HAAGA-HELIA
ammattikorkeakoulu

Ajax ja käyttöliittymäelementtien suunnittelu

Kimmo Löfman

Opinnäytetyö
Tietojenkäsittelyn koulutusohjelma
2013

HAAGA-HELIA
ammattikorkeakoulu

Tietojenkäsittelyn koulutusohjelma

Tiivistelmä

15.1.2013

Tekijä tai tekijät Kimmo Löfman	**Ryhmätunnus tai aloitusvuosi** 2009
Raportin nimi Ajax ja käyttöliittymäelementtien suunnittelu	**Sivu- ja liitesivumäärä** 24 + 9
Opettajat tai ohjaajat Sirpa Marttila	

Opinnäytetyön tavoitteena oli tuottaa kehittämissuunnitelma kohdeyrityksen toteuttaman sovelluskehikon dynaamisten ominaisuuksien parantamiseksi. Sovelluskehikkoon on aiemmin toteutettu perustoiminnallisuus asynkronisten kutsuille, joten kehityssuunnitelma painottuu yksittäisten elementtien suunnitteluun. Niiden suunnittelussa käytettävyyden parantamisella on erityinen painoarvo.

Teoriaosuudessa pyritään esittämään teoriat ja perusteet, mitkä ovat kehityssuunnitelman taustalla. Siinä käsitellään yleisesti käytettävyyttä, rikkaita selainkäyttöliittymiä sekä Ajax-teknologiaa. Niiden lisäksi perehdytään tarkemmin verkkosovellusten yleisiin suunnitteluperiaatteisiin, Ajax-sovellusten ominaisuuksiin ja suunnittelumalleihin.

Empiirisessä osuudessa käsitellään projektissa tuotettua suunnitelmaa. Minkä reaalimaailman ongelman ratkaisemiseen ne on tarkoitettu ja mihin suunnittelumalliin ne perustuvat. Suunniteltujen elementtien toiminta kuvataan karkealla tasolla kaavioiden avulla.

Lopuksi pohditaan suunnitelman vaikutusta käytätettävyyteen parantamiseen ja suunnittelumallien hyväksikäyttöä kehityssuunnitelmien teossa. Opinnäytetyöprosessin ja oman oppimisen arviointi on myös osa pohdintaa ja johtopäätöksiä.

Asiasanat
Ajax, suunnittelumallit, käytettävyys

HAAGA-HELIA
University of Applied Sciences

Abstract

15.1.2013

Degree Programme in Information Technology

Authors Kimmo Löfman	**Group or year of entry** 2009
The title of thesis **Ajax and designing of user interface elements**	**Number of pages and appendices** 24 + 9
Supervisor(s) Sirpa Marttila	

The aim of this thesis was to produce a development plan for improving the dynamic features of the Java framework made by company X. The framework has basic functionalities for handling asynchronous requests, thus the main focus of the development plan was the single elements with high usability.

In the theoretical part of the thesis, the rationale behind the development plan was presented. This part of the thesis covered usability, rich user interfaces and the Ajax technology briefly. It focused on common web design patterns, the features of Ajax applications and the Ajax design patterns.

In the empirical part of the thesis, the development plan was explained focusing on which real life problems the new planned element should resolve and which design patterns were used in the development plan. Graphical modeling was used for defining the flow of the new elements.

The results of the thesis showed what kind of effects the development plan has for usability and how to use design patterns when making development plans.

Key words
Ajax, usability, design patterns

Sisällys

1 Johdanto

Standardien teknologioiden puute rajoitti pitkään käytettävyydeltään parempien verkkosovellusten kehittämisen selaimiin. Selaimeen asennettavien liitännäisten avulla oli aiemminkin mahdollista toteuttaa työpöytäsovellusmaisia sovelluksia. Tarve oli kuitenkin kevyemmälle ja alustariippumattomalle teknologialle, mikä ei sitoisi kehittäjiä tiettyyn teknologiaan. Ajaxista alettiin puhua terminä vasta Googlen esiteltyä Gmail sähköpostipalvelunsa, vaikka teknologiat Ajaxin taustalla olivatkin kehitetty aiemmin. Googlen toteuttamat sovellukset ovat toimineet esimerkkinä muille, kuinka Ajax teknologian avulla on mahdollista toteuttaa monimutkaisia ja samalla erittäin käyttäjäystävällisiä sovelluksia selaimiin ilman liitännäisiä.

Tutkimuksen tarkoituksena on laatia kehityssuunnitelma kohdeyrityksessä kehitetyn JEE-sovelluskehikkotuotteen Ajax-ominaisuuksien parantamiseksi. Sovelluskehikkoon on jo aiemmin toteutettu perus Ajax-toiminnallisuus. Se pystyy vastaanottamaan ja käsittelemään asynkroniset kutsut sekä palauttamaan XML tai merkkijonopohjaisen vastauksen. Tarkoituksena ei ole laatia kehityssuunnitelmaa koko esityskerroksen muuttamiseksi Ajaxilla toimivaksi. Siinä keskitytään uusien käyttöliittymäkomponenttien kehittämiseen sekä nykyisten toiminnallisuuksien laajentamiseen käyttämään asynkronisia kutsuja lomakkeiden lähetyksen tai hyperlinkkien käytön sijaan.

Kehityskohteena olevaa sovelluskehikkoa on käytetty verkkopalveluiden alustana toimialoilla, joilla on korkeat vaatimukset tietoturvan ja saavutettavuuden suhteen. Verkkopalveluiden on pitänyt tukea mahdollisimman montaa selainta, sekä niiden eri versioita. Verkkopalveluiden on pitänyt myös toimia ilman, että käyttäjä on sallinut JavaScriptin suorittamista selaimessa. Myös vanhempien selain versioiden JavaScript-toteutuksissa on ollut ongelmia. Selaimet ovat kehittyneet, selainkanta on uudistunut ja mielipiteet ovat muuttuneet sallivammaksi JavaScriptin suhteen, koska käytettävyyden parantaminen verkkopalveluissa on vihdoin nähty myös kilpailutekijänä. Edellä mainitut syyt ovat luoneet osaltaan tarpeen tutkimuksen kohteena olevan kehityshankkeen käynnistämiselle.

2 Käytettävyys ja rikkaat selainkäyttöliittymät

2.1 Mitä käytettävyys on

Käytettävyydestä puhuttaessa sillä tarkoitetaan usein tuotteen tai palvelun (myöhemmin tuote) ominaisuutta. Arviointiperuste on usein hyvin mustavalkoinen, joko tuote on käytettävä tai sitten ei. Monet tutkijat ja kirjailijat yrittävät tiivistää teoriansa käytettävyydestä yhteen lauseeseen, vaikka kyseessä onkin useammasta tekijästä koostuva monimutkainen asia. Rubinin ja Chisnellin (2008, 4) mukaan tuote on käytettävä, jos käyttäjä voi tehdä mitä hän haluaa, miten hän arvioi voivansa tehdä sen esteettä ja ilman epäröintiä tai kysymyksiä. Käytettävyyttä voidaan myös miettiä menetelmä- ja teoriakentän kautta, minkä tarkoituksena on saada käyttäjän ja laiteen yhteistoiminta tehokkaammaksi ja käyttäjän kannalta miellyttävämmäksi (Sinkkonen, Kuoppala, Parkkinen & Vastamäki 2006, 17). Käytettävyyteen liitetään ominaisuuksia kuten opittavuus, tehokkuus, muistettavuus, virheettömyys ja tyytyväisyys (Usability.gov). Rubin ja Chisnell(2008, 4) lisäävät siihen myös hyödyllisyyden ja esteettömyyden.

Höydyllisyys kuvaa sitä, onko käyttäjällä mahdollisuus saavuttaa tuotteella hänen asettamansa tavoitteet (Rubin & Chisnell 2008, 4). Tämä myös vaikuttaa käyttäjän motivaatioon käyttää tuotetta. Muut käyttävyyteen vaikuttavat ominaisuudet menettävät merkityksenä, jos käyttäjä ei ole halukas käyttämään tuotetta. Hyödyllisyyttä ei yleensä kirjata käyttävyysvaatimuksiin, koska tuotteen pitäisi lähtökohtaisesti sopia siihen tehtävään, mihin se on tarkoitettu (Sinkkonen ym. 2006, 15).

Tehokkuudella tässä tarkoitetaan sitä, kuinka nopeasti ja tarkasti käyttäjä pystyy suoriutumaan halutusta tehtävästä. Virheettömyydellä taas viitataan siihen toimiiko tuote siten, kuten käyttäjä olettaa sen toimivan ja voiko hän käyttää sitä kuten aikoi. Tehokkuudelle ja virheettömyydelle voidaan asettaa konkreettisia mittareita. Tehokkuutta voidaan suoraan mitata toimintaan käytetyllä ajalla. Virheettömyyttä voidaan mitata määrällisesti virheprosentilla, esimerkiksi 95 prosenttia käyttäjistä suoriutuu annetusta tehtävästä kymmenessä minuutissa. (Rubin ja Chisnell 2008, 4.)

Opittavuus tarkoittaa sitä, kuinka nopeasti käyttäjä oppii käyttämään tuotetta tietyllä tasolla ennalta määritellyn koulutuksen jälkeen. Opittavuuteen tässä yhteydessä liitetään myös muistettavuus, eli kuinka nopeasti harvakseltaan käyttävälle palautuu mieleen tuotteen käyttö. (Rubin ja Chisnell 2008, 4.)

Tyytyväisyys kuvaa käyttäjän havaintoja, tuntemuksia ja mielipidettä tuotteesta. Käyttäjä on halukkaampi käyttämään tuotetta, jonka käyttöön hän on tyytyväinen kuin sellaisen tuotteen käyttöön, minkä käyttö on epätyydyttävää. Tyytyväisyyttä tyypillisesti mitataan kyselyillä. (Rubin ja Chisnell 2008, 4.)

Esteettömyydellä viitataan siihen, että onko tuote kaikkien käytettävissä, mukaan lukien henkilöt joiden toimintaa haittaa jokin rajoite kuten esimerkiksi näkörajoite. Englannin kielistä termiä käytetään myös kuvaamaan tuotteen saavutettavuutta, eli onko se käytettävissä käyttäjän haluamalla hetkellä. (Rubin ja Chisnell 2008, 5.)

Käytettävyyden ominaisuuksille on määriteltävissä erilaisia mittareita, joilla voidaan arvioida käytettävyyttä. Mittareiden lisäksi käytettävyyden arviointiin löytyy muitakin menetelmiä. Riihiaho(2000, 223-224) jakaa arviointimenetelmät kokeellisiin käyttäjätesteihin ja ilman käyttäjiä tehtäviin asiantuntija-arvioihin. Käytettävyyttä on käsiteltävä monelta katsantokannalta ja arviointimenetelmien yhdistelmä tuottaa yleensä kattavimman arvion käytettävyydestä sovelluksen kehitysvaiheessa. Lopullisen arvion käytettävyydestä tekevät kuitenkin tuotteen varsinaiset käyttäjät.

Asiantuntija-arviomenetelminä käytetään heuristista arviota ja kognitiivista läpikäyntiä. Heuristinen arvio on näistä menetelmistä suositumpi (Riihiaho 1998, 4). Asiantuntija-arvio on projektin ulkopuolisen, yleensä käytettävyysasiantuntijan, suorittama katselmoinnin tulos. Toimialatuntemus ei ole käytettävyysasiantuntijalle välttämätöntä, mutta suotavaa. Asiatuntijat perustavat arvionsa yleisesti hyväksyttyihin käytettävyyden periaatteisiin l. heuristiikoihin. Ne pohjautuvat tutkimusaineistoon sekä aiempaan kokemukseen (Rubin & Chisnell 2008, 19). Kognitiivisella läpikäynnillä pyritään löytämään käyttäjän tyypillisesti kohtaamat ongelmat tuotteen käytön opettelemisessa (Riihiaho 1998, 4).

2.2 Rikkaat selainkäyttöliittymät

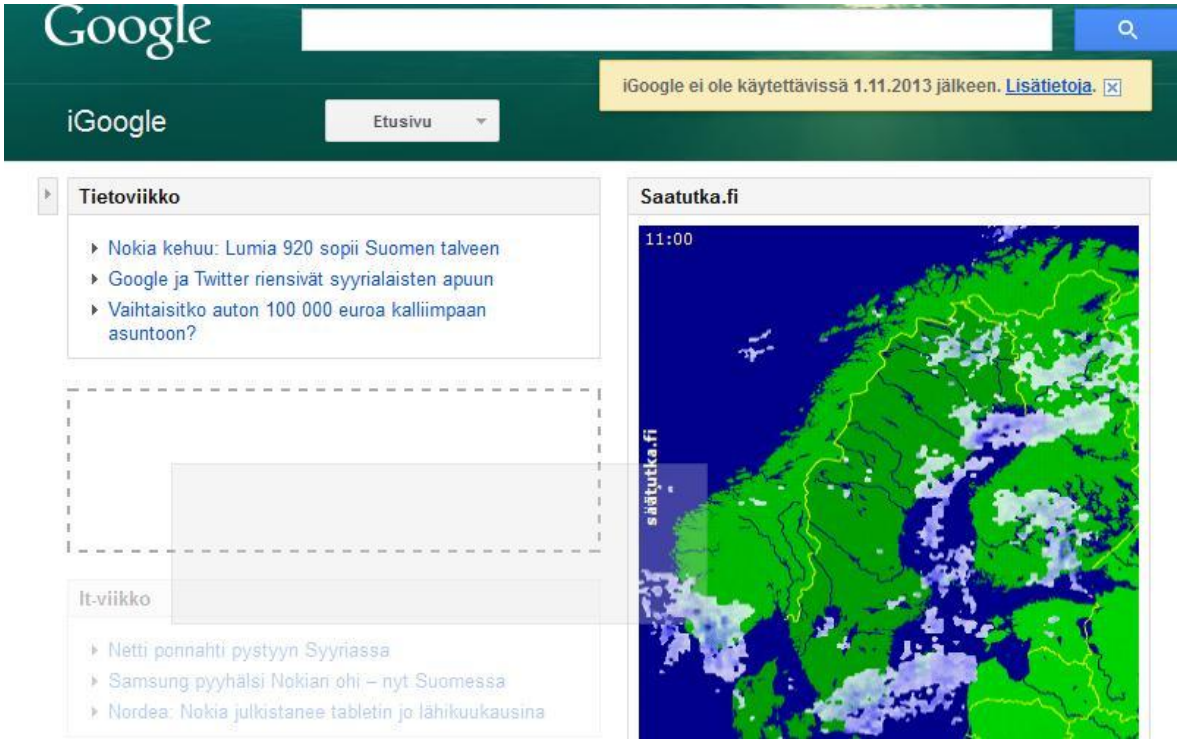
Rikkaista selainkäyttöliittymistä puhuttaessa tarkoitetaan selaimessa toimivaa sovellusta, jonka vuorovaikutusominaisuudet ovat laajempia kuin perinteisen verkkosivuston, sekä niiden ominaisuudet muistuttavat työpöytäohjelmistoista tuttuja ominaisuuksia (Crane, Pascarello & James 2006, 5). Tästä johtuen yhtenä perustavana ideana rikkaiden selainkäyttöliittymien syntyyn voidaan pitää tarvetta saada työpöytäohjelmistot, kuten tekstinkäsittely- ja taulukkolaskentaohjelmistot, toimimaan selaimessa (Mahemoff 2006, 9).

Rikkaat selainkäyttöliittymät voidaan jakaa kahteen ryhmään toteutustavan perusteella, Ajax-sovelluksiin ja selaimeen asennettavien liitännäisten avulla toimiviin sovelluksiin. Ajax-sovellukset käyttävät standardeja selainten tukemia teknologioita eivätkä vaadi JavaScriptin suorittamisen sallimisen lisäksi muita asennus tai asetustoimia käyttäjältä. Liitännäisteknologioita ovat esimerkiksi Adoben Flash, Javan Webstart ja JavaFX ja Microsoftin Silverlight. Ajaxin vahvuutena liitännäisteknologioihin verrattuna voidaan pitää asennusvapautta. Tosin liitännäisillä voidaan toteuttaa näyttävämpää grafiikkaa ja ne mahdollistavat esimerkiksi videoiden suoratoiston.

2.3 Rikkaiden selainkäyttöliittymien suunnittelumallit

Sovelluskehityksestä tutut suunnittelumallit ovat yleisiä ja uudelleenkäytettäviä ratkaisumalleja usein esiintyviin ongelmiin. Ne kehittyvät ajan myötä parhaista käytännöistä suunnittelumalleiksi. Asiayhteydestä voidaan päätellä, minkä ongelmien ratkaisemiseen ne kulloinkin on tarkoitettu. Rikkaiden selainkäyttöliittymien suunnittelumalleilla pyritään ratkaisemaan selaimessa toimivien käyttöliittymien suunnitteluun liittyviä yleisiä ongelmia. Scottin ja Neilin (2009, xiv) mukaan suunnittelumallit on jaettavissa kuuden periaatteen mukaan.

Ensimmäinen periaate on suora muokkaus. Sen mukaan, siellä missä tietoa esitetään, pistäisi pystyä esitettyä tietoa myös muokkaamaan ilman siirtymistä erilliselle muokkaussivulle. Tähän periaatteeseen kuuluvat myös suunnittelumallit sivulla olevien elementtien raahaukseen sekä valintaan sivuilta (Scott & Neil 2009, 2). Kuviossa 1 on tämän periaatteen mukainen toteutus Googlen iGoogle palvelussa. Elementin voi valita, sitä voi muokata ja sen voi uudelleen sijoittaa haluamalleen paikalle.



Kuvio 1 Sivun muokkaus iGoogle palvelussa

Toisen periaatteen mukaan sivun pitäisi olla mahdollisimman kevyt. Suunnittelussa pyritään ymmärtämään käyttäjän aikomukset ja tarjota vain asiayhteyteen liittyvät työkalut (Scott & Neil 2009, 295). Niillä tarkoitetaan työpöytäsovelluksista tuttuja, oikealla hiirenpainikkeella avattavia valikoita. Valikoiden ja työkalujen toteutukseen verkkosivuilla on monia mahdollisuuksia, kuten elementin vieressä aina näkyvät työkalurivit, hiiren ylityksellä aktivoituvat valikot yms. (Scott & Neil 2009, 81). Kuviossa 2 on Helsingin Sanomien artikkelin perässä näkyvissä olevat työkalut, artikkelin jakamiseen, tallentamiseen, kommentointiin, tulostukseen ja lähettämiseen sähköpostilla.

Kuvio 2 Työkalut Helsingin Sanomien artikkelin jälkeen

Kolmas periaate käsittelee sivulla pysymistä. Sillä tarkoitetaan sitä, ettei käyttäjän toimintojen kulkua katkaista sivun uudelleen lataukseen. Uudelle sivulle siirtymisen sijaan sivun päälle voidaan tuoda ns. modaalinen ikkuna (Scott & Neil 2009, 102). Lisäksi informaatiota tai toimintoja voidaan upottaa sivuille ja tuoda esille tarvittaessa animaatioiden avulla. Animaatioiden avulla voidaan myös sivun virtuaalista kokoa kasvattaa. Kuviossa 3 on VR:n verkkokaupan sivulle upotettu kalenteri, joka ponnahtaa päällimmäiseksi elementiksi käyttäjän painettua kalenterikuvaketta.

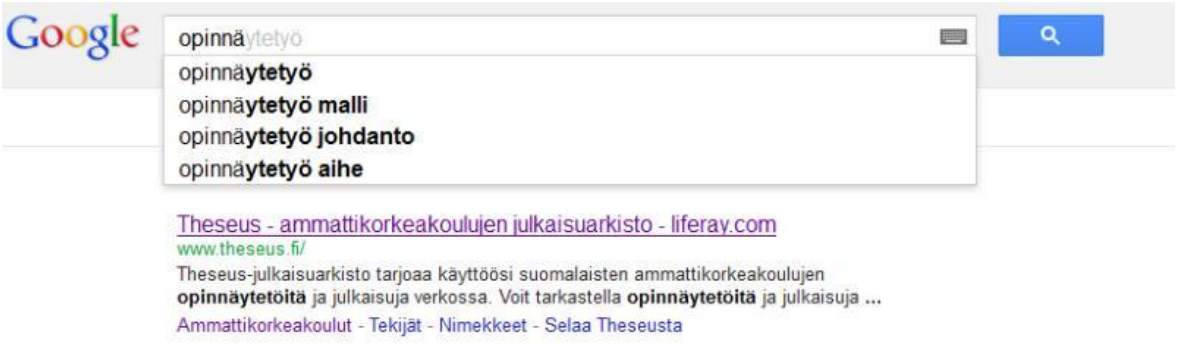
Kuvio 3 Sivulle upotettu kalenteri

Neljäs periaate esittelee staattiset ja dynaamiset kutsut. Sovellusten monet ominaisuudet jäävät usein käyttäjiltä huomaamatta ja sitä myöten käyttämättä. Kutsun tarkoitus on houkutella käyttäjä koettamaan joitain hänelle uutta ja samalla koetetaan parantaa hänen käyttökokemustaan. (Scott & Neil 2009, 296.)

Viides periaate kehottaa käyttämään siirtymiä. Siirtymillä tarkoitetaan visuaalisia tehosteita, joilla voidaan kuvata asioiden välisiä yhteyksiä, selittää tapahtunutta, kiinnittää huomiota, parantaa suorituskykyä ja luoda illuusio virtuaalisesta tilasta (Scott & Neil 2009, 296). Visuaaliset tehosteet voivat myös ärsyttää käyttäjää, joten niiden käytössä on syytä olla varovainen.

Kuudennen periaatteen mukaan käyttäjän toimien pitäisi aiheuttaa välitön vastareaktio. Suunnittelumallit automaattiseen täydennykseen, reaaliaikaiseen hakuun ja ehdotuksiin, esikatseluun sekä edistymispalkkeihin ovat tarkoitettu reagoimaan käyttäjän toimiin ja samalla kun ne luovat elävän ja reagoivan käyttöliittymän (Scott & Neil 2009, 296). Kuviossa 4 on esimerkki reaaliaikaisesta hausta ja ehdotuksesta Googlen hakukoneessa.

Käyttäjän kirjoittaessa tekstiä hakukenttään, hänelle ehdotetaan hakusanoja sanan alun, aiempien tai yleisimpien hakujen perusteella. Samalla hakutuloksiin päivittyy hakusanaa parhaiten vastaava hakutulos.



Kuvio 4 reaaliaikainen haku

2.4 Ajax

Ajax on kokoelma standardeja teknologioita, joiden avulla on mahdollista tehdä asynkroninen kutsu sovelluspalvelimelle ilman sivun uudelleen latausta. Se on lyhenne ja tulee sanoista Asynchronous JavaScript and XML. Lyhenteessä käytetyistä nimistä voi päätellä siihen liittyvän toiminnan ja osan teknologioista. Microsoft esitteli sähköpostiohjelmansa selainversiossa asynkronisten kutsujen käytön huomattavasti ennen kuin termiä Ajax edes alettiin käyttää (Bibeault & Katz 2010, 235).

Keskeinen ominaisuus Ajaxissa on asynkroniset kutsut. Niiden avulla kutsut sovelluspalvelimelle voidaan tehdä taustalla. Sovelluksen käyttö voi jatkua keskeytymättä, kun käyttäjän ei tarvitse odottaa sivun uudelleen latausta. Asynkroniset kutsut eivät ole nopeampia kuin synkroniset, mutta taustalla tapahtuvien kutsujen avulla käyttäjälle muodostuu mielikuva nopeasta ja reagoivasta sovelluksesta. (Hoffman & Sullivan 2008, 5.)

JavaScript on Ajaxin koossa pitävä teknologia. Ilman JavaScriptiä ei olisi mahdollista toteuttaa monimutkaista logiikkaa selaimeen. Muut teknologiat ovat Ajaxin kannalta hyödyttömiä ilman sitä. Sitä tarvitaan asynkronisen kutsun lähettämiseen ja vastauksen käsittelemiseen. Käyttöliittymän rakenteen ja XML:n käsittelyyn tarvitaan myös JavaScriptiä. (Crane ym 2006, 33; Hoffman & Sullivan 2008, 6.)

Asynkroniset kutsut mahdollistaa XMLHttpRequest(XHR) objekti. Sitä voidaan pitää keskeisimpänä komponenttina Ajaxissa. Se on uusimpien selaimien tukema teknologia. Vanhemmissa Microsoftin selaimissa oli oma toteutus. XHR-objektilla on muutama metodi kutsun lähetyksen ja vastauksen käsittelyn lisäksi, joilla voidaan ohjelmallisesti tutkia kutsun tilaa ja reagoida kutsun tilassa tapahtuneisiin muutoksiin. (Hoffman & Sullivan 2008, 6.)

2.5 Ajax-suunnittelumallit

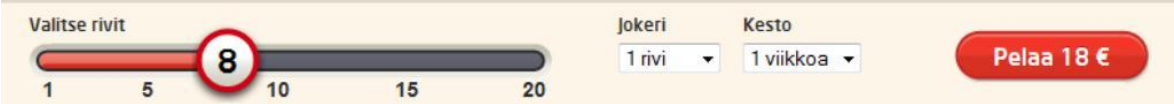
Ajax-suunnittelumallit voidaan jakaa hierarkkisesti neljään ryhmään kohdealueensa mukaan: perusteknologia-, ohjelmointi-, toiminnallisuus- ja käytettävyys- sekä kehitysmalleihin. Kolme ensimmäistä mallia liittyvät itse sovellukseen ja viimeinen prosessiin sovelluksen kehittämiseksi. Perusteknologiamallit toimivat ytimenä, mihin muut mainitut mallit pohjautuvat, vaikka ne muuten ovat melko itsenäisiä. (Mahemoff 2006, 47-48.)

Ajax on itsessään suunnittelumalli, minkä tarkoituksena on ratkaista selaimessa toimivaan käyttöliittymään liittyviä ongelmia. Ajaxia voidaan siis pitää kantana muille perusteknologiaan liittyville suunnittelumalleille. Nämä voidaan ryhmitellä näytön muokkaamisen, palvelimen ja selaimen välisen kommunikaation sekä käyttäjien aikaansaamien ja ajastettujen tapahtumien perusteella omiin ryhmiinsä. Omana ryhmänä voidaan pitää liittymiä ei-standardeihin verkkoteknologioihin. (Mahemoff 2006, 57.)

Ohjelmointiin liittyvien mallien tarkoitus on varmistaa sovelluksen laatu. Ajax ei sinällään tuo tähän laadulliseen vaatimukseen mitään uutta. Ajax on useamman teknologian yhdistelmä, mistä johtuen ohjelmoitiin liittyvissä malleissa kiinnitettään erityisesti huomiota ylläpidettävyyteen, kestävyyteen sekä suorituskykyyn (Mahemoff 2006, 159.)

Toiminnallisuuteen ja käytettävyyteen liittyvät suunnittelumallit ovat käyttäjälle näkyvin Ajaxilla toteutettua rikasta käyttöliittymää. Niissä yhdistyvät perusteknologiamallien yhteydessä mainitut näytön muokkaukseen ja selaimen ja palvelimen väliseen kommunikaatioon liittyvät mallit (Mahemoff 2006, 327). Käytettävyyteen liittyvät mallit käsittävät käyttöliittymäelementit sekä sivuarkkitehtuurin. Käyttöliittymäelementeillä tarkoitetaan työpöytäsovelluksista tuttuja elementtejä kuten liukupalkit ja ennakoiva syöttö.

Sivuarkkitehtuuri liittyy enemmän sivun toimintaan ja tiedon esittämiseen sivulla, kuten esimerkiksi elementtien raahaus, ponnahdusikkunat ja sisältöruudut (Mahemoff 2006, 51). Kuviossa 5 on liukupalkki Veikkauksen verkkosivuilla, mitä liu'uttamalla vaakatasossa valitaan pelattavien rivien määrä arvontaan.



Kuvio 5 Liukupalkki

Kehitykseen liittyvät suunnittelumallit jaotellaan sovelluksen diagnosointiin ja testaukseen liittyviin malleihin (Mahemoff 2006, 52). Diagnosoinnilla tarkoitetaan sovelluskehityksessä tyypillistä lokikirjoitusta, virheen etsimistä ja korjaamista sekä Ajaxille ominaisempaa dokumenttirakenteen dynaamista tutkimista sekä selaimen ja palvelimen välisen liikenteen tarkkailua (Mahemoff 2006, 533). Testauksen suunnittelumalleihin liittyy varsinaisen testauksen lisäksi palveluiden simuloiminen.

2.6 Ajax-sovelluksen ominaispiirteet

Ajax pohjautuu moneen standardiin verkkoteknologiaan, minkä johdosta on mahdollista tuottaa rikas ja vuorovaikutteisen käyttäjäkokemus verkossa (Mahemoff 2006, 10). Tarvittavien teknologioiden tuntemisen lisäksi Ajax-sovelluksien suunnittelu edellyttää niille muutaman ominaisen perusperiaatteen ymmärtämistä.

Selain isännöi sovellusta, ei sisältöä. Perinteisessä verkkosovelluksessa selain toimii vain tyhmänä päätteenä ja sen tehtävänä on muotoilla näytölle palvelimen lähettämä sisältö. Perinteisessä mallissa palvelinpäähän on toteutettu kaikki sovellukseen liittyvä logiikka ja käyttäjäkohtaiseen istuntoon on talletettu siihen liittyvä tieto, kuten esimerkiksi ostoskorin sisältö tai sijaintitieto sovelluksen sivuilla. Ajax-sovelluksissa osa tästä logiikasta on siirretty JavaScriptillä suoritettavaksi selaimeen. (Crane ym. 2006, 17-18.)

Palvelin lähettää tietoa, ei sisältöä. Kokonaan Ajaxilla toteutetuissa sovelluksissa sovelluspalvelimella toimiva osa lähettää vain kerran sivupohjan tarvittavine navigaatioineen ja ominaisuuksineen verkkosivun esittämiseksi. Seuraavilla kutsuilla päivitetään vain

tietosisältöä, toisin kuin perinteiset verkkosovellukset, jotka päivittävät koko sivun. (Crane ym. 2006, 19.)

Ajax-sovellukset perustuvat sulavaan ja jatkuvaan vuorovaikutukseen käyttäjän ja sovelluksen välillä. Ajax-sovelluksissa kutsut lähetetään asynkronisesti, jolloin käyttäjän normaali toiminta ei katkea sivun uudelleen lataukseen ja tiedot sivulla päivittyvät häiritsemättä käyttäjää. Jatkuva vuorovaikutus mahdollistaa esimerkiksi käyttäjän syöttämien tietojen tarkistamisen reaaliaikaisesti, eikä vasta lomakkeen lähetyksen jälkeen sovelluspalvelimella suoritettavassa sovelluksessa. (Crane ym. 2006, 21; Mahemoff 2006, 11.)

Ajax-sovelluksissa korostuu reaaliaikaisuus. Tiedonsiirron nopeutumien on mahdollistanut reaaliaikeisen tiedon esittämisen käyttäjälle (Mahemoff 2006, 11). Sovellusten on mahdollista pyytää jatkuvasti palvelimelta päivitettyä tietoa. Selaimissa toimivien pikaviestisovellusten logiikka perustuu kiertokyselyihin. Kyseinen tekniikka johtuu siitä, ettei sovelluspalvelimelta voida puskea tietoa selaimeen, vaan selaimessa toimivan sovelluksen pitää olla aktiivien osapuoli.

3 Kehitysprojekti

Projektin tarkoitus on tuottaa kehityssuunnitelma kohdeyrityksessä toteutetun sovelluskehikon Ajax-ominaisuuksien parantamiseksi. Kehityskohteena olevassa sovelluskehikossa on toiminnallisuudet asynkronisten kutsujen käsittelyyn. Suunnitellut ominaisuudet ovat toteutettavissa ilman muutoksia kutsujen käsittelyyn. Kehityssuunnitelman tarkoitus on tuotteistaa uudet toiminnallisuudet osaksi sovelluskehikkoa, jotta ne ovat helposti kehitysprojektien käytettävissä tulevaisuudessa.

Kehityssuunnitelman teko aloitettiin kevyellä määrittelyllä. Määrittelyssä kartoitettiin millaisia omaisuuksia muissa vastaavissa sovelluskehikoissa on, millaisia projektikohtaisia toteutuksia löytyy sekä kyselemällä toteutusprojekteihin osallistuneilta parannusehdotuksia tai havaittuja puutteita. Kootut kehitysideat analysoitiin tarkemmin tarpeellisuuden, sekä niiden mahdollisesti tuottaman lisäarvon perusteella. Lopuksi ne priorisointiin ja siinä järjestyksessä otettiin mukaan kehityssuunnitelmaan.

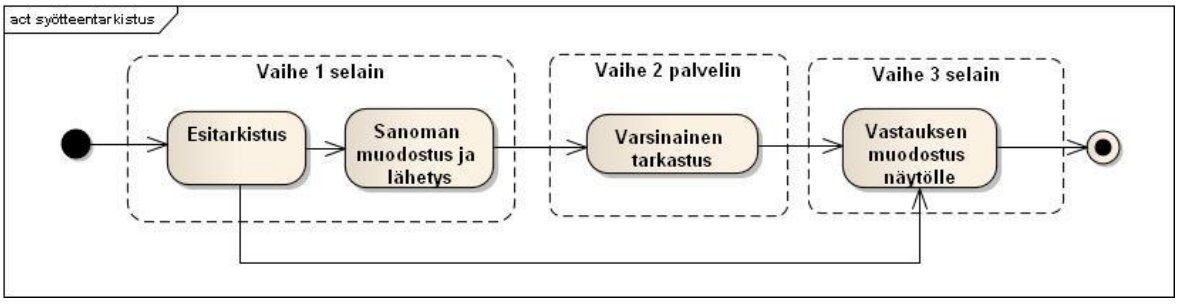
Suunnitelma on jaettu käyttöliittymä- ja sovellusosiin. Käyttöliittymällä tarkoitetaan tässä raportissa selaimessa toimivia JavaScript-osuuksia ja sovelluksella sovelluspalvelimella toimivaa sovellusta. Kehityssuunnitelmassa kehitettävien ominaisuuksien toimintaa on mallinnettu tarkoituksen mukaisin kaavion. Mallintamisessa on käytetty standardia UML:n kuvaustapaa. Suurimmat toiminnalliset kokonaisuudet on mallinnettu viestiyhteyskaaviossa ja pienimmät aktiviteettikaavioissa. Kaavioissa on kuvattu toimintaan vain keskeisesti liittyvät komponentit yleisellä tasolla.

Toimintojen mallinnuksen pohjalta kehittämissuunnitelmaan on toteutettu suunnitelmat uusista JavaScript-objekteista sekä vanhojen toimintaa on laajennettu uusilla parametreilla. Kehittäjien tuottavuuden parantamiseksi JavaScript-objektien metodi-kutsut piilotetaan sovelluskehikkokohtaisten JSP-tagien taakse. Niiden suunnittelussa on hyödynnetty olemassa olevia toteutuksia, laajentamalla niiden toiminnallisuutta sekä suunniteltu uusia silloin kun vanhojen käyttäminen ei ole ollut tarkoituksenmukaista.

3.1 Syötteen tarkistus

Reaaliaikaisella syötteen tarkastuksessa käyttäjän kirjoittaman syötteen tarkastaminen alkaa välittömästi kohdistuksen siirryttyä seuraavaan elementtiin käyttöliittymässä. Tämän tarkoituksena on tehostaa sovelluksen käyttöä, sekä vähentää käyttäjän tekemiä virheitä. Syötteen tarkistuksessa reagoidaan käyttäjän toimiin välittömästi ja sen avulla voidaan antaa välitön palaute käyttäjälle.

Syötteen tarkistus voidaan jakaa kolmeen eri vaiheeseen. Kuten kuviosta 6 voidaan huomata, syötteen tarkistus on jaettu vaiheisiin ajan ja suoritusympäristön perusteella käyttöliittymässä suorittavaan esitarkastukseen ja sanoman lähetykseen, sovelluksessa suoritettavaan varsinaiseen tarkastukseen ja vastauksen muotoilemiseen käyttöliittymässä.

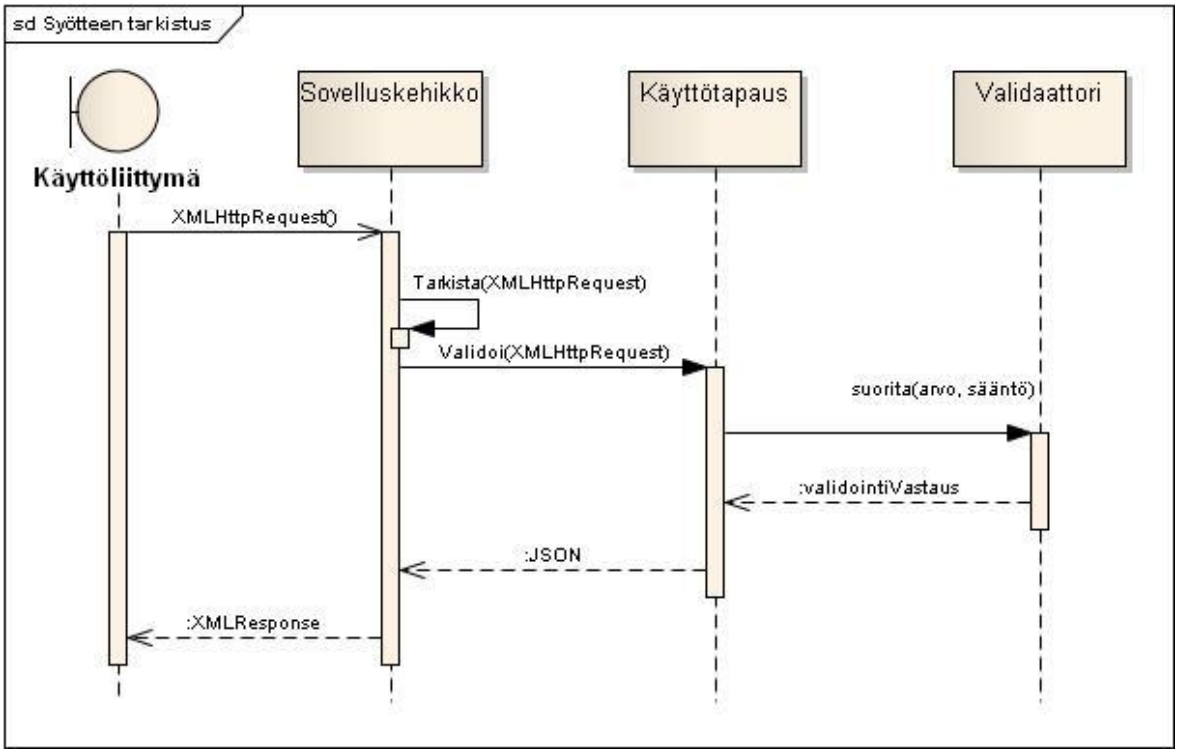


Kuvio 6 syötteen tarkistuksen vaiheistus

Ensimmäisessä vaiheessa käyttöliittymässä suoritetaan esitarkistus käyttäjän syötteelle ja lähetetään esitarkastuksen läpäissyt käyttäjän antama syöte sovellukselle varsinaiseen tarkastukseen. Käyttöliittymä tarvitsee parametreja esitarkistuksen suorittamiseksi, sekä sanoman lähettämiseksi. Käyttöliittymä lähettää sanomalla sovelluksen vaatimat parametrit, joiden perusteella sanomalle voidaan suorittaa tarvittavat tietoturvaan liittyvät tarkastukset, sekä ohjata sanoma halutun käyttötapauksen toteuttavan luokan metodille. Edellä mainittujen parametrien lisäksi pakollisena parametrina tapahtuman käynnistävän elementin tunnus. Valinnaisina parametreina käyttöliittymälle voidaan välittää elementin tietotyyppi, elementtiin liittyvät muut elementit, esitarkistuksen valinnaisuus. Elementin tietotyypillä tarkoitetaan esimerkiksi päivämäärä- tai rahatyyppiä. Tietotyyppiä hyödynnetään esitarkastuksessa, kun tarkistetaan onko käyttäjän syöte muodollisesti oikein. Kaikille syötteille ei ole tarvetta tehdä esitarkastusta, minkä johdosta esitarkastus

ei ole pakollista. Elementin tarkastukseen saattaa liittyä myös muita elementtejä, kuten esimerkiksi päivämääräväleissä. Käyttöliittymän toiminnassa huomioitavaa on myös lokalisointi. Käyttäjille näytettävät virheilmoitukset ja ohjeet pitää näyttää käyttäjän valitsemalla kielellä. Näytettävien ilmoitusten lisäksi päivä- ja rahamäärien muotoiluissa on maakohtaisia eroja, jotka tulee huomioida syötteen tarkastuksessa.

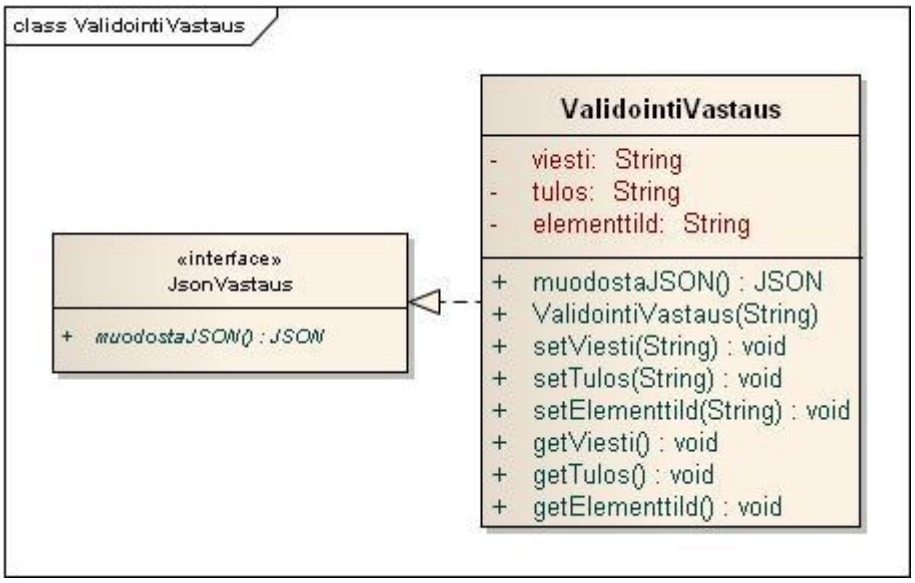
Toisessa vaiheessa käyttäjän syöte tarkistetaan sovelluksessa ja käyttöliittymälle palautetaan määrämuotoinen vastaus, jonka sisällön perusteella käyttöliittymä osaa muotoilla vastauksen käyttäjälle esitettäväksi. Kuvion 7 viestiyhteyskaaviossa on esitetty yksinkertaistettuna sanoman kulku käyttöliittymästä tarkistuksen suorittavalle luokalle sovellukseen ja sen palauttama vastaus.



Kuvio 7 Syötteen tarkistuksen viestiyhteyskaavio

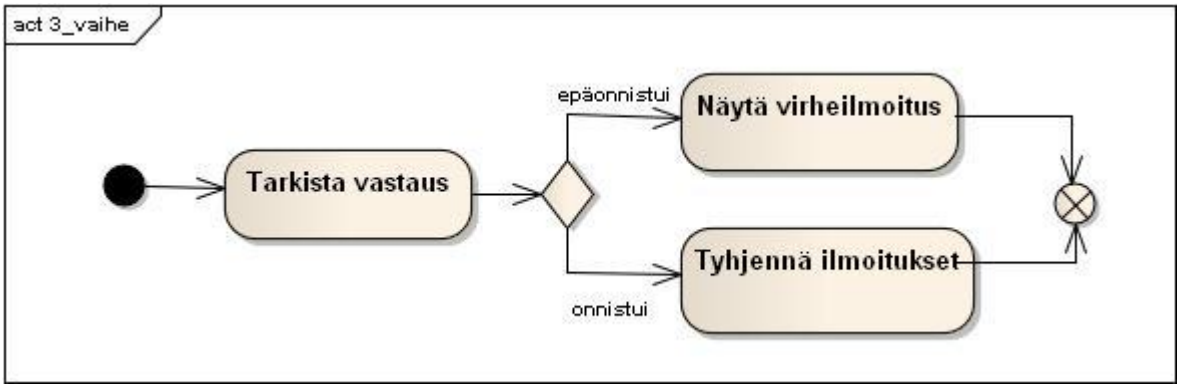
Sovelluksen kannalta on huomioitavaa, että sen on toimittava ilman että käyttäjä on sallinut JavaScriptin suorittamisen sekä, että syöte tarkistetaan uudelleen lomakkeen lähetyksen yhteydessä. Kuviossa 8 on esimerkkinä vastauksella käytettävän luokan luokkamalli, jota voidaan käyttää sekä asynkronisten että synkronisten kutsujen kanssa. Luokalla on kolme attribuuttia, sekä niille metodit arvojen asettamista ja palauttamista varten. Luokka toteuttaa myös rajapinnan mukaisen metodin, minkä avulla on käyttö-

liittymälle mahdollista palauttaa käyttöliittymän vaatima määrämuotoinen vastaus. Käyttöliittymässä näytettävien viestien lokalisointi on tehtävä sovelluksessa, toteutukseen käytettävän työmäärän ja ylläpidon minimoimiseksi.



Kuvio 8 Luokkamalli syötteen tarkistuksen vastaukselle.

Kolmannessa vaiheessa sovelluksen palauttama vastaus muotoillaan näytölle. Kuviossa 9 on esitetty kolmannen vaiheen kulku aktiviteettikaaviossa. Vastauksesta tarkistetaan pituus tyhjän vastauksen varalta, minkä jälkeen tutkitaan vastauksen tarkempi sisältö. Vastauksen perusteella virheilmoitus kohdistetaan halutun elementin viereen tai tyhjennetään mahdolliset aiemmat ilmoitukset. Ilmoituksen tarkempi sijainti elementin suhteen on määriteltävissä tyylitiedoston avulla toteutuskohtaisesti.

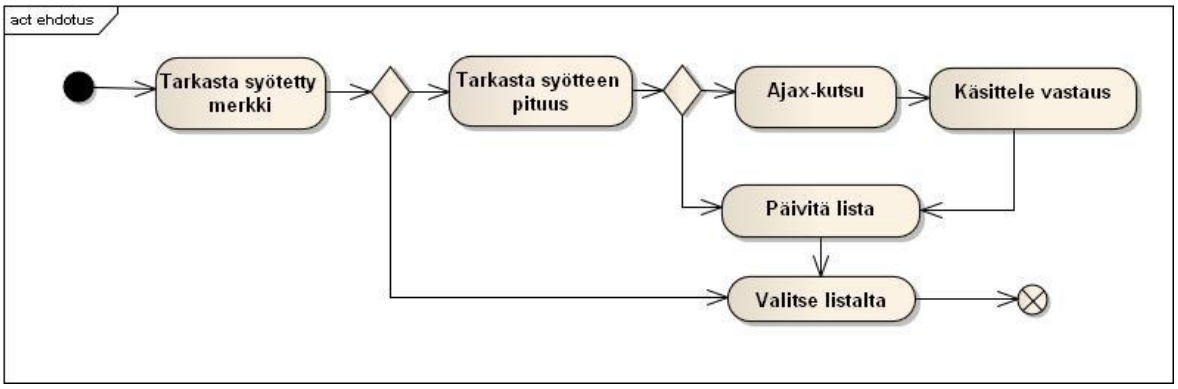


Kuvio 9 Syötteen tarkistuksen 3. vaiheen aktiviteettikaavio.

3.2 Reaaliaikainen ehdotus

Reaaliaikaisella ehdotuksella tarkoitetaan sitä, kun käyttäjän kirjoittaessa tekstikenttään hänelle tuodaan kentän viereen lista sanoista, mitkä vastaavat parhaiten hänen aloittamaansa sanaa. Reaaliaikaisilla ehdotuksilla voidaan korvata pudotusvalikot, varsinkin silloin kun kyseessä on pitkät listat. Lennonvarausjärjestelmissä lähtöpaikka ja määränpää kentät on usein toteutettu näin. Ehdotusten tarkoitus on ryhmitellä tietoa paremmin käytettäväksi ja samalla vähentää käyttäjän tekemiä kirjoitusvirheitä. Reaaliaikaiseen ehdotukseen voidaan liittää myös toimintoja kuten automaattinen täydennys ja syötteen ennustaminen.

Reaaliaikaisen ehdotuksen toiminta tapahtuu pääosin käyttöliittymän syöttökentässä ja sen viereen avutuvassa listassa ehdokkaista. Turhan verkkoliikenteen vähentämiseksi, jokaisen painalluksen jälkeen ei ole tarkoituksen mukaista suorittaa asynkronista kutsua. Kuvion 10 aktiviteettikaaviossa on kuvattu käyttöliittymän toiminta.

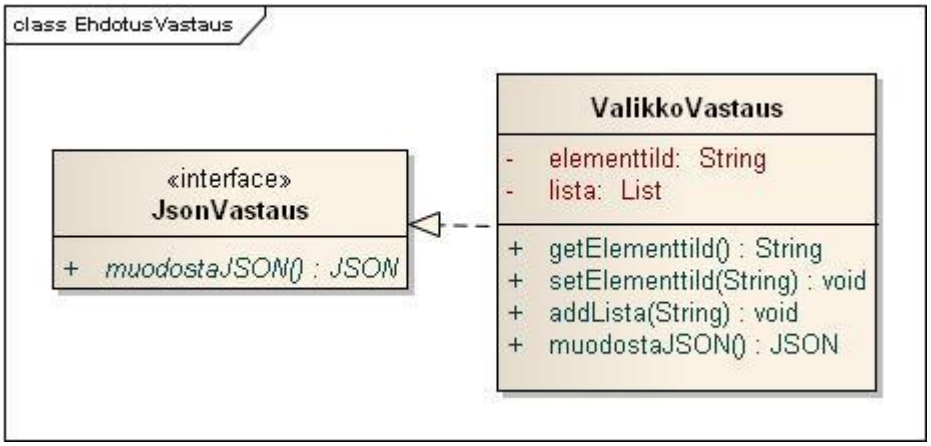


Kuvio 10 Ehdotuksen aktiviteettikaavio

Käyttöliittymässä tarkastetaan ensimmäiseksi hyväksytyt merkit syöttökentästä, joita ovat aakkosnumeeriset, nuoli-, sarkain- ja rivinvaihtomerkit. Syötetyn aakkosnumeerisen merkin perusteella jatketaan tarkistuksia. Nuolimerkeillä ylös- ja alaspäin valitaan syöttökentän viereen avautuneelta listalta seuraava. Sarkaimella tai rivinvaihdolla valitaan haluttu rivi listalta ja päivitetään syöttökenttä sen sisällöllä. Pituustarkistuksen tarkoituksena on päättää milloin sovellukselta haetaan lista ehdokkaista. Näytettävää listaa pitää käyttöliittymän toimesta muokata myös säännöllisten lausekkeiden avulla, silloin

kun listan sisältöä ei haeta jokaisen merkin painalluksen jälkeen. Näytettävältä listalta on mahdollista valita rivi myös hiiren avulla.

Käyttöliittymän lähettämä sanoma sisältää yleisten parametrien lisäksi tapahtumaan liittyvän elementin tunnuksen sekä sen sisältämän arvon. Sovelluksessa käyttötapauksen toteuttava luokka muodostaa näiden tietojen perusteella vastauksen palautettavaksi. Käyttötapauksen toteutus vastaa käytetyistä hakualgoritmeista ja hakutulosten lajittelusta. Sovelluksen palauttaman vastauksen pitää olla määrämuotoinen, jotta käyttöliittymä osaisi käsitellä sen. Kuten syötteen tarkastuksessa vastauksena palautetaan määrämuotoinen vastaus, joka sisältää listan. Kuviossa 11 on mallinnettu ehdotuksen vastauksen luokkamalli. Luokalla on kaksi attribuuttia, sekä niille metodit arvojen asettamista ja palauttamista varten. Luokka toteuttaa myös rajapinnan mukaisen metodin, jonka avulla on käyttöliittymälle mahdollista palauttaa käyttöliittymän vaatima määrämuotoinen vastaus.



Kuvio 11 Luokkamalli ehdotuksen vastaukselle

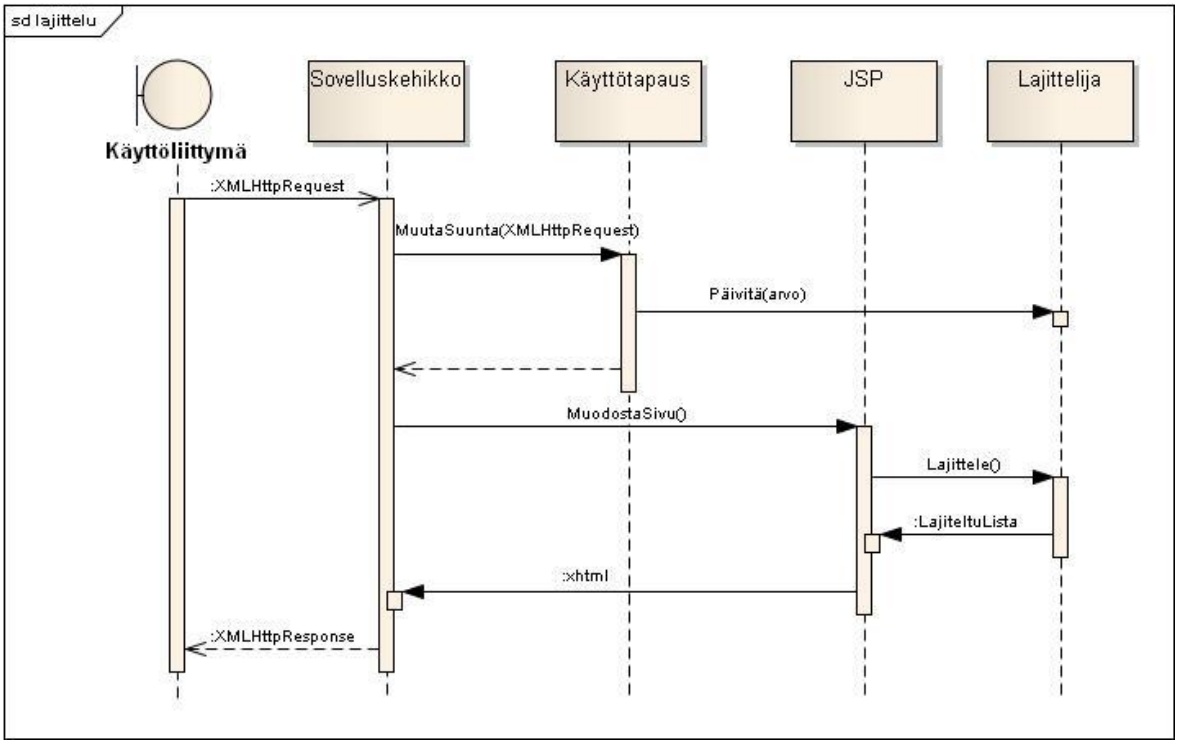
Vastauksena saadulla tiedoilla päivitetään käyttöliittymässä näkyvä lista sekä taulukko, jota käytetään apuna kun seuraavan kerran näytöllä olevaa listaa päivitetään käyttäjän toimien johdosta. Näytettävän listan muotoilu ja sen sijainti suhteessa syöttökenttään on määriteltävissä toteutuskohtaisesti tyylitiedoston avulla.

3.3 Lajittelu

Lajittelulla tässä yhteydessä tarkoitetaan käyttöliittymässä näytetyn taulukon tietojen uudelleen lajittelua. Lajittelussa käytetään hyväksi käyttäjän istuntoon talletettua tietoa taustahakujen sijaan. Asynkronisten kutsujen avulla voidaan korvata aiempi toteutus, mikä pohjautui hyperlinkkien avulla tehtyyn sivun uudelleen lataukseen. Suunnitelma perustuu periaatteeseen käyttäjän pitämiseksi avoinna olevalla sivulla.

Käyttöliittymässä lajittelu tarvitsee kaksi ominaisuutta, linkit mitkä käynnistävät taulukon lajittelun ja alueen mihin taulukko päivitetään. Taulukon lajittelu tapahtuu yhden sarakkeen perusteella nousevasti tai laskevasti. Sarakkeen otsikossa on lajittelua kuvaavat kuvakkeet tai tekstit. Ne ovat linkkejä asynkronisen kutsun käynnistävän objektin metodiin. Parametreina linkeissä on sarakkeen indeksi, haluttu lajittelusuunta sekä päivitettävän alueen tunnus.

Sovelluksen kannalta taulukko on käyttötapaukseen liittyvä käyttöliittymäkomponentti, johon voi liittyä lajittelukomponentti. Käyttötapauksen toteuttava luokka päivittää lajittelukomponentin tilan pyyntösanoman parametrien perusteella. Varsinainen lajittelu tehdään vasta sivun muodostuksen yhteydessä. Kuviossa 12 on esitetty viestiyhteyskaaviossa lajitteluun liittyvä tapahtumien yksinkertaistettu kulku ja siihen liittyvät komponentit. Sovellus välittää käyttöliittymältä tulleen sanoman käyttötapauksen toteuttavalle luokalle, joka päivittää lajittelukomponentin tilaan lajittelun perustana käytettävän sarakkeen indeksin ja lajittelun suunnan. Taulukkoa muodostettaessa lajittelukomponentti lajittelee taulukon tiedot tilansa perusteella. Sovellus palauttaa käyttöliittymälle vastauksen, jonka käyttöliittymä osaa kohdistaa oikeaan kohtaan päivitettävän alueen tunnuksen perusteella.



Kuvio 12 Lajittelun viestiyhteyskaavio

3.4 Pudotusvalikoiden sisällön päivitys

Pudotusvalikoiden sisällön päivityksessä muutos yhdessä pudotusvalikossa aiheuttaa toisen pudotusvalikon sisältämän listan päivityksen, esimerkiksi merkin valinta vaikuttaa malleja sisältävän pudotusvalikon sisältöön. Käyttäjän toimiin reagoidaan välittömästi, eikä toiminta katkea sivun uudelleen lataukseen. Pudotusvalikoiden sisällön päivitys on osa reaaliaikaisen lomakkeen kokonaisuutta, johon kuuluu myös aiemmin mainitut syötteen tarkistus ja ehdotus.

Pudotusvalikon sisällön päivitys käynnistyy käyttäjän muuttaessa valintaa pudotusvalikosta, jonka valitulla arvolla on vaikutus toiseen pudotusvalikkoon. Käyttöliittymälle välitetään tapahtuman käynnistävän pudotusvalikon sekä siihen liittyvän pudotusvalikon tunnus. Tässä voidaan käyttää samoja parametreja kuin syötteen tarkistuksessa, poikkeuksena liitetyn elementin tunnuksen pakollisuus. Käyttöliittymässä ei tehdä tarkistuksia valitulle arvolle vaan ainoastaan tyhjennetään siihen liitetyn pudotusvalikon sisältämä lista. Tämän jälkeen lähetetään, aiemmissa suunnitelmissa määriteltyjen pakollisten tietojen lisäksi, sanomalla tapahtuman käynnistäneen pudotusvalikon tunnus ja käyttäjän valinta sekä liitetyn pudotusvalikon tunnus.

Sovelluksessa käyttötapauksen toteuttava luokka vastaa palautettavan listan muodostuksesta. Sovelluksessa käyttäjän istuntoon on talletettu tieto käyttöliittymässä olevien elementtien sisällöstä. Muodostettu lista on talletettava myös käyttäjän istuntoon, ettei sivun uudelleen latauksen yhteydessä kadoteta aiemmin haettuja tietoja. Kuten syötteen tarkastuksessa ja reaaliaikaisissa ehdotuksissa sovelluksen palauttaman vastauksen pitää olla määrämuotoinen, jotta käyttöliittymä osaa käsitellä vastauksen. Tässä voidaan käyttää samaa luokkaa kuin reaaliaikaisen ehdotuksen yhteydessä, koska tarvittavan luokan attribuutit ja metodit ovat samat (Kuvio 11).

Käyttöliittymässä sovelluksen palauttama lista puretaan pudotusvalikkoon näytettäväksi. Pudotusvalikon tiedoille muodostetaan indeksi, jota käytettään lomakkeen lähetyksessä. Käyttöliittymässä sovelluksen palauttamaa tietoa ei enää muotoilla, vaan se näytetään siinä muodossa kuin sovellus on sen lähettänyt.

4 Pohdinta ja johtopäätökset

4.1 Projektin tulokset

Opinnäytetyönä tehdyn projektin tarkoitus oli laatia kehittämissuunnitelma kohdeyrityksessä kehitetyn sovelluskehikon Ajax-ominaisuuksien parantamiseksi. Projektin tuloksena syntyivät suunnitelmat kolmeen uuteen toiminnallisuuteen sekä yhden sovelluskehikossa olevan toiminnallisuuden laajentaminen dynaamisempaan malliin.

Sovelluskehikkoon suunnitellessa uusia elementtejä, hyödynnettäviksi varsinaisten verkkopalveluiden toteutuksissa, voidaan keskityttyä käytettävyyden tehokkuus ja virheettömyys ominaisuuksiin parantamiseen. Tosin varsinaisten toteutusprojektien vastuulle jää kuinka hyvin ne hyödyntävät sovelluskehikon tarjoamia ominaisuuksia käytettävyyden parantamiseksi. Hyödyllisyys, opittavuus ja tyytyväisyys ovat kuitenkin arvioitavissa vasta lopullisesta verkkopalvelusta. Opinnäytetyön tuloksena syntyivät suunnitelmat elementeistä, joiden avulla on mahdollista parantaa käytettävyyttä. Esteettömyyttä ei tässä projektissa käsitelty, mutta se voisi olla yksi jatkokehityksen aiheita.

Ajax on teknologiana jo niin vanha, että sen käytön parhaimmista käytännöistä on muodostunut lukuisia suunnittelumalleja hyödynnettäviksi suunnitteluun. Kehityssuunnitelmassa on pystytty konkretisoimaan teoreettisessa pohdinnassa esiteltyjä suunnittelumalleja varsinaisiksi suunnitelmiksi. Suunnittelumallien käyttö ei pelkästään helpota suunnittelua, vaan myös niitä yhdistelemällä on mahdollista toteuttaa vieläkin käyttäjäystävällisimpiä elementtejä, kuten esimerkiksi prosessin etenemistä kuvaavan elementin liittäminen osaksi syötteen tarkistusta. Tätä ei tämän opinnäytetyöprojektin puitteissa ehditty suunnitella, vaan ne ovat jatkokehityskohteita.

4.2 Oman oppimisen arviointi

Idea kehittämisprojektista syntyi jo muutama vuosi ennen varsinaisen opinnäytetyön projektin käynnistymistä. Ideaa piti kehittää ja tarkentaa, jotta siitä muodostui tarkemmin rajattu kokonaisuus opinnäytetyön aiheeksi. Opinnäytetyön aihetta kehittäessä keskityin vain toteutettavaan projektiin ja teoreettisen lähestymistavan valinta jäi huomioimatta tässä vaiheessa, millä oli hieman vaikutuksia raportointivaiheen aikatauluihin

ja työmääriin. Koko opinnäytetyöprosessin läpivienti projektina helpotti tehtävien, työmäärien ja aikataulujen suunnittelua. Kattavasti tehty projektisuunnitelma edesauttoi etenemisen seuraamista ja tarvittaessa oli mahdollista kiristää työtahtia, mikäli projekti oli vaarassa jäädä aikataulustaan. Projektin hallinnassa apua oli kohdeyrityksen käyttämästä projektin hallintamenetelmästä sekä prosesseista. Tosin projektin pienestä koosta johtuen, menetelmiä ja prosesseja piti soveltaa hieman. Opinnäytetyön raportoinnissa haasteellista oli teoreettisen näkökulman valinta. Vaihtoehtoja teoreettiselle pohdinnalle olivat ketterät suunnittelumenetelmät tai suunnittelumallit. Näkökulman tarkentumiseen meni liian paljon aikaa ja kirjoittaminen oli tästä johtuen aikataulustaan jäljessä koko projektin ajan. Tässä kostautui selkeästi se, että valitessani aihetta en ollut ottanut kantaa teoreettiseen taustaan. Kokonaisuuden kannalta katsoen pystyin kuitenkin yhdistämään loogisesti teoreettisen taustan varsinaiseen projektiin. Omaa oppimista opinnäyteprosessin aikana kun arvioin, niin suurimmat hyödyt tulevat projektin aikana opituista asioista, sekä aiemmin opitun soveltamisesta käytäntöön. Varsinaisten opintojen aikana eri tietolähteiden käyttö on tullut erittäin hyvin tutuksi, joten en kokenut että tiedon hankkiminen tai analysointi olisi parantunut opinnäytetyön aikana. Tosin ilman hyviä tiedonhaku valmiuksia opinnäytetyön tekeminen olisi ollut erittäin paljon haastavampaa.

Opinnäytetyön aikana opin monia uusia asioita, kartutuin tietämystäni aiemmin opitusta sekä sovelsin opittua käytäntöön. Ajax teknologiana oli minulle tuttu niin opintojen kuin työelämänkin projektien kautta. Kerätessäni vaatimuksia suunnitelman pohjaksi, perehdyin useaan vapaan lähdekoodin Ajax-kirjastoon ja niillä toteutettuihin käyttöliittymiin. Sen lisäksi, että sain opinnäytetyötäni varten monia ideoita noista kirjastoista, on karttunut tietämys varmasti hyödynnettävissä tulevaisuudessa.

Käyttöliittymiä olen toteuttanut useissa projekteissa, mutta varsinaisesti niiden suunnittelu on ollut projekteissa jonkun muun vastuulla. Tietämykseni selainkäyttöliittymien suunnittelusta ja suunnitteluun vaikuttavista tekijöistä kasvoi huomattavasti tämän projektin aikana. Suunnittelumallit ovat ohjelmoinnin puolelta tuttuja, mutta niiden käyttöön käyttöliittymien suunnittelussa en aiemmin ollutkaan kiinnittänyt huomiota. Hyvänä muistutuksena oli huomata uudelleen käytön toimivuus ja helppous myös käyttöliittymäsuunnittelussa.

Projektin vetovastuu on ollut jonkun muun vastuulla niissä projekteissa, joissa olen ollut mukana. Projektipäällikkökurssin olen käynyt aiemmin, mutta nyt ensimmäistä kertaa kurssin jälkeen pääsin soveltamaan kurssilla opittua. Tämän lisäksi projektin aikana oli mahdollista tutustua kohdeyrityksen projektien hallintaprosessiin ja soveltaa sitä tässä opinnäytetyöprojektissa. Kokonaisuudessaan olen varsin tyytyväinen opinnäytetyön aikana oppimiini asioihin ja olen varma siitä, että opitut asiat ovat hyödynnettävissä tulevaisuudessa eri projekteissa.

Lähteet

Bibeault, B. Katz, Y. 2010. JQuery in Action. Manning Publications Co. Stamford

Crane, D. Pascarello, E. James, D. 2006. Ajax in Action. Manning Publications Co. Greenwich.

Hoffman, B. Sullivan, B. 2008. Ajax security. Pearson Publications Inc. Boston

Mahemoff, M. 2006. Ajax Design Patterns. O'Reilly Media Inc. Sebastopol.

Object Management Group 2012. UML. Luettavissta: http://www.uml.org/ Luettu 19.11.2012

Riihiaho, S. 1998. Käytettävyyden arviointi ilman käyttäjiä. Systeemityö 5(4), 4-6. Systeemityöyhdistys SYTYKE ry.

Riihiaho, S. 2000. Käytettävyystestauksen muunnelmia. Teoksessa: Eero Pantzar (Toim.) Informaatio, tieto ja yhteiskunta. Suomen Akatemian Tiedon tutkimusohjelman raportteja, 4/2000, Tampereen yliopiston Tietoyhteiskunnan tutkimuskeskus, Tampere. s. 223–230

Rubin, J. Chisnell, D. 2008. Handbook of Usability Testing. Second edition. Wiley Publishing Inc. Indianapolis

Scott, B. Neil, T. 2009. Designing Web Interfaces. O'Reilly Media Inc. Sebastopol.

Sinkkonen, I. Kuoppala, H. Parkkinen, J. Vastamäki, R. 2006. Käytettävyyden psykologia. 3. uudistettu painos. Edita Publishing Oy, Helsinki

W3C 2012a. World Wide Web Consortium. JavaScript Web APIs. Luettavissa: http://www.w3.org/standards/webdesign/script.html. Luettu 19.11.2012

W3C 2012b. World Wide Web Consortium. HTML & CSS. Luettavissa: http://www.w3.org/standards/webdesign/htmlcss.html. Luettu 19.11.2012

W3C 2012c. World Wide Web Consortium. XML Essentials. Luettavissa: http://www.w3.org/standards/xml/core. Luettu 19.11.2012

Liitteet

Liite 1. Keskeiset käsitteet

Ajax, Asynchronous JavaScript and XML

Yhdistelmä JavaScript, DOM, CSS ja XML tekniikoista, minkä avulla voidaan verkkosivujen sisältöä ladata tai lähettää dynaamisesti(W3C 2012a).

CSS, Cascading Style Sheets

Verkkosivujen ulkoasujen kuvaamiseen tarkoitettu kuvauskieli. Sen avulla voidaan muotoilut erottaa rakenteesta(W3C 2012b).

DOM, Document Object Model

Ohjelmointirajapinta mikä mahdollistaa rakenteisten dokumenttien muokkaamisen(W3C 2012a)

HTML, Hypertext Markup Language

Kuvauskieli jolla kuvataan verkkosivun rakenne (W3C 2012b).

JEE, Java Enterprise Edition

Java teknologia verkkopalveluiden tuottamiseksi.

JSP, Java Server Pages

Java teknologia html sivujen tuottamiseen

JavaScript

Komentosarjakieli html dokumenttien rakenteen ja ulkoasun muokkaamiseen(W3C 2012a)

JSON, JavaScript object notation

Tekstipohjainen tiedonsiirtomuoto, jota käytetään JavaScriptillä

UML, Unified Modeling Language

Standartoitu mallinnuskieli sovellusten rakenteen, toiminnan ja arkkitehtuurin sekä liiketoimintaprosessien ja tietorakenteiden kuvaamiseen().

XHTML, eXtensible Hypertext Markup Language

Muunnelma HTML-kielestä, mikä käyttää XML:n syntaksia(W3C 2012b).

XML, Extensible Markup Language

XML on yksinkertainen tekstipohjainen muoto rakenteisen tiedon esittämiseen(W3C 2012c).

Liite 2: Salainen

Ajax ja käyttöliittymäelementtien suunnittelu

Loppuraportti

Ajax ja käyttöliittymäelementtien suunnittelu

Ajax ja käyttöliittymäelementtien suunnittelu kehitysprojektin tarkoituksena on laatia suunnitelma yritys X:ssä toteutetun sovelluskehikon dynaamisten ominaisuuksien kehittämiseksi.

Projekti toimii myös HAAGA-HELIA Ammattikorkeakoululle tehtävänä opinnäytetyönä.

1 Yhteenveto

1.1 Projektin tiedot

Sovelluskehikossa on ollut pitkään perus Ajax-toiminnallisuus. Asiakkailta sekä omalta kehitysryhmältä tulleiden palautteiden perusteella Ajax-omineisuuksia tulisi kehittää edelleen. Asiakasprojektien ja muiden kehityshankkeiden prioriteetti on ollut korkeampi kuin Ajax-ominaisuuksien kehityksen, tästä johtuen kehitys on jäänyt odottamaan suotuisampaa ajankohtaa.

Asiakasprojektit ovat pystyneet kehittämään Ajax-toiminnallisuuksia selainpohjaisiin käyttöliittymiin sovelluskehikossa olevien ominaisuuksien avulla. Projektien tehostamisen kannalta olisi kuitenkin tarpeellista, että kattavat Ajax-toiminnallisuudet löytyisivät valmiina sovelluskehikosta.

Ammattikorkeakouluopintoihin kuuluu opintojen loppuvaiheessa laaja käytännön harjoitustyö. Harjoitustyöt yleensä liittyvät käytännön ongelman ratkaisemiseen. Kehittämisprojekti täyttää opinnäytetyölle asetetut vaatimukset ja samalla opinnäytetyöprojekti mahdollistaa kehitysprojektin aloittamisen.

Projektin arvioitu kesto on noin 4 kuukautta ja työmäärä noin 400 tuntia. Projekti alkoi 3.9.2012 ja päättyy 15.1.2013

1.2 Yhteenveto opituista asioista

Projektissa opituista asioista päällimmäiseksi nousee käytettävyyden, suunnittelumallien ja toteutettujen suunnitelmien välisen yhteyden konkretisoituminen. Kehityssuunnitelma keskittyy yksittäisten elementtien suunnitteluun, mutta perimmäinen tarkoitus on kuitenkin verkkopalveluiden käytettävyyden parantaminen. Suunnitelmissa on käytetty hyväksi Ajax-sovellusten kehityksessä hyviksi todettuja suunnittelumalleja, jotka pohjautuvat verkkosivujen suunnittelumalleihin ja joiden tarkoitus on käytettävyyden parantaminen.

Toinen asia, mikä konkretisoitui projektin aikana, oli työmäärien ja aikataulujen arvioinnin tärkeys. Suunnittelua ja määrittelyä voi jatkaa määrättömästi, mutta projektin alussa asetetut tavoiteaikataulut kalenteriviikoittain, sekä niiden tarkka seuraaminen piti projektin aikataulussaan ja projekti pysyi asetettujen rajoituksien sisällä.

1.3 Suositukset jatkotoimenpiteiksi

Ensimmäisenä ja tärkeimpänä jatkotoimenpiteenä on suunnitelmien mukaisten toiminnallisuuksien toteuttaminen sovelluskehikkoon. Suunniteltujen kohteiden välillä ei ole riippuvuuksia, joten ne voidaan toteuttaa porrastetusti.

Vaatimusten kartoituksessa löytyi kehitysideoita, jotka vaativat vielä lisäselvittelyjä. Toisena jatkotoimenpiteenä olisi lisäselvitysten tekeminen näille ideoille sekä niiden tarkempi analysointi.

Kehityssuunnitelmat sisältävät vain komponenttien perustoiminnallisuuden. Kolmantena jatkotoimenpiteenä olisi lisäominaisuuksien kartoittaminen ja toteuttamien.

2 Projektin toteutuminen

2.1 Projektin tavoitteet ja niiden hyväksyminen

Projektilla oli kolme tavoitetta.

1. Kartoittaa, analysoida ja priorisoida toiminnalliset ja ei toiminnalliset vaatimukset uusille Ajax-ominaisuuksille. Kartoituksessa löytyi 11 toiminnallista vaatimusta, jotka analysoitiin tarkemmin ja priorisoitiin hyödyllisyyden perusteella. Katselmointikokouksen päätöksellä kehittämiskohteiksi valikoitui 4 kohdetta.

2. Teknisen kehittämisehdotuksen laatiminen sovelluskehikon ominaisuuksien kehittämiseksi. Ehdotuksessa on tarkoitus ottaa kantaa mitkä vaatimusmäärityksen mukaisista ominaisuuksista voidaan toteuttaa sovelluskehikkoon mitkä ominaisuudet ovat tarkoituksen mukaisempaa toteuttaa asiakasprojekteissa. Kehittämissuunnitelma laadittiin katselmointikokouksen tulosten perusteella. Kaikista katselmointikokouksessa hyväksytyistä kehityskohteista toteutettiin suunnitelmat. Asiakaskohtaisia suunnitelmia ei toteutettu.

3. Opinnäytetyön laatiminen. Opinnäytetyöhön liittyvät projekti on toteutettu ja se on raportoitu opinnäytetyönä.

Lopetuskokous hyväksyi projektin tulokset.

2.2 Edellytykset ja ulkoiset riippuvuudet

Projekti oli riippuvainen yritys X:n asiantuntijoiden avusta. Asiantuntijat olivat projektin käytettävissä tarvittaessa. Projektin ja asiantuntijoiden aikataulut olivat sovitettavissa siten, ettei projektin eteneminen estynyt. Asi-

antuntija-apua tarvittiin vaatimusten katselmoinnissa ja priorisoinnissa sekä kehitysympäristön asentamisessa.

2.3 Rajoitukset ja prioriteetit

Opinnäytetyölle asetetusta työmäärästä johtuen, projektissa oli mahdollista tuottaa vain vaatimusmääritys, kehitysehdotus ja opinnäytetyön kirjallinen osuus. Projektin tärkein prioriteetti oli tulos. Projektin tuloksena toteutti kehittämissuunnitelma ja opinnäytetyö aikataulua tai kustannuksia ylittämättä.

2.4 Projektin kustannukset

Projekti tehtiin opiskelutyönä, mutta yritys X mahdollisti projektipäällikön työskentelyn projektin parissa päivän viikossa 10 viikon ajan. Projektissa käytettiin yritys X:n omistamia laitteistoja ja ohjelmistoja, eikä näiden osalta suoritettu uusia hankintoja.

2.5 Projektin aikataulu

Projekti jakaantui kolmeen päävaiheeseen. Vaiheet yksi ja kaksi käsittävät kehittämisprojektin ja kolmas opinnäytetyön raportoinnin. Vaihe kaksi on jaettu kahteen osaan ja sen aloitus riippui ensimmäisen vaiheen valmistumisesta. Vaihe kolme, opinnäytetyön raportointi, toteutettiin vaiheiden yksi ja kaksi rinnalla.

Ensimmäisen vaiheen tavoitteena oli tuottaa projektisuunnitelma, sekä vaatimusmääritys, minkä pohjalta kehityssuunnitelman toteutetaan. Taulukossa 1 on ensimmäisen vaiheen suunnitellut ja toteutuneet työmäärät, sekä toteuma prosentteina. Toteuma on 21 prosenttia pienempi kuin suunniteltu työmäärä. Alitus johtuu osin työmäärien arviointimenetelmästä, jossa pienin työmäärä oli yksi päivä eli 8 tuntia. Pienissä tehtävissä poikkeamat korostuvat.

Vaihe 1	Projektin alusta		
	suunniteltu	toteutunut	
Tehtävä	tuntia	tuntia	%
Projektisuunnitelma	24	20	83 %
Projektihallinto	8	8	100 %
Vaatimusten kerääminen	40	32	80 %
Vaatimusten analysointi	8	8	100 %
Vaatimusten priorisointi	8	6	75 %
Vaatimusten katselmointi	8	2	25 %
Yhteensä	96	76	79 %

Taulukko 1 Ensimmäisen vaiheen työmäärät

Toisen vaiheen tavoitteena oli kehittämisehdotuksen laatiminen ensimmäisessä vaiheessa kerättyjen ja analysoitujen vaatimusten pohjalta. Taulukossa 2 on vaiheen suunnitellut ja toteutuneet työmäärät, sekä toteuma

prosentteina. Toteuma on myös alle suunnitellun työmääräarvion. Poikkeama johtuu siitä, että projektin kokonaisaikataulun vuoksi teknisille suunnitelmille varattu kahdeksan viikon jakso piti lyhentää kuuteen viikkoon.

Vaihe 2	Projektin alusta		
	suunniteltu	toteutunut	
Tehtävä	tuntia	tuntia	%
Tekniset suunnitelmat			
Ensimmäinen osa	64	46	72 %
Toinen osa	64	48	75 %
Kehittämisehdotus	16	15	94 %
Loppuraportti	8	7	88 %
Projektihallinto	8	8	100 %
Yhteensä	160	124	78 %

Taulukko 2 Toisen vaiheen työmäärät

Kolmannen vaiheen tavoitteena oli tuottaa opinnäytetyön kirjallinen osuus. Taulukossa 3 on kolmannen vaiheen suunnitellut ja toteutuneet työmäärät sekä toteuma prosentteina. Tämä vaiheen toteuma ylittyi 12 prosentilla. Teoreettisen pohdinnan lähtökohta ei ollut selvillä projektin alussa, minkä johdosta aineiston keruuseen ja läpikäyntiin meni suunniteltua enemmän aikaa. Tämän lisäksi kirjoittamiseen meni huomattavasti enemmän aikaa kun oli suunniteltu.

Vaihe 3	Projektin alusta		
	suunniteltu	toteutunut	
Tehtävä	tuntia	tuntia	%
Aineiston keruu	24	32	133 %
Johdanto	8	5	63 %
Tietoperusta	56	60	107 %
Empiirinen osuus	32	40	125 %
Johtopäätökset	8	7	88 %
Tiivistelmä	8	8	100 %
Yhteensä	136	152	112 %

Taulukko 3 Kolmannen vaiheen työmäärät

Kokonaisuudessaan projektin kokonaisaikataulu piti hyvin. Kokonaistyömäärä alittui 10 prosentilla (Taulukko 4), mikä on projektille asetettujen laatukriteerien sisällä.

Koko projekti	Projektin alusta		
	suunniteltu	toteutunut	
Yhteensä	392	352	90 %

Taulukko 4 Projektin työmäärät yhteensä

3 Yhteenveto projektimittareista

Projektimittari	Tavoitearvo	Toteutunut arvo	Huomautukset
Aikataulun pitäminen	11.1.2013	15.1.2013	Projekti valmistui aikataulussaan. Ylitys johtuu projektin lopetuskokous ajankohdan tarkentumisesta.
Työmääräarvion pitäminen	392	352	Poikkeama on 10 prosenttia, joka on asetetun kynnysarvon +/-10% sisällä.

4 Käytettyjen projektinhallintamenetelmien arviointi

4.1 Projektihallintaprosessi

Projektinhallinnassa noudatetaan yritys X:n projektinhallintaprosessia ja projektinhallintamenetelmää.

4.2 Projektin tietojen/asiakirjojen hallinta

Projektissa noudatettiin soveltaen yritys X:n dokumentointistandardeja. Opinnäytetyön raportoinnissa käytettiin HAAGA-HELIAN raportointimallia.

Projektipäällikkö talletti projektin aikana syntyneet projektin hallinnolliset dokumentit projektikirjaan. Projektikirja on talletettu sähköisessä muodossa ja ovat projektipäällikön vastuulla

4.3 Poikkeamien ja muutosten hallinta

Projektin aloituskokouksessa sovittiin, että projektipäällikkö arvioi kaikki projektiin tulevat muutosesitykset ja niiden vaikutukset sekä esittää arvion hankepäällikölle. Muutosesitykset ja niiden vaikutukset projektiin dokumentoidaan Projektin aikaiset tehtävämuutokset – luetteloon.

Muutospyynnöt käsitellään ohjausryhmän kokouksessa hankepäällikön esityksestä. Vaikutusten arvioinnin jälkeen ohjausryhmä päättää, mitä esitetylle muutokselle tehdään.

Projektin aikana ei tullut muutospyyntöjä.

4.4 Laadunvarmistus

Laadunvarmistuksessa noudatettiin yritys X:n laadunvarmistusmenettelyjä. Laadunvarmistuksessa käytetyt projektimittarit ja niiden toteuma on esitelty luvussa 3.

4.5 Viestintä

Projektin aikana pidettiin kolme kokousta, joihin liittyvät asiakirjat, kuten kokouskutsut, esityslistat sekä pöytäkirjat, toimitettiin osallistujille sähköpostitse viikkoa ennen varsinaista kokousta. Kokousten lisäksi projektin etenemisestä informoitiin asianomaisia noin neljän viikon välein edistymisraporteilla.

4.6 Riskien hallinta

Projektin alussa sovittiin, että projektin riskejä seurataan säännöllisesti ja riskiluetteloa päivitetään projektin edetessä. Riskiluettelo käsitellään ohjausryhmän kokouksessa. Riskiluettelon ylläpidosta vastasi projektipäällikkö. Mikään riskiluettelossa esitetyistä uhista ei toteutunut.

4.7 Projektin päättäminen

Projekti päättyi lopetuskokoukseen 15.1.2013. Lopetuskokouksessa työn tilaajalle luovutettiin kehittämissuunnitelma, opinnäytetyö ja loppuraportti.