



Ajax-verkkosovellukset

Timeo-ajanseurantasovellus

Viestinnän koulutusohjelma
Verkkoviestintä
Opinnäytetyö
28.5.2009

Sami Laakkonen

TIIVISTELMÄSIVU

Koulutusohjelma Viestinnän koulutusohjelma		Suuntautumisvaihtoehto Verkkoviestintä	
Tekijä Sami Laakkonen			
Työn nimi Ajax-verkkosovellukset. Timeo-ajanseurantasovellus			
Työn ohjaaja/ohjaajat Matti Rantala			
Työn laji Opinnäytetyö		Aika 28.5.2009	Numeroidut sivut + liitteiden sivut 43
<p>TIIVISTELMÄ</p> <p>Opinnäytetyö käsittelee Ajax-tekniikkaan perustuvia verkkosovelluksia sekä niihin liittyviä asioita Timeo-verkkosovelluksen kautta. Aluksi kerrotaan verkkosovellusten kehityksestä ja teknologioista sekä verkkosovellusten käytettävyydestä. Tämän jälkeen käydään läpi Timeo-ajanseurantaverkkosovellusta ja sen rakennetta. Lopuksi kerrotaan vielä Timeon jatkokehityssuunnitelmista ja sen julkaisusta avoimena lähdekoodina.</p> <p>Ajax-tekniikkaan perustuva verkkosovellus on perinteistä tietokoneohjelmaa muistuttava sovellus, jota käytetään selainohjelmalla. Verkkosovellus on toteutettu selainten tukemilla teknologioilla, kuten JavaScriptillä, HTML:llä ja CSS:llä. Verkkosovellus sijaitsee yleensä yhdellä palvelimella, josta käyttäjät käyttävät sitä selaimella Internetin välityksellä omilla käyttäjätunnuksillaan. Työssä käsitellään Ajax-tekniikkaan perustuvia verkkosovelluksia, jotka hyödyntävät selainten itsessään tukemia teknologioita, ilman tarvetta asentaa selaimelle erillisiä lisäosia, kuten Adobe Flash Playeria.</p> <p>Esimerkkinä Ajax-tekniikkaan perustuvasta verkkosovelluksesta käytetään Timeo-ajanseurantasovellusta. Timeo on ajanseurantaan suunniteltu avoimen lähdekoodin verkkosovellus, jolla voidaan pitää tuntiseurantaa projekteihin kuluneesta ajasta. Timeo-projekti aloitettiin nimellä Kellota.fi. Nimi vaihdettiin myöhemmin Timeoksi, kun projekti päätettiin julkaista avoimena lähdekoodina. Laatutavoitteista johtuen Timeo-projektin aikataulu venyi moneen kertaan, eikä kaikkia suunniteltuja ominaisuuksia saatu valmiiksi tätä opinnäytetyötä kirjoittaessa. Laatuun panostaminen kuitenkin kannatti, sillä Timeon valmiiksi saadut perusominaisuudet toimivat paremmin kuin jos projektiin olisi käytetty vähemmän aikaa. Perusominaisuuksien ollessa kunnossa, projektin jatkaminen tulevaisuudessa on myös helpompaa.</p>			
Teos/Esitys/Produktio			
Säilytyspaikka Taideteollisen korkeakoulun kirjasto, Aralis-kirjastokeskus			
Avainsanat verkkosovellus, Ajax, jQuery			

Degree Programme in Media		Specialisation New Media Design	
Author Sami Laakkonen			
Title Ajax Web Applications. Timeo Time Tracking Application			
Tutor(s) Matti Rantala			
Type of Work Bachelor´s Thesis	Date 28 May, 2009	Number of pages + appendices 43	
<p>This thesis is about Ajax-based web applications, their technologies, history and usability. The author is using a time tracking application named Timeo as an example of an Ajax-based web application. Timeo is a simple open source application that allows users to track time and see how long they have been working on their projects and tasks. Timeo is an Ajax application that uses an Ajax engine between user and the server. The engine allows the user's interaction with the application to happen asynchronously. Using Ajax in a web application facilitates a better user experience and it may even improve the performance. Timeo's Ajax engine and user-interface logic is written in JavaScript which is a popular programming language for the web. A JavaScript library named jQuery was also used in Timeo to make developing JavaScript easier.</p> <p>Ajax-based web applications are applications that are accessed via the web browser on the Internet or other network. The use of Ajax allows the development of web applications that are highly interactive and feel more like traditional desktop applications. Ajax refers to the use of several technologies that browsers already support. This thesis is not about Flash applications or other web applications that require a separate plug-in to be installed on the browser. Web applications usually exist in one server. The users use their browser to access the application over an Internet. They usually need to log in with their username and password to use the application.</p> <p>Timeo-project was first named Kellota.fi. Later, the project was decided to be released as an open source and the name was changed to Timeo. The goal was to make Timeo as high-quality as possible, and so the project took longer than originally scheduled. But it was worth it, because Timeo ended up being more robust. All the basic features were done well, so it will be easier to continue with the project in the future.</p>			
Work / Performance / Project			
Place of Storage Aralis Library and Information Center, Helsinki			
Keywords web application, Ajax, jQuery			

SISÄLLYS

1	JOHDANTO	2
2	AJAX-VERKKOSOVELLUKSET	3
2.1	Verkkosovellusten kehitys.....	4
2.2	Document Object Model.....	5
2.3	Ajax	7
2.4	Synkroninen ja asynkroninen tiedonsiirto	8
2.5	jQuery	10
2.6	Käytettävyys	12
2.7	Käytettävyys mobiililaitteissa.....	14
3	TIMEO	16
3.1	Navigointirakenne ja sivut	17
3.1.1	Uusien käyttäjien rekisteröintisivu	19
3.1.2	Käyttäjien sisäänkirjautumissivu	23
3.1.3	Timeon pääsivu	24
3.2	Timeon käyttöönotto ja tiedostot	28
3.3	Tietokantarakenne.....	32
4	TIMEON JATKOKEHITYS.....	35
4.1	Timeon kehitys omaksi sovellukseksi eri alustoille	35
4.2	Timeon julkaiseminen avoimena lähdekoodina	37
5	YHTEENVETO	41
	LÄHTEET.....	42

1 JOHDANTO

Käsittelen opinnäytteessäni Ajax¹-tekniikkaan perustuvia verkkosovelluksia. Ajax-tekniikkaa hyödyntävien verkkosivujen ja verkkosovellusten määrä kasvaa nopeasti. Käsittelen vain Ajax-tekniikkaan ja muihin nykyaikaisissa selaimissa jo valmiina oleviin teknologioihin perustuvia verkkosovelluksia, jotka eivät vaadi selaimelle asennettavia lisäosia. En käsittele esimerkiksi Adobe Flash-teknologiaan perustuvia verkkosovelluksia tai muita verkkosovelluksia, jotka vaativat selaimen asennettavia lisäosia. Käytän esimerkkinä ajanseurantaan suunniteltua Timeo-verkkosovellusta.

Timeo on verkossa toimiva helppokäyttöinen ja ilmainen ajanseurantatyökalu, jolla voidaan pitää yksinkertaista tuntikirjanpitoa työ- tai esimerkiksi koulutehtäviin käytetystä ajasta. Timeo-projekti aloitettiin syksyllä 2008 aluksi nimellä Kellota.fi. Alkuperäinen idea oli, että projektin tuloksena valmistuisi palvelu Internet-sivustona, mutta suunnitelmista luovuttiin ja nimi vaihdettiin samalla Timeoksi. Opinnäytettäni kirjoittaessa Timeo ei ollut vielä täysin valmistunut. Timeo tullaan uuden suunnitelman mukaan julkaisemaan avoimena lähdekoodina sen valmistuttua julkaisukelpoiseksi kesällä 2009. Käsittelen vielä Timeon jatkokehityssuunnitelmia ja julkaisua avoimena lähdekoodina opinnäytteeni lopussa.

1 Asynchronous JavaScript + XML

Käsittelen aluksi yleisesti verkkosovelluksia, jonka jälkeen käyn tarkemmin läpi Ajax-tekniikkaan perustuvan verkkosovelluksen mahdollista rakennetta käyttäen esimerkkinä Timeo-verkkosovellusta. Verkkosovellusten rakenne vaihtelee sovelluksen mukaan, eikä yhtä yleistä sääntöä Ajax-pohjaisten verkkosovellusten toteuttamiseen ole. Ajax-tekniikkaan perustuvilla verkkosovelluksilla on kuitenkin toisiaan muistuttavia piirteitä. Siksi tästä opinnäytetyöstä voi olla hyötyä niille, jotka ovat mukana vastaavanlaisissa verkkosovellusprojekteissa.

2 AJAX-VERKKOSOVELLUKSET

Verkkosovellus on sovellus, jota käytetään selainohjelmassa verkon, kuten Internetin välityksellä. Verkkosovellus on tietokoneohjelma, joka on ohjelmoitu selaimen tukemalla kielellä, kuten HTML²:llä, Javalla tai JavaScriptillä. Yleinen verkkosovellus on esimerkiksi webmail³. (Web application 2009.)

Verkkosovelluksen rakenne voidaan nähdä jakautuneen kahteen osaan, asiakaspuoleen ja palvelinpuoleen. Asiakaspuolen teknologioita ovat HTML, CSS⁴ ja JavaScript. Ne toimivat selaimessa ja mahdollistavat verkkosovelluksen käyttöliittymän toteuttamisen. Palvelinpuolen teknologioita ovat esimerkiksi paljon käytetyt PHP⁵-skriptauskieli ja MySQL-tietokanta. Verkkosovelluksen toimintalogiikka, HTML-sisällön tuottaminen ja prosessointi voivat jakautua asiakaspuolella JavaScriptin ja palvelinpuolella teknologioiden kuten PHP:n hoidettavaksi. Käsittelen tätä verkkosovelluksen rakennemallia myöhemmin kuvaillessani Timeon rakennetta. On olemassa useita rakennemalleja siitä, miten verkkosovelluksia ja muita tietokoneohjelmia voidaan tarkastella. Verkkosovellusten rakenne voidaan usein jakaa myös useampaan kuin kahteen osaan, mutta päädyin kaksiosaiseen malliin, sillä se kuvaa selkeästi Timeon rakennetta. Timeo on kuitenkin rakenteeltaan melko yksinkertainen.

2 Hypertext Markup Language.

3 Sähköpostisovellus, jota käytetään selaimella.

4 Cascading Style Sheets.

5 Hypertext Preprocessor.

2.1 Verkkosovellusten kehitys

Verkkosivut olivat aluksi pelkkiä tekstidokumenttien sähköisiä kopioita, eivätkä ne sisältäneet dynaamista sisältöä, vaan olivat sisällöltään staattisia. Verkkosivujen mahdollisuuksia ei vielä osattu täysin hyödyntää. Ei mennyt kauan kun Internetin käyttäjät alkoivat vaatia dynaamisempia verkkosivuja. Uusien tietokoneohjelmien julkaisun myötä, myös odotukset ja vaatimukset verkkosivuja kohtaan kasvoi. (Asleson & Schutta 2006, 3.)

CGI⁶ on ensimmäisiä ratkaisuja tehdä Internetistä dynaamisempi. Se sallii ohjelmien suorittamisen palvelimella käyttäjän pyynnöstä. CGI-skripti voi luoda esimerkiksi yhteyden tietokantaan ja näyttää tietokantahaun tulokset. Käyttämällä HTML-lomakkeita ja CGI-skriptejä voidaan toteuttaa selaimessa toimiva näyteikkuna, jonka avulla voidaan myydä tuotteita käyttäjille. CGI ei kuitenkaan ole turvallisin tapa luoda dynaamisia verkkosivuja, sillä sen avulla voi olla mahdollista suorittaa palvelimella jokin ei-haluttu ohjelma. Ongelmista huolimatta, CGI:tä käytetään vielä nykyäänkin. (Asleson & Schutta 2006, 3.)

Vuonna 1995 julkistettiin Java-niminen ohjelmointikieli. Dynaamisten verkkosivujen valtakausi alkoi, kun Netscape Navigator -selain sai Java-tuen. Tämä mahdollisti Java-sovelmien turvallisen suorittamisen selaimen sisällä. Sovelmat olivat suosittuja kuitenkin vain lyhyen aikaa. Sovelmien kehitystyö oli hankalaa, sillä oli otettava huomioon yleisin käytössä oleva Java-versio. Huonosti toteutetut Java-sovellukset veivät myös käyttäjien uskoa sovellusten käyttöön verkkosivuilla. (Asleson & Schutta 2006, 4.)

Javan rinnalle kehitettiin JavaScript-skriptauskieli. Sen tarkoitus oli tehdä sovellusten kehittäminen helpommaksi, varsinkin niille kehittäjille, joille Java oli uusi asia. JavaScriptin alkuperäisenä tarkoituksena oli helpottaa HTML-kielen tagien⁷ dynaamista muokkausta. Näin verkkosivujen käyttökokemus käyttäjille paranisi. JavaScriptin kehittymisen myötä kehittyi myös DOM⁸, joka on tapa esittää verkkosivu

6 Common Gateway Interface.

7 HTML-elementti.

8 Document Object Model.

oliokeskeisessä muodossa. JavaScriptin yleistymisen verkkosivuilla ei ole tapahtunut mutkitta. JavaScriptin selaintuki oli aluksi puutteellista ja on sitä osittain nykyäänkin. Jotkin skriptit saattavat toimia eri tavoin eri selaimilla. JavaScriptin turvallisuuspuutteetkin ovat saaneet useat käyttäjät asettamaan selaimestaan JavaScript-tuen pois päältä. JavaScript-sovellusten kehittäminen voi olla myös hankalaa, sillä JavaScript-koodin testaaminen ja virheiden jäljittäminen on monimutkaista, minkä takia monet kehittäjät välttävät JavaScriptin käyttöä. (Asleson & Schutta 2006, 5.)

2.2 Document Object Model

Document Object Model (DOM) on ohjelmointirajapinta, joka mahdollistaa HTML-dokumenttien sisällön muokkauksen esimerkiksi JavaScriptillä. DOM-malli on puumainen tapa esittää tietoa ja tiedon rakenteita. Se on myös standardoitu tapa käsitellä dokumentteja. DOM ei ole kielisidonnainen, vaikka se suunniteltiin tavaksi tehdä JavaScript-koodista siirrettävä eri selainten välillä. DOM on tapa, jolla JavaScript näkee HTML-dokumentin ja jolla JavaScript-skriptit voivat tarkastella ja muokata dynaamisesti dokumenttia ilman dokumentin uudelleenlatausta. Oliokeskeisen suunnittelun näkökulmasta DOM-malli on aito oliomalli. Se määrittelee olioita, joita tarvitaan dokumentin esittämiseen ja muokkaamiseen. Se määrittelee kyseisten olioiden käytöksen ja ominaisuudet sekä olioiden väliset suhteet. (Document Object Model 2009; Asleson & Schutta 2006, 36–37.)

Kuviossa 1 on esimerkki DOM-mallin mukaisen HTML-dokumentin lähdekoodista. Lähdekoodin rivin 1 ja 11 välissä määritellään HTML-dokumentin sisältö. Tagien <body> ja </body> välissä määritellään HTML-dokumentin varsinaisen käyttäjälle näkyvän sisällön elementit, eli objektit. Rivillä 3 määritellään kuviossa 2 esitetty Otsikko-objekti. Rivillä 4 määritellään Tekstiä-objekti. Rivillä 5 määritellään Linkki-objekti. Riveillä 6–9 määritellään Lomake-objekti ja sen sisältämät objektit, jotka ovat etunimi- ja sukunimi-kenttä.

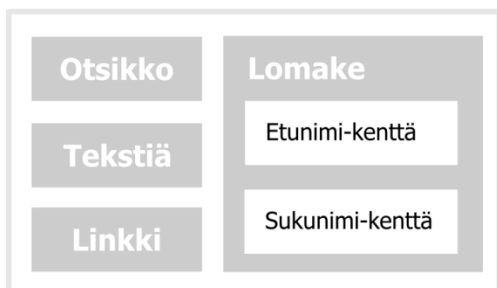
```

1 <html>
2   <body>
3     <h1>Timeo</h1>
4     <p>Tässä on tekstiä.</p>
5     <a href="#">linkki</a>
6     <form>
7       Etunimi: <input type="text" name="etunimi" /><br />
8       Sukunimi: <input type="text" name="sukunimi" />
9     </form>
10  </body>
11 </html>

```

Kuvio 1. Esimerkki DOM-mallin mukaisen HTML-dokumentin lähdekoodista.

HTML-dokumentti



Kuvio 2. Esimerkki objektien hierarkiasta DOM-mallin mukaisessa HTML-dokumentissa. Objektit Otsikko, Tekstiä, Linkki ja Lomake sijaitsevat HTML-dokumentin sisällä. Etunimi- ja Sukunimi-kentät sijaitsevat Lomake-objektin sisällä.

2.3 Ajax

Vaikka JavaScript ei ole aiemmin ollut kehittäjien suuressa suosiossa, sen käyttö on nykyään yleistynyt Ajax-tekniikan myötä. Ajax-tekniikka mahdollistaa dynaamisten verkkosovellusten luomisen ja yhteyden palvelimeen, ilman verkkosivun koko sisällön uudelleen lataamista. Verkkosivuille voidaan toteuttaa Ajax-tekniikalla tavallisia tietokoneohjelmia muistuttavia sovelluksia. Tällainen sovellus on esimerkiksi Google Docs -dokumenttienkäsittelyohjelma, joka muistuttaa tavallista tietokoneen toimisto-ohjelmistoa. Perinteisten verkkosivujen ongelmana on ollut, että aina kun haetaan ja esitetään uutta tietoa käyttäjälle, verkkosivu on pitänyt ladata palvelimelta selaimeen kokonaan uudelleen. Käyttäjä on joutunut odottamaan sivun latautumista. Tämä on ollut usein dynaamisten verkkosivujen toteutuksen harmina. Ajax-tekniikalla voidaan hakea palvelimelta tietoa ja esittää se suoraan verkkosivun tietyssä osassa ilman sivun kokonaan uudelleen latausta selaimeen. Näin Ajax on mahdollistanut perinteisiä tietokoneohjelmia muistuttavien sovellusten vaatiman käyttäjäystävällisyyden toteuttamisen.

Ajax on merkittävä tekniikka, joka mahdollistaa monipuoliset verkkosovellukset. Sen keskeisin komponentti on JavaScript. Ajaxin etuna on, että se ei vaadi selaimeen asennettavia erillisiä lisäosia. Ajax rakentuu jo olemassa olevien tekniikoiden varaan. Google on yksi merkittävimmistä Ajaxin hyödyntäjistä. Google on tehnyt Ajax tekniikkaa tutuksi hyödyntämällä sitä omissa verkkosovelluksissaan, kuten Google Mapsissa⁹, Google Suggestissa¹⁰ ja Gmailissa¹¹. Termi Ajax tulee yhdistelmästä Asynchronous JavaScript ja XML¹². Nykyään termiä Ajax käytetään kuvaamaan kaikkia niitä teknologioita joiden avulla haetaan tietoa palvelimelta, ilman sivun uudelleenlatausta selaimeen. (Asleson & Schutta 2006, 13–15)

9 Googlen tuottama karttapalvelu, jolla voi löytää ja katsoa satelliittikarttoja.

10 Mahdollistaa hakutulosten automaattisen täydentämisen, kun kirjoitetaan suosittuja hakusanoja.

11 Googlen sähköpostiverkkosovellus.

12 Extensible Markup Language.

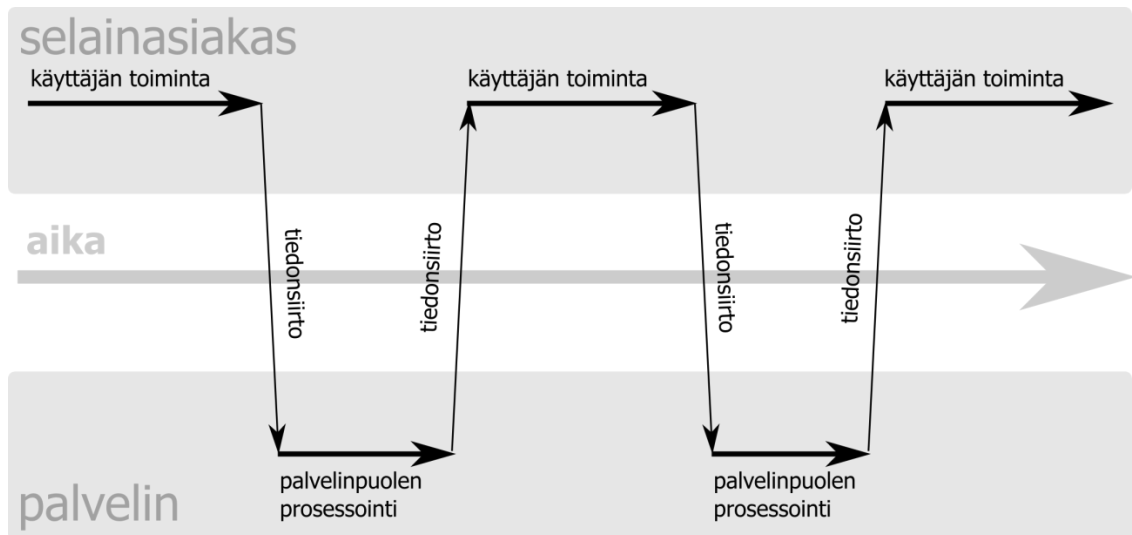
Ajax ei ole yksi teknologia, vaan se koostuu useammasta teknologiasta. Ajaxin takana ovat standardeihin perustuvat verkkosivun esitysteknologiat XHTML¹³, CSS, DOM, datan vaihdon hoitava XML sekä asynkronisen tiedon haun hoitava XMLHttpRequest. JavaScript sitoo kaikki nämä teknologiat yhteen. Ajax koostuu teknologioista, jotka ovat kehittäjille entuudestaan tuttuja. Ajaxin ydinteknologiat ovat myös kehittyneitä ja vakaita. (Garrett 2005.)

XMLHttpRequest (XHR) on tärkeä osa Ajax-tekniikkaa ja kaikki yleisimmät selaimet tukevat sitä. XHR on DOM-ohjelmointirajapinta, jota voidaan käyttää selainten skriptauskielillä, kuten JavaScriptillä. XHR mahdollistaa asynkronisen kommunikoinnin selaimen ja palvelimen välillä. Sen avulla voidaan lähettää HTTP-pyyntö suoraan palvelimelle ja saada vastaus teksti- tai XML-muodossa suoraan skriptauskieleen ilman koko verkkosivun uudelleen lataamista. (XMLHttpRequest 2009; Asleson & Schutta 2006, 23)

2.4 Synkroninen ja asynkroninen tiedonsiirto

Perinteisen verkkosovelluksen malli (ks. kuvio 3) toimii siten, että useimmat käyttäjän toimet verkkosivun käyttöliittymässä laukaisevat HTTP-pyyntöä takaisin palvelimelle. Palvelin suorittaa prosessointia tai esimerkiksi tiedonhakuja tietokannasta ja lähettää sen jälkeen HTML-sivun takaisin selaimen, eli selainasiakkaalle. Sillä välin kun palvelin suorittaa prosessointia ynnä muuta, käyttäjä joutuu odottamaan. Tämä synkroninen malli on peräisin Internetin alkuperäisestä käyttötarkoituksesta toimia vain yksinkertaisten sähköisten dokumenttien välittäjänä. (Garrett 2005.)

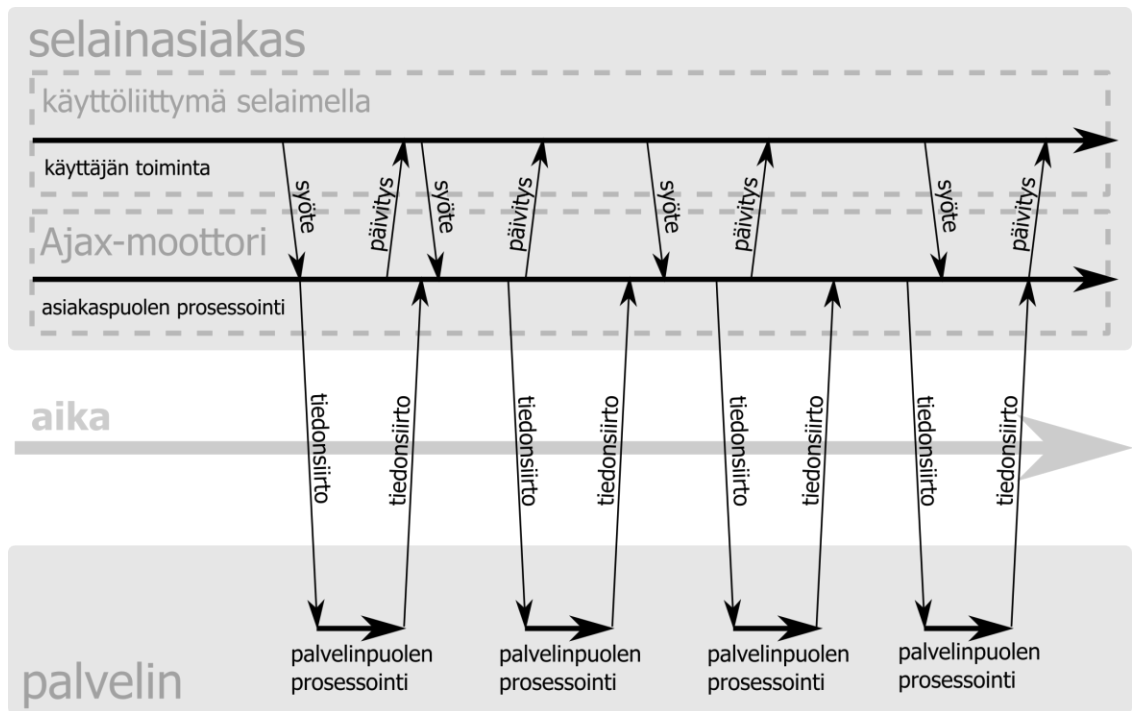
13 Extensible Hypertext Markup Language.



Kuvio 3. Perinteisen verkkosovelluksen synkroninen malli (Garrett 2005, opinnäytetyön tekijän suomennos).

Perinteinen malli ei kuitenkaan mahdollista verkkosovellukselle hyvää käyttäjäkokemusta, sillä käyttäjä joutuu odottamaan aina kun palvelin esimerkiksi prosessoi tietoa. Käytettävyyden kannalta käyttäjän ei tarvitsisi edes tietää milloin verkkosovellus on yhteydessä palvelimeen. Ajax-verkkosovellusten malli (ks. kuvio 4) muuttaa perinteisen verkkosovelluksen vuorovaikutuksen luonnetta. Se lisää JavaScriptillä toimivan Ajax-moottorin käyttäjän ja palvelimen välille. Pelkän verkkosivun sijasta selain lataa Ajax-moottorin, joka luo käyttöliittymän sekä hoitaa kommunikoinnin palvelimen kanssa. Tämä mahdollistaa käyttäjän vuorovaikutuksen asynkronisesti verkkosovelluksen kanssa riippumatta kommunikoinnista palvelimen kanssa. Käyttäjän ei tarvitse odottaa silloin, kun palvelin tekee jotakin. Jokainen käyttäjän toimi, joka normaalisti laukaisisi HTTP-pyyynnön, saa aikaan JavaScript-kutsun Ajax-moottoriin. Kaikki toiminnot hoitaa Ajax-moottori vastauksena käyttäjän toimiin, jotka eivät välttämättä vaadi yhteyttä palvelimeen. Tällaisia ovat esimerkiksi yksinkertainen tiedon validointi¹⁴, tiedon muokkaus muistissa tai jopa osa navigoinnista. Jos Ajax-moottori tarvitsee jotain palvelimelta, se hoitaa pyynnöt asynkronisesti vaikuttamalla mahdollisimman vähän käyttäjän ja verkkosovelluksen väliseen vuorovaikutukseen. (Garrett 2005.)

¹⁴ Validointi on prosessi, jossa tarkistetaan, että prosessin kohde täyttää tietyt kriteerit.



Kuvio 4. Ajax-verkkosovelluksen asynkroninen malli (Garrett 2005, opinnäytetyön tekijän suomennos).

2.5 jQuery

jQuery on ilmainen ja kevyt JavaScript-kirjasto, joka tehostaa JavaScriptin ja HTML:n vuorovaikutusta. Sen ominaisuuksiin kuuluu muun muassa CSS-tyylitietojen muuttaminen, Ajax, efektit ja animaatiot. jQueryn filosofia on, että se erottaa toiminnallisuuden määrittämisen HTML-rakenteesta, aivan kuten CSS erottaa ulkoasun määrittämisen HTML-rakenteesta. jQueryyn on saatavilla myös omia laajennuksia. (jQuery 2009.)

jQuery otettiin käyttöön Timeo-projektissa, sillä se vähentää JavaScript-koodin kirjoittamisen tarvetta. Lisäksi se auttaa selainten yhteensopivuusongelmissa. Eri selainten JavaScript-tuki on vaihtelevaa, jonka vuoksi JavaScript-koodi ei toimi aina halutulla tavalla kaikilla selaimilla. Tällöin koodiin on tehtävä muutoksia, jotka yleensä johtavat kompromisseihin tai verkkosovellukseen suunniteltujen ominaisuuksien vähentämiseen, jos ominaisuuksia ei voida toteuttaa kaikille selaimille. Voi olla mahdollista löytää keino toteuttaa jokin JavaScript-koodin ominaisuus kaikissa selaimissa, mutta se vaatii usein ylimääräistä työtä. jQuery-kirjastossa olevien funktioiden toiminta on testattu kaikissa yleisimmissä selaimissa. jQueryn käyttäminen

säästää aikaa ja vaivaa testauksen osalta. Uusien selainversioiden ilmestyessä jQueryn kehitysryhmä testaa ne ja tekee tarvittavat muutokset jQueryn funktioihin, jotta funktiot toimivat selainten uusilla versioilla. Parhaimmassa tapauksessa Timeo-projektissa säästyään selainten testaamiselta ja koodin korjaamiselta, kun vain otetaan käyttöön uusiin versio jQuery-tiedostosta.

jQuery otetaan käyttöön lataamalla sen js-päätteinen JavaScript-tiedosto omalle palvelimelle esimerkiksi jQueryn omilta verkkosivuilta¹⁵. Tiedosto sisältää jQueryn JavaScript-koodin ja se on muokattavissa tavallisella tekstieditorilla. jQueryn jquery.js-tiedosto linkitetään verkkosivulle sivun HTML-rakenteen HEAD-osiossa, jolloin jQuery on käytettävissä kaikkialla sivun JavaScript-koodissa. Kuviossa 5 rivillä 4 on HTML-kielinen esimerkki jquery.js-tiedoston linkittämisestä HTML-dokumenttiin.

```
1 <head>
2   <title>Esimerkkisivu</title>
3   <link rel="stylesheet" href="tyyli.css" type="text/css" />
4   <script type="text/javascript" src="jquery.js"></script>
5 </head>
```

Kuvio 5. Esimerkki jquery.js-tiedoston linkittämisestä HTML-sivun HEAD-osiossa.

jQueryn JavaScript-tiedostosta on jaossa kahta eri tyyppiä, pakattu ja pakkaamaton. Pakattu tiedosto soveltuu hyvin suoraan käyttöönotettavaksi ja se on pienempi kooltaan. Pakatussa tiedostossa JavaScript-koodi on pakattu eräänlaisella pakkausalgoritmilla, joka poistaa muun muassa tyhjät välilyönnit, rivinvaihdot ja turhat merkit JavaScript-koodista sekä lyhentää koodissa esiintyvien muuttujien nimet. Tällaista koodia on jälkikäteen lähes mahdotonta lukea, jos haluaa tehdä esimerkiksi muutoksia jQueryn sisältämiin funktioihin. Pakkausta tehdään sen takia, että verkkosivu, joka käyttää pakattua jQuery-tiedostoa, latautuu Internetin välityksellä nopeammin kuin jos se käyttäisi pakkaamatonta tiedostoa. Pakkaamaton jQuery-tiedosto sopii esimerkiksi kehittäjälle, joka haluaa valita mitä jQuery-funktioita hän tarvitsee. Käyttämättömät funktiot voidaan ottaa pois viemästä tilaa. Myöhemmin

15 <http://www.jquery.com>

kehittäjä voi pakata tiedoston itse, jolloin se vie vähemmän tilaa kuin jQueryn kotisivuilla jaettu pakattu tiedosto.

2.6 Käytettävyys

Ajax-tekniikkaa hyödyntävät verkkosivut eivät toimi perinteisten staattisten verkkosivujen tapaan, mikä voi aiheuttaa hämmennystä joillekin käyttäjille. Verkkosivuja varten kehitetyt teknologiat on alun perin suunniteltu staattisia sivuja varten. Internetin käyttäjät ovat tottuneet käyttämään verkkosivuja tietyllä tavalla. Esimerkiksi Takaisin-painike, joka löytyy kaikista selaimista, ei välttämättä vie käyttäjää oletetulle edelliselle sivulle, jos verkkosivu hyödyntää Ajax-tekniikkaa siten, että eri sivut on ladattu kaikki samalla kertaa. Verkkosivu saattaa näyttää dynaamisesti vain yhden sivun kerrallaan. Tällöin selain tunnistaa verkkosivun vain yhtenä sivuna, vaikka verkkosivu sisältäisi käytännössä useamman sivun. Selaimen Kirjamerkki-toiminto ei myöskään osaa aina tunnistaa oikeaa sivua. Kaikkien sivujen lataaminen yhdellä kertaa tekee kuitenkin verkkosivusta käyttäjäystävällisemmän siinä mielessä, että sivun vaihtaminen tapahtuu nopeasti, ilman eri sivujen uudelleenlatausta.

Monet käyttäjät ovat tottuneet verkkosivujen uudelleenlatautumiseen aina kun sivuilla tapahtuu muutos. He eivät välttämättä kiinnitä huomiota Ajax-tekniikkaa hyödyntävillä verkkosivuilla tapahtuviin sisällön muutoksiin, joilla muutokset päivittyvät dynaamisesti ilman sivun uudelleenlatausta. Tämä on johtanut sellaisten tekniikoiden syntyyn kuten Yellow Fade Technique. Se näyttää käyttäjälle visuaalisen vihjeen muuttuneesta sivun osasta värjäämällä sen hetkeksi keltaiseksi. Jos verkkosivun ulkoasu ei ole keltainen, käyttäjä huomaa vihjeen helposti. Ajan kuluessa keltainen väri häivytetään pikkuhiljaa takaisin värjätyn kohteen alkuperäiseen väriin. Vihjeväri voi olla myös jokin muu kuin keltainen. Toinen tapa vihjata käyttäjää muutoksesta on ilmoittaa siitä esimerkiksi tekstillä, joka ilmestyy sivun laitaan. (Asleson & Schutta 2006, 16.)

Verkkosovelluksia käytettäessä, vältetään työasemille jaettavista perinteisistä tietokoneohjelmien asennuspaketeista. Verkkosovelluksia ei tarvitse asentaa erikseen joka työasemalle, sillä verkkosovellukset sijaitsivat usein vain yhdellä palvelimella. Riittää että työasemilla on Internet-yhteys ja tarpeeksi uusi selainohjelma, jonka kautta verkkosovellusta voidaan käyttää. Internet-yhteys ja selain löytyvätkin melkein jokaiselta nykyaikaiselta tietokoneelta. Verkkosovellusten päivittäminen on helppoa, sillä riittää että verkkosovelluksen uusin versio asennetaan yhdelle palvelimelle, eikä jokaiselle työasemalle erikseen. Nykyaikaiset selaimetkin päivittävät itsensä lähes automaattisesti, joten verkkosovellusten käyttäjien ei tarvitse välttämättä ollenkaan murehtia ohjelmien asentamisesta tai päivittämisestä.

Verkkosovelluksilla on alhainen käyttöönottokynnys, sillä niiden käyttö voi olla vain yhden linkin painamisen päässä. Verkkosovellusten käyttö ei edellytä suurien tiedostojen lataamista ja asentamista, kuten tavalliset tietokoneohjelmat. Verkkosovellukset toimivat selaimessa, joten ne eivät ole riippuvaisia käyttöjärjestelmästä. Näin kehittäjät voivat saavuttaa myös eri käyttöjärjestelmien käyttäjät, kuten Linuxin tai OS X:n käyttäjät. (Asleson & Schutta 2006, 12.)

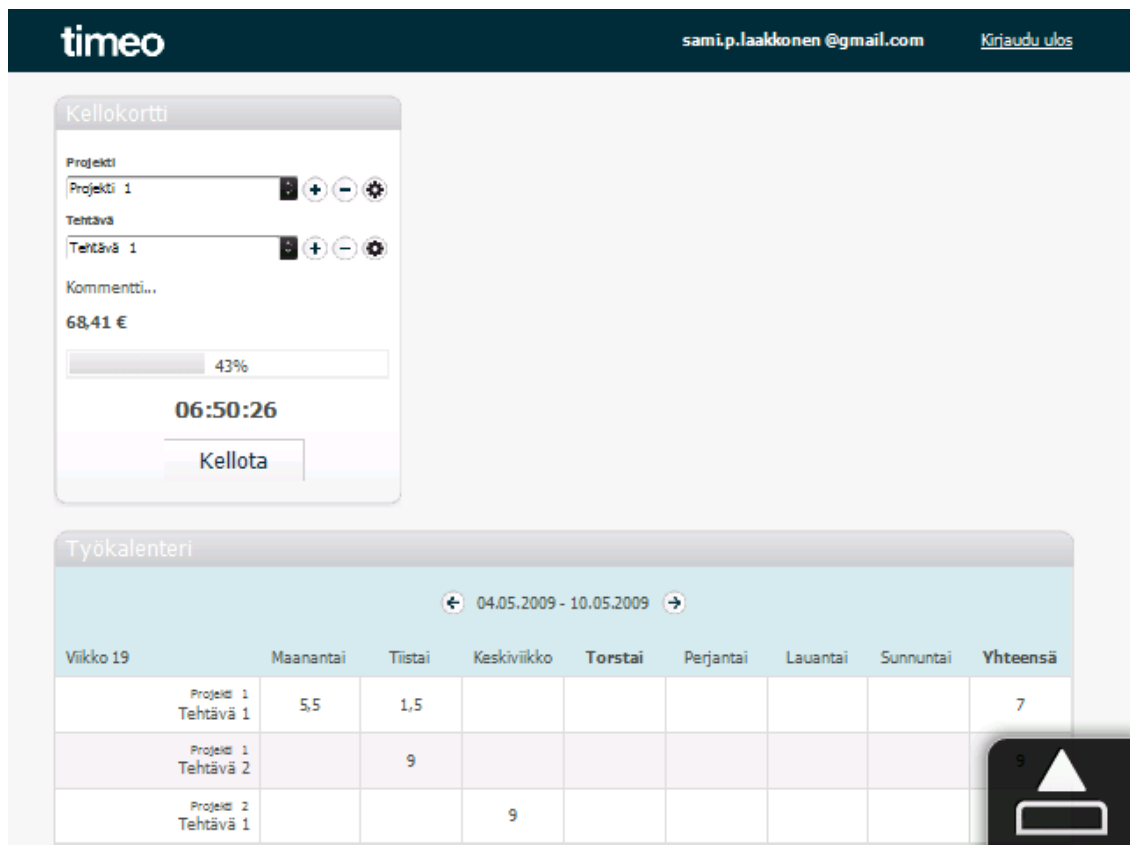
Kaksi suosittua selainta Applen Safari ja Mozilla Firefox ovat saatavilla useammalle käyttöjärjestelmälle. Safari on saatavilla sekä Mac OS X- että Windows-käyttöjärjestelmille. Mozilla Firefox on saatavilla Windowsille, OS X:lle sekä Linuxille. Kehittäjien ei tarvitse tehdä verkkosovelluksista eri versiota eri käyttöjärjestelmille, sillä sovellukset toimivat selaimilla. Selaimet tukevat yleisiä standardeja ja teknologioita, joten verkkosovellukset toimivat yleensä samalla lailla myös eri selaimilla lukuun ottamatta Internet Explorerin vanhempia versioita, kuten versiota 6, joka on edelleen hyvin suosittu. Internet Explorer 6 ei ole suosittu kuitenkaan kehittäjien keskuudessa, sillä esimerkiksi selaimen CSS-tuki on puutteellinen, jonka vuoksi kehittäjien täytyy tehdä paljon ylimääräistä työtä saadakseen verkkosivuista Internet Explorer 6 - yhteensopivia.

2.7 Käytettävyys mobiililaitteissa

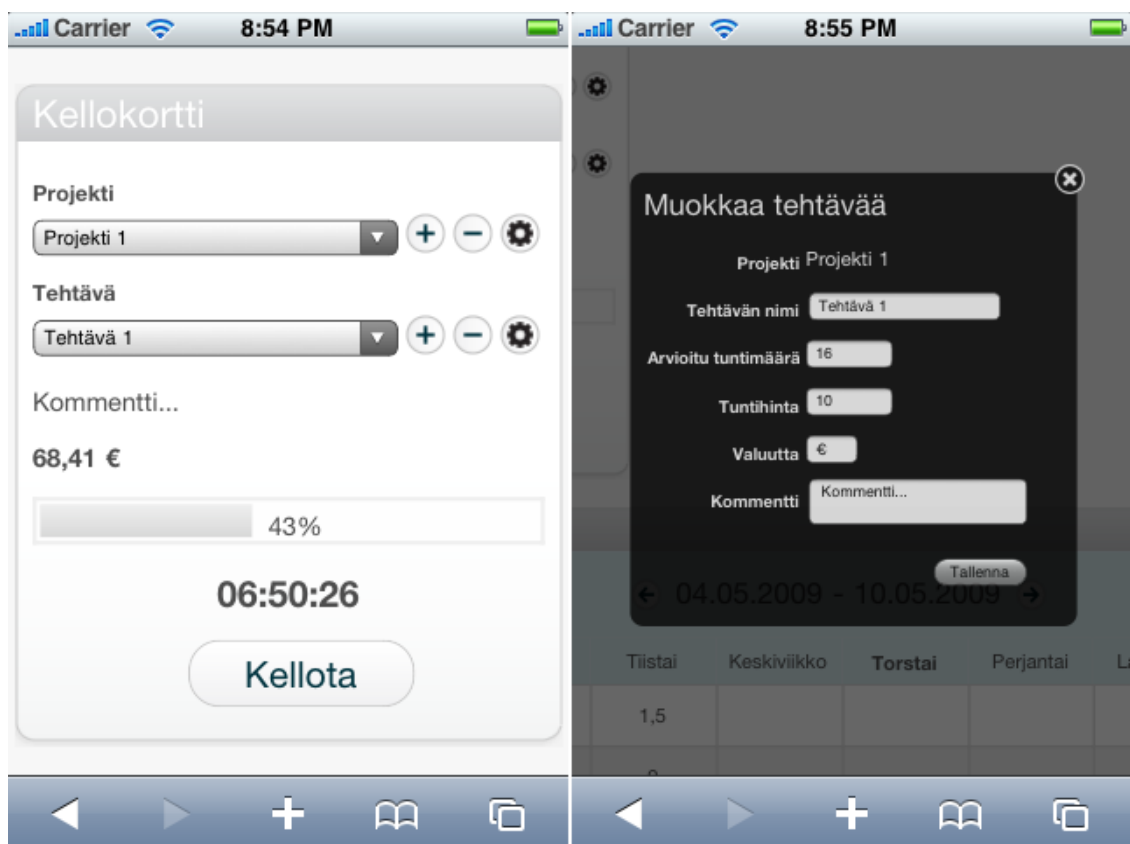
Mobiililustojen selaimet kehittyvät nopeaa vauhtia. Kehitys mahdollistaa verkkosovellusten käyttämisen mobiililaitteissa, kuten matkapuhelimissa. Applen iPhone-matkapuhelimen Safari-selain muistuttaa teknisiltä ominaisuuksiltaan hyvin paljon Applen pöytätietokoneille tarkoitettua Safari-selainta.

Nykyaikaisten mobiililaitteiden selaimissa voidaan käyttää jo teknisesti melkein yhtä vaativia verkkosovelluksia kuin pöytätietokoneidenkin selaimissa. Mobiililaitteiden suorituskyky kasvaa ja niiden selaimet kehittyvät samalla. Ainoa rajoite tulee olemaan mobiililaitteiden pienikokoiset näytöt, jotka tuovat haasteita kehittäjille. Pienikokoiselle näytölle on vaikea suunnitella toimivaa käyttöliittymää, varsinkin jos sovellus on yhtään monimutkaisempi. Mobiiliselaimille on kehitetty ominaisuuksia, jotka helpottavat verkkosivujen selausta pienellä näytöllä. Tällainen ominaisuus on esimerkiksi sivun koon suurentaminen ja pienentäminen. Verkkosivu nähdään aluksi kokonaan laitteen näytöllä. Käyttäjä voi valita sivulta tietyn osan, kuten tekstikentän, jolloin selain suurentaa tekstikentän koko näytölle käyttäjän luettavaksi. Tällainen verkkosivun suurentaminen ja pienentäminen on hyvä tapa parantaa mobiililaitteiden selainten käytettävyttä, mutta käyttömukavuus ei silti voita tavallisten pöytätietokoneiden suurta näyttöä hyödyntävien selainten käyttömukavuutta.

Timeota testattiin iPhoneen Safarilla ja Windows Mobile -mobiilikäyttöjärjestelmän Opera-selaimella. Kuvissa 1 ja 2 on kuvakaappaukset Timeosta molempien laitteiden selaimilla. Kummankin mobiiliselaimen tuki JavaScriptille ja muille teknologioille on sen verran kehittynyttä, että kaikki Timeon ominaisuudet toimivat niillä. Ainoa ongelma oli, että Timeon käyttö oli melko vaivalloista, johtuen laitteiden pienikokoisesta näytöstä verrattuna pöytätietokoneiden suurempikokoisiin ja tarkempiin näyttöihin.



Kuva 1. Kuvakaappaus Timeosta Windows Mobilen Opera 9.5-selaimella.



Kuva 2. Kuvakaappauksia Timeosta iPhone:n selaimella.

Verkkosovelluksille ja verkkosivuille voidaan tehdä oma käyttöliittymä mobiililaitteiden selaimille, jossa on otettu huomioon mobiililaitteiden näytön pienikokoisuus. Tämä onnistuu esimerkiksi erillisellä CSS-tyylitiedostolla, joka määrittelee verkkosivun ulkoasun mobiililaitteille. Erillinen tyyli tiedosto otetaan automaattisesti käyttöön, kun verkkosivu tunnistaa, että käyttäjä käyttää sivua mobiililaitteen selaimella. Erillisten CSS-tyylitiedostojen tekeminen vaatii kuitenkin aikaa ja vaivaa kehittäjiltä. Parempi ratkaisu olisi mobiililaitteiden selainten käytettävyyttä lisäävien ominaisuuksien kehittyminen sekä mobiililaitteiden näyttöjen kehittyminen.

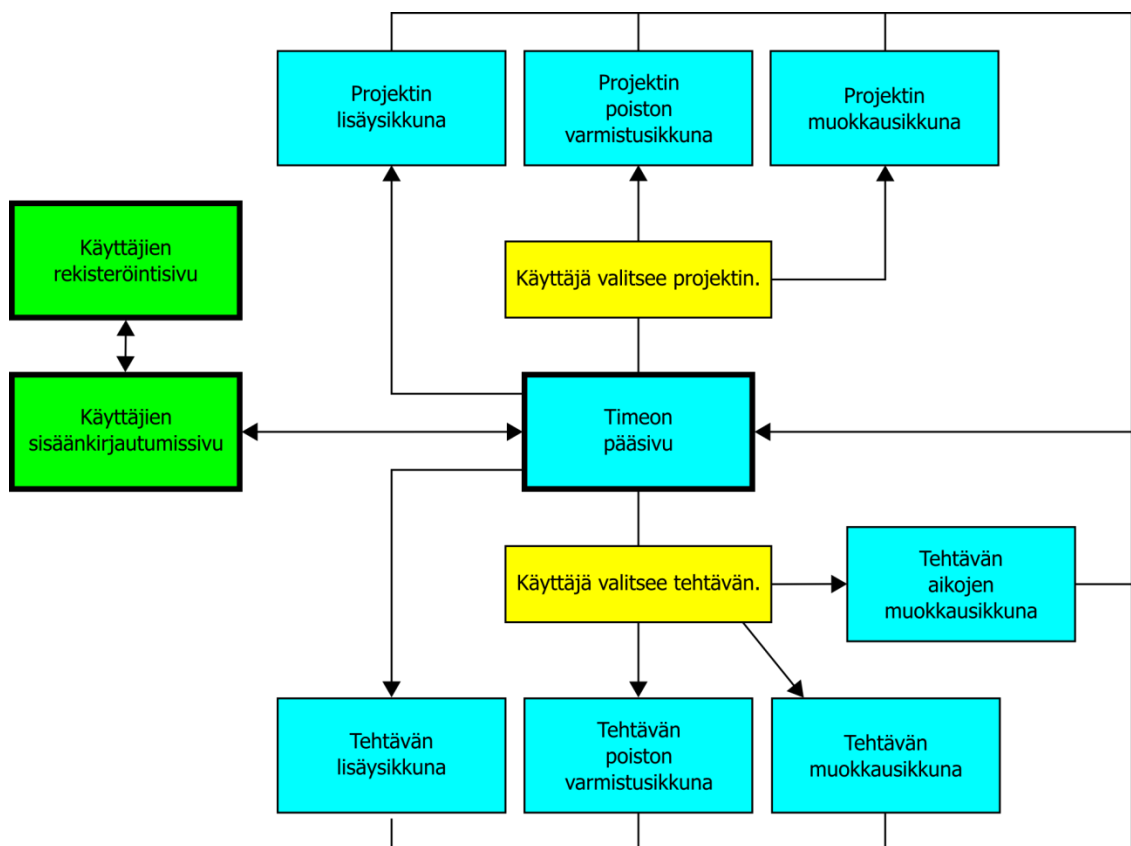
3 TIMEO

Timeo on ajanseurantaan suunniteltu avoimen lähdekoodin verkkosovellus, jolla voidaan pitää yksinkertaista tuntiseurantaa projekteihin ja niiden osatehtäviin kuluneesta ajasta. Tehtäville voidaan asettaa muun muassa tuntihinta, valuutta, tavoiteaika ja kommentteja. Timeo voidaan laittaa ottamaan aikaa työnteon ajaksi, mutta aikoja voidaan myös muokata ja lisätä itse. Timeo laskee ja näyttää otetuista ajoista tilastoja, kuten tehtävistä kertyneen palkan tuntihinnan mukaan, kuukauden aikana kertyneen palkan ja tuntimäärän sekä tehtävien tuntimäärät erikseen viikon eri päiville.

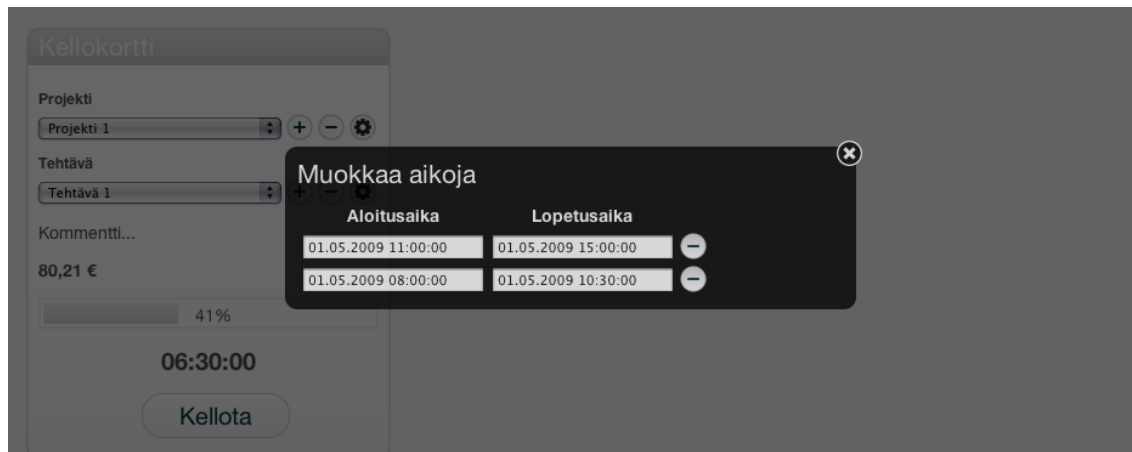
Käytän Timeota esimerkkinä JavaScriptiin ja Ajax-tekniikkaan perustuvasta verkkosovelluksesta. Käyn tässä luvussa läpi Timeon rakennetta ja toimintalogiikkaa sekä Timeon käyttöönoton vaatimuksia. Käyn läpi myös Timeossa käytettyjen palvelinpuolen teknologioiden, kuten PHP:n, ja selaimen asiakaspuolen teknologioiden, kuten JavaScriptin, välistä suhdetta.

3.1 Navigointirakenne ja sivut

Kuviossa 6 on esitetty Timeon navigointirakenne ja sivunäkymät. Kun käyttäjä tulee ensimmäistä kertaa Timeon sivuille, hänelle avautuu näkymä sisäänkirjautumissivusta. Jos käyttäjä ei ole aiemmin käyttänyt Timeota, eikä hänellä ole tunnuksia, käyttäjä voi siirtyä rekisteröintisivulle ja luoda itselleen käyttäjätunnukset. Kirjaututtuaan tunnuksillaan sisälle Timeoon, käyttäjä siirtyy Timeon pääsivulle. Pääsivulla käyttäjä voi lisätä itselleen projekteja ja projektien osatehtäviä sekä valita niitä muokattavaksi tai laittaa Timeon ottamaan osatehtäviin kuluvaan aikaa. Kuviossa 6 esitetyt siniset laatikot, kaikki paitsi pääsivu-laatikko, kuvaavat ikkunanäkymiä, jotka avautuvat pääsivun päälle (ks. kuva 3). Näkymät on toteutettu selkeyden takia ikkunanäkyminä tai toisin sanoen ikkunoina, koska ne liittyvät pääsivulla oleviin toimintoihin.



Kuvio 6. Timeon navigointirakenne.



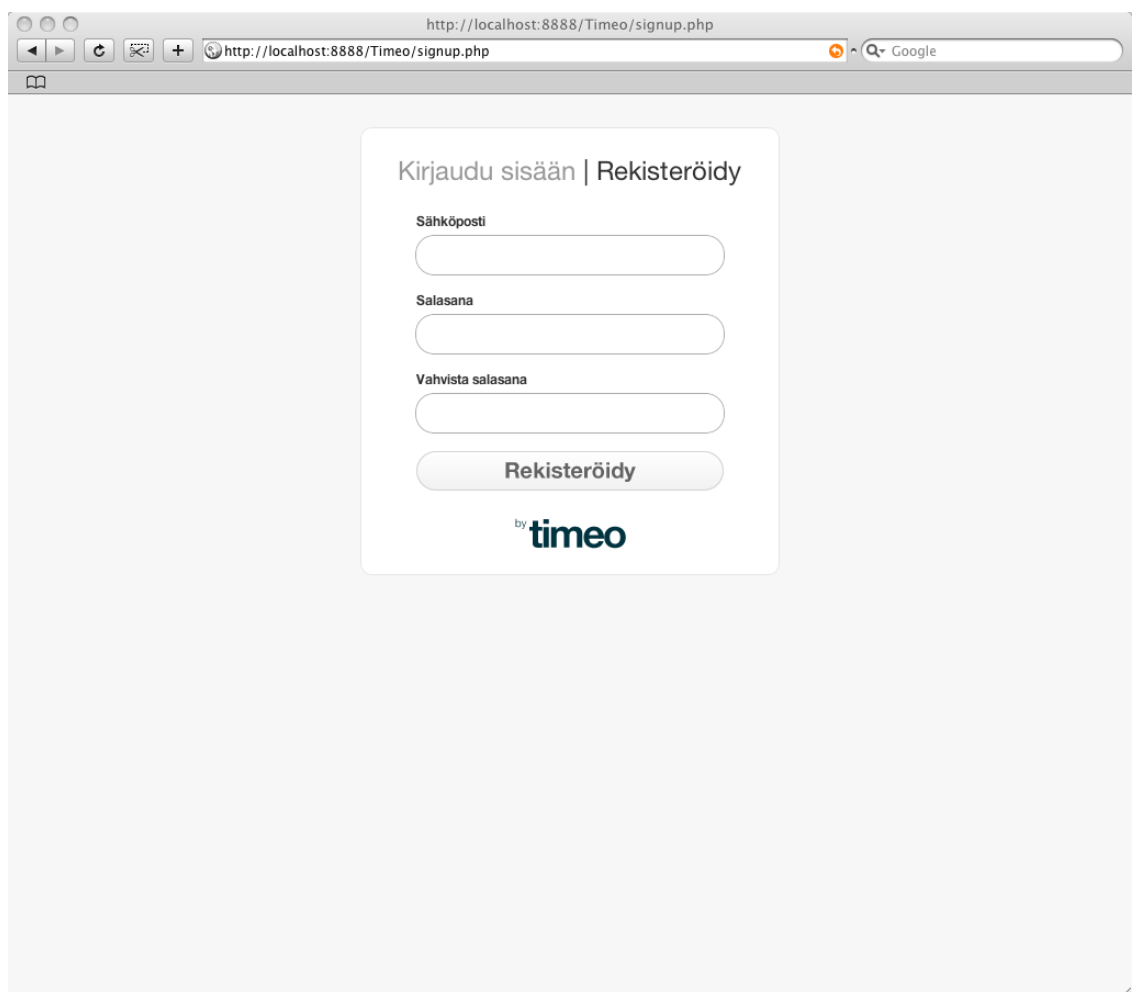
Kuva 3. Kuvakaappaus Muokkaa aikoja -ikkunanäkymästä. Taustalla näkyy pääsivu.

Kun käyttäjä esimerkiksi muokkaa aikoja, muutokset näkyvät heti ilman sivunlatauksia myös ikkunanäkymän taustalla olevalla pääsivulla. Samalla muuttuneet tiedot välittyvät palvelimelle. Tämä on mahdollista, sillä Timeon pääsivusta vastaa Ajax-tekniikkaa hyödyntävä JavaScript-moottori. Ikkunanäkymät aukeavat heti, eikä käyttäjän tarvitse odottaa niiden latautumista. Taustalla näkyvä pääsivu tummennetaan, mikä kiinnittää käyttäjän huomion avattuun ikkunaan. Tummennetut kohdat eivät reagoi käyttäjän toimiin.

Timeon navigointirakenne on pyritty pitämään mahdollisimman yksinkertaisena ja selkeänä. Timeo on ulkoasultaan jaettu kahteen yhtenäiseen osaan: käyttäjien rekisteröinti- ja kirjautumissivuihin sekä Timeon pääsivuun. Käyttäjien sisäänkirjautumissivun ja rekisteröintisivun ulkoasut ovat yhtenäiset (ks. kuva 4 sivulla 19 ja kuva 5 sivulla 24), sillä molemmat sivut liittyvät käyttäjätunnusten hallintaan, eli tunnusten luomiseen ja käyttämiseen. Sivulla ei myöskään ole Timeon esittelytekstejä tai muuta ylimääräistä sisältöä. Timeon ei ole tarkoitus olla valmis palvelusivusto, vaan avoimen lähdekoodin projekti. Ulkoasu on mahdollisimman pelkistetty, jotta Timeon mukauttaminen käyttöönottajän omaan käyttöön on selkeämpää ja helpompaa.

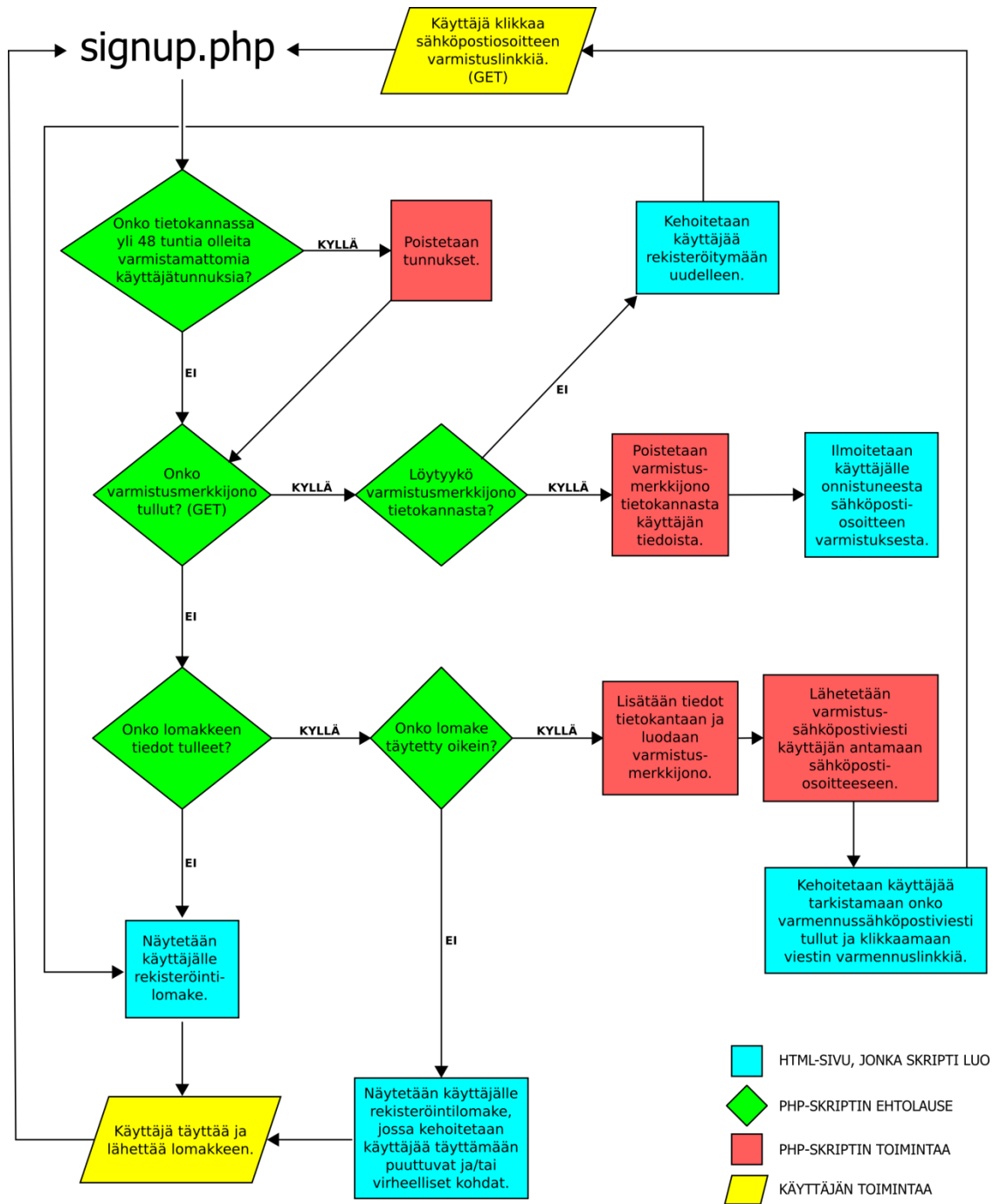
3.1.1 Uusien käyttäjien rekisteröintisivu

Käyttääkseen Timeota, käyttäjän on ensin rekisteröidyttävä. Rekisteröinti tapahtuu Timeon rekisteröintisivulla (ks. kuva 4). Rekisteröitymällä käyttäjälle luodaan omat henkilökohtaiset käyttäjätunnukset Timeon tietokantaan. Käyttäjän projektit, projektien osatehtävät ja otetut työajat pysyvät tallessa palvelimen tietokannassa. Timeon ollessa selainpohjainen sovellus, joka toimii yhdellä palvelimella, Timeota voidaan käyttää selaimella mistä päin maailmaa tahansa millä tahansa Internetiin yhteydessä olevalla tietokoneella.



Kuva 4. Kuvakaappaus Timeon uusien käyttäjien rekisteröintisivusta.

Rekisteröintisivulla on rekisteröintilomake, jossa on kolme käyttäjän täytettävää kenttää. Kentät ovat sähköposti-, salasana- ja salasana uudelleen -kenttä. Salasana uudelleen -kenttä varmistaa sen, että käyttäjä ei vahingossa kirjoita haluamaansa salasanaa väärin. Salasana-kenttiä verrataan toisiinsa ja jos ne eivät täsmää, rekisteröinti ei onnistu. Lomakkeen kentät täytettyään, käyttäjän antamaan sähköpostiosoitteeseen lähetetään varmistusviesti, jolla varmistetaan käyttäjän antaman sähköpostiosoitteen aitous. Rekisteröintisivun toiminnasta vastaa palvelinpuolen signup.php-skripti. Kuviossa 7 on esitetty skriptin toimintalogiikka.



Kuvio 7. Vuokaavio signup.php-skriptin toimintalogiikasta.

Kun käyttäjä siirtyy rekisteröintisivulle, selain kutsuu Timeon palvelimella olevaa `signup.php`-skriptiä. Skripti etsii aluksi tietokannasta käyttäjätunnuksia, joiden sähköpostiosoitetta ei ole varmistettu 48 tunnin sisällä, ja poistaa ne. Tämä siksi että Timeon tietokanta ei täyty ajan myötä turhilla käyttäjätunnuksilla, joiden rekisteröinti on tehty virheellisillä tai väärillä sähköpostiosoitteilla. Tämän jälkeen skripti tarkistaa onko käyttäjä tullut sivulle klikkaamalla rekisteröinnin yhteydessä lähetettyä varmistussähköpostiviestin sisältämää varmistuslinkkiä.

Varmistussähköpostiviesti sisältää varmistuslinkin (ks. kuvio 8), joka on voimassa 48 tuntia. Tänä aikana käyttäjän on avattava linkki selaimessa tai muuten rekisteröinti vanhenee ja rekisteröintilomake on täytettävä uudelleen, jos käyttäjä haluaa rekisteröityä Timeoon. Varmistuslinkin loppuosaan luodaan satunnainen uniikki varmistusmerkkijono, josta tunnistetaan sen käyttäjän tiedot tietokannasta, jonka sähköposti-osoitetta ollaan varmistamassa. Kun varmistuslinkki avataan selaimessa, kutsutaan Timeo-palvelimella olevaa `signup.php`-skriptiä. Linkin loppuosan varmistusmerkkijono välittyy skriptille PHP:n GET-metodilla, eli skripti tunnistaa linkin loppuosan merkkijonon muuttujaksi nimeltä `vc`. Skripti etsii Timeon tietokannasta käyttäjän tiedot, joiden kohdalta löytyy kyseinen merkkijono, ja määrittää käyttäjän sähköpostiosoitteen varmistetuksi poistamalla varmistusmerkkijonon käyttäjän tiedoista. Käyttäjän sähköpostiosoite on varmistettu ja käyttäjä voi kirjautua sisään Timeoon, kun tietokannasta on poistettu varmistusmerkkijono käyttäjän kohdalta.

<http://www.timeo.fi/signup.php?vc=78d6fb7f58af38ffa109af168928be136378ba1e>

Kuvio 8. Esimerkki sähköpostiosoitteen varmistuslinkistä.

Skripti tarkistaa käyttäjän antamat tiedot, jotta rekisteröinti voi onnistua. Kaikkien lomakkeen kenttien pitää olla täytettynä ja salasanakenttien on täsmättävä. Lisäksi sähköpostiosoitteen on oltava oikeaa muotoa, eli siitä on löydyttävä ainakin piste ja @-merkki. Timeo ei vaadi käyttäjältä muuta tietoa kuin sähköpostiosoitteen ja salasanan. Näin käyttäjän ei tarvitse keksiä itselleen erillistä käyttäjänimeä. Käyttäjänimen keksiminen ei aina ole helppoa, varsinkaan jos verkkosivulla, jonne ollaan rekisteröitymässä, on paljon muita käyttäjiä. Kaikki yleisimmät käyttäjänimet on silloin yleensä jo varattu. Jotkin sivustot ehdottavat käyttäjänimen perään numeroa, jolla nimi erotetaan muista käyttäjänimistä. Tällöin käyttäjänimestä tulee kuitenkin vaikeammin

muistettava. Timeossa käyttäjän ei tarvitse muistaa erillistä käyttäjänimeä. Riittää, että muistaa oman sähköpostiosoitteensa. Sähköpostiosoitetta käytetään käyttäjänimenä yhä useammalla verkkosivulla. Muita tunnettuja verkkosivuja, jotka käyttävät sähköpostiosoitetta käyttäjänimenä, ovat esimerkiksi Google¹⁶, Facebook¹⁷ ja Vimeo¹⁸.

3.1.2 Käyttäjien sisäänkirjautumissivu

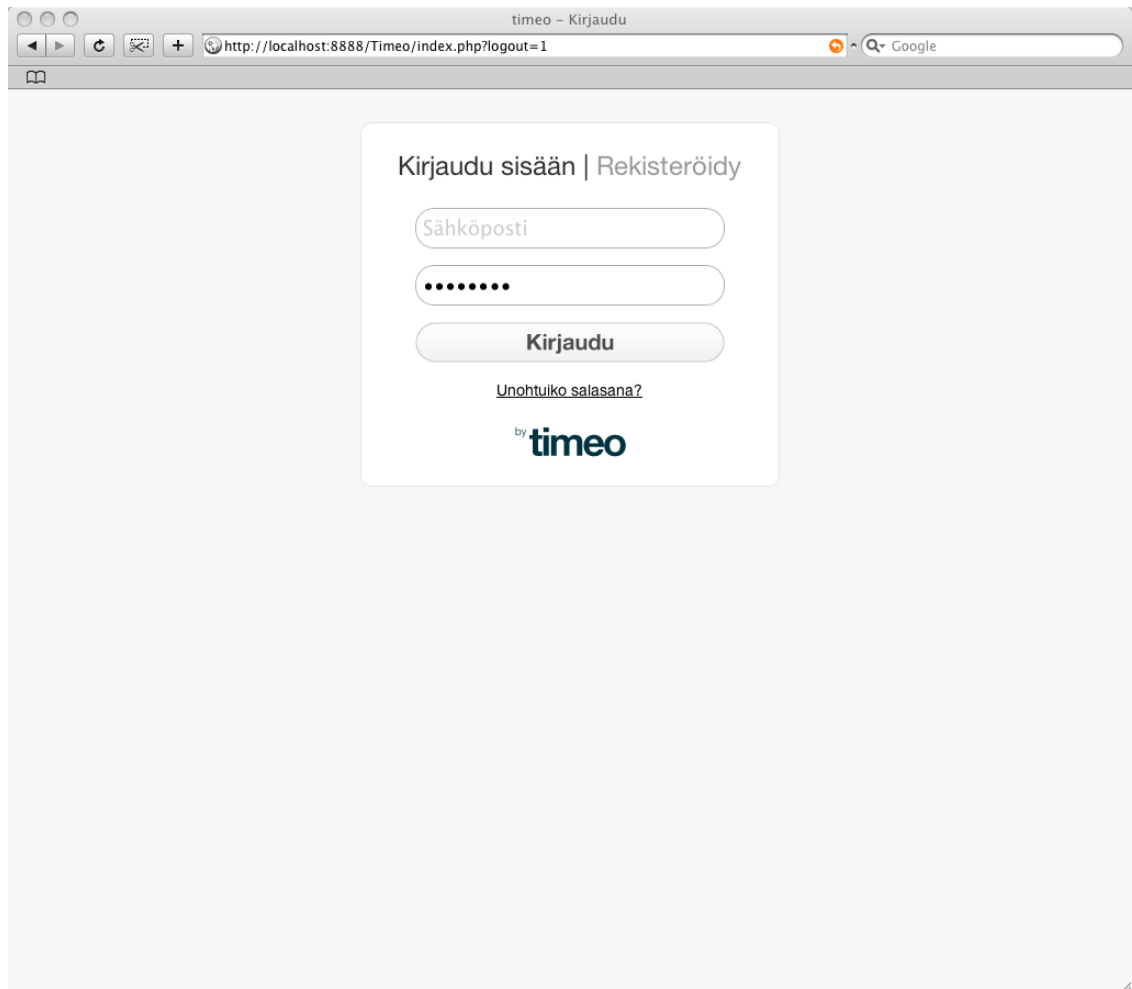
Kuvassa 5 on kuvakaappaus Timeon käyttäjien sisäänkirjautumissivusta. Sisäänkirjautumissivulla on sisäänkirjautumislomake ja linkki käyttäjien rekisteröintisivulle. Sisäänkirjautumislomakkeessa on kaksi kenttää, jotka ovat sähköposti- ja salasana-kenttä. Jos käyttäjä on rekisteröitynyt Timeoon, sisäänkirjautumiseen riittää, että käyttäjä kirjoittaa vain käyttäjätunnuksena toimivan sähköpostiosoitteen ja salasanan. Jos käyttäjä ei muista salasanaansa, hänen täytyy täyttää sähköposti-kenttä ja klikata Unohtuiko salasana? -linkkiä. Timeo etsii tämän jälkeen tietokannasta kyseisen sähköpostiosoitteen omistavan käyttäjän tiedot ja tiedoista salasanan. Salasana lähetetään sähköpostiviestinä käyttäjän antamaan sähköpostiosoitteeseen. Jos sähköpostiosoitetta ei löydy, Timeo ilmoittaa siitä ja pyytää käyttäjää rekisteröitymään Timeoon käyttäjien rekisteröintisivulla.

Timeon palvelimella oleva index.php-skripti hoitaa käyttäjien sisään- ja uloskirjautumisen. Kun käyttäjä on täyttänyt kirjautumislomakkeen ja kirjautuu sisään, skripti luo käyttäjän tiedoista erityiset sessiomuuttujat, jotka säilyvät sivunvaihdosten välillä, kunnes käyttäjä kirjautuu ulos. Sessiomuuttujista Timeon pääsivu saa tiedot kirjautumisesta ja kirjautuneesta käyttäjästä. Näillä tiedoilla Timeo osaa hakea oikean käyttäjän tiedot tietokannasta. Tietoturvasyistä jokainen PHP-skripti, joka on tekemisissä tietokannan kanssa, tarkastaa käyttäjän tiedot sessiomuuttujista. Kun käyttäjä kirjautuu ulos, index.php-skripti tuhoaa sessiomuuttujat. Käyttäjien kirjautuminen ja varmentaminen hoidetaan tietoturvasyistä PHP:llä palvelinpuolella. En kuitenkaan käsittele tässä työssä tietoturvaan liittyviä asioita, sillä aihe on hyvin laaja.

16 Google (<http://www.google.com>) on suosittu Internet-hakukone ja Internet-palveluita tarjoava yhtiö.

17 Facebook (<http://www.facebook.com>) on Internetissä toimiva yhteisöpalvelu.

18 Vimeo (<http://www.vimeo.com>) on Internetissä toimiva videopalvelu.



Kuva 5. Kuvakaappaus Timeon käyttäjien sisäänkirjautumissivusta.

3.1.3 Timeon pääsivu

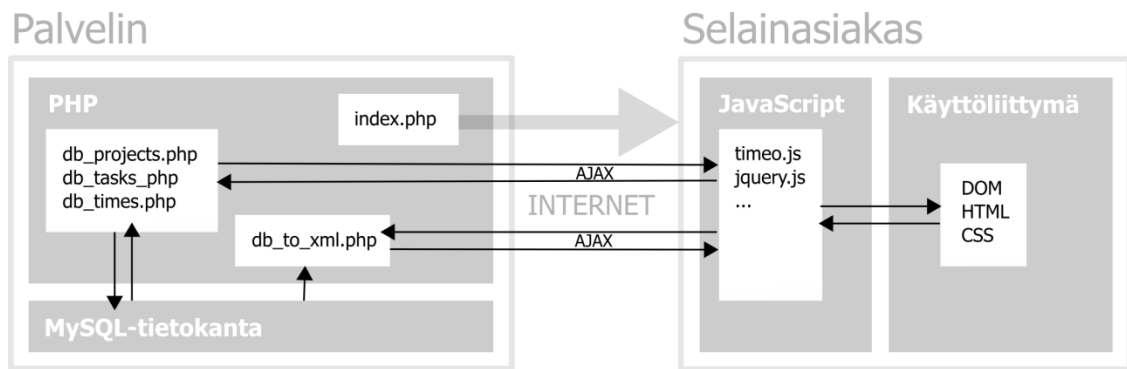
Kuvassa 6 on kuvakaappaus Timeon pääsivusta. Oikeassa yläkulmassa on Kirjaudu ulos -linkki, josta käyttäjä voi poistua pääsivulta ja kirjautua ulos Timeosta. Vasemmalla on Kellokortti-niminen osa, josta käyttäjä voi lisätä, valita, muokata ja poistaa projekteja sekä projektien osatehtäviä. Kellokortista käyttäjä voi valita osatehtävän ja laittaa Timeon "kelottamaan", eli ottamaan aikaa tehtävään kuluva ajasta. Ajanoton aikana kellokortissa näkyvä tehtävän kokonaisaika päivittyy joka sekunti. Kellokortti laskee ja näyttää tehtävään kuluneen kokonaisajan, prosenttiluvun tavoiteajasta sekä kertyneen palkan tehtävän tuntiinnan mukaan. Alhaalla näkyvä Työkaleri-osa taas laskee ja näyttää erikseen tunnit päivien ja tehtävien mukaan sekä koko kuukauden tuntimäärän ja kertyneen palkan.

Viikko 18	Maanantai	Tiistai	Keskiviikko	Torstai	Perjantai	Lauantai	Sunnuntai	Yhteensä
Projekti 1 Tehtävä 1					6,5			6,5
Projekti 1 Tehtävä 2				5	1			6
Projekti 2 Tehtävä 1	6							6
Yhteensä	**6**			**5**	**7,5**			**18,5**

 At the bottom right of the calendar section, it says 'Toukokuu 7,5 h / 80 €'.

Kuva 6. Kuvakaappaus Timeon pääsivusta.

Kuviossa 9 on kuvattu Timeon pääsivun JavaScript-moottorin toimintaa ja kommunikointia palvelinpuolen PHP-skriptien kanssa. Kuvio toimii myös esimerkkinä Ajax-sovelluksen selaimen ja palvelimen välisestä kommunikoinnista. Kun käyttäjä kirjautuu sisään Timeoon, palvelimella oleva index.php-skripti luo HTML-sivun, joka lähetetään käyttäjän selaimelle. Sivun on linkitetty Timeon JavaScript-moottori, eli tiedosto nimeltä timeo.js sekä osa muista JavaScript-tiedostoista, kuten JavaScript-kirjasto nimeltä jquery.js ja sen lisäosat. JavaScript-moottori hoitaa käyttäjälle näkyvän käyttöliittymän esityksestä ja toiminnasta. Pääsivun käyttöliittymä on DOM-mallin mukainen HTML-dokumentti, jonka elementtejä JavaScript-moottori luo ja muokkaa muutoksien tapahtuessa. Käyttäjä näkee HTML- ja CSS-tekniikoilla luodun HTML-sivun, jossa muutokset tapahtuvat dynaamisesti ilman sivunlatauksia. Käyttäjän tehdessä muutoksia projekteihin ja osatehtäviin sekä ottaessa aikoja, JavaScript moottori lähettää muuttuneet tiedot Ajax-tekniikalla palvelimen PHP-skripteille. Skriptit hoitavat yhteyden palvelimella sijaitsevaan MySQL-tietokantaan.



Kuvio 9. Timeon JavaScript-moottorin toiminta sekä pyyntöjen ja vastausten kulku selainasiakkaan ja palvelimen välillä.

Kuviossa 10 on esimerkki timeo.js-tiedoston lähdekoodista, jossa Timeon JavaScript-moottori lähettää Ajax-pyyntöä palvelimella sijaitsevalle db_to_xml.php-skriptille. Koodiesimerkki on jQuery-koodia. jQuerystä on paljon apua kehittäjille, sillä se helpottaa Ajax-komentojen kirjoittamista ja selkeyttää koodia. Kuviossa rivillä 3 määritellään, että pyyntö tapahtuu asynkronisesti. Rivillä 5 määritellään palvelimella sijaitseva PHP-skripti, jolle pyyntö lähetetään. Rivillä 6 määritellään, että palvelimelta saatava vastaus on XML-muotoa. Rivillä 7 määritellään funktio, joka suoritetaan, jos pyyntö epäonnistuu. Tässä tapauksessa koko sivu ladataan uudelleen. Riveillä 8–10 määritellään funktio, joka suoritetaan, jos pyyntö onnistuu. Funktiolle annetaan argumenttina palvelimelta vastauksena saatu tieto.

```

1 $.ajax({
2   type: "POST",
3   async: "true",
4   cache: "false",
5   url: "db_to_xml.php",
6   dataType: "xml",
7   error: function() { location.reload(); },
8   success: function(xml) {
9     TIMEO_LANG_CALENDAR =
10    $(xml).find('item[name=calendar]').text();
11  });

```

Kuvio 10. Esimerkki jQuery-koodista, joka lähettää Ajax-pyyntöä.

Kuviossa 11 on esimerkki palautettavista tiedoista XML-muodossa. Timeon JavaScript-moottori parsii tiedot XML-muodosta JavaScriptin omiksi muuttujiksi (ks. esimerkki kuvion 10 rivi 9), joita se käyttää tiedon esittämiseen ja pitämiseen muistissa sen

aikaa, kun käyttäjä on kirjautuneena Timeoon. Käyttäjän tiedot haetaan palvelimen tietokannasta vain silloin, kun käyttäjä kirjautuu sisään.

```
<?xml version="1.0" encoding="utf-8" ?>
<timeo>
  <user id="30">
    <email>sami.p.laakkonen@gmail.com</email>
    <projects>
      <project id="163">
        <user_id>30</user_id>
        <name>Projekti 2</name>
      </project>
      <project id="162">
        <user_id>30</user_id>
        <name>Projekti 1</name>
      </project>
    </projects>
    <tasks>
      <task id="311">
        <project_id>162</project_id>
        <user_id>30</user_id>
        <name>Tehtävä 1</name>
        <estimated_hours>16</estimated_hours>
        <hourly_rate>12.34</hourly_rate>
        <currency_symbol>€</currency_symbol>
        <comment>Kommentti...
        </comment>
      </task>
      <task id="313">
        <project_id>162</project_id>
        <user_id>30</user_id>
        <name>Tehtävä 3</name>
        <estimated_hours>0</estimated_hours>
        <hourly_rate>0</hourly_rate>
        <currency_symbol>€</currency_symbol>
        <comment/>
      </task>
    </tasks>
    <times>
      <time id="559">
        <task_id>311</task_id>
        <user_id>30</user_id>
        <start_time>2009-05-01 18:00:09</start_time>
        <end_time>2009-05-01 19:00:10</end_time>
      </time>
      <time id="562">
        <task_id>313</task_id>
        <user_id>30</user_id>
        <start_time>2009-05-11 06:17:00</start_time>
        <end_time>2009-05-11 06:17:39</end_time>
      </time>
    </times>
  </user>
</timeo>
```

Kuvio 11. Esimerkki db_to_xml.php-skriptin palauttamista käyttäjän tiedoista XML-muodossa.

3.2 Timeon käyttöönotto ja tiedostot

Timeo vaatii palvelimelta PHP-tuen ja MySQL-tietokannan, jotka löytyvät useimmilta palvelimilta. Timeo toimii kehitystyön aikana ilmaisella MAMP¹⁹-ohjelmistolla, joka sisältää muun muassa Apache-palvelimen, MySQL-tietokannan, phpMyAdmin-tietokantatyökalun ja PHP:n samassa paketissa. MAMP-ohjelmistosta on saatavilla myös Windowsille tehty versio WAMP²⁰ ja Linuxille tehty versio LAMP²¹. Timeon käyttöönotto vaatii, että Timeon tiedostot siirretään palvelimelle. Taulukossa 1 on listattu kaikki Timeon tiedostot.

Taulukko 1. Timeon tiedostot ja niiden kuvaukset.

Tiedostonimi	Tiedoston laji	Tiedoston kuvaus
lang	Kansio	Tämä on kansio, joka sisältää XML-muotoiset kielitiedostot. Esimerkiksi englanninkielinen kielitiedosto on nimeltään en.xml ja suomenkielinen fi.xml.
images	Kansio	Tämä kansio sisältää kuvatiedostot, joita tarvitaan Timeon verkkosivujen ulkoasun luomiseen.
timeo.js	JavaScript-skripti	Tämä tiedosto sisältää ohjelmakoodin, joka hoitaa Timeon koko toimintalogiikan.
jquery.js	JavaScript-skripti	Tämä tiedosto sisältää jQuery-JavaScript-kirjaston.
jquery.tablesorter.js	JavaScript-skripti	Tämä on jQuery:n laajennos, joka sisältää funktioita verkkosivun taulukoiden rivien uudelleenjärjestämiseen ilman sivulatauksia.
jquery.form.js	JavaScript-skripti	Tämä on jQuery:n laajennos, joka vähentää kirjoitettavan JavaScript-koodin tarvetta, kun ohjelmoidaan verkkosivujen lomakkeiden lähettämistä Ajax-tekniikalla.
jquery.corners.js	JavaScript-skripti	Tämä on jQuery:n laajennos, joka mahdollistaa verkkosivun graafisten elementtien kulmien helpon pyöristyksen.
jquery.corners.init.js	JavaScript-skripti	Tämä on tiedosto, jota jQuery:n corners-laajennos tarvitsee.
jquery.clearinginput.js	JavaScript-skripti	Tämä jQuery:n-laajennos mahdollistaa verkkosivun lomakkeiden kenttien nimen näyttämisen kentän sisällä.

19 Mac OS X + Apache + MySQL + PHP

20 Windows + Apache + MySQL + PHP

21 Linux + Apache + MySQL + PHP

jquery.alphanumeric.js	JavaScript-skripti	Tämä on jQueryn laajennos, jolla voidaan rajoittaa käyttäjien verkkosivujen lomakkeisiin syöttämien merkkien lajia. Sillä voidaan rajoittaa käytössä olevat merkit esimerkiksi vain numeroiksi.
functions.js	JavaScript-skripti	Tämä tiedosto sisältää hyödyllisiä JavaScript-funktioita.
date.js	JavaScript-skripti	Tämä on JavaScript kirjasto, joka sisältää hyödyllisiä funktioita aikojen laskemiseen ja käsittelyyn.
index.php	PHP-skripti	Tämä on PHP-skripti, joka hoitaa käyttäjien sisäänkirjautumisen ja Timeon pääsivun luomisen selaimelle.
signup.php	PHP-skripti	Tämä on PHP-skripti, joka luo uusien käyttäjien rekisteröintisivun ja hoitaa käyttäjien rekisteröinnin toimintalogiikan, kuten käyttäjien tietojen lisäämisen tietokantaan.
db_to_xml.php	PHP-skripti	Tämä PHP-skripti luo tietokannasta käyttäjän kaikista tiedoista XML-sivun, kun skriptiä kutsutaan esimerkiksi Ajax-tekniikalla.
settings.php	PHP-skripti	Tämä on PHP-skripti, jossa on määritelty kaikkien PHP-skriptien käyttämät tiedot MySQL-tietokannan asetuksista, Timeon yleisistä asetuksista sekä Timeon käyttämästä kielitiedostosta.
db_projects.php	PHP-skripti	Tämä on PHP-skripti, joka hoitaa tietokannan Projektit-taulun muokkauksen.
db_tasks.php	PHP-skripti	Tämä on PHP-skripti, joka hoitaa tietokannan Tehtävät-taulun muokkauksen.
db_times.php	PHP-skripti	Tämä on PHP-skripti, joka hoitaa tietokannan Ajat-taulun muokkauksen.
timeo.sql	MySQL-tietokantatiedosto	Tämä tiedosto sisältää tyhjät MySQL-taulut, jotka siirretään tietokantaan, kun Timeo-otetaan käyttöön palvelimella.
style.css	CSS-tyylitiedosto	Tässä tiedostossa on määritelty Time on ulkoasu.
login.css	CSS-tyylitiedosto	Tässä tiedostossa on määritelty Timeon kirjautumissivun ja rekisteröintisivun ulkoasut.

timeo.sql-tiedosto sisältää tyhjät tietokantataulut, jotka on siirrettävä MySQL-tietokantaan. Tämä onnistuu esimerkiksi ilmaisella phpMyAdmin-työkalulla, joka osaa lukea SQL-tiedostoja. MySQL-tietokannan asetukset on määriteltävä settings.php-skriptissä. Skriptiä voi muokata tavallisella tekstieditorilla. Kuviossa 12 on esimerkki

settings.php-skriptin kohdasta, jossa määritellään tietokannan asetukset. Rivillä 2 määritellään palvelimen osoite MySQL-tietokantaan. Rivillä 3 määritellään tietokannan käyttäjätunnus ja rivillä 4 salasana. Rivillä 5 määritellään tietokannan nimi. Rivillä 6 määritellään Timeon käyttämien taulujen nimen alkuosa. Jos tietokanta sisältää muitakin tauluja kuin Timeon tauluja, alkuosalla voidaan välttää käyttämästä väärää taulua. Esimerkiksi Timeon Käyttäjät-taulu on nimeltään pelkkä users ilman alkuosaa.

```

1 // mysql
2 define("TIMEO_MYSQL_SERVER", "localhost:8889");
3 define("TIMEO_MYSQL_USERNAME", "root");
4 define("TIMEO_MYSQL_PASSWORD", "root");
5 define("TIMEO_MYSQL_DB", "timeo");
6 define("TIMEO_MYSQL_PREFIX", "timeo_");

```

Kuvio 12. Esimerkki settings.php-skriptin kohdasta, jossa määritellään MySQL-tietokannan asetukset.

Tiedostot joiden tiedostonimen alussa on db_, hoitavat yhteyden palvelimeen ja vastaavat Timeon JavaScript-moottorin pyyntöihin. jquery-alkuiset tiedostot ovat Timeon JavaScript-moottorin käyttämän jQuery-kirjaston lisäosia. css-päätteiset tiedostot sisältävät Timeon ulkoasun CSS-määrittelyt. Kansio nimeltä lang, sisältää Timeon käyttämät kielitiedostot, jotka mahdollistavat tuen useammalle kielelle. Jokainen käyttäjälle näkyvä tekstipätkä ja sana Timeossa haetaan kielitiedostosta. Timeon käyttämä kielitiedosto määritellään settings.php-skriptissä.

settings.php-skripti sisältää komennon "define('TIMEO_LANG', 'fi');". Tämä komento määrittelee vakionmuuttujan nimeltä TIMEO_LANG ja antaa sille arvoksi fi. Timeon muut PHP-skriptit sisällyttävät settings.php:n, joten TIMEO_LANG-muuttujan arvo välittyy myös niille. Muuttujan arvosta skriptit ja myös Timeon JavaScript-moottori osaavat lukea oikeaa kielitiedostoa. Jos muuttujan arvo on esimerkiksi fi, käytetään fi.xml-nimistä kielitiedostoa. Jos Timeo halutaan kääntää toiselle kielelle, se onnistuu tekemällä esimerkiksi uusi kielitiedosto, jossa määritellään Timeon sisältämät tekstipätkät ja sanat uudella kielellä. Tämän jälkeen settings.php:stä TIMEO_LANG-muuttujan arvoksi muutetaan arvon fi sijasta uuden kielitiedoston nimi, ilman tiedoston lopputunnistetta xml. Kielitiedoston XML-muoto on kätevä, koska se on selkeä ja Timeon käyttöönottajana on helppo muokata sitä tavallisella tekstieditorilla. Kuviossa 13 on esimerkkinä osa Timeon suomenkielisestä kielitiedostosta fi.xml. Kehittäjän

näkökulmasta tiedon hakeminen XML-muotoisesta tiedostosta on myös helppoa PHP:llä ja JavaScriptillä.

```
<?xml version="1.0" encoding="utf-8"?>
<timeo>
  <language title="Finnish">
    <item name="email">Sähköposti</item>
    <item name="password">Salasana</item>
    <item name="log_in">Kirjaudu sisään</item>
    <item name="loading">Ladataan...</item>
    <item name="time_clock">Kellokortti</item>
    <item name="calendar">Työkalenteri</item>
    <item name="date_format">dd.MM.yyyy</item>
```

Kuvio 13. Osa Timeon suomenkielisestä kielitiedostosta (fi.xml).

PHP-skripti nimeltä `db_to_xml.php` palauttaa kutsuttaessa sillä hetkellä Timeoon kirjautuneen käyttäjän kaikki tiedot, kuten projekti, tehtävät ja ajat, tietokannasta XML-muodossa. Skriptiä voidaan kutsua Ajax-tekniikalla, jolla Timeo kutsuu skriptiä aina kun käyttäjä kirjautuu sisään Timeoon. Käyttäjä voi kutsua tiedostoa myös itse, kirjoittamalla selaimen osoitekenttään skriptitiedoston osoitepolun. Osoitepolku voi olla esimerkiksi muotoa `http://www.timeo.fi/db_to_xml.php`. Tällöin käyttäjälle palautetaan XML-tiedosto, jonka käyttäjä voi tallentaa selaimesta. Käyttäjän pitää olla kirjautuneena sisään Timeoon, sillä skripti seuraa kirjautumisessa luotuja sessiomuuttujia, muuten palautettava XML-tiedosto on tyhjä. Sessiomuuttujista skripti myös tunnistaa oikean käyttäjän tiedot. XML-tiedostosta voi olla käyttäjälle hyötyä myöhemmin, jos Timeosta tehdään tavallinen tietokoneohjelma tai iPhone-sovellus, joka ei tarvitse Internet-yhteyttä toimiakseen, ja käyttäjä haluaa siirtää omat tietonsa sille. Uuteen sovellukseen voidaan tehdä toiminto, joka osaa lukea XML-tiedostoa ja sen sisältämät käyttäjän tiedot, kuten projektit, tehtävät ja ajat.

3.3 Tietokantarakenne

Timeo tarvitsee tiedon varastomiseen tietokannan. Tietokannaksi valittiin MySQL, koska se on ilmainen, suosittu ja sen käyttö onnistuu helposti PHP:llä. Kun Timeota kehitettiin, käytössä oli MySQL:n sen hetkinen uusin versio 5.0. Tietokantaa hallinnoitiin phpMyAdmin-työkalulla. Timeon tietokannan tietorakenteita voidaan kuvailla relaatiomalli-tietomallilla. Timeon tietokanta sisältää neljä relaatiotietokantamallin mukaista taulua, Käyttäjät, Projektit, Tehtävät ja Ajat.

IBM:n tutkija E. F. Codd julkaisi relaatiotietokannan teoreettisen pohjan määrittelevän relaatiomallin (the relational model) vuonna 1970. Relaatiotietokantatuotteet ovat syrjäyttäneet aiemmat hierarkkiset ja verkkomalliset tietokantatyypit lähes kokonaan. Uudet tietojärjestelmät rakennetaan usein relaatiokantatuotteiden avulla. SQL on myös standardoitu lähes ainoaksi tietokantakieleksi. (Hovi, Huotari & Lahdenmäki 2003, 7–8.)

Relaatiomalli voidaan jakaa kolmeen osaan, rakenteeseen, käsittelyyn ja eheytyssääntöihin. Tietokannan rakenteessa peruselementti on taulu. Asiakokonaisuudet esitetään tauluina, joissa on sarakkeita ja rivejä. Taulut sisältävät perusavaimen (primary key, PK), joka on yksilöivä, eli uniikki. Viittaavalla taululla on myös viiteavain (foreign key), joka viittaa toisen taulun perusavaimen. Viittaavaa taulua sanotaan lapsitauluksi (referenced table) ja viittauksen kohdetaulua isätauluksi (master table). Viiteavaimilla tehdään liitoksia taulujen välille. Relaatiomallissa tietoja käsitellään joukko-opillisesti. Taulut sisältävät joukon rivejä, joihin voidaan kohdistaa joukko-operaatioita. Relaatiomallin eheytyssääntöihin kuuluu, että perusavaimella on aina oltava arvo, eikä se voi olla tyhjä, eli NULL-arvo. Tätä eheytyssääntöä kutsutaan avaineheydeksi (entity integrity). Toinen eheytyssääntö on viite-eheys (referential integrity), jonka mukaan isätaulusta ei pitäisi olla mahdollista poistaa riviä, johon lapsitaulu viittaa. (Hovi, Huotari & Lahdenmäki 2003, 8–12.)

Kuviossa 14 on esitetty Timeon tietokannan taulut. Otetaan esimerkiksi Käyttäjät-taulu. Taulu sisältää sarakenimiä kuten id, password, rights ja niin edelleen. Yksi taulukon rivi sisältää yhden käyttäjän tiedot tietokannassa. Tiedot on tallennettu sarakkeisiin, joiden sisällön sarakenimet kertovat. Sarake nimeltä id on perusavain, joka yksilöi käyttäjät.

Relaatiomallin eheysäännön mukaan, useammalla kuin yhdellä käyttäjällä ei voi olla samaa arvoa id-sarakkeessa, sillä eri käyttäjät tunnustetaan id:n arvosta. Jos perusavainta ei olisi ja taulussa olisi kaksi eri käyttäjää samoilla tiedoilla, niin käyttäjä ei voisi erottaa toisistaan tietokannasta. Timeon tapauksessa kuitenkin myös käyttäjän sähköpostiosoite on jokaisella käyttäjällä uniikki, eikä kahdella tai useammalla käyttäjällä voi olla samaa sähköpostiosoitetta Timeon tietokannassa. Käyttäjät-taulu on muiden käyttäjien isätaulu, sillä muut taulut viittaavat siihen. Esimerkiksi Projektit-taulu on Käyttäjät-taulun lapsitaulu, sillä se sisältää viiteavaimen nimeltä user_id, joka viittaa Käyttäjät taulun id-perusavaimeen. Näin tietokannasta voidaan tunnistaa mikä projekti kuuluu millekin käyttäjälle. Tehtävät-taulu viittaa Projektit-tauluun, sillä jokainen sen rivi eli tehtävä kuuluu johonkin projektiin. Ajat-taulu taas viittaa Tehtävät-tauluun, sillä jokainen aika kuuluu johonkin tehtävään.



Kuvio 14. Timeon relaatiotietokannan taulut.

Kaikki muut taulut viittaavat myös Käyttäjät-tauluun. Tämä helpottaa joukko-operaatioiden tekemistä, kuten tiedon hakua tietokannasta. Timeo voi tarvita esimerkiksi tiedot jokaisesta tietyin käyttäjän ottamasta ajasta, jotka kuuluvat tietylle tehtävälle. Tämä haku voidaan suorittaa yksinkertaisella SQL-lausekkeella: "SELECT * FROM timeo_times WHERE user_id=4 AND task_id=1". SELECT-komennolla haetaan tietoa. Merkki * tarkoittaa, että jokainen löytynyt rivi haetaan. Lausekkeen kohta "FROM timeo_times" tarkoittaa, että tietoa haetaan timeo_times-nimisestä taulusta. WHERE-komennon jälkeen tulee haettavan tiedon ehdot, jotka taulusta haettavilla riveillä, eli tässä tapauksessa ajoilla, on täyttyvä. user_id ja task_id ovat Ajat-taulun viiteavaimia, joilla taulun yksittäiset ajat voidaan yhdistää tietokannassa tiettyyn käyttäjän tehtävään. Tällaista hakua voidaan tarvita tulevaisuudessa, kun Timeolle kehitetään ryhmätyöominaisuus, jolloin yksittäinen tehtävä voi viitata useampaankin kuin yhteen käyttäjään.

Käyttäjät-taulu sisältää rights-sarakkeen, joka kuvaa käyttäjän oikeuksia. Timeo ei hyödynnä vielä käyttäjien oikeuksia, mutta tulevassa versiossa niitä tullaan käyttämään. Käyttäjät voidaan jakaa esimerkiksi peruskäyttäjiin ja ylläpitäjiin. Ylläpitäjätason oikeuksilla käyttäjät voivat tehdä muutoksia esimerkiksi Timeon palvelimen asetuksiin tulevassa Timeon hallintasovelluksessa, joka jäi vielä suunnitteluasteelle nykyisessä Timeon versiossa. Mitä pienempi numero käyttäjän oikeuksissa on, sitä enemmän oikeuksia käyttäjällä on. Jos käyttäjätasoa tehdään useampia, niin on hyvä pitää numerot aluksi 20 peruskäyttäjälle ja 10 ylläpitäjälle. Jos ylläpitäjän ja peruskäyttäjän väliin halutaan myöhemmin uusi käyttäjätaso, niin oikeudet voi olla 15.

4 TIMEON JATKOKEHITYS

Käyn tässä luvussa läpi asioita Timeon jatkokehityssuunnitelmista. Timeosta saatiin valmiiksi vasta ensimmäinen käytettävä versio tämän opinnäytteen kirjoittamisen aikana, joten osa suunnitelluista Timeon ominaisuuksista ei ehtinyt toteutua. Yksi näistä suunnitelluista ominaisuuksista on ryhmätyöominaisuus, joka mahdollistaisi käyttäjien jakaa projekteja ja tehtäviä muiden käyttäjien kanssa. Näin saman projektin tehtävät voidaan esimerkiksi jakaa useammalle käyttäjälle. Kun yksi käyttäjä ottaa jonkin tehtävän aikaa, muut käyttäjät eivät voi ottaa aikaa kyseiselle tehtävälle. Jaetun projektin käyttäjät näkisivät kuka ottaa tai on ottanut aikaa millekin tehtävälle, eli kuka on työskennellyt tehtävän parissa. Näin käyttäjät eivät tekisi turhaan samaa työtä saman tehtävän parissa. Myös Työkalerin tapaisia lisäosia on tarkoitus kehittää, kuten esimerkiksi erilaisia lisäosia jotka näyttäisivät raportteja tai mahdollistavat raporttien tulostamisen tai tallentamisen vaikka PDF-muodossa. Käyn seuraavaksi läpi Timeon mahdollista jatkokehitystä uudeksi sovellukseksi ja asioita, jotka liittyvät Timeon julkaisuun avoimena lähdekoodina.

4.1 Timeon kehitys omaksi sovellukseksi eri alustoille

Timeon jatkokehityssuunnitelmissa on kehittää Timeosta oma itsenäinen sovellus ainakin iPhoneille. Timeon yksi alkuperäisistä tavoitteista oli kehittää siitä yhteensopiva ja helposti käytettävä mobiililaitteiden selaimilla, kuten iPhoneen Safari-selaimella. Timeon kehittyessä monipuolisemmaksi, mobiililaitteiden selainten tuki päätettiin jättää vähemmälle huomiolle. Erillisen sovelluksen tekeminen voisi olla nopeampaa, kuin yrittää kehittää verkkosovelluksesta yhteensopiva. Timeon ensimmäisen käyttövalmiin version valmistuessa huomattiin yllättäen, että Timeo toimii varsin hyvin nykyaikaisten mobiililaitteiden selaimilla, varsinkin iPhoneen selaimella. iPhoneen selaimen JavaScript-tuki on sen verran hyvä, että Timeon kaikki ominaisuudet toimivat selaimella. Timeosta saisi täysin käytettävän iPhoneen selaimella, ainoastaan tekemällä sille oma CSS-tyylitiedosto, joka määrittelee Timeolle uuden ulkoasun iPhoneen pienelle näytölle. Uuden sovelluksen tarkoituksena onkin tehdä oma itsenäinen sovellus iPhoneelle, joka ei vaadi yhteyttä Internetiin toimiakseen, vaan tietokannan tiedot säilytetään iPhoneen muistissa.

Toinen Timeon kehitysidea on kehittää Timeosta yksinkertainen Macin OS X-käyttöjärjestelmän Dashboard-widgetti. Widgetit ovat eräänlaisia miniohjelmia, jotka toimivat erillisessä käyttöjärjestelmän Dashboard-näkymässä (ks. kuva 7). Widgetit toimivat samoilla teknologioilla kuin Timeo ja muut verkkosivut ja verkkosovellukset, eli ne koodataan käyttäen HTML:lää, CSS:ää ja JavaScriptiä. Widgetit käyttävät samaan WebKit-teknologiaa, mitä Safari-selainkin, joten Timeon siirtäminen Widget-muotoon ei ole kovin suuri työ. Timeon Widget-sovellus voisi olla yksinkertainen käyttöliittymä, jonka asetuksina on käyttäjän tunnukset ja palvelimen osoite, jolla Timeo sijaitsee. Käyttöliittymänä voisi toimia suoraan esimerkiksi Timeon Kellokortti-osa. Näin käyttäjä voisi käyttää Timeota nopeasti suoraan käyttöjärjestelmän Dashboard-näkymästä, eikä selaimen avaamista ja Timeoon erikseen kirjautumista tarvittaisi.



Kuva 7. Kuvakaappaus OS X-käyttöjärjestelmän Dashboard-näkymästä ja sillä sijaitsevista Widgeteistä.

4.2 Timeon julkaiseminen avoimena lähdekoodina

Timeo tullaan julkaisemaan avoimena lähdekoodina. Alkuperäisenä ideana oli, että Timeo tarjotaan mainosrahoitteisena käyttäjälle ilmaisena palveluna. Projektin nimi oli silloin vielä Kellota.fi. Palvelun ylläpitämiseen ei kuitenkaan ollut resursseja. Palvelun pitäminen olisi myös vaatinut suurempaa vastuuta tietoturvan ja esimerkiksi käyttäjien tietojen varmuuskopioinnin suhteen. Lisäksi markkinoilla on jo olemassa muutamia vastaavia mainosrahoitteisia palveluita. Ei ole myöskään varmaa, että palvelun rahalliset kustannukset katettaisiin pelkästään mainostuloilla. Timeo on myös projektinimi avoimen lähdekoodin projektille. Timeon tarkoitus on, että se voidaan ottaa lisäosaksi jo valmiille verkkosivustoille esimerkiksi nimellä Ajanseuranta. Timeon lähdekoodia muokkaamalla, käyttöönottaja voi yhdistää Timeon käyttäjätietokannan jo olemassa olevan verkkosivuston tietokantaan, jolloin Timeosta tulee osa sivustoa.

Avoimen lähdekoodin projektiin osallistuminen on vapaaehtoista ja jokainen voi valita itselleen sopivan tehtävän projektista. Kuka vain voi osallistua avoimen lähdekoodin projektiin ja lähteä projektista koska tahansa. Avoimen lähdekoodin projektin työvoima on hajautettua ja ilmaista. Työtä ei valvota samaan tapaan kuin perinteisessä yritysorganisaatiossa. Ideaalitapauksessa avoimen lähdekoodin projekti pyörii ohjelmiston ydinlähdekoodin ympärillä. Tätä lähdekoodia levitetään vapaasti yleensä Internetin kautta ja se on kaikkien saatavilla. Lähdekoodi on vapaasti muokattavissa ja käytettävissä jokaisen omaan vapaaseen käyttöön. Lähdekoodia voi käyttää yleensä melkein mihin tahansa. Siitä voidaan tehdä jopa oma ohjelma, jonka lähdekoodi ei ole kaikkien saatavilla. GPL-lisenssi²² rajoittaa kuitenkin tätä oikeutta. (Weber 2004, 62.)

22 GNU General Public License, GNU yleinen lisenssi.

GNU GPL on vapaa lisenssi ohjelmistoille ja muille töille. Useimpien ohjelmistojen ja töiden lisenssit on suunniteltu rajoittamaan töiden muuttamista ja jakamista. GPL-lisenssi taas takaa käyttäjälle vapauden jakaa ja tehdä muutoksia työhön. GPL-lisenssi antaa käyttäjälle laillisen luvan levittää ohjelmaa ilmaiseksi tai käyttää ohjelmiston lähdekoodin osia omissa ohjelmissaan. Ohjelmaa saa myös myydä maksua vastaan. Jos ohjelmaa levitetään, ohjelman lähdekoodi on kuitenkin annettava eteenpäin lukukelpoisena ja GPL-lisenssillä. GPL-lisenssi ei takaa käyttäjälle takuuta ohjelmiston toiminnasta. Lisenssi vaatii että kaikkiin alkuperäisen työn muunnelmiin on merkittävä, että ne ovat muunnoksia alkuperäisestä työstä. (GNU General Public License 2007.)

GPL-lisenssi tulee todennäköisesti myös Timeo-projektille. Koska emme itse vie projektia kaupalliselle tielle, niin ei ole mielekästä että annamme työmme jollekin yksittäiselle taholle, joka kehittäisi Timeosta vain sitä itseään hyödyttävän ja kaupallisesti eteenpäin levitettävän palvelun. Timeon on pysyttävä jatkossa avoimena ja siitä on oltava hyötyä mahdollisimman monelle. Yksittäisen tahon kehittäessä projektia itsenäisesti vain omiin kaupallisiin tarkoituksiinsa, projekti ei kehity tai kehityksen tulokset eivät ole kaikkien saatavilla. GPL-lisenssillä pyritään estämään tämä, sillä lisenssi vaatii että työtä saa levittää vain, jos sen lähdekoodi on vapaasti saatavilla. Kaupalliset organisaatiot voivat kyllä käyttää Timeota myös ilman sen tuotteistamista kaupallisiin tarkoituksiin.

Kaupalliset organisaatiot käyttävät avoimen lähdekoodin ohjelmia, koska ne voivat muokata ohjelmista omiin käyttötarpeisiinsa soveltuvia. Samalla kun ne käyttävät ja kehittävät ohjelmia omiin tarkoituksiinsa, kaupalliset organisaatiot voivat antaa kehitystyön tuloksia takaisin avoimen lähdekoodin yhteisölle. (Weber 2004, 72.)

Timeo-projektin mahdollista jakautumista²³ useampiin erillisiin projekteihin emme kuitenkaan voi estää, emmekä välttämättä haluakaan estää, sillä haluamme että työstämme on hyötyä mahdollisimman monelle. Jakautumisessa ei siinä mielessä ole harmia, sillä GPL-lisenssi velvoittaa, että jakautuneissakin projekteissa on oltava lähdekoodi kaikkien saatavilla, sillä projekti on luotu alkuperäisen GPL-lisenssin alaisen projektin pohjalta. Alkuperäisen projektin GPL-lisenssi siirtyy automaattisesti lisenssin mukaan myös jakautuneille projekteille.

23 englanniksi *Forking*

Kehittäjä jolla on eri näkemys kuin muilla siitä mihin avoimen lähdekoodin projektia viedään voi ottaa projektin ydinkoodin, lisätä siihen omat haluamansa muutokset ja luoda alkuperäisestä avoimen lähdekoodin projektista uuden jakautuneen projektin. Tällaiset jakautuneet projektit voivat saada kehittäjiä vanhalta projektilta. Avoin lähdekoodi takaa oikeuden luoda jakautuneita projekteja. Joidenkin mielestä tämä kuuluu oleellisesti avoimen lähdekoodin projektien vapauteen käyttää lähdekoodia. (Weber 2004, 64.)

Jakautuneet avoimen lähdekoodin projektit voidaan nähdä vapaan ohjelmiston vapauden ilmentymänä, mutta myös heikkoutena. Ne monistavat kehitystyön ponnistelut ja aiheuttavat hämmennystä vapaiden ohjelmien käyttäjien keskuudessa, sillä käyttäjät joutuvat valitsemaan mitä jakautuneen projektin kehittämää ohjelmaa käyttävät. Kehittäjien keskuudessa projektien jakauttaminen myös huonontaa siihen ryhtyneiden kehittäjien mainetta avoimen lähdekoodin yhteisössä. Esimerkki samasta projektista jakautuneista avoimen lähdekoodin projekteista ovat tietokoneen Linux-käyttöjärjestelmät Debian ja Debaniin pohjautuva Ubuntu. Näiden projektien eri kehittäjäryhmien välit ovat hyvät, mutta joissain jakautuneissa projekteissa kehittäjien välit voivat olla hyvinkin katkerat. (Fork (software development) 2009; Ubuntu 2009.)

Kun Timeo julkaistaan avoimena lähdekoodina, voimme saada apua sen kehitystyöhön. Samalla voimme oppia toisten kehittäjien tekemästä työstä. Timeo-projektissa päävastuualueeni on ohjelmoinnissa. En voi sanoa, että olisin kovin kokenut ohjelmoija, varsinkaan Javascript-kielellä, joten minua kiinnostaa vastaako tuottamani koodi kokeneempien ohjelmoijien laatutasoa ja mitä kirjoittamassani JavaScript-koodissa on mahdollisesti paranneltavaa. Projekti etenee myös nopeammin ja Timeon laatu paranee, kun kehittäjiä on enemmän.

Avoimen lähdekoodin kehittäjät etsivät tilaisuutta luoda jotain uutta ennennäkemätöntä tai tehdä jokin vaikea asia elegantisti. Kehittäjät hyötyvät oppimalla uutta ja parantavat omia kykyjään, vaikka ovatkin mukana vapaaehtoisesti ilman rahallista palkkiota. Samalla kun oppii, voi auttaa projektin kehittymistä, mikä tuo kehittäjälle mielihyvää. Kehittäjät hakeutuvat mielellään projektien pariin, joista näyttää tulevan jollain tavalla merkittäviä ja jotka tarjoavat kehittäjille mielenkiintoisia tehtäviä. (Weber 2004, 73-74.)

Timeo-projekti on julkaisukunnossa arviolta kesällä 2009 ja se tullaan todennäköisesti julkaisemaan suositussa SourceForge.net-nimisessä²⁴ avoimen lähdekoodin ohjelmistokehittäjien yhteisössä. Toinen mahdollisen vastaava avoimen lähdekoodin julkaisupaikka on Google Code²⁵.

SourceForge.net on verkkosivusto ja yhteisö, joka tarjoaa ilmaisen keskitetyn paikan ohjelmistokehittäjille säilyttää, hallita ja ylläpitää avoimen lähdekoodin projektejaan ja niiden kehitystä. SourceForge:lla on noin kaksi miljoonaa rekisteröitynyttä käyttäjää sekä yli 180 000 projektia. SourceForge tarjoaa kehittäjille säilytystilan lisäksi versionhallinnan sekä wikin²⁶ projektin dokumentointia varten. Lisäksi projekteille tarjotaan oma alidomain, joka on muotoa <http://projekti.sourceforge.net>. SourceForge tarjoaa myös pääsyn MySQL-tietokantaan. SourceForge-yhteisössä kehittäjät voivat saada palautetta projekteistaan. Aktiivisimmat projektit saavat myös enemmän näkyvyyttä sivustolla. SourceForge.net on paikka, jossa avoimen lähdekoodin kehittäjät vierailevat säännöllisesti. Siellä seurataan minkälaisia projekteja on meneillään ja mitä muut kehittäjät tekevät sekä missä projekteissa he ovat mukana. Useat projektit SourceForge.net:ssa eivät ole teknisesti suuria ja niihin osallistuu vain muutama kehittäjä. Osa projekteista on hylättyjä, eikä niitä enää kehitetä. (SourceForge.net 2009;Weber 2004, 66.)

Timeon julkaisukuntoon saattaminen vaatii vielä ainakin lähdekoodin siistimistä ja selkeämpää dokumentointia, jotta lähdekoodi on muiden kehittäjien helposti luettavista. Hyvä dokumentointi on tärkeää avoimen lähdekoodin yhteisöissä, sillä samalla projektilla voi olla hyvinkin paljon eri kehittäjiä. Lisäksi kehittäjiä tulee mukaan projekteihin projektin eri kehitysvaiheissa, joten uusien kehittäjien tulee saada tarvittava tieto nopeasti ja helposti voidakseen osallistua projektiin. Dokumentointiin kuuluu kommenttien lisääminen lähdekoodin sekaan selventämään esimerkiksi eri funktioiden toimintaa ja koodin rakennetta. Pelkän koodikielen ymmärtäminen voi olla hyvin työlästä ilman hyvää dokumentointia.

24 <http://www.sourceforge.net>

25 <http://code.google.com>

26 Wiki on verkkosivu, jonka sisältöä käyttäjät voivat itse muokata.

5 YHTEENVETO

Vastasin Timeo-projektissa ohjelmoinnista sekä teknisen puolen suunnittelusta ja toteutuksesta. Projektin aikana JavaScript-osaamiseni kehittyi paljon, sillä Timeo on suurimmaksi osaksi JavaScript-sovellus. Olen aiemmissa verkkosivu-projekteissa käyttänyt JavaScriptiä vain hyvin vähän, joten käytännössä se oli minulle vieras kieli. Päätös julkaista Timeo avoimena lähdekoodina loi paineita kirjoittaa mahdollisimman oikeaoppista ja laadukasta koodia, mikä pakotti oppimaan JavaScriptistä ja muista teknologioista, kuten PHP:stä, jQuerystä ja CSS:stä, kaiken mahdollisen. Kyseenalaistin kaikki vanhat tapani toteuttaa jokin asia ja otin selvää siitä, miten asian voisi mahdollisesti tehdä paremmin. Pienten yksityiskohtien viilaamiseen kului paljon aikaa. Timeon lähdekoodi on kaikkien nähtävillä, sillä Timeo on avointa lähdekoodia. Tavoitteeni oli että kirjoittamani koodi on mahdollisimman ammattimaista ja antaa hyvän kuvan osaamisestani. Mielestäni laatuun panostaminen kannatti ja se näkyy lopputuloksessa. Opin Timeon aikana myös ammattimaisempaa työskentelyä projektin parissa. Kehitystyössä otettiin käyttöön versionhallintapalvelin yhteistyön helpottamiseksi, mikä oli minulle uusi asia.

Laatutavoitteista johtuen Timeo-projektin aikataulu venyi monta kertaa ja jotkin suunnitellut ominaisuudet eivät ehtineet valmistua. Opin, että vastaavissa ohjelmointiprojekteissa tiukan aikataulun tekeminen ei välttämättä ole hyödyllistä, sillä vastaantulevat ongelmat ovat usein odottamattomia ja niiden ratkaisuun voi mennä hyvinkin paljon aikaa. Uskon että tulevaisuudessa suoriudun ongelmista kuitenkin nopeammin Timeosta saamani kokemuksen ansiosta.

Timeo-projekti lähti liikkeelle siitä, että olin etsinyt tuloksetta ajanseurantaohjelmaa, joka olisi ilmainen ja helppokäyttöinen. En ollut aluksi varma, että riittäisikö taitoni sellaisen sovelluksen toteuttamiseen. Lopulta kysymys oli vain ajasta, eikä niinkään taidosta. Kaikki on mahdollista, kunhan on aikaa asioiden oppimiseen. Oli ilo huomata, että työn tuloksena syntyi työkalu, josta on itsellekin paljon hyötyä. Timeo on mielenkiintoinen projekti, jonka kehitystyössä aion olla mukana jatkossakin.

LÄHTEET

Asleson, Ryan & Schutta, Nathaniel, T. 2006. Foundations of Ajax. Berkeley, CA: Apress.

Document Object Model, 2009. Wikipedia. [Verkkodokumentti].
<http://en.wikipedia.org/wiki/Document_Object_Model> (luettu 23.5.2009).

Fork (software development). 2009. Wikipedia. [Verkkodokumentti].
<[http://en.wikipedia.org/wiki/Fork_\(software_development\)](http://en.wikipedia.org/wiki/Fork_(software_development))> (luettu 2.4.2009).

Garrett, Jesse, James 18.2.2005. Ajax: A New Approach to Web Applications. [Verkkodokumentti].
<<http://www.adaptivepath.com/ideas/essays/archives/000385.php>> (luettu 20.4.2009).

GNU General Public License. 2007. Boston, MA: Free Software Foundation, Inc. [Verkkodokumentti]. <<http://www.gnu.org/copyleft/gpl.html>> (luettu 1.4.2009).

Hovi, Ari, Huotari, Jouni & Lahdenmäki, Tapio. 2003. Tietokantojen suunnittelu & indeksointi. Jyväskylä: Docendo Finland Oy.

jQuery. 2009. Wikipedia. [Verkkodokumentti]. <<http://en.wikipedia.org/wiki/JQuery>> (luettu 3.4.2009).

SourceForge.net. 2009. Wikipedia. [Verkkodokumentti].
<<http://en.wikipedia.org/wiki/SourceForge.net>> (luettu 1.4.2009).

Ubuntu. 2009. Wikipedia. [Verkkodokumentti]. <<http://en.wikipedia.org/wiki/Ubuntu>> (luettu 2.4.2009).

Web application. 2009. Wikipedia. [Verkkodokumentti].

<http://en.wikipedia.org/wiki/Web_application> (luettu 18.4.2009).

Weber, Steven 2004. The Success of Open Source. Cambridge, MA: Harvard University Press.

XMLHttpRequest. 2009. Wikipedia. [Verkkodokumentti].

<<http://en.wikipedia.org/wiki/XMLHttpRequest>> (luettu 16.5.2009).