Johdanto:

Ohjelman tavoitteena on tarjota SiipiLomat Oy:n lentomatkustajille palvelua, jossa matkustajat voivat tilata tax free –ostoksia ennakkoon. Ostokset toimitetaan paikalle, jossa käyttäjä istuu lentokoneessa. Käyttöliittymän kautta voi myös esittää toiveita mm. ruokailusta lentokoneessa. Tuotteet jakautuvat ohjelmistossa muutamaan eri tuoteryhmään ja tuotteista näytetään jotakin materiaalia käyttäjälle. Järjestelmän avulla voidaan myös päivittää tarjolla olevia tuotteita ja niitä voidaan ylläpitää. Paikkavaraustiedot saadaan toisesta paikanvarausjärjestelmä.

Ohjelman toimintaympäristö on Apache TomCat 7hjelma tuotetaan Java EE:lla, hyödyntäen todennäköisesti Spring-kehystä. 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 tilaukse
- Lentokohtaiset tilausraportit toimitusta varten
  - Ylläpitäjä saa tiedot mitä toimittaa tietyille lennoille
- Toimitusasiakirja ja lasku
  - Asiakkaalle muodostetaan lasku ja tiedot toimitetusta tavarasta
- Tuotetietojen lisäys, muokkaus ja poisto
  - Ylläpitäjä voi muokata tuotteita

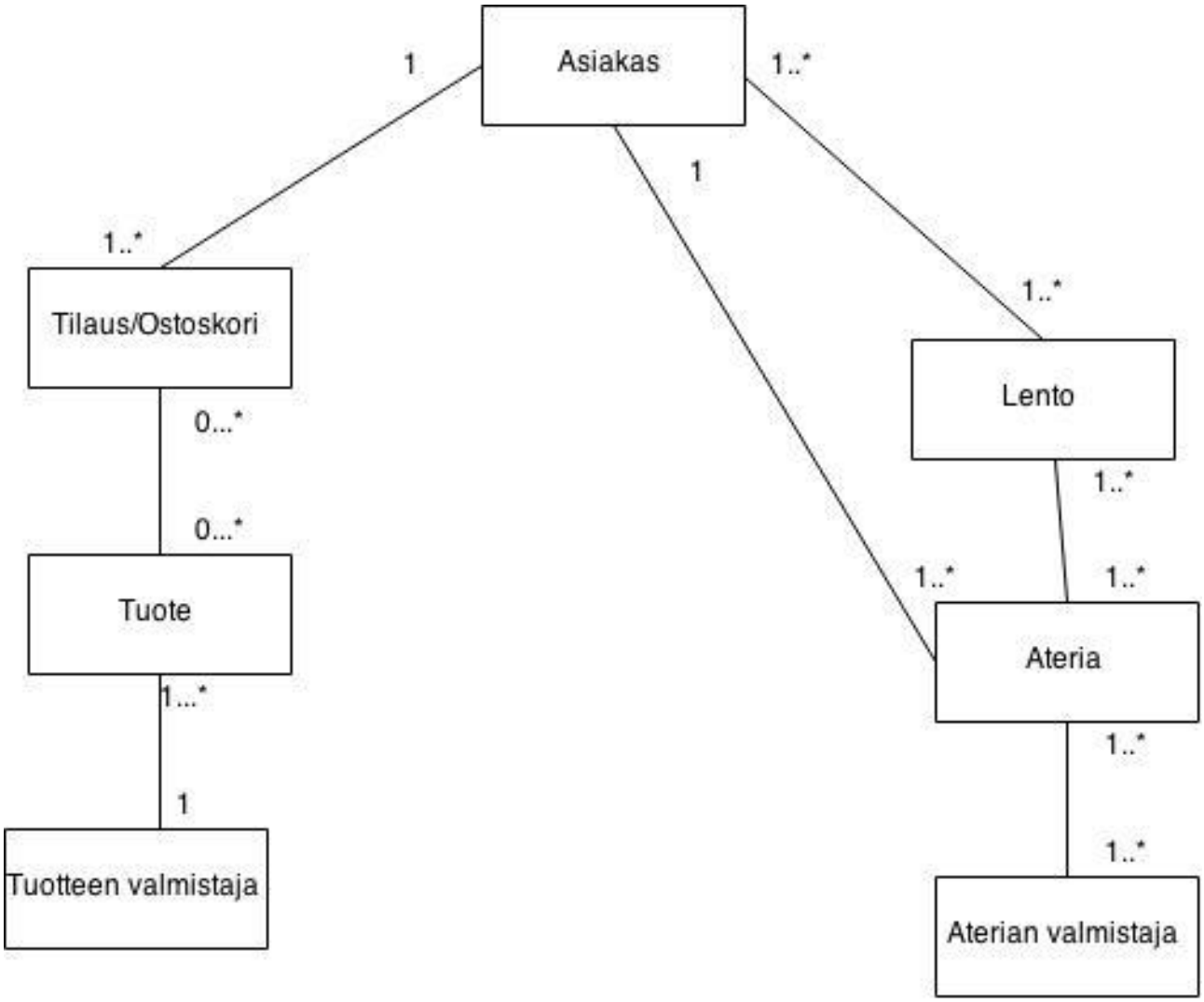
Käyttäjäryhmät:

Asiakas: Asiakkaalla tarkoitetaan käyttäjää joka käyttää järjestelmää. Kirjautuu järjestelmään, valitsee tuotteet ja luo tilauksen. Voi halutessaan muuttaa tai peruuttaa tilausta. Käyttäjälle muodostetaan myös lasku ja toimitustiedot.

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:



Asiakas

Attribuutti	Arvojoukko	Kuvailu
kayttajanimi	Merkkijono, max 50 merkkiä	Asiakkaan tunnus
Salasana	Merkkijono, max 120 merkkiä	Asiakkaan salasana
admin	boolean	Onko asiakas admin
Maa	Merkkijono, max 25 merkkiä	Asiakkaan valtio

Tuote

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
Tuotteen valmistaja	Kokonaisluku	Valmistaja joka valmistaa tuotteen
Hinta	Desimaaliluku	Tuotteen hinta
Saatavilla	Boolean	Onko saatavilla
Kuvaus	Merkkijono, max 400 merkkiä	Kertomus ateriasta
Julkaisu	Date	Milloin ilmestynyt valikoimaan

Aterian Valmistaja

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max 50 merkkiä	Tuote jonka asiakas voi ostaa
Maa	Merkkijono, max 50 merkkiä	Tuotteen valmistajan kotimaa

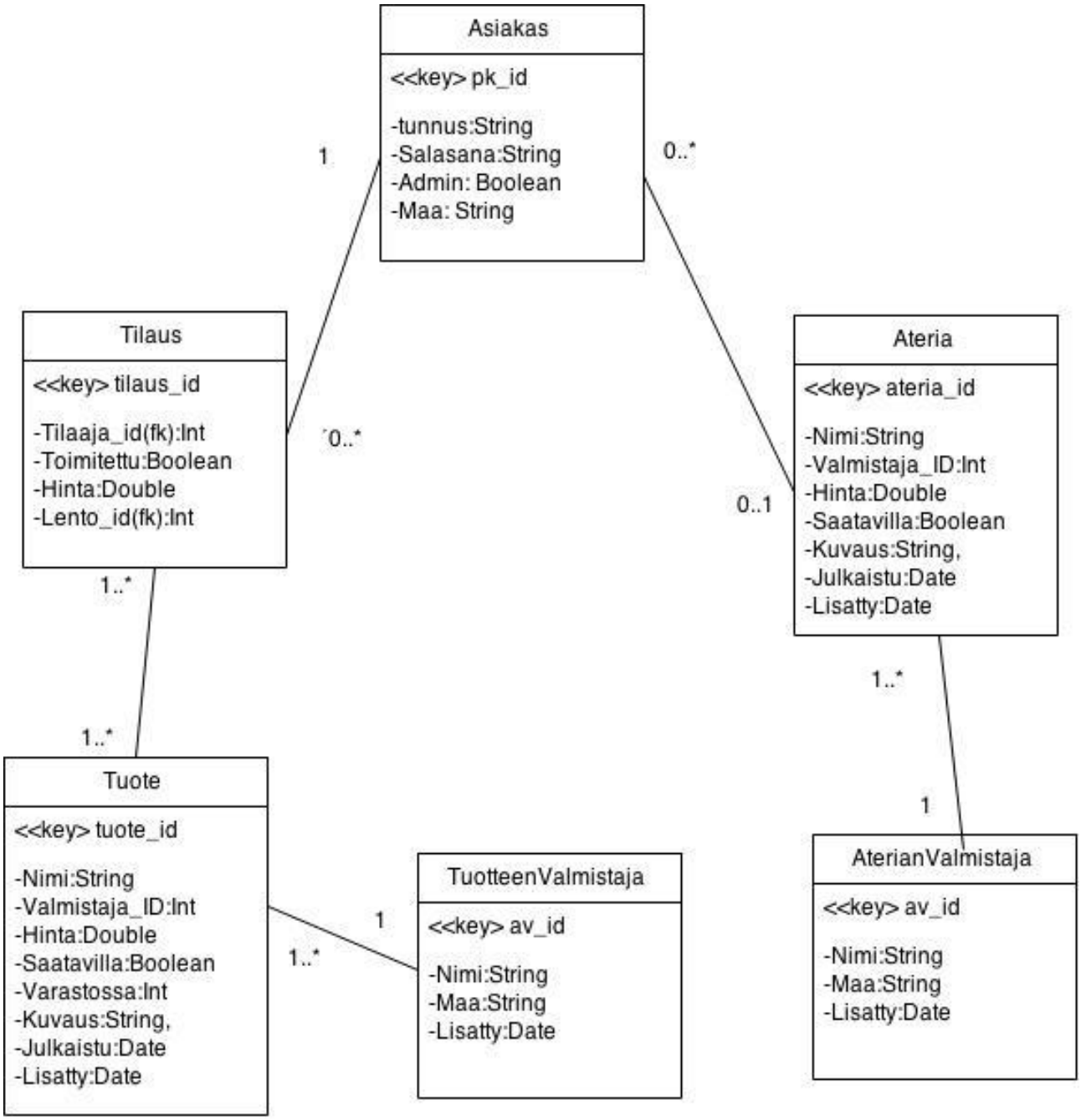
Tuotteen valmistaja

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max 50 merkkiä	Tuote jonka asiakas voi ostaa
Maa	Merkkijono, max 50 merkkiä	Tuotteen valmistajan kotimaa

Tilaus

Attribuutti	Arvojoukko	Kuvailu
Asiakas_ID	Kokonaisluku	Asiakkaan ID
Toimitettu	Boolean	Onko tilaus toimitettu
Hinta	Desimaaliluku	Tuotteen hinta
Lento_ID	Kokonaisluku	Lennon ID

Tietokantakaavio:



Yleisrakenne

Backend sovellusta (Spring) tehdessä on käytännössä noudatettu vain Controller-mallia. Eli Spring sovellukseta 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 toiminnalisuus löytyy Spring-sovelluksen Controllereista = luokannimiController eli esim. tavaraController.

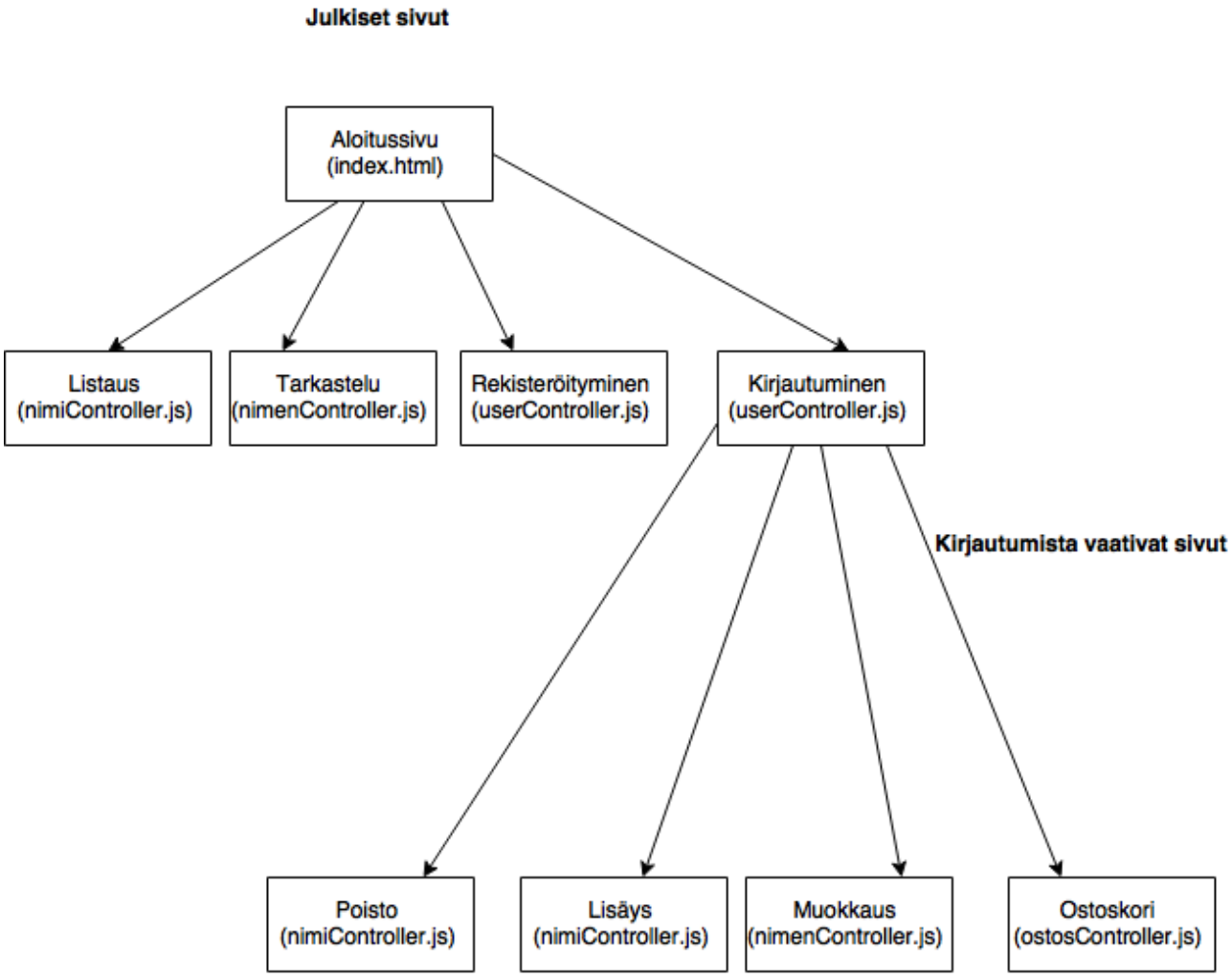
Frontend sovellus on toteutettuna Javascriptin, AngularJS ja JQuery-lisäosien avulla, joka noudattaa Model, View ja Controller periaatteita. Varsinaista Model-luokkaa ei kuitenkaan löydy. Controllerit tarjoavat BackEnd-sovelluksen tarjoamat REST-pohjaiset palvelut, joita View-luokkien avulla voidaan kutsua.

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 TavaranController tarjoaa kaiken yksittäiseen tavaraan ja sen päivittämiseen tarvittavat metodit.

JS-kansion alta löytyvätästä 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 firebasen tietokannasta, jos löytyy niin käyttäjä kirjataan sisään jos salasana on oikein. 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 haluamiansa kohteita (kuten Tavaroita 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. Kirjautumisen jälkeen käyttäjä voi poistaa, lisätä ja muokata kohteita. Sovellukseen tulee vielä toteuttaa jonkinlainen admin käyttäjä. Tällä hetkellä kaikki voivat muokata, lisätä ja poistaa kohteita.


Julkiset sivut
Aloitussivu
(index.html)
Listaus
(nimiController.js)
Tarkastelu
(nimenController.js)
Rekisteröityminen
(userController.js)
Kirjautuminen
(userController.js)
Kirjautumista vaativat sivut
Poisto
(nimiController.js)
Lisäys
(nimiController.js)
Muokkaus
(nimenController.js)
Ostoskori
(ostosController.js)