

1. Ohjelmoinnin peruskurssi Y2: Projekti - Ajanvarausjärjestelmä

Topias Pöllänen, 597348
Tieto- ja palvelujohtaminen, BIZ
22.4.2019

2. Yleiskuvaus

Projektsuunnitelman mukaisesti olen tehnyt ajanvarausjärjestelmän, joka yhdistää kalenteri-, asiakas- ja resurssitietoja. Toteutin ohjelman mielestäni keskivaikealla vaikeusasteella, jonka lisäksi ohjelmassa on muutama vaikean tason lisäominaisuus. Ohjelma estää päällekkäisten varauksen tekemisen ja käy läpi kaikki hotellin huoneet varausta tehdessä. Vapaiden huoneiden/päivien visualisoinnissa onnistunut haluamallani tavalla, mutta ohjelmasta voi hakea historiatietoja asiakas-, huone ja palvelukohtaisesti. Ohjelmaa käytetään PyQt-käyttöliittymällä toteutetussa graafisessa käyttöliittymässä.

3. Käyttöohje

Ohjelma käynnistetään avaamalla tiedostot esimerkiksi Eclipsessä ja suorittamalla main.py tiedosto. Ohjelman voi suorittaa myös komentoriviltä suorittamalla main.py tiedosto. Jos customers.csv tiedosto on tyhjä, niin ensimmäiseksi ohjelmaan tulee syöttää asiakas. Luotu asiakas ilmestyy Room- ja Service-välilehtien valikkoihin. Varauksen voi viedä loppuun painamalla OK ja painamalla Cancel, varaus nollaantuu. Tehtyjä varauksia voi tarkkailla History and statistics- painikkeesta aukeavalla välilehdellä. Kyseisellä välilehdellä on mahdollista etsiä asiakkaan tekemiä huonevarauksia ja service-hausta näkee myös asiakkaan ostamat palvelut.

4. Ulkoiset kirjastot

Käytin ohjelmassa PyQt5- ja datetime-kirjastoja ja näiden käyttämisestä olen kysynyt assistentiltani. PyQt-kirjastolla loin ohjelman graafisen käyttöliittymän ja kyseisen kirjaston eri ominaisuuksista oli paljon hyötyä käyttöliittymän rakentamisessa. Datetime kirjastoa tarvitsin varauksien päällekkäisyyden tutkimiseen ja varauksien oikeanlaiseen tallentamiseen.

5. Ohjelman rakenne

Ohjelman luokkien rakenne vastaa kohtalaisen hyvin teknisen suunnitelman yhteydessä palauttamaani UML-kaaviota. Jokainen luokka ohjelmassa on sama kuin tekemässäni kaaviossa, mutta osa luokkien metodeista sekä luokkien välisistä assosiaatioista ovat muuttuneet. Ohjelman käyttöliittymän kuvaamiseen käytetään gui.py tiedostossa useita eri luokkia, kuten UI_MainWindow sekä Calendar_widget. Ohjelman ydin on Hotel-luokka, joka luodaan valmiin csv-tiedoston pohjalta. Tämän jälkeen tuohon yksilöityyn Hotel-olioon voidaan luoda asiakkaita ja varauksia sekä etsiä tietoa. Hotel-luokkaan liittyy Customer-, Room-, Service-, Reservation- ja Calendar- luokat. Osalla luokista ei ole juurikaan omia metodeita tietojen noutamisen lisäksi. Calendar-luokka ja sen check_dates-metodi on tärkeässä uuden huonevarauksen tekemisessä. Check_dates metodia kutsutaan Hotel-luokassa.

6. Algoritmit

Kuten teknisessä suunnitelmassa arvelin, niin ohjelmassa ei ollut kovin matemaattisia algoritmeja. Calendar-luokan `check_dates` funktiossa on ohjelman olennaisin funktio, joka tutkii varausten päällekkäisyyttä/tarkastaa vapaan huoneen. Metodissa `date_a` = varauksen alkupäivä, `date_b` = varauksen päättymispäivä, `date_c` on huoneella jo olevan varauksen alkamispäivä ja `date_d` taas kuvaa huoneella jo olevan tietyn varauksen päättymispäivää. Algoritmin matemaattinen muoto on metodissa seuraavanlainen: $(c_date < a_date \text{ and } d_date \leq a_date)$ tai $(c_date \geq b_date \text{ and } d_date > b_date)$. Metodin tarkoituksena on testata, että uuden varauksen päivät eivät osu ns. keskelle vanhaa huonevarausta.

Check-dates metodiin liittyen ohjelmalle tärkeä algoritmi on Hotel-luokan `make_reservation` metodi. Nämä kaksi metodia käyvät for-loopeilla läpi kaikki huoneet ja niihin liittyvät varaukset. Ohjelma varaa ensimmäisen vastaantulevan huoneen, jossa ei ole päällekkäisiä varauksia.

7. Tietorakenteet

Mielestäni listat toimivat omassa ohjelmassani listojen tallentamiseen oikein hyvin. Ohjelman kannalta erittäin olennaista oli, että listoihin tallennettuja olioita voi käsitellä eri funktioissa eri tarpeita varten. Muuttuvissa listoissa tallensin usein olioita, joiden kautta käsittelin muuttuvaa tietoa.

8. Tiedostot

Ohjelma käsittelee kahta csv-tiedostoa, jotka ovat nimeltään `resources.csv` ja `customers.csv`. Molemmat näistä ovat tekstitiedostoja, joissa eri tieto/arvot ovat erotettu toisistaan pilkulla. Resources tiedostossa on otsikko "Resources", jonka jälkeen tulee tietoa hotellin nimestä, huoneista ja hotellin tarjoamista palveluista. Noudattamalla samaa tekstimuotoilua, kun tiedostossa jo olevat rivit, ohjelmaan voi lisätä enemmän huoneita ja palveluita. Customers.csv tiedosto tallentaa kaikki luodut asiakkaat ja tehdyt varaukset ja näiden tietojen lukeminen ohjelman sisällä on mahdollista, vaikka ohjelman sulkisi välissä. Ohjelma on palautettaessa sellaisessa muodossa, että assistentti voi sitä heti kokeilla.

9. Testaus

Ohjelma suoriutuu teknisessä suunnitelmassa suunnitelluista keskeisistä testeistä. Päästyäni vauhtiin ohjelman kehittämisessä, testien tekeminen ohjelmaa rakentaessa pääsi hieman unohtumaan. Loppua kohden aloin tekemään ohjelmalle enemmän testejä. PyQt-käyttöliittymän avulla moni virheellisiin syötteisiin yms. liittyvistä ongelmista pystytään eliminoimaan asettamalla syötteille rajoja tai kriteereitä. Tästä syystä en nähnyt tarpeelliseksi testata kaikkia virheitä, jotka voivat syntyä ainoastaan koneellisesti eikä graafisen käyttöliittymän kautta.

Ohjelmalle tärkeimmän ominaisuuden eli varauksen testaaminen onnistui helposti tekemällä 4 esimerkki varausta. Huoneita on 3, joten neljäs varaus saa arvokseen False. Tämä palauttaa graafisessa käyttöliittymässä ilmoituksen epäonnistuneesta varauksesta. Tämän lisäksi ei ole esimerkiksi mahdollista luoda tyhjää asiakasta.

10. Ohjelman tunnetut puutteet ja viat

Minulla tuli hieman kiire Services-luokkaan liittyvien varausten toteuttamisen kanssa, joten tällä hetkellä Service-olioihin tallentuu customer-olion sijasta asiakkaan nimi, mikä vaikeuttaa asiakkaan yksilöintiä. Myöskin tästä syystä history and statistics-välilehdellä ohjelmassa ostetut palvelut eivät asiakaskohtaisessa yhteenvedossa. Asiakkaat ovat kuitenkin löydettävissä myös hausta hakemalla palvelukohtaista historiaa.

Olisin halunnut myös toteuttaa paremman visuaalisen kalenteri-ikkunan, jonka kautta olisi nähnyt suoraan jo varatut päivät, joille ei voi enää tehdä varausta. History- and statistics ikkunan alareunassa olevissa "progressbareissa" on ollut satunnaisia eroja reaalityöntekseen, mutta en ole onnistunut paikantamaan eron syytä.

11. 3 parasta ja 3 heikointa kohtaa

Parhaat:

1. PyQt-käyttöliittymä: Käyttöliittymä on yksinkertainen ja siinä on hyödynnetty monipuolisesti kirjaston tarjoamia mahdollisuuksia
2. Ohjelman rakenne: Noudatin suunnitelmani rakennetta alusta asti ja mielestäni se toimii hyvin.
3. History- and statistics painikkeesta aukeavan ikkunan hakurakenne: Tämän toteuttamiseen meni kohtalaisen paljon aikaa lukuisten pienien virheiden vuoksi ja olen erittäin tyytyväinen ikkunan lopputulokseen

Heikoimmat:

1. Services-varaukset: Palveluiden varausten toteutus jäi hieman kesken ja toteutin sen suoraan Service- luokkaan, enkä Reservation-olion kautta.
2. Graafisen käyttöliittymän ikkunoiden koko: Ikkunoiden kokoa ei voi aina suurentaa, koska haluamani ikkunan luominen tietyllä layoutilla ei ollut mahdollista, enkä ollut valmis tinkimään tietynlaisesta näkymästä.
3. Hieman toistoa koodissa: Ohjelmassa esiintyy hieman toistoa koodissa. Olen ratkaissut näin muutama ongelman koodissa, kun en ole keksinyt, että miten toteuttaisin koodin toisin.

12. Poikkeamat suunnitelmasta

Aloitin ohjelman tekemisen hyvissä ajoin ja noudatin tekemääni aikataulua kohtalaisen hyvin. Ohjelma noudattaa hyvin pitkälti sitä rakennetta, minkä projektin alkuvaiheessa suunnittelin. Ajankäytön arvio osui kohtalaisen hyvin oikeaan, mutta oli kuitenkin alakanttiin loppujen lopuksi. Toteutusjärjestykseni oli hyvä ja eteneminen tuntui järkevältä.

13. Toteutunut työjärjestys ja aikataulu

Päiväkohtaisen aikataulun muistaminen on äärimmäisen hankalaa, mutta GitLabia tukena käyttäen huomaa, että aloitin projektin koodaamisen 14.3.2019. Maaliskuun loppuun asti committeja on tullut gitlabiin tasaiseen tahtiin vähintään kerran viikossa. Huhtikuun alussa tenttiviikko hieman vähensi työpanostani projektiin. Huhtikuun puolivälissä parantelin käyttöliittymää ja tutkin muutamaa samaa

ongelmaa, joten en nähnyt tarpeelliseksi usean commitin tekemistä. Sain ohjelman perusominaisuudet ajoissa valmiiksi, joten päätin käyttää aikaa lisäominaisuuksien tekoon ja graafisen käyttöliittymän käyttömukavuuden parantamiseen.

14. Arvio lopputuloksesta

Olen tyytyväinen suunnittelemaani ja kehittämäni ohjelmaan. Se ei ole täydellinen, mutta se on yksinkertainen ja toteuttaa siltä vaaditut asiat. Hyödynsin mielestäni hyvin ja monipuolisesti PyQt-kirjaston tarjoamia graafisen käyttöliittymän ominaisuuksia. Tämä ehkä paikkaa sitä, että ajanvarausjärjestelmässä ei ollut hirveästi mahdollisuuksia kehittää matemaattisesti vaativia ja poikkeuksellisia metodeita. Projektin aihe oli sopiva sivuaineopiskelijalle kauppakorkeakoulun puolelta. Tosin on todettava, että omien pääaineopintojen ja esimerkiksi juuri ohjelmointi Y2-kurssin yhteen soveltaminen ei ole aina ollut helppoa.

Service-luokkaan voisi tulevaisuudessa kiinnittää vielä enemmän huomiota. Tämän lisäksi ohjelma olisi muokattavissa sen laiseksi, että sieltä voidaan poistaa varauksia. Voi olla, että joissain tilanteissa olisi ollut järkevämpää käyttää listojen sijaan esimerkiksi sanakirjoja. Ohjelman rakenne sopii erittäin hyvin laajennusten tekemiseen. Ohjelma rakentuu Resources-tiedoston ympärille, jonka vuoksi usealla resources-tiedostolla esimerkiksi hotelliketju voisi hallita varauksiaan ohjelman avulla.

15. Viitteet

Lukuisten eri Qt-luokkien kuvaukset

<https://doc.qt.io/archives/index.html>

datetime-kirjaston kuvaus

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

Pythonin calendar-luokka

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

Useita PyQt:in käyttöön liittyviä ohjeita

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

PyQt:in calendar widgetin käyttöön liittyviä ohjeita

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

QListView-ohjeita PyQt:in käyttöön

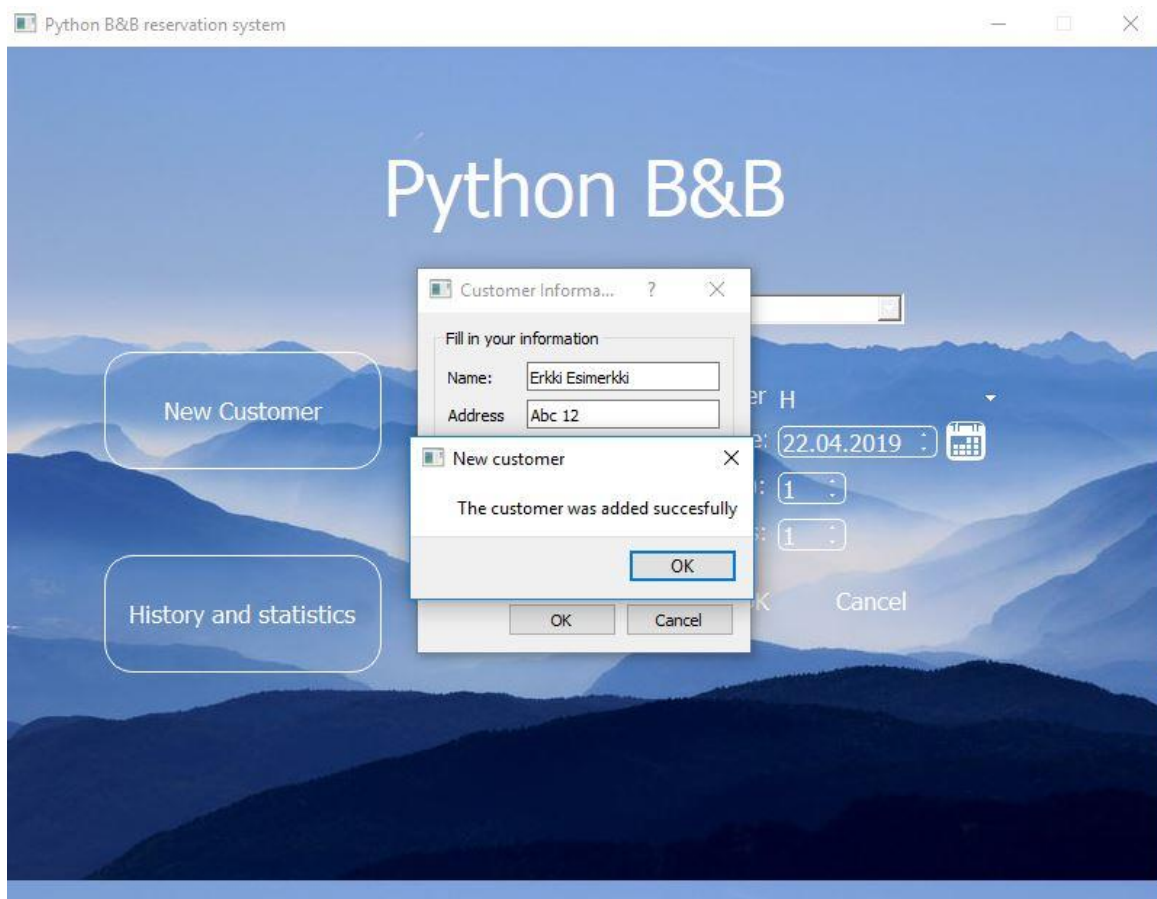
<https://www.pythoncentral.io/pyside-pyqt-tutorial-qlistview-and-qstandarditemmodel/>

Esimerkki Youtube-video

<https://www.youtube.com/watch?v=dRRpbDFnMHI>

16. Liitteet

Oheessa 3 kuvaa ohjelmasta eri käyttötilanteissa: Asiakkaan lisäys, Varauksen tekeminen ja historian tarkastelu.



☒ Customer ☐ Room ☐ Service

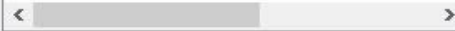
Matti Mallikas, Abc 1 00000 Helsinki, 04400000
Mikko Mallikas, Defg 123, 0503234234, aaa@sdf.co
aaa, aaa aaa aaa, aaa, aaa
Zzz, zz z zz, z, z
H, h h h, h, h
Erkki Esimerkki, Abc 12 Aaaaa 00000, 044000000



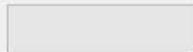
Search

History

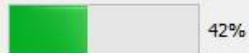
Mikko Mallikas, Defg 123, 0503234234, aaa@sdf.co
Mikko Mallikas, Defg 123, 0503234234, aaa@sdf.co



Capacity for previous week



Capacity for next week



Capacity for next month

