

Markus Hyvärinen

Tietokannan suunnittelu ja käytettävyys

Case Lapin Ajokoirayhdistys

Tietokannan suunnittelu ja käytettävyys

Case Lapin Ajokoirayhdistys

Markus Hyvärinen
Opinnäytetyö
Syksy 2015
Tietojenkäsittelyn koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma, Internet-palvelut ja digitaalinen media

Tekijä(t): Markus Hyvärinen

Opinnäytetyön nimi: Tietokannan suunnittelu ja käytettävyys -Case Lapin Ajokoirayhdistys

Työn ohjaaja: Sinikka Viinikka

Työn valmistumislukukausi- ja vuosi: Syksy 2015

Sivumäärä: 36+4

Lapin Ajokoirayhdistyksen toiminnan tavoitteena on edistää puhdasrotuisten ajokoirien kasvatusta ja jalostusta. Tavoitteen saavuttamiseksi yhdistys järjestää kursseja, valistus- ja esitelmätilaisuuksia, koiranäyttelyitä ja ajokokeita. Lisäksi yhdistys tarjoaa opastusta ja neuvoja koirien kasvattamisesta. Tämä tapahtuu alan kirjallisuuden tai mahdollisten kurssien välityksellä.

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa uusi tulostietokanta ja sen käyttöön tarvittava käyttöliittymä. Järjestelmän päätarkoituksena on tuoda ajokokeiden tulokset näkyville, jotta niitä voitaisiin hyödyntää ajokoirien jalostuksessa.

Tietoperustan ensimmäisessä osiossa käydään lyhyesti läpi ER-malli ja pyritään sen avulla selvittämään tietokannan rakenne ja kuinka se käytännössä toimii. ER-malli ja tietokannan rakenne eivät ole muuttuneet paljoa viime vuosina, joten suomenkielistä ajantasalla olevaa kirjallisuutta löytyi helposti. Tietoperustan toisessa osiossa käydään läpi mitä tarkoitetaan käytettävyydellä ja miten se vaikuttaa tietokannan käyttöliittymän suunnitteluun ja toteutukseen. Toisen osion tietoperustan aineistona pyrittiin käyttämään mahdollisimman uutta materiaalia ja tämän seurauksena suurin osa materiaalista oli englanniksi.

Opinnäytetyön tuloksena sain laadittiin Lapin Ajokoirayhdistykselle toimivan tietokannan ja käyttöliittymän minkä kautta tietoja voidaan käsitellä. Käyttöliittymään jäi parantamisen varaa, mutta se täyttää kuitenkin nykyisellään jo järjestelmälle asetetut vaatimukset.

Avainsanat: tietokanta, käytettävyys, käyttöliittymä, ER-malli

ABSTRACT

Oulu University of Applied Sciences
Business information systems, Internet services and digital media

Author(s): Markus Hyvärinen

Title of thesis: Database design and usability -Case Lapin Ajokoirayhdistys

Supervisor(s): Sinikka Viinikka

Term and year when the thesis was submitted: Autumn 2015 Number of pages: 36+4

Lapin Ajokoirayhdistys' main goal is to advance the breeding on pure blooded hounds. To achieve this goal the organization organizes courses, education and presentation opportunities, dog shows and hound tests. In addition to that the trust offers guidance and advices in matters related to dog breeding. This happens through literature and courses.

The objective of the thesis was to design and to build a new database and a user interface through which you can use the database. The systems main goal is to bring the results of hound tests to the users which they may use to advance hounds breeding.

First part of the knowledge base in the thesis is about ER-model and through explaining that we are introducing the structure for the database and it's functionality. Literature related to ER-model and basics for databases have not changed much in recent years. Thus there were plenty of literature in Finnish that could be used in the thesis. The second part of the knowledge base is about usability and how usability affects the design and creation of databases user interface. The material needed to fill this knowledge base needed to be as fresh as possible and because of that most of the material was in English.

As a product of the thesis I made a new database and a user interface to control it. The database and the user interface was made for Lapin Ajokoirayhdistys. The user interface was left with some room to improve on, but overall it meets all the requirements set for it.

Keywords: database, usability, user interface, ER-model

SISÄLLYS

1	JOHDANTO	6
2	TOIMEKSIANTAJA JA TOIMEKSIANTO.....	7
	2.1 Ajokokeet.....	7
	2.2 Vanha järjestelmä.....	9
	2.3 Toimeksianto	10
3	TIETOKANTA	13
	3.1 Fyysinen tietokanta	13
	3.2 ER-malli.....	14
	3.3 Lapin Ajokoirayhdistyksen tietokannan suunnittelu ja toteutus	19
4	KÄYTETTÄVYYS.....	22
	4.1 Käyttöliittymä	22
	4.2 SQL-komennot	24
	4.3 Tietotyypit.....	25
	4.4 Tietojen syöttäminen	28
	4.5 Tiedon muokkaamisen seuraukset.....	29
	4.6 Lapin Ajokoirayhdistyksen tietokannan käyttöliittymän toteutus	30
5	POHDINTA	34
	LÄHTEET.....	35
	LIITTEET	37

1 JOHDANTO

Suomenajokoira on yksi Suomen suosituimmista koirarodusta, jota enimmäkseen käytetään metsästyskoirana. Suomenajokoira on suomalaisen koiranjalostuksen tuote, jota moni sanoo maailman parhaaksi ajokoiraksi. Lapin Ajokoirayhdistyksen yksi merkittävimmistä tavoitteista on edistää puhtasrotuisten ja näin myös suomenajokoiran jalostusta, jotta suomenajokoira pysyy yhtenä parhaista ajokoiraroduista. (Lapin Ajokoirayhdistys 2015, viitattu 9.9.2015.)

Lapin Ajokoirayhdistyksen toimintaan kuuluu muun muassa ajokilpailujen järjestäminen, jonka avulla mitataan ajavien koirien jäniksen ajo-ominaisuuksia. Näiden ominaisuuksien avulla pystytään edistämään koirien jalostusta. Ajokokeiden tarkoituksena ei ole vahingoittaa riistaeläintä tai koiraa eikä alistaa niitä kohtuuttomalle rasitukselle. (Lapin Ajokoirayhdistys 2015.)

Mikäli ajo-ominaisuuksien tiedot eivät ole helposti saatavilla ei niistä silloin ole paljoa hyötyä koiran jalostuksessa. Lapin Ajokoirayhdistys julkaisee kaikki tulokset web-sivustollaan, jotta ajokoirien omistajat voivat seurata heidän omien koiriensa ja muiden omistamien koirien ajotaitoja ja löytää niistä ajokoiralle haluttuja periytyviä ominaisuuksia.

Lapin Ajokoirayhdistyksen vanha järjestelmä alkoi olla loppuun kulunut ja uusi järjestelmä oli tarpeen. Opinnäytetyön tavoitteeksi muodostui uuden käyttöliittymän suunnittelu ja toteutus minkä kautta kilpailutuloksia voitaisiin lisätä, muokata ja katsella. Tavoitteena oli, että uuden järjestelmän käyttö sekä myös ylläpito olisi helpompaa kuin vanhassa järjestelmässä.

2 TOIMEKSIANTAJA JA TOIMEKSIANTO

Opinnäytetyön toimeksiantajana toimiva Lapin Ajokoirayhdistys on perustettu 1988. Yhdistyksen toiminnan tavoitteena on edistää puhdasrotuisten ajokoirien kasvatusta ja jalostusta Lapin Kennelpiiri ry:n alueella. Lapin Ajokoirayhdistys toimii siis Lapissa ja on Lapin Kennelpiiri ry:n ja täten myös Suomen Ajokoirajärjestön jäsen. (Lapin Ajokoirayhdistys 2015.)

Yhdistys pyrkii toteuttamaan päämääräänsä erilaisten tapahtumien kautta. Yhdistyksen järjestämiin tapahtumiin kuuluvat kurssit, valistus- ja esitelmätilaisuudet, koiranäyttelyt ja ajokokeet. Lisäksi yhdistys tarjoaa opastusta ja neuvoja koirien kasvattamisesta. Tämä tapahtuu alan kirjallisuuden tai mahdollisten kurssien välityksellä. (Lapin Ajokoirayhdistys 2015.)

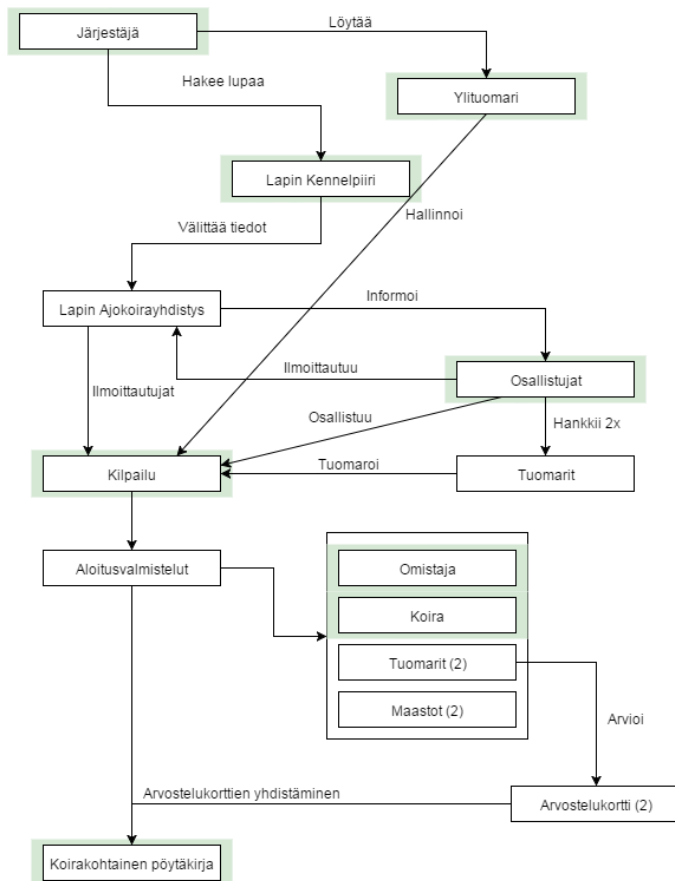
Lapin Ajokoirayhdistyksen tarkoitusperien saavuttamiseen on tärkeää olla jokin media, jonka kautta saadaan informoitua koiranomistajia kilpailuista ja näyttelyistä helposti ja vaivattomasti. Tärkeänä osana tätä informointia ovat yhdistyksen web-sivustot, joiden kautta sivustolla vierailevat voivat tarkastella tulevia kilpailuja ja näyttelyitä sekä menneiden kilpailujen tuloksia. Lisäksi sivusto toimii hyvänä alustana, jonka avulla voidaan informoida sivustolla vierailijoita mahdollisista uutisista ja muutoksista.

2.1 Ajokokeet

Ajokokeiden (Ajok) tarkoituksena on ajavien koirien jäniksen ajo-ominaisuuksien selville saaminen jalostusta varten, harrastajien ja heidän yhteistoimintansa kehittäminen sekä kilpailumahdollisuuksien tarjoaminen näihin sääntöihin pohjautuen. Ajokokeissa ei ole tarkoitus vahingoittaa riistaeläintä tai koira, eikä altistaa niitä kohtuuttomalle rasitukselle. Ajokokeet voivat olla luonteeltaan yleisiä, yhdistyksen jäsenten välisiä, rotujärjestön alaisten rotujen kokeita tai kokeita joihin osallistumisesta määrätään erillisillä kilpailusäännöillä ja ohjeilla. (Ajokokeen (Ajok ja Beaj) säännöt) Lapin Ajokoirayhdistys järjestää myös vastaavia tapahtumia, joissa seurataan ajokoirien ketun ajotaitoja (Keaj). (Lapin Ajokoirayhdistys 2015.)

Kun metsästysseura tai vastaava haluaa järjestää kilpailun, täytyy sen yleensä ensin löytää ylituomari valvomaan kilpailua, jotta kilpailun oikeellisuus voidaan varmistaa. Saatuaan tuomarin täytyy järjestäjän hakea lupa kilpailun pitämiseksi. Luvan hyväksymisestä vastaa paikallinen kennelpiiri (Lapissa luvan myöntämisestä vastaa Lapin Kennelpiiri). Mikäli lupa kilpailulle myönnetään, laitetaan tieto eteenpäin siitä tässä tapauksessa Lapin Ajokoirayhdistykselle. Saatuaan tiedon kilpailusta Lapin Ajokoirayhdistys lisää kilpailun ja sitä koskevat tiedot heidän tietokantaansa ja täten myös web-sivustollensa. Lisäksi he informoivat tarvittaessa koiranomistajia kilpailusta muiden keinojen kautta. Kun ajokoiran omistaja haluaa osallistua kilpailuun, täytyy hänen itse löytää tuomarit (2) arvioimaan koiran suoritusta. Löydettyään tarvittavat tuomarit kilpailulle koiranomistaja voi ilmoittautua kilpailuun paperi- tai sähköisellä lomakkeella. Ilmoittautuminen tallennetaan tämän jälkeen manuaalisesti Lapin Ajokoirayhdistyksen tietokantaan. (Lapin Ajokoirayhdistys 2015.)

Kilpailutoiminta vaihtelee riippuen siitä onko kyseessä niin sanottu pitkäkoe vai normaalikoe. Pitkällä kokeella tarkoitetaan koetta, joka kestää noin kaksi viikkoa. Kahden viikon aikana koiranomistaja ja tuomarit voivat sopia keskenään ajan ja koemaaston, jossa toimitaan. Sovittuaan nämä seikat voivat he suorittaa ajokokeen. Normaalisissa kokeissa kaikki ajokoirien omistajat ja tuomarit kokoontuvat sovitulle paikalle, jossa ylituomari pitää lyhyen puhuttelun ja arpoa tuomarit sekä koemaastot koiranomistajille. Tämän jälkeen koiranomistaja, koira ja kaksi tuomaria menevät näihin arvottuihin koemaastoihin ja kilpailu aloitetaan ylituomarin määräämään aikaan. Ajokoiralle annetaan tietty määrä haku-aikaa ja ajo-aikaa yhteensä. Tämän ajan käyttöä valvovat tuomarit, jotka myös kirjaavat koiran havainnot ja käyttäytymisen arviointikortille (Liite 1 ja Liite 2). Koe koostuu yleensä kahdesta erästä, jotka suoritetaan eri maastossa. Näin voidaan tasata maaston tuomia etuja ja haittoja. Tämä kilpailuprosessi voidaan kuvata olla olevan kuvion mukaisella tavalla. (Lapin Ajokoirayhdistys 2015.)



Kuvio 1. Kilpailun järjestäminen. Vihreällä pohjalla olevat tiedot tallennetaan tietokantaan.

Suorituksen päätyttyä molempien tuomareiden lomakkeet toimitetaan ylituomarille ja hän käy tuomareiden kanssa läpi kortin ja siihen tehdyt merkinnät. Tuomareilla voi olla usein eriäviä mielipiteitä siitä miten koira käyttäytyi. Koira arvesteltaessa, otetaan keskiarvo näiden kahden tuomarin antamista arvioista kirjattaessa ne koirakohtaiseen pöytäkirjaan (Liite 3 ja Liite 4). Kun kaikki tulokset on käyty läpi ja pisteet on laskettu, julkaistaan tulokset kaikkien tietoon, jotta voidaan palkita parhaiten pärjänneet osallistujat. Tuloslomakkeet, joihin on kirjattu molempien tuomareiden korteista yhdistetyt tiedot toimitetaan Lapin Ajokoirayhdistykselle, joka lisää ne omaan tietokantaansa. (Lapin Ajokoirayhdistys 2015.)

2.2 Vanha järjestelmä

Yhdistyksen vanha web-sivusto on laadittu 1999 ja on tästä johtuen hieman aikaansa jäljessä niin toiminnallisilta kuin myös käytännön ominaisuuksiltaan. Vanhalla sivustolla toimiva järjestelmä täyttää yhdistyksen perustarpeet Keaj-kokeita lukuunottamatta, mutta järjestelmän ja sivuston

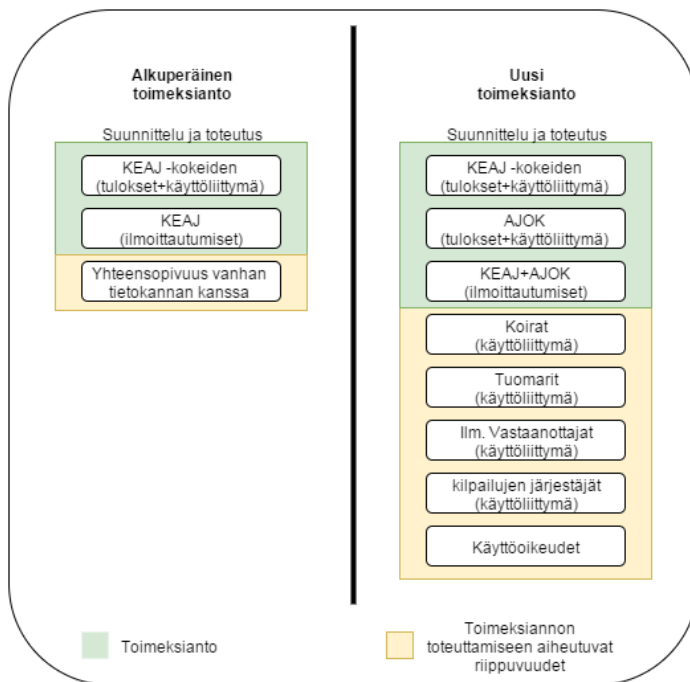
päivittäminen ja ylläpito vaativat ohjelmoinnin tuntemusta, joten päivittäminen on hidasta ja vaivalloista.

Vanha sivusto on laadittu käyttäen .asp koodausta ja tietokannat, jotka varastoivat tarpeellisen tiedon on laadittu Access-tietokantaa käyttäen. Vanhaa sivustoa laadittaessa ei ole otettu huomioon, sitä että sivua käytettäisiin 15 vuotta. Itsessään se, että sivustoa on pystytty käyttämään näin kauan osoittaa, että sivusto on aikanaan rakennettu huolellisesti. Tästä huolimatta sivustolle ja tietokantaan aiheutuvat muutokset ovat verottaneet järjestelmää ja kyseinen sivusto on tullut elinkaarensa päähän. Lukuisat pienet päivitykset ovat tehneet tietokannasta sekavan ja monimutkaisen. Sivuston laatimisesta tai siihen tehdyistä päivityksistä ei ole saatavilla dokumentaatiota ja koodia ei ole kommentoitu tekstissä ollenkaan. Tämä tekee koodin lukemisesta monimutkaista ja lähes mahdotonta.

Järjestelmän ikä ilmenee siinä tyylissä miten se kerää käyttäjiltä syötteitä. Syötteiden kerääminen tapahtuu suoraan valikko-käyttöliittymän kautta, jossa käyttäjältä kysytään yksi tai useampi kysymys kerralla. Tämän jälkeen esitetään käyttäjälle seuraava kysymys ja niin edelleen. Kun käyttäjä haluaa muokata yhtä kenttää taulussa, täytyy hänen siis navigoida useiden eri kysymysten läpi, jotta hän pääsee muokkaamaan yhden kentän arvoa. Nykyisissä tietokantojen käyttöliittymissä lähestytään enemmän graafista käyttöliittymää, joka on helpompi ja nopeampi käyttää.

2.3 Toimeksianto

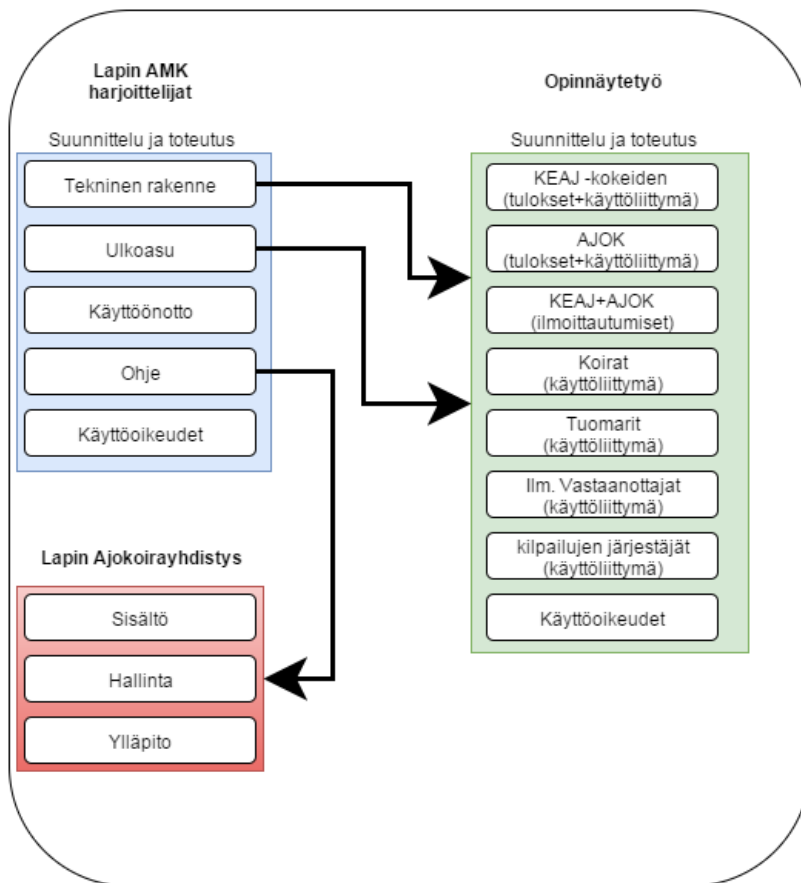
Alkuperäisenä tehtävänantona oli suunnitella ja toteuttaa uusi tulostietokanta, jota voitaisiin käyttää ketunajokokeiden tulosten tallentamiseen. Tilanteen tarkastelun jälkeen kävi nopeasti kuitenkin ilmi, että kyseistä tehtävänantoa ei olisi pystytty toteuttamaan tyydyttävällä tavalla käyttäen yhdistyksen silloista järjestelmää. Tästä seurasi se, että tehtävänantoa täytyi laajentaa kuvio 1. mukaiseksi. Saamani toimeksianto laajentui koskemaan myös jo olemassa olevien tietokantojen sekä niiden käyttöliittymien päivittämistä uusiin.



Kuvio 2. Toimeksianto

Työnjako

Lapin Ajokoirayhdistyksen alkuperäisenä tavoitteena oli päivittää ainoastaan web-sivusto uudemman näköiseksi. Tämä kuitenkin myös laajeni heidän näkökulmastaan koskemaan sivuston ja tämän myötä tietokannan täysin uudelleen rakentamista alusta lähtien. Tämä aiheutti sen, että tehtävä olisi liian laaja yhden henkilön toteutettavaksi, joten tehtävä jaettiin kahtia. Tehtävänjako suoritettiin kuvio 3. osoittamalla tavalla. Lapin Ajokoirayhdistys sai Lapin Ammattikorkeakoululta kaksi harjoittelijaa, joiden vastuulle jäi web-sivusto ja siihen liittyvät asiat. Tehtäväkseni jäi siis tietokannan ja sen käyttöliittymän suunnittelu ja toteutus. Tehtäväkseni jäi useiden taulujen suunnittelu ja toteutus. Sivuston sisällön tuottaminen, hallinta ja ylläpito jäisivät Lapin Ajokoirayhdistykselle.



Kuvio 3. Suunnittelun ja toteutuksen työnjako.

3 TIETOKANTA

Tietokanta on työkalu, joka varastoi tietoa. Tämä työkalu sallii jossakin muodossa tiedon luonnin, lukemisen, päivittämisen ja poistamisen. (Stephens 2009, 5.)

3.1 Fyysinen tietokanta

Tietokannan määritelmän alle voidaan laskea kuuluvaksi monia eri tietokantaratkaisuja. Eri tietokantatyypeillä on omat hyötynsä ja haittansa. Fyysinen tietokanta voi olla saaduista käyntikorteista laadittu nippu, tämä täyttää kaikki tietokantaa koskevat vaatimukset. Nippuun voidaan lisätä kortteja, niitä voidaan lukea nipusta, ne voidaan vaihtaa uusiin käyntikortteihin ja ne voidaan helposti poistaa nipusta. Kun käyntikorttien määrät ovat vähäisiä voidaan niitä käsitellä helposti vaikka ne olisivat fyysisiä. Tärkeimmät tai useimmin käytetyt käyntikortit voidaan tuoda nipun päälle ja vastaavasti vähiten käytetyt voidaan viedä nipun pohjalle. Käyntikorttien määrän kuitenkin kasvaessa tämän tietokannan käytettävyys heikkenee huomattavasti. Arkistokaapissa tai vastaavassa olevat tiedot on helpommin käytettävissä ja tähän ratkaisuun mahtuu huomattavasti suurempi määrä tietoa. Tietojen järjestelytavasta riippuen eri tietojen hakeminen arkistokaapista voi olla vaikeaa ja osittain lähes mahdotonta. Näissä tapauksissa täytyy jokainen arkistossa oleva paperi käydä erikseen läpi ja etsiä siitä haluttu tieto. Tämän etsinnän toteuttaminen on työlästä ja aikaa vievää. (Stephens 2009, 5.)

Ihmisen aivotkin omalla tavallaan varastoivat tietoa tietokantaan. Tavallaan ihminenkin suorittaa etsintä pyyntöjä tiettyjen avainsanojen tai vastaavien avulla kun hän ajattelee. Ihmisen aivoihin mahtuvan tiedon määrä on suuri, mutta sillä on tapana unohtaa tietoja, jolloin tietokannassa olevat tiedot menettävät luotettavuutta ja saattavat kadota kokonaan ajan kuluessa. Lisäksi aivot väsyvät ja kun aivot ovat väsyneet sen sisältämästä tiedosta tulee epätarkempaa. (Stephens 2009, 5–6.)

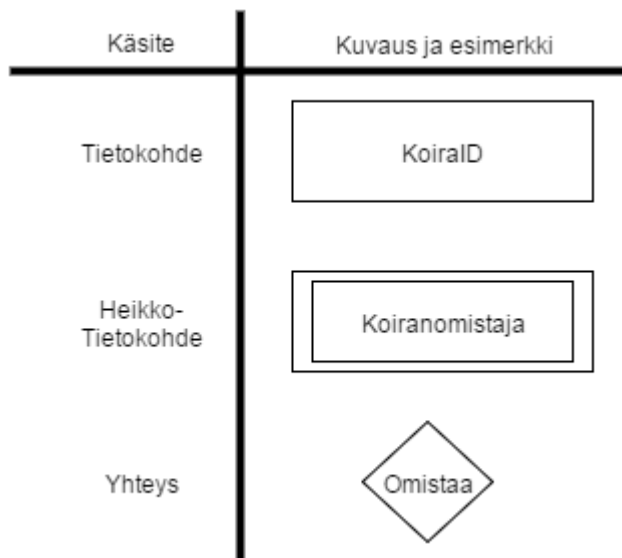
Fyysisellä tietokannalla on rajoituksensa, mitä laajempi tietokannasta tulee sitä heikompi fyysisestä tietokannasta tulee. Fyysinen tietokanta voi olla kuitenkin nykyään vielä validi ratkaisu kun tietokannat ovat pieniä ja yksi tärkeimmistä etuuksista on, että niitä on helppo käyttää ja ne ovat halpoja. (Stephens 2009, 6.)

3.2 ER-malli

ER-malli (Entity-Relationship model) määrittelee tietokannan tietokohteiden ja niiden välisten keskenäisten suhteiden avulla. Mallissa ei kuvata yksittäisiä suhteita vaan keskitytään tarkastelemaan kohteita tyyppinä. (Murthy 2008, 95.)

Tietokohteet

Tietokohteilla (Entities) tarkoitetaan niitä keskeisiä objekteja joiden keskeistä tietoa kerätään. Nämä kerättävät tiedot ovat yleensä kiinnostuksen kohteita tietokannan käyttäjälle. Näiden tietokohteiden ja niiden yhteyksiä voidaan kuvata kuvio 4. osoittamalla tavalla. Heikolla tietokohteella tarkoitetaan tietokohdetta, jota ei voida erottaa toisistaan pelkästään sen saamien arvojen avulla. Tämä kuvataan ympäröimällä tietokohde kahdella suorakulmiolla. Vastaavasti kun kyseessä on tietokohde, joka voidaan eritellä toisistaan sen saamien arvojen perusteella, sitä kuvataan ympäröimällä tietokohde yhdellä suorakulmiolla. Tietokohteiden välille muodostuvien yhteyksien kuvaamiseen käytetään timanttia eli kulmallaan seisovaa neliötä. (Lightstone, Nadeau & Teorey 2005, 14.)



Kuvio 4. Tietokohteiden kuvaus.

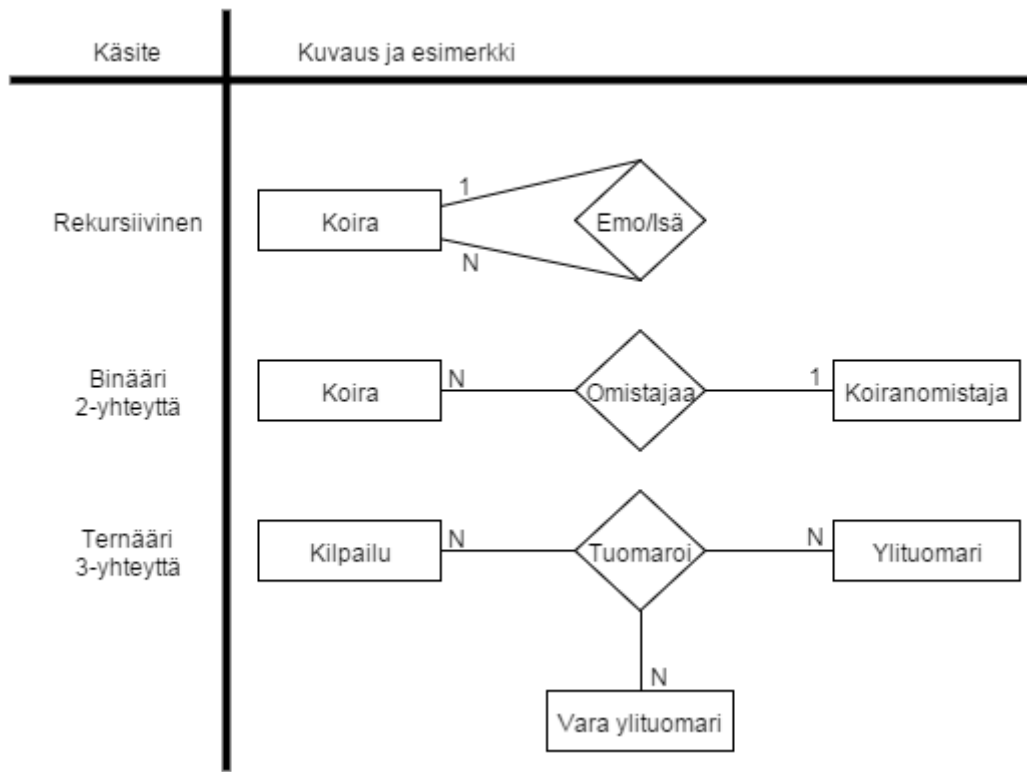
Yhteydet

Tietokannan yhteydet (Relationships) kuvaavat todellisia yhteyksiä yhden tai useiden eri tietokohteiden välillä. Näitä yhteyksiä yleensä kuvataan kolmen muuttujan avulla; yhteyksien aste (degree/cardinality), yhteydet (connectivity) ja niiden olemassaolo (existence). Tietokohteiden välille muodostuvia yhteyksiä kutsutaan rooleiksi (role), näiden roolien avulla pyritään selvemmin kuvaamaan tietokohteiden välistä suhdetta. Kuten kuviosta 4. näkyy, niin aina kuitenkin tietokohteiden välille ei tarvitse erikseen kuvata roolia, jotta voidaan määritellä tietokohteiden suhde selvästi. (Lightstone ym. 2005, 14–16.)

Kun tietokannoissa halutaan määritellä yhteyksiä taulujen välille, täytyy taulussa olla pääavainkenttiä (Primary key) tai uniikkeja kenttiä (Unique). Pääavainkenttä on yleensä juokseva numero, jolla taulun jokainen rivi saadaan yksilöityä ja täten erotettua muista riveistä. Tätä ominaisuutta tarvitaan kun halutaan viitata taulun tiettyyn riviin. Pääavainkenttiä voi jokaisessa taulussa olla vain yksi. Uniikeilla kentillä tarkoitetaan kenttiä, jotka saavat avainkentän tapaan muista eroavia arvoja ja näiden toisistaan eroavien arvoja käyttäen voidaan viitata taulun riveihin, mutta toisin kuin pääavainkenttiä uniikkeja kenttiä voi olla samassa taulussa useita ja näiden avulla voidaan luoda useita toisistaan riippumattomia yhteyksiä. Lisäksi tietokohteet voivat olla indeksejä (Index) tai tekstiä (Full text). Näitä käytetään kun kyseessä on laaja tietokanta, jonka tekemiin kyselyihin tarvitaan indeksointi, jotta ne toimisivat tarpeeksi nopeasti. Tässä opinnäytetyössä tietokanta on niin pieni että indeksoinnille ei ole tarvetta, joten sivuutamme sen käsittelyn. (W3Schools 2015, viitattu 9.9.2015.)

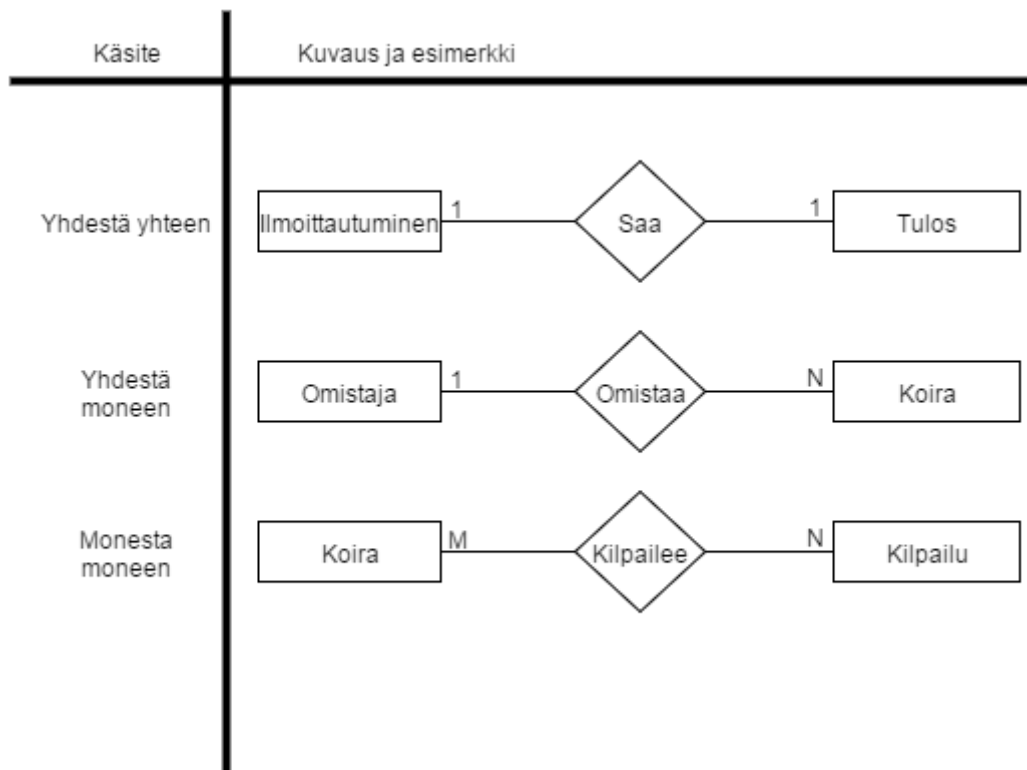
Yhteyksien aste on tapahtumien määrä yhteen tietokohteeseen, joka liittyy tai on linkitetty tapahtumien määrään toisessa tietokohteessa. Kuvio 5. on kuvattu muutamia esimerkkejä yhteyksien asteesta. Yleisin yhteys-aste on binäärinen eli kaksi tietokohdetta on yhteydessä toisiinsa eli niiden välillä on jokin suhde jota ne noudattavat. Mikäli binäärisillä yhteyksillä ei pystytä kuvaamaan yhteyttä tarpeeksi hyvin täytyy tällöin ottaa siihen lisää tietokohteita, jotta jo olemassa olevien tietokohteiden sisältö voidaan kuvata yksiselitteisesti. Laatiessa tietokantaa on kuitenkin tärkeää ottaa huomioon että kahdella tietokohteella voi olla keskenään useita binäärisiä yhteyksiä. Yhteydet voivat myös olla rekursiivisia, jolloin jokin taulun arvoista voi viitata saman taulu toisen rivin saamaan arvoon. Esimerkiksi jokin tietokannassa oleva koira voi olla toisen koiran emo tai isä. Yhteydet voivat myös olla terniäärisiä, jolloin kolmea taulua yhdistää yksi

yhteys. Terniäriset yhteydet ovat harvinaisia ja yleensä ne pystytään kuvaamaan käyttäen binäärisiä yhteyksiä..(Lightstone ym. 2005, 16.)



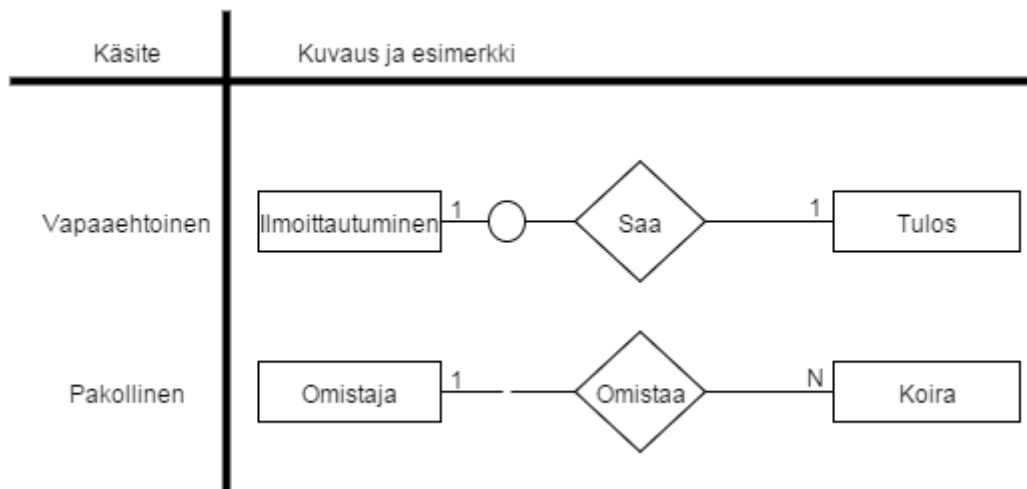
Kuvio 5. Yhteyksien aste.

Tietokannan yhteyksillä tarkoitetaan niitä rajoitteita, jotka kuvaavat sitä kuinka usein tietokohteet esiintyvät yhteydessä. Yhteyksien arvot voivat olla joko “one” (yksi) tai “many” (monta). Tietokohteiden välillä täytyy olla vähintään yksi yhteys, jotta ne ovat keskenään yhteydessä, mutta yhteyksiä voi olla monia. Nämä yhteyksien arvot määrittelevät sitä kuinka moneen tietueeseen nämä viittaavat toisessa tietokohteessa. Tätä voidaan kuvata kuvio 6. osoittamalla tavalla. Mikäli yksi tietue viittaa vain yhteen tietueeseen on kyseessä 1:1 yhteys. Vastaavasti kun kyseessä on monta yhteyttä jotka viittaavat yhteen tietokohteeseen tätä kutsutaan 1:N suhteeksi. Joissakin tapauksissa yhteyksissä viitataan tiettyyn määrään tietueita, kyseessä voi olla esimerkiksi kilpailu, jonka koko on säännöissä määrätty eikä täten voi ylittää tiettyä määrää. Näissä tapauksissa kun tiedetään kuinka moneen tietueeseen viitataan kutsutaan tätä numeroa “cardinality” nimityksellä. (Lightstone ym. 2005, 19.)



Kuvio 6. Yhteyksien suhde.

Tietokannoissa tietokohteen olemassaolo voi olla joko pakollinen tai vapaaehtoinen. Tietokohteen täytyy olla olemassa mikäli "one" tai "many" puolen tietokohteen täytyy olla aina olemassa. Ja vastaavasti mikäli tietokohdetta ei välttämättä tarvita on yhteys vapaaehtoinen. Tätä voidaan kuvata kuvio 7. osoittamalla tavalla. Kuviossa vapaaehtoista yhteyttä kuvataan ympyrällä, joka merkitään tietokohteita yhdistävälle viivalle sen tietueen puolelle, jota ei välttämättä tarvita. Tämä kuvaa sitä, että minimimäärä yhteyksiä on nolla. Vastaavasti kun yhteyden olemassa olo on pakollinen, kuvataan tätä pienellä tyhjällä välillä yhteyttä osoittavassa viivassa. (Lightstone Lightstone ym. 2005, 20.)



Kuvio 7. Tietokohteiden olemassaolo.

Attribuutit

Tietokannan attribuuteilla (Attributes) tarkoitetaan niitä ominaisuuksia, joita tietokohteet omaavat ja kuvaavat näitä. Tietokannassa attribuuteille annetaan arvo, joka kuvastaa jotakin tietokohteen ominaisuutta. Tietokannassa on kahdentyyppisiä attribuutteja. On attribuutteja, jotka auttavat erittelemään tietokohteet toisista ja muista saman tietokohteen tiedoista, tätä kutsutaan identifiseriksi tai avain-tietueeksi (key). Lisäksi tarvitaan arvoja kuvaamaan tiettyjä ominaisuuksia, joita tietue voi saada. Samoja ominaisuuksia voi olla useampia saman tietokohteen sisällä. Näitä attribuutteja kutsutaan kuvaaviksi (descriptors) attribuuteiksi. (Lightstone ym. 2005, 15.)

Attribuutit voidaan määritellä yksi- tai moniarvoisiksi. Yksiarvoisilla attribuuteilla voi olla kunkin kohteen kuvauksessa enintään yksi arvo. Moniarvoisella attribuutilla taas voi vastaavasti olla useita samanaikaisia arvoja. (Lightstone ym. 2005, 15.)

3.3 Lapin Ajokoirayhdistyksen tietokannan suunnittelu ja toteutus

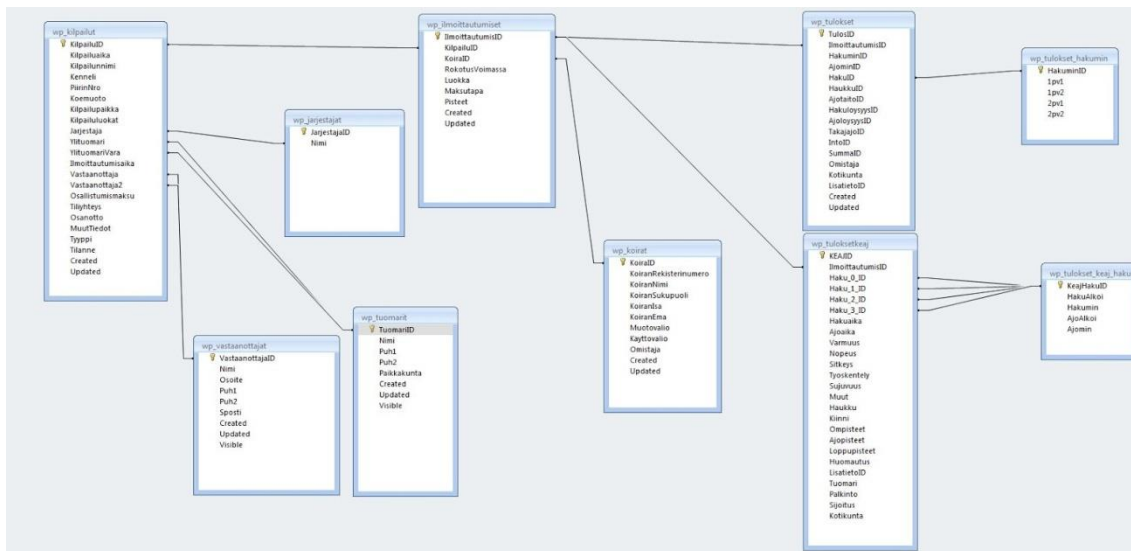
Lapin AMK:n harjoittelijat valitsivat web-sivuston alustaksi Wordpress-ohjelmiston. Wordpress vaatii toimiakseen SQL-tietokannan, joten tuntui luonnolliselta, että uudesta tietokannasta tulisi tehdä SQL-tietokanta. Tällöin pystyttäisiin käyttämään Wordpressin tarjoamia ominaisuuksia. Wordpressin ominaisuuksien avulla käyttäjien ja käyttöoikeuksien hallinta olisi helpompaa kun ne voitaisiin hoitaa Wordpressin kautta. Lisäksi kaikki tarvittavat tiedot pystyttäisiin laittamaan yhteen tietokantaan ja jatkossa järjestelmän siirtäminen toiselle palvelimelle olisi mahdollisimman yksinkertaista.

Tietokannan rakenteen hahmottelun aloitin kun olimme pitäneet ensimmäisen tapaamisen asiakkaan kanssa. Tapaamisessa asiakas esitteli miten heidän tällöinen järjestelmä toimi ja mitä hän halusi saada irti uudesta järjestelmästä. Tämän lisäksi hän antoi tarvittavat tunnukset sivustolle, jotta pystyisin testailemaan että miten järjestelmä toimi. Testaamisen helpottamiseksi sain käyttöohjeet järjestelmään. Ohjeet olivat lähes liian yksinkertaiset ja niissä käytiin askel kerrallaan kuinka lisätä, muokata tai poistaa tietokannassa olevia tietoja.

Tietokannan suunnittelussa auttoi suuresti se, että vanhaan tietokantaan oli pääsy. Tällöin pystyin näkemään millaisia syötteitä mihinkin kenttään on laitettu. Tämä helpotti käyttäjän syötteiden määrittelyä. Harmillisesti edellinen tekijä ei ollut tehnyt minkäänlaista dokumentaatiota tietokannasta, joten yhteyksien löytäminen siitä oli painajaismaista, mutta onnekseni iso osa yhteyksistä oli itsestään selviä.

Rakenne muuttui huomattavasti kun tapasin asiakkaan kanssa myöhemmin uudestaan. Tällöin kävi ilmi että useat asiat vanhassa tietokannassa olivat turhia ja niiden tietojen syöttäminen olisi ollut työlästä käyttäjälle. Onneksi nämä asiat tulivat ilmi ja saimme tietokannan rakenteen oikeille raiteille jo tässä vaiheessa. Mikäli rakenteen ongelmat olisivat ilmentyneet myöhemmin olisi tästä aiheutunut merkittäviä ongelmia. Tietokannan rakenne muodostui lopuksi kuvio 8. mukaiseksi. Kuviossa oleva wp_kilpailut taulu voidaan ajatella olevan tietokannan oleellisin taulukko, joka yhdistää taulut toisiinsa. Tähän tauluun tallennetaan kilpailua koskevat tiedot. Tähän tauluun suorien yhteyksien avulla siihen sisältyy siten myös wp_vastaanottajat, jotka ottavat vastaan kilpailuilmottautumiset, wp_tuomarit, johon on listattu Lapissa toimivat ylituomarit, joita kilpailuun tarvitaan kaksi sekä wp_järjestäjät, johon on tallennettu Lapissa toimivat kilpailujen järjestäjät. Ilmoittautumista tallennettaessa lisäys tehdään wp_ilmoittautumiset tauluun, johon tallennetaan

wp_kilpailut taulun KilpailuID käyttäen kilpailu, johon osallistutaan. Lisäksi tauluun tallennetaan muutamia osallistumista koskevat tiedot, kuten onko rokotus voimassa ja miten osallistumismaksu maksetaan. Kyseessä on ajokoe, joten luonnollisesti kilpailuun täytyy osallistua koira. Kilpailuun osallistuva koira tallennetaan käyttäen wp_koirat taulun avainkenttää. Mikäli kyseessä on ajokoe, tallennetaan tiedot wp_tulokset tauluun. Jos kyseessä on ketunajokoe, tallennetaan tiedot wp_tuloksetkeaj tauluun. Molempien taulujen rakenne on pohjimmiltaan samankaltainen. Tauluihin tallentaa ilmoittautumisID, jonka avulla voidaan yksilöidä, että mitä ilmoittautumista vastaa tämä kyseinen tulos. Näihin tulokset tauluihin tallennetaan tulokset käyttäen lukuisia alatauluja. Näitä käyttämällä järjestelmää on helpompi päivittää jatkossa, kuin se että kaikki tiedot olisi tallennettu yhteen tauluun.



Kuvio 8. Suunniteltu Lapin Ajokoirayhdistyksen tietokannan rakenne.

Tietokantaa ja käyttöliittymää suunniteltaessa yksi suurimmista kysymyserkeistä oli se, että tulisiko järjestelmässä olevia tietoja muokkaamaan loppukäyttäjät, jotka ovat rekisteröityneitä sivustolle vai tulisiko tämän olla suljettu, jolloin järjestelmän tietoja pystyisi muokkaamaan vain valikoidut henkilöt. Loppupäätös oli se, että asiakas eli Lapin Ajokoirayhdistys päätti, että pysytään nykyisessä ilmoittautumismenettelmissä eli ilmoitusten vastaanottaja pystyisi lisäämään tietoja tietokantaan ja loppukäyttäjät ilmoittaisivat hänelle sähköpostitse tai puhelimitse kun haluavat ilmoittautua. Järjestelmän suunnittelussa otin kuitenkin huomioon, että myöhemässä vaiheessa käyttäjät voitaisiin lisätä järjestelmään. Tämä voitaisiin toteuttaa tulevaisuudessa helposti katsomalla koiran omistaja ja vertaamalla sitä käyttäjätunnuksen nimi kenttään. Mikäli nämä ovat samoja, pystyisi käyttäjä muokkaamaan tietuetta tietokannassa. Osaksi tästä syystä ja sen takia että omistajan tietoja ei tarvita mihinkään päätin parhaaksi, että koiran omistajille ei

tarvita erillistä taulua, vaan siihen voidaan tulevaisuudessa käyttää Wordpressin valmista käyttäjätaulua wp_users, jolloin samalla voidaan hallita käyttäjien oikeuksia vaivattomasti.

Tietokantarakenne tuntui yksiselitteiseltä tuloksiin asti. Tuloksien tallennusmuotoa mietittäessä ilmaantui useita mahdollisia ratkaisuja, joista monet olivat hyviä ja parhaan ratkaisun löytäminen tuntui lähes mahdottomalta. Ongelmaa aiheutti myös se että vanhat Ajok-tulokset olivat 4-eräisiä ja uudet olivat 2-eräisiä. Nämä kaksi tulosmenetelmää toimivat päällekkäin jonkin aikaa, joten suoraa automaatiota ei voitu laatia. Tästä kävi hyvin ilmi, että järjestelmään täytyisi myös jättää mahdollisuus päivittää sitä tarpeen vaatiessa niin, että koko tulostietokanta ei menisi täysin uusiksi.

4 KÄYTETTÄVYYS

ISO 9241-11 standardi määrittelee käytettävyyden seuraavasti "Se vaikuttavuus, tehokkuus ja tyytyväisyys, jolla tietyt määritellyt käyttäjät saavuttavat määritellyt tavoitteet tietyssä ympäristössä". Käytettävyys kuitenkin voidaan laajentaa käytettävyyden uranuurtaja J.Nielsenin mukaan koskemaan opittavuutta, muistettavuutta ja ei katastrofaalisten virheiden hallintaa. (Steering Committee on the Usability, Security, and Privacy of Computer Systems & National Research Council 2010, 3.)

4.1 Käyttöliittymä

Kun tietokantaa halutaan tarkastella lähetetään pyyntö jonkin elementin kautta, joka lähettää kyselyn tietokannalle ja tietokanta lähettää halutun tiedon ennalta määritellyssä muodossa. Etsitty elementti voi olla yksittäinen sana, lauseke, numerokoodi tai vastaava. (Jacsó 1999, 14.)

Tietokannan käyttämiseen tarvitaan siis käyttöliittymä, jonka kautta käyttäjä voi olla yhteydessä tietokantaan. Käyttöliittymät voidaan jakaa kolmeen pääkategoriaan: valikko rakenteelliset käyttöliittymät, komentorivikäyttöliittymät ja graafiset käyttöliittymät. Käyttöliittymien erot voidaan karkeasti jakaa karkeasti kolmeen osaan kuvio 9. osoittamalla tavalla. Vasteajalla tarkoitetaan, sitä kuinka kauan kestää siitä hetkestä kun käyttäjä antaa komennon siihen että komento tapahtuu. Komentorivikäyttöliittymässä on luonnollisesti vasteaika lyhyempi kun käyttöliittymään syötetään suoraan sen tarvitsema koodi, mutta tämän koodin muistaminen ei välttämättä ole helppoa, joten komentorivikäyttöliittymää on vaikea käyttää. Vastaavasti graafisessa käyttöliittymässä komentoa vastaavat kuvakkeet, jolloin vasteaika pitenee, mutta käytettävyys helpottuu huomattavasti. Käyttöliittymän suunnittelun kannalta on helpointa laatia käyttöliittymä, joka etenee valikoiden avulla. Valikoiden avulla voidaan rajoittaa käyttäjien antamat syötteet ja pystytään helposti laatimaan malli, jonka avulla pystytään ottamaan huomioon kaikki syötteet, joita käyttäjä voi antaa. Komentorivikäyttöliittymässä vastaavasti käyttäjä voi antaa minkä tahansa käskyn järjestelmään ja järjestelmän pitää pystyä toteuttamaan tämä komento. Tästä aiheutuu huomattavasti vaikeauksia suunnittelun ja toimivuuden kannalta kun käyttäjän toimintaa ei rajoiteta lainkaan. Yleensä parhaaksi havaittu käyttöliittymä sijoittuu johonkin väliin eikä siten ole

puhtaasti ainoastaan yhden tyyppinen käyttöliittymä. (Foster & Godbole 2014, Part B: The Relational Database Model - Chapter 6.)



Kuvio 9. Käyttöliittymien erot. Nuolen suunta osoittaa kummassa suunnassa kyseiset arvot ovat paremmat.

Komentorivi

Komentorivikäyttöliittymä (CLI, Command Line Interface) on vanhin näistä kategorioista, mutta se on kuitenkin laajassa käytössä vielä nykyäänkin. CLI on yksinkertainen ja sillä onnistuvat niin monimutkaiset kuin myös yksinkertaiset toiminnot. Tästä johtuen monet kehittäjät suosivat CLI käyttöliittymää. (White 2015, 3. The Hadoop Distributed Filesystem - The Command-Line Interface.)

Valikko

Valikkopohjaisessa käyttöliittymässä käyttäjä etenee valikoista toiseen valintojen mukaan. Valikkorakenne on yleensä huomattavasti suppeampi kuin CLI. Tämä johtuu siitä, että jokainen toiminto pitää erikseen koodata eli käytännössä kun teet valinnan niin käyttöliittymän taustalla on sama käsky kuin CLI:ssä, mutta käyttäjän ei tarvitse kyseistä riviä kirjoittaa. (Libby & Wellman 2013, Chapter 9. Creating Menus.)

Graafinen

Graafinen käyttöliittymä (GUI, Graphical User Interface) on kehittyneempi versio WIMP:stä (Windows, Icons, Menus, Pointers). Graafiseen käyttöliittymään on sisällytetty useita uudestaan käytettäviä käyttöominaisuuksia. Käyttöliittymä tukee mobiililaitteita, jotka eivät välttämättä käytä hiiren osoitinta. Tämä mahdollistaa sen, että pystytään käyttämään kuvia, ikoneita ja elementtejä 2-D kuvaruudulla ja nämä toteuttavat niille määritellyt toiminnot ilman että käyttäjän tarvitsee muistaa monimutkaisia komentoja. (Lal 2013, 16.)

4.2 SQL-komennot

Käyttöliittymästä riippuen käyttäjä näkee taustalla toimivat komennot erillä tavalla, mutta taustalla kaikkien käyttöliittymien esittämät komennot ovat käytännössä samoja. Käyttöliittymä voi vaatia käyttäjältä syötettä, joka vaikuttaa siihen miten ja mihin tietokannan elementtiin komento vaikuttaa.

SELECT column FROM tablename WHERE condition

SELECT-komentoa käytetään lähettämään pyyntö tietokannalle, jonka avulla saadaan tuotua valitun taulun ("SELECT column") tietokannasta ("FROM tablename"), jotka täyttävät annetut ehdot ("WHERE condition"). Ehdoiksi voidaan asettaa jokin taulun kentän saamista arvoista, jolloin komento palauttaa ainoastaan ne tietueet (rivit), jotka täyttävät annetun ehdon. Ehtojen laatimisessa voidaan käyttää myös wildcard-arvoa (%), jotta voidaan etsiä tietueita, joiden tarkkaa arvoa ei tiedetä. (W3Schools 2015, viitattu 9.9.2015.)

CREATE TABLE tablename (column1 data type, column2 data type)

Tietokannassa täytyy olla taulu jonne tiedot tallennetaan. Tämän luominen tapahtuu CREATE TABLE-komennolla. Tämä komento luo käyttäjän määrittelemän nimisen (tablename) taulun, johon käyttäjän täytyy määrittellä jokaisen sarakkeen nimi ja tietotyyppi. Vastaavasti DROP komennolla voidaan poistaa taulu tietokannasta. (W3Schools 2015, viitattu 9.9.2015.)

INSERT INTO tablename (col1, ... , coln) VALUES (val1, ... , valn)

Tyhjästä taulusta ei ole hyötyä, joten siihen täytyy pystyä lisäämään tietoa. Tietojen lisääminen tauluun tapahtuu INSERT INTO-komennon avulla. Komennolle annetaan halutut arvot (val1, ... , valn) ja määritellään mihin tauluun (tablename) ja mille sarakkeille arvot tallennetaan (col1, ... , coln). (W3Schools 2015, viitattu 9.9.2015.)

UPDATE tablename SET colX = valX [, colY = valY, ...] WHERE condition

On tärkeää, että tietokannassa olevat tiedot ovat ajan tasalla ja että tietoja voidaan päivittää. Esimerkiksi henkilötietolaki (22.4.1999/523 29 §) velvoittaa ylläpitäjää korjaamaan tai täydentämään virheelliset tiedot. Komennon avulla taulun sarakkeille annetaan niiden uudet arvot (colX = valX), jonka jälkeen ne päivitetään riville, joka täyttää vaaditun ehdon (WHERE condition). Vastaavasti tietojen poistaminen tietokannasta tapahtuu DELETE-komennolla, joka sisältää määrittelyn siitä mitkä rivit poistetaan taulusta. (W3Schools 2015, viitattu 9.9.2015.)

4.3 Tietotyypit

Kun tietokantaan laaditaan taulua täytyy jokaiselle sarakkeelle määritellä tietotyyppi. Jokainen tietotyyppi eroaa toisistaan siinä missä muodossa se tallennetaan tietokantaan ja miten sitä pystytään käsittelemään. (W3Schools 2015, viitattu 9.9.2015.)

Tietotyyppien muuttaminen jälkikäteen voi olla työlästä ja vaikeaa, joten on tärkeää, että tehdään hyvät suunnitelmat ennen kuin koodia lähdetään tuottamaan. Väärää muotoa oleva tieto voi estää ohjelmiston toiminnan täysin. Tämän takia on tärkeää, että käyttäjältä minimoidaan mahdollisuudet tulkita kysytty asia väärin ja hyväksytään ainoastaan oikeaa muotoa olevat tiedot. (W3Schools 2015, viitattu 9.9.2015.)

Tietotyyppien valinta

SQL-tietokannassa voidaan käyttää useita eri tietotyyppiä, joista jokaiseen tietotyyppiin voidaan tallentaa tietotyyppin määrittelemän laiset tiedot. Tietotyyppiin pystytään siis ainoastaan tallentamaan tietoja, jotka täyttävät kyseisen tietotyyppin vaatimukset. Parhaan tietotyyppin

löytäminen voi olla vaikeaa kun halutaan maksimoida tietokannan tehokkuus. (W3Schools 2015, viitattu 9.9.2015.)

char(n)/varchar(n)

Tietotyyppiin pystytään tallentamaan merkkijono, jonka pituus on n, mutta maksimissaan 8 000 merkkiä. Char- ja varchar-tietotyypit eroavat toisistaan siinä, että char-tietotyyppi tallentaa kaikki sarakkeen arvot saman pituisena (n tavua), kun taas varchar-tietotyypin pituus riippuu syötteen pituudesta (2+n tavua). Saman pituisten sarakkeiden käsittely on nopeampaa ja tästä johtuen char-tietotyyppi on nopeampi, mutta kun syötteiden pituudet vaihtelevat paljon aiheutuu tästä paljon tiedon "tuhlausta", mikä hidastaa tietojen käsittelyä. (W3Schools 2015, viitattu 9.9.2015.)

int(size)

Tietotyyppiin voidaan tallentaa ainoastaan kokonaislukuja $-2,147,483,648$ ja $2,147,483,647$ väliltä (4^{16} lukua siis). Tietotyypin käyttäminen vie tällöin 4 tavua jokaista sarakkeen arvoa kohti. Int-tietotyyppiä käytettäessä on tärkeää ottaa huomioon, että se ei hyväksy desimaalilukuja vaan se pudottaa suoraan desimaaliluvun pois. Käyttäjän antamat arvot pyöristetään kun ne tallennetaan tietokantaan ja kun tietokannassa oleva luku noudetaan on se tämä pyöristetty luku. (W3Schools 2015, viitattu 9.9.2015.)

decimal(size,d)

Tietotyyppiin voidaan tallentaa arvoja $-10^{38} + 1$ ja $10^{38} - 1$ väliltä (tietotyypin +/-1 johtuu numeron etumerkistä). Tietotyyppiin täytyy määritellä kaksi arvoa. Koon lisäksi täytyy decimal-tietotyyppiä käytettäessä määritellä d:lle arvo, joka kuvaa sitä kuinka monta lukua tallennetaan pilkun jälkeen tietokantaan. Decimal-tietotyyppi vie tilaa jaksoissa. Jokainen 9 numeroa pilkun edessä vie 4 tavua tilaa (ensimmäiset 10 numeroa vie 4 tavua) ylijäämä numerot vievät noin 1 tavun per kaksi numeroa. Pilkun jälkeisten numeroiden tilan vienti riippuu tietueen arvosta. (W3Schools 2015, viitattu 9.9.2015.)

DATETIME()

DATETIME-tietotyyppi tallentaa päivän sekä kellonajan sekunnin tarkkuudella tietueeseen. Tietotyyppi tallentaa syötteen muodossa YYYY-MM-DD HH:MI:SS, myöhemmin kun halutaan käyttää toista päiväyksen ilmoitusmuotoa, voidaan muuttujan avulla muokata päiväys haluttuun

muotoon. Kun tarvitaan pelkästään aika tai päiväys tulisi silloin käyttää niille tarkoitettuja tietotyyppisiä. (W3Schools 2015, viitattu 9.9.2015.)

Null-arvot

Laitettaessa tietoa tietokantaan on niissä lähes poikkeuksetta kenttiä, jotka eivät välttämättä saa mitään arvoa, näitä kutsutaan "null" arvoksi. Kun tietue saa arvoksi 0 tämä ei tarkoita, että arvo on "null". Arvo on null ainoastaan kun tietueella ei ole olemassa arvoa. Tietokantaa laatiessa on tärkeää huomata, että kaikki sarakkeet eivät välttämättä saa arvoja, jolloin ne täytyy määritellä niin, että ne voi olla "null". Kun taulussa on yksikin sarake joka voi saada arvon "null", varaa sql-tietokanta tällöin automaattisesti 4 tavua jokaista taulun riviä kohti. Tämä 4 tavua kattaa 32 saraketta, jotka voivat saada arvon "null". Kun tämä määrä ylittyy siirrytään käyttämään 8 tavua ja 64 saraketta. Eli lähes poikkeuksetta kannattaisi käyttää tietokannassa olevaa määritelmää enemmän kuin tekisi ratkaisun käyttöliittymään, joka korvaisi "null" kentät, jollakin arvolla näin vältettäisiin "null" arvojen esiintyminen taulussa. (MySQL 5.1 Manual. 11.7.)

Tilan varaus

Tietotyypit vievät eri verran tilaa keskenään. Jokaisella tietotyypillä on hieman toisistaan eroavat ominaisuudet, jonka vuoksi oikean tietotyypin valinta voi olla vaikeaa. Tietotyyppien tilanvaraus vaihtelee tyypistä riippuen taulukko 1. mukaisella tavalla. Kun eri tietotyyppisiä käsitellään, niiden käsittelyyn kulutettava aika luonnollisesti riippuu siitä kuinka paljon tietoa sarakkeista täytyy lukea tai kuinka paljon niihin täytyy kirjoittaa. Tästä johtuen kaikki ylimääräinen osuus, joka on varattu tietotyypin määrittelyssä, on turhaa ja aiheuttaa sen, että tietokanta toimii hitaammin. Tämän hidastumisen merkitys riippuu siitä kuinka paljon turhaa tietoa tietokantaan on varattu ja kuinka iso tietokanta on. (Oracle Corporation 2015. 11.7.)

Tietotyyppi	Tiedon varaus
TINYINT, CHAR(1)	1 tavua
SMALLINT, CHAR(2)	2 tavua
INT, FLOAT, CHAR(3-4), DECIMAL(5-9)	4 tavua
DATE	4 tavua
BIGINT, DOUBLE, CHAR(5-8), DECIMAL(10-18)	8 tavua

DATETIME	8 tavua
----------	---------

Taulukko 1. Tietotyyppien tilan varaus.

4.4 Tietojen syöttäminen

Yhdenmukaiset nimitykset

Tietokannassa on tärkeää, että kaikki tiedot esitetään johdonmukaisesti ja yksiselitteisesti, joissakin tapauksissa tämä on välttämätöntä tietokannan toimivuuden kannalta. Esimerkiksi koirarekisterinumero voidaan antaa virheellisesti, kun siihen lisätään ylimääräisiä merkkejä. Tämä voi tuntua pieneltä asialta, mutta se voi olla tietokannan toiminnan kannalta kriittistä. Etsittäessä tätä koirarekisterinumeroa ei välttämättä löydetä haluttua koiraa koirarekisterinumerossa olevan virheen vuoksi. (Jacsó 2015, 25–26.)

Vaikka muun tiedon osalta tietojen yksiselitteisyys ei ole toimivuuden kannalta välttämätöntä, voi se kuitenkin aiheuttaa muita ongelmia käyttäjälle ja tietokannalle. Kun tietokannassa käytetään johdonmukaisia ja yksiselitteisiä tietoja voidaan säästää aikaa ja välttää käyttäjien turhautumista ja ärsyyntymistä siitä kun tiedot ovat eri muodossa. Lapin Ajokoirayhdistyksen tietokannassa nimeä käytettäessä voi kyseessä olla koiran nimi, sen isän tai emon nimi tai kyseessä voi olla jopa koiran omistajan nimi. On tärkeää, että nämä erotetaan toisistaan ja että ne noudattavat samaa muotoa kuten omistajan nimi on muotoa Etunimi+Sukunimi eikä toisinpäin. Näitä tietoja kysyttäessä täytyy siis lomakkeen olla yksiselitteinen ja selkeä, jotta voidaan välttää virheitä. (Jacsó 2015, 25–26.)

Toistuva tieto

Tietokannan tarkoituksena on se, että samaa tietoa ei tallennettaisi useasti samaan kertaan ja että turhaa tietoa ei olisi tietokannassa ollenkaan. Tiedon tallentaminen useaan kertaan vie luonnollisesti enemmän tilaa palvelimelta, jolla tietokanta sijaitsee. Lisäksi samojen tietojen esiintyminen moneen kertaan voi aiheuttaa ristiriitaisten tietojen ilmaantumiseen tietokannassa. Saman tiedon ollessa useassa eri paikoissa voi helposti unohtua tiedon päivittäminen johonkin näistä paikoista. Näitä ristiriitoja voi syntyä kun päivitetään, lisätään tai poistetaan tietoa tietokannasta. Toistuvan tiedon esiintymistä tietokannassa voidaan välttää hyvällä suunnittelulla.

Hyvällä suunnittelulla voidaan estää se, että mikään automaatio ei tuota toistuvaa tietoa. Toistuvaa tietoa voi lisäksi antaa käyttäjä. Tästä syystä johtuen on tärkeää, että käyttäjälle esitetään selviä ja yksiselitteisiä kysymyksiä kun halutaan käyttäjältä tietoa. (Pallaw 2010, 125–126.)

4.5 Tiedon muokkaamisen seuraukset

Taulujen keskinäisiä yhteyksiä määriteltäessä voidaan valita miten toisen taulun tietojen muokkaaminen vaikuttaa toiseen tauluun. Esimerkiksi Lapin Ajokoirayhdistyksen tietokannassa koiran kilpailuilmottautuminen sijaitsee taulussa, joka viittaa toiseen tauluun, jossa on kilpailujen tulokset. Kun koiran ilmoittautuminen poistetaan kilpailusta, luonnollisesti koiran kilpailutuloksia ei myöskin tarvita, jolloin ne tulisi poistaa tietokannasta. Näiden toimintojen toteuttamiseen voidaan käyttää seuraavia ehtoja ja määrittellä mitä niiden toteutumisesta seuraa. (Pallaw, 125–126.)

On delete/On update

Tämä määrittelee milloin ehto toteutuu. Ehto voi toteutua kun taulusta yritetään poistaa riviä (on delete) tai kun taulun riviä yritetään muokata (on update). On update käytetään yleensä kun halutaan muokata taulun avainten arvoa. Tämä tapahtuu usein kun aletaan poistamaan taulusta turhaa tietoa, jota on voinut jäädä tietokantaan. Tällöin voidaan haluta muuttaa avainten arvoja. (Microsoft 2015, viitattu 9.9.2015.)

No action/Cascade/Set null/Set default

Tämä määrittelee sen mitä seuraa ehdon toteutumisesta. No action eli mitään ei tapahdu kun tietoa muokataan taulusta, mutta yhteys taulujen kenttien välillä kuitenkin on olemassa. Cascadea käyttämällä muutokset vaikuttavat kaikkiin sarakkeisiin, joihin yhteys viittaa. Set null käyttämällä jos sarakkeelle on määriteltynä mahdollisuus olla null(tyhjä) tällöin sarake johon viitataan asetetaan arvoon null. Set default vaihtoehtoa käytetään usein kun halutaan asettaa oletusarvo kentälle. Lapin Ajokoirayhdistyksen tietokannassa kennelpiiri on lähes poikkeuksetta Lappi, jolloin on turha vaatia käyttäjää laittamaan tämä tieto jokaisella kerralla, vaan on parempi että tämä tieto on oletusarvona ja mikäli käyttäjä haluaa vaihtaa kunnan joksikin toiseksi on siihen mahdollisuus. (Microsoft 2015, viitattu 9.9.2015.)

4.6 Lapin Ajokoirayhdistyksen tietokannan käyttöliittymän toteutus

Käyttöliittymä

Web-sivusto tulee pyörimään Wordpressin kautta, joten tuntui luonnolliselta, että tämä käyttöliittymä tulisi suunnitella niin, että sitä voitaisiin muokata Wordpressin kautta eikä erillistä editoria ja ftp-yhteyttä tarvittaisi ollenkaan kun haluttaisiin muokata käyttöliittymää. Wordpress lisäosilla voidaan usein saada halutut ominaisuudet, mutta halutuista ominaisuuksista ja budjetista riippuen näistä joudutaan joskus luopumaan. Ilmaisia lisäosia Wordpressissä on lukuisia, mutta näiden joukosta ei löytynyt yhtään, joka olisi suoraan täyttänyt sille asetetut vaatimukset ja valmiin järjestelmän muokkaus on usein vaikeampaa ja monimutkaisempaa kuin uuden järjestelmän suunnittelu ja toteuttaminen alusta lähtien. Maksullisia lisäosia tarkastellessani huomasin useita potentiaalisesti riittäviä lisäosia, mutta ongelmaksi muodostui se, että lisäosista saatavat tiedot ilman sen ostamista eivät olleet tarpeeksi laajat. Ainoaksi keinoksi lisäosan toimivuudesta olisi ollut ostaa lisäosa ja testata täyttääkö se tarvittavat vaatimukset vai ei. Päädyin tämän takia käyttöliittymän suunnitteluun ja toteuttamiseen alusta lähtien itse.

Käyttöliittymän pohjaksi ilmeni kaksi vaihtoehtoa. Ensimmäisenä vaihtoehtona oli toteuttaa käyttöliittymä tekemällä lisäosa tyyppinen ratkaisu, joka mahdollistaisi Wordpressin hallintapaneelin kautta tietokannan muuttamisen. Tämä ratkaisu ei kuitenkaan ollut käytännössä toteutettavissa lyhyessä ajassa ja käyttöliittymän toteutuksesta olisi tullut monimutkainen muuttua myöhemmässä vaiheessa. Toisena vaihtoehtona oli käyttöliittymän toteuttaminen Wordpressin sivut(Page) ominaisuuden kautta. Tällöin käyttöliittymää olisi helppo ja nopea muokata jos osaa ohjelmointia hieman, mutta järjestelmän hallitseminen ei olisi yhtä kattavaa ja helppoa. Käyttöliittymän toteuttamiseen ei ollut tarpeeksi aikaa jos ensimmäistä vaihtoehtoa käytettäisiin, joten toinen vaihtoehto oli luonnollinen valinta.

Käyttöoikeudet

Testattaessa ja osittain jo suunnitteluvaiheessa kävi ilmi, että kaikkia tietoja ei haluttaisi kaikkien nähtävillä ja muokattavaksi. Tämä toteutui helpoiten suunnittelemalla ja toteuttamalla hallintapuolen sivuston ensin ja tämän jälkeen siirtämällä ja muokkaamalla tätä

hallintajärjestelmää, että se kykeni näyttämään halutut tiedot tietokannasta. Kuvio 10. osoittamat punaiset laatikot eivät näy ollenkaan kun katsellaan tulosta, mutta ne näkyvät hallintapuolella. Tekstilaatikoista ja alavetovalikoista näkyvät tiedot näkyvät tekstimuodossa taulukossa tuloksia tarkasteltaessa. Kahden erillisen sivun rakentaminen mahdollisti käyttöoikeuksien hallinnan helposti ja kätevästi Wordpress pluginin avulla (Press Permit Core) ja näin päästiin helpommalla käyttöoikeuksien toteuttamisen suhteen.

Järjestäjä	Veitiluodon Metsämies ry		Tyyppi	AJOK	
Kennelipini	Lappi		Pinnit Nro	3	
Koekunta	Tervola		Aika	28.28.2015	
Koira	Mustrovan Elvis		Rek.nro	45313 /12	
Sukupuoli	Uros				
Isä	Räntäkaaman Ringo		Rek.nro		
Emä	Mustrovan Jenka		Rek.nro		
Omistaja	Öm Eva Öm Pa		Kotikunta	Kolan	
Koe-Erät	1	2	Hyväksytyt ajominuutit	Ajoajan pisteet	
Erä alkoi	06.00	06.00			
Haku min.	8	72	101	29.46	
Ajo min.	84	17			
			Ansiopisteiden keskiarvo	Ansiopisteet yhteensä	
Haku	10	9	9.50		
Haukku	8	8	8	20.50	
Ajotaito	5	1	3		
			Tappiopisteiden summat	Tappiopisteet yhteensä	
Hakubyyisyys	0	0	0	0	
Ajotyyisyys	0	0	0	0	
kieli	Palkas maa <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Luopii <input type="checkbox"/>	Loppupisteet	49.96
	Lumkeii <input type="checkbox"/>	<input type="checkbox"/>	Palkinto	AJOK	Sijoitus
Huomautus					
Ylläomari					

Kuvio 10. Tulokortin ensimmäisen sivun eroavaisuudet hallinta- ja näyttöpuolen välillä.

Koodi

Wordpress on pääosin koodattu käyttäen HTML-koodia ja .css tyylitiedosto määrittelee teeman mukaiset asetukset elementteihin. Tuntui siis luonnolliselta koodata käyttöliittymän rakenne käyttäen HTML-koodia ja viittaamalla tarvittaessa koodissa tyylitiedostoon. Pystyin tekemään käyttöliittymän, joka oli riippumaton teeman valinnasta ja muuttuisi teeman mukana mikäli sitä vaihdettaisiin.

Toteutin käyttöliittymän toiminnallisen osion .php-koodilla. Tähän kuului perus if, while, for ja muut vastaavat loopit ja peruskomennot. Tietokannan kanssa kommunikointiin käytin Wordpress database (WPDB) funktioita, jotka tulevat Wordpressin mukana. Funktiot sisältävät pääosin kaikki samat SQL-komennot ja nämä toiminnot toteuttaa WPDB-funktio sen sijaan, että suoraan

kirjoitettaisiin SQL-komento. Tämä helpotti käyttöliittymän laatimista kun pystyttiin hieman oikeisemaan SQL-komennoissa.

Käyttöliittymä

Käyttöliittymän tarkoituksena oli se, että se olisi mahdollisimman helppokäyttöinen ja monimutkaisia ohjeita ei tarvittaisi ollenkaan. Mikäli ohjeille olisi tarvetta, voitaisiin ne sijoittaa suoraan hallintajärjestelmään. Vanhassa järjestelmässä käyttäjän syötteitä ei rajattu ollenkaan, kun niitä pyydettiin, mutta mikäli ne olivat väärässä muodossa ilmoitti järjestelmä, tällöin että syöte on viallinen. Rajaamalla käyttäjän antamia mahdollisia syötteitä pystyttiin helpottamaan järjestelmän käyttöä ja pienentämään viallisten syötteiden määrää samalla. Näiden syötteiden rajaaminen toteutettiin kuvio 11. osoittamalla tavalla. Valintaruutua käytettiin kun käyttäjältä haluttiin tietää esimerkiksi onko sää kohtalainen vai ei. Lämpötilaa ja maastoa kysyttäessä päätin käyttää avointa tekstikenttää. Lämpötilaa pyydetäessä oli mahdollista käyttää alavetovalikkoa, mutta vaihtoehtoja olisi tässä valikossa ollut liikaa, joten se ei olisi ollut käytännöllinen enää. Lisätiedoissa suuri osa kentistä pyytää käyttäjältä syötettä väliltä 1-5 (lisäksi null on mahdollinen syöte), osa kentistä kuitenkin eroaa tästä ja vaatii että syöte on kokonaisluku tai että kenttä voi saada arvoja vain 3 asti. Näissä tapauksissa käyttäjän syötteen rajaaminen onnistui helpoiten alavetovalikolla.

Erät	1	2
Olosuhteet		
11. Pajjas maa	<input type="checkbox"/>	<input type="checkbox"/>
12. Lumikeli (cm)	<input type="checkbox"/>	<input type="checkbox"/>
13. Kohtalainen tuuli (X)	<input type="checkbox"/>	<input type="checkbox"/>
14. Kuiva keli	<input type="checkbox"/>	<input type="checkbox"/>
15. Kostea keli	<input type="checkbox"/>	<input type="checkbox"/>
16. Kohtalainen sade (X)	<input type="checkbox"/>	<input type="checkbox"/>
17. Lämpötila	<input type="text"/>	<input type="text"/>
18. Maasto	<input type="text"/>	<input type="text"/>
Haku		
20. Haun laajuus ilman yöjälkeä	<input type="text"/>	<input type="text"/>
21. Vainuamistapa	<input type="text"/>	<input type="text"/>
22. Hakulöylyyden laatu	<input type="text"/>	<input type="text"/>
Haukku		
30. Kuuluvuus	<input type="text"/>	<input type="text"/>
31. Kiertovuus	<input type="text"/>	<input type="text"/>
32. Intohimoisuus	<input type="text"/>	<input type="text"/>
33. Tiheys	<input type="text"/>	<input type="text"/>
34. Äänen määrä	<input type="text"/>	<input type="text"/>
35. Sukupuolileima	<input type="text"/>	<input type="text"/>
36. Beaglen haukku	<input type="text"/>	<input type="text"/>
<input type="button" value="Muokkaa kaikki"/>		

Erät	1	2
Metsästysinto		
40. Metsästysinto haun aikana	<input type="text"/>	<input type="text"/>
41. Metsästysinto ajon aikana	<input type="text"/>	<input type="text"/>
42. Metsästysinto koetteluajana	<input type="text"/>	<input type="text"/>
Ajo		
50. Ajotaito	<input type="text"/>	<input type="text"/>
51. Nopeus	<input type="text"/>	<input type="text"/>
52. Tie- ja estotyöskentely	<input type="text"/>	<input type="text"/>
53. Vainuamistapa	<input type="text"/>	<input type="text"/>
54. Havainnot Herkkyydestä	<input type="text"/>	<input type="text"/>
55. Ajolöylyyden laatu	<input type="text"/>	<input type="text"/>
56. Ajettava nähty	<input type="text"/>	<input type="text"/>
Muut ominaisuudet		
60. Muiden eläinten ajo	<input type="text"/>	<input type="text"/>
61. Hallittavuus	<input type="text"/>	<input type="text"/>

Kuvio 11. Ajokokeen tuloslomakkeen lisätiedot osio - käyttäjän syötteiden saamistavat.

Osa käyttäjän syötteiden arvosta kuitenkin toimii nopeammin ja paremmin käytännössä kun käyttäjälle tarjotaan tekstikenttä johon hän voi antaa syötteen. Hyvänä esimerkkinä tästä on kuvio

12. mukainen lomake, jota käytetään ajokokeiden tulosten tallentamiseen. Kuviosta huomaa, että haku, haukku, ajotaito, hakulöyisyys ja ajolöyisyys pyytävät syötteekseen tekstiä tai tässä tapauksessa numeroita. Näiden toteuttamisessa olisi voitu käyttää alasvetovalikkoa, mutta käytännöllisyyden puolesta huomattiin, että tulosten syöttäminen tapahtui nopeammin kun käyttäjä pystyi kirjoittamaan numeron ja menemään sarkaimen avulla seuraavan kenttään. Vikasyötteistä ei tarvinnut niin paljoa huolehtia sillä järjestelmän tarkoituksena on siirtää tulokset lomakkeelta sähköiseen muotoon eikä kirjata tuloksia ensimmäistä kertaa. Lisäksi toisin kuin lisätietokentät, joihin ei välttämättä kirjata mitään arvoja, nämä kaikki kentät täytetään.

Järjestäjä	Veitsiluodon Metsämiehety		Tyyppi	AJOK
Kennelpiiri	Lappi		Piirin Nro	3
Koekunta	Tervola		Alka	28.28.2015
Koira	Hakaten Veera		Rek.nro	48719/11
Sukupuoli	Narttu			
Isä	Kuturan Vikke		Rek.nro	
Emä	Raimolan Heta		Rek.nro	
Omistaja	Pellinen Marko		Kotikunta	
Koe-Erät	1	2	Hyväksytyt ajominuutit	Ajoajan pisteet
Erä alkoi				
Haku min.				
Ajo min.				
			Ansio pisteiden keskiarvot	Ansio pisteet yhteensä
Haku				
Haukku				
Ajotaito				
			Tappio pisteiden summat	Tappio pisteet yhteensä
Hakulöyisyys				
Ajolöyisyys				
Itell	Paljas maa	<input type="checkbox"/>	Luopui	<input type="checkbox"/>
	Lumikeli	<input type="checkbox"/>	Palkinto	
			Loppupisteet	
Huomautus				
Ylituomari			Sijoitus	

Kuvio 12. Ajokokeen tuloslomake perustiedot ja ajotiedot - käyttäjän syötteiden saamistavat.

5 POHDINTA

Opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa uusi tietokanta ja sen käyttöliittymä Lapin Ajokoirayhdistykselle. Tehtävän suorittamista vaikeutti se että toimeksiantaja toimi Rovaniemellä, mutta tästä huolimatta yritin olla mahdollisimman hyvin yhteydessä asiakkaaseen, jotta järjestelmä vastaisi mahdollisimman hyvin asiakkaan toiveita.

Tietokantaa suunnitellessani tein parhaani miettiessäni ja selvitellessäni mahdollisia poikkeuksia ja erikoisuuksia mitkä vaikuttaisivat tietokantaan. Tästä huolimatta en kuitenkaan onnistunut omasta mielestäni tässä asiassa tarpeeksi hyvin. Järjestelmää testattaessa kävi nopeasti ilmi, että useita tietokannan tauluja piti laajentaa mahdollisten poikkeustapausten ilmaisemiseksi. Osa syynä tähän oli se, että vanhaa järjestelmää on päivitetty useaan otteeseen. Järjestelmää päivitettäessä oli usein tehty suunnittelemtomia ja dokumentoimattomia muutoksia käyttöliittymään sekä tietokantaan. Puutteellisesta dokumentoinnista johtuen muutokset ja näistä aiheutuneet ongelmat eivät tulleet esille tutkiessani järjestelmää. Tämä ei kuitenkaan muodostunut suureksi esteeksi järjestelmän suunnittelussa ja toteutuksessa. Pieniä ongelmia ilmaantui palveluntarjoajan SQL-versioiden ja niiden ristiriitojen kanssa, mutta nämä ongelmat onnistuttiin korjaamaan tai pahimmassa tapauksessa nämä ongelmat täytyi kiertää. Tietokannasta saatiin kuitenkin toimiva kokonaisuus, jota yhdistys voi mielestäni käyttää tuloksien tallentamiseen.

Käyttöliittymän suunnittelussa tein muutamia oleellisia virheitä, jotka huomasin kun käyttöliittymä laajeni. Osa näistä ongelmista pystyin korjaamaan helposti ja osa näistä ei ollut niin vakavia, että niiden puutteista olisi ollut suurta haittaa. Käyttöliittymän suunnitteluun olisi minun mielestäni pitänyt käyttää huomattavasti enemmän aikaa. Tämä olisi pidemmän päälle säästänyt koodaamiseen tarvittavaa aikaa ja käyttöliittymän koodista olisi tullut tehokkaampaa. Näistä seikoista huolimatta toteutettu käyttöliittymä täyttää sille asetetut vaatimukset, jättäen siihen kuitenkin kehittämisvaraa.

LÄHTEET

Foster, C. & Godbole, S. 2014. Database systems. New York: Apress.

Jacsó, P. & Lancaster, W. 1999. Build Your Own Database. Chicago: ALA Editions.

Lal, R. 2013. Digital Design Essentials. Beverly: Rockport Publishers.

Lapin Ajokoirayhdistys 2015. Viitattu 9.9.2015, <http://www.lapinajokoirayhdistys.fi/>.

Libby, A. & Wellman, D. 2013. jQuery UI 1.10: The User Interface Library for jQuery. Birmingham: Packt Publishing.

Lightstone, S., Nadeau, T. & Teorey, T. 2005. Database Modeling and Design: Logical Design. Burlington: Morgan Kaufmann.

Microsoft 2015. Microsoft SQL Server. Viitattu 9.9.2015, <https://technet.microsoft.com/library/bb545450.aspx>.

Murthy, C. 2008. Database management Design. Delhi: Himalaya Publishing House.

Oracle Corporation 2015. MySQL 5.1 Manual. Viitattu 9.9.2015, <http://dev.mysql.com/doc/refman/5.1/en/>.

Pallaw, V. 2010. Database Management Systems. New Delhi: Asian Books Pvt Ltd.

Steering Committee on the Usability, Security, and Privacy of Computer Systems & National Research Council 2010. Toward Better Usability, Security, and Privacy of Information Technology : Report of a Workshop. Washington, D.C.:National Academies Press.

Stephens, R. 2009. Beginning Database Design Solutions. New Jersey: Wiley

Suomen Ajokoirajärjestö 2011. Ajokokeen (Ajok ja Beaj) säännöt. Viitattu 9.9.2015, <http://www.ajokoirajarjesto.fi/ajankohtaista/AJOKOKEEN%20s%E4%E4nn%F6t%202010%20versio%20180311.pdf>.

W3Schools 2015. SQL Tutorial. Viitattu 9.9.2015. <http://www.w3schools.com/sql>.

White, T. 2015. Hadoop: The Definitive Guide, 4th Edition. Kalifornia: O'Reilly Media, Inc.



SUOMEN KENNELLIITTO -
FINSKA KENNELKLUBBEN RV.

AJOK

Ajokokeen arvostelukortti Bedömningskort vid drevprov

Koeaika Provtid	_____ / _____ 20 _____	Erän nro Omgång nr.	Erä alkoi klo Omgången började kl.
Maasto Provmark			<input type="checkbox"/> Uros Hanhund
Koiran nimi Hundens namn			<input type="checkbox"/> Narttu Tik
Ohjaaja Hundförare			

Ominaisuuksien arvostelu asteikko: ansiop. 1–10; tappiop. 0–10 Bedömning av egenskaperna skala: posit. 1–10; negat.0–10					
Ominaisuus Egenskap	Haku Sök	Haukku Skall	Ajotaito Drevförmåga	Hakulöisyys Löshet på sök	Ajolöisyys Löshet på drev
Tuomari Domaren					
Ryhmä Gruppen					
Apurivi Hjälprad	+	+			
Apurivi Hjälprad	+	+			
Apurivi Hjälprad	+	+			

Huomautuksia
Anmärkningar

LISÄTIEDOT • TILLÄGGSUPGIFTER

Asteikko 5–1

Skala 5–1

OLOSUHTEET FÖRHÅLLANDENA	
11. Paljas maa Barmark	<input type="checkbox"/>
12. Lumikeli Snöföre	cm
13. Kohtal. tai kova tuuli Mättig eller hård vind	<input type="checkbox"/>
14. Kuiva keli Torrt före	<input type="checkbox"/>
15. Kosteaa keli Fuktig före	<input type="checkbox"/>
16. Kohtal. tai kova sade Mättig eller hårt regn	<input type="checkbox"/>
17. Lämpötila Temperatur	°C
18. Maasto Provmark	

HAUKKU SKALL	
30. Kuuluvuus Hörbarhet	
31. Kertovuus Passion	
32. Intohimoinisuus Intensitet	
33. Tiheys Täthet	
34. Äänien määrä Mångtonighet	
35. Sukupuolileima Könsprägel	
36. Beaglen haukku Beaglens skall	

AJO DREV	
50. Ajotaito Drevförmåga	
51. Nopeus Fart	
52. Tie- ja estetyökentely Arbete på väg o. vid hinder	
53. Vainuamistapa Spårningsätt	
54. Skallbenägenhet Havainnot herkkyydestä	
55. Ajolöisyyden laatu Arten av löshet på drev	
56. Ajettava nähty Drevdjumoteringar	kertaa ggr

HAKU SÖK	
20. Laajuus ilman yöjälkeä Vidd utan nattslag	
21. Vainuamistapa Spårningsätt	
22. Hakulöisyyden laatu Arten av löshet på sök	

METSÄSTYSINTO JAKTLUST	
40. Haun aikana Under söket	
41. Ajon aikana Under drevet	
42. Koetteluaikana Under provtiden	

MUUT OMINAIS. ÖVRIGA EGENSK.	
60. Muiden eläinten ja sorkkaeläinten ajo Drev av övrigt vilt och klovvilt	
61. Hallittavuus Lydnad	

Huomautuksia
Anmärkningar

Ryhmätuomari
Gruppdomare

Allekirjoitus
Underskrift
Nimen selvennys
Förtygande av underskrift

HAKUTAULUKKO • SÖKTABELL

Klo/Kl. _____

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60

Huomautuksia • Anmärkningar:

.....

.....

Hakuaikaa käyttämättä:
Söktid kvar:

Hakuaika yht. min.
Sammanlagd söktid i min.

Hakuaikaa käyttämättä:
Söktid kvar:

AJOTAULUKKO • DREVTABELL

Klo/Kl. _____

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60

Huomautuksia • Anmärkningar:

.....

.....

Ajoaika yht. min.
Sammanlagd drevtid i min.

Ryhmän toteama ajoaika
yht. min.
Gruppens sml. konstaterande drevtid i min.

Ajolöysyittä
Löshet på drev _____ min.

AJOKOKEEN KOIRAKOHTAINEN PÖYTÄKIRJA

LIITE 3

SKL:n tunnus		Rotukoodi	
Kennelpiiri		Piirin nro	
Koekunta		Koeaika	
Järjestäjä			
<input type="checkbox"/> Uros <input type="checkbox"/> Narttu <input type="checkbox"/> Rokotus <input type="checkbox"/> Tunnistus			
Koiran nimi		Rek.nro	
Koiran isä		Rek.nro	
Koiran emä		Rek.nro	
Omistaja		Kotikunta	
KOE-ERÄT	1	2	Hyväksytyt ajominuutit
Erä alkoi			
Haku min.			
Ajo min.			
		Ansiopisteiden keskiarvot	Ansiopisteet yht.
Haku			
Haukku			
Ajotaito			
		Tappiopisteiden summat	Tappiopisteet yhteensä
Hakulöyisyys			
Ajolöyisyys			
Keli		Palkinto	LOPPUPISTEET
<input type="checkbox"/> Paljas maa <input type="checkbox"/> Lumikeli			SUOITUS /
Huomautukset <input type="checkbox"/> Luopui <input type="checkbox"/> Suljettu <input type="checkbox"/> Keskeytetty			

Ajokokeen koirakohtainen pöytäkirja

LISÄTIEDOT		Asteikko 1-5	
Erät		1	2
Olosuhteet			
11. Paljas maa (x)			
12. Lumikeli (cm)			
13. Kohtalainen tai kova tuuli			
14. Kuiva keli			
15. Kosteaa keli			
16. Kohtalainen tai kova sade			
17. Lämpötila (°C)			
18. Maasto			
Haku			
20. Haun laajuus ilman yöjälkeä			
21. Vainuamistapa			
22. Hakulöyisyyden laatu			
Haukku			
30. Kuuluvuus			
31. Kertovuus			
32. Intohimosisuus			
33. Tiheys			
34. Äänen määrä			
35. Sukupuolileima			
36. Beaglen haukku			
Tuomareiden allekirjoitukset ja nimenselvennykset			
Ryhmittöumari		K	
Nimenselvennös		J	
Palkintotuomari		K	
Nimenselvennös		J	
Ylituomari			
Nimenselvennös			

SKL:n tunnus		Rotukoodi	
Kennelpiiri		Piirin nro	
Koekunta		Koeaika	
Järjestäjä			
<input type="checkbox"/> Uros <input type="checkbox"/> Narttu <input type="checkbox"/> Rokotus <input type="checkbox"/> Tunnistusmerkintä			
Koiran nimi		Rek.nro	
Koiran isa		Rek.nro	
Koiran emä		Rek.nro	
Omistaja		Kotikunta	
Koemaasto			
Haku alkoi klo	Haku-min.	Ajo alkoi klo	Ajo-min.
1. Lisä-haku klo	Haku-min.	Lisäajo alkoi klo	Ajo-min.
2. Lisä-haku klo	Haku-min.	Lisäajo alkoi klo	Ajo-min.
3. Lisä-haku klo	Haku-min.	Lisäajo alkoi klo	Ajo-min.
Hakutyöskentely-aika yht. min.		Ajotyöskentely-aika yht. min.	
		Koiria kiinni klo	

OMINAISUUS	PISTEET
Hakuvarmuus ja tehokkuus	
Hakunopeus ja eteneminen	
Metsästysintä ja hakusitkeys	
Hakutyöskentelyn muut ominaisuudet	
Ajon sujuvuus ja nopeus	
Metsästysintä ja ajon muut ominaisuudet	Ominaisuuspisteet yht.
Haukku	
Keli <input type="checkbox"/> P <input type="checkbox"/> L Hyväksytty ajoaika min.	Ajopisteet
Sijoitus kokeessa / Palkintosija	Loppupisteet

Huomautukset: Luopui Suljettu Keskeytetty

Ketunajokokeen koirakohtainen pöytäkirja KEAJ

6. LISÄTIEDOT

6.1. TYÖSKENTELYLOSUHTEET

11. Paljas maa (x)	
12. Lumikeli (cm)	
13. Kuiva keli	
14. Kostea keli	
15. Lämpötila (°C)	

6.2. HAKU ILMAN YÖJALKEÄ

21. Metsästysintä	
22. Laajuus	

6.3. TYÖ YÖJALJELLA

31. Metsästysintä	
32. Väinuumistapa	
33. Herätteley ja hakulöyisyys	
34. Herätteley ja hakulöyisyiden laatu	
35. Takajälkeen jäljitys	
36. Yöjäljen löytötapa	
37. Yöjäljellä seurattu matka	

6.4. AJOTAPA

41. Nopeus	
42. Sujuvuus	
43. Väinuumistapa	
44. Ajolöyisyiden ja takajälk. ajon laatu	

6.5. TYÖ HUKALLA

51. Tehokkuus hukkatyöskentelyssä	
-----------------------------------	--

6.6. HAUKKU

60. Kuuluvuus	
61. Kertovuus	
62. Intohimoisuus	
63. Tiheys	
64. Äänien määrä	
65. Sukupuolileima	
66. Havainnot herkkyydestä	

6.7. MUUT OMINAISUUDET

71. Tietyöskentely	
72. Muiden eläinten ajo	
73. Hallittavuus	
74. Louheen menon ilmoittaminen	
75. Ajo kuulumattomissa	
76. Ajon päättymisen syy	
77. Suurpetohavainto	
78. Ajettavan ketun toteaminen	
79. Ampumatilaisuus	

ÄKELI:
 1. Kennelliitto
 2. rotujärjestö
 3. kennelpiiri
 4. koiranomistaja

Tuomareiden allekirjoitukset ja nimenselvennykset

Ryhtätuomari	K
Nimenselvennös	J
Palkintotuomari	K
Nimenselvennös	J
Yliuomari	
Nimenselvennös	