

Olio-ohjelmoinnin perusteet

Harjoitustyö – Olio-Ohjelmointi

Lappeenrannan teknillinen yliopisto
Laskennallisen tekniikan koulutusohjelma

Olio-ohjelmoinnin perusteet
Kevät 2019

Petra Laakkonen
Petra.Laakkonentudet.lut.fi

Jarno Mikkola 0452648
Jarno.Mikkolatudent.lut.fi

SISÄLLYSLUETTELO

1. Ohjelman kuvaus
 - 1.1 Ohjelman kuvaus
 - 1.2 Työvaiheet
2. Ohjelman suunnitelma
 - 2.1 Käsitelmä
 - 2.2 Relaatiomalli
 - 2.3 Luokkakaavio
3. Toiminnallisuudet
 - 3.1 Lista Toiminnallisuuksista
4. Mitä opimme?
5. Terveiset tarkastajalle

Ohjelman kuvaus

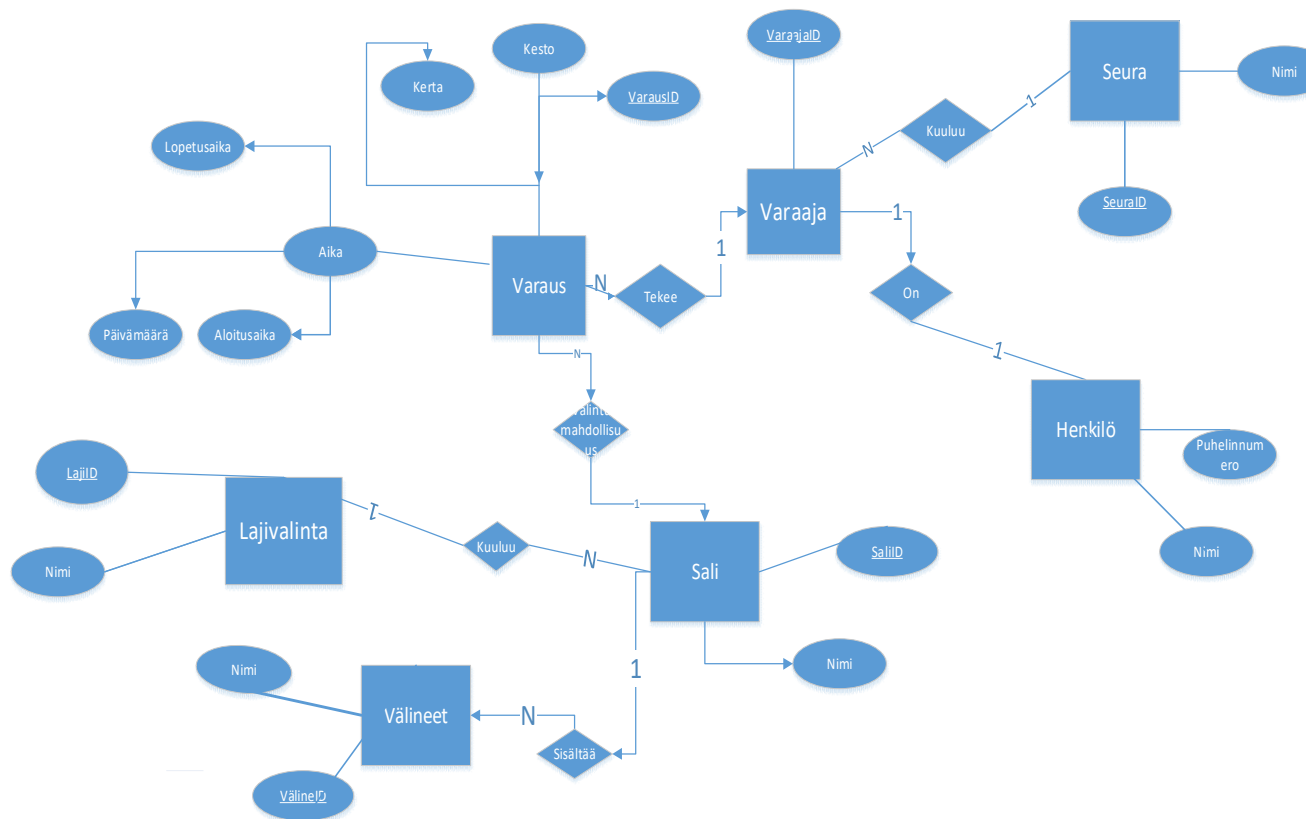
Kehitimme sovelluksen androidille monitoimihallin salien vuorojen varaamiseen. Tällä sovelluksella käyttäjän on mahdollista varata verkosta käsin vuoroja monitoimihallin saleihin, mikä helpottaa huomattavasti varausta, kun ei tarvitse enää soittaa vastaanottoon ja varata sitä kautta. Sovelluksessa käyttäjä pystyy varauksen lisäksi tarkastelemaan varauksia, muokkaamaan varauksia, poistamaan varauksia ja tarkastelemaan varauksia. Tämä mahdollistaa myös sen, että vastaanoton työntekijöille jää arvokasta työaika johonkin muuhun. Sovellusta on mahdollista käyttää älypuhelimella ja tabletilla.

1.2 Työvaiheet

2. Ohjelman suunnitelma

2.1 Käsitemalli

*Pääavaimet ovat alleviivattu ja kohdetyyppien väliset suhteet on merkitty näkyviin.



2.2 Relaatiomalli

*Pääavaimet on relaatiokaavoissa merkattu alleviivauksella ja vierasavaimet ovat alleviivattu katkoviivalla.

Henkilö(VaraajaID, Nimi, Puhelinnumero)
Seura(SeuraID, Nimi)

Varaaja(VaraajaID, SeuraID)
Varaus(VarausID, VaraajaID, SaliID, Kesto, Kerta, Aika(Lopetusaika),
Aika(Päivämäärä), Aika(Aloitusaika))
Lajivalinta(LajiID, Nimi)
Sali(SaliID, LajiID, Nimi)
Välineet(VälineID, SaliID, Nimi)

Henkilön ja Varaajan välinen suhde on 1:1, koska varaaja voi olla vain yksi henkilö. Tästä johtuen transformointi säännön 6 eli binäärisen transformointi suhteen mukaan kohdetyppi Henkilön vierasavain on VaraajaID, joka on samalla myös Henkilön pääavain.

Seuran ja varaajan välinen suhde on 1:N, koska yhdessä seurassa on mahdollista olla monta henkilöä, jotka pystyvät tekemään varauksia. Tästä johtuen käytimme sääntöä 7 eli Binäärisen 1:N –suhdetyyppin transformointi sääntöä, mistä seuraa että kohdetyypin Seura pääavain on SeuraID ja Varaajan pääavain on VaraajaID ja vierasavain on SeuraID.

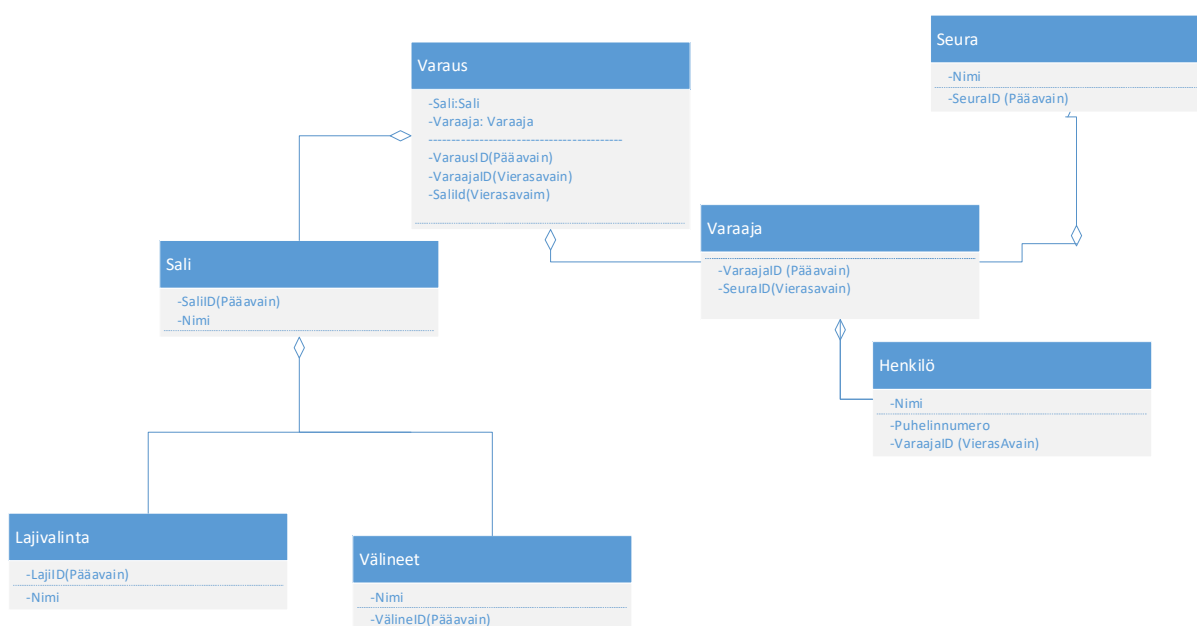
Varaajan ja Varauksen välillä on myös käytetty transformoinnissa sääntöä 7, koska yksi varaaja voi tehdä monta varausta, mutta yhdellä varauksella ei voi olla monta varaajaa. Tästä johtuen varauksen pääavain on VarausID ja vierasavaimia ovat VaraajaID ja SaliID. SaliID on toinen vierasavain, koska Varauksen ja Salin välillä vallitsee myös N:1, koska yhdellä salilla voi olla monta varausta, mutta yhdellä varauksella ei voi olla monta salia. Tietokannan

puolesta on mahdollista tehdä kaksi varausta samaan aikaan samaan paikkaan. Tähän tehdään esto käyttöliittymän puolella.

Lajivalinnan ja Salin välillä vallitsee myös 1:N-suhde, koska yhtä lajia pystyy harrastamaan useassa salissa, mutta yhdessä salissa voi harrastaa vain yhtä lajia. Esimerkiksi sulkapallokentällä voi pelata vain sulkapalloa, mutta sitten tällaisia koulunliikuntasali varauksia voi varata lajilla muut palloilulajit, jossa voi sitten harrastaa palloilulajeja. Tästä johtuen Salin pääavain on SaliID ja vierasavain on LajiID.

Salin ja Välineiden välillä vallitsee 1:N-suhdetyyppi, koska yhdessä salissa on oma välinevarasto, jossa on valittavina erilaisia välineitä, mutta yhtä salia kohden on vain yksi välinevarasto salissa. Tämän seurauksena Välineiden pääavain on VälineID ja vierasavaimena toimii SaliID.

2.3 Luokkakaavio



3. Toiminnallisuudet

3.1 Lista toiminnallisuuksista:

- Mahdollisuus luoda käyttäjä
- Mahdollisuus muuttaa käyttäjän tietoja
- Mahdollisuus tehdä varauksia
- ~~— Mahdollisuus muokata tai poistaa omia varauksia~~
- Mahdollisuus tarkastella varauksia Seuran, Henkilön tai Lajin mukaan.

Lista vielä olevista virheistä ja puutteista

- Ei voi poistaa tai muokata tehtyjä varauksia.
 - o Tämä on tarkoitus korjata. Varauksen poistaminen pitäisi olla helppo työ, muokkaamiseen saattaa mennä hieman aikaa, kun pomppii aktiviteetti-ikkunasta toiseen.
- Ei voi tehdä viikottaista varausta.
 - o Päivämäärien ja kellonaikojen tallentaminen stringeinä oli suuren suuri virhe.
 - o Saman kuukauden sisällä pystyisi tekemään varauksen seuraavalle viikolle, mutta kuukauden tai vuoden vaihtaminen hajoittaa matikan. Niin monta if-lausetta kuukausien eri pituuksista, että itku tulisi.
 - o Vaatii luultavasti datatyyppin vaihdon, mikä varmaan kipuilee muihin asioihin.
 - o Ihän tehtävissä, ainakin jollakin ei niin nätillä purkkamenetelmällä.
- XML/JSON -tiedoston kirjoittaminen
 - o Ei olla vielä perehdytty. Kovaa googlausta, ei pitäisi olla paha.
- Uusi käyttäjä jokaisella ajokerralla
 - o Kun ohjelman ajaa uudelleen, sillä ei ole käyttäjää.

- Pitää tehdä uusi käyttäjä, jolla on uusi varaajaID.
 - Tätä ei mielellään korjaisi.
 - Voisi kirjoittaa tiedostoon, kun luo käyttäjän, ja uudella ajokerralla tarkistaa, onko tiedostossa käyttäjä.
 - Voisi kovakoodata käyttäjälle oman varaajaID:n ja sitä kautta tarkistaa, onko käyttäjä jo luoto. Ja toivoo ettei tarkistaja tätä huomaa.
 - Kumpikaan ylläolevista ratkaisutavoista ei tunnu mielekkäältä.
- Koodin kommentoini
 - puuttuu vielä
 - Koodin luettavuudessa on kaksi tasoa. Ensimmäinen on, että koodin kirjoittanut ymmärtää sitä ja toinen että joku muu ymmärtää, mitä tapahtuu.
 - Nyt ollaan tasolla 1 suurimmalta osin koodia. EditBooking.java ja MakeBooking.javan int-char-string -veivailut ei ihan mene jakeluun, mutta ne metodit pitää kommentoida blackbox-tyyppisesti kiinnittäen huomiota siihen, mitä syöttää ja mitä paluttaa, eikä niinkään välivaiheeseen.
 - Koodin siisittäminen.
 - Debuggauksen jälkeisiä System.out.println();-komentoja on tosi paljon.
 - DataBaseHelper.javan metodissa on Upgrade on tällä hetkellä ehto if (1 > 2)
Johtuu harkkityön suurimmasta päähakkauksesta, kun ei tajuttu että database.execSQL("Create table.... Inset into...") pysähtyy heti ensimmäiseen ;-merkkiin eikä välitä sen jälkeisiä komennoista.
 - Raporttia pitää täydentää.

4. Mitä opimme?

- Älä ikinä korvaa date- ja time-datatyyppejä Stringeillä.
- aika paljon muuta ei vielä kirjoitettua

5 Terveiset tarkastajalle

Pistetavoite: loppusuora 13 pistettä kohti on alkanut.

Kun palautus tulee bumerangina takaisin, olisi kiva tietää, mitkä asiat pitää korjata ja mitä lisätä, että tavoitteeseen päästään.

Raportin lopun hieman epäasiallinen tyyli johtuu viimeisen 15 tunnin Android Studio -maratonista.