

AJANVARAUSJÄRJESTELMÄ KOIRAKESKUS DOGGIELLE

Seela Ihanainen
Jaakko Pakkala

Opinnäytetyö
Huhtikuu 2011

Mediatekniikka
Tekniikan ja liikenteen ala



JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES



Tekijä(t) IHANAINEN, Seela PAKKALA, Jaakko	Julkaisun laji Opinnäytetyö	Päivämäärä 29.04.2011
	Sivumäärä 130	Julkaisun kieli Suomi
	Luottamuksellisuus () saakka	Verkkojulkaisulupa myönnetty (X)
Työn nimi AJANVARAUSJÄRJESTELMÄ KOIRAKESKUS DOGGIELLE		
Koulutusohjelma Mediatekniikan koulutusohjelma		
Työn ohjaaja(t) NIEMI, Kari		
Toimeksiantaja(t) Koirakeskus Doggie		
Tiivistelmä Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa internetissä toimiva ajanvarausjärjestelmä, jonka kautta asiakkaat voivat varata trimmaus- ja koulutusaikoja. Tavoitteena oli myös suunnitella ja dokumentoida koko järjestelmä mahdollisimman hyvin, sekä tehdä siitä mahdollisimman selkeä ja yksinkertainen, jotta varausten lisääminen ja selailu olisi helppoa kaikille käyttäjille. Toteutuksessa käytettiin pääasiassa PHP- ja XHTML-kieliä sekä MySQL -tietokantaa. Varaukset tallennetaan Googlen kalenteriin, joten tärkeässä osassa järjestelmässä on Zend_Gdata-komponentin avulla käytettävä Google Data API. Muitakin teknologioita käytettiin, mutta ainoastaan tukitarkoituksessa: sovelluksen haluttiin toimivan myös ilman lisäosia tai javascriptiä. Tuloksena syntyi helppokäyttöinen, Googlen kalenteria hyödyntävä ja ohjelmistosuunnittelun menetelmien pohjalta dokumentoitu ajanvarausjärjestelmä. Työssä selvitettiin myös oman ohjelmointityön ja valmiiden komponenttien sopivaa suhdetta. Kaikkia alkuperäisiä suunnitelmia ei ehditty toteuttamaan; tärkein tavoite, eli palveluaikojen varaaminen, kuitenkin toteutui kuten suunniteltiin. Toimeksiantaja oli tyytyväinen järjestelmään, joka oli odotettu ja tarpeellinen. Järjestelmä tehosti työajan käyttöä, sillä Doggien työntekijöiden ei tarvitse enää hallita ajanvarauksia puhelimitse, vaan heidän asiakkaansa voivat varata mieluisan ajan internetin kautta.		
Avainsanat (asiasanat) Ajanvarausjärjestelmä, kalenteri, Zend_Gdata, Google Data API, rajapinta, dokumentointi, ohjelmistosuunnittelu, projektinhallinta		
Muut tiedot		



Author(s) IHANAINEN, Seela PAKKALA, Jaakko	Type of publication Bachelor's / Master's Thesis	Date 11.05.2011
	Pages 130	Language Finnish
	Confidential () Until	Permission for web publication (X)
Title ONLINE BOOKING SYSTEM FOR KOIRAKESKUS DOGGIE		
Degree Programme Media Engineering		
Tutor(s) NIEMI, Kari		
Assigned by KOIRAKESKUS DOGGIE		
Abstract The aim of this thesis was to design and implement an online booking system, with which guests can book grooming and training sessions for dogs. The aim was also to plan and document the system as well as possible, and to make it easy to use by having a clear and simple interface. Primary technologies used were PHP, XHTML and MySQL. The reservations are sent to and retrieved from Google Calendar, thus an important part of the system is Google Data API, used through Zend_Gdata PHP interface. Other technologies were also used, but only for support purposes: the application needed to function without additional plugins or Javascript. The result is an easy to use booking system that utilizes Google Calendar and is documented using the principles of software development methodology. Comparing the suitable ratio between the author's own code and components made by others was also a part of the thesis. There was not enough time to actualize every planned feature; however, the main objective - the possibility to book services online – was realized as planned. The client was pleased with the system, which was welcomed and necessary. The system helps to optimize working time, because the employees of Koirakeskus Doggie no longer have to manage time reservations by phone.		
Keywords OBS, online booking system, calendar, Zend_Gdata, Google Data API, documenting, software development, project management		
Miscellaneous		

SISÄLTÖ

1	LÄHTÖTILANNE.....	4
1.1	Toimeksiantaja	4
1.2	Taustaa	4
1.3	Tavoitteet	5
2	PROJEKTIN YLEINEN LÄPIVIENTI.....	5
2.1	Mikä projekti on?.....	5
2.2	Projektin vaiheistaminen	7
2.3	Tarpeen tunnistaminen projektissa	11
2.4	Projektin määrittelyvaihe.....	12
2.5	Projektin suunnitteluvaihe	14
2.6	Projektin toteutusvaihe ja läpi vieminen	16
2.6.1	Toteutusvaihe	16
2.6.2	Resurssien hallinta	18
2.6.3	Muutosten hallitseminen	18
2.7	Testausvaihe	20
2.8	Projektin päättäminen	26
2.9	Tulosten luovuttaminen	28
3	KOIRAKESKUS DOGGIEN AJANVARAUSJÄRJESTELMÄN TARPEIDEN MÄÄRITTELY	28
3.1	Tarpeen tunnistaminen.....	28
3.2	Ajanvarausjärjestelmän määrittely	29
3.2.1	Ajanvarausjärjestelmän toiminnot	30
3.2.2	Adminin toiminnot.....	33
4	TOTEUTUSVAIHTOEHTOJEN KARTOITUS	36
4.1	Kartoituksen tavoitteet.....	36
4.2	Ajanvarausjärjestelmältä vaaditut toiminnot	37

4.3	Ajanvarausjärjestelmien vertailu	38
4.4	Kalenterin valinta.....	39
4.5	Valmis ohjelmistokehys	40
4.6	Olio-ohjelmointi vai proseduraalinen ohjelmointi?	41
4.7	Toteutusteknologian valinta	41
5	SUUNNITTELU JA TOTEUTUS	42
5.1	Ajanvarausjärjestelmän suunnittelu.....	42
5.2	Ajanvarausjärjestelmän toteutus	44
5.2.1	Käytettävät tekniikat	44
5.2.2	Sivut	45
5.2.3	Kansiorakenne	45
5.2.4	Tietokanta.....	46
5.3	Sivuston runko	46
5.4	Kalenterin rajapinnan käyttö	48
5.4.1	Käyttöönotto.....	48
5.4.2	Kalenterin lukeminen taikakeksiautentikoinnin avulla	48
5.4.3	Varauksen lisääminen ClientLogin-autentikoinnin avulla	50
5.5	Järjestelmän testaus	51
6	KÄYTTÄJÄKOULUTUS	52
7	PÄÄTÖSVAIHE.....	53
8	TULOKSET JA JATKOKEHITYS	54
9	YHTEENVETO	54
	LÄHTEET	56
	LIITTEET.....	57
	Liite 1. Vaatimusmäärittely.....	57
	Liite 2. Käyttötapauskaavio-käyttäjä.....	70
	Liite 3. Käyttötapaukset-käyttäjä.....	71

Liite 4. Käyttötapauskaavio-Admin.....	85
Liite 5. Käyttötapaukset-Admin.....	86
Liite 6. Testaussuunnitelma.....	104
Liite 7. Testitapaukset.....	110
Liite 8. Tietokannan luokkakuvaukset.....	126

KUVIOT

KUVIO 1. Kuvio projektin yleisestä etenemisestä	7
KUVIO 2. Esimerkki vesiputousmallista	8
KUVIO 3. Kuvio tarpeentunnistamisvaiheen paikasta projektin etenemisessä	11
KUVIO 4. Kuvio määrittelyvaiheen paikasta projektin etenemisessä	12
KUVIO 5. Kuvio suunnitteluvaiheen paikasta projektin etenemisessä	14
KUVIO 6. Kuvio toteutusvaiheen sijainnista projektin etenemisessä	16
KUVIO 7. Muutosten hallinnan vaiheita	20
KUVIO 8. Testausvaiheen sijainti projektin etenemisessä.	21
KUVIO 9. Testaustasot käsittävä V-malli	23
KUVIO 10. Kuvio päätösaiheen sijainnista projektin etenemisessä	26
KUVIO 11. Määrittelyprosessin kulku.....	29
KUVIO 12. Esimerkki näkymä ajanvarausjärjestelmän Omat tiedot -sivusta	31
KUVIO 13. Esimerkki näkymä ajanvarausjärjestelmän Valitse palvelu -osiosta	32
KUVIO 14. Esimerkki näkymä ajanvarausjärjestelmän Valitse aika -osiosta.....	33
KUVIO 15. Esimerkki näkymä Adminin käyttämästä järjestelmästä	34
KUVIO 16. Sivuston toimintaperiaate	47
KUVIO 17. Sama ajankohta Googlen ja varausjärjestelmän kalentereissa	49
KUVIO 18. Lisätty tapahtuma Googlen kalenterissa	51

1 LÄHTÖTILANNE

1.1 Toimeksiantaja

Opinnäytetyön toimeksiantajana toimi Jyväskyläläinen Koirakeskus Doggie. Yritys on melko nuori ja se on perustettu vuonna 2008. Tällä hetkellä Koirakeskus Doggie työllistää kaksi henkilöä. He tarjoavat monenlaisia palveluita, kuten trimmausta, koulutuksia, pentutestausta, koirien päivähoitoa ynnä muuta sellaista.

Yrityksen tavoitteena on parantaa koirien ja heidän omistajiensa elämänlaatua ammattitaitoisella koira-alan osaamisella. Koiran koulutuksessa he auttavat koira ja omistajaa saavuttamaan yhteisymmärryksen niin arkielämässä kuin kisakentillä. Apua tarjotaan myös jo syntyneisiin käytösongelmiin ja niiden ennaltaehkäisyyn.

Omistajien työpäivien ajaksi virikkeellisiä hetkiä koirille tarjoaa koirapäiväkot. Lisäksi neuvoja saa turkinhoidosta sekä muista käytännönasioista. (Koirakeskus Doggie 2011)

Heidän visionsa on levittää koiratietoutta yrityksen kautta ja helpottaa koiraperheiden arkea.

1.2 Taustaa

Koirakeskus Doggie tarvitsi ajanvarausjärjestelmän ja asiakasrekisterin helpottamaan yrityksen arkea. Ennestään heillä ei ollut minkäänlaista ajanvarausjärjestelmää eikä kunnollista asiakasrekisteriä. Kaikki oli hoidettu paperilla. Nykyään heidän resurssejaan kuluu paljon puhelimesta olemiseen ja tietojen etsimiseen erinäisiltä

paperilappusilta. He toivoivat järjestelmän tuovan helpotusta näihin asioihin ja asiakastietojen nopeampaan järjestelyyn.

1.3 Tavoitteet

Opinnäytetyön tavoitteena oli luoda toimiva ajanvarausjärjestelmä, jonka kautta yrityksen asiakkaat voivat varata niin trimmausaikoja kuin kurssiaikoja internetin välityksellä. Järjestelmästä pyrittiin tekemään mahdollisimman selkeä ja yksinkertainen, jotta varauksien lisääminen ja selailu olisi helppoa kaikille käyttäjille. Käyttöliittymä suunniteltiin noudattamaan Koirakeskus Doggien nettisivujen nykyistä tyyliä.

2 Projektin yleinen läpivienti

2.1 Mikä projekti on?

Projekti-sana on peräisin latinasta ja tarkoittaa ehdotusta tai suunnitelmaa. Lyhyesti sanottuna projekti koostuu useasta ihmisestä ja mahdollisesti muista resursseista, jotka on väliaikaisesti koottu yhteen toteuttamaan tietynlaista tehtävää. Projektilla on myös usein budjetti ja aikataulu. (Ruuska 2007, 18-19.)

Ruuska (2007, 19-20) mainitsee seuraavia tyypillisiä asioita projekteille:

- Projektilla on yksi tai useampi tavoite. Projekti päättyy, kun tavoitteet on saavutettu.
- Projektilla on oltava päätepiste, joka on määritelty etukäteen.

- Projekti on järkevästi rajattu kokonaisuus, jossa vastuu on keskitetty yhteen pisteeseen, vaikka mukana olisi monia osapuolia.
- Jotta projektin tavoitteet saavutettaisiin, edellytetään usein ryhmätyöskentelyä. Ryhmän jäsenet voivat koostua eri organisaatioista.
- Projektin aikana havaitaan useita eri vaiheita niin kasvusta kypsymiseen kuin kuihtumiseen. Projekti on myös oppimisprosessi. Siitä saatavaa kokemusta ja tietoa tulisi aina hyödyntää seuraavissa hankkeissa.
- Kun projekti on meneillään, ei voida varmaksi tietää mitä projektin seuraavassa vaiheessa tapahtuu. Yksityiskohdat työn edetessä tarkentuvat, joten edellisen vaiheen tulokset vaikuttavat aina seuraavan vaiheen tehtäviin.
- Projektin takana on aina asiakkaan tilaus. Asiakas määrittelee projektille vaatimuksia ja reunaehdoja. Ne rajaavat projektin toimintaa.
- Jokaiseen projektiin liittyy aina riskejä ja epävarmuutta, mutta ne kuuluvat projektityön luonteeseen. Riskien määrä on verrannollinen siihen kuinka projekti viedään sen eri vaiheissa läpi. Jos projekti on rajattu epäselvästi ja suunniteltu huonosti, riskien määrä ja niiden todennäköisyys kasvaa. (Ruuska 2007, 20)

Yksi tärkein projektin määrittelevä asia on ainutlaatuisuus. Projekti sisältää aina omat erityispiirteensä ja poikkeaa siten muista vastaavista hankkeista. (Kettunen 2003, 16)

Projektin lopputuloksena ei aina tarvitse olla mikään konkreettinen tuote, vaan se voi olla myös ratkaisu johonkin ongelmaan (Ruuska 2007, 20).

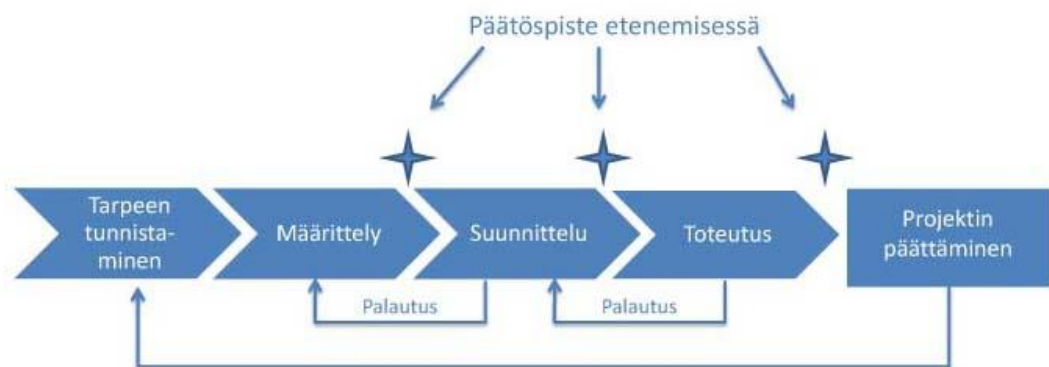
Kettusen (2003, 16) mukaan projektien onnistumisen ja etenemisen kannalta on tärkeää, että niitä vastuutettaisiin. Vastuuttaminen merkitsee hänen mukaansa sitä, että projektin läpivieminen annettaisiin yhden henkilön esim. projektipäällikön vastuulle. Täten tiedettäisiin kenen puoleen kääntyä, jos projektin etenemisessä ilmenee ongelmia. Jaettu vastuu saattaisi aiheuttaa tilanteen, jossa lopulta

kenelläkään ei olisi päävastuuta projektin vetämisestä tai syntyneistä tuloksista.
(Kettunen 2003, 16.)

2.2 Projektin vaiheistaminen

Projekti jaksetaan useaan ajallisesti peräkkäiseen vaiheeseen (Pelin 2002, 110). Ne seuraavat toisiaan tai saattavat olla osittain päällekkäisiä. Tavallisesti projektit etenevät suoraviivaisesti vaiheesta toiseen. Projektissa on kuitenkin mahdollista palata edelliseen vaiheeseen, jos tarvitaan vielä kehittelyä tai tulokset sitä vaativat.
(Kettunen 2003, 41.)

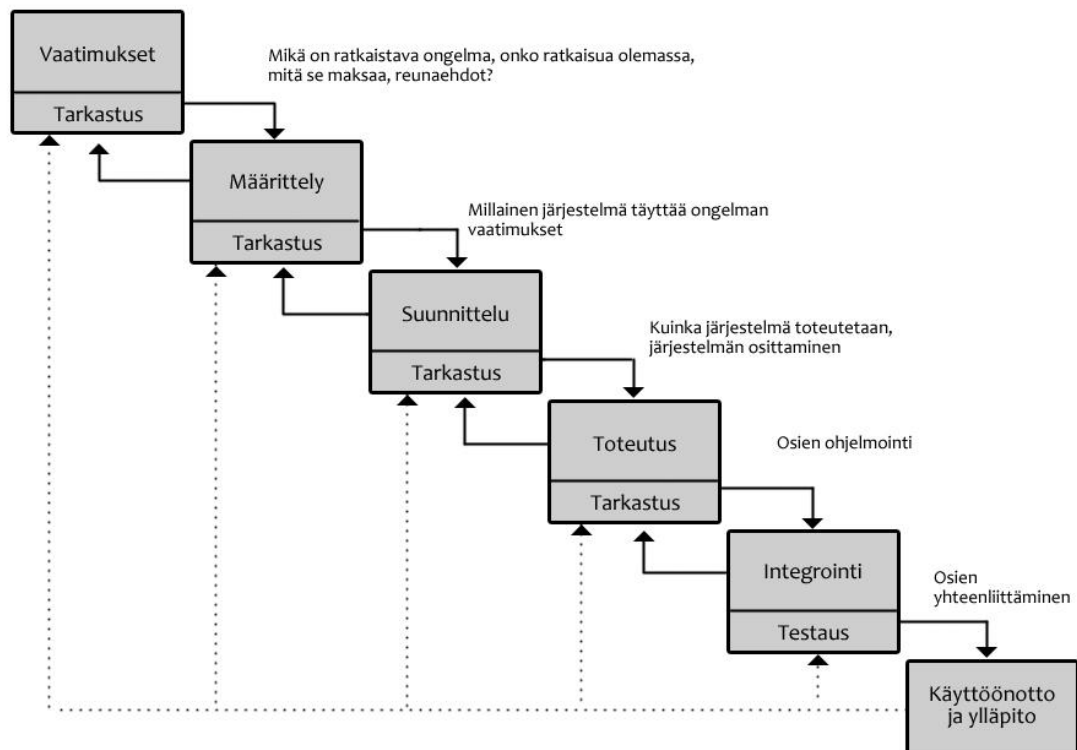
Seuraavassa Kuviossa 1 esitellään, miten projekti useimmiten etenee.



KUVIO 1. Kuvio projektin yleisestä etenemisestä (ks. alkuperäinen kuvio: Kettunen 2003, 41)

Tässä mallissa oletetaan, että vaiheet seuraavat toisiaan ja niissä on mahdollisuus palata tarvittaessa edellisiin vaiheisiin. Projekti lähtee liikkeelle siitä, että ensin sille on tunnistettu tarve tai idea. Määrittelyvaiheessa arvioidaan, onko se tarpeeksi hyvä ja kannattava toteutettavaksi. (Kettunen 2003, 41.)

Toinen hyvä vaihejakomalli, jota käytetään enemmän ohjelmistotuotannossa on niin sanottu vesiputousmalli. Malli on esitelty kuviossa 2. Mallista on tehty useita eri muunnelmia, mutta useimmiten niistä voidaan erottaa ainakin määrittely-, suunnittelu- ja toteutusvaiheet. Lisän voidaan erottaa testausvaihe, jota kuviossa 1 ei ole. Määrittelyvaihetta edeltää useimmiten esitutkimukseksi tai tarvekartoitukseksi kutsuttu vaihe. (Haikala & Märijärvi. 2006, 36-37.)



KUVIO 2. Esimerkki vesiputousmallista (ks. alkuperäinen kuvio: Haikala ym. 2006, 36)

Esitutkimuksen ja tarpeen tunnistamisvaiheen tehtävänä on asettaa yleiset järjestemätason vaatimukset. Näitä vaatimuksia kutsutaan usein asiakasvaatimuksiksi. Ne määrittelevät asiakkaan tarpeet, mutta eivät ota kantaa millainen järjestelmä vastaa asiakkaan vaatimuksia. Esitutkimus kertoo miksi ohjelmisto tai järjestelmä tulisi toteuttaa tai miksi sitä ei kannattasi tehdä. Usein esitutkimuksen ajatellaan olevan osa määrittelyvaihetta, koska asiakkaan tarpeiden analysointi ja tarkentaminen jatkuu yleensä koko määrittelyvaiheen ajan. (Mts. 37.)

Määrittelyvaiheessa (vaatimusmäärittely) asiakkaan vaatimuksia analysoidaan. Niistä johdetaan ohjelmistovaatimukset, joissa määritellään toteutettava järjestelmä. (Mts. 38.)

Kun määrittelyvaiheen tulokset ovat tarpeeksi mielekkäitä ja kannattavia, siirrytään itse suunnitteluvaiheeseen. Tässä vaiheessa tarkennetaan määrittelyvaiheen tuloksia ja tavoitteet asetetaan konkreettisiksi suunnitelmiksi. Suunnitteluvaiheen tuloksena syntyy muun muassa projektisuunnitelma. Tähän dokumenttiin kirjataan, kuinka tavoite toteutetaan käytettävissä olevilla resursseilla, aikataululla ja budjetilla. (Kettunen 2003, 42.)

Ilman kunnollisia määrittelyvaiheen tuloksia suunnitteluvaiheeseen ei voi onnistua kunnolla. Määrittelyssä yksityiskohtia tarkennetaan niin kauan, kunnes suunnitteluvaiheen edellytykset ovat täyttyneet. (Mts. 42)

Usein sanotaan, että määrittelyvaihe vastaa kysymykseen ”mitä” järjestelmä tekee. Suunnitteluvaiheessa taas vastataan kysymykseen ”miten” se tehtävänsä suorittaa. (Haikala ym. 2006, 40.)

Yleensä suunnittelun jälkeen projektista on tehty projektisuunnitelma, joka ihanteellisesti sisältäisi aikataulun, budjetin, riskianalyysin, resurssianalyysin, projektiorganisaation, työsuunnitelman sekä viestintä- ja

dokumentointisuunnitelman. Tässä vaiheessa on vielä mahdollista keskeyttää projekti, sillä toteutusvaiheen alkaessa keskeyttäminen saattaa tulla kalliiksi. Jos projektin päätöspisteissä havaitaan epävarmuutta ja epäselvyyttä projektin tavoitteista tai toimintamalleista, projektin suunnittelua täytyy jatkaa ennen siirtymistä toteutukseen. (Kettunen 2003, 42.)

Toteutusvaihe seuraa tehtyä suunnitelmaa. Tämä tavoite kuitenkin harvemmin toteutuu käytännössä, koska projekti elää aina jonkin verran. Tällöin projektisuunnitelmaa tulee päivittää tai muuttaa. Projekti voidaan myös palauttaa hetkeksi suunnitteluvaiheeseen, jos sen perusteet muuttuvat esimerkiksi tavoitteiden tai budjetin suhteen. Toteutusvaiheen tuloksena syntyy projektisuunnitelmassa määritelty tuotos. (Mts. 42.)

Testausvaiheen tarkoituksena on löytää ohjelmistosta virheitä ja poistaa ne. Testaus toteutetaan usein monella tasolla niin sanotun V-mallin mukaisesti. V-mallissa testaus jaetaan moduulitestaukseen, integrointitestaukseen ja järjestelmätestaukseen. Modulitestaus etsii virheitä yksittäisistä moduuleista, integrointitestauksessa vikoja etsitään moduulien yhteistoiminnasta ja järjestelmätestauksessa taas koko järjestelmän toiminnoista ja suorituskyvystä. (Haikala ym. 2006, 40.)

Projektin viimeinen vaihe on projektin päättäminen. Päätös vaihe sisältää loppuraportoinnin, projektiorganisaation purkamisen ja mahdollisien jatkoideoiden esitlemisen. Monesti projektien sivutuotteena syntyy uusia projekti-ideoita ja ne kannattaa ottaa käsittelyyn sekä työstöön. (Kettunen 2003, 42-43.)

Laadunvarmistustoimenpiteet liittyvät olennaisesti kaikkiin vaiheisiin. Niitä ovat muun muassa tarkastukset, katselmukset ja testaukset. Tarkastuksien ja testauksien tarkoituksena on pyrkiä estämään virheitä järjestelmästä mahdollisimman varhaisessa vaiheessa. (Haikala ym. 2006, 37)

2.3 Tarpeen tunnistaminen projektissa

Kaikkien projektien alussa (ks. kuvio 2) projektin omistaja tunnistaa ensin tarpeen, joka täytetään projektin kautta. Jotkut projekteista perustetaan esimerkiksi asiakkaan tilauksesta, sisäisen idean perusteella tai kehitystarpeen seurauksena. (Kettunen 2003, 46.)



KUVIO 3. Kuvio tarpeentunnistamisvaiheen paikasta projektin etenemisessä (ks. alkuperäinen kuvio: Kettunen 2003, 46)

Jokaisella projektilla on oltava omistaja tai ohjausryhmä. Heille raportoidaan projektin etenemisestä. Projektin omistajan nimeäminen esimerkiksi organisaatiossa on tärkeää, jotta saadaan projektille henkilö, joka on kiinnostunut saaduista tuloksista. Useimmiten projektin omistaja on se henkilö, jolle projektin tulos luovutetaan projektin päättyttyä. (Mts. 47.)

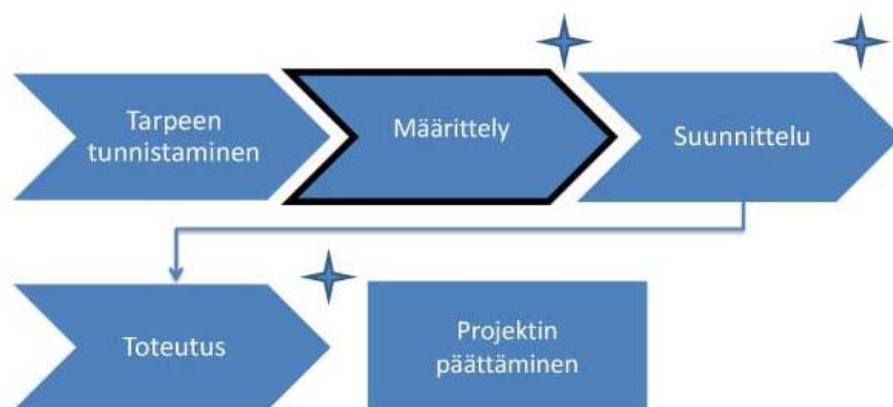
Kun projektilla on asiakas, saadaan tehtävänanto ja muut tarkat määrittelyt asiakkaalta (mts. 47).

Kettunen (2003, 47) toteaa, että projektin perustamisvaiheessa tulee olla hyvin tarkkaan mietittynä, miksi projektiin ryhdytään, mitkä ovat odotukset sen suhteen ja mitkä ovat käytettävät resurssit. Hänen mukaansa projektien perustaminen pohjautuu turhan usein liian vähäiseen pohjatyöhön. Jos projektia ei ole valmisteltu hyvin jo heti alussa, se on tuomittu epäonnistumaan jo ennen käynnistysvaihetta. (Kettunen 2003, 47.)

2.4 Projektin määrittelyvaihe

Useasti ennen projektin suunnittelun käynnistämistä tarvitaan erillinen määrittelyvaihe. Tämä vaihe sijoittuu tarpeentunnistamis- ja suunnitteluvaiheen väliin. (ks. kuvio 3)

Tämän vaiheen tavoitteena on selventää, mitä hankkeen lopputuloksena halutaan saada. Määrittelyvaiheessa tutkitaan myös vaihtoehtoisia toimintamalleja ja mietitään, kuka kyseessä olevan projektin voisi tehdä. (Kettunen 2003, 48.)



KUVIO 4. Kuvio määrittelyvaiheen paikasta projektin etenemisessä (ks. alkuperäinen kuvio: Kettunen 2003, 48)

Pienissä ja tavoitteeltaan selkeissä hankkeissa tämä vaihe voidaan sisällyttää osaksi tarpeen tunnistamis- tai suunnitteluvaihetta. Määrittely voi olla siis osana vaihetta, jossa projektia suunnitellaan syvällisemmin. (Mts. 48.)

Tarpeellinen määrittelyvaihe on etenkin silloin, jos projektin tavoite on jäänyt epäselväksi. Vaiheen läpivienti antaa tietoja myös siitä, jatketaanko hankkeen kanssa vai ei. Tämän vaiheen kustannukset ovat yleensä pieniä. (Mts. 48-49.)

Kettusen (2003, 49) mukaan määrittelyvaiheen tuloksena syntyvät raamit tehtävälle projektille:

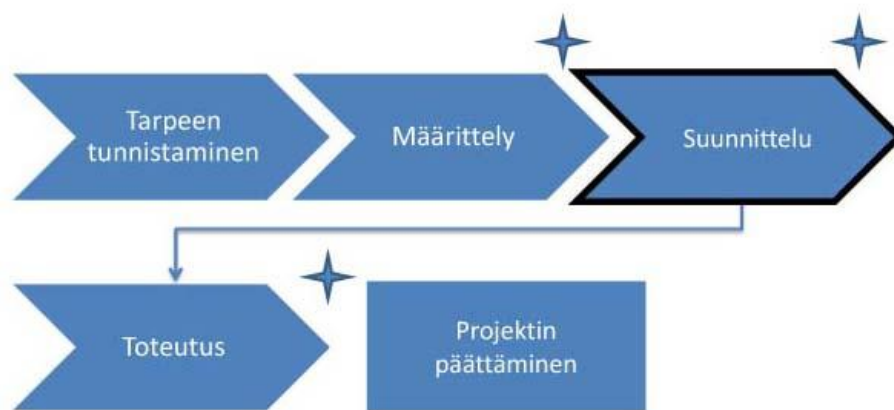
- Kuka omistaa projektin?
- Missä ympäristössä työ tehdään sekä mihin ympäristöön tulokset tulevat?
- Minkä ratkaisun ja mitä uutta projekti tuottaa?
- Mitä resursseja projektiin on käytettävissä?
- Projektin yhteydet toisiin projekteihin tai kehityshankkeisiin?
- Millaisia tuloksia projektilta odotetaan?
- Mikä on projektin aikataulu?
- Millaiset ovat projektin taloudelliset rajat?

Määrittelyvaiheen tuloksena syntyy muun muassa toiminnallinen määrittelydokumentti (Vaatimusmäärittely). Muita mahdollisia tulosdokumentteja ovat alustava testaussuunnitelma, toteutusprojektin projektisuunnitelma, alustava käyttöohje ja käyttötapauskuvaukset ja-kaaviot, sekä tarvittaessa luokkakaavio ja luokkakuvaukset. (Haikala ym. 2006, 73, 78-81.)

Projektin omistaja johtaa määrittelyä. Vaiheen lopuksi tehdään päätös siitä, eteneekö projekti suunnitteluvaiheeseen. (Kettunen 2003, 49.)

2.5 Projektin suunnitteluvaihe

Yksi tärkeimmistä vaiheista projektin aikana on suunnitteluvaihe, joka sijoittuu määrittely- ja toteutusvaiheen väliin (ks. kuvio 4). Tässä vaiheessa kiinnitetään suurin osa kustannuksista ja resursseista. Itse toteutusvaiheessa on hankala enää karsia kustannuksia tai vaihtaa resurssien käyttöä. Tämän takia suunnitteluvaiheeseen on hyvä varata paljon aikaa. (Kettunen 2003, 49-50.)



KUVIO 5. Kuvio suunnitteluvaiheen paikasta projektin etenemisessä (ks. alkuperäinen kuvio: Kettunen 2003, 49)

Suunnitteluvaiheessa voidaan useimmiten syventää projektille määrittelyvaiheessa asetettuja vaatimuksia. Suunnitteluvaiheessa varmistetaan myös se, että projektin tilaajalla ja toteuttajalla on samat näkemykset työn lopputuloksesta.

Projektsuunnitelmaa tehtäessä täytyy olla selvillä, että molemmat osapuolet - projektin tilaaja ja tekijä - ovat samaa mieltä siitä, miten ja milloin asiat tehdään.

(Mts. 50.)

Kokonaisuuden hahmottaminen on tärkeää, sillä se auttaa ymmärtämään hankkeen etenemistä ja eri työtehtävien tarpeellisuutta. Kun lopputulos ja tavoite ovat selkeästi tiedossa, myös projektin jäsenten on helpompi motivoitua työhön. Hyvä suunnittelu lisää yhteisymmärrystä ja selventää tavoitteita. Se myös lisää tehokkuutta ja kommunikaatiota. Täten hyvä suunnittelu vähentää myös epävarmuutta ja riskejä. (Mts. 51.)

Hyvä suunnitelma elää koko projektin ajan ja se ei ole siksi koskaan valmis. Suunnitelmaa tulee päivittää jatkuvasti ja tehdä siihen muutoksia tarvittaessa. Projekti menettää merkityksensä ja siltä katoaa helposti yksi ohjausmekanismi, jos projektisuunnitelma jätetään päivittämättä. (Mts. 51.)

Kettusen (2003, 51) mukaan suunnitteluvaihetta kritisoidaan joskus sen vaatiman ajan ja kustannusten takia. Kuitenkin, jos projektisuunnitelma on tehty huonosti tai ylimalkaisesti, voi se kostautua ylimääräisinä kustannuksina projektin myöhemmissä vaiheissa.

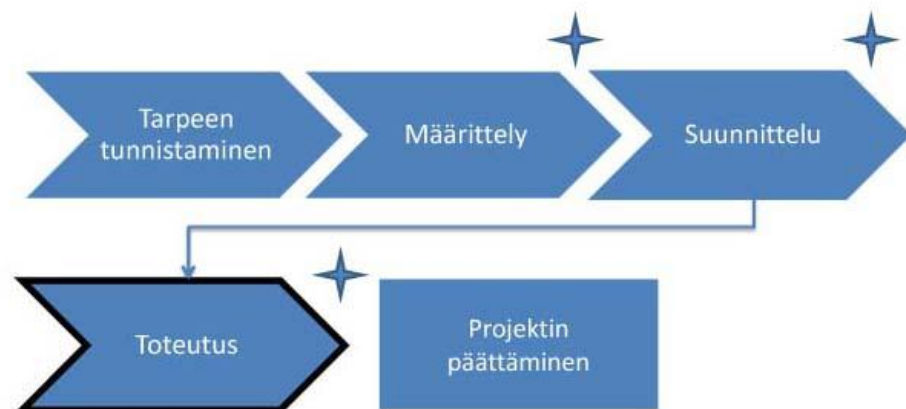
Suunnitteluvaiheessa määrittelyssä kuvattujen ohjelmistovaatimuksien toteutus kuvataan määrittelemällä tekniset vaatimukset järjestelmään. Teknisistä vaatimuksista ilmenee, miten ohjelmistovaatimukset täyttävä järjestelmä toteutetaan. (Haikala ym. 2006, 40)

Suunnitteluvaiheen dokumentteina ja tuloksina syntyvät esimerkiksi näyttökartta/moduulihahmotelma, visualisoinnit, testaussuunnitelma ja testitapaukset sekä Tietokantaohjelmoinnin puolelta tietokanta ja SQL-lauseet (Haikala ym. 2006, 73, 81-83, 283).

2.6 Projektin toteutusvaihe ja läpi vieminen

2.6.1 Toteutusvaihe

Projektin toteutusvaiheeseen siirrytään suunnitteluvaiheen lopussa, kun käynnistyspäätös on tehty (ks. kuvio 5). Käynnistyspäätöksen antaa yleensä projektin omistaja tai ohjausryhmä. Päätös tehdään suunnitteluvaiheen dokumentaatioiden ja tietojen pohjalta. (Kettunen 2003, 142)



KUVIO 6. Kuvio toteutusvaiheen sijainnista projektin etenemisessä (ks. alkuperäinen kuvio: Kettunen 2003, 142)

Toteutuksen käynnistyessä täytyy projektin toteuttajalla olla tiedossaan, mitä tehdään, miten ja mitkä ovat resurssit. Kannattaa kuitenkin tarkistaa vielä, että suunnitteluvaiheessa tehdyt suunnitelmat ja resurssit pitävät paikkansa. Sen jälkeen voidaan alkaa miettiä, millä toimenpiteillä projektia lähdetään käynnistämään. (Mts. 142.)

Usein käytetty ja hyväksi havaittu tapa käynnistää hanke on järjestää projektin käynnistyspalaveri. Siinä ryhmänjäsenet tapaavat toisensa ja kertaavat projektin tavoitteet sekä jakavat tehtävät keskenään. (Mts. 142.)

Toimivana oman työn johtamisen välineenä voi käyttää keskeneräisten töiden listaa ja sen ylläpitoa. Tällaisia muistilistoja on hyvä pitää, sillä ilman niitä on vaikea muistaa kaikkia tehtäviä töitä. Muistilistaan voi myös kerätä niitä asioita, joita voi jakaa toisten ryhmän jäsenten kesken. (Mts. 144.)

Toteutusvaiheen tuloksena syntyy aikaisemmissa vaiheissa määritelty tuotos.

2.6.2 Resurssien hallinta

Yksi tavanomainen haaste projekteissa on resurssien hallinta. Voidaan sanoa, että melkein kaikissa projekteissa ilmenee ongelmia resurssien saatavuudesta tai käytöstä. Resurssit, jotka yleisimmin nostattavat keskustelua, ovat Kettusen (2003, 148) mukaan

- projektin tekijät
- osaaminen
- laitteistot ja kalustot
- materiaali.

Useissa hankkeissa työskennellään liian pienten tekijäresurssien kanssa, mikä on ongelmallista joissain tilanteissa. Projektille ei aina välttämättä ole saatavilla riittävästi tekijöitä, joilla on juuri oikea osaamistausta. Joskus taas asia voi olla toisinpäin eli osaajia löytyy, mutta heidän taitonsa ei sovellu kyseisiin tehtäviin. (Mts. 148.)

Puutetta voi myös tulla hankkeen vaatimista laitteista tai kalustosta (mts. 148).

2.6.3 Muutosten hallitseminen

Kaikissa projekteissa tulee eteen tilanteita, jolloin aikaisemmin tehtyjä suunnitelmia täytyykin muuttaa. Kettunen (2003, 149) toteaa muutoksien johtuvan monista syistä ja listaa muutamia:

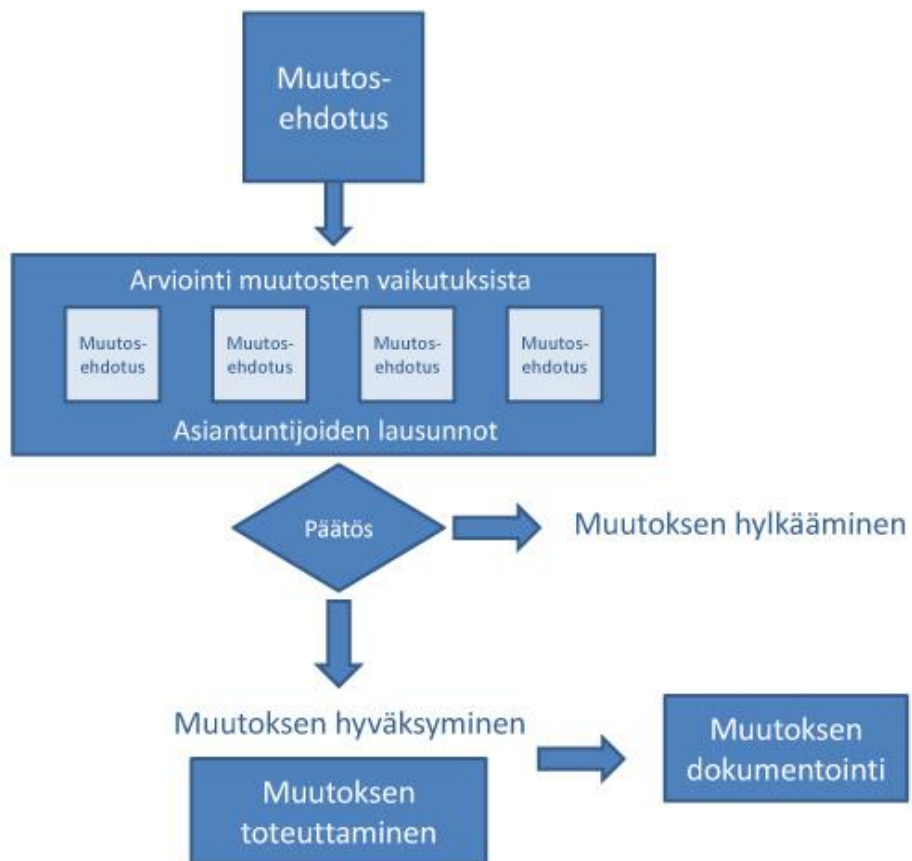
- Projektin tilaaja saattaa oppia uutta projektin aikana ja osaa vaatia muutoksia. Työn aikana tilaajan ymmärrys kasvaa ja hän pystyy itsekin hahmottamaan lopputuloksen paremmin ja siksi osaa vaatia muutoksia.
- Projektiympäristö tai sen tavoitekin saattavat muuttua suunnittelun ja työn tekemisen välillä.
- Suunnitelmissa voi olla puutteita tai muuta muutettavaa. Joskus voidaan havaita, että projektin alussa on tehty selviä suunnitteluvirheitä tai suunnittelussa on puutoksia. Aina isojen projektien suunnitteluvaiheessa ei kaikkia yksityiskohtia voida suunnittelemaan heti täsmälleen oikein.

Muutosten teko ei saa olla kuitenkaan täysin hallitsematonta, vaan suurten muutosten tekeminen vaatii hyväksymismenettelyn (Pelin 2002, 211).

Muutosten hallinnan vaiheet ovat Pelinin (2002, 211) mukaan seuraavanlaiset:

1. Muutosehdotuksen laadinta
2. Arviointi muutosten vaikutuksista
3. Lausunnot asiantuntijoilta
4. Muutosten käsittely: hyväksytäänkö vai hylätäänkö muutokset
5. Muutoksen suorittaminen
6. Muutoksen dokumentointi
7. Muutoksesta tiedottaminen

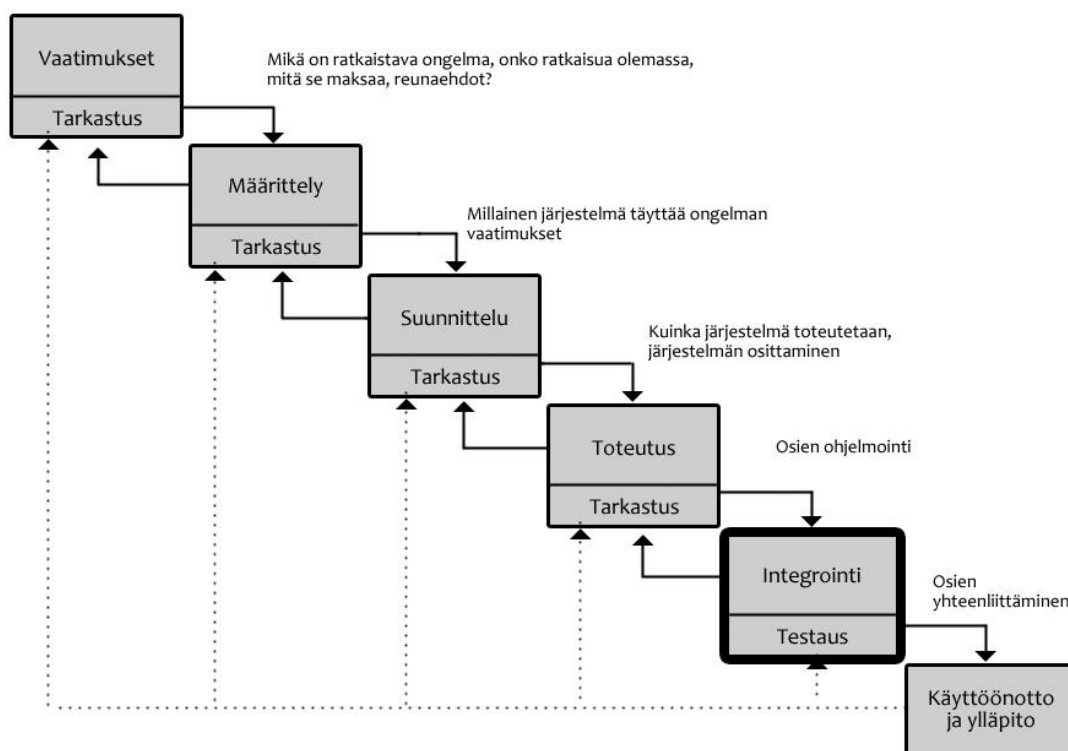
Kuviossa 6 havainnollistetaan mainittuja vaiheita.



KUVIO 7. Muutosten hallinnan vaiheita (ks. alkuperäinen kuvio: Pelin 2002, 212)

2.7 Testausvaihe

Testausvaihe sijoittuu yleensä toteutusvaiheen jälkeen, mutta saattaa olla myös päällekkäinen toteutusvaiheen kanssa (Katso kuvio 8). Tämä johtuu siitä, että järjestelmää voidaan testata samanaikaisesti, kun sitä toteutetaan.



KUVIO 8. Testausvaiheen sijainti projektin etenemisessä. (ks. alkuperäinen kuvio: Haikala ym. 2006, 36)

Testaukseen liittyviä työvaiheita ovat muun muassa testauksen suunnittelu (testaussuunnitelma, testitapaukset), testiympäristön luonti, testin suorittaminen ja tulosten tarkastelu. Testauksen läpivientiin kannattaa kiinnittää huomiota, sillä virheiden jäljitykseen ja korjauksiin saattaa kulua jopa yli puolet ohjelmistoprojektin resursseista. (Haikala ym. 2006, 283.)

Testaus määritellään perinteisesti suunnitelmalliseksi virheiden etsimiseksi suorittamalla ohjelmaa tai jotain sen osaa. Useimmiten testaus tapahtuu kokeilemalla ohjelmaa umpimähkäisesti jonkinlaisella syöttöaineistolla. Jos testaajana on ohjelman tekijä, on tärkeämpää osoittaa ohjelman toimivuus, kuin etsiä virheitä. Testauksen määrä (työtunnit ja testitapausten määrä) ei välttämättä ole suoraan verrannollinen testauksen tehokkuuteen. Testaus muutassa tunnissa

pienellä hyvin suunnitellulla testitapausjoukolla voi olla tehokkaampaa ja johtaa parempaan tulokseen kuin päiväkausia kestävä umpimähkäinen kokeilu. (Mts. 284.)

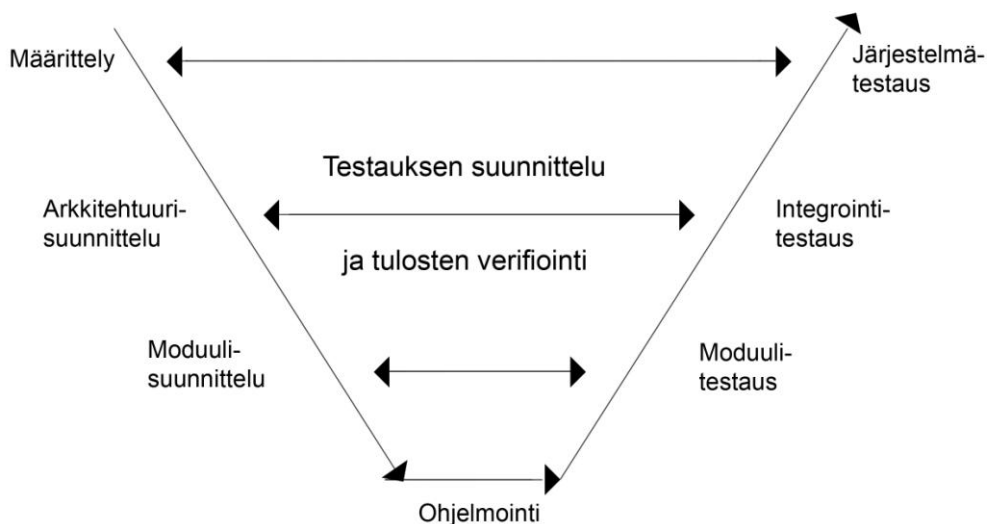
Testauksen avulla voidaan siis osoittaa, että ohjelmassa on virheitä. Ohjelman virheettömyyttä sillä taas ei voi voida osoittaa edes yksinkertaisissa tapauksissa. Ohjelman testauksessa pystytään kattamaan käytännössä vain pieni osa kaikista mahdollisista tilanteista. Tämä ei tarkoita sitä etteikö testaukseen kannattaisi panostaa. Ei vaan kannata liian paljon luottaa ohjelman toimivuuteen hyvistä testaustuloksista huolimatta. Tarkastuksilla ja katselmuksilla pyritään avustamaan projektin elinkaaren etenemistä niin, että eteneminen olisi näkyvää ja poistamaan mahdolliset virheet tuotteessa aikaisessa vaiheessa. Tarkastelun seurauksena ohjelman huolellinen suunnittelu ja ohjelmoinnin merkitys korostuu. (Haikala ym. 2006, 287.)

Aloittelevat ohjelmoijat tekevät yleensä tavanomaisen virheen Haikalan ym. (2006, 287) mukaan siinä, että he alkaavat koodaamaan äkkiä ja debuggaavat virheet sitten pois. Tämä on toimintapa on ilmeisen väärä.

Testaustasot

Testaustasoja on erilaisia muun muassa V-mallin mukaisesti moduulitestaus, yksikkötestaus, integrointitestaus ja järjestelmätestaus (Katso kuvio 9).

Järjestelmätestausta voi seurata erillinen kenttätestaus ja/tai hyväksymistestaus, kertoo Haikala ym. (2006, 288).



KUVIO 9. Testaustasot käsittävä V-malli (ks. alkuperäinen kuvio: Haikala ym. 2006, 289)

Kuten kuviosta 9 on nähtävissä, testauksen suunnittelu tapahtuu testaustasoa vastaavalla suunnittelutasolla. Järjestelmätestaus suunnitellaan määrittelyvaiheen aikana, integrointitestaus suunnitteluvaiheessa ja moduulitestaus moduulisuunnitteluvaiheessa. Tulokset taas todetaan oikeanlaisiksi vertaamalla niitä vastaaviin dokumentteihin. (Mts. 289.)

Haikalan ym. (2006, 289) mukaan moduulitestaus tarkoittaa sitä, että testattavana on yksittäinen moduuli. Moduuli saattaa koostua noin 100-1000 ohjelmarivistä. Sen toimintaa verrataan moduulisuunnittelun ja arkkitehtuurisuunnittelun tuloksiin eli yleensä tekniseen määrittelydokumenttiin. Useimmiten testauksen suorittaa moduulin tekijä. Jotta moduulin toimivuutta voidaan kokeilla testauksen aikana, joudutaan usein tekemään testipetejä. Niihin voi kuulua ohjelman ympäristöä simuloivia osia esim. testiajureita ja tynkämoduuleita. Testiajureiden avulla on mahdollista kutsua moduulin toteuttamia palveluita ja tarkastella tuloksia.

Tynkämoduuleilla taas korvataan testattavan moduulin tarvitsemat muut moduulit ellei niitä jo ole olemassa. (Haikala ym. 2006, 289.)

Integroititestauksesta Haikala ym. (2006, 290) kertovat, että siinä moduuleita yhdistellään yhteen tai moduuliryhmiä. Tärkeintä on siis moduulien välisten rajapintojen toimivuuden tutkiminen. Testauksesta saatuja tuloksia verrataan useimmiten tekniseen määrittelyyn. Integroititestaus etenee yhdessä samaan aikaan moduulitestauksen kanssa. Usein sitä onkin tarpeetonta tarkastella erillään moduulitestauksesta. Toisaalta, jos testauksesta haluaa kattavan, on moduulitestauksessa helpompi saavuttaa kunnollinen testikattavuus, kertovat Haikala ym. (2006, 290). Tämä johtuu heidän mukaansa siitä, kun testattava kokonaisuus kasvaa alkaa olla vaikea käydä läpi kaikkea koodia. Integrointi etenee yleensä alimman tason moduuleista ylöspäin. Jäsentävässä integroinnissa etenemissuunta on päinvastainen. (Mts. 290.)

Järjestelmätestauksessa on koko järjestelmä tarkastelun kohteena. Tuloksia verrataan määrittelydokumentaatioon (ohjelmiston toiminnalliseen määrittelyyn) ja asiakasdokumentaatioon. Kenttätestaus ja hyväksymistestaus voi myös olla osana järjestelmätestausta. Järjestelmätestauksen suorittajina täytyy olla henkilöitä jotka ovat kehitystyöstä mahdollisimman riippumattomia. Järjestelmätestauksen kohteena ovat myös järjestelmän ei-toimimalliset ominaisuudet: kuormitustestit, luotettavuustestit, asennustestit, käytettävyytestit ja niin edelleen. (Mts. 290.)

Haikala ym. (2006, 290) kertoo, että mitä korkeammalle V-mallin testaustasolle mennään, sitä kalliimmaksi virheiden korjaus tulee. Joskus virheiden korjaus voi aiheuttaa lisää virheitä. Esimerkiksi jos järjestelmätestauksessa havaitaan virhe ja se korjataan, voi korjaus aiheuttaa muutoksia useisiin moduuleihin. Tämän takia olisikin hyvä tehdä regressiotestaus eli uudelleentestaus. Siinä testataan myös muutkin moduulit ja lopuksi suoritetaan järjestelmätestaus uudelleen, jotta voidaan havaita onko jokin muutostarve jäänyt huomaamatta. (Mts. 290.)

Käytettävyytestauksella yritetään varmistaa se, että pystyykö käyttäjä suoriutumaan hyvin tehtävistä, joiden suorittamista varten järjestelmää ollaan toteuttamassa.

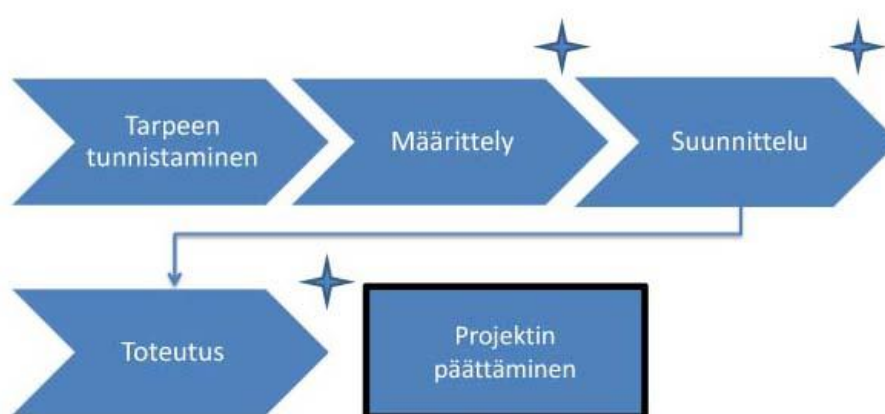
Testaus on tavallisesti järjestelmän käyttöliittymän testausta, jota toteutetaan jo mahdollisesti määrittelyvaiheessa käyttöliittymän prototyypin avulla. Joskus käytettävyytestauksen sijasta tai lisäksi voidaan käyttää hyväksi ohjelmistojen käytettävyyteen erikoistuneita ammattilaisia. (Mts. 291.)

Testaussuunnitelmasta selviää muun muassa mitä testejä tehdään, milloin ja miten ne järjestetään ja millaisia lopputuloksia odotetaan (mts. 299).

2.8 Projektin päättäminen

Toteutusvaiheen jälkeen voidaan siirtyä projektin päättämiseen (ks. kuvio 10.).

Projektin päättäminen ei aina tarkoita vain loppuraportin kirjoittamista, vaan sen on myös päätyttävä päätöksenteon tasolla ja henkisesti. Aina ei kuitenkaan projektin loppuessa siitä tehdä virallista päätöstä. (Kettunen 2003, 169.)



KUVIO 10. Kuvio päätös vaiheen sijainnista projektin etenemisessä (ks. alkuperäinen kuvio: Kettunen 2003, 169)

Projektin virallinen päättäminen kuuluu ohjausryhmälle sekä projektin omistajalle, jotka kutsutaan koolle projektin päätteeksi. Sille esitetään projektin tulokset ja päätösesityksenä projektin päättäminen. Kun ohjausryhmä ja omistaja ovat nähneet tulokset, he voivat vaatia projektia täydennettäväksi tai hyväksyä tulokset ja siten päättää projektin. (Mts. 169.)

Joissain tapauksissa projekti voidaan päättää, vaikka se olisi kesken. Tällainen tilanne voi syntyä, jos projektin perusteet muuttuvat merkityksellisesti kesken kaiken. (Mts. 169.)

Päätösvaiheessa on suotavaa, että molemmat osapuolet, projektin tilaaja ja tekijä ovat joustavia. Työntekijän on jaksettava toteuttaa tilaajan viimeisiä toiveita, kun taas tilaajan on osattava jossain vaiheessa panna piste lisä- ja muutostöille ja todettava hankkeen tulokset riittäviksi. (Mts. 170.)

Monet projektit jättävät jälkeensä mahdollisuuksia jatkokehittämiseen. Uusia ideoita ja tarpeita saattaa syntyä jälkikäteen. Nämä ideat kannattaa kirjoittaa ylös seuraavien projektien pohjaksi. (Mts. 170.)

Projektin päättämisen jälkeen, kannattaa olla vielä tietyn ajan jälkeen yhteydessä tilaajaan ja varmistaa, että kaikki on hyvin. Tällainen jälkiseuranta antaa myönteisen kuvan ja kertoo projektiryhmän halusta tehdä hyvää työtä. (Mts. 170.)

2.9 Tulosten luovuttaminen

Projektin tulosten luovuttamisesta sovitaan jo ennen projektin käynnistämistä. Silloin tehdään selväksi millä tavoin ja millä hyväksymismenettelyllä tulokset hyväksytään. Projektin tilaaja tekee työlle hyväksymistestauksen ja tarkistaa, että työ vastaa vaatimusmäärittelyä sekä läpäisee kaikki toiminnalliset testit. (Kettunen 2003, 171.)

3 KOIRAKESKUS DOGGIEN AJANVARAUSJÄRJESTELMÄN TARPEIDEN MÄÄRITTELY

3.1 Tarpeen tunnistaminen

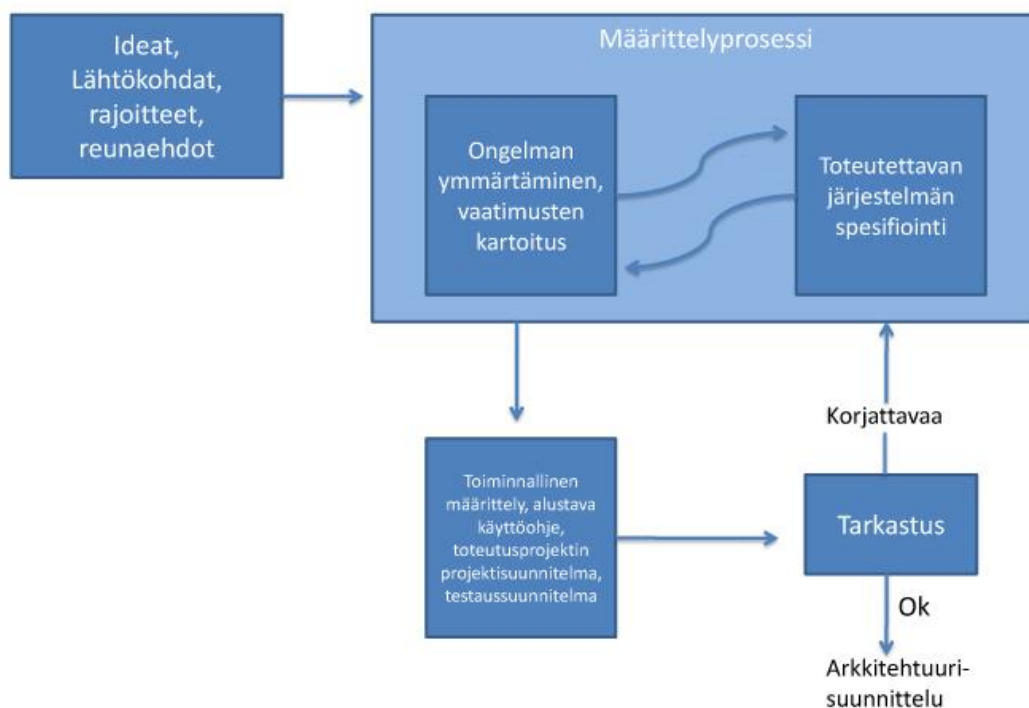
Opinnäytetyö lähti liikkeelle asiakkaan tilauksesta ja tarpeen tunnistamisesta. Asiakkaalla oli konkreettinen tarve saada helpotusta trimmaus- ja kurssiaikojen varaamiseen ja asiakastietojen käsittelyyn. Projektin omistajia ovat Koirakeskus Doggien työntekijät.

Koska projektilla oli asiakas, tehtävänanto ja muut tarkat määrittelyt tulivat asiakkaalta. Ensi alkuun opinnäytetyön tekijät pitivät asiakkaan kanssa aloituspalaverin, jossa kartoitettiin asiakkaan vaatimukset ja määriteltiin jo hieman toteutettavaa järjestelmää.

Kun ajanvarausjärjestelmän tarve ja toteuttamiskelpoisuus oli todettu, siirrettiin projekti määrittelyvaiheeseen.

3.2 Ajanvarausjärjestelmän määrittely

Alussa opinnäytetyöntekijät asettivat yhdessä asiakkaan kanssa tavoitteet ja vaatimukset sekä laativat vaatimusmäärittelydokumentin (Liite 1). Siihen on dokumentoitu järjestelmän vaatimukset ja järjestelmän kuvaus pääpiirteittäin. Järjestelmän vaatimuksia kartoitaessa sille asetettiin yleisiä tavoitteita kuten maksuttomuus, toimivuus internetissä eri selaimilla, mahdollisuus selata kalenteria kännykkällä, helppokäyttöisyys ja niin edelleen. Kun ongelma oli ymmärretty ja vaatimukset kartoitettu, alkoi toteutettavan järjestelmän määrittely. Kuvio 11 selventää määrittelyprosessin kulkua.



KUVIO 11. Määrittelyprosessin kulku (ks. alkuperäinen kuvio: Haikala ym. 2006, 79)

Määrittelyn tuloksena syntyi Haikalan ym. (2006, 78-81) esimerkin mukaisesti vaatimusmäärittely, käyttötapauskaavio(t) ja käyttötapauskuvaukset (kts. liitteet 1, 2, 3, 4 ja 5).

3.2.1 Ajanvarausjärjestelmän toiminnot

Järjestelmään kirjaudutaan sisään ensin rekisteröitymällä sen käyttäjäksi Koirakeskus Doggien sivuilta löytyvän linkin kautta.

Sisäänkirjauduttua avautuu käyttäjälle Omat-tiedot -sivu jossa pääsee muokkaamaan tai korjaamaan omia tietojaan, jotka käyttäjä on syöttänyt järjestelmään rekisteröityessään. Siellä pystytään lisäämään järjestelmään koiria, muokkaamaan koiran tietoja, sekä poistamaan koiria järjestelmästä. Sivuston alareunaan on sijoitettuna linkki jota klikkaamalla asiakas pääsee varaamaan aikoja trimmauksiin ja koulutuksiin. Omat tiedot -sivun alareunaan listautuvat myös mahdolliset asiakkaan aikaisemmin varatut ajat muistutukseksi hänelle. Kuviossa 12 on havainnollistettu Omat tiedot -sivun näkymää.



Kirjautuneena: [Testaja](#) (id: 37)
[Kirjaudu ulos](#)
 - HYVÄKSI HAUKUTTU



Omat tiedot

Sähköposti:
 Etunimi:
 Sukunimi:
 Katuosoite:
 Postinumero:
 Postitoimipaikka:
 Puhelinnumero:

Koirat

[Suihke](#)
[Lisää koira](#)

Uusi salasana:
 Vahvista salasana:
 Vanha salasana:

Ajanvarauksesi

[Uusi trimmausajka](#)
 Ajanvarauksia ei vielä listata tässä

KUVIO 12. Esimerkki näkymä ajanvarausjärjestelmän Omat tiedot -sivusta

Itse ajanvarausta tehtäessä ensimmäisenä avautuu sivu, jossa valitaan halutut palvelut valitulle koiralle (katso kuvio 13). Asiakas ei saa itse päättää varauksen kestoja, vaan järjestelmä laskee sen automaattisesti valituista palveluista. Sen jälkeen asiakas pääsee valitsemaan ajankohdan palvelulle.



Kirjautuneena: [Testaaja](#) (id: 37)
[Kirjaudu ulos](#)
- HYVÄKSI HAUKUTTU

1. Valitse palvelu
2. Valitse aika
3. Vahvista varaus



(Listaukseen on tarkoituksella tullut myös koirien kuvia)

Mahdolliset palvelut koiralle **Smirre** (villakoirat):

☐ Trimmaus (22 min)
☐ Pesu (11 min)
☐ Punkin poisto (18 min)

Koko varauksen pituus: minuuttia

Edellinen
Seuraava

KUVIO 13. Esimerkki näkymä ajanvarausjärjestelmän Valitse palvelu -osiosta

Ajankohdan asiakas pääsee valitsemaan Valitse aika -osiosta. Asiakkaan valittavana on kaksi kalenteria. Asiakas saa itse päättää kumman työntekijän haluaa trimmaavan koiransa. Ajankohdan valinta tehdään etsimällä haluttu päivä ja klikkaamalla hiirellä vapaata aikaa. Kuvio 14 havainnollistaa kalenterinäköymää.

Ajan valinnan jälkeen varaus vahvistetaan erillisellä sivulla. Silloin on myös mahdollista perua varauksen tekeminen.

1. Valitse palvelu 2. Valitse aika 3. Vahvista varaus

(Tämänhetkisen päivän kaikki vapaat ajankohdat ovat varattavissa, vaikka ajankohta olisi jo mennyt)

Varauksen pituus: 22 minuuttia

13.04.2011 Päivä

Edellinen päivä 13.04.2011 Seuraava päivä

Dona	Hanna
8.00	8.00
8.30	8.30
9.00	9.00
9.30	9.30
10.00	10.00
10.30	10.30
11.00	11.00
11.30	11.30
12.00	12.00
12.30	12.30
13.00	13.00
13.30	13.30
14.00	14.00
14.30	14.30
15.00	15.00
15.30	15.30
16.00	16.00
16.30	16.30
17.00	17.00
17.30	17.30
18.00	18.00
18.30	18.30
19.00	19.00
19.30	19.30
20.00	20.00
20.30	20.30

Edellinen Seuraava



KUVIO 14. Esimerkki näkymä ajanvarausjärjestelmän Valitse aika -osiosta

Toiminnot löytyvät tarkemmin määriteltyinä Liitteistä 1, 2 ja 3.

3.2.2 Adminin toiminnot

Admin, eli pääkäyttäjä, käyttää kahta järjestelmää: Googlen kalenteria, joka on yhteydessä Koirakeskus Doggien ajanvarausjärjestelmään, ja omaa hallinnointijärjestelmäänsä. Googlen kalenterin kautta hän pystyy selailemaan omia ja asiakkaiden tekemiä varauksia sekä muokkaamaan ja poistamaan niitä. Admin voi varata Googlen kalenterin kautta itselleen aikaa, joihin ei halua varauksia.

Esimerkiksi, jos Koirakeskus Doggien työntekijät haluavat pitää lomaa tai lopettaa jonain päivänä työnsä aikaisemmin, voi Googlen kalenterin kautta varata haluttu ajanjakso, minkä jälkeen kyseinen aika ei ole enää asiakkaiden varattavissa.

Adminin toisen järjestelmän kautta pystyy hakemaan, lisäämään, poistamaan ja muokkaamaan asiakkaita. Lisäksi Admin voi lisätä järjestelmään kursseja ja poistaa niitä, sekä seurata kursseille ilmottautuneita. Käytössä on myös jonotuspaikkatoiminto. Sen avulla Admin pystyy lisäämään kursseille jonotuspaikalla olevia henkilöitä, jos kurssipaikkoja vapautuu. Hallinnointijärjestelmän kautta voidaan myös lisätä järjestelmään pankista saatuja maksuviitteitä, joita asiakkaat käyttävät kurssimaksujen suorittamisessa. Kuviossa 15 esitellään Adminin järjestelmän ulkoasua.

Et ole kirjautuneena sisään

HYVÄKSI HAUKUTTU

Doggie
KOIRAKESKUS

Kursien hallinta
Asiakashallinta
Lisää viitteitä

Asiakashallinta

Asiakashaku

Etunimi:

Sukunimi:

Puhelinnumero:

Hae

[Kukka Matilla](#)
[Kukka Kuvio](#)

KUVIO 15. Esimerkki näkymä Adminin käyttämästä järjestelmästä

Pääasiassa asiakkaalle tehtyihin ajanvarauksiin Admin käyttää samaa järjestelmää mitä asiakkaatkin, mutta eri kautta. Admin pystyy listaamaan asiakkaita Asiakashallinta -sivun kautta haku toiminnolla. Klikkaamalla asiakkaan nimestä muodostunutta linkkiä pääsee Asiakkaan tiedot -sivulle, jossa on erillinen linkki ajanvarauksen suorittamiseen.

Toiminnot löytyvät tarkemmin määriteltynä liitteistä 1, 4 ja 5.

4 TOTEUTUSVAIHTOEHTOJEN KARTOITUS

4.1 Kartoituksen tavoitteet

Kartoituksen tavoitteena oli selvittää, oliko helposti käyttöönotettavaa, mahdollisimman monta haluttua toiminnallisuutta sisältävää ilmaista järjestelmää jo olemassa, vai olisiko järkevämpää ohjelmoida järjestelmän runko itse.

Pääasiallinen ongelma kartoituksessa oli se, että vaaditun muokattavuuden tutkiminen on varsin hidasta. Oli olemassa olennainen riski siihen, että syvälinen järjestelmien vertailu söisi leijonanosan käytettävissä olevista resursseista, mutta käteen ei jäisi yhtään sopivaa järjestelmää. Tällöin ei jäisi jäljelle myöskään tarpeeksi resursseja järjestelmän ohjelmoimiseen itse. Eli kyseessä ei ollut syväluotaava vertailu, vaan nopea tarkistus siltä varalta, että haluttu järjestelmä olisi jo saatavilla.

Koska kartoituksen suorittajalla oli kokemusta teknologioista, joilla sovelluksen ohjelmoiminen olisi mahdollista, yksi vaihtoehto oli ohjelmoida runko itse. Pienen testisovelluksen jälkeen kävi selväksi, että kalenteri on ainoa komponentti, jonka ohjelmoiminen itse ei olisi realistinen tavoite käytettävissä olevien resurssien puitteissa.

Päätettiin, että mikäli sopivaa järjestelmää ei löydy riittävän nopeasti, siirrytään suosiolla ohjelmoimaan järjestelmän pohja itse PHP:llä ja XHTML:llä, mutta käytetään valmista kalenterikomponenttia.

4.2 Ajanvarausjärjestelmältä vaaditut toiminnot

Koska osa halutuista ominaisuuksista oli melko epätavallisia, kaikki vaaditut ominaisuudet ilman räätälöintiä sisältävän järjestelmän olemassaoloa ei pidetty todennäköisenä. Täten alusta lähtien yksi tärkeimmistä vaadituista asioista oli helppo muokattavuus ja laajennettavuus; jos nämä näyttivät puuttuvan, ei aikaa vaihtoehdon syvempään tutkimiseen tuhlatu.

Järjestelmältä vaaditut ominaisuudet:

1. Ajanvaraus lomakkeella, joka laskee varauksen pituuden dynaamisesti valittujen palveluiden perusteella, koiran rodun vaikuttaessa palvelun pituuteen.
2. Vain mahdollisen varausajankohtien näyttäminen käyttäjälle, sen sijaan että näytettäisiin kaikki varatut ajat.
3. Oma kalenteri kummallekin työntekijälle
4. Asiakastietokanta ja jokaiselle asiakkaalle mahdollisuus lisätä ja muokata koiria.
5. Asiakkaan tulee voida itse rekisteröityä järjestelmään ja varata aikoja ilman panostusta Koirakeskus Doggien puolelta.
6. Ilmainen

Mikäli jonkin yllä olevan ominaisuuden olemassaolo ei varmistunut kohtuullisella vaivalla, järjestelmää ei tutkittu enempää.

Lisäksi oli toivottavaa, että järjestelmän ulkoasua voisi muokata vastaamaan Koirakeskus Doggien olemassaolevaa nettisivua.

4.3 Ajanvarausjärjestelmien vertailu

Koska Koirakeskus Doggien nettisivut käyttävät Joomlaa, oli luonnollista aloittaa vaihtoehtojen kartoitus Joomla-lisäosista, osoitteesta

<http://extensions.joomla.org/>.

Kaikki testatut lisäosat hylättiin joko kaupallisuuden tai liialliseen erikoistumisen vuoksi, yleensä molempien. Erityisesti hotellinvarausjärjestelmiä oli saatavilla paljon, ja moni lisäosa olisi suoraan sopinut kurssien varaamiseen. Halutun kaltainen trimmausajan varaaminen koirille olisi kuitenkin vaatinut muokkaamista, ja lisäosien muokattavuus vaikutti olevan yleisesti ottaen heikkoa. Vertailun suorittajalla ei myöskään ollut kokemusta Joomla-lisäosista tai Joomla:sta yleensäkkään, eikä opettelu vaikuttanut olevan mahdollisesti hukkaan menevän ajan arvoista.

Myös muita kuin Joomla:lle tarkoitettuja sovelluksia tutkittiin. Periaatteessa esimerkiksi moni parturikampaamoille tarkoitettu ajanvarausjärjestelmä olisi saattanut toimia, mutta välttämättömänä vaatimuksena oli aikojen varaaminen asiakkaiden koirille, ei asiakkaille itselleen. Lisäksi nämä järjestelmät harvemmin olivat ilmaisia.

Loppujen lopuksi ilmaisia, kaikkia vaatimuksia täyttäviä järjestelmiä ei löytynyt. Koska kaikki vaaditut ominaisuudet olivat välttämättömiä, yhtäkään tutkituista järjestelmistä ei pidetty vaihtoehtona, eikä tarkempi vertailu järjestelmien välillä ollut tarpeen.

Koko vertailun ajan taustalla oli käsitys siitä, että ajanvaraus voidaan ohjelmoida myös itse. Täten vertailuun ei saanut mennä niin paljon resursseja ettei kaiken ohjelmointiin itse olisi enää aikaa. Tämä periaate johti siihen, että kun ilmeisimmät vaihtoehdot eivät vakuuttaneet, katsottiin järkeväksi aloittaa ohjelmointi itse.

4.4 Kalenterin valinta

Vaikka sopivia valmiita ajanvarausjärjestelmiä ei löytynyt, kalenterin ohjelmoimiseen itse ei ollut resursseja. Tarvittiin jokin valmis vaihtoehto.

Ensin päätettiin, mitä kalenterilta halutaan. Tavanomaisten kalenteritoimintojen lisäksi kalenterissa tuli olla PHP-rajapinta, joka tukee seuraavia asioita:

- Vapaan ajan palauttaminen kalenterista vapaa/varattu-formaatissa, ilman että paljastetaan mahdollista muuta tietoa.
- Ajan varaamisen tulee olla mahdollista rajapinnan kautta ilman tunnusten luomista asiakkaalle kalenteriin
- Nopeasti käyttöönotettava

Kalenterin tuli olla ilmainen. Lisäksi oli toivottavaa, että kalenteri tukisi seuraavia asioita:

- Kalenterin tarkastelu ja käyttö kännykällä

Ensimmäinen vaihtoehto ulkopuoliseksi kalenteriksi oli jo suosion vuoksi Googlen kalenteri. Kalenterille oli hyvä PHP-rajapintadokumentointi, joista selvisi nopeasti se, että se tuki kaikkia kalenterilta vaadittavia ominaisuuksia.

Google Apps -palvelusta, joka sisältää muun muassa Gmailin ja kalenterin, on ilmaisversio 50 käyttäjän rajoituksella. Lisäksi mahdollisen oman domainin sähköpostiosoitetta voi käyttää Gmailissa. (Google Inc. 2011a)

Maksullinen Google Apps Business lisää sähköpostilaatikon tilaa, sisältää puhelintuen ja joitakin suunnitellun järjestelmän kannalta merkityksettä palveluita. Hinta tälle olisi 40 euroa/vuosi/käyttäjä (Google Inc. 2011a.) Maksullisen version ominaisuuksia ei pidetty hinnan arvoisina.

Vaikka muitakin kalentereita olisi ollut, Googlen kalenteri tuki kaikkia haluttuja ominaisuuksia ilmaiseksi, joten pidemmälle tutkimukselle ei nähty tarvetta.

4.5 Valmis ohjelmistokehys

Valmiin PHP-ohjelmistokehyksen käyttämistä harkittiin vakavasti, ja kokeeksi asennettiin CakePHP. Ohjelmistokehysten hyöty on suurimmillaan isoissa projekteissa, etenkin mikäli kehys ei ole ennestään tuttu: asentamiseen, konfigurointiin ja kehyksen opetteluun menee aikaa, ja tämän ajan määrää on hankala arvioida.

Jos PHP-suunnittelija on kehittämässä sivustoa, jossa tulee olemaan yli kymmenen eri sivua, suuri tietokanta, paljon logiikkaa ja tarve käyttäjähallinnalle, niin projekti kannattaa tehdä ohjelmistokehystä käyttämällä. (Taipalus 2009, 2)

Jos resursseja olisi ollut käytettävissä enemmän, olisi ohjelmistokehystä todennäköisesti käytetty. On mahdollista, että käyttämällä kehystä olisi paremman laajennettavuuden lisäksi myös säästetty aikaa, mutta tärkein tavoite oli saada toimiva järjestelmä käytettävissä olevilla resursseilla valmiiksi mahdollisimman suurella todennäköisyydellä.

4.6 Olio-ohjelmointi vai proseduraalinen ohjelmointi?

Proseduraalisessa ohjelmoinnissa moduulit vaikuttavat toisiinsa käyttämällä jaettua tietoa. Olio-ohjelmoinnissa moduulit (joita olio-ohjelmoinnissa kutsutaan olioiksi) vuorovaikuttavat lähettämällä toisilleen viestejä.

Puhdas olio-ohjelmointi on mahdollista PHP:llä, mutta sivusto päätettiin ohjelmoida pääasiallisesti proseduraalisesti. Tärkeimmät syyt tähän olivat sivuston rakenteen yksinkertaisuus ja todennäköinen resurssien säästäminen.

Olio-ohjelmointi on suositeltavaa suurien sivustojen kanssa, sillä ylläpito, laajentaminen ja koodin uudelleenkäyttäminen myöhemmissä projekteissa on helpompaa. Puhtaasti oliopohjaisessa toteutuksessa alkuun pääseminen vaatii kuitenkin proseduraalista ohjelmointia enemmän aikaa (Poburan 2008).

Vaikka olio-ohjelmointi olikin sovelluksen pääasialliselle ohjelmoijalle tuttua, sivuston rakenteen ollessa yksinkertainen ei olio-ohjelmoinnista katsottu olevan niin paljoa hyötyä, että se oikeuttaisi aloittamiseen vaaditun lisätyön. Jos olio-ohjelmointia olisi käytetty, olisi otettu käyttöön myös valmis ohjelmistokehys.

4.7 Toteutusteknologian valinta

Kaikki kriteerit täyttäviä valmiita ajanvarausjärjestelmiä ei löytynyt. Päätettiin ohjelmoida sivuston runko itse PHP:tä, (X)HTML:ää ja MySQL:ää käyttäen, sillä näistä teknologioista oli jo valmiiksi kokemusta, eikä aiempaan kokemukseen peilaten työtä pidetty liian suurena. Ulkopuolisena kalenterina päätettiin käyttää Googlen kalenteria.

5 SUUNNITTELU JA TOTEUTUS

5.1 Ajanvarausjärjestelmän suunnittelu

Suunnitteluvaihetta aloitettaessa varmistettiin, että projektin toteuttajalla ja tilaajalla oli samat näkemykset työn lopputuloksesta, sekä siitä miten asiat toteutettiin. Kuten luvussa 2.5 kerrottiin, hyvä suunnittelu kannattaa, sillä se selkeyttää tavoitteita, kasvattaa motivaatiota, ehkäisee riskejä ja epävarmuutta. Tämän takia resursseja oli varattu suunnitteluvaiheeseen suhteellisen paljon. Näin myös siksi, että projektin loppuvaiheessa ei tarvinnut palata välttämättä enää takaisin aikaisempiin vaiheisiin, vaan suunnitelmat olivat jo selviä. Siten resurssit sekä työenergia voitiin suunnata projektin loppuvaiheisiin.

Luvussa 2.5 mainittiin suunnitelmien elämisestä yleensä projekteissa ja ne myös elivät tässäkin opinnäytetyössä. Dokumentteja sekä itse järjestelmää täytyi opinnäytetyöntekijöiden päivittää ja tehdä niihin muutoksia koko prosessin etenemisen ajan.

Suunnittelu lähti liikkeelle määrittelyvaiheessa tehdystä vaatimusmäärittelystä. Sen pohjalta oli lähdetty suunnittelemaan muun muassa ensimmäisiä näyttökarttoja ja visualisointeja. Näyttökartat muuttuivat useaan kertaan, koska suunnitelmat elivät melko paljon. Lopulta siirryttiin suoraan visualisointeihin, kun suunnitelmat ja järjestelmän rakenne alkoi olla selvillä.

Käyttöliittymän suunnittelussa otettiin huomioon Koirakeskus Doggien sen hetkiset internetsivut. Asiakkaan toiveena oli, että ajanvarausjärjestelmä noudattaisi heidän sivujensa tyyliä. Pääpaino oli käyttöliittymän selkeydessä ja yksinkertaisuudessa. Tärkeää oli, että navigointi on selkeää ja käyttäjälle on tarjottu selkeästi merkityt

poistumistiet. Joka toiminnossa tulisi siis olla mahdollisuus peruuttaa tai kumota toiminto. Tämä vahvistaa Kempvaisen (2004, 23) mukaan järjestelmän oppimista kokeilun kautta siten, että käyttäjä voi luottaa siihen, että väärin suoritettun toiminnon voi peruuttaa. Käyttäjän on myös tiedettävä missä on sivustolla menossa, ja että hän saa tarpeeksi selkeää palautetta tekemistään toiminnoistaan (Nielsen 2000, 188-189).

Selkeä ja esteettinen ulkoasu helpottaa sovelluksen käyttöä. Kun symboleja ja painikkeita suunnitellaan, mielessä täytyy pitää se, että tietokonesovelluksissa käyttäjä on tottunut luomaan tietynlaisia mielleyhtymiä tiettyihin symboleihin. Joillain symboleilla on oma merkityksensä, jota kannattaa noudattaa. Esimerkiksi nuoli oikealle yhdistetään liikkumisena eteenpäin ja nuoli vasemmalle tarkoittaa liikkumista taakse päin. Näitä merkityksiä ei pitäisi mennä muuttamaan, sillä se hämmentää käyttäjää. (Keränen, Lamberg & Penttinen 2006, 166.)

Tärkeää on, että sivusto olisi mahdollisimman yhdenmukainen. Kun käyttäjä on huomannut kuinka sovellusta käytetään jossain tietyssä kohtaa, hän soveltaa tätä tietoa myös muissa kohdissa. Käyttäjän pitäisi siis pystyä käyttämään järjestelmän uusiakin ominaisuuksia aikaisemmin opittujen toimintamallien mukaisesti ja silti päädyttäisiin vastaavissa tilanteissa samaan lopputulokseen. (Keränen ym. 2006, 166.)

Suunnitteluvaiheessa kuvattiin järjestelmän tekniset vaatimukset, joista ilmeni kuinka vaatimukset täyttävä järjestelmä toteutetaan. Haastavaa oli suunnitella jo tässä vaiheessa testitapauksia ja testaussuunnitelmaa, koska toteutettavan järjestelmän prototyyppi ei ollut vielä valmis.

5.2 Ajanvarausjärjestelmän toteutus

5.2.1 Käytettävät tekniikat

PHP

PHP on palvelimella ajettava ohjelmointikieli, jota käytetään muun muassa dynaamisten verkkosivujen HTML:n generointiin ja tietokannan kanssa kommunikointiin. Kävijä näkee ainoastaan PHP:n tulostaman sisällön, ei itse lähdekoodia.

XHTML

Selain lukee palvelimen lähettämää XHTML-merkintäkieltä, ja renderoi sen perusteella sivuston.

CSS

CSS-kielen avulla voi määritellä erilaisia tyylejä, joita XHTML-elementit käyttävät. Yhtenäiset tyylit helpottavat ohjelmointia ja ylläpitoa. Myös ladattavan HTML:n määrä vähenee, kun jokaisessa elementissä määritetään koko ulkoasun sijaan vain CSS-luokka tai -tunnus.

MySQL

MySQL on SQL-tietokannan hallintajärjestelmä. Tietokantaan tallennetaan muun muassa asiakkaiden ja koirien tiedot sekä mahdolliset palvelut.

Google Calendar

Ajanvaraukset lisätään Googlen kalenteriin, ja vapaa aika haetaan kalenterin perusteella. Järjestelmä keskustelee kalenterin kanssa Google Data API:n avulla.

Järjestelmä hakee kalenterista vain vapaa-varattu -tiedot, tarkempi selaaminen tapahtuu Googlen kalenterisovelluksessa.

Google Data API

Rajapinta, jonka avulla sovellus tiedustelee vapaat ajat kalenterilta, ja varaa uudet ajat kalenteriin. Keskustelee kalenterin kanssa APP:llä (Atom Publishing Protocol). Työssä Google Data API:a käytetään Zend Gdata -kirjaston avulla.

APP

Atom Publishing Protocol. HTTP-pohjainen protokolla, jota Google Data API käyttää keskustellessaan kalenterin kanssa. APP:tä käytetään usein blogien päivittämiseen. Google Data API vastaa APP:n käytöstä tässä sovelluksessa.

Zend Gdata

Zend Gdata -kirjasto tarjoaa PHP5-rajapinnan Google Data API:lle. Kirjaston voi ladata osoitteesta <http://framework.zend.com/download/gdata>.

5.2.2 Sivut

Ajanvaraussovelluksen sivurakenne on verrattain yksinkertainen: rekisteröinti/kirjautuminen yhdessä näkymässä, käyttäjän etusivu, koiran tietojen näyttäminen ja muokkaus, ja koiran lisäyslomake. Lisäksi on kolmiosainen lomake ajan varaamiseen.

5.2.3 Kansiorakenne

Juurihakemistossa on useita kansioita, joihin erityyppiset tiedostot sijoitetaan.

Kaikki PHP-tiedostot ovat www-kansiossa. Skriptit, jotka eivät tulosta mitään vaan ohjaavat toiselle sivulle, merkitään _-merkillä tiedostonimen alussa. Eri lomakkeet sijaitsevat alikansioissa, esimerkiksi trimmausajan varauslomake on kansiossa nimeltä "trimmaus".

JavaScript- ja CSS-tyylitiedostot ovat src-kansiossa.

Include-tiedostot (header.inc, footer.inc, validate.inc...) sekä rajapintatiedostot ovat lib-kansiossa.

Sivuston ulkoasussa käytettävät kuvat sijaitsevat img-kansiossa.

5.2.4 Tietokanta

Ajanvaraukset lähetetään Googlen kalenteriin. Tieto varauksesta tallennetaan kuitenkin myös sovelluksen omaan tietokantaan.

Sovellus toimii myös asiakasrekisterinä, josta löytyvät asiakkaat ja koirat.

Lisäksi tietokannassa ovat kaikki rodut ja mahdolliset palvelut. Palveluihin käytettävä aika voi vaihdella roduttain, mikä monimutkaisti tietokantaa (luokkakuvaukset liitteessä 8).

5.3 Sivuston runko

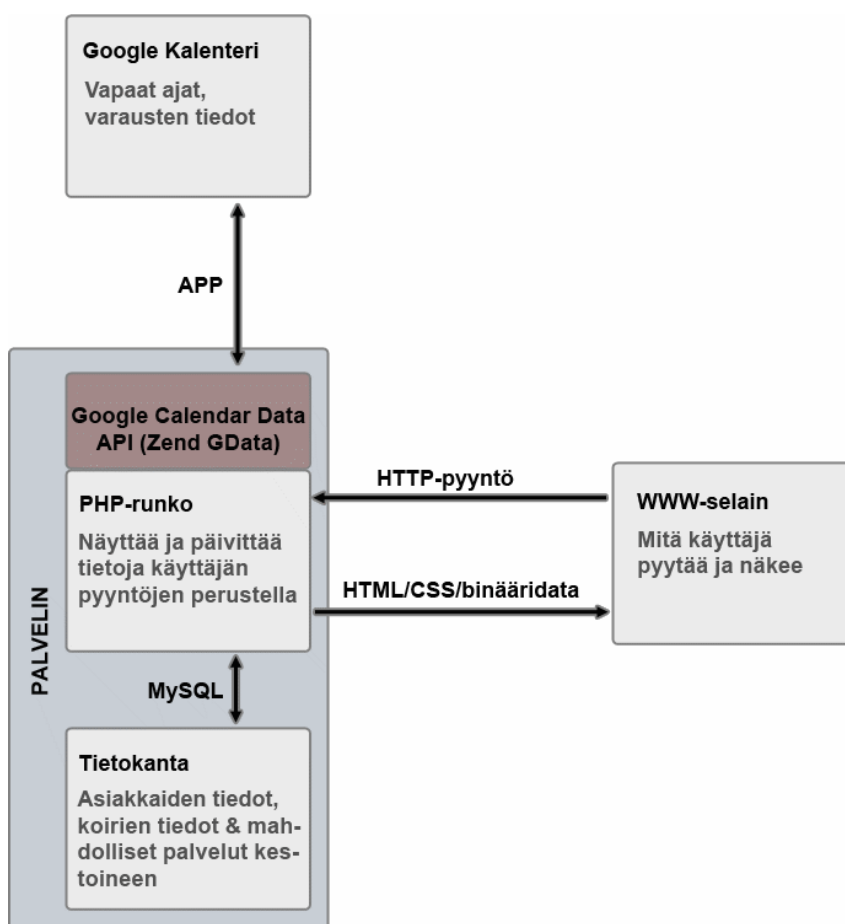
Joka sivulla tulostetaan ensin header.inc, sitten sisältösivu, ja lopuksi footer.inc. Sisältösivu voi olla mitä tahansa kirjautumissivusta kalenteriin. Lisäksi kaikki muut

sivut paitsi kirjautuminen sisällyttävät alussa validate.inc-tiedoston, joka pysäyttää sivun lataamisen ja ohjaa kirjautumissivulle, mikäli käyttäjä ei ole oikeaoppineesti kirjautuneena sisään.

Tietokannan kanssa keskustellaan tietokantaluokan avulla. Tietokantaluokka vastaa mm. tietokantaan yhdistämisestä ja MySQL-lauseiden oikeellisuuden tarkistamisesta.

Monisivuiset lomakkeet käyttävät sessiomuuttujia. Ajaxia hyödyntävä lomake oli pitkään vaihtoehtona, mutta sovelluksen ei haluttu vaativan javascriptiä

Periaate on yksinkertainen, mutta toimiva tähän tarkoitukseen.



KUVIO 16. Sivuston toimintaperiaate

5.4 Kalenterin rajapinnan käyttö

5.4.1 Käyttöönotto

Googlen kalenterin käyttöön vaaditaan Google-tili. Kalenterin asetuksista löytyy yksityinen avain, private-url-key, jota voidaan käyttää kalenterin lukemiseen rajapinnan avulla (Google Inc. 2011b).

PHP-rajapinnan tarjoaa Zend Gdata -luokkakirjasto. Tämä asennettiin lib/Zend -kansioon, josta tarvittavat luokat sisällytetään kalenterin kanssa keskusteleviin sivuihin.

5.4.2 Kalenterin lukeminen taikakeksiautentikoinnin avulla

Yksinkertaisimmillaan yksityisestä kalenterista voi hakea tietoja taikakeksiautentikoinnilla. Tämä toimii edellämainitun yksityisen avaimen avulla, eikä vaadi salasanaa tai kirjautumista. Taikakeksiautentikoinnin avulla haetaan kalenterista vapaa-varattu-tiedot ajanvarauskalenteriin. Tällä autentikointitavalla ei ole mahdollista muokata kalenteria, ainoastaan lukea sitä (Google Inc. 2011b).

```
// Luodaan uusi kalenteriolio ja sille tapahtumakysely:
$gdataCal = new Zend_Gdata_Calendar();
$query = $gdataCal->newEventQuery();

// Määritetään käyttäjäksi kalenterin omistajan käyttäjätunnus:
$query->setUser('kayttajatunnus@gmail.com');

// Käytetään yksityistä avainta, joka löytyy kalenterin asetuksista:
$query->setVisibility('private-abcdefg');
```

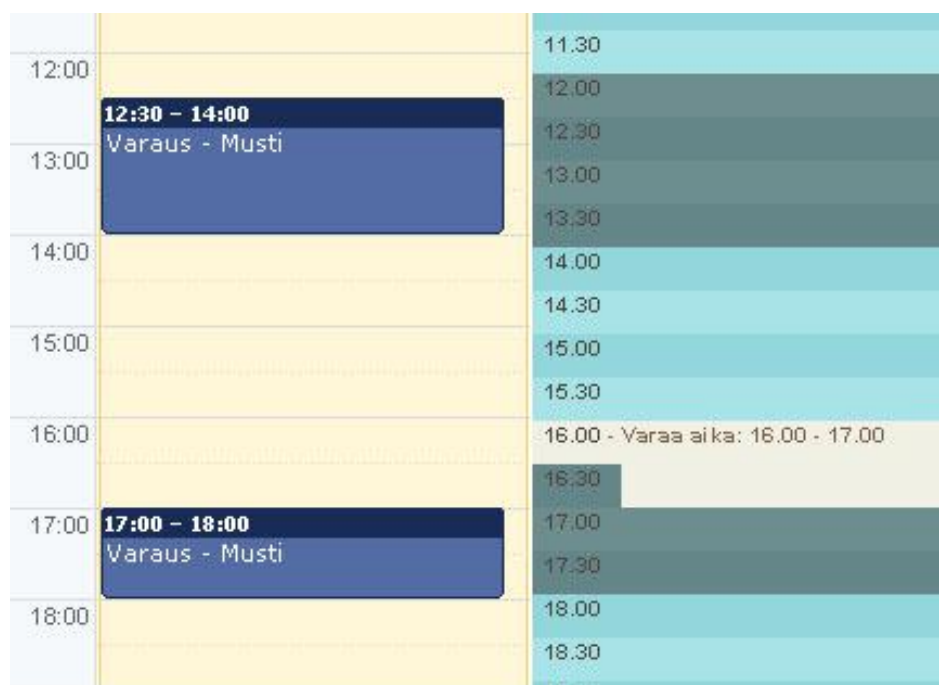
```
// Määritetään haettavaksi vain tiedot siitä, ovatko palautettavat
ajat vapaita vai varattuja:
$query->setProjection('free-busy');
```

```
//Määritetään eventFeed-muuttuja tapahtumakyselyn tulostaulukoksi:
$eventFeed = $gdataCal->getCalendarEventFeed($query);
```

Palautettava taulukko sisältää kaikki päivän varatut ajat olioina, joilla on mm.

startTime- ja endTime-muuttujat. Tätä taulukkoa iteroimalla luodaan XHTML/CSS-kalenterinäköymä, jossa näkyy valittavissa olevat vapaat ajat.

Kuviossa 9 vasemmalla näkyy sama ajankohta Googlen kalenterissa, oikealla varausjärjestelmän luoma näkymä kalenterilta saamistaan tiedoista. Koska klo 12.30 on varaus, voi tunnin mittainen varaus alkaa klo 11:30, mutta ei enää klo 12.



KUVIO 17. Sama ajankohta Googlen ja varausjärjestelmän kalentereissa

5.4.3 Varauksen lisääminen ClientLogin-autentikoinnin avulla

Mikäli asiakas päättää varata valitsemansa ajan, ajanvaraus lähetetään kalenterille ClientLogin-autentikoinnin avulla. ClientLogin-autentikoinnissa käytettävän kalenterin käyttäjätunnus ja salasana määritellään lähdekoodissa (Google Inc. 2011b).

```
$user = 'kayttajatunnus@gmail.com';
$password = 'salasana';
$service = Zend_Gdata_Calendar::AUTH_SERVICE_NAME;

$title = 'Varaus - Musti';
$desc = 'Trimmaus, pesu';
$date = '23.03.2011';
$startTime = '10:00';
$endTime = '11:00';
$where = "Jyväskylä";

// Luodaan tunnuksilla uusi kalenteriolio:
$client = Zend_Gdata_ClientLogin::getHttpClient( $user, $password,
$service ); $gdataCal = new Zend_Gdata_Calendar($client);

// Luodaan uusi tapahtuma, jolle annetaan aikaisemmin määritetyt ta-
pahtuman tiedot:
$newEvent = $gdataCal->newEventEntry();
$newEvent->title = $gdataCal->newTitle($title);
$newEvent->where = array($gdataCal->newWhere($where));
$newEvent->content = $gdataCal->newContent("$desc");

// Muunnetaan aika googlen ymmärtämään RFC 3339 -
aikaleimaformaattiin:
$date = date( "Y-m-d", strtotime($date) );
$startTime = date( "H:i:s", strtotime($startTime) );
$endTime = date( "H:i:s", strtotime($endTime) );
$when = $gdataCal->newWhen();
$when->startTime = "{$date}T{$startTime}.000+02:00";
$when->endTime = "{$date}T{$endTime}.000+02:00";
$newEvent->when = array($when);

// Lähetetään uusi tapahtuma kalenteriin:
$createdEvent = $gdataCal->insertEvent($newEvent);
```



KUVIO 18. Lisätty tapahtuma Googlen kalenterissa

Lopputuloksena Googlen kalenteriin tulee tapahtuma määritetyillä tiedoilla.

5.5 Järjestelmän testaus

Luvun 2.7 mukaisesti, tässä opinnäytetyössä testauksen työvaiheisiin kuuluivat muun muassa testauksen suunnittelu, joka sisälsi testaussuunnitelman ja testitapaukset, sekä testiympäristön luonti, testin suorittaminen ja tulosten tarkastelu.

Testaussuunnitelma toimi viitekehyksenä testauksen suorittamiselle (liite 6). Siinä kerrottiin testauksen periaatteet, tehtävät ja testausympäristö ynnä muuta sellaista. Testitapaukset taas kuvasivat ohjelman eri osiin liittyviä testejä (liite 7).

Testausvaihe oli osittain päällekkäinen toteutusvaiheen kanssa, koska järjestelmää testattiin myös samalla, kun sitä toteutettiin. Testaus suoritettiin projektin loppuvaiheessa V-mallin mukaisesti (kts. kuvio 9). Apuna käytettiin myös

testitapausjoukkoa, kuten luvussa 2.7 suositellaan. Sen avulla testauksen suorittaminen oli tehokkaampaa ja järjestelmällisempää, kuin ohjelman umpimähkäinen kokeilu. Tehokkuutta tarvittiinkin, koska varsinainen testaus oli jäädä hieman vähäiseksi, sillä toteutus saatiin valmiiksi melko viime hetkellä. Testaukseen, jota suoritettiin samalla kun järjestelmää toteutettiin, oli panostettu enemmän aikaa, jotta järjestelmä toimisi mahdollisimman moitteettomasti.

Järjestelmälle suoritettiin myös käytettävyydestestausta. Tällä pyrittiin varmistamaan se, pystyykö asiakas suoriutumaan tehtävistä, joiden suorittamista varten järjestelmä tehtiin. Kokematon käyttäjä ei aina mahdollisesti reagoi ja toimi samanlailla, kuin järjestelmän toteuttaja tai niitä käyttämään tottunut henkilö. Oli siis tärkeää saada kokemattomia testaajia mukaan projektiin, jotta saataisiin arvokasta tietoa ja palautetta käyttäjien toiminnoista.

Tarvittavaa testauksen määrää oli vaikea arvioida etukäteen. Testausta ja järjestelmän korjausta jatkettiin niin pitkään, kunnes järjestelmä toimi moitteettomasti ja se pystyttiin ottamaan käyttöön.

6 KÄYTTÄJÄKOULUTUS

Käyttäjäkoulutus järjestettiin kyseisessä projektissa henkilökohtaisena koulutuksena ja ohjauksena. Koulutusta tarjottiin siinä vaiheessa, kun projekti oli lopussa ja ajanvarausjärjestelmä valmiina. Tavoitteena oli tarjota asiakkaille käytännönläheistä sekä selkokielistä opetusta, joka mahdollistaisi järjestelmän omatoimisen käytön sekä riippumattomuuden projektin toteuttajista. Koulutus suunniteltiin soveltumaan henkilöille, joilla ei ole aiempaa kokemusta tietotekniikasta tai tällaisten järjestelmien ylläpidosta.

Koulutuskertoja suunniteltiin järjestettäväksi kaksi kappaletta, sillä kaikille osapuolille sopivien aikojen löytämisen arveltiin olevan haastavaa, ja koulutuksen jakaminen kahteen kokonaisuuteen oli järkevintä oppimisen ja ymmärtämisen kannalta. Ensimmäisellä koulutuskerralla ohjelmaksi suunniteltiin järjestelmän toimintojen ja rakenteen esittely. Tällöin koulutettavat saivat myös esittää kysymyksiä projektin tekijöille sekä tutustua omatoimisesti järjestelmän käyttöön. Toisella kerralla koulutus keskittyisi syventämään jo aiemmin opittuja asioita järjestelmän käytöstä. Molemmilla kerroilla asiakkailta olisi myös mahdollisuus kysyä järjestelmän erilaisista ominaisuuksista.

Yhteistyötä on myös mahdollista jatkaa tulevaisuudessa. Mikäli asiakkaat kokevat tarvitsevansa lisäopastusta tai tukea järjestelmän ylläpitämisessä ja mahdollisissa ongelmatilanteissa, heidän on mahdollista pitää yhteyttä projektin tekijöihin. Jos on tarvetta, suunnitelmissa on myös luoda heille ohjekirja, joka kattaa päivittäisen käytön ohjeistuksen sekä mahdollisten ongelmatilanteiden itsenäisen selvittämisen.

7 PÄÄTÖSVAIHE

Pitkään kestänyt projekti asiakkaiden kanssa päättyi keväällä 2011. Asiakkaiden kanssa käytiin läpi järjestelmä ja lopputulos hyväksyttiin sen jälkeen yhteisessä loppukokouksessa. Järjestelmä luovutettiin asiakkaille ja vastuu lopputuotteen käytöstä siirtyi heille. Projektin tekijät siirsivät järjestelmän Oulanka Networksin tarjoamalle palvelimelle, jossa Koirakeskus Doggien nykyiset internetsivut sijaitsevat.

Projektin kirjallinen osuus hyväksyttiin asiakkaiden lisäksi myös ohjaavalla opettajalla. Tämän jälkeen projekti esiteltiin opinnäytetyöseminaarissa. Hyväksytyn tuloksen jälkeen opinnäytetyö vietiin painoon ja projektiorganisaatio purettiin. Yhteistyö asiakkaan kanssa jatkui vielä projektin virallisen päättymisen jälkeen

kuukauden ajan, jolloin varmistettiin ajanvarausjärjestelmän toimivuus sekä asiakkaiden tietämys järjestelmän käytöstä.

8 TULOKSET JA JATKOKEHITYS

Järjestelmän tärkein tavoite - eli mahdollisuus aikojen varaamiseen internetin avulla puhelinvarausten sijaan - toteutui kuten suunniteltiin. Muita tuloksia oli muun muassa Koirakeskus Doggien domainille luotu Google Apps -tili ja asiakastietokanta. Tulevaisuudessa jatkuvan yhteistyön aikana järjestelmää on jatkossa mahdollista kehittää eteenpäin toteuttamalla asiakkaiden muita alkuperäisiä toiveita, joita ei ollut mahdollista toteuttaa projektin kuluessa.

9 YHTEENVETO

Opinnäytetyön aihevalinta oli mielestämme onnistunut. Projekti oli tarpeellinen ja odotettu eikä jäänyt niin sanotusti keräämän pölyä, vaan tuli todella käyttöön. Haastavaa opinnäytetyössä oli sen rajaaminen. Asiakkaan vaatimukset olivat suhteellisen suuria opinnäytetyön laajuuteen nähden, joten toteutettavien ominaisuuksien priorisointi oli tärkeää. Ajankäyttöä ja aikataulutusta olisi voinut käyttää paremmin ja projektin toteutus venyi ajallisesti suunniteltua pitemmäksi. Emme osanneet alussa hahmottaa, kuinka paljon ajallisia resursseja projekti olisi kokonaisuudessaan vienyt, joten karsimista oli tehtävä. Ensisijainen tavoite oli saada toteutettua toimiva ajanvarausjärjestelmä, jolla saisi ainakin trimmaus- ja kurssiaikoja varattua. Vaativaa oli myös se, ettei samanlaista järjestelmän pohjaa

ollut valmiina, vaan jouduimme soveltamaan ja tekemään todella paljon itse. Hieman samanlaisia parturi-kampaamoiden ajanvarausjärjestelmiä oli jo olemassa, mutta nekään eivät sellaisenaan soveltuneet projektiimme, koska Koirakeskus Doggien asiakkaina ovat pääasiassa koirat, eivätkä ihmiset. Jokaisella eri koirarodulla oli omat vaatimuksensa ja se toi haasteita järjestelmän toteuttamiselle.

Projektin suunnittelu oli mielestämme vaativaa, mutta onnistui kuitenkin hyvin. Monia asioita täytyi ottaa huomioon ja oli hyvä suunnitella useat asiat etukäteen valmiiksi, että toteutusvaihe oli sitten helpompi suorittaa. Useamman kerran jouduimme kuitenkin palaamaan muutamissa vaiheissa taaksepäin ja muuttamaan suunnitelmia. Asiakas oli mukana koko ajan aktiivisesti eikä sen puolesta ollut ongelmia.

Yhteistyö opinnäytetyön tekijöiden välillä toimi hyvin. Kummankin vahvuudet ja erikoistumisen kohteet olivat voimavarana ja täydensivät toisiaan. Pakkalalla oli muun muassa vahva tausta ohjelmoinnissa sekä järjestelmien tuntemuksessa. Ihanainen oli taas hyvä muun muassa graafisessa suunnittelussa ja projektin hallinnassa. Molemmilta luonnistui hyvin myös järjestelmän suunnittelu.

Projektin tulosten lisäksi opimme projektin aikana muun muassa ajankäyttöä, kokonaisuuksien hahmottamista, suunnittelun tärkeyttä sekä tietysti myös asiakkaiden kanssa erilaista vuorovaikutusta, asiakkaiden toiveiden, ideoiden ja tavoitteiden kehittämistä käytännössä toimiviksi asioiksi. Järjestelmää ei siis kannata viskata nopeasti kasaan vaan kannattaa pysähtyä miettimään kokonaisuutta ja toimivuutta, että järjestelmää voi sitten päivittää, ylläpitää ja kehittää helposti eteenpäin. Kun alusta asti järjestelmän rakentaa huomioiden niin kokonaisuuden kuin järjestelmän muutkin eri osat, pystyy helpommin huomaamaan kaikki ongelmat ja korjaamaan ne. Täten myös testaamisen tärkeys korostuu.

Kuten ohjelmointiprojekteissa aina, myös ohjelmointitaidot vahvistuivat työn aikana; aina on jotain opittavaa, ja aina kannattaa etsiä parempaa tapaa päästä tavoitteeseen.

LÄHTEET

Google Inc. 2011a. Google apps -vaihtoehtojen vertailutaulukko. Viitattu 15.3.2011.
<http://www.google.com/apps/intl/en/group/index.html>

Google Inc. 2011b. Data API Developer's guide: PHP. Opas Google kalenterin PHP-rajapinnalle. Viitattu 17.3.2011.
http://code.google.com/apis/calendar/data/1.0/developers_guide_php.html

Haikala, I. & Märijärvi, J. 2006. Ohjelmistotuotanto. 11. p. Jyväskylä: Gummerrus Kirjapaino Oy.

Kemppainen, R. 2004. Käyttöliittymä ja käytettävyys -Valueframe Psa-. Opinnäytetyö. Jyväskylän ammattikorkeakoulu, Informaatioteknologian instituutti.

Keränen, V., Lamberg, N. & Penttinen, J. 2006. Web-julkaiseminen ja multimedia. 1p. Porvoo: WS Bookwell Oy.

Kettunen, S. 2003. Onnistu projektissa. Juva: WS Bookwell Oy.

Koirakeskus Doggie. 2011. WWW-sivut. Viitattu 4.4.2011.
<http://www.doggie.fi/yritys>

Pelin, R. 2002. Projektihallinnan käsikirja. 3p. Jyväskylä: Gummerrus Kirjapaino Oy.

Poburan, L. 2008. Procedural vs Object Oriented PHP Programming. Artikkelit Debug Magazine -sivustolla. Viitattu 17.3.2011.
<http://www.debugmagazine.com/freelancing-tips-tricks/procedural-vs-object-oriented-php-programming>

Ruuska, K. 2007. Pidä projekti hallinnassa. 6p. Jyväskylä: Gummerrus Kirjapaino Oy.

Taipalus, J. 2009. Symfony framework. Opinnäytetyö. Seinäjoen ammattikorkeakoulu, Tekniikan yksikkö, tietotekniikan koulutusohjelma. Viitattu 17.3.2011. <http://urn.fi/URN:NBN:fi:amk-2009101649>

LIITTEET

Liite 1. Vaatimusmäärittely

Ajanvarausjärjestelmä Koirakeskus Doggielle Vaatimusmäärittely

Seela Ihanainen

Jaakko Pakkala

Huhtikuu 2011

Mediatekniikka

Tekniikan ja liikenteen ala



VERSIONHALLINTA

JP = Jaakko Pakkala

SI = Seela Ihanainen

Versio	Päivämäärä	Muutos(/muutokset)	Muokkaaja
010	15.3.2010	Sisällysluettelon teko ja vaatimusmäärittelyn aloittaminen	SI
011	16.3.2010	Vaatimusmäärittelyn tekoa	SI, JP
012	16.3.2010	Muutoksia kappaleisiin 3.3, 4.4, 5	SI
013	25.3.2010	Muutos kappaleeseen 4.3	SI
014	4.1.2011	Muutoksia kappaleisiin 4.2-4.9, 5, 6.1.2	SI
015	3.4.2011	Oikolukua ja korjailua kaikkiin kappaleisiin	SI
016	27.4.2011	Pieniä lisäyksiä kappaleeseen 4.3	SI

SISÄLTÖ

1	JOHDANTO	60
1.1	Opinnäytetyön tavoitteet	60
1.2	Dokumentin rakenne	60
2	YLEISKUVAUS.....	61
2.1	Kuvaus toteutettavasta järjestelmästä.....	61
2.2	Toteutustyökalut.....	62
3	KÄYTTÄJÄT.....	62
3.1	Käyttäjä	62
3.2	Järjestelmän ylläpitäjä	62
3.3	Käyttöoikeudet	63
4	TOIMINNOT	63
4.1	Asiakastilin luominen	63
4.2	Sisäänkirjautuminen: Admin ja yrityksen asiakas	63
4.3	Aikojen varaaminen.....	64
4.3.1	Tilanvaraus.....	66
4.4	Käyttäjätiedot	66
4.5	Asiakashallinta.....	66
4.6	Paikkatiedot.....	66
4.7	Kurssienhallinta.....	66
4.8	Ohjeet	67
4.9	Asiakasrekisteri ja hakutoiminto.....	67
4.10	Tulostettava versio	67
4.11	Uloskirjaus	67
5	KÄYTTÖLIITTYMÄ	68
6	ARKKITEHTUURI	68
6.1	Palvelin.....	68
6.1.1	Testipalvelin	68
6.1.2	Tuotantopalvelin.....	68
7	TURVALLISUUS	69
8	YHTEENVETO	69

1 JOHDANTO

1.1 Opinnäytetyön tavoitteet

Opinnäytetyön pääprioriteettina on toteuttaa jyvaskyläläiselle Koirakeskus Doggielle ”portaalityyppinen” ajanvarausjärjestelmä ja asiakasrekisteri verkkoon, sekä näkyvyyskysely. Jos resursseja ja osaamista löytyy, toteutetaan muitakin mahdollisia toimeksiantajan toiveita.

1.2 Dokumentin rakenne

Kappale 1 sisältää johdannon toteutettavasta järjestelmästä. Kappaleessa kuvataan projektia ja sen tavoitteita yleisesti, sekä tämän dokumentin rakennetta.

Kappaleessa 2 käsitellään järjestelmää yleisesti, sen kohderyhmää ja toteuttamistyökaluja.

Kappale 3 sisältää kuvauksen järjestelmään tehtävistä käyttäjistä ja heidän käyttöoikeuksistaan järjestelmään.

Kappaleessa 4 käsitellään järjestelmään mahdollisesti tulevia toimintoja. Kappaleen sisällä määritellään tärkeimmät toiminnot ja kuvataan niiden toteutusta.

Kappaleessa 5 käsitellään alustavasti järjestelmän käyttöliittynyä ja kuvaillaan hieman sitä, millainen siitä mahdollisesti tulee.

Kappale 6 käsittelee järjestelmän arkkitehtuuria, kuten palvelinympäristöä.

2 YLEISKUVAUS

2.1 Kuvaus toteutettavasta järjestelmästä

Koirakeskus Doggie tarvitsee ajanvarausjärjestelmän sekä asiakasrekisterin helpottamaan yrityksen toimintaa. Nykyään heillä menee aikaa todella paljon puhelimesta roikkumiseen, joten he toivovat järjestelmän tuovan apua sekä siihen, että asiakastietojen nopeampaan järjestelyyn. Tarkoituksena on luoda asiakastietokanta, jossa on kaikki yrityksen asiakkaat koirineen. Asiakkaiden olisi sen avulla mahdollista varata trimmaus- ja kurssiaikoja kummaltakin työntekijältä internetin välityksellä. Mahdollisesti varauksia tekevä asiakas saisi myös maksutiedot heti varauksen tehtyään.

Järjestelmästä on tarkoitus tehdä mahdollisimman selkeä ja yksinkertainen, jotta varausten lisääminen ja selailu olisi helppoa kaikille käyttäjille. Selkeyttä tavoitellaan niin käytön kuin ulkoasun kannalta. Tavoitteena on, että käyttöliittymä olisi yksinkertainen, helppokäyttöinen ja noudattaisi Koirakeskus Doggien nettisivujen nykyistä tyyliä. Asiakkaan toiveena oli, että järjestelmän ajanvarauksia voisi selailla matkapuhelimella, ja että järjestelmästä ei koituisi paljon kuluja heille toteutusvaiheessa tai sen jälkeen.

Tärkeysjärjestyksessä ensimmäisenä on saada valmiiksi toimiva järjestelmä trimmausaikojen varaamiseen. Toiseksi tärkein ominaisuus, joka toteutetaan resurssien salliessa, on saada mahdolliseksi järjestelmään myös kurseille ilmoittautuminen internetin välityksellä. Tämän lisäksi toteutetuksi pitäisi saada Admin-sivut, jossa asiakashallinnan lisäksi mahdollista olisi lisäällä, muokata ja poistaa kursseja järjestelmästä. Asiakkaan toiveena oli myös saada jonotuspaikka-toiminto kursseja varten. Kolmanneksi oli priorisoitu tilan varaus. Koirakeskus Doggiella on lämmin halli, jota he vuokraavat eri koulutustarkoituksiin, ja tälle hallille pitäisi saada tilanvarausjärjestelmä.

Järjestelmä luovutetaan valmistuttuaan toimeksiantajalle eli Koirakeskus Doggielle.

2.2 Toteutustyökalut

Järjestelmä toteutetaan kummankin opinnäytetyöntekijän omavalintaisella www-sivujen teko-ohjelmalla, kuvankäsittelyohjelmilla, ja MySQL-tietokannoilla. Sivut ohjelmoidaan PHP- ja HTML/XHTML-kielillä, hyödyntäen mahdollisesti muita teknologioita (esim. Javascript).

3 KÄYTTÄJÄT

3.1 Käyttäjä

Järjestelmää käyttävät pääkäyttäjän oikeuksin Koirakeskus Doggie -yrityksen osakkaat, ja rajallisin ajanvarausmahdollisuuksin heidän asiakkaansa. Pääkäyttäjiä on alussa kaksi, muita käyttäjiä ovat Doggien asiakkaat.

3.2 Järjestelmän ylläpitäjä

Toimeksiantaja koulutetaan ylläpitämään järjestelmää, jos ongelmia syntyy, voi yritys alkuvaiheessa ottaa yhteyttä järjestelmän tekijöihin.

3.3 Käyttöoikeudet

Asiakkailla tulee olemaan oikeudet varata itselleen aikoja ja perua niitä (esim. kurssiaikoja viikkoa ennen tapahtumaa) ajanvarausjärjestelmän kautta. Toimeksiantajalle annetaan admin-tunnukset, joiden avulla he voivat poistaa ja tarvittaessa lisätä käyttäjiä, sekä muokata asiakkaiden ja henkilökunnan tietoja. Admin-tunnuksilla voi myös muokata kalenteita sekä lisätä ja poistaa tapahtumia.

4 TOIMINNOT

4.1 Asiakastilin luominen

Asiakas voi itse luoda tilin siirtymällä yrityksen nettisivuilta ajanvarausjärjestelmän sivuille ja antamalla perustiedot itsestään. Mikäli asiakas ei syystä tai toisesta halua käyttää internetiä, tulisi hänen silti olla asiakastietokannassa. Tällöin admin voi lisätä asiakkaan tietokantaan koirineen.

4.2 Sisäänkirjautuminen: Admin ja yrityksen asiakas

Sisäänkirjautuminen tapahtuu internetissä yrityksen nettisivuilla olevan linkin kautta. Jos asiakkaalla on valmiit tunnukset järjestelmään, pystyy hän kirjautumaan suoraan sisään ajanvarausjärjestelmään täyttämällä käyttäjätunnus- ja salasanakentät, sekä klikkaamalla ”Kirjaudu sisään” -painiketta. Jos asiakkaalla ei ole valmiita tunnuksia järjestelmään, hän voi rekisteröityä käyttäjäksi järjestelmän etusivulla. Sieltä löytyy rekisteröintilomake, jossa kysytään asiakkaan nimi, yhteystiedot ja salasana

Kun työntekijä kirjautuu sisään, aukeaa kurssienhallintasivu, jossa työntekijä pääsee tarkastelemaan tietoja kursseille ilmottautuneista ja muusta olennaisesta. Sivulla on myös linkki asiakashallintaan, jossa listataan kaikki asiakkaat..

Kun asiakas kirjautuu sisään, avautuu omat tiedot -sivu, jossa näkyy asiakkaan tiedot. Näitä tietoja asiakas voi itse muokata.

4.3 Aikojen varaaminen

Järjestelmään on tarkoituksena luoda kummallekin yrityksen työntekijälle oma varauskirja (yhteensä kaksi kappaletta), johon tallennetaan asiakkaan tekemät ajanvaraukset tietoineen. Varausten lisääminen ja selailu pyritään saamaan mahdollisimman yksinkertaiseksi käyttää.

Kun asiakas haluaa varata aikoja, on hänen oltava rekisteröityneenä ja sisäänkirjautuneena järjestelmään. Ensin asiakkaan on lisättävä koira(t) tietokantaan Omat tiedot -sivulta löytyvän Lisää koira -linkin kautta. Koiria voi lisätä useammankin. Sen jälkeen asiakas valitsee Omat tiedot -etusivulta joko Uusi trimmausaika- tai Ilmottaudu kursseille -linkin/napin. Jos asiakas valitsee trimmausajat, aukeaa ikkuna johon on listattu asiakkaan koira(t) erikseen. Sieltä asiakas voi valita yhdelle tai useammalle koiralleen ”rasti ruutuun” -toiminnalla haluamansa trimmauspalvelut. Asiakas ei itse saa valita palvelun kestoa, vaan aikojen kestoajat lasketaan automaattisesti kalenteriin palveluun kuluvaan ajan mukaan (aikamuuttujat valmiiksi määriteltä toimeksiantajan kanssa). Eli järjestelmä laskee kuinka kauan aikaa menee mihinkin palveluun.

Tämän jälkeen asiakas klikkaa Seuraava-nappia ja siirtyy eteenpäin. Seuraavaksi asiakas pystyy valitsemaan kahdesta kalenterista kummalle Doggien työntekijöistä hän haluaa koiransa viedä trimmattavaksi. Sitten asiakas hiirellä klikkaamalla valitsee haluamansa ajankohdan kalenterista, ja siirtyy varauksen vahvistamiseen. Varauksenvahvistamissivulla asiakas saa yhteenvedon valitsemistaan palveluista, muistutuksen valitsemastaan ajankohdasta, sekä muistutuksen maksun suorittamisesta (käteisen mukaan ottamisesta) ja ohjeet varauksen perumisesta

(varauksen voi perua milloin vain lääkärintodistuksella, ilman todistusta 1 vrk kautta ennen trimmausaikaa). Klikkaamalla Vahvista varaus -linkkiä asiakas voi nyt vahvistaa varauksensa. Kun varaus on vahvistettu, asiakas saa siitä tiedon ja siirtyy takaisin Omat tiedot -sivulle, jossa on listattuna hänen varaamansa aika/ajat.

Jos asiakas on valinnut kurseille ilmottautumisen, aukeaa uusi ikkuna, jossa valitaan ensin palvelu, esim. agilityn alkeet, sen jälkeen ajankohta ja koira jonka kanssa kurssille mennään. Järjestelmä ilmoittaa, jos kurssipaikka on täynnä (6 henk. max. mahtuu kurssille), ja kysyy asiakkaalta haluaako tämä jonotuspaikalle. Jos asiakas siirtyy jonotuspaikalle, Koirakeskus Doggien työntekijät ottavat ensimmäisenä jonotuspaikalla olevaan yhteyttä, kun kurssipaikkoja vapautuu. Jos kurssilla oli vapaata, asiakas pystyy siirtymään järjestelmässä eteenpäin Seuraava-napilla. Tämän jälkeen kysytään asiakkaalta haluaako hän vahvistaa varauksen, ja myönteisen vastauksen jälkeen asiakkaalle annetaan tili- ja viitenumero, joiden avulla kurssimaksu maksetaan. Järjestelmä ilmoittaa myös, että kurssimaksu on maksettava ennen kurssille tuloa ja kurssille ilmottautumisen voi perua viimeistään yhtä viikkoa ennen kurssin alkamista, mutta omalla tai koiran lääkärintodistuksella milloin vain.

Lyhimmän ajanvarauksen väliksi sovittiin 30 minuuttia. Ajanvarauksia pystyy tekemään aikavälille 9-21. Kalenteri näkymä näyttää aina 1 päivän kerrallaan. Päiviä pystyy vaihtamaan Seuraava päivä- ja Edellinen päivä-napeista. Kalenterin yläpuolella on myös pikkukalenteri jota klikkaamalla saa vaihdettua kuukausia ja valitsemalla päivän sitäkin kautta. Oletusaikaväli on adminin muokattavissa Googlen kalenterin kautta. Kalenterissa ajanvarauksesta näkyy seuraavat tiedot adminille: asiakkaan sukunimi (+ etunimi) ja koiran nimi.

Työntekijän on mahdollista myös varata ns. omia aikoja Googlen kalenterin kautta ja merkitä kalenteriin sillä tavoin oma poissaolonsa. Tällä ehkäistään se, ettei hänelle varata aikoja hänen poissaollessaan.

4.3.1 Tilanvaraus

Tilanvaraus toteutetaan, jos resurssit riittävät. Niitä Admin voi lisätä itse lisää tietokantaan.

4.4 Käyttäjätiedot

Järjestelmän etusivu on Omat tiedot -sivu. Sivulta löytyy mm. seuraavia kenttiä: etunimi, sukunimi, katuosoite, puhelinnumero, sähköposti, käyttäjätunnus ja salasana.

Lisää koira -linkin kautta lisätyt koirat tulevat käyttöliittymän oikealle puolelle listattuna, ja niitä klikkaamalla pääsee muokkaamaan seuraavia tietoja koirasta: kutsumanimi, rotu, syntymävuosi, sukupuoli ja lisätietoja.

4.5 Asiakashallinta

Asiakashallinta-valikossa työntekijä voi lisätä, muokata ja poistaa asiakkaita. Lisäksi käytössä on hakutoiminto.

4.6 Paikkatiedot

Paikkatiedot-kohdassa työntekijä voi lisätä, muokata ja poistaa paikkoja tietokannasta.
(Toteutetaan resurssien salliessa)

4.7 Kurssienhallinta

Kurssienhallinnassa työntekijä voi lisätä ja poistaa kursseja, ja seurata kursseille ilmoittautuneita ihmisiä. Jonotuspaikat ovat myös siellä näkyvillä. Eli työntekijä näkee järjestetyssä listassa

ilmottautuneiden järjestyksen, ja siten pystyy siirtämään peruutuspaikoille jonotuspaikalla olleita.

4.8 Ohjeet

Järjestelmä näyttää pelkistetyt ohjeet sen päätoiminnoista asiakkaan näkymässä.

4.9 Asiakasrekisteri ja hakutoiminto

Koko asiakasrekisteriä voi selata asiakashallinta-linkin takaa työntekijän alueelta. Järjestelmässä voidaan suorittaa asiakashakuja ja mahdollisesti jopa aikahakuja. Asiakashaussa tietoja voidaan hakea asiakkaan sukunimen, etunimen, koiran nimen tai puhelinnumeron perusteella. Haku käynnistetään Hae-painikkeella. Haun tulokset –kenttään tulee näkyviin ilmoitus haun tuloksista. Nämä toiminnot ovat vain työntekijöille

4.10 Tulostettava versio

Järjestelmä voisi mahdollisesti antaa kalenterista myös tulostettavan version. Myös laskun voisi tulostaa valmiiksi täytettynä asiakkaan tiedoilla.

4.11 Uloskirjaus

Päänäkymän uloskirjautumispainikkeella pääsee kirjautumaan ulos kummastakin järjestelmästä. Tämän jälkeen avautuu sivu josta on mahdollisuus kirjautua uudestaan sisään.

5 KÄYTTÖLIITTYMÄ

Käyttöliittymä toimii Internet-selaimen kautta. Tuettavia selaimia ovat uusimmat versiot mm. Mozilla Firefoxista, Internet Explorerista ja Operasta. Työn toteutuksessa otetaan huomioon, että järjestelmä toimii kokonaisuudessaan jokaisella edellä mainitulla selaimella.

Sisäänkirjautumisen jälkeen käyttäjä/asiakas voi valita haluamansa kahdesta vaihtoehdosta. Osa-alueita voisi olla esimerkiksi etusivu/omat tiedot, ajanvaraus, (kursseille ilmottautuminen) ohjeet, ja kirjaudu ulos. Lisäksi adminille tulee mm. näkyviin asiakashallinta, kurssien hallinta ja paikkatiedot. Käyttöliittymä tehdään noudattaen yrityksen nykyistä tyyliä

6 ARKKITEHTUURI

6.1 Palvelin

6.1.1 Testipalvelin

Testipalvelimenä käytettävä oppilaitoksen palvelin sijaitsee Jyväskylän ammattikorkeakoulun Teknologiayksikön laboratorioverkon serverihuoneessa.

6.1.2 Tuotantopalvelin

Tuotantopalvelimen tulisi tukea samoja käyttöjärjestelmä-, PHP-, Apache- ja MySQL-versiota testipalvelimen kanssa, jotta järjestelmän voi siirtää ongelmitta testipalvelimelta tuotantopalvelimelle. Nykyiset www-sivut sijaitsevat Oulanka Networksin tarjoamalla

palvelimella (<http://www.doggie.fi/>), jonne lopputuotekin todennäköisesti siirretään. Palvelimen tulisi tukea PHP:tä ja MySQL:ää.

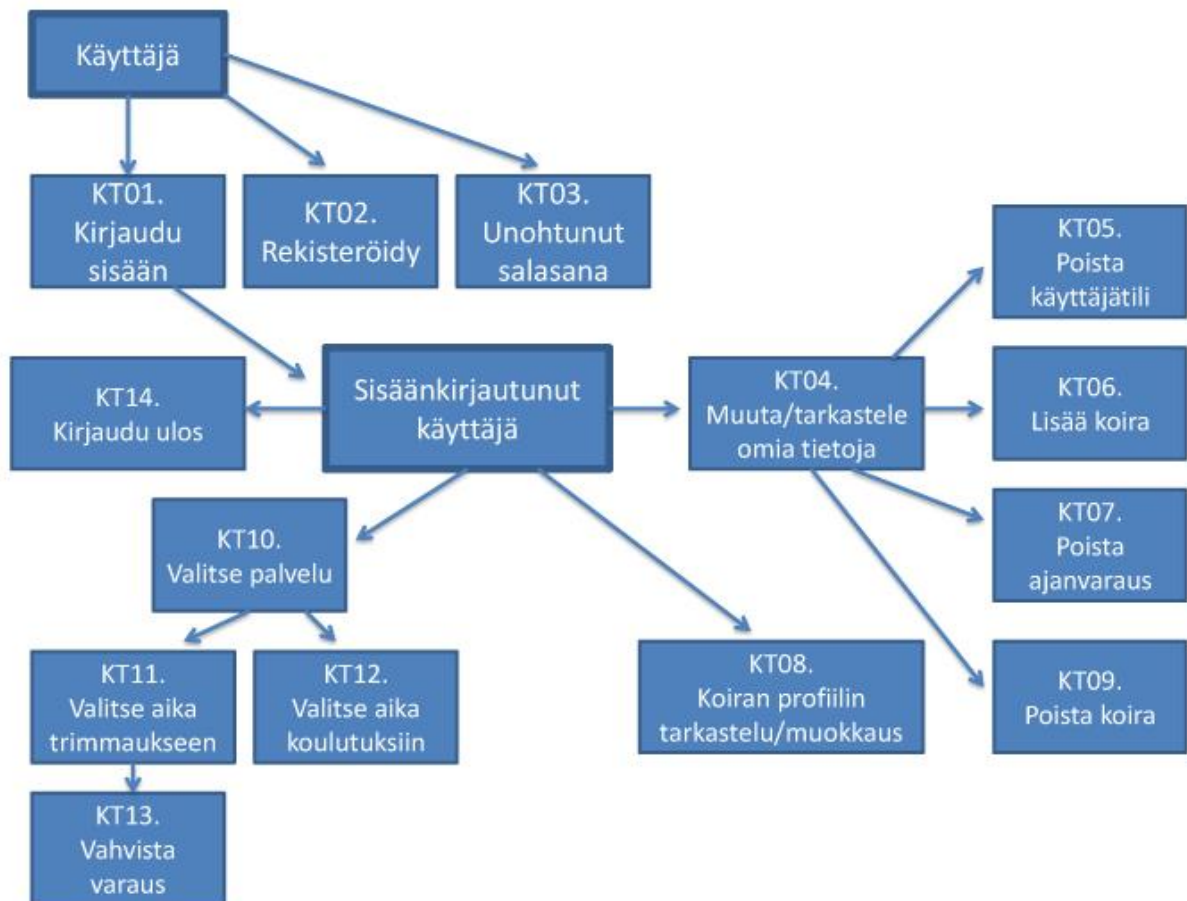
7 TURVALLISUUS

Käyttäjien materiaalit sijaitsevat tietokannassa, joka on suojattu tunnuksella ja salasanalla. Huomioon täytyy ottaa myös se, että jos jossain sovelluksen osiossa on vika, koko sovellus ei saa kaatua, vaan toiminta muualla jatkuu normaalisti. Sovellus ei saa myöskään kaatua silloin, kun sisäänkirjautuneita on useita ja tietokanta lähes täynnä materiaalia. Asiakastiedot kryptataan eli salataan tietokannassa.

8 YHTEENVETO

Tästä dokumentista löytyy ajanvarausjärjestelmän ja asiakasrekisterin toiminnot pääpiirteittäin. Dokumenttia päivitetään tietojen muuttuessa.

Liite 2. Käyttötapauskaavio-käyttäjä



Liite 3. Käyttötapaukset-käyttäjä

Nimi:	KT01 Kirjaudu sisään
Esiehdot:	Käyttäjillä ja adminilla on järjestelmään tunnukset
Suorittajat:	Käyttäjä, Admin
Kuvaus:	Kirjautusmisikkuna, johon syötetään käyttäjätunnus ja salasana. Tämän jälkeen painetaan ”Kirjaudu” –nappia.
Poikkeukset:	Väärä rekisteröimätön käyttäjätunnus ja/tai väärä salasana. Järjestelmä ilmoittaa väärästä käyttäjätunnuksesta ja/tai salasanasta ja pyytää yrittämään kirjautumista uudelleen.
Lopputulos:	Käyttäjä tai admin pääsee sisään ajanvarausjärjestelmään joko käyttäjän tai adminin oikeuksilla.
Muut vaatimukset:	Tunnuksissa huomioidaan erikseen isot ja pienet kirjaimet.

Nimi:	KT02 Rekisteröidy
Esiehdot:	Käyttäjä on sisäänkirjautumissivulla, eikä hänellä ole vielä tunnuksia järjestelmään.
Suorittajat:	Käyttäjä
Kuvaus:	Järjestelmän sisäänkirjautumissivulla on useita kenttiä, jotka on täytettävä rekisteröityäkseen järjestelmään. Pakollisia tietoja ovat sähköposti (muodossa xxxx@xxx.xx), etu- ja sukunimi, puhelinnumero, katuosoite, postinumero (5 numeroa) ja –toimipaikka, sekä salasana ja salasanan vahvistaminen. Käyttäjätunnuksena toimii asiakkaan antama sähköpostiosoite, mutta salasanan asiakas saa päättää itse (min. 5 merkkiä). Sen jälkeen painetaan rekisteröidy.
Poikkeukset:	Jos pakollisia tietoja ei ole täytetty, rekisteröityminen ei onnistu ja siitä tulee ilmoitus. Myös kahta tai useampaa samaa käyttäjätunnusta ei saa olla. Jos käyttäjätunnus on jo varattu, järjestelmä antaa siitä ilmoituksen ja kehottaa yrittämään toista käyttäjätunnusta.
Lopputulos:	Käyttäjä saa ilmoituksen rekisteröitymisen onnistumisesta ja pääsee sisään ajanvarausjärjestelmään.
Muut vaatimukset:	Joihinkin kenttiin on kirjoitettava jotain, tai rekisteröitymistä ei suoriteta.

Nimi:	KT03 Unohtunut salasana
Esiehdot:	Käyttäjä on unohtanut salasanansa
Suorittajat:	Käyttäjä
Kuvaus:	Järjestelmän kirjautumissivulla on linkki ”Unohditko salasanasi?”, jonka kautta pääsee kirjoittamaan sähköpostiosoitteen siihen varattuun kenttään. Kun osoite on kirjoitettu valittuun kohtaan painetaan ”Jatka”-painiketta.
Poikkeukset:	Sähköposti osoitteen on oltava sama kuin rekisteröidytessä järjestelmään.
Lopputulos:	Järjestelmä lähettää uuden salasanan annettuun sähköpostiosoitteeseen. Jos järjestelmä ei löydä tietokannastaan annettua sähköpostiosoitetta, seuraa ilmoitus siitä.
Muut vaatimukset:	Sähköpostiosoitte kenttään on kirjoitettava sähköpostiosoite muotoon <u>****@***.***</u> , jossa *-merkki vastaa kirjaimia. Muuten salasanaa ei lähetetä asiakkaan antamaan osoitteeseen.

Nimi:	KT04 Muuta/tarkastele omia tietoja
Esiehdot:	Käyttäjä on siirtynyt sisäänkirjautumissivulta Omat tiedot -sivulle
Suorittajat:	Käyttäjä
Kuvaus:	Sisäänkirjautumissivulta siirrytään suoraan asiakkaan Omat tiedot -sivulle. Siellä asiakas voi katsella ja muuttaa omia tietojaan. Asiakastieto -kenttiä ovat etunimi, sukunimi, puhelinnumero, katuosoite, postitoimipaikka, sähköpostiosoite, käyttäjätunnus ja salasana. Näistä kaikki ovat pakollisia. Näihin kenttiin siirtyvät automaattisesti rekisteröitymisvaiheessa annetut tiedot. Käyttäjä voi muuttaa tietojaan korjaamalla kentissä olevat tiedot ja painamalla ”Tallenna muutokset” – nappia. Avautuu vahvistusikkuna tai vahvistusteksti, jossa kerrotaan toimenpiteen onnistuneen. Salasanan voi vaihtaa sivulta löytyvän ”Uusi salasana” - kentän kautta.
Poikkeukset:	Jokin pakollinen tieto puuttui, joten avautuu ikkuna tai ilmestyy teksti, jossa sanotaan ettei tallennus onnistunut.
Lopputulos:	Asiakkaan tiedot tallennetaan tietokantoihin.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT05 Poista käyttäjätili
Esiehdot:	Käyttäjä on siirtynyt Omat tiedot –sivulle.
Suorittajat:	Käyttäjä
Kuvaus:	Omat tiedot -sivulta löytyy painike ”Poista käyttäjätili”. Tämän jälkeen avautuu ponnahdusikkuna joka kysyy ”Haluatko poistaa tämän käyttäjätilin?” Josta valitaan ”Kyllä” tai ”Ei”. Avautuu vahvistusikkuna/vahvistusteksti, jossa kerrotaan toimenpiteen onnistuneen.
Poikkeukset:	Ei poikkeuksia.
Lopputulos:	Järjestelmä poistaa asiakkaan käyttäjätilin, muttei asiakastietoja kokonaan tietokannoista. Asiakastiedot jäävät Koirakeskus Doggien säilytettäväksi.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT06 Lisää koira
Esiehdot:	Käyttäjä on siirtynyt Omat tiedot –sivulle.
Suorittajat:	Käyttäjä
Kuvaus:	Järjestelmän Omat tiedot -sivulla on linkki ”Lisää uusi koira järjestelmään”, josta järjestelmä siirtyy Koiran tiedot-sivulle. Sen kautta voi lisätä uuden koiran tietokantaan. Koiran tiedot –sivulta löytyy kentät: Koiran kutsumanimi, rotu (pudotusvalikko), koiran syntymävuosi, koiran sukupuoli ja lisätietoja. Pakollisia kenttiä ovat kutsumanimi, rotu, syntymävuosi ja sukupuoli. Lisäksi vapaaehtoisena toimintona sivulta löytyy koiran kuvan lisääminen. Kuvan saa tuotua omalta koneelta painamalla ”Selaa” -painiketta. Koiran saa lisättyä järjestelmään painamalla ”Tallenna” –painiketta. Avautuu vahvistusikkuna, jossa kerrotaan toimenpiteen onnistuneen.
Poikkeukset:	Jokin pakollinen tieto puuttui, joten avautuu ikkuna, jossa sanotaan ettei tallennus onnistunut.
Lopputulos:	Koira lisätään tietokantoihin ja koiran nimi profiililinkeineen siirtyy Omat tiedot –sivulle.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT07 Poista ajanvaraus
Esiehdot:	Käyttäjä on siirtynyt Omat tiedot –sivulle.
Suorittajat:	Käyttäjä
Kuvaus:	Omat tiedot –sivulle ilmestyy asiakkaan tekemät ajanvaraukset. Asiakas voi poistaa ajanvarauksen klikkaamalla ”Poista varaus” linkkiä/nappia. Sen jälkeen aukeaa ikkuna jossa kysytään ”Haluatko poistaa tämän ajanvarauksen? Kyllä, Ei?”
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä poistaa ajanvarauksen tietokannoistaan.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT08 Koiran profiilin tarkastelu/muokkaus
Esiehdot:	Käyttäjä on siirtynyt Omat tiedot –sivulle.
Suorittajat:	Käyttäjä
Kuvaus:	Omat tiedot –sivulla on koiran nimestä muodostunut linkki, jota klikkaamalla pääsee tarkastelemaan/muokkaamaan koiran tietoja. Kenttiä voi nyt muokata ja koiran kuvaa vaihtaa. Painamalla ”Tallenna” -nappia muutokset tallentuvat tietokantaan ja avautuu vahvistusikkuna, jossa kerrotaan toimenpiteen onnistuneen
Poikkeukset:	Koira on pitänyt lisätä jo aiemmin Koiran tiedot –sivulta tietokantaan, jotta päästään katsomaan koiran profiilia.
Lopputulos:	Järjestelmä tallentaa muutokset tietokantaan.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT09 Poista koira
Esiehdot:	Käyttäjä on siirtynyt Omat tiedot -sivulle tai käyttäjä on siirtynyt Omat tiedot -sivulta Koiran profiili -sivulle.
Suorittajat:	Käyttäjä
Kuvaus:	Järjestelmän Omat tiedot -sivulla on linkki koiran nimestä koiran profiiliin. Sen kautta pääsee poistamaan koiran. Tai koiran voi poistaa myös suoraan Omat tiedot -sivulta koiran nimen vieressä olevasta Poista koiran tiedot -painikkeesta. Tämän jälkeen avautuu ponnahdusikkuna joka kysyy ”Haluatko poistaa tämän koiran?” Josta valitaan ”Kyllä” tai ”Ei”. Avautuu vahvistusikkuna, jossa kerrotaan toimenpiteen onnistuneen.
Poikkeukset:	Ei poikkeuksia.
Lopputulokset:	Järjestelmä poistaa koiran tietokannoista.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT10 Valitse palvelu
Esiehdot:	Käyttäjä on siirtynyt Omat tiedot -sivulle
Suorittajat:	Käyttäjä
Kuvaus:	Omat tiedot -sivulla on painike ”Uusi trimmausaika”, jonka kautta pääsee varaamaan trimmausajan. Omat tiedot -sivulla on myös toinen linkki ”Ilmottaudu koulutuksiin”, jonka kautta asiakkaat pääsevät varaamaan kurssiaikoja. Palveluryhmä kohtaisesti valitaan palvelut ja halutut toimenpiteet, sekä valitaan koira jolle aika varataan ja painetaan Seuraava -nappia.
Poikkeukset:	Jos koira ei löydy pudotusvalikosta, sitä ei ole vielä järjestelmässä. Silloin voi painaa ”Lisää koira” – painiketta, jolloin käynnistyy ponnahdusikkuna ”Koiran tiedot” -sivulle. Kun koira on lisätty (Tallenna-painikkeella), ponnahdusikkuna voidaan nyt sulkea ja koira löytyy ”Valitse palvelu” -sivulta. Jokin pakollinen tieto puuttui, joten avautuu ikkuna, jossa sanotaan ettei tallennus onnistunut.
Lopputulos:	Koira tallentuu tietokantoihin.
Muut vaatimukset:	Ei muita vaatimuksia.

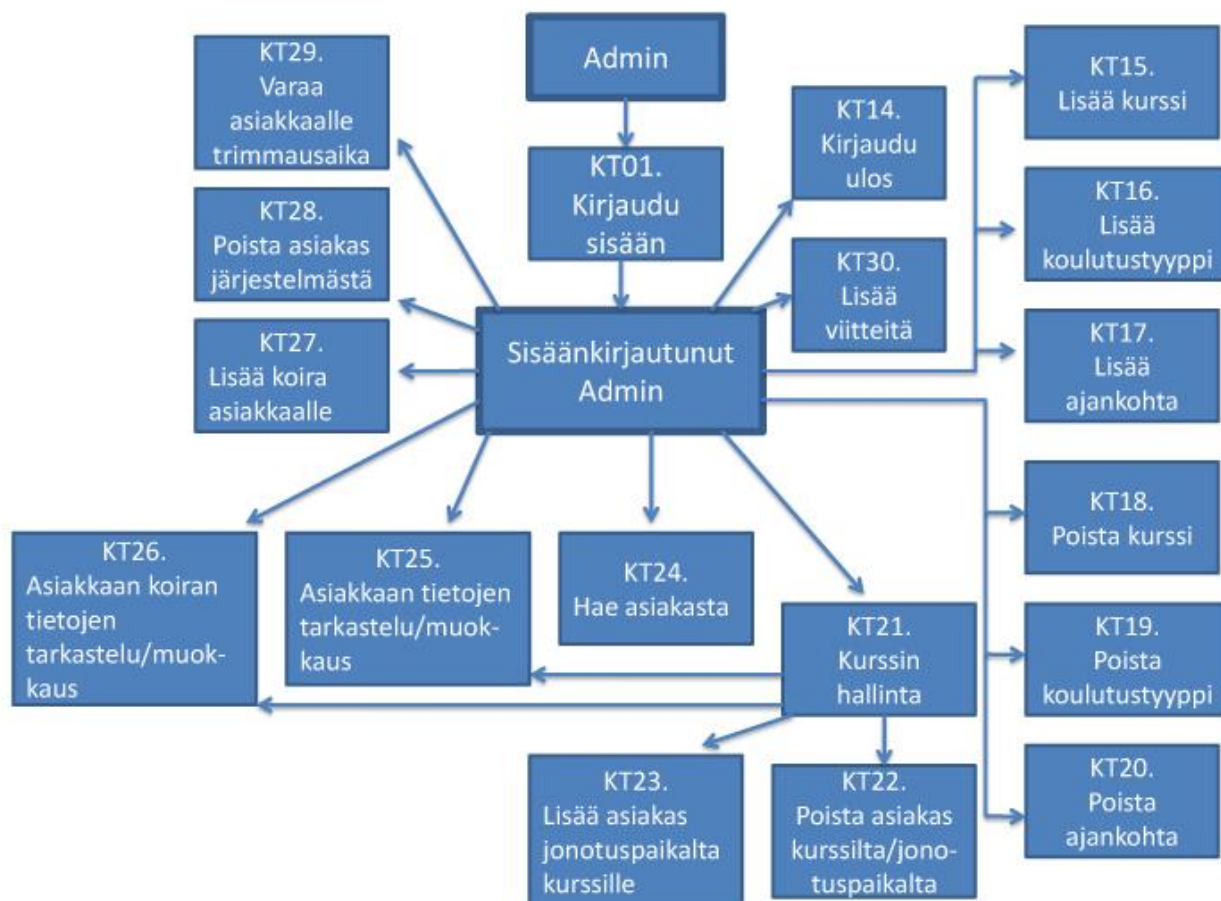
Nimi:	KT11 Valitse aika trimmaukseen
Esiehdot:	Käyttäjä on valinnut Uusi trimmausaika -linkin, siirtynyt ajavaraukseen osioon sekä valinnut haluamansa palvelut koiralle.
Suorittajat:	Käyttäjä
Kuvaus:	Ajanvaraukshoidossa näkyy kalenteri, josta voi varata haluamansa vapaan ajan hiirellä klikkaamalla. Vapaat ajat näkyvät sinisenä. Kalenterissa voi vaihtaa päiviä painamalla kalenterin yläreunoissa olevia ”Seuraava päivä” ja ”Edellinen päivä” painikkeita. Kalenterin yläosasta löytyy myös pikkukalenteri, jota klikkaamalla voi helposti selata viikkoja ja kuukausia eteenpäin ja taaksepäin. Näkyvissä ovat molempien työntekijöiden kalenterit, ja asiakas voi varata ajan kummasta tahansa. Kun vapaa aika on löytynyt, käyttäjä pääsee eteenpäin klikkaamalla sivun alaosasta olevaa Seuraava -nappia.
Poikkeukset:	Ei poikkeuksia, mutta mikäli seuraavalle sivulle menee valitsematta aikaa, asiasta huomautetaan, eikä varaamisprosessia voi jatkaa.
Lopputulos:	Aika tallentuu tietokantoihin.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT12 Valitse aika koulutuksiin
Esiehdot:	Käyttäjä on valinnut Ilmottaudu kursseille -linkin ja siirtynyt kursseilleilmottautumisosioon.
Suorittajat:	Käyttäjä
Kuvaus:	Käyttäjä on valinnut Kursseille ilmottautumisen palveluryhmäksi. Sen jälkeen aukeaa sivu jossa on ”Valitse palvelu”-pudotusvalikko, josta käyttäjä valitsee mieleisensä kurssin esim. Agilityn alkeet. Tämän jälkeen aukeaa vielä toinen pudotusvalikko jonne on listattu kaikki tulevat Agilityn alkeet -kurssit ja niiden aloitus- ja lopetuspäivämäärät. Käyttäjä valitsee sitten mieleisensä päivämäärän ja koiran , jonka ilmoittaa kurssille. Tämän jälkeen järjestelmä ilmoittaa onko kurssilla vapaita paikkoja. Jos on käyttäjä painaa Seuraava -painiketta, muuten käyttäjä saa ilmoituksen siitä, että kurssi on täynnä. Samalla järjestelmä kysyy haluaako käyttäjä jonotuspaikan kurssille vai haluaako käyttäjä kokeilla onko eri ajankohdassa kurssilla tilaa. Jos käyttäjä valitsee haluavansa jonotuspaikan tulee ilmoitus eri ponnahdusikkunaan kurssin ajankohdasta ja vahvistus jonotuspaikasta, sekä viesti, että asiakkaalle ilmoitetaan, kun tilaa on.
Poikkeukset:	Jos kursseille ei ole tilaa, asiakas ei pääse eteenpäin painamalla Seuraava -nappia.
Lopputulos:	Varaus tallentuu tietokantoihin.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT13 Vahvista varaus
Esiehdot:	Käyttäjä on siirtynyt ajanvaraussivulle ja sieltä Vahvista varaus -osioon.
Suorittajat:	Käyttäjä
Kuvaus:	Järjestelmä kysyy haluaako käyttäjä vahvistaa varauksensa ja antaa samalla käyttäjälle tiedot varauksen päivämäärästä ja kestosta sekä varauksen perumisehdoista (Perua saa vain omalla tai koiran lääkärin todistuksella). Koulutusten vahvistuksissa järjestelmä myös kehoittaa käyttäjän maksamaan Doggien tilille annetulla viitteellä tarvittavan summan viimeistään viikkoa ennen kurssin alkua (5 kerran kurssi=55 e ja 10 kerran kurssi 110 e). Tarvittaessa koulutukseen osallistuminen peruttava 1 vkoa ennen kurssin alkua. Trimmausta vahvistettaessa järjestelmä kehottaa lisäksi varaamaan mukaan käteistä ja tarvittaessa perumaan varauksen viimeistään 1 vrk ennen trimmausaikaa. Muuten veloitetaan summa joka muutenkin olisi mennyt trimmaukseen
Poikkeukset:	Asiakas voi painaa Peruuta -nappia, jolloin varausta ei vahvisteta ja se peruuntuu. Asiakas siirtyy takaisin Omat tiedot -sivulle.
Lopputulos:	Varaus tallentuu tietokantoihin.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT14 Kirjaudu ulos
Esiehdot:	Käyttäjä/Admin on kirjautuneena järjestelmään
Suorittajat:	Käyttäjä, Admin
Kuvaus:	Käyttäjä/Admin voi olla millä sivulla tahansa järjestelmässä ja painaa Kirjaudu ulos-nappia.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä kirjaa käyttäjän ulos ja siirtyy sisään kirjautumis-sivulle
Muut vaatimukset:	Ei muita vaatimuksia.

Liite 4. Käyttötapauskaavio-Admin



Liite 5. Käyttötapaukset-Admin

Nimi:	KT01 Kirjaudu sisään
Esiehdot:	Käyttäjillä ja adminilla on tunnukset järjestelmään.
Suorittajat:	Käyttäjä, Admin
Kuvaus:	Kirjautusmisikkuna, johon syötetään käyttäjätunnus ja salasana. Tämän jälkeen painetaan ”Kirjaudu” –nappia.
Poikkeukset:	Väärä rekisteröimätön käyttäjätunnus ja/tai salasana. Järjestelmä ilmoittaa väärästä käyttäjätunnuksesta ja/tai salasanasta ja pyytää yrittämään kirjautumista uudelleen.
Lopputulos:	Käyttäjä tai admin pääsee sisään ajanvarausjärjestelmään joko käyttäjän tai adminin oikeuksilla.
Muut vaatimukset:	Tunnuksissa huomioidaan erikseen isot ja pienet kirjaimet.

Nimi:	KT14 Kirjaudu ulos
Esiehdot:	Käyttäjä/Admin on kirjautuneena järjestelmään
Suorittajat:	Käyttäjä, Admin
Kuvaus:	Käyttäjä/Admin voi olla millä sivulla tahansa järjestelmässä ja painaa Kirjaudu ulos-nappia.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä kirjaa käyttäjän ulos ja siirtyy sisään kirjautumissivulle
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT15 Lisää kurssi
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt etusivulta kurssienhallintaosioon. Klikkaamalla Lisää kurssi -linkkiä Admin siirtyy Lisää kurssi sivulle. Siellä valitaan ensin koulutustyyppi, jonka alle uusi kurssi lisätään. Sen jälkeen kirjoitetaan sivulla oleviin kenttiin kurssin nimi ja haluttu ajankohta. Kenttiä ei saa jättää tyhjäksi. Sitten painetaan Tallenna-nappia.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä lisää uuden kurssin tietokantaan.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT16 Lisää koulutustyyppi
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt etusivulta kurssienhallintaosioon. Klikkaamalla Lisää koulutustyyppi -linkkiä Admin siirtyy Lisää koulutustyyppi -sivulle. Siellä kirjoitetaan uusi koulutustyyppi tekstikenttään ja painetaan Tallenna-nappia. Tyhjiä kenttiä ei saa olla.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä lisää uuden koulutustyyppin tietokantaan.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT17 Lisää ajankohta
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt etusivulta kurssienhallintaosioon. Klikkaamalla Lisää ajankohta-nappia/linkkiä Admin siirtyy Lisää ajankohta -sivulle. Siellä valitaan ensin koulutustyyppi ja sen jälkeen kurssi, jolle lisätään uusi ajankohta kirjoittamalla se tekstikenttään. Sitten painetaan Tallenna-nappia. Tyhjiä kenttiä ei saa olla.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä lisää uuden ajankohdan tietokantaan valitulle kurssille. Eli tietokantaan ilmestyy valitun kurssin nimi ja ajankohta heti kurssin nimen perään siinä muodossa missä se on kirjoitettu esim. Agilityn alkeet 14.5.2011-13.6.2011 klo 17.00-18.00.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT18 Poista kurssi
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt etusivulta kurssienhallintaosioon. Klikkaamalla Poista kurssi -linkkiä Admin siirtyy Poista kurssi -sivulle. Siellä valitaan kurssi ja painetaan Poista -nappia.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä poistaa kurssin tietokannasta.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT19 Poista koulutustyyppi
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt etusivulta kurssienhallintaan. Klikkaamalla Poista koulutustyyppi -linkkiä Admin siirtyy Poista koulutustyyppi -sivulle. Sieltä valitaan koulutustyyppi ja painetaan Poista -nappia.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä poistaa koulutustyyppin tietokannasta.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT20 Poista ajankohta
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt etusivulta kurssienhallintaosioon. Klikkaamalla Poista ajankohta -linkkiä Admin siirtyy Poista ajankohta -sivulle. Sieltä valitaan koulutustyyppi, kurssi ja haluttu ajankohta, joka poistetaan painamalla Poista -nappia.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä poistaa ajankohdan tietokannasta, joka on muotoa kurssin nimi-ajankohta (Agilityn alkeet 14.5.2011-13.6.2011 klo 17.00-18.00.).
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT21 Kurssin hallinta
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt etusivulta kurssienhallintaosioon. Klikkaamalla Kurssien hallintasivulla näkyviä kurssien nimiä, Admin pääsee kurssin hallinta sivulle. Siellä Admin voi valita kyseessä olevan kurssin ajankohdan ja klikata Hae -nappia. Tämän jälkeen listataan siihen ajankohtaan kurssille ilmottautuneet henkilöt ja jonotuspaikalla olijat.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä listaa kurssille ilmottautuneet.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT22 Poista asiakas kurssilta/jonotuspaikalta
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt etusivulta kurssien hallintaosioon ja sieltä kurssinhallintasivulle. Admin pystyy poistamaan kurssilta ja jonotuspaikalta henkilöitä klikkaamalla henkilön vieressä olevaa Poista -linkkiä/nappia.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä poistaa henkilön jonotuspaikalta/kurssilta, mutta ei tietokannasta.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT23 Lisää asiakas jonotuspaikalta kurssille
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt etusivulta kurssien hallintaosioon ja sieltä kurssinhallintasivulle. Admin pystyy lisäämään henkilön jonotuspaikalta kurssille, klikkaamalla jonotuspaikalla olevan henkilön vieressä olevaa Lisää kurssille -linkkiä/nappia.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä lisää henkilön kurssille jonotuspaikalta.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT24 Hae asiakasta
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt asiakashallintasivulle. Admin pystyy hakemaan asiakasta nimen ja/tai puhelinnumeron perusteella. Kun hakukenttä/-kentät on täytetty, painetaan Hae -nappia. Haun tulokset listataan linkkeinä hakukenttien alapuolelle.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä hakee asiakkaan tietokannasta.
Muut vaatimukset:	Johonkin hakukenttään on kirjoitettava jotain, että haku suoritettaisiin.

Nimi:	KT25 Asiakkaan tietojen tarkastelu/muokkaus
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt kurssinhallintasivulle tai asiakashallintasivulle. Kurssinhallintasivulta Admin pääsee koiran tietoihin käsiksi klikkaamalla ensin asiakkaan (koiran omistajan) nimestä muodostunutta linkkiä. Asiakashallintasivulta, kun Admin on hakenut asiakasta, hän pääsee asiakkaan tietoihin klikkaamalla samanlailla asiakkaan nimestä muodostunutta linkkiä. Tietojen muokkaaminen tapahtuu muuttamalla haluttuja kenttiä ja painamalla Tallenna muutokset -nappia.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä tallentaa muutokset tietokantaan.
Muut vaatimukset:	Pakollinen tieto tulee olla aina täytettynä, muuten muutoksia ei tallenneta.

Nimi:	KT26 Asiakkaan koiran tietojen tarkastelu/muokkaus
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt Kurssin hallinta -sivulle tai Asiakashallinta -sivulle. Kurssin hallinta sivulta Admin pääsee koiran tietoihin käsiksi klikkaamalla ensin asiakkaan (koiran omistajan) nimestä muodostunutta linkkiä. Asiakashallinta -sivulta, kun Admin on hakenut asiakasta, hän pääsee asiakkaan koirantietoihin klikkaamalla samanlailla asiakkaan nimestä muodostunutta linkkiä. Kun Asiakkaan tiedot -sivu aukeaa sieltä löytyy listattuina linkkeinä asiakkaan koira/(t), joita klikkaamalla aukeaa Koiran tiedot-sivu. Siellä voi tarkastella koiran tietoja tai muokata niitä muuttamalla kenttiä ja painamalla Tallenna -nappia.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä tallentaa muutokset tietokantaan.
Muut vaatimukset:	Pakollinen tieto tulee olla aina täytettynä, muuten muutoksia ei tallenneta.

Nimi:	KT27 Lisää koira asiakkaalle
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt Asiakkaan tiedot -sivulle. Klikkaamalla sivuston oikealla olevaa Lisää koira -linkkiä Admin pystyy lisäämään asiakkaalle koiria (kts. käyttötapaus KT06).
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Järjestelmä tallentaa koiran tietokantaan.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT28 Poista asiakas järjestelmästä
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt Asiakkaan tiedot -sivulle. Klikkaamalla Poista asiakas järjestelmästä -nappia, asiakas poistetaan.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Asiakas poistetaan pysyvästi tietokannasta.
Muut vaatimukset:	Ei muita vaatimuksia.

Nimi:	KT29 Varaa asiakkaalle trimmausaika
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt Asiakkaan tiedot -sivulle. Klikkaamalla Varaa asiakkaalle trimmausaika -linkkiä, Admin pystyy tekemään asiakkaalle ajanvarauksia (Kts. käyttötapaus KT11)
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Asiakkaan ajanvaraus tallentuu tietokantaan.
Muut vaatimukset:	Ei muita vaatimuksia.

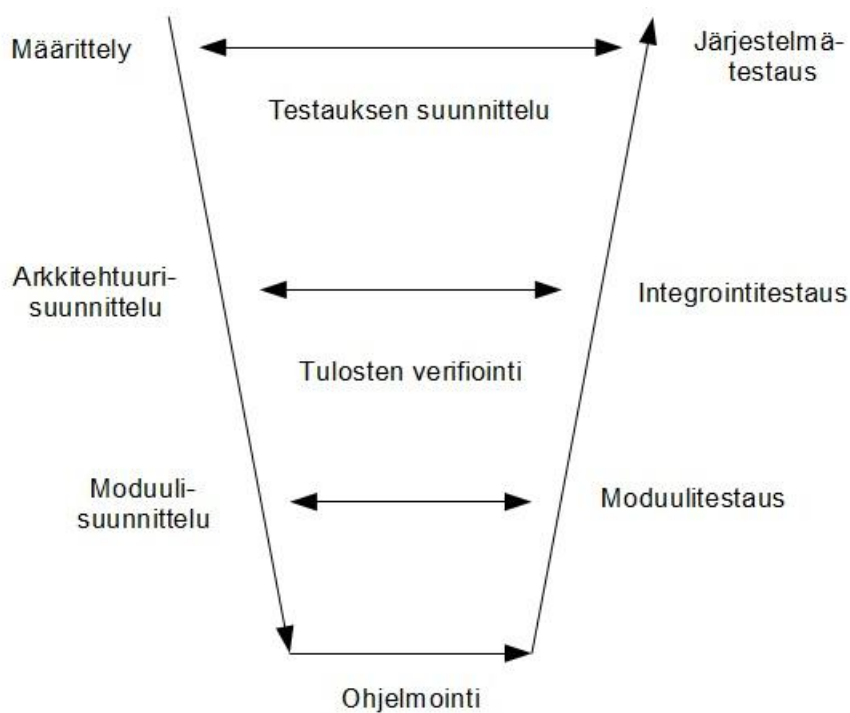
Nimi:	KT30 Lisää viitteitä
Esiehdot:	Admin on kirjautuneena järjestelmään
Suorittajat:	Admin
Kuvaus:	Admin on siirtynyt etusivulta Lisää viitteitä -sivulle. Täällä Admin pystyy lisäämään pankista saatuja viitteitä tietokantaan, joita käytetään kurssien maksuissa. Admin kirjoittaa Syötä viite -kenttään viitteen ja painaa Lisää -nappia.
Poikkeukset:	Ei poikkeuksia
Lopputulos:	Viite tallentuu tietokantaan.
Muut vaatimukset:	Ei muita vaatimuksia.

Liite 6. Testaussuunnitelma

Versio	Päivämäärä	Kommentit	Muokkaaaja
0.1	10.4.2011	Testaussuunnitelman aloittaminen	Seela Ihanainen
0.2	19.4.2011	Testaussuunnitelman korjausta	Seela Ihanainen
0.3	26.4.2011	Testaussuunnitelman tarkennusta	Seela Ihanainen

1 TESTAUKSEN PERIAATTEET

Testaus toteutetaan V-mallin mukaisesti ja sen tarkoituksena on löytää ohjelmasta mahdollisia virheitä.



Kuvio 1. Testauksen V-malli

1. Moduulitestauksessa (osatestaus) testataan yksittäisiä moduuleja. Testaus toteutetaan ulkoisena testauksena, eli tulosarvojen oikeellisuutta tarkastellaan ja verrataan niitä syöttöarvoihin. Tarkoituksena on siis testata sitä, että toimiiko järjestelmän osa siten kuin sen on suunnitelmissa kuvattu.

2. Integroititestauksessa testataan moduulin yhteistoimintaa. Tämä tarkoittaa sitä, että järjestelmää testatattaessa katsotaan toimivatko järjestelmän osat yhdessä rakennekuvauksien (arkkitehtuurin) mukaisesti.
3. Järjestelmätestauksessa testataan koko valmista järjestelmää. Sitä verrataan määrittyskuvausten sääntöihin eli testataan, että toimiiko järjestelmä määritellyllä tavalla.
4. Testiaineistot tullaan suunnittelemaan siten, että ne sisältävät sekä oikeita, että virheellisiä syöttöarvoja.

2 TESTAUSSUUNNITELMA

- Sivujen toimivuus testataan eri selaimilla (Internet Explorer 7, Mozilla Firefox 3 ja Opera 11.10).
- Aloitussivu
- Tietojen syöttö ja syötön oikeellisuus
- Tietojen haku ja haun oikeellisuus
- Tietojen poisto ja poiston oikeellisuus.
- Virhetilanteet ja niistä toipuminen
- Eri käyttäjäryhmien toiminnot
- sisäänkirjautuminen ja järjestelmästä poistuminen
- Määrittys- ja suunnitteluvaiheen suunnitelmien mukainen toiminta

Testaukset aloitetaan kun testattavia sivuja ja toimintoja saatu valmiiksi. Testauksessa käydään läpi, sitä ovatko sivut määrittely ja suunnittelukuvausten mukaiset sekä niiden toimivuus ja ulkoasu.

Asiakkaan suorittama testaus

Asiakas saa sovelluksen testattavakseen ja voi silloin arvioida vastaako sovellus ja käytettävyys hänen tarpeitansa. Asiakas testaa sovellusta ja katsoo, vastaako sen käytettävyys ja sopivuus hänen tarkoituksiaan. Tästä on hyötyä myös siksi, että yleensä loppukäyttäjien toimintatavat poikkeavat järjestelmän suunnittelijoiden/tekijöiden tottuneista toimintatavoista. Testaustapana on mahdollisesti kognitiivinen läpikäynti.

Testauksen läpikäynti

Läpikäynnissä käyttäjälle annetut tehtävät on jaettu pieniin osiin ja suoritetaan järjestyksessä. Sen jälkeen järjestelmän tekijät käyvät osatehtävät läpi ja arvioivat sitä käyttäen apuna neljää seuraavaa kysymystä:

1. Yrittääkö käyttäjä päästä oikeaan tavoitteeseen ja ymmärtääkö hän, että vaihe on osa kokonaisuutta?
2. Onko oikea toiminto löydettävissä käyttäjälle eli huomaako hän sen heti?
3. Onko käyttäjälle asianyhteys selvä järjestelmän toiminnon ja hänen tavoittelemansa toiminnon välillä?
4. Kun käyttäjä toimii oikein (suorittaa oikean toiminnon) huomaako hän sitä?
 - eli saako käyttäjä riittävän selkeää palautetta toiminnon suorittamisesta.

3 TOIMINNALLISET VAATIMUKSET

Testataan onko lähtökohtakuvauksen vaatimukset toteutuneet.

#	Koirakeskus Doggien asiakkaan käyttöliittymän toiminnot
1	Kirjaudu sisään
2	Rekisteröidy
3	Kirjaudu ulos
4	Unohtunut salasana
5	Muokkaa omia tietoja
6	Poista käyttäjätili
7	Lisää koira
8	Poista ajanvaraus
9	Muokkaa koiran tietoja
10	Poista koira
11	Valitse palvelu
12	Valitse aika trimmaukseen
13	Vahvista varaus (trimmaus)
14	Valitse aika koulutuksiin
15	Vahvista varaus (koulutukset)

#	Ylläpitäjän/Adminin käyttöliittymän toiminnot
16	Lisää kurssi
17	Lisää koulutustyyppi
18	Lisää ajankohta
19	Poista kurssi
20	Poista koulutustyyppi
21	Poista ajankohta
22	Kurssin hallinta
23	Poista asiakas kurssilta/jonotuspaikalta

24	Lisää asiakas jonotuspaikalta kurssille
25	Hae asiakasta
26	Asiakkaan tietojen tarkastelu/muokkaus
27	Asiakkaan koiran tietojen tarkastelu/muokkaus
28	Lisää koira asiakkaalle
29	Poista asiakas järjestelmästä
30	Varaa asiakkaalle trimmausaika
31	Lisää viitteitä

Muut vaatimukset

#	Vaatus
32	Järjestelmässä on voitava olla useita käyttäjiä kirjautuneena samanaikaisesti.
33	Järjestelmä toimii internetissä.

Liite 7. Testitapaukset

Versio	Päivämäärä	Kommentit	Muokkaaja
0.1	16.4.2011	Testitapausten aloittaminen	Seela Ihanainen
0.2	20.4.2011	Testitapausten jatkaminen	Seela Ihanainen
0.3	25.4.2011	Testitapausten jatkaminen ja korjailu	Seela Ihanainen
0.4	26.4.2011	Testitapausten korjailu	Seela Ihanainen

SISÄLTÖ

1	TESTITAPAUKSET	112
1.1	Kirjaudu sisään	112
1.2	Rekisteröidy	112
1.3	Kirjaudu ulos	114
1.4	Unohtunut salasana	114
1.5	Muokkaa omia tietoja	115
1.6	Poista käyttäjätili	115
1.7	Lisää koira	115
1.8	Poista ajanvaraus	116
1.9	Muokkaa koiran tietoja	116
1.10	Poista koira	116
1.11	Valitse palvelu	117
1.12	Valitse aika trimmaukseen	117
1.13	Vahvista varaus (trimmaus)	117
1.14	Valitse aika koulutuksiin	118
1.15	Vahvista varaus (koulutukset)	119
1.16	Lisää kurssi.....	119
1.17	Lisää koulutustyyppi	120
1.18	Lisää ajankohta	120
1.19	Poista kurssi.....	120
1.20	Poista koulutustyyppi	121
1.21	Poista ajankohta	121
1.22	Kurssin hallinta	121
1.23	Poista asiakas kurssilta/jonotuspaikalta.....	122
1.24	Lisää asiakas jonotuspaikalta kurssille.....	122
1.25	Hae asiakasta.....	122
1.26	Asiakkaan tietojen tarkastelu/muokkaus	123
1.27	Asiakkaan koiran tietojen tarkastelu/muokkaus	124
1.28	Lisää koira asiakkaalle.....	124
1.29	Poista asiakas järjestelmästä	124
1.30	Varaa asiakkaalle trimmausaika	125
1.31	Lisää viitteitä	125

1 TESTITAPAUKSET

Tässä kappaleessa on kuvattuna testitapaukset, jotka on ryhmiteltyinä tehtävittäin. Järjestelmän tehtäviä kohti on useita testitapauksia. Niistä osa voi päättyä virhetilanteeseen. Toimintojen tulee olla suunnitteludokumentin toimintojen mukaisia, jotka puolestaan perustuvat vaatimusmäärittely-dokumentin vaatimuksiin. Testitapaukset on suunniteltu ”Harmaalaatikko” -periaatteella.

Harmaalaatikotestauksessa käytetään hyväksi tietoa järjestelmän toteutusperiaatteista, eli silloin tunnetaan jo järjestelmän toimintaa.

1.1 Kirjaudu sisään

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT01	Koirakeskus Doggien asiakas	Kirjaudutaan järjestelmään oikealla käyttäjätunnuksella ja salasanalla	Siirrytään Omat tiedot - sivulle	OK
TT02	Koirakeskus Doggien asiakas	Kirjaudutaan järjestelmään oikealla käyttäjätunnuksella ja väärällä salasanalla	Tulee ilmoitus kirjautumisen epäonnistumisesta	OK
TT03	Koirakeskus Doggien asiakas	Kirjaudutaan järjestelmään väärällä käyttäjätunnuksella ja jollain oikealla salasanalla	Tulee ilmoitus kirjautumisen epäonnistumisesta	OK
TT04	Koirakeskus Doggien asiakas	Kirjaudutaan järjestelmään pelkällä oikealla käyttäjätunnuksella	Tulee ilmoitus kirjautumisen epäonnistumisesta	OK
TT05	Koirakeskus Doggien asiakas	Kirjaudutaan järjestelmään pelkällä oikealla salasanalla	Tulee ilmoitus kirjautumisen epäonnistumisesta	OK
TT06	Koirakeskus Doggien asiakas	Kirjaudutaan järjestelmään pelkällä väärällä käyttäjätunnuksella	Tulee ilmoitus kirjautumisen epäonnistumisesta	OK

1.2 Rekisteröidy

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT07	Koirakeskus Doggien asia	Täytetään kaikki kentät oikeilla tiedoilla	Tulee ilmoitus rekisteröitymisen onnistumisesta	OK
TT08	Koirakeskus Doggien	Täytetään oikeassa muodossa pelkkä sähköpostiosoite (xxx@xx.xx)	Rekisteröityminen ei onnistu ja tulee ilmoitus puuttuvista	OK

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
	asiakas		tiedoista	
TT09	Koirakeskus Doggien asiakas	Täytetään oikeassa muodossa sähköposti- ja etunimi-kenttä	Rekisteröityminen ei onnistu ja tulee ilmoitus puuttuvista tiedoista	OK
TT10	Koirakeskus Doggien asiakas	Täytetään oikeassa muodossa sähköposti- ,etunimi- ja sukunimi-kenttä	Rekisteröityminen ei onnistu ja tulee ilmoitus puuttuvista tiedoista	OK
TT11	Koirakeskus Doggien asiakas	Täytetään oikeassa muodossa sähköposti-, etunimi-, sukunimi-, katuosoite-kenttä	Rekisteröityminen ei onnistu ja tulee ilmoitus puuttuvista tiedoista	OK
TT12	Koirakeskus Doggien asiakas	Täytetään oikeassa muodossa sähköposti-, etunimi-, sukunimi-, katuosoite-, postinumerokenttä	Rekisteröityminen ei onnistu ja tulee ilmoitus puuttuvista tiedoista	OK
TT13	Koirakeskus Doggien asiakas	Täytetään oikeassa muodossa sähköposti-, etunimi-, sukunimi-, katuosoite-, postinumero-, postitoimipaikkakenttä	Rekisteröityminen ei onnistu ja tulee ilmoitus puuttuvista tiedoista	OK
TT14	Koirakeskus Doggien asiakas	Täytetään oikeassa muodossa sähköposti-, etunimi-, sukunimi-, katuosoite-, postinumero-, postitoimipaikka-, puhelinnumero- kenttä	Rekisteröityminen ei onnistu ja tulee ilmoitus puuttuvista tiedoista	OK
TT15	Koirakeskus Doggien asiakas	Täytetään oikeassa muodossa kaikki kentät paitsi jätä vahvista salasana- kenttä tyhjäksi	Rekisteröityminen ei onnistu ja tulee ilmoitus puuttuvista tiedoista	OK
TT16	Koirakeskus Doggien asiakas	Täytetään oikeassa muodossa kaikki kentät paitsi jätä salasana-kenttä tyhjäksi	Rekisteröityminen ei onnistu ja tulee ilmoitus puuttuvista tiedoista	OK
TT17	Koirakeskus Doggien asiakas	Täytetään kaikki kentät oikeassa muodossa	Tulee ilmoitus rekisteröitymisen onnistumisesta	OK
TT18	Koirakeskus Doggien asiakas	Täytetään kaikki kentät oikeassa muodossa paitsi sähköposti muotoa (xxx@)	Tulee ilmoitus vääränlaisesta sähköpostiosoitteesta	OK
TT19	Koirakeskus Doggien asiakas	Täytetään kaikki kentät oikeassa muodoossa paitsi postinumerokenttä vain 4 numerolla	Tulee ilmoitus vääränlaisesta postinumerosta	OK

1.3 Kirjaudu ulos

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT20	Koirakeskus Doggien asiakas	Kirjaututaan ulos heti sisään kirjautumisen jälkeen.	Ulos kirjautuminen onnistuu ja siirrytään sisäänkirjautumis-sivulle	OK

1.4 Unohtunut salasana

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT21	Koirakeskus Doggien asiakas	Kirjoitetaan unohtunut salasana kohdassa olevaan sähköposti- kenttään väärä sähköpostiosoite, jota ei löydy tietokannasta.	Tulee ilmoitus väärästä sähköpostiosoitteesta	OK
TT22	Koirakeskus Doggien asiakas	Kirjoitetaan unohtunut salasana kohdassa olevaan sähköposti- kenttään oikea sähköpostiosoite, joka löytyy tietokannasta.	Tulee ilmoitus uuden salasanan lähettämisestä ja maili siitä saapuu oikeaan sähköpostiin	OK

1.5 Muokkaa omia tietoja

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT23	Koirakeskus Doggien asiakas	Muokataan mitä tahansa tietoa ja paina tallenna	Tulee ilmoitus onnistuneista muutoksista	OK
TT24	Koirakeskus Doggien asiakas	Muokataan mitä tahansa tietoa ja jätä pakollinen kenttä esim. puhelinnumero tyhjäksi	Tulee ilmoitus pakollisen tiedon muuttumisesta, eikä muutosten tallentaminen onnistu	OK

1.6 Poista käyttäjätili

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT23	Koirakeskus Doggien asiakas	Painetaan Poista käyttäjätili -painiketta	Ilmestyy kysymys halutaanko käyttäjätili varmasti poistaa, jos otetaan kyllä, käyttäjätili poistuu. Tiedot eivät kuitenkaan katoa tietokannasta vaan ovat vielä Koirakeskus Doggien käytettävissä	OK
TT24	Koirakeskus Doggien asiakas	Painetaan Poista käyttäjätili -painiketta. Kun kysytään halutaanko käyttäjätili poistaa, otetaan ei	Tulee ilmoitus pakollisen tiedon muuttumisesta, eikä muutosten tallentaminen onnistu	OK

1.7 Lisää koira

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT25	Koirakeskus Doggien asiakas	Täytetään kaikki kentät	Lisäys järjestelmään onnistuu	OK
TT26	Koirakeskus Doggien asiakas	Jätetään kaikki kentät täyttämättä	Lisäys järjestelmään epäonnistuu, tulee ilmoitus puuttuvista tiedoista	OK
TT27	Koirakeskus Doggien asiakas	Jätetään kutsumanimi-kenttä täyttämättä, täytetään muut kentät	Lisäys järjestelmään epäonnistuu, tulee ilmoitus puuttuvista tiedoista	OK
TT28	Koirakeskus Doggien asiakas	Jätetään syntymävuosi-kenttä täyttämättä, täytetään muut kentät	Lisäys järjestelmään epäonnistuu, tulee ilmoitus puuttuvista tiedoista	OK

1.8 Poista ajanvaraus

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT29	Koirakeskus Doggien asiakas	Klikataan Poista ajanvaraus-kohtaa	Varaus poistetaan	OK

1.9 Muokkaa koiran tietoja

TT30	Koirakeskus Doggien asiakas	Muokataan mitä tahansa tietoa ja paina tallenna	Tulee ilmoitus onnistuneista muutoksista	OK
TT31	Koirakeskus Doggien asiakas	Muokataan mitä tahansa tietoa ja jätä pakollinen kenttä esim. kutsumanimi tyhjäksi	Tulee ilmoitus pakollisen tiedon muuttumisesta, eikä muutosten tallentaminen onnistu	OK

1.10 Poista koira

TT32	Koirakeskus Doggien asiakas	Painetaan Poista koiran tiedot-linkkiä	Tulee ilmoitus onnistuneesta poistosta	OK
------	-----------------------------------	--	--	----

1.11 Valitse palvelu

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT33	Koirakeskus Doggien asiakas	Valitaan yksi tai useampi palvelu ja siirrytään eteenpäin	Siirrytään ajanvaraus kohtaan	OK
TT34	Koirakeskus Doggien asiakas	Ei valita yhtään palvelua ja klikataan Seuraava-nappia	Tulee ilmoitus, ettei ole palvelua valittuna	OK

1.12 Valitse aika trimmaukseen

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT35	Koirakeskus Doggien asiakas	Klikataan vapaata aikaa	Siirrytään varauksen vahvistamiseen	OK

1.13 Vahvista varaus (trimmaus)

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT36	Koirakeskus Doggien asiakas	Klikataan Vahvista varaus-nappia	Tulee ilmoitus varauksen vahvistamisesta	OK

1.14 Valitse aika koulutuksiin

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT37	Koirakeskus Doggien asiakas	Valitse Omat tiedot -sivulta Ilmottaudu koulutuksiin -linkki. Älä valitse mitään palvelua	Tulee ilmoitus puuttuvista tiedoista	OK
TT38	Koirakeskus Doggien asiakas	Valitse vain palvelu esim. agilityn alkeet	Tulee ilmoitus puuttuvista tiedoista	OK
TT39	Koirakeskus Doggien asiakas	Valitse palvelu esim. agilityn alkeet ja ajankohta sekä koira.	Siirrytään varauksen vahvistamiseen	OK
TT40	Koirakeskus Doggien asiakas	Valitaan kaikki tarvittavat tiedot, mutta valitaan koulutusajankohta, joka on täynnä (6 mahtuu aina yhdele kurssille)	Tulee ilmoitus kurssipaikan täynnä olemisesta ja ehdotus haluaako asiakas jonotuspaikalle	OK
TT41	Koirakeskus Doggien asiakas	Painetaan, että halutaan jonotuspaikalle	Henkilö siirtyy jonotuspaikalle	OK
TT42	Koirakeskus Doggien asiakas	Painetaan sivulle ilmestyvistä linkeistä linkkiä, josta voi tarkistaa kurssiaikataulut	Aukeaa kurssiaikataulu ikkuna	OK
TT43	Koirakeskus Doggien asiakas	Painetaan sivulle ilmestyvistä linkeistä linkkiä, josta voi tarkistaa kurssikuvaukset	Aukeaa kurssikuvaus ikkuna	OK
TT44	Koirakeskus Doggien asiakas	Painetaan sivulle ilmestyvää Lisää koira-linkkiä, kun on valittu ensin Koulutukset, Palvelu, Ajankohta	Aukeaa ikkuna, jossa voi lisätä uuden koiran järjestelmään	OK

1.15 Vahvista varaus (koulutukset)

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT45	Koirakeskus Doggien asiakas	Klikataan Vahvista varaus- nappia	Tulee ilmoitus varauksen vahvistamisesta	OK

1.16 Lisää kurssi

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT46	Koirakeskus Doggien työntekijä	Klikataan Lisää kurssi -linkkiä/nappia. Täytetään kaikki kentät.	Tulee ilmoitus onnistuneesta kurssin lisäämisestä.	OK
TT47	Koirakeskus Doggien työntekijä	Klikataan Lisää kurssi -linkkiä/nappia. Jätetään kaikki kentät tyhjäksi.	Tulee ilmoitus pakollisten tietojen puuttumisesta.	OK
TT48	Koirakeskus Doggien työntekijä	Klikataan Lisää kurssi -linkkiä/nappia. Jätetään Kurssin nimi -kenttä tyhjäksi.	Tulee ilmoitus pakollisten tietojen puuttumisesta.	OK

1.17 Lisää koulutustyyppi

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT49	Koirakeskus Doggien työntekijä	Klikataan Lisää koulutustyyppi - linkkiä/nappia. Täytetään pakollinen kenttä.	Tulee ilmoitus onnistuneesta koulutustyyppin lisäämisestä.	OK
TT50	Koirakeskus Doggien työntekijä	Klikataan Lisää koulutustyyppi - linkkiä/nappia. Jätetään pakollinen kenttä tyhjäksi.	Tulee ilmoitus pakollisten tietojen puuttumisesta.	OK

1.18 Lisää ajankohta

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT51	Koirakeskus Doggien työntekijä	Klikataan Lisää ajankohta - linkkiä/nappia. Täytetään kaikki kentät.	Tulee ilmoitus onnistuneesta ajankohdan lisäämisestä.	OK
TT52	Koirakeskus Doggien työntekijä	Klikataan Lisää koulutustyyppi - linkkiä/nappia. Kaikki kentät tyhjiksi.	Tulee ilmoitus pakollisten tietojen puuttumisesta.	OK

1.19 Poista kurssi

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT53	Koirakeskus Doggien työntekijä	Klikataan Poista kurssi -linkkiä/nappia. Valitaan kurssi ja painetaan poista.	Tulee ilmoitus onnistuneesta poistosta.	OK

1.20 Poista koulutustyyppi

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT54	Koirakeskus Doggien työntekijä	Klikataan Poista koulutustyyppi - linkkiä/nappia. Valitaan koulutustyyppi ja painetaan poista.	Tulee ilmoitus onnistuneesta poistosta.	OK

1.21 Poista ajankohta

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT55	Koirakeskus Doggien työntekijä	Klikataan Poista ajankohta -linkkiä/nappia. Valitaan kurssi ja ajankohta ja painetaan poista.	Tulee ilmoitus onnistuneesta poistosta.	OK

1.22 Kurssin hallinta

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT56	Koirakeskus Doggien työntekijä	Klikkaa kaikkia kursseja vuorotellen Kurssien hallinta - sivulla.	Jokaisesta kurssista avautuu oma Kurssin hallinta -sivu.	OK
TT57	Koirakeskus Doggien työntekijä	Klikkaa mitä tahansa kurssia. Valitse kurssin ajankohta ja paina Hae -nappia.	Kurssille ilmottautuneet ja jonotuspaikalla olijat listautuvat allekkain.	OK

1.23 Poista asiakas kurssilta/jonotuspaikalta

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT58	Koirakeskus Doggien työntekijä	Klikkaa minkä tahansa kurssille ilmottautuneen nimen vieressä olevaa Poista -linkkiä.	Tulee ilmoitus onnistuneesta poistosta.	OK
TT59	Koirakeskus Doggien työntekijä	Klikkaa minkä tahansa jonotuspaikalla olevan henkilön nimen vieressä olevaa Poista -linkkiä.	Tulee ilmoitus onnistuneesta poistosta.	OK

1.24 Lisää asiakas jonotuspaikalta kurssille

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT60	Koirakeskus Doggien työntekijä	Klikkaa minkä tahansa kurssille ilmottautuneen nimen vieressä olevaa Lisää kurssille -linkkiä.	Tulee ilmoitus onnistuneesta lisäyksestä.	OK

1.25 Hae asiakasta

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT61	Koirakeskus Doggien työntekijä	Täytä kaikki haku kentät (Etunimi: Kukka, Sukunimi: Matila, Puhelinnumero: 06660666).	Haettu henkilö ilmestuu haun tuloksiin.	OK
TT62	Koirakeskus Doggien työntekijä	Jätä kaikki kentät tyhjiksi.	Tulee ilmoitus puuttuvista tiedoista.	OK
TT63	Koirakeskus Doggien työntekijä	Hae pelkällä etunimellä ”Kukka”.	Kaikki Kukka etunimiset henkilöt ilmestyvät haun tuloksiin.	OK
TT64	Koirakeskus Doggien työntekijä	Hae pelkällä sukunimellä ”Matila”.	Kaikki Matila sukunimiset henkilöt ilmestyvät haun tuloksiin.	OK
TT65	Koirakeskus Doggien työntekijä	Hae pelkällä puhelinnumerolla ” 06660666”.	Puhelinnumeroa vastaava henkilö ilmestyy haun tuloksiin.	OK

1.26 Asiakkaan tietojen tarkastelu/muokkaus

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT66	Koirakeskus Doggien työntekijä	Klikkaa asiakkaan hakemisen jälkeen asiakkaan nimestä muodostunutta linkkiä. Muuta mitä tahansa tietokenttää ja paina Tallenna muutokset -nappia.	Tulee ilmoitus onnistuneesta muutoksesta.	OK
TT67	Koirakeskus Doggien työntekijä	Klikkaa asiakkaan hakemisen jälkeen asiakkaan nimestä muodostunutta linkkiä. Jätä jokin kentistä tyhjäksi esim. puhelinnumero. Paina Tallenna muutokset -nappia.	Tulee ilmoitus puuttuvista tiedoista.	OK
TT68	Koirakeskus Doggien työntekijä	Jätä kaikki kentät tyhjäksi. Paina Tallenna muutokset -nappia.	Tulee ilmoitus puuttuvista tiedoista.	OK
TT69	Koirakeskus Doggien työntekijä	Mene Kurssin hallinta -sivulle, klikkaa jotain kurssille ilmoittautunutta henkilöä.	Siirrytään asiakkaan tiedot sivulle.	OK

1.27 Asiakkaan koiran tietojen tarkastelu/muokkaus

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT70	Koirakeskus Doggien työntekijä	Mene Asiakkaan tiedot -sivulle. Klikkaa asiakkaan koiran nimestä muodostunutta linkkiä.	Siirrytään koiran tiedot sivulle.	OK
TT71	Koirakeskus Doggien työntekijä	Muuta yksi kerrallaan jokaista koiran tieto kenttää. Paina Tallenna -nappia.	Tulee ilmoitus onnistuneesta muutoksesta.	OK
TT72	Koirakeskus Doggien työntekijä	Jätä kaikki kentät tyhjäksi. Paina Tallenna -nappia.	Tulee ilmoitus puuttuvista tiedoista.	OK

1.28 Lisää koira asiakkaalle

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT73	Koirakeskus Doggien työntekijä	Mene Asiakkaan tiedot -sivulle. Klikkaa Lisää koira -linkkiä. Täytä kaikki koiran tiedot ja paina Tallenna -nappia.	Tulee ilmoitus onnistuneesta lisäyksestä.	OK
TT74	Koirakeskus Doggien työntekijä	Jätä tyhjäksi kaikki koiran tiedot ja paina Tallenna -nappia.	Tulee ilmoitus puuttuvista tiedoista.	OK

1.29 Poista asiakas järjestelmästä

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT75	Koirakeskus Doggien työntekijä	Mene Asiakkaan tiedot -sivulle. Klikkaa Poista asiakas järjestelmästä -nappia.	Asiakas poistuu tietokannoista.	OK

1.30 Varaa asiakkaalle trimmausaika

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT76	Koirakeskus Doggien työntekijä	Mene Asiakkaan tiedot -sivulle. Klikkaa Varaa asiakkaalle trimmausaika -linkkiä/nappia.	Siirrytään ajanvaraus sivustolle.	OK

1.31 Lisää viitteitä

ID	Toimija	Tapaus	Odotettu lopputulos	Tulos
TT77	Koirakeskus Doggien työntekijä	Mene Lisää viitteitä -sivulle. Kirjoita pankista saatu viite sivulla olevaan kenttään. Paina Lisää -nappia.	Tulee ilmoitus onnistuneesta lisäyksestä.	OK
TT78	Koirakeskus Doggien työntekijä	Mene Lisää viitteitä -sivulle. Jätä kenttä tyhjäksi ja paina Lisää -nappia.	Tulee ilmoitus puuttuvista tiedoista.	OK

Liite 8. Tietokannan luokkakuvaukset

Tiedot ovat pakollisia, ellei toisin mainita.

CLIENTS Asiakkaat		
ATTRIBUUTTI/OPERAATIO	DATATYYPPI	SELITYS
ClientID	integer	Asiakkaan id, automaattisesti määritetty
FirstName	varchar	Asiakkaan etunimi
LastName	varchar	Asiakkaan sukunimi
StreetAddress	varchar	Katuosoite
PostalCode	varchar	Postinumero, 5 numeroa
PostalOffice	varchar	Postitoimipaikka
PhoneNumber	varchar	Puhelinnumero
EmailAddress	varchar	Sähköpostiosoite, toimii käyttäjätunnuksena
Note	longvarchar	Ylläpitäjän kommentit asiakkaasta Ei pakollinen Ei näy asiakkaalle
Password	varchar	Salasana, MD5-hashattu

ADMINS Ylläpitäjät		
ATTRIBUUTTI/OPERAATIO	DATATYYPPI	SELITYS
AdminID	integer	Ylläpitäjän id, automaattisesti määritetty
EmailAddress	varchar	Sähköpostiosoite, toimii käyttäjätunnuksena
Password	string	Salasana, MD5-hashattu

DOGS Koirat		
ATTRIBUUTTI/OPERAATIO	DATATYYPPI	SELITYS
DogID	integer	Koiran id, automaattisesti määritetty
Name	varchar	Koiran kutsumanimi
BreedID	integer	Rodun id, taulusta BREEDS
ClientNote	longvarchar	Asiakkaan kommentit koirasta Ei pakollinen
AdminNote	longvarchar	Ylläpitäjän kommentit koirasta Ei pakollinen Ei näy asiakkaalle
OwnerID	integer	Koiran omistajan id, taulusta CLIENTS
Gender	boolean	Koiran sukupuoli 1 – uros 0 – narttu
BirthDate	Date	Syntymäpäivämäärä

BREEDS Rodut		
ATTRIBUUTTI/OPERAATIO	DATATYYPI	SELITYS
BreedID	integer	Rodun id, automaattisesti määritetty
Name	varchar	Rodun nimi, esim 'Villakoira'

SERVICES Palvelut		
ATTRIBUUTTI/OPERAATIO	DATATYYPI	SELITYS
ServiceID	integer	Palvelun id, automaattisesti määritetty
Description	varchar	Palvelun kuvaus, esim. 'Pesu'
DefaultDuration	Integer	Palvelun oletuspituus minuutteina. Tätä käytetään, mikäli rodulle ei ole määritetty palvelun pituutta SERVICE_BREED-taulussa

SERVICE_BREED Rodulle määritetyt palvelut		
ATTRIBUUTTI/OPERAATIO	DATATYYPI	SELITYS
BreedID	integer	Rodun id, taulusta BREEDS
ServiceID	integer	Palvelun id, taulusta SERVICES
Duration	integer	Palvelun pituus minuutteina

RESERVATIONS Varaukset		
ATTRIBUUTTI/OPERAATIO	DATATYYPPI	SELITYS
ReservationID	integer	Varauksen id, automaattisesti määritetty
StartTime	timestamp	Varauksen alkamispäivämäärä ja -aika
EndTime	timestamp	Varauksen päättymispäivämäärä ja -aika
OrderTime	timestamp	Varauspäivämäärä ja -aika
DogID	integer	Koiran id, taulusta DOGS

RESERVATION_SERVICE Varaukseen valitut palvelut		
ATTRIBUUTTI/OPERAATIO	DATATYYPPI	SELITYS
ReservationID	integer	Varauksen id, taulusta RESERVATIONS
ServiceID	integer	Palvelun id, taulusta SERVICES