

# Dokumentaatio

## Ominaisuudet ja ohjelman käyttö

Ohjelma tukee asiakkaiden luomista ja työkohteiden lisäämistä asiakkaalle (T1). Ohjelma osaa muodostaa yhteyden tietokantaan, joka on toteutettu metodilla ConnectDB. Ohjelma tukee työsuoritusten ja tarvikkeiden lisäämistä työkohteisiin (T2). Jokaiselle päivälle on olemassa yksi työsuoritus, jonne sinä päivänä käytetyt tarvikkeet lisätään. Näin voidaan myös lisätä erilaisia työsuorituksia samalle kohteelle. Samana päivänä voi lisätä erilaisia työsuorituksia. (R1). Raportti 2 listaa raportissa 1 muodostettuun hinta-arvioon laskun (R2).

Toteutimme yksinkertaisen komentorivipohjaisen käyttöliittymän, jota käyttää ohjelman käyttäjä eli Seppo Tärsky.

Päävalikossa on seuraavat komennot:

- 1. Luo uusi asiakas. Luo uuden asiakkaan, jolle annetaan puhnumero, nimi ja osoite
- 2. Hae asiakastiedot Haetaan asiakastiedot nimellä tai asiakasId:llä. Jos kirjoittaa list, niin ohjelma tulostaa listan asiakkaista. Kirjoittamalla poistu poistutaan päävalikkoon.
- 3. Poistu ohjelmasta Poistutaan ohjelmasta
- 4. Hallitse tarvikkeita Ohjelman osio, jossa luodaan uusia tarvikkeita tietokantaan

Hallitse tarvikkeita on seuraavat komennot

- 1. Luo uusi tarvike Tässä luodaan uusi tarvike ja sille asetetaan nimi, sisäänostohinta, yksikkötyyppi ja alv luokka.
- 2. Näytä tarvikelista Näyttää luotujen tarvikkeiden listan ja tulostaa sen
- 3. Palaa päävalikkoon Menee ohjelman käynnistysvalikkoon

Kun asiakas ollaan valittu, niin tulee seuraavanlainen asiakkaan käsittely valikko

- 1. Lisää uusi projekti Lisätään uusi työkohde ja asetetaan työkohteen osoite
- 2. Tyhjennä asiakasvalinta Tyhjentää asiakasvalinnan ja palaa päävalikkoon
- 3. Poista asiakas Poistaa asiakkaan ja hänen työkohteensa

Seuraava osio tulee näkyviin, jos asiakkaalle on luotu työkohteita

- 5. Valitse projekti Antaa lyhyen listan asiakkaan projekteista. ID:tä painamalla päästään projektiin käsiksi. Tästä päästään Projektin käsittelyvalikkoon
- 6. Listaa projektit Listaa asiakkaan kaikki projektit ja tulostaa toString- tulosteen

Kun projekti ollaan valittu, niin päästään projektimenuun

- 1. Lisää työsuoritus Lisätään käsitellyssä olevalle projektille työsuoritus. Ensin valitaan onko kyseessä 1 – tavallinen- vai 2 – urakkatyö. Urakakatyössä syötetään vain työnosuuden kokonaissumma. Tavallisen työn lisäämisessä valitaan 1- suunnittelu, 2- normaalityö tai 3- aputyö. Sen jälkeen asetetaan työn tuntimäärä, jolloin ohjelma laskee työn hinnan.

- 2. Avaa työsuoritus Ominaisuus, jossa avataan jokin työsuoritus ja sinne voidaan lisätä tarvikkeita. Siitä päästään työsuorituksen käsittelyvalikkoon.
- 3. Tyhjennä projektivalinta Tyhjentää projektivalinnan, jolloin palataan asiakkaan käsittelyvalikkoon

Työsuorituksen käsittelyvalikossa on seuraavat ominaisuudet

- 1. Lisää käytetty tarvike Lisätään käytetty tarvike nimen perusteella. Jos tarviketta ei ole luotu aiemmin, niin se luodaan tässä samalla tavalla kuin ”1. Luo uusi tarvike” operaatiossa. Sen jälkeen annetaan se määrä mitä työsuoritukseen kyseistä tarviketta halutaan lisätä.
- 2. Hae kaupan tarvikkeet Tulostaa listan kaikista jo luoduista tarvikkeista.
- 3. Näytä työsuorituksen tarvikelista // Näyttää tarvikkeiden listan ja määrän, jotka ovat laitettu kyseiseen työsuoritukseen.
- 4. Takaisin Palataan takaisin edelliseen valikkoon
- 5. poista tarvike työsuorituksesta Poistaa tarvikkeen työsuorituksesta. Näyttää paljonko tarviketta on työsuorituksesta. Sitten valitaan tarvikkeen id ja määrä montako tarvikeyksilöä vähennetään. Kyseinen toiminto ei toimi täydellisesti, mutta toimii kuitenkin.

## Työnjako

Teimme projektia kaiken aikaa yhdessä. Suunnitteluosuudesta teimme ensin karkea versio, jota sitten lähdimme korjaamaan ennen ohjelmoinnin aloittamista. Teemun vastuulla on tietokanta-asiat ja Petteri otti suuremman vastuun itse ohjelmoinnista. Teemu teki paljon testausta ja muutaman ominaisuudenkin, jotka sovitimme lopulta yhteiseksi ohjelmaksi. Karkeasti Teemu suunnitteli enemmän ja Petteri teki enemmän.

Työskentelytapa oli tehdä samanaikaisesti etäyhteyden kautta niin, että näimme toistemme näytöt. Yleensä toinen ohjelmoi ja toinen ideoi ja testasi uutta koodia.

## Oma arvio

Työn aloittaminen oli selkeästi vaikein osa-alue, mutta kun saimme suunnitelman tehtyä, niin työn tekeminenkin alkoi onnistua. Jouduimme korjaamaan paljon tietokantaa ohjelmoinninkin aikana, josta tuli haasteta ja yllätyksiä, jotka veivät aikaa huomattavasti enemmän kuin olimme olettaneet.

Puutteita työssämme oli, että koodimme pääluokka oli pituudeltaan noin 1300- riviä. Siitä olisimme voineet kyllä nipistää huomattavasti, mutta operaatiomme ovat kuitenkin suhteellisen selkeitä. Sen lisäksi teimme vain ominaisuudet T1 ja T2, sillä olimme jo valmiiksi yliajalla ja laskutuksen tekeminen olisi ollut vielä työläämpää.

Muutimme paljon asioita alkuperäisestä dokumentistamme. Tietokanta meni pitkälti uusiksi, sillä olimme tehneet sen virheellisesti. Samoin ER-kaavio vaati korjausta paljon.

## Lopulliset taulujen luontilauseet

```
CREATE TABLE asiakas(  
    asiakasID SERIAL,  
    osoite VARCHAR(255),  
    puhelinnumero VARCHAR(255),  
    nimi VARCHAR(255),  
    PRIMARY KEY(asiakasID)  
);
```

```
CREATE TABLE tyoprojekti (  
    projektiID SERIAL,  
    asiakasID INT,  
    tila BOOLEAN,  
    projektihinta DECIMAL,  
    kohteenOsoite VARCHAR(255),  
    PRIMARY KEY (projektiID),  
    FOREIGN KEY (asiakasID) REFERENCES asiakas ON DELETE CASCADE  
);
```

```
CREATE TABLE tyosuoritus (  
    tyosuoritusID SERIAL,  
    projektiID INT,  
    kokonaishinta DECIMAL,  
    tyosuoritustyyppi VARCHAR(255),  
    tuntimaara INT,  
    tuntihinta FLOAT(2),  
    kotitalousvahennys FLOAT(2),  
    alv FLOAT(2),  
    tyoTyyppi VARCHAR(255),  
    PRIMARY KEY(tyosuoritusID),  
    FOREIGN KEY(projektiID)REFERENCES tyoprojekti ON DELETE CASCADE  
);
```

```
CREATE TABLE tarvike(  
    tarvikeID SERIAL,  
    tarvikenimi VARCHAR(255),  
    sisaanostohinta FLOAT(2),  
    myyntihinta FLOAT(2),  
    yksikkotyyppi VARCHAR(255),  
    alv DECIMAL,  
    PRIMARY KEY (tarvikeID)  
);
```

```
CREATE TABLE tarvikeluettelo(  
    TyosuoritusID INT,  
    TarvikeID INT,  
    maara INT,  
    FOREIGN KEY (TyosuoritusID) REFERENCES tyosuoritus ON DELETE CASCADE,  
    FOREIGN KEY (tarvikeID) REFERENCES tarvike  
);
```

```
CREATE TABLE lasku(  
    laskuID SERIAL,  
    ProjektiID INT,  
    laskutuslisa FLOAT(2),  
    maksettuPvm DATE,  
    laskunPaivamaara DATE,
```

```

        laskutyyppi VARCHAR(255),
        laskujarjestys INT,
        viivastyskorko FLOAT(2),
        erapaiva DATE,
        PRIMARY KEY(laskuID),
        FOREIGN KEY (ProjektiID) REFERENCES tyoprojekti
    );

```

```

CREATE TABLE laskunTila(
    laskuID INT,
    asiakasID INT,
    tila BOOLEAN,
    FOREIGN KEY(laskuID) REFERENCES lasku,
    FOREIGN KEY(asiakasID) REFERENCES asiakas
);

```

## Tietojen lisäyslauseet

### Lisää asiakas

```
INSERT INTO asiakas(puhelinnumero,osoite,nimi) VALUES(?,?,?);
```

### Poista asiakas

```
DELETE FROM asiakas WHERE asiakasid = ' " + asiakas.getAsiakasid() + " '";
```

### Luo uusi tarvike

```
"INSERT INTO tarvike(tarvikenimi, sisanostohinta, myyntihinta, yksikkotyyppi, alv)VALUES(?,?,?,?);"
```

### Hae asiakas

#### Tulostaa asiakkaat

```
SELECT asiakasid, osoite, puhelinnumero, nimi FROM asiakas
```

#### Hakee asiakkaat

```
"SELECT * FROM asiakas WHERE asiakasid=?::int OR nimi=?::varchar";
```

### Lisää projekti

```
"INSERT INTO tyoprojekti(asiakasid,tila,projektihinta,kohteenosoite)
(?,?,?,?)";
```

### Hae projektit

```

("SELECT projektiid, " +
    "asiakasid, tila, projektihinta, kohteenosoite " +
    "FROM tyoprojekti " +
    "WHERE asiakasid = ' " + asiakas.getAsiakasid() +
    " '");

```

### Hae suoritukset

```

"SELECT * FROM tyosuoritus
WHERE projektiid = ' " + aktiivinenProjekti.getId() + " '";

```

### Lisätään tyosuoritus

```
"INSERT INTO tyosuoritus(projektiid,kokonaishinta,tyosuoritustyyppi, " +
```

```
"tuntimaara,tuntihinta,kotitalousvahennys,alv,tyotyyppi)VALUES(?,?,?,?,?,
?,?,?);"
```

### Tulosta tarvikeluettelo

```
("SELECT tarvikeid, tarvikenimi, sisanostohinta, myyntihinta,
yksikkotyyppi, alv FROM tarvike ");
```

### Hae tarvikkeet työsuorituksessa

```
"SELECT DISTINCT t.tarvikeID, tl.maara,t.yksikkotyyppi, t.tarvikenimi,
t.sisanostohinta, t.myyntihinta,t.alv
FROM tarvikeluettelo as tl, tarvike as t
WHERE tl.tarvikeID = t.tarvikeID AND tl.tyosuoritusID = " +
aktiivinenSuoritus.getId() + "');
```

### Hae tarvike

```
"SELECT DISTINCT *
FROM tarvike
WHERE tarvikenimi LIKE '%" + nimi.replace("\'", "'") + "%';";
```

### Poista tarvike suorituksesta

Jos poistetaan vain osa tarvikkeista

```
"UPDATE tarvikeluettelo
SET maara = " + (value - lkm)
WHERE tyosuoritusID = " + aktiivinenSuoritus.getId() AND tarvikeID = " +
poistettava;
```

Jos poistetaan kaikki tai liikaa

```
"DELETE FROM tarvikeluettelo
WHERE tyosuoritusID =" + aktiivinenSuoritus.getId() AND tarvikeID = " +
poistettava;
```

### Lisää tarvike suoritukseen

Jos lisätään sama tarvike sinne

```
"UPDATE tarvikeluettelo
SET maara = " + (maara + aktiivinenSuoritus.getTarvikkeet().get(tarvike))
WHERE tyosuoritusID = " + aktiivinenSuoritus.getId()
AND tarvikeID = " + tarvike.getId();
```

Jos lisätään uusi tarvike suoritukseen

```
"INSERT INTO tarvikeluettelo(tyosuoritusid, tarvikeid, maara) VALUES
(?,?,?);"
```

### Päivitetään suorituksen hinta

```
"UPDATE tyosuoritus
SET kokonaishinta = "+ uusiHinta +
WHERE tyosuoritusID = " + aktiivinenSuoritus.getId();
```

### Päivitetään projektin hinta

```
"UPDATE tyoprojekti
SET projektihinta = " + hintaDelta.add(new
BigDecimal(aktiivinenProjekti.getHinta())) +
" WHERE projektiid = " + aktiivinenProjekti.getId();
```

### Tulosta tyosuorituksen tarvikelista

```
"SELECT DISTINCT tl.tarvikeID, tl.maara, t.yksikkotyyppi, t.tarvikenimi,  
tl.tyosuoritusID  
FROM tarvikeluettelo as tl, tarvike as t, tyosuoritus as ty  
WHERE tl.tarvikeID = t.tarvikeID AND tl.tyosuoritusID = '"' +  
aktiivinenSuoritus.getId() + '"';
```