|  |
| --- |
|  |
| Tietokantaohjelmointi |
| Harjoitustyö – ryhmä 1 |
| Äänestysjärjestelmä – Vaihe 2 |
|  |
|  |

|  |
| --- |
| Joni Järvinen  [joni.jarvinen@uta.fi](mailto:joni.jarvinen@uta.fi)  Lauri Vene  [lauri.vene@uta.fi](mailto:lauri.vene@uta.fi)  Ville Murtonen  [ville.murtonen@uta.fi](mailto:ville.murtonen@uta.fi)  28.04.2012 |

Sisällys

[Ohjelman ominaisuudet 1](#_Toc323486603)

[Työnjako 2](#_Toc323486604)

[Kuvaus toteutuksesta 2](#_Toc323486605)

[Ohjelman käyttö 2](#_Toc323486606)

[Muutokset 1-vaiheesta 3](#_Toc323486607)

[Oma arvio 3](#_Toc323486608)

# Ohjelman ominaisuudet

Tässä osuudessa kuvataan ohjelman ohjelmiston ominaisuudet.

* Välitulosten laskemisesta ja esittämisestä. Arvontoja ei tarvitse tehdä välitulosten laskemiseksi, vaan saman äänimäärän/vertailuluvun ehdokkaat voidaan sijoittaa välituloksissa keskenään määrittelemättömästi. Välitulosten laskeminen voi tapahtua joko automatisoidusti tai käyttäjän erillisestä pyynnöstä. (12p)
  + Välituloksia ei voi tällähetkellä esittää koska tulossivu jarjestysluvut.php suorittaa myös arvonnat. Pienellä muutoksella (jättämällä arvonnat pois), voitaisiin tosin näyttää myös välituloksia tämän sivun avulla.
  + Kyseinen sivusto siis näyttää jokaiselle liitolle sen saamat kokonaisäänimäärät ja tämän lisäksi asettaa ehdokkaat liiton sisällä järjestykseen, sekä kertoo ko. ehdokkaan liittovertailuluvun. Kyseinen sivusto suorittaa myös arvonnat jo tässä kohtaa mikäli kahdella ehdokkaalla on sama vertailuluku.
  + Vastaavasti sama sivu kertoo kyseiset tiedot myös renkaille.
* Toiminnosta, jolla ehdokkaiden/liittojen/renkaiden tiedot saa syötettyä tiedostoina ohjelmalle. (12p)
  + lisaa\_filesta.php sivun avulla voi käyttäjä uploadata listan joka sisältää opiskelijoita esiformatoidussa muodossa ”etunimi:sukunimi:opiskelijanumero”, tämän jälkeen ohjelma parsii tiedostosta rivi kerrallaan kantaan ja antaa virheilmoitukset mikäli kohtaa vääriä syötteitä. Toteutus on suppea koska se sisältää ainoastan opiskelijoiden syötön.
  + Vastaavalla koodipätkällä voitaisiin hyvin helposti syöttää myös liitot ja renkaat mikäli näin haluttaisiin. Tällöin vaan tiedosto olisi esiformatoitu muotoon ”nimi:tunnus” ja koodi sisältäisi sitten vastaavat tarkistukset kuin normaalin formin kautta syötettäessä.
* Tuesta usean käyttäjän samanaikaiselle käytölle (esim. kaksi ihmistä pystyisi syöttämään ehdokkaiden ääniä yhtä aikaa tai yksi pystyisi syöttämään ja toinen seuraamaan välituloksia tms.). (12p)
  + Tätä toiminnallisuutta ei ole testattu, joten emme voi tietää toimiiko se. Ei tosin ole tällähetkellä mitään syytä miksi tämä ei toimisi koska ohjelman rakenne perustuu siihen että äänestyspaikalta annetaan vain kokonaisäänet, näin ristiin meneviä luku/kirjoitus tapahtumia ei pitäisi tapahtua.
* Käyttöliittymästä ehdokkaiden, liittojen, renkaiden ja äänestyspaikkojen lisäämiseen ja käsittelyyn. (12p)
  + Opiskelija.html sivulta voi lisätä opiskelijoita järjestelmään
  + edustaja.php sivulta voi lisätä edustajia järjestelmään
  + lisaarengas.php sivulta voi lisätä renkaita järjestelmään
  + vaaliliitto.php sivulta voi lisätä vaaliliittoja järjestelmään
  + lisaa\_paikka.php sivulta voi lisätä äänestyspaikkoja järjestelmään
* Raportointitoiminnosta, jolla saa selville liittojen ja renkaiden paikkajakaumat (voit olettaa, että paikkoja olisi yhteensä 40). (12p)
  + Aanijakaumat.php sivulta voi valita näytettäväksi haluamansa liiton/renkaan paikkajakaumat
* Raportointitoiminnosta, jolla saa selville eri äänestyspaikoilla annettujen äänien määrän ja kuinka äänet ovat jakautuneet eri liittojen/renkaiden/ehdokkaiden välillä. (12p)
  + Raportti\_jakaumat.php sivulta voi valita äänestyspaikan ja nähdä äänijakauman tämän äänestyspaikan osalta
* Muista ominaisuuksista, kuten kehittynyt tietoturva. (12)
  + Kaikki syötteet on sanitoitu ja virheiden käsittelystä huolehdittu

# Työnjako

Ville Murtonen: Ohjelmointi / Dokumentointi

Lauri Vene: Testaus / Ohjelmointi

Joni Järvinen: Testaus / Dokumentointi

# Kuvaus toteutuksesta

Toteutus pohjautuu php + html koodiin postgresql tietokanta-alustalla. Kaikki syötteet hoidetaan form/post metodilla jonka käsitettystä huolehtii php. Suurinosa syötteistä viittaa script.php tiedostoon joka hoitaa suurimman osan koodin käsittelystä. Muutamat sivut kuten esimerkiksi Raportti\_jakaumat.php & Aanijakaumat.php sisältävät sekä form postauksen että sen käsittelyn, eli näissä tapauksissa sivusto viittaa itseensä.

SQL injektioilta on pyritty suojautumaan käyttämällä php:n omaa filter\_var funktiota jonka avulla kaikki syötteet on sanitoitu ennen kuin niiden pohjalta on tehty kyselyitä. Vastaavasti kaikki kyselyt sisältävät muuttujien perustarkistukset jotta esim. tyhjillä muuttujilla ei päästä suoritusvaiheeseen.

Koodissa on panostettu pääosin toiminnallisuuden toteuttamiseen eikä niinkään graafiseen ulkoasuun, eli suurinosa html-näkymistä on hyvin yksinkertaista ja sisältää lähinnä table-elementtejä. Tämä toteutusvalinta pohjaa siihen että kyseessä on tietokantaohjelmointiin keskittyvä kurssi, eikä niinkään www-toteutukseen pohjaava.

# Ohjelman käyttö

Ohjelman käyttö tapahtuu sivun index.php kautta johon on linkitetty kaikki oleelliset tapahtumasivut.

Ohjelman käyttöä voi testata esimerkiksi osoitteessa <http://www.cs.uta.fi/~vm92179/index.php> josta löytyy esimerkkitietokanta jolla voi suorittaa esimerkkitapahtumia.

Ohjelman käyttö lähtee rakentumaan opiskelija.html sivun kautta jonka avulla voidaan syöttää opiskelijoita kantaan. Sama toiminnallisuus voidaan hoitaa myös tiedoston avulla käyttäen lisaa\_filesta.php sivua.

Kun opiskelijoita on syötetty haluttu määrä, voidaan alkaa määrittää opiskelijoista edustajia. Tämä tapahtuu edustaja.php sivun kautta joka käytännössä ottaa opiskelijat taulusta opiskelijoita ja tekee niistä uusia rivejä edustaja-tauluun. Näin saamme opiskelijoista edustajia äänestykseen.

Tämän jälkeen voidaan syöttää vaaliliittoja ja vaalirenkaita vaaliliitto.php & lisaarengas.php sivuilta. Kun liittoja ja renkaita on syötetty haluttu määrä, voidaan kytkeä liittoja renkaisiin liita\_liitto.php sivulla.

Vastaavasti voidaan kytkeä edustajia liittoihin / renkaisiin liitaopiskelija.php sivun avulla.

Kun liitot / renkaat ja niiden sisällöt on kunnossa, voidaan syöttää äänestyspaikat. Näiden syöttäminen tapahtuu lisaa\_paikka.html sivun kautta. Kun paikat ovat kunnossa, voidaan aloittaa varsinainen ääntenlasku.

Tämä tapahtuu lisaa\_aanet.php sivun kautta jossa valitaan äänestyspaikka, edustaja ja tämän saamat kokonaisäänet paikalla. Tässäkohtaa siis vaaditaan että äänestyspaikalla suoritetaan jo manuaalista laskentaa. Tällä saavutetaan myös äänten anonymiteetti. Kun jokaisen edustajan saamat kokonaisäänet on saatu ja lisätty tämän sivun kautta, löytyvät ne paikka\_aanet taulusta jossa on listattu edusta,paikka ja saadut äänet.

Kun kaikki äänet on syötetty, viheltää ylläpito pelin poikki, ja poistaa syöttötunnuksilta oikeudet kirjoittaa kantaan (tätä ei voitu harjoitustyössä toteuttaa koska kannat olivat yliopiston ylläpitämät joten käyttöoikeuksia ei päästy tarpeeksi tehokkaasti muuttamaan). Näin estetään se että ääniä ei pääse enää jälkikäteen lisäämään/muuttamaan.

Seuraavaksi voidaankin laskea jo alustavia tuloksia jarjestysluvut.php sivun avulla. Tämä sivu käy kaikki liitot läpi ja laskee liiton jäsenille sijaluvut liiton sisällä sekä tästä seuraavat liittovertailuluvut. Tämän jälkeen järjestelmä läpikäy kaikki renkaat ja muodostaa näiden pohjalta vastaavat sijaluvut sekä rengasvertailuluvut. Kyseinen sivusto myös tulostaa nämä tiedot ruudulle ylläpidon katselmoitavaksi.

Tämän jälkeen voidaan siirtyä varsinaisten tulosten laskemiseen. Tässäkohtaa sivulla tulokset.php voidaan saada näytölle lopulliset tulokset jossa vertailulukujen pohjalta on arvottu myös lopulliset sijat mikäli liittoon/renkaaseen kuulumattomalla olisi ollut sama vertailuluku kuin niihin kuuluvalla. Näistä tuloksista kirjataan myös viralliset\_tulokset taulu johon tulokset kirjataan.

Tämän jälkeen voidaan vielä viralliset\_tulokset taulusta tulostaa erilaisia raportteja Aanijakaumat.php sekä Raportti\_jakaumat.php sivujen avulla.

# Muutokset 1-vaiheesta

Alkuperäisessä suunnitelmassa oli tarkoitus käsitellä äänestäjä ja ehdokas yhtenä tyyppinä, mutta päätimme ottaa ehdokkaan omaksi tyypikseen, jotta käsittely yksinkertaistuu, sillä ehdokasnumero toimii suoraan vierasavaimena eri relaatioissa. Päätimme myös luopua Ääni-tyypistä kokonaan, sillä se ei toiminut vaalisalaisuuden säilyvyyden kanssa. Korvasimme tämän laittamalla ehdokkaan suhteeseen äänestyspaikan kanssa ja suhteelle attribuutit ehdokas\_id, paikka\_tunnus ja lkm. Näin saavutimme sen, että äänestäjien identiteetti ei esiinny. Loimme myös alkuperäisestä suunnitelmasta poiketen erillisen viralliset\_tulokset-tyypin, joka ei ole suhteessa muihin tyyppeihin, vaan se johdetaan muista.

# Oma arvio

Työn toteutus sujui loppujenlopuksi järkevästi koska ymmärsimme aloittaa sen toteuttamisen ajoissa ja lisäksi meillä oli kohtuu järkevä suunnitelma jo apunamme. Ainoaksi ongelmaksi muodostui käytännössä järkevän sanitoinnin ja muutaman pidemmän SQL kyselyn muodostamien. Myöskään optimointia ei liiemmin suoritettu koska sen testaaminen olisi vaatinut aika paljon enemmän syötteitä kuin meillä oli käytössämme.

Puutteina työhön jäi parempi SQL injektioneilta suojautuminen, vaikka kaikki syötteet sanitoidaan, ei tämä silti ole paras tapa suojautua injektioneja vastaan. Lisääksi visuaaliseen ulosantiin olisi voinut panostaa enemmän sekä muutamia ominaisuuksia parantaa käytettävyyden näkökulmasta.

Myöskin useiden yhtäaikaisten syötteiden testaaminen jäi tekemättä joten tältäosin ei voida sanoa toimisiko koodi tälläisessä tilanteessa ja miten hyvin.