

Opinnäytetyö (AMK)  
Tietojenkäsittelyn koulutusohjelma  
Sähköisen liiketoiminnan järjestelmät  
2013

Markku Siipola

# OPETUKSEN SUUNNITTELUTYÖKALUN SUUNNITTELU JA TOTEUTUS KIELIKESKUKSELLE



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittelyn koulutusohjelma | Sähköisen liiketoiminnan järjestelmät

Maaliskuu 2013 | Sivumäärä: 34

Ohjaaja: Päivi Killström

Markku Siipola

# OPETUKSEN SUUNNITTELUTYÖKALUN SUUNNITTELU JA TOTEUTUS KIELIKOULUTUSKESKUKSELLE

Tämän opinnäytetyön tavoitteena oli suunnittelutyökalun määrittely, suunnittelu ja toteutus Turun Ammattikorkeakoulun kielikoulutuskeskukselle

Kielikoulutuskeskus toteuttaa kieli- ja viestintäkoulutusta Turun Ammattikorkeakoululle ja julkishallinnon yhteisöille. Opetuksen suunnittelutyökalun tarkoituksena on helpottaa opintojaksojen suunnittelua ja seuranta. Suunnittelutyökalun sovelluskehittimenä on käytetty Oracle Application Express ohjelmaa. Sovellus koostuu interaktiivisista raporteista, lomakkeista ja sovelluksen kehitystyö on pyritty tekemään Application Express sovelluksen omilla toiminnoilla, että myös sovelluksen jatkokehitys olisi mahdollista ilman suurempia ohjelmointitaitoja.

Opinnäytetyössä kuvataan sovelluksen määrittely- ja toteutusvaiheet sekä esitellään lyhyesti sovelluskehittimenä käytetyn Application Express toimintoja.

Opetuksen suunnittelutyökalun toteutus on vaatimusmäärittelyn mukainen ja sovellus toimii vaaditulla tavalla. Kielikoulutuskeskus testasi järjestelmää kehitysympäristössä oikealla datalla ja palaute oli positiivista, joten asennusvaiheessa voitiin siirtää valmis järjestelmä käytettäväksi.

ASIASANAT:

HTML, Oracle Application Express, PL/SQL, Apex, Kielikoulutuskeskus

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme in Business Information Technology | e-Business System

March 2013 | Total number of pages: 34

Instructor: Päivi Killström

Markku Siipola

## DESIGN AND IMPLEMENTATION OF COURSE PLANNING TOOL FOR THE LANGUAGE CENTER

The purpose of this project was to define, design and implement an education planning tool for the language center of the Turku University of Applied Science.

The language center operates language and communication studies for the degree programs and offers language education for their corporate customers. The purpose of the planning tool is to simplify course planning and follow-ups. The planning tool is implemented with Oracle Application Express web application development tool. The planning tool consist of interactive reports, forms and the development is made by using the applications own functions so that further development would be possible to do even with minor programming skills.

The thesis describes defines and implements the planning tool and briefly presents the functions of the web application development tool.

The course planning tool is as it is defined in the requirement specification and it will perform as intended. The language center tested the planning tool in the development environment with real data and feedback was positive so the complete application installation could be made.

### KEYWORDS:

HTML, Oracle Application Express, PL/SQL, Apex, Language center

# SISÄLTÖ

<b>1. JOHDANTO</b>	<b>6</b>
<b>2. OHJELMISTOTUOTANNON VAIHEJAKOMALLI</b>	<b>7</b>
<b>3. ORACLE APPLICATION EXPRESS TYÖVÄLINE</b>	<b>9</b>
2.1 Sovelluskehittimen osat	9
2.2 Sivunmääritykset	10
2.3 Tietoturvallisuus	11
2.4 Raportointi	13
<b>4. SUUNNITELUTYÖKALUN VAATIMUSMÄÄRITTELY</b>	<b>16</b>
4.1 Toiminnallisuuksien määrittely	16
4.2 Käyttäjien roolit	18
<b>5. TYÖKALUN SUUNNITTELU</b>	<b>20</b>
5.1 Tietokantataulut	20
5.2 Näkymät	21
5.3 Navigointi	22
<b>6. SUUNNITELUTYÖKALUN TOTEUTUS</b>	<b>25</b>
6.1 Todennus	25
6.2 Suunnittelu - ja tulosaluenäkymät	25
6.3 Raportointi ja ylläpito	28
<b>7. SUUNNITELUTYÖKALUN TESTAUS</b>	<b>29</b>
<b>8. KÄYTTÖÖNOTTO</b>	<b>32</b>
<b>9. ARVIOINTI JA JATKOKEHITYKSEN IDEOINTI</b>	<b>33</b>
<b>LÄHTEET</b>	<b>34</b>

## KUVAT

Kuva 1. Feature Drive Development malli.	7
Kuva 2. Sovelluskehittimen osat.	9
Kuva 3. Sivuston määrittäminen.	10
Kuva 4. Oracle Application Express ympäristö.	12
Kuva 5. Perusraportti.	13
Kuva 6. Interaktiivinen raportti.	14
Kuva 7. Sovelluksen Home sivu.	23
Kuva 8. Arvoluettelon kysely.	26
Kuva 9. Suunnittelulomakkeen hakutoiminto.	27
Kuva 10. Sivun prosessit suunnittelunäkymässä.	30
Kuva 11. Javascript sivun Header osiossa.	31
Kuva 12. Javascript lomakkeen opintopisteet Column Definition osiossa.	31

## KUVIOT

Kuvio 1. Sovelluksen käyttötapauskaavio.	18
Kuvio 2. Tietokannan rakenne.	20
Kuvio 3. Tietokannan näkymät.	22
Kuvio 4. Suunnittelutyökalun sivustorakenne.	23

## TAULUKOT

Taulukko 1. Tietokantataulujen sisältöjen kuvaukset.	21
--	----

# 1. JOHDANTO

Turun Ammattikorkeakoulun kielikoulutuskeskus toteuttaa ammattikorkeakoulun tutkintoon johtavaa viestintä- ja kielikoulutusta, sekä tarjoaa koulutusta yrityksille ja julkishallinnon yhteisöille. Kielikoulutuskeskuksen tavoitteena on ollut saada käyttöön selainpohjainen suunnittelu-sovellus, missä koulutusta koskevat tiedot talletettaisiin tietokantaan ja samalla sovellus toimisi myös raportointijärjestelmänä kurssien suunnittelua ja opettajien resurssien seuranta varten.

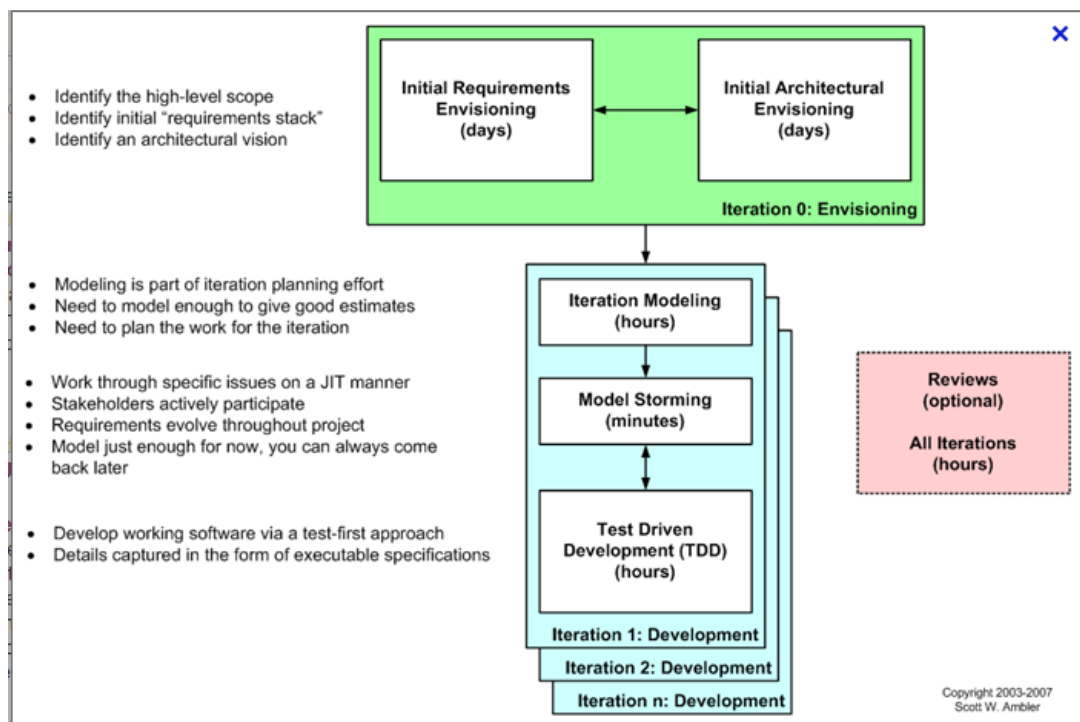
Kielikoulutuskeskuksen kurssien suunnittelu on toteutettu Microsoft Office Excel taulukkolaskentaohjelmalla, minkä on todettu olevan riittämätön kurssien suunnitteluun ja raportointiin. HTML-pohjaisen sovelluksen tulisi sijaita ammattikorkeakoulun lähiverkossa, jolloin järjestelmän käyttö olisi vaivatonta ja turvallista.

Tehtävänäni oli suunnitella ja toteuttaa vaatimusmäärittelyn mukainen suunnittelutyökalu. Opinnäytetyö kuvaa projektin eri vaiheet sovelluksen suunnittelusta sen käyttöönottoon. Projekti toteutettiin Oracle Application Express sovelluskehittimellä, mikä sisältää valmiit raportointitoiminnot. Application Express on helppokäyttöinen sovelluskehitin, jolla pystytään helposti luomaan tehokkaita sovelluksia vähäisilläkin ohjelmointitaidoilla. Sovelluskehittimessä käytettävät ohjelmointikieliset ovat PL/SQL ja Javascript.

Opetuksen suunnittelutyökalussa alkavien kurssien tiedot syötetään järjestelmään joko kopioimalla olemassa oleva kurssi tai tuomalla tiedot tekstitiedostosta suunnittelua varten. Tietojen syöttämisen jälkeen opettajien resursseja voidaan helposti seurata ja muokata suunnittelunäkymässä, jos siihen on tarvetta. Alkavista kursseista voidaan myös tehdä kurssiehdotuksia, jotka hyväksytään tai hylätään kielikoulutuskeskuksen toimesta. Suunniteltavien kurssien ja toteutuneiden kurssien seuranta varten järjestelmässä on valmiita raportteja, joilla voidaan seurata opettajien resursseja tehokkaasti.

## 2. OHJELMISTOTUOTANNON VAIHEJAKOMALLI

Ohjelmistoprojektista kehitystyö voi olla varsin haasteellinen toteuttaa ja dokumentoida. Projektin onnistumisen varmistamiseksi kehitystyö jaetaan osiin, joissa suunnittelulle ja toteutukselle on omat vaiheet ja tätä ositusta kutsutaan vaihejakomalliksi. Vaihejakomalleja on useita erilaisia ja tässä luvussa käsiteltävä ketterät menetelmät ovat yksi niistä. Vuonna 2001 luodussa ketterien menetelmien Agile Manifestossa on 12 periaatetta, mitkä määrittelevät toimintatavat ohjelmistoprojekteille, joissa käytetään ketteriä menetelmiä. Ketterien menetelmien päätarkoituksena on tehdä ohjelmistoprojekteista yksinkertaisia ja tehokkaita, minkä avulla asiakastyytyväisyys pysytään varmistamaan. Projektit toteutetaan lyhyissä iteraatioissa, jolloin asiakkaalle pystytään toimittamaan uusin versio sovelluksesta mahdollisimman lyhyessä ajassa testattavaksi. Periaatteena ketterissä menetelmissä on, että asiakkaalle toimitettavan version kehitys kestäisi yhdestä neljään viikkoa. Jatkuva vuorovaikutus asiakkaan ja kehitystii-  
mien välillä takaa projektin onnistumisen sekä tehokkaamman reagoimisen muutostarpeisiin. (Agile Manifesto 2001)



Kuva 1: Agile Modeling Driven Development (Scott Ambler 2007, AMDD)

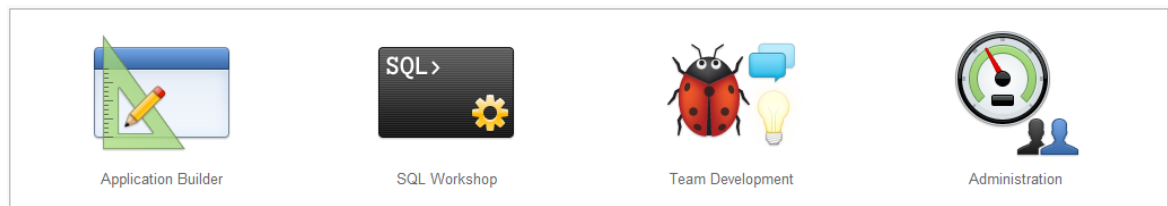
Kuvassa 1 on esitelty AMDD - malli, mikä kuvaa ketterien menetelmien eri vaihteita. Vihreällä pohjalla oleva osio on menetelmien ylätasoa, missä toteutetaan ohjelmiston vaatimusmäärittely ja ohjelman rakenne. Alla olevissa sinisellä pohjalla olevat vaiheet kuvaavat iteraatiivisia vaihteita, mihin kuuluu suunnittelu (Iteration Modelling), palautteiden ja kehitysideoiden kokoaminen (Model Storming) ja kehitystyö (Test Driven Development). Elinkaarimallin tarkoituksena on ominaisuuslistojen luominen ja ominaisuuksien toteuttaminen jokaisessa iteraatiovaiheessa. Projektissa voi olla useita iteraatioita riippuen ohjelman toiminnallisuuden määrästä. Tällä tavalla toteutettuna kehitystyöstä saadaan tehokas, ja kun edellinen vaihe on valmis, voidaan aloittaa uusi iteraatio toiminnon lisäämiseksi sovellukseen. Elinkaarimallin avulla myös dokumentointi pystytään tekemään tarkemmalla tasolla projektin aikana. Elinkaarimallin hyvänä puolena on se, että myös uusien toimintojen lisääminen iteraatioon on mahdollista kehitystyön aikana.



### 3. ORACLE APPLICATION EXPRESS TYÖVÄLINE

Oracle Application Express, lyhyesti kutsuttuna Apex, on Oraclen ilmaisen tietokantatuotteen mukana tuleva sovelluskehitin, jolla pystytään luomaan tehokkaita ja turvallisia sovelluksia joiden käyttöliittymä on selainpohjainen. Sovelluskehittimellä pystytään luomaan perinteisiä html-sivustoja tai tietokantaan integroitua html-sivustoja, jollainen myös tuntien suunnitteluohjelma on. Kehitysympäristönä tälle sovellukselle on käytetty Oraclen Internet sivustolla sijaitsevaa Oracle Apex harjoitusympäristöä. Oraclen sivustolta käyttäjät voivat anoa käyttöönsä työtilan, jossa sovelluskehittimen käyttöä pystyy harjoittelemaan. Myös järjestelmän kehittämiseen ja ylläpitoon liittyvät dokumentit löytyvät kyseiseltä sivustolta (Oracle Application Express Overview).

#### 2.1 Sovelluskehittimen osat



Kuva 2. Sovelluskehittimen osat.

Kuvassa 2 on esitelty Application Express sovelluskehittimen työtilan (Workspace) sisältämät osiot kehitystyön eri vaiheille. Varsinainen kehitystyö toteutetaan Application Builder osiossa, missä kehittäjä luo varsinaisen sovelluksen. Application Builder työkalulla kehittäjän pystyy luomaan joko tavallisen HTML-sivuston tai tietokantaan integroidun HTML-sivuston. Application Builder sisältää valmiita esimerkkisovelluksia, joita kehittäjä voi asentaa käyttöönsä.

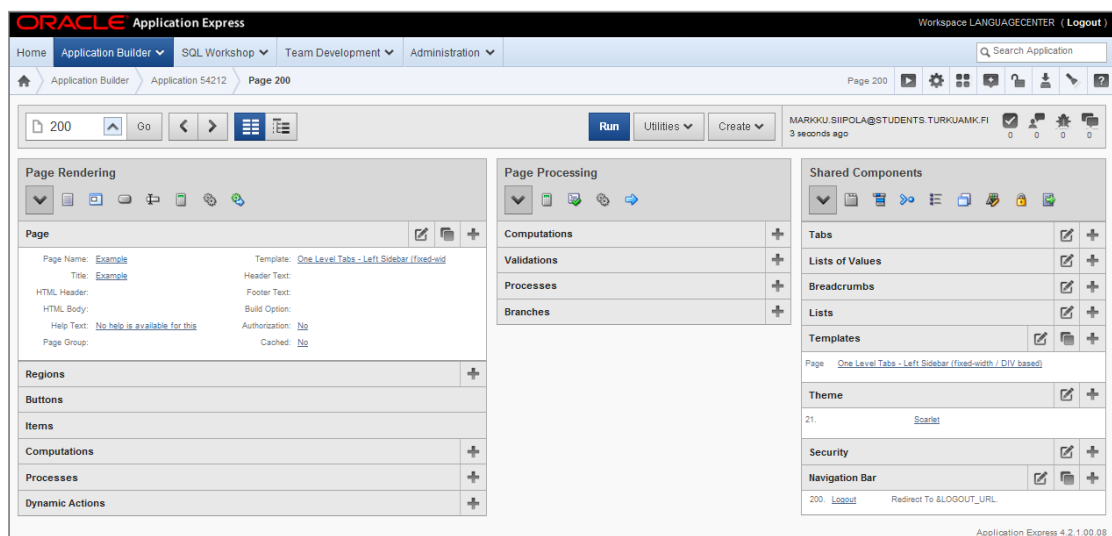
SQL Workshop osio on tarkoitettu tietokantaan liittyvien toimintojen hallintaan ja ylläpitoon. SQL Workshop osiossa kehittäjä voi esimerkiksi luoda uusia tietokantaobjekteja, muokata olemassa olevia objekteja sekä luoda kyselyjä ohjatulla toiminnolla.

Team Development osiossa kehittäjä pystyy luomaan erilaisia tehtäviä sovel-  
luskehityksen hallintaa. Team Development osio sisältää toiminnot projektin  
aikataulutukselle, tehtävälidoille sekä palautteen keräämiselle.

Administration osio on tarkoitettu Application Express järjestelmä ylläpitoa var-  
ten. Administration osiossa ylläpidetään järjestelmän käyttäjät ja käyttäjäryhmät  
sekä pystytään seuramaan sivustojen käyttöön liittyviä raportteja.

## 2.2 Sivunmääritykset

Kun sovellus luodaan Application Builder:lla, luodaan varsinaisesti sivuja, jotka  
sisältävät käyttöliittymän elementtejä kuten listoja, painikkeita ja sivun osioita.  
Näitä käyttöliittymän elementtejä kehittäjä pystyy muokkaamaan sivuston mää-  
rittelynäkyssä (Page definition). Määrittelynäky on jaettu kolmeen osaan:  
Page Rendering, Page Processing ja Shared Components.



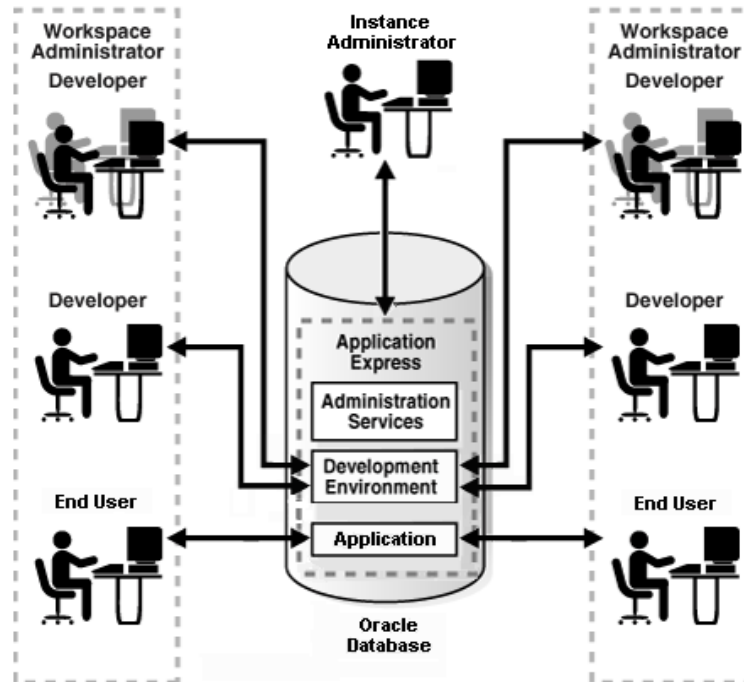
Kuva 3: Sivuston määrittelynäky

Kuvassa 3 on esitelty Page Rendering osio, mikä sisältää sivulla olevat käyttö-  
liittymäelementit. Page Rendering on prosessi, jolla elementit ladataan tieto-  
kannasta, kun sivu ladataan selaimeen. Page Processing osio suoritetaan sivun  
latauksen jälkeen järjestyksessä ylhäältä alaspäin. Osiossa suoritetaan sivulle

määritetyt toiminnot, jotka voivat olla laskutoimituksia, tiedon tarkistamista, tiedon käsittelyä tai toiselle sivulle siirtyminen. Shared Components osiossa on yleisessä käytössä toimintoja ja elementtejä. Osio sisältää sivun navigaatioelementin, sivulla käytettävän mallin ja teeman.

## 2.3 Käyttäjien tunnistus ja autorisointi

Oracle Apex sovelluskehittimellä luotujen sovellusten käyttäjätunnistus on mahdollista toteuttaa useammalla tavalla. Sovellukseen kuuluu valmiina erilaisia autentikointiin liittyviä skeemoja ja käyttäjärooleja, joita kehittäjät voivat hyödyntää. Myös omien käyttäjätunnistusmenetelmien luonti onnistuu helposti, jos valmiita skeemoja ei haluta hyödyntää järjestelmässä. Omien käyttäjätunnistusmenetelmien luonti järjestelmään toteutetaan PL/SQL-ohjelmakoodilla ja näiden tekemiseen löytyy paljon hyviä esimerkkejä esimerkiksi Google-hakukoneen avulla. Järjestelmässä tullaan käyttämään sovelluksen omaa käyttäjätunnistusmenetelmää, jolloin jokainen käyttäjä pitää lisätä erikseen järjestelmään. Käyttäjätunnistuksessa on myös mahdollista hyödyntää Windows-verkossa käytettävää autentikointia, jolloin käyttäjät voivat kirjautua järjestelmään samoilla tunnuksilla ja salasanoilla kuin mitä verkkoon kirjautumisessa käytetään. Järjestelmässä on valmiina neljä erilaista käyttäjäryhmää valmiina, Workspace Administrator, Developer ja End User. Instance Administrator ryhmään kuuluvat käyttäjät voivat yllä pitää kaikkia järjestelmän toimintoja Application Express Administration Services ohjelmalla.



Kuva 4: Oracle Application Express ympäristö (Oracle 2 days + Developer guide, 1-5, 2013)

Kuvassa 4 on esitelty sovelluksen ympäristö ja käyttöoikeustasot. Workspace Administrator ryhmään kuuluvilla jäsenillä on oikeudet tiettyyn työtilaan, missä he voivat ylläpitää käyttäjätietoja ja hyödyntää työtilan käyttöön liittyviä lokitietoja. Developers ryhmään kuuluvat käyttäjät voivat luoda tai muokata työtiloja ja muokata järjestelmässä olevia tietokantaobjekteja. End Users ryhmään kuuluvat jäsenet voivat ainoastaan käyttää työtiloja. (Oracle 2 days + Developer guide, 2013)

Apex-sovelluksessa olevien tietojen näkyvyyttä pystytään rajamaan tehokkaasti erilaisten käyttäjäryhmille määritettyjen oikeuksien avulla. Html-sivustot tai sivustoilla olevien objektien näkyvyydet pystytään rajamaan juuri näiden käyttäjäryhmien perusteella ja näin pystytään varmistamaan, että ainoastaan tiettyihin käyttäjäryhmiin kuuluvat jäsenet pystyvät näkemään tai poistamaan tietoa järjestelmästä.

## 2.4 Raportointi

Apex sovellukseen sisältyy monipuolinen raportointityökalu, jolla kehittäjä pystyy helposti luomaan näyttäviä raportteja erilaisiin käyttötarpeisiin. Raportointityökalussa on mahdollista luoda joko perusraportteja tai interaktiivisia raportteja, joiden ominaisuudet eroavat aikailla toisistaan. Raporteilla olevia ominaisuuksia ei ole pakko ottaa heti käyttöön, vaan ne voidaan lisätä jälkikäteen, jos tulee tarvetta niiden käytölle. Alla on kuvaukset raporttien ominaisuuksista.

### Perusraportti

Apex sovelluksen Classic report on perusraportti, johon sisältyy vain muutamia toiminnallisia ominaisuuksia. Raportille voidaan sisällyttää raportilla olevien tietojen lajittelu, tulostaminen PDF - formaatissa, tietojen vienti .csv - formaatissa sekä lisätä hakutoiminto, jolla saadaan raportille saadaan halutut tiedot näkyviin. Perusraportin kysely on myös mahdollista toteuttaa PL/SQL- kyselyllä, minkä avulla raportille haettava tieto voidaan rajata järjestelmässä olevien käyttäjäryhmien tai haluttujen tietojen perusteella. PL/SQL kyselyllä toteutetun raportin tiedot voidaan rajata siten, että esimerkiksi koulutuspäälliköt näkevät kaikki vastuulleen kuuluvat koulutukset ja niiden resurssit mutta opettajat näkevät ainoastaan omat kurssit ja niiden resurssit.

Cust First Name ▼	Cust Last Name	Cust Street Address1	Cust City	Cust State	Cust Postal Code	Phone Number 1
Albert	Lambert	10701 Lambert International Blvd.	St. Louis	MO	63145	314-555-4022
Edward	Logan	1 Harborside Drive	East Boston	MA	02128	617-555-3295
Edward "Butch"	OHare	10000 West OHare	Chicago	IL	60666	773-555-7693
Eugene	Bradley	Schoephoester Road	Windsor Locks	CT	06096	860-555-1835
Fiorello	LaGuardia	Hangar Center	Flushing	NY	11371	212-555-3923
John	Dulles	45020 Aviation Drive	Sterling	VA	20166	703-555-2143
William	Hartsfield	6000 North Terminal Parkway	Atlanta	GA	30320	404-555-3285

[Spread Sheet](#)

Kuva 5. Perusraportti

Kuvassa 5 on esitelty sovelluksen perusraportti. Perusraportin toiminnot ovat rajalliset, koska raportilla voidaan hyödyntää pelkästään yhtä hakuehtoa, eikä käyttäjä pysty muokkaamaan valmiita raportteja lainkaan.

### Interaktiivinen raportti

Apex sovelluksen Interactive report on kehittyneempi raporttipohja, johon myös loppukäyttäjä pystyy tekemään omia muokkauksia. Raportilla ei ole mahdollista hyödyntää PL/SQL-kyselyä kuten perusraportilla pystyy tekemään, mutta vastaavasti interaktiivinen raportti sisältää monipuoliset työkalut raportin tietojen hakemiseen, esittämiseen ja muokkaamiseen. Raportilla loppukäyttäjä pystyy esimerkiksi valitsemaan haluamansa tietosarakkeet raportille ja määrittelemään näytettävien raporttirivien määrät. Interaktiiviselle raportille järjestelmän ylläpitäjä pystyy tekemään erilaisia raporttipohjia, joita loppukäyttäjät voivat helposti hyödyntää tietoa haettaessa. Myös loppukäyttäjä pystyy tallettamaan henkilökohtaisia raporttipohjia järjestelmään, jolloin tiedon saanti järjestelmästä on nopeaa ja vaivatonta. Myös erilaisten kaavioiden lisääminen raportille on mahdollista, jolloin raportin sisältö saadaan visuaaliseksi. Kaavioissa on myös monipuoliset funktiot, joiden avulla raportilla olevat tiedot pystytään ryhmittelemään käyttäjän haluamalla tavalla. Interaktiivisella raportilla on myös erinomaiset toiminnot tiedon siirtämiseen järjestelmästä.



The screenshot shows an interactive report interface. At the top, there is a search bar with a magnifying glass icon, a 'Go' button, and an 'Actions' dropdown menu. Below the search bar, there is a filter section with a funnel icon and the text 'Department Id != 60', followed by a green checkmark and a red 'X' icon. The main part of the interface is a table with the following columns: Department Id, Department Name, Number Of Employees, Manager Name, and Location. Each row in the table has a yellow pencil icon in the first column, indicating that the data can be edited. The table contains 10 rows of data.

Department Id	Department Name	Number Of Employees	Manager Name	Location
90	Executive	3	S.King	United States of America
100	Finance	6	N.Greenberg	United States of America
30	Purchasing	6	D.Raphaely	United States of America
50	Shipping	45	A.Fripp	United States of America
80	Sales	34	J.Russell	United Kingdom
10	Administration	1	J.Whalen	United States of America
20	Marketing	2	M.Hartstein	Canada
40	Human Resources	1	S.Mavris	United Kingdom
70	Public Relations	1	H.Baer	Germany
110	Accounting	2	S.Higgins	United States of America

1 - 10

Kuva 6. Interaktiivinen raportti

Kuvassa 6 on esitelty sovelluksen interaktiivinen raportti. Käyttäjä pystyy siirtämään raportilla olevat tiedot joko tiedostoon tai lähettämään tiedot sähköpostin liitteenä halutulle henkilölle. Lisäksi raportilla on toiminto, jolla käyttäjä pystyy lataamaan raportin tiedot HTML-muodossa. HTML-tiedostoon sisältyy valmiina hakutoiminto, jolla tietoja pystytään hakemaan sivulta.

## 4. SUUNNITELUTYÖKALUN VAATIMUSMÄÄRITTELY

Ohjelmistoprojektin ensimmäinen vaihe on vaatimusmäärittelyn tekeminen. Vaatimusmäärittelyssä pyritään dokumentoimaan toteuttavan ohjelman toiminnallisuudet ja vaatimukset joita sovelluksessa halutaan olevan. Kielten opetuksen suunnittelutyökalun toteutus alkoi varsinaisesti Advanced SQL- opintojaksoilta, jossa piti tehdä harjoitustyönä kielikoulutuskeskuksen määrittelyn mukainen suunnittelutyökalu. Kielikoulutuskeskus on aikaisemmin hyödyntänyt opetuksen suunnittelussa Microsoft Office Excel taulukkolaskenta ohjelmaa, mihin tiedot on syötetty kurssien suunnittelua varten. Koulutustarpeiden tiedot on kerätty koulutuspäälliköiltä pääsääntöisesti sähköpostin välityksellä, minkä jälkeen tiedot on syötetty erikseen suunnittelutyökaluun.

Järjestelmän määrittely aloitettiin palaverilla, jossa käytiin läpi järjestelmän toiminnallisia vaatimuksia. Palaverin tarkoitus oli saada kuva siitä, mitä järjestelmältä vaaditaan ja millaiset odotukset käyttäjillä tulevasta järjestelmästä ovat. Palaverissa katsoimme Advanced SQL – kurssille tekemääni Apex sovellusta, minkä pohjalle suunnittelutyökalu perustuu. Tarkoituksena oli näyttää mitä toimintoja sovelluksessa on ja millaisia ominaisuuksia ja toimintoja suunnittelutyökaluun olisi mahdollista saada. Lisäksi katsoimme myös kurssien suunnittelussa käytettävää taulukkolaskentaohjelmaa ja tarkoituksena oli että pystyisimme määrittelemään, millä tavalla tietoa halutaan kerätä tulevaan järjestelmään ja millä tavalla järjestelmässä olevaa tietoa halutaan raportoida ja hyödyntää suunnittelussa.

### 4.1 Toiminnallisuuksien määrittely

Suunnittelutyökalun tarkoituksena on helpottaa sekä alkavien kurssien suunnittelua, että opettajien resurssien seuranta. Suunniteltaville kursseille pitää olla mahdollista syöttää tulosarvo, yksikkö, koulutuspäällikkö, toimipiste, kurssi, opetuskieli, opettaja, opintopisteet, resurssit ja jakso. Edellä mainitut kentät eivät ole pakollisia järjestelmässä, joten tyhjäksi jätetyt tiedot eivät estä kurssien tal-



lentamista järjestelmään. Tallennettujen kurssien tietoja pitää myös pystyä muokkaamaan jälkikäteen.

Suunniteltavien kurssien pitää olla näkyvissä yhdellä sivulla, josta suunniteltavia kursseja voidaan muokata, hyväksyä toteutukseen tai poistamaan järjestelmästä. Hyväksynnän jälkeen muokattu kurssi ei saa näkyä suunnittelunäkymässä ja hylätty kurssi poistetaan järjestelmästä.

Suunnittelutyökalussa pitää olla myös mahdollisuus luoda kurssiehdotuksia, jotka näkyvät omalla sivustolla, ja myös ehdotukset joko hyväksytään suunniteluun tai hylätään. Hyväksytty kurssiehdotus siirtyy suunnittelunäkymään käsiteltäväksi ja hylätty kurssi poistetaan järjestelmästä.

Useampien kurssien yhtäaikainen tuonti tekstitiedostosta pitää olla mahdollista. Toiminnon tarkoitus on vähentää manuaalista tiedon syöttämistä järjestelmään. Tietojen tuonnin jälkeen uusien kurssien tulee näkyä suunnittelunäkymässä.

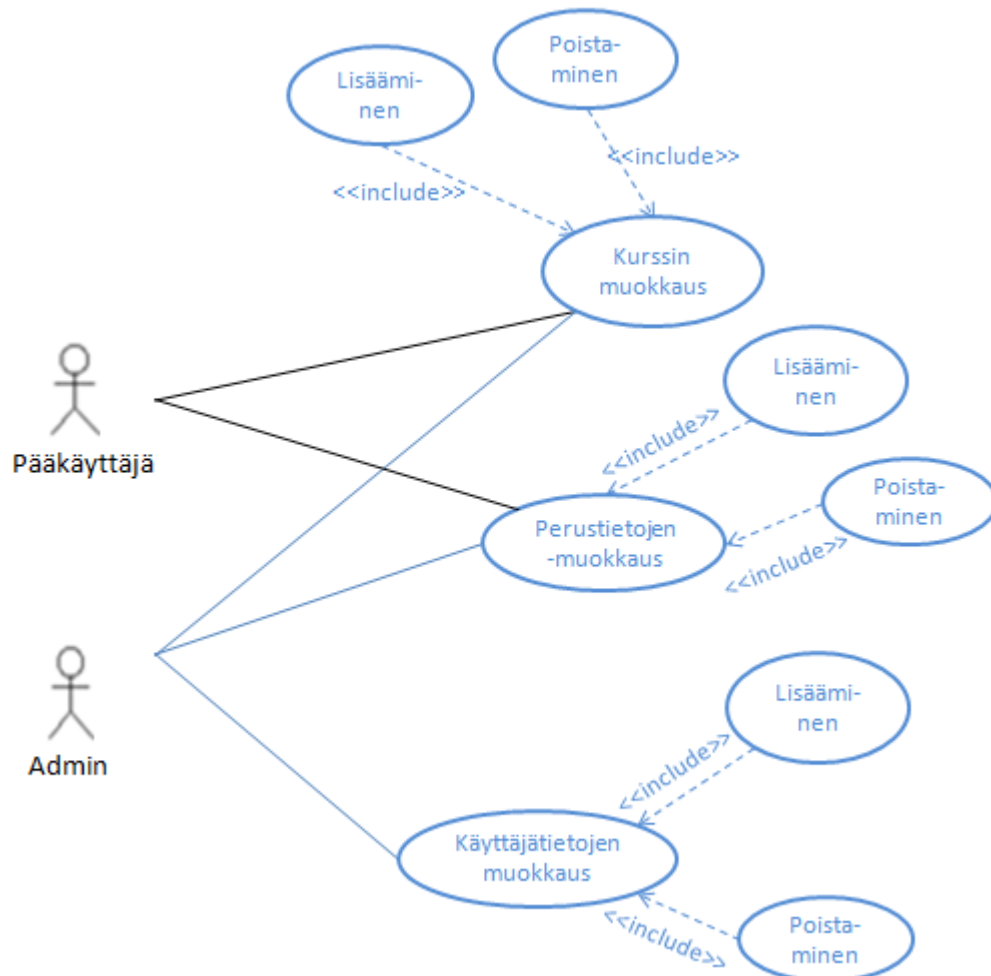
Järjestelmässä olevan yksittäisen kurssin kopiointi pitää olla myös mahdollista. Kurssin kopioinnin yhteydessä kurssin tietoja on pystyttävä muuttamaan ja kopioinnin jälkeen kurssin tulee näkyä suunnittelunäkymässä.

Suunnittelutyökalussa raportointi on yksi tärkeimmistä ominaisuuksista. Työkalussa pitää olla raportteja, joilla opettajille suunniteltuja ja toteutuneita resursseja pystytään seuraamaan ja hyödyntämään suunnitteluvaiheessa. Raporteilla ja lomakkeilla pitää olla toiminto, jolla tiedot pystytään siirtämään tekstitiedostoon sähköpostin liitetiedostona lähettämistä tai tiedon muokkaamista varten. Myös muokatuista kursseista pitää saada raportti.

Suunnittelutyökalussa pitää olla hakutoiminto, jolla kurssien tietoja pystytään hakemaan. Hakukriteereinä halutaan olevan ainakin koulutuspäällikkö, opettaja tai kurssi.

## 4.2 Käyttäjien roolit

Järjestelmän käyttäjinä toimivat kielikoulutuskeskuksen työntekijät, jotka syöttävät kurssitiedot järjestelmään. Tietojen syöttämisellä tarkoitetaan lomakkeiden manuaalista täyttämistä tai kurssitietojen tuontia tekstitiedostosta.



Kuvio 1: Käyttötapauskaavio

Kuviossa 1 on kuvattu järjestelmän kaksi käyttäjäroolia sekä käyttötapaukset näihin rooleihin liittyen. Alla on kuvattu käyttäjäroolien oikeuksia järjestelmään:

Pääkäyttäjä

- pystyy ylläpitämään järjestelmässä käytettäviä perustietoja kuten tulosalue, toimipiste ja kieli

- pystyy muokkaamaan, lisäämään ja poistamaan kurseja järjestelmästä
- ei pysty muokkaamaan tai lisäämään käyttäjiä järjestelmään

### Administrator

- pystyy ylläpitämään järjestelmässä käytettäviä perustietoja kuten tulosalue, toimipiste ja kieli
- pystyy muokkaamaan, lisäämään ja poistamaan kurseja järjestelmästä
- pysty lisäämään, muokkaamaan tai poistamaan käyttäjän järjestelmästä.

Alkuvaiheessa sovelluksessa on käytössä kaksi yllä mainittua käyttäjäryhmää. Sivustolle on lisätty valmiiksi autorisointi, jolla myös koulutuspäälliköille voidaan antaa lukuoikeus oman tulosalueen tietoihin. Koulutuspäälliköiden ryhmää ei ole kuvattu käyttötapauskaavioon, koska tämä toiminnallisuus ei varsinaisesti kuulunut määrittelyvaiheeseen.

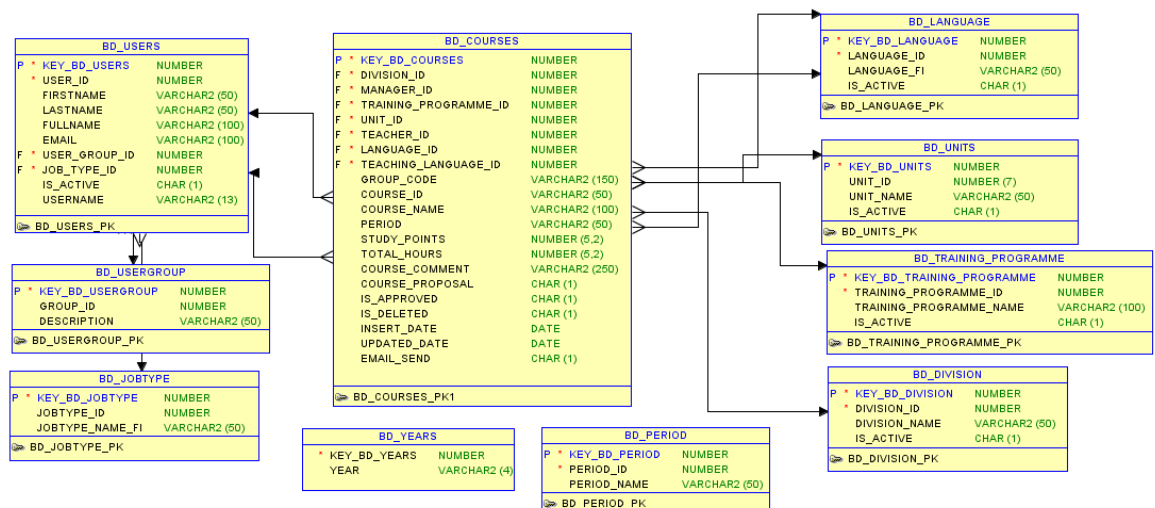
Tulosalueiden käyttäjäryhmät on lisätty valmiiksi järjestelmään, joten jos tämä toiminto halutaan ottaa käyttöön, pitää järjestelmän administratorin lisätä pelkästään käyttäjät oikeisiin ryhmiin.

## 5. TYÖKALUN SUUNNITTELU

Suunnitteluvaiheessa sain melko vapaat kädet sivustojen ja navigaation suunnitteluun, koska näiden osalta ei varsinaisesti tehty määrittämiä eikä sivustojen ulkoasun ei tarvinnut vastata minkään käytössä olevan sivuston mallia. Suunnitteluvaiheessa pyrin hyödyntämään mahdollisimman paljon aikaisemmin mainitsemaani ketterien menetelmien vaihejakomallia, mikä mahdollisti sovelluksen toimintojen toteutuksen jakamisen erillisiin vaiheisiin. Koska toteutettu sovellus pohjautuu aikaisemmin tekemääni harjoitustyöhön, oli tietokantataulujen, sovelluksen toimintojen ja sivuston ulkoasun suunnittelu helppo toteuttaa sykleissä.

### 5.1 Tietokantataulut

Suunnittelutyökalussa tiedot talletetaan BD\_COURSES tietokantatauluun, missä on viiteavaimet muihin kurssien suunnittelussa tarvittavien tietojen tietokantatauluihin.



Kuvio 2. Tietokannan rakenne.

Kuviossa 2 on esitelty tietokannan rakenne ja tietokantatauluihin kuuluvat pääavaimet (Primary key) ja viiteavaimet (Foreign key). Kuviossa olevien taulujen pääavaimentät näkyvät sinisellä P tunnuksella ja taulukuvauksen alaosassa

näkyy pääavaimen tietokantasovelluksessa käytettä nimi. Pääavaimeen liittyvät viiteavaimet näkyvät mustalla F tunnuksella ja taulujen välissä olevat nuolet osoittavat taulun, minkä taulun pääavaimeen viiteavain on yhdistetty. Taulujen kenttien perässä näkyvät käytettävät tietotyypit ja kenttien suurin sallittu merkkimäärä näkyvät kentän nimen perässä vihreällä tekstillä.

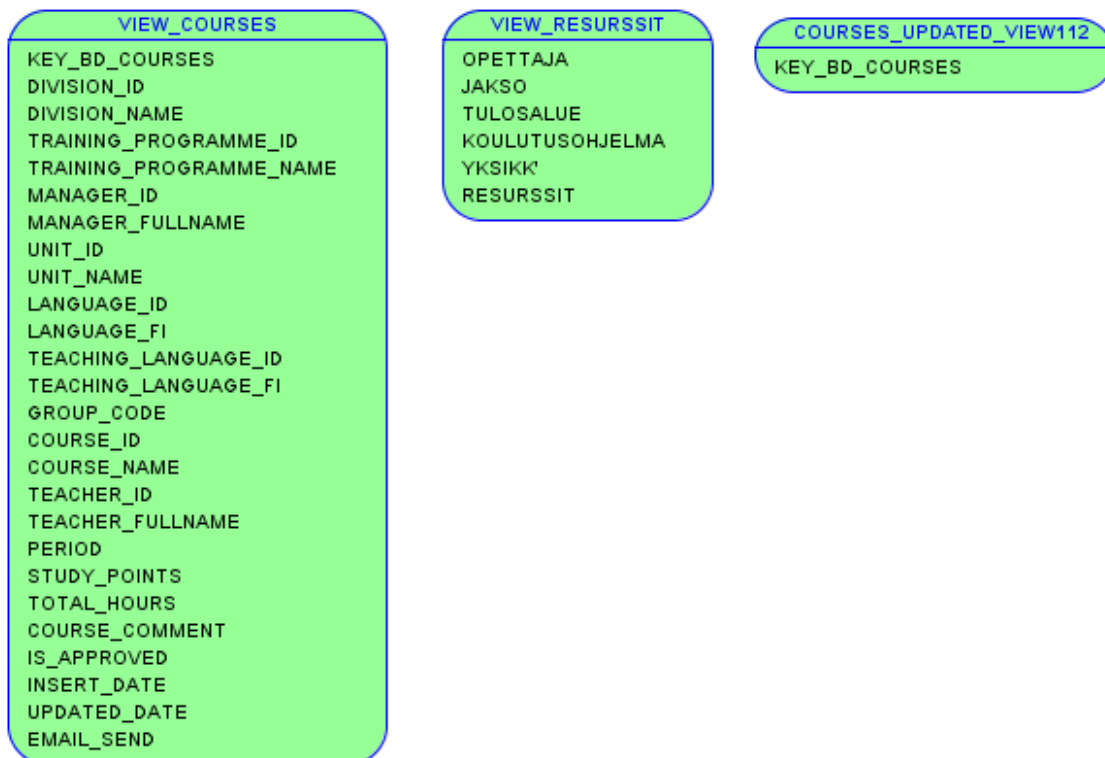
Tietokantataulu	Sisältö
BD_USERS	Opettajien ja koulutuspäälliköiden tiedot
BD_USERGROUP	Käyttäjryhmät
BD_JOBTYPE	Työsuhteen tyyppi
BD_UNITS	Ammattikorkeakoulun yksiköt
BD_DIVISION	Ammattikorkeakoulun tulosalueet
BD_LANGUAGE	Koulutuksissa käytettävät kielet
BD_TRAINING_PROGRAMME	Ammattikorkeakoulun koulutusohjelmat
BD_YEARS	Järjestelmässä käytettävät vuodet
BD_PERIOD	Opetuksen jaksot

Taulukko 1: Tietokantataulujen sisältöjen kuvaukset

Taulukossa 1 on kuvattu, mitä tietoa tietokantataulut sisältävät.

## 5.2 Näkymät

Kuviossa 3 on esitelty tietokantaan luotuja näkymiä (Views), joita käytetään pelkästään raportoinnissa ja tietojen päivitysten yhteydessä. Näkymät ovat SQL-kyselyllä tehtyjä virtuaalisia tauluja, joissa tiedot on haettu yhdestä tai useammasta taulusta tietyin rajauksin. Koska BD\_COURSES taulussa viiteavaintiedot ovat numereenisia, ei käyttäjä pysty tiedon perusteella tietämään mistä tiedosta on kyse, jos raportit luodaan suoraan edellä mainitusta tietokantataulusta.



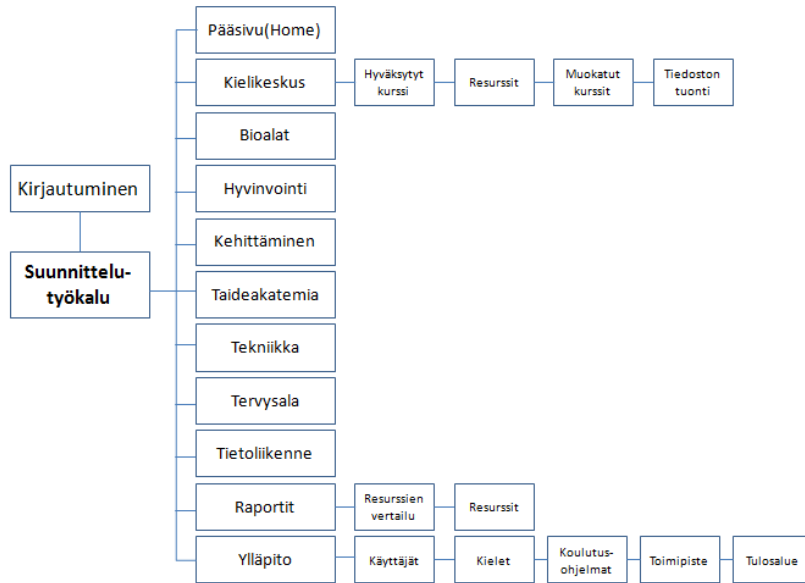
Kuvio 3: Näkymät tietokannasta

Näkymässä on yhdistetty BD\_COURSES tietokantataulun vierasavainkentät ja tietoon liittyvät pääavainkentät, jolloin näkymään saadaan lisättyä kyseisen kentän kuvaustieto pääavaimen tietokantatulusta. VIEW\_COURSES näkymään on haettu koulutuspäällikkö-, koulutusohjelma-, toimipiste-, tulosalue- ja opettajatietojen kuvaukset. VIEW\_RESURSSIT näkymä on rakennettu VIEW\_COURSES näkymästä ja siinä lasketaan opettajien resurssit tietylle jaksolle. COURSES\_UPDATED\_VIEW112 näkymää käytetään pelkästään tietojen päivittämiseen, kun kurssien tietoja muutetaan tietokantaan. Näkymässä verrataan kurssitietoja tilanteeseen, mikä oli ennen päivittämistä ja päivittämisen jälkeen, ja sen perusteella muokatuille kursseille lisätään muutosajankohta tietokantaan.

### 5.3 Navigointi

Navigointi haluttiin alun perin toteuttaa koulutusohjelmien perusteella, mutta koska koulutusohjelmia on yli 50 kappaletta, olisi navigoinnista saattanut tulla

liian vaikea käyttää. Tästä syystä toteutin navigoinnin tulosalueiden perusteella.



Kuvio 4: Suunnittelu-työkalun sivustorakenne.

Kuviossa 4 on esitelty sovelluksen sivustorakenne. Kirjautumisvaiheessa sovelluksessa suoritetaan proseduri, mikä tarkistaa mihin käyttäjäryhmään kyseinen käyttäjä kuuluu, ja ohjaa käyttäjän oikealle sivulle.



Kuva 7: Suunnittelu-työkalun Home sivu.

Kuvassa 7 on esitelty sovelluksen Home-sivu ja navigaatio palkki. Navigoinnin toteutus tulosalueiden perusteella mahdollistaa tiedon tehokkaamman rajaamisen käyttäjäryhmätasolla. Sovelluksessa on aluksi käytössä kaksi käyttäjäryhmää, jotka käyttävät sovellusta. Sovelluksen autorisointi on rakennettu valmiiksi.

siten, että myös koulutuspäälliköiden on mahdollista käyttää sovellusta, jos siihen on myöhäisemmässä vaiheessa tarvetta.



## 6. SUUNNITTELUTYÖKALUN TOTEUTUS

Suunnittelutyökalun toteutuksessa on pyritty käyttämään pelkästään järjestelmän omia toiminnallisuuksia ja toteutuksessa on käytetty Oraclen sivustolta löytyviä dokumentteja (Oracle Application Express Documentation). Toteutusvaiheessa pääsin tekemään suunnitteluvaiheen toiminnot järjestelmään iteraatiossa. Ensimmäinen vaihe oli luoda tietokantarakenteessa kuvatut tietokantataulut järjestelmään (Kuvio 2). Tietokantataulujen luonnin jälkeen pystyin aloittamaan dimensioissa käytettävien tietojen lataamisen järjestelmään, minkä jälkeen aloitin kurssitietojen siirtämisen järjestelmään. Tietojen tuontia varten tein järjestelmään näkymän ja raportin, joiden avulla pystyin tarkistamaan tietojen oikeellisuuden verratessani tietoja taulukkolaskentaohjelman ja suunnittelutyökalun välillä. Kun tietojen tuonti oli tehty ja tietojen oikeellisuus tarkistettu, pääsin luomaan varsinaisia lomakkeita, joita käytetään tietojen tallettamisessa ja muokkaamisessa.

### 6.1 Todennus

Todennuksen tarkoituksena on varmistaa, että järjestelmään kirjautuva käyttäjä on oikeutettu näkemään sovelluksessa olevat tiedot. Suunnittelutyökalussa käyttäjien todennus on toteutettu Oracle Apex:n omalla Application Express Account toiminnolla. Jokaiselle loppukäyttäjälle on perustettava käyttäjätunnus järjestelmään, jotta sovelluksen käyttäminen on mahdollista. Application Express Account käyttäjiä voivat lisätä vain Apex järjestelmän Administrator oikeudet omaavat käyttäjät.

### 6.2 Suunnittelu - ja tulosaluenäkymät

Suunnittelunäkymässä on käytetty järjestelmässä olevaa Tabular Form tyyppistä lomaketta ja lomaketta käytetään pelkästään tietojen päivittämiseen tai pois-

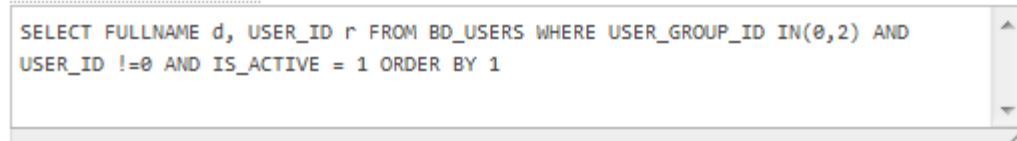
tamiseen tietokannasta. Lomakkeen luonnin yhteydessä lomakkeelle pitää määrittää lähdetaulu, käytettävät kentät ja ohjattu toiminto luo lomakkeen kyselyn automaattisesti. Lomakkeen kyselyssä haetaan tietokannasta ne kurssit joita ei ole hyväksytty. Lomakkeen luonnin yhteydessä sivulle luodaan tiedon tarkistamiseen tarvittavat toiminnot sekä tiedon päivittämiseen ja poistamiseen vaaditut prosessit. Luodulla lomakkeella on kolme eri osiota:

- Region Definition mikä sisältää varsinaisen SQL-kyselyn.
- Report Attributes sisältää SQL-kyselyssä olevat attribuutit eli kentät.
- Print Attributes osiossa määritetään lomakkeen tulostukseen tarvittavia tietoja kuten otsikkotiedot, käytettävä fontti sekä alatunniste.

Lomakkeen luonnin jälkeen kehittäjä pystyy muuttamaan Reports Attributes osiossa lomakkeella olevien kenttien muotoilun tai piilottamaan kentät lomakkeelta. Koska kyseisessä taulussa tiedot ovat viiteavaimia, piti osalle attribuuteista luoda arvoluettelo viiteavaimen kohdetaulusta ja muuttaa kenttä valintalistatyypiksi kentäksi.

List of values definition

```
SELECT FULLNAME d, USER_ID r FROM BD_USERS WHERE USER_GROUP_ID IN(0,2) AND
USER_ID !=0 AND IS_ACTIVE = 1 ORDER BY 1
```



Kuva 8: Arvoluettelon kysely.

Kuvassa 8 on esitelty sovelluksessa oleva arvoluettelo. Arvoluettelo (List of values) on SQL-kielellä luotu kysely, missä määritellään luettelossa näytettävä tieto (d) ja tieto mikä palautetaan tietokantaan (r) päivityksen yhteydessä. Yllä olevassa esimerkissä luettelossa (kuva 7) näytetään BD\_USERS taulusta käyttäjän kokonimi ja päivityksen yhteydessä tietokantatauluun talletetaan kyseisen käyttäjälle määritetty yksilöivä käyttäjätunnus.

Suunnittelunäkymässä pystytään suorittamaan myös hakuja suunnittelussa olevien kurssien osalta. Sivun yläosassa oleville valintalistoilta käyttäjä voi valita arvoja ja suorittaa näiden perusteella hakuja.

The screenshot shows a web application interface for course planning. At the top, there is a navigation menu with tabs: Home, Kielikeskus (highlighted in red), Bioalat ja liiketalous, Hyvinvointipalvelut, Kehittäminen, Taideakatemia, Tekniikka, ympäristö ja talous, Terveysala, Tietoliikenne ja sähköinen kauppa, Raportit, and Ylläpito. Below the menu is a search bar with five dropdown menus: 'Valitse tulosalue', 'Valitse koulutusohjelma', 'Valitse koulutuspaikallisko', 'Valitse kieli', and 'Valitse opettaja'. A 'Hae' button is on the right.

Below the search bar is a table titled 'Suunnittelu'. The table has columns: 'Tulosalue', 'Koulutusohjelma', 'Koulutuspaikallisko', 'Toimipiste', 'Opettaja', and 'Kursin'. The table contains 8 rows of data. The first two rows are for 'BIOALAT JA LIIKETALOUS' with 'Bio- ja elintarviketekniikka' as the program. The remaining six rows are for 'TEKNIikka, YMPÄRISTÖ JA TALOUS' with 'Auto- ja kuljetustekniikka' as the program. The 'Opettaja' column lists names like Paula Aali and Marjo Aaltonen. On the right side of the table, there are small icons and text indicating the course language, such as 'Suomen viesti', 'saksan aikeet', and 'Englän kielen'.

Tulosalue	Koulutusohjelma	Koulutuspaikallisko	Toimipiste	Opettaja	Kursin
BIOALAT JA LIIKETALOUS	Bio- ja elintarviketekniikka	Seikola, Aila	Lemminkäisenkatu	Aali, Paula	Suomen viesti
BIOALAT JA LIIKETALOUS	Bio- ja elintarviketekniikka	Seikola, Aila	Lemminkäisenkatu	Aali, Paula	Suomen viesti
TEKNIikka, YMPÄRISTÖ JA TALOUS	Myyntityö	Kumpula, Marjo	Sepänkatu	Aaltonen, Marjo	saksan aikeet
TEKNIikka, YMPÄRISTÖ JA TALOUS	Auto- ja kuljetustekniikka	Lindström, Kari	Sepänkatu	Aaltonen, Marjo	Englän
TEKNIikka, YMPÄRISTÖ JA TALOUS	Auto- ja kuljetustekniikka	Lindström, Kari	Sepänkatu	Aaltonen, Marjo	Englän
TEKNIikka, YMPÄRISTÖ JA TALOUS	Auto- ja kuljetustekniikka	Lindström, Kari	Sepänkatu	Aaltonen, Marjo	Englän kielen
TEKNIikka, YMPÄRISTÖ JA TALOUS	Auto- ja kuljetustekniikka	Lindström, Kari	Sepänkatu	Aaltonen, Marjo	Saksan aikeet
TEKNIikka, YMPÄRISTÖ JA TALOUS	Auto- ja kuljetustekniikka	Lindström, Kari	Sepänkatu	Aaltonen, Marjo	Saksan

Kuva 9. Suunnittelulomakkeen hakutoiminto.

Kuvassa 9 on esitelty suunnittelulomakkeen hakutoiminnossa käytettävät valintalistoilta toteutetut hakukentät, joista käyttäjä voi valita hakuehdot haettaville tiedoille. Hakutoiminto on rakennettu siten, että kun haku suoritetaan, siirtyy käyttäjä uudelle sivulle ja hakukenttien arvot siirretään uudella sivulla olevien kenttien arvoiksi ja näiden tietojen perusteella suoritetaan haku tietokannasta.

Sovelluksessa on valmiina omat raportointisivut Ammattikorkeakoulun tulosalueille. Tulosalueet ovat Ammattikorkeakoulun yksiköitä, jotka vastaavat koulutuksesta, kuten Bioalat- ja liiketalous, Tietoliikenne ja sähköinen kauppa sekä Kehittäminen. Tulosalueiden näkymät on toteutettu raporteilla, koska varsinaista ylläpitoa ei haluta tehdä muualla kuin suunnittelunäkymässä. Myös tulosalueiden sivuilta pystytään kurssitiedot lataamaan tekstitiedostoon. Raporttien rakenne vastaa suunnittelunäkymää ja tekstitiedostoa pystytään käyttämään pohjana, kun uusia kurssitietoja tuodaan järjestelmään tekstitiedostosta.

## 6.3 Raportointi ja ylläpito

Raportointi opetuksen suunnittelussa on erityisen tärkeä, koska raportoinnilla pystytään seuraamaan opettajien suunniteltuja ja toteutuneita resursseja. Suunnittelutyökalussa on oma sivu raportoinnille, mistä löytyy erilaisia raportteja opettajien resurssien seurantaan. Raporteilla voidaan verrata esimerkiksi opettajien toteutuneita resursseja eri vuosilta keskenään.

Sovelluksessa järjestelmässä olevien tietojen ylläpitoon on oma sivu, jonne kielikeskus ja admin käyttäjäryhmiin kuuluvilla henkilöillä on pääsy. Ylläpitosivulla on järjestelmässä käytettävien perustietojen ylläpitoon vaaditut lomakkeet. Sivulla voidaan lisätä ja poistaa käyttäjiä, tulosalueita, toimipisteitä, opetuskieliä, koulutusohjelmia.

## 7. SUUNNITTELUTYÖKALUN TESTAUS

Testausvaihe on yksi tärkeimmistä vaiheista ohjelmistoprojektissa jossa vaihejakomallin iteraation toiminnallisuus hyväksytään. Testausvaiheen tarkoituksena on löytää virheet, joita sovellus sisältää ja varmistaa, että sovellus toimii vaatimusmäärittelyn mukaisesti. Suunnittelutyökalun testaaminen päästiin aloittamaan kun lomakkeet kurssien ylläpidolle olivat valmiina ja varmistettu, että tiedot kirjautuvat oikein tietokantaan. Suunnittelutyökalussa käytettävät tiedot laddattiin käytössä olevasta Microsoft Excel taulukkolaskentaohjelmasta järjestelmään. Excelissä olevista tiedoista erotettiin dimensioiden tiedot ja kurssien tiedot omiksi tiedostoiksi, että testiympäristön tiedot saatiin vastamaan todellista suunnittelutilannetta. Testiympäristössä oleva data ladataan asennusvaiheessa uuteen ympäristöön suunnittelua ja raportointi varten.

Seuraavaksi kielikoulutuskeskuksen henkilökunta aloitti järjestelmän testikäytön eli varsinaisen testaamisen. Testaaminen on jaettu kahteen osaan, joissa ensimmäisessä vaiheessa testataan lomakkeiden toiminnot ja toisessa vaiheessa raportointi.

Suunnittelutyökalun testikäytön tarkoituksena oli varmistaa, että järjestelmä on vaatimusmäärittysten mukainen. Testikäytössä sovellus toimi hienosti, mutta joitakin muutoksia jouduttiin tekemään.

Kurssien suunnittelunäkymä oli alun perin toteutettu raportilla ja kurssin tietoja pääsi muuttamaan muokkauslomakella, kun kurssi avattiin muokkaustilaan raportilla olevasta linkistä. Testikäytössä kurssien ylläpito todettiin oleva hidasta ja tähän haluttiin tehdä muutos, koska jokainen kurssi oli avattava erikseen, että muokkaus oli mahdollista. Muutoksen tarkoituksena oli saada useamman kurssin tiedot näkyviin ja mahdollisuus muokata useampaa kurssia saman aikaisesti. Muutoksen vuoksi suunnittelunäkymä tehtiin Tabular Form -lomakkeella, että toiminnallisuus voitiin toteuttaa. Muutoksen jälkeen kurssien tiedot näkyvät lomakkeella ja lomakkeella pystyy ylläpitämään tai poistamaan yhden tai useamman kurssin tietoja samalla kertaa.

Testikäytön aikana tuli tarve nähdä muokattujen kurssien tietoja. Kuvassa 10 on suunnittelusivustolla olevan lomakkeen prosessit, joita käytetään kun sivulla olevia tietoja päivitetään.

Processes			
After Submit			
10	<a href="#">LOAD COURSES</a>	PL/SQL anonymous block	Conditional
20	<a href="#">ApplyMRU</a>	Multi Row Update	Conditional
30	<a href="#">ApplyMRD</a>	Multi Row Delete	Conditional
40	<a href="#">UPDATE COURSES</a>	PL/SQL anonymous block	Conditional

Kuva 10: Sivun prosessit suunnittelunäkymässä

Haasteelliseksi muodostui kurssien muokattu tieto, koska päivityksen yhteydessä kaikki lomakkeen tiedot talletetaan tietokantaan eikä näin ollen muokattuja tietoja saanut selville helposti. Tästä syystä jouduin tekemään kaksi uutta prosessia sivulle, jotka suoritetaan kun päivityspainiketta painetaan sivulla.

Päivityksen alussa kaikki suunnittelussa olevien kurssien tiedot talletetaan tietokannassa olevaan työtauluun `LOAD_COURSES` prosessilla. Tämän jälkeen järjestelmän oma päivitysprosessi päivittää lomakkeella olevat tiedot tietokantaan. Viimeisenä suoritetaan `UPDATE_COURSES` prosessi, missä verrataan työtauluun tallennettuja tietoja sekä kurssien tietokantataulussa olevia tietoja, ja vain niille kurseille joilla jokin tieto on muuttunut, lisätään päivitetty merkki.

Testikäytön aikana nähtiin tarpeelliseksi myös opettajan resurssien automaattinen laskeminen suunnittelunäkymässä. Opettajien resurssien automaattinen laskenta toteutettiin Javascript – koodilla, mikä on yleisesti käytössä HTML-sivustoissa ja Javascript lisätään sivun Page Rendering osiossa olevaan HTML-Header – kohtaan.

```

HTML Header
<script type="text/javascript">

function calculateResources(pThis)
{
var vRow = pThis.id.substr(pThis.id.indexOf('_')+1);
html_GetElement('f14_'+vRow).value =
  parseInt(html_GetElement('f13_'+vRow).value,10) * 23;
}
</script>

```

Kuva 11. Javascript sivun Header osiossa.

Jotta Javascript saadaan toimimaan sivulla, pitää lomake kääntää HTML-koodiksi ja hakea tarvittavien kenttien nimet koodin sisältä. Kun Javascript -koodi on lisätty sivulle, pitää lomakkeella olevan kentän Element Attributes koh-  
taa lisätä Javascript – koodi.

**Column Attributes**

<u>Display As</u>	<input type="text" value="Text Field"/>
<u>Number / Date Format</u>	<input type="text"/>
	<small><a href="#">Numeric format mask: 999G999G999G999G990</a></small>
	<small><a href="#">Graphical formatting for percentages, whole numbers between 0 and 100</a></small>
<u>Field Template</u>	<input type="text" value="- render form field without template -"/>
<u>Element Width</u>	<input type="text" value="2"/>
<u>Element CSS Classes</u>	<input type="text"/>
<u>Element Attributes</u>	<input calculateresources(this);"="" type="text" value="onchange="/>
<u>Element Option Attributes</u>	<input type="text"/>

Kuva 12: Javascript lomakkeen opintopisteet Column Definition osiossa.

Kuvassa 12 on esitelty suunnittelulomakkeella olevan opintopisteiden laskemiseen käytetty Javascript koodi. Tämän jälkeen kun käyttäjä muuttaa lomakkeella olevan opintopisteen arvon ja poistuu kentästä, laskee järjestelmä automaattisesti opettajan resurssit ja tiedot voidaan päivittää tietokantaan.

## 8. KÄYTTÖÖNOTTO

Sovelluksen käyttöönotto on ohjelmistoprojektin viimeinen vaihe, jossa testikäytön jälkeen hyväksytty järjestelmä asennetaan varsinaisesti loppukäyttäjille tarkoitettuun ympäristöön. Suunnittelutyökalun käyttöönotto oli itse asiassa helppo, koska testiympäristössä ollutta järjestelmää käytettiin kurssien suunnittelussa.

Käyttöönottoa varten ammattikorkeakoulun verkkoon asennettiin Windows Server 2008 R2 palvelin sovellusta varten. Ensimmäinen vaihe oli asentaa Oracle Database 11g Express Edition tietokantasovellus, mihin Oracle Application Express sisältyy. Tietokantasovelluksen asennuksen jälkeen piti Apex sovelluskehittäjän ohjelmisto päivittää uudempaan versioon, koska testiympäristössä oli käytössä uudempi versio, eikä asennus vanhempaan versioon ole mahdollista.

Versiopäivityksen jälkeen pääsin luomaan tarvittavat tietokantataulut, joita suunnittelutyökalussa käytetään. Tietokantataulujen luonti oli nopea, koska testiympäristöstä pystyi ottamaan DDL script - tiedoston jolla sovelluksessa käytettävät taulut, näkymät, triggerit ja sequensit pystyi luomaan uuteen ympäristöön. Myös tietokantatauluissa olleet tiedot siirrettiin testiympäristöstä omiksi tiedostoiksi ja tietojen vienti uuteen ympäristöön sujui ongelmitta ja nopeasti Apex sovelluksen import toiminnon avulla.

Varsinaisen sovelluksen asennus oli myös nopea, koska Apexista on mahdollista siirtää sovelluksia asennuspaketiksi esimerkiksi omalle tietokoneelle ja asentaa toiseen ympäristöön ohjatun toiminnon avulla. Ohjatussa toiminnossa asennuspaketti ladataan sovellukseen minkä jälkeen sovelluksen asennus tapahtuu taustalla.

Asennuksen ja tietojen tuonnin jälkeen oli käytettävien ryhmien luonti järjestelmään, minkä jälkeen sovellus oli valmis käytettäväksi. Varmistin sovelluksen toimivuuden lisäämällä testidataa järjestelmään ja testi osoitti sen, että järjestelmä on valmis käyttöä varten.



## 9. ARVIOINTI JA JATKOKEHITYKSEN IDEOINTI

Opinnäytetyöprojektin tarkoituksena oli toteuttaa kielikoulutuskeskukselle järjestelmä, jolla kieltenopetuksen kurssien suunnittelu olisi vaivatonta ja tehokasta. Projekti eteni mielestäni aikatauluuni nähden hyvin, vaikka välillä testaus ja kehityksen ajankäyttö eivät oikein kohdanneet toisiaan.

Ohjelman testikäytön ja asennuksen jälkeen voi sanoa, että projekti onnistui ehkä paremmin kuin mitä aluksi ajattelin ja sovelluksesta saatiin vaatimusmäärittelyn mukainen. Vaikka olin tutustunut Oracle Application Express sovelluskehittäjän käyttöön Advanced SQL – opintojaksolla, missä harjoitustyönä tehtiin suunnitteluohjelma kielikeskukselle, ei sovelluksen tekeminen ollut helppoa. Harjoitustyönä tehty sovellus ei sisältänyt yhtä paljon toimintoja, kuin mitä käyttönotettu sovellus sisältää ja opin todella paljon uusia asioita sovelluskehittäjästä ja sen toiminnallisuuksista. Koska sovelluskehittäjä on helppokäyttöinen ja sisältää valmiina paljon erilaisia toiminnallisuuksia, soveltui se erinomaisesti tähän projektiin. Testikäytön aikana saamani positiivinen palaute vaikutti suuresti siihen, että sain sovelluksen valmiiksi ja asennetuksi opetuksen suunnitteluun varten.

Suunnittelutyökalun jatkokehitykseen liittyen olisi muutama toiminto, mitkä kannattaisi lisätä järjestelmään. Ensimmäinen vaihe voisi mielestäni olla se, että koulutuspäälliköille annetaan mahdollisuus lisätä kurssiehdotuksia järjestelmään ja kurssien hyväksyntä suunnitteluun jäisi kielikeskuksen tehtäväksi.

Seuraava vaihe voisi olla se, että opettajat pääsisivät katsomaan omia tulevia kursseja ja toteutuneita kursseja järjestelmästä suoraan. Voisihan järjestelmää hyödyntää myös kaikkien opetusten suunnittelussa, jos sellaiseen on tarvetta, koska järjestelmä sisältää valmiina kaikki koulutusohjelmat ja tulosalueet.

## LÄHTEET

- What is Oracle Application Express? Viitattu 15.01.2013  
<http://www.oracle.com/technetwork/developer-tools/apex/overview/index.html>
- Oracle Application Express Documentation. Viitattu 15.01.2013  
<http://www.oracle.com/technetwork/developer-tools/apex/documentation/index.html>
- Oracle Application Express Overview. Viitattu 15.01.2013  
<http://www.oracle.com/technetwork/developer-tools/apex/overview/index.html>
- Oracle 2 days + Developer guide. Viitattu 15.01.2013  
[http://docs.oracle.com/cd/E37097\\_01/doc/doc.42/e35122.pdf](http://docs.oracle.com/cd/E37097_01/doc/doc.42/e35122.pdf)
- Agile Manifesto 2001. Principles behind the Agile Manifesto. Viitattu 15.01.2013  
<http://agilemanifesto.org/principles.html>
- Scott W. Ambler. Feature Driven Development (FDD) and Agile Modeling. Viitattu 15.01.2013  
<http://www.agilemodeling.com/essays/fdd.htm>