

CT60A4301 Tietokannat ja CT60A2410 Olio-ohjelmointi

Harjoitustyö

Lappeenrannan teknillinen yliopisto
Innovation and Software (IS), LUT LBM

CT60A4301 Tietokannat ja CT60A2410 Olio-ohjelmointi
Kesäkurssi 2016

0452334(op. nro.) Aleksi Ruokoniemi aleksi.ruokoniemi@student.lut.fi

SISÄLLYSLUETTELO

SISÄLLYSLUETTELO	1
1 TIETOKANNAN MÄÄRITYS.....	2
2 TIETOKANNAN KÄSITEMALLI.....	3
3 OHJELMAN KUVAUS	4
4 OHJELMALLINEN TOTEUTUS JA LISTA TOIMINNALLISUUKSISTA	5
5 OHJELMAN LUOKKAKAAVIO	8
6 YHTEENVETO.....	8

1 TIETOKANNAN MÄÄRITYS

Harjoitustyön tietokanta toteutetaan mahdollistamaan kuvitteellisen SmartPost-postiautomaattiverkoston käytön. Tietokannan täytyy säilöä tarvittavat tiedot sekä postiautomaateista että luoduista paketeista ja niissä olevista esineistä. Sen lisäksi tietokantaan kerätään myös lokitietoja lähetetyistä paketeista. Postiautomaateista säilötään XML-dokumentista vapaasti saatavat tiedot. Tietokantaa käytetään säilömään luotujen paketti- ja esineolioiden tietoja, jotta ohjelma voi hakea niitä tarvittaessa. Tämän vuoksi paketeista ja esineistä tulee varastoida kaikki tiedot paitsi ohjelmallisesti määritetyt arvot esim. pakettiluokkiin ja esineisiin liittyen.

Pakettien lähettämisen mahdollistamiseksi tietokantaan täytyy voida tehdä päivityksiä käyttöliittymän avulla ilman, että käyttäjän täytyy hallita tietokantaa. Lähetetyt paketit täytyy pystyä asettamaan lähetetyiksi. Lähetyksen aikana mahdollisesti rikkoutuneet esineet täytyy myös pystyä asettamaan rikkoutuneiksi.

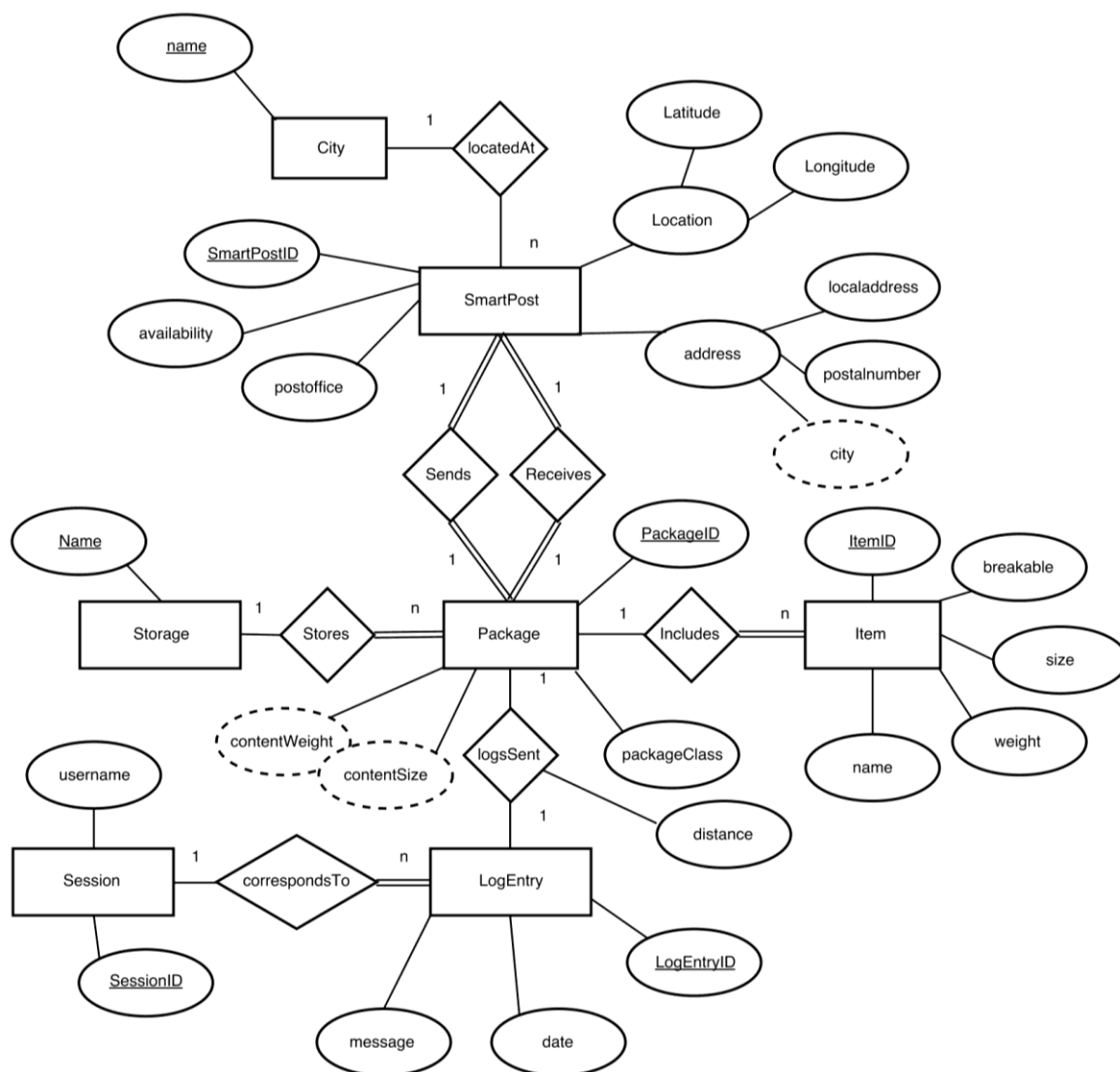
Joitain tietokannassa olevia tietoja tulee myös voida hallita käyttöliittymän avulla. Pakettiautomaattien osalta tietokantaan täytyy voida lisätä uusia automaatteja. Jo olemassa olevien automaattien tietojen muuttamisen tai valitun automaatin poistamisen tietokannasta täytyy olla mahdollista. Tietokantaan tulee pystyä lisäämään uusia varastoja ja niihin uusia paketteja, jotka sisältävät esineitä. Luotuja varastoja, paketteja ja esineitä täytyy voida poistaa tietokannasta. Pakettien tietoja ja sisällä olevia esineitä tulee voida tarkastella katsoa ennen paketin lähettämistä sekä sen jälkeen, jolloin voidaan tarkistaa mitä sisällä oleville esineille on tapahtunut lähetyksen aikana (ovatko ne särkyneet).

Tietokannan käyttäjiä ei ryhmitellä millään tasolla. Ohjelman käytön alussa käyttäjä voi kuitenkin antaa vapaaehtoisesti käyttäjätunnuksen, joka liitetään istuntoon ja esitetään lokitietojen yhteydessä.

Käytetyt tietokantakyselyt löytyvät ohjelman DatabaseHandler -luokasta. JOIN-määritettä on käytetty getSmartPost -metodissa, rivillä 1040.

2 TIETOKANNAN KÄSITEMALLI

Linkki kuvaan: <https://www.dropbox.com/s/yxrkr19f5bxul9h/kasitemalli-1.png?dl=0>



3 OHJELMAN KUVAUS

Luotu ohjelma antaa käyttöliittymän ennalta määriteltujen esineiden paketoimisen ja pakettien lähettämisen kartalla näkyvien postiautomaattien välillä. Lisäksi ohjelma mahdollistaa tietokannassa olevien SmartPost -, varasto- ja pakettitietojen muokkaamisen ja lokitietojen tarkastelemisen.

Ohjelma koostuu useassa näkymästä. Alkunäkymässä käyttäjä voi valita, millä asetuksilla tietokannan käyttö aloitetaan. Käyttäjä voi valita tyhjennetäänkö loki- tai varastotiedot ennen käyttöä. Käyttäjällä on myös mahdollisuus ladata SmartPost -tiedot tietokantaan XML-tiedostosta, jolloin tietokannan kaikki taulut tyhjennetään ennen uusien tietojen lataamista tiedon päällekkäisyyden vähentämiseksi. Käyttäjä voi myös syöttää käyttäjätunnuksen tekstikenttään, jolloin syötetty käyttäjätunnus liitetään istunnon aikana lisättyihin lokitietoihin. Jos käyttäjä ei syötä käyttäjätunnusta, asetetaan käyttäjätunnuksen arvoksi 'Guest'. Ensimmäisessä näkymässä asetetaan myös SQLite-tietokannan sijainti. Tekstikenttä näyttää virheilmoituksen, jos annettu sijainti on virheellinen tai osoittaa vääräntyyppiseen tiedostoon. Seuraavaan näkymään voidaan siirtyä, kun yhteys tietokantaan toimii.

Seuraava näkymä koostuu kahdesta välilehdestä. Ensimmäiseltä välilehdeksi löytyvät karttaan liittyvät toiminnallisuudet, pakettien luonti ja lähettäminen sekä tietokannan hallinta. Ennen pakettien luontia täytyy valita varasto, johon luotu paketti asetetaan. Ohjelma ohjeistaa joko valitsemaan varaston tai lisäämään uuden sellaisen tietokantaan, mikäli varastoja ei ole luotuna. Tietokannan hallinta avautuu uuteen ikkunaan, josta löytyvät välilehdet varastojen ja SmartPost -automaattien hallinnalle. Varastojen välilehdeksi voidaan tarkastella varastoitujen pakettien sisältöä, lisätä uusia varastoja, poistaa varastoja tai varastoissa olevia paketteja. Varastojen poistaminen poistaa myös kaikki niissä olleet paketit. Pakettien poistaminen poistaa lisäksi niihin paketoituneet esineet. SmartPost -välilehdellä voidaan lisätä uusia automaatteja, muokata olemassa olevien automaattien tietoja tai poistaa niitä. Automaatit ryhmitellään kaupungin mukaan. Valittavissa on vain sellaisia kaupunkeja, joista löytyy vähintään yksi automaatti. Kaupunki

poistetaan listalta, jos SmartPost -automaattiin kohdistuvan päivityksen jälkeen kaupungista ei enää löydy automaatteja.

Kun varasto on valittu, voidaan sinne luoda paketteja. Pakettejen luonti tapahtuu uudessa ikkunassa. Valitun pakettiluokan rajat päivittyvät pakettiluokkaa vaihdettaessa. Esineitä voi lisätä pakettiin, jolloin paketin koko ja paino kasvavat. Paketin koko ja paino näkyvät pakettiluokan rajojen vieressä ja värittyvät punaiseksi, jos ne ylittävät sallitut rajat. Pakettia ei voi luoda, jos rajat ovat ylittyneet.

Pakettien lähettäminen onnistuu valitsemalla paketti sen ID:n mukaan ja valitsemalla kaksi SmartPost -automaattia joiden välillä paketti lähetetään. Valittavissa ovat vain ne automaattit, jotka on merkitty kartalle. Automaattien välinen matka tarkistetaan ennen lähetystä. Jos matka on liian suuri valitulle pakettiluokalle, näytetään virheilmoitus. Lähetetyn paketin sisällön voi tarkastaa Loki-välilehdeltä. Välilehdellä on mahdollisuus tarkastella lokitietoja nykyiseltä istunnolta tai kaikilta ajoilta.

4 OHJELMALLINEN TOTEUTUS JA LISTA TOIMINNALLISUUKSISTA

Ohjelma hyödyntää tietokantaa SmartPost, paketti ja esine -olioiden tallettamiseen. Näin ohjelmassa ei tarvitse pitää muistissa esim. jokaista SmartPost-automaattia, vaan niiden tiedot haetaan tarvittaessa tietokannasta ja olio luodaan.

Harjoitustyössä täyttyvät olio-ohjelmoinnin osalta laajimman vaatimustason vaatimukset, tietokantojen osalta perusvaatimukset.

Lista **olio-ohjelmoinnin** harjoitustyöhön liittyvistä toiminnallisuuksista:

Perustaso:

- Valitun kaupungin kaikki SmartPost -automaattit saa lisättyä kartalle painiketta painamalla.
- Uusia paketteja voi luoda valittuun varastoon. Paketteihin voi lisätä esineitä valitun pakettiluokan rajoissa.

- Luotuja paketteja voi lähettää kartalla olevien SmartPost -automaattien välillä. Matkan pituus tarkastetaan ennen lähettämistä. Paketin lähettämistä ei sallita, jos matka on liian pitkä valitulle pakettiluokalle.
- Painiketta painamalla kartalle piirretyt reitit saa tyhjennettyä.
- SmartPostHolder -luokka pitää kirjaa kartalla olevista SmartPost -olioista. Oliot haetaan kaupungeittain tietokannasta.
- XMLParser -luokka syöttää XML-tiedostosta saadut tiedot tietokantaan mikäli käyttäjä valitsee asetuksen ohjelman alkuvalikossa.
- Storage -luokkaan varastoidaan luotuja paketteja. Varastoon asetetut paketit haetaan tietokannan kautta.
- Package.java tiedostosta löytyy abstrakti pakettiluokka ja kolme siitä periytettyä pakettiluokkaa, joilla on erilaisia rajoja paketoitavien esineiden ja lähetysmatkan suhteen.
- Item.java tiedostosta löytyy abstrakti esineluokka ja neljä siitä periytettyä esineluokkaa, joilla on erilaisia ominaisuuksia.

Laajennetut vaatimukset:

- Lähetetyistä paketeista kirjataan lokitiedot tietokantaan. Lokitiedot esitetään Loki-välilehdellä. Näkymästä on mahdollista tarkastella lähetyspäivämääriä, paketin lähettäjää, lähtö- ja saapumipaikkoja sekä matkojen pituuksia. Pakettien tietoja ja sisällä olevia esineitä voi tarkastella painiketta painamalla.
- Ohjelmassa käytetään useita uusia ikkunoita pakettien lisäämiseen, tietokannan hallintaan jne. Lokitietojen esittämiseen käytetään välilehteä.

Laajin vaatimustaso:

- Käyttöliittymää on muokattu visuaalisesti itse luodulla CSS-tiedostolla.
- Ohjelma käyttää tietokantaa mahdollistaakseen edellisen käyttökerran varasto/lokitilanteen käytön. Lokitietoja on mahdollista selata tämänhetkiselä istunnolta tai kaikilta ajoilta.

Lista **tietokantojen** harjoitustyöhön liittyvistä toiminnallisuuksista:

Suunnitteluosuus:

- Käsitemalli koostuu 7 yksilötyypistä. Valmis tietokanta koostuu 9 taulusta.
- SELECT-käskyjä on yhteensä 22. JOIN-lausetta käytetty yhdessä. Käytetyt tietokantakyselyt löytyvät ohjelman DatabaseHandler -luokasta. JOIN-määritettä on käytetty getSmartPost -metodissa, rivillä 1040.

Ohjelmallinen toteutus:

- Tietokanta on käytettävissä SQLite3:n ja Java-käyttöliittymän kautta. Java-käyttöliittymän kautta toteutettavat toiminnot rajoittuvat luonnollisesti tehtävänannon tarpeiden mukaan.
- Tietokannastaan voi lisätä uusien SmartPost -automaattien tietoja. Olemassa olevia tietoja voi päivittää tai poistaa. Varastoja on mahdollista lisätä tai poistaa. Poistamisen yhteydessä tietokannasta poistetaan myös varastoidut (ei lähetetyt) paketit ja niiden sisältävät esineet. Uusia paketteja voi luoda varastoihin ja niihin voi asettaa esineitä. Varastossa olevia paketteja voi poistaa. Myös lähettämättömien pakettien sisältämiä esineitä voi poistaa. Pakettien ja sisällä olevien esineiden tietoja on mahdollista tarkastella.
- XMLParser -luokka lukee XML-muotoista tietoa ja lisää tietoa DatabaseHandler -luokan kautta tietokantaan.

[illegible]

Toteutustapa: Harjoitustyön tehtävänannon mukainen.

Ohjelmaa käytetty versiolla 3.8.6., mutta pitäisi toimia myös uudemmilla versioilla. Ajurin toiminnassa saattaa olla satunnaisesti ongelmia. Esim. demovideota kuvatessa Linux-luokan tietokoneilla ajuri ei suostunut toimimaan pidempiä aikoja vaan antoi `ClassNotFoundException:n`. NetBeansin uudelleenkäynnistäminen korjaa ongelman.

Ohjelma hakee ensisijaisesti tietokannan sijainnin 'settings.properties' tiedostosta. Testaamisen helpottamiseksi tietokannan tiedostopolun voi asettaa k.o. tiedostoon, jolloin tietokannan sijaintia ei tarvitse asettaa uudestaan jokaisella käynnistyskerralla.

Koodissa huomioitavaa: Tietokannan tauluihin liittyvien eheyssääntöjen tarkistukset toteutettu ohjelmallisesti.