asiakas hankkii itse oman tietokantaratkaisunsa. Voimme auttaa tietokannan järjestämisessä ja asentamisessa maksua vastaan.

1.3 Riskit

- Koska sovellus on kehitetty opiskelijoiden toimesta ammattilaisten sijaan, skaalautuvuudessa ja jatkokehityksessä saattaa esiintyä ongelmia.
- Tietokantaratkaisu ei ylläpidä varmuuskopioita, joten oikeassa maailmassa riski tietojen katoamiselle ongelman sattuessa on suuri. Jos tietokanta siirretään esim. Amazonille, riskit vähenevät huomattavasti, koska palveluun kuuluu varmuuskopiot.

- Opiskelijoiden työmoraaalin lasku.

1.4 Edut

- Sovellus on ilmainen, joten kuka tahansa voi kokeilla sovellusta tarpeen vaatiessa.
- MVC-malli helpottaa sovelluksen uudistamista.

1.5 Henkilöstötarve

- Sovelluksella ei ole erityisiä henkilöstövaatimuksia, sovellukseen on vakiona sisäänrakennettu neljä käyttöoikeustasoa, jotka vaikuttavat siihen, mihin tietoon tai toiminnallisuuteen käyttäjä pääsee käsiksi.

1.6 Virstanpylväät

- Sovelluksen kehitys alkoi 10.1.2022.
- Sovelluksen julkaisupäivämäärä: 6.5.2022.

2 Sovelluksen yleiskuvaus

2.1 Kehittäjät

- Anna Raevskaia (anna.raevskaia@metropolia.fi)
- Antti Taponen (antti.taponen@metropolia.fi)
- Nicklas Sundell (nicklas.sundell@metropolia.fi)
- Samu Luoma (samu.luoma@metropolia.fi)
- Lassi Piispanen (lassi.piispanen2@metropolia.fi)

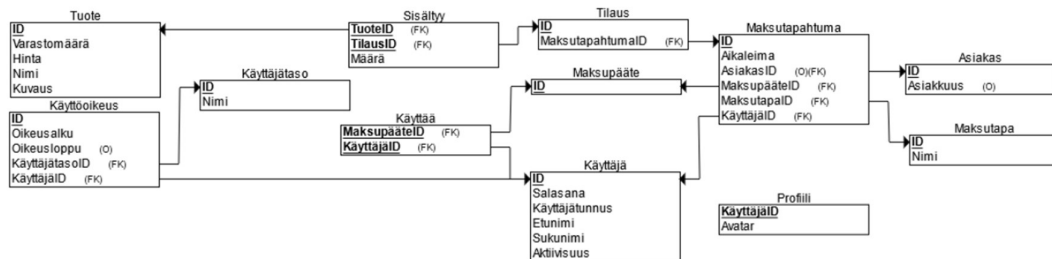
Handle that POS on luotu JavaFX Maven projektina, jonka yhteys tietokantaan käsitellään Hibernaten avulla. Ohjelmointiympäristönä käytämme JetBrainsin IntelliJ IDEA Ultimatea.

2.2 Maven riippuvuudet ja lisäosat

- At.favre.lib
 - Bcrypt (0.9.0), <https://github.com/patrickfav/bcrypt>
- Org.apache.maven.plugins
 - Maven-surefire-plugin (3.0.0-M5)
 - Maven-compiler-plugin (3.8.1)
- Org.junit.jupiter
 - Junit-jupiter-params (5.7.2)
 - Junit-jupiter-api (5.7.2)
- Mysql
 - Mysql-connector-java (8.0.28)
- Org.hibernate
 - Hibernate-core (5.6.5 Final)
- Org.controlsfx
 - Controlsfx (11.1.1)
- Org.openjfx
 - Javafx-controls (17.0.2)
 - Javafx-fxml (17.0.2)
 - Javafx-controls (17.0.2)
 - javafx-swing (18)
 - javafx-media (18)
- Com.dlsc.formsfx
 - Formsfx-core (11.4.2)

3 Sovelluksen suunnitteluratkaisut

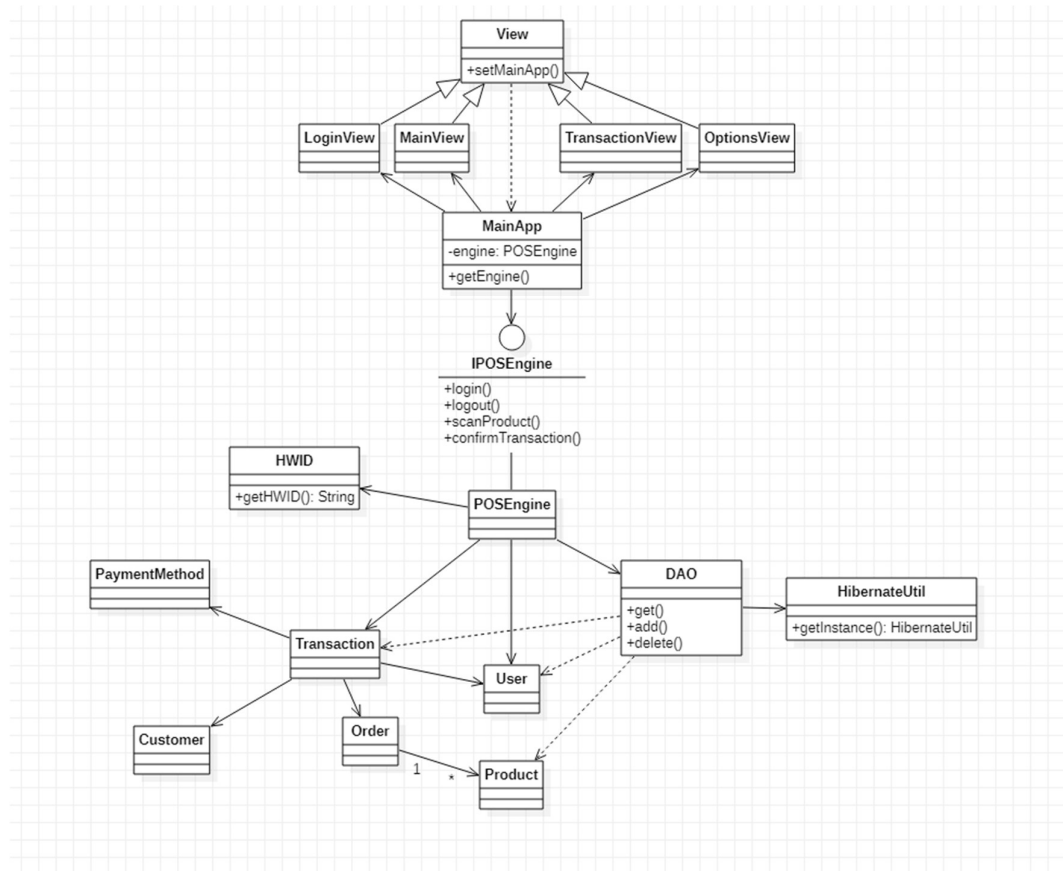
Handle that POS käyttää MariaDB relaatiotietokantajärjestelmää (Kuva 1, sivu 4) maksutapahtumien sekä muiden siihen liittyvien tietojen tallentamisessa. Sovelluksen kehittämisessä käytetään Java-ohjelmointikieltä, JavaFX käyttöliittymäkirjastoa graafisen käyttöliittymän luomisessa sekä olio-relaatio-mallinnuskirjasto Hibernatea tietokantayhteyksien rakentamisessa.



Kuva 1. Tietokannan relaatiokaavio

Sovelluksessa käytettyjä ratkaisuja:

- Sovellus noudattaa MVC-mallia jatkokehittämisen helpottamiseksi. (Kuva 2, sivu 5)
- Päädyimme tekemään omat DAO luokat eri olioiden tallentamiseksi, sillä tarvitsimme erilaisia operaatioita eri olioita käsitellessä.
- JDBC-cascading annotaation avulla Transaction-luokan ilmentymiä käsitellessä kaikki sen alaiset oliot joko tallennetaan tai haetaan tietokannasta tarpeen mukaan.
- Pyrimme pitämään luokkia löyhästi liitettyinä toisiinsa ja saavutimme tämän tavoitteen käyttämällä POSEngine-luokkaa eräänlaisena mediaattorina, joka ylläpitää yhteyksiä niiden välillä.
- Tietoturvallisuuden parantamiseksi käytämme bcrypt-tiivistealgoritmia salasanojen salauksessa.
- Aina tuotteen skannauksen yhteydessä POSEngine tarkastaa, onko tilaus olemassa. Jos tilausta ei ole, ohjelmisto luo uuden. Paikallinen tilaus ilmentymä poistetaan maksutapahtuman valmistuttua ja tallennuttua tietokantaan.

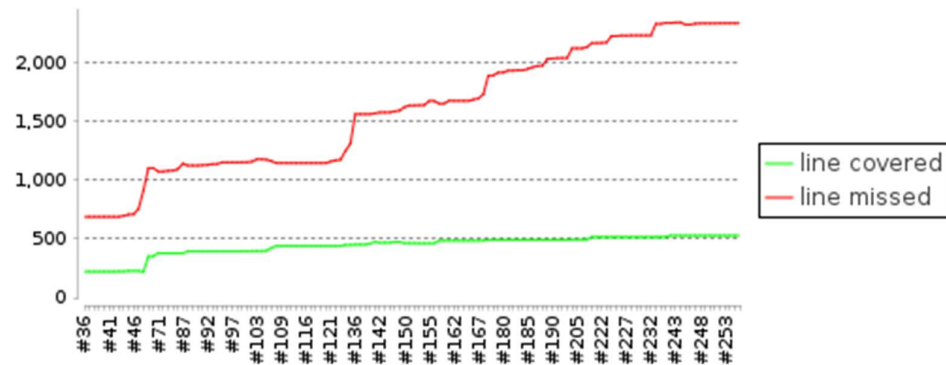


Kuva 2. Yksinkertaistettu luokkakaavio

4 Testit

Sovellusta on testattu sekä manuaalisesti että automaattisesti yksikkötesteillä. Automaattinen testaus on suoritettu jatkuvan integraation palvelinohjelmistolla, Jenkinsillä. Testaamattoman koodin määrä kasvaa, koska suurin osa lisääntyneestä koodista on testaamatonta (Kuvat 3 ja 4, sivu 6). JavaFX-koodin testaus

on suoritettu käsin sovellusta ajamalla. Aina kun uusi virhe löytyi, tilanteesta raportoitiin kehitysryhmän tiedotetaululle. Tietokantaa käyttäviä testejä varten on oma testitietokanta.



Kuva 3. JaCoCo raportti testatuista ja testaamattomista riveistä

Overall Coverage Summary

name	instruction	branch	complexity	line	method	class
all classes	15% M: 9881 C: 1751	12% M: 496 C: 70	23% M: 545 C: 159	18% M: 2334 C: 528	33% M: 279 C: 136	39% M: 35 C: 22

Coverage Breakdown by Package

name	instruction	branch	complexity	line	method	class
model.DAOs	M: 462 C: 523 53%	M: 68 C: 18 21%	M: 46 C: 31 40%	M: 203 C: 201 50%	M: 9 C: 25 74%	M: 0 C: 7 100%
model.classes	M: 776 C: 1228 61%	M: 57 C: 52 48%	M: 80 C: 128 62%	M: 192 C: 327 63%	M: 40 C: 111 74%	M: 3 C: 15 83%
model.interfaces	M: 0 C: 0 100%	M: 0 C: 0 100%	M: 0 C: 0 100%	M: 0 C: 0 100%	M: 0 C: 0 100%	M: 0 C: 0 100%
view	M: 8643 C: 0 0%	M: 371 C: 0 0%	M: 419 C: 0 0%	M: 1939 C: 0 0%	M: 230 C: 0 0%	M: 32 C: 0 0%

Kuva 4. JaCoCo yleiskuva sovelluksen testaamisesta

5 Operaatioiden kuvaus

Sovelluksen tietokannan varmuuskopiointi tehdään lähteenä olevien ohjeiden mukaisesti [1].

Kun sovelluksella on asetettu pikanäppäimiä, ilmestyy sovelluksen juureen "hot-keys.properties"-tiedosto, jota voi käyttää muissa kassapäätteissä ja varmuuskopioida. Myös HandleThatPos.properties -tiedostoa voi käyttää muissa maksupäätteissä ja varmuuskopioida. Tämä tiedosto sisältää yrityksen tiedot (osoite yms.), joita käytetään mm. kuiteissa.

Olemme pyrkineet antamaan ajonaikaisten virheiden sattuessa asianmukaiset ilmoitukset, jotka ohjaavat käyttäjää virheen korjaamiseen. Voit referoida tämän dokumentin "Ongelman sattuessa" yleisimpien virhetilanteiden ratkaisuun. Käytöstä voi olla myös hyötyä. Jos kuitenkin löytyy ylitsepääsemätön ongelma, suosittelemme ottamaan yhteyttä sovelluksen kehittäjiin.

5.1 Ongelman sattuessa

Tässä tarjoamme ratkaisuja yleisimpiin ongelmatilanteisiin, suorita ratkaisut järjestyksessä alkaen a-kirjaimesta. Jokaisen ratkaisun jälkeen varmista onko ongelma vielä olemassa vai ratkaistu.

Yleisimmät ongelmatilanteet:

- Eikö sovellus käynnisty?
 1. Tarkista nettiyhteys.
 2. Käynnistä sovellus uudelleen.
 3. Käynnistä tietokone uudelleen.
 4. Ota yhteys yrityksesi it-asentajaan tai yritykseen, jonka kautta ostat it-tukea.
- Eivätkö tunnukset pelitä?
 5. Varmista että kirjoitit tiedot oikein
 6. Ota yhteys yrityksesi it-asentajaan tai yritykseen, jonka kautta ostat it-tukea.
- Sovellus jäätyy tilauksen vahvistuksen yhteydessä
 7. Tarkista nettiyhteys ja varmista että tilaus meni läpi Asetukset näkymän kautta.
 8. Jos tilaus ei mennyt läpi suorita tilaus uudelleen.
 9. Ota yhteys yrityksesi it-asentajaan tai yritykseen, jonka kautta ostat it-tukea.
- Tuotteita ei tilauksessa
 10. Tarkista nettiyhteys.
 11. Kokeile käynnistää sovellus uudelleen.

12. Käynnistä tietokone uudelleen.

13. Ota yhteys yrityksesi it-asentajaan tai yritykseen, jonka kautta ostat it-tukea.

6 Vaatimusten kuvaus

6.1 Käyttötapaukset

- Tuotteen skannaus ja tilauksen luominen: Kun tuote skannataan, järjestelmä luo uuden tilauksen, jos sellaista ei ole vielä olemassa. Jos tilaus on olemassa, skannattu tuote lisätään tilauksen tuotelistaan.
- Tilauksen muokkaus: Tilausta voi tarvittaessa muokata. Tuotteiden poisto ja lisääminen ovat tilauksen muokkaustoimintoja.
- Tilauksen vahvistus: Tilaus täytyy vahvistaa ennen maksutapahtumaa. Vahvistuksen yhteydessä valitaan maksutapa (maksukortti tai käteinen), kuitin tulostus sekä kanta-asiakkuus.
- Maksu: Kun tilaus on vahvistettu, asiakas suorittaa maksun. Jos asiakas on bonusasiakas, hän saa alennuksen tuotteista ja bonusasiakkuustunnus tallennetaan maksutapahtumaan.
- Kuitin tulostus: Kassahenkilö voi valita kuitin tulostuksen maksun yhteydessä.
- Tuotteen ja käyttäjien hallinta: Käyttäjät, joilla on MANAGER-tason käyttöoikeudet voivat lisätä, poistaa ja muokata tuotteita ja käyttäjiä tietokannassa.

6.2 Käyttäjätarinat

Käyttäjätarinat ovat tärkeysjärjestyksessä. Ylin on tärkein, alin vähiten tärkeä.

- Kassahenkilönä haluan nähdä käyttöliittymän
- Kassahenkilönä haluan suorittaa kassatapahtuman
- Kassahenkilönä haluan poistaa tuotteen jota asiakas ei haluakaan ostaa
- Kassahenkilönä minun pitää poistaa tuote, minkä skannasin vahingossa tuplana
- Kassahenkilönä haluan skannata tuotteen

- Kassahenkilönä haluan tulostaa kuitin maksutapahtumasta
- Kassahenkilönä haluan kirjautua maksupäätelle
- Yrityksen johtajana haluan mahdollisuuden lisätä uusia tuotteita kassajärjestelmän tietokantaan
- Yrityksen johtajana haluan mahdollisuuden poistaa tuotteita kassajärjestelmän tietokannasta
- Myymäläpäällikkönä haluan asettaa yritykseni tiedot, jotta ne näkyvät kuitissa
- Kassahenkilönä haluan pikakuvakkeita usein käyttämilleni toiminnoille
- Kassahenkilönä haluan valita pikanäppäinten nimet vapaasti
- Monikulttuurisen yrityksen johtajana haluan englanninkielisen lokalisaation kassajärjestelmästä
- Kaupan asiakkaana haluan hyötyä etuhinnoittelusta
- Yrityksen johtajana haluan tarjota asiakkaille kanta-asiakkuusohjelman
- Myymäläpäällikkönä haluan itsepalveluversion
- Myymäläpäällikkönä haluan tarkastella käyttäjien tietoja
- Sovelluksen käyttäjänä haluan tietää käynnistyykö sovellus
- Kassahenkilönä haluan opasteita sovelluksen käytön tueksi
- Punavihervärisokeana kassatyöntekijänä haluan, että kassajärjestelmässä käytetään sinistä väriä vihreän sijaan
- Yrityksen johtajana haluan tietoa tuotteiden myynnistä ja nähdä tilastoja
- Myymäläpäällikkönä haluan, että työntekijöillä on mahdollisuus ladata kuva profiiliinsa
- Itsepalvelukassan käyttäjänä haluan valita itselleni sopivan kielen
- Kassahenkilönä haluan valita käyttöliittymän teeman (vaalea/tumma)
- Käyttäjänä haluan käyttäjäpalautetta ostoksestani

6.3 Laadulliset vaatimukset

Varmistusponnahdusikkunat ja ilmoitukset kertovat käyttäjälle tietoa sovelluksen tilasta. Jos käyttäjä yrittää esimerkiksi lisätä tietokannasta puuttuvaa tuotetta tilaukseen, epäonnistuneesta yrityksestä ilmoitetaan käyttäjälle.

Koska kassasovelluksia käytetään vakiintuneesti kosketusnäyttöjen avulla, pitää kaikkien UI-elementtien tukea tätä käyttötarkoitusta.

Saavutettavuus sovelluksessa on huomioitu mm. kuvakkeilla (ratas ilmaisee asetuksia, kynä muokkausmahdollisuutta, roskakori poistomahdollisuutta). Kaikille kuvakkeille tarjotaan tekstivaihtoehtot. Tuoteselaus-listassa on myös väriominaisuus, joka värittää rivin tuotteiden määrän mukaan ja on värisokeaystävällinen. Haluamme vastaavia saavutettavuustoiminnallisuuksia myös uusille ominaisuuksille. Itsepalvelukassan käyttö

7 Järjestelmädokumentaatio

Projekti on toteutettu JavaFX Maven projektina. Sovellus toteuttaa MVC (Model View Controller) -arkkitehtuuria, jossa näkymä, malli, sekä näkymän ja mallin yhdistävä kontrolleri ovat eristetty omiksi kokonaisuuksiksi.

7.1 MVC - Model-View-Controller (Malli-Näkymä-Ohjain)

Sovelluksen konkreettisessa MVC-toteutuksessa näkymät ovat pelkästään JavaFX:n FXML-tiedostoja ja niiden CSS-tyylitiedostoja. Näkymissä ei itsessään ole mitään toiminnallisuutta, vaan niiden ainoa tarkoitus on määrittää käyttöliittymäelementit, niiden sijainnit, tyylit sekä viittaukset mahdollisiin ohjaimiin.

Kontrollerikokonaisuus koostuu näkymäkohtaisista kontrollereista, jotka määrittävät näkymille toiminnallisuuden ja joiden kautta näkymät voivat tarvittaessa olla yhteydessä malliin. Myös sovelluksen MainApp-luokka voidaan mieltää ohjaimiin, koska se toimii keskeisessä roolissa ohjainluokat yhteydessä malliin.

Mallikokonaisuus pitää sisällään sovelluksen toimintalogiikan tiedon tallentamisen, ylläpidon ja käsittelyn suhteen. Malli koostuu Data Access Object -luokista, utiliteettiluokista ja "POSEngine" -pääluokasta, joka yhdistää kaikki mallin luokat yhdeksi kokonaisuudeksi ja jonka kautta ohjaimet voivat olla yhteydessä mallin muihin luokkiin.

7.2 Käyttöoikeudet

Emme halua, että järjestelmäämme pääsee käsiksi kuka tahansa sivullinen, sillä ylimääräinen toiminta kassapääätteellä sotkisi tietokantaa turhalla tiedolla. Sovelluksen käyttöoikeudet on jaoteltu neljään tasoon: "ADMIN", "MANAGER", "USER" ja "SELF". "USER"-tason käyttöoikeus on tarkoitettu kassahenkilölle, "MANAGER"-tason käyttöoikeus myymäläpäällikölle, "SELF" itsepalvelukassan oikeustaso ja "ADMIN"-tason käyttöoikeus sovelluksen kehittäjille. Oikeudet eri toimintojen suorittamiseen toteutuu sisäänkirjautumisen perusteella. Eri käyttäjillä voi olla erilaiset oikeudet, joka johtaa eri saatavilla oleviin toimintoihin.

Asiakas on järjestelmää epäsuorasti käyttävä toimija, ellei käytä itsepalveluominaisuutta (SELF). Asiakas haluaa maksaa ostoksensa, ja tämä on asiakkaan ainoa interaktio järjestelmän kanssa. Kanta-asiakas on asiakkaan perivä käyttäjärooli, ja saa alennuksen tuotteista.

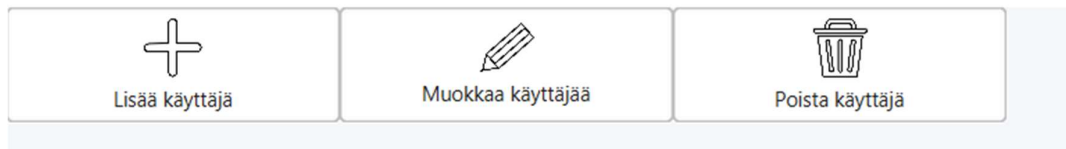
- SELF on itsepalvelukäyttäjä ja voi ainoastaan suorittaa maksutapahtumia.
- USER voi päästä käyttäjien haku näkymään (mutta ei voi hakea käyttäjiä), hakea tuotteita (mutta ei muokata tuotteita) sekä lisätä ja poistaa bonusasiakkuuksia. Näiden lisäksi USER voi suorittaa maksutapahtumia.
- MANAGER hallinnoi järjestelmää ja omaa oikeudet sekä tuotteiden että käyttäjien lisäykselle, poistolle ja muokkaukselle. Tämän lisäksi MANAGER perii myös oikeudet kaikkiin myyjälle mahdollisiin toimintoihin.
- ADMIN on järjestelmän korkein käyttöoikeus, mutta ei varsinaisesti tee nykyisessä versiossa mitään mitä MANAGER ei pystyisi.

7.3 Näkökulmia jatkokehitykseen

Tällä hetkellä kaikkien näkymien ohjaimet sisältävät "setMainApp" -metodin, jota kutsutaan näkymän luonnin yhteydessä. Jatkokehitysideana uusien näkymien toteutuksessa voitaisiin tehdä niin, että jokainen näkymän kontrolleri toteuttaa rajapinnan, joka määrittelee "setMainApp" -metodin, jonka jokaisen näkymän kontrollerin tulee toteuttaa, mikäli yhteys malliin on taattava. Metodien keskeisin

tarkoitus on asettaa kontrollerille viite "MainApp"-olioon, jota kautta kontrolleri voi kutsua mallia.

Sovelluksessa on paljon lähes identtisiä fxml-tiedostoja. Niissä on paljon identtistä koodia, koska niissä käytetään identtisiä suunnitteluratkaisuja (Kuvat 5 ja 6, sivu 12). Tiedostoja tulisi karsia, ja eri näkymien toiminnallisuus toteutettaisiin muuttamalla yhden fxml-tiedoston toiminnallisuutta.



Kuva 5. user-management-view.fxml näkymän valikko.



Kuva 6. products-view.fxml näkymän valikko.

7.3.1 Asynkroninen tietokannan päivitys

Mahdollistaa palautumisen virheistä estämättä sovelluksen toimintaa, kuten tuotteiden lisääminen taustalla kun virheestä on palauduttu. Nykyisessä versiossa tuotteiden lisääminen tulee suorittaa uudelleen virheen sattuessa.

7.3.2 Kuittivaihtoehtoja

Tällä hetkellä kuitit tulostetaan A4-muotoisesti, eikä niiden tulostamista kuittitulostimella ole testattu. Jatkokehitysideana kuittitulostintuki. Kuitti voitaisiin myös lähetettää asiakkaan antamaan sähköpostiin sovelluksen sisältä tulostamisen sijaan.

7.3.3 Bonusasiakkaista enemmän tietoja

Tällä hetkellä bonusasiakkaista ylläpidetään vain bonusasiakasnumero. Relevantteja tietoja voisivat olla osoite, nimi, sähköposti ja puhelinnumero, jotka kyttäisiin asiakkaalta bonusasiakkuuden rekisteröimisen yhteydessä.

7.3.4 Tuotteiden palautus

Tällä hetkellä tuotteita ei voi palauttaa ilman, että tietokantaa muokataan manuaalisesti. Jatkokehitysideaksi tuotteiden palautus esim. asetuksista kuitenkin id:llä.

7.3.5 Mobiililaitetuki

Mobiilisovellus bonusasiakkaille, missä näkyisi esim. tarjouksia ja mitä voisi käyttää bonusasiakkaaksi rekisteröitymiseen. Asiakkaille sovellukseen qr-koodi, jonka lukijalle näyttämällä bonusasiakkuus rekisteröityy tilaukseen.

7.3.6 Mobiiliviivakoodinluku

Yritykset voisivat käyttää vanhaa kamerallista mobiilipuhelinta viivakoodinlukemiseen. Tätä varten pitäisi tehdä mobiilisovellus, mikä välittää viivakoodin datan sovellukselle.

Lähteet

- 1 <https://mariadb.com/kb/en/backup-and-restore-overview/>