

Yleissuunnitelma

Varausjärjestelmä

Henkilötiedot

Matti Yli-Ojanperä

355153

AUT

2012

24.2.2018

Yleiskuvaus

Kuvitteelliselle yritykselle toteutettu varausjärjestelmä, joka yhdistää kalenteri-asiakas-, palvelu- ja resurssitietoja. Ohjelmaa käytetään graafisen käyttöliittymän kautta. Ohjelmaan liitetään tiedot yrityksen käytössä olevista resursseista sekä yrityksen tarjoamista palveluista. Käyttäjä voi muokata näitä tietoja vastaamaan yrityksen kulloistakin tilannetta.

Varausta tehdessä ohjelma tallentaa tiedot tehdyistä varauksista kalenteriin ja merkitsee valitut resurssit varatuiksi, jolloin konfliktitilanteessa (resurssi on jo varattu) varausta ei voida toteuttaa. Varaukseen ohjelma liittää tarvittavat tiedot asiakkaasta sekä asiakkaan valitsemat palvelut.

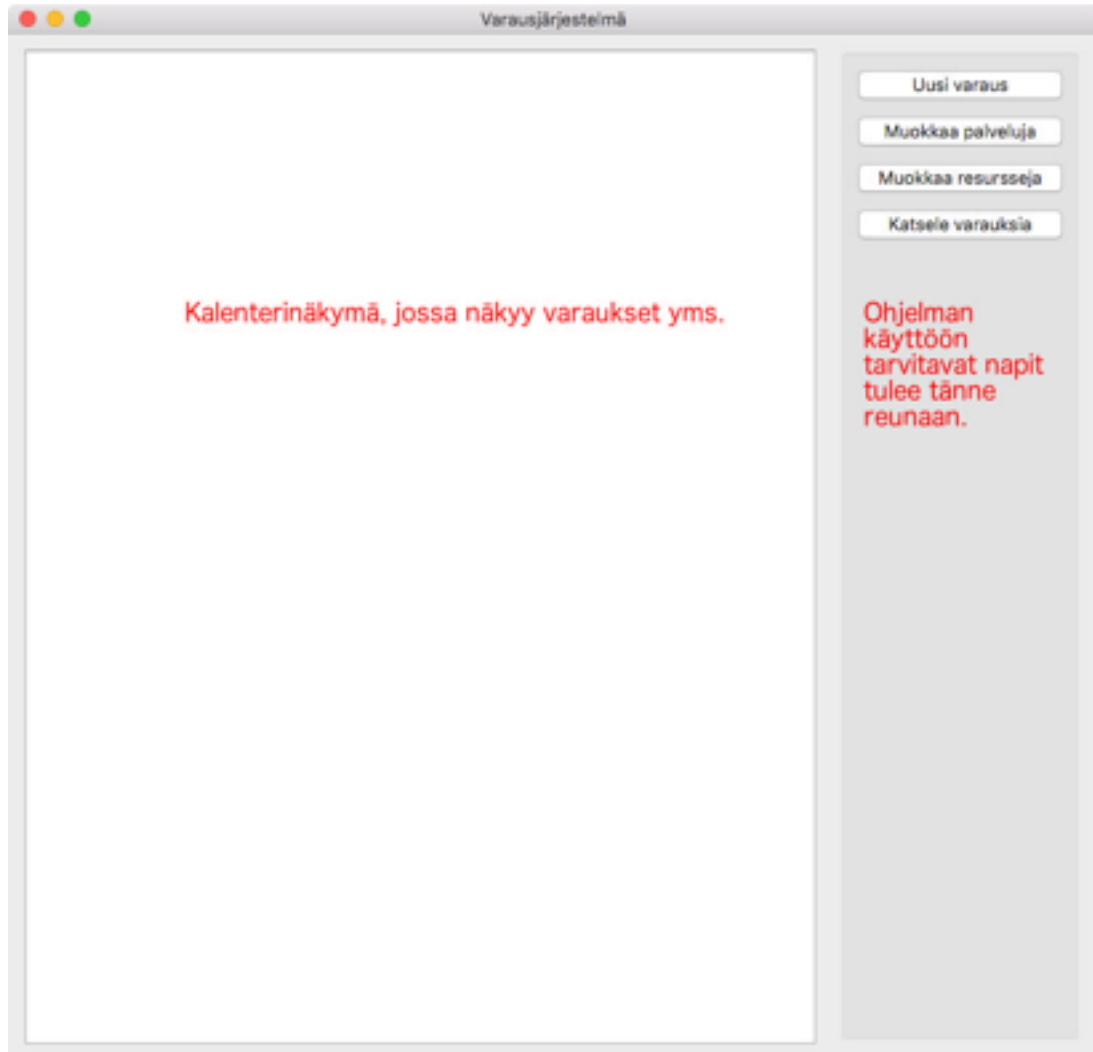
Ohjelmalla voi tarkistaa valitun ajankohdan varaustilanteen ja tulostaa varaustilanteen käyttäjän määrittämältä aikaväliltä. Ohjelmalla voi katsella palveluun, asiakkaaseen tai resurssiin liittyviä historiatietoja. Ohjelma laskee myös varauksen pituuden ja siihen valittujen palveluiden perusteella hinnan.

Ohjelma tallentaa keräämänsä tiedot SQLite realaatiotietokantaan.

Olen ajatellut toteuttaa työn vaikeana siten, että on ensin toimiva keskivaikea versio ohjelmasta ja sitten siihen lisää ominaisuuksia.

Käyttöliittymän luonnos

Kuvassa luonnos ohjelman pääikkunasta. Lisäksi ohjelma käyttää pop-up ikkunoita varausta lisätessä, palveluja ja resursseja muokatessa sekä katsellessa varauksia ja historiatietoja.



Tiedostot ja tiedostoformaatit

Ohjelma käyttää SQLite-relaattiotietokantaa tietojen tallentamiseen.

Tietokantaan tulee ainakin seuraavat taulut:

- Resources(ID, Name, Type)
- Services(ID, Name, Price, Description)
- Customers(ID, Name, Address, Email, Phone)
- Reservations(ID, CustomerID, ResourceID, Start, End)
- ReservationServices(ID, ReservationID, ServiceID)

Järjestelmätestaussuunnitelma

Ohjelman toiminnan kannalta on tärkeää, että yhteys tietokantaan toimii moitteettomasti. On siis tärkeää testata, että tietokannasta haku sekä sinne tietojen lisääminen toimii.

Varauksien teko on yksi tärkeimmistä ohjelman toiminnoista, joten sen toimintaa pitää testata. Testitapauksia on ainakin seuraavat:

- Varauksen lisääminen vapaalle resurssille
- Varauksen lisääminen jo varatulle resurssille
- Varauksen lisääminen asiakkaalle, joka ei ole vielä tietokannassa
- Varauksen lisääminen asiakkaalle, joka on tietokannassa.
- Varauksen lisääminen menneisyyteen (ei pidä onnistua)
- Varauksen lisääminen siten, että alkamisaika > päättymisaika (ei pidä onnistua)

Resurssitietojen muokkaus on myös tärkeä testattava ominaisuus.

Testitapauksia on ainakin seuraavat:

- Resurssin lisääminen
- Resurssin tietojen muuttaminen
- Resurssin poistaminen, kun sitä ei ole varattu
- Resurssin poistaminen, kun se on varattu (ei pidä onnistua ts. varaukset pitää muuttaa/poistaa ensin)

Palvelutietojen muokkaus testataan samaan tyyliin kuin resurssitietojen.

Varausten ja historiatietojen hakeminen ja katseleminen pitää myös testata. Näissä pitää testata, että kaikki halutut tiedot näkyvät muttei mitään ylimääräistä.

Testitapauksia on ainakin seuraavat:

- Aikavälin alku ja loppu
- Tyhjä haku
- Yksittäinen tulos
- Monta tulosta

Myös hinnan laskemista pitää testata, että hinta lasketaan oikein.

Testitapauksia on ainakin seuraavat:

- Yksi palvelu
- Monta palvelua
- Ei yhtään palvelua

Lopullista ohjelmaa kannattaa testata käyttämällä ohjelmaa sen graafisen käyttöliittymän kautta, jolloin myös käytettävyyttä voi arvioida.