Suunnitteludokumentti

”Sanakysely”-tietokantasovellus

Pete Saarela
pete.saarela@cs.helsinki.fi

Aineopintojen harjoitustyö:
Tietokantasovellus
Kurssikoodi 582203

19.10.2014
Helsingin yliopisto
tietojenkäsittelytieteen laitos

Sisällys

1. Johdanto

'Sanakysely' on tietokantasovellus, joka on tarkoitettu apuvälineeksi sanojen merkityksen ja kirjoitusasun opiskeluun. Tietokantaan tallennetaan sanastoja halutuista aiheista eri kielillä ja vaikeustasoilla. Tämä tapahtuu järjestelmän ylläpitäjän (opettajan) toimesta. Opettaja voi halutessaan lisätä tai poistaa sanastoja ja muokata kannassa olevien sanastojen sanoja. Peruskäyttäjä (oppilas) voi käyttää ohjelmaa tenttaamaan itseltään sanastoja sana kerrallaan. Sanat annetaan satunnaisessa järjestyksessä ja listalla oleva sana säilyy kyseltävänä siihen asti, kunnes siihen on annettu hyväksytty käännös tai jos oppilas ohittaa kyseisen sanan. Sovellus pitää kirjaa oppilaan tuloksista ja tenttikerroista. Nämä tilastot ovat opettajan tarkasteltavissa. Ohjelmaa voi mainiosti käyttää itseopiskeluun sanakokeiden lähestyessä.

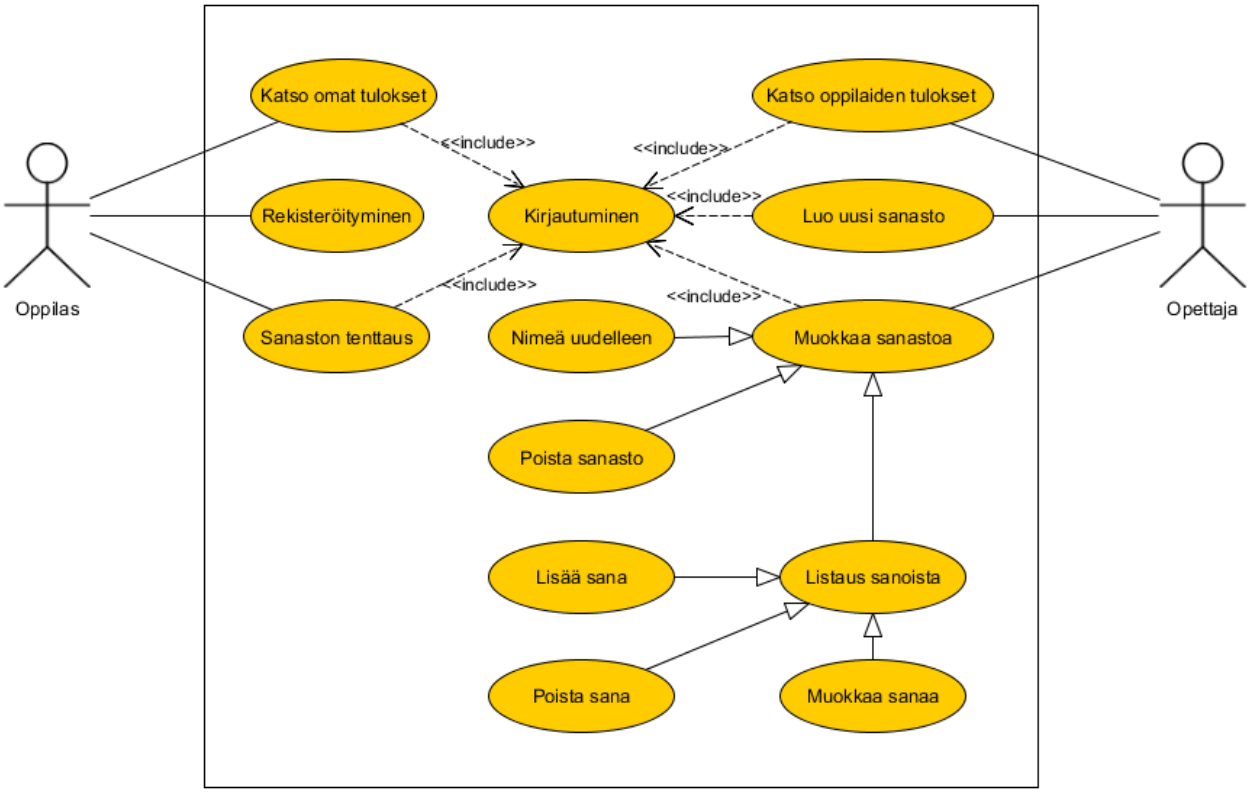
Toimintoja:

- Kirjautuminen
- Rekisteröityminen
- Sanastojen tenttaus
- Oppilaalle omien tulosten tarkistaminen
- Opettajalle kaikkien oppilaiden tulosten tarkistaminen
- Sanastojen muokkaaminen, lisäys ja poisto
- Yksittäisen sanan muokkaaminen, lisäys ja poisto kussakin sanastossa
- Sanojen kyselyjärjestys arvotaan kullakin käyttökerralla erilaiseksi
- Sanojen ja niiden käännösten kyselysuunnan voi valita mieleisekseen kullakin kyselykerralla

Järjestelmä toteutetaan PHP-kielellä ja se hyödyntää Helsingin yliopiston tietojenkäsittelytieteen laitoksen PostgreSQL-tietokantapalvelinta.

2. Yleiskuva järjestelmästä

Käyttötapauskaavio



Käyttäjäryhmät

Oppilas on sovelluksen peruskäyttäjä. Rekisteröidyttyään hän voi käyttää ohjelmaa sanastojen tenttaamiseen ja omien aikaisempien tenttien tulosten tarkistamiseen.

Opettaja on sovelluksen sisällön ylläpitäjä, joka hallinnoi sanastoja ja niihin kuuluvia sanoja sekä tarkistaa oppilaiden suorituksia.

Käyttötapauskuvaukset

Käyttötapaus 1:

- *Käyttäjä*: oppilas
- *Tavoite*: suorittaa uuden sanaston tenttaus
- *Laukaisija*: oppilaan tarve
- *Esiehto*: oppilas on rekisteröitynyt järjestelmään
- *Käyttötapauksen kulku*:

1. Oppilas kirjautuu järjestelmään
2. Järjestelmä näyttää luettelon saatavilla olevista sanastoista
3. Oppilas valitsee luettelosta haluamansa sanaston
4. Järjestelmä kysyy, kysytäänkö sanan vieraskielistä vai omakielistä muotoa
5. Oppilas tekee valinnan kyselyn suunnasta
6. Järjestelmä noutaa sanaston sisällön tietokannasta ja arpoo sanoille järjestyksen
7. Järjestelmä esittää listan seuraavan sanan ja kysyy sille käännöstä
8. Oppilas ehdottaa vastausta
9. Järjestelmä tarkistaa, onko vastaus oikein ja kertoo tuloksen oppilaalle
    - jos vastaus on väärin, sana säilyy kysyttävien listalla
    - jos vastaus on oikein, sana poistetaan listalta ja oppilaalle lisätään piste
10. Järjestelmä tarkistaa, onko sanasto käyty lävitse kokonaan
11. Jos listalta löytyy vielä sanoja, palataan kohtaan 7
12. Järjestelmä kertoo oppilaalle tentin tuloksen ja lisää suorituksen tietokantaan
13. Oppilas poistuu ohjelmasta painikkeella ’Kirjaudu ulos’

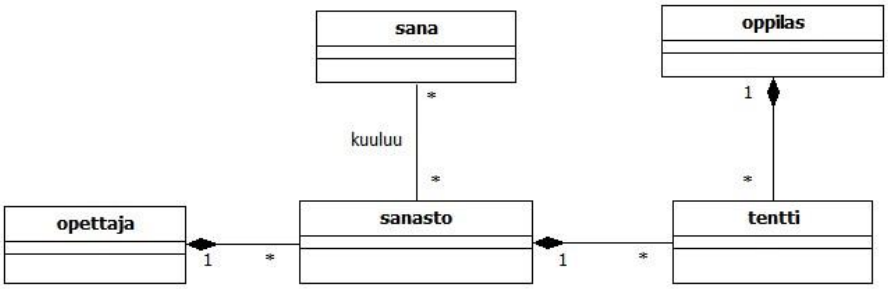
- *Poikkeuksellinen toiminta*:
    a. Oppilas keskeyttää kyselyn, vaikka ei ole vielä vastannut oikein kaikkiin kysymyksiin
    b. Järjestelmä palauttaa oppilaan takaisin etusivulle, kohtaan 2

Käyttötapaus 2:

- *Käyttäjä*: opettaja
- *Tavoite*: muokata jo olemassa olevaa sanastoa
- *Laukaisija*: sanaston laajentuminen
- *Esiehto*: opettajalle on annettu tarvittaviin käyttöoikeuksiin oikeuttavat tunnukset
- *Käyttötapauksen kulku*:
    1. Opettaja kirjautuu järjestelmään
    2. Järjestelmä näyttää luettelon saatavilla olevista sanastoista
    3. Opettaja valitsee luettelosta haluamansa sanaston (esimerkiksi ’Tietotekniikkasanasto 2, eng-suo’)
    4. Järjestelmä näyttää luettelon sanastoon kuuluvista sanoista käännöksineen
    5. Opettaja valitsee toiminnon ’Lisää sana’
    6. Järjestelmä antaa tarvittavat kentät täytettäväksi
    7. Opettaja lisää sanan, sen käännöksen ja tarvittavat lisätiedot
    8. Järjestelmä tarkistaa, ovatko syötteet sallituissa rajoissa
    9. Järjestelmä joko tallentaa sanan tietokantaan tai raportoi opettajalle virheestä
    10. Opettaja palaa halutessaan kohtaan 5
    11. Opettaja palaa etusivulle painikkeella ’Sanastot’
    12. Järjestelmä näyttää luettelon sanastoon kuuluvista sanoista käännöksineen
    13. Opettaja poistuu ohjelmasta painikkeella ’Kirjaudu ulos’

3. Järjestelmän tietosisältö

Tietosisältökaavio



Tietokohteiden kuvaukset

opettaja

Attribuutti	Arvojoukko	Kuvailu
<<key>>opettajatunnus	serial	Numeerinen pääavain
nimi	varchar(80)	Opettajan nimi
salasana	varchar(12)	Salasana
tehdyt	integer ARRAY	Taulukko niistä sanastoista, jotka opettaja on tehnyt

Opettaja on järjestelmän sisällön ylläpitäjä, joka laatii sanastot ja tekee niihin muutoksia. Hän voi katsella kaikkien oppilaiden tuloksia.

sana

Attribuutti	Arvojoukko	Kuvailu
<<key>>sanatunnus	serial	Numeerinen pääavain
kohde	varchar(30)	Sanan omakielinen muoto
kieli	varchar(9)	Kohteen kieli, esim. suo - eng
kaannos	varchar(30)	Sanan perusmuoto vieraskielisessä muodossa
sluokka	varchar(15)	Sanaluokka, esim. adjektiivi, substantiivi, verbi, pronomini, partikkeli, adverbi
artikkeli	varchar(5)	Substantiiviin liittyvä liite, esim. en, ett, a, an
taivutus	varchar(50)	Sanan mahdolliset taivutusmuodot, esim. en katt, katten, katter, katterna

Sana on itsenäinen tietokohde, joka voi liittyä yhteen tai useampaan sanastoon.

oppilas

Attribuutti	Arvojoukko	Kuvailu
<<key>>oppilastunnus	serial	Numeerinen pääavain
nimi	varchar(80)	Oppilaan nimi
salasana	varchar(12)	Salasana
tehdyt	integer	Montako sanastoa suoritettu
viimeksi	date	Viimeksi kirjautuneena

Oppilas on järjestelmän peruskäyttäjä, joka voi suorittaa tenttejä ja saada niistä tuloksia. Hän voi katsella vain omia tuloksiaan.

sanasto

Attribuutti	Arvojoukko	Kuvailu
<<key>>sanastotunnus	serial	Numeerinen pääavain
nimi	varchar(80)	Sanaston nimi, joka kuvaa sen sanojen aihepiiriä
kieli	varchar(12)	Sanastoon kuuluvien sanojen kieli, esim. suo - eng
maara	integer	Montako sanaa sanastossa on
kuvaus	varchar(200)	Sanaston tarkempi kuvaus
tehty	date	Milloin sanasto on laadittu

Sanastoon kuuluu joukko sanoja. Sen laatijana on opettaja. Sanastoon kohdistuu oppilaan tekemä tentti.

kuuluu

Attribuutti	Arvojoukko	Kuvailu
<<fkey>>sanastotunnus	integer	Viittaus sanasto-tauluun
<<fkey>>sanatunnus	integer	Viittaus sana-tauluun

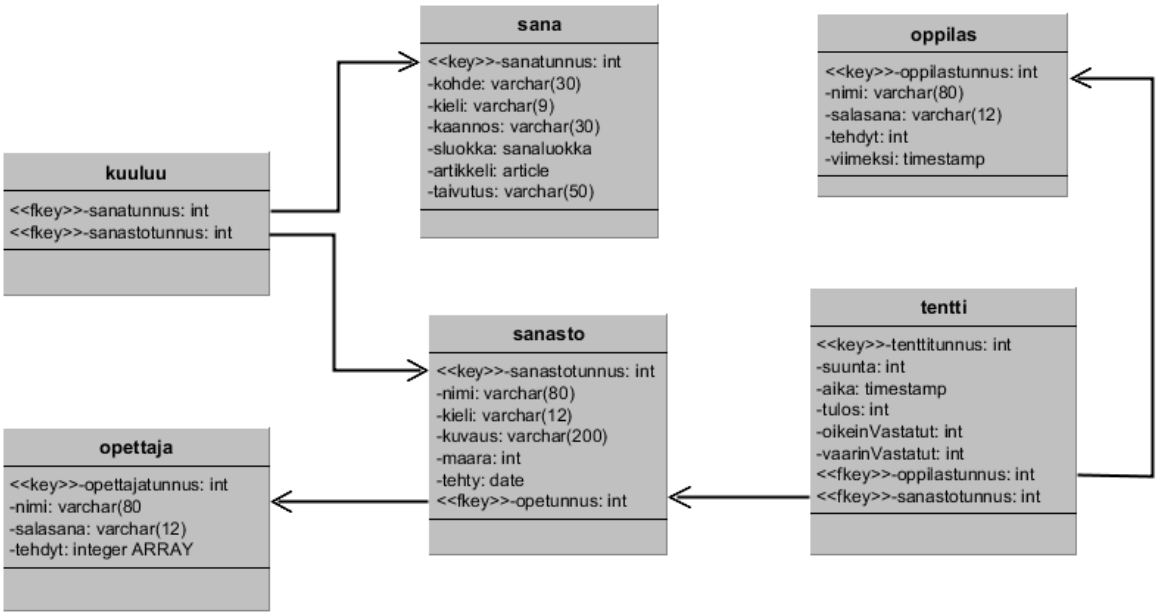
Kuuluu on välitaulu, joka purkaa Sanaston ja Sanan välisen monesta-moneen-yhteyden.

tentti

Attribuutti	Arvojoukko	Kuvailu
<<key>>tenttitunnus	serial	Numeerinen pääavain
suunta	integer	Kumpaan suuntaan sanastoa kysellään tietyssä tentissä
tulos	integer	Kokonaistulos
aika	timestamp	Milloin tentti suoritettiin
oikeinVastatut	integer	Montako oikeaa vastausta
vaarinVastatut	integer	Montako väärää vastausta

Tentti on oppilaan tiettyyn sanastoon kohdistettu kyselykerta. Oppilas voi tenttiä samaa sanastoa useita kertoja. Kullakin oppilaalla voi olla useita tenttejä.

4. Relaatiotietokantakaavio



5. Järjestelmän yleisrakenne

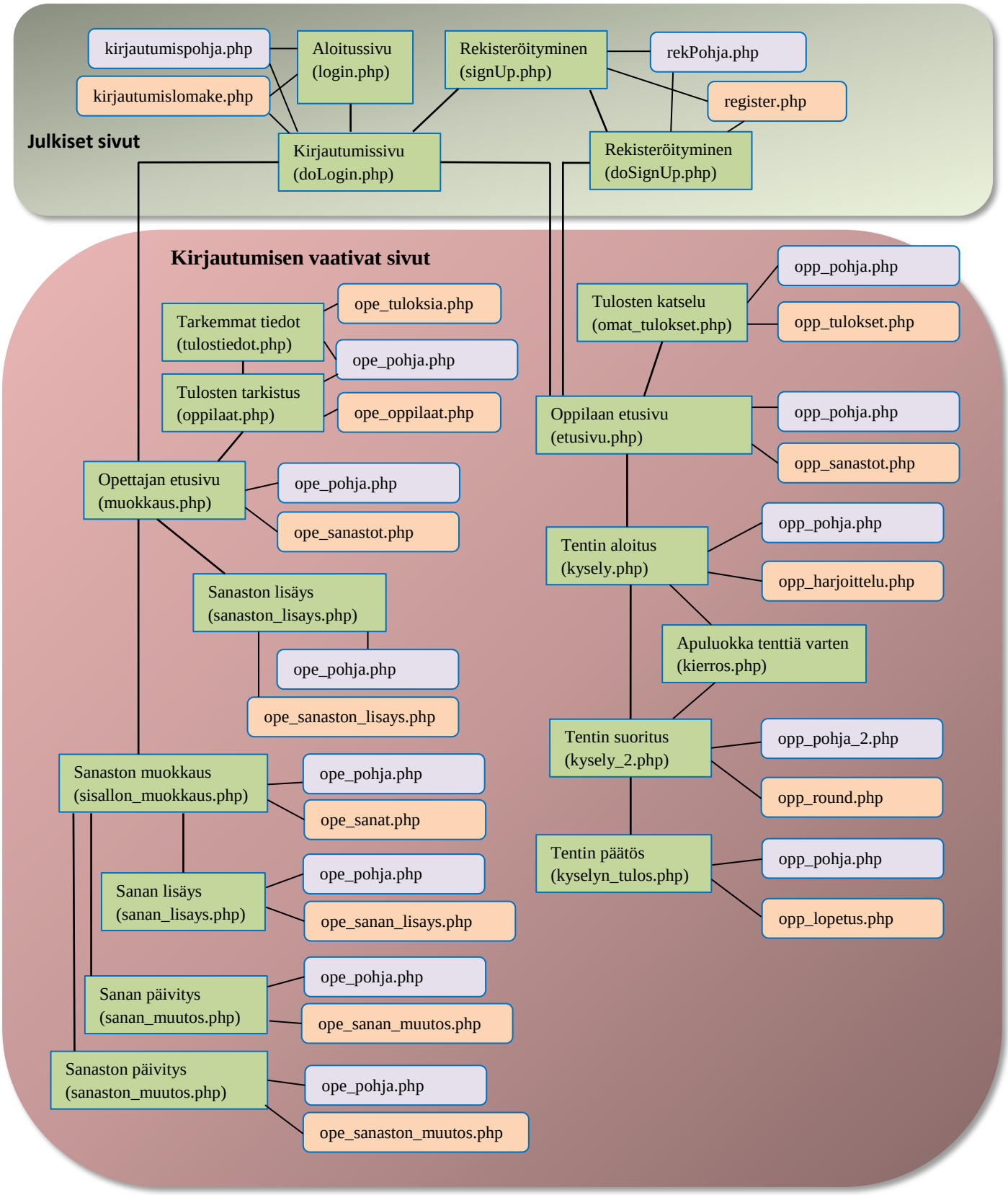
Sovelluksen toteutuksessa on käytetty MVC-mallin yleistä rakennetta. Kontrollerit sijaitsevat juurihakemistossa, näkymät hakemistossa ../views, yleiskäyttöiset apukirjastot ja muut tiedostot hakemistossa ../libs ja malliluokat hakemistossa ../libs/models. Tietokannan yhteystiedot ovat tiedostossa ..libs/tietokantayhteys.php ja yleiskäyttöiset metodit tiedostossa ..libs/common.php.

Tiedostojen nimet on kirjoitettu pienaakkosilla. Metodien ja muuttujien nimissä on käytetty camelCase-käytäntöä (jos nimi koostuu useasta sanasta, nimen keskellä olevat sanat alkavat isolla kirjaimella) ja nimissä olevat numerot on erotettu kirjaimista _alaviivalla.

Sovelluksessa on hyödynnetty globaalia $_SESSION-muuttujaa istunnonhallintaan ja olioiden välittämiseen kontrollerilta toiselle. Myös $_GET- ja $_POST-muuttujia on käytetty lomakkeiden tietojen välittämiseen näkymien ja kontrollerien välillä. Kaikki tietokantatransaktiot tehdään malliluokkien kautta. SQL-injektioilta on suojauduttu kierrättämällä käyttäjän syötteet common.php-tiedostossa sijaitsevan putsaaString()-metodin kautta. Lisäksi syötteiden soveltuvuus kuhunkin tarkoitukseen tarkistetaan onkoKelvollinen()-metodilla kussakin kantaa käyttävässä malliluokassa.

Näkymissä ylimääräistä HTML-koodin toistoa on vähennetty jakamalla näkymät pohjiin ja varsinaisiin sisältötiedostoihin. Pohjia voidaan tehdä erilaisia eri käyttötarkoituksiin; esimerkiksi opettajalle tarkoitetut sivut käyttävät tiedostoa ope_pohja.php ja oppilaille tarkoitetut sivut tiedostoa opp_pohja.php. Yleiskäyttöiselle naytaNakyma()-metodille annetaan kutsuttaessa parametreina käytettävien tiedostojen nimet.

6. Käyttöliittymä ja järjestelmän komponentit



7. Asennustiedot

Käytettävän palvelimen tulee tukea PHP:tä ja PostgreSQL-tietokantaa. Hakemistoille on annettava laajat lukuoikeudet ja läpikulkuoikeudet. Sen voi tehdä esimerkiksi komennolla

chmod -R a+X $HOME $HOME/<juurihakemiston_nimi>

Asenna sovellus kopioimalla sen tiedostot palvelimen Internetiin näkyvään hakemistoon (esimerkiksi htdocs-hakemisto). Määritä sen jälkeen tietokannan yhteystiedot oikein tiedostoon libs/tietokantayhteys.php. Tässä toteutuksessa tietokantayhteyden välittämiseen käytetään staattista muuttujaa $yhteys = new PDO('pgsql:'); jossa PDO-olion parametreiksi riittää

$yhteys->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);

PostgreSQL-tietokannan pystytykseen voit ajaa sql-kansiosta löytyvän tiedoston create-tables.sql, joka ajetaan toiminnassa olevan tietokannan komentorivillä komennolla psql <. Sql-kansiossa on myös taulujen purkutiedosto ja joitakin tiedostoja, joiden avulla tauluihin voi lisätä testauskäyttöön tarkoitettua sisältöä.

Sovelluksessa on käytetty relatiivista osoitteiden toteutusta, joten asennuskohteella ei ole sisäisen toiminnan suhteen väliä, kunhan hakemistopuu on kopioinnin jälkeen samarakenteinen kuin tässä työssä.

8. Käynnistys- /käyttöohje

Sovellus on kokeiltavana osoitteessa http://petesaar.users.cs.helsinki.fi/Sanakysely/login.php .
Esittelysivu löytyy osoitteesta http://petesaar.users.cs.helsinki.fi/Sanakysely/esittelysivu.html .

- Opettajan (pääkäyttäjän) tunnus ja salasana: **Ope, ope**
- Opiskelijan (peruskäyttäjän) tunnus ja salasana: **Erkki, passu**

Uudet peruskäyttäjät saavat tunnukset rekisteröitymisen kautta. Peruskäyttäjä voi tenttiä valitsemiaan sanastoja ja tarkastella omia tuloksiaan. Opettajan tunnuksilla pääsee muokkaamaan sisältöä ja katsomaan oppilaiden suorituksia.

Jos kirjautuu sovellukseen opettajana, voi luoda uuden sanaston painikkeella ’Lisää sanasto’ tai valita listalla näkyvän valmiin sanaston muokkaamista varten. Sanaston voi poistaa kyseisen rivin punaisella painikkeella. Kun sanasto on avattu muokattavaksi, sen nimeä, kuvausta ja kielivalintoja voi muuttaa painikkeella ’Päivitä tiedot’. Sanastoon liittyvät sanat näkyvät luettelossa, josta niitä voi poistaa ja valita muokattavaksi yksi kerrallaan. Sanan lisäykseen löytyy oma painikkeensa.

Oppilas saa tentin aloittaessaan lyhyet ohjeet kyselyn aikana käytettävistä painikkeista. Niiden käyttö on hyvin yksinkertaista ja pitäisi sujua ohjeita lukemattakin jo yhden harjoittelukerran jälkeen.

9. Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

Varsinaista luokkien ja metodien yksikkötestausta en ehtinyt toteuttaa, joten en voi tähän liittää samanlaisia testausmetodeja kuin aikaisemmin Java-ohjelmoinnin harjoitustyöhön. Vaikka täyttä rivikattavuutta ei tullutkaan, testasin toki sovellusta jatkuvalla systeemitestauksella kautta linjan. Matkan varrella tuli korjattua useita löytyneitä bugeja (esimerkiksi tyhjien sanastojen päätyminen oppilaan tentittäväksi) ja puutteita käytettävyydessä.

Sovellusta on testattu antamalla sille kaikenlaisia virheellisiä syötteitä ja myös suoranaisia SQL-injektioyrityksiä. Lisäksi tarkistin toiminnan yleisimmillä selaimilla (Firefox, IE, Chrome, Safari). Se näyttäisi toimivan tarkoituksenmukaisesti mainituilla alustoilla, mutta skaalautuvuudessa olen huomioinut vain tavallisimmat resoluutiot. Myös tabletilla ja kosketusnäytöllä sovellus on vielä käyttökelpoinen, mutta aivan kaikkia mobiililaitteita varten en edes lähtenyt käyttöliittymää suunnittelemaan; se voisi tietysti olla yksi jatkokehityksen kohde.

Kaikkien yksittäisten sanojen listausmahdollisuus ei ehtinyt mukaan tähän versioon. Olisi hyödyllistä, että opettaja (siis sanaston laatija) voisi listata kaikki tietokannassa olevat sanat aakkosjärjestyksessä ja siirrellä niitä sieltä vapaasti sanastosta toiseen. Myös sanoja pitäisi voida lisätä sisällyttämättä niitä mihinkään sanastoon.

Myöskin aika loppui kesken muutamien alun perin mielessä olleiden ominaisuuksien loppuun asti viemisessä. Esimerkiksi sanaan sisältyy attribuutti ’taivutus’, joka olisi sellaisen löytyessä hyvä esittää oppilaalle tentin yhteydessä (esim. ruotsin kielen sanan ’katt’ taivutusmuodot: en katt, katten, katter, katterna). Nyt oppilaalle kerrotaan lisätietona vain sanan sanaluokka.

Tämä versio on ajateltu lähinnä itseopiskelussa käytettäväksi. Jatkokehityksessä voisi pohtia lisää ominaisuuksia varsinaiseen opetuskäyttöön. Esimerkiksi nyt opettaja näkee listauksessa ainoastaan oppilaan sanastokohtaiset yrityskerrat ja parhaat tulokset, ei tarkkoja tietoja vääristä vastauksista. Myös tuloksen suoritusajankohta olisi tarpeellinen tieto siinä tapauksessa, että kyseistä sanastoa laajennetaan tai muutetaan myöhemmin.

Oppilaan kannalta voisi olla mukavaa, jos sovelluksessa olisi enemmän ’tekoälyä’ esimerkiksi antamaan väärään vastaukseen vinkkejä, oliko vastauksessa edes jotakin oikein. Tentin aikana voisi kenties erityisellä ’vihjepainikkeella’ saada pieniä apuja, vaikkapa kysyttävän sanan alkukirjaimen tai sanan pituuden tms.

Tässä harjoitustyössä otin tavoitteeksi CRUD-nelikoitten toteuttamisen vain tietokohteille ’sana’- ja ’sanasto’. Jatkokehityksessä kannattaisi sisällyttää käyttöliittymään myös järjestelmänvalvojan sivut, joilla voisi kätevästi hallinnoida käyttäjiä, sekä oppilaita että opettajia. Näin myös muut taulut saisivat ansaitsemansa crudit.

10. Omat kokemukset

Tämän harjoitustyön tekeminen oli samalla kertaa sekä hauskaa että haastavaa. Mukavaa oli uusien tekniikoiden oppiminen (PHP, PostgreSQL, MVC-malli) ja uurastuksen tulosten konkretisoituminen heti silmien edessä. Oppimateriaalin esimerkit antoivat hyvän pohjan omien ideoiden toteuttamiselle ja jäsensivät työn etenemistä.

Haastavaa oli aikataulussa pysyminen (muiden kurssien ohella) ja etenkin työn edetessä yhä laajenevan kokonaisuuden pitäminen hallussa. Idealtaan näinkin yksinkertainen sovellus vaati lopulta arvaamattoman paljon kontrollereita ja näkymiä, että käytettävyys säilyi siedettävänä ja työlle asetetut kriteerit tulivat täytettyä.

Kurssi oli sangen opettavainen ja harjoitustyön tekeminen antoi paljon ’kipinää’ uuden opiskeluun. Esimerkiksi ikivanhat HTML-oppini osoittautuivat aikansa eläneiksi; sen huomaa alkukiireessä valitsemastani tavasta suunnitella käyttöliittymän runko perustuen taulukkorakenteisiin, soluihin ja sarakkeisiin. Tällä aikataululla ei ollut mahdollista kunnolla opiskella nykyaikaista käytäntöä, jossa lähes kaikki muotoilu tapahtuu tyylitiedostojen avulla. HTML5:n ja CSS3:n tarjoamia uusia mahdollisuuksia tuli siis vain raapaistua tässä työssä.

Vaikka jouduin monissa kohdin tyytymään kompromisseihin, olen lopputulokseen suhteellisen tyytyväinen. Ehkä sitä kuvaa parhaiten se, että aion palata kurssin päättymisen jälkeenkin projektin pariin ja jatkaa sen kehittelyä vapaalla aikataululla ja rohkeampia kokeiluja mielessä.