

CT60A2410 Olio-ohjelmointi

Harjoitustyö 1 – TIMOTEI-ohjelma

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

CT60A4301 Tietokannat
Kevät 2016

0453731 Joonas Virmajoki
joonas.virmajoki@student.lut.fi

SISÄLLYSLUETTELO

SISÄLLYSLUETTELO	1
1 TEHTÄVÄNANNON KUVAUS JA TYÖN RAJAUKSET	2
2 TIETOKANNAN KÄSITEMALLI JA EHEYSSÄÄNNÖT	3
3 OHJELMAN KUVAUS	5
4 RAPORTTI OHJELMALLISESTA TOTEUTUKSESTA JA LISTA TOIMINNALLISUUKSISTA	6
5 OHJELMAN TÄYDELLINEN LUOKKAKAAVIO	8
6 YHTEENVETO	9

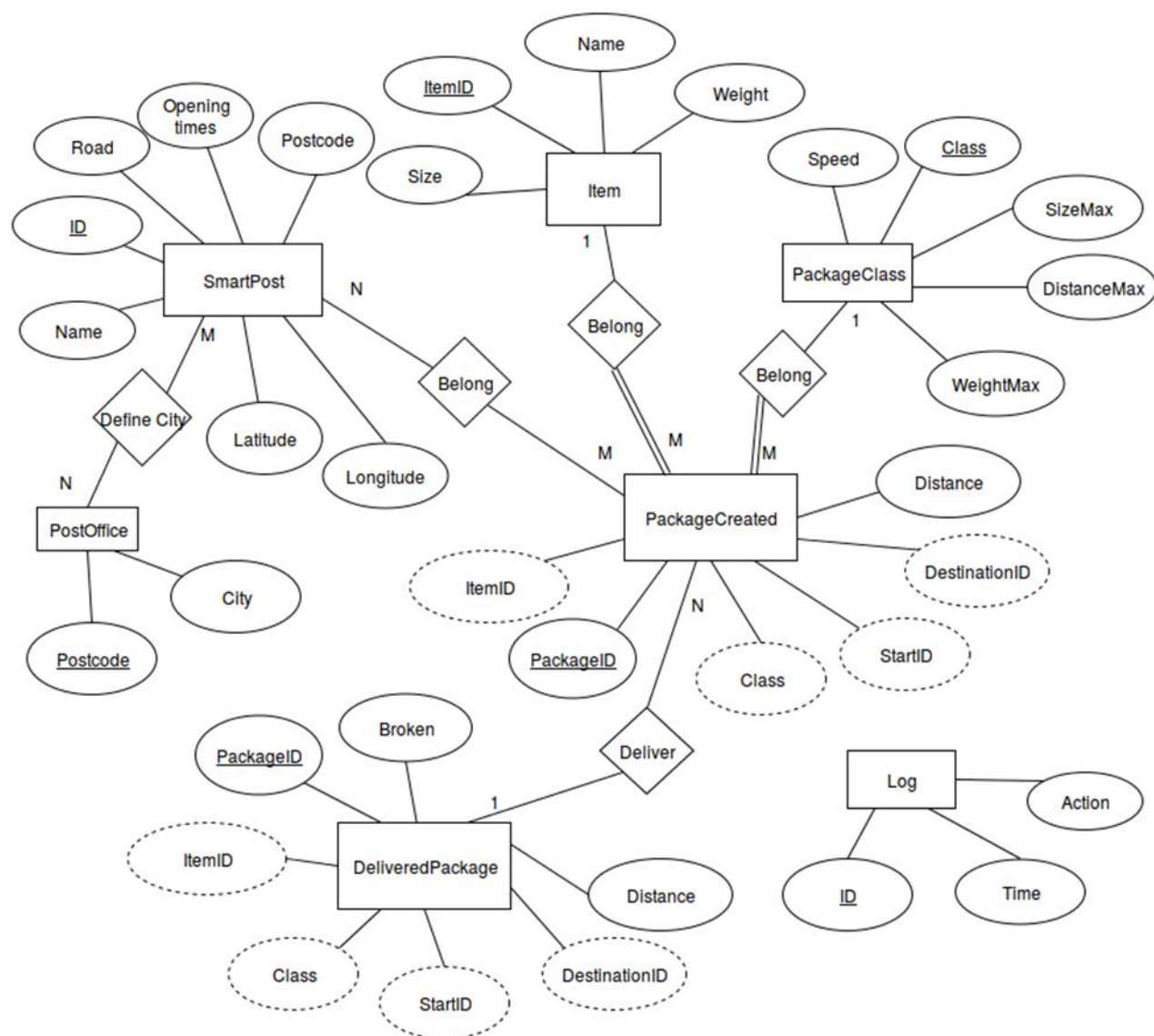
1 TEHTÄVÄNANNON KUVAUS JA TYÖN RAJAUKSET

Harjoitustyössä suunnitellaan ja toteutetaan tietokanta kuvitteelliselle loppukäyttäjälle tai asiakkaalle, sekä tehdään SmartPost-verkosta hyödyntävä TIMOTEI (Toiminnaltaan Itellaa Muistuttava Ohjelmisto, Tietokantaa Edellyttävä Integraatio) ohjelmallinen käyttöliittymä. Tietokanta toteutetaan SQLite3 työkalulla ja käyttöliittymä käyttää Javaa ja JavaFX-työkalua. Ohjelman tulee hakea avoimena datana olevat SmartPostit verkosta, jotka sisältävät dataa kuten nimen, sijainnin ja aukiolotiedot. Ohjelman pitää lisätä SmartPostien data tietokantaan. Lisäksi tietokantaan tarvitaan oma osio paketeille ja paketteja sisältäville tavaroille. Tietokannassa täytyy olla valmiiksi neljä oletustavaraa. Paketeilla on kolme eri mahdollista pakettiluokkaa, joissa on rajoituksia paketin koolle, painolle ja matkalle. Pakettiluokka vaikuttaa paketin nopeuteen ja tavarän ehjänä säilymiseen kuljetuksessa. Pakettiluokka toteutetaan tietokantaa käyttäen. Kuvitteellisen loppukäyttäjän pitää olla mahdollista poistaa ja lisätä tietokannasta SmartPosteja, paketteja ja tavaroita käyttäen TIMOTEI-ohjelmaa. Ohjelman tulee käyttää opettajan tekemää paikallista Google Maps html-tiedostoa ja sen JavaScript-rajapintaa, jolla on mahdollista piirtää merkki kartalle, luoda reitti ja poistaa reitit kartalle. Ohjelman pitää esittää visuaalisesti SmartPosteja ja pakettien lähettämistä SmartPostista toiseen. Ohjelmaan ei tehdä käyttäjänhallintaaja ja käyttäjällä on kaikki oikeudet muokata tietokantaa.

2 TIETOKANNAN KÄSITEMALLI JA EHEYSSÄÄNNÖT

Työn toteutus aloitettiin määrittelemällä tietokannan laajuus ja luomalla siitä käsittemalli. Sen jälkeen käsittemalli konvertoitiin relaatiotietokannaksi.

Kuvassa ER-malli, jossa kohdetyyppeinä on tavara, SmartPost, paketin luokka, postitoimipaikka, luotu paketti, toimitettu paketti ja loki.



SmartPost	
PK	"ID" INTEGER ROWID NOT NULL
FK	"Postcode" VARCHAR(8) NOT NULL
"Name" VARCHAR(32) NOT NULL	
"Latitude" VARCHAR(12) NOT NULL	
"Longitude" VARCHAR(12) NOT NULL	
* "Road" VARCHAR(32) NOT NULL	
"OpeningTime" VARCHAR(64) NOT NULL	

DeliveredPackage	
PK	"PackageID" INTEGER ROWID NOT NULL
FK	"ItemID" INTEGER NOT NULL
FK	"StartID" INTEGER NOT NULL
FK	"DestinationID" INTEGER NOT NULL
"Class" INTEGER NOT NULL	
"Distance" REAL NOT NULL	
"Broken" INTEGER NOT NULL CHECK("Broken" IN (0, 1))	

Item	
PK	"ItemID" INTEGER ROWID NOT NULL
"Name" VARCHAR(32) NOT NULL	
"Weight" INTEGER NOT NULL CHECK("Weight" > 0)	
"ItemSize" INTEGER NOT NULL CHECK("ItemSize" > 0)	
"Breakable" INTEGER NOT NULL CHECK("Breakable" IN (0, 1))	

PostOffice	
PK	"Postcode" VARCHAR(8) NOT NULL
"City" VARCHAR(32) NOT NULL	

DeliveredPackage	
PK	"PackageID" INTEGER ROWID NOT NULL
FK	"ItemID" INTEGER NOT NULL CHECK("ItemID" > 0)
FK	"StartID" INTEGER NOT NULL CHECK("StartID" > 0)
FK	"DestinationID" INTEGER NOT NULL CHECK("DestinationID" > 0)
"Class" INTEGER NOT NULL CHECK("Class" IN (1, 2, 3))	
"Distance" REAL NOT NULL CHECK("Distance" > 0)	

PackageClass	
PK	"Class" INTEGER ROWID NOT NULL
"Speed" INTEGER NOT NULL CHECK("Speed" > 0)	
"WeightMax" REAL NOT NULL CHECK("WeightMax" > 0)	
"SizeMax" REAL NOT NULL CHECK("SizeMax" > 0)	
"DistanceMax" REAL NOT NULL CHECK("DistanceMax" > 0)	

Log	
PK	"LogID" INTEGER ROWID NOT NULL
"Time" VARCHAR(32) NOT NULL	
"Action" VARCHAR(64) NOT NULL	

3 OHJELMAN KUVAUS

Harjoitustyössä tehtävä TIMOTEI-ohjelma käyttää valmista avointa SmartPost-verkostoa. Ohjelma hakee SmartPostien tiedot xml:nä ja parseaa tiedot niistä tietokantaan. Tietokannassa on omat osiot paketeille, tavaroille ja pakettiluokille. Paketeilla on kolme eri mahdollista luokkaa, jotka vaikuttavat paketin nopeuteen, matkan pituuteen ja paketissa olevan tavaran ehjänä säilymiseen. Käyttäjän on mahdollista luoda uusia paketteja jotka sisältävät tavaroita, sekä lähettää paketti. Lähettäessä paketin kartalle ilmestyy visuaalinen esitys paketin kulusta. Käyttäjän on myös mahdollista luoda omia tavaroita neljän oletustavaran lisäksi. Käyttäjä voi ohjelman käyttöliittymän avulla poistaa tai lisätä SmartPosteja/paketteja/tavaroita tietokantaan. Käyttäjällä on mahdollisuus lisätä kaupungin SmartPostit kartalle ja poistaa kartalta lisätyt reitit sekä merkit.

4 RAPORTTI OHJELMALLISESTA TOTEUTUKSESTA JA LISTA TOIMINNALLISUUKSISTA

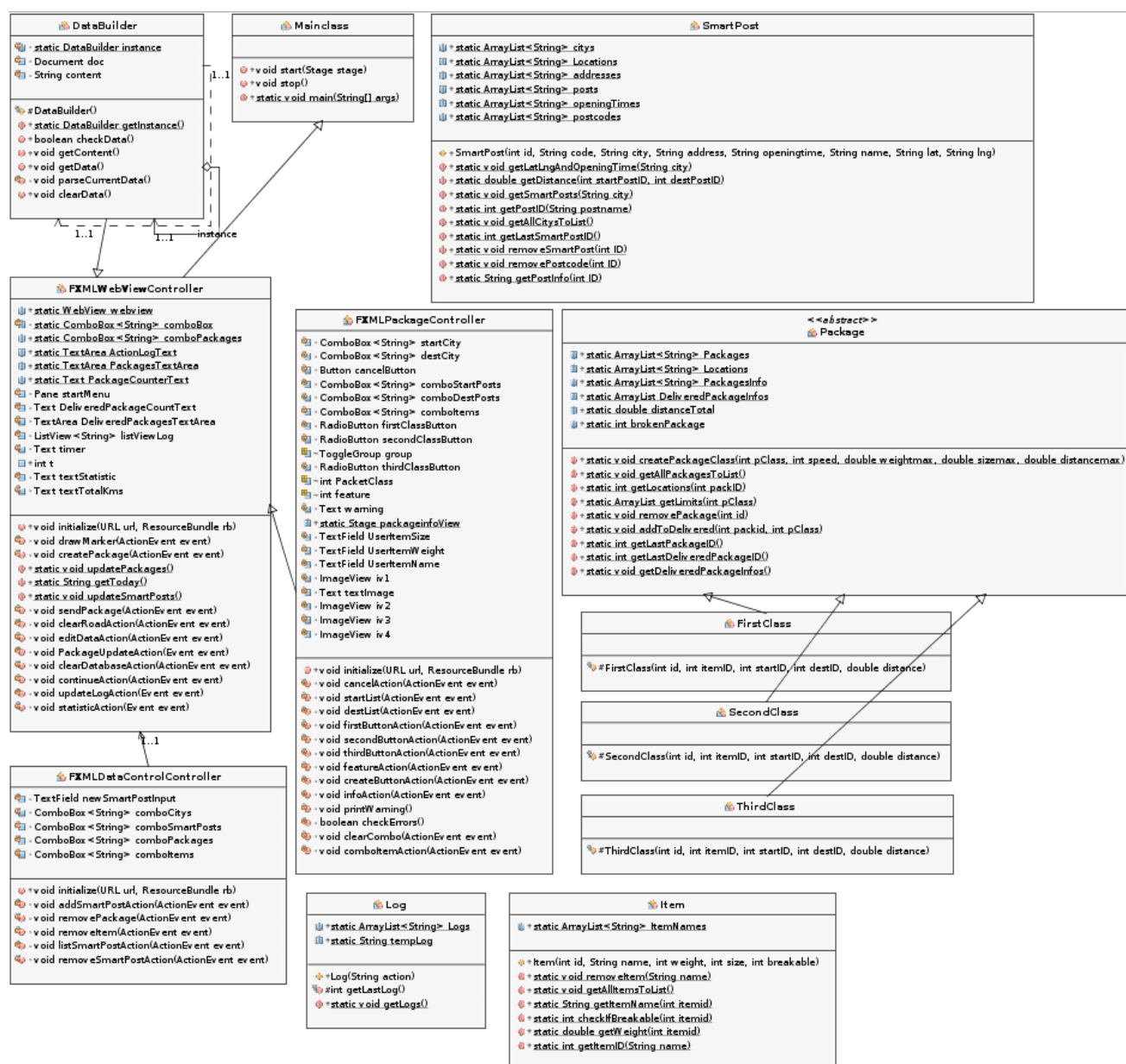
Ohjelmassa on pääluokka, joka avaa graafisen käyttöliittymän FXMLWebView:n. Sen kontrolleri alustaa ajastimen käyntiin, avaa paikallisen html-tiedoston ja tarkistaa onko tietokanta tyhjä. Jos tietokanta on tyhjä, se hakee tiedot automaattisesti, muuten kysytään käyttäjältä halutaanko tietokanta tyhjentää. SmartPostien tiedot haetaan DataBuilder luokalla. Se pystyy hakemaan datan, tarkistamaan onko tietokanta tyhjä, parseamaan datan tietokantaan tai tyhjentämään koko tietokannan. WebView:ssä on metodi piirtää valitun kaupungin kaikki SmartPost-automaatit kartalle JavaScript-rajapintaa käyttäen. Tein opettajan antamaan html-tiedostoon seuraavat muokkaukset. Ohjelma piirtää kartalle merkit SmartPostien paikkatietoja käyttäen. Osoitteita käyttäessä havaitsin Googlen rajoittavan pyyntöjä 9-10 kappaleeseen sekunnissa, jolloin osa SmartPosteista ei ilmestynyt kartalle. Lisäsin html-tiedostoon funktion joka mittaa kahden SmartPostin etäisyyden ja palauttaa sen, sekä kehitin funktion joka poistaa kartalta myös lisätyt merkit. WebView ikkunasta on mahdollista aukaista uusi ikkuna joko paketin luontiin tai datan hallintaan. SmartPostille, paketille, tavaralle ja lokille on omat luokkansa ja rakentajat. Luokat sisältävät siihen liittyviä toimintoja esimerkiksi paketti-luokassa on tietokantakysely paketin poistoon. Jokaiselle tarvittavalle tietokantakyselylle on tehty oma metodi, mutta yhdessä metodissa voi olla käytetty montaa eri tietokantakyselyä. Ohjelmassa on tietokannasta luettaessa käytetty apuna listoja tiedon välityksessä.

Lista ohjelman toiminnallisuuksista:

- Mahdollisuus lisätä SmartPosteja kartalle kaupungin perusteella
- Mahdollista luoda uusi esine, paketoita se ja lisätä varastoon
- Mahdollista lähettää paketti SmartPost-automaattien välillä
- SmartPostia klikkaamalla saa esiin valitun SmartPost-automaatin nimen ja aukiolotiedot
- Lähettäessä paketin, ohjelma piirtää merkin lähtö- ja saapumispaikkaan ja piirtää visuaalisen reitin merkkien välille
- Tyhjentää kartalta piirretyt reitit ja merkit
- Ohjelma pitää kirjaa luotujen ja lähetettyjen pakettien määrästä, lähtö- ja saapumispaikoista, matkan pituuksista ja paketeissa olevista esineistä
- Pitää lokia toiminnoista tietokannassa

- Lopetettaessa ohjelma kirjoittaa lokin tekstitiedostoon istunnon aikaisesta kirjanpidosta ja varaston tilanteesta.
- Ohjelmassa on Statistiikka-välilehti joka sisältää tietoja miten pitkään ohjelma on ollut käynnissä, suhteen montako pakettia on mennyt ehjänä perille ja kuinka monta kilometriä paketit ovat kulkeneet yhteensä
- Ohjelma ei anna aukaista useita ikkunoita ja suljettaessa se kulkee kaikki ikkunat
- Ohjelman käyttöliittymän komponentteja on muokattu visuaalisesti
- Ohjelmassa mahdollisuus aloittaa tyhjältä pöydältä tai jatkaa vanhalla varastotilanteella ja lokitiedoilla
- Neljällä oletustavaralla on kuvat

5 OHJELMAN TÄYDELLINEN LUOKKAKAAVIO



6 YHTEENVETO

Harjoitustyössä havaittiin että hyvä tietokannan suunnittelu alussa olisi helpottanut ja nopeuttanut ohjelman koodausta, koska koodia ei tarvitsisi kokoajan muokkailla uusiksi monesta eri paikasta. Ongelmia tuli opettajan html-tiedoston funktioiden kanssa, mutta muokkaamalla niitä paremmaksi ja etsimällä netistä apua ongelmiin, saatiin ongelmat aina ratkaistua. Käsitemallin toteutuksessa relaatiomalliin havaittiin postinumeron ja kaupungin välinen riippuvuus. SmartPostien postitoimipaikoille luotiin oma relaatio toiston vähentämiseksi ja sen attribuutit ovat postinumero ja kaupunki.