

OMAJAAKAAPPI

OTP1 - Väliraportti

Ryhmä 9

Lauri Uotila, Ville Kerri, Tuukka Rissanen

Sisällys

Sovelluksen visio	1
Sovelluksen toiminnallisuudet.....	1
Toteutettavat ominaisuudet.....	2
Graafinen käyttöliittymä	2
Sovelluksen arkkitehtuuri ja käytetyt työkalut.....	3

Kuvat

Kuva 1, Sovelluksen tietokantarakenne	3
Kuva 2, Sovelluksen luokkakaavio	4
Kuva 3, Instanssisekvenssikaavio: Tuotteiden haku	5
Kuva 4, Instanssisekvenssikaavio: Tuotteen lisäys jääkaappiin	6
Kuva 5, Instanssisekvenssikaavio: Reseptin lisäys	7

Sovelluksen visio

Projektin myötä syntyvälle ohjelmistolle on annettu nimi OmaJääkaappi. Sovelluksen päätarkoituksena on auttaa kuluttajaa seuraamaan jääkaappinsa ruokatilannetta. Käyttäjä kertoo sovellukselle tuotteiden tietoja, joiden perusteella ohjelma auttaa käyttäjää seuraamaan tuotteiden päiväyksiä. Sovellus pyrkii auttamaan käyttäjää ehkäisemään ruokahävikkiä tiedottaen käyttäjää jääkaapin sisällöstä ja vanhentuvista tuotteista. Lisäksi sovellus tarjoaa mahdollisuuden seurata omaa kulutusta käytettyjen kaloreiden ja hävikiksi menneiden tuotteiden avulla.

Ohjelma myös auttaa käyttäjää ruokareseptien käytössä, tarjoamalla mahdollisuuden tallentaa reseptejä, joiden pohjalta ohjelma antaa löytyvien tuotteiden perusteella mahdollisia reseptiehdotuksia. Myös reseptien toiminnot auttavat hävikin ehkäisyssä, sillä sovelluksen reseptejä hyödyntämällä käyttäjä näkee samalla jääkaapista löytyvät tuotteet. Näin ylimääräisiä tuotteita ei tule ostettua ja käyttäjä ohjataan käyttämään jääkaapista löytyviä tuotteita.

Sovelluksen toiminnallisuudet

Tähän mennessä sovellukseen on toteutettu kaikkien osien perustoiminnot, jotka toteuttavat sovellukselle laadittujen käyttäjätarinoiden perustoiminnallisuudet. Näitä käyttäen ja soveltaen sovellukseen pystyy kehittämään monimutkaisempia toimintoja.

Ohjelmaa käytetään tällä hetkellä tekstipohjaisella käyttöliittymällä. Käyttöliittymä toimii valitsemalla osa-alueen vaihtoehtoista: tuotteet, jääkaappi tai reseptit. Seuraavaksi avautuu numeroitu lista valitun aiheen mahdollisista toiminnollisuuksista. Käyttäjä voi valita toiminnon syöttämällä sitä vastaavan numeron, ja suorittaa toiminnan seuraamalla käyttöliittymän ohjeita. Ohjelma voidaan aina lopettaa syöttämällä numero 0. Tekstikäyttöliittymästä on poistettu virhesyötteiden mahdollisuus niin, että esimerkiksi int-määräisiä syötteitä pyydetessä ohjelma ei kaadu, vaikka käyttäjä antaisi string-syötteen. Käyttöliittymässä voi myös numerokomennoilla navigoida eri välilehdiltä takaisin päävalikkoon ilman, että mitään toimintoa on pakko suorittaa.

Tämänhetkiset toiminnallisuudet luokittain:

Tuotteet:

- Kyky lisätä/muokata/poistaa tuotteita ohjelman tietokannasta.
- Kaikkien lisättyjen tuotteiden listaus.
- Hakutoiminnallisuus tuotteen nimen/ID:n perusteella.

Jääkaappi:

- Tuotteiden lisäys/muokkaus/poisto jääkaapista.
- Tuotteiden statusten merkkäminen käytetyksi/hävikiksi
- Lähenevien viimeisten käyttöpäivämäärien listaus.
- Erinäisiä hakuja
 - Käytetyt tuotteet.
 - Hävikki tuotteet.
 - Tuotteen haku nimellä/päivämäärällä/ID:llä.
 - Kaikkien jääkaapissa olevien tuotteiden haku.

Ruokapäiväkirja

- Uuden merkinnän lisäys/muokkaus/poisto ruokapäiväkirjaan.
- Kaikkien haku
- Haku ID:llä/nimellä/jääkaapissa olleen tuotteen ID:llä.

Reseptit:

- Reseptien lisäys/muokkaus/poisto.
- Kaikkien reseptien haku.
- Reseptin haku nimellä/ID:llä.
- Reseptin kalorimäärän lasku siinä käytettävien tuotteiden pohjalta.

Aines:

- Aineiden lisäys/muokkaus/poisto.
- Kaikkien aineiden haku.
- Aineiden haku ID:llä/tuotteen nimellä/reseptin nimellä.

Toteutettavat ominaisuudet

Sovellukseen tullaan lisäämään tietokantaan liittyviä edistyneempiä ominaisuuksia sekä parannuksia. Lisäksi sovellukseen on tarkoitus lisätä aiheeseen liittyviä toiminnollisuuksia. Tärkein tavoite on luoda sovellusta varten graafinen käyttöliittymä, jossa panostetaan käyttäjäkeskeiseen suunnitteluun.

Tuotteet

- Tietokantaan on tarkoitus lisätä laaja lista tuotteita valmiiksi, sovelluksen käyttöönoton helpottamiseksi.

Ruokapäiväkirja

- Ruokapäiväkirjaan on tarkoitus toteuttaa kalorimäärien laskenta päiväkohtaisesti, sekä valitulla aikavälillä. Myös toiminta hävikkiruoan määrän laskenta valitulta aikaväliltä on tarkoitus lisätä.

Resepti

- Resepteihin on tarkoitus lisätä toiminta, joka tarkasta mitä reseptiin merkittyjä aineksia löytyy jääkaapista ja mitä puuttuu. Tämän pohjalta lisätään myös toiminta, joka osaa ehdottaa käyttäjälle reseptejä, joita hän voi valmistaa.

Edellä mainittujen ominaisuuksien lisäksi sovellukseen on tarkoitus lisätä ruoanlaittoon käytettäviä ajastimia. Yleisiä ajastimia löytyisi sovelluksesta valmiina ja käyttäjällä olisi myös mahdollisuus luoda ajastin itse määritetyllä ajalla.

Sovellukseen lisätään myös toiminta luoda ostoslista. Ostoslistaan lisätään sovelluksen tietokannasta löytyviä tuotteita, sekä näiden haluttuja määriä. Tarkoituksena on toteuttaa myös mahdollisuus lähettää ostoslista käyttäjälle sähköpostiin.

Graafinen käyttöliittymä

Käyttöliittymä on tarkoitus luoda käyttäen JavaFX:ää. Käyttöliittymän navigoinnin osalta alustava tarkoitus on toteuttaa toiminnallisuudet välilehdille niin, että välilehdet vastaavat tekstikäyttöliittymän toimintoja (tuotteet, jääkaappi, ja reseptit). Mahdollisesti Osa nyt jääkaappi valikon alta löytyvistä toiminnoista tullaan lisäämään uuteen ruokapäiväkirja nimiseen valikkoon.

Välilehtitoteutuksessa käyttäjän on helppo navigoida sovelluksessa ilman, että toimintoihin täytyy kiertää erillisen etusivun kautta. Neljä välilehteä on myös helposti hallittavissa ja toiminnallisuudet on helppo kategorisoida kyseisten välilehtien alle. Graafinen käyttöliittymä noudattaa samoja toiminnollisuuksia kuin nykyinen käyttöliittymä.

Sovelluksen arkkitehtuuri ja käytetyt työkalut

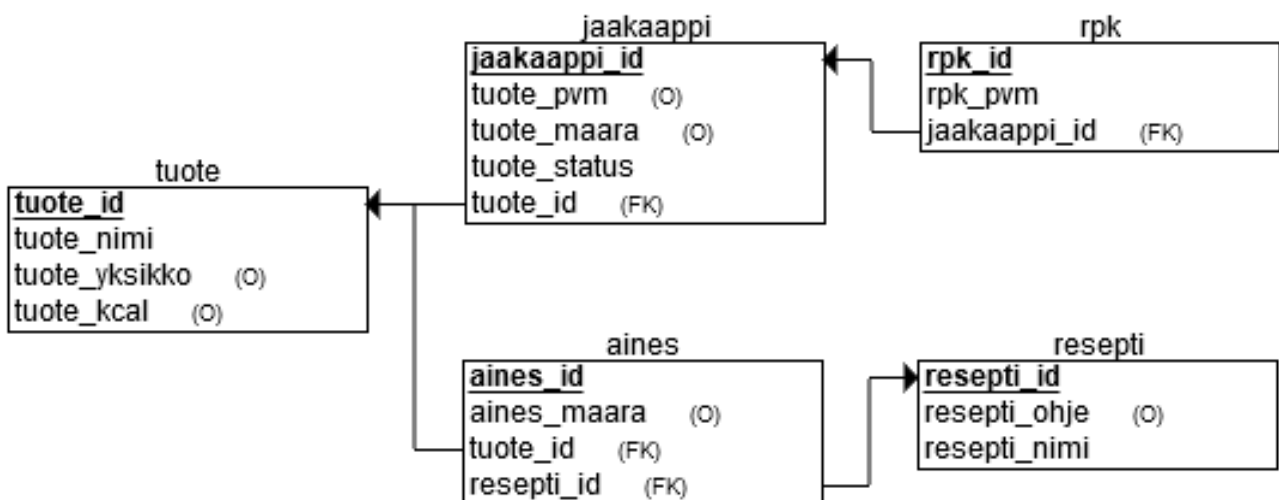
OmaJääkaappi-sovellusta toteutetaan Java-ohjelmointikielellä, josta käytetään Java 8-versiota. Ohjelmaan syötettyjen tietojen tallentamiseen käytetään MariaDB-tietokantaa, sekä Javan Hibernate-kirjastoa, joka toimii ORM-rajapintana ohjelmointikielen ja tietokannan välillä.

Projektin versionhallinnassa hyödynnetään Git-versionhallintatyökalua, sekä GitHub-palvelua etäsäilönä projektin lähdetiedostoille.

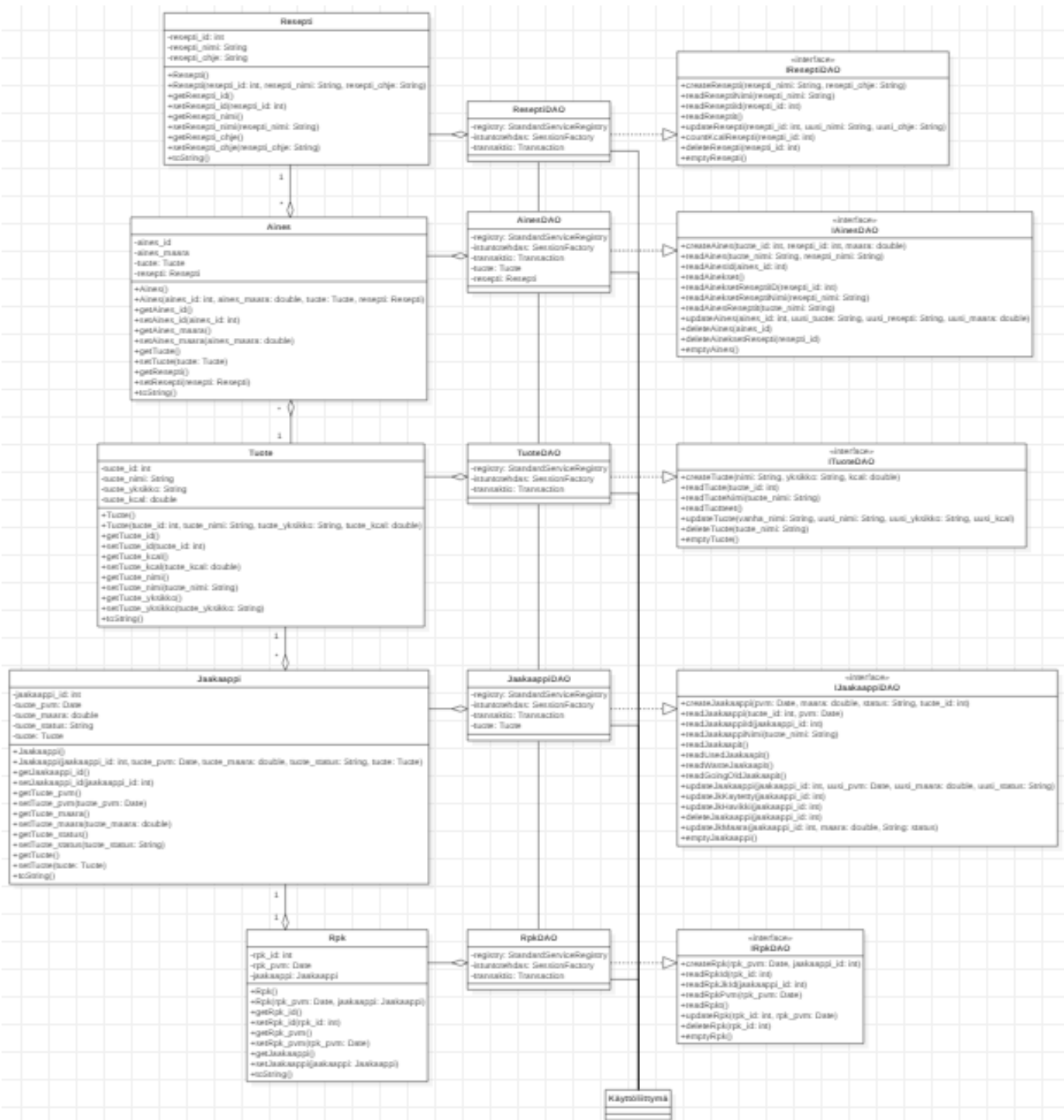
Ohjelman testaamisessa käytetään JUnit 5-kirjastoa, jonka avulla jokaiselle luokalle luodaan omat yksikkötestit ohjelman toimivuuden takaamiseksi. Apuna on myös Metropolian sisäverkossa toimiva palvelin, jolla pöyritetään kaikille projektiin osallistujille yhteistä tietokantaa sekä Jenkins-jatkuvan integraation automaatiopalvelua. Kun uusi versio ohjelmasta pusketaan GitHub-etäsäilöön, Jenkins automaattisesti kokoaa ohjelman ja ilmoittaa epäonnistuneista testeistä.

Projektin koonnissa käytetään Maven-kasaustyökalua, jonka avulla varmistetaan, että jokaisella projektiin osallistujalla on sama versio jokaisesta käytetystä kirjastosta.

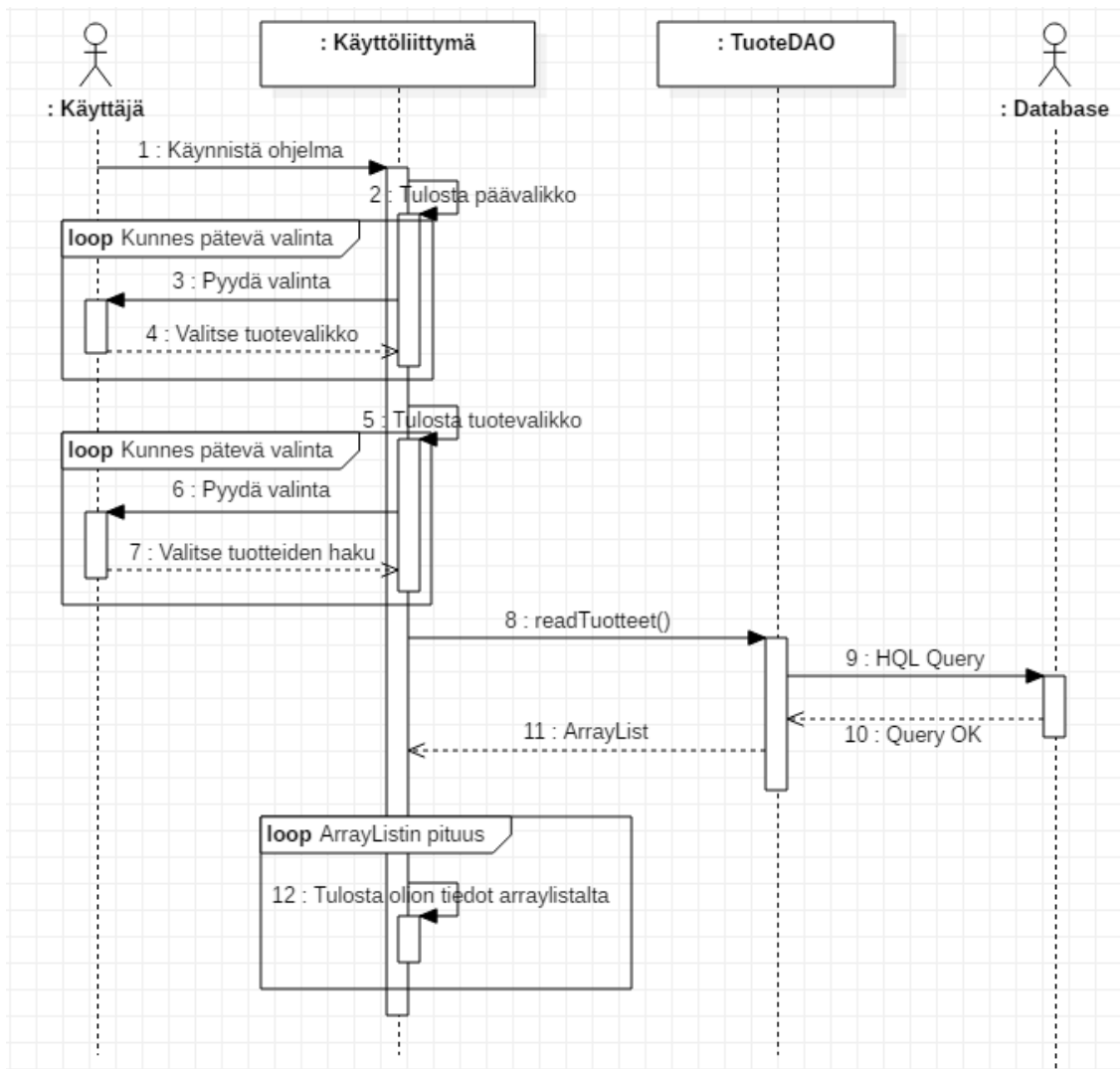
Seuraavassa on esitetty sovelluksen tietokannan rakenne sekä luokkakaavio pohjautuen sovelluksen nykyiseen rakenteeseen.



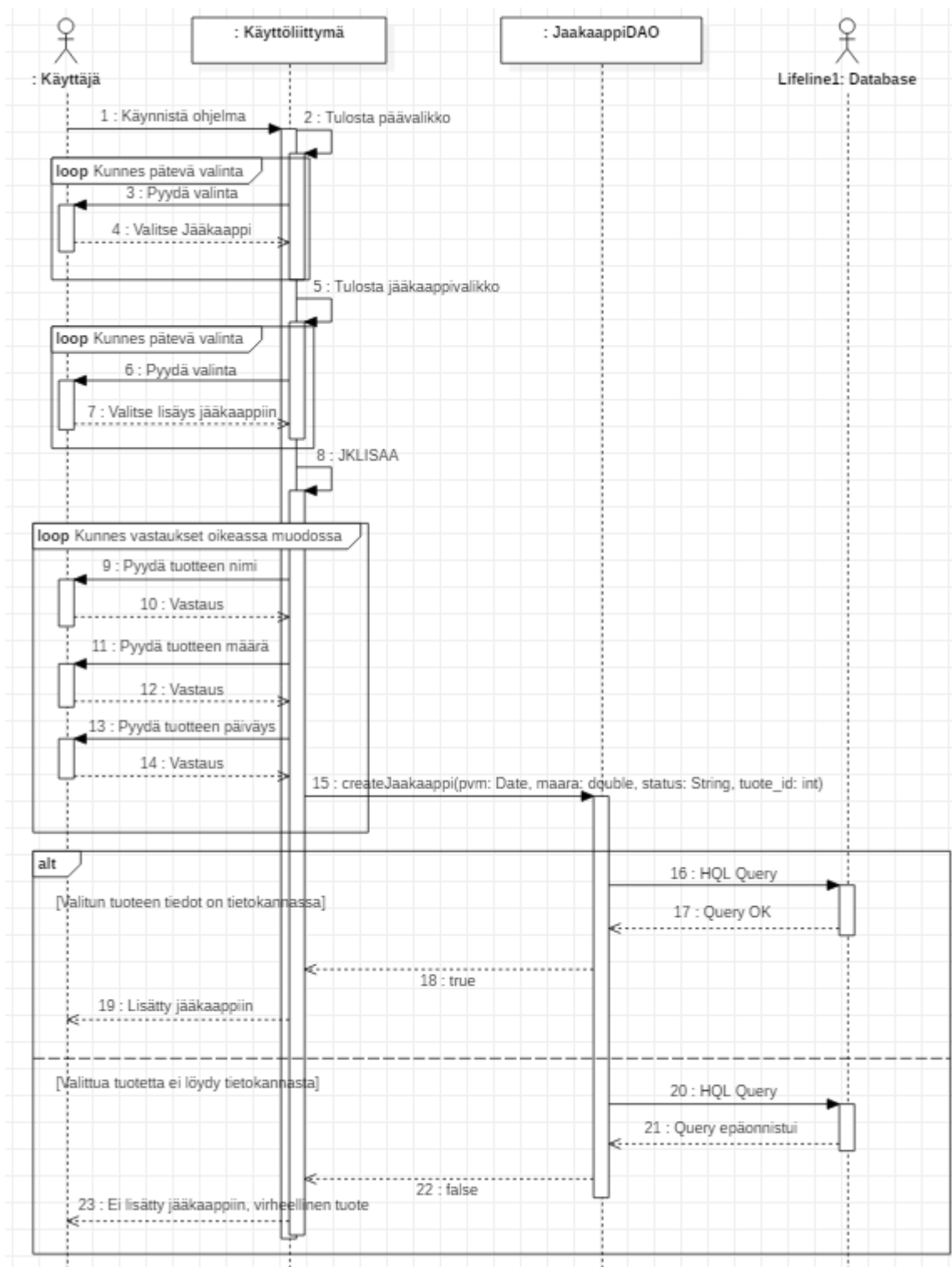
Kuva 1, Sovelluksen tietokantarakenne



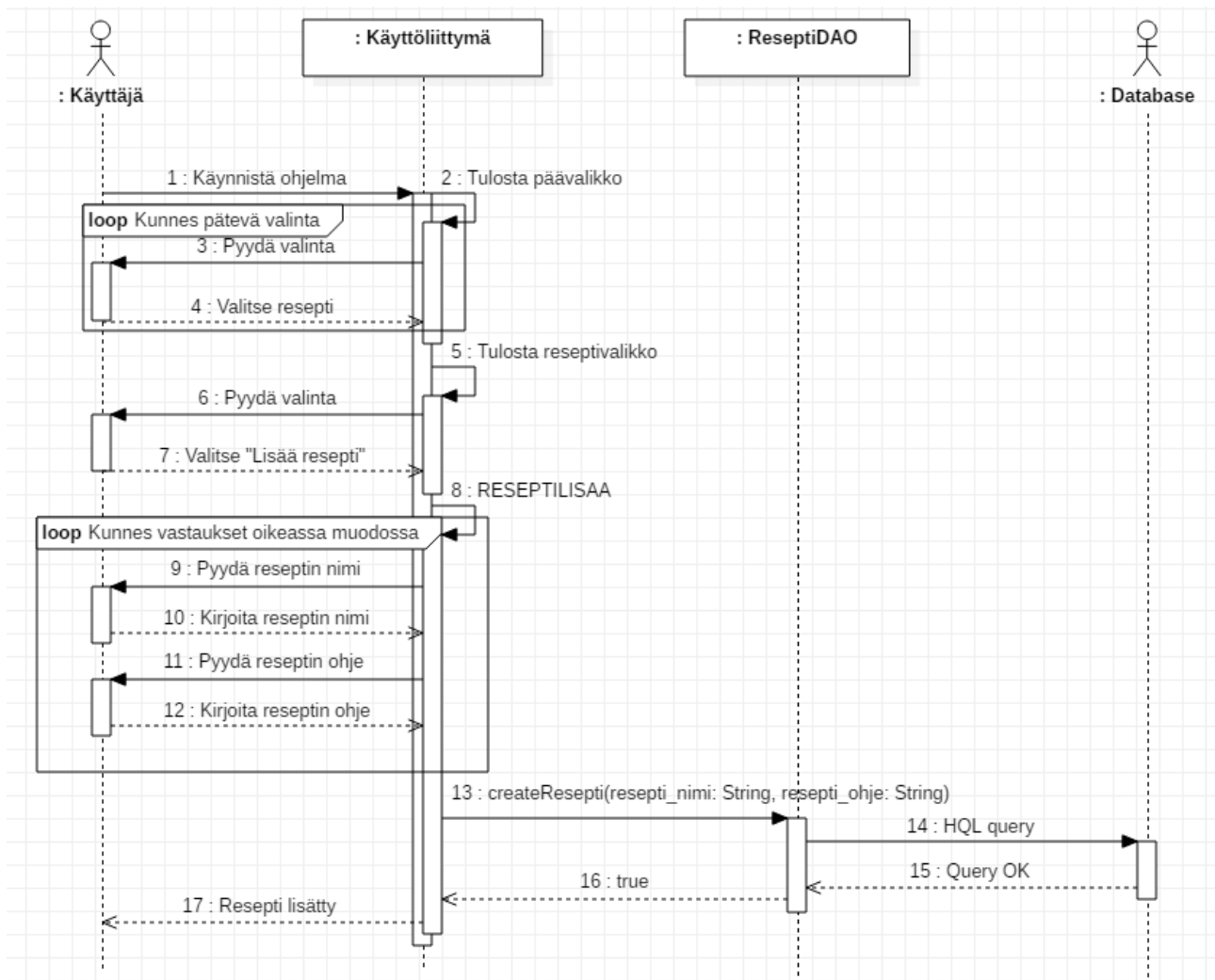
Kuva 2, Sovelluksen luokkakaavio



Kuva 3, Instanssisekvenssikaavio: Tuotteiden haku



Kuva 4, Instanssisekvenssikaavio: Tuotteen lisäys jääkaappiin



Kuva 5, Instanssisekvenssikaavio: Reseptin lisäys