

Käyttöliittymäsuunnittelusta toteutukseen - Ohjausnäköy ja sen luonti toimintajärjestelmään

Outi Niemelä



Tietojenkäsittelyn koulutusohjelma

<p>Tekijä tai tekijät Outi Niemelä</p>	<p>Ryhmätunnus tai aloitusvuosi 2010</p>
<p>Raportin nimi Käyttöliittymäsuunnittelusta toteutukseen - Ohjausnäkyä ja sen luonti toimintajärjestelmään</p>	<p>Sivu- ja liitesivumäärä 45 + 33</p>
<p>Opettajat tai ohjaajat Sirpa Marttila</p>	
<p>Tässä työssä käydään läpi ohjausnäkyä käsitteenä, sekä käyttöliittymän suunnittelun eri vaiheet (suunnittelu, toteutus, testaus ja validointi). Työn kohteena on tilaajan, IMS Business Solutions Oy:n kehittämä toimintajärjestelmä IMS, johon ohjausnäkyä käyttöliittymän osalta toteutetaan.</p> <p>Työssä sivutaan mobiilia käyttöliittymäsuunnittelua sekä itse IMS-toimintajärjestelmää. Työhön ei kuulu palvelinpuolen ratkaisua, eikä tietokantapuolta.</p> <p>Teorian osalta perehdytään ohjausnäkyä lisäksi käytettävyyden ja mallintamisen eri osa-alueisiin. Tutkitaan, mitä eri keinoja mallintamisessa voidaan käyttää, ja perehdytään millä työkaluilla se voidaan toteuttaa. Työssä käydään läpi validoinnin ja testauksen eri työkalut ja verrataan niitä. Työssä perehdytään eri arkkitehtuureihin ja ohjelmointikieliin, millä ohjausnäkyä käyttöliittymä voidaan toteuttaa.</p> <p>Työssä käydään askel askeleelta läpi vaiheet, mitä ohjausnäkyä käyttöliittymää suunnitellessa ja luodessa tehtiin. Lopullisena tuotoksena on valmis käyttöliittymä suunnittelun, mallinnuksen, testauksen, validoinnin ja koodin siistimisen jälkeen.</p> <p>Työ tehtiin opinnäytetyönä kevään 2013 aikana. Työn teon aikana opittiin paljon uusia menetelmistä ja tehdyn työn analysoinnista ja aikataulutuksesta. Tuloksena syntyi käytettävyydeltään ja toiminnallisuudeltaan kattava ratkaisu toimintajärjestelmän sisäiseksi ohjausnäkyäksi.</p>	
<p>Asiasanat Ohjausjärjestelmät, käyttöliittymät, käytettävyys, mallintaminen, toteuttamissuunnittelu, toteuttaminen</p>	

Degree Programme in Business Information Technology

<p>Authors Outi Niemelä</p>	<p>Group or year of entry 2010</p>
<p>The title of thesis Designing and Making a User Interface – Creating an Inner Dashboard into a Management System</p>	<p>Number of report pages and attachment pages 45 + 33</p>
<p>Advisor(s) Sirpa Marttila</p>	
<p>The purpose of this thesis was to clarify the concept of a dashboard, and to go through the stages of user interface (UI) designing (planning, execution, testing, and validating). The target for the dashboard was to attach it into IMS, which is a management system owned by the client company, IMS Business Solutions Ltd.</p> <p>In addition, the thesis also looked into some facts about mobile UI designing and IMS in general. Server side- or database-solutions, however, were not included in this thesis. The thesis was written during the spring of 2013.</p> <p>The theoretical part of thesis was structured as follows: first the theory of dashboard was studied, and then it continued by explaining the different parts of usability and UI modeling. Furthermore, different methods and tools that can be used in modeling were investigated, and they were compared to each other for validating and testing user interfaces. Finally, the different architectures and programming languages that can be used in making the user interface for the dashboard were examined.</p> <p>In the empirical part of the thesis, all the different steps regarding the designing and creating a user interface were taken.</p> <p>The final product of this thesis was the finished user interface after it had been designed, modeled, tested, refactored, and validated with the chosen methods. It was a complete solution for a management system's dashboard, considering its usability and functionalities.</p>	
<p>Key words Control systems, user interfaces, usability, modeling, execution planning, executing</p>	

Sisällys

1 Johdanto	1
2 Ohjausnäkyvä	3
3 Ohjausnäkyvän suunnittelu ja toteutus	5
3.1 Käyttöliittymäsuunnittelu	5
3.2 Mallintaminen	7
3.3 Ohjausnäkyvän toteuttaminen	10
3.4 Koodin validointi ja testaus	13
4 Kohde ja tavoitteet.....	16
5 Kohteen ohjausnäkyvän suunnittelu ja toteutus.....	20
5.1 Suunnittelu ja kartoitus	20
5.2 Mallintaminen	22
5.3 Käyttöliittymän teko	30
5.4 Testaus ja validointi.....	33
6 Pohdinta	36
6.1 Keskeiset tulokset ja hyödynnettävyys	38
6.2 Luotettavuus	43
6.3 Suositukset ja jatkokehitysehdotukset	43
6.4 Ammatillinen kehittyminen ja oppiminen.....	44
Lähteet.....	46
Liitteet.....	50
Liite 1. Käsitteet	50
Liite 2. Mallinnustyökalujen vertailu.....	54
Liite 3. Aloituskäytön sisällön kysely	55
Liite 4. Tehtäväluettelo ja ajoitus suunnitelma	56
Liite 5. Toteutumaseuranta	60
Liite 6. Aikataulu, suunniteltu verrattuna toteutuneeseen	63
Liite 7. Käyttötapaukset.....	65
Salaiset liitteet	66

1 Johdanto

Yritykselle on tärkeää tietää, miten sillä kullakin hetkellä menee. Tämän tiedon avulla se voi varautua muutoksiin ja ennakoida markkinoita sekä tehdä päätöksiä, joilla voi olla kauaskantoisiakin vaikutuksia. Perustavanlaatuinen ongelma yrityksen hallinnassa on, ettei yrityksen johto tai sen työntekijät ole tietoisia, miten yrityksellä kullakin hetkellä menee. Tätä ongelmaa varten on kahden viime vuosikymmenen kuluessa vähitellen kehittynyt idea ohjausnäköymästä, jonka avulla yrityksen toimintaa voidaan monitoroida.

Ohjausnäköymä esittää käyttäjälle suoraan yrityksen sen hetken tilanteen, ja toimii siten yrityksen sisäisenä informointikanavana. Se on näköymä, jonka avulla voidaan yhdellä silmäyksellä nähdä tärkeimmät tiedot yrityksen toiminnasta. Se sisältää usein mittareita ja kuvaajia, ja sen perimmäisenä tarkoituksena on tiedon selkeyttäminen ja läpinäkyvyyden parantaminen. (Few 2004.)

Ohjausnäköymän kohteena on tässä työssä toimintajärjestelmä IMS, jonka sisältämillä työkaluilla yrityksen toimintaa voidaan hallinnoida. Sisältö on yrityskohtaista ja siten hyvin dynaamista. Lisäksi sisällön laajuus ja käyttöasteet vaihtelevat toimintajärjestelmää käyttävästä tahosta riippuen. Vaikka sisältökokonaisuudet ovat käyttötarkoituksiltaan yrityksen toimintaa luotaavaa, poikkeavat niiden esitystavat hyvin laajasti toisistaan.

Tämä opinnäytetyö kertoo, mitä ohjausnäköymän suunnittelua ja toteutusta tehdessä tulee pitää mielessä. Lisäksi keskitytään siihen, mitä eri tekijöitä voi tulla vastaan johtuen kehitysalustasta. Työ tehdään toimeksiantona IMS Business Solutions Oy:lle.

Työn lopputuloksena syntyy valmis käyttöliittymäpuoli toimeksiantajan IMS-toimintajärjestelmän sisäiselle ohjausnäköymälle. Tavoitteena on luoda käytettävyydeltään hyvä ja käyttökohteelleen ominainen näköymä, josta yritys saa tietoonsa kokonaistilanteen yrityksen toiminnasta. Työssä painotetaan web-käyttöliittymäsuunnittelua sivujen mobiilia alustaa.

Työn tehtävänä on määritellä, suunnitella ja käyttöliittymän osalta luoda IMS-sovelluksen kirjautumisen jälkeinen aloitusosio mobiili- sekä web-käyttöliittymille. Käyttöliittymän luonnilla tarkoitetaan, että käyttöliittymä tehdään toimivaksi kokonaisuudeksi.

Käyttöliittymä tulee pitämään sisällään eri tietokokonaisuuksia toimintajärjestelmän sisällöstä, sekä muokkausmahdollisuuden, jotta asiakasyritykset voivat tehdä omalle kohdalleen parhaiten toimivan ohjausnäkyvän. Se, mitä eri tietokokonaisuuksia ohjausnäkyvään tulee, ratkaistaan työn aikana.

Työssä käydään läpi mallintamisen lisäksi sitä, mitä eri asioita käytettävyyden osalta suunnittelussa tulee vastaan. Suunnittelun lisäksi pureudutaan eri käyttöliittymän toteuttamisen keinoihin, käyden läpi testausta ja työn validointia.

Työ on rajattu kattamaan vain käyttöliittymäosuus ohjausnäkyvän luonnista. Työhön ei kuulu tietokannan luomista tai muokkaamista. Palvelinpuolen muutoksia työhön ei myöskään kuulu. Mobiilikäyttöliittymällä tarkoitetaan tässä, että käyttöliittymä taipuu mobiilikokoon ja toimii Safari-selaimella. Web-käyttöliittymän osalta taas tarkoitetaan sitä, että käyttöliittymä toimii IMSin tukemilla selaimilla niin pienempien kuin suurempien resoluutioiden kanssa.

Luvuissa 2 ja 3 esitellään se tietotaito ja informaatio, mitä työn tekemisessä hyödynnetään. Luvun 4 kohdalla esitellään työn kohde ja tavoitteet tarkemmin. Luvussa 5 mietitään, miten käydyt tiedot voidaan ottaa huomioon valitun käyttökohteen tapauksessa, sekä käydään yksityiskohtaisesti läpi työn tehdyt vaiheet ja aikaansaannokset. Luku 6 on varattu tehdyn työn pohdinnalle. Muun muassa mitä tehtiin hyvin ja mikä olisi voinut onnistua paremmin, ovat tämän luvun vastaavia kysymyksiä.

Liitteessä Liite 1 esitellään raportissa vastaan tulevat keskeiset käsitteet.

2 Ohjausnäkyvä

Mikä ohjausnäkyvä on? Mikä sen merkitys on ja mitä sillä voidaan saavuttaa, jos se luodaan yrityksen käyttöön? Vaikka toimintajärjestelmä on sellaisenaan toimiva osa ylläpitämään kuvaa siitä, mitä yritys tekee ja mitkä sen tavoitteet ovat, voidaan käyttöön ottaa myös ohjausnäkyvä ja kasvattaa entisestään yrityksen sisäistä kommunikaatiota ja tilanteenhallintaa.

Wayne Eckersonin, Stephen Fewn ja Edward Tuften merkitys ohjausnäkyvän yleistykselle ja sen käytänteiden ja parhaiden käytäntöjen hahmottamiselle on verraton. Kukin heistä on antanut oman panoksensa ohjausnäkyvän kehitykselle ja ilman heitä, käsitteenäkin ohjausnäkyvä saattaisi yhä olla tuntematon.

Ohjausnäkyvä tunnetaan paremmin sen englanninkielisellä nimellä dashboard. Tiedonesittämisen konsultointiyrityksen perustajajäsenen ja kirjailijan Stephen Fewn (2004) mukaan ohjausnäkyvä on näkyvä, jossa esitetään tavoitteisiin pääsemiseen tarvittavat tärkeimmät tiedot. Hän lisää, että tieto on oltava luettavissa yhdellä silmäyksellä.

Ideana ohjausnäkyvä on ollut jo pitkään tiedossa, esimerkiksi autojen kojelautoissa. Wayne Eckerson, kirjailija, konsultti ja tietotekniikan tutkija, kertoo kirjassaan *Performance dashboards* (2011a, 5), että alkujaan ohjausnäkyvän tuonti tietotekniikan ja yritysten hallinnan käyttöön alkoi 1980-luvulla. Yritysjohdajat toivoivat pystyvänsä hallitsemaan yrityksiään omista työhuoneistaan käsin. Ajatus ei kuitenkaan sellaisenaan kantanut. Tiedon kasaaminen ja laitteet tulivat liian kalliiksi ja toisaalta tieto oli liian harvan ihmisen nähtävillä ollakseen hyödyksi.

Syntyi tarve luoda kokonaisuus, josta pystyisi näkemään, mitä yrityksessä kulloinkin tapahtuu sekä tarve pystyä hallitsemaan kulloistakin tilannetta. Erinäisiä ratkaisuja syntyi seuraavan kahden vuosikymmenen aikana, joista merkittävin oli idea yritystietoisuudesta (engl. *Business Intelligence*). Tämä yhdistettiin yrityksen toiminnanhallintaan, ja idea tiedon ohjausnäkyvästä oli syntynyt. (Eckerson 2011a, 6.)

Ohjausnäkyvän tunnusmerkkeihin kuuluu tiedon läpinäkyvyys. Tietoa ei salailta tai yritetä kaunistella. Tiedon avulla voidaan yrityksen toimintaa parantaa ja lisätä sisäistä sekä ulkoista koordinaatiota. Parannetun toiminnan ja tiedon näkyvyyden myötä myös yrityksen työntekijät motivoituvat tekemään parempaa työtä (Eckerson 2011a, 7-8).

Ohjausnäkyviä on kolmenlaisia. Toiminnallinen ohjausnäkyvä keskittyy enemmän tiedon monitorointiin. Tieto on ajantasaista ja tarkkaa, ja sitä voidaan pienissä määrin hallinnoida. Taktinen ohjausnäkyvä keskittyy tiedon analysointiin ja sen myötä toiminnan parantamiseen. Strateginen ohjausnäkyvä taas on yrityksen hallintaan. Se ei keskity niinkään tiedon ajantasaisuuteen, vaan tietyn ajanjakson tuloksiin verrattuna aiempaan tai suunnitelmiin. (Eckerson 2011a, 101–121, 260.)

Vain toiminnallisen ohjausnäkyvän tarkoitusperiin kuuluu jonkinasteinen tiedon muokkaus (Eckerson 2011a, 260). Kriittisissä tilanteissa voidaan suorittaa massa- tai yksittäiskomentoja, jotka raukaisevat tilanteen. Nämä toiminnalliset vaatimukset tulee ottaa huomioon ohjausnäkyvää suunniteltaessa.

Jos saman tehtävän voi tehdä toisella keinolla tehokkaammin, ei ohjausnäkyvä silloin kata tehtäväänsä. Ohjausnäkyvää ajateltaessa tulee huomioida, kenelle ohjausnäkyvää luodaan, miten ohjausnäkyvä tulee auttamaan häntä työskentelemään paremmin, älykäämmin ja nopeammin, ja mitä eri toimenpiteitä ohjausnäkyvän käyttö vaatii käyttäjältä (Optimal Solutions 2011).

Ohjausnäkyvän suunnittelu on aikaa vievää ja vaatii paljon ajattelua. Se, kenelle tai mihin tarkoitukseen ohjausnäkyvä tullaan luomaan, luo omat haasteensa suunnitteluun. Näytettävällä tiedolla on omat vaatimuksensa. Jos tietoa on paljon näytettäväksi, sen karsiminen ja jakaminen osiin vaatii taitoa ja tietämystä aiheesta.

3 Ohjausnäkyvän suunnittelu ja toteutus

Tässä luvussa keskitytään siihen, miten ohjausnäkyvä voidaan luoda ja mitä sen suunnittelun aikana tulisi pitää mielessä. Mallintamisesta lopullisen version toteuttamiseen sisältää monia riskitilanteita, jotka voidaan ehkäistä kattavalla käyttöliittymäsuunnittelulla. Lopuksi tulee varmistaa, että tehty tuote toimii odotetulla tavalla kattavan validoinnin ja testauksen avulla.

3.1 Käyttöliittymäsuunnittelu

Edellisen luvun perusteella tiedon läpinäkyvyys, käytettävyys ja käyttöystävällisyys kuuluvat ohjausnäkyvän käyttöliittymäsuunnittelun perusvaatimuksiin. Riippuen ohjausnäkyvän tyypistä tulee myös vaatimuksiin lisätä ajantasaisuus ja tiedon saannin nopeus.

Vaikka suunnittelussa huomioitaisiinkin jokainen näistä vaatimuksista, voidaan siinä silti kompastua virheisiin. Yleisimpiin kompastuskiviin kuuluu se, että näkyvä ahdetaan liian täyteen näytettävää tietoa. Näkyvästä tulee tällöin vaikeaselkoinen ja sen käyttö hankaloituu. Näytettäväksi tiedoksi voidaan myös kuvitella jokin, mikä ei näkyvään kuuluisi lainkaan. Käyttäjälle ei tällöin kerrota, mitä hän haluaisi tietää. (Eckerson 2011a, 223–224.)

Käyttöliittymää suunnitellessa on siksi hyvä ensin kartoittaa käyttäjä - minkä taitotasoinen käyttäjä on ja mitä hän haluaa käyttöliittymältä. Käyttöliittymäsuunnittelu alkaa siis tietojen kartoittamisella ja aiheeseen tutustumisella. (Eckerson 2011b.)

ISO 9241–11 -standardin mukaan käytettävyydellä tarkoitetaan sitä laajuutta, millä tuotetta voidaan käyttää tietyllä käyttäjäryhmällä tiettyyn käyttötarkoitukseen tiettyjen tavoitteiden saavuttamiseksi tehokkaasti, vaikuttavasti ja tyydyttävästi (Bevan 2006). Toisin sanoen käytettävyys on kiinnitetty kontekstiin, jossa käyttöä tehdään. Käyttäjän henkilökohtaiset piirteet, kuten käyttötaito tai kokemus laitteistosta, vaikuttaa myös käytettävyyteen, kuten myös käyttötarkoitus. Käytettävyyden kokemukseen taas vaikuttaa käytön vaikuttavuus ja tehokkuus, sekä käyttäjän tyytyväisyys suoritukseen.

Edward Tufte on Yhdysvaltojen Yalen yliopiston poliittisen tieteen ja tilastotieteen professori. Hän kertoo kirjassaan, että lukijaa tulee ohjata keskittymään olennaiseen, eli itse tietoon. Se, miten tieto esitetään, on myös tärkeää, sillä erinäisillä korostuksilla voidaan lukijan huomio keskittää eri kohtiin. (Tufte 2001, 13.)

Jos mahdollista näytettävää tietoa on liikaa, se voidaan ryhmittää. Ryhmitettäessä tulee kysyä, miten eri osakokonaisuudet liittyvät toisiinsa ja mitä samaa niissä on, vai aiheuttaako toinen kenties toisen (Few 2009, 11). Ryhmän tietokokonaisuuksia voi erottaa toisista tyhjällä alueella tai rajaamalla tai ympäröimällä ne erilleen. Eri tietokokonaisuuksien ryhmittämisessä voidaan tukeutua Gestaltin lakeihin (Tuck 2010) kun taas yhden tai useamman erottamisessa kokonaisuudesta voidaan käyttää pohjana ihmisen vaistonvaraisen käsittelyn ominaisuuksia (Healey 2009).

Vaistonvarainen käsittely tapahtuu silmissä, ennen kuin tieto saapuu aivoihin. Jopa 70 % ihmisen aistisoluista on silmissä (Few 2009, 29). Vaistonvaraisen käsittelyn määreitä ovat malliin viittaavat pituus, leveys, kallistus, koko, muoto ja rajaus, avaruudelliseen sijaintiin viittaava 2D sijainti ja väritykseen viittaavat sävy ja värikylläisyys (Few 2012, 68). Tietyn kohteen erottamiseen muista parhaiten sopivat näistä muoto ja sävy (Few 2009, 40).

Näytettävän tiedon karsiminen on myös ratkaisu tietopaljouteen. Karsimisella tarkoitetaan, että tietokokonaisuudesta poistetaan kaikki ylimääräinen, jolloin mahdollistetaan vain olennaiseen keskittyminen (Few 2012, 298). Fewn mukaan tietoa karsittaessa auttaa esimerkiksi se, ettei näkymässä tarvitse ilmoittaa jos kaikki on hyvin (Eckerson 2011a, 232). Tulkitsemme hiljaisuuden myöntämisen merkiksi, eikä ole poikkeus, että ajattelempa ilmoitusten puuttumisen merkiksi siitä, että kaikki on hyvin.

Karsimisen ohella tiedon suodattaminen voidaan mahdollistaa. Siinä tulisi miettiä, onko suodatuskeino käytössä kaikille näkymän tietokokonaisuuksille vai vain joillekin. Tietylle kaaviolle tarkoitettu suodatin on hyvä esittää suoraan kyseisen kaavion yläpuolella. (Eckerson 2011a, 235.)

Ohjausnäkyvän voi luoda niin web- kuin mobiilillekin alustalle. Mobiilin käyttöliittymän kohdalla W3C, kansainvälinen yritysten ja yhteisöjen yhteenliittymä, on luonut listan parhaita käytäntöjä (2008), joita tulisi huomioida suunnittelussa. Lista kattaa 60 eri kohtaa aina tekstin luonteesta linkkien sijaintiin.

Pieni näyttökoko katsotaan usein suurimmaksi haasteeksi mobiilia käyttöliittymää suunniteltaessa. Koon ollessa pienempi se, miten ruudun tilaa käyttää ja miten eri sivun osat rakentaa yhteen nousee tärkeäksi suunnitteluvaiheeksi. Usein turhaa tilaa vievät visuaaliset tehosteet jätetään pois tilaa viemästä, ja tilaa kulutetaan vain yksinkertaisiin kaavioihin, mittareihin ja taulukoihin. Vaikka koko onkin pieni, täytyy silmä silti kohdistaa johonkin. Se, että käyttäjille huomautetaan tärkeästä kohdasta, on tärkeää niin mobiili- kuin web-käyttöliittymää suunniteltaessa. (Chiang 2011.)

3.2 Mallintaminen

Suunnitelmia voidaan kuvata auki eräänlaisina alkukahmotelmina, joita kutsutaan mallinnuksiksi. Käyttöliittymää voidaan mallintaa monin eri tavoin, muun muassa hahmottelemalla kuvakokonaisuuksia wireframe-tekniikalla, mockup-tekniikalla tai tekemällä jotenkuten interaktiivinen demoversio eli prototyyppi kohteesta.

Mallintaessa käytettävyyden tutkintaan kuuluu osanaan affordanssien hahmottaminen. Affordansseilla tarkoitetaan sitä, miten tunnistamme eri osat sivulla ja miten reagoimme niihin. William Gaver (1991) jakoi nämä tyypeittäin kolmeen eri osaan: aistittavat, piilotetut ja valheelliset. Aistittavat ovat selkeitä affordansseja. Kohotettua neliötä voi painaa, mutta harmaana painike on suljettu, Gaver selittää. Oikealle osoittavasta nuolesta voi päästä seuraavaan osioon. Piilotetut affordanssit ovat hänen mukaan taas niitä, jotka ovat tapahtumanlähteitä, muttemme niitä sellaisiksi miellä. Valheelliset ovat taasen niitä, mitkä miellämme tapahtumanlähteiksi, mutta jotka eivät todellisuudessa tee mitään.

Jopa ruudulla valon uskotaan tulevan ylhäältä päin. Siksi alhaalta päin tummemmaksi varjostettu painike näyttää, että sitä voidaan painaa, kun taas päinvastoin varjostettu antaa kuvan, että painiketta on jo painettu. (Few 2012, 62.)

Teksti ei affordoi muuta kuin lukemaan. Ihminen käsittelee kuvat eri tavalla kuin tekstin. Kuvaajien kohdalla silmä kertoo, mitä kuvassa tapahtuu, toisin kuin tekstin kohdalla aivot joutuvat töihin, jotta teksti ymmärrettäisiin. (Few 2005.)

Kuten aiemmin on mainittu, ohjausnäkyvän tulisi olla sisäistettävissä yhdellä silmäyksellä. Ohjausnäkyvän sisältö on tästä syystä melkein poikkeuksetta kaavioita, mittareita ja erinäisiä kuvaajia.

Kuvaajia tulisi käyttää silloin, kun halutaan keskittyä kokonaiskuvaan, tai kun piirretty malli tuo lisätietoa aiheesta. Jos taas halutaan keskittyä yksittäisiin arvoihin tai verrata arvoja keskenään, kannattaa käyttää taulukkoja. (Few 2012, 44-51.)

Kuvaajien lisäksi lukijan huomion voi kiinnittää monin muin visuaalisin keinoin. Värejä tulee kuitenkin käyttää säästeliäästi. Niihin ei saisi tukeutua, ottaen huomioon värisokeat. Värit voivat myös hämätä katsojaa ja estää häntä keskittymästä olennaiseen eli itse tietosisältöön. (Eckerson 2011a, 238-239.)

Alexander Chiang, Dundas Data Visualization yhtiön tuotehallinnon varapääjohtaja omaa vuosien kokemuksen käyttöliittymien teosta (Chiang 2011). Hän mainitsee artikkelissaan (2009), että yleensä yrityksillä on omat värimaailmansa jota seurata. Hän jatkaa, että värimaailman ollessa suppea (alle kahdeksan väriä) voidaan sopivia värejä lisätä erinäisillä palettigeneraattoreilla. Tällaisia ovat esimerkiksi Collor ja Pictaculous.

Tausta- tai liukuvärejä tulisi välttää varsinkin kaavioiden pohjalla, sillä ne voivat vääristää kuvaajan sanomaa (Few 2012, 252). Värien lisäksi fonttien ja fonttikoon valinnoissa tulisi keskittyä luettavuuteen (Eckerson 2011a, 236).

Jo pelkällä sijainnilla on merkitystä. Länsimaalaiset aloittavat sivun lukemisen vasemmasta ylänurkasta. Tärkein ja kriittisin tieto tulisikin kohdistaa sinne. Myös keskikohta mielletään usein tärkeäksi, varsinkin jos se on jotenkin korostettuna muusta sisällöstä. (Eckerson 2011a, 234.)

Se, miltä jokin asia tulee myöhemmin näyttämään, hahmotellaan usein mallintamalla asia erinäisin keinoin. Näitä keinoja ovat muun muassa wireframe-tekniikka, mockup-tekniikka sekä prototyypin tekeminen mallinnuksen kohteesta.

Wireframe-tekniikan avulla kuvataan usein ohjelman käyttöliittymää ja navigaatiota, harvemmin toiminnallisuutta. Tekniikan käytöstä syntyy karkeita hahmotelmia siitä, miltä varsinainen sovellus tulisi näyttämään ja sisältämään mallinnuksen pohjalta (Chapman, C. 2010). Varjopuolena tekniikan käytössä on toiminnallisuuden ja elävöittämisen puute. Se, miten sivu latautuu ja miten eri sivun osat ilmestyvät näkyviin, tai millainen toiminto tai ääni syntyy klikattaessa jotain sivun osaa, jää kuvakaappauksista ulkopuolelle ja täytyy usein selittää sanoin kuvan ulkopuolella tai puhekuplin kuvassa.

Wireframe-tekniikan käyttöön on olemassa monia eri työkaluja. Useimmat työkaluista ovat maksullisia, kuten ForeUI ja Balsamiq Mockups. On myös olemassa ilmaisia tai osittain ilmaisia sovelluksia, riippuen käyttötarkoituksesta (muun muassa yritys- vai yksityiskäyttöön). Tällaisia ovat esimerkiksi Mockup Builder ja Pencil. Kuvauksia voi myös tehdä perinteiseen tapaan kynällä ja paperilla, tai eri kuvankäsittely-työkaluilla. Eri työkaluja on verrattu dokumentaation, lisenssivaatimusten, maksullisuuden, helppokäyttöisyyden ja ominaisuuksien laajuuden osalta liitteessä Liite 2.

Wireframe-tekniikan lisäksi mallinnuksia voidaan toteuttaa mockup-tekniikalla. Toisin kuin wireframe-tekniikassa, mockup-tekniikkaa käytettäessä keskitytään ulkoasuun ja yleisilmeeseen. Myös tästä syntyy käyttöliittymästä kuvakaappauksia, mutta ne ovat jo hyvin tarkempia ja lähempänä todellisuutta kuin hyvin karkean tason kuvaelmat, joita wireframe-tekniikasta syntyy (Chapman, C. 2010). Myös tämän tekniikan käyttö omaa huonona puolenaan toiminnallisuuden kuvaamisen puutteen.

Mallinnettaessa prototyypin tekeminen kuvattavasta ohjelmasta, toiminnallisuudesta tai käyttöliittymästä antaa usein hyvin yksityiskohtaisen ja kokonaislaatuksen kuvan ohjelmasta (Chapman, C. 2010). Usein uusinta ja toimivinta demoversiota myös lähdetään suoraan jatkokehittämään varsinaiseksi sovellukseksi. Html-pohjaisen demoversion Ajax-toiminnallisuutta voidaan hahmotella esimerkiksi Protoscript-kielellä.

Prototyypin tekeminen sovelluksesta voi olla hyvin aikaa vievää. Yhdenkin demoversi-
on tekeminen voi viedä yhtä paljon aikaa ja vaivaa kuin varsinaisen ohjelman teko.
Demoversi-
on teko jokaisesta eri prototyypiversiosta erikseen voi olla paljon raskaam-
paa kuin yleisilmeen ja toiminnallisuuden kuvaaminen sanoin ja kuvauksin.

3.3 Ohjausnäkyvän toteuttaminen

Mallinnusten luonnin jälkeen käyttöliittymäsuunnittelu on ohi, ja varsinaisen käyttöliit-
tymän toteuttaminen voi alkaa.

Se, millä arkkitehtuurilla käyttöliittymä toteutetaan, on usein perustavanlaatuinen kysy-
mys. Miten palvelimen ja käyttöliittymän välinen rajapinta luodaan? Minkä muotoista
tietoa käyttöliittymä ottaa kyselyistä vastaan? Mahdollisista tiedonsaannin arkkitehtuu-
reista käytetyimpiä ovat SOAP ja REST. SOAP-pohjaisia ratkaisuja on ollut käytössä jo
pitkään. REST taas on uudempi rakenne, tosin se on ollut ideana jo HTML-kielen syn-
nystä lähtien.

RESTin koko idean voi pohjauttaa siihen, että se rakentaa kaikista asioista sivuilla yksi-
löitä, joiden tilaa säädetään URL-kyselyillä. Kyselyn vastauksena saadaan yksilön kaikki
tiedot mukaan lukien yhteydet muihin yksilöihin ja toimintoihin, sekä yksilön senhetki-
nen tila (muun muassa katselu- ja muokkaustilat). Näin sekä palvelimen että selaimen ei
tarvitse muistaa, missä tilassa yksilö on. (Tilkov 2007.)

RESTin avulla voidaan rakentaa URL-kyselyitä, joilla palvelimelta voidaan saada tietoa
eri sivupohjille. Sama tieto voidaan siten esittää eri lailla. Tämän lisäksi samaa sivupoh-
jaa voidaan käyttää moneen eri tarkoitukseen, ja saadun tiedon mukaan se voidaan täyt-
tää eri osakokonaisuuksilla. Tämän toteutukseen voidaan käyttää mallintamista (engl.
template). Osakokonaisuuden pohja mallinnetaan esimerkiksi Handlebars- ja JSON
Template-kielillä. Sen jälkeen malliin syötetään palvelimelta saadut tiedot ja se kopioi-
daan haluttuun kohtaan sivua. Näin mallista voidaan tehdä useita ikään kuin valoksia.

Käytettävän arkkitehtuurin päätettyä voidaan päättää myös käytettävät ohjelmointikie-
let. Tehdäänkö käyttöliittymä HTML- vai XHTML-pohjaisena? Muun muassa vir-

heidenkäsittely eroaa vahvasti HTML:n ja XHTML:n kesken. HTML:ää käyttävät selaimet usein jättävät huomiotta virheet ja lataavat sivun niiltä osin kuin se toimii.

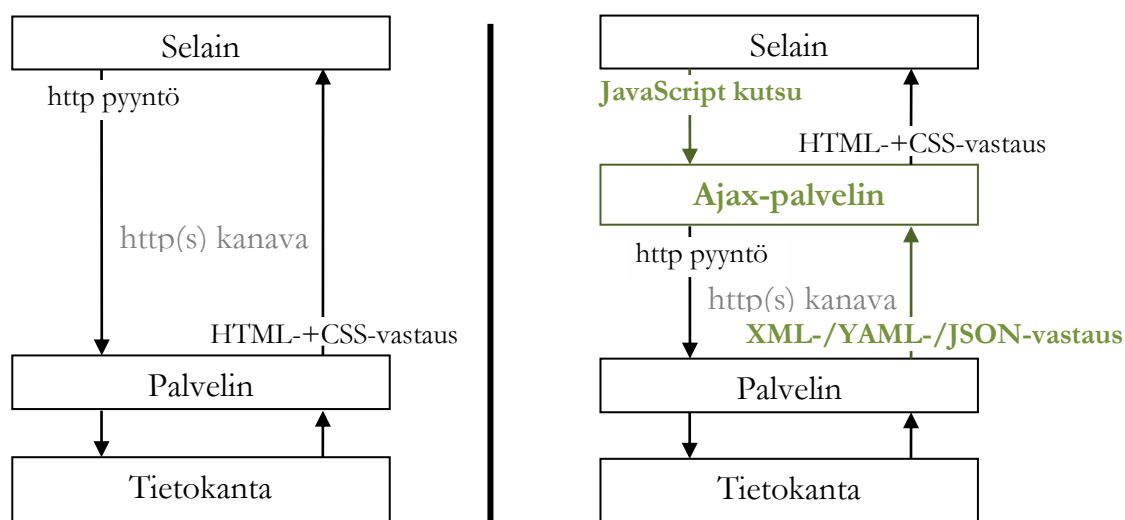
Ongelmana XHTML:n käytössä on, että Microsoft Internet Explorer (IE) – selain ei vielä tue sitä ollenkaan. Kyseinen selain antaa vain latausvaihtoehdon, jos sovellus on merkitty XHTML-kielen viittaavalla `application/xhtml+xml` mime tyyppillä. XHTML-ohjelmoijat vaihtavatkin mime tyyppiä HTML-kielen `text/html`, jolloin selaimet käsittelevät sivun HTML-kielenä, tai vaihtoehtoisesti tekevät erillisen sivun IE:n käyttöön. Siten he varautuvat tulevaisuuteen, jossa IE alkaisi tukea heidän paremmaksi ratkaisuksi katsomaansa XHTML:ää. Useimmat ohjelmoijat kuitenkin vain tyytyvät toteuttamaan sivunsa pelkällä HTML:llä. (Stachowiak 2006.)

Kenties halutaan ottaa valitun pohjan kanssa käyttöön myös PHP, JSP tai ASP, joilla palvelimelta selaimen tuotua sisältöä voidaan liittää pohjaan dynaamisesti. Koska nämä kaikki käsittelevät sivun ennen sen lähettämistä käyttäjälle, täytyy pohja ladata aina uudelleen, jos tietoa halutaan päivittää sivulla (Php.net, 2013; Oracle.com, 2010; Cluts 1997).

Tämän sivunlatausongelman voi ratkaista lisäämällä JavaScript-tuen vaatimuksena sivun katsomiseen. Jos toteutusta aiotaan lähteä tekemään JavaScriptin, JavaScript-kirjasto jQueryn tai JavaScriptiin vahvasti liittyvän Ajaxin avustuksella, täytyy kyseisen tuen olemassaolo varmistaa. Toinen vaihtoehto on tehdä näillä vain pieniä tarkistuksia tai elävöittämistä, joka ei vaikuta sovelluksen toimivuuteen ratkaisevasti. Tarkistukset voidaan esimerkiksi tehdä uudelleen palvelinpuolella. (Chapman, S. 2009.)

JavaScriptin toisena ongelmana on, että se käyttää pohjanaan DOM-kieltä. DOMin selaintenväliset erot aiheuttavat ongelmaa. Selainten eri käsitykset ja ratkaisut sen eri ominaisuuksiin aiheuttavat ongelmaa myös sitä käyttävälle JavaScriptille. Yleinen ratkaisu onkin vuodesta 2011 lähtien ollut liittää JavaScriptiin kirjasto, joka yhdistää nämä käsitykset. Siten voidaan varmistaa, että joka selaimella sama toiminnallisuus toimii samalla tavalla. Tällaisia kirjastoja on esimerkiksi mainittu jQuery, Prototype, ja YUI. (Mozilla Developer Network 2012.)

Erona tavallisen web-sovelluksen toiminnassa verrattuna Ajaxilla rakennettuun on, että Ajaxilla toteutetun sovelluksen kohdalla sivun osia pystytään päivittämään reaaliaikaisesti ilman, että koko sivu joudutaan hakemaan uudelleen palvelimelta. Näkymä toimii tällöin vain runkona, johon syötetään kuhunkin tilanteeseen sopivaa tietoa ilman, että käyttäjän tarvitsisi odottaa palvelimen vastausta tai sivun latautumista uutta tietoa haettaessa kannasta. (Kuvio 1.)



Kuvio 1. Klassisen ja Ajax-websovelluksen toiminta (Garrett 2005)

Toiminnallisuuden jälkeen kuvaan nousee visuaalisen ilmeen parantaminen suunnitteluvaiheena. Sen osalta käytössä on Cascading Style Sheet-kieli, jonka avulla sivun visuaalista ulkonäköä voidaan muuttaa.

CSS-kielen uusimmasta versiosta CSS3-kielestä on johdettu eri laajennoksia, kuten Sass (Syntactically Awesome Stylesheets), SCSS (Sassy CSS) ja LESS. Niiden avulla CSS:ää voidaan toteuttaa joustavammin ja vähentää CSS-ohjelmoinnissa tapahtuvaa toistoa. Ne omaavat kukin lisäominaisuuksia, joita CSS3 yksin ei kata, kuten ankkureita ja selainriippumattomia määreitä. (Sass-lang.com 2006; Graham 2011; Sellier 2010.)

Käyttöliittymä voi olla hyvin tyhjä, jos se ei sisällä minkäänlaista grafiikkaa. Kuvien ja värien avulla käyttöliittymän ilmettä voidaan parantaa paljon. Tähän tarkoitukseen voidaan joko luoda itse omia kuvia tai käyttää internetin kautta löydettyjä, luvalla saatuja kuvia. Käyttöliittymän tekoon voidaan saada pieniä kuvia esimerkiksi Icomoon- tai FatCow-sivustoilta.

3.4 Koodin validointi ja testaus

Web-käyttöliittymiä tehtäessä tulee usein tehtyä virheitä, jotka eivät välttämättä näy päällepäin. Joillain selaimilla saattaa kuitenkin ilmetä ongelmia, jos lähdekoodissa esiin-tyy virheitä. Siksi tehty koodi on hyvä validoida säännöllisin väliajoin.

Taulukko 1. Validaattorien vertailu

Käyttö	Validaattori				
	WDG	CSE	Html	W3C	JSONLint
URL syöttö, sivun validointi	X			X	X
Tiedoston validointi	X	X		X	
Koodisyötön validointi	X			X	X
Validointi selaimen kautta			X		
Validoi HTML-kieltä	X	X	X		
Validoi HTML5-kieltä		X	X	X	
Validoi XHTML-kieltä	X	X		X	
Validoi CSS-kieltä		X		X	
Validoi CSS3-kieltä		X			
Validoi JavaScriptiä		X			
Validoi JSONia					X
Toimii ilman Internet-yhteyttä	X	X			

Validaattoreita on useita, kuten WDG Validator, CSE Validator ja Firefox -selaimelle tarkoitettu Html Validator-lisäosa. Tunnetuin ja käytetyin validaattori on kenties W3C Validator. Validaattorien vertailussa voidaan huomioda esimerkiksi niiden kielikatta- vuus ja käytön helppous (Taulukko 1).

Kun sivu on valmis, se täytyy testata, ennen kuin se voidaan ottaa käyttöön. Testausta suoritetaan jatkuvasti kehitysvaiheessa. Kun kaikki haluttu toiminnallisuus ja tyyli- muutokset on tehty, täytyy sovellus läpikäydä vielä viimeinen, kattava testaus. Koska kaikki toiminnallisuudet eivät välttämättä toimi yhdessä tai jokin selainalusta on jäänyt testaa-

matta kaikilta osin toteutuksen aikana, tämä vaihe on hyvin kriittinen loppukäyttäjän tyytyväisyyden takaamiseksi.

Käytettävyyttä testatessa sen vaikuttavuuden mittarina voidaan käyttää onnistumisprosenttia, eli sitä, kuinka monta prosenttia jonkin tehtävän suorittamista kokeillutta käyttäjää onnistui tehtävässään. Tyydyttävyyden mittarina taas on tehtävän suorittamisen vaikeusaste sekä myös kommenttien positiivisuus tai negatiivisuus. (Travis 2009.)

Tehokkuuden mittaamiseen käytetään sitä aikaa, joka käyttäjiltä kului tehtävän suorittamiseen verrattuna muihin keinoihin ja käyttökertoihin (Travis 2009). Tiedon saannin nopeus on siis tärkeää käytettävyydelle. Yleisesti käyttäjä turhautuu, jos hän joutuu odottamaan ohjelman reagointia kauemmin kuin viisi sekuntia (Eckerson 2011a, 256).

Web-käyttöliittymän toimintaa voidaan testata esimerkiksi Selenium -testeillä. Selenium on työkalu, johon voidaan tallentaa käyttötapaus. Käyttötapaus ajetaan ohjelmaan käymällä käyttötapaus sivun eri rakenteiden läpi. Kun ohjelmaa muutetaan tai uusi ominaisuus lisätään joka on yhteydessä vanhaan osioon, on hyvä ajaa käyttötapaukset uudelleen. Tällöin voidaan varmistua, ettei uusi ominaisuus tai koodi ole rikkonut vanhaa toiminnallisuutta.

Seleniumin lisäksi on olemassa monta muutakin työkalua käyttöliittymän toiminnallisuuden testaukseen, kuten Ranorex ja Watir. Molemmat näistä ovat maksullisia, mutta omaavat ilmaisen koekäyttöajan. Toisin kuin Selenium, ne toimivat myös eri selaimilla. Seleniumin selainkattavuus Firefoxin lisäksi on vasta kehitteillä.

Eri selainten välisten käyttöliittymäerojen testausta varten voidaan jokainen tuettu selain ladata koneelle ja sitten katsoa, miten sivu toimii. Jos tuettu selaimen versio on vanhempi kuin koneelle on asennettu, voi muodostua ongelmia selainten asennuksen kanssa. Tätä varten on olemassa maksullinen CrossBrowserTesting – sovellus. Maksutomaan käyttöön on muun muassa Adobe Browserlab-, Browsershots- ja Lunascape 6 (Taulukko 2).

Taulukko 2. Testaustyökalujen vertailu (Chapman, C. 2011.)

Työkalu	Tuettuja selaimia	IE?	Interaktiivinen testaus	Vierekkäinen selaus	Maksullisuus
CrossBrowserTesting	100+	IE6+	Kyllä	Kyllä	Maksullinen
Adobe Browserlab	13	IE6+	Ei	Kyllä	Maksuton
BrowserShots	60+	IE6+	Ei	Ei	Maksuton
Lunascape 6	3	IE6+	Kyllä	Kyllä	Maksuton
SuperPreview	5+	IE6+	Kyllä	Kyllä	Osittain maksuton
Spoon	16+	Ei	Kyllä	Ei	Maksuton

4 Kohde ja tavoitteet

IMS-sovellus on kehitysalusta, joka toimii käyttäjäyritysten laadunhallinta- toiminta- ja johtamisjärjestelmänä. Se on selainpohjainen, jonka toiminnallisuuksiin kuuluu tiedon säilyttämisen lisäksi sisällön muokkaus. Sovelluksella voidaan katsella, tehdä ja siihen voidaan tallentaa prosesseja, dokumentteja, käsikirjoja, raportteja, mittareita ja viestejä. (IMS 2013a.)

IMS-toimintajärjestelmää käyttää niin pienet, keskikokoiset kuin suuretkin yritykset. Yrityksiä ovat muun muassa Tampereen ammattiopisto, Iittala ja Helsingin kaupunki. Asiakkaita on tällä hetkellä yli 300. (IMS 2013b.)

Taulukko 3. IMS-toimintajärjestelmän osiot

Osion nimi	Kuvaus
Prosessit	Osion avulla yritys voi mallintaa toimintaansa liittyviä prosesseja.
Käsikirjat	Käsikirja-osiossa käyttäjä voi luoda esimerkiksi yrityksen toimintaohjeet.
Dokumentit	Käyttäjällä voi tallentaa tai luoda osioon kaikki yrityksen käyttöön eri dokumentit, tallenteet ja muistiinpanot.
Mittaristo	Osio avaa mahdollisuuden luoda yrityksen toiminnasta kaavioita ja kuvaajia.
Raportit	Käyttäjällä voi muun muassa luoda kyselylomakkeita ja palauteraportteja osioon sisältyvän editorin avulla.
Viestit	IMSiin kuuluu sisäinen viestintämahdollisuus. Niin yksityisten kuin järjestelmäviestien lähetyks on mahdollista Viestit-osion avulla.
Uutiset	Osiossa voidaan kirjoittaa ja välittää IMSin sisäisesti uutisia käyttäjien luettavaksi.
Haku	Hakuosion avulla käyttäjä voi hakea sisältöä tietystä osiosta tai koko IMSistä.
Ylläpito	Osio sisältää kaikki ne toiminnot, joilla IMSin toimintaa voidaan säätää.

IMS:n avulla voidaan seurata yrityksen kehitystä ja resursseja. Se on täysin räätälöitävissä kunkin yrityksen näköiseksi, jolloin se sulautuu osaksi yrityksen omaa intranettiä. Sovellus on jaettu yhteensä yhdeksään eri osioon, joista jokaisella on oma tarkoituksensa, ja joista jokainen tuo oman lisänsä yrityksen hallinnan käyttöön. (Taulukko 3.)

Kunkin osion hallintaan kuuluu mahdollisuus valita käytettävä kieli, sekä kenellä on oikeudet päästä kuhunkin eri osioon tai yksittäisiin sisältökokonaisuuksiin. IMS:n avulla voidaan myös lähettää tietoiskuja uudesta sisällöstä, ja päivityspyyntöjä vanhoiksi käyviin sisältöihin. Ohjelmistosta löytyy myös kattavat ohjeet kunkin osion käyttöön.

IMS-sovelluksen tekovaiheessa sekä nykyään sitä kehitettäessä käytettävyys on hyvin tärkeä tekijä. Sovelluksen halutaan olevan mahdollisimman lähestyttävä ja helposti käytönotettava. Tästä syystä aina uusia ominaisuuksia luotaessa sovelluksen käyttöliittymäsuunnitteluun jätetään runsaasti aikaa.

IMS-toimintajärjestelmän eri osioita käytetään samalla laajuudella ilman, että yksikään niistä olisi toisia laajempi tai tärkeämpi. Siksi aloitusosioiksi usein jätetäänkin oletusasetuksena oleva prosessit-osio taikka viestit-osio. Nämä yksittäiset osiot eivät kuitenkaan anna täyttä kuvaa siitä, mitä IMS:ssä on tapahtunut viime kirjautumisen jälkeen, ja yrityksen työntekijät joutuvatkin käymään jokaisen osion läpi erikseen, jotta he näkevät kaikki mahdolliset tulleet muutokset.

Tätä käytettävyysongelmaa varten on jo muutaman vuoden ajan haluttu luoda IMS:iin erillinen aloitusosio. Kysyntää tällaiselle osiolle on ollut yritysasiakkaiden keskuudessa paljon, mutta tavoitteen mukainen ominaisuus on jäänyt tekemättä muiden sovelluskehityshankkeiden korkeamman priorisoinnin vuoksi. Toteuttamisen aloittamista edisti aloitusosion tarve myös IMS:n kehiteillä olevaan mobiiliin alustaan.

Kun työ lopulta liitetään IMS-sovellukseen, se tulee lisäämään IMS-sovelluksen käytöstävällisyyttä ja käytön tehokkuutta. Yritysasiakkaiden työntekijät tulevat näkemään yhdellä silmäyksellä IMS:iin viime kirjautumisen jälkeen tulleen sisällön, eikä aikaa tule kulumaan jokaisen osion läpikäyntiin. Myös yritysasiakkaat tulevat siten saamaan talou-

dellista hyötyä. Käytettävyyden parantuessa myös käytön aloittaminen helpottuu. IMS sovelluksena selkenee ja osiot saadaan liitettyä tiiviimmin yhteen yhdeksi sovellukseksi.

Aloitusosio päätettiin toteuttaa hyödyntämällä ohjausnäkyvän ideaa. Ohjausnäkyvän taustalla on ajatus siitä, että näytettävä tieto on selkeää, eikä sitä yritetä kaunistella. Se ideana haroo kiinni yrityksen johdon ja työntekijöiden välistä tiedonkulun aukkoa. Sen avulla yrityksen tila tulee olemaan yrityksen sisällä yleisemmässä tiedossa. Ohjausnäkyvän valinta tähän kohteeseen myös edistää ohjausnäkyvän idean tunnettavuutta, ja kenties kannustaa esimerkinomaisesti muita yrityksiä luomaan omiin sovelluksiinsa omanlaisensa ohjausnäkyvän.

The Data Warehousing Instituten (TDWI) teettämän tutkimuksen mukaan ohjausnäkyvien tuonti yrityksen raportoinnin ja analysoinnin välineeksi on kokenut hurjan kasvun. Vuonna 2004 51 % tutkimukseen osallistuneista yrityksistä oli ottanut ohjausnäkyvän käyttöön. Vuoteen 2009 mennessä osuus oli jo 72 %. (Eckerson 2011a, 6.)

Vaarana ohjausnäkyvän käytössä oli, että se ahdettaisiin liian täyteen tietoa – varsinkin, kun tiedon lähteenä oli jokainen IMSin eri osio. Tiedon esittämisen lisäksi niin IMSin kuin ohjausnäkyvän perimmäisenä kohteena on käytettävyyden parantaminen. Siksi kaikkea tietoa ei tarvinnut näyttää. Se, mitä tiedosta tarvittiin ja mitä tietoa voitiin käyttää, karsivat sisältöä näkyvästä. Myös se, että jokin asia voitiin tehdä helpommin vain menemällä asianomaiseen osioon sen sijaan, että se tuotiin näkyväseen, karsi näytettävää tietoa.

Kun vertailtiin eri ohjausnäkyvätyyppejä, parhaiten työn tekoon sopi toiminnallinen ohjausnäkyvä. Sen tarkoituksena on hallinnoida sekä monitoroida melko ajantasaista tietoa. Siksi se oli omiaan käyttötarkoitusta ajatellen.

Aloitusosion käyttäjänä tulevat toimimaan IMSiä käyttävien yritysten työntekijät sekä esimiehet. Heidän IT-osaamisensa tasoerot voivat olla hyvin suuria. Myös niiden, jotka eivät käytä internetiä usein eivätkä siksi ole luoneet hahmotusta sen sisältämisestä eri affordansseista, täytyy pystyä käyttämään IMSiä oppimisvaiheen jälkeen tehokkaasti. Tätä

yrityksen aloitusosion toteutuksessa huomioida poistamalla turhaa visuaalista ilmettä näkymältä ja keskittymällä sisältöön.

5 Kohteen ohjausnäkyvän suunnittelu ja toteutus

Tässä luvussa kuvataan työvaiheita, millä työ toteutettiin sekä valittuja menetelmiä. Ensimmäinen työvaihe oli osakokonaisuuksien kartoittaminen ohjausnäkyvälle. Tässä vaiheessa hahmoteltiin, mitä ohjausnäkyvän halutaan kertovan käyttäjälle, ja mitä eri käyttötarinoita siihen yhdistyy. Tämän vaiheen jälkeen käsiteltiin käyttöliittymän mallintamista työkalujen ja työtapojen valinnasta aina lopullisten mallinnusten luontiin. Mallinnuksia luotiin tämän vaiheen osalta neljän ”mallinnus ja analysointi” – kierroksen ajan.

Kun mallintaminen saatiin valmiiksi, valittiin käytettävät toteutuskeinot ja lähdettiin luomaan varsinaista käyttöliittymää. Toteutuksen jälkeen työ eteni tuotetun käyttöliittymäkoodin validointiin ja lopputestaukseen.

5.1 Suunnittelu ja kartoitus

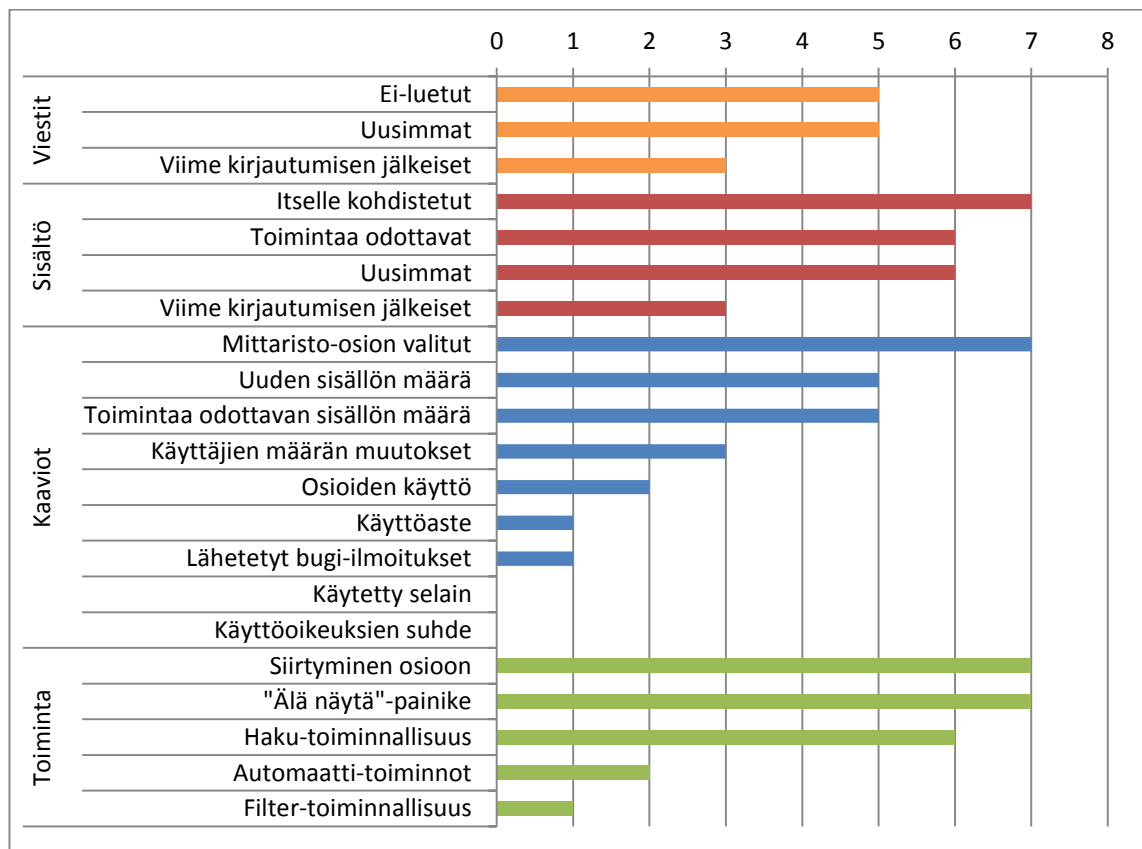
Ennen työn aloitusta tehtiin projektisuunnitelma. Työtä lähdettiin tekemään kartoittamalla käytettävät tekniikat ja kuvauskeinot. Kartoittaminen suoritettiin valitsemalla tilanteeseen, kohteeseen ja yrityksen omiin tarpeisiin parhaiten sopivat menetelmät. Tämä vaihe päätettiin toteuttaa pienimuotoisella kyselyllä, joka lähetettiin täytettäväksi yrityksen työntekijöille.

Kysely valittiin kartoituskeinoksi sen yksinkertaisuuden ja runsaan asialähtöisen rakenteen vuoksi. Kysely lähetettiin sähköpostitse eteenpäin siitä syystä, että haluttiin nopeasti tavoittaa työntekijät, mutta silti antaa heille aikaa päättää, milloin ehtivät parhaiten kyselyyn vastaamaan. Toisena vaihtoehtona kartoituksen toteuttamiseen olisi ollut henkilökohtainen kyseleminen, johon työntekijät voisivat vastata vapaamuotoisesti. Sillä yrityksen sisällä myös etättyö on mahdollista ja jotkut työntekijät ovat usein poissa työpaikalta yritystapaamisissa, tätä ei nähty toimivana ratkaisuna.

Kyselyssä kartoitettiin, mitä eri ominaisuuksia aloitusosioon toivottaisiin. Analysointia varten luotu kysely on kuvattu liitteessä Liite 3. Jokaisesta eri kohdasta avattiin myös viitteellä, mitä kullakin kohdalla tarkoitetaan.

Kyselyyn vastasi kymmenen yrityksen työntekijää. Kyselyn vastaukset analysoitiin katsoen kunkin ehdotuksen suosio, ja analysoinnista tehtiin kuvaaja (Kuvio 2). Sen jälkeen tulokset arvioitiin ottaen huomioon eri IMS-toimintajärjestelmän käyttäjäroolit ja oikeudet tehden lopullinen määrittely halutuista toiminnallisuuksista käyttötapauksiksi.

Taulukon perusteella viestit haluttaisiin tuoda näkymään, mutta viestien tyypistä ei löydetty selkeää voittajaa. Sisällön osalta ääniä saaneet ”itselle kohdistetut”, ”toimintaa odottavat” ja ”uusimmat” voitaisiin mahdollisesti yhdistää. Kaavioiden kohdalla Mittaristo-osion valitut palat, uuden sisällön ja toimintaa odottavan sisällön määrät olivat vaihtoehtoista suosituimpia. Uuden sisällön ja toimintaa odottavan sisällön määrät voitaisiin yhdistää sisältöosiin. Toiminnallisuuksista kärkeen nousi "älä näytä"-painike, osioon siirtyminen ja haku, tosin osioon siirtyminen mainittiin muutenkin vakioasetukseksi IMS-sovelluksessa. Automaattitoiminnot täytyisi ajatella auki, jotta voitaisiin käyttää. (Kuvio 2.)



Kuvio 2 Sisältökyselyn tulokset

Eri osioiden korvaavuuteen tuli hyvin paljon ehdotuksia. Uutisten ja muiden käyttäjien ilmoitusten tuominen näkymään katsottiin tarpeelliseksi, samoin kuin intralinkkien ja ylläpitäjien luomien linkkien. Kyselyssä ehdotettuun hakutoimintoon haluttiin myös liittää hakuosioon jo kaavailtu hakuhistoria ja suosikkien haku.

IMS:n käytölle toivottiin pientä ja alustavaa ohjeistusta. Käytön jatkamisen helpottamiseksi toivottiin myös listausta omista viimeksi muokatuista sisällöistä. Myös lähestyvien katselmointien ja aikataulujen kirjanpidon osalta toivottiin helpotusta kalenterin muodossa, jollaista IMS-sovelluksessa ei vielä ole.

Kyselyä tehtäessä ei vielä haluttu ottaa kantaa peruskäyttäjän ja ylläpitäjän eriäviin oikeuksiin, mutta näiden osalta tuli paljon palautetta siltikin. Näiden kahden käyttäjän näkymät haluttiin erottaa, sillä käyttökohteet ja – tarkoitukset eroavat hyvin paljon toisistaan. Varsinkin kaavioiden kohdalla katsottiin, että vain ”valitut palat” voitaisiin näyttää peruskäyttäjille, kun taas muut kaaviot hyödyntäisivät lähinnä vain ylläpitäjiä.

Myös hyvin toivottu ”älä näytä ohjausnäkyä”-painike jakoi mielipiteitä siitä, pitäisikö painike olla käyttäjä- vai järjestelmäkohtainen. Kaikkien asiakkaiden ei välttämättä katsottu haluavan koko ohjausnäkyä käyttöön. Ylläpitäjille toivottiin myös listausta käyttäjistä, jotka ovat olleet jonain ajanjaksona aktiivisena tai jotka ovat kyseisellä hetkellä kirjautuneena IMS-sovellukseen. Kenties kiinnostavin ehdotus oli idea käyttäjien omista ohjausnäkyistä, joita he voisivat muokata tarpeisiinsa sopiviksi.

Uutisten saama suosio yllätti. Sen perusteluina oli esimerkiksi, että aloitusosioista luotaisiin eräänlainen intrasivu. Toiminnallisuuksien kohdalla oltiin oletettu, että suodatus-toiminnallisuus saisi kannatusta, mutta haku jäisi vähempään kannatukseen sekä sen epämääräisen merkityksen vuoksi että sen vuoksi, että IMS-sovelluksessa jo on erillinen Haku-osio. Kannatus meni kuitenkin juuri toisinpäin.

5.2 Mallintaminen

Näiden määrittelyjen pohjalta lähdettiin kartoituksessa valituin menetelmin luomaan mallinnuksia kustakin ominaisuudesta sekä aloitusosioista kokonaisuutena. Mallinnuksia

tehdessä pidettiin mielessä eri laitealustojen vaatimukset ja haasteet, sekä eri lähteistä johdetut suunnitteluohjeistukset.

Mallinnuksia toteutettiin ensin wireframe-tekniikan avulla kuvaten käyttöliittymän rakennetta ja osakokonaisuuksien sijoittelua. Tämän katsottiin olevan tehokas ratkaisu ajankäytön ja vaivan osalta alkuhahmottelun toteuttamiseen. Tämän jälkeen käyttöliittymän yleisilmettä lähdettiin hahmottelemaan mockup-tekniikan avulla. Sen arveltiin olevan hyvä ja nopea keino kuvata haluttua käyttöliittymän ilmettä, sillä se sisälsi samat edut kuin wireframe-tekniikan käyttö, mutta vain tarkemmalla tasolla. Mockup- sekä wireframe-työkaluna käytettiin ilmaista Pencil-sovellusta. Sen katsottiin olevan helppokäyttöisyyden ja käyttömahdollisuuksiensa puolesta hyvä vaihtoehto. Vertailua tehdessä myös huomattiin, että sen käyttö oli sujuvaa ja vaivatonta.

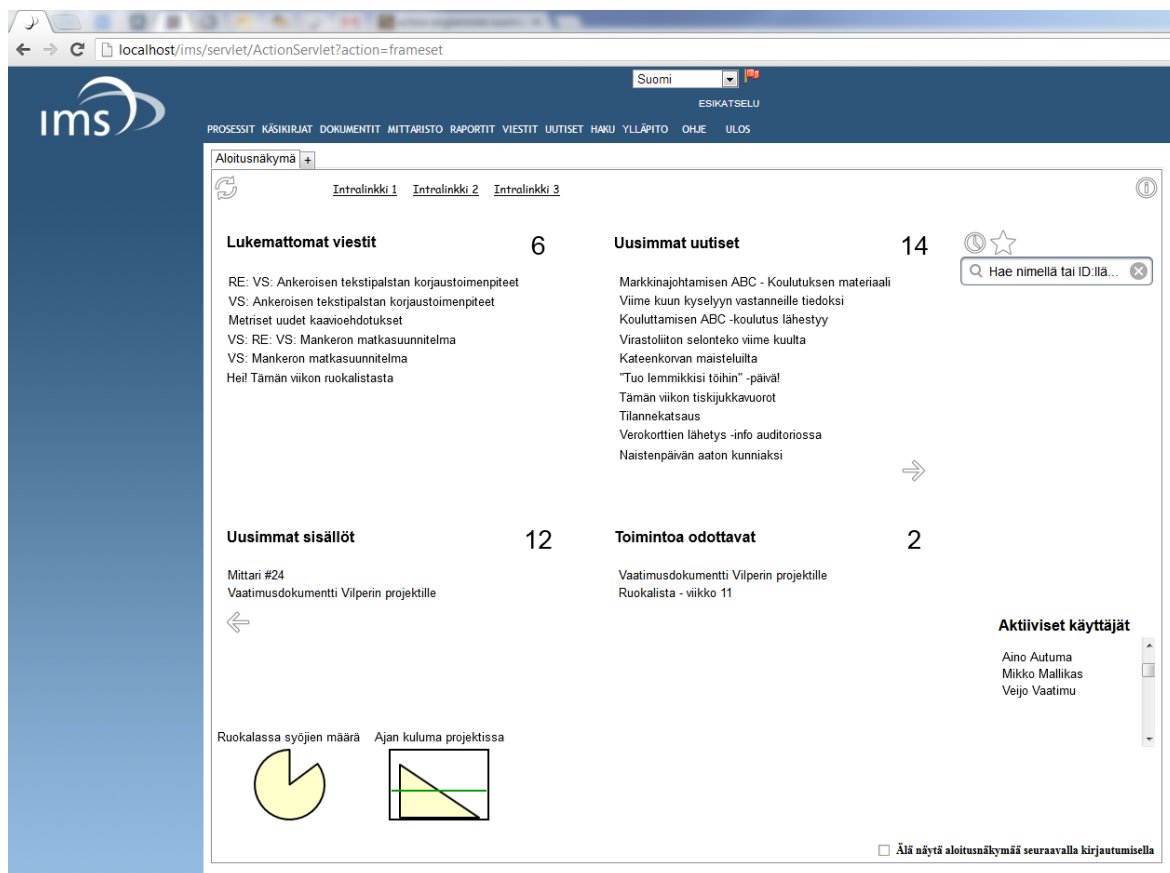
Mallinnusten toiminnallisuutta kuvattiin sanoin ja selityksin kuvakaappauksien ohessa. Tämän katsottiin olevan ajansäästön ja ymmärrettävyyden vuoksi käypä vaihtoehto demoversio-tekniikan sijaan ottaen huomioon, että ymmärrys toiminnallisuudesta kasvoi mallinnuksista keskusteltaessa.

Jokaisesta mallinnuksesta keskusteltiin kolmen yrityksen toimihenkilön kanssa (tuotekehityspäällikkö sekä päivitys- ja kehityshaaroista vastaavat tahot). Keskusteluissa analysoitiin kunkin mallinnuksen hyvät ja huonot ominaisuudet. Analysoinnin jälkeen luotiin uusi mallinnus kustakin ominaisuudesta huomioiden analysoinnin tulokset suunnittelua tehdessä. Tätä jatkettiin, kunnes mallinnuksesta ei enää löydetty huonoja ominaisuuksia tai parannusehdotuksia. Kaiken kaikkiaan tämä vaihe kesti noin kaksi viikkoa.

Ensimmäisen mallintamiskierroksen kohdalla toiminnallisuutta ja elävöittämistä ei vielä mietitty, vaan lähinnä kunkin osakokonaisuuden yleisilmettä ja asettelua. Ensimmäiseen mallintamiseen käytettiin kaksi vaihetta, joista ensimmäinen oli kaikkien osakokonaisuuksien hahmottelu. Siinä kaavailtiin, mitä ylipäätään haluttiin tuoda aloitusosioon ilman niiden varsinaista sijoittelua. (Liite 8.)

Alustavan mallinnuksen pohjalta lähdettiin kehittämään ideaa pidemmälle. Sitä tehtäessä pidettiin mielessä Gestaltin lainalaisuudet, joissa muun muassa läheisyys vaikuttaa

ryhmän hahmottamisessa. Rajoja ei katsottu tarvittavan, ja ne otettiin pois. Painikkeet muutettiin niitä kuvaaviksi ikoneiksi, ja tärkein tieto aseteltiin vasempaan ylänurkkaan sekä siitä keskelle. (Kuvio 3.)



Kuvio 3. Mallinnus 1

Sivu rajattiin "Aloituskäyttöksi", mutta ylävasemmalla välilehtipainikkeesta sai mallissa käyttäjä luotua oman ohjausnäkömäänsä. Ajatuksena oli, että omassa näkömässä eri osakokonaisuuksien paikkoja voitaisiin siirrellä, poistaa näkyvistä tai lisätä näkömään. Huolena tosin oli, että toiminnallisuus olisi liian iso toteuttaa ja jätettäisiin toteuttamatta. Ideana sen katsottiin tuovan käyttäjäystävällisyyttä ja käytettävyyttä. Fontiksi valittiin IMS-sovelluksessa käytetty Trebuchet MS. (Kuvio 3.)

Epätärkeäksi käytettäessä havaittu "Älä näytä aloitusosiota" siirrettiin oikeaan alakulmaan, jossa se olisi eristettynä yhä selkeästi esillä, muttei häiritse käyttöä eikä vaatisi huomiota.

Puutteita mallissa oli, ettei se sellaisenaan toimisi pienempien näyttökokojen kanssa. Koon pienentyessä osakokonaisuuksia ei voida erottaa tyhjin aluein. Myös mittarit on pudotettu tärkeysasteessa hyvin alas, pienentäen niiden kokoa ja asettamalla ne näkymän alalaitaan. Kuitenkin katsottiin, että ne visuaalisuudellaan voivat saada katseita. Piirakkamallia ei saisi suosia ohjausnäkyessä, mutta nämä mittarit voi ylläpitäjä päättää aloitusosioon.

Mallinnuksesta keskusteltaessa päätettiin muun muassa, ettei ”Älä näytä aloitusosiota”-painiketta tarvitse esittää aloitusosiossa, vaan sen voi sijoittaa Ylläpito-osion käyttäjän tietoihin. Omien näkymien teon ei katsottu olevan liian raskas toteuttaa. Ylläpitäjän ja käyttäjän eriävien näkymien osalta päätettiin, että olisi järkevää luoda yksi yhtenevä käyttäjän sekä ylläpitäjän näkemä aloitusnäky. Tätä oletusaloitusnäkyä voisivat ylläpitäjät muokata. Lisäksi jokainen käyttäjä ja ylläpitäjä voisi myös luoda omia näkymiä sillä erolla, että ylläpitäjä voisi myös luoda uusia oletusaloitusnäkyä kaikille näytettäväksi. Tästä voisi olla hyötyä esimerkiksi siinä, että uusille käyttäjille voitaisiin luoda ohjeistusnäkyä.

Sivun päivityksen mahdollistava painike katsottiin turhaksi. Sen sijaan nähtiin ratkaisuna, että sivun voisi asettaa päivittymään tietyn ajanjaksoin. Intralinkkien kohdalla taas keskusteltiin siitä, ovatko ne tarpeellisia näkyessä. Intralinkit esitetään muutenkin jo ylävalikon osio-linkkien yläpuolella, joten sivulla olisi toisintoa, jos ne tuotaisiin esiin. Lisäksi jokaisen linkin alta löytyy puurakenne alilinkkejä, jota ei linkkejä tuotaessa näkyeseen oltu huomioitu. Näistä syistä päädyttiin jättämään intralinkit pois aloitusnäkyestä.

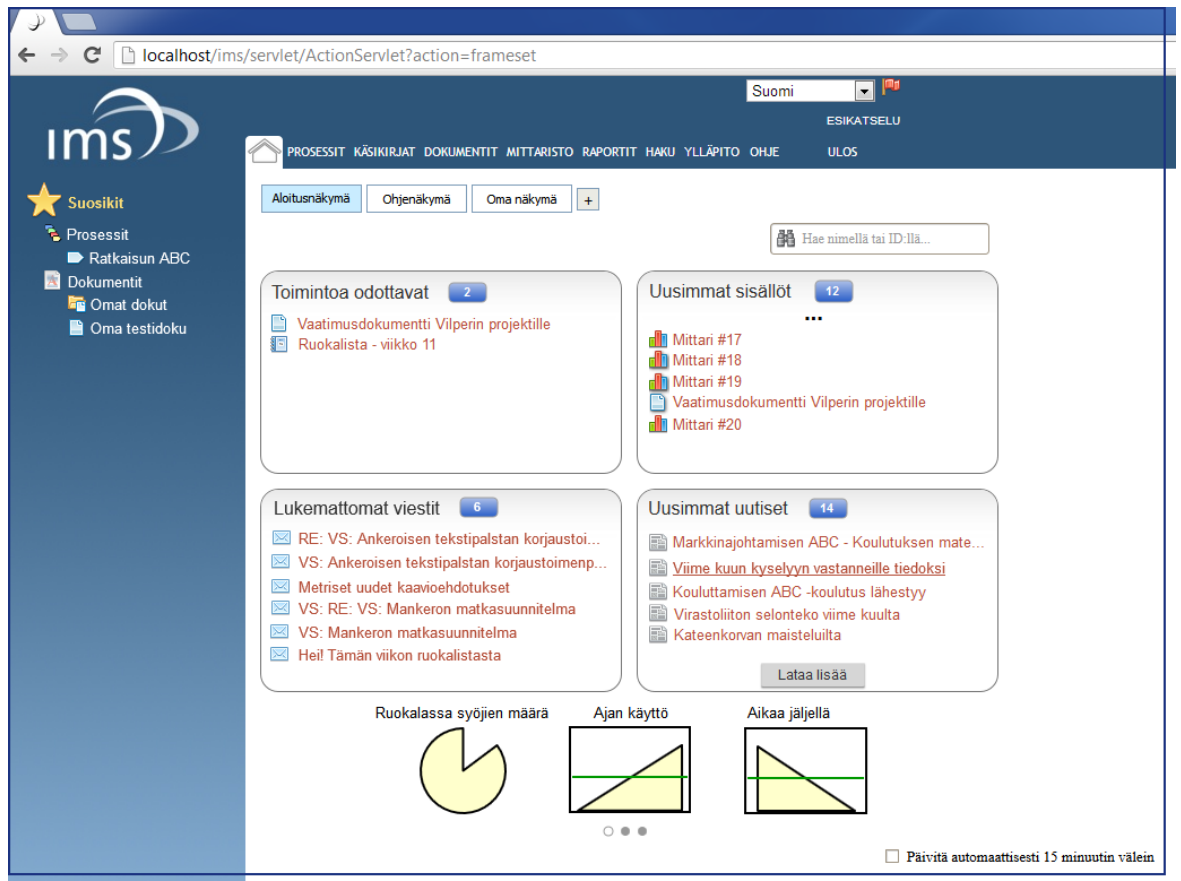
Sivuhistorian erillinen painike haluttiin yhdistää suoraan hakukenttään niin, että kirjoitettaessa hakuhistoria aukeaisi alavetovalikkona. Myös suosikkien erillinen painike katsottiin paremmaksi toteuttaa niin, että suosikit näkyisivät vasemmassa palkissa. Hakutuloksen osalta toiminnallisuus haluttiin niin, että sivu vaihtuisi Haku-osioon, jossa haku varsinaisesti suoritettaisiin, eikä tilaa siksi tarvitsisi jättää hakutulosten esittämiseen.

Tilan osalta nähtiin paremmaksi rajata edes jossain määrin neljää vallitsevaa osakokonaisuutta (viestit, uutiset, sisällöt ja toimintaa odottavat). Niiden lukumäärä haluttiin myös esitettäväksi suoraan nimen yhteyteen. Sivutusidean katsottiin olevan hyvä, mutta haluttiin kokeilla, josko olisi mahdollista vaihtaa se rullaukseksi ja ”Näytä lisää”-linkiksi. Tämä idea toimisi tapauksessa, jossa uutta sisältöä saattaisi tulla vanhaa luettaessa, eikä sivutus silloin toimisi halutulla tavalla (ensimmäisellä sivulla olisi silloin 11 tietoa kymmenen maksimin sijaan). Neljän vallitsevan osakokonaisuuden järjestys tuli myös puheeksi. Sisällöt ja toimintaa odottavat katsottiin itse asiassa tärkeämmiksi kuin uutiset ja viestit.

Aloituskäytävään pääseminen tuli myös puheeksi. Näkymään haluttiin päästä myös muulla keinolla kuin kirjautumalla ulos ja sisään uudelleen. Yläpalkin linkkeihin nähtiinkin vaihtoehtona aloituskäytävän lisääminen prosessit-osion eteen. Ylälinkeistä puhuttaessa myös nähtiin tarpeellisena poistaa Viestit ja Uutiset kokonaan linkeistä, sillä niihin pääsisi suoraan aloituskäytävästä käsin kustakin otsikosta painaen.

Keskustelun pohjalta lähdettiin rakentamaan uutta mallinnusta käyttäen tällä kertaa mockup-tekniikkaa. Mallinnuksessa eri otsikoiden väriä haalennettiin ja eri osakokonaisuudet ympäröitiin rajoin ja tasaisin välimatkoin. Numerointi muutettiin painikkeeksi. Sivutus vaihdettiin "Lataa lisää"-painikkeeksi ja vieritykseksi. Kerralla näytettävien sisältöjen määrää vähennettiin. Eri sisältölinkkeihin lisättiin tyyppikuvakkeet, ja värit muutettiin IMS-sovelluksen tyyliin sopivaksi oranssinsävyiseksi. Mittarien sivutusratkaisu muutettiin painikeriviksi. Mallinnuksessa myös huomioitiin 1024x768 mittasuhteen rajat kuvaamalla ne mallinnukseen sinisellä laatikolla. (Kuvio 4.)

Puutteita saattaa olla, että suosikit-näkymän tähti-ikoni on hyvin vangitsevan voimakassävyinen, samoin kuin numeroinnit. Numerointien nappulatyyli saattaa myös ilmentyä virheellisenä affordanssina.



Kuvio 4. Mallinnuksen 2 alustava hahmotelma

Seuraavaksi testattiin eri rajausvaihtoehtoa, ja suosikit-tähden vahvaa väriä himmennettiin (Liite 8). Reunusten jälkeen lähdettiin avaamaan eri näkymiä. Ensin mallinnettiin näkymä, jonka käyttäjä on luonut itselleen katselutilassa (Liite 8). Sen jälkeen sama näkymä avattiin muokkaustilassa (Liite 8). Näkymää kuvatessa hahmoteltiin tilannetta, jossa käyttäjä on painanut yhden osakokonaisuuden kohdalla muokkauspainiketta.

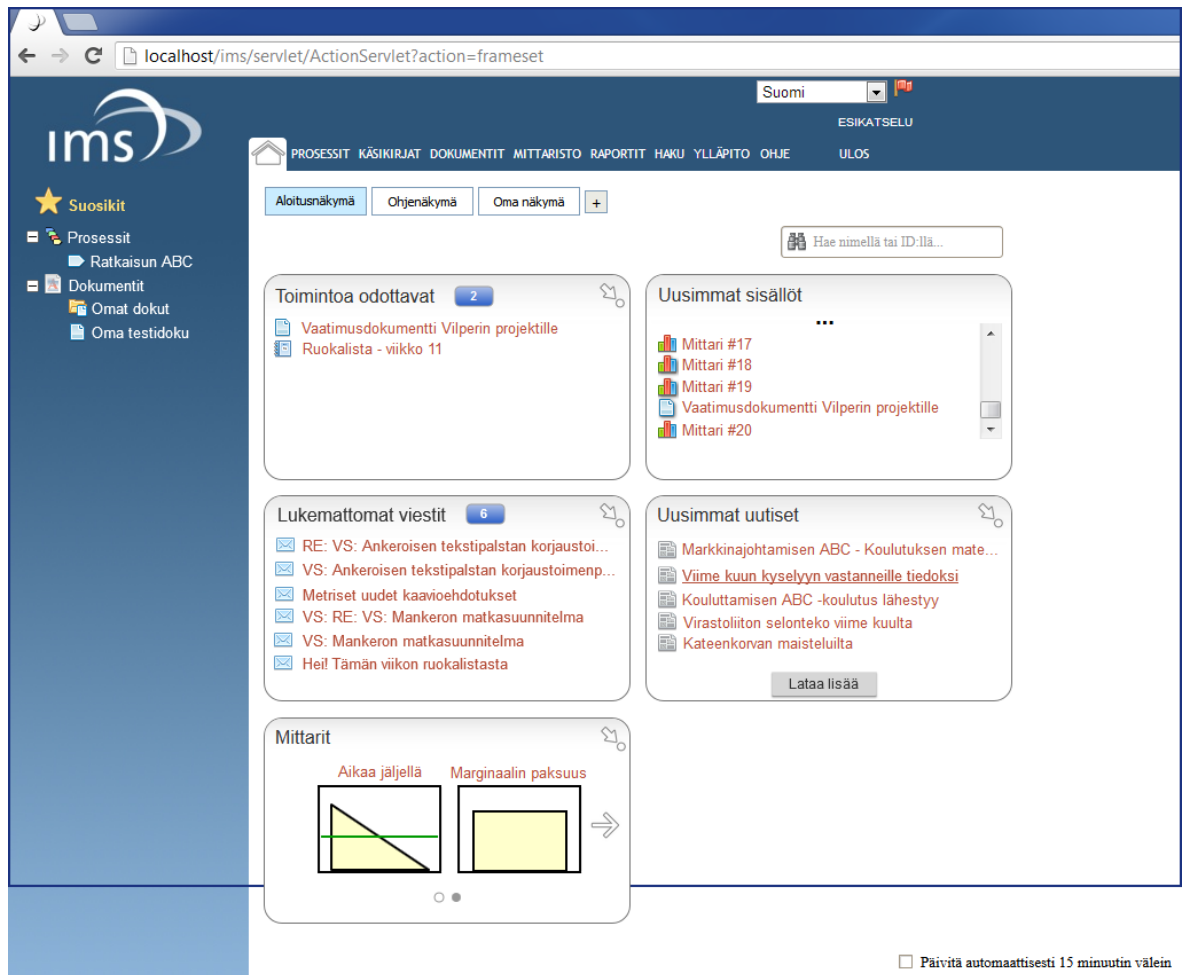
Mallinnuksen 2 näkymistä puhuttaessa katsottiin, että ensimmäinen, rajattu osakokonaisuuksien erottelu on parempi ratkaisu. Sen ajateltiin selkeyttävän kokonaisuuksien hahmottamista varsinkin vierityspalkkien tullessa kuvaan. Sen myös katsottiin toimivan selkeämmin suuren sisältömäärän kanssa, ja pystyvän myös koon pienentyessä pitämään eri kokonaisuudet erillään toisistaan.

Suosikit-listauksen tähden katsottiin olevan parempi pitää alkuperäisen keltaisen väriseinä. Vaihtoehtona ehdotettiin sen koon pienentämistä. Aktiiviset käyttäjät-osakokonaisuuden huomautettiin olevan turha pelkkänä listauksena. Siihen haluttiin

lisää toiminnallisuutta. Oman näkymän mallinnus nähtiin hyvänä ratkaisuna sellaise-
naan, vaikkakin se, että jokainen pystyy muokkaamaan näkymänsä oikeusasetuksia pe-
lätettiin ruuhkauttavaksi. Vaihtoehtona ehdotettiin, että oikeusasetusten muuttaminen
annettaisiin toimintona vain ylläpitäjille. Myös eri osakokonaisuuksien kokoa haluttiin
keskittää, eli tuoda kunkin osakokonaisuuden viemä tila lähemmäs toisiaan. Tämän
katsottiin tuottavan vähemmän ”ilmaa” sivulle.

Näiden kommentointien perusteella lähdettiin luomaan mallinnusta 3. Suosikit -
listaukseen lisättiin pienennysmahdollisuus. Osioihin siirtymisen selventäminen to-
teutettiin lisäämällä kuvake jokaiseen kohtaan, mistä voi siirtyä. Sisällön vierittämistä
varten lisättiin osakokonaisuuksille vierityspalkit. Sisällön määrän tiedot otettiin pois
uusimmista sisällöistä ja uutisista tarpeettomuuden vuoksi. Mittarit-osakokonaisuus
muutettiin vastaamaan paremmin muita kokonaisuuksia kooltaan ja rajaukseltaan.
(Kuvio 5.)

Käyttäjälistaus-osakokonaisuuteen lisättiin toiminnallisuutta. Käyttäjän nimestä pai-
nettaessa tulisi mahdollisuus nähdä käyttäjän yhteystiedot ja listauksesta siirryttäisiin
Ylläpito-osion ”aktiiviset käyttäjät”-taulukoon. Aktiiviset käyttäjät-osakokonaisuus
levennettiin yhtä leveäksi kuin muut osakokonaisuudet. (Liite 8.)



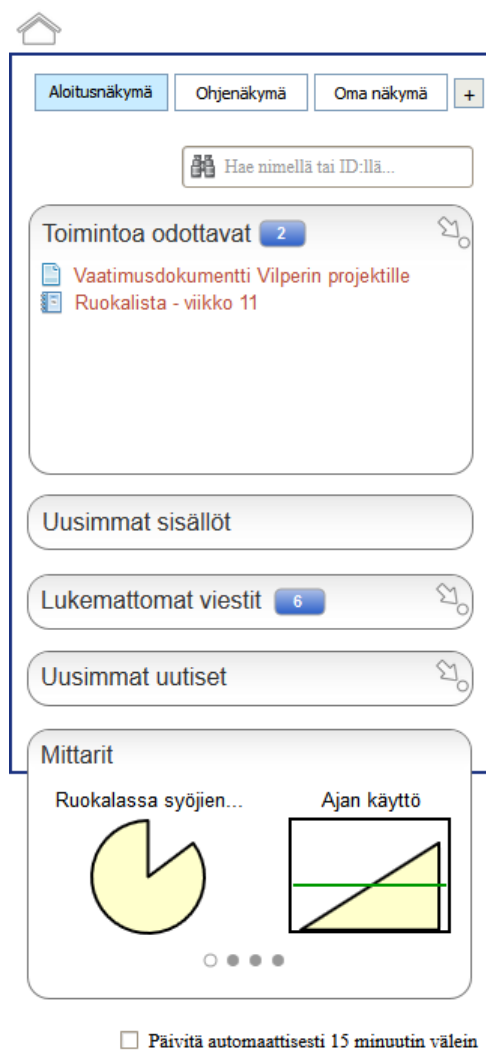
Kuvio 5. Mallinnuksen 3 aloitusnäkö

Viimeisessä mallinnuksessa sisältövaihtoehtolistaukseen vaihdettiin "Linkkejä IMS-sisältöön" yhden linkin sijaan. Oikeuksien muutos-painike päätettiin näyttää vain ylläpitäjille. Uuden osakokonaisuuden lisäspainike muutettiin ikoniksi ja tekstiksi. "Linkkejä IMS-sisältöön"- ja "Mittarivalikoima"-tyylisissä osakokonaisuuksissa sisällön voi itse määrittää. Näiden kohdalla muokkaustilassa näytetäänkin listana, mitkä kohteet on lisätty. Jokaisen kentän kokoa voidaan myös muokata ristikkoratkaisun avulla. Vierekkäiset valintaruudut painettua osakokonaisuus levenee valinnan mukaisesti. (Liite 8.)

Tämän jälkeen lähdettiin kuvaamaan sitä, miten näköala käyttäytyisi sivukokoa pienennettäessä. Malleja tehtäessä pidettiin jatkuvasti mielessä W3C:n (2008) mobiilien käyttööntymien parhaiden käytäntöjen listaus. Mallien sisältämät osakokonaisuudet kaavailtiin kokoon 320x480 pikseliä. Jokainen osakokonaisuus voidaan pienentää vain otsikoksi,

jota painamalla voidaan sisältöä tarkastella. Sama raamitus on käytössä, mutta rivit heittäytyvät allekkain. (Kuvio 6.)

Liitteessä Liite 8 kuvattiin omaa näkymää katselutilassa, kun taas liitteessä Liite 8 hahmoteltiin näkymän muokkaustilaa.



Kuvio 6. Aloituspäätelmä mobiilikooassa

5.3 Käyttöliittymän teko

Mallinnusvaiheen jälkeen mallinnuksien pohjalta luotiin varsinainen aloitusosio. Totuudessa pyrittiin käyttämään valmiita pakettiratkaisuja ja pitämään mielessä jatkokehitys- ja koodin uusiokäyttömahdollisuudet.

Varsinaisen aloitusosion luonnissa käytettiin REST-arkkitehtuuria, sillä sen avulla voitiin suoraan luoda yksi rajapinta mobiili- sekä web-alustalle. SOAPia ei aloitusosion käyttöliittymän teossa käytetty, sillä sitä ei katsottu tarpeelliseksi ottaen huomioon, ettei toteutuksessa käytetty tai tehty varsinaista palvelinpuolta.

Käyttöliittymän pohja luotiin XHTML-kielellä, mutta valiten mime tyyppiä kuitenkin text/html-kielen. Tämän katsottiin olevan toimiva ratkaisu ottaen huomioon, että selaintuen tulee kattaa myös vanhemmat selaimet sekä Internet Explorer. Jotta tulevaisuuden tukimuutoksiin varauduttaisiin, toteutettiin sivu kuitenkin XHTML:n mukaisesti.

Pohjan teon apuna käytettiin Handlebars-mallinnusta. Mallinnuskielen katsottiin vähentävän tarpeetonta toistoa sivun toteutuksesta. Se myös tukee erinomaisesti REST-arkkitehtuuria ja uuden sisällön tuontia Ajax-ratkaisun avulla.

Handlebars valittiin eri mallinnuskielivaihtoehdoista sen hyvän dokumentaation ja suorituskyvyn, ominaisuuksien ja selkeyden lisäksi siksi, että IMSin työntekijöistä löytyy sen osaamista, joilta pystyi tarvittaessa saamaan ohjeistusta. Handlebars-templaattien lisäksi kokeiltiin myös JSON Template-ratkaisua, mutta koska sen katsottiin olevan vähemmän sekä ajankohtainen että päivitetty, ei sitä otettu käyttöön.

Toteutukseen käytettiin JavaScriptiä. Sen vaatima tuki on vaadittu IMSiä käytettäessä, joten ongelmia sen käyttöönottoon ei ollut. Koska sen avulla pystytään tekemään monet mallinnusvaiheessa esiin tulleet ominaisuudet, sen käyttöönotto oli selvää. Lisäksi hyödynnettiin sen käyttöä yhdentävää ja helpottavaa jQuery-kirjastoa.

jQuery-kirjasto valittiin muiden JavaScript-kirjastojen sijasta sen tuttuuden vuoksi ja koska IMS-toimintajärjestelmässä jo käytetään kyseistä kirjastoa laajasti. Ajaxia käytettiin myös, tosin vain runkona tiedostosta tietoa haettaessa. Varsinaisia kyselyitä ei lähetetty olemattomaan palvelimeen, vaan tiedot kehitysvaiheessa haettiin Ajax-kyselyillä tiedostoista.

Tiedostot sisälsivät kaikki tarvittavat tiedot, mitä ohjelma joutuisi palvelimelta hakemaan JSON-muotoisena. JSON valittiin XML:n ja YAMLin sijasta sen helppimman Ajax- ja jQuery-integroitavuuden vuoksi. YAMLiä harkittiin myös, mutta koska se oli jo hyvin tuttu keino, haluttiin lähteä syventämään enemmän JSON-kielen taitoa.

Käyttöliittymän sisältämä mittarikuva-karuselliratkaisu toteutettiin kolmannen osapuolen jQueryyn pohjautuvalla jCarousel-lisäosalla. Käyttöliittymää tehtäessä testattiin myös jQueryn EasyRotator- sekä Basic jQuery Slider-lisäosia. JCarousel-ratkaisu kuitenkin valittiin näiden kahden sijasta, koska sen katsottiin olevan helpompi ottaa käyttöön ja muokata omanlaisekseen.

Pienien viestien näyttämiseen ponnahdusikkunoiden tapaan IMSin käytössä oli jo valmiiksi jälleen jQueryyn pohjautuva qTip-lisäosa. Tämä otettiin käyttöön, mutta koska testauksessa löydettiin niin ylitsepääsemättömiä virheitä, jotka johtuivat qTip-lisäosasta, sen käytöstä luovuttiin. Sen sijaan päädyttiin tekemään oma ratkaisu viestien näyttämiseen, joka toimi ilman ongelmia, oli paljon yksinkertaisempi ja vei jopa vähemmän tilaa.

Tyylien tekoon valittiin CSS tukeutuen vain tarvittaessa CSS3:een. CSS3 on uusin versio ja standardi CSS-kielestä. Uusimmat selaimet tukevat sitä, mutta koska käyttöliittymän tuli toimia myös vanhemmilla selainten versioilla, ei sen sisältämiä ominaisuuksia voitu käyttää kokonaisuudessaan. Koska Sass, LESS ja SCSS kaikki kääntyvät osittain CSS3:eksi, myös niiden käytön katsottiin toimivuudeltaan epävarmaksi. Lisäksi ei nähty syytä niiden käyttöön, koska toistoa ei tapahtunut merkittävästi toteutuksen aikana.

Kuvien osalta käytettiin IMSin valmiiksi sisältämiä kuvia, joista suurin osa on aikoinaan hankittu FatCow-palvelusta. Tämän lisäksi käytettiin Icomoon-sivuston tarjontaa, josta saatiin käyttöliittymän tarpeisiin sopivia harmaasävyisiä ja hyvin yksinkertaisia kuvia. Icomoon-sivusto valittiin muista, sillä sen käyttöä oli testattu jo aiemmin toisen yrityksen työntekijän toimesta ja se oli huomattu tuolloin luotettavaksi sivustoksi.

Lopullisen tuotoksen näkymäkuvat on esitetty liitteessä Liite 9.

5.4 Testaus ja validointi

Aloitusosion toteutuksen jälkeen tehtiin testaus suunnitelma, jonka mukaisesti tehtyä käyttöliittymää lähdettiin testaamaan. Löydetyt ongelmakohdat korjattiin ja koodi refaktoitiin.

Käyttöliittymän toiminnallisuuden testaus toteutettiin käyttäen Selenium-työkalua. Se valittiin käyttöön sen jonkinasteisen tuttuuden ja tunnettavuuden vuoksi. Koska testaukseen ei ollut ajateltu käytettävän kahta viikkoa enempää, ei haluttu lähteä kokeilemaan toimivuuksiltaan epävarmoja testauskeinoja. Seleniumista käytettiin sen Firefox-selaimen lisäosamuotoa. Selenium-testejä tehtiin seuraavista käyttötapauksista:

- Näkymää voidaan katsoa käyttäjäoikeuksin.
- Näkymää voidaan katsoa ylläpitäjäoikeuksin.
- ”Linkkejä IMS-sisältöön”-osakokonaisuus toimii.
- ”Toimintoa odottavat”-osakokonaisuus toimii.
- ”Mittarivalikoima”-osakokonaisuus toimii.
- ”Hakukenttä”-osakokonaisuus toimii.
- Näkymää voidaan vaihtaa.
- Näkymän muokkaus voidaan avata.
- Näkymän muokkauksen voidaan peruuttaa.
- Uusi näkymä voidaan luoda.
- Osakokonaisuudet voidaan uudelleenjärjestää.
- Osakokonaisuuden tyyppiä voidaan muuttaa.
- Osakokonaisuus voidaan poistaa.
- Uusi osakokonaisuus voidaan lisätä näkymään.
- Osakokonaisuuden sisältörivi voidaan poistaa.
- Näkymän nimeä voidaan muuttaa.

Tietyt toiminnallisuudet ja niiden toteutus jätettiin jatkokehitykseen, sillä niiden liittäminen onnistuu vasta, kun palvelinpuolen ratkaisu käyttöliittymälle toteutetaan. Näihin kuuluu

- näkymän oikeuksien muuttaminen
- osakokonaisuuden sisältörivien lisäys
- näkymän muokkaus käyttäjäoikeuksin
- suosikit-listauksen toimivuus
- näkymän tallentaminen
- näkymän poistaminen.

Käyttöliittymän ulkoasun testaus eri selainten osalta päätettiin ensin suorittaa Adobe Browserlab-sovelluksella, joka olisi kattanut kaikki vaadittavat selaimet. Tämä sovellus kuitenkin suljettiin 13. maaliskuuta 2013. Tämän seurauksena vaihdettiin käyttämään Internet Explorer- ja Mozilla Firefox 3.0+-selainten testauksen osalta SuperPreview-työkalua, sekä uusimpien Mozilla Firefox-, Safari ja Chrome-selainten versioiden osalta Spoon-sovellusta. Molempien käyttö havaittiin niitä arvioidessa helpolta ja vaivattomalta, joskin testauksen aikana ilmeni jonkin verran hankaluuksia niiden käytössä. Muun muassa SuperPreview-ohjelman kanssa joutui taistelemaan, että se suostui testaamaan vanhempia Mozilla Firefox-selaimen versioita.

Tuotoksen teon aikana sekä lopuksi validoinnissa käytettiin W3C-validaattoria sen tuttuuden ja tunnettavuuden ja siten sen luotettavuuden sekä käytön helppouden vuoksi. Sen validoitavien kielten valikoima on myös laaja. Sillä ei tosin voitu validoida CSS3- tai JavaScript-kieliä, mutta niiden validointi voitiin suorittaa käyttöliittymän ulkoasun testauksen yhteydessä. JSONin validointi hoidettiin JSONLint-palvelulla, jonka käytön katsottiin olevan vaivatonta ja luotettavaa.

XHTML:än osalta sivun validointi onnistui ilman ongelmakohtia, mutta sivun CSS ei validoitunut täydellisesti. Tämä johtui siitä, että sivulla käytettiin osittain CSS3:ea. Lisäksi käytettiin sivun kopioinnin estävää user-select-attribuuttia, joka ei kuulu W3C:n spesifikaatioihin (Mozilla Developer Network 2013). User-selectin osalta koodi jouduttiin testaamaan toiminnallisuutta testatessa, ja se osoittautui toimivan.

Refaktoroitaessa eli tuotoksen koodia paranneltaessa käytiin läpi jokainen käyttötapaus ja siihen liittyvät koodirivit ja metodit. Koodin jälkikäytön mahdollistaminen, turhan

koodin siistiminen ja suurten metodien jako osiin olivat suurimmat työvaiheet refaktorointiin.

Jokainen metodi kommentoitiin toimintoa selittävin lausein. Samalla varmistettiin, että jos virhetapauksia syntyy myöhemmin, ne pystytään jäljittämään lokituksen avulla. Lokitus toteutettiin tekemällä konsoliloki-merkintöjä, jotka näkyvät, jos käyttöliittymää käyttää lokimerkintöjä tukevalla selaimella. Lokimerkinnät tehtiin englanninkielisiksi ja selkeiksi, ymmärrettäviksi lauseiksi.

6 Pohdinta

Kun tämän työn tekeminen alkoi, ei ollut tiedossa muuta kuin että jonkinlainen osiot yhdistävä osio tulisi toteuttaa IMS-sovellukseen. Se, mitä tuo osio pitäisi sisällään, minkälainen rakenne sillä olisi, mitä kaikkea sillä olisi tarkoitus tehdä tai että minkä oikeudelliset käyttäjät sitä käyttäisivät, olivat asioita, joista kukaan ei ollut vielä tehnyt päätöstä. Ideana oli kuitenkin jo alusta alkaen toteuttaa ohjausnäkyä.

Myös se, miten työ rajattaisiin, oli auki vielä siihen asti, kun aloituskokous pidettiin projektille. Se oli kuitenkin päätetty jo alkuun, että ajatuksena olisi toteuttaa vain käyttöliittymäpuoli. Lopulta projektille lisättiin myös mobiilituki ikään kuin jälkiajatuksena.

Työ aloitettiin, jotta IMS-sovelluksen käyttäjien pitkään pyytämä ominaisuus saataisiin viimein toteutettua. Ominaisuutta varten käyttöliittymäsuunnitteluun tuli jättää runsaasti aikaa, sillä jotta ohjausnäkyä tekisi tehtävänsä, sen tulee esittää ne tiedot ja tietokokonaisuudet, mitä käyttäjän tarvitsee nähdä kaikesta mahdollisesta olemassa olevasta tiedosta erotettuna.

Koska näytettävää tietoa oli niin paljon, sitä oli pakko sekä karsia että ryhmitellä. Ylipäätään se, mitä ohjausnäkyssä näytetään, muodostui tärkeäksi kysymykseksi, joka tämän työn aikana ratkaistiin.

Hankalaksi nähtiin alusta lähtien se, että ohjausnäkyä liitettäisiin toimintajärjestelmään. Molemmat ovat kokonaisuuksia, jotka keskittyvät yrityksen tilanteen kuvaamiseen ja toiminnan tehostamiseen. Se, että ohjausnäkyä periaatteessa tiivistäisi entisestään kaiken sen, minkä toimintajärjestelmä kokonaisuutena jo kertoo, oli hyvin haasteellista. Tärkeäksi nähdystä tiedosta täytyi löytää kokonaisuuksia, mitkä olisivat tärkeämpiä kuin muut kokonaisuudet.

Projekti aloitettiin hahmottelemalla työtä projektisuunnitelman, vaatimusmäärittelyn, käyttäjäkyselyn sekä opinnäytetyöraportin tietoperustan teon avulla. Kun työ ja siihen kuuluvat osat olivat selkeytetty, aloitettiin käyttöliittymän hahmottelu neljän mallinnuskierroksen muodossa. Opinnäytetyöraporttia päivitettiin jatkuvasti työtä tehdessä. Mal-

linnsvaiheen jälkeen ruvettiin toteuttamaan varsinaista käyttöliittymää, jonka valmistuttua tehtiin testaussuunnitelma. Sen jälkeen tehty käyttöliittymä testattiin, validoitiin ja refaktoroiitiin. Nämä vaiheet on esitetty liitteessä Liite 4.

Liitteessä Liite 5 on esitetty työvaiheet, mitä projektissa läpikäytiin. Toteutumaseurannasta huomaa, että muutamassa kohdin aikataulun riittämättömyyden ja erinäisten pois-saolojen vuoksi aikataulu on muutamassa sprintissä lähtenyt heittämään seuraavalle sprintille. Tästä ei kuitenkaan ilmennyt suurempaa ongelmaa, vaan kaikki tehtävät saatiin suuremmista ongelmista toteutettua.

Riskien osalta projektin aikana toteutui riski aikataulun venymisestä. Varmistamistoimenpiteinä aikataulua tehostettiin ja seuranta tehostettiin entisestään. Projektin lisättiin kolmen viikonlopun osalta täydet päivät, että aikataulusta saatiin jälleen kiinni. Muita kartoitettuja riskejä ei toteutunut.

Työtä aloitettaessa oli varattu projektin budjettiin tilaa erinäisiä kirjahankintoja varten, mutta budjettia ei käytetty. Sen sijaan käytettiin mahdollisuutta tilata kirjastoon uutta aineistoa, jonka johdosta tarvittavat hankinnat saatiin.

Liitteessä Liite 6 on esitetty kuvaajina projektin kulku aikataulun osalta. Työtä tehtiin maanantaisin ja keskiviikkoisin kahdeksan ja perjantaisin kymmenen tuntia, sekä muutamana viikonlopun osalta lauantaisin ja sunnuntaisin kahdeksan tuntia. Suunniteltu tuntimäärä oli 365 tuntia, joka toteutui tarkalleen.

Projektin tekoon käytettiin työpaikan tarjoamaa Windows 7-käyttöjärjestelmäistä Dell Optiplex 990-pöytäkonetta sekä etätöiden osalta Windows 7- käyttöjärjestelmällä varustettua Dell Latitude E6500 kannettavaa. Versionhallintaan käytettiin Git-versionhallintaohjelmaa ja sovelluksen tekoon IntelliJ IDEA-sovellusta, mallinnuksen osalta Pencil-sovellusta. Projektin hallinnollisiin toimenpiteisiin käytettiin IMS-sovellusta sekä Microsoft Office-työkaluja.

6.1 Keskeiset tulokset ja hyödynnettävyys

Tuloksena syntyi toimiva ja käyttäjäystävällinen käyttöliittymä, joka yksinkertaisuudeltaan ja toisaalta toimintojen kattavuudeltaan vastaa asetettuja tavoitteita. Käyttöliittymän avulla IMSin käyttäjät pystyvät saamaan käyttöoikeuksiensa rajoissa näkyviin muuttuneen sisällön ja tapahtumat helposti sisäistettävässä muodossa. Näkymä on selailtavissa kaikilla IMSin tukemilla selaimilla, sekä lisäksi Safari-selaimella, jota ei projektisuunnitelmassa huomioitu ottaa mukaan.

Projektia aloitettaessa ajateltiin, että käyttöliittymän liittäminen IMSiin jätettäisiin jatkokehitykseen. Näkymä päätettiin kuitenkin toteuttaa niin, että se oli jo valmiiksi IMSissä kiinni, sillä toteutuksen aikana se katsottiin helpommaksi ratkaisuksi kuin näkymän tekeminen erillään. Tämä helpottaa jatkokehityksen aloittamista huomattavasti. Kokonaisuus tehtiin kuitenkin helposti erotettavaan muotoon niin, että tarvittaessa se voidaan eriyttää IMSistä vähäisellä vaivalla.

Koska näkymä tehtiin suoraan osaksi IMSiä, sovelluksen valmiita osia pystyttiin käyttämään ja monikielisyys pystyttiin alustamaan. Näkymä toteutettiin kuitenkin vain suomenkielisenä. Näkymän toimivuus pystyttiin myös varmistamaan, eikä siten mahdollisia ongelmatilanteita näkymän IMSiin liiton yhteydessä pääse syntymään.

Käytettävyyso ongelmia, mitä ohjausnäky mällä haluttiin poistaa, oli kaksi. Ensimmäinen oli se, että osiot olivat niin hajanaisia, ettei käyttäjä näe kaikkea käymättä jokaisessa osiossa erikseen. Toinen oli se, etteivät sekä yrityksen johto että työntekijät ole tietoisia kunkin hetkisestä yrityksen tilanteesta. Tehty ohjausnäky män käyttöliittymä ratkaisee nämä molemmat. Jokainen osio saadaan edustetuksi käyttöliittymään osana kokonaisuutta. Lisäksi yrityksen tilan saa näky mästä selville. Sekin, että näky miä voidaan muokata yrityskohtaisiksi, poistaa sen todennäköisyyden, että tieto olisi liian yleisluonteista.

Työn kartoitusvaiheessa päätettiin, että käytettäisiin kyselylomaketta eri osakokonaisuuksien tarpeen hahmottamiseen. Tämä toimi periaatteessa hyvin, mutta kenties se, että päätettiin yhdistää mielipiteet ja laskea jokaisen osakokonaisuusehdotuksen suosio, ei ollut parhain ratkaisu toteuttaa. Tärkeimmät ja asiasta eniten tietävät mielipiteet jäivät

massan alle. Vaikka heidän enemmän painoa kantavat mielipiteensä sanoivatkin mahdollisesti toista, ei tuo mielipide suuremman ryhmän suosiossa merkinnyt kuin yhtä ääntä kymmenestä.

Jo kyselyn tuloksia analysoitaessa huomioitiin loistava idea siitä, että näkymään tuotaisiin kalenteri, joka tähän mennessä vieläkin puuttuu koko IMS-toimintajärjestelmästä. Tämä kuitenkin jätettiin toteuttamatta pelkästä huolimattomuudesta johtuen.

Mallintamisvaiheeseen kului kenties eniten aikaa toteuttamisesta. Tähän on totuttu jo muiden projektien kohdalla, ja siihen olikin varattu riittävästi aikaa. Kuitenkin muut tehtävät veivät enemmän aikaa kuin oli kuviteltu, joka sakotti mallintamiseen varattua aikaa. Mallinnusten tekeminen eteni hyvää vauhtia, ja kommentteja mallinnuksiin saatiin kiitettävän paljon. Runsaan kommentoinnin ansiosta työn käytettävyys parani huomattavasti jokaisen mallinnuskierroksen jälkeen.

Vaikka mallintaminen eteni nopeaan tahtiin ja siihen oli varattu tarpeeksi aikaa, muiden aiempien tehtävien arvioitua pidemmän keston takia aika ei riittänyt kaikkien mahdollisten käyttöliittymän ratkaisumallien kokeiluun. Jatkokehitystä ajatellen ja myöhempää käyttöliittymän refaktorointia varten koodista tehtiin mahdollisimman muokattava ja helposti lähestyttävä.

Mallintamisen keinoksi valittu mallinnus-analysointisykli toimi erinomaisesti. Palautetta saatiin kiitettävästi joka kierrokselta, tosin aina vähemmässä määrin mitä pidemmälle kehityksessä mentiin, kuten oli odotettu. Siksi lopullisesta mallista ollaankin ylpeitä, vaikka kaikkia ideoita ei ehditty kokeilla.

Mallintamisen työkaluksi valittiin Pencil-sovellus. Sen käyttö oli hyvin sujuvaa, ja sillä pystyi helposti jopa luomaan HTML-pohjaisen mallinnusten esittelysivun kommentti-teksteineen. Tästä toiminnallisuudesta Pencil-tökalua valitessa ei oltu tietoisia. Lisäksi työkalua käyttäessä nähtiin erinomaisena piirteenä se, että omia palikoita pystyi luomaan omista osakokonaisuuksista, jonka avulla sivua pystyi muokkaamaan ja uusia tekemään todella tehokkaasti.

Pohjakuvista tuli työkalulla tehtyinä selkeitä, joskin wireframe-tekniikan käyttö työkalun avulla oli hankalaa – työkalun tarjoamat palikat ja kokonaisuudet olivat liian selkeitä, jotta niitä mielletäisiin vain summittaisiksi hahmotelmiksi eikä suoranaiseksi kuvakaappauksiksi tulevasta ohjelmasta.

Käyttöliittymän varsinaiseen tekoon valittiin HTML, CSS, JavaScript ja jQuery, Ajax sekä Handlebars-templaattit. Näillä valinnoilla onnistuttiin toteuttamaan käyttöliittymä ilman ongelmia, ja testauksessa huomattiin ratkaisun toimivan vaadituilla selaimilla. Mittarilistaus-osakokonaisuutta tehdessä käytettiin hyväksi kolmannen osapuolen tekemää kirjastoa, jossa halutusta sisällöstä voidaan tehdä karusellin tyylinen esitys. Tätä kirjastoa voidaan käyttää myös muita uusia IMSin ominaisuuksia toteutettaessa.

Käyttöliittymän teko sujui ongelmitta. Viimeisen mallinnuskierroksen ominaisuudet saatiin tehtyä valmiiseen ohjausnäkömään pitäen hahmoteltu ilme ennallaan. Toiminnallisuudessa ei jouduttu pahemmin tinkimään, paitsi ominaisuuksissa, joita ei pystytty demoamaan tai toteuttamaan, kuten näkömään poisto, oikeuksien muutokset ja näkömään muokkauksen tallentaminen. Nämäkin toiminnallisuudet saatiin kuitenkin alustettua käyttöliittymään, jossa ne odottavat jatkokehitystä. Ainut ominaisuus, mikä olisi voitu toteuttaa jos aikaa olisi ollut, oli suurempien osakokonaisuuksien teon mahdollistaminen. Tämä jätettiin jatkokehitykseen.

Valmiita, uudelleenkäytettäviä osia on käytetty toteutuksessa. Työssä käytettiin muutamia valmiita tyyliiedostoja ja kuvia, sekä taulukon seeprakuvioinnin tuottavaa jQuery-kirjastoa. Toteutus tehtiin myös niin, että ohjausnäkömälle voidaan luoda myöhemmin uusia eri osakokonaisuuksia, esimerkiksi juuri toteutuksesta pudotettu kalenteriosakokonaisuus.

Ohjausnäkömään käsitteen mukaisesti tehdystä käyttöliittymästä kyllä löytyy mittareita, mutta vain pienenä osana kokonaisuutta. IMS-sovelluksen sisältö on usein vain eri kohteita, joista ei voida suoraan luoda kaavioita. Kuitenkin jokaista näkömää pystytään käyttöliittymässä muokkaamaan halutunlaiseksi. Siten IMSin käyttäjät voivat luoda itselleen parhaan ohjausnäkömään.

Toiminnallisuuden testaus toteutettiin Selenium-työkalulla, kun taas yhtenevän ilmeen testaus eri selaimien välillä toteutettiin kolmea eri tapaa käyttäen. Jo toteutusta suunniteltaessa ja sitäkin aiemmin testaustyökaluja kokeiltaessa ja niitä verratessa huomattiin Adobe Browserlabsin olevan parhain vaihtoehto. Sen avulla selainten erot olisi voitu tarkistaa yhden työkalun avulla. Kuitenkin ennen kuin testaukseen päästiin, kyseinen työkalu poistettiin Adoben toimesta.

Testaus näillä kolmella eri tavalla onnistui odotetun vaikeasti. SuperPreview-työkalun avulla eri Internet Explorer-selaimen versiot pystyi kiitettävän nopeasti käymään läpi ja testaamaan, mutta jotta työkalulla pystyi Firefoxin vanhempia versioita testaamaan, täytyi kyseiset versiot selaimesta ladata ensin koneelle. Vaikka testaus näiden työkalujen kanssa osoittautuikin vaikeaksi, se silti onnistui helpommin kuin kaikkien vaadittavien selainten lataus koneelle.

Käyttöliittymää testatessa virhetapauksia ei syntynyt kuin muutama yhtenevän ilmeen osalta sekä yksi toiminnallisuuden osalta, jotka kaikki korjattiin saman tien. On kuitenkin totuttu, että testauksessa syntyy enemmän virheitä korjattavaksi. Kenties jokin virhetapaus on jäänyt testauksessa huomaamatta, vaikka testit ovatkin kattaneet kaikki löydetyt käyttötarinat.

Validointi toteutettiin W3C- ja JSONLint-validaattoreilla. Molemmat olivat toimiva ja selkeä ratkaisu. Ne myös antoivat selväkielisen ilmoituksen jos jokin kohta validoitavasta sisällöstä ei validoitunut oikein. Siksi virheet pystyttiin helposti korjaamaan.

Tehty koodi dokumentoitiin ja toteutettiin selkeästi niin, että sitä voi helposti seurata. Koodi jäsennettiin eri osiin näkymän muokkauksen ja katselun osilta, sivupohjan ja sivupalikoiden osalta, eri käytettyjen koodikielien osalta sekä eri tapahtumien ja tapahtumista seuraavien toimintojen osalta. Tehty sekä projektin apuna käytetty koodi kattaa liitteessä Liite 10 esitetyn sisällön. Liitteen kuvioista on poistettu näkyvistä mahdolliset muut projektiin kuulumattomat tiedostot. Varsinaiset luodut tiedostot ovat ”dashboard”-kansion alla.

Koodin mahdollista uudelleenkäyttöä mietittiin muun muassa kuvien valinnoissa sekä JavaScript-toiminnallisuuksien toteuttamisessa. Tyyli tiedostojen osalta koodi jaettiin näkymää varten tehtyihin kokonaisuuksiin sekä uudelleenkäytettäviin osiin. Uudelleenkäyttöä ajateltiin myös siinä, että koodi jaettiin muualla tarpeettoman muokkauksen sekä mahdollisen muualta katselun mahdollistavan koodin jakamiseen (katseluun tarvittavat koodit erillään muokkaukseen tarvittavista koodeista).

Dokumentoinnin osalta työn aikana tehtiin projektisuunnitelma, vaatimusmäärittelydokumentti, testaussuunnitelma, loppu- ja tämä opinnäytetyöraportti, kokousten kutsut, esityslistat ja pöytäkirjat sekä useita edistymisraportteja. Reilusti eniten ja suunniteltua enemmän aikaa meni kuitenkin opinnäytetyöraportin tekemiseen ja hiomiseen.

Vaatimusmäärittely läpikäytiin asiakasedustajan kanssa ja kattaa suurimmat käyttötapa-ukset, mitä näkymä pitää sisällään. Näihin käyttötapauksiin kuuluu seuraavien listojen osat.

Kirjautunut käyttäjä voi

- katsella aloitusosion rajoitettua sisältöä
- vaihtaa näkymästä toiseen
- muokata näkymiä
- luoda uuden näkymän.

Ylläpitäjäoikeudellinen käyttäjä voi

- katsella aloitusosion rajoittamatonta sisältöä
- vaihtaa näkymästä toiseen
- muokata näkymiä
- luoda uuden näkymän.

Liittessä Liite 7 käyttötapaukset on avattu tarkennettuna.

6.2 Luotettavuus

Tietoperustaa tehtäessä käytettiin aikaa lähteen luotettavuuden tutkimiseen. Työn lopuksi valitut lähteet vielä käytiin uudelleen läpi tarkistaen, että alustava luotettavuuden tarkastelu piti edelleen paikkaansa. Lähteiden luotettavuuteen vaikutti lähteen asiaan-kuluvuus, ajantasaisuus sekä sen kirjoittajan asiantietämys ja saavutukset aihepiiristä. Jos lähde ei näiltä osin vastannut haluttua tasoa, etsittiin toinen lähde sen tilalle käytettäväksi. Joidenkin lähteiden kohdalla tekijä oli niin keskeinen asiantuntija, että ajantasaisuudesta tingittiin.

Tärkeimpiin lähteisiin lukeutuu muun muassa Stephen Few, joka esitellään työn aikana kirjailijaksi ja konsultointiyrityksen perustajajäseneksi. Hän on taho, jonka kirjoitukset ohjausnäkökulmasta ja sen käyttötarkoituksesta ovat kyseisellä osa-alalla arvostettuja. Se, missä Few keskittyy enemmän yrityspuoleen tiedonesittämisessä, hänen esikuvansa Edward Tufte, jota lainataan lyhyehkösti työssä, keskittyy enemmän tiedon esteettiseen puoleen. (ExcelCharts 2009.)

Wayne Eckerson, joka myös esiteltiin työn aikana, on myös keskeinen lähde työn eri osa-alueilla. Mallintamisessa ja testauksessa esiin nousee Cameron Chapman, joka on monen vuoden kokemuksen omaava käyttöliittymäsuunnittelija.

6.3 Suositukset ja jatkokehitysehdotukset

Jatkokehitykseen jää palvelin- ja tietokantapuolen ratkaisut, jotka tähän työhön eivät kuuluneet, samoin kuin Suosikit-listauksen varsinainen liittäminen ja oikeuksien muuttamisen toteuttaminen - ylläpitäjäoikeudellisen käyttäjän tulee suunnittelun mukaan pystyä luomaan ohjausnäkyviä kaikkien näkyviin. Myös ohjausnäkökulman poisto jätettiin jatkokehitykseen.

Käyttöliittymän toteutuksessa alustettiin se, että jokin osakokonaisuus voisi olla suurempi kuin vakioleveys ja korkeus antavat myöten. Tämän lopullinen toteutus kuitenkin jätettiin odotetusti jatkokehitykseen ajanpuutteen vuoksi.

Koska rasi- ja palvelintestit ei voitu ilman palvelinta rakentaa uskottavasti, ne jätettiin tekemättä. Suosituksena onkin, että nämä suoritetaan heti, kun tietokanta- ja palvelinpuolen ratkaisut ovat tehty. Tällöin nähdään mahdollisimman aikaisin osion kehitystä, tarvitseeko käyttöliittymän tekemistä kyselyjä muuttaa kevyemmiksi.

Monikielistämistä aloitettiin työn aikana, tosin liittämällä vain suomenkieliset käännökset. Työ saatiin ajateltua helpommin liitettyä IMS-sovellukseen kiinni, jonka vuoksi tämä oli mahdollista. Väliaikaisen testisyysolosuhteiden suhteen monikielisyttä ei kuitenkaan mietitty.

Työn aikana hahmoteltiin ja lisättiin näkymään yhteensä yhdeksän erilaista osakokonaisuustyyppiä, joita voidaan liittää ohjausnäkökulmaan käyttäjien toimesta. Näitä osakokonaisuuksia voidaan myöhemmin lisätä, kun löydetään uusia käyttötarkoituksia ja -tarpeita. Myöhemmin toteutettaviin kuuluu muun muassa kalenteriosakokonaisuus, jossa lähestyvät katselmointipäivämäärät voitaisiin nähdä.

”Aktiiviset käyttäjät”-osakokonaisuuden nimistä painaessa aukenevat kyseisen käyttäjän tiedot tarkemmin. Jatkokehitykseen jää pohdinta, muutetaanko tämän toimintaa niin, että nimestä painaessa aukeaisi mahdollisuus lähettää käyttäjälle viesti.

Vaikka tietoperustaa ja mallinnuksia tehdessä huomioitiin tarkkaan eri osakokonaisuuksien sijoittelu niin, että ne olisivat mahdollisimman esillä tärkeysjärjestykseen nähden, ei lopullisen käyttöliittymän tekovaiheessa tätä voitu enää juuri huomioida. Rakennettaessa pitää lähellä mallinnusta, mutta jotta eri osakokonaisuuksia voitiin toteuttaa, testata, validoida ja esitellä käyttöliittymässä, ne täytyi tuoda näkymään. Tästä syystä käyttöliittymä voi näyttää ruuhkaiselta. Tämä kuitenkin tullaan korjaamaan palvelinpuolen ratkaisua tehtäessä, kun osakokonaisuudet rajataan eri näkymiin.

6.4 Ammatillinen kehittyminen ja oppiminen

Työtä tehdessä opittiin, kuinka paljon projektin hallinnollisiin tehtäviin voi kulua aikaa. Työn alussa aikataulu lipsui hieman, mikä aiheutti stressiä. Siihen saatiin kehitettyä ratkaisu lisäpäivien muodossa, ja sen seurauksena aikatauluttamisesta opittiin paljon. Työn

aikana pidetyt kokoukset itsessään olivat arvokas kokemus kokouskäytäntöjen rutinoitumisen osalta. Määrittelyn aloittaminen havaittiin yllättävän vaikeaksi ja aikaa vieväksi, ja tietoperustan hahmottelu toi varmuutta työn käynnistämiseen.

Tätä raporttia kirjoitettaessa opittiin paljon työn aiheesta, eli ohjausnäkömystä ja käyttöliittymäsuunnittelusta. Näistä havainnoista tulee jatkossa olemaan paljon hyötyä, sillä ne avarsivat näkemään eri puolia käyttöliittymäsuunnittelussa ja toivat ymmärrystä siihen, että kannattaa keskittyä olennaiseen - sisältöön. Dokumentoidessa opittiin asalähtöistä ja lähteisiin perustuvaa kirjoittamista. Samalla opittiin jäsentämään tekstiä ja järjestämään asiat niille kuuluviin paikkoihin. Se, miten asiat saadaan liitettyä toisiinsa, oli päälimmäisenä oppimiskokemuksena.

Lähteiden arviointi osoittautui hankalaksi, vaikka siinä lopulta onnistuttiinkin. Liian usein joutui hylkäämään potentiaalisen lähteen vain siitä syystä, ettei artikkelin tai teoksen kirjoittajasta löytynyt lisätietoa. Omakohtaista tietoa oli paljon, mutta siihen ei yksin voinut tukeutua, vaan kaikesta tiedosta täytyi olla lähde.

Dokumentoinnin harjoittelun lisäksi opittiin uusia tekniikoita käyttöliittymän teon kanssa. JSON-kieltä harjoiteltiin ja se sujuu nyt entistä paremmin. Lisäksi harjoiteltiin uutta tekniikkaa, eli Handlebars-templaatteja, joilla sivun osa voidaan kopioida uudelleen. Työssä kokeiltiin myös Less-kieltä, jolla CSS3-kielen kirjoittamista voitaisiin helpottaa, mutta se nähtiin tarpeettomana lisänä toteutukseen. Samalla harjaannuttiin IMS-sovelluksen sisältämien eri valmiiden palikoiden käytössä muun muassa tyyli tiedostojen ja suosikkiosion osalta.

Koko opinnäytetyö prosessina oli hyvin avartava ja arvokas kokemus. Tietoperustaa oli paljon eikä läheskään kaikki siitä mahtunut mukaan. Joissain kohdin juurtuneita käsityksiä eri asioista joutui muuttamaan, ja toisaalta joidenkin asioiden kohdalla ymmärrys kasvoi entisestään. Vaikka virheitä sattui, eikä kaikkeen voi kai koskaan olla tyytyväinen, saatiin prosessi vedettyä kunnialla läpi. Tämän työn aikana kertynyt tietotaito tulee auttamaan vielä pitkään, ja sen päälle on hyvä jatkossa kerätä uutta tietoa.

Lähteet

- Bevan, N. 2006. International Standards for HCI. Selonteko. Luettavissa: http://nigelbevan.com/papers/International_standards_HCI.pdf. Luettu: 3.3.2013.
- Chapman, S. 2009. Why JavaScript. Artikkele. Luettavissa: <http://javascript.about.com/od/hintsandtips/a/whyjavascript.htm>. Luettu: 22.4.2013.
- Chapman, C. 2010. Ultimate Guide to Website Wireframing. Artikkele. Luettavissa: <http://sixrevisions.com/user-interface/website-wireframing>. Luettu 17.4.2013.
- Chapman, C. 2011. Review of Cross-Browser Testing Tools. Artikkele. Luettavissa: <http://www.smashingmagazine.com/2011/08/07/a-dozen-cross-browser-testing-tools>. Luettu 6.3.2013.
- Chiang, A. 2009. The Art of Dashboarding. Artikkele. Dashboard Insight. Luettavissa: http://www.dashboardinsight.com/articles/digital-dashboards/fundamentals/the_art_of_dashboarding.aspx. Luettu: 26.4.2013.
- Chiang, A. 2011. Data visualization on smart phones. Artikkele. Dashboard Insight. Luettavissa: <http://www.dashboardinsight.com/articles/digital-dashboards/fundamentals/data-visualization-on-smart-phones.aspx>. Luettu: 27.2.2013.
- Cluts, N. 1997. An ASP You Can Grasp: The ABCs of Active Server Pages. Artikkele. Luettavissa: <http://msdn.microsoft.com/en-us/library/ms972317.aspx>. Luettu: 22.4.2013.
- Crockford, D. 2006. JSON: The Fat-Free Alternative to XML. Artikkele. Luettavissa: <http://www.json.org/fatfree.html>. Luettu: 4.3.2013.
- Eckerson, W. 2011a. Performance dashboards. Toinen painos. John Wiley & Sons, Inc. New Jersey, USA.

Eckerson, W. 2011b. How to Design Effective Dashboard Displays. Artikkele. Luettavissa: <http://www.dashboardinsight.com/articles/digital-dashboards/fundamentals/how-to-design-effective-dashboard-displays.aspx?page=1>. Luettu: 26.4.2013.

ExcelCharts 2009. God and Moses? The Differences Between Edward Tufte and Stephen Few. Artikkele. Luettavissa: <http://www.excelcharts.com/blog/god-and-moses-the-differences-between-edward-tufte-and-stephen-few>. Luettu: 17.4.2013.

Few, S. 2004. Dashboard confusion. Artikkele. InformationWeek. Luettavissa: <http://www.informationweek.com/software/business-intelligence/dashboard-confusion/18300136>. Luettu: 27.2.2013.

Few, S. 2005. Information dashboard design. Presentaatio. Perceptual Edge. Luettavissa: <http://blogs.ischool.berkeley.edu/i247s12/files/2012/01/Dashboard-Design-Overview-Presentation.pdf>. Luettu: 3.3.2013.

Few, S. 2009. Now You See It, Simple Visualization Techniques for Quantitative Analysis. Analytics Press. California, USA.

Few, S. 2012. Show Me the Numbers, Designing Tables and Graphs to Enlighten. Toinen painos. Analytics Press. California, USA.

Garrett, J. 2005. Ajax: A New Approach to Web Applications. Artikkele. Luettavissa: <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>. Luettu: 4.3.2013.

Gaver, W. 1991. Technology affordances. Tutkielma. Luettavissa: <http://www.cs.umd.edu/class/fall2003/cmsc828s-0301/p79-gaver.pdf>. Luettu: 3.3.2013.

Graham, J. 2011. SCSS Manual-Edition 0.4.2. Ohjeistus. Luettavissa: <http://www.nongnu.org/scss/scss.html>. Luettu: 22.4.2013.

Healey, C. 2009. Perception in Visualization. Selonteko. Luettavissa:
<http://www.csc.ncsu.edu/faculty/healey/PP>. Luettu: 4.3.2013.

IMS 2013a. Esittely – Ohjelmisto. Luettavissa: <http://www.ims.fi/ohjelmisto/esittely>.
Luettu: 1.3.2013.

IMS 2013b. Asiakkaat. Luettavissa: <http://www.ims.fi/asiakkaat>. Luettu: 1.3.2013.

Mozilla Developer Network 2012. JavaScript technologies overview. Artikkel. Luettavissa: https://developer.mozilla.org/en-US/docs/JavaScript/JavaScript_technologies_overview?redirectlocale=en-US&redirectslug=JavaScript_technologies_overview. Luettu: 4.3.2013.

Mozilla Developer Network 2013. User-select. Ohjeistus. Luettavissa:
<https://developer.mozilla.org/en/docs/CSS/user-select>. Luettu: 17.4.2013.

Optimal Solutions 2011. Usability Key to Dashboard Design. Artikkel. Luettavissa:
<http://www.optimalsol.com/usability-key-to-dashboard-design/?pp=1808>. Luettu: 8.3.2013.

Oracle.com, 2010. JavaServer Pages Overview. Selonteko. Luettavissa:
<http://www.oracle.com/technetwork/java/overview-138580.html>. Luettu: 22.4.2013.

Php.net, 2013. What is PHP? Selonteko. Luettavissa:
<http://www.php.net/manual/en/intro-what-is.php>. Luettu: 22.4.2013.

Sass-lang.com, 2006. Sass. Selonteko. Luettavissa: <http://sass-lang.com>. Luettu: 22.4.2013.

Sellier, A. 2010. LESS. Ohjeistus. Luettavissa:
<https://github.com/cloudhead/less#readme>. Luettu: 22.4.2013.

Stachowiak, M. 2006. Understanding HTML, XML and XHTML. Artikkele. Luettavissa: <https://www.webkit.org/blog/68/understanding-html-xml-and-xhtml>. Luettu: 4.3.2013.

Tilkov, S. 2007. A Brief Introduction to REST. Artikkele. Luettavissa: <http://www.infoq.com/articles/rest-introduction>. Luettu: 22.4.2013.

Travis, D. 2009. User Experience as Business Intelligence. Artikkele. Luettavissa: <http://www.userfocus.co.uk/articles/dashboard.html>. Luettu: 8.3.2013.

Tuck, M. 2010. Gestalt Principles Applied in Design. Artikkele. Luettavissa: http://sixrevisions.com/web_design/gestalt-principles-applied-in-design. Luettu: 4.3.2013.

Tufte, E. 2001. The visual display of Quantitative Information. Toinen painos. Graphics Press LLC. Connecticut, USA.

W3C 2000. Simple Object Access Protocol (SOAP) 1.1. Ohjeistus. Luettavissa: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>. Luettu: 4.3.2013.

W3C 2008. Mobile Web Best Practices. Ohjeistus. Luettavissa: <http://www.w3.org/TR/mobile-bp>. Luettu: 2.3.2013.

Weizenbaum, N. 2009. Sass and Less. Blogikirjoitus. Luettavissa: <http://next.com/posts/83-sass-and-less>. Luettu: 4.3.2013.

Liitteet

Liite 1. Käsitteet

Ajax	<i>(Asynchronous JavaScript And XML)</i> Selainpohjainen ohjelmointikieli, jonka avulla verkkosivu voidaan päivittää ilman, että käyttäjän tulee odottaa palvelimen vastausta.
Aloitusosio	Tässä työssä aloitusosion tarkoitus on selainikkunaa, joka näytetään käyttäjälle ensimmäisenä, kun hän on kirjautunut IMS-toimintajärjestelmään. Voi sisältää monta eri ohjausnäkyä.
Aloitusnäky	Tässä työssä aloitusnäkyllä tarkoitetaan aloitusosion sisältä ohjausnäkyä, mikä kattaa sisälleen sille ominaisia osakokonaisuuksia. Aloitusnäky on vain yksi aloitusosion sisältämisestä mahdollisista ohjausnäkyistä.
Arkkitehtuuri	Tässä työssä arkkitehtuurilla tarkoitetaan suunnittelumallia, minkä mukaisesti sovellus toteutetaan. Eri arkkitehtuureita on useita, ja niiden käyttökohteet vaihtelevat.
ASP	<i>(Active Server Pages)</i> Ohjelmointikieli palvelinpuolen toteuttamiseen.
IMS	<i>(Integrated Management System)</i> IMS Business Solutions Oy:n tuottama toimintajärjestelmä, jolla asiakasyritykset voivat ohjata ja hallinnoida omia yrityksiään.
JavaScript	Ohjelmointikieli, jolla web-sivua voidaan elävöittää. JavaScript rakentuu pohjakieli ECMAScriptistä ja DOM-arkkitehtuurista <i>(Document Object Model)</i> (Mozilla Developer Network 2012.)

JavaScript-kirjasto	Kokoelma JavaScript-pohjaisia funktioita, jotka useimmiten on kasattu yhdeksi paketiksi.
JSON	(<i>JavaScript Object Notation</i>) Tiedontallennuskieli, jonka muotoista tietoa voidaan välittää palvelimelta toiselle tai selaimeen ja takaisin esimerkiksi Ajaxin avulla. Muita tällaisia kieliä ovat muun muassa XML ja YAML. JSON pohjautuu ECMA-262-standardiin, jonka pohjalta on myös luotu JavaScriptin pohjana toimina ECMAScript (Crockford 2006).
JSP	(<i>Java Server Pages</i>) Java-ohjelmointikieleen pohjautuva templaattikieli, jolla palvelimen tieto voidaan liittää HTML-pohjaan.
LESS	Sass- ja SCSS-kieliin pohjautuva tyylinmuokkauskieli (Weizenbaum 2009). Voidaan kääntää komentoriviltä tai erillisellä selaimen lisäosalla. Käännettäessä sitä muodostetaan CSS3:ea. Avoimen lähdekoodin LESS on Apache License 2.0:n alainen.
Ohjausnäkyvä	(engl. <i>Dashboard</i>) Näkyvä, josta käyttäjä voi suoraan yhdellä silmäyksellä nähdä yrityksen senhetkisen tilanteen. Sisältää eri osakokonaisuuksia.
Osakokonaisuus	Tässä työssä osakokonaisuudella tarkoitetaan suuremman kokonaisuuden sisältämää yksittäistä, pienempää kokonaisuutta. Esimerkiksi ohjausnäkyvän sisällä olevaa Käyttäjälisäosausta voidaan pitää osakokonaisuutena. Koko IMSin sisällä yksittäinen osio, esimerkiksi Mittaristo-osio, on osakokonaisuus.

Osio	Tässä työssä osiolla tarkoitetaan IMS-sovelluksen kutakin erillistä osakokonaisuutta, joita ovat muun muassa Raportit ja Dokumentit.
PhP	(<i>PHP: Hypertext Preprocessor</i>) Ohjelmointikieli, jolla palvelinpuoli voidaan toteuttaa. Kieli on hyvin yleisesti käytössä yksinkertaisten verkkosivujen teossa.
Protoscript	Ohjelmointikieli, joka toimii samalla tavalla kuin Ajax, mutta jonka rakenne on yksinkertaistettu ja jonka avulla sivua voidaan elävöittää myös hyvin vähäisellä ohjelmointitaidolla.
Prototyyppi	Esiversio suunnitellusta ohjelmasta.
Refaktorointi	Koodin korjausta ja uudelleenkirjoittamista. Siinä sisäinen koodi kirjoitetaan selkeämpään muotoon ilman, että toiminnallisuutta muutetaan.
REST	(<i>Representational State Transfer</i>) HTTP-protokollaan perustuva web-arkkitehtuurityyppi, jossa palvelimen ja selaimen välinen keskustelu hoidetaan yleensä HTTP-protokollalla käyttäen apuna URL-kutsuja. Siinä selain huolehtii tiedon näyttämisen lisäksi kunkin session tilasta. Kullakin asiakokonaisuudella jota sivustolla esiintyy, on oma nimensä.
Sass	CSS3-kieleen pohjautuva tyylinmuokkauskieli. Voidaan kääntää komentoriviltä tai erillisellä selaimen lisäosalla. Käännettäessä siitä muodostetaan CSS3:ea. Sass on lisensoitu MIT-lisenssillä.
SCSS	Sass-kieleen pohjautuva tyylinmuokkauskieli. Voidaan kääntää komentoriviltä tai erillisellä selaimen lisäosalla. Käännettäessä siitä muodostetaan CSS3:ea. SCSS on lisensoitu MIT-lisenssillä.

SOAP	<i>(Simple Object Access Protocol)</i> Arkkitehtuurityyli, joka mahdollistaa palvelinten välisen kommunikaation HTTP-protokollalla. XML-koodiin pohjautuva SOAP-protokolla käsittää kolme osaa: käytettävän arkkitehtuurin, koodisäännösten ja kutsujen ja vastausten käytännöt (W3C 2000).
Validointi	Koodin oikeaoppisuuden muodon tarkistamista. Validoimalla voidaan varmistaa, että koodi rakenteellisesti on oikeaa.
W3C	<i>(World Wide Web Consortium)</i> Maailmanlaajuinen yhteisö, joka tarjoaa vapaita standardeja tavoitteenaan internetin pitkän tähtäimen kasvu.
Wireframe-tekniikka	Ohjelman tai sovelluksen karkea kuvaaminen näyttökuvaksi.
XHTML	HTML-ohjelmointikieli tiukemmin muotoilusäännöin. Se, missä tavallinen HTML pohjautuu SGML:ään (<i>Standard Generalized Markup Language</i>), XHTML pohjautuu XML:ään (<i>Extensible Markup Language</i>) (Stachowiak 2006).
XML	<i>(Extensible Markup Language)</i> Tiedontallennuskieli, jonka muotoista tietoa voidaan välittää palvelimelta toiselle tai selaimen ja takaisin esimerkiksi Ajaxin avulla. Muita tällaisia kieliä ovat muun muassa XML ja JSON.
YAML	<i>(YAML Ain't Markup Language)</i> Tiedontallennuskieli, jonka muotoista tietoa voidaan välittää palvelimelta toiselle tai selaimen ja takaisin esimerkiksi Ajaxin avulla. Muita tällaisia kieliä ovat muun muassa XML ja JSON.

Liite 2. Mallinnustyökalujen vertailu

Työkalu	Dokumentaatio	Käytön helppous	Ominaisuuksien laajuus	Lisenssi / maksullisuus
ForeUI	Laaja	Helppo	Laaja	Maksullinen
Balsamiq	Keskinkertainen	Helppo	Laaja, toimii myös applikaationa	Maksullinen, 7 päivän ilmainen kokeilu
Mockups				
App Scetcher	Vähäinen, ellei olematon	Helppo	Laaja, luo toimivan prototyypin, ei vain kuvaa	Maksullinen, 14 päivän ilmainen kokeilu
FlairBuilder	Laaja, myös videotutoriaaleja	Helppo	Laaja	Maksullinen, 15 päivän ilmainen kokeilu
Wireframe Scetcher	Laaja, myös videotutoriaaleja	Helppo	Laaja, toimii myös monen selaimen lisäosana	Maksullinen, 14 päivän ilmainen kokeilu
MockingBird	Vähäinen	Helppo	Keskinkertainen	Maksullinen /maksuton (raja 1 projekti, 10 sivua, 2 käyttäjää)
Mockup Builder	Laaja, myös videotutoriaaleja	Helppo, toimii myös suoraan selaimelta	Keskinkertainen	Maksuton
Pencil	Laaja	Helppo, toimii myös selaimen lisäosana	Laaja, omat pakettinsa myös mobiilien alustojen suunnitteluun	Maksuton, GNU GPL V2
DENIM	Vähäinen	Vaatii piirto-pöydän	Vähäinen	Maksuton

Liite 3. Aloitusnäkyvän sisällön kysely

Aloituskäytännön sisällön kysely	
<p>Aloituskäytännön näkymä, joka tulee tulemaan heti kirjautumisen jälkeen näkyviin. Toimii kuin dashboard, eli näyttää ohjelman tilan kullakin hetkellä ja antaa kokonaisvaltaisen kuvan siitä, mitä tapahtuu. Se, mitä osio/näkymä näyttää tai tulee pitämään sisällään, on työn alla. Kyselyllä koitetaan ratkaista, mitä osioon ylipäätään "voisi" tulla. Toteuttamisen hankaluutta tms. ei vielä lähdetä miettimään.</p>	
Kirjoita vain X laatikoihin, jotka ideana voisivat olla hyviä.	
Mitä aloitusosiossa halutaan näkyvän?	
<input type="checkbox"/>	Viestit
<input type="checkbox"/>	Uusimmat
<input type="checkbox"/>	Viime kirjautumisen jälkeiset
<input type="checkbox"/>	Ei-luetut
<input type="checkbox"/>	Sisällöstä tietoa
<input type="checkbox"/>	Uusimmat
<input type="checkbox"/>	Viime kirjautumisen jälkeiset
<input type="checkbox"/>	Itselle kohdistetut
<input type="checkbox"/>	Toimintaa odottavat
<input type="checkbox"/>	Kaavioita käytöstä
<input type="checkbox"/>	Mittaristo-osion "Valitut palat"
<input type="checkbox"/>	"Osioiden käyttö" -piirakka (esim. Sisällön määrä osioittain)
<input type="checkbox"/>	Käyttäjien määrän muutokset
<input type="checkbox"/>	Uuden sisällön määrä
<input type="checkbox"/>	Toimintaa odottavan sisällön määrä
<input type="checkbox"/>	Käyttöaste (sisällön teko vs. Katselija)
<input type="checkbox"/>	Käytetty selain
<input type="checkbox"/>	Käyttöoikeuksien suhde (ylläpitäjiä, katsojia, muokkaajia)
<input type="checkbox"/>	Lähetettyjen bugi-ilmoitusten määrä
<input type="checkbox"/>	Muuta, mitä?
<div style="border: 1px solid black; height: 60px; width: 100%;"></div>	
Mitä aloitusosion halutaan pystyvän tekevän?	
<input type="checkbox"/>	Hakutoiminnallisuus
<input type="checkbox"/>	Siirtyminen osioon /sisältöön (launch pad)
<input type="checkbox"/>	Filter-toiminnallisuus
<input type="checkbox"/>	Automaattitoiminnot (esim. Jos jokin sisältö ei ajan tasalla, lähetä huomautus)
<input type="checkbox"/>	"Älä näytä aloitusosiota kirjaututtaessa"-painike
<input type="checkbox"/>	Muuta, mitä?
<div style="border: 1px solid black; height: 100px; width: 100%;"></div>	

Liite 4. Tehtäväluettelo ja ajoitussuunnitelma

Seuraaviin taulukoihin merkitty vihreällä suunnitellun mittaiset päivät, tummanvihreällä ekstratunteja sisältäneet päivät sekä punaisella päivät, jolloin aikataulussa on tapahtunut lipsahdus.

Teh- tänä nro	Tehtävä	Lopputulos	Aloituskriteeri	Vastuu	Tunti- määrä	Tammikuu						Hel
						4			5			6
						ma	ke	pe	ma	ke	pe	ma
1	Projektin käynnistys	Hyväksytty projektisuunnitelma	Hyväksytty aihe-ehdotus	or	60			-7		-6		-6

Kuvio 8 Projektin käynnistuksen tehtäväluettelo ja ajoitus

Teh- tänä nro	Tehtävä	Lopputulos	Aloituskriteeri	Vastuu	Tunti- määrä	Helmikuu													
						6		7		8		9							
						ke	pe	ma	ke	pe	ma	ke	ma	ke	to	pe	la	su	
2	Aloituskokous	Käynnistetty projekti	Hyväksytty projektisuunnitelma, aloituskokouksen esityslista ja materiaali jaettu		1								25.2.						
	Pyhäpäivät	29.3. pitkäperjantai, 1.4. 2. pääsiäispäivä, 1.5. Vappu																	
	Hiihtoloma	18.-22.2.																	
	IMS Muutto	Yrityksen muutto uusiin tiloihin 7.-31.3., enintään kahden päivän heitto aikatauluun																	
	ONT-seminaari	28.2. kuuntelu, X.X opponointi																	
	Kypsyyssnäyte	8.5. klo 13:00																	
	Seminaari	Opponointi ja esitys																	
	Ekstrapäivät	Ekstrapäivät aikataulun kiinnisaamiseen																	
3	Sprint 1	Määrittely ja kuvauskeinot käyty läpi, ont-raportti aloitettu	Aloituskokouksen pöytäkirja tehty ja jaettu	pp	69		-5	-7	-7	-10			1					8	8

Kuvio 9 Sprintin 1 aikainen tehtäväluettelo ja ajoitus

Teh- tänä nro	Tehtävä	Lopputulos	Aloituskriteeri	Vastuu	Tunti- määrä	Maaliskuu												
						10				11				12				
						ma	ke	pe	la	su	ma	ke	pe	la	su	ma	ti	
	Ekstrapäivät	Ekstrapäivät aikataulun kiinnisaamiseen																
4	Sprint 2	Prototyypit luonnosteltu ja arvioitu, ont-raportin tietoperusta tehty	Määrittely aloitettu ja kuvauskeinot valmiina	pp	68					8	8					8	2	

Kuvio 10 Sprintin 2 aikainen tehtäväluettelo ja ajoitus

Teh- tänä nro	Tehtävä	Lopputulos	Aloituskriteeri	Vastuu	Tunti- määrä	Maaliskuu						Huhtikuu						
						12			13			14				15		
						ke	to	pe	ma	ke	pe	ma	ke	to	pe	ma		
	Pyhäpäivät	29.3. pitkäperjantai, 1.4. 2. pääsiäispäivä, 1.5. Vappu																
	IMS Muutto	Yrityksen muutto uusiin tiloihin 7.-31.3., enintään kahden päivän heitto aikatauluun																
	Seminaari	Opponointi ja esitys																
	Ekstrapäivät	Ekstrapäivät aikataulun kiinnisaamiseen																
5	Ohjauskokous	Edistyminen hyväksytty	Ohjauskokouksen esityslista ja materiaali jaettu	or	2	20.3.												
6	Sprint 3	Käyttöliittymä luotu, ont-raporttia päivitetty	Ohjauskokouksen pöytäkirja tehty ja jaettu	pp	66			3	-2	-8								

Kuvio 11 Sprintin 3 aikainen tehtäväluettelo ja ajoitus

				Huhtikuu							Toukokuu							
				15		16		17			18		19		20			
Tehtävä	Lopputulos	Aloituskriteeri	Vastuu	Tunti- määrä	ke	pe	ma	ke	pe	ma	ke	pe	su	ma	ke	pe	ma	ke
Pyhäpäivät	29.3. pitkäperjantai, 1.4. 2. pääsiäispäivä, 1.5. Vappu																	
Kypsyysnäyte	8.5. klo 13:00																	
Ekstrapäivät	Ekstrapäivät aikataulun kiinnisaamiseen																	
Sprint 4	Testaus ja refaktorointi tehty, löydetty poikkeukset korjattu, loppuraportti kirjoitettu, ont-raportti viimeistelty, dokumentoinnit päivitetty	Käyttöliittymä valmis	pp	93	8	-2	8	8	-2	8	3	10	8	10	5			
Loppukokous	Projekti viety loppuun	Loppukokouksen esityslista ja materiaali jaettu	or	2											6.5.			
Jälkipyykki	Loppukokouksen pöytäkirja tehty ja jaettu	Loppukokous pidetty, projektin tuote luovutettu	or, pp	?														

Kuvio 12 Sprintin 4 aikainen tehtävluettelo ja ajoitus

Tehtävä nro	Tehtävä	Lopputulokset	Valmistumisaste	Aloitettu (pvm)	Tuntiarvio	Käytetty tuntimäärä	Päätetty (pvm)
1	Projektin käynnistys	Hyväksytty projektisuunnitelma	valmis	21.1.2013	60	31	25.2.2013
		Projektisuunnitelman teko	valmis	21.1.2013	40	21	25.2.2013
		Aiheeseen syventyminen	valmis	25.1.2013	20	10	28.1.2013
2	Aloituskokous	Käynnistetty projekti	valmis	30.1.2013	9	11	25.2.2013
		Aloituskokouksen järjestäminen	valmis	30.1.2013	1	3	13.2.2013
		Kokouskutsun ja esityslistan teko	valmis	1.2.2013	3	3	1.2.2013
		Aloituskokouksen pitäminen	valmis	25.2.2013	1	1	25.2.2013
3	Sprint 1	Pöytäkirjan laatiminen ja lähettäminen	valmis	25.2.2013	4	4	25.2.2013
		Määrittely aloitettu ja kuvauskeinot valmiina	jatkettaaan s2	1.2.2013	60	60	3.3.2013
		Määrittelydokumentin aloittaminen	kesken	1.2.2013	20	11	sprint 2
		Kuvauskeinojen läpikäynti	kesken	8.2.2013	10	2	sprint 2
		Opinnäytetyöraportin aloittaminen	valmis	27.2.2013	22	32	3.3.2013
		edistymisraportin kirjoittaminen	valmis	1.3.2013	3	3	1.3.2013
		toteutumaseurannan aloittaminen	valmis	1.3.2013	2	2	1.3.2013
	työajanseurannan aloittaminen	valmis	1.2.2013	2	9	27.2.2013	
	Opinnäytetyön kansilehden teko	valmis	1.2.2013	1	1	1.2.2013	

Kuvio 13 Projektin aloituksen ja sprintin 1 tehtävät

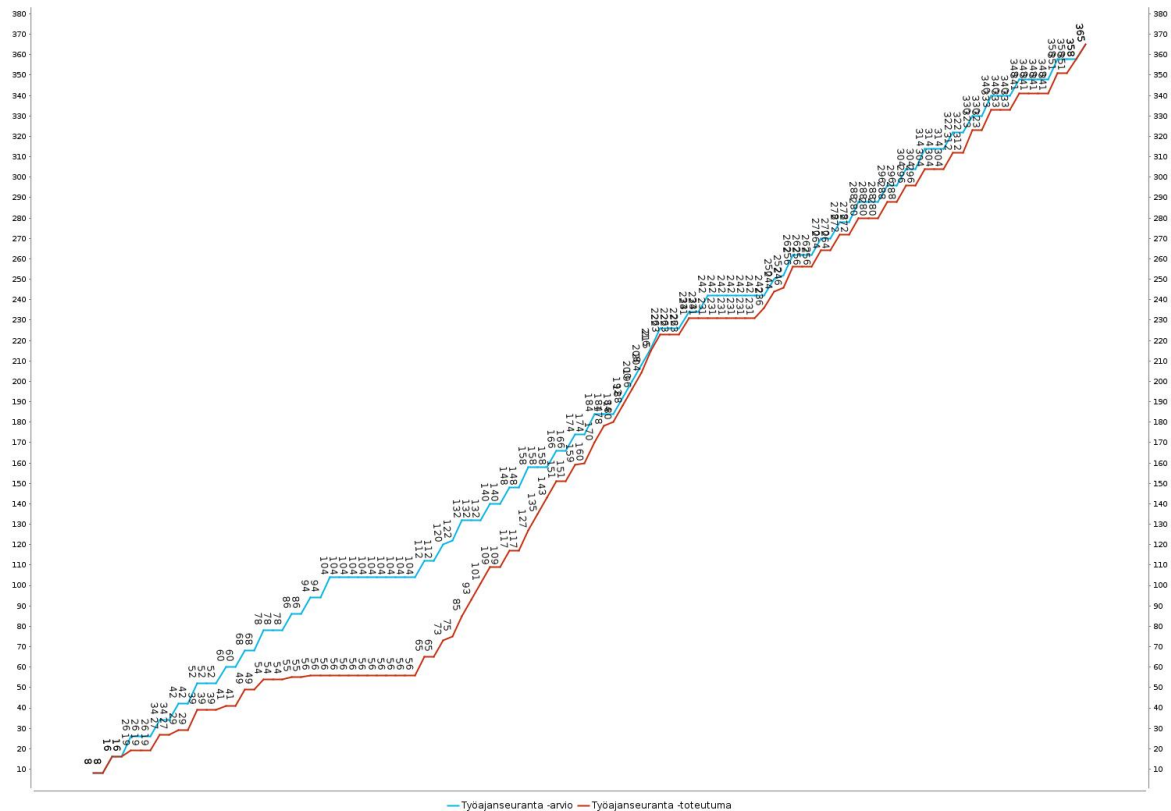
Tehtävä nro	Tehtävä	Lopputulokset	Valmistumisaste	Aloitettu (pvm)	Tuntiarvio	Käytetty tuntimäärä	Päätetty (pvm)
4	Sprint 2	Prototyypit luonnosteltu ja arvioitu, ont-raportin tietoperusta tehty	jatketaan s3	4.3.2013	100	94	19.3.2013
		Määrittelydokumentin jatkaminen	valmis	9.3.2013	20	11	11.3.2013
		Kuvauskeinojen läpikäynti	valmis	6.3.2013	5	4	6.3.2013
		ONT-raportin tulosten ja johdannon kirjoittaminen	kesken	4.3.2013	20	16	sprint 3
		ONT-raportin tietoperustan ja käsitteiden täydentäminen	kesken	4.3.2013	16	13	sprint 3
		edistymisraportin kirjoittaminen	valmis	13.3.2013	3	2	13.3.2013
		esityslistan ja kokouksun kirjoittaminen	valmis	4.3.2013	1	1	4.3.2013
		Prototyypien teko ja parantelu	valmis	11.3.2013	29	43	19.3.2013
		Mielipiteiden keruu, lomakkeen teko ja lähetys	valmis	4.3.2013	2	2	4.3.2013
		Mielipiteiden keruu, vastausten analysointi	valmis	6.3.2013	4	2	6.3.2013
6	Sprint 3	Käyttöliittymä luotu, ont-raporttia päivitetty	valmis	20.3.2013	84	68	8.4.2013
		ONT-raportin tulosten kirjoittaminen	valmis	21.3.2013	4	1	8.4.2013
		ONT-raportin tietoperustan ja käsitteiden täydentäminen	valmis	21.3.2013	9	21	8.4.2013
		edistymisraportin kirjoittaminen	valmis	3.4.2013	3	2	3.4.2013
		Ohjauskokoukseen valmistautuminen ja osallistuminen	valmis	20.3.2013	2	2	20.3.2013
		ohjauskokouksen pöytäkirjan laatiminen ja lähettäminen	valmis	20.3.2013	4	1	21.3.2013
		Seminaariin valmistautuminen, esittäminen	valmis	3.4.2013	8	6	4.4.2013
		Seminaariin valmistautuminen, opponointi	valmis	1.4.2013	8	6	4.4.2013
		Käyttöliittymän luonti	valmis	20.3.2013	30	28	8.4.2013
		muuton aiheuttamat työt	valmis	3.4.2013	16	1	3.4.2013

Kuvio 14 Sprintin 2 ja 3 tehtävät

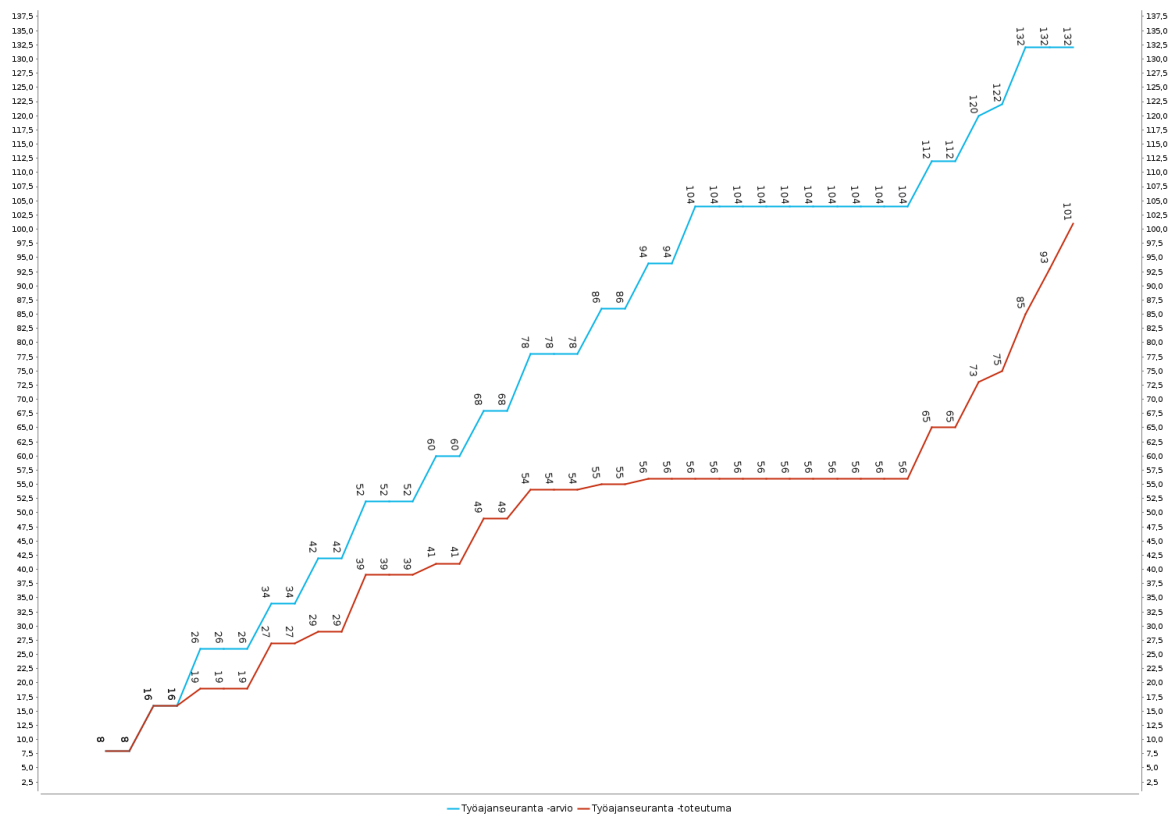
Tehtävä nro	Tehtävä	Lopputulokset	Valmistumisaste	Aloitettu (pvm)	Tuntianvio	Käytetty tuntimäärä	Päätetty (pvm)
7	Sprint 4	Testaus ja refaktorointi tehty, löydetty poikkeukset korjattu, loppuraportti kirjoitettu, ont-raportti viimeistelty, dokumentoinnit päivitetty	valmis	9.4.2013	93	101	6.5.2013
		sprintin aloittaminen	valmis	10.4.2013	1	1	10.4.2013
		ONT-raportin kirjoittaminen loppuun, johdanto	valmis	15.4.2013	4	3	5.5.2013
		ONT-raportin kirjoittaminen loppuun, tiivistelmä	valmis	19.4.2013	4	3	5.5.2013
		ONT-raportin kirjoittaminen loppuun, teoriaperusta	valmis	17.4.2013	4	9	5.5.2013
		ONT-raportin kirjoittaminen loppuun, tulokset	valmis	15.4.2013	4	7	5.5.2013
		ONT-raportin kirjoittaminen loppuun, pohdinta	valmis	17.4.2013	4	12	5.5.2013
		ONT-raportin kirjoittaminen loppuun, yleinen siistintä	valmis	19.4.2013	4	19	5.5.2013
		loppuraportin kirjoittaminen	valmis	26.4.2013	4	5	29.4.2013
		loppukokouksen valmistautuminen ja osallistuminen	valmis	29.4.2013	5	6	6.5.2013
		käyttöliittymän refaktorointi, uuden luonti	valmis	10.4.2013	4	2	10.4.2013
		käyttöliittymän refaktorointi, vanhan muokaus	valmis	10.4.2013	4	7	26.4.2013
		käyttöliittymän refaktorointi, katselu, ylläpitäjä	valmis	10.4.2013	4	2	10.4.2013
		käyttöliittymän refaktorointi, katselu, käyttäjä	valmis	12.4.2013	4	2	12.4.2013
		käyttöliittymän refaktorointi, yleinen koodin siistiminen	valmis	19.4.2013	4	6	24.4.2013
		testaussuunnitelman teko	valmis	12.4.2013	8	5	24.4.2013
		käyttöliittymän testaus toiminnallisuuden osalta	valmis	12.4.2013	4	2	24.4.2013
		käyttöliittymän testaus yhtenevän ilmeen osalta	valmis	15.4.2013	4	2	15.4.2013
		käyttöliittymän validointi	valmis	17.4.2013	4	1	17.4.2013
		testauksen poikkeusten korjaus, toiminnallisuus	valmis	15.4.2013	5	1	15.4.2013
		testauksen poikkeusten korjaus, yhtenevä ilme	valmis	15.4.2013	5	1	15.4.2013
		määritysdokumentin päivitys	valmis	19.4.2013	5	2	19.4.2013
		muut projektin lopun toimenpiteet	valmis	5.5.2013	4	3	6.5.2013

Kuvio 15 Sprintin 4 tehtävät

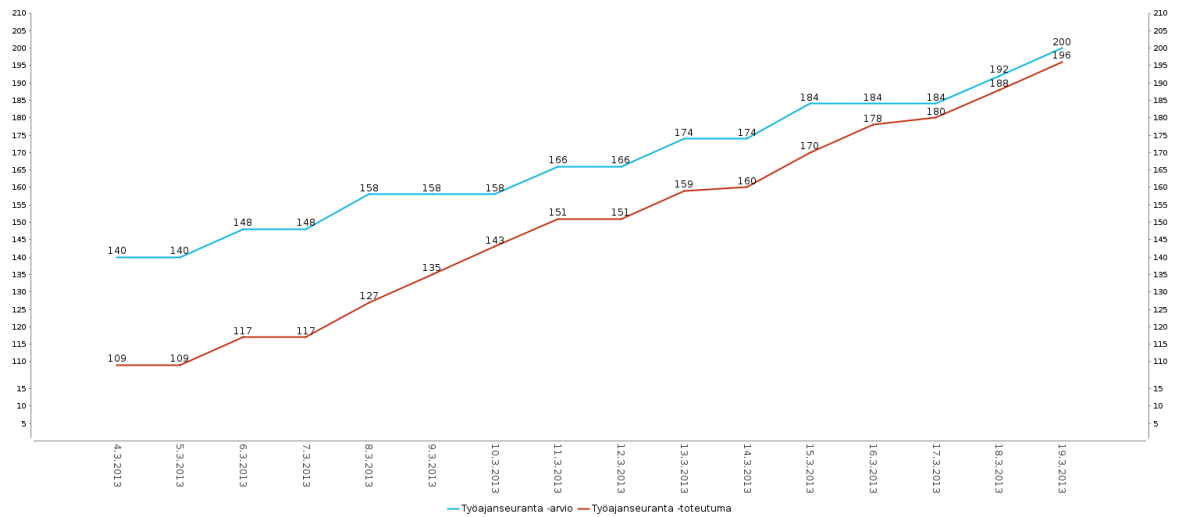
Liite 6. Aikataulu, suunniteltu verrattuna toteutuneeseen



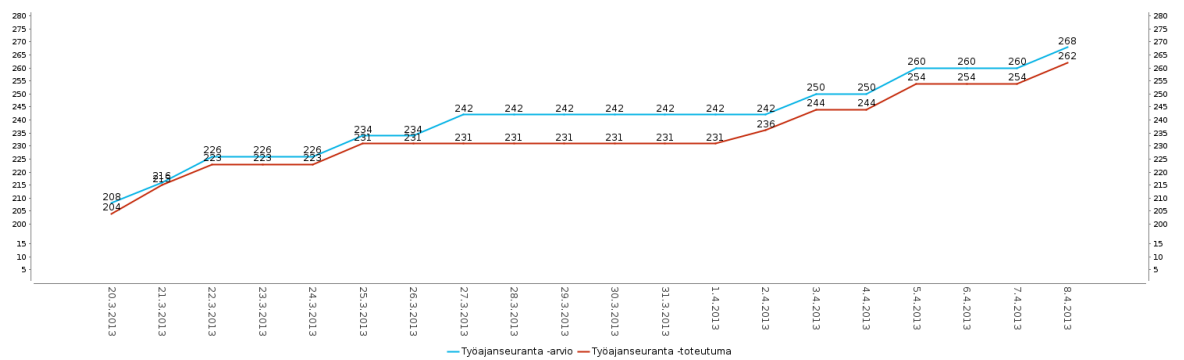
Kuvio 16 Tuntikuluma koko projektin ajalta



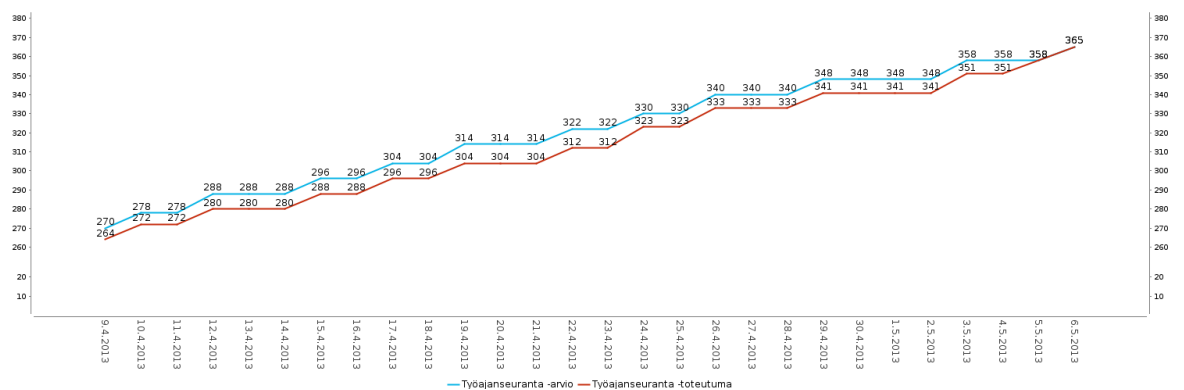
Kuvio 17 Tuntikuluma projektin alusta sprintin 1 loppuun



Kuvio 18 Tuntikuluma sprintin 2 ajalta



Kuvio 19 Tuntikuluma sprintin 3 ajalta



Kuvio 20 Tuntikuluma sprintin 4 ajalta

Toimijana minun tulee voida...
Käyttäjä	nähdä minulle kohdistetut lukemattomat viestit
Käyttäjä	nähdä uusimmat uutiset
Käyttäjä	nähdä sovelluksen uusimmat sisällöt
Käyttäjä	nähdä, ketkä käyttäjät ovat tällä hetkellä aktiivisena
Käyttäjä	nähdä suosikkini
Käyttäjä	nähdä toimintoani odottavat sisällöt
Käyttäjä	nähdä valitut mittarit
Käyttäjä	nähdä valitut IMS-sisällöt
Käyttäjä	nähdä yhteiset ja omatekemäni näkymät
Käyttäjä	etsiä tietty sisältö sovelluksesta syöttämällä sisällön nimi tai id
Käyttäjä	siirtyä valitsemaani sisältöön
Käyttäjä	siirtyä toiseen näkymään
Käyttäjä	luoda oma aloitusnäkyminen-sivuni
Käyttäjä	liittää omaan aloitusnäkymykseen haluamiani sisältökokonaisuuksia
Käyttäjä	asettaa näkymän päivittymään tietyn ajan välein
Käyttäjä	muokata omia näkymiäni järjestyksen, osakokonaisuuksien tyy- pin, näkymän nimen ja sen sisältämien osakokonaisuuksien osal- ta (poisto ja lisäys)
Ylläpitäjä	muokata omia ja yhteisiä näkymiä järjestyksen, osakokonaisuuk- sien tyyppin, näkymän nimen ja sen sisältämien osakokonaisuuk- sien osalta (poisto ja lisäys)

Salaiset liitteet

- Liite 8. Mallinnukset
- Liite 9. Valmis käyttöliittymä
- Liite 10. Koodin kansiorakenne ja tiedostot