

Jarno Rantanen

YRITYSREKISTERIN HALLINTAOHJELMISTON  
KÄYTETTÄVYYDEN PARANTAMINEN

Tekniikka ja merenkulku Pori  
Tietotekniikan koulutusohjelma  
Ohjelmistotekniikan suuntautumisvaihtoehto  
2009



## YRITYSREKISTERIN HALLINTAOHJELMISTON KÄYTETTÄVYYDEN PARANTAMINEN

Rantanen, Jarno  
Satakunnan ammattikorkeakoulu  
Tekniikka ja merenkulku Pori  
Tietotekniikan koulutusohjelma  
Joulukuu 2009  
Työn ohjaaja: Kivi, Karri  
Sivumäärä: 40

Asiasanat: tietokanta, verkkosivu, Microsoft Access

---

Tämän opinnäytetyön aiheena oli vähentää Kehitys-Parkki Oyn ylläpitämän yritysrekisterin yritystietojen ajan tasalla pitämiseen tarvittavaa työpanosta. Kehitys-Parkki Oy tarjoaa verkkosivuillaan noin 750 yrityksen osoitetietoja, mutta verkkosivut mahdollistava rekisteri oli hajanainen ja sisälsi päällekkäistä sekä ristiriitaista tietoa. Tästä johtuen yritystietojen muuttaminen kulutti paljon työaikaa ja oli virhealtista.

Tutkielma aloitettiin suunnittelemalla uusi tietokantarakenne ja siirtämällä tietokanta palvelimelle, jossa se olisi kaikkien järjestelmän eri osien käytettävissä. Käytössä olleen yritysrekisterin tavoin Kehitys-Parkki Oyn työntekijä muuttaa edelleen yritystietoja Microsoft Accessin kautta, mutta paikallisen tietokannan sijaan tiedot ohjataan uudessa yritysrekisterissä palvelimella sijaitsevaan tietokantaan. Kaikille avoimet yritysrekisterin verkkosivut rakentuvat automaattisesti tietokannan tietojen pohjalta, joten ylimääräistä työtä ei enää tässä kohtaa vaadita.

Yritystietojen keskittäminen yhteen, rakenteeltaan hyvin suunniteltuun tietokantaan, ratkaisi tiedon hajanaisuudesta aiheutuvat virheet. Samalla myös yritystietojen muuttamisesta poistui useampi työvaihe, joten projektin tavoitteeksi asetettu muutosprosessin tehostaminen toteutui. Uudessa järjestelmässä tiedon muuttaminen kerran saa muutoksen heijastumaan automaattisesti kaikkiin järjestelmän eri osiin ilman ylimääräistä työtä.

## INCREASING USABILITY OF COMPANY REGISTRY MANAGEMENT

Rantanen, Jarno  
Satakunta University of Applied Sciences  
Faculty of Technology and Maritime Management Pori  
Degree Programme in Information Technology  
December 2009  
Supervisor: Kivi, Karri  
Number of pages: 40

Key words: database, webpage, Microsoft Access

---

The purpose of this thesis was to decrease the amount of work needed to keep company information up-to-date on a Kehitys-Parkki Oy maintained registry of companies. Kehitys-Parkki Oy offers address information of 750 companies on its webpage but registry containing the information was scattered and it contained overlapping and conflicting data. Because of this updating company's address information required a lot of work and was prone to errors.

In the first part of this thesis new database architecture was designed and the resulting database was moved to a server in order to make it available for all elements of the system. In the new system Kehitys-Parkki Oy employees still change company's information manually using Access as in the original registry of companies, but instead of using a local database the data is now transferred to a database server. The openly available webpage for browsing company information builds itself automatically using the data on the database, thus no extra work is required.

Concentrating all company information together in a structurally well designed database solved the issues originating from dispersed data. This also removed several required steps from the company information update process, which was the goal of this thesis. In the new system changing company's information once is reflected throughout the whole system without any additional work.

# SISÄLLYS

1	JOHDANTO.....	6
2	VAATIMUSMÄÄRITTELY .....	7
2.1	Projektin tavoitteet .....	7
2.2	Käyttötapaukset .....	8
2.2.1	Julkiset internet-sivut .....	8
2.2.2	Kehitys-Parkki Oy.....	8
2.3	Ongelmakohdat ja parannusehdotukset .....	8
3	TIETOKANTA .....	11
3.1	Tietokantasuunnittelu.....	11
3.1.1	Hierarkkinen taulun sisältö tietokannassa .....	14
3.2	Taulujen luonti.....	18
3.3	Olemassa olevien yritystietojen siirtäminen uuteen tietokantaan.....	18
4	VERKKOSIVUT.....	20
4.1	Database-luokka.....	25
4.2	Testaus .....	27
4.3	Käyttöönotto .....	27
5	ACCESS .....	29
5.1	ODBC .....	29
5.2	Käyttöliittymä .....	31
6	VALMIIN PROJEKTIN ARVIOINTI.....	36
6.1	Kehitysideoita .....	37
	LÄHTEET.....	40

## TERMILUETTELO

Tietue	Relaatiotietokannan ollessa kyseessä tietueella tarkoitetaan taulun yhtä riviä. Esimerkiksi Yritys-taulun tietue sisältää yhden yrityksen osoite- ja yhteystiedot.
CSS	Cascading Style Sheets, yleisesti verkkosivuilla käytettävä tyyliohje, eli tyylin määrittelevä säännöstö.
HTML	Hypertext Markup Language, erityisesti verkkosivujen luomiseksi käytetty hyperlinkkejä sisältävän tekstin merkkauskieli.
SSH	Secure Shell, verkkoprotokolla, jonka kautta verkkoyhteydet voidaan suojatusti ohjata.
MySQL	My Structured Query Language, SQL-tietokannan hallintajärjestelmä.
Python	Tulkattava, korkean tason ohjelmointikieli. Laaja standardikirjasto ja Python-koodin hyvä luettavuus tekevät kehitystyöstä nopeaa ja tehokasta.
PHP	PHP: Hypertext Preprocessor, erityisesti verkkopalvelinympäristössä käytetty korkean tason ohjelmointikieli

## 1 JOHDANTO

Kehitys-Parkki Oy on Parkanossa sijaitseva elinkeinoyhtiö, jonka tavoitteena on Parkanon ja Kihniön alueen elinkeinoelämän tukeminen ja kehittäminen. Palveluita tarjotaan aloittavan yrityksen liikeidea-analyysistä pidempi-ikäisten yritysten omistajanvaihdoksiin sekä kehityssuunnitelmiin.

Muiden palveluiden lisäksi Kehitys-Parkki Oy ylläpitää Parkanon ja Kihniön alueen yritysrekisteriä. Yritysrekisterissä on noin 750 yrityksen tiedot, joista 600 sijaitsee Parkanossa ja 150 Kihniössä.

Kehitys-Parkki Oyn sivuilla alueen yrityksistä näytetään yhteystiedot ja yhteyshenkilöt. Nämä tiedot ovat näkyvissä kaikille sivuille vierailijoille. Yrityksiä on mahdollista etsiä toimialoittain tai aakkosellisesta hakemistosta.

Yritysrekisteri on koettu hyödylliseksi ja sen sivut toimiviksi, mutta yritysrekisterin ylläpitäminen on ollut liian vaivalloista. Yrityksen tiedot on säilytetty Microsoft Access -tietokannassa, josta tiedot on päivitetty html-tiedostoon ja lopuksi tiedosto on siirretty palvelimelle, jolloin muutokset ovat päivittyneet Kehitys-Parkki Oyn internet-sivuille. Lisätyötä aiheutti toimialahaun puumainen rakenne, jossa sama yritys voi sijaita useamman eri toimialan alla. Tästä syystä samat yritystiedot piti päivittää useaan eri toimialaan ja yhden toimialan muutoksien unohtaminen jätti samalle yritykselle näkyviin kahdet ristiriitaiset yhteystiedot.

Yritystietojen muuttamisprosessi haluttiin saada mahdollisimman yksinkertaiseksi ottamalla käyttöön yritystiedot yhteen paikkaan kokoava tietokanta, jossa ei esiintyisi päällekkäistä tai ristiriitaista tietoa. Kehitys-Parkki Oyn työntekijät muuttaisivat yritystietoja tuttuun tapaan Accessin kautta, mutta muutokset ohjattaisiin tietokantaan. Yritysrekisterin verkkosivut lukisivat yritystiedot tietokannasta ja loisivat sivut niiden pohjalta.

## 2 VAATIMUSMÄÄRITTELY

### 2.1 Projektin tavoitteet

Parkanon ja Kihniön alueen yrityksiltä tuli säännöllisesti ilmoituksia osoitemuutoksista tai muista yhteystietojen muutoksista. Yritykset laajensivat toimintaansa muille toimialoille ja osa lopetti kokonaan toimintansa. Samalla myös uusia yrityksiä perustettiin. Kaikki nämä muutokset vaativat Parkanon ja Kihniön alueen yritysrekisterin verkkosivuilla näkyäkseen Kehitys-Parkki Oyn työntekijän työpanosta, joka kasvoi helposti suhteettoman suureksi ja virhealttiiksi järjestelmän monimutkaisen ja pirstaloituneen rakenteen vuoksi. Tämä työpanos oli pois kaikesta muusta alueen elinkeinoelämän kehittämisestä.

Projektin päällimmäisin tavoite on yritystietojen muuttamisprosessin yksinkertaistaminen. Tämän saavuttamiseksi on tärkeää saada yritystiedot keskitettyä yhteen tietokantaan, jotta voidaan olla varmoja järjestelmän osien käyttävän täsmälleen samoja yritystietoja.

Yritystietojen muutosprosessin virtaviivaistaminen halutaan kuitenkin saavuttaa menettämättä ennestään Kehitys-Parkki Oyn sisäisessä käytössä olleita yritysrekisterin toimintoja. Näitä ovat mm. yrityksiä osoitetietojen tarratulostus erinäisten kriteerien pohjalta, kuten henkilöstömäärältään suurimmat yritykset.

Projekti aloitettiin keskikesällä ja syksyllä Parkanossa alkaisi uuden asuin- ja yritysalueen kaavoitusprojekti, joka vaatisi työntekijöiden huomion. Tästä johtuen yritysrekisteri haluttiin saada mahdollisimman valmiiksi syksyyn mennessä.

## 2.2 Käyttötapaukset

Projektin alussa selvitettiin millaisiin toimintoihin yritysrekisteriä yleisimmin käytetään. Näitä tietoja käytettiin myöhemmin tietokannan ja käyttöliittymän toteuttamiseen. Kehitys-Parkki Oyillä oli jo ennestään yritystietoja tallennettuna, joten uudet vaatimukset rakennettiin pitkälti vanhojen kokemusten pohjalta parannusehdotuksia mieltien.

### 2.2.1 Julkiset internet-sivut

Sivujen kautta yritykset ja yksityishenkilöt voivat hakea yritysten yhteystietoja. Yrityksiä voi myös hakea toimialan perusteella, jolloin on mahdollista esimerkiksi listata sähköasennuksia tekevät yritykset. Sivujen rakenteeseen ja ulkoasuun oltiin tyytyväisiä, joten näihin asioihin ei koettu tarpeelliseksi tässä projektissa juurikaan puuttua.

### 2.2.2 Kehitys-Parkki Oy

Vanhassa järjestelmässä yritystiedot tallennettiin Microsoft Access –tietokantaan. Tietojen tallentamisen lisäksi Accessia oltiin käytetty osoitetietojen helppoon tulostamiseen tarra-arkille. Osoitetietojen lisäksi yrityksestä kirjattiin myös liikevaihto ja työntekijöiden lukumäärä. Näitä tietoja käytettiin kun haluttiin lähettää kirjeitä esimerkiksi sadalle suurimmalle yritykselle.

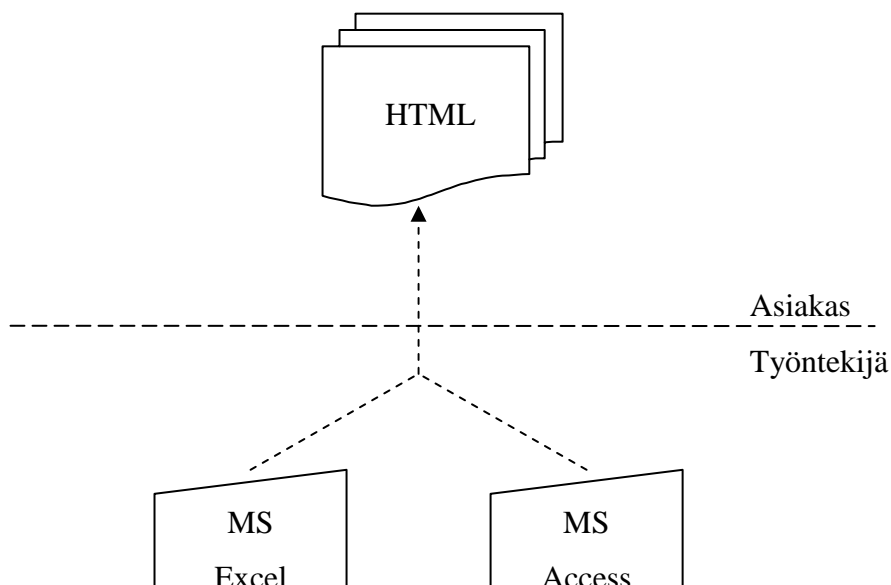
Julkisilla sivuilla yrityksiä on mahdollista listata toimialan mukaan. Tästä johtuen myös yrityksen toimialat piti tallentaa tietokantaan. Lisäksi toimialatietoja tarvittiin tulostettaessa tietyn toimialan yritysten osoitteita.

## 2.3 Ongelmakohdat ja parannusehdotukset

Yritysrekisterin kasvaessa yritystietojen päivittäminen tietokantaan ja siitä eteenpäin Kehitys-Parkki Oyn sivuille on muodostunut liian aikaa vieväksi tapahtumaksi. Tä-

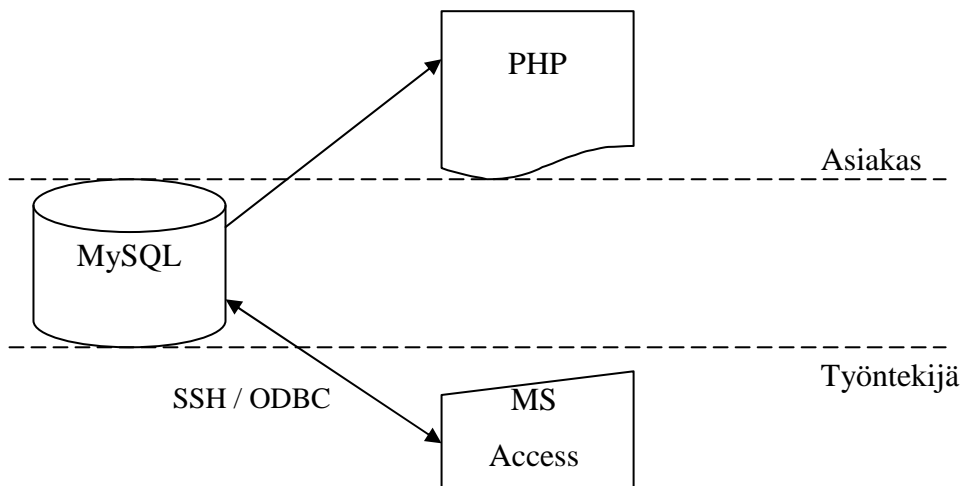


mä haluttiin yksinkertaistaa niin, että yrityksen tietojen muuttaminen näkyy välittömästi sivuilla ja päällekkäiseltä tiedolta vältytään.



Kaavio 2.1. Vanhassa muutosprosessissa Kehitys-Parkki Oyn työntekijä muuttaa tietoja kolmeen eri paikkaan. Access ja Excel ovat sisäisessä käytössä ja näiden tiedoista kootaan ulkopuolelle näkyvät HTML-sivut. Päivitys ei ole automaattinen, vaan vaatii työntekijän muokkaamaan sivuja ja lataamaan uudet sivut palvelimelle.

Kaikki hajallaan olevat yritystiedot oli siirrettävä yhteen tietokantaan, jotta ristiriitaisista yritystiedoista päästäisiin eroon. Lisäksi kaikki aikaisemmin yritysrekisterin työntekijöiden käytettävissä olleet toiminnot haluttiin säilyttää uuteen järjestelmään. Tästä johtuen Access säilytettiin työntekijöiden käyttöliittymänä, mutta varsinainen tietokanta siirrettiin palvelimelle, jotta yritysrekisterin sivut voivat käyttää suoraan tietokannan tietoja.



Kaavio 2.2. Uudessa muutosprosessissa muutokset tehdään ainoastaan Accessiin, josta muutokset siirtyvät tietokantapalvelimelle. PHP muodostaa tietokannan tiedoista asiakkaalle näkyvän sivun.

Yritysrekisterin www-sivut oltiin jo käytössä todettu toimiviksi, joten suurin osa työstä kohdistettaisiin yritystietojen päivittämisen helpottamiseen sivujen ulkoasun muuttamisen sijaan.

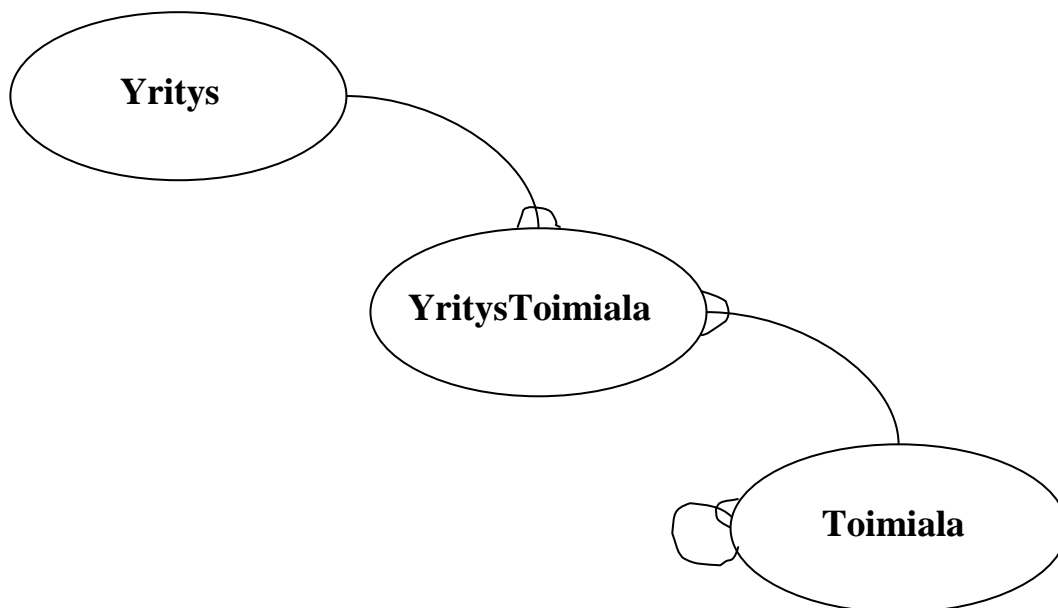
### 3 TIETOKANTA

Projektin alussa yritysrekisterissä oli noin 750 yrityksen tiedot, jotka haluttiin saada myös uuteen yritysrekisteriin. Uusi MySQL-tietokanta tulisi sijaitsemaan samalla palvelimella, jossa Kehitys-Parkki Oyn sivut olivat jo näkyvissä. MySQL-tietokanta korvaa vanhan yksitauluisen Microsoft Access –tietokannan, joten tietokannan rakenne haluttiin samalla suunnitella uudestaan ennen yritystietojen siirtämistä.

#### 3.1 Tietokantasuunnittelu

Uusi datamalli suunniteltiin yllä mainittuihin vaatimuksiin pohjautuen. Oli selvää, että yritysrekisteri tulee sisältämään yrityksen tietoa ja nämä yritykset tulevat kuulumaan tiettyihin toimialoihin. Tästä johtuen tietokannassa tulee olemaan Yritys- ja Toimiala-nimiset taulut.

Yhdessä toimialassa voi olla useampi yritys ja vastaavasti sama yritys voi kuulua useaan eri toimialaan. Tällainen monen suhde moneen yhteys tietokannan taulujen välillä on tarpeen purkaa käyttämällä ylimääräistä taulua Yritys- ja Toimiala-taulujen välissä, jolloin lopulliseksi tietokannan suunnitelmaksi muodostui kaavion 3.1 esittämä tietokantarakenne.



Kaavio 3.1. Yritysrekisterin tietokannan taulut ja niiden väliset yhteydet.

Taulujen kenttien määrittelyssä hyödynnettiin Kehitys-Parkki Oyn edellistä, olemassa olevaa paikallista tietokantaa sekä sen käytöstä nousseita parannusehdotuksia. Vanha tietokanta koostui vain yhdestä taulusta. Tämän vuoksi taulussa oli yhdellä tietueella samankaltaisia kenttiä. Jokaista yritystä kohti oli esimerkiksi kentät toimiala1, toimiala2 ja toimiala3, joita täytettiin tarpeen mukaan. Uudessa tietokannassa tieto hajautettaisiin eri tauluihin, jonka vuoksi osa vanhoista kentistä voitiin poistaa. Uuden tietokantarakenteen taulujen välisten yhteyksien vuoksi muutama kenttä myös lisättiin.

Vaikka tarvittavat kentät määriteltiin tarkasti ennen työn aloittamista, niin kenttiä jouduttiin tästä huolimatta projektin loppuvaiheilla lisäämään ohjelman käytössä esiin nousseiden kokemusten tukemana. Tämä oltiin osattu ennakoida projektin alussa, joten uusien kenttien sovittaminen käytössä olevaan ohjelmakoodiin ei ollut työläs operaatio.

Taulukko 3.1. Yritys-taulu.

Kenttä	Tietotyyppi	Rajoitteet	Suhde (relaatio)
id	Unsigned int(10)	Pääavain	
ytunnus	Varchar(9)	Yksilöllinen	
toiminimi	varchar(128)	Ei tyhjiä arvoja, yksilöllinen	
aputoiminimi	varchar(128)		
kayntiosoite	varchar(128)		
postiosoite	varchar(128)		
postinro	varchar(5)		
ptoimipaikka	varchar(128)		
puh	varchar(64)		
gsm	varchar(64)		
fax	varchar(64)		
etunimi	varchar(64)		
sukunimi	varchar(64)		
wwwosoite	text		
email	text		
henkilosto	int(11)		
perustamispvm	date		
yritysmuoto	varchar(32)		
liikevaihto	bigint(20)		
alue	varchar(32)		
info	text		

Taulukko 3.2. Toimiala-taulu.

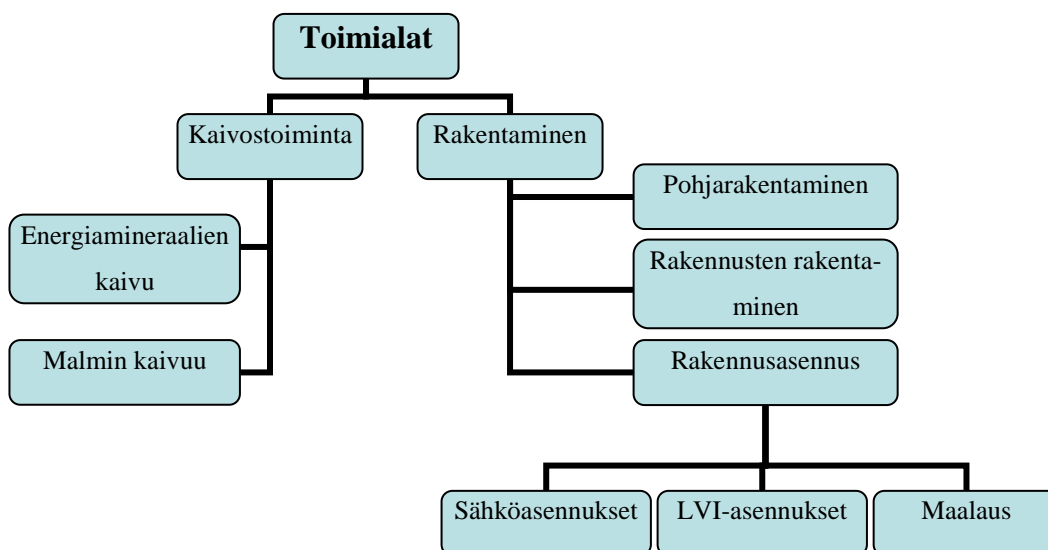
Kenttä	Tietotyyppi	Rajoitteet	Suhteet
tunnus	char(3)	Pääavain	
kuvaus	text	Ei tyhjiä arvoja	
ylätoimiala	char(3)		Toimiala.tunnus

Taulukko 3.3. YritysToimiala-taulu. Toimii viitetauluna Yritys- ja Toimiala-taulujen välillä.

Kenttä	Tietotyyppi	Rajoitteet	Suhteet
yritysviite	unsigned int(10)	Ei tyhjiä arvoja	Yritys.id
toimialaviite	char(3)	Ei tyhjiä arvoja	Toimiala.tunnus

### 3.1.1 Hierarkinen taulun sisältö tietokannassa

Tietokannan taulun rakenne ei ole hierarkinen, vaan sisältää yksittäisiä, toisiinsa liittymättömiä tietueita. Yritysrekisterin tietokannassa Toimiala-taulu kuitenkin sisältää puumaisen listan toimialoista. Toimiala voi olla päätoimiala, päätoimialan alatoimiala, tai alatoimialan alatoimiala. Tätä syvemmäksi hierarkia ei kasva. Myös yritysrekisterissä olevat yritykset kuuluvat yhteen tai useampaan toimialaan, mutta tämä tapahtuu normaaleilla tietokantaviitteillä Yritys- ja Toimiala-taulujen välillä, eikä sinällään liity Toimiala-taulun hierarkiseen rakenteeseen.



Kaavio 3.2 Esimerkki Toimiala-taulun sisällöstä ja niiden suhteista.

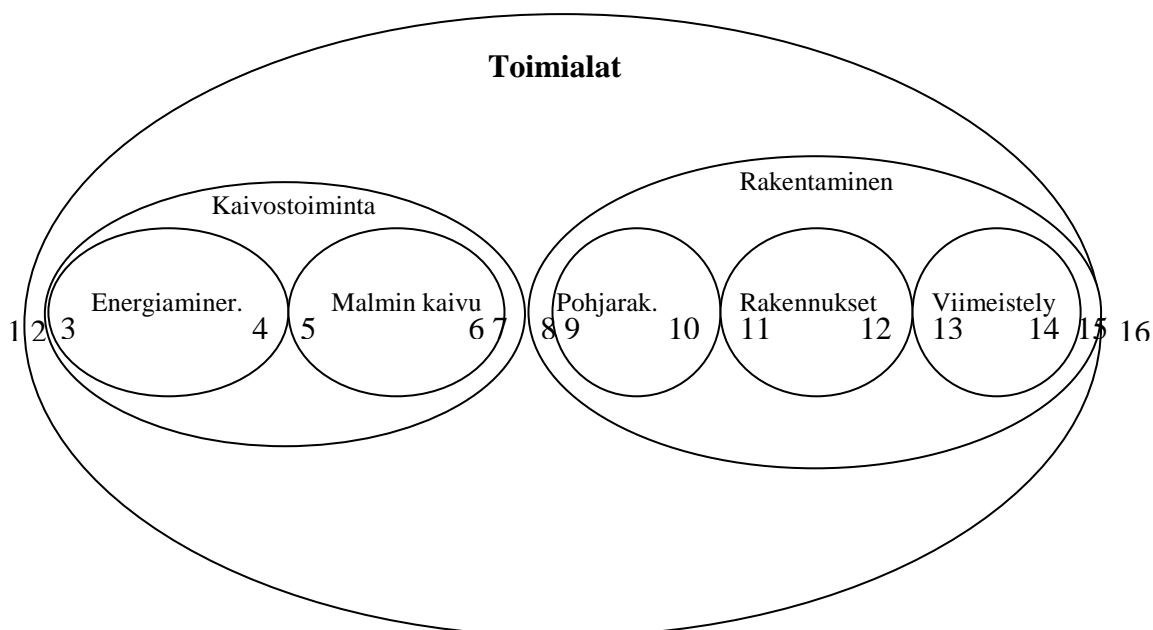
Hierarkisen tiedon esittämiseen taulun sisällä on olemassa pari yleisesti käytettyä tapaa. Perinteisesti taulussa otetaan käyttöön sisäinen yhteys, jolloin lapsi viittaa vanhempansa, eli Adjacency List –malli. Yritysrekisterin tapauksessa tämä tarkoittaa ylätoimialan tallentamista taulun tietueeseen. Päätoimialoilla ylätoimialan kenttä jää tyhjäksi.

Taulukko 3.4. Adjacency List -mallin mukaiset arvot tietokannassa. Tietue sisältää tiedon vanhemmastaan, eli tässä tapauksessa ylätoimialastaan.

Tunnus	Kuvaus	Ylätoimiala
C	Kaivostoiminta	
CA	Energiamineraalien kaivu	C
CB	Malmin kaivu	C
F	Rakentaminen	
FA	Pohjarakentaminen	F
FB	Rakennusten ja niiden osien rakentaminen	F
FC	Rakennusasennus, viimeistelytyöt	F
451	Sähköasennukset	FC
452	LVI-asennukset ja -urakointi	FC
453	Maalaus	FC

Adjacency List -mallissa kaikkien toimialojen hakeminen yhdellä haulla olisi hankalaa. Puhdas sql-haku vaatii taulun sisäisen JOIN-komennon käyttöä jokaista hierarkian tasoa kohti, jonka vuoksi tämä käykin hankalaksi hierarkian kasvaessa. Käytännössä tämä ongelma on kierrettävissä tietokannan ulkopuolella kirjoittamalla koodia, joka noutaa silmukassa taulun tietoa ja lopuksi yhdistää tiedot yhdeksi helposti käsiteltäväksi listaksi.

Saman ongelman ratkaisemiseksi on olemassa toinen tapa. Nested Set -malli jakaa toimialat sisäkkäisiin säiliöihin ja toimialojen reunat numeroidaan vasemmalta oikealle.



Kaavio 3.3. Toimiala-aulun sisältö Nested Set –mallin mukaisessa muodossa.

Toimialojen vasemmat ja oikeat arvot säilytetään tietokannassa. Esimerkiksi kaivostoiminta kattaa kaavion 3.3 tapauksessa toimialat rajoista kahdesta seitsemään.

Taulukko 3.5. Nested Set –mallin mukaiset arvot tietokannassa.

Tunnus	Kuvaus	vasen	oikea
1	Toimialat	1	16
2	Kaivostoiminta	2	7
3	Energiamineraalien kaivu	3	4
4	Malmin kaivu	5	6
5	Rakentaminen	8	15
6	Pohjarakentaminen	9	10
7	Rakennukset	11	12
8	Viimeistely	13	14

Nested Set -malli tekee toimialojen hakemisesta helppoa, sillä alitoimialan rajojen numeroarvot kuuluvat aina päätoimialansa vasen- ja oikea-arvojen sisään. Esimerkki 3.1 mukaisen haun laajuutta voi helposti rajoittaa määrittämällä ylätoimialaksi jonkun muun kuin ”Toimialat”-säiliön, esimerkiksi ”Kaivostoiminta” hakisi ainoastaan sen alle kuuluvat toimialat.



```

SELECT tmiala.kuvaus
FROM Toimiala AS tmiala,
Toimiala AS ylatmiala
WHERE tmiala.vasen BETWEEN ylatmiala.vasen AND ylatmiala.oikea
AND ylatmiala.kuvaus = 'Toimialat'
ORDER BY tmiala.vasen;

```

Esimerkki 3.1 SQL-komento kaikkien toimialojen kuvauksien hakuun Nested Set -mallissa

Myös alatoimialojen hakeminen on tässä mallissa helppoa, sillä vasemman ja oikean rajan erotus on yksi, jota voidaan hyödyntää hakuena. Toimialojen lisääminen puolestaan vaatii kaikkien lisätyn toimialan oikealle puolelle jäävien toimialojen rajojen numeroarvojen kasvattamista kahdella. Jos lisäämme ”Kaivostoiminta”-toimialan alle uuden alitoimialan nimeltään ”Jalostus”, niin ensiksi on selvitettävä lisättävän toimialan paikka. Esimerkissä 3.2 lisäämme uuden toimialamme ”Malmin kaivu”-alitoimialan perään. Lisäyskohdan selvittämisen jälkeen päivitämme sitä suuremmat rajojen arvot ja lopuksi lisäämme toimialamme lisäyskohtaan.

```

SELECT @lisayskohta := oikea FROM Toimiala
WHERE kuvaus = 'Malmin kaivu';
UPDATE Toimiala SET oikea = oikea + 2 WHERE oikea > @lisayskohta;
UPDATE Toimiala SET vasen = vasen + 2 WHERE vasen > @lisayskohta;
INSERT INTO Toimiala(kuvaus,vasen,oikea)
VALUES('Jalostus', @lisayskohta + 1, @lisayskohta + 2);

```

Esimerkki 3.2. SQL-komento toimialan lisäykselle ja siitä seuraava Nested Set -mallin rajojen päivitys.

Ensimmäiseksi mainitun Adjacency List -mallin haut käyvät sitä monimutkaisemmaksi, mitä enemmän taulussa on toimialojen tasoja. Yritysrekisterissä on runkona tilastokeskuksen käyttämä toimialaluokitus, joista tarpeellisiksi on todettu kolme ensimmäistä tasoa. Vaikka Toimialat-tauluun saattaa tulla lisää toimialojen kuvauksia, niin hierarkian syvyyttä ei tulla kasvattamaan. Lisäksi yritysrekisterin sivujen luonti tehdään PHP:llä, jossa tietokantahaun tuloksien yhdistely on helppoa. Näistä tietokantahakua helpottavista syistä johtuen yritysrekisteri tulisi käyttämään Adjacency List -mallia toimialojen esittämiseksi.

### 3.2 Taulujen luonti

Vaatimusmäärittelyn mukaiset tietokannan taulut ja taulujen kentät kirjattiin sql-tiedostoon ”create table”-komennoina ja tiedosto suoritettiin tietokantapalvelimella.

### 3.3 Olemassa olevien yritystietojen siirtäminen uuteen tietokantaan

Yritystiedot piti saada siirrettyä Microsoft Access –tietokannasta uuteen, palvelimella sijaitsevaan MySQL-tietokantaan. Tiedon siirtämiseen eri tietokantojen välillä on saatavilla lukuisia eri ohjelmia. MySQL-tietokantoihin on tätä varten olemassa MySQL Migration Toolkit -niminen työkalu, jota myös tässä projektissa käytettiin.

Tiedon siirtäminen toiseen tietokantaan vaatii ensimmäiseksi alkuperäisen tiedon hakemisen Microsoft Access –tietokannasta. Oletuksena Microsoft Access ei päästä ulkopuolisia ohjelmia käsiksi tietokantoihinsa, joten tämä on erikseen sallittava. Käytännössä tämä tapahtuu ensin paljastamalla sisäiset järjestelmätaulut asetusten ”näkyvä”-välilehdeltä. Tarvittavat järjestelmätaulut ovat MSysObjects, MsysQueries ja MsysRelationships. Tämän jälkeen kyseisiin järjestelmätauluihin on annettava pääkäyttäjän oikeudet tietoturvalavikon kautta. Kun nämä kaksi asiaa on saatu tehtyä, niin MySQL Migration Toolkit pystyy lukemaan kaiken tarvitsemansa tiedon.

Usein uuteen tietokantaan siirtyessä myös tietokannan rakenne on muuttunut, joten siirtymä vaatii muitakin muutoksia kuin pelkän syntaksin sovittamisen. Näin oli myös tämän projektin tapauksessa. MySQL Migration Toolkit mahdollistaa koko prosessin kerralla tekemisen, eli tiedon hakemisen vanhasta tietokannasta, tiedon rakenteen muuttamisen skripteillä, tiedon sisällön manuaalisen muuttamisen sekä lopuksi tiedon siirtämisen uuteen tietokantaan. Yhtenäisen siirtoprosessin luomisesta hyödytään kuitenkin vasta kun siirto halutaan toistaa useamman kerran. Tämä ei ollut tarpeen, joten vanhat yritystiedot otettiin muuttamattomina SQL-tiedostoihin MySQL Migration Toolkitin avulla. SQL-tiedostot sisältävät komennot taulujen luomiseen (create table) sekä tiedon tallentamisen tauluihin (insert into).

```
Insert into Yritys (tmi1,tmi2,tmi3,nimi,osoite) values ('F','Fa','231','Nimi','tie 2A');
```

Esimerkki 3.3. Alkuperäisen tietokantarakenteen mukainen insert-lause.

Tietokannan rakenteen muuttamisen vuoksi SQL-tiedoston yritystietoja ei voitu hyödyntää sellaisenaan, joten tiedoston sisältö piti muokata uusien taulujen mukaiseksi. Tähän käytettiin Python-ohjelmointikieltä, jolla rivien sarakkeet pilkottiin osiin ja kirjoitettiin toiseen SQL-tiedostoon uuden tietokannan rakenteen mukaisesti. Toinen tapa olisi ollut kopioida SQL-tiedoston sisältö sellaisenaan MySQL-tietokantaan vanhaa rakennetta käyttäen, jonka jälkeen tieto oltaisiin voitu siirtää tietokannan sisällä uusiin tauluihin tietokannan komentoja käyttäen. Lopputuloksen kannalta eroa ei ole ja Python koettiin tässä tapauksessa vaivattommaksi.

```
Insert into Yritys (id, toiminimi, postiosoite) values (1, 'Nimi', 'tie 2A');  
Insert into YritysToimiala (yritysviite, toimialaviite) values (1, 'F');  
Insert into YritysToimiala (yritysviite, toimialaviite) values (1, 'Fa');  
Insert into YritysToimiala (yritysviite, toimialaviite) values (1, '231');
```

Esimerkki 3.4 Uuden tietokantarakenteen mukaiset, useampaa taulua käyttävät insert-lauseet

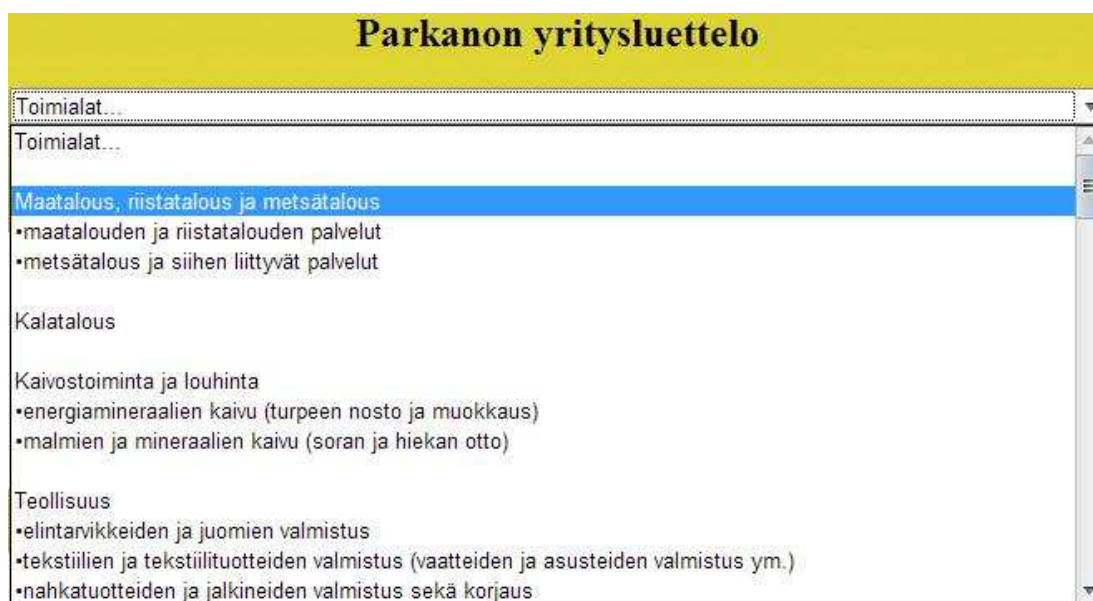
Toimialojen tietoa ja hierarkiaa ei ollut mahdollista kopioida olemassa olevasta yritysrekisteristä, joten toimialatiedot jouduttiin kirjoittamaan uuteen tietokantaan manuaalisesti ennen SQL-tiedostojen sisällön kopiointia. Toimialojen ja yritysten tietojen kopiointi uuteen tietokantaan tapahtui siirtämällä SQL-tiedosto palvelimelle, jonka jälkeen tiedosto ladattiin komentokehityksen kautta tietokantaan.

```
shell> mysqlimport [options] tietokannan_nimi uudet_tiedot.sql
```

Esimerkki 3.5 Komentokehoite uuden tiedon lataamiselle tietokantaan mysqlimport-sovellusta käyttäen.

## 4 VERKKOSIVUT

Käytössä olleet yritysrekisterin sivut oli jaettu Parkanon ja Kihniön yrityksiin omille sivuilleen. Sivuilta yrityksiä hakeva vierailija valitsi alasvetovalikosta toimialan, jonka jälkeen näkymä siirtyi kyseisen toimialan sivulle. Näitä yritykset listaavia toimialasivuja tarvittiin yksi jokaista toimialaa kohti, joten Parkanon yrityksiä varten palvelimella oli 93 html-tiedostoa ja lisäksi sama määrä Kihniön yrityksiä varten. Usein yritys kuuluu useampaan toimialaan, jonka vuoksi yritystietojen muuttaminen oli työlästä ja virhealtista.

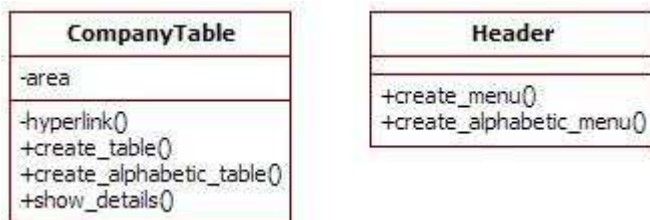


Kuva 4.1. Toimialan valinta Parkanon alueen yritysrekisterissä.

Uusien sivujen yrityslistaus muodostettaisiin dynaamisesti valitun toimialan mukaan. PHP:ssä toimialojen alasvetovalikko välittää funktion parametriksi valitun toimialan, jonka perusteella funktio hakee ja listaa tietokannasta kyseisen toimialan ja sen alatoimialojen yritykset näkyviin.

Uudet sivut rakennetaan ulkoasultaan vanhojen sivujen näköisiksi. Sivun rakenne voidaan jakaa kahteen osaan. Sivun yläosaan kuuluu toimialavalikko sekä aakkosellinen hakemisto. Riippuen sivuilla vierailevan valinnasta, valikoiden alapuolella näy-

tetään joko aakkosellisen haun tulokset tai tietyn toimialan yritykset. Näiden luomisesta huolehtivat luokat ovat elements.php-nimisessä tiedostossa.



Kuva 4.2. Elements.php tiedoston luokkarakenne. Vaatii database.php tiedoston toimiakseen.

Header luokka sisältää jäsenfunktiot valikoiden luomiseksi. Create\_alphabetic\_menu listaa aakkoset riviksi html formin input-elementtejä.



Kuva 4.3. Parkanon toimialavalikko, jonka alla aakkosellisen yrityshaun valikko.

Kuvassa 4.4 CSS-tiedosto vaihtaa aakkosten tyylin linkeiksi kookkaiden painikkeiden sijaan.

```

input.alpha_menu {
  font-family: "Lucida Console", Courier, "Courier New", monospace;
  background-color: transparent;
  border-top: none;
  border-left: none;
  border-right: none;
  border-bottom: 1px solid #0000FF;
  cursor: pointer;
  padding: 0;
}
  
```

Kuva 4.4. Input-elementin muuttaminen linkin kaltaiseksi.

Create\_menu-jäsenfunktio luo toimialojen alasetovalikon. Parametrina se voi ottaa toimialan tunnusteen, jolloin tämä toimiala näkyy valikossa valittuna. Ensiksi jäsenfunktio hakee kaikki ylätoimialat. Kaikki näin löydetty toimialat käydään yksitellen silmukassa läpi, jossa haetaan niiden alatoimialat. Tätä seuraa sisempi silmukka, joka lisää päätoimialan alasetovalikkoon, sisentää sen alle alatoimialat ja näiden alle vielä alitoimialojen alitoimialat.

Toimialaa alavetovalikosta valittaessa lomake lähettää valinnan heti eteenpäin. Tämä vaatii kuitenkin selaimelta JavaScript-tuen toimiakseen, joka on huomioitu lisäämällä ”Hae”-painike alavetovalikon perään JavaScript-tuen puuttuessa.

```

public function create_menu($selected) {
// create a drop-down menu for categories
try {
$db = Database::connection();
$mainres = $db->send_query("select tunnus from Toimiala where ylatoimiala is NULL");
echo "<form action='".$_SERVER['PHP_SELF']."' method='post'>\n";
echo "<select name='category' onchange='this.form.submit()'>\n";
echo "<option value='empty'> Toimialat... \n";
$IDs = array("");
while($mainrow=mysql_fetch_row($mainres)) {
    $main = $mainrow[0];
    $query = "SELECT t1.kuvaus AS lev1, t2.kuvaus as lev2, t3.kuvaus as lev3, ".
        "t1.tunnus AS id1, t2.tunnus as id2, t3.tunnus as id3 ".
        "FROM Toimiala AS t1 LEFT JOIN Toimiala AS t2 ON t2.ylatoimiala = t1.tunnus ".
        "LEFT JOIN Toimiala AS t3 ON t3.ylatoimiala = t2.tunnus ".
        "WHERE t1.tunnus='".$main."'";
    $leafres = $db->send_query($query);
    while($col=mysql_fetch_row($leafres)) {
        if (!in_array($col[3],$IDs)) { // top category
            echo "<option value='empty'> <br />\n";
            if ($col[3] == $selected) {
                echo "<option selected value='".$col[3]."'>".$col[0]."</option>\n";
            } else {
                echo "<option value='".$col[3]."'>".$col[0]."</option>\n";
            }
            $IDs[] = $col[3];
        }
        if (!in_array($col[4],$IDs)) { // middle category
            if ($col[4] == $selected) {
                echo "<option selected value='".$col[4]."'>&bull;".$col[1]."</option>\n";
            } else {
                echo "<option value='".$col[4]."'>&bull;".$col[1]."</option>\n";
            }
            $IDs[] = $col[4];
        }
        if (!in_array($col[5],$IDs)) { // bottom category
            if ($col[5] == $selected) {
                echo "<option selected value='".$col[5]."'>&nbsp;&bull;".$col[2]."</option>\n";
            } else {
                echo "<option value='".$col[5]."'>&nbsp;&nbsp;&bull;".$col[2]."</option>\n";
            }
            $IDs[] = $col[5];
        }
    }
}
echo "</select>\n";
echo "<noscript><input type='submit' value='Hae' /></noscript>\n";
echo "</form>\n";
} catch (Exception $e) {
    echo $e->getMessage();
}
}

```

Kuva 4.5. Header-luokan create\_menu-jäsenfunktio.

Create\_menussa on huomioitava, että hakuja tehdään yksi jokaista päätoimialaa kohden. Näin on välttytty tietokanta-luvussa mainituilta ongelmilta liittyen kaiken tiedon hakemiseen yhdellä tietokantahaulla.

CompanyTable-luokka sisältää jäsenfunktiot sivun sisällön luomiseen. Luokasta uutta instanssia luodessa parametriksi annetaan haluttu alue, eli käytännössä Parkano tai Kihniö. Tämä asetus rajoittaa luokan tietokantahakuja niin, että ainoastaan halutun alueen yritykset haetaan.

Create\_table-jäsenfunktiota kutsutaan toimialan valinnan jälkeen, jolloin parametriksi saadaan halutun toimialan tunnus. Kyseessä voi olla myös päätoimiala ja tässä tapauksessa sivujen on tarkoitus näyttää myös kyseisen toimialan alatoimialojen yritysten tiedot. Tästä johtuen koodissa haetaan ensin parametriksi saadun toimialan alatoimialat, jonka jälkeen toimialat lisätään yritystunnuksien haun ”where toimiala.tunnus in ‘.\$categories’” -ehtoon. Lopuksi yritystunnuksilla haetaan yritysten halutut tiedot ja nämä tiedot kirjoitetaan taulukkomuotoon sivuille.

Eräs kenttä yritystaulussa on yrityksen verkkosivujen osoite sekä sähköpostiosoite. Näistä tehdään linkit, johon on olemassa luokan sisäiseen käyttöön tarkoitettu hyperlink-jäsenfunktio.

Create\_alphabetic\_table toimii samalla periaatteella kuin create\_table, mutta saa parametriksi aakkosten kirjaimen. Tietokantahaku on yksinkertaisempi, sillä kaikkien toimialojen yritykset haetaan kunhan alkukirjain täsmää annettua parametria. Yritystietojen taulukkoon lisätään tässä tapauksessa yrityksen nimen kohdalle myös lisätietoihin vievä painike. Lisätietoja-sivu listaa yrityksen toimialat, sillä muuten niitä ei pysty aakkosellisesta hausta päättämään. Näiden tietojen näyttämistä huolehtii show\_details-jäsenfunktio.

Sivuilla yritystietoja etsivä vierailija ohjataan joko Parkano.php tai Kihnio.php -sivuille, riippuen kumman alueen yrityksistä ollaan kiinnostuneita. Nämä PHP-tiedostot luovat käyttäjille näkyvät sivut riippuen käyttäjän valinnoista kutsumalla Header ja CompanyTable -luokkia.

Yritysrekisterin sivut käyttävät HTTP:n POST-metodia kaikkien käyttäjiltä tulevien pyyntöjen lähettämiseen. Käyttäjän vieraillessa sivuilla ensimmäistä kertaa selain lähettää yritysrekisteriin GET-pyyntön, jolloin sivujen kaikki elementit näytetään alkutilassaan. Yritysrekisterin sivuilla käyttäjä ei voi syöttää tietoja rekisteriin, joten tässä tapauksessa POST-pyyntöillä tarkoitetaan aakkosellisen hakemiston painikkeiden painamista, toimialan valitsemista tai yrityksen lisätietojen pyytämistä. Yritysrekisterin valikoiden luomisessa ainoa muutos on toimialavalikossa, joka muistaa edellisen valinnan. Tästä johtuen Header-luokan jäsenfunktioita kutsuessa GET- ja POST-metodien käsittelyssä ei ole suuria eroja.

```

<div id="header">
  <h2>Parkanon yritysluettelo</h2>
  <?php
    if ($_SERVER["REQUEST_METHOD"]=="POST") {
      $category = $_POST['category'];
      $header = new Header;
      // keeps the requested category selected in the menu:
      $header->create_menu($category);
      $header->create_alphabetic_menu();
    } else { // GET
      $header = new Header;
      $header->create_menu();
      $header->create_alphabetic_menu();
    }
  ?>
</div>

```

Kuva 4.6. Sivun yläosan valikoiden luominen Header-luokkaa käyttäen.

Sivujen sisältöä luodessa GET-metodin kautta tulevat kutsut (alkutila) eivät näytä käyttäjälle kuin tekstin, jossa pyydetään valitsemaan toimiala. Kaikki varsinainen sisältö näytetään POST-metodin kautta ja sisältö vaihtelee käyttäjän valinnan mukaan. Valinta selvitetään normaaliin tapaan hakemalla lomakkeiden asettamien muuttujien sisällöt ja mikäli tietty muuttuja sisältää tietoa (esim. toimialat säilövä category-muuttuja) niin käyttäjälle tiedetään näyttää yritykset toimialahaun mukaisesti. Aakkoselliselle haulle ja yrityksiin lisätietojen pyytämiseen on myös omat muuttujansa.



```

<div id="content">
<?php
    if ($_SERVER["REQUEST_METHOD"]=="GET") {
        echo "<br /><h3>Tervetuloa! Voit valita haluamasi toimialan yllä olevasta valikosta.</h3><br />\n";
    }
    elseif ($_SERVER["REQUEST_METHOD"]=="POST") {
        $category = $_POST['category']; // User selected category from the drop-down menu
        $letter = $_POST['alpha_menu']; // User clicked on alphabetic letter links
        $company_name = $_POST['company_name']; // User is requesting company details
        $company_id = $_POST['id']; // User is requesting company details
        $companies = new CompanyTable("Parkano");
        if (isset($category)) {
            $companies->create_table($category);
        } elseif(isset($letter)) {
            $companies->create_alphabetic_table($letter);
        } elseif(isset($company_name) && isset($company_id)) {
            echo "<h3>Yrityksen ".$company_name." toimialat: </h3>\n";
            $companies->show_details($company_id);
        }
    }
?>
</div>

```

Kuva 4.7. Sivun sisällön luominen CompanyTable-luokkaa käyttäen.

Kuvissa 4.6 ja 4.7 olevat koodipätkät ovat molemmat Parkano.php-tiedostosta, mutta Kihnio.php on logiikaltaan identtinen. Ainoat erot löytyvät sivun ylälaidan otsikosta, CSS-tiedostosta löytyvistä tyyleistä sekä CompanyTablen luomisessa, jolloin luokalle annetaan parametrina ”Kihniö”. Periaatteessa kummatkin sivut olisi voinut esittää yhdenkin tiedoston kautta, sillä tällä hetkellä tiedostot sisältävät lähinnä toistensa kanssa päällekkäistä koodia. Mikäli Parkanon tai Kihniön sivuja yksilöidään enemmän tulevaisuudessa on muutoksien tekeminen kuitenkin helpompaa, kun käytettävissä on kaksi eri tiedostoa.

#### 4.1 Database-luokka

CompanyTable ja Header –luokat kumpikin tekevät tietokantahakuja, mutta ne eivät ole vastuussa tietokantayhteyksien hallinnasta. Yhteyksien luomiseen, hakujen suorittamiseen ja yhteyksien sulkemiseen luotiin Database-luokka. Luokka sijaitsee omassa database.php-tiedostossaan. Elements.php ei yritä suorittaa koodiaan ilman database.php-tiedostoa.

Database
-conn -instance
+connection() +send_query()

Kuva 4.8. Database.php tiedoston luokkarakenne.

Luokan konstruktori ei ole kutsuttavissa luokan ulkopuolelta, vaan siihen tarkoitukseen on connection-niminen staattinen metodi. Tämä palauttaa ensimmäisellä kerralla luokan uuden instanssin ja avaa tietokantayhteyden. Mikäli luokkaa yritetään luoda uudestaan, niin staattinen metodi palauttaa seuraavilla kerroilla suoraan jo aiemmin luodun luokan instanssin. Tällä tavalla luokka ei yritä turhaan avata useampaa tietokantayhteyttä. Luokan destruktori sulkee lopuksi tietokantayhteyden.

Send\_query-jäsenfunktio ottaa vastaan sille annetun SQL-komennon tekstimuodossa ja välittää sen eteenpäin käyttäen aiemmin luotua ja luokkamuuttujaan tallennettua tietokantayhteyttä. Lopuksi jäsenfunktio palauttaa resource-viittauksen, jota muut luokat voivat vapaasti käsitellä PHP:n tarjoamilla funktioilla. Database-luokkaa käytettäessä ei tarvitse huolehtia päällekkäisistä tai auki jäävistä tietokantayhteyksistä.

```
$companyquery = "select toiminimi from Yritys";
$db = Database::connection();
$res = $db->send_query($companyquery);
while($row=mysql_fetch_row($res)) {
    echo $row[0];
}
```

Esimerkki 4.1. Database-luokan käyttäminen koodissa.

Tietokantayhteyksien hoitaminen yhden luokan kautta mahdollistaa myös tietokannan salasanan tallentamisen ainoastaan yhteen paikkaan, joka helpottaa koodin muokkausta salasanaa vaihdettaessa. Luokka tallentaa kaikki tietokantayhteyksissä tapahtuvat mahdolliset virheet omaan tekstitiedostoonsa, jolloin sivujen vierailijoille näkyneet virheet tulevat lopulta myös sivujen ylläpitäjien tietoon. Suurempien käyttäjämäärien tapauksessa luokkaan voitaisiin helposti ja keskitetysti lisätä mahdollisuus useamman tietokantayhteyden käyttämiseen.

## 4.2 Testaus

Yritysrekisterin testaus tapahtui muuttamalla yritystietoja, poistamalla yrityksiä, lisäämällä niitä eri toimialoihin ja tarkistamalla, että muutokset näkyvät oikein sivuilla. Myös kaikki toimialat todettiin olevan oikein toimialavalikossa. Suurin osa yritysrekisterin koodista liittyy yritystietojen hakemiseen tietokannasta ja tämän kaltaisella koodilla on tapana toimia. Lisäksi yritysrekisterin sivut ovat projektin lopussa valmiit, eikä niihin tule kohdistumaan muutospaineita. Näin ollen testaus suoritettiin manuaalisesti pääosin yritysrekisterin kehityksen aikana, eikä testauksen automatisaatioon kiinnitetty huomiota.

Sivut sijaitsevat vuokratulla virtuaalipalvelimella, joten kohteliaisuusyistä rasisustestausta ei suoritettu, eikä se tämän kokoluokan projektilla olisi tarpeellistakaan. Mahdolliset suorituskykyongelmat havainnoitaisiin käyttöönoton aikana.

## 4.3 Käyttöönotto

Uudet PHP- ja CSS -tiedostot oli siirretty palvelimelle, jonka jälkeen Kehitys-Parkin sivuilla oleva yritysrekisterin linkki vaihdettiin osoittamaan uusiin sivuihin. Lopullinen tiedostolistaus:

- database.php
- elements.php
- kihnio.php
- parkano.php
- kihnio\_style.css
- parkano\_style.css
- style.css

Ennen käyttöönottoa tietokannan sisältö varmuuskopioitiin. Tämä on toistettu pari kertaa käyttöönoton jälkeenkin. Tietokannan tiedon saamiseen käytetään mysqldump-ohjelmaa. Yksinkertaisimmillaan taulujen rakenteen ja sisällön kopiointi onnistuu ”mysqldump --password=salasana tietokannan\_nimi > dump.sql” –komennolla, joka tallentaa tietokannan sisällön dump.sql-tiedostoon.

Oletuksena mysqldump lukitsee taulut ennen varmuuskopiointia varmistaakseen kaiken tiedon säilyvän eheänä ja muuttumattomana kopioinnin aikana. Yritysrekisterin tapauksessa tämä ei kuitenkaan onnistu, sillä meillä ei ole oikeuksia lukita tauluja vuokratussa tietokannassa. Ongelman voi kiertää antamalla mysqldumpin parametriksi "--lock-tables=-1". Tämä estää lukitsemisen, mutta samalla lukitsemisen hyöty menetetään. Onneksi yritysrekisterissä käytetty tietokantamoottori, InnoDB, tukee transaktioita ja mysqldump osaa hyödyntää tätä ominaisuutta. Antamalla mysqldumpin parametriksi "--single-transaction", saadaan transaktion kautta lukittujen taulujen hyöty, mutta samalla pystytään kiertämään vuokratun palvelimen käyttöoikeusrajoitukset lukitsemiselle.

Rajoitettu taulujen lukitseminen on myös huomioitava tilanteessa, jossa varmuuskopioita joudutaan joskus käyttämään. Oletuksena mysqldump lisää varmuuskopiointitiedostoon lukitsemiskäskyt ennen jokaista taulun luontia ja tietojen lisääystä. Yritysrekisterin tapauksessa tämä estäisi varmuuskopioinnin käyttöönottamisen, joten tämä ominaisuus on syytä ottaa pois päältä "--skip-add-locks"-parametrilla varmuuskopioita luodessa.

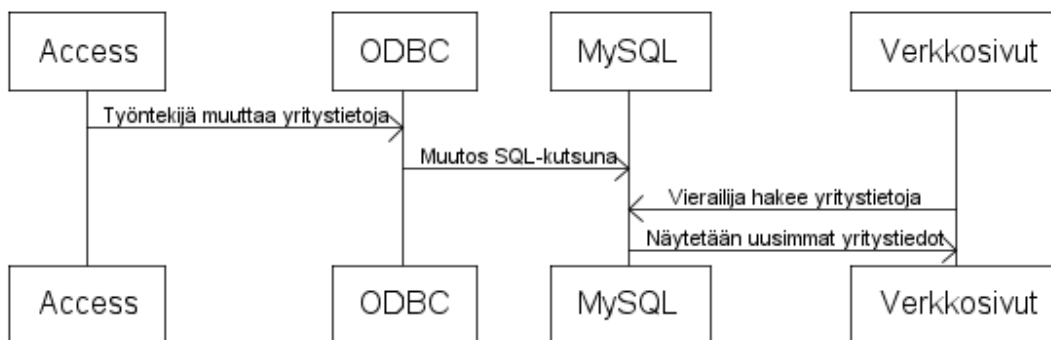
```
shell> mysqldump --password=salasana --single-transaction --skip-add-locks  
kehitysparkkiDB_rekisteri > backup.sql
```

Esimerkki 4.2. Komentokehoite yritysrekisterin tietokannan varmuuskopiointiin mysqldump-sovellusta käyttäen.

Käyttöönoton aikana virhetilat tallentavaan lokitiedostoon kiinnitettiin erityistä huomiota, mutta virheitä ei ilmaantunut ja käyttöönotto sujui muiltakin osin ongelmitta.

## 5 ACCESS

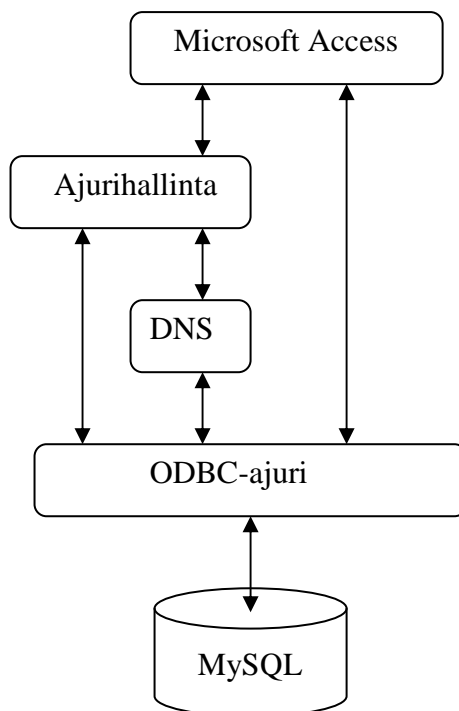
Microsoft Access kuuluu Microsoftin Office-tuoteperheeseen. Access on tietokantojen hallintaohjelma, mutta se mahdollistaa myös paikallisen tietokannan luomisen. Kehitys-Parkki Oyssa Accessia tultaisiin käyttämään uuden yritysrekisterin tietojen muokkaamiseen ja erinäisten raporttien tulostamiseen alueen yrityksistä. Yritystiedot säilytetään ulkoisessa tietokannassa, joten Access noutaa aina uusimmat yritystiedot palvelimelta työntekijän avatessa Accessin taulun.



Kuva 5.1. Yritystietojen muutosprosessi.

### 5.1 ODBC

Access käyttää normaalisti Microsoftin Jet-tietokantamoottoria. Yritysrekisterin tapauksessa MySQL-palvelin ja Kehitys-Parkki Oyn työntekijöiden käyttämä Access pitäisi saada ymmärtämään toisiaan. ODBC-rajapinta mahdollistaa tämän ottamalla vastaan Accessin tietokantakutsuja ja muuttamalla kutsut MySQL:n ymmärtämään muotoon.



Kaavio 5.1. ODBC:n arkkitehtuuri (MySQL 5.0 Reference Manual. 2009).

Käytännössä ODBC:n käyttöön ottaminen alkaa asentamalla tietokantaa tukevan ODBC-ajurin, joka yritysrekisterin tapauksessa on ”MySQL Connector/ODBC”. Tämän jälkeen ajuri on valittavissa, kun tietokoneeseen määritellään DNS-asetukset (Data Source Names). DNS-asetukset sisältävät tietokantapalvelimen osoitteen, portin, käyttäjätunnukset ja palvelimella olevan tietokannan nimen. Kehitys-Parkki Oyn tietokoneissa käytössä on Windows XP, jolloin DNS-asetukset voi määrittää ohjauspaneelin kautta ”Tietolähteet (ODBC)”-painikkeen takaa.

Aiemmin luodut asetukset ovat käytettävissä Accessissa valittaessa ulkoista tietolähdettä, kunhan tiedostotyyppiä valitsee ODBC-tietokannat. Access ottaa yhteyden tietokantaan ja listaa taulut näkyviin, jonka jälkeen ne ovat käytettävissä paikallisen tietokannan tapaan.

ODBC-arkkitehtuurissa on myös paikka ajurien hallinnalle, mutta tämä tuki löytyy yleensä valmiina käyttöjärjestelmästä ja on piilossa käyttäjiltä. Ajurihallinta toimii käytettävän ohjelman ja DNS-asetuksien välissä, selvittämällä DNS-asetuksista minikälaista ODBC-ajuria yhteyden on käytettävä.

Usein tietokantaan ei voi ottaa yhteyttä verkosta päin vaan yhteys onnistuu ainoastaan suoraan palvelimelta. SSH-yhteys on kuitenkin sallittu, joten tietokantayhteys voidaan tunneloida verkon yli SSH-yhteyttä käyttäen, jolloin tieto saadaan myös salattua. Yritysrekisterissä tähän käytetään Plink-nimistä ohjelmaa, joka on määritetty ohjaamaan paikallisen koneen portin 3306 liikenne kehitysparkin palvelimen 3306 porttiin. Tämän jälkeen on huomioitava, että myös ODBC:n DNS-asetuksissa liikenne tulee ohjata localhost-osoitteeseen, jotta se käyttäisi avattua SSH-tunnelia.

## 5.2 Käyttöliittymä

Access mahdollistaa yritystietojen muuttamisen suoraan taulujen sisältöä muokkaamalla, mutta tämä ei ole käyttäjäystävällistä ja vääränlaisen tiedon syöttämisen riski tietokantaan kasvaa. Yritysrekisterissä käytössä on lomakkeita ja raportteja. Lomakkeissa tietoa voi muuttaa, poistaa tai lisätä uutta. Raportit ovat staattisia ja ovat tarkoitettu tulosteiden luomiseen.

Yksinkertaisimmillaan lomake sisältää yhden taulun sisällön. Usein taulun kentät asetellaan näkyviin taulukkomuodossa tai sarakemuodossa, jolloin näkyvissä on ainoastaan yksi taulun tietue kerrallaan ja tietueiden välillä siirrytään sivunvaihtopainikkeilla. Yritysrekisterissä yritykset liittyvät aina tiettyyn toimialaan, jolloin vaaditaan alilomakkeiden käyttöä. Käytännössä tämä voi tarkoittaa esimerkiksi sarakemuotoisen Toimialat-lomakkeen luomista, jonka alle lisätään taulukkomuotoinen Yritys-lomake. Access osaa hyödyntää tietokannasta löytyviä taulujen välisiä yhteyksiä, jonka ansiosta teollisuus-toimialan alla näkyy ainoastaan siihen toimialaan kuuluvat yritykset.

Y-tunnus	Toiminimi	Aputoiminimi	Yritysmuoto	Käyntiosoite
1980677-6	Kuosmasen Kala		Yeh	Montunpohjankatu 6

Kuva 5.2. Toimialojen lomake. Ylimmäisenä valittu toimiala tietoineen ja alhaalla osa yritysten alilomakkeesta, joka listaa kaikki valitun toimialan yritykset.

Yritys- ja Toimiala- taulujen välillä ei kuitenkaan ole käytössä yhden suhde moneen – yhteys, vaan alilomakkeiden kannalta asiaa monimutkaistava monen suhde moneen – yhteys. Normaalisti yritys- ja toimialatietojen yhdistämisessä alilomake voisi hakea tietonsa suoraan Toimiala- taulusta, mutta monen suhde moneen –yhteyksissä haku on kohdistettava yhteyden kahteen yhden suhde moneen –yhteyteen jakavaan YritysToimiala- tauluun. Käytännössä tälle tietokantakyselylle tehdään erillinen haku Accessissa, jota yritysten alilomake käyttää tietolähteenään.

```
SELECT YritysToimiala.toimialaviite, YritysToimiala.yritysviite, Yritys.*
FROM Yritys INNER JOIN YritysToimiala ON Yritys.id=YritysToimiala.yritysviite;
```

Esimerkki 5.1. Yritys- ja YritysToimiala- taulut yhteen liittävä tietokantakysely.

Yritysrekisterissä on olemassa kuvassa 5.2 esitelty Toimiala- lomake, sekä vastaava Yritys- lomake. Yritys- lomake näyttää yhden yrityksen tiedot ja listaa alilomakkeessa kyseisen yrityksen toimialat. Lomake on luotu täysin samalla tavalla kuin Toimiala- lomake. Erilliset lomakkeet ovat tarpeellisia, sillä yrityksen liittäminen uuteen toimialaan onnistuu helposti Yritys- lomakkeessa, mutta se ei ole mahdollista Toimiala- lomakkeessa.

Käytettävissä on myös kaikki yritykset kerralla taulukkomuotoon listaava lomake. Tämä lomake ei ole helppokäyttöisin yritystietojen muuttamiseen, mutta kaikkien yritystietojen kerralla näyttäminen mahdollistaa Accessin valmiiden työkalujen käyttämisen. Yritykset on helppo laittaa järjestykseen esimerkiksi henkilöstömäärän tai



perustamispäivämäärän mukaan. Lisäksi lomakkeeseen voi suodattaa näkyviin vaikkapa tietyn toimialan tai yritysmuodon mukaiset yritykset.

Yksi Kehitys-Parkki Oyn yleinen käyttötapaus yritysrekisterin sisäiselle käytölle on osoitetarrojen tulostaminen tietyt kriteerit täyttävistä yrityksistä, joka helpottaa suurien kirjemäärien lähettämistä. Esimerkiksi yksi yleinen kriteeri on suurimpiin yrityksiin yhteyden ottaminen. Kehitys-Parkki Oyn esittämien vaatimuksien pohjalta luotiin tarratulostukseen oma käyttöliittymänsä, jonka valinnoista riippuen listataan halutut yritykset tarra-arkin muodossa olevaan raporttiin.

Kuva 5.3. Osoitteiden tulostamiseen tarkoitettu käyttöliittymä.

Tarratulostuksen pohjana toimii raportti, joka hakee yritysten osoitetiedot ja listaa ne näkyviin tarra-arkin muodossa. Yritysrekisterissä tämä tarkoittaa 3x7 yrityksen osoitetietoa sivua kohti. Access sisältää valmiiksi joidenkin valmistajien tarra-arkkien koot raporttien luomisen helpottamiseksi ja omien mittojenkin antaminen oli helppoa.

Kuvassa 5.3 näkyvän tarratulostuksen käyttöliittymän valinnat käynnistävät valinnan määräämän makron. Accessissa makrot mahdollistavat erinäisten tietokantaan tai

käyttöliittymään liittyvien toimenpiteiden suorittamisen ja niiden helpon toistamisen kutsumalla makroa toistuvasti. Tarratulostuksen makrot ovat yksinkertaisia. Ensiksi ne hakevat tarratulostuksen käyttöliittymästä käyttäjän mahdollisesti syöttämät tiedot ja tekemät valinnat. Tämän tiedon pohjalta muodostetaan tietokantahakuun sopiva kriteeri. Lopuksi makro käynnistää tarratulostuksen raportin ja antaa samalla hakuparametrina aiemmin luodun kriteerin. Käyttäjälle näytetään ainoastaan pyydettyt yritykset ja raportti on valmis tulostettavaksi.

Yritysrekisterin päivittäinen käyttö rajoittuu yritystietojen muuttamiseen. Joskus kuitenkin saattaa tulla tilanne, jossa halutaan muuttaa toimialojen kuvauksia tai poistaa ja lisätä toimialoja. Toimiala-aulun muokkaukseen luotiin oma lomakkeensa, jonka avaaminen tapahtuu harvinaisuutensa vuoksi ylläpito-valikon kautta.

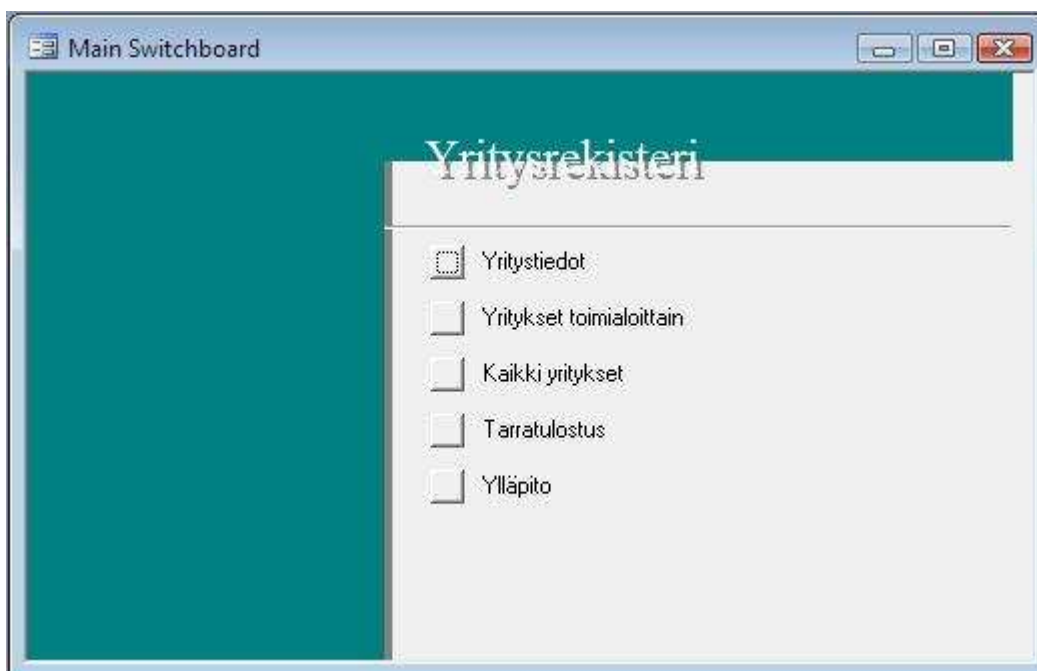
The screenshot shows a window titled "Ylläpito : Form" with a standard Windows-style title bar. Below the title bar, there is a text box containing the instruction: "Lisätietoa mahdollisista ylläpitotoimenpiteistä löytyy yritysrekisterin käyttöohjeista." Below this, the window is divided into two columns. The left column is titled "Toimialat" and contains two bullet points: "- Toimialan kuvauksen muuttaminen" and "- Toimialan lisääminen tai poistaminen". Below these points is a button labeled "Muokkaa...". The right column is titled "Toimialattomat yritykset" and contains a bullet point: "- Ilman toimialaa olevat yritykset eivät ole löydettävissä verkkosivujen toimialahakua käyttämällä". Below this point is a button labeled "Yritykset...".

Kuva 5.4. Ylläpito-lomake toimintoinen.

Toimialoja poistaessa on myös huomioitava mitä tapahtuu niihin liitettyillä yrityksillä. Yritysrekisterissä tietokannan taulujen väliset yhteydet on asetettu niin, että toimialan poistettaessa siihen kuuluneiden yritysten toimialaviittaus häviää. Mikäli yritys ei kuulu muihin toimialoihin, niin se jää ilman liitettyjä toimialoja eikä ole enää löydettävissä verkkosivujen toimialahaun kautta. Tällaisten yritysten löytämisen helpottamiseksi luotiin oma HylätytYritykset-lomake, johon löytyy painike ylläpito-valikosta. HylätytYritykset-lomake on täsmälleen sama lomake kuin aiemmin esitelty Yritykset-lomake, mutta se rajoittaa tietokantahakunsa ainoastaan yrityksiin joiden tunnuksen ei viitata YritysToimiala-aulussa.

Käynnistyessään Access näyttää valikon, jossa ovat mm. käytettävissä olevat lomakkeet, raportit ja makrot. Kehitys-Parkki Oyn työntekijä voisi käynnistää tästä valikosta haluamansa lomakkeen, mutta valikko on sekava sillä se sisältää myös kaikki muiden lomakkeiden käyttämät alilomakkeet, joita ei ole tarkoitus käynnistää yksinään. Selkeä päävalikko olisi huomattavasti parempi vaihtoehto.

Accessin työkaluvalikossa on päävalikon luomiseksi oma työkalunsa. Päävalikkoon lisätään otsikko ja valitaan halutut lomakkeet, sekä annetaan niille kuvaava nimi. Työkalu luo lomakkeille napit ja päävalikko on valmis. Se on muokattavissa muiden lomakkeiden tavoin switchboard-nimellä lomake-valikossa, jolloin ulkoasun voi hioa yritykselle sopivaksi. Valmis päävalikko aukeaa automaattisesti joka kerta, kun Access käynnistetään.



Kuva 5.5. Yritysrekisterin päävalikko.

## 6 VALMIIN PROJEKTIN ARVIOINTI

Projektin alussa tavoitteeksi asetettiin yritystietojen muuttamisen helpottaminen ja tämä tavoite saavutettiin. Muuttamalla kerran yritystietoja Accessissa saadaan sama muutos ohjattua automaattisesti myös kaikkiin muihin yritysrekisterin osiin.

Projektin aikana tehdyt tekniset valinnat osoittautuivat järkeviksi, eikä suurempia ongelmia ilmaantunut. MySQL:lle löytyi hyvät työkalut, jotta vanhan yritysrekisterin tiedot saatiin siirrettyä uuteen tietokantaan. Uudessa järjestelmässä muutokset saatiin siirrettyä Accessista tietokantaan MySQL:n ODBC-ajurilla. Lisäksi MySQL on niin yleisesti käytetty, että Kehitys-Parkki Oyn verkkosivuja ylläpitävä yritys tarjoaa myös MySQL:n käyttämismahdollisuuden palvelimellaan.

PHP:lla tietokannan tietojen käsittely oli vaivatonta ja PHP:n tarjoama dokumentointi toiminnoistaan on kattava. Projektin aikana ei törmätty ongelmiin PHP:n kanssa, joita ei pystynyt ratkaisemaan verkosta dokumentointia tai muiden ohjelmoijien kokemuksia ja ratkaisuja etsimällä.

Access tarjoaa kehittäjälle mahdollisuuden saada nopeasti aikaan käyttöliittymä tietokannan käsittelyyn. Työntekijän muokatessa lomakkeiden kautta saadaan suoraa taulun muokkausta selkeämpi käyttökokemus ja myös annettujen syötteiden tarkistus on mahdollista virheiden karsimiseksi.

Accessin heikkoudet ovat samat kuin missä tahansa muussa graafisen käyttöliittymän luomisen helpottamiseen tarkoitetussa työkalussa. Kun siirrytään liiaksi järjestelmäkohtaisten yksityiskohtien hiomiseen, joita Accessin kehittäjät eivät ole voineet ennakoida, nousee tarvittava työpanos äkisti seuraavalle tasolle. Tälläisiin tapauksiin Accessissa voidaan käyttää Visual Basic –ohjelmointikielen Visual Basic for Applications –nimistä toteutusta. Yritysrekisterissä tätä pyrittiin välttämään ja järjestelmä saatiin toteutettua hyvin pienellä määrällä ylimääräistä Visual Basic –koodia.

Projektin aikataulussa pysyttiin riittävästi. Uudet verkkosivut otettiin virallisesti käyttöön elokuun lopulla ja tuolloin yritystietojen muuttaminen hoidettiin jo Accessin kautta. Tämän jälkeen lisättiin vielä tärkeä tarratulostus-ominaisuus. Lisäksi järjestelmän käytöstä ilmenneitä parannusehdotuksia toteutettiin hiljalleen koko talven ajan, mutta nämä eivät enää vaatineet keneltäkään suurta työpanosta.

## 6.1 Kehitysideoita

Projektin tavoitteena ollut työtaakan vähentämistä voisi tulevaisuudessa parantaa lisääkin. Vaikka yritystietojen muutosprosessista saatiin huomattavasti tehokkaampi, niin tietojen päivittäminen vaatii edelleen Kehitys-Parkki Oyn työntekijän aikaa.

Yhtenä parannusehdotuksena mietittiin yritystietojen muuttamiseen tarkoitettua verkkosivua, johon jaettaisiin käyttäjätunnuksia yrityksille. Tällä hetkellä yritystietojen muutos tapahtuu verkkosivuilta löytyvällä lomakkeella, jonne yrittäjä syöttää muuttuneet tietonsa ja lomake lähettää tiedot sähköpostina Kehitys-Parkki Oyn työntekijälle. Sähköpostin saatuaan työntekijä syöttää muutokset Accessin kautta tietokantaan. Antamalla yrittäjälle oikeudet tietojensa muuttamiseen voitaisiin verkkosivuilla oleva lomake siirtää salasanan taakse ja lomakkeeseen syötetyt muutokset päivittyisivät suoraan tietokantaan.

Yllä olevassa ratkaisussa yritysrekisterin verkkosivut siirtyisivät tietojen esittämisestä myös tietojen vastaanottamiseen, joten tietoturvaan ja syötteiden sisällön tarkistamiseen pitäisi kiinnittää enemmän huomiota. Samalla tarvittaisiin myös tapa varmistaa kyseessä olevan oikea henkilö jonkun pyytäessä käyttäjätunnuksia tietyn yrityksen tietojen muuttamiseen. Tästä pitäisi saada riittävän yksinkertainen, että Kehitys-Parkki Oyn työntekijän työaika ei vain siirry yritystietojen muuttamisesta käyttäjätunnuksien hallinnointiin ja käyttäjien koulutukseen.

Yritykset ilmoittavat tietonsa patentti- ja rekisterihallituksen ja verohallinnon yhteiseen yritystietojärjestelmään (YTJ). Järjestelmässä ovat kaikki yritykset ja yhteisöt, jotka on merkitty:

- kaupparekisteriin

- säätiörekisteriin
- arvonlisäverovelvollisten rekisteriin
- ennakkoperintärekisteriin
- työnantajarekisteriin tai
- verohallinnon asiakasrekisteriin

Yritysrekisterin kaikki tiedot voitaisiin ohjelmallisesti päivittää esimerkiksi puolivuositain YTJ:n tietojen pohjalta. Käytännössä tämä tapahtuisi sovittamalla Parkanon ja Kihniön alueen yritykset y-tunnuksen perusteella toisiinsa YTJ:n ja yritysrekisterin välillä. Näin saataisiin pidettyä ajantasalla myös yritykset joiden yrittäjä ei syystä taikka toisesta ole päivittänyt tietojaan yritysrekisteriin tai yritykset, jotka ovat jo lopettaneet toimintansa.

Säännöllisesti YTJ:n tietoja käyttämällä säästettäisiin sekä yrittäjien, että Kehitys-Parkki Oyn työntekijöiden aikaa, sillä yritystietoja ei tarvitsisi enää olla niin aktiivisesti lähettämässä ja vastaavasti muuttamassa. Valitettavasti YTJ on kieltänyt sivuillaan hakurobottien käytön yritystietojen hakemiseen ja tätä sääntöä ylläpidetään IP-kohtaisilla hakurajoituksilla. Vastineeksi YTJ tarjoaa yhteistyökumppaniensa kautta maksullisia hakupalveluja, mutta näiden hintatasoa ei projektin aikana alettu selvittää.

Ennen yllä olevien parannusehdotusten toteuttamista on syytä miettiä kuinka suuriin säästöihin niillä todellisuudessa päästäisiin. Tästä saadaan kuva työntekijöiden kokemuksien perusteella. Kuluuko heidän mielestään yritystietojen muuttamiseen huomattava määrä aikaa vaikka käytössä on uusi yritysrekisteri vai onko tarvittava työn määrä vähentynyt riittävästi.

Toinen kehityksen kohde voisi olla yritysrekisterin verkkosivujen parantaminen. Sivujen rakenne on käytännöllinen ja selkeä, eikä sitä juurikaan muutettu projektin aikana. Kehitys-Parkki Oyn sivut ovat kuitenkin muilta osin uudistuneet ja yritysrekisterin ulkopuoliset sivut näyttävät moderneilta, joten yritysrekisterinkin ulkoasua voisi yhtenäistää paremmin muita sivuja vastaavaksi. Uusiksi kirjoitetut yritysrekisterin sivut tarjoavat ulkoasun kohentamiseen tiettyyn pisteeseen saakka helpotusta, sillä sivujen ulkoasuun vaikuttavat valinnat on kerätty erilleen sivujen rakenteesta CSS-

tiedostoihin. CSS-tiedostot mahdollistavat helposti sivujen tyylin muokkauksen. Suuremmat rakenteelliset ja tiedon hakuun liittyvät muutokset kuitenkin vaatisivat PHP-koodin muokkausta.

## LÄHTEET

MySQL Migration Toolkit documentation. 2009. Preparing a Microsoft Access Database for Migration. Viitattu 15.7.2008. <http://dev.mysql.com/doc/migration-toolkit/en/mysql-migration-toolkit-accessprep.html>

MySQL 5.0 Reference Manual. 2009. Mysqlexport — A Data Import Program. Viitattu 15.7.2008. <http://dev.mysql.com/doc/refman/5.0/en/mysqlexport.html>

Hillyer, M. Managing Hierarchical Data in MySQL. Viitattu 1.7.2008. <http://dev.mysql.com/tech-resources/articles/hierarchical-data.html>

Tilastokeskus. 2008. Toimialaluokitus 2008. Viitattu 9.11.2009. <http://www.stat.fi/meta/luokitukset/toimiala/001-2008/index.html>

MySQL 5.0 Reference Manual. 2009. Mysqldump — A Database Backup Program. Viitattu 9.11.2009. [http://dev.mysql.com/doc/refman/5.0/en/mysqldump.html#option\\_mysqldump\\_single-transaction](http://dev.mysql.com/doc/refman/5.0/en/mysqldump.html#option_mysqldump_single-transaction)

MySQL 5.1 Reference Manual. 2009. The InnoDB Storage Engine. Viitattu 1.7.2008. <http://dev.mysql.com/doc/refman/5.1/en/innodb.html>

MySQL 5.0 Reference Manual. 2009. General Information About ODBC and Connector/ODBC. Viitattu 9.11.2009. <http://dev.mysql.com/doc/refman/5.0/en/connector-odbc-general-information.html>

Patentti- ja rekisterihallituksen ja verohallinnon yhteinen yritystietojärjestelmä. Viitattu 9.11.2009. <http://www.ytj.fi/>

Uuden yritysrekisterin verkkosivut. Viitattu 9.11.2009. <http://www.kehitysparkki.fi/yritykset.html>