

Helsingin Uudenkaupungin kiltä ry:n jäsenpalvelut ja kotisivut

Tuomas Kujala

Opinnäytetyö
Tietojenkäsittelyn koulutusohjelma
2013



Tekijä tai tekijät Tuomas Kujala	Aloitusvuosi 2011
Raportin nimi Helsingin Uudenkaupungin kilta ry:n jäsenpalvelut ja kotisivut	Sivu- ja liitesivumäärä 33+42
Opettajat tai ohjaajat Juhani Välimäki	
<p>Helsingin Uudenkaupungin kilta ry halusi kehittää toimintaansa hankkimalla web-sovelluksen, joka toimisi yhdistyksen kotisivuna ja johon sisältyisi yhdistyksen toimintaa helpottavia toimintoja, kuten tapahtumista ilmoittaminen ja niihin ilmoittautuminen. Työn konkreettisia tuloksia ovat vaatimusmääritysdokumentti, käytettävyyss- ja käyttöliittymäraportti sekä sovelluksen ohjelmakoodi.</p> <p>Tämän opinnäytetyön tarkoitus oli määritellä ja toteuttaa toimeksiantajan vaatimuksia vastaava web-sovellus, joka olisi sovellusarkkitehtuuriltaan mahdollisimman kevyt ja tukisi jatkokehitystä.</p> <p>Sovellus toteutettiin PHP-komentosarjakielellä ja tietokantana käytettiin MySQL:a. Sovelluksen web-sivujen toteutuksessa käytettiin HTML5:ä, JavaScriptiä ja jQueryä. Sovelluksen käyttöliittymien suunnittelussa seurattiin teoriaosassa mainittujen tutkimusten ohjeistusta joka ottaa huomioon vanhemman väestön edustajat.</p> <p>Opinnäytetyön tuloksena syntyneen web-sovelluksen tärkeimmät ominaisuudet ovat yhdistyksen tapahtumien organisointiin liittyvät toiminnallisuudet. Toimeksiantaja oli erittäin tyytyväinen työn tuloksiin.</p>	
Asiasanat olio-ohjelmointi, web-sovellus, käytettävyys	



HAAGA-HELIA

University of Applied Sciences

Degree Programme in Information Technology

Abstract

Authors Tuomas Kujala	Year of entry 2011
The title of thesis The member services and website for Helsingin Uudenkaupungin kilta –association	Number of report pages and attachment pages 33+42
Advisor(s) Juhani Välimäki	
<p>Helsingin Uudenkaupungin kilta is an association which wanted to improve their activity by acquisition of their own web service. The web service should serve as the association's home page and have functionalities which will automatize the association's data processing operations, such as organizing events. In addition, requirement analysis and partial implementation of the web service were included in this thesis.</p> <p>The purpose of this thesis was to produce a web service which will meet the association's current requirements and provide an easily expandable software architecture for further development.</p> <p>The service was programmed using the PHP server-side scripting language and a MySQL Database. HTML5, JavaScript and jQuery were used on the client side programming.</p> <p>The tangible results of this thesis were, for example, requirement analysis document, usability report and the source code of the implemented service. The guidelines concerning the possible usability issues for the target group were taken into notice during the user interface design of the service.</p>	
Key words Object oriented programming, web service, usability	

Sisällys

Sanasto.....	iii
1 Johdanto	1
1.1 Yhdistysten tietotarpeet.....	1
1.2 Helsingin Uudenkaupungin kiltä ry:n kehitysprojekti	2
1.2.1 Kehitysprojektin tavoitteet.....	3
2 Teoriatausta.....	4
2.1 Web-sovelluksen ohjelmointi PHP:lla	4
2.1.1 PHP:n historia.....	4
2.1.2 PHP:n ajaminen ja käyttö	4
2.1.3 Olio-ohjelmointi PHP:lla.....	5
2.1.4 Luokat	5
2.1.5 Luokkailmentymät.....	6
2.1.6 PDO – PHP Data Objects.....	7
2.2 HTML5.....	7
2.2.1 HTML-dokumentit	7
2.2.2 HTML5:n uudet elementit	8
2.3 Käytettävyys ja käyttöliittymäsuunnittelu kohderyhmälle, jossa on myös vanhemman väestön edustajia	9
2.3.1 WAI-AGE-projekti	9
2.3.2 Hyväksi havaitut käytännöt.....	10
2.3.3 WCAG – web-sovellusten helppokäyttöisyysohjeistus	11
3 Työn suunnittelu	12
3.1 Projektinhallinta.....	12
3.2 Aikataulutus.....	12
3.3 Työkalut.....	12
4 Vaatimusmäärittäminen	13
4.1 Yleiskuvaus.....	13
4.2 Käyttötapaukset.....	13
4.3 Säilytettävät tiedot	14
4.4 Käyttöoikeudet	14

4.5	Rajaus	14
5	Palvelun tekninen suunnittelu ja toteutus	15
5.1	Kehitysympäristö.....	15
5.2	Tietokanta.....	15
5.3	Sovellusarkkitehtuuri.....	15
5.3.1	Kohdeluokat.....	15
5.3.2	Ohjausluokat	18
5.3.3	Apuluokat	18
5.3.4	Näkymät.....	19
5.3.5	Pääsivun näkymien vaihto jQuerylla	21
5.4	Käyttöliittymät	21
5.5	Testaus	24
6	Tulokset ja johtopäätökset	26
7	Yhteenveto	28
7.1	Jatkokehitys	28
	Lähteet	30
	Liitteet.....	33
	Liite 1. Vaatimusmäärittämisdokumentti.....	1
	Liite 3. Tapahtumaan ilmoittautumisen lähdekoodi	25
	Liite 4. Arkkitehtuurikuvaus	31
	Liite 5. Käyttöönoton ohjeistus.....	32
	Liite 6. Tietokannan luontilauseet	34

Sanasto

Käsite	Selite
htdocs	Tiedostopolku, josta Apache-serveri etsii verkkosivuston tiedostoja. Htdocs-kansioon tai sen alikansioon sijoitetulla .htdocs-piilotiedostolla voi hallinnoida esimerkiksi salasanoja ja käyttäjän uudelleenohjauksia.
iframe	HTML:n elementti, joka mahdollistaa web-sivun osan upottamisen toiseen web-sivuun
XAMPP	Kehitysympäristö, joka sisältää mm. Apachen, PHP-tuen ja MySQL:n
Apache	Web-serveri
Config.php	PHP-tiedosto, joka sisältää sovelluksen asetukset
MySQL	Relaatiotietokantatuote, Structured Query Language
DOM	Document-Object-Model on HTML-dokumentin rakennetta kuvaava rajapinta, jonka avulla dokumenttien sisältöä voidaan muokata esimerkiksi JavaScriptillä.

1 Johdanto

Yhdistysten hallinnointiin liittyy usein tehtäviä, joissa joudutaan käyttämään pieniä tietovarastoja yhdistyksen toimintoja varten. Erilaisten nimilistojen, yhteystietojen ja jäsenmaksutilanteen ylläpitäminen on välttämätöntä aktiivisissa yhdistyksissä.

Koska toimintojen sähköistämiseen on lähdetty Suomen yritysmaailmassakin melko verkkaisesti, on turvallista olettaa, että yhdistyksien osalta sähköistämistä on tapahtunut hyvin vähän, ja perinteinen kynä ja paperi-järjestelmä on hallitseva. Kynnys oman verkkopalvelun hankkimiseen yhdistykselle voi olla taloudellisista syistä tai epätietoisuuden vuoksi korkea. Oman verkkopalvelun tuomia etuja, kuten näkyvyyttä omalle toiminnalle, ei välttämättä ymmärretä.

Ei kuitenkaan voida sanoa, että oma palvelu olisi tarpeellinen ratkaisu kaikille. Sosiaalisen median työkalut, kuten Facebookin ryhmä- ja tapahtumatoiminnot, voivat oikein käytettynä helpottaa tietojenkäsittelyä yhdistyksissä huomattavasti. Yhdistyksille suunnatut palvelut, kuten Yhdistysavain.fi ja saman yrityksen Yhdistystieto.fi, tarjoavat mahdollisuuden kotisivujen pitämiseen ja sähköisen jäsenrekisterin ylläpitämiseen.

1.1 Yhdistysten tietotarpeet

Yhdistyslaki vaatii, että kaikki yhdistykset pitävät luetteloa jäsenistään. Luetteloon jäsenistä on merkittävä vähintään koko nimi ja kotipaikka. Näiden tietojen lisäksi käytännön syistä jäsenistä säilytettäviin tietoihin kuuluvat usein osoite, puhelinnumero ja sähköpostiosoite. Yhdistystieto.fi tarjoaa työkalut jäsenluettelon ylläpitämiseen. Palvelun kautta on mahdollista seurata myös yhdistyksen jäsenmaksuja.

Kaikille yhdistyksille yhteisten tietojen säilyttämiseen ja käsittelyyn on siis olemassa palvelu, jota hyödyntää, ja monille yhdistyksille kyseinen palvelu onkin epäilemättä sopiva ratkaisu. Mikäli tarvetta ei ole muulle, kuin jäsenrekisterin hallinnalle, keskustelupals-

talle ja ilmoitusten lähettämiseksi, on tuskin perusteltua lähteä hankkimaan omaa sivustoa. Valmiiksi olemassa olevaa, kustannustehokasta ratkaisua on turha sivuuttaa, elleivät yhdistyksen tietojenkäsittelytarpeet sitä vaadi.

Entä jos yhdistyksen tavoitteet oman palvelun suhteen ovat kuitenkin pidemmällä kuin tämä palvelu yksissään mahdollistaa? Esimerkiksi tapahtumien organisointiin palvelu ei tarjoa erillistä työkalua. Jos tavoitteet ovat laajemmat kuin valmiin palvelun toiminnallisuudet tarjoavat, on tarpeen kartoittaa muita mahdollisuuksia.

1.2 Helsingin Uudenkaupungin kilta ry:n kehitysprojekti

Helsingin Uudenkaupungin kilta on yhdistys, joka haluaa saada oman verkkopalvelun. Olemassaolevien palvelujen tarjonta ei vastaa yhdistyksen tarpeita, joten oman kehitysprojektin käynnistäminen on perusteltua.

Tarvittavan palvelun ensisijainen funktio olisi saada yhdistystä varten työkalu, joka helpottaa tapahtumien organisointia. Tapahtumien järjestämiseen liittyvä tietojen ylläpitäminen on tähän asti tapahtunut käsin joko paperille tai Excel-tiedostoon. Toiminta on epäkäytännöllistä niin tapahtumaa organisoivien henkilöiden kuin jäsenienkin kannalta. Yhdistys tarvitsee palveluun myös ominaisuuksia, jotka edistäisivät yhdistyksen jäsenien välistä toimintaa. Tällaisia voivat olla erilaiset sisällöntuotantotavat, kuten kirjoitusten kirjoittaminen, pienten talkoiden järjestäminen ja ilmoittaminen ja vastaavat pienimuotoiset toiminnot yhdistyksen niin sanotun ”naapurit tutuksi”-vuorovaikutuksen edistämiseksi.

Kehitysprojektissa halutaan tehdä määrittely ja toteutus palvelun ominaisuuksille. Yhdistys tahtoo projektissa tehtävän myös käytettävyysraportin palvelusta, sillä osa yhdistyksen jäsenistä kuuluu vanhempaan väestöön ja se on otettava huomioon palvelun käyttöliittymäsuunnittelussa.

1.2.1 Kehitysprojehtin tavoitteet

Kehitysprojehtin tavoite on määritellä ja kohderyhmän eri osat huomioon ottaen toteuttaa web-sovellus, jossa yhdistyksen jäsenet voivat ilmoittautua yhdistyksen tapahtumiin. Kohderyhmän huomiointi todennetaan käyttöliittymäraportilla, jossa viitataan aiheesta aikaisemmin tehtyihin tutkimuksiin ja osoitetaan lopputuloksesta ominaisuuksia, joissa tutkimusten tulokset on huomioitu.

2 Teoriatausta

Teoriaosassa käsitellään sovelluksen kehitykseen käytettyjä menetelmiä ja tekniikoita.

2.1 Web-sovelluksen ohjelmointi PHP:lla

PHP on avoimeen lähdekoodiin perustuva komentosarjakieli (skriptikieli) (The PHP Group, 2013a). Se on suosituin kieli web-sovelluskehityksessä. Suosiota selittää erityisesti kielen helppokäyttöisyys, joka kuvaakin hyvin PHP-projektin onnistumista; PHP:n nykyinen versio kehitettiin yksinkertaistamaan web-kehitystä. (Lavin, P. 2006, 3-4.)

2.1.1 PHP:n historia

PHP sai alkunsa vuonna 1994, kun Rasmus Lerdorf käytti C-kielellä kirjoitettuja skriptejä sähköisen ansioluettelonsa kävijöiden seuraamiseen. Alun perin skriptit kulkiivat nimellä ”Personal Home Page Tools”. Muutaman vuoden sisällä Lerdorf kirjoitti skriptit useita kertoja uudelleen, ja niistä tuli rajapinta tietokantaa käyttäviin dynaamisiin sovelluksiin. (The PHP Group, 2013b.)

PHP 3.0 oli kielen kehityksen kannalta käännteentekevä. Lerdorf oli tehnyt kehitystä vuoteen 1997 yksin, mutta silloin kolme israelilaista opiskelijaa ottivat häneen yhteyttä ja keskustelivat kehitysmahdollisuuksista. Tuloksena syntyi jo hyvin paljon nykyistä syntaksia muistuttava versio 3.0, joka sai myös nykyisen lyhenteen mukaisen nimen ”Hypertext Preprocessor”. (The PHP Group, 2013b.)

2.1.2 PHP:n ajaminen ja käyttö

PHP-koodin tulkitseminen ja ajaminen tapahtuu sovelluksen palvelinpuolella ja tulos lähetetään vastaanottajalle HTML-muodossa. Yleensä vastaanottaja on Internet-selain. PHP:ta voi käyttää myös ilman serveriä, ainoastaan PHP-tulkkia hyväksikäyttäen. Tällainen käyttö on tyypillistä ajoitettujen tehtävien asettamisessa esimerkiksi cron-ajastuspalvelua käytettäessä. PHP:ta voi käyttää myös työpöytäsovelluksissa, mutta se on harvinaista. (The PHP Group, 2013c)

Yleisin käytätapa on PHP-koodin HTML:n upottaminen. Upottaminen mahdollistaa dynaamisen sisällön ylläpitämisen vähällä työllä. Esimerkiksi sivuston navigaatioelementti voidaan lisätä joka sivulle vain yhdellä koodirivillä, sen sijaan että sen koko koodin joutuisi kirjoittamaan joka sivulle uudelleen (Lavin, P. 2006, 2).

```
<body>  
    <?php include('includes/admin_nav.php');?>  
</body>
```

Kuva 1. PHP-koodia HTML:n upotettuna.

2.1.3 Olio-ohjelmointi PHP:lla

Koska PHP on komentosarjakieli eikä ohjelmointikieli, sen soveltuvuus olio-ohjelmointiin on aiemmin ollut kyseenalainen. Argumentteina sitä vastaan ovat esimerkiksi kysymys olio-ohjelmoinnin tarpeellisuudesta pienissä projekteissa (joita PHP:lla usein toteutetaan), ja epäily PHP:n muuttumisesta liian monimutkaiseksi tarkoitukseensa, eli helppoon web-sovelluskehitykseen. Kuitenkin, kuten HTML5:nkin tapauksessa, kielen kehittäjien täytyy työssään ottaa huomioon alan yleiset kehityslinjat sekä trendit ja toteuttaa niiden tarpeita vastaavia työkaluja. (Lavin, P. 2006, 2-4.)

Siten PHP5:ssa julkaistiin ominaisuuksia, jotka tukivat luokkapohjaisten sovellusten rakentamista. Uudet näkyvyysmääritteet ja muodostin (Lavin, P. 2006, 26) toivat PHP:ta lähemmäksi yleisemmin käytettyjä olio-ohjelmointikieliä ja tekivät luokkien muodostamisesta sekä käyttämisestä olio-ohjelmoinnin käytäntöjä vastaavat.

2.1.4 Luokat

Luokat (class) ovat tietotyyppejä, jotka luokan ominaisuuksien (attribuuttien) ja funktioiden kanssa muodostavat mallin luokan ilmentymille, eli olioille (Rouse, P. 2005).

PHP5:ssä luokan attribuutteja määritellään tapaan *näkyvyysmääre \$attribuutti*. Näkyvyysmääreitä (access modifiers) on kolme: julkinen (public), yksityinen (private) ja suojeltu (protected). Julkiset attribuutit ovat käytettävissä kaikkialta, yksityiset vain luokan sisältä ja suojellut vain luokan ali- ja ylliluokista (The PHP Group, 2013e). Hyvään ohjelmoin-

titapaan kuuluu yleensä määrittää niin paljon yksityiseksi kuin mahdollista. Ennen versiota 5.0 näkyvyysmääreitä ei ollut käytössä, vaan sen tilalla oli esimerkiksi JavaScriptissä käytetty *var* (Lavin, P. 2006, 26).

```
private $name;
```

Kuva 2. Yksityinen attribuutti.

Funktiot määritellään samalla periaatteella. Yksityisiksi määritellyille attribuuteille täytyy luoda julkiset funktiot joilla niiden arvoja voi palauttaa (Lavin, P. 2006, 27).

```
public function getName(){  
    return $this->name;  
}
```

Kuva 3. Julkinen funktio.

2.1.5 Luokkailmentymät

Luokkailmentymien luominen PHP:ssä on hyvin samantapaista kuin muissakin kielissä. Ensin käytettävä luokka täytyy tuoda tiedostoon *require* tai *require_once*-komennolla, jonka jälkeen ilmentymä voidaan muodostaa käyttämällä komentoa *new luokan_nimi()*. (Lavin, P. 2006, 22.)

```
require_once('utility-class/sql.class.php');  
$this->connection = new sql();
```

Kuva 4. Luokkailmentymän luominen.

Jotta tarvittavia luokkia ei jouduta tuomaan erikseen jokaiseen niitä käyttävään tiedostoon, käytetään usein config-tiedostoa, jossa ladataan valmiiksi kaikki sovelluksessa jatkuvasti käytössä olevat luokat. Config-tiedostossa säilytetään myös muita, usein muuttumattomia tietoja, kuten sovelluksen URI-osoitteita.

2.1.6 PDO – PHP Data Objects

PDO on PHP:n tietokantarajapinta. Rajapinnan kaikkia funktioita voi soveltaa usean eri tietokantatuotteen kanssa työskentelyyn. (The PHP Group, 2013d).

2.2 HTML5

Web-dokumentteja on luotu eri merkintätavoin (HTML 4.01, XHTML 1.1) ja niitä luetaan monilla erilaisilla selaimilla. Eri selaimet käsittelevät DOM-rakennetta eri tavalla, ja siten sama dokumentti voi näyttää erilaiselta tai toimia virheellisesti eri selaimissa.

Tietty syntaksi web-dokumenttien luomisessa voi siis olla jonkin selaimen tulkinnan mukaan väärin, ja jonkin toisen mukaan oikein.

HTML5 on uusi versio web-kehityksen tärkeimmästä kielestä. Se tulee korvaamaan edellisen version suositeltuna standardina. Aikaisemmat standardit ovat olleet HTML 4.01, XHTML 1.0 ja XHTML 1.1. (W3C, 2013a; Wikipedia 2013.)

Internetin sisältö on kahden viime vuosikymmenen aikana muuttunut staattisista sivuista jatkuvasti dynaamisempiin sivustoihin. Yhä useampaa sivustoa kuvaa paremmin termi verkkosovellus tai applikaatio. Rakenteiden ja käyttöliittymien kehittyessä myös vaatimukset kehitystyökaluja kohtaan kasvavat. Koska alalla toimivien sisällöntuottajien, tuotantoalustojen ja kohdelaitteiden määrä kasvaa jatkuvasti, tarvitaan kaikille yhteisiä standardeja. HTML5 pyrkii vastaamaan kasvaviin vaatimuksiin ja olemaan laiteriippumaton, avoin ja vapaakäyttöinen standardi. (Lawson, B., Sharp, R. 2011, 13.)

2.2.1 HTML-dokumentit

HTML-dokumentit koostuvat HTML-elementeistä. Elementit merkitään `<>`-hakasten sisään. Useimmilla elementeillä on avausmerkintä ja lopetusmerkintä; sisällöttömillä elementeillä, kuten *input*, saa olla ainoastaan aloitusmerkintä. Lopetusmerkinnässä elementin nimen eteen merkitään oikeallepäin kääntynyt kauttaviiva (W3C, 2013b). Suositeltavan muotoinen HTML-dokumentti koostuu seuraavanlaisesti:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset=UTF-8>
    <title>Insert title here</title>
  </head>
  <body>

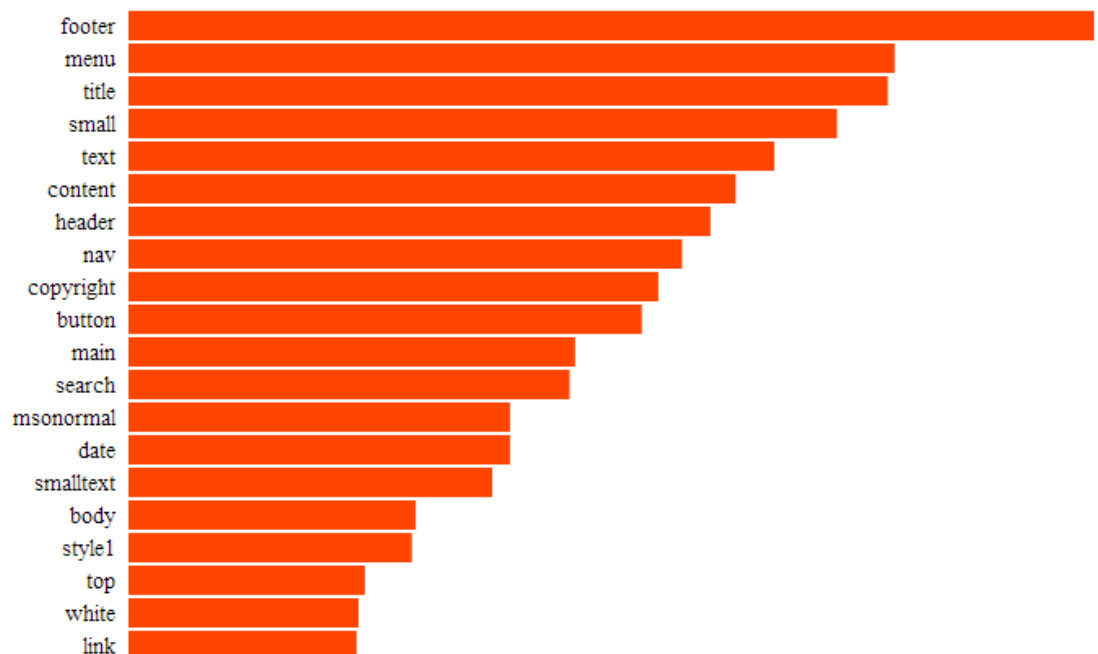
  </body>
</html>

```

Kuva 5. HTML5-dokumentin runko. (Lawson, B., Sharp, R. 2011, 8-15.)

2.2.2 HTML5:n uudet elementit

HTML-elementtien luokkanimiä (class) kuten header, footer, nav ja content on käytetty aiemmin kuvaamaan elementtien sisältöä tai käyttötarkoitusta dokumentissa. HTML5:n kehittäjät tahtovat, että elementtien nimet itsessään kertovat, millaista sisältöä elementti sisältää, ilman että sovelluskehittäjien tarvitsee määritellä niille kuvaavaa nimeä. (Lawson, B., Sharp, R. 2011, 15-18.) Kuvassa 6 ja taulukossa 1 esitetään, kuinka syntaksiin on johdettu elementtejä yleisistä luokkanimistä.



Kuva 6. Eniten käytetyt luokkien nimet web-sivuilla (Google Web Authoring Statistics, 2005)

Taulukko 1. Suositut luokannimet ja toteutuneet HTML5-elementit
(Google Web Authoring Statistics, 2005)

Suosittu luokka	HTML5-elementti
footer	footer
menu	menu
title, header, top	header
small, smalltext	small
text, content, main, body	article
nav	nav

2.3 Käytettävyys ja käyttöliittymäsuunnittelu kohderyhmälle, jossa on myös vanhemman väestön edustajia

Kun suunnitellaan uutta sovellusta, jonka tulevaan käyttäjäryhmään kuuluu vanhemman väestön edustajia, sovelluksen käyttöliittymäsuunnittelussa on otettava huomioon tämän mahdollisesti asettamia vaatimuksia. Niitä on tutkittu käyttäjätesteillä ja ohjeistus verkkopalveluiden helppokäyttöisyyden edistämiseksi on olemassa.

2.3.1 WAI-AGE-projekti

Web Accessibility Initiativen projektissa ”Ageing Education and Harmonisation” tutkittiin laajasti web-sovellusten käytettävyyttä vanhempien käyttäjäryhmien keskuudessa. Projektin tavoite oli tutkia ja saada lisää näkyvyyttä käyttöliittymäsuunnittelulle, jossa otetaan huomioon käyttäjäryhmät joille tavallisten käyttöliittymien käyttäminen on haastavaa. (W3C, 2012.)

Projektissa havaittiin, että jo aikaisemmissa tutkimuksissa oli tunnistettu valtaosa vanhempien käyttäjien käyttökokemukseen vaikuttavista seikoista. Vaikka ne ovat tiedossa, suunnittelijat ottavat niitä harvoin huomioon. (Arch, A., Abou-Zahra, S., Lawton-Henry, S. 2008.)

Kaksi merkittävää tekijää erottaa vanhemmat käyttäjät muista käyttäjistä: heillä on todennäköisesti vähemmän kokemusta tietokoneiden ja Internetin käytöstä, ja heidän käyttökokemukseensa saattaa vaikuttaa vanhenemisprosessin mukana tapahtuvat kognitiivisten ja fyysisten toimintojen heikentyminen. Noin puolella yli 60-vuotiaista on jokin kognitiivista toimintaa häiritsevä haitta tai invaliditeetti. (Newell, A., Dickinson, A., Smith, M., Gregor, P. 2006, 4; Arch 2009, 2.) Yleisiä muutoksia ovat aisti- ja muistihäiriöt, jotka vaikuttavat käyttäjän kykyyn käsitellä ja hyödyntää esitettyä tietoa. (Arch 2009, 1-2.) Siten vanhemmilla käyttäjillä on usein vaikeuksia oppia käyttämään uutta palvelua tai sovellusta (Newell ym. 2006).

2.3.2 Hyväksi havaitut käytännöt

WAI-AGE:n lopputuloksena syntyi useita julkaisuja, joissa käyttäjätutkimusten avulla pystyttiin määrittämään yksityiskohtaisia ohjeita käyttöliittymäsuunnittelun tueksi. Erään sähköpostiohjelmiston vanhoilla käyttäjillä suoritettiin käyttäjätutkimus, jossa haettiin esimerkiksi seuraavaa (Newell ym. 2006, 9):

- sovelluksessa oli liikaa toiminnallisuuksia
- sovelluksessa käytettiin liikaa teknologisia ilmaisuja
- yhteen ikkunaan oli asetettu liian paljon sisältöä

Vanhempi väestö käyttäjäryhmänä on monimuotoinen, ja kun puhutaan käytettävyyteen mahdollisesti liittyvistä haittatekijöistä, täytyy ottaa huomioon että ne koskettavat yleensä vain käyttäjäryhmän osaa eikä siis koko ryhmää. Tutkimustulosten ohessa todetaankin, että tämän monimuotoisuuden vuoksi tämän ja vastaavan tutkimuksen tuloksia tulisi tulkita enemmän suuntaa-antavina kuin suorina ohjeina käyttöliittymäsuunnittelussa (Newell ym. 2006, 9).

Useissa muissa tutkimuksissa on havaittu samoja ja samankaltaisia asioita kuin edellä mainitussa. Yksi WAI-AGE:n tutkijoista, Andrew Arch, esittää kirjoituksissaan havaintoja tukevia väittämiä. Ikääntyneelle kohderyhmälle suunnittelussa olennaisia ovat selkeät rakenteet ja käyttöliittymien helppo opittavuus sekä rakenteiden yhdenmukaisuus

palvelun eri osien välillä. (Arch 2009, 3-4.) Yhtä näyttöä tulisi käyttää vain yhden toiminnallisuuden esittämiseksi ja vain toiminnallisuuden kannalta välttämättömät tiedot kannattaa esittää (Newell ym. 2006, 9). Kompleksiset käyttöliittymät ja liian suuren tietomäärän esittäminen yhdellä sivulla heikentävät käyttökokemusta. (Arch 2009, 2.)

Niin aiemmin mainitussa käyttäjätutkimuksessa kuin Archin teksteissä mainitaan, että hyvä keino parantaa ikääntyneiden käyttäjien käyttökokemusta on hyödyntää normaalia suurempia fontteja ja kuvakkeita. Niiden on havaittu parantavan näkö- ja motoriikka-häiriöistä kärsivien käyttäjien käyttökokemusta. (Newell ym. 2006, 10; Arch 2009, 3.)

2.3.3 WCAG – web-sovellusten helppokäyttöisyysohjeistus

Web Content Accessibility Guidelines on kokoelma suosituksia, joiden avulla web-sivujen käytettävyyttä voi parantaa niin yleisesti, kuin myös käyttöä haittaavista tekijöistä kärsivien käyttäjien osalta (W3C, 2008). Artikkelissa ”Improving WCAG for Eldery Web Accessibility” (Lara, S., Watanabe, W., dos Santos, E., Fortes, R. 2010) on tutkittu ja paranneltu WCAG:n ohjeistusta erityisesti vanhempia käyttäjiä ajatellen. Yksityiskoh-
taisia ohjeita on kymmenittäin, ja alla on listattu joitain niistä kategorioittain.

Rakenteen ymmärrettävyys (Lara, S. ym. 2010, 5):

- Sivun sisältö kannattaa sijoittaa keskelle ruutua, sillä vanhemmat käyttäjät keskittyvät usein eniten ruudun keskiosaan.
- Sivun eri elementtien täytyy erottua toisistaan riittävän selvästi.

Käytettävyys (Lara, S. ym. 2010, 5):

- Vierityspalkkien käyttöä tulee välttää, koska vanhemmat käyttäjät eivät ole aina selvillä niiden käyttötarkoituksesta.
- Sivuston täytyy eritellä linkit joissa on vierailtu linkeistä, joissa ei ole käyty. Erittelyn tekemättä jättäminen saattaa aiheuttaa vaikeuksia heikentyneestä muistista kärsiville.

Sisällön ymmärrettävyys (Lara, S. ym. 2010, 6):

- Yhdellä sivulla kannattaa esittää vain välttämättömät tiedot.
- Käyttäjäsyötteen avustaminen erilaisin interaktiivisin toiminnoin, kuten esimerkiksi tekstikentän automaattitäyttämisen, kun mahdollista, on suositeltavaa.

3 Työn suunnittelu

3.1 Projektinhallinta

Projektihallinnassa käytetään ketterää mallia huomioon ottaen, että kyseessä on yhden työntekijän kehitystyö. Sen vuoksi katsotaan, ettei projektissa tarvitse ylläpitää yksityiskohtaista product backlogia, koska työntekijä on itse selvillä työn vaiheesta ja kesken-eräisistä tehtävistä. Työn katselmoinnit tehdään projektin ohjauskokousten yhteydessä.

Työ jaetaan sprintteihin. Kullekin sprintille nimetään toiminnallisuus tai tehtävä, jonka valmistuttua sprintin tavoite täyttyy. Sprinteille määrätään suuntaa-antava toteutusjärjestys. Jokaiseen sprinttiin kuuluu sprintille nimetyn toiminnallisuuden vaatimusmäärittäminen.

3.2 Aikataulutus

Lähtökohtaisesti sprinttejä on kahdeksan ja ne jaetaan kahden kuukauden ajalle. Kuten sprinttien toteutusjärjestys, myös niiden kesto määritellään joustavasti. Seuraavaan sprinttiin siirrytään, kun meneillään olevalle nimetty toiminnallisuus on valmis.

3.3 Työkalut

Asiakas ei määrittele tekniikoita palvelun toteuttamiseen. Työntekijän aiemman kokemuksen perusteella palvelun kehityskieleksi valitaan PHP ja tietokannaksi MySQL-tietokanta. Kehitysympäristönä käytetään kevyttä XAMPP Apache-palvelinta ja Eclipsen Kepler-versiota. Työssä käytetään HAAGA-HELIAn laitteistoa.

4 Vaatimusmäärittäminen

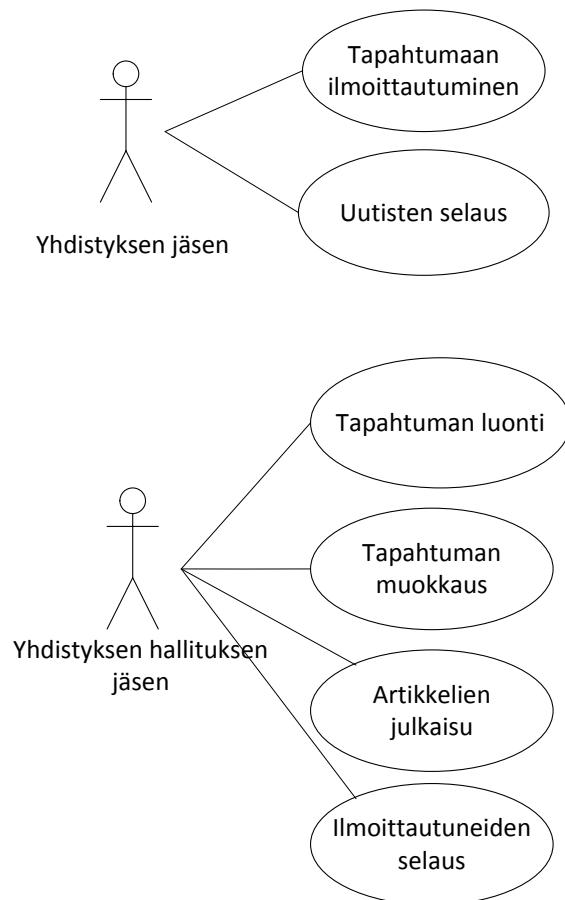
Toteutettavan palvelun toimintoja ovat Helsingin Uudenkaupungin kiltä ry:n toiminnan esittely, tapahtumien organisoimiseen liittyvien tietojenkäsittelytehtävien suorittaminen ja jäsenille tarkoitettu sisällönlisäystoiminto. Palvelun käyttäjiä ovat yhdistyksen jäsenet ja yhdistyksen toiminnasta kiinnostuneet henkilöt.

4.1 Yleiskuvaus

Projektissa toteutetaan palvelu Helsingin Uudenkaupungin kiltä ry:n kotisivuksi. Palvelun tarkoitus lisätä yhdistyksen näkyvyyttä, helpottaa tapahtumanjärjestämistä niin hallituksen kuin muidenkin jäsenten osalta ja lisätä jäsenten välistä kanssakäymistä.

4.2 Käyttötapaukset

Käyttötapauskuvaukset ovat vaatimusmäärittäyskuvastossa (liite 1).



Kuva 7. Käyttötapauskaavio.

4.3 Säilytettävät tiedot

Sovelluksessa säilytetään tietoa yhdistyksen tapahtumista ja niiden osallistujista, sekä yhdistyksen esittelytekstiä ja uutisia. Tietokantakaavio ja luokkakuvaukset ovat vaatimusmäärityskuvastossa (liite 1).

4.4 Käyttöoikeudet

Palveluun ei tehdä käyttäjäkohtaista kirjautumista, vaan palvelun hallintasivut suojataan htdocs-kirjautumisella. Htdocs on Apachen kansio, josta serveri lähtökohtaisesti etsii verkkosivuja näytettäväksi. Kansiossa ja sen alikansioissa voi käyttää piilotettua .htdocs-tiedostoa, johon voi määrittää salasanasuojauksen tiedot eli käyttäjänimen ja salasanan. Salasanasuojausta käytetään ainoastaan hallintasivuilla. Muut sivut ovat julkisia.

4.5 Rajaus

Jäsenille suunnatun sisällön luonti- ja jakotoiminnallisuudet sekä hallintapuolen kuvanvaihtotoiminnallisuus ovat rajattu toteutuksen ulkopuolelle. Kuvan vaihtamisen käyttötapauskuvaus on kuitenkin tehty määritysdokumentissa.

5 Palvelun tekninen suunnittelu ja toteutus

5.1 Kehitysympäristö

Suurin osa kehitystyöstä tehtiin HAAGA-HELIAn eri työasemilla. Kehitysympäristönä käytettiin XAMPP:a, joka sisältää Apache-serverin, PHP:n ja MySQL:n. XAMPP:sta asennettiin Portable-versio massamuistille, jolloin kehitysympäristö ja tietokanta eivät rajoitu yhdelle työasemalle. XAMPP:n hakemistosta oli ajettava `setup_xampp`-tiedosto, joka suorittaa kehitysympäristölle tarvittavien asetusten muokkaamisen. Kehitysympäristö sisältää Apache-palvelimen ja MySQL-tietokannan.

5.2 Tietokanta

Työ alkoi tietokannan ja testidatan luomisella. Kyselyt suoritettiin kehitysympäristössä valmiina olevan phpMyAdmin-ohjelman kautta. Luontilauseet kannan tauluille ja testidatalle ovat liitteessä 6.

5.3 Sovellusarkkitehtuuri

Sovellusarkkitehtuurilta vaaditaan, että se on yksinkertainen ja jatkoa ajatellen mahdollisimman vaivaton muokata. Prosessuaalinen sovellusrakenne vaikeuttaisi tähän tavoitteeseen pääsemistä. Siksi kehitystä lähdettiin tekemään oliopohjaisesti PHP-ohjelmointikielellä. Sovelluksen käsittelemät tiedot muodostettiin kohdeluokiksi. Tietoja eri sovelluksen osien välillä toimittavat funktiot sijoitettiin ohjaimiin. Tiedot näytetään näkymissä.

Arkkitehtuurikuvaus on liitteenä (liite 4).

5.3.1 Kohdeluokat

Kohdeluokat muodostettiin PHP5:n ominaisuuksia hyödyntäen. Jokaisella kohdeluokalla on muodostin `__construct()` ja yksityisiksi määriteltäviä ominaisuuksia. Kohdeluokien rooli sovelluksessa on käyttäjän syöttämien tietojen palvelinpään validointi.

Event-luokka sisältää tapahtuman tiedot. Suoraan tapahtumaan liittyvät tiedot ovat luokan attribuutteina, ja epäsuorasti liittyvät tiedot on kuvattu viitteellä muihin luokkiin. Epäsuorasti liittyvistä tiedoista on johdettu Contact-, Catering- ja Payment-luokat.

Event
name
descp
venue
date
enroll_dl
venue_site
directions
parking
transportation
program
addinfo
payment Payment
contact Contact
catering Catering

Kuva 8. Event-luokka

Contact-luokka sisältää tapahtuman yhteys henkilön tiedot.

Contact
name
surname
email
phone

Kuva 9. Contact-luokka

Catering-luokka sisältää tapahtumaan liittyvän ruokailun tiedot. Kuten tapahtumalla, myös ruokailulla on oma yhteys henkilönsä. Siksi Catering-luokan yksi attribuutti on Contact-luokan ilmentymä.

Catering
contact Contact time addinfo menu Menu

Kuva 10. Catering-luokka

Menu-luokassa on ruokailuun liittyvien ruokalajien nimet.

Menu
array menu starter main dessert

Kuva 11. Menu-luokka

Payment-luokassa kuvataan tapahtuman maksutavan tiedot.

Payment
amount method

Kuva 12. Payment-luokka

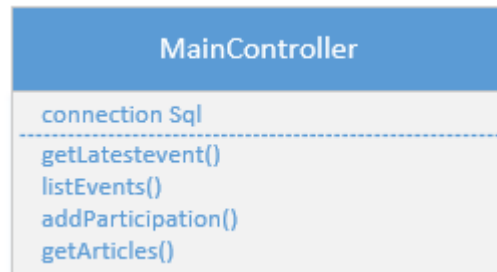
Participation-luokassa käsitellään tapahtumaan ilmoittautumisen yhteydessä syötettävät tiedot. Tapahtumaan ilmoittautumista käsittelevän Participation-luokan ja siihen liittyvä ohjausluokan ja ilmoittautumisnäkyvän lähdekoodi on liitteessä 2.

Participation
event_id member_name member_surname member_email member_phone member_addinfo

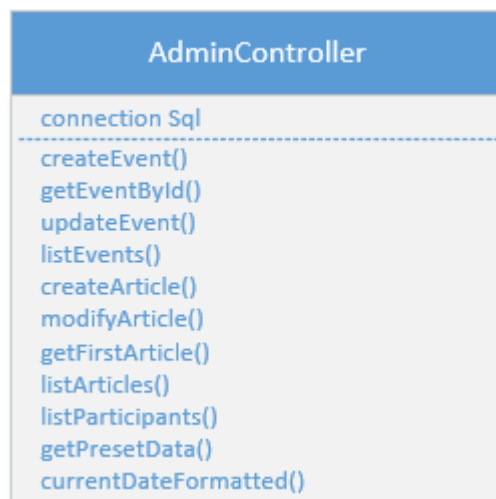
Kuva 13. Participation-luokka.

5.3.2 Ohjausluokat

Ohjausluokat ovat luokkia, jotka suorittavat kaiken dynaamisen tiedon siirron sovelluksen eri osien välillä. Sovelluksessa on kaksi ohjausluokkaa: pääohjain, joka sisältää funktiot joilla kommunikoidaan palvelun julkisten toimintojen ja tietokannan välillä, sekä hallintaohjain, jossa on funktiot sovelluksen hallintapuolen toiminnoille.



Kuva 14. Yleiskontrolleri



Kuva 15. Hallintakontrolleri

5.3.3 Apuluokat

Kohdeluokat käyttävät joidenkin käsittelysääntöjen vahvistamiseksi Validation-luokkaa, jossa on funktioita joita kohdeluokkien tietojen tarkistamisessa käytetään useita kertoja. Apuluokkaa käyttämällä funktiot tarvitsee kirjoittaa vain kerran, sen sijasta että sama tietojenkäsittelysääntö koodattaisiin joka kohdeluokkaan uudestaan.

Ohjausluokat käyttävät tietokantayhteyteen SQL-apuluokkaa.

5.3.4 Näkymät

Näkymät ovat osittaisia palvelun sivuja, joilla esitetään jokin tietty toiminto tai tietokonaisuus. Palvelussa näkymät tuodaan sivuille iframe-elementin avulla. Palvelun julkisen osan dynaamisen sisällön esittäminen on toteutettu näkymien avulla. Tapahtumiin ilmoittautumiselle, yhdistyksen esittelylle ja uutisille on omat näkymänsä. Tapahtumia ei listata julkisilla sivuilla missään, koska yhdistyksellä on avoinna enintään yksi tapahtuma kerrallaan johon voi ilmoittautua.

Ilmoittautumisnäkyssä esitetään tapahtuman tiedot ja ilmoittautumislomake kyseiseen tapahtumaan. Ilmoittautumisen jälkeen näytetään teksti ilmoittautumisen onnistumisesta. Ilmoittautumisnäkyman ulkoasun määrittelevät CSS-merkinnät ovat liitteen 2 lopussa.

The screenshot shows a web page titled "Helsingin Uudenkaupungin kilta". At the top, there are navigation links: "Toimintamme", "Ajankohtaista", "Ilmoittaudu tapahtumaan", and "Liity jäseneksi". The main content is divided into two columns. The left column is titled "Itsenäisyyspäivän juhlat/pikkujoulut" and contains the following information: "Aika 6.12.2013 Klo 17.50", "Paikka Herttoniemen kartano", "Tervetuloa itsenäisyyspäivän ja pikkujoulujen viettoon!", "Osallistumismaksu 50€ maksetaan Tapahtuman yhteydessä.", "Ilmoittauduthan viimeistään 1.12.2013.", and "Pukukoodi: Formal". The right column is titled "Ilmoittaudu" and contains a red warning message: "Tähdellä merkityt kentät ovat pakollisia." Below this are five input fields: "Etunimi*", "Sukunimi*", "Sähköposti*", "Puhelinnumero", and "Lisätietoja". At the bottom of the right column is a "Lähetä" button.

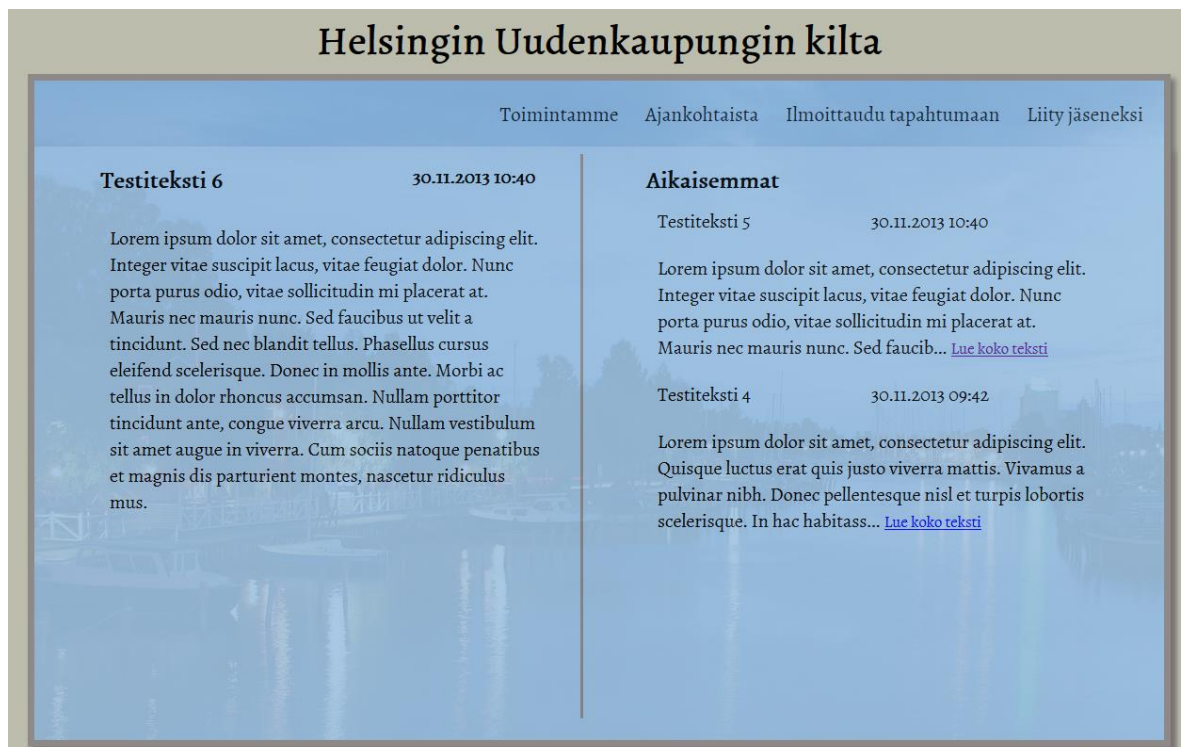
Kuva 16. Ilmoittautumisnäky

Esittelynäkylässä on yleistä tietoa yhdistyksestä sekä yhdistyksen hallituksen esittely.



Kuva 17. Esittelynäkymä

Uutisnäkylässä esitetään yhdistyksen esittelyteksti ja viimeisimmät uutiset.



Kuva 18. Uutisnäkymä

Ilmoittautumistilannenäkymä on hallintapuolen osa, jolla listataan tiettyyn tapahtumaan ilmoittautuneet henkilöt tietoineen.



Osallistujat: 3

Etunimi	Sukunimi	Sähköposti	Puhelin	Lisätiedot
Rauli	Karvinen	testimail@testimail.com	0401231234	Testilisätieto
Ernesti	Moilanen	testimail@testimail.com	0401231234	
Antti	Karhu	testimail@testimail.com	0401231234	

Kuva 19. Ilmoittautumistilannenäkymä

5.3.5 Pääsivun näkymien vaihto jQuerylla

jQuery on JavaScript-kirjasto, jonka avulla DOM-puuta ja siinä tapahtuvia toimintoja voi tarkkailla ja manipuloida. Pääsivulle on sijoitettu iframe-elementti, joka piilotetaan dokumentin latauksen ollessa valmis. Navigaation linkkien painaminen kutsuu elementin id-tribuutin mukaan jQuery-funktiota, joka asettaa iframen lähdeattribuutiksi halutun sivun. Toimintaperiaatteen koko lähdekoodi on esitetty liitteessä 2.

5.4 Käyttöliittymät

Palvelussa on oma käyttöliittymänsä palvelun julkisten osien käyttäjille ja hallinnan käyttäjille. Käyttöliittymäsuunnittelussa otettiin huomioon teoriaosassa esitettyjä vanhemman väestön mahdollisia vaikeuksia palvelun osien käytössä. Rakenne pyrittiin pitämään yksinkertaisena: yksi näyttö palvelee yhtä toiminnallisuutta ja navigaatio toimii yhdellä klikkauksella. Sisältö esitetään aina sivun keskiosassa, fontin valinnassa painotui luettavuus ja fonttikoko pidetään riittävän suurena. Alla on kuvana osa palvelun pääsivusta. Kuvan oikeassa yläreunassa on palvelun julkisen osan navigaatio.



Kuva 20. Otos pääsivusta

Navigaation valinnasta avautuu iframessa valitun toiminnallisuuden näkymä. Iframe sulkeutuu klikkaamalla sivulta mistä tahansa muualta kuin iframen sisältä. Toimintamme-painikkeesta avautuu uutisnäkymä ja ilmoittautumispainikkeesta tapahtumanäkymä. Jäseneksi liittyminen-toiminnallisuutta ei toteuteta tämän projektin puitteissa. Tapaamisessa tilaajan kanssa kuitenkin sovittiin, että liittymiseen johtava linkki esitetään navigaatiossa jo nyt.

Kuva 21. Valittu Ilmoittaudu tapahtumaan-painike navigaatiosta

Hallintapuolen käyttöliittymästä näkee suoraan tietokannassa olevat tapahtumat. Tapahtumiin ilmoittautuneita henkilöitä (ilmoittautuneet-näkymään) pääsee katsomaan tapahtuman tietojen yhteydessä olevasta painikkeesta. Uusi tapahtuma-linkistä pääsee luomaan uuden tapahtuman tietokantaan. Uutiset-näkymän sisältöä pääsee muokkaamaan Muokkaa tekstejä-linkistä.



Kuva 22. Hallinnan pääsivu, ilmoittautumiset selattavana

Tapahtuman perustiedot

Tapahtuman nimi: Tapahtumapaikka:

Tapahtuman aika: 30 Marraskuu 2013 Kello:

Viimeinen ilmoittautumispäivä: 30 Marraskuu 2013

Tapahtuman kuvaus:

Osallistumismaksu: Maksutapa: Tapahtuman yhteydessä: [Muu maksutapa?](#)

Yhteyshenkilö

Etunimi: Anne Sukunimi: Valsta

Puhelinnumero: 04000000 Sähköposti: anne.valsta-haagaheli

Tapahtuman lisätiedot

Saapumisohjeet:

Kuva 23. Tapahtuman luominen hallintasivuilla



Kuva 24. Tekstien julkaisu

5.5 Testaus

Testauksen kohteena olivat kohdan 4.2 käyttötapauskaaviossa nimetyt käyttötapaukset. Kaikki toteutetut käyttötapaukset todettiin toimiviksi.

Taulukko 2. Pääsivun toiminnallisuuksien testausraportti

Toiminnallisuus	Odotettu tulos	Todettu tulos
Tapahtumaan ilmoittautuminen	Lomakkeelle syötetyt tiedot tallentuvat tietokantaan ja käyttäjä saa vahvistuksen ilmoittautumisesta	Toimii määrittelyn mukaan
Artikkelien selaus	Palvelu näyttää artikkelit tietokannasta	Toimii määrittelyn mukaan

Taulukko 3. Hallintatoiminnallisuuden testausraportti

Toiminnallisuus	Odotettu tulos	Todettu tulos
Tapahtuman lisäys	Lomakkeelle syötetyt tiedot tallentuvat tietokantaan.	Toimii määrittelyn mukaan
Tapahtumaan ilmoittautuneiden selaus	Palvelu näyttää taulukon, jossa on valittuun tapahtumaan ilmoittautuneet henkilöt	Toimii määrittelyn mukaan
Artikkelien julkaisu	Lomakkeelle syötetyt tiedot tallentuvat tietokantaan.	Toimii määrittelyn mukaan
Tapahtuman muokkaus	Valitun tapahtuman muokatut tiedot tallentuvat tietokantaan.	Toimii vain tapahtuman pakollisten tietojen ja lisätietokentän osalta

6 Tulokset ja johtopäätökset

Projektin tuloksena syntyi vaatimusmäärittäydokumentti ja oliopohjainen PHP:lla toteutettu palvelu. Palvelun tärkeimmät ominaisuudet ovat tapahtumien organisointiin liittyvät toiminnallisuudet: tapahtuman luonti, tapahtumiin ilmoittautuminen ja ilmoittautumisten selaaminen. Tulokseen sisältyy suppea sisällönhallintajärjestelmä tekstien julkaisun ja pääsivun kuvan vaihto-ominaisuuden kautta.

Palvelusta pyrittiin alusta asti tekemään rakenteeltaan ja ulkoasultaan yksinkertainen, ja siinä onnistuttiin niin työn tekijän kuin toimeksiantajankin mielestä. Teoriaosassa esitetyt ohjeet käyttöliittymäsuunnitteluun ja käytettävyyteen auttoivat tekemään palvelusta toimeksiantajan vaatimusten mukaisen.

Projektissa onnistuttiin etenemään suunnitellulla tavalla loppuun asti. Kevyt projektinhallinta, jossa suuntaa-antavasti määritettiin palvelun toteutuksen aikataulu ja toiminnallisuuksien toteutusjärjestys, oli riittävä. Toteutuksen vähimmäisvaatimuksiin päästiin suunnitellusti aikaisessa vaiheessa projektia. Kehitystyön aikana toimeksiantajan kanssa oli kaksi katselmointikertaa.

Toteutukseen valitut menetelmät ja tekniikat soveltuivat projektiin hyvin. Olio-ohjelmointi PHP:lla soveltuu hyvin kevyen web-sovelluksen rakentamiseen. HTML5:n elementtien avulla web-sivujen rakenteista saa tehtyä aiempaa selkeämpiä vähemmällä luokkanimien ja tunnisteiden nimeämisellä.

Opinnäytetyössä käytetyt lähteet olivat pääosin sähköisiä, ja useasti teknisistä työkaluista käytettäessä parhaat tekstit löytyvät kyseisen tekniikan omalta tai tekniikan kehittäjien sivustoilta. Silloin teksteissä harvoin korostetaan tekniikan heikkouksia. Käytettyjen sähköisten lähteiden tekstit olivat kuitenkin pääosin objektiivisia ja kirjallisista lähteistä saadut tiedot tukivat sähköisten tarjoamia tietoja.

Opinnäytetyön tekijän mielestä projekti onnistui tavoitteisiin nähden oikein hyvin. Alussa asetetut tavoitteet toiminnallisuuksien ja käytettävyyden kannalta saavutettiin.

Lopputulos on hyvä, mutta projektin aikana ilmeni myös paljon parannettavaa. Suunnittelutyön merkitys korostui. Paremmalla suunnittelutyöllä olisi yksinkertaisesti säästynyt aikaa, ja projektin aikana ilmi tulleita toimeksiantajan toivomia lisätoiminnallisuuksia olisi ehtiny kehittää pidemmälle.

Tekijän näkökulmasta projekti oli opettavainen kokemus. Tekijän aiempi kokemus ohjelmistoprojekteista oli hankittu työharjoittelusta eikä oppilaitoksen ohjelmistokehittäjalinjalta. Vähäinen kokemus määrittästyöstä lisäsi työn haasteellisuutta. Määrittästyön teko oli varmasti projektin hyödyllisimpiä asioita. Käyttötapausten kuvaaminen sekä sanallisesti että kaavioin oli tekijälle hyvä oppimiskokemus.

7 Yhteenveto

Opinnäytetyössä määriteltiin ja toteutettiin Helsingin Uudenkaupungin kilta ry:n jäsenpalvelut ja kotisivut. Palvelu toteutettiin PHP-kielellä ja HTML5:llä. Tietokantana käytettiin MySQL:a.

Projektin alussa todettiin, että on olemassa yhdistyksille suunnattuja palveluita, jotka vastaavat osaan toimeksiantajan vaatimuksista. Niiden ominaisuudet ovat kuitenkin rajattuja, ja tämän projektin toimeksiantajan tavoin muidenkin yhdistyskohtaisempia toiminnallisuuksia tarvitsevien kannattaa harkita täysin oman palvelun hankkimista.

Palvelun teknisessä toteutuksessa keskeinen osa oli oliopohjainen PHP-ohjelmointi. Kohdeluokat ja ohjausluokat toteutettiin PHP5:n olio-ohjelmointia tukevia ominaisuuksia hyväksikäyttäen. Palvelun julkisen osan navigointi toteutettiin HTML:n iframe-elementin ja jQuery:n avulla. Navigoinnista linkkejä valittaessa ei ladata koko sivua uudelleen, vaan ainoastaan iframen sisältö päivittyy. Toiminnon tällä tavoin toteuttaminen toi perinteiseen ja rakenteeltaan yksinkertaiseen sivustoon modernia asynkronisaation tuntua.

Toteutetun palvelun käyttöliittymäsuunnittelussa noudatettiin WAI-AGEN ja WCAG:n ohjeistusta vanhempiin väestöryhmiin kuuluvien henkilöiden käyttökokemukseen vaikuttavista asioista. Ohjeistus oli sekä opinnäytetyön tekijän että toimeksiantajan mielestä loogista, ja toteutuneen palvelun ulkoasun katsottiin vastaavan ohjeistusta. Palvelun ulkoasun rakenne, sisällön sijoittaminen ruudun keskiosaan ja vain välttämättömien tietojen esittäminen toiminnallisuuksien yhteydessä ovat esimerkkejä ohjeistuksen toteuttamisesta.

7.1 Jatkokehitys

Opinnäytetyön alusta asti oli selvää, että työn puitteissa ei toteuteta palvelun kaikkia ominaisuuksia vaan kehitystyötä tehdään jatkuvasti jatkokehitys huomioon ottaen.

Tässä luvussa esitetyissä jatkokehitysehdotuksissa tuodaan esille ominaisuuksia, jotka parhaiten täydentävät palvelun nykyistä toteutusmuotoa.

Palveluun ei tässä projektissa toteutettu minkäänlaisia käyttöoikeuksia, lukuunottamatta hallinnan suojaamista htdocs-salasanan taakse. Palvelun julkinen etusivu on kaikkien käytettävissä. Ei ole toivottavaa, että kuka tahansa sivulle saapuva voi ilmoittautua yhdistyksen tapahtumiin. Sen vuoksi jatkokehitystä ajatellen sisäänkirjautumisjärjestelmä on korkeimman prioriteetin tehtävä.

Tämän kehitysprojektin puitteista rajattiin pois jäsenten mahdollisuus luoda sisältöä sivulle. Sivuston elävöittämisen ja Helsingin Uudenkaupungin kiltä ry:n pitkän aikajakson tavoitteiden saavuttamiseksi sen toteuttaminen olisi erittäin edullista. Toiminnallisuus edellyttää, että edellisessä kappaleessa mainittu sisäänkirjautumisjärjestelmä on toteutettu. Projektin viimeisessä tapaamisessa Killan edustajien kanssa ominaisuudesta puhuttiin nimellä 'jäsenpalsta'.

Kehitysprojektissa ei myöskään toteutettu uusien jäsenten liittymismahdollisuutta palvelun kautta. Sen tekeminen voisi madaltaa uusien jäsenten liittymiskynnystä. Toiminnallisuuden toteuttaminen saattaa olla suuritöinen, mikäli harkitaan integraatiota suoraan yhdistyksen jäsenrekisteriin.

Tällä hetkellä palvelu ei tue mobiilikäyttäjiä millään tavoin. Tämä kannattaa ottaa jatkossa huomioon muuttamalla sivut responsiivisiksi.

Palvelussa voi tällä hetkellä kirjoittaa uutisia ja julkaista ne, mutta kirjoitettuja uutisia ei voi muokata. Uutisen ulkoasua ei voi myöskään muokata minkään tekstikentän yhteydessä olevan käyttöliittymän avulla, joka sallisi lihavoinnit ja muut tyypilliset tekstikentän työkalut. Muokkaamistoiminto ja tekstityökalut toisivat sovellukseen paljon lisää.

Lähteet

Arch, A. 2008. Web accessibility for older users: a literature review. WC3 Working Draft 14.5.2008. Luettavissa: <http://www.w3.org/TR/wai-age-literature/>. Luettu: 24.10.2013

Arch, A. 2009. Web accessibility for older users: successes and opportunities. W4A '09: Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibility (W4A). Search - Web accessibility for older users: successess and opportunities. Luettavissa: <http://dl.acm.org/>. Luettu: 23.10.2013

Arch, A., Abou-Zahra, S., Lawton-Henry, S. 2008. Older users online: WAI Guidelines address older users web experience. Luettavissa: <http://www.w3.org/WAI/posts/2009/older-users-online>. Luettu: 19.11.2013

Lara, S., Watanabe, W., dos Santos, E., Fortes, R. 2010. Improving WCAG for elderly web accessibility. Search – Improving WCAG for elderly web accessibility. Luettavissa: <http://dl.acm.org/>. Luettu: 1.12.2013

Lavin, P. (2006). Object-Oriented PHP: Concepts, Techniques, and Code. No Starch Press.

Lawson, B., Sharp, R. 2011. Introducing HTML5. New Riders. Berkeley, California.

Newell, A., Dickinson, A., Smith, M., Gregor, P. 2006. Designing a portal for older users: a case study of an industrial/academic collaboration. ACM Transactions on Computer-Human Interaction. Search – Designing a portal for older users. Luettavissa: <http://dl.acm.org/>. Luettu: 24.10.2013.

Rouse, M. 2005. Definition: class. Luettavissa: <http://whatis.tech-target.com/definition/class>. Luettu: 11.11.2013.

The PHP Group. 2013a. What is PHP? Luettavissa: <http://fi1.php.net/manual/en/intro-whatis.php>. Luettu: 23.10.2013.

The PHP Group. 2013b. History of PHP. Luettavissa: <http://www.php.net/manual/en/history.php.php>. Luettu: 11.11.2013

The PHP Group. 2013c. What can PHP do? Luettavissa: <http://www.php.net/manual/en/intro-whatcando.php>. Luettu: 11.11.2013

The PHP Group. 2013d. PDO: Introduction. Luettavissa: <http://www.php.net/manual/en/intro.pdo.php>. Luettu: 6.11.2013

The PHP Group. 2013e. Visibility. Luettavissa: <http://php.net/manual/en/language.oop5.visibility.php>. Luettu: 11.11.2013.

W3C. 2008. Web Content Accessibility Guidelines 2.0. Luettavissa: <http://www.w3.org/TR/WCAG20/>. Luettu: 19.11.2013

W3C. 2012. WAI-AGE Project. Luettavissa: <http://www.w3.org/WAI/WAI-AGE/>. Luettu: 19.11.2013

W3C. 2013a. HTML5: A vocabulary and associated APIs for HTML5 and XHTML. Luettavissa: <http://www.w3.org/TR/html5/>. Luettu: 28.10.2013

W3C. 2013b. HTML5 New elements. Luettavissa: http://www.w3schools.com/html/html5_new_elements.asp. Luettu: 31.10.2013

Wikipedia. 2013. HTML5. Luettavissa: <http://en.wikipedia.org/wiki/HTML5>. Luettu: 31.10.2013.

Liitteet

HH amk

Tekijän nimi
palvelut

Vaatimusdokumentti

1 (20)

Helsingin Uudenkaupungin kiltä / jäsen-

29.11.2013

Liite 1. Vaatimusmäärittäydokumentti

VAATIMUSDOKUMENTTI

Helsingin Uudenkaupungin kiltä ry

Jäsenpalvelut ja kotisivut
Järjestelmän/ohjelmiston työnimi

Versio 1.0

Tekijä
Tarkastaja
Hyväksyjä

Luonnos | Ehdotus | **Hyväksytty**

1.11.2013 Tuomas Kujala
Anne Valsta
Anne Valsta

29.11.2013

Versiohistoria

Versiono.	Pvm	Tekijä	Muutoksen kuvaus/syy
0.1	5.10.2013	Tuomas Kujala	Dokumentti on luotu → Luonnos
0.2	1.11.2013	Tuomas Kujala	Valmis katselmoitavaksi → Ehdotus
1.0	14.11.2013	Tuomas Kujala	Katselmoinnin havainnot korjattu ja varmistettu → Hyväksytty

Tämän dokumenttimallin pohjana on käytetty seuraavia ohjeita
Bray, I. K. 2002. An Introduction to Requirements Engineering
Volere, Requirements. Specification Template, Edition 13, Aug. 2007.

29.11.2013

SISÄLTÖ

1	KEHITETTÄVÄN TOIMINNAN KUVAUS.....	4
1.1	TOIMINNAN YLEISKUVAUS.....	4
1.2	SIDOSRYHMÄT	4
1.3	SAAVUTETTAVAT HYÖDYT.....	4
1.3.1	<i>Prosessit.....</i>	5
1.3.2	<i>Palvelun käyttäjät.....</i>	5
1.3.3	<i>Liittymät muuhun toimintaan.....</i>	5
1.3.4	<i>Reunaehdot toiminnan kehittämiseksi.....</i>	5
1.3.5	<i>Toiminnan turvallisuus ja laatu.....</i>	5
1.4	LIIKETOIMINNAN KÄSITTEET, NIMEÄMINEN JA ASIAYHTEYDET.....	6
2	VAATIMUKSET RATKAISUN TOIMIVUUDELLA	6
2.1	TOIMINNALLISET VAATIMUKSET RATKAISULLE	6
2.1.1	<i>Ratkaisun yleiskuvaus</i>	6
2.1.2	<i>Käyttötapauskuvaukset.....</i>	7
2.2	SÄILYTETTÄVÄT TIEDOT	11
2.2.1	<i>Tietojen käyttö</i>	15
2.2.2	<i>Alustavat näyttö- ja tulostehahmotelmat</i>	16
2.2.3	<i>Käsittelysäännöt</i>	18
2.2.4	<i>Liittymät muihin järjestelmiin.....</i>	18
2.3	SUORIUTUMISVAATIMUKSET.....	18
2.3.2	<i>Käytettävyyksivaatimukset.....</i>	19
2.3.3	<i>Turvallisuus ja laatu</i>	19
2.4	SUUNNITTELUN RAJOITTEET JA YMPÄRISTÖVAATIMUKSET.....	19
2.5	AVOIMET KYSYMYKSET	20
3	TIETOHAKEMISTO	20
4	VIITTAUKSET JA LIITTEET	20

29.11.2013

1 KEHITETTÄVÄN TOIMINNAN KUVAUS

Helsingin Uudenkaupungin kilta on yhdistys, jonka pääasiallinen tarkoitus on ylläpitää Helsingissä ja sen ympäristössä asuvien entisten uuskaupunkilaisten kotiseutuyhteyksiä ja edistää vanhan kielen säilyttämistä. Tarkoituksensa toteuttamiseksi kilta järjestää kokouksia ja muita tilaisuuksia jäsentensä kesken.

1.1 Toiminnan yleiskuvaus

Tuleva palvelu keskittyy killan tapahtumien järjestämiseen liittyvien tietojenkäsittelytehtävien automatisointiin. Aikaisemmin tapahtumiin liittyvät tietojenkäsittelytehtävät on hoidettu käsin. Tuleva palvelu helpottaa tapahtumien hallinnoinnista vastaavien henkilöiden tehtäviä.

1.2 Sidosryhmät

Kohdealueella on kaksi toimijaa: killan tapahtumia hallinnoivat henkilöt sekä killan jäsenet.

1.3 Saavutettavat hyödyt

Tapahtumien hallinnoiminen on aikaisemmin tapahtunut täysin manuaalisesti. Osallistujamäärien seuranta ja tapahtumiin liittyvien tietojen säilytys on ollut epäkäytännöllistä.

Tuleva palvelu automatisoi tapahtumien hallintaan liittyvät toiminnot ja siten keventää tapahtumien järjestämistä koordinoivien henkilöiden työtaakkaa. Palvelun asiakkaita ovat tilaajan jäsenet, jotka tulevat käyttämään palvelua

29.11.2013

tapahtumiin ilmoittautumiseen. Palvelun käyttöönotolla poistetaan puhelimen ja sähköpostin käyttö ilmoittautumisessa.

1.3.1 Prosessit

Tapahtuman tietojenkäsittelyprosessia kuvaava prosessikartta on liitteessä A.

1.3.2 Palvelun käyttäjät

Palveluun ei toteuteta käyttäjäryhmiä/käyttöoikeuksia. Käyttäjäroolit määritellään tilaajan organisaatorakenteen avulla:

1. Tilaajan hallituksen jäsenet

Hallituksen jäsenillä on pääsy salasanasuojattuun käyttöliittymään, jonka kautta voi luoda/poistaa/muokata tietoja tietokannan tapahtuma (event)-taulussa ja listata kaikkia event-tauluun liittyviä tietoja. Mainittuja toimintoja käyttää pääasiassa tilaajan emäntänä toimiva henkilö.

2. Tilaajan muut jäsenet

Muut jäsenet voivat ilmoittaa osallistumisensa tapahtumiin. Ei peruutusta

1.3.3 Liittymät muuhun toimintaan

Tilaajalla ei ole ennestään käytössä olevia palveluita.

1.3.4 Reunaehdot toiminnan kehittämiselle

Palvelun värimaailman halutaan mukailevan Uudenkaupungin vaakunan värejä. Lisäksi lomake- ja muissa malleissa täytyy ottaa huomioon, että kohde-ryhmästä osa kuuluu vanhempaan väestöön.

Tiettyjä teknologioita tai tuotantoympäristöjä ei ole vaadittu.

1.3.5 Toiminnan turvallisuus ja laatu

29.11.2013

Palvelussa ei käsitellä arkaluontoisia tietoja, kuten henkilötunnuksia. Palvelua käyttävistä henkilöistä säilytetään nimet, sähköpostit ja/tai puhelinnumerot. Tilaajalla ei ennestään ole käytössä mitään tietoturvakäytäntöjä.

Palvelu pyritään rakentamaan mahdollisimman kevyeksi.

1.4 Liiketoiminnan käsitteet, nimeäminen ja asiayhteydet

Normaalin yhdistystoiminnan käsitteistö.

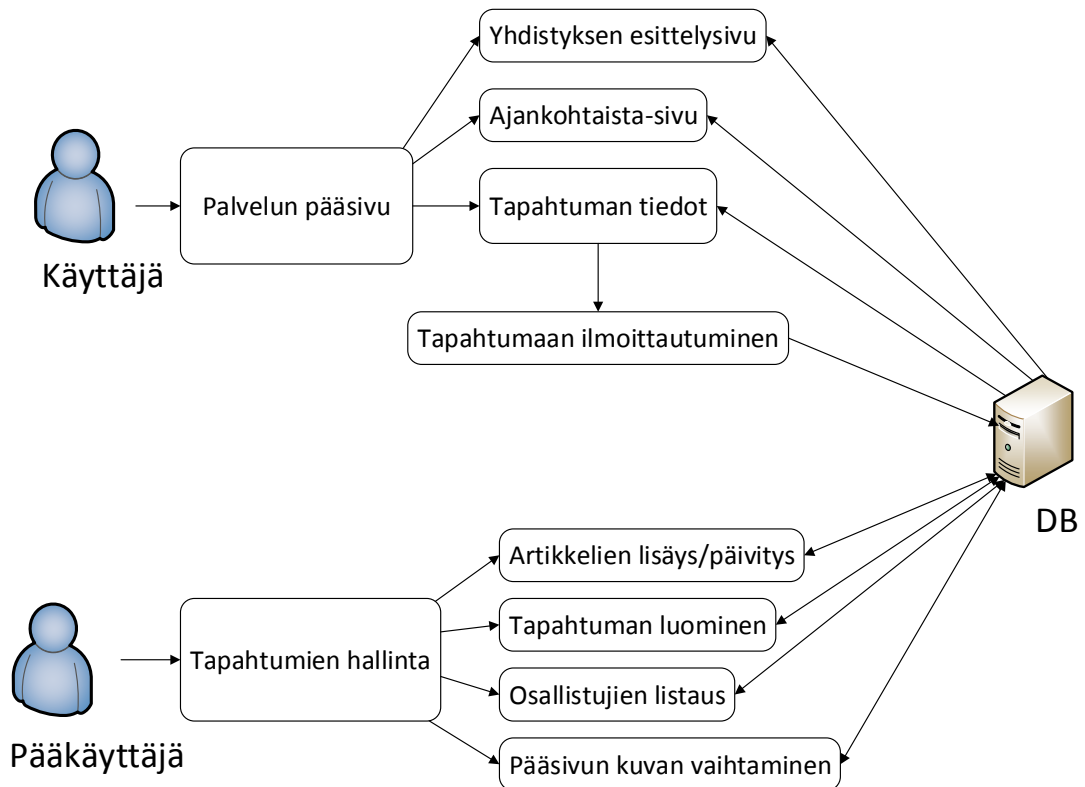
2 VAATIMUKSET RATKAISUN TOIMIVUUDELLE

Tämä osa dokumenttia kuvaa tulevan ratkaisun eli vaatimukset ohjelmistolle.

2.1 Toiminnalliset vaatimukset ratkaisulle

2.1.1 Ratkaisun yleiskuvaus

29.11.2013



2.1.2 Käyttötapauskuvaukset

1. Tapahtuman luominen

Toimijat: Tilaajan emäntänä toimiva henkilö tai muut tilaajan hallituksen jäsenet

Esiehto: Käyttäjä saapuu tapahtuman luominen-sivulle.

Lopputulos: Käyttäjän määrittämät tapahtumatiedot tallentuvat tietokantaan.

Käyttötiheys: 2-5 kertaa vuodessa

Step-by-step

1. Palvelu näyttää lomakkeen, jolle täytetään tapahtuman tiedot
2. Käyttäjä syöttää haluamansa tiedot lomakkeelle ja painaa tallenna-nappia.
3. Palvelu tarkistaa syötteen

29.11.2013

- a. (Syötettyjen tietojen oikeellisuus tarkistetaan serveripuolen ja clientpuolen tekniikalla?) Tietojen hyväksyminen/hylkäys. Jos hylätty, palataan kohtaan 2.
 - b. Lomake lähettää tiedot palvelimelle
 - c. Palvelin tarkistaa ja tallentaa tiedot tietokantaan.
4. Palvelu ilmoittaa, että tiedot on tallennettu.

2. Tapahtumaan ilmoittautuminen

Toimijat: Kaikki tilaajan jäsenet

Esiehto: Toimija on valinnut palvelun etusivulta ilmoittautumisen tapahtumaan.

Lopputulos: Käyttäjä tallentaa osallistumistietonsa valittuun tapahtumaan.

Käyttötiheys: Tapahtumakohtaisesti 20-100 kertaa.

Step-by-step

1. Palvelu näyttää tapahtuman johon voi ilmoittautua ja ilmoittautumislomakkeen.
2. Käyttäjä täyttää ilmoittautumiseen vaaditut tiedot ja painaa tallenna-painiketta.
3. Palvelu tarkistaa tiedot ja tallentaa ilmoittautumisen kyseiseen tapahtumaan. Palvelu ilmoittaa ilmoittautumisen onnistumisesta.
 - a. Syötettyjen tietojen oikeellisuus tarkistetaan serveripuolen ja clientpuolen tekniikalla. Tietojen hyväksyminen/hylkäys. Jos hylätty, palataan kohtaan 2.

3. Ilmoittautumisten listaus

Toimijat: Tilaajan emäntänä toimiva henkilö tai muut tilaajan hallituksen jäsenet

Esiehto: On olemassa tapahtuma, johon voi ilmoittautua ja toimija on hallintaisivulla.

Lopputulos: Käyttäjä näkee sivulla valitsemaansa tapahtumaan ilmoittautuneiden henkilöiden tiedot ja lukumäärän.

Käyttötiheys: Muutamia kertoja kunkin tapahtuman osalta

Step-by-step

29.11.2013

1. Palvelu näyttää tapahtumat.
2. Toimija valitsee tapahtuman.
3. Palvelin hakee tietokannasta valittuun tapahtumaan ilmoittautuneiden henkilöiden tiedot ja listaa tiedot sivulle.

4. Yhdistyksen päivitys

Toimijat: Tilaajan hallituksen jäsenet

Esiehto: Toimija on valinnut hallintasivulta Tekstit-sivun.

Lopputulos: Toimijan kirjoittama teksti tallentuu tietokantaan ja toimija saa ilmoituksen tallennuksesta.

Step-by-step

1. Palvelu näyttää aiemman esittelytekstin HTML5-artikkelina tietokannasta.
2. Toimija painaa muokkaa-linkkiä.
3. Palvelu muuttaa artikkelin muokattavaksi tekstikentäksi.
4. Toimija tekee muutokset ja painaa tallenna-nappia.
5. Palvelu tallentaa tiedot tietokantaan ja antaa ilmoituksen tallentamisesta.
 - a. Syötetyt tiedot tarkistetaan. Jos tulos on hylätty, palataan kohtaan 4.

5. Uutisten lisäys ja päivitys

Toimijat: Tilaajan hallituksen jäsenet

Esiehto: Toimija on valinnut hallintasivulta Tekstit-sivun.

Lopputulos: Toimijan kirjoittama teksti tallentuu tietokantaan ja toimija saa ilmoituksen tallennuksesta.

1. Palvelu näyttää tekstikentän, jonka päällä on teksti "Kirjoita uutinen".
2. Toimija kirjoittaa tekstin ja painaa tallenna-nappia.
3. Palvelu tallentaa tiedot tietokantaan ja antaa ilmoituksen tallentamisesta.
 - a. Syötetyt tiedot tarkistetaan. Jos tulos on hylätty, palataan kohtaan 2.

29.11.2013

6. Pääsivun kuvan vaihtaminen

Toimijat: Tilaajan hallituksen jäsenet

Esiehto: Toimija on valinnut hallintasivulta Vaihda kuva-sivun.

Lopputulos: Toimijan valitsema kuva tallentuu palvelimelle ja yliajaa olemassaolevan kuvan.

1. Palvelu näyttää painikkeen, josta pääsee selaamaan toimijan laitteella olevia tiedostoja.
2. Toimija valitsee laitteeltaan kuvan.
3. Palvelu ilmoittaa onko kuva mittasuhteiltaan käyttöön sopiva.
 - a. Jos ei ole, palataan kohtaan 2.
4. Toimija painaa tallenna-painiketta.
5. Palvelu tallentaa kuvan.

7. Artikkelien ja esittelyn selaaminen

Toimijat: Kaikki tilaajan jäsenet

Esiehto: Toimija on saapunut palvelun etusivulle.

Lopputulos: Toimija näkee valitut tiedot sivulla.

1. Toimija valitsee navigaatiosta Toimintamme/Ajankohtaista-sivun.
2. Palvelu näyttää tietokannasta esittelytekstin/viimeisimmät uutiset.

8. Tapahtuman muokkaus

Toimijat: Tilaajan emäntänä toimiva henkilö tai muut tilaajan hallituksen jäsenet

Esiehto: Käyttäjä valitsee hallinnan pääsivulta (Ilmoittautumiset) muokkaa tapahtumaa-linkin.

Lopputulos: Käyttäjän määrittämät tapahtumatiedot tallentuvat tietokantaan.

Step-by-step täysin sama kuin käyttötapauksessa 1 ensimmäistä kohtaa luennottamatta.

1. Palvelu näyttää lomakkeen, jolla ovat valitun tapahtuman aikaisemmat tiedot.

29.11.2013

2.2 Säilytettävät tiedot

Tietokantakaavio on tämän dokumentin liitteenä B.

Luokkakuvaukset

1.0 Event

Määritelmä Tapahtuma					0..*
Synonyymit					
Yliluokan nimi (jos kuvattu luokka on aliluokka)					
Attribuutin nimi	tietotyyppi	Pituus	arvoalue	kuvaus	pakollisuus
event_id	int			tapahtuman tunniste, pää-avain	not null
event_name	varchar	128		tapahtuman nimi	not null
event_descp	varchar	256		tapahtuman kuvaus	not null
event_venue	varchar	128		tapahtuma-paikka	not null
event_date	date			tapahtuma-aika	not null
event_enroll_dl	date			viimeinen ilmoittautumispäivä	not null
event_payment_id	int			osallistumismaksun tunniste	
event_contact_id	int			tapahtuman yhteyshenkilö	not null
event_catering_id	int			tapahtuman catering-yhteyshenkilö	
event_venue_site	varchar	256		tapahtumapaikan url	
event_directions	varchar	512		kulkutiedot tapahtumapaikalle	

29.11.2013

event_parking	boolean			tapahtumapaikalla parkkitilaa, kyllä/ei	
event_transportation	varchar	256		kuljetusmuoto	
event_program	varchar	512		tapahtuman ohjelma	
event_addinfo	varchar	512		tapahtuman lisätiedot	

1.1 member

Määritelmä tapahtumaan ilmoittautuva tilaajan jäsen					0..*
Synonyymit					
Yliluokan nimi (jos kuvattu luokka on aliluokka)					
Attribuutin nimi	tietotyyppi	Pituus	arvoalue	kuvaus	pakollisuus
member_id	int			jäsenen tun- niste, pää- avain	not null
member_name	varchar	28		jäsenen etu- nimi	not null
member_sur- name	varchar	56		jäsenen suku- nimi	not null
member_email	varchar	128		jäsenen säh- köposti	not null
member_phone	varchar	28		jäsenen puhe- linnumero	
member_addinfo	varchar	256		lisätiedot	

1.2 participation

Määritelmä jäsenen osallistuminen tapahtumaan					0..*
Synonyymit					
Yliluokan nimi (jos kuvattu luokka on aliluokka)					
Attribuutin nimi	tietotyyppi	Pituus	arvoalue	kuvaus	pakollisuus
event_id	int			tapahtuman tunniste	not null

29.11.2013

member_id	int			jäsenen tun- niste	not null
-----------	-----	--	--	-----------------------	----------

1.3 participation_price

Määritelmä Osallistumismaksu					0..*
Synonyymit					
Yliluokan nimi (jos kuvattu luokka on aliluokka)					
Attribuutin nimi	tietotyyppi	Pituus	arvoalue	kuvaus	pakollisuus
price_id	int			maksun tun- niste	not null
amount	int			maksun määrä	not null
price_pmet- hod_id	int			maksutavan tunniste	not null

1.4 payment_method

Määritelmä osallistumismaksun maksutapa					0..*
Synonyymit					
Yliluokan nimi (jos kuvattu luokka on aliluokka)					
Attribuutin nimi	tietotyyppi	Pituus	arvoalue	kuvaus	pakollisuus
method_id	int			maksutavan tunniste	not null
method_name	varchar	128		maksutavan nimi/kuvaus	not null

1.5 contact

Määritelmä tapahtuman yhteyshenkilö					0..*
Synonyymit					
Yliluokan nimi (jos kuvattu luokka on aliluokka)					
Attribuutin nimi	tietotyyppi	Pituus	arvoalue	kuvaus	pakollisuus
contact_id	int			yhteyshenkilön tunniste, pää- avain	not null
contact_name	varchar	28		yhteyshenkilön etunimi	not null
contact_surname	varchar	56		yhteyshenkilön sukunimi	not null

29.11.2013

contact_email	varchar	128		yhteyshenkilön sähköposti	not null
contact_phone	varchar	28		yhteyshenkilön puhelinnumero	

1.6 catering

Määritelmä catering-yhteyshenkilö					0..*
Synonyymit					
Yliluokan nimi (jos kuvattu luokka on aliluokka)					
Attribuutin nimi	tietotyyppi	Pituus	arvoalue	kuvaus	pakollisuus
catering_id	int			catering-yhteyshenkilön tunniste	not null
catering_name	varchar	28		catering-yhteyshenkilön etunimi	not null
catering_sur-name	varchar	56		catering-yhteyshenkilön sukunimi	not null
catering_email	varchar	128		catering-yhteyshenkilön sähköposti	not null
catering_menu_id	int			menun tunniste	not null
catering_time	date			ruokailun ajankohta	
catering_addinfo	varchar	512		lisätietoa ruokailusta	

1.7 menu

Määritelmä ruokailun ruokalajit					0..*
Synonyymit					
Yliluokan nimi (jos kuvattu luokka on aliluokka)					
Attribuutin nimi	tietotyyppi	Pituus	arvoalue	kuvaus	pakollisuus
menu_id	int			menun tun- niste	not null
menu_starter	varchar	56		alkuruoka	not null
menu_main	varchar	128		pääruoka	not null
menu_dessert	varchar	56		jälkiruoka	not null

29.11.2013

1.8 news

Määritelmä esittelyteksti ja artikkelit					0..*
Synonyymit					
Yliluokan nimi (jos kuvattu luokka on aliluokka)					
Attribuutin nimi	tietotyyppi	Pituus	arvoalue	kuvaus	pakollisuus
id	int			artikkelin tun- niste	not null
name	varchar	128		artikkelin nimi	not null
post	varchar	1056		artikkelin si- sältö	not null
posted	varchar	28		artikkelin li- säyspäivä- määrä	not null

2.2.1 Tietojen käyttö

Taulukko 2. Tietojen käyttöyhteenveto, jossa

R = hakee, C = lisää, U = muuttaa, D = poistaa

Käyttötapaukset	Luokat							
	Event	Parti- cipa- tion	Mem- ber	Con- tact	Cate- ring	Men u	Participa- tion_pric e	News
Tapahtuman lisää- minen	C			C	C	C	C	
Tapahtumaan il- moittautuminen	R	C	C	R			R	
Ilmoittautuneiden selaaminen	R	R	R					
Uutisen lisäys ja päivitys								RCU
Uutisten selaus								R

29.11.2013

2.3.3 Alustavat näyttö- ja tulostehahmotelmat

The screenshot shows a web application titled 'Helsingin Uudenkaupungin kilta'. At the top, there are navigation links: 'Ilmoittautumiset', 'Uusi tapahtuma', 'Muokkaa tekstejä', 'Vaihda kuva', and 'Pääsivustoon'. The main content area is titled 'Tapahtuman perustiedot' and contains several input fields and dropdown menus. The fields are organized as follows:

- Tapahtuman nimi:** A text input field.
- Tapahtumapaikka:** A text input field.
- Tapahtuman aika:** A date selector with a dropdown for the month (currently 'Marraskuu') and a year (currently '2013').
- Kello:** A time selector with two input fields for hours and minutes.
- Viimeinen ilmoittautumispäivä:** A date selector with a dropdown for the month (currently 'Marraskuu') and a year (currently '2013').
- Tapahtuman kuvaus:** A large text area for the event description.
- Osallistumismaksu:** A text input field.
- Maksutapa:** A dropdown menu with the option 'Tapahtuman yhteydessä' and a link 'Muu maksutapa?'.
- Yhteyshenkilö:** A section header for the contact person.
- Etunimi:** A text input field with the value 'Anne'.
- Sukunimi:** A text input field with the value 'Valsta'.
- Puhelinnumero:** A text input field with the value '04000000'.
- Sähköposti:** A text input field with the value 'anne.valsta-haagaheli'.
- Tapahtuman lisätiedot:** A section header for additional event information.
- Saapumisohjeet:** A text input field for arrival instructions.

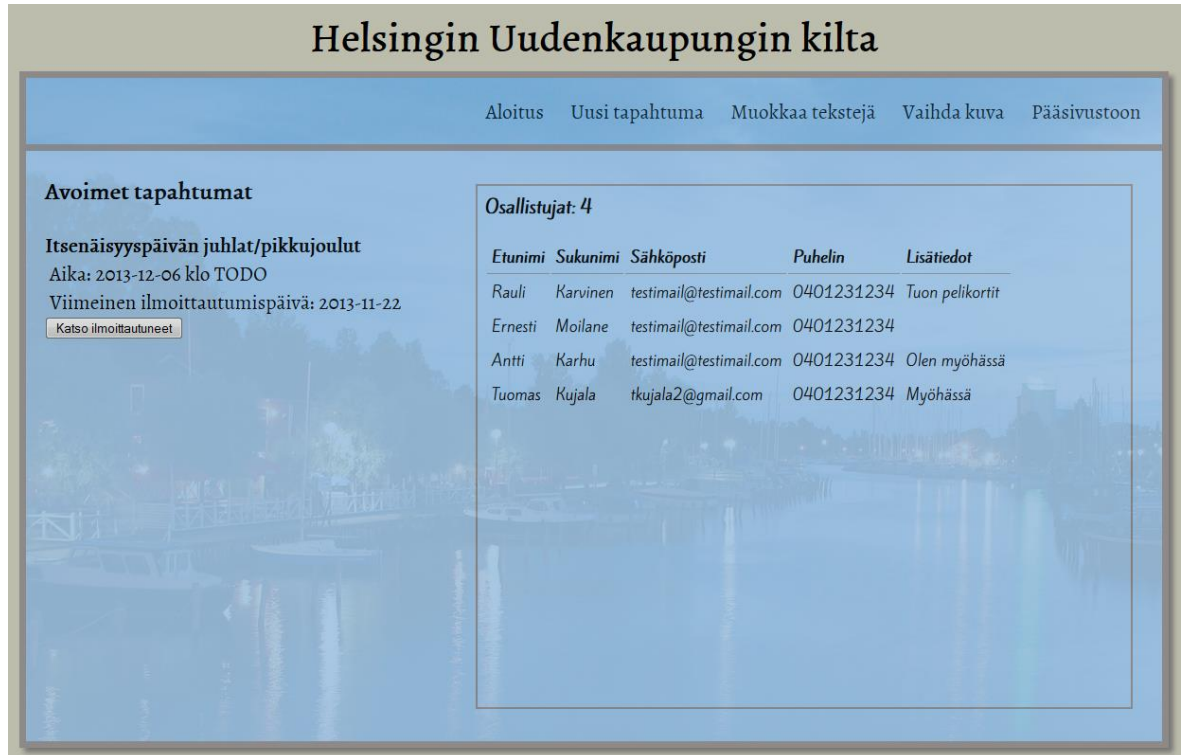
Kuva 1. Tapahtuman luominen

The screenshot shows a web application titled 'Helsingin Uudenkaupungin kilta'. The main content area is titled 'Ilmoittaudu tapahtumaan' and contains a form for signing up for an event. The form is organized as follows:

- Itsenäisyyspäivän juhlat/pikkujoulut:** A section header for the event.
- Aika:** A text input field with the value '2013-12-06'.
- Paikka:** A text input field with the value 'Herttoniemen kartano'.
- Tervetuloa itsenäisyyspäivän ja pikkujoulujen viettoon!** A text input field for a welcome message.
- Osallistumismaksu 50€ maksetaan Tapahtuman yhteydessä.** A text input field for the participation fee.
- Ilmoittauduthan viimeistään 2013-11-22.** A text input field for the deadline.
- Ilmoittautumisen tiedot:** A section header for the registrant's information.
- Etunimi:** A text input field.
- Sukunimi:** A text input field.
- Sähköposti:** A text input field.
- Puhelinnumero:** A text input field.
- Lisätietoja:** A text input field.
- Lähetä:** A button to submit the form.

Kuva 2. Tapahtumaan ilmoittautuminen

29.11.2013



Kuva 3. Ilmoittautuneiden selaus



Kuva 4. Uutisten lisäys ja päivitys

29.11.2013

2.3.4 Käsittelysäännöt

Tapahtumaan ilmoittautuminen

Tapahtumaan ilmoittautumislomakkeessa käyttäjän tiedot tarkistetaan sekä selaimessa että palvelimella. Nimistä tarkistetaan, että kentät eivät ole tyhjiä eikä syöte ole alle 2 merkin pituinen, ja että kentät sisältävät ainoastaan kirjaimia. Sähköpostiosoitteesta tarkistetaan, että syöte sisältää @-merkin sekä .-merkin. Jos puhelinnumero on syötetty, tarkistetaan, että se sisältää vain numeroita ja sen pituus on 6-10 merkkiä. Selaimessa tarkistukset tehdään JavaScriptin RegExp-luokan avulla.

Selaimessa tapahtuvat virheet esitetään käyttäjälle ilmoittautumislomakkeen alla. Kun syötetyissä tiedoissa havaitaan virhe, lomakkeen käsittely keskeytetään ja virhekohtainen virheilmoitusteksti esitetään.

Palvelinpuolella tehdään samat tarkistukset PHP:lla. Sähköpostin validointiin palvelinpuolella käytetään PHP:n omaa filter_var-funktiota.

Tapahtuman luominen

Tapahtuman luomisessa pakollisiksi kentiksi on määritetty seuraavat tiedot:

- tapahtuman nimi
- tapahtumapaikka
- päivämäärä
- ilmoittautumisen päättymispäivämäärä
- kaikki yhteyshenkilön tiedot

Pakollisen tiedon puuttumisesta ilmoitetaan muuttamalla kyseessä olevan kentän reunukset punaiseksi.

2.3.5 Liittymät muihin järjestelmiin

Liittymiä muihin järjestelmiin ei ole.

2.4 Suoriutumisvaatimukset

29.11.2013

2.4.3 Käytettävyyysvaatimukset

Web Content Accessibility-suosituksista johdettuja käytettävyyysvaatimuksia sovellukselle:

- sivun sisältö keskelle ruutua
- selkeä jaottelu sivun eri elementtien välillä
- ei vierityspalkkeja
- yhdelle sivulle vain sivun toiminnallisuuden kannalta välttämättömät tiedot
- käyttäjäsyötteen avustaminen ja esitäyttö, kun mahdollista

2.4.4 Turvallisuus ja laatu

Taulukko 3. Toimijoiden käyttöoikeudet

Käyttötapaukset	Toimijat	
	Hallinnan käyttäjä	Peruskäyttäjä
Tapahtuman lisäys	x	
Tapahtumaan ilmoittautuminen	x	x
Uutisten lisäys/päivitys	x	
Uutisten selaus		x
Tapahtumaan ilmoittautuneiden selaus	x	
Kuvan vaihtaminen	x	
Uutisten selaus	x	x
Tapahtuman muokkaus	x	

Työssä lähdetään käytettävyyšnäkökulmasta. Laadulliset vaatimukset liittyvät käytettävyyteen ja edellisessä kappaleessa mainittuihin käytettävyyysvaatimuksiin.

Turvallisuuteen liittyvät seikat on hallintaosion salasanasuojausta lukuunottamatta rajattu tämän projektin ulkopuolelle.

2.4 Suunnittelun rajoitteet ja ympäristövaatimukset

29.11.2013

Palvelun kehitys- ja tuotantoympäristöltä vaaditaan PHP5-tuki ja MySQL-tietokantojen käyttömahdollisuus. Kehitystyössä käytetään Haaga-Helia ammattikorkeakoulun työasemia, Eclipse Kepleriä ja XAMPP-palvelinympäristöä.

Toteutuksessa syntyy valmis kaksipalstainen sivumalli sovelluksen julkisen osan jatkokehityksessä toteutettavia toiminnallisuuksia varten.

2.5 Avoimet kysymykset

Toimeksiantajan kanssa on sovittu tarkasta rajauksesta.

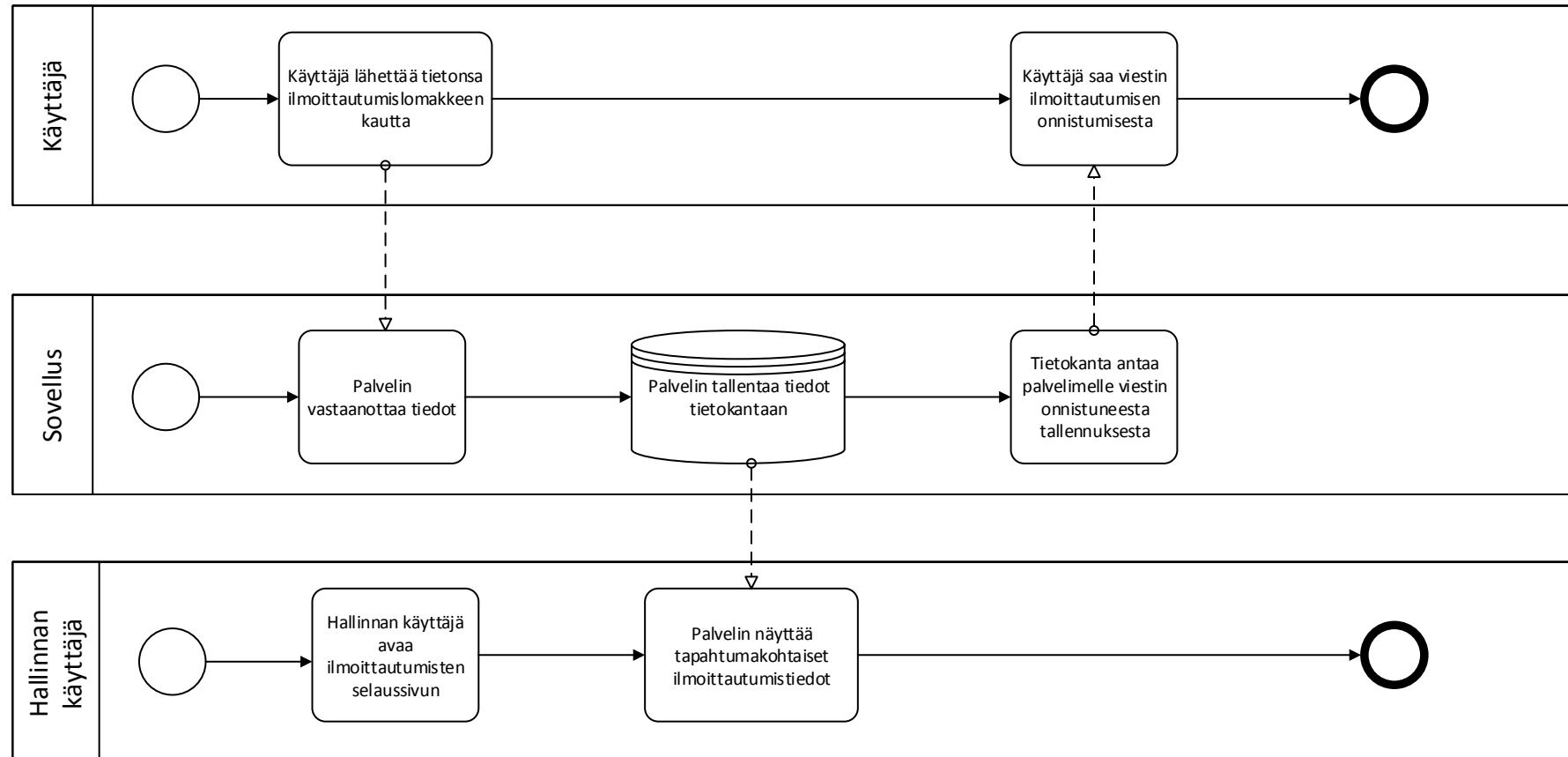
3 TIETOHAKEMISTO

Kaikki luokkien ominaisuudet on kuvattu nimeämistavalla *luokannimi_attri-buutti*, esimerkiksi *\$event_date*. Tätä nimeämistapaa noudatetaan tietokantaoperaatioihin liittyvissä sovelluksen osissa, joita ovat SQL-lauseet ja palvelimelle lähetettävän datan käsittelyfunktiot. PHP:n \$_POST-taulukkoon tulevat tiedot saavat nimensä HTML-lomakkeiden name-attribuutilta.

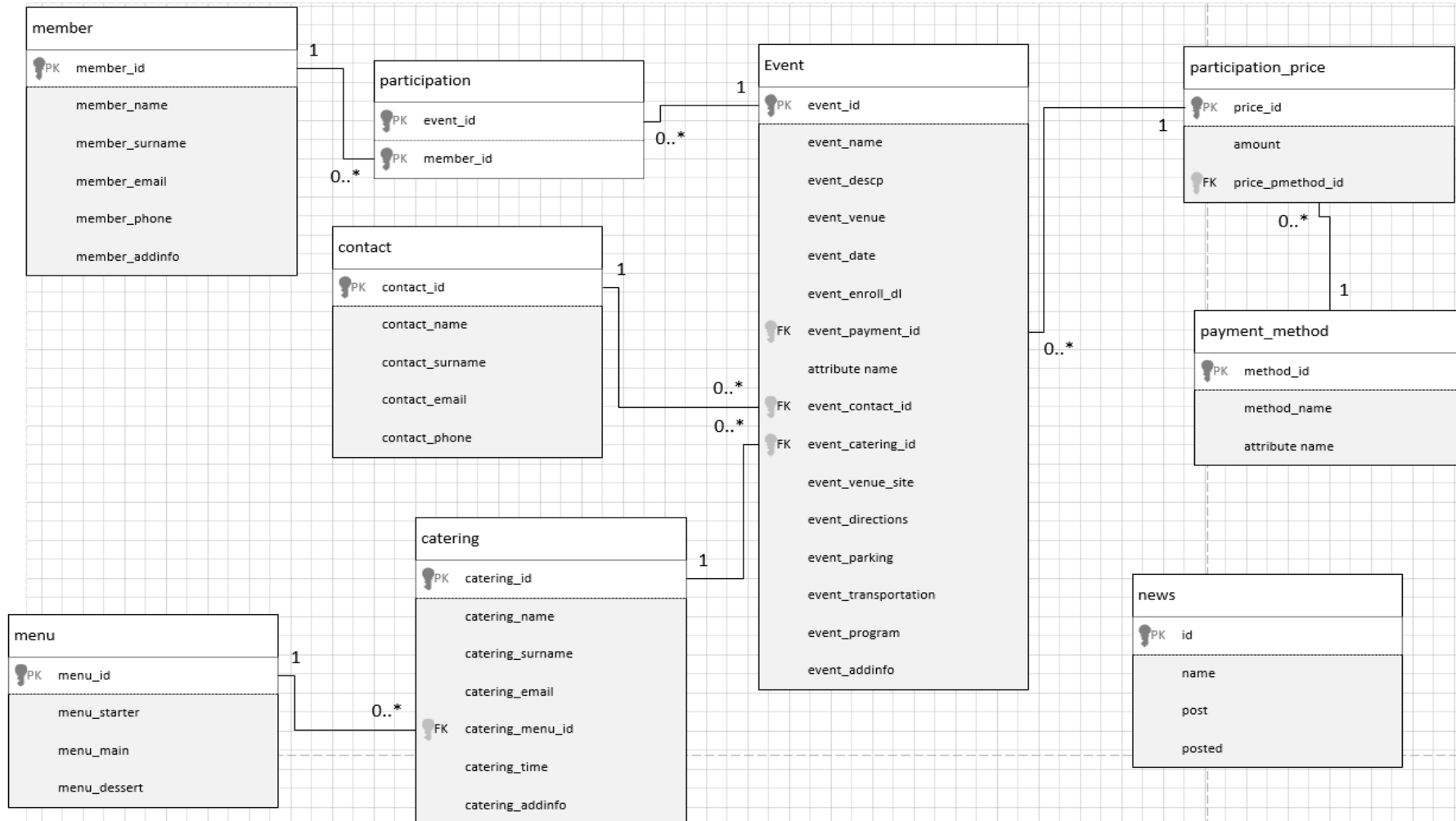
4 VIITTAUKSET JA LIITTEET

- | | |
|---------|--|
| Liite A | Tapahtuman tietojenkäsittelyä kuvaava prosessikartta |
| Liite B | Tietokantakaavio |

Liite A. Prosessikaavio



Liite B. Tietokantakaavio



Liite 2. Iframe-navigoinnin lähdekoodi

Main.js

```
$( 'document' ).ready(function(){  
    // excluding navigation link id's from document.click  
  
    $(document).click(function(e){  
        if(e.target.id == "enroll" || e.target.id == "join" || e.target.id == "newsPage" || e.target.id == "association"){  
            return false;  
        }  
  
        // hide iframe onclick anywhere from the nav items above and the iframe itself  
  
        $( 'iframe' ).fadeOut("slow", function(){  
  
        });  
  
    });  
});
```

```
// navigation - setting iframe src

$('#association').click(function(){
    $('#iframe').attr('src', 'association.php');
    $('#iframe').fadeIn('slow');
});

$('#newsPage').click(function(e){
    $('#iframe').attr('src', 'content.php');
    $('#iframe').fadeIn('slow');
});

$('#enroll').click(function(e){
    $('#iframe').attr('src', 'event.php');
    $('#iframe').fadeIn('slow');
});

$('#join').click(function(e){
    $('#iframe').attr('src', 'join.php');
    $('#iframe').fadeIn('slow');
});
```

```
// hide the iframe document.ready

$('#iframe').hide();
});
```

Liite 3. Tapahtumaan ilmoittautumisen lähdekoodi

```
<?php
require_once('../..controller.php');

// controller instance

$controller = new Controller();

$event = $controller->getLatestEvent();

// post enrollment data on submit

if(isset($_POST['enroll'])){

    $controller->addParticipation();
}

?>
<!doctype html>

<head>
<link href='http://fonts.googleapis.com/css?family=Alegreya' rel='stylesheet'>
<link href='css/iframe.css' rel='stylesheet'>
<link href="css/event.css" rel='stylesheet'>
<script src="http://code.jquery.com/jquery-1.9.1.js"></script>
<script src="js/enrollValidation.js"></script>
</head>
```



```

<div class="event_wrapper">

    <h3 class="event_header"><?php echo($event['event_name']);?>
    </h3>

    <!-- Event info -->

    <table>

    <tr><td><b>Aika</b></td><td><?php echo $event['event_date'];?></td></tr>
    <tr><td><b>Paikka</b></td><td><?php echo $event['event_venue'];?></td></tr>
    </table>

    <p class="event_descp"><?php echo($event['event_descp']);?></p>

    <p class="event_pprice"><?php print("Osallistumismaksu ".$event['amount']."€ maksetaan ".$event['payment_method']);?>

    <p class="event_enroll_dl"><?php print("Ilmoittauduthan viimeistään ".$event['event_enroll_dl']);?>.</p>

    <p class="event_addinfo"><?php print($event['event_addinfo']);?>

</div>

<div id="middleBorder"></div>

```

```

<!-- Enroll form -->

<form method="post" id="enrollForm" onsubmit="return validateEnrollForm()">
<h3>Ilmoittaudu</h3>
<div class="required">Tähdellä merkityt kentät ovat pakollisia.</div>
<table>

    <tr><td>Etunimi<span class="required">*</span></td><td><input name="name" type="text"></td></tr>
    <tr><td>Sukunimi<span class="required">*</span></td><td><input name="surname" type="text"></td></tr>

    <tr><td>Sähköposti<span class="required">*</span></td><td><input name="email" type="text"></td></tr>
    <tr><td>Puhelinnumero</td><td><input name="phone" type="text"></td></tr>

<tr><td>Lisätietoja</td>

<td><textarea name="addinfo" form="enrollForm"></textarea></td></tr>

<!-- hidden field with event id -->

<tr><td><input id="hidden" type="hidden" name="hidden" value="<?php echo $event['event_id'];?>"></td></tr>

<tr><td><input id="enrollButton" type="submit" name="enroll" value="Lähetä"></td></tr>

<tr><td></td><td class="required" id="error"></td></tr>
</table>
</form>

</html>

```

Controller.php – ilmoittautumisen tallennus

```
function addParticipation(){  
    require_once('class/Participation.class.php');  
    $participation = new Participation();  
    // return 1 for success, 0 for fail  
    $saveStatus = $participation->saveParticipation();  
    return $saveStatus;  
}
```

Participation.class – osallistumisen tallennus

```
function saveParticipation(){  
    $saveMemberQuery = "insert into member  
        (member_name, member_surname, member_email, member_phone, member_addinfo)  
        values  
        ('$this->member_name', '$this->member_surname', '$this->member_email', '$this->member_phone', '$this->member_addinfo')";  
    $this->connection->sql->exec($saveMemberQuery);  
    // id of inserted member ---> participant id  
    $this->member_id = $this->connection->sql->lastInsertId();  
    // insert into participation  
    $saveParticipationQuery = "insert into participation  
        (event_id, member_id)  
        values  
        ($this->event_id, $this->member_id)";  
    if($this->connection->sql->exec($saveParticipationQuery)){  
        return 1;  
    }  
}
```

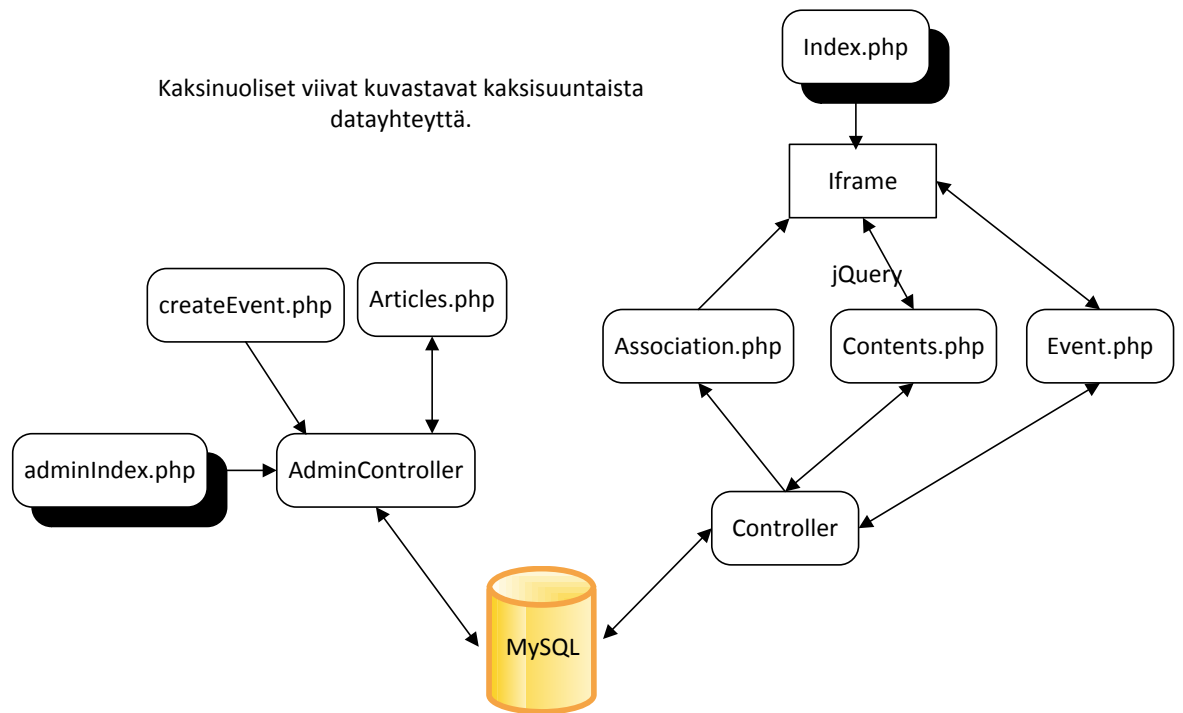
Iframe.css – framen sisällön tyylitiedosto

```
body{  
  
  font-family: 'Alegreya', serif;  
  font-size: 20px;  
  
}  
  
.event_wrapper{  
  
  position: absolute;  
  left: 5%;  
  width: 40%;  
  padding: 10px;  
  
}  
  
.event_wrapper table, p, h3, td{  
  
  padding: 10px;  
  font-size: 20px;  
  
}  
  
#enrollForm{  
  
  position: absolute;  
  width: 40%;  
  right: 5%;  
  font-size: 20px;  
  padding: 10px;  
  
}
```

```
#enrollForm td{  
  
  font-size: 20px;  
}  
  
#enrollForm input, textarea{  
  
  border: 2px solid rgba(0, 0, 100, 0.6);  
  background-color: rgba(100, 100, 100, 0.5);  
  display: block;  
  font-size: 20px;  
  
}  
  
h3{  
  
  font-size: 25px;  
  padding: 0;  
  
  display: inline;  
  
}  
  
#enrollButton{  
  
  position: absolute;  
  width: auto;  
  
}
```

```
#middleBorder{  
  
  position: fixed;  
  left: 580px;  
  width: 3px;  
  height: 95%;  
  
  background-color: #8e8a8a;  
}
```

Liite 4. Arkkitehtuurikuvaus



Liite 5. Käyttöönoton ohjeistus

Ohjeessa kuvataan Helsingin Uudenkaupungin kilta ry:n verkkopalvelun käyttöönoton vaiheet.

Palvelinympäristö

Palvelun käyttöä varten käyttäjällä tulee olla pääsy kotisivutilaan, jossa on vähintään PHP 5.0-tuki ja MySQL-tietokannan käyttömahdollisuus.

Tietokannan asennus

Palvelun lähdekoodin mukana toimittiin tiedosto kilta_db.sql, joka sisältää palvelun tietokannan luontilauseet. Webhotellien MySQL-käyttöliittymänä on yleensä phpMyAdmin. Sen navigaatiopalkissa on Tuo (tai Import)-linkki, josta pääsee työkaluun, jolla voit syöttää tietokannan luontitiedoston palvelimelle. Valitse kilta_db.sql ja paina Jatka (tai Go)-painiketta. Mikäli palveluntarjoajallasi on käytössä jokin muu kuin phpMyAdmin, voit syöttää luontilauseet myös yksitellen kopioimalla.

Tietokantayhteyden konfigurointi

Siirry palvelimen pääkansioon. Siirry kansioon utility-class ja avaa tekstieditorilla tiedosto sql.class.php. Sen alussa on seuraavat rivit:

```
private $host = '127.0.0.1';  
private $username = 'username';  
private $pass = 'salasana';  
  
private $dbName = 'kilta';
```

Vaihda kolmen ensimmäisen kentän (host, username, pass) tiedot palveluntarjoajasi toimittamiin tietoihin. Ensimmäinen kenttä on palveluntarjoajan palvelimen ip-osoite, toinen on käyttäjänimesi tietokantaan, ja kolmas salasanasasi tietokantaan.

Tiedostojen siirtäminen palvelimelle

Siirrä toimitetut tiedoston palvelimellesi joltain tiedostonsiirto-ohjelmaa käyttäen seuraavasti:

- kansion www sisältö menee palveluntarjoajasta riippuen kansioon htdocs tai public_html (vaihtoehtoja voi olla myös muita, jos näitä ei ole, tiedustele palveluntarjoajalta mihin pääsivu tulisi laittaa).

- siirrä loput tiedostot www-kansion sisällöstä yksi askel ylöspäin (siis siihen kansioon, missä äsken siirrettyjen tiedostojen www-kansio sijaitsee).

Hallinnan salasanan vaihtaminen

Hallinnan .htdocs-salasanan voi vaihtaa avaamalla public_html/admin/-kansiossa sijaitsevan .htdocs-tiedoston tekstieditorilla. Nykyiset tiedot näkyvät tiedostossa malliin käyttäjänimi : salasana. Vaihda tiedot ja paina CTRL + S. Uudet salasana-asetukset ovat nyt voimassa.

Liite 6. Tietokannan luontilauseet

PhpMyAdminista viedyt tietokannan luontilauseet ja testidatan syöttölauseet.

```
-- Tietokanta: `kilta`
```

```
CREATE DATABASE IF NOT EXISTS `kilta` DEFAULT CHARACTER SET utf8  
COLLATE utf8_general_ci;  
USE `kilta`;
```

```
Rakenne taululle `catering`
```

```
CREATE TABLE IF NOT EXISTS `catering` (  
  `catering_id` int(11) NOT NULL AUTO_INCREMENT,  
  `catering_name` varchar(28) COLLATE utf8_bin NOT NULL,  
  `catering_surname` varchar(56) COLLATE utf8_bin NOT NULL,  
  `catering_email` varchar(128) COLLATE utf8_bin NOT NULL,  
  `catering_menu_id` int(11) DEFAULT NULL,  
  `catering_time` time DEFAULT NULL,  
  `catering_addinfo` varchar(512) COLLATE utf8_bin DEFAULT NULL,  
  PRIMARY KEY (`catering_id`),  
  KEY `menu_fk` (`catering_menu_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin  
AUTO_INCREMENT=2 ;
```

```
--
```

```
-- Vedos taulusta `catering`
```

```
--
```

```
INSERT INTO `catering` (`catering_id`, `catering_name`, `catering_surname`, `cater-  
ing_email`, `catering_menu_id`, `catering_time`, `catering_addinfo`) VALUES  
(1, 'Kirsi', 'Puustinen', 'puustinen@fscatering.testmail', 1, '19:30:00', NULL);
```

--

-- Rakenne taululle `contact`

--

```
CREATE TABLE IF NOT EXISTS `contact` (  
  `contact_id` int(11) NOT NULL AUTO_INCREMENT,  
  `contact_name` varchar(28) COLLATE utf8_bin NOT NULL,  
  `contact_surname` varchar(56) COLLATE utf8_bin NOT NULL,  
  `contact_email` varchar(128) COLLATE utf8_bin NOT NULL,  
  `contact_phone` varchar(28) COLLATE utf8_bin NOT NULL,  
  PRIMARY KEY (`contact_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin  
AUTO_INCREMENT=2 ;
```

--

-- Vedos taulusta `contact`

--

```
INSERT INTO `contact` (`contact_id`, `contact_name`, `contact_surname`, `con-  
tact_email`, `contact_phone`) VALUES  
(1, 'Anne', 'Valsta', 'anne.valsta-haagahelia-fi', '04000000');
```

-- -----

--

-- Rakenne taululle `event`

--

```
CREATE TABLE IF NOT EXISTS `event` (  
  `event_id` int(11) NOT NULL AUTO_INCREMENT,  
  `event_name` varchar(128) COLLATE utf8_bin NOT NULL,  
  `event_descp` varchar(256) COLLATE utf8_bin NOT NULL,
```

```

`event_venue` varchar(128) COLLATE utf8_bin NOT NULL,
`event_date` varchar(28) COLLATE utf8_bin NOT NULL,
`event_enroll_dl` varchar(28) COLLATE utf8_bin NOT NULL,
`event_payment_id` int(11) DEFAULT NULL,
`event_contact_id` int(11) DEFAULT NULL,
`event_catering_id` int(11) DEFAULT NULL,
`event_venue_site` varchar(128) COLLATE utf8_bin DEFAULT NULL,
`event_direction` varchar(512) COLLATE utf8_bin DEFAULT NULL,
`event_parking` tinyint(1) DEFAULT NULL,
`event_transportation` varchar(256) COLLATE utf8_bin DEFAULT NULL,
`event_program` varchar(512) COLLATE utf8_bin DEFAULT NULL,
`event_addinfo` varchar(512) COLLATE utf8_bin DEFAULT NULL,
PRIMARY KEY (`event_id`),
KEY `contact_fk` (`event_contact_id`),
KEY `event_catering_fk` (`event_catering_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin
AUTO_INCREMENT=2 ;

```

```
--
```

```
-- Vedos taulusta `event`
```

```
--
```

```

INSERT INTO `event` (`event_id`, `event_name`, `event_descp`, `event_venue`,
`event_date`, `event_enroll_dl`, `event_payment_id`, `event_contact_id`, `event_cater-
ing_id`, `event_venue_site`, `event_direction`, `event_parking`, `event_transporta-
tion`, `event_program`, `event_addinfo`) VALUES
(1, 'Itsenäisyyspäivän juhlat/pikkujoulut', 'Tervetuloa itsenäisyyspäivän ja pikkujoulujen
viettoon!', 'Herttoniemen kartano', '6.12.2013', '22.11.2013', 1, 1, 1, 'http://hertonas-
gard.fi/suomi/kotisivu/', 'Poistu Itäväylältä Herttoniemen kohdalla ja käänny Laajasa-
lon suuntaan. Noin kilometrin ajon jälkeen kartanon parkkipaikka on tien vasemmalla
puolella.', 1, NULL, NULL, 'Pukukoodi: Formal');

```

```
-- -----
```

```
--
-- Rakenne taululle `member`
--

CREATE TABLE IF NOT EXISTS `member` (
  `member_id` int(11) NOT NULL AUTO_INCREMENT,
  `member_name` varchar(28) COLLATE utf8_bin NOT NULL,
  `member_surname` varchar(56) COLLATE utf8_bin NOT NULL,
  `member_email` varchar(128) COLLATE utf8_bin NOT NULL,
  `member_phone` varchar(28) COLLATE utf8_bin DEFAULT NULL,
  `member_addinfo` varchar(256) COLLATE utf8_bin DEFAULT NULL,
  PRIMARY KEY (`member_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin
AUTO_INCREMENT=14 ;

--
-- Vedos taulusta `member`
--

INSERT INTO `member` (`member_id`, `member_name`, `member_surname`,
`member_email`, `member_phone`, `member_addinfo`) VALUES
(10, 'Rauli', 'Karvinen', 'testimail@testimail.com', '0401231234', 'Tuon pelikortit'),
(11, 'Ernesti', 'Moilane', 'testimail@testimail.com', '0401231234', ''),
(12, 'Antti', 'Karhu', 'testimail@testimail.com', '0401231234', 'Olen myöhässä'),
(13, 'Tuomas', 'Kujala', 'tkujala2@gmail.com', '0401231234', 'Myöhässä');

-- -----
--
-- Rakenne taululle `menu`
--
```

```

CREATE TABLE IF NOT EXISTS `menu` (
  `menu_id` int(11) NOT NULL AUTO_INCREMENT,
  `menu_starter` varchar(56) DEFAULT NULL,
  `menu_main` varchar(128) DEFAULT NULL,
  `menu_dessert` varchar(56) DEFAULT NULL,
  PRIMARY KEY (`menu_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

--
-- Vedos taulusta `menu`
--

INSERT INTO `menu` (`menu_id`, `menu_starter`, `menu_main`, `menu_dessert`)
VALUES
(1, 'Rosépippurisilakkaa', 'Hirveä kahdella tavalla kukkakaalipyreen kera', 'Lakkapannacotta');

-----

--
-- Rakenne taululle `news`
--

CREATE TABLE IF NOT EXISTS `news` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(128) NOT NULL,
  `post` varchar(1056) NOT NULL,
  `posted` varchar(28) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=20 ;

--
-- Vedos taulusta `news`

```

--

```
INSERT INTO `news` (`id`, `name`, `post`, `posted`) VALUES
```

```
(1, 'Testi', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer vitae susci-  
pit lacus, vitae feugiat dolor. Nunc porta purus odio, vitae sollicitudin mi placerat at.  
Mauris nec mauris nunc. Sed faucibus ut velit a tincidunt. Sed nec blandit tellus.
```

```
Phasellus cursus eleifend scelerisque. Donec in mollis ante. Morbi ac tellus in dolor  
rhoncus accumsan. Nullam porttitor tincidunt ante, congue viverra arcu. Nullam ves-  
tibulum sit amet augue in viverra. Cum sociis natoque penatibus et magnis dis parturi-  
ent montes, nascetur ridiculus mus.', '2013-11-27 14:49:58'),
```

```
(17, 'Testiteksti 4', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque  
luctus erat quis justo viverra mattis. Vivamus a pulvinar nibh. Donec pellentesque nisl  
et turpis lobortis scelerisque. In hac habitass.', '30.11.2013 09:42'),
```

```
(18, 'Testiteksti 5', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer vi-  
tae suscipit lacus, vitae feugiat dolor. Nunc porta purus odio, vitae sollicitudin mi plac-  
erat at. Mauris nec mauris nunc. Sed faucibus ut velit a tincidunt. Sed nec blandit tellus.  
Phasellus cursus eleifend scelerisque. Donec in mollis ante. Morbi ac tellus in dolor  
rhoncus accumsan. Nullam porttitor tincidunt ante, congue viverra arcu. Nullam ves-  
tibulum sit amet augue in viverra. Cum sociis natoque penatibus et magnis dis parturi-  
ent montes, nascetur ridiculus mus.', '30.11.2013 10:40'),
```

```
(19, 'Testiteksti 6', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer vi-  
tae suscipit lacus, vitae feugiat dolor. Nunc porta purus odio, vitae sollicitudin mi plac-  
erat at. Mauris nec mauris nunc. Sed faucibus ut velit a tincidunt. Sed nec blandit tellus.  
Phasellus cursus eleifend scelerisque. Donec in mollis ante. Morbi ac tellus in dolor  
rhoncus accumsan. Nullam porttitor tincidunt ante, congue viverra arcu. Nullam ves-  
tibulum sit amet augue in viverra. Cum sociis natoque penatibus et magnis dis parturi-  
ent montes, nascetur ridiculus mus.', '30.11.2013 10:40');
```

--

```
-- Rakenne taululle `participation`
```

--

```

CREATE TABLE IF NOT EXISTS `participation` (
  `event_id` int(11) NOT NULL,
  `member_id` int(11) NOT NULL,
  KEY `participation_event_fk` (`event_id`),
  KEY `participation_memberid` (`member_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

```

--
-- Vedos taulusta `participation`
--

```

```

INSERT INTO `participation` (`event_id`, `member_id`) VALUES
(1, 10),
(1, 11),
(1, 12),
(1, 13);

```

```

-----

```

```

--
-- Rakenne taululle `participation_price`
--

```

```

CREATE TABLE IF NOT EXISTS `participation_price` (
  `price_id` int(11) NOT NULL AUTO_INCREMENT,
  `amount` int(11) NOT NULL,
  `payment_method` varchar(128) COLLATE utf8_bin NOT NULL DEFAULT
'Tapahtuman yhteydessä',
  PRIMARY KEY (`price_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin
AUTO_INCREMENT=24 ;

```

```

--
-- Vedos taulusta `participation_price`
--

INSERT INTO `participation_price` (`price_id`, `amount`, `payment_method`) VAL-
UES
(1, 50, 'Tapahtuman yhteydessä');
--
-- Rajoitteet vedostauluille
--

--
-- Rajoitteet taululle `catering`
--

ALTER TABLE `catering`
  ADD CONSTRAINT `menu_fk` FOREIGN KEY (`catering_menu_id`) REFER-
ENCES `menu` (`menu_id`) ON DELETE CASCADE ON UPDATE NO AC-
TION;

--
-- Rajoitteet taululle `event`
--

ALTER TABLE `event`
  ADD CONSTRAINT `event_catering_fk` FOREIGN KEY (`event_catering_id`)
REFERENCES `catering` (`catering_id`) ON DELETE CASCADE ON UPDATE
NO ACTION,
  ADD CONSTRAINT `contact_fk` FOREIGN KEY (`event_contact_id`) REFER-
ENCES `contact` (`contact_id`) ON DELETE CASCADE ON UPDATE NO AC-
TION;

--
-- Rajoitteet taululle `participation`
--

```



```
ALTER TABLE `participation`  
  ADD CONSTRAINT `participation_memberid` FOREIGN KEY (`member_id`)  
REFERENCES `member` (`member_id`);
```