

# Taloyhtiön palvelut

## Johdanto

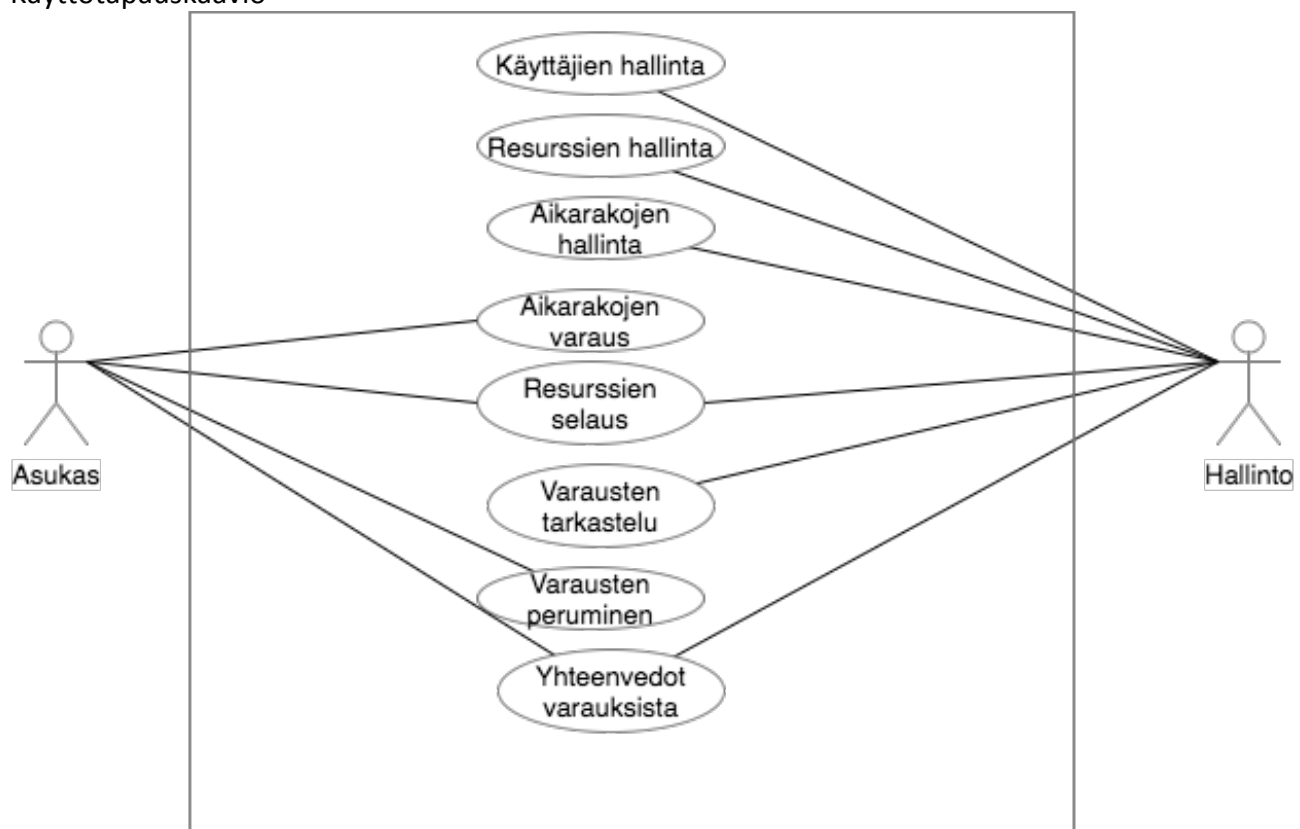
Toteutetaan tietojärjestelmä jota taloyhtiön asukkaat ja hallinto (isännöitsijä/huoltoyhtiö) voi käyttää taloyhtiön yhteiskäytössä olevien resurssien (sauna, pesutupa) hyödyntämiseen. Toteutettava tietojärjestelmä on web-sovellus jota voi käyttää www-selaimella niin mobiililaitteessa kuin tietokoneella. Järjestelmä tarjoaa taloyhtiön asukkaille mahdollisuuden varata käyttöönsä hallinnon määrittelemiä resursseja. Hallinto ja asukkaat voivat tulostaa laskuja resurssien käytön mukaan.

Tietojärjestelmä toteutetaan Heroku-pilvipalveluun seuraavilla teknologioilla:

|                      |                      |
|----------------------|----------------------|
| Tietokanta:          | Mariadb              |
| Palvelin:            | Node.js + Express.js |
| Template engine:     | Pug                  |
| Front-end framework: | Bootstrap            |

## Käyttötapaukset

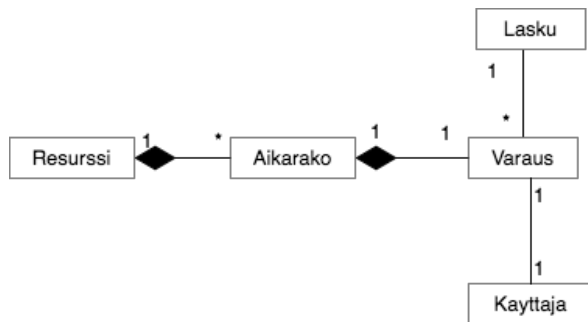
Käyttötapauskaavio



| Käyttötapaus                     | Käyttäjä          | Huomioitavaa |
|----------------------------------|-------------------|--------------|
| Järjestelmään kirjautuminen      | hallinto & asukas |              |
| Varauksen tekeminen              | hallinto & asukas |              |
| Varauksen peruutus               | hallinto & asukas |              |
| Laskun laatiminen                | hallinto & asukas |              |
| Yhteenveto resurssi(e)n käytöstä | hallinto & asukas |              |
| Yhteenveto maksuista             | hallinto & asukas |              |
| Maksetun laskun kirjaus          | hallinto          |              |
| Salasanan vaihto                 | hallinto & asukas |              |
| Aikojen lisäys                   | hallinto          |              |
| Varattavien resurssien lisäys    | hallinto          |              |
| Aikojen poisto                   | hallinto          |              |
| Varattavien resurssien poisto    | hallinto          |              |
| Aikojen muokkaus                 | hallinto          |              |
| Varattavien resurssien muokkaus  | hallinto          |              |

# Tietosisältö

## Käsitekaavio



## Tietokohteet

### Tietokohde: Resurssi

| Attribuutti       | Arvojoukko                                   | Kuvailu   |
|-------------------|--|---|
| id                | kokonaisluku                                 | Tietokohteen yksilöivä pääavain   |
| resurssinimi      | merkkijono (max 32 merkk.)                   | Resurssin nimi (esim. sauna)  |
| kayttoaikaalkaa   | aika   | kellonaika josta lähtien resurssi on oletusarvoisesti käytettävissä. Ensimmäisessä versiossa alkamisajankohdan on oltava tasalta tai puolelta (10:00, 10:30, 11:00 jne).            |
| kayttoaikapaattyy | aika   | kellonaika jonka jälkeen resurssi ei oletusarvoisesti ole enää käytettävissä. Ensimmäisessä versiossa päättymisajankohdan on oltava tasalta tai puolelta (10:00, 10:30, 11:00 jne). |
| varausyksikko     | aika   | kiinteä pituus varausyksikölle. Järjestelmän ensimmäisessä versiossa pituus on on oltava tunti tai puoli tuntia.  |
| hinta             | 4-numeroinen kaksidesimaalinen desimaaliluku | resurssin varausyksikön mittaiselle käytölle asetettu hinta   |

Taloyhtiön varattavissa oleva resurssi kuten sauna, pyykkitupa tms. Resursseja voi olla useita ja yhteen resurssiin voi kohdistua useita aikarakoja.

### Tietokohde: Aikarako

| Attribuutti | Arvojoukko   | Kuvailu                                    |
|-------------|--------------|--|
| id          | kokonaisluku | tietokohteen yksilöivä pääavain            |
| resurssi_id | kokonaisluku | viiteavain resurssiin jota aikarako koskee |
| paivamaara  | päivämäärä   | varauksen päivämäärä                       |
| aika        | kellonaika   | kellonaika jolloin varaus alkaa            |
| kesto       | aika         | varauksen kesto                            |

Resurssiin liittyvä aikarako eli ”slotti” jolloin resurssi on varattavissa. Yhteen resurssiin voi liittyä useita aikarakoja. Aikarakoihin voi kohdistua varauksia ja yhteen aikarakoon voi kohdistua vain yksi varaus.

### Tietokohde: Varaus

| Attribuutti | Arvojoukko             | Kuvailu  |
|-------------|------------------------|--|
| id          | kokonaisluku           | tietokohteen yksilöivä pääavain  |
| tunnusluku  | merkkijono (4 merkkiä) | varauksen tekijälle luovutettava tunnusluku resurssin käyttöä varten. <b>Ei käytössä tässä versiossa</b> |
| aikarako_id | kokonaisluku           | viiteavain kyseessä olevaan aikarakoon   |
| varaaja_id  | kokonaisluku           | viiteavain varauksen tehneeseen käyttäjään   |
| lasku_id    | kokonaisluku           | viiteavain laskuun johon varaus on liitetty, <b>ei käytössä tässä versiossa</b>                          |

Varaus jolla varataan aikarako. Varaus voi kohdistua vain yhteen aikarakoon. Käyttäjä tekee varauksen ja varaus voi kohdistua vain yhteen käyttäjään. Yhdellä käyttäjällä voi olla useita varauksia.

### Tietokohde: Kayttaja

| Attribuutti  | Arvojoukko                              | Kuvailu  |
|--------------|---|--|
| id           | kokonaisluku                            | tietokohteen yksilöivä pääavain                            |
| nimi         | merkkijono (64 merkkiä)                 | Käyttäjän merkkijono muodossa ”Jarkko Lampinen”            |
| asunto       | merkkijono (64 merkkiä)                 | käyttäjän asunto muodossa ”C64”                            |
| salasana     | merkkijono (64 merkkiä)                 | käyttäjän salasana   |
| kayttooikeus | määritelty tyyppi (hallinto tai asukas) | käyttäjän oikeudet järjestelmässä joko hallinto tai asukas |

Järjestelmän käyttäjä joka voi tehdä varauksia ja laskuja. Yhdellä käyttäjällä voi olla useita varauksia ja laskuja. Käyttäjällä on käyttöoikeus joka on joko asukkaan oikeudet tai suppeammat hallinnon oikeudet.

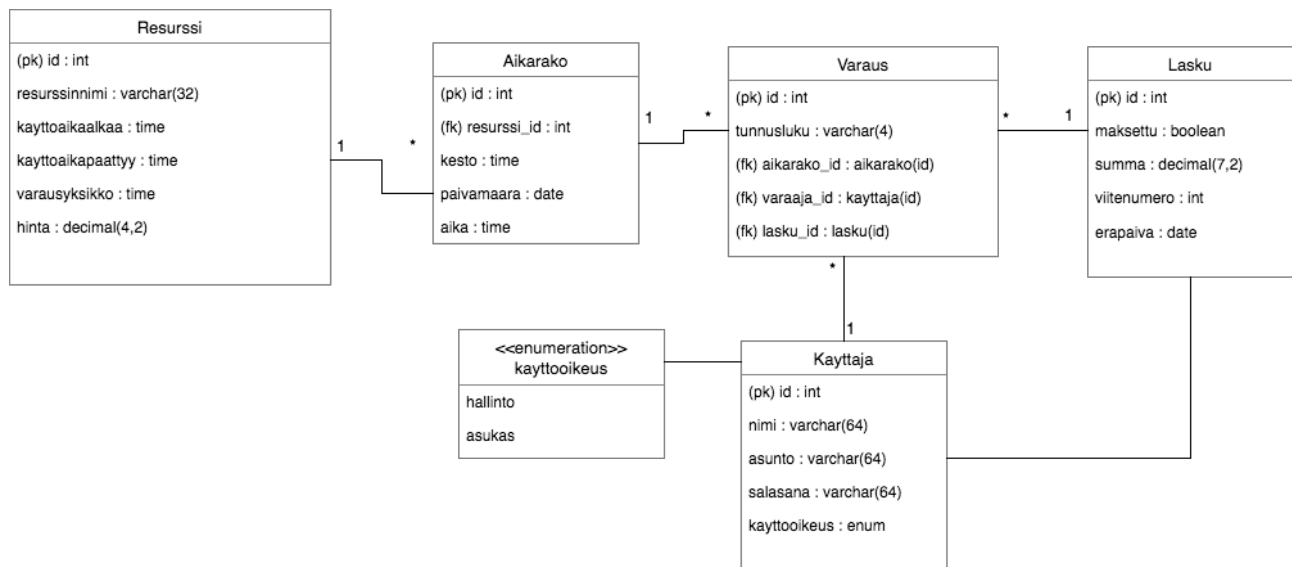
## Tietokohde: Lasku

Laskutusta ei toteuteta järjestelmän tässä versiossa.

| Attribuutti | Arvojoukko                                   | Kuvailu   |
|-------------|--|---|
| id          | kokonaisluku                                 | tietokohteen yksilöivä pääavain   |
| maksettu    | totuusarvo                                   | true, jos lasku maksettu; muuten false                                      |
| summa       | 7-numeroinen kaksidesimaalinen desimaaliluku | laskun loppusumma   |
| viitenumero | kokonaisluku                                 | laskun viitenumero joka on laskulla olevan viimeisen varauksen varausnumero |
| erapaiva    | päivämäärä                                   | laskun eräpäivä   |

Lasku liittyy yhteen tai useampaan varaukseen.

## Tietokantakaavio

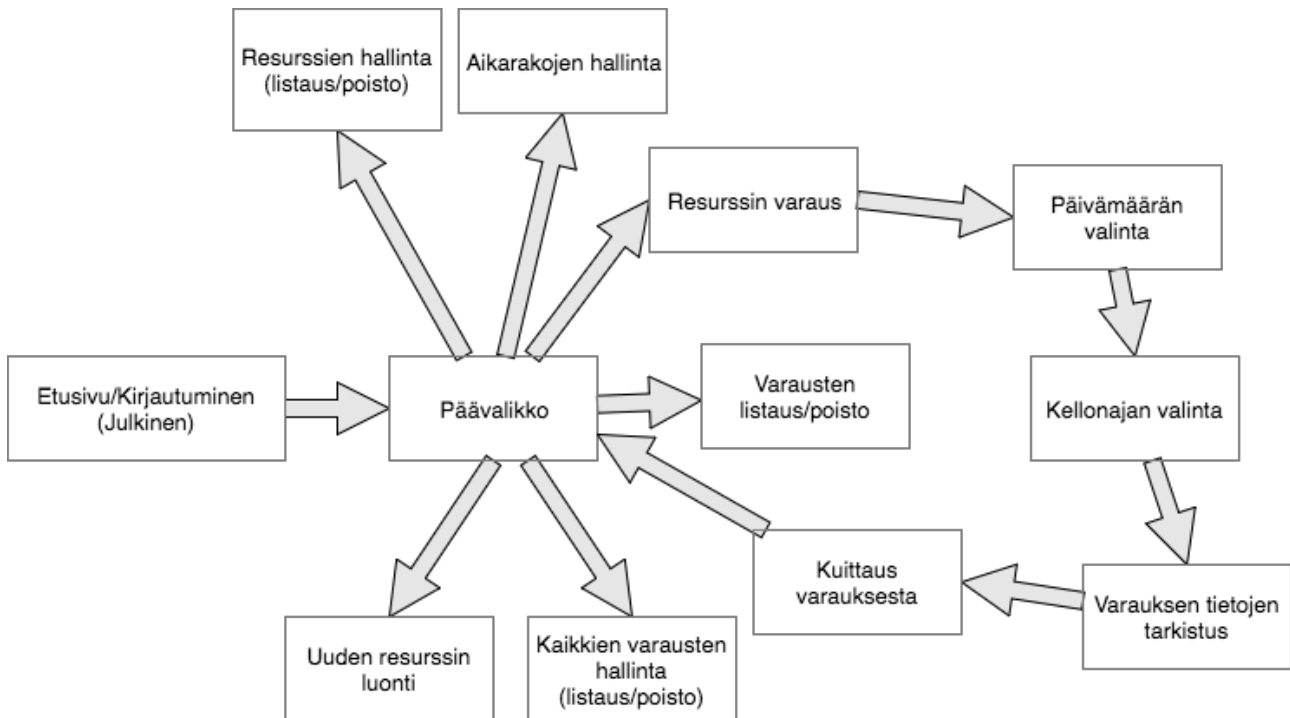


## Yleisrakenne

Järjestelmä on kahdessa js-tiedostossa jotka ovat projektin juuressa. Main.js-tiedostossa tuodaan projektin tarvitsemat pakkaukset/plugarit, määritellään kirjautuminen ja määritellään routet (get/post-pyyntöt). Tietokantaa ohjaavat kontrollerit ovat dbcontroller.js –tiedostossa jossa käytetään mariasql-plugarin funktioita SQL-lauseilla joihin lisätään parametreja.

Näkymät ovat views-kansiossa pug-tiedostoina joista pug template engine renderöi HTML-sivuja. Tyylimäärittelyissä käytetään Bootstrap CDN:ää joka on tuotu head.pug-tiedostossa joka sisältyy kaikkien järjestelmän renderöimien HTML-sivujen head-tageihin.

## Käyttöliittymäkaavio



## Asennus

Projekti asennetaan kloonaamalla git-repositorio omalle koneelle (git clone).

Seuraavaksi deployataan appi heorkuun heroku create:lla ja pushataan lopuksi herokuun git push heroku master –komennolla.

JawsDB Maria –tietokannan saa käyttöön Herokussa lisäämällä JawsDB Maria –tietokannan.

Appin asetuksissa Herokussa tulee asettaa Config Variabeleihin DB\_DATABASE, DB\_HOST, DB\_PASSWORD, DB\_USER luodun tietokannan mukaiset tiedot.

Sovelluksen toiminnan kannalta vaikuttaa olevan välttämätöntä, että tietokanta ja Heroku app ovat samalla regionilla (Europe).

## Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

Koska käytin työssäni itseleni niin paljon uusia teknologioita ei aikaa riittänyt oikeaan testaukseen. Myöskään käyttäjän syötteiden validointia en ehtinyt toteuttaa. Aion perehtyä Chai.js ja Mocha.js

–testausframeworkkeihin sekä express validator –pakkaukseen validoinnissa. Myös tietokannan testaukseen liittyen aion perehtyä myös mysql-test-frameworkkiin.

## Omat kokemukset

Päätökseni toteuttaa harjoitustyö monella minulle kokonaan uudella teknologialla osoittautui todella suureksi. Vaikka lopputulos näyttää vaatimattomalta niin olen ollut todella innoissani valitsemastani tiestä, koska olen joutunut hakemaan tietoa runsaasti ja oppinut todella paljon sen sijaan, että olisin seurannut valmiita ohjeita tai käyttänyt valmista pohjaa. Käytin apunani paljon netistä löytyviä tutoriaaleja, mutta mikään tutoriaali ei käyttänyt täysin samoja teknologioita. Esimerkiksi Mozillan Developer-sivuilta löytyvästä kirjastoesimerkistä ([https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\\_Nodejs/Tutorial\\_local\\_library\\_website](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Tutorial_local_library_website)) sain paljon apua, mutta tietokantaratkaisu oli siinäkin erilainen.