

Johdanto:

Ohjelman tavoitteena on tarjota SiipiLomat Oy:n lentomatkustajille palvelua, jossa matkustajat voivat tilata tax free –ostoksia ennakoon. Ostokset toimitetaan paikalle, jossa käyttäjä istuu lentokoneessa. Palvelussa käyttäjä valitsee vain lennon jolla on menossa. Käyttöliittymän kautta voi tilata Aterian lennolle. Tuotteita voidaan lisätä admin-käyttäjätunnuksien avulla lisää järjestelmään ja niitä voidaan poistaa. Tuotteita voidaan myös ottaa pois myynnistä väliaikaisesti. Järjestelmän avulla voidaan myös päivittää tarjolla olevia tuotteita ja niitä voidaan ylläpitää. Lentojen tiedot saadaan toisesta järjestelmästä.

Ohjelman toteutusympäristö on Java ja Spring-sovelluskehys. Spring-tarjoaa sovelluksen ”back-end” osuuden eli se hoitaa yhteyden tietokantaan ja tarjoaa REST-rajapinnan jonka avulla ”Front-end” sovellus voi käyttää haluttuja toimintoja. Front end sovellus toteutaan Javascriptin, JQueryn ja AngularJS:n avulla. Selaimen tulee tukea JavaScript-kieltä. Ohjelmisto tulee käyttämään vain PostgreSQL-tietokantaa.

Käyttötapaukset:

Ohjelmisto käyttötapauksia ovat mm:

- Kirjautuminen
 - Asiakas tai ylläpitäjä kirjautuu järjestelmään.
- Tilauksen laatiminen
 - Asiakas laatii tilauksen
- Tilauksen muuttaminen
 - Asiakas muuttaa tilausta
- Tilauksen peruutus
 - Asiakas peruu tilauksen
- Ylläpitäjä voi tarkastella kaikkia tilauksia
 - Ylläpitäjä voi asettaa tilauksen toimitetuksi tai poistaa sen
- Tuotetietojen lisäys, muokkaus ja poisto
 - Ylläpitäjä voi muokata tuotteita
- Ateria tietojen lisäys, muokkaus ja poisto
 - Ylläpitäjä voi muokata Aterioita

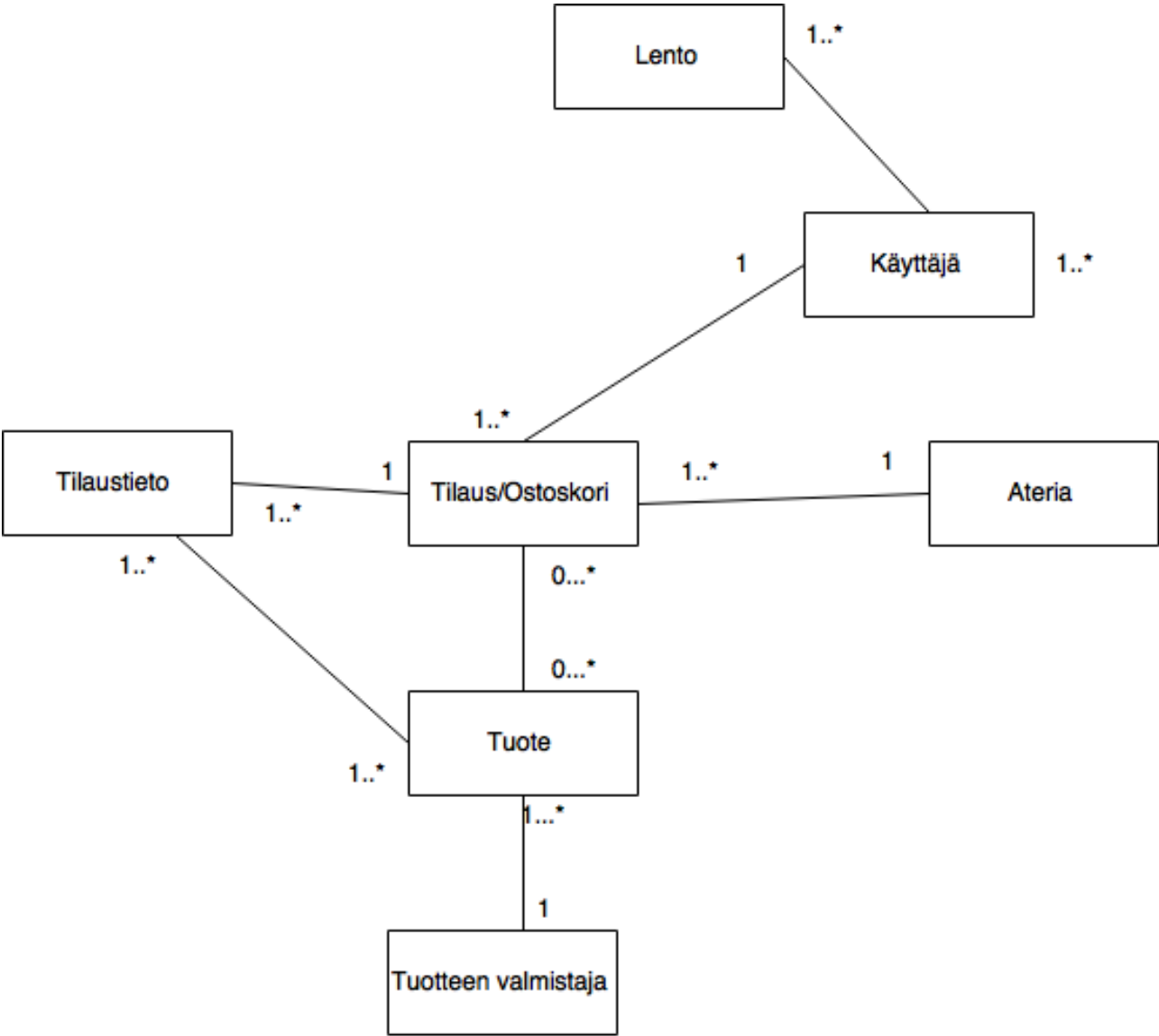
Käyttäjryhmät:

Asiakas: Asiakkaalla tarkoitetaan käyttäjää joka käyttää järjestelmää. Kirjautuu järjestelmään, valitsee tuotteet ja luo tilauksen. Voi halutessaan muuttaa tai peruuttaa tilausta. Käyttäjä voi tarkastella kaikkia tekemiään tilauksia

Ylläpitäjä: Ylläpitäjä pitää järjestelmässä olevia tuotteita ajankohtaisena. Muokkaa, poistaa tai lisää uusia tuotteita. Voi peruuttaa myös asiakkaan tilauksia.

Todennäköisesti myös muut yhtiön työntekijät voivat käyttää järjestelmää, mm täyttäessään tilauksia.

Järjestelmän tietosisältö:



Käyttäjä

Attribuutti	Arvojoukko	Kuvailu
email	Merkkijono, max 50 merkkiä	Asiakkaan sähköposti
admin	boolean	Onko asiakas admin

Tavara

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max 50 merkkiä	Tuote jonka asiakas voi ostaa
Tuotteen valmistaja	Kokonaisluku	Valmistaja joka valmistaa tuotteen
Hinta	Desimaaliluku	Tuotteen hinta
Saatavilla	Boolean	Onko saatavilla
Kuvaus	Merkkijono, max 400 merkkiä	Tuotteen kuvaus

Julkaisu	Date	Milloin ilmestynyt valikoimaan
----------	------	--------------------------------

Ateria

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max 50 merkkiä	Tuote jonka asiakas voi ostaa
Hinta	Desimaaliluku	Tuotteen hinta
Saatavilla	Boolean	Onko saatavilla
Kuvaus	Merkkijono, max 400 merkkiä	Kertomus ateriasta
Julkaisu	Date	Milloin ilmestynyt valikoimaan

Tavaranylvalmistaja

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max 50 merkkiä	Tuote jonka asiakas voi ostaa
Maa	Merkkijono, max 50 merkkiä	Tuotteen valmistajan kotimaa
Lisätty (added)	Date	Milloin lisätty järjestelmään

Tilaus

Attribuutti	Arvojoukko	Kuvailu
Tilaaja_ID	Kokonaisluku	Asiakkaan ID
Toimitettu	Boolean	Onko tilaus toimitettu
Ateria_ID	Kokonaisluku	Tilatun aterian ID
Yhteishinta	Desimaaliluku	Tilauksen kokonaishinta
Lento_ID	Kokonaisluku	Lennon ID
Lisätty (added)	Date	Milloin lisätty järjestelmään

Lento

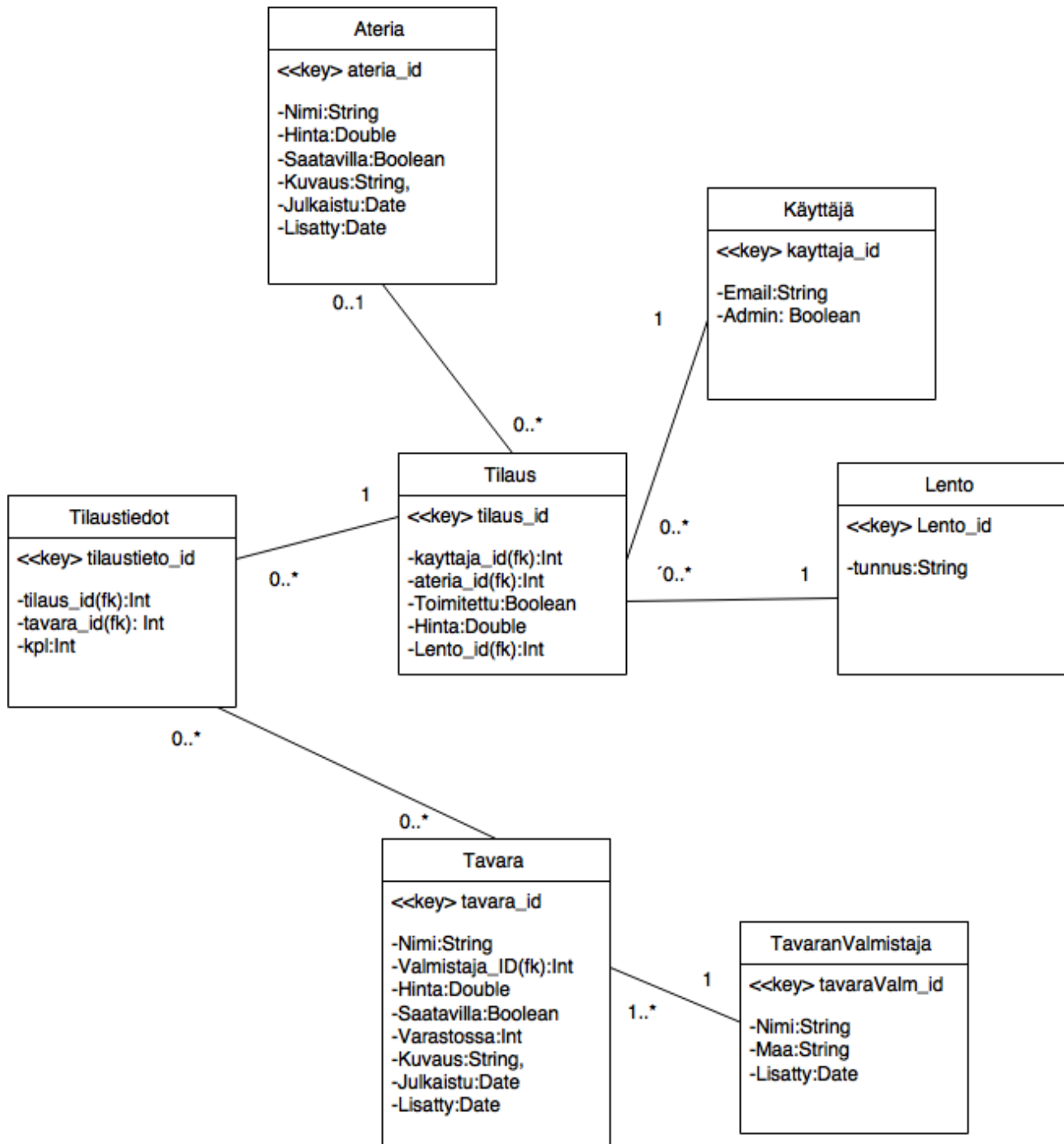
Attribuutti	Arvojoukko	Kuvailu
Tunnus	Merkkijono, max 50 merkkiä	Lennon tunnus (esim. SIPI-111)

Tilaustiedot

Attribuutti	Arvojoukko	Kuvailu
Tilaus_ID	Kokonaisluku	Tilauksen ID
Tavara_ID	Kokonaisluku	Tavaran ID
kpl	Kokonaisluku	Monta kappaletta tilattu

Tietokantakaavio:

Käyttäjä voi tehdä tilauksia, joihin liittyy Ateria, Lento ja Tavaraita. Monesta moneen yhteys löytyy Tilauksen ja Tavarain väliltä. Tilautstiedot toimii näiden välitauluna. Jokaisessa Tilautstiedossa on tilauksen id, tavarain id ja montako kappaletta tavaraa on tilattu. Tavarain liittyy myös tavarain valmistaja.



Asennus ja käyttöohje:

Sovellus voidaan helpoiten asentaa Heroku-Palveluun. Helpoiten tämä onnistuu cloonaamalla projekti Git-repositorystä ja luomalla sille Herokun ohjeiden perusteella oma sovellusosoite. Heroku sovellukseen tulee myös aktivoida tietokanta käyttöön. Sovelluksen käyttäminen omalla koneella onnistuu myös kunhan muuttaa ohjelmasta löytyvien tietokantayhteyksien asetuksia vastaamaan oman koneensa asetuksia. Projektin tarvitsema riippuvuudet voi ladata mavenin clean install:lla, jonka jälkeen sen voi käynnistää vaikka IDE:stä tai mvn spring-boot:run avulla. Tämä voi vaatia kuitenkin pientä säätöä.

Sovelluksen käyttäminen on helppoa, kaikki hoituu käyttöliittymän avulla.

Yleisrakenne

Backend sovellusta (Spring) tehdessä on käytännössä noudatettu vain Controller-mallia. Eli Spring sovelluksetta löytyy vain Controllerit, joiden vastuulla on tarjota REST-pohjaiset palvelut halutuille "Malleille". Varsinaisia Malleja ei kuitenkaan löydy Spring sovelluksesta, vaan ne tarjotaan Angular Front End sovelluksessa. Controllerit hoitavat tietokanta toiminnot JDBC-Moduulin avulla. Eli tietokantaan liittyvä toiminnallisuus löytyy Spring-sovelluksen Controllereista = luokannimiController eli esim. tavaraController.

Frontend sovellus on toteutettuna Javascriptin, AngularJS ja JQuery-lisäosien avulla, joka noudattaa Model, View ja Controller periaatteita. Angularin controllerit tarjoavat BackEnd-sovelluksen tarjoamat REST-pohjaiset palvelut, joita View-luokkien avulla voidaan kutsua (eli Angular kutsuu back-end sovelluksen REST-palveluita).

Sovelluksen rakenne on varsin selkeä: Controllerit löytyy js-kansion alta Controller-kansiosta nimiController-periaatteella. Views-kansiosta löytyvät kaikki HTML-tiedostot. Monikossa oleva (tavarat) tarjoaa listaus ja uuden asian lisäämisen kun yksikössä oleva (tavara) muokkauksen ja yhden "esiintymän" tarkastelemisen. Controllereissa taas esim TavaraController tarjoaa metodit listaukseen ja lisäämiseen, kun TavarainController tarjoaa kaiken yksittäiseen tavarahan ja sen päivittämiseen tarvittavat metodit.

JS-kansion alta löytyvästä libs-hakemistosta löytyvät kaikki sovelluksen käyttämät javascript liitännäiset: jquery, angular, angular-route ja firebase. Firebasea sovellus käyttää käyttäjien hallintaan: kirjautumiseen ja rekisteröintiin. Käyttäjän kirjautuessa sovellus tarkistaa löytyykö käyttäjä firebaseen tietokannasta, jos löytyy niin käyttäjä kirjataan sisään jos salasana on oikein. Jos käyttäjää ei löydy, lisätään se firebaseen tietokantaan sekä sovelluksen omaan tietokantaan. Sovelluksen omassa tietokannassa on tallessa turvallisuussyistä vain sähköpostit ja tieto siitä onko käyttäjä admin. CSS-tiedostot (bootstrap) on hakemiston polussa johtuen ongelmista sen löytymisen kanssa.

Käyttöliittymä ja järjestelmän komponentit

Alla käyttöliittymän ja järjestelmän tärkeimmät komponentit yleisellä tasolla. Aloitussivulta käyttäjä pääsee tarkastelemaan menubar:n kautta haluamiensa kohteita (kuten Tavaraita tai Aterioita). Alla olevassa kaaviossa nimi kuvaa siis jonkin kohteen nimeä: eli esimerkiksi tavaraController. Ilman kirjautumista käyttäjä voi tarkastella kaikkien kohteiden listausta ja tarkastella kohteiden yksittäisiä kohteita. Käyttäjä voi myös rekisteröityä palveluun, jonka jälkeen kirjautua järjestelmään. Kirjaututtuaan käyttäjä voi tilata

tavaroita ja valita aterian haluamalleen lennolleen. Kirjautumisen jälkeen Admin käyttäjä voi poistaa, lisätä ja muokata kohteita. Eli Adminille ja tavalliselle käyttäjälle on erilliset tunnukset.

Tehtyään tilauksen käyttäjä voi tilaukset sivulta tarkastella tekemiään tilauksia, peruuttaa sen tai vähentää tai lisätä tilaamiaan tuotteita. Aterialla käyttäjä ei pysty vaihtamaan. Tilauksen voi peruuttaa vain jos sitä ei ole asetettu toimitetuksi. Admin käyttäjä voi tarkastella kaikkien tekemiä tilauksia ja asettaa ne toimitetuksi.

Admin käyttäjä voi lisätä tuotteita tietokantaan, muokata ja poistaa niitä. Jotta tilaukset pysyisivät ehjänä, järjestelmästä ei voida poistaa tuotteita tai aterioita jotka ovat tilattuja jossain tilauksessa. Tämä on estetty asettamalla nappi "disabled"-tilaan.

