

Ajovuorotietokanta Taksiyrityksille

Hossein Bahmanpour

Table of Contents

1. Johdanto.....	3
2. Yleiskuva järjestelmästä.....	4
2.1 Käyttötapaukset.....	4
2.2 Käyttäjäryhmät.....	4
2.3 Käyttötapauskuvaus.....	4
2.4 Käyttöliittymä.....	5
3. Järjestelmän tietosisältö.....	6
3.1 Tietosisältökaavio.....	6
3.3 Tietokohteet.....	6
3.2 Relaatiotietokantakaavio.....	7
4. Ohjeet.....	8
4.1 Asennus.....	8
4.2 Käyttö.....	8
5. Ohjelmistokehitys.....	9
5.1 Testaus.....	9
5.2 Tunnetut bugit ja puutteet.....	9
5.3 Jatkokehitys.....	9
5.4 Omat kokemukset.....	9
6. Muu.....	10
6.1 Sql taulujen luonti.....	10

1. Johdanto

Ajovuoro tietokantasovelluksen tarkoitus on olla kätevä tietokanta suomalaisille taksiyrityksille, joilla on useampia autoja – ja kuljettajia. Tietokantaan tallennetaan kaikki yrityksen kuljettajat, autot, ajetut työvuorot, ja jokaisen työvuoron kaikki kyydit.

Tietokannan avulla yritys voi seurata yleistä taloudellista tilannetta, esimerkiksi vuoden kassoja, kuukauden kassoja, tai päivän kassoja, tai yksittäisten kuljettajien suorituksia valitsemaltaan ajanjaksolta – kaikki kätevästi sähköisessä muodossa.

Tietokantaa voi käyttää netissä. Tämä sovellus on esimerkiksi ladattu Helsingin Yliopiston users-palvelimelle Tomcat palvelimen alla.

Sovelluksen pohjana on Java ja SQL kielet, sekä HTML. Sovelluksen käyttämiseen ei tarvitse muuta kuin nettiselaimen. Ohjelmisto käyttää PostgreSQL-tietokantaa, ja sen vaihtaminen muulle alustalle vaatii työtä.

2. Yleiskuva järjestelmästä

2.1 Käyttötapaukset



2.2 Käyttäjärühmät

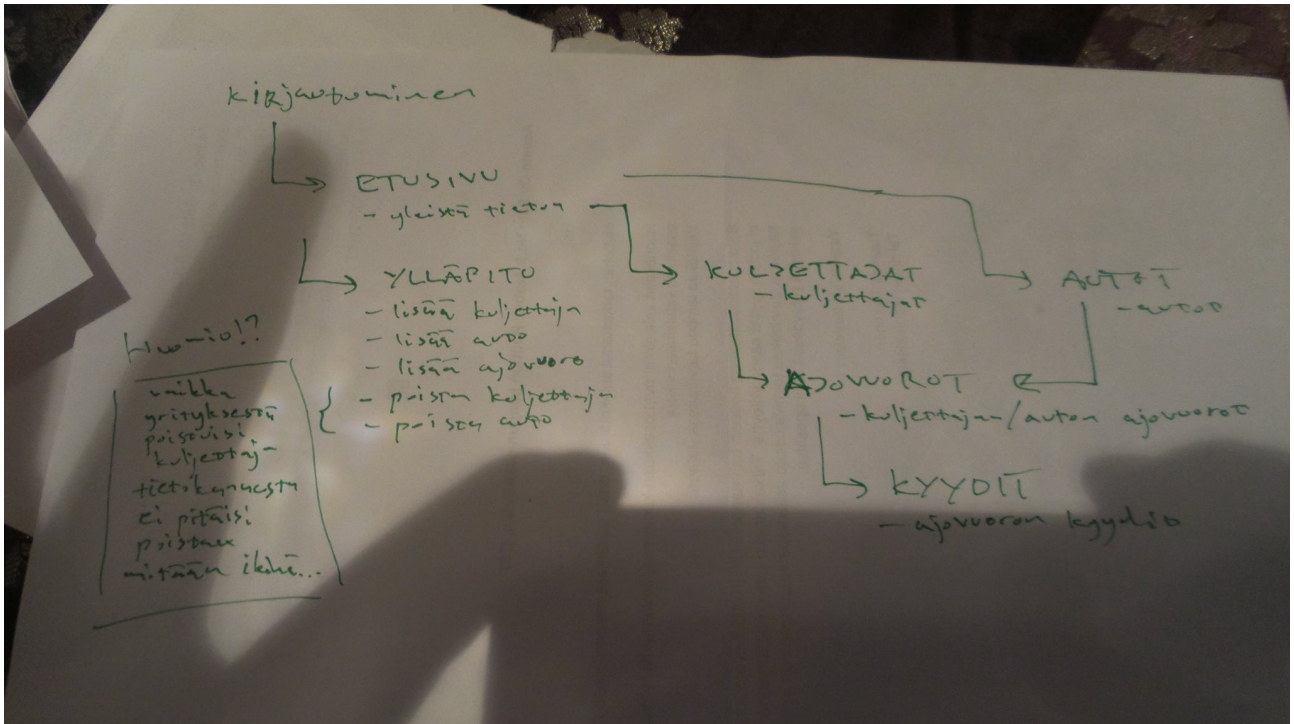
Tietokannalle on suunniteltu toistaiseksi vain yksi käyttäjäryhmä: yrittäjä, joka on sovelluksen ostanut taksi yritykselleen.

2.3 Käyttötapauskuvaus

Käyttäjä / yrittäjä:

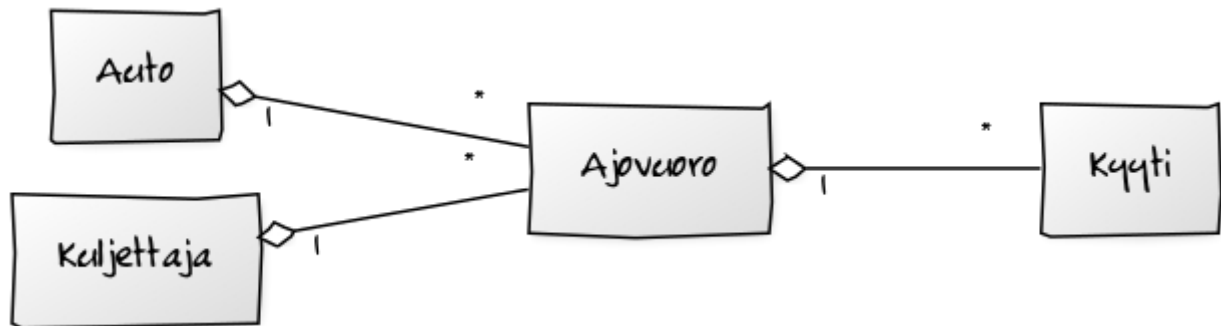
Käyttäjä kirjautuu järjestelmään. Käyttäjä voi luoda uusia ajovuoroja, minkä yhteydessä hän voi luoda uusia autoja ja kuljettajia. Käyttäjä voi luoda uusia kyytejä ajovuoroille. Käyttäjä voi myös selata autoja tai kuljettajia. Selatessa näkyy kyseisen auton tai kuljettajan ajovuorot uusimmasta vanhimpaan, auton tapauksessa näytetään kuljettajat ja heidän ajamat ajovuorot, kuljettajan tapauksessa autot ja auton ajovuorot. Ajovuoroa tarkasteltaessa voi nähdä kuljettajan ajovuoron ajaneen kuljettajan tietoja, ajovuoroon käytetyn auton tietoja, sekä kaikki kyseisen vuoron kyydit. Kyytiä tarkasteltaessa voi nähdä kyydin tiedot. Mahdollisesti käyttäjä voi myös nähdä montako kilometriä, ja montako euroa jokin kuljettaja tai auto on ajanut valitsemallaan ajanjaksolla.

2.4 Käyttöliittymä



3. Järjestelmän tietosisältö

3.1 Tietosisältökaavio



3.3 Tietokohteet

Tietokohde: Auto

Attribuutti	Arvojoukko	Kuvailu
id	serial	id
rekkari	merkkijono, max 22 kirjainta	Auton rekisterikilven sisältö
asemapaikka	merkkijono, max 22 kirjainta	Taksin asemapaikka
merkki	merkkijono, max 22 kirjainta	Auton merkki
malli	merkkijono, max 22 kirjainta	Automerkin malli

Tietokohde: Kuljettaja

Attribuutti	Arvojoukko	Kuvailu
id	serial	id
etunimi	merkkijono, max 22 kirjainta	Ammattiautoilijan etunimi
sukunimi	merkkijono, max 22 kirjainta	Ammattiautoilijan sukunimi

Tietokohde: Ajovuoro

Attribuutti	Arvojoukko	Kuvailu
id	serial	id
kuljettaja_id	luku	Ajovuoron suorittaneen kuljettajan id
auto_id	luku	Ajovuoron suoritukseen käytetyn auton id

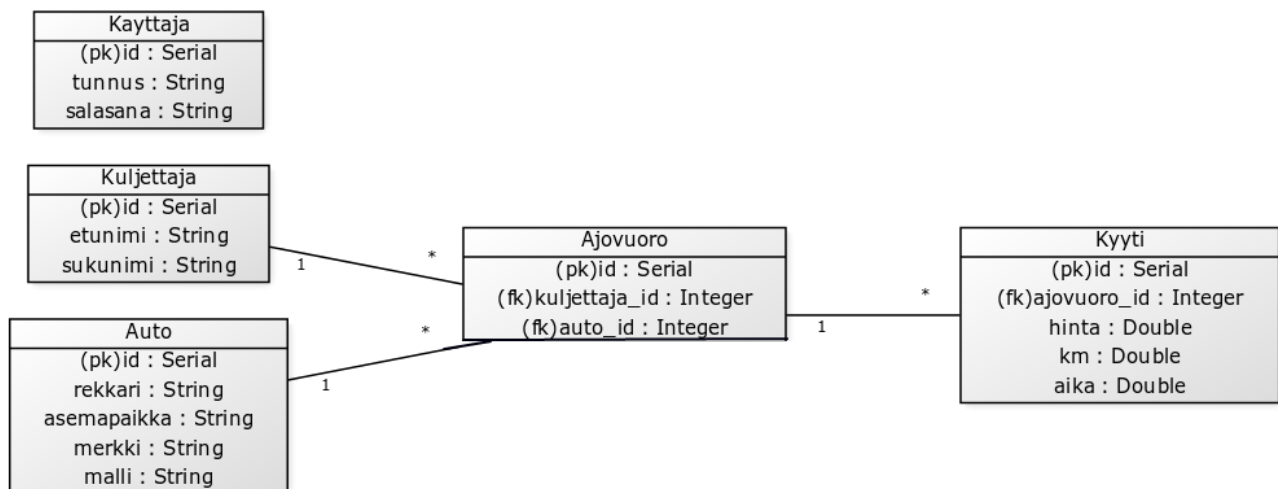
Tietokohde: Kyyti

Attribuutti	Arvojoukko	Kuvailu
id	serial	id
ajovuoro_id	luku	Kyydin ajovuoron vierasavain
hinta	desimaaliluku	Kyydin euromääräinen hinta
km	desimaaliluku	Maksetun kyydin kilometrit
aika	luku	Kyydin kesto minuuteissa

Tietokohde: Kayttaja

Attribuutti	Arvojoukko	Kuvailu
id	serial	id
tunnus	merkkijono, max 8 kirjainta	tunnus
salasana	merkkijono, max 8 kirjainta	salasana

3.2 Relaatiotietokantakaavio



4. Ohjeet

4.1 Asennus

Asenna sovellus kopioimalla /dist hakemiston sisällä oleva taksidb.war tiedosto nettiin näkyvään hakemistoon. Tämän jälkeen aseta tietokannan yhteystiedot oikeaksi hakemistossa /web/META-INF/ olevaan context.xml tiedostossa. Asetustiedoston mallin voi kopioida tiedostosta /web/META-INF/context.xml.dist.

4.2 Käyttö

Sovellus löytyy osoitteesta:

t-xbax.users.cs.helsinki.fi/taksidb/

Tarkemmat käyttöohjeet löytyy README tiedostosta mikä on githubissa osoitteessa:

<https://github.com/hosseinbahmanpour/tsoha/blob/master/README.md>

5. Ohjelmistokehitys

5.1 Testaus

Ohjelmistoa on testattu noin 10 henkilön voimin. Sovellus on toiminut hyvin ja ei ole kaatuillut.

5.2 Tunnetut bugit ja puutteet

Symbolit ja ääkköset:

Yrityksistä huolimatta, String syötteiden kohdalla ei ole onnistuttu estämään tai saamaan toimimaan ääkkösten sekä symbolien syöttämistä.

Ajovuoro liitostaulu:

Kun luodaan uusi Kyyti tietokohde, valitaan Auto ja Kuljettaja, ja sitten syötetään kyydin tiedot.

Kyyti tietokohteessa ei ole Autoa ja Kuljettajaa, vaan viite ajovuoro liitostauluun. Uutta Kyytiä luodessa, tietokantaa pyydetään etsimään Ajovuoroa missä esiintyy valitun Auton ja Kuljettajan id:t.

Jos kysyttyä Ajovuoroa ei ole olemassa, luodaan uusi.

Kuitenkin ohjelma löytää olemassa olevan Ajovuoron vain jos valittu Auton ja Kuljettajan id# = 1 ja näin Ajovuoron id # = 1. Jos luodaan uusi Auto ja uusi Kuljettaja, ja luodaan heille kaksi uutta Kyytiä, Ajovuoroja syntyy kaksi, vaikka uusien Kyytien pitäisi olla samaa Ajovuoroa.

5.3 Jatkokehitys

Sovellukseen olisi hyvä luoda paljon lisäominaisuuksia, joista päälimmäisenä mieleen tulee kuljettajien ja autojen vertailu ajettujen kilometrien, ajossa olevan ajan, sekä tietenkin euromääräisen menestyksen suhteen.

5.4 Omat kokemukset

Kiitos hienosta kurssista, sai työskennellä itsenäisesti ja valita aiheen vapaasti. Kaikista miellyttävintä oli oppia uutta, kuten servlettien, jsp tiedostojen sekä tägitiedoston tekeminen. Haastavinta oli lisäys ja muokkaus toimintojen tekeminen. Mukavinta oli sovelluksen ja tietokantojen suunnittelu.

6. Muu

6.1 Sql taulujen luonti

```
CREATE TABLE Auto (  
    id SERIAL PRIMARY KEY,  
    rekkari VARCHAR(22) NOT NULL,  
    asemapaikka VARCHAR(22) NOT NULL,  
    merkki VARCHAR(22) NOT NULL,  
    malli VARCHAR(22) NOT NULL  
);  
CREATE TABLE Kuljettaja (  
    id SERIAL PRIMARY KEY,  
    etunimi VARCHAR(22) NOT NULL,  
    sukunimi VARCHAR(22) NOT NULL  
);  
CREATE TABLE Ajovuoro (  
    id SERIAL PRIMARY KEY,  
    kuljettaja_id INTEGER NOT NULL,  
    auto_id INTEGER NOT NULL,  
    FOREIGN KEY (kuljettaja_id) REFERENCES kuljettaja(id) ON DELETE CASCADE ON  
UPDATE CASCADE,  
    FOREIGN KEY (auto_id) REFERENCES auto(id) ON DELETE CASCADE ON UPDATE  
CASCADE  
);  
CREATE TABLE Kyyti (  
    id SERIAL PRIMARY KEY,  
    ajovuoro_id INTEGER NOT NULL,  
    hinta DECIMAL(10,2) NOT NULL,  
    km DECIMAL(10,2) NOT NULL,  
    aika INTEGER NOT NULL,  
    FOREIGN KEY(ajovuoro_id) REFERENCES ajovuoro(id) ON DELETE CASCADE ON  
UPDATE CASCADE  
);  
CREATE TABLE Kayttaja (  
    id SERIAL PRIMARY KEY,  
    tunnus VARCHAR(8) NOT NULL,  
    salasana VARCHAR(8) NOT NULL  
);
```