CT60A4301 Tietokannat

**Harjoitustyö 1 – Tietokannan suunnittelu**

Lappeenrannan teknillinen yliopisto

Innovation and Software (IS), LUT LBM

CT60A4302 Tietokannat

Kesä 2017

0453757 Kuisma Närhi

Kuisma.narhi@student.lut.fi

SISÄLLYSLUETTELO

[SISÄLLYSLUETTELO 1](#_Toc445902737)

[1 määritys 2](#_Toc445902738)

[2 käsitemalli 4](#_Toc445902739)

[3 tietokantatoteutus 6](#_Toc445902740)

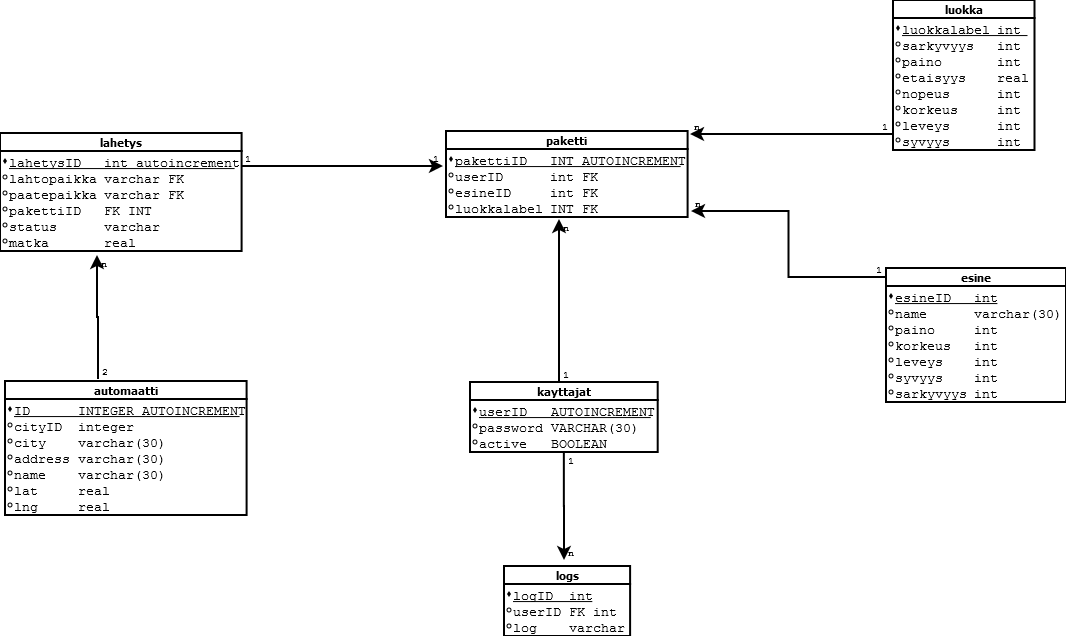
[4 keskustelu 7](#_Toc445902741)

# määritys

Tietokanta luotiin smartpost-javasovellusta varten. Tietokantaa käyttää oletetut postin käyttäjät java-sovelluksen kautta. Ohjelma vaati toimiakseen tietokantataulun kaikista smartpost-pisteistä, eri pakettiluokista, postitettavista esineistä ja käyttäjistä. Näitä tietoja yhdistelemällä luotiin taulut ohjelman toiminnan kannalta järkevään muotoon. Ohjelma myös tallensi logit tietokantaan toiminnastaan.

Käyttäjä voi tallentaa kantaan oman käyttäjänsä, uusia postipisteitä, esineitä ja oman toimintansa logit. Lähetyksistä tallentuu tieto lähetys- ja päätepisteistä, lähettäjästä, sisällöstä, pakettiluokasta ja lähetyksen onnistumisesta.

# käsitemalli



*kuva1. tietokannan rakenne ER-mallina*

Eheysrajoitteet on toteutettu varsinaisessa ohjelmassa NOT NULL-käskyillä pääavaimissa ja ohjelmapuolen rajoitteilla. Tietokantaan ei voi ottaa yhteyttä ilman vaadittavia muuttujia, joten yksikään sarake ei jää tyhjäksi, lukuun ottamatta mahdollisesti yhtä käyttäjää ja useaa salasanaa jos käyttäjät näin haluavat.

# tietokantatoteutus

**Yleistä:** Koodasin ohjelman jo kerran tietokantarakenteella, joka oli loppujenlopuksi liian yksinkertainen ja aiheutti spagettikoodia, joka vuorostaan aiheutti purkkaratkaisuja tietokantaan. Tehtävänannossa vaadittu seitsemäs taulu on vähän väkisin sisään tungettu, koska ohjelma suoltaa ulos myös log.txt-tiedoston. Sinänsä osoitteen olisi voinut pilkkoa atomisemmaksi normaalimuotoon (postinumero-kaupunkitaulu vielä erikseen), mutta koin tämän suoraviivaisemman automaatti-taulun selkeäksi ja ohjelman toimivuuden kannalta järkeväksi, sillä XML-taulun lukeminen suoraan tietokantaan ja siitä kaupunkikohtainen pistehaku käyttöliittymässä tuotti halutun lopputuloksen.

**Toteutustapa:** Java-koodipohja, johon loin tietokantaa varten kattavan database-rajapinnan. Alustin tietokannan schema- ja build-tiedostoilla, jotka sisälsivät tietokannan luonnin ja ohjelman kannalta oleelliset perusinsertit.

**Testatessa huomioitavaa:** ohjelman saa kyllä hajalle, mutta tietokantainjektioita ei pitäisi voida tehdä.

**Koodissa huomioitavaa:** Eheyssäännöt on toteutettu enimmäkseen javan puolella; tietokantarajapinta vaati muuttujiksi jokaisen oleellisen kentän joten ohjelma heitti erroria jos virheellisiä syötteitä annettiin. Logit on luotu join-operaattoreita käyttäen.

# keskustelu

Luokka- ja esinetaulut näyttävät äkkiseltään samankaltaisilta mutta niissä on periaatteellinen ero, sillä luokka määrittää rajoitteita ohjelmalle ja tuote ”mahdutetaan” näiden rajoitteiden sisään.  
  
Jos jotain tästä projektista oppi, on koodaamisen tehon radikaali laskeminen yli ~12h pitkien rupeamien aikana.