



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Mikko Pukkila

HUONEVARAUS- JA ASIAKASTIETO- JÄRJESTELMÄ

Tekniikka ja liikenne

2015

TIIVISTELMÄ

Tekijä	Mikko Pukkila
Opinnäytetyön nimi	Huonevaraus- ja asiakastietojärjestelmä
Vuosi	2015
Kieli	suomi
Sivumäärä	47 + 1 liite
Ohjaaja	Timo Kankaanpää

Geriatrinen kuntoutuskeskus Geritrim (Selkämeren sairaskotisäätiö) tarvitsi käyttöönsä paremman järjestelmän huonevarausten seurantaan ja ylläpitoon. Heidän vanha järjestelmä oli hyvin vaikeasti seurattava ja työläs ylläpidettävä. He tarvitsivat järjestelmän, jota olisi helppo käyttää, mutta joka olisi myös monipuolinen ja toisi mukanaan uusia mahdollisuuksia huonevaraus- ja asiakastietojen käyttöä helpottamaan sekä tuottamaan yrityksen johdolle reaaliaikaisia taloushallinnon mittareita.

Projektin tuloksena toteutettiin Windows-työpöytäsovellus Visual Studiolla, ohjelmointikielen ollessa VB.NET. Sovellus käyttää tietolähteenään MySQL -tietokantaa. Sovellusta käytetään yhdeltä tietokoneelta paikallisesti ja sovelluksen käyttämä tietokanta asennettiin samaan tietokoneeseen.

Opinnäytetyöprojekti onnistui odotusten mukaisesti ja asiakkaan jokapäiväiseen käyttöön saatiin toteutettua järjestelmä, jonka avulla voidaan toteuttaa muun muassa huonevarausten kirjaukset, erilaisia varaus- ja asiakashakuja, kutsukirjeiden tulostusta sekä johdon laskelmia.

SISÄLLYS

1	JOHDANTO.....	5
2	MÄÄRITTELY	6
	2.1 Vaatimusmäärittely	6
	2.1.1 Yleiskuvaus	6
	2.1.2 Toiminnalliset vaatimukset	7
	2.1.3 Ei-toiminnalliset vaatimukset.....	9
	2.2 Toiminnallinen määrittely.....	10
	2.2.1 Toiminnot.....	10
	2.2.2 Tiedot ja syötteen	12
3	SUUNNITTELU	13
	3.1 Suunnittelussa käytetyt ohjelmat	13
	3.2 Tietokannan suunnittelu.....	13
	3.3 Käyttöliittymän suunnittelu	16
	3.3.1 Suunnitteluperiaatteet ja suunnittelun alkuvaiheet.....	16
	3.3.2 Käyttöliittymäkuvaukset	18
	3.4 Ohjelmiston suunnittelu ja toteutus	31
	3.4.1 Yleistä	31
	3.4.2 Kalenterinäköymän suunnittelu	32
	3.4.3 Muut ohjelmistosuunnittelun vaiheet.....	39
4	TESTAUS JA ASENNUS.....	44
	4.1 Järjestelmän testaus.....	44
	4.2 Järjestelmän asennus ja varmuuskopioinnin järjestäminen	45
5	YHTEENVETO	46
	LÄHTEET.....	47
	LIITTEET	

KUVIOLUETTELO

Kuvio 1.	Käyttötapauskaavio	s. 7
Kuvio 2.	ER-kaavio	s. 16
Kuvio 3.	Kalenterinäkymän ensimmäinen mallinnuskuva	s. 17
Kuvio 4.	Kalenterinäkymä	s. 18
Kuvio 5.	Valikon rakenne	s. 19
Kuvio 6.	Asiakkaat-ikkuna	s. 20
Kuvio 7.	Asiakaslista	s. 21
Kuvio 8.	Uusi varaus	s. 21
Kuvio 9.	Varauksen tiedot	s. 22
Kuvio 10.	Asiakkaan tiedot	s. 23
Kuvio 11.	Kutsukirjeen tulostuskielen valinta	s. 23
Kuvio 12.	Kutsukirje	s. 24
Kuvio 13.	Asiakastyypit	s. 25
Kuvio 14.	Varauslista	s. 26
Kuvio 15.	Laskut	s. 26
Kuvio 16.	Laskelmat-raportti	s. 27
Kuvio 17.	Loki	s. 28
Kuvio 18.	Lokitietojen tallennus	s. 28
Kuvio 19.	Lokitiedot tekstitiedostossa	s. 28
Kuvio 20.	Pyhäpäivät	s. 29
Kuvio 21.	Poista varaus	s. 30
Kuvio 22.	Tietojen poistovaroituksia	s. 30
Kuvio 23.	Asiakastietuetta ei voi poistaa	s. 31
Kuvio 24.	Kalenterin kolme ensimmäistä soluriviä	s. 32
Kuvio 25.	Kalenterin alkumäärittäjä	s. 33
Kuvio 26.	Kalenterin solujen alkumäärittäjä	s. 34
Kuvio 27.	Arkipyhien käsittely	s. 35
Kuvio 28.	Ote koodin Case-rakenteesta	s. 37
Kuvio 29.	Kalenterin selaustoiminnot	s. 38
Kuvio 30.	Tyhjän solun klikkaus	s. 39
Kuvio 31.	”Valitse asiakas” -alasvetovalikko	s. 40
Kuvio 32.	Kaksoisklikkaustapahtuma	s. 40
Kuvio 33.	”Uusi varaus” -lomakkeen asiakkaan valinta	s. 41
Kuvio 34.	”Varauksen tiedot” -lomakkeen toimintoja	s. 41
Kuvio 35.	Asiakkaan huonevaraustietojen haku	s. 42
Kuvio 36.	Huonevarausten kuormitusasteen laskeminen	s. 43

TERMIT JA LYHENTEET

Visual Studio	Ohjelmointiympäristö (sovellus), jossa työ on suunniteltu.
VB.NET	Visual Studiossa käytössä oleva Visual Basic -perustainen ohjelmointikieli, jolla työn koodaus on toteutettu.
MySQL	Avoimeen lähdekoodiin perustuva relaatiotietokantojen hallintajärjestelmä, joka on käytössä tässä työssä.
MySQL Server Connector/.NET	Tietokantapalvelin, johon tietokanta asennetaan. Tietokanta-ajuri tietokannan ja sovelluksen välisen tietoliikenteen muodostamista varten.
Käyttötapauskaavio	Voidaan kuvata eri käyttäjätapauksia, eli mitä kukin voi ohjelmalla tehdä.
ER-kaavio	Luokkakaavio, jolla voidaan kuvata tietolähteitä ja niiden tietoyhteyksiä (esim. tietokannan taulut).
Työpöytäsovellus	Sovellus, joka on asennettu paikallisesti tietokoneeseen.
Sovellus / Ohjelma Ikkuna	Viittaa huonevaraus- ja asiakastietojärjestelmään. Tällä tarkoitetaan ohjelmassa olevia ikkunoita, jotka sisältävät eri toimintoja, esimerkiksi ohjelman päänäkökymä on kalenteri-ikkuna.
Lomake, Form	Kaikki ohjelman ikkunat ovat lomakkeita (Form), jotka sisältävät eri tietoja.
MainForm	Ohjelman päälomake (tässä sovelluksessa "Kalenterinäkökymä"), johon ohjelma oletuksena käynnistyy.
Ohjelmakirjasto	Valmiita (määriteltyjä) joukkoja ohjelmassa käytettäviä luokkia ja toimintoja.
While, Case	Ohjelmoinnissa käytettyjä toisto- ja valintarakenteita.
DataGridView	Solurakenteinen ohjelmakomponentti, jolla voidaan toteuttaa ja määrittää taulukkomaisia rakenteita.
DataTable	Ohjelmassa käytettävä objekti, jonka avulla tietoa säilytetään (tieto pysyy muistissa).
PrintDocument	Dokumenttien tulostuksessa käytettävä metodi.
PrintDialog	Metodi, joka mahdollistaa tulostusasetusten valinnat.
Graphics.DrawString	Metodi (tulostettavan tai tallennettavan) dokumentin tekstin piirtämiseen ja muotoilemiseen.
ListView	Ohjelmakomponentti nimikkeiden listaukseen.

LIITELUETTELO**LIITE 1.** Geritrim Huonevaraussovelluksen käyttöohjeet

1 JOHDANTO

Tämän opinnäytetyön tavoitteena oli suunnitella, toteuttaa ja asentaa huonevaraus- ja asiakastietojärjestelmä kristiinankaupunkilaiselle geriatriselle kuntoutuskeskus Geritrimille (Selkämeren sairaskotisäätiö).

Asiakasyrityksellä oli tarve saada selkeä ja hyvin toimiva järjestelmä ensisijaisesti huonevarausten kirjaukseen ja seurantaan. Muita tarvittavia liitännäisominaisuuksia olivat muun muassa huonevaraus- ja asiakastietojen hakutoiminnot, johdon laskelmien, kuten huonevarausten kuormitusasteen, raportointi, sekä mahdollisuus tuottaa huonevarausten mukaisia kutsukirjeitä järjestelmästä käsin.

Tässä raportissa käydään läpi työn määrittely, suunnittelu, testaus sekä asennus ja lopuksi pohditaan työn onnistumista ja kehittymismahdollisuuksia.

2 MÄÄRITTELY

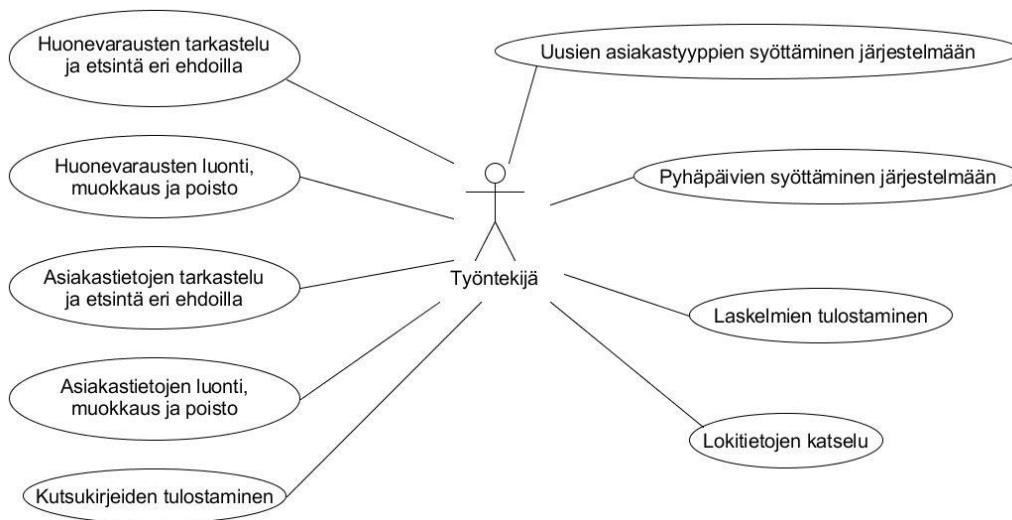
2.1 Vaatimusmäärittely

2.1.1 Yleiskuvaus

Huonevaraus- ja asiakastietojärjestelmä toimii Windows-ympäristössä työpöytäsovelluksena (Windows Form Application) ja sen tehtävä on pitää kirjaa asiakasyrityksen huonevarauskirjauksista sekä näihin liittyvistä liitännäistoimista, kuten asiakastiedoista. Ohjelman päänäkymän on oltava kalenterimainen, jossa huonevaraukset näkyvät aikapalkkeina sen mukaan, mihin vuodepaikkaan ja ajanjaksoon huonevaraus linkittyy.

Ohjelman tarvitsemat tiedot ja kirjaukset tallennetaan pääsääntöisesti samassa tietokoneessa sijaitsevaan MySQL-tietokantaan ja lisäksi joissain tapauksissa käytetään tiedon tallentamista tekstitiedostoihin (samalla tietokoneella). Ohjelma sekä kirjoittaa että lukee tietoja edellä mainituin keinoin.

Ohjelmaa käyttävät Geritrimin työntekijät tietokoneelta, joka on tarkoitettu varta vasten huonevaraus- ja asiakastietojärjestelmän käyttöön. Työntekijät seuraavat, kirjaavat ja muokkaavat huonevarauksia ja asiakastietoja. Lisäksi he voivat hakea huonevaraus-, asiakas-, loki-, sekä laskelmatietoja sekä tulostaa kutsukirjeitä ja tehdä erinäisiä ohjelman ylläpitotoimia, kuten päivittää tietoja tulevista arkipyhistä. Kuvio 1 on käyttötapauskaavio, joka kuvaa sitä, mitä ohjelman käyttäjä voi ohjelmalla tehdä.



Kuvio 1. Käyttötapauskaavio, joka kuvaa ohjelman käyttäjätoiminnot.

2.1.2 Toiminnalliset vaatimukset

Seuraavassa on listattu ohjelmalle asetetut toiminnalliset vaatimukset, jotka tulivat esiin ensimmäisessä ohjelman suunnittelukokouksessa, joka pidettiin syyskuussa 2014:

- Ohjelman tulee toimia itsenäisesti yhdellä, erikseen ko. käyttöä varten hankittavalla Windows-pöytäkoneella.
- Huoneiden varaukset näkyvät kalenterissa, jota voidaan selata sekä eteen- että taaksepäin ajassa, lisäksi voidaan hakea tietty ajankohta kalenterista.
- Kalenterin on oltava ajan tasalla, eli kalenterinäköymän on oltava sellainen, että siinä näkyy kaikki sen hetkiset varaukset.
- Huonevarauksia pitää voida luoda, muokata (kaikkia tietoja, mm. tieto siitä, onko varaus toteutunut, on voitava muuttaa suunnitelmasta toteutuneeksi) ja poistaa.
- Kalenterin on näytettävä varaustilanne noin kuukauden ajalta kerrallaan siten, että kaikki varaukset ko. ajalla näkyvät näyttöruutua vierittämättä.
- Huonevarauksiin kirjataan tiedot asiakkaasta (nimi, henkilötunnus/syntymäaika ja yhteystiedot), huoneen vuorokausihinta, vuodepaikka (huone ja

vuode), status (asukas, itse maksava, leikkaus vai kuntoutus), tieto siitä, onko varus toteutunut vai suunnitelma, tulo- ja lähtöaika sekä mahdolliset erityistoiveet.

- Huonevaraukset näkyvät kalenterissa erivärisinä statuksen mukaan (asukas = keltainen, itse maksava = sininen, kuntoutus = vihreä ja leikkaus = punainen).
- Huonevarauksessa on oltava nähtävillä tieto siitä, koska varaus on luotu ja koska sitä on viimeksi muokattu.
- Aina kun tallennetaan, muokataan tai poistetaan tietoja, on käyttäjältä kysyttävä selkeästi vähintään kertaalleen ”Oletko varma?”.
- Ohjelmaan jää tiedot siitä, milloin mitäkin on tehty. Eli lokitietoihin merkintä tietojen tallennuksista, muokkauksista ja poistoista.

Ohjelman vaatimus oli toimia ensisijaisesti huonevarausten kirjauksessa ja näin ollen ylläpitää tietoja lähinnä huonevarauksista. Asiakkaiden tietoja ylläpidetään asiakasyrityksessä toki muuallakin, ja nimenomaan tarkemmat tiedot asiakkaista sijaitsevat muualla kuin tässä ohjelmassa. Ohjelman toimivuuden ja jouhevuuden (ettei jokaiseen huonevaraukseen tarvitse kirjata kaikkia asiakkaan tietoja aina uudelleen, vaan asiakkaan tiedot voidaan liittää huonevaraukseen erikseen) vuoksi myös asiakastiedot on pystyttävä kirjaamaan järjestelmään. Sitä kautta voidaan sanoa, että toiminnallisia vaatimuksia olivat myös asiakastietojen luonti ja muokkaus (ja myös joissain tapauksissa poistomahdollisuus, josta tarkemmin luvussa 3). Oli myös helpompi ja turvallisempi vaihtoehto, että sellaiset asiakastiedot, jotka tämä sovellus tarvitsee käyttöönsä, ovat eriytettyinä muista asiakastietolähteistä. Asiakastietoihin tulee voida kirjata asiakkaan nimi, henkilötunnus/syntymäaika, yhteystiedot sekä mahdolliset lisätiedot.

Ohjelman suunnittelun edetessä toiminnalliset vaatimukset lisääntyivät laskutoiminnoilla, joissa voitaisiin hakea tietty aikaväli ja laskea toteutuneiden huonevarausten kuormitusaste ja keskimääräinen vuorokausihinta sekä toteutuneille että suunnitelluille huonevarauksille. Lisäksi asiakkaiden tietoihin haluttiin lisätä asiakastyyppejä. Näitä asiakastyyppejä ovat asukkaat, invalidit, veteraanit ja muut. Vielä

myöhemmin ko. vaatimus täsmentyi sellaiseksi, että edellä mainitut neljä asiakastyyppejä ovat ohjelmaan ”kovakoodatut” vakiotyypit ja lisäksi haluttiin mahdollisuus lisätä uusia asiakastyyppejä ohjelman käyttäjän toimesta.

Ohjelmaan haluttiin myös kutsukirjeen tulostustoiminto. Aikaisemmin kutsukirjeet oli tulostettu erikseen siten, että kaikki muuttuvat tiedot, kuten huonevaraustiedot ja asiakkaan tiedot, kirjoitettiin aina jokaiseen kutsukirjeeseen erikseen. Ohjelmalla tulisi voida tulostaa kutsukirje, jossa kutsukirjeen pohjatekstin lisäksi muuttuvat tiedot olisivat jo valmiina. Kirje tulisi voida tulostaa sekä suomen- että ruotsinkielisenä.

Ohjelmassa haluttiin olevan myös hyvät hakutoiminnot, joiden avulla asiakkaita ja huonevarauksia voi hakea nimen ja asiakastyypin mukaan.

Viimeisimpiä toiminnallisia vaatimuksia oli lisäksi asiakastietoihin tieto siitä, kuinka monta päivää kukin asiakas on ollut talossa asukas-statusella kuluvana vuotena ja kuinka monta tällaista päivää on kuluvana vuotena vielä tulossa.

2.1.3 Ei-toiminnalliset vaatimukset

Ohjelman ei-toiminnalliset vaatimukset kohdistuivat ensisijaisesti ohjelman käytön helppouteen ja käyttöliittymän selkeyteen. Lisäksi ohjelman kehittäjän tulisi olla tarpeen vaatiessa tavoitettavissa myös jatkossa, mikäli ohjelman käyttöön tai toimintaan liittyvää avun tarvetta ilmaantuu, tai mikäli ohjelmaa on tarve jatkokehittää.

Ohjelman käyttöliittymän kaikkine toimintoineen on oltava selkeä ja mahdollisimman helppokäyttöinen, eikä ohjelmassa saa olla turhia ja epäselviä toimintoja.

Asiakasyrityksen aikaisempi huonevarausjärjestelmä, joka käsitti taulukkolaskentaohjelmalla suunnitellun pohjan, oli melkoisen sekava ja vaikeakäyttöinen (tiedot erityyppisistä varauksista ja asiakkaiden nimistä sijaitsivat useilla eri välilehdillä). Uuden järjestelmän tulisi olla selkeä parannus edelliseen ja säästää henkilökunnan

aikaa ja hermoja. Tämä tarkoittaa sitä, että uuden ohjelman käyttöliittymän tulisi olla niin selkeä, että huonevarausten kokonaistilanteesta, aina tietyllä aikajaksolla, on nähtävillä kalenterinäkylässä selvä kokonaiskuva ja, että huonevarauksia kuvaavat aikapalkit kertovat oleellisimmat tiedot, kuten huonevarauksen statuksen, sen onko varaus toteutunut ja asiakkaan nimestä ainakin alkuosan, ilman sen tarkempaa yksittäisten varausten tarkistelu.

Kalenterinäkymän on oltava sellainen, että siitä nähdään selkeästi, mikä päivä on kyseessä (päivämäärä, viikonpäivä ja viikkonumero). Sunnuntait ja arkipyhät on oltava esillä eri värillä (punainen) kuin muut päivät. Nämä olivat ensimmäisen suunnittelukokouksen jälkeen esiin tulleita vaatimuksia.

2.2 Toiminnallinen määrittely

2.2.1 Toiminnot

Kalenterinäkymä, joka on ohjelman päänäkyvä ja johon ohjelma avautuu kun se käynnistetään, koostuu kalenterista, joka on kuin taulukko soluineen. Kalenterin ylimpien rivien solut kertovat päivän (päivämäärä, viikonpäivä ja viikkonumero) ja näiden solujen alapuoliset solut ovat ko. päivää kuvaavia, vuodepaikan kertovia soluja. Huonevaraukset tulevat näkymään näissä vuodepaikkaa merkkäavissa soluissa aina sitä mukaa, kun tietokannasta löytyy sellainen huonevaraus, joka vastaa vuodepaikaltaan ja päivämäärältään ko. solua.

Kun ohjelma käynnistetään, kalenterinäkymään latautuu oletuksena näkyvä, joka on noin kuukauden mittainen alkaen sen hetkisestä päivästä. Kalenterinäkylässä on selkeillä paikoilla painonapit, joita klikkaamalla saa ladattua sekä näkyvää jaksoa edeltävän että sitä seuraavan yhtä pitkän näkymän. Lisäksi on painonappi, jota klikkaamalla voi palata oletusnäkymään. On myös mahdollista ladata kalenterinäkymä vapaasti valittavasta päivämäärästä alkaen.

Kun kalenterissa klikataan jotain haluttua päivää ja vuodepaikkaa osoittavaa solua, avautuu ikkuna. Mikäli ko. solu sisältää huonevarauksen, avautuva ikkuna käsittää

tiedot tästä huonevarauksesta ja asiakkaasta. Huonevarausta on mahdollista muokata ja sen voi myös poistaa. Mikäli klikattava solu on tyhjä, avautuvan ikkunan kautta voi luoda uuden huonevarauksen. Avautuvan ikkunan lomakkeessa on esitäytetyt tiedot (päivä ja vuodepaikka), jotka vastaavat klikattua solua.

Uuden huonevarauksen voi luoda myös sovelluksen vasemmassa ylänurkassa olevan valikon kautta (edellisen kuvauksen mukaisesti). Lisäksi kalenterinäköymän valikosta käsin pääsee asiakastieto-ikkunaan, varausten poisto -ikkunaan ja tietojen haku -ikkunoihin (asiakastiedot, varaustiedot, loki, laskelmat) sekä lisäämään pyhäpäiviä ja asiakastyyppejä. Kalenterinäköymän valikon kautta pääsee myös lukemaan sovelluksen käyttöohjetta.

Asiakastiedot-lomakkeessa on valikko, josta voi valita halutun asiakkaan ja nähdä tämän kaikki tiedot yhteystietoineen ja huonevaraustietoineen. Lomakkeessa on myös painike, jota painamalla pääsee asiakastietojen hakuun, jonka kautta asiakastietoja voi hakea täsmennetyin ehdoin. Saman lomakkeen kautta olemassa olevan asiakkaan tietoja voi myös muokata ja poistaa (poistaminen onnistuu vain, mikäli ko. asiakkaalla ei ole huonevarauksia). Lisäksi voi luoda uuden asiakkaan.

Varausten poisto -lomakkeessa voidaan huonevarauksia listata tulopäivän perusteella ja sitten poistaa listattuja huonevarauksia joko yksitellen tai useamman kerrallaan.

Asiakastietojen haku -lomakkeessa voidaan listata joko kaikkien asiakkaiden tiedot tai hakea asiakkaita eri ehdoilla (nimi ja/tai asiakastyppi). Lisäksi listauksen voi tallentaa tekstitiedostoon. Kaksoisklikkaamalla jotain listassa olevaa asiakastietoriviä ko. asiakkaan tiedot avautuvat erilliseen asiakastieto-ikkunaan tarkempaa tarkastelua varten.

Varaustietoja voidaan listata samoin ja samoilla ehdoilla kuin asiakastietojakin, mutta lisäksi voidaan hakea huonevarauksia tulopäivän perusteella siten, että valitaan aikaväli, jolta varaukset noudetaan. Myös varaustiedot voi tallentaa tekstitiedostoon. Kuten asiakastietojen haussa, myös jotain varaustietoriviä klikkaamalla saadaan avattua ko. varaustiedot erilliseen varaustieto-ikkunaan.

Lokitiedot voidaan hakea listaan joko kaikki tai, kuten huonevarauksia, tulopäivän perusteella. Listatut lokitiedotkin voidaan tallentaa tekstitiedostoon. Lokitiedoista nähdään mitä on tehty ja koska, sekä kaikki kyseisen tietueen tiedot.

Laskelmia voidaan hakea siten, että valitaan aikaväli, jolta laskelmat halutaan hakea. Sitten nämä laskelmat voidaan tulostaa raportiksi. Näitä laskelmia ovat toteutuneiden huonevarausten kuormitusaste, toteutuneiden huonevarausten keskimääräinen vuorokausihinta, suunniteltujen huonevarausten keskimääräinen vuorokausihinta, huonevarauspäivien yhteenlaskettu lukumäärä sekä toteutuneiden huonevarauspäivien osuus kaikista huonevarauspäivistä.

Pyhäpäivien lisäys -lomakkeella käyttäjä voi lisätä listaan pyhäpäiviä, jotka osuvat arkiviikonpäiville. Lisäys tapahtuu siten, että valitaan kalenterikomponentista haluttu päivä ja kirjoitetaan lisäksi halutessa jokin selite, kuten vappu. Syötettyjä päiviä voidaan myös poistaa. Kalenterinäkyvässä sunnuntait näkyvät punaisella kun muut päivät näkyvät harmaana. Myös nämä itse lisätyt pyhäpäivät tulevat näkymään kalenterissa punaisella.

Asiakastyypien lisäys -lomakkeessa voidaan asiakastyyppejä lisätä (annetaan lisättävälle asiakastyypille haluttu nimi) tai poistaa. Ohjelmassa on lisäksi vakioasiakastyypit, joita ei voi poistaa. Nämä vakioasiakastyypit ovat: asukkaat, invalidit, veteraanit ja muut.

Kalenterinäkymän valikon ”Ohje”-välilehdeltä löytyvää käyttöohje-valintaa klikkaamalla avautuu pdf-muotoinen sovelluksen käyttöohje. Lisäksi löytyy valinta ”tietoa ohjelmasta”, jota klikkaamalla pääsee lukemaan yleistä tietoa ohjelmasta.

2.2.2 Tiedot ja syötteet

Ohjelman käsittelemät tiedot ovat huonevaraustietoja sekä asiakastietoja. Ohjelmaan voi syöttää uusia tietoja, muokata tietoja ja poistaa tietoja. Lähes kaikki ohjelman käsittelemä tieto sijaitsee tietokannassa, kahta poikkeusta lukuun ottamatta. Nämä poikkeukset ovat toiminto, jossa käyttäjä voi syöttää ohjelmaan pyhäpäiviä

(arkipyhiä), sekä toiminto, jossa käyttäjä voi syöttää ohjelmaan asiakastyyppejä. Näissä kahdessa tapauksessa tiedot tallennetaan tekstitiedostoihin.

Ohjelmassa voi tallentaa monia eri tietoja tekstitiedostoihin. Tekstitiedostoon voi tallentaa yksittäisen asiakkaan tiedot, asiakaslistan, johon voi listata useamman asiakkaan tietoja, huonevaraustietoja listattuna sekä lokitietoja listattuna.

Tietoja voi toki myös tulostaa epäsuorasti, eli esimerkiksi kalenterinäkymän voi tulostaa ”print screen” toiminnon avulla ja kaikki edellä kuvatut tekstitiedostot voi tulostaa (esim. Notepad-ohjelmalla). Johdon laskelmia voi tulostaa suoraan ohjelmasta käsin raportiksi ja samoin voidaan tulostaa kutsukirjeitä sekä suomen- että ruotsinkielisinä.

3 SUUNNITTELU

3.1 Suunnittelussa käytetyt ohjelmat

Huonevaraus- ja asiakastietojärjestelmän suunnittelussa on käytetty seuraavia ohjelmia:

- Tietokantasuunnittelussa (tietokanta ja ER-kaavio) on käytetty ohjelmaa MySQL Workbench 6.2 CE.
- Käyttötapauskaavio on laadittu ohjelmalla UMLet 13.3.
- Huonevaraus- ja asiakastietojärjestelmä (käyttöliittymä- ja ohjelmistosuunnittelu) on toteutettu Microsoft Visual Studio 2010 Professional -ohjelmalla.

3.2 Tietokannan suunnittelu

Koska huonevaraus- ja asiakastietojärjestelmää tultaisiin käyttämään yhdeltä, juuri siihen tarkoitukseen tarkoitettulta, paikalliselta tietokoneelta, myös tietokanta olisi helpointa ja järkevintä asentaa samaan tietokoneeseen. Näin toimittuna ratkaisu

olisi ylläpidon kannalta helpoin, koska kaikki järjestelmän tarvitsemat osat sijaitsevat samalla tietokoneella. Toki ko. tietokoneen rikkoutuessa myös tietokanta jouduttaisiin uudelleen asentamaan, mutta kyseinen toimenpide on varsin nopea ja helppo, ja yhtäläillä tietokannan uudelleen asennustarvetta saattaisi ilmaantua tietokannan ollessa eri sijainnissa. Tietokannan varmuuskopiointi on toki hyvä järjestää niin, että tietokannasta tallennetaan varmuuskopioita johonkin muuhun sijaintiin, esimerkiksi verkkolevylle. Kun tietokanta sijaitsee samassa tietokoneessa, on se myös kustannustehokasta, kun ei tarvita omaa tai joltain ulkopuoliselta taholta ostettua palvelintilaa tietokantaa varten. Lisäksi tietoyhteys tietokannan ja sovelluksen välillä on varmatoiminen, koska järjestelmä ei tarvitse verkkoyhteyttä toimiakseen, ja näin ollen myös tietoturva on parempi, kun ei tarvitse pohtia sovelluksen ja tietokannan välisen yhteyden turvallisuutta.

Tietokantajärjestelmäksi päätettiin valita MySQL ennen kaikkea siksi, että se perustuu avoimeen lähdekoodiin ja on siten ilmainen käyttää. MySQL on myös kevyt asentaa, eikä se syö paljoa resursseja tietokoneesta.

Jotta tietokanta voisi toimia, tarvitsee se tietokantapalvelimen, eli MySQL-serverin. MySQL-serveri asennetaan samaan tietokoneeseen, johon koko huonevaraus- ja asiakastietojärjestelmä asennetaan kaikkine järjestelmän osineen (tästä tarkemmin luvussa 4.2).

Tietokannan suunnittelussa ja laatimisessa käytetään MySQL Workbench 6.2 CE -ohjelmaa, jonka avulla laaditaan ensin relaatiokaavio eli ER-kaavio (Entity Relationship diagram). ER-kaaviolla voidaan kuvata tietokannan tarvitsemat taulut soiluineen, tietotyypeineen ja tietovirtoineen. MySQL Workbench -ohjelmalla voidaan ensin suunnitella tietokanta ER-kaaviolla, jonka jälkeen voidaan ko. kaavion perusteella ajaa MySQL-luontilauseet, joilla tietokanta saadaan perustettua. Kuvio 2 on toteutettu ER-kaavio.

Huonevaraus- ja asiakastietojärjestelmän tietokantaan tarvittiin neljä tietokantataulua. Taulu joka kuvaa huonevarauksen, taulu joka kuvaa asiakkaan, taulu joka kuvaa huoneen, sekä taulu, joka kuvaa lokimerkinnän.

Huonevaraustaulun on sisällettävä yksilöllisen varauksen tunnisteiden (juokseva id-numerointi), tulopäivän, lähtöpäivän, statuksen, huonetunnuksen, vuodenumeron, erikoistoiveet, vuorokausihinnan ja asiakkaan tiedot. Lisäksi varauksessa täytyy olla tieto siitä, onko varaus toteutunut vai suunnitelma sekä milloin varaus on luotu ja milloin sitä on viimeksi muokattu.

Asiakastaulun on sisällettävä yksilöllisen asiakastunnisteiden (juokseva id-numerointi), etu- ja sukunimen, syntymäajan tai henkilöturvattunnuksen, yhteystiedot, omaisen tiedot ja muita tietoja.

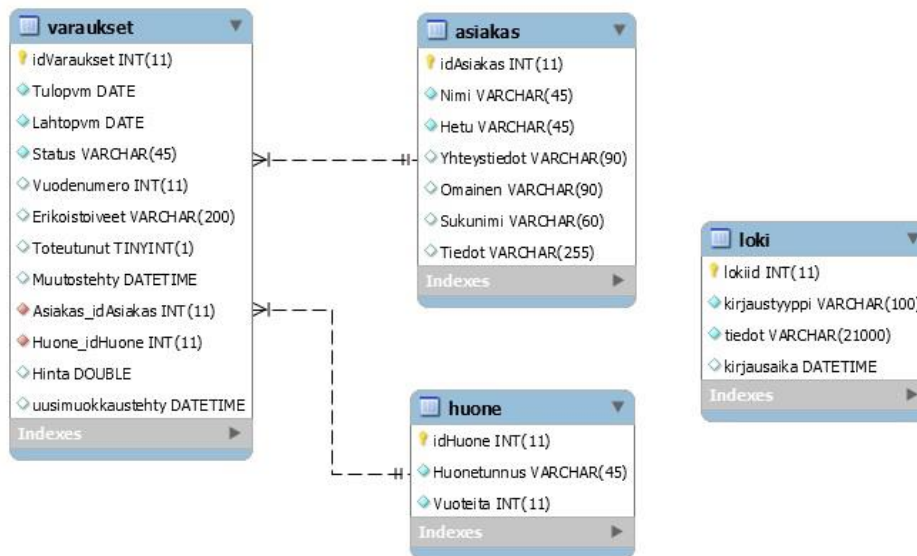
Huonetaulun on sisällettävä yksilöllisen huonetunnisteiden (juokseva id-numerointi), huonetunnuksen ja vuoteiden lukumäärän.

Lokitaulun on sisällettävä yksilöllisen lokitunnisteiden (juokseva id-numerointi), kirjaustyypin, kirjausajan sekä kaikki oleelliset tiedot siitä mitä on tehty.

Asiakastaulun tietovirta on yhdensuhde moneen suhteessa varaustauluun, koska asiakkaalla voi olla useita huonevarauksia, mutta yhdellä huonevarauksella voi olla vain yksi asiakas. Samoin huonetaulun tietovirta on yhdensuhde moneen suhteessa varaustauluun, koska yhdellä huoneella ja vuodepaikalla voi olla monia varauksia, mutta yhdellä varauksella voi olla vain yksi huone ja vuodepaikka.

Lokitaulu on täysin erillään muista tauluista (ei yhteisiä tietovirtoja). Lokitauluun tehtävät merkinnät toteutetaan siten, että sovelluksesta käsin ajetaan lokitauluun aina uusi tieto samalla, kun jotain kirjoitetaan varaus- tai asiakastauluun, tai jos niistä jotain poistetaan.

Tietokannan soluille määritetään tietotyypit sen mukaan, minkälaista tietoa ne pitävät sisällään ja miten tietoa käsitellään. Kun käsitellään kokonaislukuja, tietotyyppi käy INT. Esimerkiksi rahasummia ja muita desimaaleja sisältäviä lukuja käsiteltäessä tietotyyppi sopii DOUBLE. Kun käsitellään merkkijonoja, tietotyyppi on VARCHAR. Päivämäärien tietotyyppi on DATE.

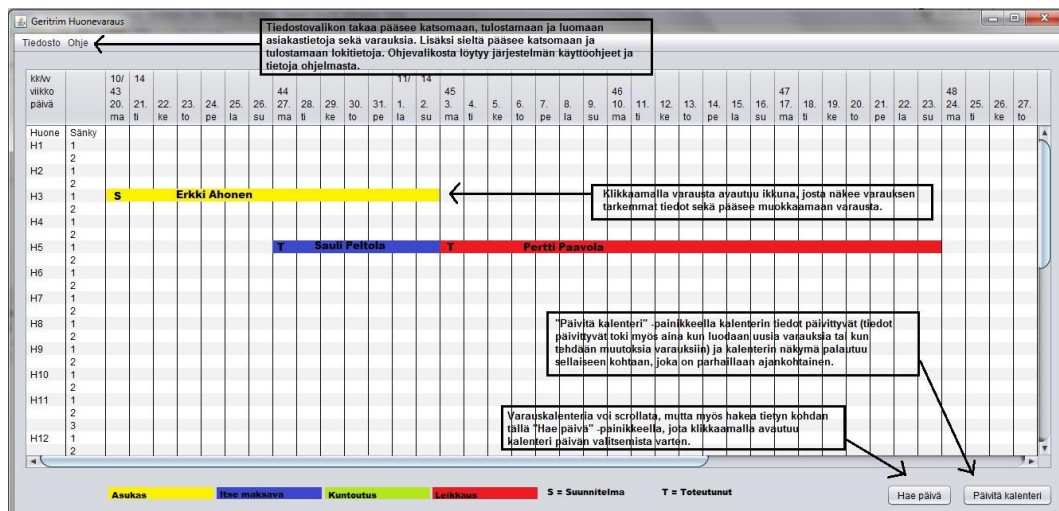


Kuvio 2. ER-kaavio tietokannan tauluista.

3.3 Käyttöliittymän suunnittelu

3.3.1 Suunnitteluperiaatteet ja suunnittelun alkuvaiheet

Käyttöliittymän suunnittelun ensivaiheet on toteutettu raakaluonnoksina paperille. Ensimmäisiä käyttöliittymään ja käyttöön vaikuttavia suunnitelmia olivat käyttötapauskaavio, josta laadittiin UMLet 13.3 -ohjelmalla siistimpi versio (**Kuvio 1.**), ja kalenterinäkömän mallinnus, josta piirrettiin kuva Microsoft Paint -kuvankäsittelyohjelmalla (**Kuvio 3.**). Käyttötapauskaaviosta nähdään, mitä kaikkea Geritrimin työntekijä voi huonevaraus- ja asiakastietojärjestelmällä tehdä. Kalenterinäkömän ensimmäinen mallinnuskuva antoi sekä suunnitteluun että asiakkaalle alkutietoa siitä, minkä tyylistä käyttöliittymää lähdetään suunnittelemaan.



Kuvio 3. Kalenterinäkömän ensimmäinen mallinnuskuva. Kuvassa huonevaraukset näkyvät statuksen mukaan erivärisinä aikapalkkeina. Aikapalkeista näkee asiakkaan nimen ja sen, onko varaus toteutunut vai suunnitelma. Tiedostovalikon ”ta-kaa” löytyy lisätoimintoja, ja kalenterinäkömäksi voi hakea halutun ajankohdan.

Seuraavaksi käyttöliittymäsuunnittelua jatkettiin suoraan samalla ohjelmalla, jolla koko ohjelman toteutus tapahtuu, eli erillisiä käyttöliittymäsuunnitteluun tarkoitettuja ohjelmia ei käytetty. Apuna käytettiin ainoastaan perinteistä paperille mallinusta antamaan hieman suuntaviivoja käyttöliittymän hahmottelussa.

Ohjelman toteutus, kuten myös käyttöliittymän suunnittelu suurimmilta osin, tapahtui Visual Studio 2010 Professional -ohjelmistolla. Sovellus, jota lähdettiin suunnittelemaan, on työpöytäsovellus, joten Visual Studion projektimalliksi otettiin ”Windows Forms Application”, jossa jokainen käyttöliittymän ikkuna/lomake on oma forminsa (Form). Seuraavaksi aloitettiin näiden lomakkeiden suunnittelu. Ensimmäisenä työn alla oli ohjelman päänäkymä, eli kalenterinäkömä, johon ohjelma avautuu kun se käynnistetään.

Kalenterinäkömän suunnittelun lähtökohdana oli se, että tarvittaisiin jonkinlainen taulukkopohja, joka toimisi kalenterina. Visual Studiassa on ”DataGridView” -komponentti, joka toimisi hyvin tässä tarkoituksessa.

3.3.2 Käyttöliittymäkuvaukset

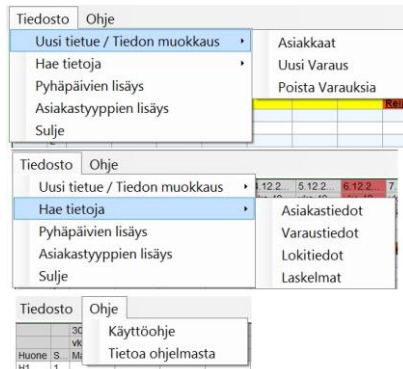
(Käyttöliittymäkuvissa esiintyvät asiakastiedot ovat fiktiivisiä.)

Kuvio 4 on sovelluksen kalenterinäkymä, joka on kuva tilanteesta, jossa ohjelma on jo toiminnassa. Ilman koodia ”DataGridView” -komponentin ”takana” kalenteri olisi vain tyhjä neliskulmainen laatikko, joten on kuvaavampaa näyttää käyttöliittymäkuvat sellaisina, kun ne näyttävät ohjelman ollessa valmis ja toiminnassa. Lisäksi kalenterinäkymässä on lomakkeen alaosassa painonapit, joilla kalenterinäkymäksi voi ladata halutun ajanjakson, ja lomakkeen yläkulmassa on valikko, josta pääsee avaamaan ohjelman muita ikkunoita. Kaikki vuodepaikat näkyvät ruutua vierittämättä, aina neljän viikon ajalta kerrallaan. Ensimmäiset solurivit kertovat päivämäärän, viikkonumeron sekä viikonpäivän. Solut, jotka merkkäavat sunnuntaipäiviä tai muita pyhäpäiviä, on värjätty punaisella värillä, jotta ne erottuvat arkipäivistä selkeästi. Aikapalkit ovat tietyn värisiä statuksen mukaan. Samoin asiakkaan nimen näkyvän osan fontti on tummennettu ja alleviivattu, mikäli varaus on toteutunut. Näistä informoidaan käyttäjää lomakkeen vasemmassa alakulmassa, jossa on näkyvillä myös asiakasyrityksen logo. Kun hiiren kohdistin viedään jonkin solun päälle, ko. solun teksti näytetään kokonaisuudessaan.

The screenshot shows a calendar application window titled "Gentrim Huonevaraus". The main area is a grid with columns for dates from 7.12.2015 to 31.12.2015 and rows for rooms (Huone) H1 through H19. Each cell in the grid contains a date and a colored bar representing a booking. The colors correspond to different booking statuses: red for "ASUKAS" (occupied), green for "KUNTOOUTUS" (maintenance), and blue for "LEIKKAUS" (cutting). The text in the cells is bold and underlined, indicating that the booking is confirmed. The status bar at the bottom contains several sections: "Statusvärit" (Status colors) with a legend for "ASUKAS", "KUNTOOUTUS", and "LEIKKAUS"; "Toteutuneissa varauksissa varausID ja asiakkaan nimi ovat lihavoitu ja alleviivattu" (In confirmed bookings, booking ID and customer name are bold and underlined); "Kalenterinäkymän selaustoiminnot" (Calendar view navigation) with a button to "Lataa tämänhetkinen ajankohta (tästä päivästä lähtien 4 viikkoa eteenpäin)"; "Valitse kalenterinäkymän ajankohta (valitusta päivästä lähtien 4 viikkoa eteenpäin):" with a date selector set to "7. joulukuuta 2015"; and two buttons: "Selaa kalenteria 4 viikkoa taaksepäin" and "Selaa kalenteria 4 viikkoa eteenpäin".

Kuvio 4. Kalenterinäkymä.

Valikon kautta (**Kuvio 5.**) pääsee avaamaan ohjelman eri toimintoikkunoita (lomakkeita). Seuraavaksi kerrotaan kaikkien näiden lomakkeiden toiminnot.



Kuvio 5. Valikon rakenne.

Kuvio 6 on ”Asiakkaat” -ikkuna, jolle pääsee kalenterinäkymän valikon kautta. Tällä lomakkeella voidaan katsella asiakastietoja. Lisäksi uusia asiakastietueita voidaan luoda, niitä voidaan muokata ja niitä voidaan poistaa. ”Valitse asiakas” -valikosta nähdään kaikki asiakkaat listattuna sukunimen mukaan aakkosjärjestyksessä. Tässä listauksessa asiakkaasta nähdään nimi, syntymäaika/henkilöturvatus sekä asiakasnumero. Listaan voi kirjoittaa sukunimen alkua ja painaa sitten näppäimistön nuolinäppäintä alaspäin, jolloin asiakaslistaan noudetaan ensimmäinen nimi, joka vastaa kirjoitettua. Jos haluaa käyttää tarkempaa asiakashakua, voi painaa painiketta ”Laaja asiakashaku”, jolloin avautuu ”Asiakaslista” -ikkuna (**Kuvio 7.**). Asiakaslista-lomakkeella asiakkaita voidaan hakea eri ehdoilla. Voidaan joko listata kaikki asiakkaat tai hakea asiakkaita nimen ja/tai asiakastyypin mukaan. Listatuista asiakkaista voidaan haluttua asiakasriviä kaksoisklikata, jolloin ko. asiakkaan tiedot siirtyvät valituiksi ”Asiakkaat” -ikkunassa. Myös asiakaslistaan pääsee kalenterinäkymän valikon kautta.

”Asiakkaat” -lomakkeella nähdään kaikki asiakkaan tiedot. Näitä ovat asiakasID, etunimi, sukunimi, henkilötunnus (tai syntymäaika), yhteystiedot, lähiomaisen tie-

dot ja ”muuta tietoja”. Lisäksi asiakkaalle voidaan valita asiakastyyppi. Lomakkeelta nähdään myös asiakkaan kaikki huonevaraukset listattuna siten, että huonevaraukset eritellään sen mukaan, ovatko varaukset menneitä, tulevia vai onko asiakas talossa juuri tällä hetkellä. Nähtävillä on myöskin asukas-statusella käytetyt ja tulevat päivät kuluvana vuotena. Lomakkeen alaosassa olevilla painikkeilla voidaan tyhjentää lomakkeen kentät (kun aloitetaan uuden asiakastietueen kirjoittaminen), tallentaa uudet tai muokatut tiedot tai poistaa asiakastietue.

The screenshot shows a window titled "Asiakkaat" with the following fields and sections:

- Valitse asiakas:** A dropdown menu showing "Meikäläinen Matti --- HETU: 101020-047Z --- AsiakasNro: 25".
- AsiakasID:** A text field containing "25" and a button labeled "Laaja asiakashaku".
- Etunimi:** A text field containing "Matti".
- Sukunimi:** A text field containing "Meikäläinen".
- Henkilötunnus:** A text field containing "101020-047Z".
- Osoite (kirjoita tähän ainoastaan lähiosoite ja postiosoite allekkain):** A text area containing "Suomisillantie 1 a 5" and "06200 Porvoo".
- Lähimaisen tiedot:** A text area containing "Maija Meikäläinen (vaimo)" and "puh. 050 1234 432".
- Asiakastyyppi:** A dropdown menu showing "Invalidit (I)".
- Muita tietoja (kirjoita tähän myös muut yhteystiedot esim. puh):** A text area containing "Asiakkaan_Tyyppi:I. puh. 050 5678 765".
- Asiakkaan huonevaraukset:** A scrollable list showing:
 - ASIAKAS ON TÄLLÄ HETKELLÄ TALOSSA:
 - 1. varaus | Aika: 1.12.2015 - 11.12.2015 | Huone: H4 | Sänky: 2 | Status: itse maksava
 - TULEVAT HUONEVARAUKSET:
 - 2. varaus | Aika: 14.12.2015 - 23.12.2015 | Huone: H4 | Sänky: 2 |
- Asukas-statusella käytetyt päivät tänä vuonna:** A text field containing "0".
- Asukas-statusella tulevat päivät tänä vuonna:** A text field containing "0".
- At the bottom, there are three buttons: "Tyhjennä kentät" (grey), "Tallenna asiakastiedot" (green), and "Poista" (red).

Kuvio 6. Asiakkaat-ikkuna.

Hakutoiminnot: Sukunimi tai etunimi: Asiakastyypit: Veteraanit (V) **Hae yhdellä tai molemmilla hakuehdolla** **Tallenna lista tiedostoon**

Listaa kaikki asiakkaat

Kaksoisklikkaamalla haluttua riviä, ko. asiakkaan tiedot siirtyvät joko "Asiakkaat"-ikkunaan tai "Uusi varaus"-ikkunaan, mikäli jompi kumpi niistä ikkunoista on avoinna.

ASIAKAS_ID	NIMI	HETU	YHTEYSTIEDOT	LÄHIOMAISEN TIEDOT
13	Anttila Aku	11.12.1930	Paratistie 1021100 Naantali	-
15	Herttonen Pelle	23051919	Veljeskyläntie 1426200 Rauma	Liisa Nieminen (tytär), puh. 010001111

Kuvio 7. Asiakaslista.

Kuvio 8 on ”Uusi varaus” -ikkuna. Tälle lomakkeelle päästään sekä kalenterinäkömään valikon kautta että klikkaamalla tyhjää solua (johon voidaan asettaa huonevaraus) kalenterinäkömään kalenterissa. Asiakas valitaan joko suoraan ”Valitse asiakas” -valikosta tai laajan asiakashaun kautta, kuten asiakkaat-ikkunassakin. Lisäksi valitaan varauksen aikaväli. Mikäli tälle lomakkeelle tullaan kalenterinäkömään jonkin tyhjän solun kautta, tulo- ja lähtöpäivät ovat oletuksena ko. solua vastaavat päivät. Samoin vuodepaikka (huone ja sänky) on klikattua solua vastaava. Valitaan myös status, kirjoitetaan mahdollisia erityistoiveita varaukseen liittyen, varauksen vuorokausihinta, sekä valitaan, onko varaus toteutunut vai suunnitelma.

Uusi varaus

Valitse asiakas: **Laaja asiakashaku**

Tulopäivä: 23. marraskuuta 2015 **Lähtöpäivä:** 23. marraskuuta 2015

Status:

Huone: H4 **Sänky:** 2

Erityistoiveet:

Hinta €/vrk:

Varaus on toteutunut (toteutunut = kyllä, suunnitelma = ei):

Tallenna varaus

Kuvio 8. Uusi varaus.

Kun klikataan kalenterinäkylässä solua, joka sisältää varauksen, aukeaa ”Varauksen tiedot” -ikkuna (**Kuvio 9.**). Myös varauslistan (**Kuvio 14.**) kautta saa haluttua varausta kaksoisklikkaamalla avattua ko. ikkunan. Tämä lomake näyttää sen varauksen tiedot, jota merkkavaa solua klikattiin. Lomakkeelta nähdään myös, koska varaus on luotu, ja jos sitä on muokattu, koska viimeisin muokkaus on tehty. ”Näytä asiakkaan kaikki tiedot” -painiketta painamalla avautuu ”Asiakkaan tiedot” -ikkuna (**Kuvio 10.**). Lisäksi voidaan tulostaa kutsukirje (**Kuviot 11. ja 12.**) joko suomen- tai ruotsinkielisenä tai sekä että. Varauksen tietoja voidaan muuttaa, jonka jälkeen painetaan ”Tallenna muutokset”, tai varaus voidaan poistaa.

Varauksen tiedot

Varausno: 63 Varaus on luotu 7.12.2015 22:21:29
Etunimi: Matti Varausta on muokattu 7.12.2015 22:23:04
Sukunimi: Meikäläinen Tulosta kutsukirje
Henkilötunnus: 101020-047Z Näytä asiakkaan kaikki tiedot
Tulopäivä: 15. joulukuu ta 2015 Lähtöpäivä: 23. joulukuu ta 2015
Status: kuntoutus
Huone: H4 Sänky: 2
Erityistoiveet: Laktositon ruokavalio.
Hinta €/vrk: 220
Varaus on Suunnitelma / Toteutunut: Toteutunut
Tallenna muutokset Poista

Kuvio 9. Varauksen tiedot.

Asiakkaan Tiedot

AsiakasID:

Etunimi:

Sukunimi:

Henkilötunnus:

Yhteystiedot:

Lähimmäisen tiedot:

Muita tietoja:

Asiakkaan huonevaraukset:

ASIAKAS ON TÄLLÄ HETKELLÄ TALOSSA:
Aika: 1.12.2015 - 11.12.2015 | Huone: H4 | Sänky: 2 | Status: itse maksava

TULEVAT HUONEVARAUKSET:
Aika: 15.12.2015 - 23.12.2015 | Huone: H4 | Sänky: 2 | Status: kuntoutus

Asukas-statusella käytetyt päivät tänä vuonna:

Asukas-statusella tulevat päivät tänä vuonna:

Tallenna tiedostoon

Kuvio 10. Asiakkaan tiedot.

Kutsukirje

Tulosta suomenkielinen kutsukirje

Tulosta ruotsinkielinen kutsukirje

Kuvio 11. Kutsukirjeen tulostuskielen valinta.

Matti Meikäläinen
Suomisillantie 1 a 5
06200 Porvoo

GERITRIM
Selkämeren sairauskoti
Lapväärtintie 10
64100 KRISTIINANKAUPUNKI

HYVÄ ASIAKAS

Teille on varattu paikka Geritrimissä ajalla 15.12.2015 - 23.12.2015.

Pyydämme Teitä saapumaan 15.12.2015 klo 08.00.

Lähtöpäivänä Teillä on oikeus ruokailuun klo 16.00 ennen lähtöä.

Lähetämme oheisena esitietokyselyn. Olkaa ystävällinen ja vastatkaa kysymyksiin huolellisesti sekä palauttakaa lomake meille mahdollisimman pian.

Ottakaa mukaanne:

- lääkkeet, lääkemääräykset, lääkeannostelija (dosetti jos on) + lista käyttämistänne lääkkeistä
- ulkoiluasu, verryttelypuku ja uimahousut sekä vaihtovaatteita
- parranajovälineet, apuvälineet (esim. kynärsauvat, keppi)
- Kela-kortti

Mikäli ette voi saapua Teille varattuna aikana, ilmoittakaa siitä välittömästi puh. 06-2218584.

Tervetuloa meille!

Kristiinankaupunki 7.12.2015



GERITRIM®

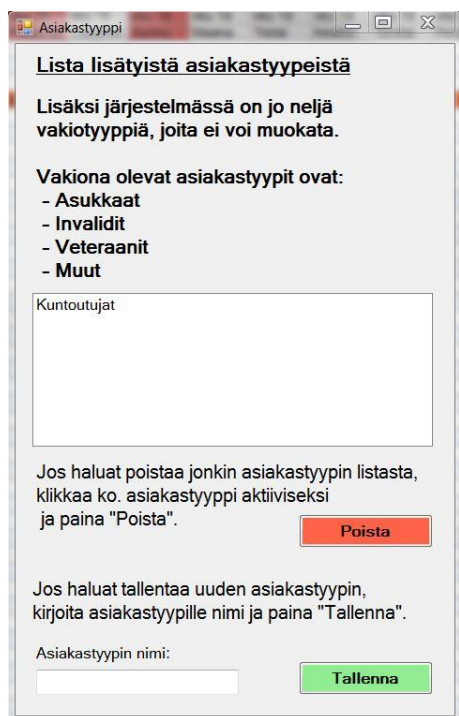
Hyvinvointia, kuntoutusta ja elämäniloa!

Lapväärtintie 10 • 64100 Kristiinankaupunki • Puh: 06-2218 585 • S-posti: info@geritrim.fi • www.geritrim.fi

Kuvio 12. Kutsukirje.

Kuvio 13 on ”Asiakastyypit” -ikkuna, joka löytyy kalenterinäkömän valikosta. Sovelluksessa on neljä lähdekoodiin koodattua vakioasiakastyyppeä (asukkaat, invalidit, veteraanit ja muut). Näiden vakiotyyppien lisäksi ohjelman käyttäjä voi itse lisätä asiakastyyppejä ohjelmaan. Asiakastyypin voi lisätä kirjoittamalla asiakastyyp-

pille nimi ja tallentamalla se. Lisätyt asiakastyypit tulevat näkyviin listaukseen. Lisätyjä asiakastyyppejä voi myös poistaa. Asiakastyypin avulla asiakkaita voidaan jaotella loogisesti, ja asiakkaita voidaan myös hakea näiden asiakastyypin mukaan.



Kuvio 13. Asiakastyypit.

Kuvio 14 on "Varauslista" -ikkuna, jonka saa esiin kalenterinäköymän valikosta. Varauslistassa huonevarauksia voidaan listata eri ehdoilla, tai kaikki varaukset voidaan hakea listaan. Listan voi halutessaan tallentaa tekstitiedostoksi. Kaksoisklikkaamalla halutun varauksen riviä saadaan avattua ko. varauksen tiedot tarkempaa tarkastelua, muokkausta tai vaikkapa kutsukirjeen tulostusta varten "Varauksen tiedot" -ikkunaan (**Kuvio 9**).

ID_VARAUKSET	ID_LASIAKAS	NIMI	HETU	TULOPÄIVÄ	LAHTOPÄIVÄ	STATUS	HUONE	SÄNKY	HINTA/VRK	TOTEUTUNUT	ERI
62	25	Meikäläinen Matti	101020-047Z	1.12.2015	11.12.2015	ise maksava	4	2	200	Kyllä	Lakt
57	16	Puheloinen Nalle	101010-XXXX	3.12.2015	10.12.2015	leikkaus	PS	3	190	Kyllä	-
56	15	Herttonen Pelle	23051919	7.12.2015	21.12.2015	kuntoutus	14	1	230	Kyllä	-
61	17	Virtanen Roope	101010	7.12.2015	9.12.2015	ise maksava	10	2	200	Kyllä	-

Kuvio 14. Varauslista.

Kalenterinäkömään valikosta löytyy myös ”Laskut” (**Kuvio 15.**). Tämän toiminnon avulla voidaan laskea halutulta aikaväliltä huonevarausten kuormitusastetta, toteutuneiden huonevarausten keskimääräistä vuorokausihintaa, suunniteltujen huonevarausten keskimääräistä vuorokausihintaa sekä nähdä, kuinka monta huonevarauspäivää ko. aikavälillä on ja kuinka monta prosenttia näistä päivistä on toteutuneita. Lisäksi laskut voidaan tulostaa raportiksi (**Kuvio 16.**).

Valitse aikaväli jolta haluat laskelmat

Ensimmäinen päivä: 1. heinäkuuta 2015

Viimeinen päivä: 18. marraskuuta 2015 **Laske**

Tulokset:

Toteutuneiden huonevarausten kuormitusaste: 2,849 %

Toteutuneiden huonevarausten keskimääräinen vuorokausihinta: 143,708 €

Suunniteltujen huonevarausten keskimääräinen vuorokausihinta: 250 €

Huonevarauspäiviä yhteensä 91, joista toteutuneita 98 %

Tulosta

Kuvio 15. Laskut.



GERITRIM LASKELMAT -RAPORTTI Ajalta 01-07-2015 - 18-11-2015

Toteutuneiden huonevarauksien kuormitusaste: 2,849 %

Toteutuneiden huonevarauksien keskimääräinen vuorokausihinta: 143,708 €

Suunniteltujen huonevarauksien keskimääräinen vuorokausihinta: 250 €

Huonevarauspäiviä yhteensä 91, joista toteutuneita 98 %

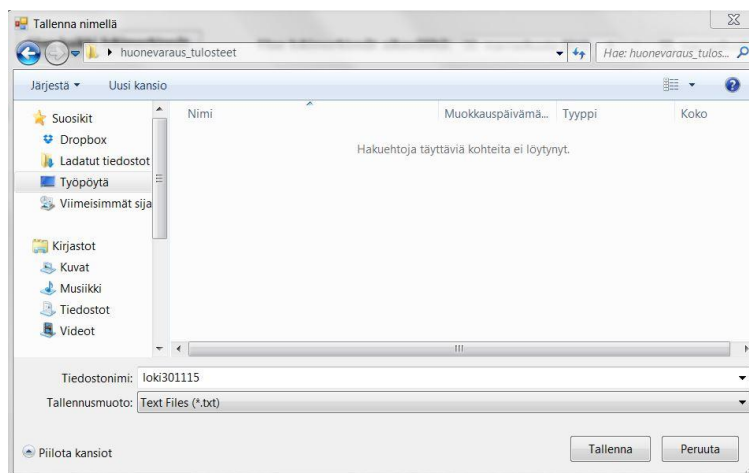


Kuvio 16. Laskelmat-raportti.

Kalenterinäköymän valikon ”Loki” -ikkunan (**Kuvio 17.**) avulla nähdään kaikki ohjelmassa tehdyt toimet, jotka ovat muuttaneet asiakas- tai varaustietueita jollain tavalla. Lokiin tulee merkintä aina, kun luodaan uusi asiakas tai uusi varaus tai kun asiakas- tai varaustietoja muutetaan tai poistetaan. Lokitiedot voidaan listata kerralla kaikki, tai määrittäen aikaväli, jolta lokitiedot noudetaan. Listan voi myös tulostaa tekstitiedostoksi (**Kuviot 18. ja 19.**).

LOKI_ID	KIRJAUSTYYPPI	KIRJAUSAIKA	TIETUEEN KAIKKI TIEDOT
433	Varausta Muokattu	17.11.2015 10:44:21	VarausID: 27 Tulopv: 19.5.2015 Lahtopv: 31.5.2015 Status: asukas Huone: H7 Sänky: 1 Varaus toteutunut Toteutunut Enhytystoiveet: - Vuorokausihinta: 300 AsiakasID: 15 Nimi: Herman
434	Uusi Varaus	17.11.2015 4:35:57	VarausID: 54 Tulopv: 2015-11-18 Lahtopv: 2015-12-06 Status: asukas Huone: H2 Sänky: 2 Varaus toteutunut Ei Enhytystoiveet: - Vuorokausihinta: 250 AsiakasID: 18 Nimi: Rovvo Roope
435	Uusi Varaus	17.11.2015 4:39:43	VarausID: 55 Tulopv: 2015-11-02 Lahtopv: 2015-12-31 Status: itse maksava Huone: H6 Sänky: 1 Varaus toteutunut Kyllä Enhytystoiveet: - Vuorokausihinta: 220 AsiakasID: 13 Nimi: Aekki
436	Uusi Varaus	17.11.2015 4:40:41	VarausID: 56 Tulopv: 2015-11-24 Lahtopv: 2015-12-13 Status: kuntolous Huone: H14 Sänky: 1 Varaus toteutunut Ei Enhytystoiveet: - Vuorokausihinta: 230 AsiakasID: 15 Nimi: Hermanni
437	Uusi Varaus	17.11.2015 4:41:30	VarausID: 57 Tulopv: 2015-12-03 Lahtopv: 2016-01-10 Status: leikkauk Huone: PS Sänky: 3 Varaus toteutunut Kyllä Enhytystoiveet: - Vuorokausihinta: 190 AsiakasID: 16 Nimi: Puh Nalle
438	Varausta Muokattu	17.11.2015 4:45:38	VarausID: 55 Tulopv: 2.11.2015 Lahtopv: 31.12.2015 Status: itse maksava Huone: H7 Sänky: 1 Varaus toteutunut Toteutunut Enhytystoiveet: Laktoositon ruokavalio Vuorokausihinta: 220 AsiakasID: 15 Nimi: Hermanni Pelle HETU: 23456789 Yhteyshetot: sirkusmaalmatvöll Lähimmäisen tiedot: - Muita tietoja: Asiakkaan_Tyyppi V. Asiakkaan huonevaraukset: MENNEET
439	Asiakastieto Muokat.	18.11.2015 12:39:19	

Kuvio 17. Loki.

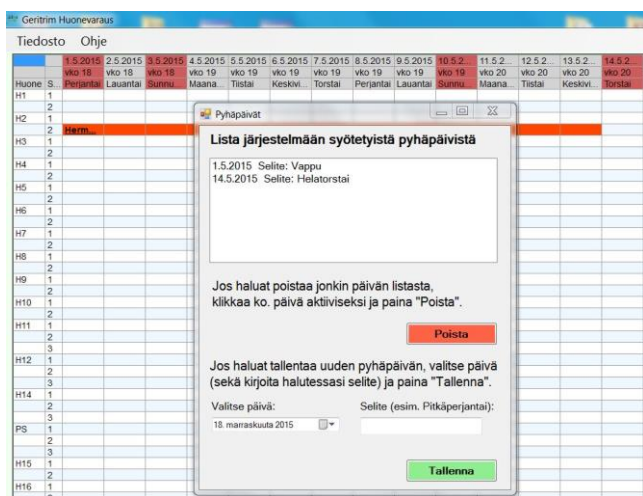


Kuvio 18. Lokitietojen tallennus.

ID	KIRJAUSTYYPPI	KIRJAUSAIKA	TIETUEEN KAIKKI TIEDOT
1	Uusi Astakas	10.4.2015 12:17:35	AsiakasID: 1 Nimi: Jeromen Heikki HETU: 120
2	Uusi Varaus	10.4.2015 12:28:03	VarausID: 1 Tulopv: 2015-03-02 Lahtopv: 201
3	Varausta Muokattu	10.4.2015 12:30:29	VarausID: 1 Tulopv: 2.3.2015 Lahtopv: 31.11
4	Varausta Muokattu	10.4.2015 12:32:06	VarausID: 1 Tulopv: 2.3.2015 Lahtopv: 10.4.
5	Uusi Astakas	10.4.2015 12:36:53	AsiakasID: 2 Nimi: Wikkola Osmo HETU: 0603
6	Uusi Varaus	10.4.2015 12:39:36	VarausID: 2 Tulopv: 2015-04-27 Lahtopv: 201
7	Uusi Astakas	10.4.2015 12:52:26	AsiakasID: 3 Nimi: Dahlback Atle HETU: 2211
8	Uusi Varaus	10.4.2015 12:53:42	VarausID: 3 Tulopv: 2015-01-11 Lahtopv: 201
9	Varausta Muokattu	10.4.2015 12:54:53	VarausID: 1 Tulopv: 2.3.2015 Lahtopv: 31.11
10	Varausta Muokattu	10.4.2015 1:04:24	VarausID: 1 Tulopv: 2.3.2015 Lahtopv: 31.11
11	Varausta Muokattu	29.4.2015 9:59:26	VarausID: 3 Tulopv: 11.1.2015 Lahtopv: 31.11
12	Varausta Muokattu	29.4.2015 10:43:13	VarausID: 1 Tulopv: 2.3.2015 Lahtopv: 31.11
13	Varausta Muokattu	29.4.2015 10:46:54	VarausID: 2 Tulopv: 27.4.2015 Lahtopv: 4.5.
14	Varaus Poistettu	29.4.2015 3:04:32	VarausID: 3 Tulopv: 11.1.2015 Lahtopv: 31.11

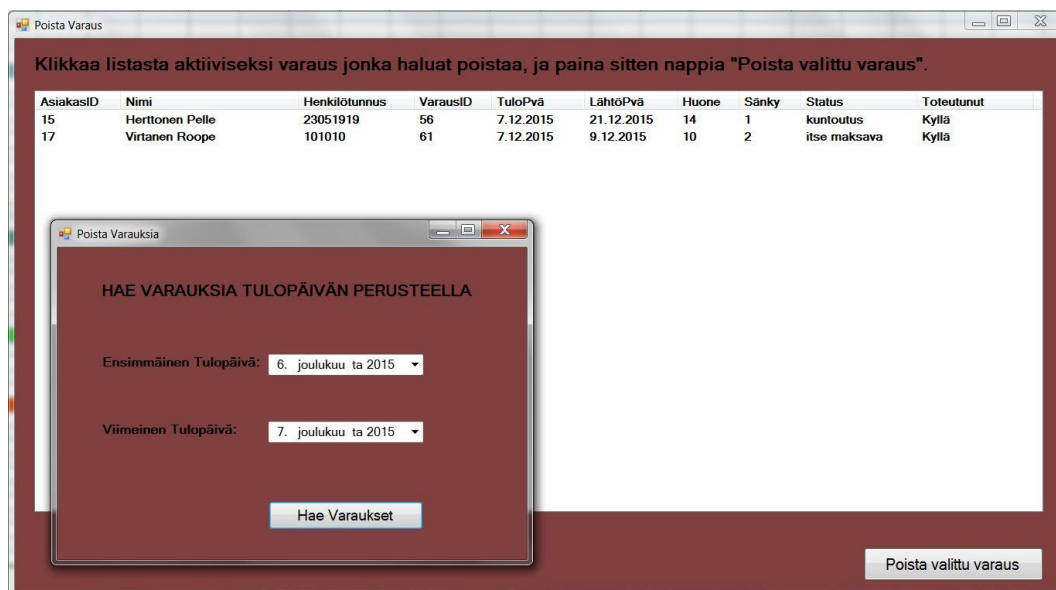
Kuvio 19. Lokitiedot tekstitiedostossa.

Kuvio 20 on ”Pyhäpäivät” -ikkuna (kalenterinäkymän valikosta). Tällä lomakkeella voidaan ohjelmaan syöttää pyhäpäiviä (arkipyhät, esim. pitkäperjantai). Pyhäpäivä syötetään valitsemalla päivä ja kirjoittamalla tälle halutessa jokin selite (esim. pitkäperjantai). Kun päivä tallennetaan, se ilmestyy listaan ja kalenterinäkymässä ko. päivä värjätään punaisella värillä, kuten sunnuntaipäivätkin. Listasta voi myös poistaa lisättyjä pyhäpäiviä.



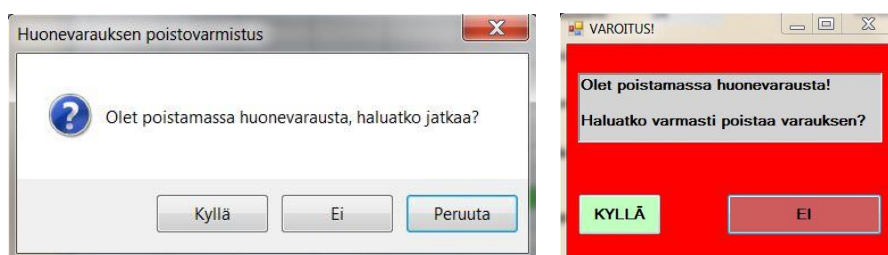
Kuvio 20. Pyhäpäivät. Kuvassa lisätyt pyhäpäivät näkyvät kalenterinäkymässä sunnuntaipäivien tavoin punaisella värillä.

Ohjelmassa huonevarauksia voidaan poistaa myös muuta kautta kuin ”Varauksen tiedot” -lomakkeen kautta. Ohjelmassa on ”Poista varaus” -lomake, jolle haetaan varaukset tietyltä aikaväliltä, ja sen jälkeen varauksia voidaan valita joko yksi tai useampi kerrallaan poistettavaksi (**Kuvio 21.**).



Kuvio 21. ”Poista Varaus” -lomakkeella varauksia voidaan poistaa useampiakin kerralla.

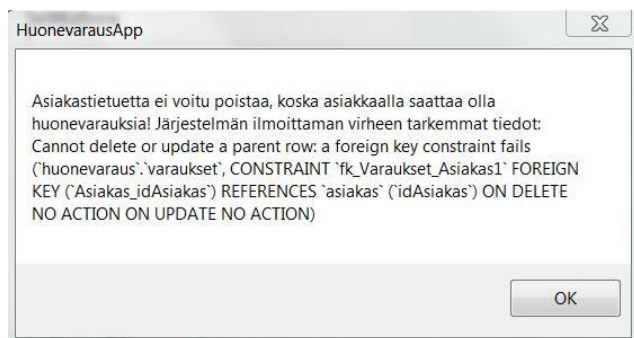
Aina, kun jotain tietoja tallennetaan, muokataan ja poistetaan, ennen toiminnon loppuunsaattamista kysytään ohjelman käyttäjältä varmistukset ko. toimenpiteelle. Varmistuksista on esimerkkinä Kuvio 22.



Kuvio 22. Tietojen poistovaroituksia. Tässä käyttäjältä kysytään kahteen kertaan, haluaako hän varmasti poistaa huonevarauksen.

On tilanteita, joissa ohjelman käyttäjä ei voi poistaa jotain tietuetta. Näin on esimerkiksi silloin, jos sellaista asiakastietuetta, jolle on linkitettyjä varauksia, ollaan

poistamassa (**Kuvio 23.**). Nämä asiakkaan huonevaraukset voivat olla menneitä tai tulevia, tai asiakas voi olla talossa juuri tällä hetkellä.



Kuvio 23. Asiakastietuetta ei voi poistaa, jos asiakkaalla on huonevarauksia.

3.4 Ohjelmiston suunnittelu ja toteutus

3.4.1 Yleistä

Suunnittelukieleksi valittiin VB:NET ja ohjelmassa käytetyt ohjelmointikirjastot ovat Visual Studion omat kirjastot ”MySQL.Data.MySqlClient” ja ”System.IO”. Ensin mainittu on kokoelma aliohjelmia, joiden avulla saadaan toteutettua MySQL-tietokantayhteydet ohjelman ja tietokannan välillä. ”System.IO”-kirjasto sisältää aliohjelmia, joiden avulla ohjelmassa on toteutettu muun muassa tulostus- ja tallennustoimintoja.

Ennen varsinaisen koodaustyön aloittamista täytyi asentaa tietokantapalvelin sekä tietokanta-ajuri, jolla muodostetaan yhteysväylä ohjelman ja tietokannan välille. Tietokantapalvelin on MySQL Server 5.7 ja tietokanta-ajuri MySQL:n Connector\NET (ADO.NET). Ajuri linkitetään (asennetaan) Visual Studioon, jonka jälkeen MySQL-tietokantayhteys on mahdollista muodostaa.

3.4.2 Kalenterinäkymän suunnittelu

Kun koodaustyö aloitettiin, ensimmäiseksi lähdettiin toteuttamaan kalenterinäkymän perusrunkoa. Kalenterinäkymä on ohjelman pääikkuna (main form), johon ohjelma käynnistyessään avautuu ja jonka sulkeminen sulkee koko ohjelman. Kalenterinäkymän kalenteriobjektina toimii ”DataGridView” -komponentti.

Kalenterialue määritettiin vakiomittaiseksi. Kalenteritaulukon koko on 46x30 (rivit x sarakkeet). Tästä alueesta sellaisia soluja, joihin huonevarauksia voidaan sijoittaa, on 43x28. Ylimmät kolme soluriviä on varattu päivämäärän, viikkonumeron ja viikonpäivän kertomiseen (**Kuvio 24.**), ja kaksi ensimmäistä saraketta merkkavat vuodepaikkaa (huone ja sänky).

	2.12.2	3.12.2	4.12.2	5.12.2	6.12.2	7.12.2	8.12.2	9.12.2	10.12	11.12	12.12	13.12	14.12	15.12	16.12	17.12	18.12	19.12	20.12	21.12	22.12	23.12	24.12	25.12	26.12	27.12	28.12	29.12	
	Wko 49	Wko 49	Wko 49	Wko 49	Wko 50	Wko 50	Wko 50	Wko 50	Wko 50	Wko 50	Wko 50	Wko 51	Wko 51	Wko 51	Wko 51	Wko 51	Wko 51	Wko 51	Wko 51	Wko 52	Wko 52	Wko 52	Wko 52	Wko 52	Wko 52	Wko 52	Wko 53	Wko 53	
Huone	S	Keskivi	Torstai	Perjantai	Lauantai	Suomi	Maana	Tistai	Keskivi	Torstai	Perjantai	Lauantai	Suomi	Maana	Tistai	Keskivi	Torstai	Perjantai	Lauantai	Suomi	Maana	Tistai	Keskivi	Torstai	Perjantai	Lauantai	Suomi	Maana	Tistai

Kuvio 24. Kalenterin kolme ensimmäistä soluriviä.

Kalenterin määrittystä varten luodaan ensin ”DataTable” -objekti, jonka avulla käsiteltävä data säilyy muistissa aina tarvittavan ajan. Kuvio 25 on sellainen osa koodia, jossa ”DataTable” määritellään. Sen jälkeen syötetään ohjelmaan vuodepaikkojen nimet sekä liitetään nämä rivit kalenteriin.

```

'Määritellään DataTable kalenteria varten.
Dim dt As New DataTable
Dim f1 = 1
Dim f2 = 2
Dim f3 = 3
dt.Columns.Add(f1)
dt.Columns.Add(f2)

'Kalenterin vasemman reunan solukkoihin syötetään tieto vuodepaikkojen nimistä (huoneet ja vuoteet).
Dim row1 As DataRow = dt.NewRow
row1.Item(f1.ToString) = ""
row1.Item(f2.ToString) = ""
Dim row2 As DataRow = dt.NewRow
row2.Item(f1.ToString) = ""
row2.Item(f2.ToString) = ""
Dim row3 As DataRow = dt.NewRow
row3.Item(f1.ToString) = "Huone"
row3.Item(f2.ToString) = "Sänky"
Dim row4 As DataRow = dt.NewRow
row4.Item(f1.ToString) = "H1"
row4.Item(f2.ToString) = "1"
Dim row5 As DataRow = dt.NewRow
row5.Item(f2.ToString) = "2"

'Liitetään yllä luodut rivit kalenteriin.
dt.Rows.Add(row1)
dt.Rows.Add(row2)
dt.Rows.Add(row3)
dt.Rows.Add(row4)

```

Kuvio 25. Kalenterinäkömän kalenterin alkumäärittämiä.

Seuraavaksi määritetään kalenterin värit siten, että kolme ylintä soluriviä värjätään tummemmalla värillä kuin muu alue, ja siitä alaspäin joka toinen rivi värjätään hieman eri värillä, jotta rivit erottuvat helpommin ja niitä on miellyttävämpi seurata/lukea.

Solujen leveydet määritetään koodissa kiinteän mittaisiksi, jotta kalenterinäkömässä näkyisi aina kerrallaan sama määrä päiviä, eli 28 (neljä viikkoa). Tämä vaikuttaa myös kalenterin latausnopeuteen merkittävästi, kun tietokoneen ei tarvitse laskea joka solun/sarakkeen leveyttä erikseen sen mukaan, mitä sen sisältö tulee olemaan. Mikäli kalenterinäkömään tulee paljon huonevarauksia, varausten piirtäminen saattaa kestää jopa useita minuutteja riippuen tietokoneen tehokkuudesta (ennen kaikkea keskusmuistin määrästä), jos solujen leveydet määritellään kukin erikseen soluun tulevan merkkimäärän mukaan. Mutta koska solujen leveydet määritellään vakiomittaisiksi, latausnopeus on täydelläkin kalenterinäkömällä luokkaa sekunti siinä tietokoneessa johon ohjelma on asennettu.

Seuraavassa koodissa (**Kuvio 26.**) määritetty ”DataTable” -objekti asetetaan tietolähteeksi kalenteriobjektille (DataGridView). Tämän lisäksi koodissa on kohta, jossa määritellään kalenterin toisen solurivin viikkonumerointi ja kolmannen solurivin viikonpäivät.

```
'Tässä luodaan datagridview (kalenteri) ja määritetään aiemmin luotu datatable sille datan lähteeksi,
'ja lisätään tarpeellinen määrä soluja sekä päivämäärät yms.
DataGridView1.DataSource = dt
Dim lisaaSolu As Integer

lisaaSolu = DataGridView1.ColumnCount

Dim vko53 = 0 'vuodenvaihteen viikkonumeroinnin korjausta varten
Do Until lisaaSolu = 30
    lisaaSolu = lisaaSolu + 1
    dt.Columns.Add(lisaaSolu)
    row1.Item(lisaaSolu.ToString) = EkaTuloP.Date.ToString("d.M.yyyy")

'Viikkonumerointi. IF-rakenteella vuodenvaihteen viikkonumeroinnit menevät oikein.
Dim viikkonro = Format(DatePart(DateInterval.WeekOfYear, EkaTuloP, FirstDayOfWeek.Monday, FirstWeekOfYear.FirstFourDays))
If Format(EkaTuloP, "dd:MM") = "28:12" And viikkonro <> 53 Then
    vko53 = 1
End If
If Format(EkaTuloP, "dd:MM") = "29:12" And vko53 = 1 And viikkonro = 53 Then
    viikkonro = 1
    row2.Item(lisaaSolu.ToString) = "vko " & viikkonro
ElseIf Format(EkaTuloP, "dd:MM") = "30:12" And vko53 = 1 And viikkonro = 53 Then
    viikkonro = 1
    row2.Item(lisaaSolu.ToString) = "vko " & viikkonro
ElseIf Format(EkaTuloP, "dd:MM") = "31:12" And vko53 = 1 And viikkonro = 53 Then
    viikkonro = 1
    row2.Item(lisaaSolu.ToString) = "vko " & viikkonro
Else
    row2.Item(lisaaSolu.ToString) = "vko " & viikkonro
End If

'Viikonpäivän asettaminen
If EkaTuloP.DayOfWeek = DayOfWeek.Monday Then
    row3.Item(lisaaSolu.ToString) = "Maanantai"
ElseIf EkaTuloP.DayOfWeek = DayOfWeek.Tuesday Then
    row3.Item(lisaaSolu.ToString) = "Tiistai"
ElseIf EkaTuloP.DayOfWeek = DayOfWeek.Wednesday Then
    row3.Item(lisaaSolu.ToString) = "Keskiviikko"
ElseIf EkaTuloP.DayOfWeek = DayOfWeek.Thursday Then
    row3.Item(lisaaSolu.ToString) = "Torstai"
ElseIf EkaTuloP.DayOfWeek = DayOfWeek.Friday Then
    row3.Item(lisaaSolu.ToString) = "Perjantai"
ElseIf EkaTuloP.DayOfWeek = DayOfWeek.Saturday Then
    row3.Item(lisaaSolu.ToString) = "Lauantai"
ElseIf EkaTuloP.DayOfWeek = DayOfWeek.Sunday Then
    row3.Item(lisaaSolu.ToString) = "Sunnuntai"
End If
EkaTuloP = EkaTuloP.AddDays(1)

Loop
```

Kuvio 26. Kalenterinäkömän kalenterin solujen alkumäärittäminen.

Sunnuntait sekä käyttäjien lisäämät pyhäpäivät näkyvät ohjelmassa punaisella värillä siten, että kalenterin kolmella ylimmällä solurivillä ko. päivät värjätään. Seuraavassa koodissa (**Kuvio 27.**) suoritetaan käyttäjien lisäämien pyhäpäivien käsittely. Ensin pyhäpäivät luetaan tekstitiedostosta listaksi, jonka jälkeen listan päivät käydään läpi siten, että niitä verrataan kalenterinäkymän päiviin ja jos löytyy täsmäävä päivä, ko. päivää merkkavat solut värjätään punaisella (IndianRed).

```
'PYHÄPÄIVÄT PUNAISELLA
Try
    ListBoxPiilo.Items.Clear()
    Dim readerPiilo As StreamReader = My.Computer.FileSystem.OpenTextFileReader("C:\Huonevaraussovellus\pyhapaiivat_huonevaraussovellus.txt")
    Dim a As String
    Dim muuttuja As String = ""
    a = readerPiilo.ReadLine
    If a <> Nothing Then
        Do
            If a.Contains("Päivä:") Then
                a = readerPiilo.ReadLine
                muuttuja = a
                a = readerPiilo.ReadLine
                a = readerPiilo.ReadLine
                ListBoxPiilo.Items.Add(muuttuja)
            End If
            a = readerPiilo.ReadLine
        Loop Until a Is Nothing
    End If
    readerPiilo.Close()
Catch ex As Exception
    MsgBox(ex.Message)
End Try

Try
    solut = 2
    Dim laskuri = ListBoxPiilo.Items.Count
    Dim laskurialkup = laskuri
    Dim laskuri2 = 0
    Dim nro = 0
    Do Until solut = 28
        ListBoxPiilo.Items(nro) = ListBoxPiilo.Items(nro).ToString.Replace(" ", "")
        If ListBoxPiilo.Items(nro).ToString = row1.Item(solut).ToString Then
            DataGridView1.Item(solut, 0).Style.BackColor = Color.IndianRed
            DataGridView1.Item(solut, 1).Style.BackColor = Color.IndianRed
            DataGridView1.Item(solut, 2).Style.BackColor = Color.IndianRed
        End If
        laskuri = laskuri - 1
        nro = nro + 1
        laskuri2 = laskuri2 + 1
        If laskuri = 0 Then
            solut = solut + 1
            laskuri = laskurialkup
            nro = 0
        End If
    Loop
Catch ex As Exception
    MsgBox("Syötä pyhäpäiviä järjestelmään (Tiedosto -> Pyhäpäivien lisäys) ja päivitä sitten kalenterinäkymä.")
End Try
```

Kuvio 27. Järjestelmään syötettyjen arkipyhien käsittely.

Kun kalenterin perusmääritykset on tehty, eli kalenterin solukko on rakennettu, aletaan kalenteriin hakea huonevarauksia. Huonevarausten haku on toteutettu siten, että aluksi haetaan tietokantakyselyllä tietokannasta sellaiset huonevaraukset, jotka osuvat kalenterinäkymän päivien kanssa yhteen, eli sellaiset varaukset, joissa yksi tai useampi varauksen päivä osuu ko. aikaväliin. Huonevaraukset käydään yksitellen (while) läpi Case-rakenteella (**Kuvio 28.**) siten, että jokaista vuodepaikkaa kohtaan on oma Case. Oikea Case löydetään siis vuodepaikan perusteella, jonka jälkeen tutkitaan mistä kohtaa solujen täyttö aloitetaan ja kuinka monta solua täytetään, ja määritetään miten solut täytetään ja värjätään.

```

'Käydään läpi Select Case -rakenteella, mihin vuodepaikkaan mikäkin huonevaraus kuuluu,
'jotta se osataan sijoittaa oikealle riville kalenterissa.
Select Case huoneID
    .....H1 Sänky 1'.....
    Case 1 And vuodenro = 1
        .....Tutkitaan tulopäivän ja lähtöpäivän erotus
        erotusTuloLahto = DateDiff(DateInterval.Day, tuloP, lahtoP)

        Dim muuttujanro As Integer = 0
        Dim pvrajalaskuri As Integer = 0

        soluID = 3

        'Verrataan huonevarauksen ensimmäistä päivää kalenterinäkömään sen hetkiseen ensimmäiseen päivään,
        'jotta osataan 'piirtää' oikean mittainen varaus.
        If tuloP < vertailupv Then
            Do Until tuloP = vertailupv
                tuloP = tuloP.AddDays(1)
                erotusTuloLahto = erotusTuloLahto - 1
            Loop
        ElseIf tuloP > vertailupv Then
            Do Until vertailupv = tuloP
                vertailupv = vertailupv.AddDays(1)
                pvrajalaskuri = pvrajalaskuri + 1
                soluID = soluID + 1
            Loop
        End If

'Ensimmäinen senhetkisessä kalenterinäkömässä näkyvä huonevarauksen solun teksti isommalla fontilla,
'jälkimmäisissä soluissa teksti näkyy vaan taustalla, eli kun solu on aktiivinen.
'Toteutuneet varaukset lihavoituna ja alleviivattuna
'Varauksen väri statuksen mukaan (asukas=keltainen...)
'Tehdään niin kauan kuin soluja 'piirrettävänä'.
Do Until erotusTuloLahto < 0 Or pvrajalaskuri >= 28
    If muuttujanro = 0 Then
        row4.Item(soluID.ToString) = AsSukunimi & " " & AsNimi & nollat & varausID
        If vahvista = True Then
            DataGridView1.Rows(3).Cells(soluID.ToString).Style.Font = New Font("Arial", fo1, FontStyle.Bold Or FontStyle.Underline)
        Else
            DataGridView1.Rows(3).Cells(soluID.ToString).Style.Font = New Font("Arial", fo1, FontStyle.Regular)
        End If
    Else
        row4.Item(soluID.ToString) = AsSukunimi & " " & AsNimi & nollat & varausID
        If vahvista = True Then
            DataGridView1.Rows(3).Cells(soluID.ToString).Style.Font = New Font("Arial", fo2, FontStyle.Bold Or FontStyle.Underline)
        Else
            DataGridView1.Rows(3).Cells(soluID.ToString).Style.Font = New Font("Arial", fo2, FontStyle.Regular)
        End If
        If AsStatus = "asukas" Then
            DataGridView1.Item(soluID.ToString, 3).Style.ForeColor = Color.Yellow

        ElseIf AsStatus = "leikkaus" Then
            DataGridView1.Item(soluID.ToString, 3).Style.ForeColor = Color.OrangeRed

        ElseIf AsStatus = "kuntoutus" Then
            DataGridView1.Item(soluID.ToString, 3).Style.ForeColor = Color.LimeGreen

        ElseIf AsStatus = "itse maksava" Then
            DataGridView1.Item(soluID.ToString, 3).Style.ForeColor = Color.CadetBlue
        End If
    End If

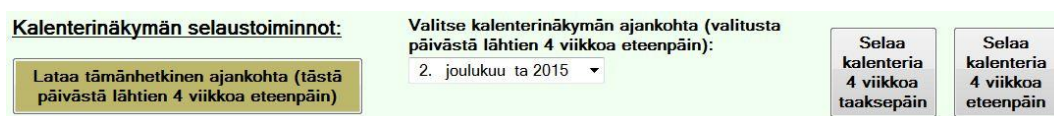
    If AsStatus = "asukas" Then
        DataGridView1.Item(soluID.ToString, 3).Style.BackColor = Color.Yellow
    ElseIf AsStatus = "leikkaus" Then
        DataGridView1.Item(soluID.ToString, 3).Style.BackColor = Color.OrangeRed
    ElseIf AsStatus = "kuntoutus" Then
        DataGridView1.Item(soluID.ToString, 3).Style.BackColor = Color.LimeGreen
    ElseIf AsStatus = "itse maksava" Then
        DataGridView1.Item(soluID.ToString, 3).Style.BackColor = Color.CadetBlue
    End If

    muuttujanro = muuttujanro + 1
    pvrajalaskuri = pvrajalaskuri + 1
    erotusTuloLahto = erotusTuloLahto - 1
    soluID = soluID + 1
Loop

```

Kuvio 28. Ote koodin Case-rakenteesta. Koodi kuuluu funktioon, joka täyttää sellaiset solut, joihin kuuluu sijoittaa tietokannasta löytyvä huonevaraus, oikeilla tiedoilla, oikeilla korostuksilla ja oikealla värillä (tietokantakyselyllä haetut varaukset ovat sellaisia, jotka osuvat sen hetkiseen kalenterinäkömään).

Kalenterinäkymää ohjataan siten, että oletuksena kalenterin ensimmäinen päivä on sen hetkinen päivä ja siitä neljä viikkoa eteenpäin, mutta lisäksi kalenteria voi selata eteen- ja taaksepäin aina neljän viikon periodeina, ja lisäksi kalenterin voi ladata alkamaan haluamastaan päivämäärästä (**Kuvio 29.**). Aina kun kalenteri ladataan (mistä tahansa päivämäärästä lähtien), aiemmin kuvatut toimet tehdään, eli kalenterinäkymän solukko valmistellaan ja käydään Case-rakenteella sen hetkiseen kalenterinäkymään osuvat huonevaraukset läpi sijoittaen ne kalenteriin aikapalkkeina.



Kuvio 29. Kalenterin selaustoiminnot sijaitsevat kalenterinäkymän alakulmassa.

Kun huonevaraukset on käsitelty ja tuotu kalenterinäkymään, muita toimintoja ”MainFormissa” on suunniteltujen huonevarausten automaattinen tarkkailu, sekä määrittely, että mitä tehdään kun kalenterinäkymän jotain (huonevarausalueen) solua klikataan. Suunniteltujen huonevarausten automaattinen tarkkailu tarkoittaa sitä, että mikäli tietokannasta löytyy huonevarauksia, jotka on merkattu suunnitelmiksi ja joiden tulopäivä on sen hetkinen päivä, käyttäjältä kysytään halutaanko tällaiset huonevaraukset muuttaa toteutuneiksi vai poistetaanko ne. Kalenterin huonevarausalueen solun klikkaus avaa joko ”Uusi varaus” -ikkunan tai ”Varauksen tiedot” -ikkunan sen mukaan, onko klikattu solu tyhjä vai sisältääkö se huonevarauksen. Mikäli solu sisältää huonevarauksen, luetaan solusta sen sisältämä ”huonevarausID”, jonka perusteella tietokannasta luetaan oikean varauksen tiedot ”Varauksen tiedot” -ikkunaan. Jos solu on tyhjä, suoritetaan Kuvio 30:n koodi. Klikatusta solusta selvitetään vuodepaikka ja päivämäärä, ja nämä tiedot viedään ”Varauksen tiedot” -lomakkeelle.

```

Dim solunsisalto = DataGridView1.Rows(e.RowIndex).Cells(e.ColumnIndex).Value

Dim rivitunniste = DataGridView1.Rows(e.RowIndex).Index
Dim solutunniste = DataGridView1.Rows(e.RowIndex).Cells(e.ColumnIndex).ColumnIndex

'Jos klikattu solu on tyhjä (ei huonevarausta), avataan formi jossa voi luoda uuden huonevarauksen ja asetetaan klikattua solua
'vastaava päivämäärä ja vuodepaikka oletukseksi huonevarauspohjaan
If solunsisalto Is DBNull.Value Then

    Dim rivinhuonetieto = DataGridView1.Rows(rivitunniste).Cells(0).Value
    Dim vuodelaskuri As Single = 1
    Dim katsoaika As Date = DataGridView1.Rows(0).Cells(e.ColumnIndex).Value
    If rivinhuonetieto Is DBNull.Value Then

        Do While rivinhuonetieto Is DBNull.Value
            rivitunniste = rivitunniste - 1
            rivinhuonetieto = DataGridView1.Rows(rivitunniste).Cells(0).Value
            vuodelaskuri = vuodelaskuri + 1
        Loop

    End If

    rivinhuonetieto_siirto = rivinhuonetieto
    vuodelaskuri_siirto = vuodelaskuri
    katsoaika_siirto = katsoaika
    FormUusiVaraus.Show()

```

Kuvio 30. Kalenterinäkömään kalenterin tyhjää solua klikataan.

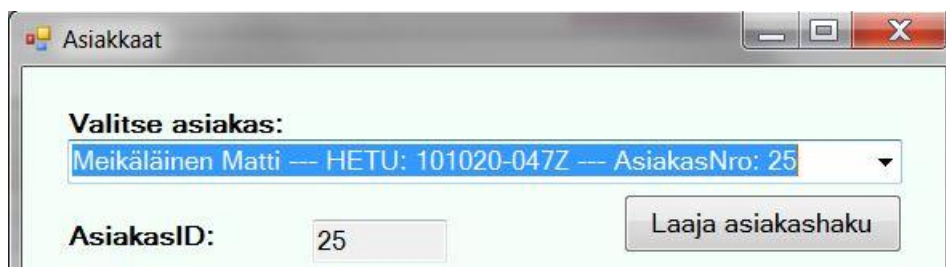
3.4.3 Muut ohjelmistosuunnittelun vaiheet

Kun ”MainForm” eli kalenterinäkömä oli toteutettu, seuraavaksi siirryttiin muiden formien toiminnallisuuksien toteuttamiseen. Näiden lomakkeiden visuaalisuus ja toiminnot on kuvattu luvussa 3.3. Tietokantaan syötettävien tietojen tarkistamisesta ja päällekkäisyyksien estämisestä kerrotaan luvussa 4.1. Tässä luvussa käydään läpi vielä muiden formien suunnittelua.

Kalenterinäkömää suunniteltaessa tietokantaan syötettiin tietoja aluksi MySQL Workbench:n kautta. Seuraava vaihe oli saada syötettyä ja luettua sekä asiakas- että huonevaraustiedot tietokantaan ohjelman kautta, eli seuraavaksi koodattaisiin toiminnot ”Asiakkaat” -formiin, ”Uusi varaus” -formiin ja ”Varauksen tiedot” -formiin.

”Asiakkaat” -lomakkeelle haetaan tiedot tietokannasta aina sen mukaan, kenen asiakkaan nimi on valittuna alasvetovalikosta (**Kuvio 31.**). Oikean asiakkaan tiedot haetaan AsiakasID:n avulla, joka luetaan piilotetusta alasvetolaatikosta, jossa on luettavissa pelkästään AsiakasID, aina näkyvää alasvetovalikkoa vastaavalla rivillä.

Toinen tapa hakea halutun asiakkaan tiedot lomakkeelle on käyttää ”Laaja asiakashaku” -toimintoa. ”Laaja asiakashaku” -painiketta painamalla aukeaa ”Asiakaslista” -lomake, josta voidaan eri ehdoilla listata asiakkaita ja tästä listasta voidaan kaksoisklikkaamalla viedä halutun asiakkaan tiedot ”Asiakkaat” -lomakkeelle. Kuvio 32 on koodi tästä kaksoisklikkaustapahtumasta. Koodissa tutkitaan, onko ”Asiakkaat” -lomake avoinna, ja mikäli on, asiakastiedot viedään ”Asiakkaat” -lomakkeelle. Jos taas ”Uusi varaus” -lomake on avoinna mutta ”Asiakkaat” -lomake ei, asiakastiedot viedään ”Uusi varaus” -lomakkeelle.



Kuvio 31. ”Asiakkaat” -lomakkeen ”Valitse asiakas” -alasvetovalikko, josta valitaan halutun asiakkaan kaikki tiedot lomakkeelle tuotaviksi.

```
Private Sub ListViewAsiakkaat_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs) Handles ListViewAsiakkaat.DoubleClick
    Try
        Dim asid = ListViewAsiakkaat.SelectedItems(0).SubItems(0).Text
        If FormAsiakkaat.IsHandleCreated Then
            Dim s = 0
            Dim l = 0
            Do Until l = 1
                If FormAsiakkaat.ComboBoxValitseAsiakas.Items(s).ToString.EndsWith(asid) Then
                    l = 1
                    FormAsiakkaat.ComboBoxValitseAsiakas.SelectedIndex = s
                End If
                s = s + 1
            Loop
        ElseIf FormUusiVaraus.IsHandleCreated Then
            Dim s = 0
            Dim l = 0
            Do Until l = 1
                If FormUusiVaraus.ComboBoxValitseAsiakas.Items(s).ToString.EndsWith(asid) Then
                    l = 1
                    FormUusiVaraus.ComboBoxValitseAsiakas.SelectedIndex = s
                End If
                s = s + 1
            Loop
        End If
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
```

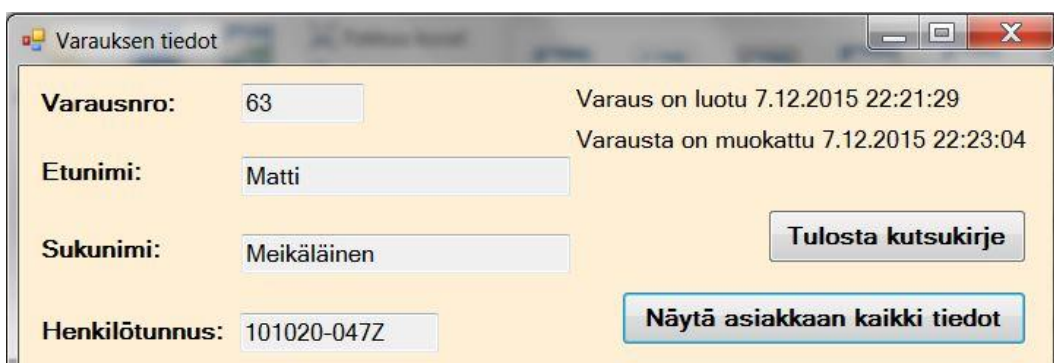
Kuvio 32. Koodi asiakaslistan kaksoisklikkaustapahtumasta.

Kun ”Asiakkaat” -lomake oli toteutettu, voitiin seuraavaksi toteuttaa ”Uusi Varaus” -lomake. ”Uusi varaus” -lomakkeelle haetaan halutun asiakkaan tiedot samalla periaatteella kuin ”Asiakkaat” -lomakkeelle, eli asiakas valitaan joko alavetovalikosta tai laajaa asiakashakua käyttäen (**Kuvio 33.**).



Kuvio 33. ”Uusi varaus” -lomakkeen asiakkaan valinta.

Sen jälkeen, kun ohjelmassa on voitu luoda sekä asiakkaita että huonevarauksia, voitiin toteuttaa ”Varauksen tiedot” -lomake, jolle oikean asiakkaan tiedot haetaan sen solun mukaan, mitä on klikattu kalenterinäkylässä. Tällä lomakkeella nähdään kaikki huonevarauksen tiedot ja niitä voidaan paitsi muokata ja varaus voidaan poistaa, mutta lisäksi ko. huonevarauksen asiakkaan kaikki tiedot päästään näkemään ja tulostamaan kutsukirje (**Kuvio 34.**). Kutsukirjeen tulostuksessa käytetään ”PrintDialog”, ”PrintDocument” ja ”Graphics.DrawString” -metodeja.



Kuvio 34. ”Varauksen tiedot” -lomakkeelta käsin päästään tarkistamaan ko. varauksen asiakkaan kaikki tiedot sekä tulostamaan kutsukirje.

Kuvio 35 on koodi, jolla noudetaan asiakkaan huonevaraustietoja asiakastietolomakkeilla. Koodissa selvitetään, onko asiakkaalla huonevarauksia, ja jos niitä on, listataan erikseen menneet varaukset, tulevat varaukset ja mahdollisesti tämänhetkinen varaus.

```
'Tietokantayhteys
Dim conn As MySqlConnection
conn = New MySqlConnection("server=localhost;user id=xxxxxxx;password=xxxxxxx;database=xxxxxxx;")
conn.Open()

'Haetaan tietokannasta asiakkaan huonevaraukset, ja kerrotaan niistä RichTextBoxVaraukset-kentässä mikäli niitä on
Dim cmd1 As MySqlCommand = New MySqlCommand("select tulopvm, lahtopvm, huonetunnus, vuodenumero, status from varaukset" +
"LEFT JOIN huone ON varaukset.huone_idhuone=huone.idhuone" +
"where asiakas_idasiakas = '" & TextBoxAsID.Text & "' order by lahtopvm", conn)

Dim reader As MySqlDataReader = cmd1.ExecuteReader()
Dim laskuri As Single = 0
Dim laskuri2 As Single = 0
If reader.Read = False Then
    RichTextBoxVaraukset.Text = "Asiakkaalla ei ole huonevarauksia."
Else
    reader.Close()
    reader = cmd1.ExecuteReader()
    While reader.Read()
        Dim tulo = reader.GetDateTime(0)
        Dim lahto = reader.GetDateTime(1)
        Dim huone = reader.GetString(2)
        Dim vuode = reader.GetInt16(3)
        Dim status = reader.GetString(4)

        If lahto < Now.Date Then
            If laskuri = 0 Then
                RichTextBoxVaraukset.AppendText("MENNEET HUONEVARAUKSET:" & vbCrLf)
                laskuri = 1
            End If
            RichTextBoxVaraukset.AppendText("Aika: " & tulo & " - " & lahto & " | Huone: " & huone & _
                " | Sänky: " & vuode & " | Status: " & status & vbCrLf)
        End If
        If tulo <= Now.Date And lahto >= Now.Date Then
            RichTextBoxVaraukset.AppendText("ASIAKAS ON TÄLLÄ HETKELLÄ TALOSSA:" & vbCrLf & "Aika: " & _
                & tulo & " - " & lahto & " | Huone: " & huone & " | Sänky: " & _
                & vuode & " | Status: " & status & vbCrLf)
        End If
        If tulo > Now.Date Then
            If laskuri2 = 0 Then
                RichTextBoxVaraukset.AppendText("TULEVAT HUONEVARAUKSET:" & vbCrLf)
                laskuri2 = 1
            End If
            RichTextBoxVaraukset.AppendText("Aika: " & tulo & " - " & lahto & " | Huone: " & huone & _
                " | Sänky: " & vuode & " | Status: " & status & vbCrLf)
        End If
    End While
End If
reader.Close()
```

Kuvio 35. Asiakkaan huonevaraustietojen haku.

”Varauslista” -lomake toimii samalla periaatteella kuin ”Asiakaslista” -lomake. Huonevarauksia voi listata eri ehdoilla ListView-elementtiin. Tässä listassa näytetään varauksen kaikki tiedot, ja varauksen voi siirtää lähempään tarkasteluun ”Varauksen tiedot” -lomakkeelle kaksoisklikkaamalla haluttua varausta listassa.

Muita ohjelman lomakkeita (form) ovat muun muassa lomake, jolle noudetaan poistettavia huonevarauksia, loki, jossa voidaan listata joko kaikki tai tietyltä ajalta lokimerkinnät ja johon tiedot on kerätty siten, että aina kun jotain tietoa on tallennettu tai poistettu, siitä tallennetaan tieto tietokannan lokitauluun, sekä laskut-lomake, jolle haetaan erilaisia johdon laskelmia, kuten huonevarausten kuormitusaste (**Kuvio 36.**).

```

Dim eka = DateTimePickerEka.Value
Dim vika = DateTimePickerVika.Value

Dim laskuri2 = 0
Dim varauksia = 0
Dim summa As Double = 0
Dim hinta As Double = 0

Dim cmd1 As MySqlCommand = New MySqlCommand("select count(idvaraukset) from varaukset where " +
    "" & eka.ToString("yyyy-MM-dd") & " between tulopvm and lahtopvm and toteutunut = 1", conn)
Dim cmd2 As MySqlCommand = New MySqlCommand("select sum(hinta) from varaukset where " +
    "" & eka.ToString("yyyy-MM-dd") & " between tulopvm and lahtopvm and toteutunut = 1", conn)

Dim reader As MySqlDataReader = cmd1.ExecuteReader()
reader.Read()
Dim laskuri = reader.GetInt16(0)
reader.Close()

If laskuri > 0 Then
    reader = cmd2.ExecuteReader
    reader.Read()
    hinta = reader.GetDouble(0)
    reader.Close()
End If

varauksia = varauksia + laskuri
summa = summa + hinta
hinta = 0
laskuri2 = laskuri + 1
Do Until eka >= vika
    eka = eka.AddDays(1)
    cmd1 = New MySqlCommand("select count(idvaraukset) from varaukset where "" & eka.ToString("yyyy-MM-dd") & "" +
        "between tulopvm and lahtopvm and toteutunut = 1", conn)
    cmd2 = New MySqlCommand("select sum(hinta) from varaukset where "" & eka.ToString("yyyy-MM-dd") & "" +
        "between tulopvm and lahtopvm and toteutunut = 1", conn)

    reader = cmd1.ExecuteReader()
    reader.Read()
    laskuri = reader.GetInt16(0)
    reader.Close()
    If laskuri > 0 Then
        reader = cmd2.ExecuteReader
        reader.Read()
        hinta = reader.GetDouble(0)
        reader.Close()
    End If
    varauksia = varauksia + laskuri
    summa = summa + hinta
    hinta = 0
    laskuri2 = laskuri2 + 1
Loop

Dim kuormitus1 = (varauksia / (22 * (laskuri2))) * 100
LabelTulokset.Text = "Tulokset:"
LabelKuormitus1.Text = "Toteutuneiden huonevarausten kuormitusaste: " & Math.Round(kuormitus1, 3) & " %"

```

Kuvio 36. Huonevarausten kuormitusasteen laskeminen.

4 TESTAUS JA ASENNUS

4.1 Järjestelmän testaus

Ohjelmaa suunniteltiin siten, että sitä testattiin jatkuvasti. Aina kun tehtiin jokin uusi toiminto, testattiin ohjelman toimintaa mahdollisimman perusteellisesti. Asiakkaan kanssa pidimme säännöllisiä videoneuvotteluita, joissa asiakas sai nähdä, missä vaiheessa ohjelma milloinkin on, mitä toimintoja siinä on ja miltä se näyttää, ja joissa sovittiin mitä tehdään seuraavaksi. Tämä käytäntö edesauttoi jatkuvaa testausta.

Asiakkaan kanssa sovimme, että asiakasyrityksessä suoritetaan järjestelmän lopullinen testaus siten, että he käyttävät ohjelman testausvaiheessa vielä rinnalla vanhaa huonevarausjärjestelmäänsä. Asiakas pystyy parhaiten itse suorittamaan lopullisen testauksen, sillä heillä järjestelmä asetetaan heti jokapäiväiseen käyttöön ja tietokantaan tullaan syöttämään runsaasti asiakas- ja huonevaraustietoja.

Ohjelmaa suunniteltaessa oli selvää, että ohjelman tulisi estää virheellisten ja päällekkäisten tietojen syöttäminen tietokantaan. Jokaisessa ohjelman formissa on erilaisia tarkistuksia, joilla varmistetaan oikeanlaisten tietojen tallennus. Esimerkiksi huonevarauksia tehdessä tarkistetaan, onko ko. asiakkaalla jo olemassa oleva huonevaraus, joka osuu samaan ajanjaksoon yhdeltä tai useammalta päivältä. Mikäli tällainen päällekkäisyys löytyy, ei huonevarauksen tallennus onnistu ja siitä informoidaan käyttäjää.

Ohjelmakoodissa on useita Try...Catch -virheenkäsittelymetodeja. Kaikki toiminnot on ympäröity näillä metodeilla siten, että jotkut toiminnot on ympäröity omalla virheenkäsittelymetodillaan ja joillakin formeilla saattaa olla vain yksi Try...Catch, jonka sisällä kaikki koodi sijaitsee. Tällä virheenkäsittelymetodilla saadaan napattua poikkeuksen aiheuttama virhe ja informoitua siitä ohjelman käyttäjää ilman, että ohjelma kaatuisi siihen virheeseen.

4.2 Järjestelmän asennus ja varmuuskopioinnin järjestäminen

Järjestelmä asennettiin asiakkaan tietokoneelle huhtikuussa 2015. Tällöin asennettiin tietokanta sekä ensimmäinen versio sovelluksesta, joka oli vielä asiakkaan testausta vaativa versio. Myöhemmin sovellukseen tehtiin vielä joitain muutoksia sekä lisäominaisuuksia. Samalla, kun tietokanta ja ensimmäinen sovellusversio asennettiin, opastettiin lisäksi henkilökuntaa järjestelmän käytössä. Myöhemmät versiot ohjelmasta on toimitettu asiakkaalle sähköpostitse asennusohjeiden kera. Versiopäivitykset on helppo toteuttaa asentamalla uudet ohjelmatiedostot vanhojen vastaavien ohjelmatiedostojen päälle. Tietokantaan ei ole tarvinnut tehdä muutoksia jälkeenkään.

Tietokantaa varten tietokoneeseen asennettiin MySQL Server 5.7 -tietokantapalvelin ja se asetettiin käynnistymään automaattisesti Windowsin käynnistymisen yhteydessä (Windowsin palveluna). Lisäksi asennettiin Connector/NET -ajuri sovelluksen ja tietokannan kommunikointia varten. Itse sovellus asennettiin tietokoneeseen tavanomaisen työpöytäsovelluksen tavoin, jonka jälkeen järjestelmä oli käyttövalmis.

Tietokannan varmuuskopiointi järjestettiin siten, että asiakkaalle laadittiin ohjeet varmuuskopiointien laadintaan ja kerrottiin myös saatavilla olevista sovelluksista, joilla varmuuskopioinnin voi hoitaa ajastetusti. Asiakkaalla on käytössään ulkoistettu järjestelmä- ja mikrotukipalvelu ja tarkoitus oli, että asiakas hoitaa varmuuskopioinnin sitä kautta. Asiakasta opastettiin hoitamaan varmuuskopiointi siten, että varmuuskopiot otettaisiin säännöllisesti ja ne talletettaisiin fyysisesti eri sijaintiin, esimerkiksi johonkin pilvipalvelu-tallennustilaan. Sovimme kuitenkin, että ohjelman kehittäjäkin on käytettävissä myös varmuuskopioinnin järjestämisessä ja muissa siihen liittyvissä asioissa. Varmuuskopio, eli ”dump file” MySQL -tietokannasta, voidaan luoda Windowsin komentokehoteella seuraavanlaisen esimerkkilauseen mukaisesti: ”C:\Program Files\MySQL\MySQL Server 5.7\bin>mysqldump -u root pSalasana tietokanta > c:\varmuuskopio.sql”.

Mikäli tietokone, johon järjestelmä on asennettu, rikkoutuu, järjestelmän saa otettua nopeasti käyttöön uudessa tietokoneessa. Uuden järjestelmän saa käyttövalmiiksi

noin tunnissa sisältäen tietokantapalvelimen käyttöönoton, tietokannan asentamisen varmuuskopiosta ja itse sovelluksen asentamisen.

5 YHTEENVETO

Projekti onnistui hyvin ja lopputulos oli odotusten mukainen. Järjestelmän käyttöönotto on sujunut hyvin ja järjestelmä on ollut tämän raportin kirjoitushetkellä asiakasyrityksen jokapäiväisessä käytössä noin puoli vuotta. Asiakasyritys sai käyttöönsä heidän tarpeitaan vastaavan järjestelmän, joka on toiminut moitteettomasti ja joka on ollut selvä parannus entiseen järjestelmään.

Jatkossa voisi olla mahdollista, mikäli tarvetta ilmaantuu, että järjestelmää kehitetään esimerkiksi siten, että tietokanta asennetaan erilliselle palvelimelle ja ohjelmistoa aletaan käyttää useammasta tietokonepäätteestä käsin. Myös erilaisten lisäominaisuuksien suunnittelu järjestelmään on mahdollista, eikä poissuljettua ole sekään, että järjestelmää voisi yhteensovittaa keskustelemaan jonkin muun sovelluksen/järjestelmän kanssa.

Järjestelmä toimii omassa ympäristössään hyvin, mutta hyvä jatkokehitysidea voisi olla myös se, että ohjelmiston ja tietokannan välille suunniteltaisiin sellainen yhteysväylä, jonka avulla ohjelmistoa olisi helpompi yhteensovittaa, jos esimerkiksi tietokantana toimisi jokin muu kuin MySQL.

Tämän opinnäytetyön tekeminen on ollut erittäin hyödyllistä ja opettavaista. Työ on opettanut paljon niin ohjelmistotuotannosta kuin asiakasprojektin läpiviemisestäkin. Projektin työmäärä osoittautui arvioitua suuremmaksi, joten ensimmäiset arviot projektin aikataulusta muuttuivat myöhemmin. Kaiken kaikkiaan järjestelmän valmiiksi saattaminen kesti noin vuoden (9/2014 – 8/2015), kun ensimmäiset varovaiset arviot projektin kestosta olivat noin puoli vuotta. Toki ensimmäinen toimiva versio otettiin asiakasyrityksessä käyttöön 5/2015, myöhemmin toteutettiin järjestelmään lisäominaisuuksia.

LÄHTEET

- /1/ dev.mysql.com, Data Types. Viitattu 28.11.2015. <http://dev.mysql.com/doc/refman/5.7/en/data-types.html>
- /2/ dev.mysql.com, Move from Microsoft SQL Server. Viitattu 28.11.2015. https://dev.mysql.com/tech-resources/articles/move_from_microsoft_SQL_Server.html
- /3/ dev.mysql.com, MySQL dump. Viitattu 7.12.2015. <http://dev.mysql.com/doc/refman/5.7/en/mysqldump.html>
- /4/ dotnetperls.com, DataTable. Viitattu 7.12.2015. <http://www.dotnetperls.com/datatable-vbnet>
- /5/ edu.phkk.fi, Käyttötapauskaavio. Viitattu 7.12.2015. <http://edu.phkk.fi/opis-kehu/oliomallinnus/Kayttotapaus.htm>
- /6/ Immonen J, Ohjelmistotuotannon luentomoniste. Viitattu 20.11.2015. http://cs.joensuu.fi/~jimmonen/jot_moniste/jot_moniste_121.html
- /7/ msdn.microsoft.com, DataGridView. Viitattu 7.11.2015. <https://msdn.microsoft.com/en-us/library/ms171628%28v=vs.110%29.aspx>
- /8/ msdn.microsoft.com, Graphics.DrawString. Viitattu 7.12.2015. <https://msdn.microsoft.com/en-us/library/21kdfbzs%28v=vs.110%29.aspx>
- /9/ msdn.microsoft.com, Library. Viitattu 10.12.2015. <https://msdn.microsoft.com/en-us/library/2x7h1hfk%28v=vs.100%29.aspx>
- /10/ msdn.microsoft.com, PrintDocument. Viitattu 7.12.2015. <https://msdn.microsoft.com/en-us/library/system.drawing.printing.printdocument.print%28v=vs.110%29.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2>
- /11/ tutorialspoint.com, ER diagram. Viitattu 7.12.2015. http://www.tutorialspoint.com/dbms/er_diagram_representation.htm
- /12/ tutorialspoint.com, ListView. Viitattu 7.12.2015. http://www.tutorialspoint.com/vb.net/vb.net_listview.htm

Geritrim Huonevaraussovelluksen käyttöohjeet 30.4.2015 -

SISÄLLYSLUETTELO

1 Päänäkymä -----	2
2 Varaukset -----	3
2.1 Varauksen luonti -----	3
2.2 Varauksen tarkistelu ja muokkaus -----	4
2.3 Varauksen poistaminen -----	5
3 Asiakkaat -----	5
3.1 Asiakkaan luonti -----	6
3.2 Asiakastiedon muokkaus -----	7
3.3 Asiakkaan poistaminen -----	7
4 Tietojen haku -----	7
4.1 Asiakastiedot -----	7
4.2 Varaustiedot -----	8
4.3 Lokitiedot -----	8
4.4 Laskelmat -----	9
5 Ohjelman uudet ominaisuudet (2.8.2015) -----	10
5.1 Pyhäpäivät -----	10
5.2 Asiakastyypit ja uudet hakutoiminnot -----	11
5.3 Kutsukirje -----	15
5.4 Asukas-statuksella käytetyt päivät -----	15
5.5 Varaus toteutuneeksi (autom. kysymys) ---	15
6 Ohjelman uudet ominaisuudet (9.8.2015) -----	16

1 Päänäkymä

Tiedosto	Ohje	30.4.2015	1.5.2015	2.5.2015	3.5.2015	4.5.2015	5.5.2015	6.5.2015	7.5.2015	8.5.2015	9.5.2015	10.5.2015	11.5.2015	12.5.2015	13.5.2015	14.5.2015	15.5.2015	16.5.2015	17.5.2015
Huone	Sänky	Torstai	Perjantai	Lauantai	Sunnuntai	Maanantai	Tiistai	Keskiviikko	Torstai	Perjantai	Lauantai	Sunnuntai	Maanantai	Tiistai	Keskiviikko	Torstai	Perjantai	Lauantai	Sunnuntai
H1	1																		
H2	1																		
H3	1																		
H4	1																		
H5	1																		
H6	1																		
H7	1																		
H8	1																		
H9	1																		
H10	1																		
H11	1																		
H12	1																		
H14	1																		
H15	1																		
H16	1																		
H17	1																		
H18	1																		
H19	1																		

Kuva 1.

Päänäkymä johon ohjelma avautuu kun se käynnistetään, käsittää kuvan 1 mukaisesti solurakenteen, jonka vasemmassa laidassa huonepaikat ja ylhäällä päivämäärät. Oletusnäkyvän ensimmäinen päivä on sen hetkinen päivä ja varaukset näkyvät siitä 4 viikkoa eteenpäin (kerrallaan näkyy 18 päivää, alhaalla olevaa vierityspalkkia viemällä oikealle, nähdään loput päivät).

Varaus luetaan siten, että katsotaan mikä solu(t) vastaa minkäkin vuodepaikan ja päivämäärän yhtymäkohtaa.

Alareunassa olevilla painikkeilla voidaan ladata haluttu näkymä:

- Tämänhetkinen näkymä (sen hetkisestä päivästä 4 viikkoa eteenpäin, joka on myös oletusnäkymä)
- kalenterista valittu päivä ja siitä 4 viikkoa eteenpäin
- 4 viikkoa taaksepäin
- 4 viikkoa eteenpäin.

Väreillä (keltainen, sininen, vihreä ja punainen) ilmaistaan asiakkaan ”maksustatus”. Nämä selitteet ovat päänäkymän vasemmassa alareunassa.

Toteutuneen varauksen tunnistaa lihavoidusta ja alleviivatusta tekstistä.

Ohjelma sammutetaan joko punaisesta ruksista tai ”Tiedosto -> Sulje”.

Nämä ohjeet sekä tietoa ohjelmasta löytyy yläreunan menu –valikon ”Ohje” kautta. Tämä huonevaraussovellusohjeet.pdf tulee olla tallennettuna c-aseman juureen (C:\).

2 Varaukset

2.1 Varauksen luonti

The screenshot shows the Geritrim Huonevaraus application interface. On the left, there is a calendar grid with columns for weeks (vko 18, vko 19) and days (Torstai, Perjantai). The grid contains room numbers (H1-H18) and bed counts (1, 2, 3). A dialog box titled 'Uusi varaus' is open in the foreground, containing the following fields:

- Valitse asiakas:** A dropdown menu.
- Tulopäivä:** 30. huhtikuu ta 2015
- Lähtöpäivä:** 30. huhtikuu ta 2015
- Status:** A dropdown menu.
- Huone:** H0
- Sänky:** 1
- Erityistoiveet:** A text input field.
- Hinta €/vrk:** A text input field.
- Varaus on toteutunut (toteutunut = kyllä, suunnitelma = ei):** A dropdown menu.
- Tallenna varaus** button.

Kuva 2.

Huonevaraus voidaan luoda joko siten, että klikataan haluttua tyhjää solua, jolloin oletuksena varausikkunassa (kuva 2) ko. solua vastaava vuodepaikka ja päivä, tai siten että valitaan "Tiedosto -> Uusi tietue / Tiedon muokkaus -> Uusi Varaus".

Avautuvaan "Uusi varaus" -ikkunaan täytetään varauksen tiedot ja painetaan "Tallenna varaus". Jos varaus on hyväksytysti täytetty, sinun tulee vielä vahvistaa tallennus seuraavaksi avautuvassa ikkunassa, muussa tapauksessa ohjelma ilmoittaa, mikäli täytettävissä tiedoissa on ongelmia.

2.2 Varauksen tarkistelu ja muokkaus

The screenshot shows the 'Geritrim Huonevaraus' application interface. A modal window titled 'Varauksen tiedot' is open, displaying the following information:

- Varausno:** 21
- Elunimi:** Antti
- Sukunimi:** Ahma
- Henkilötunnus:** 1234
- Tulopäivä:** 6. huhtikuu ta 2015
- Lähtöpäivä:** 11. heinäkuu ta 2015
- Status:** kuntoutus
- Huone:** 114
- Sänky:** 1
- Erityistoiveet:** -
- Hinta €/vrk:** 325
- Varaus on Suunnitelma / Toteutunut:** Suunnitelma

Additional details in the window include: 'Varaus on luotu 30.4.2015 9:33:11' and 'Varausta on muckattu 30.4.2015 9:35:53'. Buttons for 'Näytä asiakkaan kaikki tiedot', 'Tallenna muutokset', and 'Poista' are present at the bottom of the window.

Kuva 3.

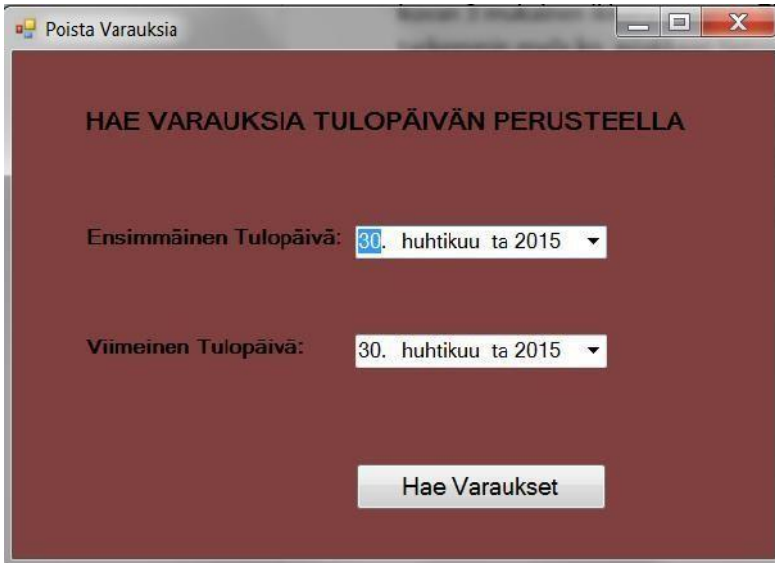
Varaus voidaan avata tarkisteltavaksi ja/tai muokattavaksi klikkaamalla haluttua varausta, jolloin kuvan 3 mukainen ikkuna avautuu. Tässä ikkunassa nähdään varauksen tiedot ja voidaan katsoa tarkemmin myös ko. asiakkaan tietoja klikkaamalla "Näytä asiakkaan kaikki tiedot".

Varauksen tietoja voidaan muokata, muuttamalla haluttuja tietoja ja sitten klikkaamalla "Tallenna muutokset". Mikäli muutokset ovat ok, seuraavaksi muutosten tallennus vielä varmennetaan, muussa tapauksessa ohjelma ilmoittaa jos muutoksissa on ongelmia.

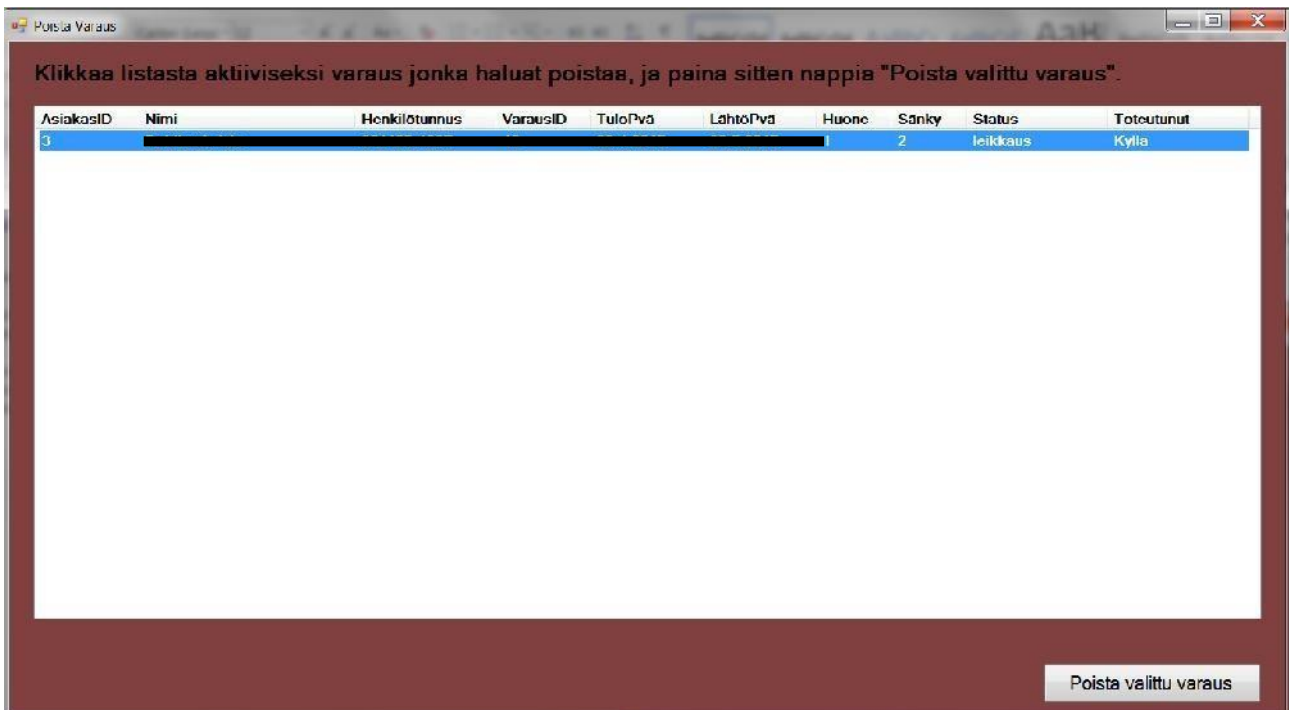
Asiakkaan tietoja ei tässä voi muuttaa, vaan niitä muutetaan valitsemalla "Tiedosto -> Uusi tietue / Tiedon muokkaus -> Asiakkaat".

2.3 Varauksen poistaminen

Varaus voidaan poistaa joko samassa ikkunassa, jossa muokkauksetkin tehdään (Kuva 3), siten että klikataan "Poista" ja sen jälkeen varmennetaan vielä varauksen poistaminen, tai siten että valitaan "Tiedosto -> Uusi tietue / Tiedon muokkaus -> Poista Varauksia". Seuraavaksi avautuvassa ikkunassa (kuva 4) valitaan tulopäivän perusteella varausten aikaväli ja klikataan "Hae Varaukset". Tämän jälkeen klikataan siniseksi listasta (kuva 5) poistettava varaus (voit myös valita useamman varauksen kerralla) ja klikataan "Poista valittu varaus". Tämän jälkeen vielä varmennukset.



Kuva 4.



Kuva 5.

3 Asiakkaat

3.1 Asiakkaan luonti

Valitaan ”Tiedosto -> Uusi tietue / Tiedon muokkaus -> Asiakkaat”. Avautuvassa ikkunassa (kuva 6) täytetään uuden asiakkaan tiedot (kentät Etunimi...Muita tietoja) ja klikataan ”Tallenna asiakastiedot” ja lopuksi varmennetaan.



The screenshot shows a window titled "Asiakkaat" with a light green background. At the top left, there is a label "Valitse asiakas:" followed by a dropdown menu. Below this are four input fields: "AsiakasID:" (a small grey box), "Etunimi:" (a white box), "Sukunimi:" (a white box), and "Henkilötunnus:" (a white box). Further down are three larger text areas: "Yhteystiedot:", "Lähiomaisen tiedot:", and "Muita tietoja:". At the bottom of the form is a large grey area labeled "Asiakkaan huonevaraukset:". At the very bottom, there are three buttons: "Tyhjennä kentät" (grey), "Tallenna asiakastiedot" (green), and "Poista" (red).

Kuva 6.

3.2 Asiakastiedon muokkaus

Samassa kuvan 6 mukaisessa ikkunassa voidaan myös muokata asiakastietoja valitsemalla ikkunan yläreunassa olevasta ”Valitse asiakas:” –listasta muokattava (tai tarkisteltava) asiakas, jonka jälkeen muutetaan halutessa tietokenttiä ja klikataan sitten klikataan ”Tallenna asiakastiedot” ja lopuksi varmennetaan.

3.3 Asiakkaan poistaminen

Edelleen samassa ikkunassa (kuva 6) tehdään myös asiakkaan tietojen poistaminen valitsemalla listasta (kuten edellä) haluttu asiakas ja klikkaamalla ”Poista”, jonka jälkeen vielä varmennetaan poisto. Asiakkaan tietoja ei voi poistaa, mikäli asiakkaalla on huonevarauksia (joko menneitä, senhetkisiä tai tulevia).

4 Tietojen haku

4.1 Asiakastiedot

Asiakastiedot on nähtävissä edellä mainitulla tavalla (”Tiedosto -> Uusi tietue / Tiedon muokkaus -> Asiakkaat”), varauksen tiedoista kuten luvussa 2.2 (”Näytä asiakkaan kaikki tiedot”, avautuvasta ikkunasta käsin asiakkaan tiedot voidaan myös tallentaa tiedostoon ”Tallenna tiedostoon” – painiketta painamalla...), sekä valitsemalla ”Tiedosto -> Hae tietoja -> Asiakastiedot”.

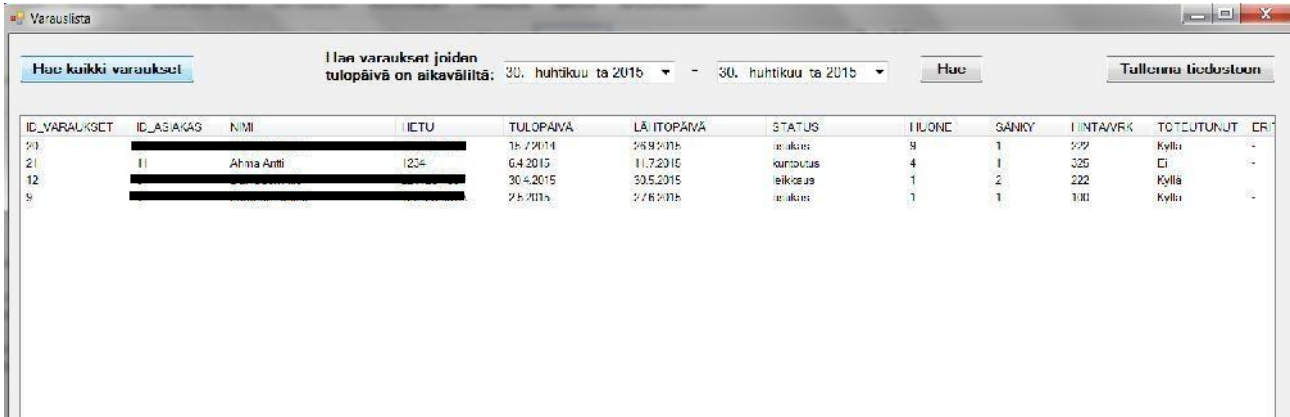
Viimeksi mainitulla tavalla avautuu kuvan 7 mukainen ikkuna, johon listautuu kaikki asiakkaat ja tiedot voi tallentaa tiedostoon klikkaamalla ”Tallenna tiedostoon”...

ASIAKAS_ID	NIMI	HILU	YHTEYSTIEDOT	VARMANEN TIEDOT
5	a e	1	1	
11	Aina Aini	1251	11	
6	b b	2	2	
7	c c	3	3	
8	d d	4	4	
3	[redacted]	[redacted]	[redacted]	
9	e e	5	5	
10	H	6	6	
1	[redacted]	[redacted]	[redacted]	
2	[redacted]	[redacted]	[redacted]	
4	[redacted]	[redacted]	[redacted]	

Kuva 7.

4.2 Varaustiedot

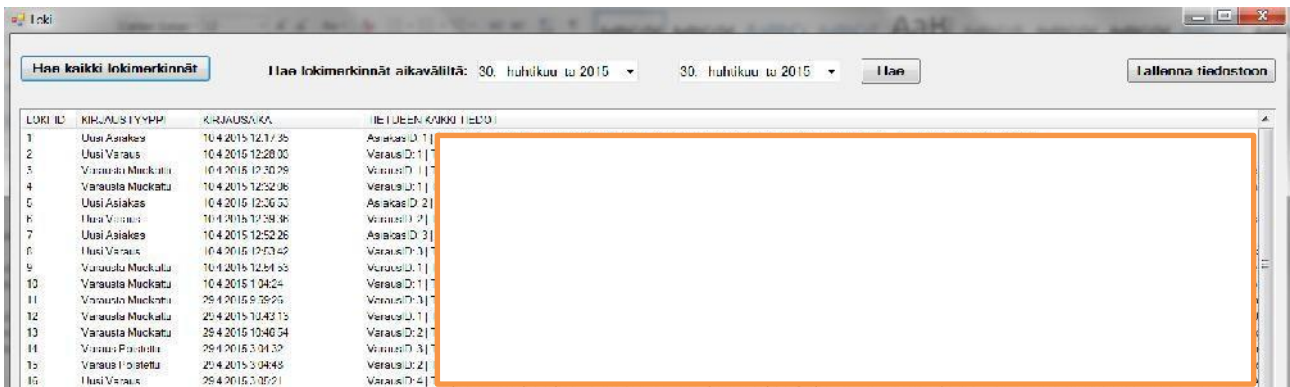
Valitsemalla ”Tiedosto -> Hae tietoja -> Varaustiedot”, avautuu ikkuna (kuva 8), josta käsin voidaan listata joko kaikki varaukset (”Hae kaikki varaukset”) tai hakea varaukset tulopäivän mukaan tietyltä aikaväliltä (valitse päivät ja ”Hae”) ja tiedot voidaan sitten tallentaa tiedostoon klikkaamalla ”Tallenna tiedostoon”...



Kuva 8.

4.3 Lokitiedot

Valitsemalla ”Tiedosto -> Hae tietoja -> Lokitiedot”, avautuu ikkuna (kuva 9), josta käsin voidaan listata joko kaikki lokimerkinnät (”Hae kaikki lokimerkinnät”) tai hakea lokimerkinnät tietyltä aikaväliltä (valitse päivät ja ”Hae”) ja tiedot voidaan sitten tallentaa tiedostoon klikkaamalla ”Tallenna tiedostoon”...



Kuva 9.

4.4 Laskelmat

Laskelmatietoja pääsee tarkastelemaan valitsemalla ”Tiedosto -> Hae tietoja -> Laskelmat”, avautuvassa ”Laskut”-ikkunassa (kuva 10) valitaan ensin aikaväli, jolta laskut halutaan suorittaa ja klikataan ”Laske”, jonka jälkeen tulokset tulevat näkyviin. Tulokset voidaan tulostaa (”Tulosta”) esim. suoraan tulostimeen tai tulostaa vaikka pdf-tiedostoksi pdf-tulostimella.



The screenshot shows a window titled "Laskut" with a light blue background. At the top, it says "Valitse aikaväli jolta haluat laskelmat". Below this, there are two date selection fields: "Ensimmäinen päivä:" with a dropdown menu showing "1. huhtikuu ta 2015" and "Viimeinen päivä:" with a dropdown menu showing "30. huhtikuu ta 2015". To the right of the second dropdown is a button labeled "Laske". Below the date fields, there is a section titled "Tulokset:" in red text. Underneath, the following information is displayed: "Toteutuneiden huonevarauksien kuormitusaste: 4,839 %", "Toteutuneiden huonevarauksien keskimääräinen vuorokausihinta: 222 €", "Suunniteltujen huonevarauksien keskimääräinen vuorokausihinta: 325 €", and "Huonevarauspäiviä yhteensä 59, joista toteutuneita 56 %". At the bottom right of the window is a button labeled "Tulosta".

Kuva 10.

5 Ohjelman uudet ominaisuudet (2.8.2015)

Uusia ominaisuuksia ja päivityksiä Huonevaraussovellukseen ovat:

- Sunnuntait ja järjestelmään syötettävät muut pyhäpäivät punaisella värillä kalenterissa
- Asiakkaille voi syöttää ”koodin” sen mukaan onko hän esim. veteraani
- Paremmat asiakastietojen hakutoiminnot (mm. edellä mainitun ”koodin” mukaan)
- Asiakkaan huonevarausta vastaavan kutsukirjeen voi tulostaa sekä suomenkielisenä että ruotsinkielisenä
- Asiakkaan tiedoista näkee asukas-statusella käytettyjen päivien määrän, sekä myös tulossa olevat asukas-statusella merkityt päivät kuluvana vuonna
- Sovellus kysyy automaattisesti ohjelman käynnistyessä sekä kalenteria selattaessa/ladattaessa, että muutetaanko sellaiset huonevaraukset, jotka ovat merkitty suunnitelmiksi ja joiden tulisi nyt alkaa, toteutuneiksi vai poistetaanko ne
- Laskelmien tulostuksen ulkoasua on parannettu mm. logoilla
- 4 viikon tilanne kaikkine vuodepaikkoineen näkyy kalenterinäkyvässä kokonaisuudessaan sivua vierittämättä ja varausnumero näkyy varatuissa soluissa viimeisenä, asiakkaan nimen jälkeen
- Varatut solut ovat kaikki samankokoisia, jotta 4 viikkoa näkyy kerrallaan vierittämättä sivua. Tämän vuoksi solun sisällöstä näkyy vain osa, mutta viemällä hiiren cursorin solun päälle, näkyy solun koko sisältö
- Kalenterin latausnopeus parempi
- Logot lisätty kalenterinäkyvän vasempaan alareunaan sekä sovelluksen ”kuvakkeeseen”.

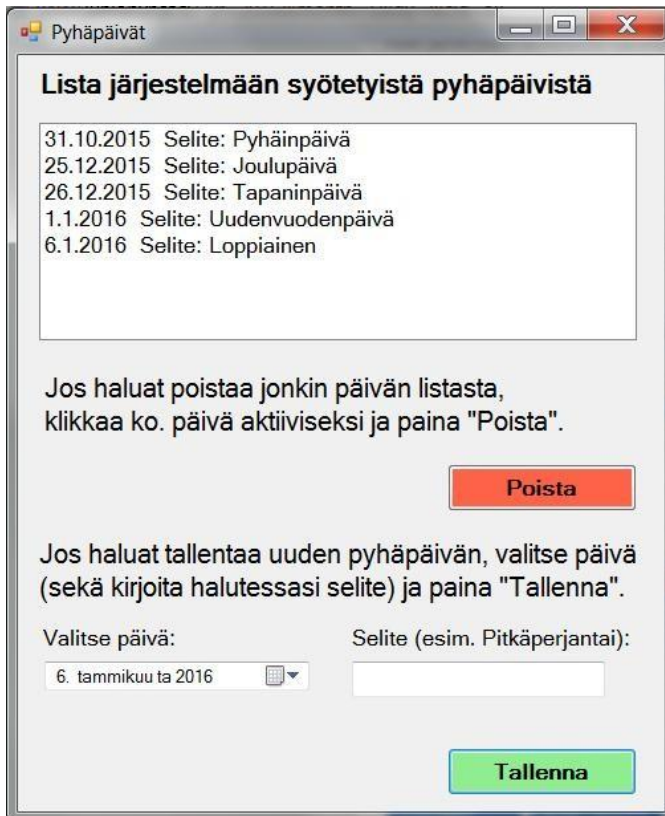
5.1 Pyhäpäivät

Sunnuntait näkyvät punaisella kalenterinäkyvässä (kuva 11).

		2.8.2015	3.8.2015	4.8.2015	5.8.2015	6.8.2015	7.8.2015	8.8.2015	9.8.2015	10.8.2015	11.8.2015
		vko 31	vko 32	vko 32	vko 32	vko 32	vko 32	vko 32	vko 32	vko 33	vko 33
Fuone	S...	Sunnu...	Maana...	Tiistai	Keskivi...	Torstai	Perjantai	Lauantai	Sunnu...	Maana...	Tiistai
F1	1										
	2										
F2	1										
	2										
F3	1										

Kuva 11. Sunnuntait sekä järjestelmään syötetyt pyhäpäivät näkyvät punaisella kalenterinäkyvässä.

Järjestelmään voi syöttää pyhäpäiviä ja ne tulevat myös näkymään kalenterinäkyvässä punaisella. Pyhäpäivien syöttö ominaisuus löytyy sijainnista ”Tiedosto -> Pyhäpäivien syöttö” (kuva 12).



Kuva 12. Pyhäpäivien syöttö.

Pyhäpäivien syöttö tapahtuu siten, että valitaan haluttu päivä ("Valitse päivä:") ja kirjoitetaan lisäksi halutessa selite ja painetaan "Tallenna". Tallennettu päivä ilmestyy ylläolevan listan loppuun.

Kun halutut päivät on syötetty, sulje "Pyhäpäivien syöttö" -ikkuna ja päivitä kalenterinäkömä painamalla esim. "Lataa tämänhetkinen ajankohta.." -painiketta.

Päiviä pystyy myös poistaa listalta, jos esimerkiksi on syötetty väärä päivä, klikkaamalla listasta aktiiviseksi poistettava päivä ja painamalla "Poista".

Pyhäpäivät tallentuvat tekstitiedostoon, joka on sijainnissa "C:\Huonevaraussovellus\pyhapaivat_huonevaraussovellus.txt". Ko. tiedostoa ei saa poistaa tai siirtää kyseisestä sijainnista, se on tarkoitettu ainoastaan Huonevaraussovelluksen käyttöön.

5.2 Asiakastyypit ja uudet hakutoiminnot

Asiakkaalle voi antaa oman koodin (Asiakastyypit):

- Asukas/Intervalli (A)
- Invalidit (I)
- Veteraanit (V)
- Muut (M).

Asiakastyypin voi asettaa asiakkaan tiedoissa (Tiedosto -> Uusi tietue / Tiedon muokkaus -> Asiakkaat) (kuva 13).

The screenshot shows a window titled "Asiakkaat" with the following content:

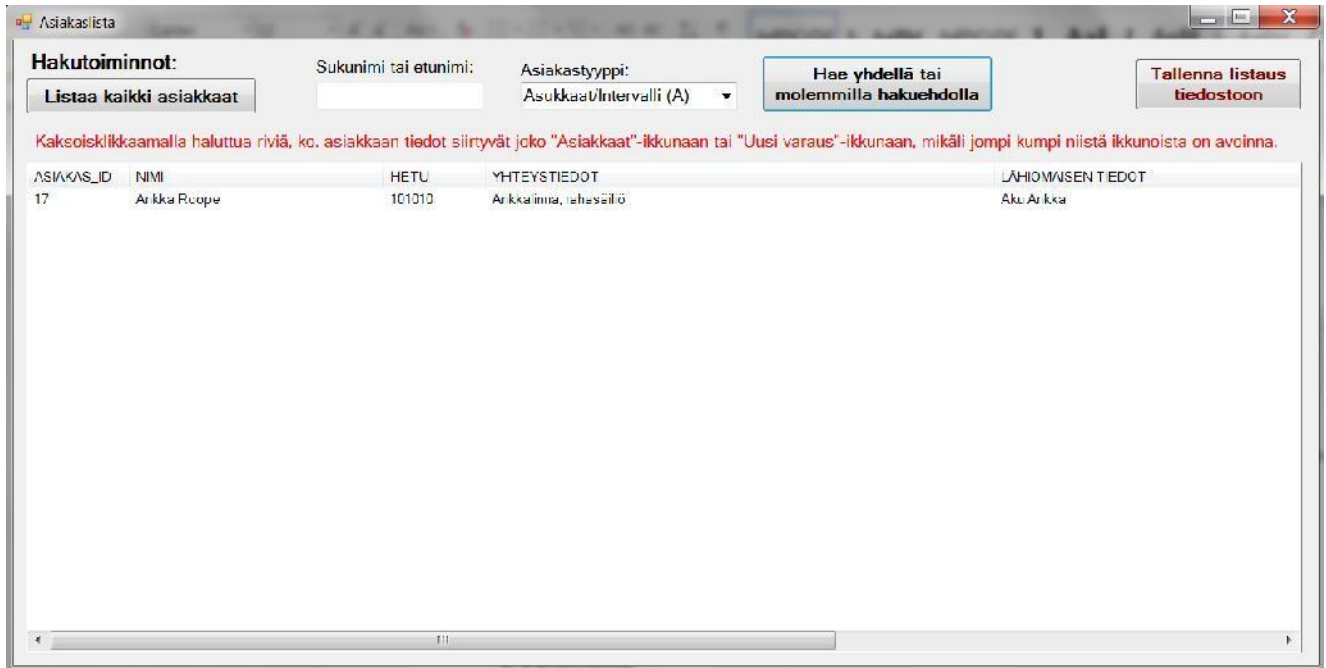
- Valitse asiakas:** A dropdown menu showing "Reipas Risto --- ICTU: 405060 --- AsiakasN:o: 14".
- AsiakasID:** Input field with "14" and a "Laaja asiakashaku" button.
- Etunimi:** Input field with "Risto".
- Sukunimi:** Input field with "Reipas".
- Henkilötunnus:** Input field with "405060".
- Osoite (kirjoita tähän ainoastaan lähiosoite ja postiosoite allekkain):** Input field with "Puolenhehtaarinmetsä 1" and "00000 PM".
- Lähiomaisen tiedot:** An empty input field.
- Asiakastyypin:** A dropdown menu with "Veteraanit (V)".
- Muita tietoja (kirjoita tähän myös muut yhteystiedot esim. puh):** Input field with "Asiakkaan_Tyyppi:V.". Below this field, the text "Asiakkaan huonevaraukset:" is visible.
- Asiakkaan huonevaraukset:** A text area containing "ASIAKAS ON TÄLLÄ HETKELLÄ TALOSSA: 1. varaus | Aika: 2.8.2015 - 26.9.2015 | Huone: H5 | Sänky: 1 | Status: itse maksava".
- Asukas-statusella käytetyt** päivät tänä vuonna: 0
- Asukas-statusella tulevat** päivät tänä vuonna: 0
- Buttons: "Tyhjennä kentät", "Tallenna asiakastiedot", and "Poista".

Kuva 13. Asiakkaat-ikkuna ja uusi ominaisuus "Asiakastyypin".

Asiakastyypin voi valita valintalistasta, mutta se ei ole välttämätön sovelluksen toimimisen kannalta.

Kun asiakastyypin valitaan ja sitten tallennetaan asiakastiedot, asiakkaan tyyppi tulee näkyviin "Muita tietoja"-kohtaan, sillä sovellus lukee asiakastyypin tästä. Sillä ei ole merkitystä muutoin ohjelman käytön kannalta. Jos myöhemmin tallennetaan (eli päivitetään) muita uusia tietoja kohtaan "Muita tietoja", voi tuon "Asiakkaan_Tyyppi.." -lauseen poistaa, sillä järjestelmä lisää sen automaattisesti ko. kentän alkuun, mikäli valintalistassa vaan on valittuna haluttu asiakastyypin. Ja jos muutettaessa tietoja ko. kohtaan "Asiakkaan_Tyyppi.." -lause jätetään poistamatta, se kopioituu siihen toistamiseen, joka ei tosin sekään aiheuta mitään ongelmaa ohjelman toimivuuteen, vaan on lähinnä visuaalinen haitta.

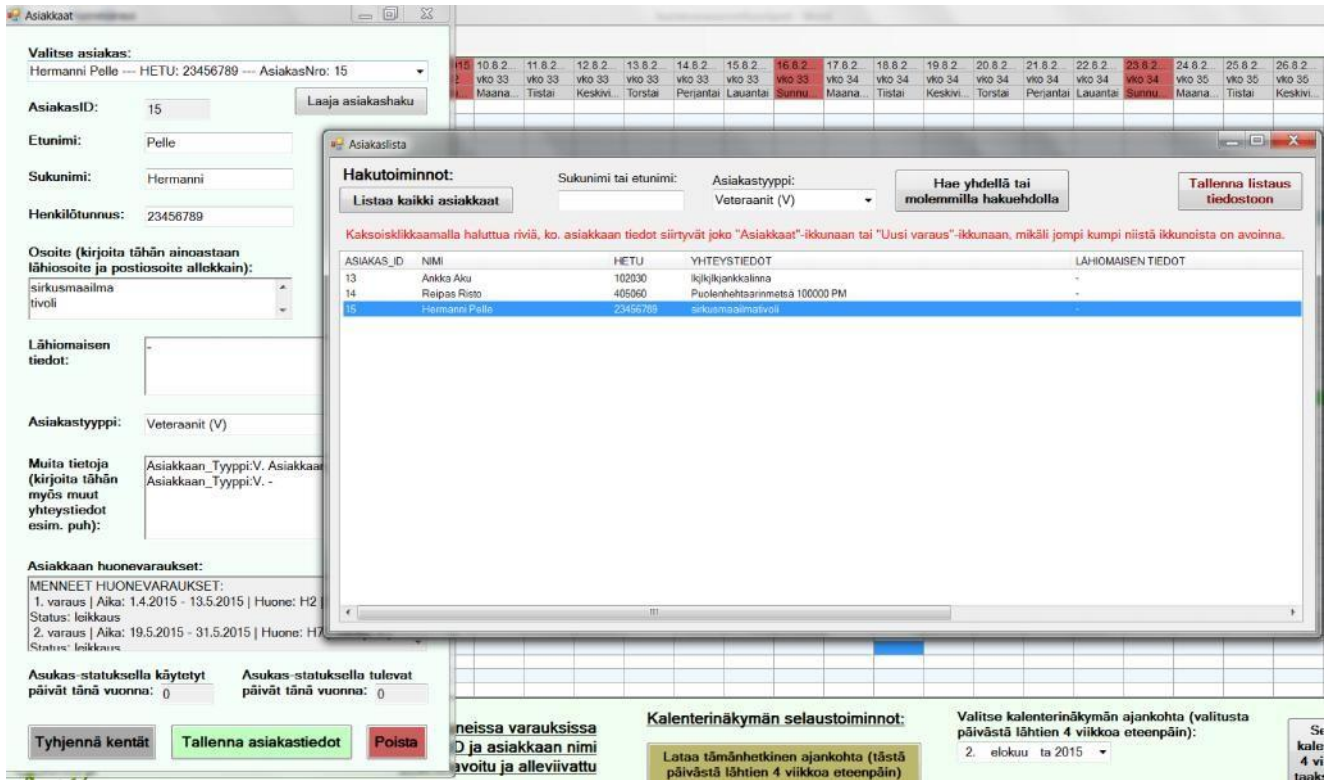
Asiakastietojen hakutoimintoja on parannettu siten, että asiakkaita voi nyt hakea myös asiakastyypin mukaan, etunimen mukaan, sukunimen mukaan tai asiakastyypin ja etunimen/sukunimen yhdistelmällä. Asiakastietojen haku löytyy sijainnista ”Tiedosto -> Hae tietoja -> Asiakastiedot” (kuva 14).



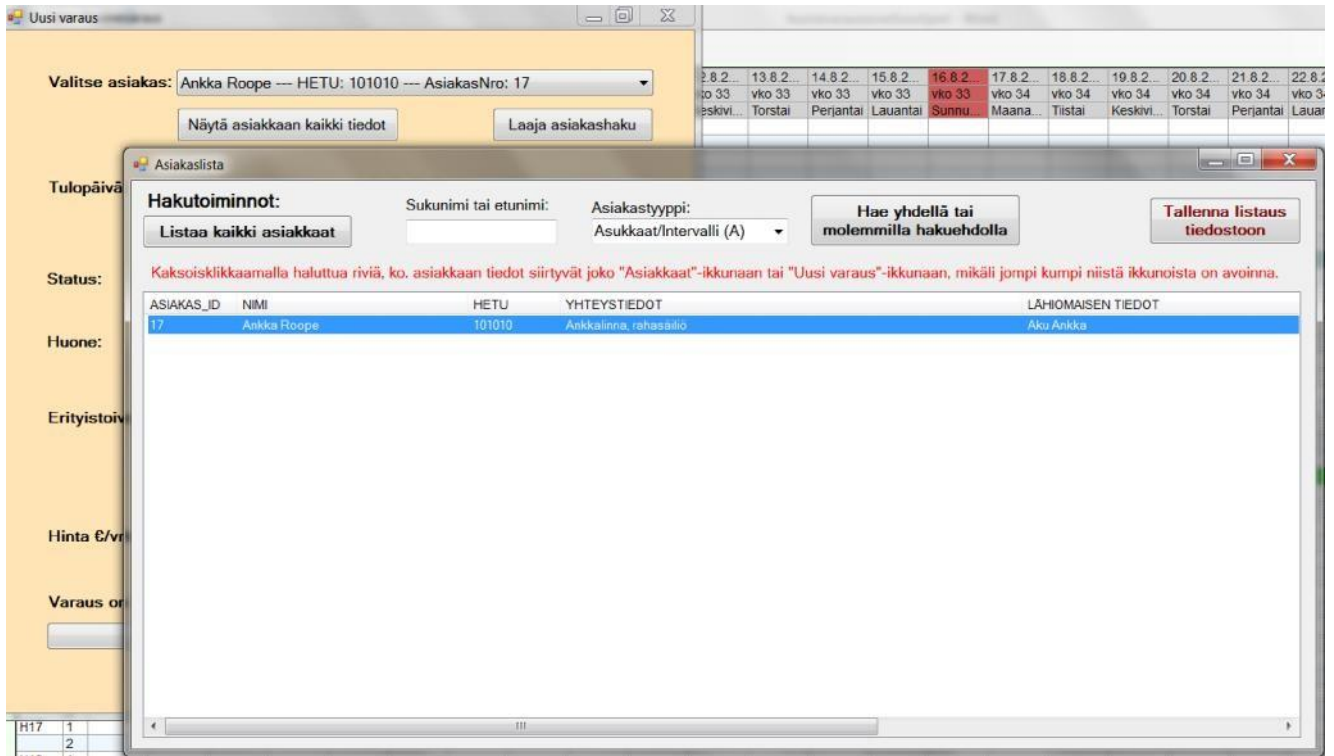
Kuva 14. Asiakaslista, jossa paremmat hakutoiminnot.

Voidaan esim. hakea kaikki asiakkaat, joiden asiakastyypin on ”Asukas/Intervalli”, valitsemalla ko. asiakastyypin valintalistasta ja painamalla sitten ”Hae yhdellä tai molemmilla hakuehdoilla”.

Lisäksi asiakas voidaan liittää ”Asiakaslistan” kautta joko ”Asiakkaat”-ikkunaan tai ”Uusi varaus”-ikkunaan siten, että jompikumpi ko. ikkunoista on auki ”Asiakaslista”-ikkuna lisäksi, ja kaksoisklikkaamalla halutun asiakkaan riviä asiakaslistassa, ko. asiakkaan tiedot siirtyvät ”eteenpäin” (kuva 15 ja 16).



Kuva 15. Asiakastiedon siirto asiakaslistasta "Asiakkaat"-ikkunaan kaksoisklikkauksella.



Kuva 16. Asiakastiedon siirto asiakaslistasta "Uusi varaus"-ikkunaan kaksoisklikkauksella.

Sekä "Asiakkaat"-ikkunasta että "Uusi varaus"-ikkunasta pääsee suoraan "Asiakaslista"-ikkunaan klikkaamalla nappia "Laaja asiakashaku".

5.3 Kutsukirje

Klikkaamalla kalenterinäkylässä haluttua huonevarausta, esiin tulee ikkuna "Varauksen tiedot", josta voi tulostaa kutsukirjeen painamalla nappia "Tulosta kutsukirje". Sen jälkeen esiin tulee ikkuna, josta voi valita tulostetaanko suomenkielinen vai ruotsinkielinen kutsukirje. Seuraavaksi aukeavasta ikkunasta valitaan oikea tulostin ja painetaan OK. Tämän jälkeen aukeaa tulostuksen esikatselu, josta voi tulostaa kutsukirjeen painamalla ikkunan vasemmassa yläreunassa olevaa tulostimen kuvaa.

Jotta kutsukirjeeseen tulostuisi asiakkaan osoite oikein, tulisi "Asiakkaat"-ikkunassa, asiakkaan tiedoissa kirjoittaa "Osoite"-kenttään (entinen Yhteystiedot-kenttä) ainoastaan lähiosoite ja postiosoite (allekkain siten, että ensimmäisellä rivillä lähiosoite ja toisella rivillä postiosoite). Muut yhteystiedot voidaan kirjoittaa esim. kohtaan "Muita tietoja".

5.4 Asukas-statuksella käytetyt päivät

Kunkin asiakkaan asukas-statuksella käytetyt päivät nähdään sekä "Asiakkaat"-ikkunassa että painamalla nappia "Näytä asiakkaan kaikki tiedot" uutta varausta tehtäessä "Uusi varaus"-ikkunassa tai tarkasteltaessa tehtyä varausta "Varauksen tiedot"-ikkunassa.

"Asukas-statuksella käytetyt päivät tänä vuonna" tarkoittaa päiviä jotka ovat jo menneet, mukaan lukien tämänhetkinen päivä, eli toteutuneet päivät, jotka asiakas on ollut talossa kuluvana vuotena asukas-statuksella.

"Asukas-statuksella tulevat päivät tänä vuonna" tarkoittaa kuluvana vuotena tulevia päiviä, joihin asiakkaalle on tehty suunniteltu huonevaraus asukas-statuksella.

5.5 Varaus toteutuneeksi (autom. kysymys)

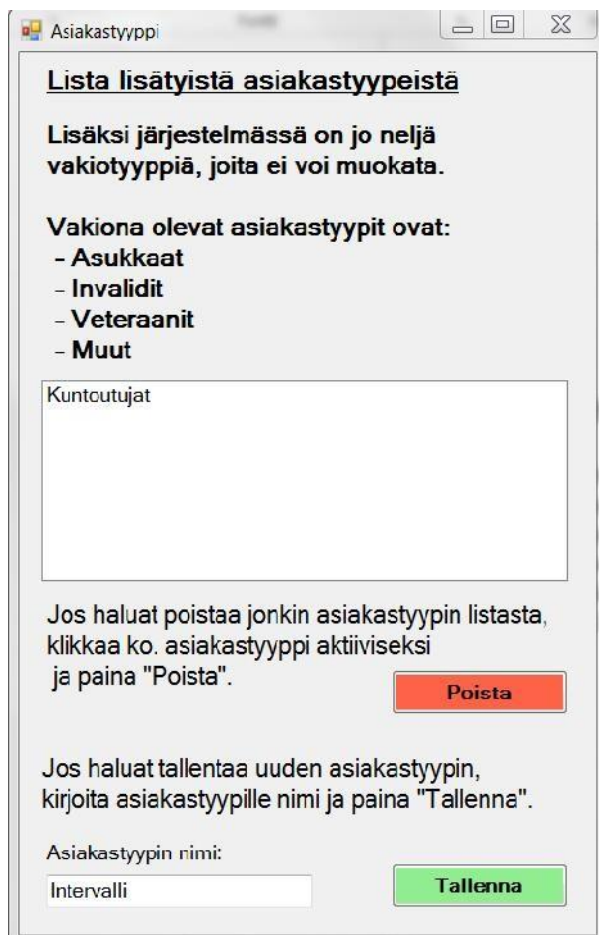
Jos järjestelmästä löytyy huonevarauksia, jotka ovat merkitty suunnitelmiksi ja joiden tulopäivä on kuluva päivä (tai jo aiempi), sovellus kysyy käynnistettäessä (tai kun kalenteria selataan), että haluatko muuttaa huonevarauksenvarauksen toteutuneeksi.

Mikäli vastaa "kyllä", ko. varaus muutetaan toteutuneeksi heti, ja mikäli vastaa "ei", ohjelma kysyy haluatko poistaa varauksen. Mikäli vastaa "kyllä", varaus poistetaan kun seuraavan kerran päivitetään kalenteria, eli painetaan jotain selauspainiketta, tai kun sovellus käynnistetään uudelleen. Mikäli vastaa poistokysymykseen "ei", järjestelmä toistaa kysymykset joka kerta kun sovellus käynnistetään tai kun kalenteria selataan.

6 Ohjelman uudet ominaisuudet (9.8.2015)

Asiakastyyppejä voi luoda lisää (vakiona ovat: Aukkaat, Invalidit, Veteraanit ja Muut). "Asiakastyypin lisäys"-ominaisuus sijaitsee "Tiedosto -> Asiakastyypin lisäys". Asiakastyypin lisäys tapahtuu kirjoittamalla haluttu asiakastyypin nimi tekstikenttään "Asiakastyypin nimi:" ja painamalla "Tallenna". Asiakastyyppejä voi luoda useita. Asiakastyypin poisto tapahtuu klikkaamalla aktiiviseksi haluttu asiakastyypin asiakastyypin listalta ja painamalla "Poista". (ks. Kuva 17).

Asiakastyyppejä voi antaa asiakkaille tämän käyttöohjeen kohdan 5.2. mukaisesti.



Kuva 17. Asiakastyypin-ikkuna.

Huonevarauksia voi hakea asiakkaan etu/sukunimen ja/tai asiakastyypin mukaan myös "Varauslista"-ikkunasta, eli sijainnista "Tiedosto -> Hae tietoja -> Varaustiedot". Uudet hakutoiminnot ovat samankaltaiset kuin tämän ohjeen kohdassa 5.2. kerrotaan. Haluttu huonevaraus voidaan viedä "Varauksen tiedot"-ikkunaan lähempään tarkasteluun ja/tai muokattavaksi, kaksoisklikkaamalla varauslistasta haluttua varausta.