

Tietokantasovellus:
Mökkivarausten välityspalvelu

Paavo Huhtala

25. huhtikuuta 2016

Sisältö

1	Johdanto	3
1.1	Toiminta ja tavoitteet	3
1.2	Toteutus ja ympäristö	3
2	Käyttötapaukset	4
2.1	Käyttötapauskaavio	4
2.2	Käyttäjärühmät	4
2.3	Käyttötapauskuvaukset	5
2.3.1	Kirjautumattoman käyttäjän käyttötapaukset	5
2.3.2	Kaikkien käyttäjien käyttötapaukset	5
2.3.3	Vuokraajien käyttötapaukset	5
2.3.4	Järjestelmänvalvojan käyttötapaukset	6
2.3.5	Vuokranantajan käyttötapaukset	6
2.3.6	Maksunvälittäjän käyttötapaukset	6
3	Järjestelmän tietosisältö	7
3.1	Käsitekaavio	7
3.2	Selitteet	8
3.2.1	Osoite	8
3.2.2	Käyttäjä	8
3.2.3	Sessio	8
3.2.4	Myyjätiedot	8
3.2.5	Mökki	9
3.2.6	Varuste	9
3.2.7	Varaus	9
3.2.8	Sisäisen varauksen tiedot	9
3.2.9	Ulkoisen varauksen tiedot	10
3.2.10	Arvostelu	10
4	Relaatiotietokantakaavio	11
5	Järjestelmän rakenne	12
5.1	Yleisrakenne	12
5.2	Tiedostojen nimeämiskäytännöt	13

6	Asennus	14
6.1	Ennakkovaatimukset	14
6.2	Lataaminen	14
6.3	Tietokannan valmistelu	14
6.4	Riippuvuudet	14
6.5	Kääntäminen	15
7	Käynnistys ja käyttö	16
7.1	Käynnistys	16
7.2	Käyttäjätunnukset	16

Luku 1

Johdanto

1.1 Toiminta ja tavoitteet

Sovellus on suunnattu mökkien omistajille sekä niiden vuokraamisesta kiinnostuneille.

Omistajille sovellus tarjoaa mahdollisuuden asettaa mökkejään julkisesti nähtäville ja vuokrattavaksi.

Vuokraajille sovellus tarjoaa mahdollisuuden selata mökkejä, vuokrata niitä ja selata varauksiaan.

Sovellus huolehtii myös maksunvälityksestä, joskin tässä projektissa sen toteuttaminen ohitetaan.

1.2 Toteutus ja ympäristö

Palvelinpuoli toteutetaan Node.js-alustalla ja TypeScript-ohjelmointikielellä. Jos sille ilmenee tarvetta, palvelin laitetaan Nginx tai Apache-proxyn taakse, joka cacheaa ja salaa liikenteen. Tietokantana käytetään PostgreSQL:ää.

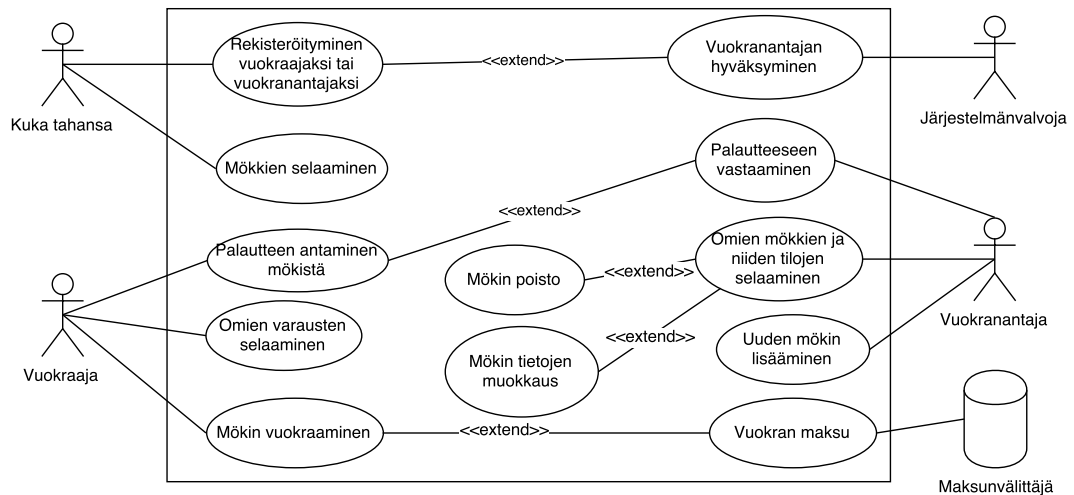
Sovellus suunnitellaan alustariippumattomaksi (Linux ja Windows), mutta lopullinen asennus tehdään Ubuntu Server 14.04:ään. Tällä hetkellä sovellusta hostataan DigitalOceanissa, osoitteessa <http://lodgist.paavo.me>.

Palvelimen tavoin käyttöliittymä toteutetaan TypeScript-ohjelmointikielellä, joka käännetään automaattisesti JavaScriptiksi. Alustavasti käyttöliittymän ja MVC-mallin toteutuksessa hyödynnetään Angular.js-kirjastoa.

Luku 2

Käyttötapaukset

2.1 Käyttötapauskaavio



2.2 Käyttäjärühmät

Kuka tahansa

Kenellä tahansa tarkoitetaan kaikkia rekisteröityneitä ja rekisteröitymättömiä käyttäjiä.

Vuokraaja

Vuokraaja on vuokraajaksi rekisteröitynyt käyttäjä.

Vuokranantaja

Vuokranantaja on loma-asuntoja vuokralle antava henkilö. Järjestelmänvalvoja hyväksyy uudet vuokranantajat erikseen.

Järjestelmänvalvoja

Järjestelmänvalvoja huolehtii palvelun moderoinnista ja ylläpitotehtävistä, sekä uusien vuokranantajien hyväksynnästä.

Maksunvälittäjä

Maksunvälittäjä on ulkopuolinen, automatisoitu maksunvälityspalvelu. Tässä kyseisessä toteutuksessa sitä ei toteuteta.

2.3 Käyttötapauskuvaukset

2.3.1 Kirjautumattoman käyttäjän käyttötapaukset

Rekisteröityminen

Järjestelmään voi rekisteröityä joko vuokraajaksi tai vuokranantajaksi. Vuokraajalta vaaditaan lisätietoja, ja järjestelmänvalvoja hyväksyy uudet vuokraajat käsin. Käyttäjä saa vahvistuksen hyväksynnän jälkeen.

Kirjautuminen

2.3.2 Kaikkien käyttäjien käyttötapaukset

Mökkien selaaminen

2.3.3 Vuokraajien käyttötapaukset

Mökin vuokraaminen

Löydettyään kiinnostavan mökin vuokraaja valitsee vuokrausajankohdan (alku, loppu, kesto) ja näkee, onko mökki saatavilla sinä ajankohtana sekä varauksen hinnan. Jos vuokraaja päättää vuokrata mökin, käyttäjä siirretään maksamaan. Maksun onnistuessa mökki lisätään käyttäjän varauksiin.

Varausten selaaminen

Käyttäjä näkee entiset, nykyiset ja tulevat varaukset.

Palautteen antaminen

Käyttäjä voi antaa palautetta entisistä ja nykyisistä varauksistaan. Palaute voi olla julkista tai vain varaajalle näkyvää.

2.3.4 Järjestelmänvalvojan käyttötapaukset

Vuokraajien hyväksyminen

Palautteen moderointi

2.3.5 Vuokranantajan käyttötapaukset

Uuden mökin lisääminen

Mökkeihin kuuluu nimi, kuvaus, saatavuustiedot ja tiedot varustelusta. Mökkeihin voi liittyä myös kuvia, sijaintidataa yms.

Mökkien selaaminen

Mökin tietojen muokkaaminen

Mökin poisto

Palautteeseen vastaaminen

2.3.6 Maksunvälittäjän käyttötapaukset

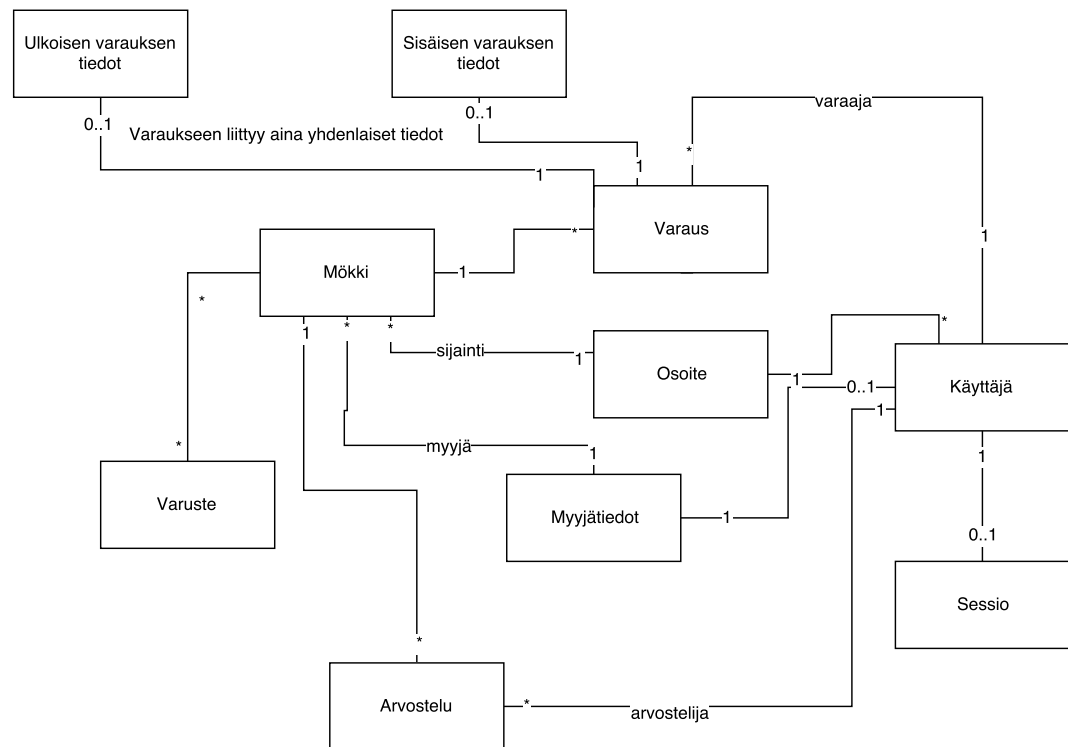
Vuokranmaksu

Ulkopuolinen maksunvälittäjä osallistuu vuokranmaksuun pyytämällä käyttäjältä maksutiedot ja välittämällä onnistuneet maksut vuokraajille.

Luku 3

Järjestelmän tietosisältö

3.1 Käsitekaavio



3.2 Selitteet

Lähes jokaiseen tietokohteeseen liittyy listattujen ominaisuuksien lisäksi myös uniikki kokonaislukutunniste.

3.2.1 Osoite

Koostuu kahdesta katuosoiterivistä, paikkakunnasta ja postinumerosta. Sisältää myös kaksimerkkisen maakoodin mahdollista laajennusta varten.

3.2.2 Käyttäjä

Attribuutti	Arvojoukko	Kuvaus
Rooli	Yksi järjestelmän rooleista	
Nimi	Rajoittamaton määrä tekstiä	Käyttäjän kutsumanimi.
Sähköposti	255 merkkiä tekstiä	
Salasana	60 merkkiä tekstiä	BCrypt-salasanatiiviste ja salt.
Puhelin	Rajoittamaton määrä tekstiä	
Osoite	Validi viite osoitetauluun	Käyttäjän laskutusosoite.

Käyttäjäroolit

unverifiedUser Käyttäjä, jonka sähköpostiosoitetta ei ole vahvistettu.

user Tavallinen käyttäjä.

unapprovedSeller Vuokranantaja, joka odottaa hyväksyntää.

seller Vuokranantaja.

admin Järjestelmänvalvoja, joka moderoi sisältöä ja hyväksyy uusia vuokranantajia.

3.2.3 Sessio

Käyttäjäkohtainen määräaikainen sessio, jota käytetään kirjautumisten tallentamiseen. Sisältää uniikin tunnuksen (32 merkkiä), voimassaoloajan ja käyttäjäviitteen.

3.2.4 Myyjätiedot

Lisätietoa käyttäjään, jos käyttäjä on vahvistettu ja hyväksytty myyjä.

3.2.5 Mökki

Attribuutti	Arvojoukko	Kuvaus
Nimi	Rajoittamaton määrä tekstiä	
Omistaja	Validi viite myyjätietotauluun	
Osoite	Validi viite osoitetauluun	Mökin sijainti.
Kuvaus	Rajoittamaton määrä tekstiä	
Näkyvyys	Tosi tai epätosi	Onko mökki näkyvissä käyttäjille?
Varauksen alkuaika	Kellonaika	Uusien varausten alkamisaika.
Varauksen loppumisaika	Kellonaika	Uusien varausten päättymisaika, eli milloin mökki vapautuu.
Hinta	Kaksidesimaalinen positiivinen luku	Yökohtainen hinta uusille varauksille.
Pinta-ala	Positiivinen desimaaliluku	Vapaaehtoinen.
Kerrosten määrä	Positiivinen kokonaisluku	Vapaaehtoinen.
Rakennusvuosi	Positiivinen kokonaisluku	Vapaaehtoinen.
Remontointivuosi	Positiivinen kokonaisluku, suurempi tai yhtä suurin kuin rakennusvuosi	Vapaaehtoinen.

3.2.6 Varuste

Mökin varuste tai myyntiargumentti (*"amenity"*). Kostuu nimestä ja vapaaehtoisesta kuvakkeesta.

3.2.7 Varaus

Varaukseen kuuluu aina tyypistä riippuen joko sisäisen tai ulkoisen varauksen lisätietoja.

Attribuutti	Arvojoukko	Kuvaus
Mökki	Validi viite mökkitauluun	
Tyyppi	Varaustyyppi (kts. alla)	
Varattu aika	Aikaväli, ei saa leikata saman mökin muiden varausten kanssa.	

Varaustyytit

user Järjestelmän sisäinen varaus.

external Järjestelmän ulkopuolinen varaus.

3.2.8 Sisäisen varauksen tiedot

Lisätietoa varaukseen, jos varaus on tehty järjestelmän sisällä (= sivustolla). Sisältää varaajan käyttäjätunnuksen ja varauksesta maksetun hinnan.

3.2.9 Ulkoisen varauksen tiedot

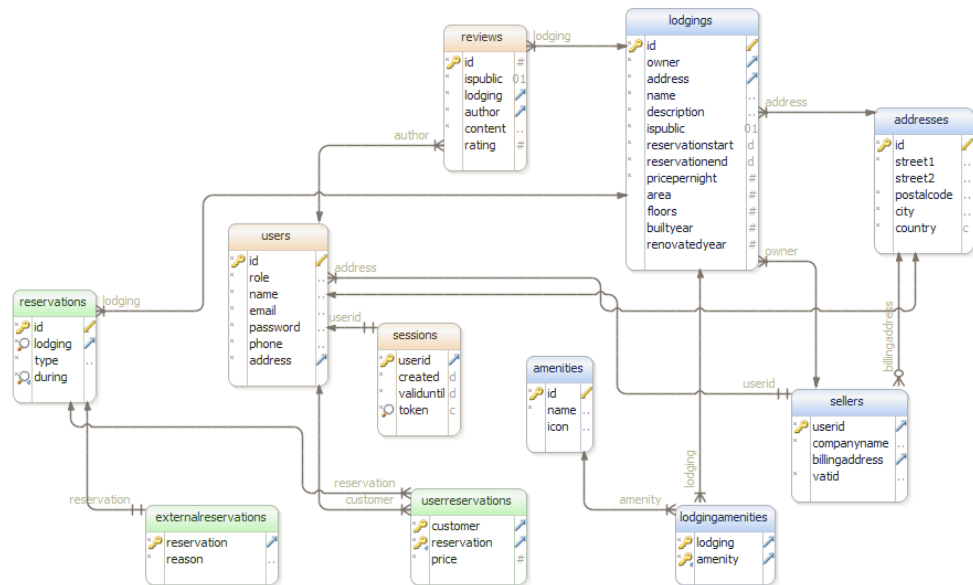
Lisätietoa varaukseen, jos varaus on tehty järjestelmän ulkopuolella mökin omistajan toimesta (esimerkiksi oman varauksen tai remontin takia). Sisältää selittekentän.

3.2.10 Arvostelu

Arvostelu mökistä. Sisältää arvosanan kokonaislukuna väliltä 1 - 5, mökkitunnuksen, käyttäjätunnuksen ja selitetekstin.

Luku 4

Relaatiotietokantaavaio



Generated using DbSchema

Luku 5

Järjestelmän rakenne

5.1 Yleisrakenne

Järjestelmä perustuu MVC-arkkitehtuuriin, jota sovelletaan sekä palvelimen että selaimen puolella. Osa toiminnoista on toteutettu REST-rajapintoina, joiden toteutukset löytyvät kansiota `server/controllers/api/`. Palvelimen kaikki sivut ja rajapinnat rekisteröidään tiedostossa `server/Routes.ts`.

Käyttäjille näkyvät sivut koostuvat neljästä komponentista:

Malli Mallit haetaan tietokannasta ja/tai luodaan kyselyn perusteella. Suurimmalle osalle tietokantatauluista on tyyppikuvaukset, jotka sijaitsevat kansiossa `server/models/`.

Näkymä Näkymät on kirjoitettu Jade-templatekielellä. Staattinen osa näkymän sisällöstä täytetään palvelimella, ja dynaaminen sisältö täytetään Angular.js:n avulla. Templatet löytyvät kansiota `views/`.

Palvelinkontrolleri Palvelinkontrollerit luovat kyselyyn perustuvat mallit, hakevat tietokannasta tietuita, yhdistävät ne näkymiin ja lähettävät tuloksen käyttäjälle. Myös REST-rajapinnoille on palvelinkontrollerit, vaikka muut komponentit puuttuvat niistä. Ne löytyvät kansiota `server/controllers/`

Käyttöliittymäkontrolleri Angular.js:llä toteutettuja käyttöliittymäkontrollereita ajetaan käyttäjän selaimella. Ne huolehtivat dynaamisesta sisällöstä, lomakkeiden vahvistamisesta sekä rajapintakyselyistä. Ne löytyvät kansiota `client/js/controllers/`.

Kommunikointi tietokannan kanssa tapahtuu pääasiallisesti DAOilla eli Data Access Objecteilla, jotka löytyvät kansiota `server/database/daos/`. Lähes jokainen niistä perustuu BaseDao-luokkaan, jossa on suurin osa tietokantalogiikasta. BaseDao löytyy tiedostosta `server/database/BaseDao.ts`.

Kirjautuminen ja sessionhallinta on toteutettu kansion `server/authentication/` tiedostoilla. Vastaavasti tunnistautuminen (eli mitä eri käyttäjät saavat tehdä) on toteutettu kansion `server/authorization/` tiedostoilla.

5.2 Tiedostojen nimeämiskäytännöt

- Palvelimen TypeScript-tiedostot käyttävät **PascalCasea**.
- Selaimessa ajettavat TypeScript-tiedostot käyttävät **snake_casea**.
- Näkymämallit sekä tyylitiedostot käyttävät **snake_casea**.

Luku 6

Asennus

6.1 Ennakkovaatimukset

Sovellus vaatii modernit versiot Node.js:stä (5.9.0) [ja NPM:stä] sekä PostgreSQL:stä (9.3.11). Ohjelma toimii ainakin Ubuntu Linux- ja Windows-ympäristöissä.

Sovelluksen riippuvuuksien asentaminen sekä kääntäminen vaatii myös NPM-paketit `gulp`, `typings` ja `bower` globaalisti asennettuna.

6.2 Lataaminen

Sovelluksen lähdekoodi ladataan Gitin kautta. Siirry haluamaasi hakemistoon (esim. `/var/node/lodgist`) ja kloonaa repositorio:

```
git clone https://github.com/paavohuhtala/lodgist.git .
```

6.3 Tietokannan valmistelu

Kansiosta `/sql` löytyvät tietokantaskriptit. Sovellus olettaa, että käytössä on tietokanta nimeltään `lodgist`, ja että se on saatavilla portissa 5432 nykyisellä koneella. Luo oikean niminen tietokanta, ja suorita skriptit `create.sql` ja `test_data.sql` käyttäen `psql`-komentiriviohjelmaa tai PgAdmin 3- työpöytäohjelmaa.

Tietokannan yhdistämistietoja (connection string) voi toistaiseksi muuttaa ainoastaan muokkaamalla tiedostoa `/server/app.ts`.

6.4 Riippuvuudet

- Asenna NPM -paketit:
`npm install`
- Asenna Bower -paketit:
`bower install`

- Asenna TypeScript -tyyppimäärittelyt:
`npm run typings`

6.5 Kääntäminen

Sovellus käännettään komennolla `gulp`. Jos kääntäminen onnistuu, tiedostot ilmestyvät kansioon `/app`.

Luku 7

Käynnistys ja käyttö

7.1 Käynnistys

Sovellus voidaan käynnistää ajamalla komento `npm run start` sovelluksen juurihakemistosta. Oletuksena sovellus on käytettävissä portissa 8080, jota voidaan vaihtaa `server/app.ts` -tiedostosta.

Sovellukseen pääsee selaimessa navigoimalla osoitteeseen `http://localhost:8080`.

7.2 Käyttäjätunnukset

Voit luoda uuden käyttäjän käyttämällä rekisteröintisivua, mutta uudet käyttäjät eivät voi rekisteröidä mökkejä. Testaamisen helpottamiseksi järjestelmään on luotu testitunnuksia, yksi joka roolille. Järjestelmä käyttää kirjautumiseen sähköpostiosoitteita.

Kaikkien esiluotujen käyttäjien salasana on `salasana`.

Asiakas asiakas@example.com

Myyjä myyjä@example.com

Järjestelmänvalvoja admin@example.com