

# Loppuraportti NoQ (OTP2-kurssi)

Kimmo Perälä, Samu Rinne, Arttu Seuna, Mikael Tikka

Sovellus on ravintolan tai kahvilan (myöh. ravintola) itsetilauuskioski. Tuote on tarkoitettu ravintoloitsijalle, joka haluaa nopeuttaa tilausjärjestelmää. Sovellus antaa mahdollisuuden ravintolan asiakkaalle tehdä tilauksensa rauhassa omaan tahtiin sekä vapauttaa ravintolan henkilökuntaa palvelemaan muita asiakkaita ja vähentää myös ravintolan jonoja. Käyttäjän vahvistettua tilauksensa tilaus lisätään tietokantaan. Ravintolahenkilökunta näkee tilauksen ja merkitsee tilauksen valmistuttua valmiiksi.

Järjestelmässä on ruokailijan käyttöliittymän lisäksi myös ravintoloitsijan käyttöliittymä, jossa ravintoloitsija voi lisätä uusia tuotteita tai tuotekategorioita, tarkastella tilauksia, muokata jo olemassa olevien tuotteiden tietoja ja poistaa tuotteita ruokalistalta tai tuotekategorioita. Lisäksi järjestelmään kuuluu tilausten tila -näkyvä, joka voisi olla esillä ravintolan näytöillä.

## Ohjelmiston tila

Järjestelmällä ruokailija pystyy selaamaan ravintolan tarjontaa, luomaan tilauksen haluamistaan tuotteista ja muokkaamaan tilauksensa tuotteita. Tilauksen voi valita syötäväksi ravintolassa tai mukaan otettuna. Ruokailija voi vaihtaa käyttöliittymästä kielen suomen ja englannin välillä. Jos ruokailija jättää tilauksen kesken ja järjestelmä pysyy koskemattomana hetken aikaa se varoittaa käyttäjää oletustilaan palautumisesta. Jos toimeton tila jatkuu, järjestelmä tyhjentää ostoskorin ja palautuu oletustilaan.

Ravintolan keittiön/kassan puolella tilaus pystytään vastaanottamaan. Käyttöliittymästä voi nähdä tilauksen sisältävät tuotteet tilauskohtaisesti. Tarkastelun lisäksi ravintoloitsija voi muokata ruokalistaa haluamallaan tavalla poistamalla, lisäämällä tai muokkaamalla tuotteita, kategorioita ja tuotteiden raaka-aineita. Ravintoloitsijan käyttöliittymästä pystyy myös hakemaan vanhoja tilauksia päivämääriä käyttäen.

Järjestelmään on lisätty myös raakaversio tilausten tila -näkyvästä, josta ruokailijat voivat nähdä oman tilausnumeron tilanteen: vasemmalla näkyvät vastaanotetut tilaukset ja oikealla valmiit tilaukset. Samassa näkymässä on muokkausmahdollisuus ravintoloitsijalla, joka voi kuitata tilauksen valmiiksi vastaanotetut-näkyvästä sekä poistaa noudetut tuotteet valmiit-näkyvästä. Kuittaus tapahtuu klikkaamalla tilauksen numeroa.

Demoa varten ohjelmiston rakenne on suunniteltu siten, että asiakkaan käyttöliittymä, ravintoloitsijan käyttöliittymä sekä tilausten tila -näkyvä avautuvat samasta aloitusnäkyvästä.

## Jatkokehitys

Tilausjärjestelmään olisi mahdollista integroida ruokailijoille kohdennettua mainontaa, lisätä tuotteita kampanjahinnoilla, ja parantaa käyttöliittymän käytettävyyttä tekemällä pikavalinta- ja tilauspainikkeet suosituimmille tuotteille. Järjestelmään voisi myös lisätä tuotteiden ainesosien hinnoittelun siten, että tuotteiden räätälöinti ainesosien mukaisesti muuttaisi lopputuotteen hintaa.

Ravintoloitsijan käyttöliittymää voisi kehittää parantamalla vanhojen tilausten hakua erilaisia hakuehtoja käyttäen ja yhdistellen. Hakua voisi parantaa esimerkiksi hakemalla tilausten sisällön perusteella. Käyttöliittymä voisi myös näyttää tilauksista tietoja, joita se ei tällä hetkellä näytä, esimerkiksi tilauksen hinnat ja tilauksen sisällön kustomoinnin. Käyttöliittymää on mahdollista parantaa myös käytettävyyden kannalta.

Tilausjärjestelmään olisi mahdollista myös lisätä ravintolan kassahenkilön tilausnäkyvä, jonka voisi saavuttaa ruokailijan tilauskäyttöliittymästä pienillä muutoksilla.

Toteutetut toiminnallisuudet:

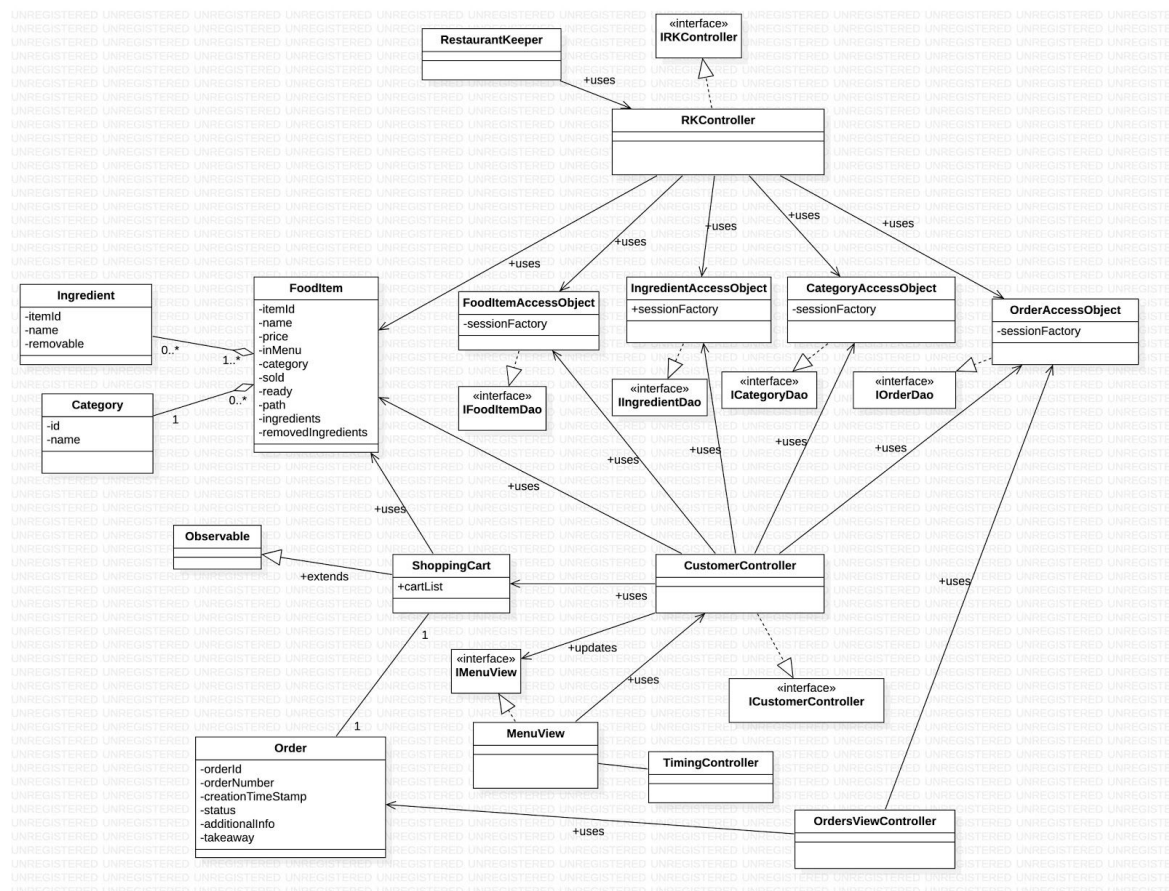
1. Ravintoloitsijana haluan lisätä uusia tuotteita ruokalistalle.
2. Ravintoloitsijana haluan poistaa tuotteita ruokalistalta.
3. Ravintoloitsijana haluan selata ruokalistaa.
4. Ruokailijana haluan selata tuotteita listassa.
5. Ruokailijana haluan lisätä tuotteita ostoskoriin.
6. Ruokailijana/ravintoloitsijana haluan selkeät painikkeet ja kuvat järjestelmään.
7. Ruokailijana haluan vahvistaa tilaukseni.
8. Ravintoloitsijana haluan muokata tuotteiden tietoja ruokalistalla.
9. Ruokailijana haluan lisätä tai vähentää tuotteen määrää ostoskorissa.
10. Ruokailijana haluan poistaa tuotteita ostoskorista.
  - Mahdollista tyhjentää koko ostoskori tai poistaa yksittäinen tuote.
11. Ruokailijana haluan maksaa ostoskorin sisällön ja lähettää sen keittiölle/kassalle.
12. Kassahenkilönä haluan kuitata tilauksen vastaanotetuksi.
13. Ravintoloitsijana haluan nähdä tilaukset keittiön puolella ruudusta.
14. Ruokailijana haluan räätälöidä tilaustani rajatuissa puitteissa.
  - Esimerkiksi ainesosat on poistettavissa tilatusta tuotteesta.
15. Ravintoloitsijana haluan tarkistaa vanhoja tilauksia eri hakuehtoja käyttäen.
16. Ruokailijana haluan nähdä oman tilaukseni statuksen.
17. Ruokailijana haluan valita, syönkö paikan päällä vai otanko tuotteen mukaan.
18. Ruokailijana haluan valita kielen, jolla käytän automaattia.

19. Ravintoloitsijana haluan, että laitteessa on oletustila, johon voi laittaa esimerkiksi ravintolan logon ja tervehdystekstin.
20. Ravintoloitsija/ruokailijana haluan, että laite palautuu oletustilaan, mikäli laite on tietyn ajan käyttämättä.
21. Kassahenkilönä haluan merkitä tilauksen valmiiksi toimitettavaksi.
22. Ruokailijana haluan käyttää selkeää järjestelmää, joka on käyttäjäystävällinen.
23. Ruokailijana haluan selata tuotteita listassa valitsemassani järjestyksessä kategorioittain.

Toteuttamattomat ominaisuudet:

24. Ravintoloitsijana haluan tarjota tuotteita alennetulla kampanjahinnalla.
25. Ravintoloitsijana haluan antaa automaatin käyttäjille lisätietoa esimerkiksi puuttuvista raaka-aineista.
26. Ravintoloitsijana haluan lisätä mainoksia järjestelmään lisämyyntiä kasvattamaan.
27. Kassahenkilönä haluan muokata tilauksia asiakkaan pyynnöstä.

## Ohjelmiston arkkitehtuuri



FoodItem on tuoteolio, jolla on mm. id-numero, nimi, hinta ja kuvan tiedosto-osoite. FoodItem-olioissa on myös tieto kategoriasta (esim. juomat tai hampurilaiset). Niihin sisältyy mahdollisesti myös lista Ingredient-olioita, mikäli tuotteessa on ainesosia (esimerkiksi hampurilainen sisältää pihvin, sämpylän jne). Ostoskori lähetetään ShoppingCart-oliona Order-luokkaan. ShoppingCart sisältää ostoskorin sisällön FoodItem-olioina ja se on myös Observable-olio, jonka muutoksia seuraa CustomerControllerin ShopCartObserver, joka päivittää ostoskorin hinnan MenuView-näkymään. Myös FoodItem on Observable-olio, jossa tapahtuvia ainesosien poistoja seuraa CustomerControllerin foodItemObserver ja päivittää näkymän ostoskorissa olevien tuotteiden poistetut ainesosat -listoja.

Tietokantaan otetaan yhteys AccessObjectien avulla (FoodItemAccessObject, IngredientAccessObject, CategoryAccessObject, OrderAccessObject). Jokainen DAO-olio toteuttaa vastaavan rajapinnan. HibernateUtil-luokassa luodaan hibernaten SessionFactory-instanssi, jota kaikki DAO-oliot käyttävät.

Start-tiedosto kasaa yhteen kaikki View-näkymät: Demo (View-näkymien valintaikkuna), MenuView (asiakkaan käyttöliittymä), RestaurantKeeper (ravintoloitsijan käyttöliittymä) ja OrdesViewController (tilausten tila -näkymä). CustomerController käyttää DAO- ja FoodItem-olioita ja päivittää MenuView'ta. MenuView myös käyttää CustomerControllerin kautta mm. ShoppingCart-oliota. Samoin RestaurantKeeper käyttää DAO-olioita RKControllerin avulla. Kielen vaihto kaikissa näkymissä on toteutettu yhdellä Bundle-luokalla, joka on toteutettu Singletonina.

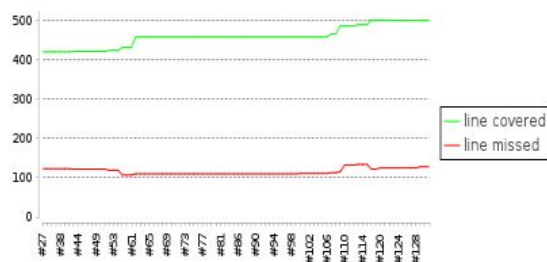
## Testaus

### JaCoCo Coverage Trend

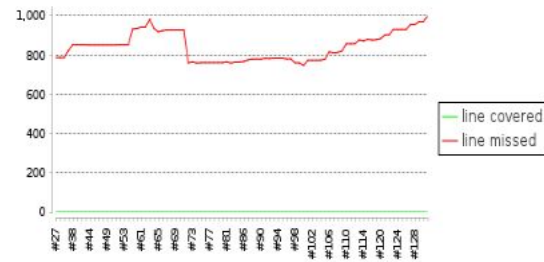


Nopeasti katsottuna sovelluksemme testikattavuus ei ole kovin hyvä, sillä vain noi  $\frac{1}{4}$  koodista on testattua. Tämä johtuu siitä, että emme ole testanneet JUnit testeillä käyttöliittymiä ollenkaan. Olemme kuitenkin pitäneet huolta siitä, että malli-tasolla testit on kunnossa. Siellä käyrä näyttääkin huomattavasti paremmalta, sillä noin  $\frac{5}{6}$  koodista on testattua.

Package: model



Package: view



Yksikkötestien lisäksi testasimme sovellusta manuaalisesti. Useimmiten kun joku ryhmästä päivitti uuden omasta mielestään toimivan ominaisuuden, ainakin yksi ryhmästä kokeili tietoisesti rikkoa sitä, jotta löytäisimme siitä mahdolliset bugit. Tämä käytäntö on mielestämme hyvä, sillä tässä vaiheessa muilla ryhmän jäsenillä ei välttämättä ole tietoa miten kyseinen ominaisuus on toteutettu, joten sen rikkomiseen saattaa löytyä uusia ideoita.

Metodia käytimme esimerkiksi ajastimeen, joka palauttaa asiakkaan näkymän takaisin aloitusikkunaan, sekä ainesosien poistamiseen tuotteesta. Molemmista löysimme joitain bugeja, jotka olisivat muuten ehkä jääneet huomaamatta.