

Varausjärjestelmän tekninen suunnitelma

Topias Pöllänen 597348

Information and Service Management (BIZ) 2vsk.

12.3.2019

1. Ohjelman rakennesuunnitelma

Varausjärjestelmä sisältää useita erilaisia resursseja ja olioita, joilla on useita luokkien välisiä suhteita. Ohjelma sisältää seuraavat luokat, jotka on myöhemmin avattu UML-kaaviossa:

- Calendar (kalenterin hallinta)
- Customer (asiakasolion luominen)
- Reservation (varauksen tekeminen asiakkaalle)
- Hotel (olio kaikille huoneille ja palveliulle)
- Room (varattavissa olevan huoneen tiedot)
- Service (lisäpalveluiden luonti)
- GUI (graafisen käyttöliittymän toteutus PyQt- käyttöliittymällä)

Hotel-olioon kerätään kaikki hotellissa olevat huoneet, asiakkaat, varaukset ja hotellin tarjoamat palvelut. Luokkien keskinäiset suhteet on avattu tarkemmin liitteenä olevassa UML-kaaviossa.

2. Käyttötapauskuvaus

Yleisin kuvitteellinen käyttötapaus ohjelmalle on uuden huonevarauksen tekeminen asiakkaalle. Aluksi käyttäjän on annettava ohjelmalle omat yhteystietonsa, jotka sisältävät nimen, osoitteen, puhelinnumeron ja sähköpostiosoitteen. Näin saadaan luotua uusi Customer-luokan olio, jolle voidaan kohdistaa varauksia. Kun yhteystiedot on täytetty, asiakas valitsee painikkeista huonevarausta vastaavan painikkeen. Tämän jälkeen ohjelma aloittaa Reservation-olion luomisen. Tässä vaiheessa käyttäjältä kysytään varauksen päivämäärät muodossa pp.kk.vvvv sekä yöpyjien ja varattavien huoneiden määrä. Reservation-olion kautta tarkistetaan huoneen saatavuus Room- ja Calendar olioiden kautta ja estetään päällekkäisten varauksien tekeminen. Ohjelma ilmoittaa käyttäjälle varauksen onnistumisesta tai epäonnistumisesta.

3. Algoritmit

Varausjärjestelmä sisältää hyvin vähän matemaattisia kaavoja. Vapaana oleva huone tai palvelu etsitään yksinkertaisesti käymällä resurssit läpi listan mukaisessa järjestyksessä. Lopputuloksena on joko onnistunut tai epäonnistunut varaus. Tarkistin etsii päällekkäisyyksiä varattavan resurssin ajankohdalta Calendar-oliosta. Käytettävissä olevista resursseista laskettavat mahdolliset lisätiedot ovat yksinkertaisia kerto-, jako- ja yhteenlaskuja.

4. Tietorakenteet

Hotel-luokassa säilytetään koottuna tiedot kaikista hotellin asiakkaista, huoneista, palveluista ja varauksista. Eri olioiden sisältämät tiedot sarakkeet ovat vakioita, mutta esimerkiksi yhden asiakkaan varausten määrä voi muuttua. Asiakastiedot voidaan esittää taulukossa ja hotelliin tehdyt varaukset muuttuvassa listassa asiakaskohtaisesti. Projektin edetessä päätös sanakirjojen, listojen ja muiden tietorakenteiden käytöstä varmistuu lopullisesti.

5. Aikataulu

Ohjelma toteutukseen on tällä hetkellä noin 4-5 viikkoa aikaa, mikä tarkoittaa noin 20 tunnin työmäärää viikottain. Ensimmäisen kahden viikon aikana on tarkoituksena saada ohjelman kriittisimmät luokat kuten Customer, Reservation, Room ja hotel toimimaan vähintään tekstipohjaisella käyttöliittymällä. Toisella viikolla alkaa graafisen käyttöliittymän integrointi jo toimiviin luokkiin. Puoliväliin mennessä uskon, että ohjelman luokkien toteutus on vienyt 30 tuntia ja graafisen käyttöliittymän toteutus 10 tuntia. Ohjelman perustoimintojen testaus suoritetaan hyvissä ajoin, jotta luokkien välisten suhteiden toiminta voidaan varmistaa. Jos checkpointiin mennessä ohjelman perustoiminnot eivät ole kunnossa, niin loppuprojektin fokus tulee olemaan ohjelman toimimisessa eikä niinkään graafisessa käyttöliittymässä. Viimeistään kolmannella viikolla on tarkoitus toteuttaa viimeisetkin suunnitellut luokat ja ominaisuudet ohjelmaan. Tämä vie aikaa noin 20 tuntia. Viimeisten viikkojen aikana on tarkoitus keskittyä graafisen käyttöliittymän ehostamiseen ja ohjelman perusteelliseen testaamiseen erilaisten käyttötapausten kautta. Testaamista ja ehostamista varten on varattu noin 20 tuntia. Tuntimäärä saattaa kasvaa, mikäli ohjelman kehitys ei etene halutulla tavalla.

6. Yksikkötestaussuunnitelma

Heti projektin alkuvaiheessa testataan ohjelman resurssien luomisen onnistuminen. Jos olioiden luonnissa on ongelmia, niin varausten tiedot eivät päivyty keskitetysti Hotel-olioon eikä Reservation-olion suorittama päällekkäisten varausten tarkistus onnistu. Käyttäjän antamat syötteet eivät saa sisältää ylimääräisiä pilkkuja, jotta tietojen tallennus ja lukeminen onnistuu käyttäen csv-tiedostoa. Asiakastietoja syöttäessä nimi saa sisältää vain kirjaimia ja välilyöntejä, osoite on vapaamuotoinen merkkijono ilman pilkkuja, puhelinnumero on merkkijono numeroita ja sähköpostiosoite on merkkijono, joka sisältää @-merkin eikä pilkkuja. Varausta tehdessä tulee testata, että Reservation-luokan metodi palauttaa onnistuneen varauksen tai kertoo, että päivämäärän tai keston syöte on virheellinen. Virheellisiä syötteitä sekä päällekkäisiä varauksia on helppo testata muuttamalla syötteitä.

7. Kirjallisuusviitteet ja linkit

Etsin tietoa Googlen avulla useilta eri internetsivulta kohdatessani ongelmia ja aloittaessani koodaamaan uutta ohjelman osa-aluetta. Tähän mennessä olen perehtynyt seuraaviin verkkosivuihin ohjelman toteutusta varten:

Pythonin calendar-luokka

<https://www.guru99.com/calendar-in-python.html>

Useita PyQt:in käyttöön liittyviä ohjeita

<https://www.tutorialspoint.com/pyqt/index.htm>

PyQt:in calendar widgetin käyttöön liittyviä ohjeita

<http://pyqt.sourceforge.net/Docs/PyQt4/qcalendarwidget.html>

Pythonin perusluokkakirjastojen API-kuvauksia

<https://docs.python.org/3/library/calendar.html>

<https://docs.python.org/3/library/datetime.html#module-datetime>

8. Liitteet

1. UML-kaavio varausjärjestelmästä

