

SELAINPOHJAISEN VARAUSJÄRJESTELMÄN SUUNNITTELU

LAHDEN AMMATTIKORKEAKOULU
Muotoilu- ja taideinstituutti
Viestinnän koulutusohjelma
Multimediatuotannon pääaine
Selainpohjaisen varausjärjestelmän
suunnittelu
2012
Samu-Erkki Aaramaa

TIIVISTELMÄ

Lahden ammattikorkeakoulu
Viestinnän koulutusohjelma
Medianomi (AMK)

AARAMAA, SAMU-ERKKI: Selainpohjaisen varausjärjestelmän
suunnittelu

Multimediatuotannon opinnäytetyö, 63 sivua, 11 liitesivua
Kevät 2012

Tiivistelmä

Opinnäytetyössä tutkin Lahden Ammattikorkeakoulun Viestinnän koulutusohjelmassa olevia laitteiston ja tilojen varauskäytänteitä. Tähän pohjautuen teen selainpohjaisen varausjärjestelmän alustavan konseptisuunnitelman.

Tarkoitukseni on luoda pohja selainpohjaisen varausjärjestelmän toteuttamiselle.

Toteutin nykytilanteen kartoituksen kyselytutkimuksin. Muut selvitykset ja suunnitelmat perustuvat tutkimustuloksiin sekä lähdemateriaaliin.

Kyselytutkimusten tulokset osoittavat, että Lahden Ammattikorkeakoulun Viestinnän koulutusohjelman lainausjärjestelmät ovat vanhahkoja ja puutteellisia.

Kaikkiin käytössä oleviin lainauskäytänteisiin tarvittaisiin parannuksia. Selainpohjainen varausjärjestelmä vastaisi tarpeisiin. Idea selainpohjaisesta varausjärjestelmästä on herättänyt kiinnostusta ja saanut positiivista palautetta sekä henkilökunnan että opiskelijoiden tahoilta.

Avainsanat

selainsovellukset, varaukset, konseptisuunnittelu, ohjelmistokehitys, sovelluskehikset

ABSTRACT

Lahti University of Applied Sciences
Visual Communication
Bachelor of Culture and Arts

AARAMAA, SAMU-ERKKI: Designing a browser-based
booking system

Bachelor's Thesis in Visual Communication, 63 pages, 11 appendices
Spring 2012

Abstract

In this thesis I research equipment and property booking systems currently used in Visual Communication in Lahti University of Applied Sciences. Preliminary concept plan for browser-based booking system is made based on these researches.

The purpose is to lay the groundwork for a browser-based booking system for its implementation.

Research of current situation was accomplished by surveys. Other studies and plans are based on surveys, as well as the general research of the field.

Results of these surveys point out that booking systems in Visual Communication in Lahti University of Applied Sciences are outdated and they lack needed features.

Improvements in all booking systems in Visual Communication in Lahti University of Applied Sciences are needed. A browser-based reservation system would correspond to these needs. Idea of a browser-based reservation system has attracted interest and has received positive feedback from both staff and students.

Key words

browser applications, reservations, concept design, software development, application frameworks

SISÄLLYSLUETTELO

JOHDANTO	1	JÄRJESTELMÄN TOTEUTTAMISESTA	46
WEB-OHJELMAN LUOMISPROSESSI	2	Resursointi	48
Vaatimusmäärittely ja etenemissuunnitelma	4	Suosittamani sovelluskehukset (engl. framework)	49
Käyttäjien kartoitus ja turvallisuusmääritelmät	5	Mikä on sovelluskehys ja miksi käyttää sellaista?	49
Käyttömäärittelyt tai ominaisuuksien määrittelyt	8	Sovelluskehys JavaScript ohjelmointikielelle	50
Kolmannen osapuolen palveluntarjoajien kartoitus	9	Sovelluskehys PHP -ohjelmointikielelle	51
Tekniset määrittelyt	11	MVC-arkkitehtuuri	52
Ongelman ja ratkaisun määrittely	13	Sovelluskehys CSS-tyylimäärittelyille	54
Aivoriihi	13	YHTEENVETO	56
Esimerkkikäyttäjien odotusten tunnistaminen	14	Työn tarkoitus	58
Kontekstiskenaarioiden luominen	14	Keskeinen sisältö	58
Vaatimusten määrittely	14	Tärkeimmät tulokset	58
Visuaalinen ohjeistus ja käyttöliittymäsuunnittelu	15	Johtopäätelmiä tulosten pohjalta	59
Web-ohjelman kehittäminen	17	Työn onnistumisen arviointi	59
Testaus ja virheenkorjaus	18	Selainpohjaisen varausjärjestelmän tuottaminen	59
NYKYTILANTEEN KARTOITUS	20	Miten varausjärjestelmä toteutetaan käytännössä?	60
WinhaResurssit-ohjelma	22	LÄHTEET	62
Elokuva ja TV-ilmaisu	24	Kirjalliset lähteet	62
Multimediatuotanto	25	Elektroniset lähteet	62
Valokuvaus	27	Suulliset lähteet	63
KESKEISIMMÄT TOIMINNOT	32	LIITTEET	64
Järjestelmän käyttäjät, laitteet ja tilat	35	LIITE 1 Erikoislaboratoriomestari Jouni Kärkisen haastattelu	64
Käyttäjät	35	Sähköpostikysely 1	64
Laitteet	36	Sähköpostivastaus 1	65
Tilat	37	Sähköpostikysely 2	65
Käyttäjätasot	38	Sähköpostivastaus 2	65
Käyttäjärühmät ja yksilöt	38	Keskustelu 8.11.2011	66
Laitteiden ja laitteen osien linkittäminen	38	Keskustelu 22.11.2011	66
Varausaika	39	Keskustelu 10.02.2012	67
Varauksen hyväksyminen	41	LIITE 2 Tuntiopettaja Tommi Liuksen haastattelu	68
Kommunikointi	42	LIITE 3 Erikoislaboratoriomestari Markku Reiniharjun haast.	70
Omat tiedot	43		
Viivakoodi tai QR-koodi	44		
Laiteryhmät ja laitekohtaiset ominaisuudet	45		
Laajennettavuus	45		

JOHDANTO

Opinnäytetyöni on selainpohjaisen varausjärjestelmän konseptisuunnitelma Lahden Ammattikorkeakoulun Viestinnän koulutusohjelman pääaineille.

Idea selainpohjaisen varausjärjestelmän toteuttamisesta syntyi käytännön tarpeesta, varatessani laitteita opiskelun aikana. Koin vihkoihin perustuvan laitteiston varaamissysteemin epäkäytännölliseksi. Esittelin idean Arto Kutvoselle (Kutvonen 2011-2012), jolloin idea jalostui järjestelmän toteuttamisesta tutkielman tekemiseen. Tutkielmani on tarkoitus auttaa järjestelmän toteuttamisessa, jos Lahden Ammattikorkeakoulun Viestinnän koulutusohjelma päättää hankkia järjestelmän käyttöönsä. Asiasta on keskusteltu Viestinnän koulutusohjelmassa ja järjestelmälle on havaittu olevan selvä tarve.

Nykyisin kullakin Viestinnän koulutusohjelman pääaineella on oma varaussysteemi, joku toimivampi kuin toinen, silti ei ole yhtään sellaista, joka ei kaipaisi vähintään päivitystä. Tavoitteenani on siis suunnitella nykyaikainen varausjärjestelmä, joka voitaisiin ottaa yhtenäisesti käyttöön koko viestinnän koulutusohjelmassa. Selainpohjaisen varausjärjestelmän tarkoituksena olisi yksinkertaistaa, yhtenäistää ja helpottaa laitteiston varausta, seuranta ja varastokirjanpitoa, sekä ylläpitäjien että käyttäjien näkökulmasta. Selainpohjainen järjestelmä mahdollistaa varausten tekemisen mistä ja milloin vain, kunhan internetyhteys on saatavilla.

Luvussa **Web-ohjelman luomisprosessi** esittelen yleisen internetohjelmien tuotantoprosessin. Luomisprosessi perustuu Kohanin listaan (Kohan 2012) sekä alan kirjallisuuteen, joita täydennän omiin kokemuksiini perustuvilla kommentteilla.

Nykytilanteen kartoituksessa tutkin Lahden Ammattikorkeakoulun Viestinnän koulutusohjelmassa käytössä olevia varaussysteemejä. Toteutan tutkimukset keräämällä kyselyillä tietoa järjestelmistä.

Luvussa **Keskeisimmät toiminnot** kartoitan selainpohjaisen varausjärjestelmän tärkeimmät ominaisuudet. Luku pohjautuu **Nykytilanteen kartoitukseen**, sekä alan yleiseen tutkimukseen.

Yhteenvedossa tiivistän opinnäytetyön kokonaisuudeksi, sekä pohdin jatkotoimenpiteitä.

WEB-OHJELMAN LUOMISPROSESSI

Tässä luvussa esittelen web-ohjelman luomisprosessin käyttäen pohjana Kohanin listaa (Kohan 2012), jota täydennän muilla lähteillä, sekä omiin kokemuksiini perustuvilla kommentteillani.

Vaatimusmäärittely ja etenemissuunnitelma

Tämän vaiheen tarkoituksena on luoda vankka kivijalka projektille sekä saattaa sen eri osapuolet yhteisymmärrykseen siitä mitä ollaan tekemässä. Tarkoituksena on kartoittaa mitä asiakas tarvitsee, mihin tarkoitukseen ohjelmaa käytetään, mitkä ovat tavoitteet ja miten niihin päästään. On tärkeää että projektin eri osapuolilla on yhteisymmärrys lopputuloksesta ja kokonaisprosessista jotta väärinkäsitykset voidaan minimoida.

Vaatimusmäärittelyssä ja etenemissuunnitelmassa määritellään web-ohjelman tavoitteet ja tarkoitus. Ne antavat projektille selkeän suunnan sekä auttavat pysymään suunnitelmassa ja tavoitteiden saavuttamisessa. Etenemissuunnitelma määrittelee web-ohjelman tulevaisuudensuunnitelmat ja sisältää alustavat aikataulut ohjelman toteutuksesta. (Kohan 2012.)

Vaatimusmäärittelyssä kartoitetaan ohjelman ominaisuudet. Tärkeintä on ymmärtää, mitä käyttäjät odottavat ohjelmalta. Miksi he käyttäisivät ohjelmaa, ja miten he saavuttavat tavoitteensa? Etenemissuunnitelma on prosessin läpiviennin kannalta tärkeä, mutta se ei saa rajoittaa suunnittelua sitomalla projektia tiettyihin suunnitteluratkaisuihin tai teknologioiden valintaan.

On huomioitava, että käyttäjien monet toiminnot ovat rutinoituneet tai heiltä jää jostakin muusta syystä kertomatta kokemuksistaan ja toiveistaan. Tämän vuoksi pelkkiin haastatteluihin ei voi luottaa, vaan suunnittelijan on myös tutkittava käyttäjien toimia oikeissa tai mahdollisimman oikeissa käyttötilanteissa (Kuutti 2003, 144-146).

Suunnittelutyökalut tarjoavat tarpeelliset yhteydet käyttäjämallien ja muiden mallien välille ja antavat viitekehyksen suunnittelulle. Tämä vaihe tarjoaa skenaarioihin perustuvat apuvälineet ennen kaikkea käyttäjien tavoitteiden saavuttamiseksi ja tarpeiden tyydyttämiseksi. Käyttäjät auttavat ymmärtämään mitkä toiminnot ovat oikeasti tärkeitä ja miksi. Tämä auttaa mahdollisimman helppokäyttöisen ja miellyttävän järjestelmän luomisessa. (Cooper 2007, 21.)

Käyttäjien kartoitus ja turvallisuusmääritelmät

Tämän vaiheen tavoite on määrittää minkälaisia ovat ohjelman käyttäjät ja mitkä ovat ohjelman tai sen käytön turvallisuusriskit. Käyttäjien tunteminen auttaa tekemään heidän tarkoitukseensa sopivan ohjelman. Tietoturvariskien tunteminen auttaa luomaan turvallisemman ohjelman tahattomia virheitä sekä tahallisia vahingoittamisyriä vastaan.

Käyttäjien kartoittaminen ja analyttinen raportti sisältää seuraavat näkökulmat:

- Käyttäjätyypit käytettävyyssnäkökulmasta:
 - Käyttäjien kartoitus prosentuaalisesti:
 - Tietoteknisesti aloittelevat käyttäjät
 - Tietoteknisesti edistyneet käyttäjät
 - Tietotekniikan ammattilaiset ja muut tietoteknisesti hyvin edistyneet käyttäjät
 - Ikäjakaumat
 - Sukupuolijakaumat
 - Käyttötyypit ja -tasot:
 - Intranetin käyttäjät
 - Internetin käyttäjät
 - Ekstranetin käyttäjät
 - Yksitasoinen
 - Monitasoinen

(Kohan 2012.)

Opinnäytetyöni selainpohjainen varausjärjestelmä on suunniteltu käytettäväksi internetin välityksellä, joten intra- ja ekstranetin käyttäjät voidaan jättää analysoinnin ulkopuolelle. Järjestelmässä tulee olemaan monen tasoisia käyttäjiä aina ylläpitäjästä laitteen lainaajaan, joten eri käyttäjätasojen käyttäjien analysointi on tärkeää.

Esitutkimusvaiheessa tutkitaan mahdollisia, tai todellisia käyttäjiä kenttätutkimusmenetelmin, jotta saataisiin mahdollisimman tarkkaa tietoa järjestelmän käyttäjistä. Kenttätutkimuksella tarkoitetaan käyttäjien tarkkailemista esimerkiksi samantapaisten ohjelmien käytössä tai tätä tarkoitustavarten rakennettujen prototyyppien avulla. Yksi esitutkimusvaiheen tärkeimmistä tehtävistä on tutkia ihmisten käyttäytymismalleja, jotka auttavat tulevan tai olemassa olevan järjestelmän käyttötapojen määrittämisessä (Cooper 2007, 20-21).

Vuorovaikutusratkaisut tulisi kehittää mahdollisimman pitkälle ennen teknisten ratkaisujen suunnittelua, jotta tekniset suunnittelijat tietäisivät mitä heiltä odotetaan. Tällöin vuorovaikutusratkaisut ohjaisivat suunnittelua vailla teknisiä rajoituksia (Saariluoma 2010, 116).

- Käyttäjien ja ympäristön analysointi turvallisuusnäkökulmasta:
 - Vyöhykkeen turvataso
 - Ohjelman tietoturva-aukot
 - Käyttäjien tekemien virheiden minimointi

(Kohan 2012.)

Käyttäjien tekemiä virheitä voidaan minimoida erilaisilla käyttöliittymäsuunnittelullisilla ratkaisuilla. Käsien kirjoitettavien syötteiden määrää tulee välttää aina kun se on mielekästä, päivämäärän voi esimerkiksi valita kalenterista, jottei sitä tarvitse käsien kirjoittaa. Käyttäjien syötteet, olivat ne sitten kirjoitettuja tai erilaisista valikoista valittuja, on tarkistettava ohjelmallisesti. (Hakala 2006, 319-326.)

Ympäristön turvatasoanalysointi on sinänsä haasteellista, koska internetiin pääsee niin monenlaisilta laitteilta. Kotona yksin pöytäkoneelta järjestelmää käyttävän henkilön ympäristö on kohtalaisen turvallinen, kun taas vaikkapa linja-autossa tai junassa tuntemattoman ihmisen vieressä tabletilla järjestelmään kirjautuvan käyttäjän ympäristö on tietoturvan kannalta turvattomampi. Näin ollen sensitiivinen data tulisi suojata ainakin ulkopuolisilta katseilta.

Salasanakenttien käyttäminen kirjautumissivulla on yksi tällainen suojausmekanismi. Salasanakentät eivät näytä käyttäjän syötettä vaan palloja tai tähtiä merkkien tilalla.

Internetin turvallisuuden kartoittaminen ja turvallisen järjestelmän suunnittelemisen on erittäin tärkeää. Järjestelmän turvallisuus olisi hyvä delegoida sellaiselle taholle, joka on tietoturvaan ja erityisesti internetin tietoturvaan perehtynyt.

- Käyttäjämäärien/käyttöasteen arviointi:
 - Sopiviin aikajaksoihin perustuva käyttäjämäärien arviointi

(Kohan 2012).

Käyttäjämäärien ja etenkin todellisen kuormituksen arvioiminen ei ole helppo tehtävä. Sen ovat kerta toisensa jälkeen todistaneet isot yritykset, jotka kaikista suunnitteluista ja testauksista huolimatta ovat tehneet virheitä kuormituksen arvioinnissa. Esimerkkinä mainittakoon VR Oy:n sähköisten palveluiden uudistaminen, joka näkyi käyttäjille vuoden 2011 loppupuolella järjestelmien lamaantumisenä - korjaustyöt kestivät viikkoja. Näin suuren mittaluokan työstä ei tässä tietenkään ole kyse, ja pienemmässä mittakaavassa myös ongelmat on helpompi ja nopeampi korjata. Joka tapauksessa, palvelinkapasiteettia suunnitellessa on hyvä laskea riittävästi pelivaraa, niin suorituskyvyn kuin tallennustilankin suhteen. Tämän mittaluokan projektissa palvelinkapasiteetin järkevä ylimitoitus ei ole myöskään kallista.

Selainpohjaisen varausjärjestelmän käyttäjien kartoitus on käyty pintapuolisesti läpi luvussa **Käyttäjät**. Ennen järjestelmän varsinaista toteuttamista on syytä tehdä paljon perusteellisemmat käyttäjäprofiilit, joissa määritellään kunkin käyttäjän toiminnot seikkaperäisesti.

Käyttömäärittelyt tai ominaisuuksien määrittelyt

Käyttömäärittelyjen tavoite on selvittää miten käyttäjät pääsevät tavoitteisiinsa. Tämän selvityksen pohjalta voidaan suunnitella ohjelmaan ominaisuuksia, joiden avulla nämä tavoitteet saavutetaan mahdollisimman helposti. Näin minimoidaan käyttäjien tekemien virheiden mahdollisuus ja tuotetaan käyttäjille mahdollisimman miellyttäviä käyttökokemuksia.

Web-ohjelman käyttömäärittely ja ominaisuuksien määrittely on erittäin tärkeää, sillä määrittelyt listaavat kaikki toiminnalliset ja tekniset seikat, jotka tarvitaan järjestelmän toteuttamiseksi. Isomman mittaluokan projekteissa tämä saattaa olla hankalaa, mutta suositeltavaa tehdä, koska nämä määrittelyt toimivat viestintävälineenä projektiryhmän sisällä ja selkeyttävät työtä myöhemmässä vaiheessa (Kohan 2012).

Käyttömäärittely ja ominaisuuksien määrittely voidaan tehdä tekstuaalisena listauksena. Apuna voidaan käyttää erilaisia grafiikoita.

Käyttötapaukset ovat kustannustehokas väline käyttäjän tietyn tavoitteen saavuttamisen kuvaamiseksi. Markkinoilla on valmiita ohjelmia graafisten käyttötapausten tekemiseksi, näistä tyypillisimpinä UML-ohjelmat (engl. Unified Modeling Language), mutta niitä voi piirtää käsinkin. Ajan säästämiseksi käyttötapaukset on syytä priorisoida, toimintojen esiintymistiheyden ja tarpeellisuuden mukaan. (Kuutti 2003, 132-134.)

* UML-ohjelmilla piirretään kaavioita, joissa erimuotoiset elementit kertovat esimerkiksi käyttäjän tai ohjelman toimista. Erilaiset toiminnot liitetään toisiinsa nuolien avulla, mikä helpottaa tietyn toiminnon suorittamisen seuraamista ja hahmottamista.

Riippumatta siitä, mitä tekniikoita käytetään, on mielestäni tärkeää, että määrittelyissä mietitään tarkasti käyttäjän tärkeimmät toimet ja pilkkotaan ne riittävän pieniin osiin, jotta mahdolliset ongelmakohdat löytyvät helposti. Ohjelman toiminnot on pintapuolisesti esitelty kappaleessa

Keskeisimmät toiminnot. Samalla tavoin kuin käyttäjienkin tarkempi profilointi, on toimintojen tarkempi määrittely tehtävä ennen järjestelmän toteuttamista.

Kolmannen osapuolen palveluntarjoajien kartoitus

Tavoitteena on löytää tarkoitukseen sopiva ohjelman toteuttaja ja palvelinratkaisu sekä mahdollisesti muita tarvittavia tuotteita ja palveluja. Kriteerejä ovat muun muassa turvallisuus, liikenne- ja tallennuskapasiteetti, laajennettavuus ja hinta. Valintaan vaikuttaa mahdollisesti myös palvelukieli ja maantieteellinen sijainti.

Tässä kohtaa tutkitaan muun muassa seuraavia palveluita ja tuotteita:

- Web-ohjelmia tuottava yritys
- SSL-sertifikaatin tarjoaja
- Vuokrapalvelin / konosalipalvelut
- Palvelin, verkko, palomuuuri, kuormituksen tasaus

(Kohan 2012.)

* SSL (Secure Socket Layer) on tekniikka, jolla suojataan käyttäjän päätteen ja palvelimen välinen yhteys, jolloin tietojen lähettäminen näiden välillä on turvallisempaa.

* Vuokrapalvelin on käytännössä hosting-palvelu, jossa asiakas ostaa palvelintilan, missä kaikki ohjelmistot ovat valmiina. Tämä on PK-yritysten tyypillisin tapa hankkia pohja verkkosovelluksilleen. Konesalipalvelu tarkoittaa sitä, että asiakkaalla on oma fyysinen palvelin palveluntarjoajan tiloissa.

* Kuormituksen tasauksella tarkoitetaan rauta- tai ohjelmistokokonaisuutta, joka ehkäisee ylikuormitusta eli tasaa kuormituspiikkejä muille palvelimille.

Kohanin listan kyseisestä kohdasta olen jättänyt pois maksusysteemit, koska järjestelmästä ei ole tarkoitus myydä mitään. Myös logistiikan pudotin pois, koska siihen ei tarvita ulkoista palveluntarjoajaa. Kaikki varausjärjestelmän laitteet löytyvät koulun sisältä ja ne noudetaan henkilökohtaisesti paikan päältä. Kohanin listan muita kohtia voi harkita ja kartoittaa tarpeen mukaan.

Omaa palvelinta ei yleensä ole järkevää rakentaa. Mieluummin valitaan lukuisista palveluntarjoajista, ja heidän palveluistaan sopiva paketti, jossa on myös riittävä laajennusmahdollisuus. Palvelintiloja ja erilaisia palvelukokonaisuuksia tarjoavilla yrityksillä on valvotut palvelinympäristöt, varmuuskopiokäytännöt ja muut tarvittavat ominaisuudet jo valmiina, joten iso osa palvelimiin liittyvistä huolista saadaan ulkoistettua niille vannoutuneille tahoille.

Ulkomailta saa yleensä halvempia palvelinratkaisuja, kun taas suomalaisilla saattaa olla palvelimet Suomessa, ja heiltä saattaa saada suomenkielistä asiakaspalvelua. Tällä tarkoitan, että vaikka yrityksen nimi olisi suomalainen, ja toimipaikka olisi Suomessa, se ei välttämättä takaa sitä, että heidän palvelimensa olisivat Suomessa. He voivat olla vain suomalaisia välittäjiä ulkomaalaiselle palveluntarjoajalle. Suomenkielisellä asiakaspalvelulla tarkoitan sitä, että ongelmat ja ratkaisut esitetään tavallisen ihmisen ymmärtämällä kielellä, aina tällaista ei Suomestakaan saa. Todellinen asiakaspalvelualttius selviää kokeilemalla. Palvelinten sijainti löytyy usein kyselemällä tai kaivelemalla omatoimisesti tietoa yrityksestä. Itselläni on hyviä ja huonoja kokemuksia, sekä Suomesta että ulkomailta.

Tekniset määrittelyt

Tavoitteena on määrittää tekniset raja-arvot, sopia aikatauluista, valita käytettävät teknologiat sekä luoda ohjelman rakenne.

Suunnitelma, joka kattaa alustan valinnan, kehitysympäristön määrittelyt, web-ohjelman rakenteen sekä käytettävät sovelluskehyykset ja teknologiat (Kohan 2012).

Esitutkimusvaiheessa tutkitaan olemassa olevia vastaavia tai samantyyppisiä järjestelmiä ja alan aikaisempia tutkimuksia sekä keskustellaan alan eri osa-alueiden asiantuntijoiden kanssa (Cooper 2007, 20).

Teknologian valinta kattaa myös julkaisualustan valinnan. Julkaisualustoja ovat esimerkiksi Wordpress ja Drupal. Olen yrittänyt löytää avoimen lähdekoodin maailmasta ilmaista ohjelmaa tai julkaisualustaa, jolle selainpohjaisen varausjärjestelmän voisi rakentaa. Sellaista ei ole ainakaan vielä löytynyt. Lähimmäksi pääsisi useamman erillisen ohjelman yhdistämisellä. Varsinainen varaussysteemi on aika lähellä verkkokauppaa, joten jotakin verkkokauppaa muokkaamalla voisi päästä aika lähelle suunniteltua lopputulosta. Järjestelmän muut toiminnot tulisi yhdistellä muista ohjelmista, kuten julkaisualustoista. Usean alun perin erilliseksi tarkoitettujen järjestelmien liittäminen yhdeksi toivotuksi kokonaisuudeksi, vaikuttaa kuitenkin todella työläältä ja hankalalta projektilta, suhteessa siihen että rakentaisi koko järjestelmän alusta asti. Jokaista koodiriviä ei kuitenkaan tarvitse keksiä alusta ja kirjoittaa käsin, kuten luvussa **Järjestelmän toteuttamisesta** valotetaan, vaan ohjelmistokehyyksiä voi, ja on syytä käyttää aina kun se on mahdollista.

Kehitysympäristöt käsittävät eri järjestelmiä kehittäjän työkonesta erilaisten testaukseen tarkoitettujen palvelinten kautta loppusijoituspaikkaan eli palvelintilaan, jonne järjestelmä asennetaan julkaisuvalmiina. Kehitysympäristövaatimukset riippuvat projektista.

Pienemmissä systeemeissä, kuten tavallisilla verkkosivustoilla, kehitysympäristö on yleensä kehittäjän oma työasema, ja testaukseen käytetään loppusijoituspalvelimella olevaa kansiota, johon testaajat ohjataan. Suuremmissa projekteissa käytetään usein ns. hiekkalaatikoita eli vain testaamiseen pyhitettyjä palvelimia, joissa testaajat pääsevät kokeilemaan sovelluksia.

Tekninen vaatimusmäärittelydokumentti määrittelee tarkasti käytettävän teknologian, lisenssit, versiot ja ennusteet. Aikajanadokumentti määrittelee aikarajat ohjelman ominaisuuksien tai moduulien valmistumiselle. (Kohan 2012.)

Tämän vaiheen tärkein tehtävä on selvittää miten käyttäjätutkimusten tuloksia voidaan hyödyntää parhaalla mahdollisella tavalla luodaksemme ohjelman, joka tyydyttää ja inspiroi käyttäjiä. Ensimmäisenä kehitetään käyttäjätarinoita tai skenaarioita, joiden pohjalta tehdään vaatimusmäärittely. Vaatimusmäärittelyn pohjalta luodaan sitten rautalankamalleja, joita aletaan hioa lopullisiksi suunnitelmiksi. (Cooper 2007, 109.)

Tarinoiden tarkoitus on visioida parempaa huomista käyttäjän näkökulmasta. Tässä vaiheessa voidaan käyttää nopeita piirroksia tai storyboardeja, joiden etu on nimenomaan nopeasti kertoa tarinan juoni, tuhlaamatta kuitenkaan aikaa liiallisiin yksityiskohtiin. tarinat rakennetaan käyttäjien kartoituksessa luotujen esimerkkikäyttäjien ympärille. (Cooper 2007, 110.)

Vaatimusmäärittely kiertyy suunnitteluperiaatteen ympärille, joka lyhykäisyydessään kehottaa suunnittelemaan ensin mitä sovelluksen tulee tehdä, ennen kuin suunnitellaan miten sovellus sen tekee. Cooper varoittaa hyppäämästä liian nopeasti tekemään erilaisia mallinnuksia sovelluksesta. Se johtaa yleensä kaikkien osapuolten turhautumiseen ja loputtomaan viritelmien kehittämiseen. Yleinen virhe on olettaa, että vaatimusmäärittely on vain lista ominaisuuksista ja toiminnoista. Sen sijaan Cooper ehdottaa että asian ajattelisi tarvelistana. (Cooper 2007, 114.)

Cooperin ehdotus on helppo ymmärtää ja hyväksyä, jos ajattelee asian ongelma-ratkaisu vastakkainasetteluna. Tietenkään ei ole järkevää lähteä kehittämään ensin ratkaisua ja sitten vasta miettiä mikä ongelmana olikaan.

Cooperin vaatimusmäärittelyprosessi koostuu seuraavista vaiheista.

1. Ongelman ja ratkaisun määrittely
2. Aivoriihi
3. Esimerkkikäyttäjien odotusten tunnistaminen
4. Kontekstiskenaarioiden luominen
5. Vaatimusten määrittely

Vaikka edellä mainitut kohdat ovat järjestyksessä, kyse on elävästä prosessista, jossa joudutaan toistamaan ainakin kohtia 3-5 useaan otteeseen.

(Cooper 2007, 116.)

Ongelman ja ratkaisun määrittely

Tässä vaiheessa pitäisi kaikilla osapuolilla olla jo hyvin kirkkana mielessä, mihin suuntaan ollaan menossa, mikä on kohdeyleisö ja mitkä ovat tärkeimpiä tavoitteita. **Ongelma** käsittää ne järjestelmän kohdat, jotka kaipaavat vielä kehittämistä. **Ratkaisu** määrittelee kehityksen suunnan. Sekä ongelma että ratkaisu tulisi perustua suoraan esitutkimukseen ja käyttäjämalleihin. (Cooper 2007, 116-117.)

Aivoriihi

Aivoriiehen tehtävä on tässä vaiheessa pudistaa tutkimusvaiheessa kertyneet pölyt kengistä. Aivoriiehen ei tulisi käyttää liikaa aikaa, parin tunnin sessio työryhmän kesken riittää. Tarkoituksena on saada valmiista järjestelmästä muodostuneet mielikuvat mahdollisimman hyvin pois päästä, ja kirjata ideat lipaston ylälaitikkoon odottamaan. Tarkoitus ei siis ole keksiä mitään vaan päinvastoin tyhjentää pöytä. (Cooper 2007, 117-118.)

Esimerkkikäyttäjien odotusten tunnistaminen

Odotukset perustuvat ihmisten kykyyn käsittää ja käsitellä maailmaa. Ne perustuvat koko elämän mittaiseen kokemusmaailmaan ja ovat syvästi juurtuneet alitajuntaan. Käyttäjän odotuksiin vaikuttavat monet tekijät, kuten asenteet, kokemukset, tavoitteet sekä sosiaaliset, kulttuuriset ja ympäristötekijät. Kaikki tällaiset tekijät olisi syytä selvittää käyttäjätutkimuksista. (Cooper 2007, 118-119.)

Kontekstiskenaarioiden luominen

Kontekstiskenaariot ovat kertomuksia siitä, miten käyttäjät saavuttavat tavoitteensa järjestelmää käyttämällä. Kontekstiskenaarioiden ei pitäisi mennä liian teknisiin vuorovaikutusmalleihin käyttäjän ja järjestelmän välillä, vaan pintapuolisesti kuvailla järjestelmän käyttöä käyttäjän näkökulmasta. Näissä skenaarioissa hahmotellaan mitä ominaisuuksia todella tarvitaan, miten usein niitä tarvitaan, millaisessa ympäristössä järjestelmää käytetään ja niin edelleen.

Cooperilla on mielestäni äärimmäisen kiehtova ajatus siitä, miten suunnittelija voi välttyä jumiutumasta teknologian rajoituksiin miettiessään miten tietty asia teknisesti toteutetaan. Yksinkertaisesti voi miettiä, että käyttäjällä on taikavoimia. Mitä lähemmäs tätä taianomaista asioiden hoitamista pääsee valmiissa järjestelmässä, sitä paremmin suunnittelu on onnistunut. (Cooper 2007, 119-122.)

Vaatimusten määrittely

Cooper muistuttaa edelleen, että vaatimusmäärittely ei ole lista ominaisuuksista. Pikemminkin vaatimusmäärittely selvittää mitä tietoa käyttäjälle tulee näyttää, mitä toimintoja tulee suorittaa ja mitä muita vaatimuksia on otettava huomioon esimerkiksi teknologian suhteen, tai

ylläpitoon liittyen, tai mitä pitää tehdä, jotta saadaan käyttäjälle tietynlainen käyttökokemus aikaiseksi. Vaatimusmäärittely antaa järjestelmästä karkean yleiskuvan, jonka pohjalta voidaan lähteä tekemään rautalankamalleja. (Cooper 2007, 122-123.)

Visuaalinen ohjeistus ja käyttöliittymäsuunnittelu

Tarkoituksena on luoda rautalankamalleja, käyttöliittymäsuunnitelma, ohjelman visuaalinen ohje sekä graafinen suunnitelma. Ohjeiden tulee olla niin viimeisteltyjä, että niiden pohjalta voidaan toteuttaa ohjelma.

Rautalankamallit ovat yksinkertaisia viivapiirroksia ohjelman rakenteesta (KUVA 1). Niiden tarkoituksena on sovittaa ohjelman osat paikoilleen. Tällainen malli on helposti ymmärrettävä ja huomio kiinnittyy olennaisiin osiin. Viivapiirros on helpompi, nopeampi ja halvempi tehdä sekä muuttaa, kuin pitkälle viety graafinen esitys. Kun rautalankamalli on hyväksytty, voidaan ohjelman ulkonäköä ryhtyä tyyliittelemään, siirtyen vaiheittain kohti yksityiskohtaisempia suunnitelmia.



KUVA 1. Rautalankamalli paperilla.

Hyvässä web-ohjelmassa käytetään vuorovaikutusmalleja, käyttöliittymä-ratkaisuja ja -elementtejä, jotka ovat tunnetusti helppokäyttöisiä ja luovat positiivisia käyttökokemuksia. Käyttöliittymäsuunnittelijat tekevät visuaalisen ohjeen ja käyttöliittymän rautalankamalleja tai yksinkertaisesti piirtävät luonnoksia paperille. Kun käyttöliittymä- ja vuorovaikutusmallit on hyväksytty, graafikot tekevät käyttöliittymän graafisen suunnitelman. (Kohan 2012.)

Tässä vaiheessa käytettävyyssuunnittelijat suunnittelevat toiminnot, joilla päästään haluttuihin päämääriin. Nämä päämäärät tulevat käyttäjälähtöisestä ongelma-ratkaisu-analyysistä sekä yleisistä käytettävyyteen kehitetyistä periaatteista. Ratkaisumalleja kehitellään, kunnes on enää pari hyvältä tuntuvaa vaihtoehtoa jäljellä. Kun toiminnot on suunniteltu ja tehty niistä rautalankamalleja, ryhtyvät graafiset suunnittelijat tekemään malleja ulkoasusta (KUVA 2). He määrittelevät väripaletit, fontit ja muut ulkonäköseikat. Tämän vaiheen jälkeen tehdään tarkemmat määritelmät, jotka kulkevat samaa rataa, mutta menevät yksityiskohtaisemmiksi. (Cooper 2007, 22-23.)



KUVA 2. Graafinen esitys pienine yksityiskohtineen.

Web-ohjelman kehittäminen

Tavoitteena on varsinaisen ohjelman tekeminen, jotta valmista ohjelmaa päästään testaamaan. Tämä vaihe koostuu pääasiassa ohjelmoinnista.

Web-kehittäjätiimi saa graafisen ohjeen ja aloittaa ohjelman kehittämisen seuraavalla tavalla:

- Ohjelman arkkitehtuurin ja sovelluskehityksen luominen
- Tietokannan rakenteen suunnittelu
- Ohjelman moduulien, kirjastojen ja luokkien kehittäminen/räätälöinti
- Ohjelman tekeminen ja kaikkien toimintojen sisällyttäminen - versio 1.0

(Kohan 2012.)

Ohjelma-arkkitehtuurin ja sovelluskehityksen luominen on ainakin osittain myös valitsemista jo valmiista rakenteista. Monimutkaisen ohjelma-arkkitehtuurin luominen on verrattain helppoa, mutta hyvä ohjelma-arkkitehtuuri on sellainen, josta ei saa enää mitään pois. Se on puhdas, ymmärrettävä ja ennen kaikkea ylläpidettävä. Tietokannan rakenteen suunnittelu on monivaiheinen prosessi. Rakenteesta tehdään karkeita luonnoksia, ja niitä optimoidaan, kunnes rakenne on niin yksinkertainen kuin mahdollista. Moduulit, kirjastot ja luokat ovat usein käytettyjä ohjelman osia, joita kutsutaan tarvittaessa. Kun rakenne on vankalla pohjalla, tehdään itse ohjelma aiemmin tuotettujen vaatimusmäärittelyiden pohjalta.

Testaus ja virheenkorjaus

Tavoitteena on kokeilla ohjelmaa mahdollisimman monipuolisesti virheiden löytämiseksi. Kun mahdolliset virheet on korjattu, ohjelma on julkaisuvalmis. Ohjelma siirretään lopulliseen sijoituspaikkaan ja avataan käyttäjille. Pienemmissä projekteissa testaus voidaan suorittaa myös suoraan lopullisessa sijoituspaikassa.

Hyvin toteutettu testaus auttaa tuottamaan mahdollisimman turvallisen ja luotettavan web-ohjelman. Web-ohjelman Versio 1.0 on läpikotaisin testattu ja kaikki ohjelmavirheet on dokumentoitu ja korjattu (Kohan 2012.)

Edellisestä voisi päätellä, että beta-testauksen ja virheenkorjauksen jälkeen web-ohjelma on valmis. Tätä on kuitenkin vaikea allekirjoittaa. Ohjelman ns. valmistumisen jälkeen tulee luovutus, ja hyvin todennäköisesti vielä siinäkin vaiheessa huomataan puutoksia tai parannettavaa. Myös loppukäyttäjien taholta saattaa tulla muutosvastausta tai parannusehdotuksia, jotka saatetaan vielä toteuttaa. Jokainen projekti ja ohjelmien käyttöönottilanne on yksilöllinen. Ylipäätään on aika rohkeaa puhua valmiista ohjelmasta. Mielestäni sellaista ei olekaan, sillä vaikka ohjelma toimisi juuri niin kuin se on suunniteltu, maailma kehittyy jatkuvasti, joten myös ohjelmaa on päivitettävä ajan hermolle.

Kohanin listaa voidaan hyödyntää monelta osin ihan suoraakin, mutta muutama tärkeä seikka täytyy ottaa huomioon. Ensinnäkin kyseinen kahdeksankohertainen lista edustaa vanhahtavaa vesiputousmallia, jossa edetään kohdasta toiseen ja lopuksi esitellään valmis tuote. Tämä ei ole mielestäni järkevää, koska valmiin järjestelmän muuttaminen ja korjaaminen on hidasta. Suosittelen ketterien kehitysmenetelmien harkinnanvaraista käyttöä kautta kehitysprosessin.

* Ketterissä kehitysmenetelmissä tehdään ja testataan pienempiä osia kerrallaan, jolloin päästään korjaamaan vikoja ja puutteita jo kehitysvaiheessa (Shore 2008, 18). Näin vähennetään lopulliseen järjestelmään tehtäviä muutoksia, jotka ovat verrattain kalliita.

Tunnetuimmat suunnitteluprosessin kuvaukset ovat vesiputousmalleja, jossa suunnittelu tehdään vaiheittain loppuun. Vesiputousmalli on organisoinnin kannalta järkevä ja mahdollistaa suunnittelusta puhumisen. On kuitenkin syytä huomioida mallin heikkoudet, eikä sitoutua liian aikaisessa vaiheessa noudattamaan tiettyä ratkaisua vaan ratkoa ongelmia rinnakkaistensuunnittelun avulla. (Saariluoma 2010, 115.)

Koska itseään korjaavaa tai päivittävää järjestelmää ei voida nykytekniikalla luoda, on syytä tehdä jonkinlainen ylläpitosopimus tai -suunnitelma. Se on hyvä ottaa huomioon heti projektin alkuvaiheessa, sillä se voi vaikuttaa työnkulkuun ja käytettävien teknologioiden valintaan.

Kohanin listasta puuttuu myös koulutus. Järjestelmästä on toki tarkoitus tehdä niin helppokäyttöinen kuin mahdollista, ja ideana on että ainakin ylläpitäjät pääsevät jo suunnittelu- ja testausvaiheessa mukaan, joten heidän kouluttamisensa tai perehdyttämisenä ei vaatine isoja ponnisteluja, eikä paljoa aikaa. On silti realistista ottaa mukaan ainakin jonkinlainen perehdytysuunnitelma. Käytännössä tämä voisi tarkoittaa, että uusille opiskelijoille näytetään ryhmän kesken kollektiivisesti miten järjestelmää käytetään. Perehdyttäminen voitaisiin järjestää ensimmäisenä lukuvuonna esimerkiksi joulun jälkeen, koska syksyllä tulee niin paljon uutta asiaa muutenkin, että koulutus menisi todennäköisesti hukkaan ja unohtuisi pian. Tämä on tietenkin linjakohtaista, sillä joillakin linjoilla laitteita ei saa ensimmäisenä vuonna edes lainata, kun taas joillakin linjoilla lainalaitteiden käyttö aloitetaan heti opiskelujen alettua. Näin ollen jokainen linja voisi sisäisesti päättää, miten ja milloin uudet opiskelijat opetetaan järjestelmän käyttöön. Joka tapauksessa järjestelmään tulee myös sisällyttää apu-osio, josta saa missä tahansa vaiheessa etsittyä omatoimisesti ohjeita ja apuja ainakin yleisimpiin kysymyksiin.

NYKYTILANTEEN KARTOITUS

Tässä luvussa tutkin Lahden Ammattikorkeakoulun Viestinnän koulutusohjelmassa nykyisin käytössä olevia varaussysteemejä. Olen toteuttanut tutkimukset keräämällä kyselyillä tietoa järjestelmien heikkouksista ja vahvuuksista.

WinhaResurssit-ohjelma

Tapasin Viestinnän koulutusohjelman opintosihteeri Virve Pohjolan (Pohjola 2012), selvittääkseni minkälaisia ohjelmia koululla on käytössä erilaisten resurssien varaamiseen ja voisiko niitä päivittää, tai niiden pohjalta tehdä suunnittelemani varausjärjestelmän. Opintosihteerinä hänellä on tietoa tällaisista järjestelmistä.

Pohjola näytti muun muassa lukujärjestysten suunnitteluun käyttämäänsä WinhaResurssit-ohjelmaa. Muotoiluinstituutissa WinhaResurssit-ohjelmaa käytetään lukujärjestysten suunnittelussa, opettajien työaikasuunnittelussa sekä tilavarauksissa. Kävimme läpi, mitä eri osioita ohjelmassa on. Pohjola resursoi kohderyhmän, esimerkiksi multimediatuotannon neljännen vuosikurssin, lukuvuoden opintojaksoille muun muassa toteutukset, tunnit periodeittain, opettajat, opettajien kokonaistuntiresurssit ja tilatiedot. Hän sanoi, että vaikka tässä vaiheessa varataan tietty tila opintojaksolle, tila ei varsinaisesti ole vielä varattuna. Varauksen tekemisen esimerkkinä Pohjola näytti, miten hän tekee lukujärjestyksen. Lukujärjestykset suunnitellaan kohderyhmälle WinhaResurssit-ohjelman viikkonäkymäikkunalla. Ohjelma antaa virheilmoituksen, jos ollaan tekemässä päällekkäistä varausta.

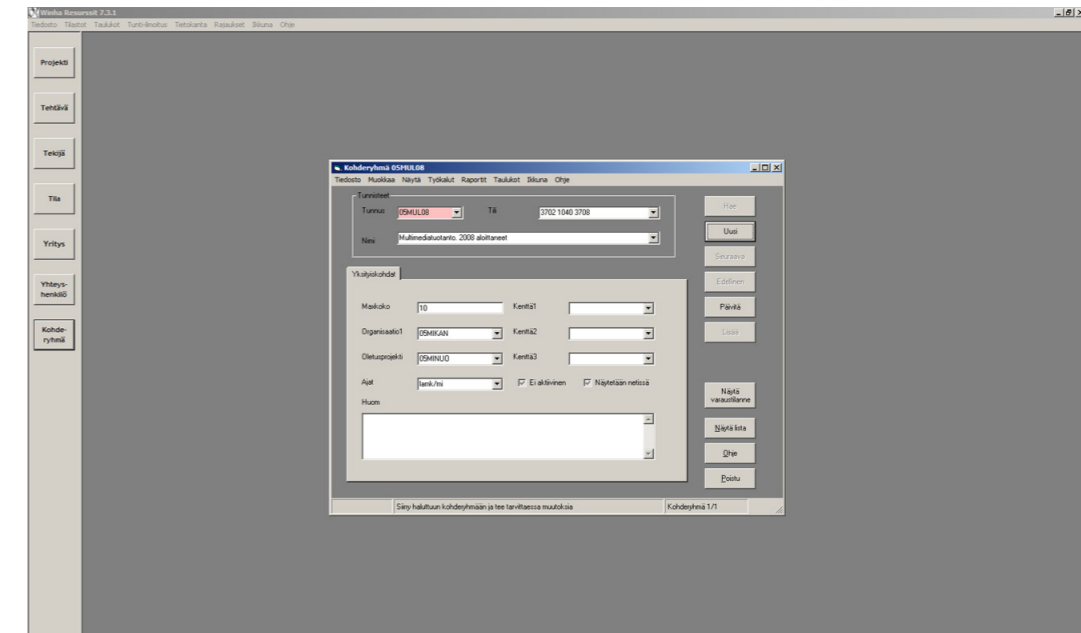
Ressu-lukujärjestyksen väline-osiossa voi selata päiväkohtaisesti mitä on varattu ja mille kellonajalle, mutta siellä ei ole ohjeita mistä tai kenen kautta varauksia voi tehdä.

Pohjola arvioi, että selainpohjainen varausjärjestelmä toimisi paremmin meidän yksikössämme. Koska viestinnän alalla on visuaalisia ihmisiä, visuaalinen varausjärjestelmä toimisi paremmin, kuin varausominaisuuden lisääminen WinhaResurssit-ohjelmaan.

WinhaResurssit-ohjelma toimii, jos sitä osaa käyttää. Se on kuitenkin vaikeakäyttöisen oloinen, kankea ja vanhanaikainen. Vanhanaikaisuudesta kielii myös ohjelman karkea ulkonäkö, joka muistuttaa Windows 1.0-versiota 80-luvulta (KUVA 3). Ohjelma vaatii käyttäjältä erilaisia koodeja eri vaiheissa. Koodien syöttäminen halutun toiminnon saavuttamiseksi, on nykyään käytössä enää lähinnä tekstipohjaisissa käyttöjärjestelmissä, kuten

Windowsin komentokehotteessa tai Mac OS X:n ja Linuxin terminaalissa. Graafisten käyttöliittymien yleistyttyä 80-luvulla ja osoitinlaitteiden kehittymisen myötä, tällaiset koodien syöttämiset ovat jääneet lähinnä ns. nörttien huviksi ja harmiksi. On huonoa käyttöliittymäsuunnittelua, jos ns. peruskäyttäjän pitää muistaa koodeja ja hänen täytyy syöttää niitä ohjelmalle, jotta hän saisi jonkun tehtävän tehtyä.

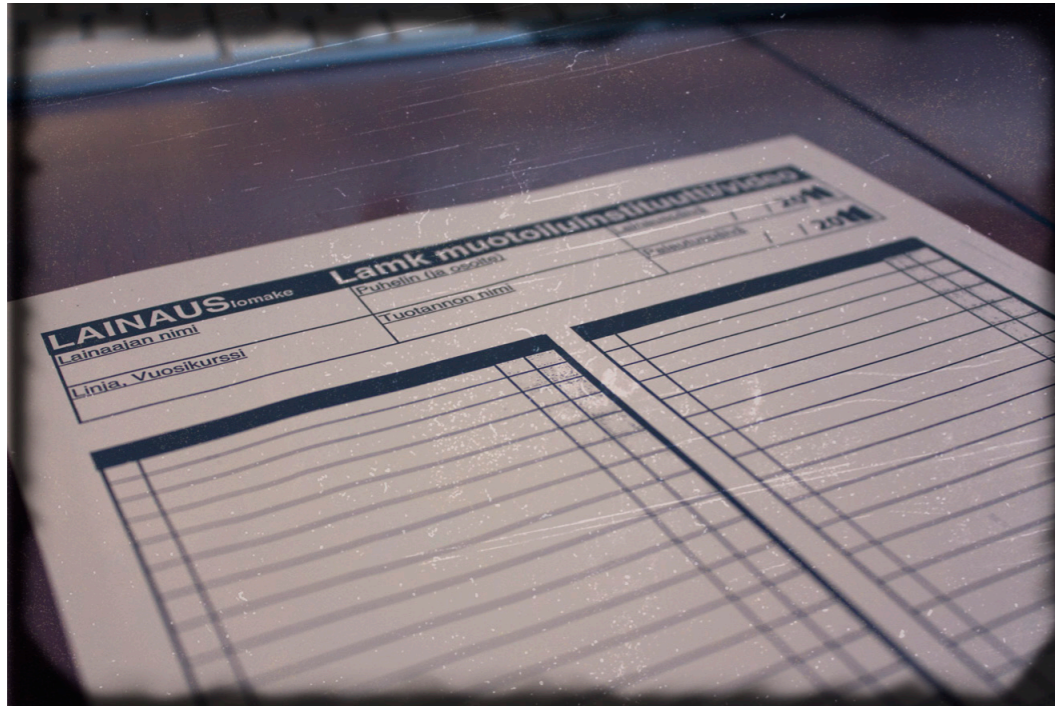
Yhteenvedona todettakoon, että WinhaResurssit-ohjelma ei sovi selainpohjaisen varausjärjestelmän pohjaksi, eikä sitä voi päivittää niin, että se sopisi käyttötarkoitukseen. Näin ollen uuden järjestelmän tekeminen on järkevämpää. Olimme Pohjolan kanssa yksimielisiä siitä, että selainpohjaiselle varausjärjestelmälle olisi tarvetta. (Pohjola 2012.)



KUVA 3. WinhaResurssit-ohjelma, kohderyhmän valinta.

Elokuva ja TV-ilmaisu

Elokuva- ja TV-ilmaisuuden erikoislaboratoriomestari Jouni Kärkinen kertoo, että heillä laitteiston ja tilojen varaus hoidetaan Excel-pohjaisen systeemin ja erillisen paperilomakkeen kautta (KUVA 4). Opiskelijat merkitsevät haluamansa laitteet tai tilat Excel-tiedostoon. Hakiessaan tavarat varastosta, opiskelijat täyttävät erikseen lainauslomakkeen. Huonona puolena hän pitää paperin käsin kirjoittamista. Varauslistan pitäisi myös pystyä tulostamaan kahtena kappaleena, toinen opiskelijalle ja toinen hänelle. Kärkinen kertoo, että usein opiskelijat muistavat vasta varastolla, mitä kaikkea muuta tarvitsevat. Tämän vuoksi lainauksen päivittäminen pitäisi onnistua helposti myös varastossa. Kärkinen kertoo Elokuva- ja TV-tuotannon varauslistan olevan enemmän tai vähemmän suuntaa antava varaston tilanteesta. Haasteena on myös kaksi fyysisesti erillään olevaa varastoa. (LIITE 1)



KUVA 4. Elokuva- ja TV-ilmaisuuden lainauslomake.

Multimediatuotanto

Multimediatuotannon nykyinen varaussysteemi perustuu opiskelijoiden täyttämiin kalenterivihkoihin (KUVA 5). Vihkoissa on sarakkeet kullekin laitteelle ja sille kuuluville varusteille. Opiskelija kirjoittaa nimensä ja puhelinnumeron sarakkeeseen niille päiville, jolloin hän haluaa ko. laitetta käyttää. Palauttaessaan laitetta, opiskelijan pitää huolehtia akut lataukseen ja tyhjentää muistikortit. Palautusta ei erikseen valvota palautustilanteessa.



KUVA 5. Multimediatuotannon varausvihko.

Multimediatuotannon laitteistosta vastaava tuntiopettaja Tommi Lius pitää suurimpana ongelmana nykyisessä järjestelmässä sitä, että liikaa jää opiskelijan vastuulle (LIITE 2).

Käytännössä opiskelijat huolehtivat itse varaamisesta, palauttamisesta ja esimerkiksi rikkoutuneesta laitteesta raportoinnista henkilökunnalle. Se ei olisi huono asia, jos homma toimisi jouhevasti, mutta se ei toimi (KUVA 6).



KUVA 6. Sekalaista tavaraa Multimediatuotannon editissä.

Nykyisessä systeemissä on ongelmana laitteen palauttaminen oikeaan paikkaan ja oikealla tavalla. Laitetta palautettaessa pitäisi laittaa akut lataukseen, formatoida muistikortit ja laittaa laitteet niiden omille paikoilleen. Usein laitteita kuitenkin lainataan ja palautetaan kiireessä, väsyneenä tai välinpitämättömällä asenteella. Liian helposti ajatellaan, että kyllä joku muu hoitaa asian. Joskus laitteita ei palauteta vihkoon merkittyyn aikaan mennessä, vaan oletetaan että varaajalle soitetaan jos laitetta tarvitaan.

Ongelmana on myös, että laitteiden paikat vaihtuvat. Laukuissa ei välttämättä aina ole samoja laitteita sisällä. Esimerkiksi kameralaukussa pitäisi olla kaksi akkua, laturi ja muistikortti, mutta aina näin ei ole. Lius kertoo myös, että nykyinen vihkojärjestelmä ei ole, eikä pysy ajan tasalla. Laitteita on tullut lisää, mutta niitä ei ole päivitetty kalenterivihkoihin. Jos laitteita tulee lisää, myös vihkoja tarvitaan lisää. (LIITE 2.)

Tilankäytön kannalta tämä on epäkäytännöllistä. Se hankaloittaa laitteiston varaamista myös käyttäjän kannalta, koska tiedot ovat hajallaan eri vihkoissa. Yhteen vihkoon eivät kaikki laitteet sovi.

Valokuvaus

Valokuvausosaston nykyinen laitteiston varaus- ja lainausjärjestelmä on Windows-sovellus (KUVA 7), joka on asennettuna ja toimii vain yhdellä tietokoneella. Järjestelmää käyttää vain erikoislaboratoriomestari Markku Reiniharju tai hänen alaisuudessaan kulloinkin oleva harjoittelija. Järjestelmässä on lainaus- ja palautus-, varasto-, varaus- sekä raportit-osio.

Lainassa olevat				
Nimike	Tarkenne	Merkki	Lainauspvm	P

KUVA 7. Valokuvausosaston lainausohjelman aloitusikkuna.

Lainaus tapahtuu lukemalla lainaajan henkilötiedot opiskelija- tai KELA-kortin viivakoodista. Jos lainaajan tietoja ei ole järjestelmässä, voidaan uuden käyttäjän tiedot manuaalisesti lisätä. Henkilötiedoissa näkyy henkilötunnus, etunimi, sukunimi, postiosoite, puhelinnumero ja luokka (KUVA 8). Kun henkilökortin viivakoodi on luettu, luetaan lainattavan laitteen viivakoodi, jolloin järjestelmä lisää laitteen tiedot lainaajan profiliin. Laitteen lainaustiedoissa näkyy kolmen viimeisimmän lainaajan henkilötiedot, lainaajan henkilötiedoissa lainaushistoriaa ei näy. Jos halutaan tarkastella tietyllä henkilöllä lainassa olevia laitteita, haetaan henkilön nimellä tietoja lainausosiosta. Käyttäjät eivät poistu automaattisesti järjestelmästä, vaan ne pitää yksitellen poistaa.

The screenshot shows a web form for entering borrower information. At the top left, there is a label 'Kelakoodi :'. To its right is a button 'Syötä kelakoodi'. Below this is a section titled 'Lainajan tiedot' (Borrower information) containing several input fields: 'Sukunimi' (Surname) and 'Etunimi' (First name) in a two-column layout, 'Lähiosoite' (Address), 'Postinumero' (Postcode) and 'Postitoimipaikka' (Post office) in a two-column layout, 'Puhelin' (Phone), and 'Luokka' (Class). To the right of these fields is a vertical column of buttons: 'Hae tiedot' (Search), 'Lisää tiedot' (Add), 'Muuta tiedot' (Edit), 'Poista tiedot' (Delete), and 'Tyhjennä' (Clear). At the bottom right of the form area are two buttons: 'Lainat' (Loans) and 'Varaukset' (Reservations). Below the form is a table with three columns: 'Sukunimi', 'Etunimi', and 'Luokka'. The table is currently empty.

KUVA 8. Lainajan tiedot Valokuvausosaston lainausohjelmassa.

Laitteen palautuksen ja lainaamisen seuraavalle henkilölle voi suorittaa myös lennosta. Palautettaessa ohjelma ei kerro kuka palautti ja mitä, vaan pelkästään että palautus on suoritettu onnistuneesti.

Varaston tilannetta tarkastellaan syöttämällä ohjelmaan hakutietoja, jolloin ohjelma tulostaa hakutermejä vastaavan listan laitteista. Laitteiden yksityiskohtaiset tiedot avautuvat uuteen ikkunaan. Laitteen tietoja ovat viivakoodi, nimike, tarkenne, merkki ja valintaruutu joka kertoo onko laite lainassa vai ei. Tarkenteessa on joskus laitteen sarjanumero. Jos tarkenteen teksti on pitkä, sarjanumero ei näy tekstikentän rajallisen koon vuoksi. Varasto sekä huolto, korjaus ja poistot ovat järjestelmän käyttäjiä. Kun laite tulee takaisin esimerkiksi huollosta, se palautetaan järjestelmässä samalla tavalla, kuin se olisi ollut lainassa jollakin henkilöllä.

Varaus ei käytännössä toimi ollenkaan. Varaussysteemi kyllä tunnistaa laitteet viivakoodien perusteella, mutta laitteen ollessa lainassa se ei kerro, että varastossa on muita samanlaisia laitteita vapaana. Tästä johtuen varaukset merkitään paperikalenteriin.

Raportit-osiosta voi valita joko lainat, varaukset, myöhästyneet lainat tai kaikki lainattavat välineet. Ohjelma avaa Excel-taulukkolaskentaohjelman ja vie tiedot taulukkoon. Excelin kautta voi selata tietoja, mutta ei muutaa niitä.

Laittevarausjärjestelmän hyviä puolia on viivakoodin lukumahdollisuus ja se, että samassa järjestelmässä on varaston seuranta sekä huolto- ja korjausseuranta. Viivakoodin haittapuolena on, että se lähtee helposti pois mekaanisen kulumisen johdosta. Kuluneen viivakoodin vaihtaminen on hankalaa. Käytännössä tehdäänkin uusi viivakoodi, jolloin myös laitteen varaushistoria nollautuu. Uudelleen koodittamisen pitäisi olla helpompaa, koska koodeja joutuu vaihtamaan usein. Valokuvauslinjan viivakoodiprintteristä ei Reiniharjun mukaan ole käytännössä mitään hyötyä, koska sillä printattu viivakoodi lähtee heti pois. Viivakoodin lisäksi laitteessa tai laitteen osassa pitää olla tunnistenumero eli sarjanumero, joka löytyy myös järjestelmästä.

Sarjanumeroa tarvitaan etenkin silloin, kun viivakoodi on kulunut pois. Viivakoodin käyttö pienissä tai pehmeissä tavaroissa on ongelmallista. Joitakin laitteita lainataan useimmiten pakettina, jolloin esimerkiksi laukussa voi olla viivakoodi ja sen takaa löytyy tiedot siitä, mitä laukun sisällä kuuluu olla. Laukun sisällä olevissa laitteissa voi myös olla omat viivakoodit.

Valokuvauslinjan nykyisen varaus- ja lainausjärjestelmän huonoja puolia ovat järjestelmän sisäiset laitteiden, käyttäjien ja varaston selausominaisuudet. Selkeitä puutteita ovat laitteiden varausmahdollisuus, automaattiset ilmoitukset laina-ajan umpeutumisesta sekä laina-ajan muuttaminen. Laina-aika on aina viikon mittainen.

Reiniharjun mukaan järjestelmän ylläpito ja lainausten hoito on työlästä ja vaatisi yhden henkilön kokopäiväisen työpanoksen, jotta se toimisi jouhevasti. (LIITE 3.)

KESKEISIMMÄT TOIMINNOT

Tässä luvussa kartoitan selainpohjaisen varausjärjestelmän tärkeimmät ominaisuudet. Luku pohjautuu **Nykytilanteen kartoitukseen**, sekä alan yleiseen tutkimukseen.

Selainpohjaisen varausjärjestelmän tarkoituksena olisi yksinkertaistaa, yhtenäistää ja helpottaa niin laitteiston varausta ja seuranta, kuin varastokirjanpitoakin. Varauksen pitäisi pystyä tekemään etäältä, koska aina ei pääse fyysisesti paikalle vaikkapa sairaudesta johtuen. Tällöin joku muu joutuu tekemään varauksen toisen puolesta.

Multimediatuotannon tuntiopettaja Tommi Lius sanoo, että järjestelmän käyttöopastus olisi sisällytettävä opetussuunnitelmaan ensimmäiselle vuodelle, koska jo ensimmäisen vuoden opiskelijat saavat lainata ainakin piirtotabletteja. Puolen päivän koulutus syksyn aikana riittäisi antamaan tarvittavat tiedot järjestelmän käyttämiseksi. (LIITE 2.)

Kuten jo luvussa **Beta-testaus ja virheenkorjaus** mainitsin, perehdyttämisestä voitaisiin päättää kunkin pääaineen sisällä, koska Viestinnän koulutusohjelman pääaineiden käytännöt poikkeavat toisistaan.

Järjestelmässä pitäisi olla käyttöönottilanteessa näkyvissä pikaopas, joka kertoisi järjestelmän toiminnan pääpiirteet. Sama pikaopas tulisi näkyviin myös, jos käyttäjä ei ole kirjautunut järjestelmään esimerkiksi viimeisen kolmen kuukauden aikana. Tämä olisi hyvä senkin vuoksi, että jos järjestelmän käyttö opetetaan heti koulun alussa, se unohtuu hyvin helposti kaiken muun informaatiotulvan alle. (LIITE 2.)

Pikaopas ei korvaa hyvää dokumentaatiota tai muita aputoimintoja, vaan nimenomaan näyttää nopeasti keskeisimmät toiminnot.

Järjestelmän käyttäjät, laitteet ja tilat

Käyttäjät

Käyttäjä voisi varata laitteita käyttöönsä internetin välityksellä tietylle ajanjaksolle. Käyttäjällä on [käyttäjä]tunnus, salasana, yksilöintitunnus eli ID, käyttäjätaso, lainaushistoria ja status, joka kertoo voiko hän lainata laitteita vai ei.

Tommi Liuksen kanssa ideoimme, että käyttäjätunnuksissa pitäisi olla voimassaoloaika, ettei käyttäjiä tarvitsisi käsin poistaa valmistumisen jälkeen. Järjestelmä voisi antaa ilmoituksen käyttäjälle, että valmistuminen lähestyy ja järjestelmä aikoo poistaa tunnuksen. Samassa yhteydessä voisi olla kohta, josta käyttäjä pääsee lähettämään ylläpidolle viestin, jos valmistumisajankohta on muuttunut. Ensimmäisen vuoden opiskelijan käyttäjätunnuksen voimassaoloaika olisi oletuksena neljä vuotta, mutta sitä voisi muuttaa tarvittaessa. Tämä muutosmahdollisuus toimisi hyvin myös lyhytaikaisten käyttäjätunnusten kohdalla, kuten vaihto-opiskelijoilla ja vierailevilla luennoitsijoilla. (LIITE 2.)

Erikoislaboratoriomestari Markku Reiniharju, kertoo, ettei hän välttämättä pidä hyvänä ideana automaattista poistamista, sillä joistakin opiskelijoista saattaa tulla opettajia tai yhteistyökumppaneita, jotka pitäisi sitten jälkikäteen lisätä uudestaan järjestelmään. Reiniharju sanoo, että olisi hyvä jos järjestelmän pystyisi linkittämään opiskelijarekisteriin, josta järjestelmä hakisi automaattisesti opiskelijan tiedot, ettei niitä tarvitsisi käsin lisätä järjestelmään. (LIITE 3.)

Mielestäni opiskelijarekisteriin linkittäminen olisi erinomainen asia. Sen sijaan Reiniharjun maalaama uhkakuva automaattisen poistamisen haitoista on korjattavissa ohjelmasuunnittelulla, kuten Liuksen kanssa edellä ideoimme. Ylläpitäjä voisi vaihtaa käyttäjän roolin järjestelmässä, jos opiskelijasta olisi tulossa yhteistyökumppani tai työntekijä.

Ylläpitäjä voisi hallinnoida käyttäjiä ja laitteita yksilöinä tai ryhminä, valvoa keskustelua ja hyväksyä tai hylätä varauksia. Ylläpitäjällä on tunnus, salasana ja ID.

Laitteet

Laitteella olisi ID, vaatimustaso/laiteryhmä (vrt. käyttäjätaso), status, lainaus- ja viestihistoria.

Laitteella voi olla myös laitteen osia, joilla on ainakin osittain samanlaisia ominaisuuksia kuin laiteella. Esimerkiksi järjestelmäkameran objektiivilla olisi oma ID, status, lainaus- ja viestihistoria.

Tommi Lius pitää laitteiston tunnistamista haasteena. Opiskelijat eivät aina tiedä laitteen nimeä tai mallia. Tähän hän kehottaa kiinnittämään huomiota järjestelmän suunnittelussa. Järjestelmässä olevien laitteiden tiedoissa pitäisi olla laitteiden ID:t, kuvia, paino, väri ja niin edelleen, jotta tunnistaminen olisi mahdollisimman helppoa, eikä virheitä pääsisi syntymään. Rikkoutuneista laitteista ei mielellään raportoida henkilökunnalle. Lius arvelee että opiskelijat ehkä pelkäävät joutuvansa korvaamaan hajonneen laitteen. Todellisuudessa opiskelija joutuu äärimmäisen harvoin korvaamaan mitään, mutta olisi tärkeää tietää hajonneista laitteista, että ne voitaisiin ottaa pois varauslistalta ja toimittaa huoltoon. Ei saisi käydä niin, että hajoaminen huomataan vasta kun laitetta ollaan käyttämässä seuraavan kerran. Tulevassa järjestelmässä pitäisi olla houkutteleva ja rohkaiseva palautteenantosysteemi siltä varalta, että laite menee rikki. Siinä pitäisi olla selkeä maininta, että käyttäjää ei aseteta edesvastuuseen hajonneesta laitteesta, mutta miten tärkeää on ilmoittaa asiasta labrimestarille. Syynä nykyisen systeemin toimimattomuuteen Lius pitää riittämätöntä perehdyttämistä. (LIITE 2.)

Markku Reiniharju kertoo, että heidän laitteitaan lainataan myös muiden linjojen, esimerkiksi Elokuva- ja TV-ilmaisun opiskelijoille (LIITE 3). Tilanne on tullut eteen myös muilla linjoilla, joten tämänkin vuoksi yhteisen järjestelmän tekeminen on mielestäni perusteltua.

Tilat

Tilalla olisi ID, käyttötarkoitus, varustelu ja varaushistoria. ID:nä voisi käyttää esimerkiksi tilan numeroa, joka löytyy luokkien ovien pielistä. Tunnistamisen helpottamiseksi voisi näkyvillä olla myös tilojen vakiintuneet nimitykset, kuten editti, pommari ja niin edelleen. Käyttötarkoitus ilmaisisi heti mihin käyttöön luokka tai tila on varattu. Tilänäkymässä voisi olla sekä kalenterinäkyvä että koulun pohjapiirros. Pohjapiirros olisi interaktiivinen siten, että sitä klikkaamalla voisi valita tilan, jonka jälkeen kalenterista valittaisiin aika, jolloin tila halutaan varata. Kun tila on klikattu, johonkin sivupalkkiin voisi tulla tiedot tilasta myös tekstuaalisena. Tekstistä voi tarkastaa, että kyseessä on oikea tila. Varustelu kertoo tilan normaalisti sisältämät varusteet, esimerkiksi piirtoheitin, valkotalu ym. voivat kuulua vakiovarusteluun. Jos tilojen varusteet halutaan piilottaa ns. suurelta yleisöltä, voisi varusteiden lisäys- ja muutososioon luoda mahdollisuuden valita käyttäjäryhmiä siten, että vain tietyt käyttäjäryhmät voivat nähdä mitä kaikkea tila sisältää.

Käyttäjätasot

Koska varattavien laitteiden kirjo on laaja, niin ominaisuuksien kuin hintojenkin puolesta, järjestelmässä pitäisi olla systeemi, jolla voisi luoda käyttäjätasoja, joilla olisi oikeudet tiettyihin laitteisiin tai laiteryhmiin. Kun opiskelija on käynyt tietyn kurssin, hänellä on valmius käyttää tiettyjä laitteita tai laiteryhmiä. Näin hän pääsee tasolle, jolla on oikeus lainata näitä laitteita. Järjestelmän ylläpitäjät vastaavat käyttäjätasojen antamisesta opiskelijoille tai ryhmille.

"Käyttäjärühmät/käyttöoikeustasot kuulostavat järkevältä. On tärkeää, ettei kuka tahansa pääse lainaamaan mitä tahansa."

~ Elokuva- ja TV-ilmaisun erikoislaboratoriomestari Jouni Kärkinen. (LIITE 1.)

Käyttäjärühmät ja yksilöt

Käyttäjiä pitäisi pystyä hallinnoimaan ryhmänä ja yksilöllisesti. Esimerkiksi jos tietty luokka on suorittanut D-tason kurssin, joka oikeuttaa D-tasoon kuuluvien laitteiden käyttämisen, pitää järjestelmän ylläpito-oikeudet omaavalla henkilöllä olla mahdollisuus ylentää kaikki ko. luokan opiskelijat kerralla D-tasoon. Vastaavasti jos joku opiskelija on ryhmäänsä ylempänä tai alempana, eli hän ei ole esimerkiksi suorittanut vaadittavaa kurssia, pitää olla mahdollista ylentää tai alentaa hänen käyttäjätasoaan muista irrallisena.

Laitteiden ja laitteen osien linkittäminen

Usein laitteisiin liittyy osia, jotka pitää pystyä varaamaan erikseen tai pitää muistaa varata ne, jotta laitetta voidaan käyttää. Esimerkiksi järjestelmäkamera on rungosta, objektiivista, akusta ja muistikortista koostuva kokonaisuus. Tämän vuoksi tiettyjen osien lainaaminen pitäisi linkittää yhteen. Olisi myös kätevää, jos varaajalle ilmoitettaisiin, mahdollisesti tarvittavista lisälaitteista. Esimerkiksi järjestelmäkameraa lainattaessa, järjestelmä ilmoittaisi, että

myös kameran jalusta voisi olla tarpeellinen. Käyttäjä voisi klikkaamalla lisätä myös jalustan varauslistalle. Kyseinen ominaisuus ehkäisi myös sitä, että vasta varastolla muistetaan mitä kaikkea voisikaan tarvita, jonka Jouni Kärkinen kertoo olevan heillä yleistä (LIITE 1).

Multimediatuotannon tuntiopettaja Arto Kutvonen ideoi, että järjestelmän pitäisi ehdottaa käyttäjälle vastaavanlaisia laitteita siinä tapauksessa, kun haettua laitetta ei ole yhtään vapaana tai varaaminen ei onnistu. Jos esimerkiksi käyttäjä haluaa varata Canonin järjestelmäkameran, järjestelmä ilmoittaisi ettei niitä ole vapaana, mutta vastaavia kameroita löytyy Nikonilta. (Kutvonen 2012.)

Varausaika

Varausaika valittaisiin kätevästi kalenterista ns. maalamalla päiviä. Varausaikaa rajoitettaisiin hallintapaneelista pääkäyttäjän toimesta, jotta esimerkiksi opiskelijat eivät voisi varata laitteita liian pitkäksi ajaksi tai vaikkapa kesäloman aikana. Pääkäyttäjällä tulisi olla myös oikeus laittaa peruskäyttäjiä karenssiin tietyksi ajaksi, vaikkapa siitä syystä, että edellinen palautus on ollut myöhässä. Kun karenssiaika olisi ohi, järjestelmä palauttaisi lainausoikeuden automaattisesti.

Tommi Lius, arvelee että olisi hyvä, jos järjestelmä ilmoittaisi sähköpostilla tai tekstiviestillä, että laina-aika on menossa umpeen, ja kertoisi mitä tapahtuu, jos palautus tapahtuu myöhässä. Samassa viestissä voi kertoa, että laina-aikaa voi pyytää lisää laboratoriomestarilta. (LIITE 2.)

Laina-ajan jatkamisenkin voisi mielestäni hoitaa suoraan järjestelmästä.

Markku Reiniharju pitää sähköposti-ilmoitusta hyvänä ideana. Samalla Reiniharju kuitenkin huomauttaa, että kaikki eivät lue sähköpostia. Lisäksi hän kertoo, että koulumme järjestelmä poistaa automaattisesti neljän vuoden kuluttua sähköpostitilit, ellei opiskelija erikseen ilmoita, että opiskelu jatkuu. Tämä on hänen mielestään hyvin hankala systeemi, koska usein juuri silloin usein pitäisi ehdottomasti saada yhteys opiskelijoihin. Reiniharjun mukaan sähköposti-ilmoituksen piiristä pitäisi saada osa kalustosta pois. Tämän voisi toteuttaa vaikka siten, että järjestelmässä olisi laitteen asetuksissa valintaruutu ilmoituksen lähettämisestä.

Reiniharju on sitä mieltä, että nykyään Facebookin kautta saa ihmiset parhaiten kiinni. Mielestäni Facebook-ilmoitus on mielenkiintoinen idea, mutta sen toteuttaminen käytännössä voi olla hankalaa. Se tarkoittaisi Facebook-applikaation tekemistä ja sitä että käyttäjien pitäisi asentaa applikaatio omaan profiliinsa saadakseen ilmoitukset vanhenevista lainoista. Idea on mielenkiintoinen ja sitä voi mielestäni pitää avoinna ainakin lisäoptiona. Ongelmana tässä on se, että kaikilla ei ole Facebook-tiliä ja kaikki eivät välttämättä halua asentaa ko. lisäapplikaatiota profiliinsa. (LIITE 3.)

Todettakoon että sellaista järjestelmää tai väylää on tuskin olemassa, joka saavuttaa sataprosenttisesti kaikki käyttäjät ja vaikka viesti myöhästymisestä saataisiinkin käyttäjälle perille, se ei silti ole tae siitä, että itse laite palautetaan määräaikaan mennessä, sovittuun paikkaan ja asiallisessa kunnossa. Sellaista järjestelmää, joka tähän pystyy, ei yksinkertaisesti ole eikä voi pelkillä ohjelmasovelluksilla tehdä.

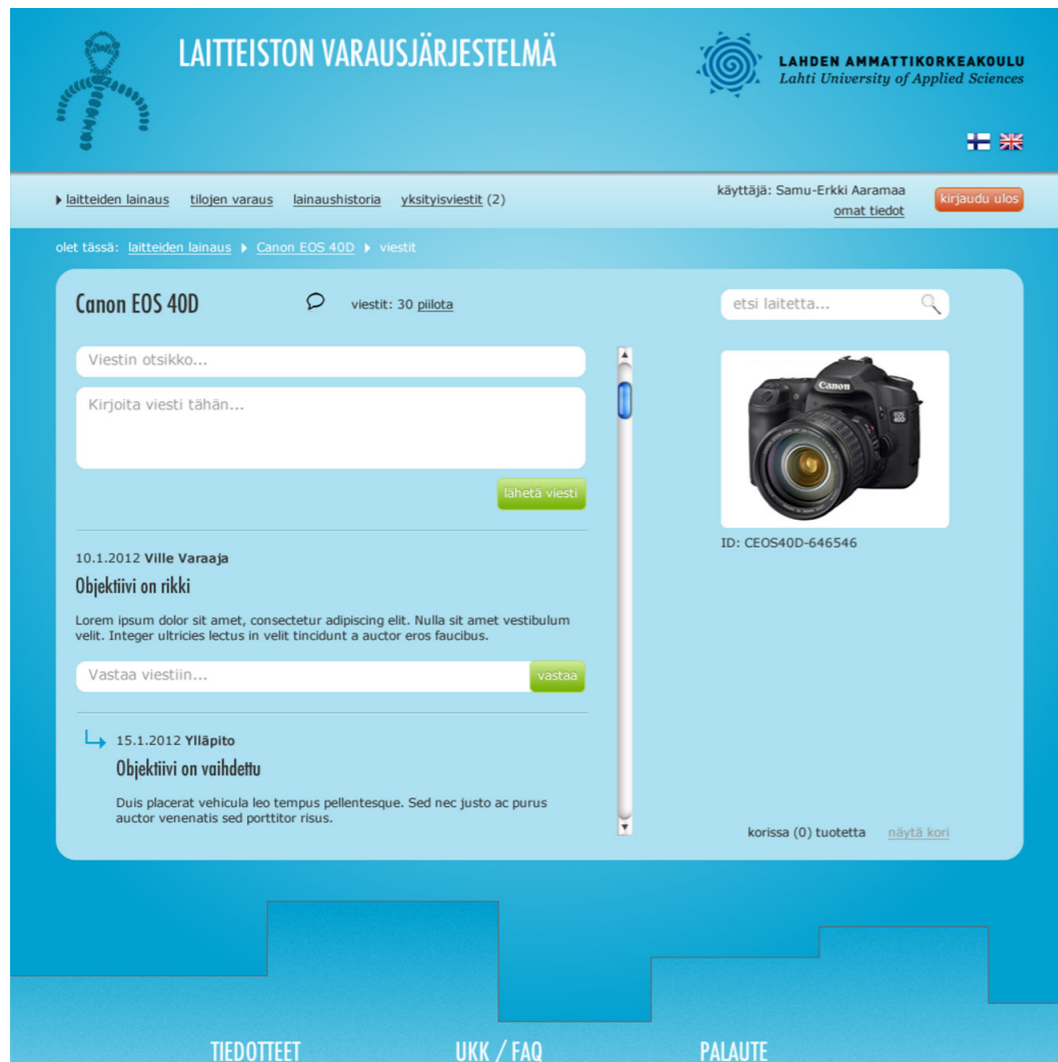
Varauksen hyväksyminen

Jouni Kärkinen ideoi, että varauksen vahvistaminen voisi mennä ylläpitäjän kautta. Jos vahvistaminen menisi ylläpitäjän kautta, varauksissa voisi olla ns. arvojärjestys, joka tarkoittaa sitä, että päällekkäisissä varauksissa laboratoriomestarit/opettajat hyväksyvät varaukset käyttötarkoituksen perusteella päällekkäisissä varauksissa. Jokainen voisi tehdä varauksen koska tahansa, myös päällekkäin, mutta henkilökunta päättäisi kuka lopulta lainaa tavaran tai varaa tilan käyttöönsä. Laitteen varauksessa pitäisi näin ollen olla alusvetovalikko, josta käyttötarkoituksen voisi valita tai tekstikenttä johon sen voisi kirjoittaa. Esimerkkitapaus: Henkilö A varaa kameran kuvataksaan kissaansa huvin vuoksi. Henkilö B tekee varauksen samalle ajalle, koska on sovittu että hän menee kuvaamaan yritys yhteistyöprojektiin liittyviä kuvia. Labramestari/opettaja hyväksyy henkilön B varauksen, koska tällä on tärkeämpi käyttötarkoitus. Samalla henkilölle A menee tieto, että hänen varauksensa on hylätty ja mahdollisesti ehdotuksena seuraavat mahdolliset varausajat ko. laitteelle. (LIITE 1.)

Tommi Lius pitää tällaista tärkeysjärjestysperustaista lainaussysteemiä hyvänä ideana.

Kommunikointi

Selainpohjaisessa varausjärjestelmässä tulisi olla mahdollisuus ilmoittaa laitteesta havaitusta virheestä tai ongelmasta, tai kertoa niksejä, joita on huomannut laitetta käytettäessä. Kuhunkin laitteeseen tai sen osaan liittyisi siis kommentointimahdollisuus (KUVA 9).



KUVA 9. Laitteiden kommentointi helpottaisi ongelmien raportointia.

Pääkäyttäjät moderoisivat viestejä hyväksymällä, hylkäämällä, muokkaamalla tai poistamalla niitä. Järjestelmän asetuksissa pitäisi olla mahdollisuus valita, menevätkö viestit läpi suoraan vai moderoinnin kautta, täten asetusta voitaisiin muuttaa tarvittaessa. Väärinkäytösten välttämiseksi käyttäjille voisi perehdytyksen yhteydessä kertoa, että kommunikoinnin väärinkäyttö tai ilkivalta voi johtaa karenssiin.

Laitteen kommentointimahdollisuuden lisäksi järjestelmässä voisi olla mahdollisuus lähettää yksityisviestejä. Ylläpito voisi käyttää tätä käyttäjien kanssa kommunikointiin, mutta ehkä myös käyttäjät voisivat lähettää keskenään viestejä. Tämä voisi olla kätevää, jos esimerkiksi käyttäjän varaus on hyväksytty ja hän haluaisi sopia laitteen sen hetkisen lainaajan kanssa palautusajankohdasta.

Omista asetuksista tulisi voida valita, lähettäkö järjestelmä esimerkiksi sähköpostiviestin, kun käyttäjälle on tullut viesti järjestelmän sisällä.

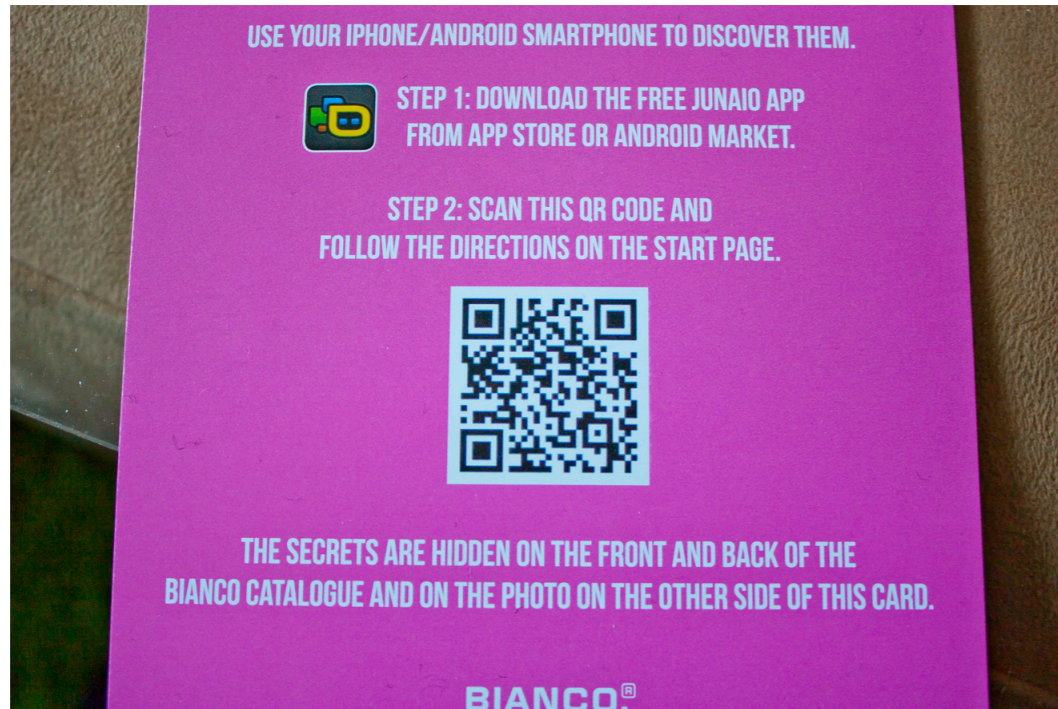
Omat tiedot

Kullakin käyttäjällä tulisi olla mahdollisuus selata omaa varaushistoriaansa, nähdä järjestelmän tallentamat tiedot käyttäjästä ja mahdollinen status ja käyttäjätaso. Status näyttäisi voiko käyttäjä tehdä varauksia vai onko hän karenssissa. Karenssista pitäisi nähdä karenssin syy ja aika.

Tommi Lius avasi keskustelun käyttäjäkuvakkeista. Tulimme siihen tulokseen, että ne voisivat olla järjestelmässä pakollisia. Käyttäjän kuva helpottaisi labramestaria tunnistamaan käyttäjiä, se nopeuttaisi järjestelmän käyttöä. Jos taas pitää etsiä tietty henkilö fyysisesti, voisi tarkastaa järjestelmästä minkä näköistä henkilöä etsitään. Multimediatauotannon koulutusohjelmassa ensimmäisen vuoden opiskelijat ottavat nykyäänkin kuvat toisistaan, joista tehdään nimi-kasvo-lista opettajille. Näitä kuvia voisi käyttää myös järjestelmässä. (LIITE 2.)

Viivakoodi tai QR-koodi

Laitteet varustettaisiin mahdollisuuksien mukaan viivakoodeilla tai QR-koodeilla (engl. Quick Response Code) (KUVA 10) laitteiden yksilöimiseksi sekä lainauksen ja varaston hallinnan helpottamiseksi.



KUVA 10. QR-koodi mainoksessa.

Ideoimme Markku Reiniharjun kanssa myös RFID-sirun (engl. Radio Frequency IDentification) upottamista laitteisiin. Hänen mielestä se olisi erittäin hyvä juttu. RFID-sirun asentamisen ja lukulaitteen kustannukset ovat kuitenkin toistaiseksi iso kysymysmerkki. (LIITE 3.)

Laiteryhmät ja laitekohtaiset ominaisuudet

Kuten aiemmin olen sivuuttanut, laitteilla tulisi olla omia ominaisuuksia ja niiden tulisi myös kuulua laiteryhmisiin. Omiin ominaisuuksiin voisi kuulua saatavuus eli onko laite varattavissa määrättyä aikana vai onko se jo varattu tai huollossa. Laiteryhmät kokoaisivat laitteet ryhmiin, joita vastaavaan tasoon kuuluvat käyttäjät voisivat lainata. Esimerkiksi tason D käyttäjä ei voisi lainata laiteryhmän B laitteita, mutta laiteryhmän E laitteita hän voisi varata.

Tommi Liuksen näkemyksen mukaan selainpohjaisessa varausjärjestelmässä pitäisi olla jonkinlainen manuaali ainakin tärkeimmille laitteille. Myös käyttövinkit, linkki valmistajan FAQ-sivulle sekä käyttö- ja lainauskäytännöt olisi hyvä olla esillä laitetiedoissa. Myös yleiset käyttöehdot tulisi olla luettavassa muodossa ja pituudessa sekä helposti löydettävissä järjestelmästä.

Tietokoneiden ja muiden ns. kiinteiden laitteiden laitteiden sekä tilojen varaus pitäisi Liuksen mukaan toimia saman järjestelmän kautta ja yksinkertaisesti. (LIITE 2.)

Laajennettavuus

Tommi Liuksen mielestä lainattavien laitteiden laukuissa voisi tulevaisuudessa olla viivakoodi, joka sisältää tiedon laukun sisällöstä. Käyttäjän pitäisi pystyä lukemaan laukun viivakoodi esim. kännykällä, jotta hän näkisi mitä laitteita laukkuun kuuluu. Tällainen systeemi on ainakin osittain käytössä Valokuvauslinjalla. (LIITE 2.)

"Selainpohjaisessa varausjärjestelmässä voisi olla sellainen ominaisuus, että kurssin vetäjä voisi luoda "tuotannon" ja sille aikajanan. Opiskelija valitsisi tuotannon, jota varten hän lainaa tavaroita."

~ Jouni Kärkinen.

Tässä yhteydessä Kärkinen kuitenkin varoittaa byrokraatiasta, ettei järjestelmästä tule liian kankea käyttöä. (LIITE 1.)

JÄRJESTELMÄN TOTEUTTAMISESTA

Tässä luvussa pohdin järjestelmän toteuttamiseen tarvittavia resursseja. Esittelen myös hyviä käytänteitä web-sovelluksen luomisessa.

Resursointi

Järjestelmän toteuttaminen vaatii luonnollisesti ihmisiä, eli osaamista sekä aikaa ja rahaa.

Työryhmän koko on suhteutettava projektiin. Ryhmässä on hyvä olla ainakin ryhmän vetäjä ja graafisia suunnittelijoita, joilla on mielellään ymmärrystä internetsovellusten erityispiirteistä, sekä ohjelmoijia, joilla on tieto-taitoa niin front end, kuin back end ohjelmoinnista.

* Front end ohjelmointi keskittyy siihen mitä näkyy loppukäyttäjälle (HTML, JavaScript, CSS), back end ohjelmointi siihen mitä tapahtuu kulissien takana, tätä on esimerkiksi palvelinpuolen ohjelmointi (PHP, MySQL). Isommissa ja monimutkaisemmissa projekteissa front end ja back end-ohjelmointi on jaettu eri osa-alueisiin erikoistuneille ohjelmoijille, pienemmissä projekteissa kummankin puolen hoitaa usein sama tai samat henkilöt. Se kumpaa tapaa käytettäisiin tässä tapauksessa, tulisi miettiä järjestelmää pohtivan työryhmän toimesta.

Aikaa on syytä varata riittävästi tutkimuksiin, suunnitteluun, toteutukseen ja testaamiseen. Prosessin läpimenoaika riippuu työryhmän koosta ja yhteistyökyvystä, sekä monista ulkoisista tekijöistä, joita on mahdotonta tässä vaiheessa ennustaa. Ensimmäisiä prototyyppejä testaukseen voisi saada parissa kuukaudessa, mutta julkaisuvalmiin ohjelman tekemiseen voi mennä vuosikin. Myös järjestelmän ylläpitoon ja päivittämiseen on syytä varata resursseja. Kuten aiemminkin olen huomauttanut, niin hyvää järjestelmää ei voi tehdä, etteikö sitä tarvitsisi jossakin vaiheessa päivittää. Maailma ja tekniikka kehittyvät jatkuvasti, joten myös järjestelmän on pysyttävä ajan hermolla.

Hyvään suunnitteluun menee aikaa, mutta jos tässä vaiheessa yritetään mennä sieltä mistä aita on matalin, se kostaustuu moninkertaisesti erilaisten korjausten muodossa myöhemmin. Hyvä suunnittelu on kaiken perusta. Suunnittelussa on syytä ottaa huomioon käyttäjäryhmä, ja tehdä yhteistyötä

oikeiden käyttäjien kanssa. Parhaimmatkaan oletukset eivät korvaa faktaa, joka saadaan oikeiden käyttäjien kanssa tehdyistä kokeista ja kyselyistä. Rahaa on myös syytä budjetoida riittävästi. Toimivat ratkaisut auttavat kaikkia osapuolia. Rahallista resursointia voi hyvinkin ajatella tulevaisuuden sijoituksena, se mitä järjestelmään menee nyt, tulee takaisin tulevaisuudessa yhtenäisten käytäntöjen ja ajansäästön muodossa. Kannustavan ja helpon palautteenantosysteemin ansiosta huollosta tulisi järjestelmällisempää, joten laitteet pysyisivät oletettavasti paremmassa kunnossa. Parempikuntoiset laitteet toisivat säästöjä investoinneissa.

Suositteleni sovelluskehukset (engl. framework)

Tässä osiossa pyrin esittelemään hyviä käytänteitä selainpohjaisen sovelluksen suunnittelussa ja toteutuksessa. En mene liian syvälle yksittäisiin teknologioihin, mutta koetan valottaa selkokielellä, mitä on hyvä ottaa huomioon ja mitä tekniikoita kannattaa käyttää. Varsinaisessa toteutuksessa tämä on korkeintaan muistilista sovelluksen kehittäjälle, sillä jos nämä tekniikat ovat hänelle entuudestaan täysin vieraita, on syytä miettiä toteuttajan vaihtamista.

Mikä on sovelluskehys ja miksi käyttää sellaista?

Sovelluskehys on kokoelma työkaluja, kirjastoja ja hyviä toimintatapoja. Sovelluskehukset pyrkivät abstrahoimaan rutiinitehtäviä ja tekemään niistä keskeisiä moduuleja, joita voidaan käyttää uudelleen sovellusten eri osissa. Tavoitteena on, että suunnittelija tai web-kehittäjä voisi keskittyä tehtäviin, jotka ovat projektissa ainutlaatuisia, sen sijaan että hän tuhlaisi aikaa yksinkertaisten, yleisten ja maailmassa jo ratkaistujen ongelmien kanssa painimiseen. (Croft 2011.)

Sovelluskehys nopeuttaa ohjelmoinnin työnkulkua, koska sovelluskehystä käytettäessä ei tarvitse kirjoittaa niin paljon koodia käsin tai keksiä pyörää uudelleen. Se myös auttaa tekemään siistimpää koodia, jota on helpompi

lukea ja huoltaa myöhemmin. (Dorf 2011.)

Sovelluskehys on joka lähtöön ja useille ohjelmointikielille. Tämän projektin puitteissa huomioon otettavia sovelluskehyskiä ovat JavaScript-, PHP- ja CSS-kielille tehdyt kehykset.

Tärkeintä on valita edes joku sovelluskehys ja mielellään sellainen, joka on riittävän tunnettu, sillä se helpottaa osaavan tekijän löytymistä myöhemmin jos tarve niin vaatii.

Sovelluskehys JavaScript ohjelmointikielille

JavaScriptille on olemassa kymmeniä erilaisia sovelluskehyskiä. Buckley listaa viisi parasta JavaScript sovelluskehystä ja kirjastoa seuraavasti:

1. jQuery <http://jquery.com>
käyttöliittymäsovellukset <http://jqueryui.com>
2. Prototype <http://prototypejs.org>
3. YUI <http://developer.yahoo.com/yui>
4. Dojo <http://dojotoolkit.org>
5. MooTools <http://www.mootools.net>

(Buckley 2011.)

Yllämainituista sovelluskehyskiä olen käyttänyt MooToolsia ja jQueryä. Valitsisin henkilökohtaisesti selainpohjaisen varausjärjestelmän tueksi jQuery-sovelluskehyskiä. Tämä valinta perustuu omaan osaamiseeni ko. sovelluskehyskiä puitteissa. Jos tarkastelen asiaa objektiivisesti, kysymykseen tulisi myös YUI-sovelluskehys sen laajuuden vuoksi. YUI:n etuna jQuery:n verrattuna on se, että YUI tarjoaa käyttöliittymäsovellukset sisäänrakennettuna, kun taas jQuery:ssä täytyy ladata erikseen jQuery UI-paketti.

Sovelluskehys PHP -ohjelmointikielille

Kuten otsikko antaa ymmärtää, puhun seuraavassa vain PHP-ohjelmointikielen ominaisuuksista ja sovelluskehyskiä. Monet peruseräkkeet pätevät myös muihin palvelinpuolen ohjelmointikieliin, mutta tässä yhteydessä en ota niihin kantaa.

* Palvelinpuolen ohjelmointikielillä tarkoitetaan ohjelmointikieliä, jotka suoritetaan palvelimella. Tällaisia ovat esimerkiksi PHP, Python, Ruby ja ASP. Asiakaspuolen ohjelmointikielillä kirjoitetut ohjelmat suoritetaan sen sijaan verkkosivuilla vierailevan käyttäjän selaimessa. Esimerkiksi JavaScript, AJAX, jQuery ja JSON ovat asiakaspuolen ohjelmointikieliä.

Palvelinpuolen ohjelmat siis suoritetaan ensin ja näytetään sitten suorituksen tulos, esimerkiksi internetsivu, sivustolla vierailevan kävijän selaimessa. Asiakaspuolen ohjelmia voidaan ajaa ns. lennosta, jolloin sivua ei tarvitse pyytää eli ladata uudestaan palvelimelta. Asiakaspuolen ohjelmilla voidaan päivittää myös osia sivusta, siten että koko sivua ei tarvitse ladata uudestaan. Asiakaspuolen sovellusten lähdekoodia voidaan tarkastella selaimessa, palvelinpuolen sovellusten lähdekoodia ei pääse näkemään selaimen välityksellä.

PHP-kehyskiä lisää arvoa tuovat käyttäjäystävälliset osoitteet eli URL:t (Dorf 2010). Vertaile vaikkapa kahta seuraavaa esimerkkiosoitetta:

1. http://www.osoite.fi/index.php?sivu=kayttaja&kayttaja_id=44
2. <http://www.osoite.fi/kayttaja/44>.

Helposti luettavista osoitteista on hyötyä etenkin julkisilla verkkosivuilla muun muassa hakukoneoptimoinnissa. Suljetussakin järjestelmässä helposti luettavista osoitteista on apua esimerkiksi vikatilanteissa; voidaan miettiä, kumpi osoite on helpompi kertoa puhelimen välityksellä tukihenkilölle.

PHP-sovelluskehysissä on helpompaa toteuttaa ja hallita tietoturvallisuusasiat, kuin jos kirjoittaisi kaiken koodin käsin.

Esimerkiksi CodeIgniter-sovelluskehysessä turvallisen lomakkeen tekeminen onnistuu lisäämällä lomake ja siihen kenttiä valmiiden moduulien avulla. Lomakkeen kenttiin lisätään ehtoja ja tarkenteita tarpeen mukaan ja sovelluskehys huolehtii automaattisesti, että käyttäjän syötteet vastaavat sallittuja arvoja. CodeIgniterissa on otettu huomioon SQL-injektointi, XSS- eli Cross-Site Scripting ja lomakkeiden syötteiden tarkistaminen. (EllisLab 2011.)

MVC-arkkitehtuuri

Model-View-Controller- eli malli-näkymä-käsittelijä-arkkitehtuuri helpottaa ohjelmointityötä ja ylläpitoa. Lyhyesti MVC-ohjelmistoarkkitehtuurin idea on erottaa käyttöliittymä sovelluksesta, jolloin näitä voidaan luoda, huoltaa ja käyttää toisistaan riippumatta. (Dorf 2010.)

PHP-sovelluskehystä käyttävän järjestelmän huolto on hallitumpaa, kun siinä on erotettu kuvauskieli ja PHP-koodi toisistaan. Käytännössä tämä tarkoittaa sitä, että sovelluksen varsinaista toimintaa pyörittävä PHP-ohjelma on erillään loppukäyttäjälle näytettävästä kauniista kuoresta, joka on toteutettu JavaScript, HTML- ja CSS-koodeilla. Erilliset osat ovat myös fyysisesti eri paikoissa, joka nopeuttaa tarvittavien osien löytämistä, lukemista ja huoltamista.

Dorf listaa PHP-sovelluskehykset LearnComputer-sivuston blogissa seuraavasti:

1. CodeIgniter <http://codeigniter.com>
2. Yii <http://www.yiiframework.com>
3. Cakephp <http://cakephp.org>
4. Zend <http://www.zend.com>
5. Symfony <http://www.symfony-project.org>

(Dorf 2010.)

Edellä mainituista sovelluskehysistä valitsisin CodeIgniter-ympäristön selainpohjaisen varausjärjestelmän alustaksi. Se on itselleni tuttu, helposti omaksuttava ja hyvin dokumentoitu. Lisäksi sen online-yhteisö on laaja ja aktiivinen. CodeIgniterissa ei ole sisäänrakennettua kirjastoa autentikointia varten, mutta siihen on saatavana sellainen ns. lisäosana tai sellaisen voi ohjelmoida itse.

* Autentikoinnilla tarkoitetaan käyttäjien varmennusta, joka toteutetaan esimerkiksi käyttäjätunnus-salasana pari -tyyppisellä tunnistuksella kirjautumislomakkeen avulla. Esimerkiksi Facebook-tilille kirjautuminen on osa tällaista autentikointia.

Sovelluskehys CSS-tyylimäärittelyille

CSS sovelluskehukset ovat tulleet suosituiksi viime aikoina. Ne antavat web-kehittäjälle mahdollisuuden tehdä nopeita prototyyppisiä suunnitelmistaan. CSS sovelluskehysten tarkoitus, kuten sovelluskehysten yleensäkin, on hoitaa usein toistuvat rakenteet web-kehittäjän puolesta, jotta kehittäjä pääsisi nopeasti työstämään projektin ainutlaatuisia ominaisuuksia. (Noupe 2009.)

Noupe listaa viisi suosituinta CSS sovelluskehystä seuraavasti:

1. 960 Grid System <http://960.gs>
2. Blueprint CSS Framework <http://blueprintcss.org>
3. jQuery UI CSS Framework http://docs.jquery.com/UI/Theming/API#The_jQuery_UI_CSS_Framework
4. YAML <http://www.yaml.de/en>
5. YUI 2: Grids CSS <http://developer.yahoo.com/yui/grids>

(Noupe 2009.)

Yllämainituista CSS-sovelluskehyksistä olen käyttänyt 960 Grid System- ja jQuery UI -kehysä. Kuten aiemminkin, tässäkin on vaikeus valita monesta hyvästä, mutta selainpohjaisen varausjärjestelmän tueksi valitsisin näistä molemmat. Lopullinen valinta kannattaa suhteuttaa myös valittuun JavaScript-kehukseen. Jos järjestelmässä käytetään Yahoo!n JavaScript-kehystä, on perusteltua valita CSS-kehukseksikin Yahoo!n tarjoama YUI 2: Grids CSS.

YHTEENVETO

Tässä luvussa tiivistän opinnäytetyön kokonaisuudeksi, sekä pohdin jatkotoimenpiteitä.

Työn tarkoitus

Opinnäytetyöni tarkoituksena oli luoda konseptisuunnitelma selainpohjaiselle varausjärjestelmälle Lahden Ammattikorkeakoulun Viestinnän koulutusohjelman pääaineille. Tavoitteenani oli tehdä suunnitelma, joka toimisi pohjana ja auttaisi varsinaisen järjestelmän toteuttamisessa, jos Lahden Ammattikorkeakoulu päättäisi hankkia järjestelmän Viestinnän koulutusohjelman käyttöön.

Keskeinen sisältö

Työssäni kävin läpi web-ohjelman luomisprosessin, tutkin nykyisiä lainauskäytänteitä Lahden Ammattikorkeakoulun Viestinnän koulutusohjelman pääaineissa, suunnittelin keskeisimmät toiminnot selainpohjaiseen varausjärjestelmään yhdessä henkilökunnan kanssa sekä pohdin järjestelmän toteuttamista käytännössä.

Tärkeimmät tulokset

Tekemäni selvitys osoittaa kiistattomasti, että Lahden Ammattikorkeakoulun Viestinnän koulutusohjelman pääaineiden lainauskäytännöt ovat vanhanaikaisia ja puutteellisia. Jokaisella pääaineella on omat lainauskäytännöt ja niiden laajentaminen muiden pääaineiden käyttöön on hankalaa tai mahdotonta. Viestinnän koulutusohjelman pääaineiden tarpeet ovat monelta osin niin ainutlaatuisia, että valmista sovellusta niiden tarpeisiin ei löydy. Olemassa olevien järjestelmien muokkaaminen on työlästä, hidasta ja kallista. Eikä olemassa olevaa järjestelmää muokkaamalla, välttämättä päästä toivottuun lopputulokseen.

Yleensä kaupallisten sovellusten muokkaaminen on lisäksi kielletty tai sitä on rajoitettu lisensseillä, eikä niiden muokkaaminen ole välttämättä edes teknisesti mahdollista. Ainoaksi mielekkääksi vaihtoehdoksi jää räätälöidyn selainsovelluksen tekeminen alusta asti, siten että se vastaa Viestinnän koulutusohjelman pääaineiden tarpeisiin.

Johtopäätelmiä tulosten pohjalta

Tekemäni selvityksen pohjalta voidaan päätellä, että nykyiset lainauskäytännöt vaativat uudistusta ja selainpohjainen varausjärjestelmä voitaisiin rakentaa vastaamaan tarpeisiin, yhtenäiseksi työkaluksi kaikille Viestinnän koulutusohjelman pääaineille.

Työn onnistumisen arviointi

Kokonaisuutena työni onnistui hyvin. Tärkeimpänä sisällöllisenä tavoitteenani oli selainpohjaisen varausjärjestelmän alustavan suunnitelman tekeminen. Tämä tavoite toteutui suunnitelmieni mukaan.

Selainpohjaisen varausjärjestelmän tuottaminen

Selainpohjaisen järjestelmän tuottaminen Lahden Ammattikorkeakoululle, tapahtuisi luultavasti parhaiten siten, että konseptisuunnitelma esiteltäisiin henkilökunnalle ja opiskelijoille. Tällöin heille tulisi tarve saada järjestelmä käyttöön. Kiinnostus järjestelmää kohtaan on herännyt jo tämän opinnäytetyön tekemisen aikana. Tämän tarpeen pohjalta voitaisiin perustella, että järjestelmän hankkiminen on kannattavaa taloudellisesti. Ehdotuksen tueksi tarvitaan tietenkin konkreettista näyttöä, laskelmia, taulukoita sekä hyvä esitys, joka uppoaa päättäjiin.

Miten varausjärjestelmä toteutetaan käytännössä?

Järjestelmän varsinainen toteuttaminen vaatisi työryhmän kokoamisen. Tiimi saataisiin koottua pitkälle Lahden Ammattikorkeakoulun opiskelijoista. Vahvistusta voitaisiin hakea yritysyhteistyökumppaneista, esimerkiksi pyytämällä konsultaatiota. Opiskelijat saisivat työstä erinomaista kokemusta ja opintopisteitä. Näin saataisiin koottua motivoitunut työryhmä ja kustannukset jäisivät verrattain alhaisiksi.

Kirjalliset lähteet

- Cooper, A., Reinmann, R., Cronin, D. 2007. About Face 3 – The Essentials of Interaction Design. Indianapolis, Indiana: Wiley Publishing, Inc.
- Hakala, M., Vainio, M., Vuorinen, O. 2006. Tietoturvallisuuden käsikirja. Porvoo: WS Bookwell.
- Kuutti, W. 2003. Käytettävyys, suunnittelu ja arviointi. Saarijärvi: Gummerus Kirjapaino Oy.
- Kärkinen, Jouni. 2011. Erikoislaboratoriomestari, Elokuva- ja TV-ilmaisu. Lahden Ammattikorkeakoulu. LIITE 2.
- Saariluoma, S., Kujala, T., Kuuva, S., Kymäläinen, T., Leikas, J., Liikkanen, L.A., Oulasvirta, A. 2010. Ihminen ja teknologia - Hyvän vuorovaikutuksen suunnittelu. Tampere: Tammerprint.
- Shore, J., Warden, S. 2008. The Art of Agile Development. Sebastopol, Kalifornia: O'Reilly Media, Inc.

Elektroniset lähteet

- Buckley, Heather. 2011. The Best JavaScript Frameworks and Libraries for Web and Mobile. Silicon Beach Training [Viitattu 14.12.2011]. Saatavissa <http://www.siliconbeachtraining.co.uk/blog/best-javascript-framework-libraries-web-mobile>
- Croft, Jeff. 2007. Frameworks for Designers. A List Apart Magazine [viitattu 14.12.2011]. Saatavissa <http://www.alistapart.com/articles/frameworksfordesigners>
- Dorf, Michael. 2010. Why Use a PHP Framework?. LearnComputer [viitattu 14.12.2011]. Saatavissa <http://www.learncomputer.com/why-use-php-framework>

- EllisLab, Inc. 2011. Form Validation : CodeIgniter User Guide. EllisLab Inc [viitattu 7.3.2012]. Saatavissa http://codeigniter.com/user_guide/libraries/form_validation.html
- Kohan, Bernard. 2012. Web Application Development Process. Comentum 360 [Viitattu 02.01.2012]. Saatavissa <http://www.comentum.com/web-application-development-process.html>
- Noupe. 2009. 5 Popular CSS Frameworks + Tutorials & Tools for Getting Started. Smashing Media GmbH [viitattu 16.12.2011]. Saatavissa <http://www.noupe.com/css/5-popular-css-frameworks-tutorials-tools-for-getting-started.html>

Suulliset lähteet

- Kutvonen, Arto. Tuntiohjaaja, multimediatuotanto. Lahden Ammattikorkeakoulu. Keskustelut 2011-2012.
- Kärkinen, Jouni. Erikoislaboratoriomestari, Elokuva- ja TV-ilmaisu. Lahden Ammattikorkeakoulu. Keskustelut 2011-2012. LIITE 2.
- Lius, Tommi. Tuntiohjaaja, multimediatuotanto. Lahden Ammattikorkeakoulu. Haastattelu 7.12.2012. LIITE 3.
- Pohjola, Virve. 2012. Opintosihteri. Lahden Ammattikorkeakoulu. Keskustelu 22.2.2012.
- Reiniharju, Markku. Erikoislaboratoriomestari, Valokuvaus. Lahden Ammattikorkeakoulu. Haastattelu 9.12.2011. LIITE 4.

Liitteissä olevat haastattelut ovat kirjoittajan omia referaatteja ja muistilistoja haastatteluista. Sähköpostit ovat alkuperäisessä muodossaan.

LIITE 1 Erikoislaboratoriomestari Jouni Kärkisen haastattelu

Kärkinen, Jouni. Erikoislaboratoriomestari, Elokuva- ja TV-ilmaisu. Lahden Ammattikorkeakoulu. Keskustelut 2011-2012 ja sähköpostihaastattelu 7.11.2011.

Sähköpostikysely 1

Moi,
Olen tekemässä opinnäytetyökseni suunnitelmaa laitteiston sähköisen varaus- ja lainausjärjestelmän toteuttamisesta. Sähköistä varausjärjestelmää hallinnoisivat labrimestarit ja sitä käyttäisivät opiskelijat laitteiden (esim. valot, linssit, kameran jalat, yms.) lainaamiseen sekä varaamiseen. Tässä kohtaa olisin kiitollinen, jos kertoisitte nykyään käytössä olevista järjestelmistänne. Eli ihan vapaa sana nykyisten järjestelmienne (vihko, excel, tm.) hyvistä ja huonoista puolista. Lisäksi olisi hyvä kuulla teidän näkemyksiä siitä, mitä toimintoja te haluaisitte sähköiseen varausjärjestelmään tai mitä siinä ehdottomasti pitäisi olla. Ajattelin lähestyä sähköpostilla, jotta asiasta saataisiin kerralla mustaa valkoiselle ja voisitte vastata teille parhaiten sopivaan aikaan. Jos kuitenkin haluatte mieluummin kertoa kasvotusten näkemyksistänne, ottakaa yhteyttä ja sovitaan tapaamisaika.

Ystävällisin terveisin,

--

Samu-Erkki Aaramaa

Sähköpostivastaus 1

Terve,
ETV-llä on excel-lista lainattavista kamoista ja editeistä. Eli sinne opiskelijat merkkää haluamansa kaluston ja varaavat edittivuorot. Sitten kun kamat otetaan varastosta ne kirjoitetaan erikseen lainauslomakkeelle.

t: Jouska

Sähköpostikysely 2

Moi,
Kiitos nopeasta vastauksesta. Mikä tässä järjestelyssä on hyvää ja mikä huonoa? Tai osaatko visioida mitä toimintoja sähköiseen varausjärjestelmään pitäisi ehdottomasti kuulua?

Ystävällisin terveisin,
Samu-Erkki Aaramaa

Sähköpostivastaus 2

No huonoa on ainakin se paperin käsin kirjoittaminen. Eli ainakin se varauslista pitäisi saada tulostettua esim. kahtena kappaleena toinen lainaajalle ja toinen minulle. Sitten kuitenkin kun mennään tonne varastolle tulee mieleen että tarviikin vielä tuon ja tuon...eli se lisääminen pitäisi hoitua helposti, niin ja tietty sitten kirjautua sinne varausjärjestelmään. Meillä siis toi varauslista on kuitenkin enempi ja vähempi suuntaa-antava varaston tilanteesta. Eli siis järjestelmän pitäisi olla millintarkka jokaista pikkunippiä myöten mutta toisaalta joustava....heh heh

jouska

Keskustelu 8.11.2011

Varausjärjestelmässä voisi olla sellainen ominaisuus, että kurssin vetäjä voisi luoda ns. tuotannon ja sille aikajanan. Opiskelija valitsisi esimerkiksi alasetoalikoista tuotannon, jota varten hän lainaa tavaroita. Pitää kuitenkin varoa, ettei järjestelmästä tule liian byrokraattista ja kankeata käyttöä. Meillä on haasteena myös kaksi fyysisesti eri tilassa olevaa varastoa. Olisi hyvä jos kirjautuminen toimisi AD-tunnuksilla. Tässä olisi etuna se, että järjestelmään ei pääsisi ulkopuoliset tahot. Käyttäjryhmät tai käyttöoikeustasot kuulostavat järkevältä. On tärkeää, ettei kuka tahansa pääse lainaamaan mitä tahansa.

Keskustelu 22.11.2011

Pitäisikö tilojen varaaminen mennä opettajien tai labramestareiden kautta? Pitäisikö järjestelmään pystyä kirjaamaan ensin tuotantosuunnitelma ja deadlinet, joille varausten aikarajat perustuvat?

Tässä on vaarana, että järjestelmästä tulee liian kankea käyttöä ja jää sen takia kokonaan käyttämättä eli että laitteita annetaan kirjanpidon ohi. Varauksissa pitäisi olla ns. arvojärjestys eli siten että labramestarit tai opettajat hyväksyvät varaukset käyttötarkoituksen perusteella päällekkäisissä varauksissa. Jokainen voisi siis tehdä varauksen koska tahansa, myös päällekkäin, mutta henkilökunta päättää kuka lopulta lainaa tavaran tai varaa tilan käyttöönsä. Laitteen varauksessa pitäisi näin ollen olla alasetoalikko, josta voi valita tai tekstikenttä johon voi kirjoittaa käyttötarkoituksen. Esimerkitapaus: Henkilö X varaa kameran kuvatakseen kissaansa huvin vuoksi, henkilö Y tekee varauksen samalle ajalle, koska on sovittu että hän menee kuvaamaan yritys yhteistyöprojektiin liittyviä kuvia. Labramestari/opettaja hyväksyy henkilön Y varauksen, koska tällä on tärkeämpi käyttötarkoitus. Samalla henkilölle Y menee tieto, että hänen varaus on hylätty ja mahdollisesti ehdotuksena seuraavat mahdolliset varausajat ko. laitteelle.

Varaukset pitää voida tehdä päivittäin tai tunneittain, koska joskus saattaa olla vain parin tunnin homma ja siinä tapauksessa laite ei saa olla koko päivää varattuna.

Viivakoodi on hyvä juttu joillekin tuotteille, mutta kaikille pikkunippeleille sitä ei voi laittaa. Eli tuotteen varauksessa pitää olla mahdollisuus ohittaa viivakoodi.

Keskustelu 10.02.2012

Tilojen vuokrauksessa, ainakin ETV:n editeissä, pitäisi olla vuorot. Eli aamuvuoro 8-16, iltavuoro 16-24 ja yövuoro 00-8. Tämä siksi, ettei opiskelija päästä, että herään tuossa yhden aikaan ja otan parhaan vuoron päältä kuljeksimasta. Järjestelmässä voisi siis olla vaihtoehtona varaus vuorojen mukaan tai tunneittain ja vuorot voisi määritellä järjestelmästä käsin ylläpitäjän toimesta.

LIITE 2 Tuntiopettaja Tommi Liuksen haastattelu

Lius, Tommi. Tuntiopettaja, multimediatuotanto. Lahden Ammattikorkeakoulu. Haastattelu 7.12.2012.

Laitteen tunnistaminen haasteena. Tunnistaminen pitää tehdä helpoksi. Laitteesta pitää olla kuvia, painot, värit, ID:t jne.

Suurin ongelma nykyisessä järjestelmässä on se, että liikaa jää opiskelijan vastuulle. Se ei olisi huono asia, jos homma toimisi, mutta kun homma ei toimi. Ei ole ehkä riittävää perehdyttämistä. Ongelmana myös, että laitteiden paikat vaihtuvat eli laukuissa ei ole välttämättä aina samoja kamoja sisällä. Vihkojärjestelmä ei pysy ajan tasalla, tavaraa on tullut lisää, mutta niitä ei ole merkitty kansioihin. Eli järjestelmään pitää olla helppo lisätä uusia tuotteita.

Nykyinen ongelma laitteen palauttaminen oikeaan paikkaan ja oikealla tavalla -> akut lataukseen jne. Vastuuntunto! Joku muu, ei ehdi jne. Palauttaminen labramestarin kautta - vain labbis voi hyväksyä palautuksen ja kuitata sen järjestelmään. Ei palauteta oikeaan aikaan vaan oletetaan, että varaajalle soitetaan, kun laitetta tarvitaan. Ei ilmoiteta kun tavaraa hajoaa. Eli olisi hyvä jos järjestelmä ilmoittaisi emaililla tai jopa viestillä, että laina-aika on menossa umpeen ja kertoisi mitä tapahtuu, jos palautus tapahtuu myöhässä. Samassa viestissä voi kertoa, että laina-aikaa voi pyytää lisää. Järjestelmässä pitää olla houkutteleva/rohkaiseva palautteenantosysteemi siltä varalta, että laite menee rikki. Siten että opiskelija tajuaa, että häntä ei aseteta edesvastuuseen hajonneesta laitteesta, mutta miten tärkeää on ilmoittaa asiasta labramestarille.

Jonkinnäköinen manuaali ainakin tärkeimmistä laitteista. Vinkkejä ja normaaleja käytäntöjä. Linkit laitteiden FAQ-sivuille.

Käyttäjäkuvat pakolliseksi. Käyttäjän kuva helpottaa labramestarin toimintaa. Joskus voi olla nimi hukassa, mutta muistaa kasvot jne. Tai jos pitää etsiä tietty henkilö käsiin, voi tarkastaa järjestelmästä minkä näköistä henkilöä etsitään.

Järjestelmän käyttöopastus sisällytettävä opetussuunnitelmaan

ensimmäiselle vuodelle. Puolen päivän koulutus.

AD-tunnuksien käyttö olisi hyvä juttu, jotta käyttäjien ei tarvitsisi muistaa monia tunnuksia. Käyttäjä voisi toisaalta määritellä käsin saman tunnuksen, mutta silloin salasanan päivittäminen ei tapahtuisi automaattisesti, vaan salana pitäisi käydä muuttamassa varausjärjestelmään erikseen.

Tärkeysjärjestysperustainen lainan hyväksyminen on hyvä juttu. Tästä maininta käyttäjäehtoihin.

Käyttäjiin voimassaoloaika olisi hyvä, ettei käyttäjiä tarvitse käsin poistaa. Tämä toimii hyvin myös väliaikaisiin käyttäjiin. Ensimmäisen vuoden opiskelijan voimassaoloaika on oletuksena neljä vuotta.

Käyttäjryhmät drag-n-drop -tyyppisesti esim. MUL08 kerrallaan raahataan johonkin ryhmään, mutta voi poistaa yksitellen epäpätevät opiskelijat.

Järjestelmässä pitäisi olla käyttöönottilanteessa joku pikaopas tms. joka kertoisi järjestelmän toiminnan pääpiirteet ja sama tulisi näkyviin, jos käyttäjä ei ole kirjautunut järjestelmään esim. viimeisen kolmen kuukauden aikana. Tämä on hyvä siksikin, että jos järjestelmän käyttö opetetaan heti koulun alussa, se unohtuu hyvin helposti kaiken muun informaatiotulvan alle.

Tietokoneiden ym. ns. kiinteiden laitteiden laitteiden, sekä tilojen varaus pitää toimia saman järjestelmän kautta ja yksinkertaisesti.

Tulevaisuutta: laukussa viivakoodi, joka sisältää tiedon siitä, mitä laukun sisällä pitää olla. Tulvaisuudessa käyttäjän pitäisi pystyä lukemaan laukun viivakoodi esim. kännykällä, jotta hän näkisi mitä laukkuun kuuluu.

LIITE 3 Erikoislaboratoriomestari Markku Reiniharjun haastattelu

Reiniharju, Markku. Erikoislaboratoriomestari, Valokuvaus. Lahden Ammattikorkeakoulu. Haastattelu 9.12.2011.

Valokuvausosaston nykyinen laitteiston varaus- ja lainausjärjestelmä on Windows-sovellus, joka on asennettuna ja toimii vain yhdellä tietokoneella. Järjestelmää käyttää vain erikoislaboratoriomestari Markku Reiniharju tai hänen alaisuudessaan kulloinkin oleva harjoittelija. Järjestelmässä on lainaus- ja palautus-, varasto-, varaus- sekä raportit-osio.

Lainaus tapahtuu lukemalla lainaajan henkilötiedot opiskelija- tai KELA-kortin viivakoodista. Jos lainaajan tietoja ei ole järjestelmässä, avautuu uusi ikkuna, johon uuden käyttäjän tiedot voidaan manuaalisesti lisätä. Henkilötiedoissa näkyy henkilötunnus, etunimi, sukunimi, postiosoite, puhelinnumero ja luokka. Kun henkilön tiedot löytyvät järjestelmästä ja henkilökortin viivakoodi on luettu, luetaan lainattavan laitteen viivakoodi. Laitteen viivakoodi luetaan kaksi kertaa, koska ensimmäisellä lukukerralla järjestelmä vasta tulkitsee, että jotain halutaan lainata ja avaa ikkunan, johon viivakoodi voidaan syöttää tai lukea lukijasta. Toisella kerralla laite lukee viivakoodin ja lisää laitteen tiedot lainaajan profiiliin. Jos kyseinen laite on jo lainattuna samalle henkilölle eli halutaan uusia laina, pitää laitteen viivakoodi lukea kolmeen kertaan, jotta laite tulee uudelleen varatuksi kyseiselle henkilölle. Laitteen lainaustiedoissa näkyy kolmen viimeisimmän lainaajan henkilötiedot, lainaajan henkilötiedoissa ei näy lainaushistoriaa lainkaan. Jos halutaan tarkastella tietyllä henkilöllä lainassa olevia laitteita, haetaan kyseisen henkilön nimellä tietoja lainausosiosta. Ohjelman tulostaman nimelistan haluttua nimeä vastaavaa riviä klikkaamalla, nähdään kyseisellä henkilöllä lainassa olevat laitteet. Käyttäjätiedot eivät poistu automaattisesti järjestelmästä, vaan ne pitää yksitellen poistaa. Reiniharju kertoo, ettei hän välttämättä pidä hyvänä ideana automaattista poistamista, sillä joistakin opiskelijoista saattaa tulla opettajia tai yhteistyökumppaneita, jotka pitäisi

sitten jälkikäteen lisätä uudestaan järjestelmään. Laitteita lainataan myös Valokuvauslinjan ulkopuolelle, esimerkiksi Elokuva- ja TV-ilmaisun opiskelijoille. Mielestäni tässäkin kohtaa Viestinnän koulutusohjelman yhteinen järjestelmä olisi hyvä juttu. Reiniharju sanoo, että olisi hyvä jos järjestelmän pystyisi linkittää opiskelijarekisteriin, josta järjestelmä hakisi automaattisesti opiskelijan tiedot, ettei niitä tarvitsisi käsin lisätä järjestelmään.

Palautus tapahtuu syöttämällä 000-koodi laitteen lainaustietoihin, jolloin laitteen tila muuttuu palautetuksi ja laitteen tiedot poistuvat kyseisen henkilön varaustiedoista. Järjestelmää on päivitetty siten, että palautuksen ja lainaamisen seuraavalle henkilölle voi suorittaa myös lennosta eli että laitetta ei tarvitse ensin palauttaa ja sitten vasta lainata seuraavalle henkilölle. Palautettaessa ohjelma ei kerro kuka palautti ja mitä, vaan pelkästään että palautus on suoritettu onnistuneesti. Tämä ei ole Reiniharjun mukaan suuri ongelma, mutta näkisin että se altistaa virheen tekemiselle. Esimerkiksi jos kahdella ihmisellä on samanlainen optiikka lainassa, on mahdollista palauttaa vahingossa "väärän" ihmisen optiikka, eikä järjestelmä anna virheilmoitusta tai varoitusta asiasta.

Mielestäni tässä voisi olla sellainen systeemi, että valitaan ensin henkilö, joka on palauttamassa laitetta tai laitteen osaa ja hänen varauslistastaan merkittäisiin laite palautetuksi, tällä tavoin ei voisi palauttaa toisella henkilöllä lainassa olevia laitteita vahingossa.

Varaston tilannetta voidaan seurata syöttämällä ohjelmaan hakutietoja, jolloin ohjelma tulostaa listan laitteista, jotka vastaavat syötettyjä hakutermejä. Laitteiden yksityiskohtaisia tietoja pääsee katsomaan klikkaamalla laitteen nimen riviä listassa. Kun riviä on klikattu, aukeaa uusi ikkuna, jossa näkyy tarkemmat laitetiedot. Laitteen tietoja ovat viivakoodi, nimike, tarkenne, merkki ja valintaruutu joka kertoo onko laite lainassa vai ei. Tarkenteessa on joissakin tapauksissa laitteen sarjanumero, mutta ei aina. Jos tarkenteen teksti on pitkä, sarjanumero ei näy tekstikentän rajallisen koon vuoksi. Varasto on myös yksi lainaaja, joka helpottaa Reiniharjun mukaan inventaarion tekemistä. Samalla tavoin lainaajia ovat huolto, korjaus ja poistot. Kun laite tulee takaisin esimerkiksi huollosta, se palautetaan järjestelmässä samalla tavalla, kuin se olisi ollut lainassa jollakin henkilöllä.

Varaus ei käytännössä toimi ollenkaan. Varaussysteemi kyllä tunnistaa

laitteet viivakoodien perusteella, mutta laitteen ollessa lainassa, se ei kerro että varastossa on muita samanlaisia laitteita vapaana. Tästä johtuen varaukset merkitään paperikalenteriin, joka toimii muistilappuna Reiniharjulle.

Raportit-osiosta voi valita joko lainat, varaukset, myöhästyneet lainat tai kaikki lainattavat välineet. Kun jokin edellä mainituista on valittu, painetaan nappia, joka avaa Excel-taulukkolaskentaohjelman ja vie tiedot Excel-taulukkoon. Excel-taulukko on hyvä, jos esimerkiksi halutaan lähettää vakuutusyhtiölle lista varastossa olevista laitteista. Excelin puolella on kätevämpi tehdä listauksia, kuin nykyisessä lainausjärjestelmässä, mutta siellä ei voi muuttaa mitään tietoja. Eli jos tiedoista löytyy virhe, se pitää mennä muuttamaan järjestelmän puolelta. Reiniharjun mukaan Excel-pohjainen järjestelmä ei taas sovi varaus- ja lainausjärjestelmäksi, koska sen kautta on helppo tuhota tietoja tai koko järjestelmä.

Laitevarausjärjestelmän hyviä puolia on viivakoodinlukumahdollisuus ja se että samassa järjestelmässä on varaston seuranta ja huolto- sekä korjausseuranta.

Reiniharjun mielestä tunnistuspohjainen varausjärjestelmä on hyvä. Viivakoodi on hyvä juttu, mutta se myös kuluu helposti pois mekaanisen kulutuksen johdosta. Kuluneen viivakoodin vaihtaminen on hankalaa ja käytännössä se menee niin, että vanha koodi poistetaan ja tehdään uusi koodi. Koodin vaihtamisen myötä, myös varaushistoria nollautuu. Uudelleenkodeittamisen pitäisi olla helpompaa, koska koodeja joutuu vaihtamaan usein juuri mekaanisen kulumisen vuoksi. Valokuvauslinjalla on olemassa myös viivakoodiprintteri, mutta siitä ei Reiniharjun mukaan ole käytännössä mitään hyötyä, koska sillä printattu viivakoodi lähtee heti pois. Muovitettu viivakoodi pysyy vähän paremmin. Viivakoodin lisäksi laitteessa tai laitteen osassa pitää olla tunnistenumero eli sarjanumero, joka löytyy myös järjestelmästä. Sarjanumeroa tarvitaan etenkin silloin, kun on kyseessä niin sanottu pitkä laina. Esimerkiksi neljä vuotta käytössä olleessa kamerassa ei usein ole enää tietoaakaan viivakoodista. Viivakoodin käyttö pienissä nippeleissä, kuten muistikorteissa ja pehmeissä tavaroissa, kuten laukuissa on ongelmallista. Joitakin laitteita lainataan useimmiten pakettina, jolloin esimerkiksi laukussa voi olla viivakoodi ja sen takaa löytyy tiedot siitä, mitä laukun sisällä on tai kuuluu olla. Laukun sisällä olevissa laitteissa

voi myös olla omat viivakoodit. Ideoimme Reiniharjun kanssa mikrosirun upottamista laitteisiin ja hänen mielestä se olisi erittäin hyvä juttu. Mikrosirujen asentamisen kustannus on toistaiseksi iso kysymysmerkki. Elektroniikan hinta on toki tullut alaspäin; Reiniharju muistelee että esimerkiksi Valokuvausosaston vanha viivakoodinlukija maksoi toista tonnia parikymmentä vuotta sitten, kun nykyinen lukija maksoi Gigantissa 35 euroa.

Markku Reiniharju sanoo, että TaiK:ssa on käytössä järjestelmä, joka muistuttaa lainaajaa sähköpostilla lainan eräpäivän lähestymisestä. Reiniharju pitää sähköposti-ilmoitusta hyvänä ideana. Samalla Reiniharju kuitenkin huomauttaa, että moni ei lue sähköpostia. Lisäksi hän kertoo, että koulumme järjestelmä poistaa automaattisesti neljän vuoden kuluttua sähköpostitilit, ellei opiskelija erikseen ilmoita, että opiskelu jatkuu. Tämä on hänen mielestään hyvin hankala systeemi, koska juuri silloin usein tarvitsisi kuumeisimmin saada yhteys opiskelijoihin. Ehdotin tekstiviesti-ilmoitusta, mutta Reiniharju sanoo, että sekin on jo vanha juttu. Hän sanoo että nykyään Facebookin kautta saa ihmiset parhaiten kiinni. Mielestäni Facebook-ilmoitus on mielenkiintoinen idea, mutta sen toteuttaminen käytännössä voi olla hankalaa. Se tarkoittaisi Facebook-applikaation tekemistä ja sitä että käyttäjien pitäisi asentaa applikaatio omaan profiiliinsa saadakseen ilmoitukset vanhenevista lainoista. Idea on mielenkiintoinen ja sitä voi mielestäni pitää avoinna ainakin lisäoptiona. Ongelmana tässä on se, että kaikilla ei ole Facebook-tiliä ja moni ei välttämättä halua asentaa ko. lisäapplikaatiota profiiliinsa. Todettakoon että sellaista järjestelmää tai väylää on tuskin olemassa, joka saavuttaa 100% kaikki käyttäjät ja vaikka viesti myöhästymisestä saataisiin käyttäjälle perille, se ei silti ole tae siitä, että itse laite palautetaan määräaikaan mennessä, sovittuun paikkaan ja asiallisessa kunnossa. Sellaista järjestelmää, joka tähän pystyy, ei yksinkertaisesti ole eikä voi pelkillä ohjelmavälineillä tehdä.

Reiniharju kertoo, että sähköposti-ilmoituksen piiristä pitäisi myös saada pois niin sanottu peruskalusto. Tämän voisi mielestäni käytännössä toteuttaa siten, että laitteen ominaisuuksissa olisi valintaruutuperiaatteella kohta, josta valitaan lähetetäänkö myöhästymisestä ilmoitus käyttäjälle vai ei.

Valokuvauslinjan nykyisen varaus- ja lainausjärjestelmän huonoja puolia ovat järjestelmän sisäiset laitteiden, käyttäjien ja varaston selausominaisuudet. Selkeitä puutteita ovat laitteiden varausmahdollisuus, automaattiset ilmoitukset laina-ajan umpeutumisesta sekä laina-ajan muuttaminen, laina-aika on aina viikon.

Itselleni ohjelmasta tuleemieleen Windows-ohjelma 90-luvun alkupuoliskolta. Ohjelma vaikuttaa kankealta käyttäjä. Ohjelman osiot on jaoteltu välilehtiin, mutta jotkut osiot avautuvat erilliseen ikkunaan. Raportit voi viedä ohjelmasta Excel-taulukon, joka on sinänsä hyvä juttu, mutta mielestäni niitä pitäisi pystyä selata järkevästi myös järjestelmän sisällä, siten että Exceliä ei tarvitsisi välttämättä käyttää. Nyt listaukset menevät Exceliin, muun muassa siksi sieltä on helpompi tehdä suodatuksia listoihin. Minusta on epäkätevää, että pitää avata Excel, sitä varten että voi näennäisen kätevästi selaila järjestelmän tietoja. Excelin kautta ei voi kuitenkaan muuttaa järjestelmän tietoja, vaan jos virhe löytyy, se pitää käydä järjestelmän puolella muuttamassa. Ohjelmien ja ikkunoiden välillä hyppiminen on mielestäni epäkäytännöllistä, aikaa vievää ja sekavaa. Erikoislaboratoriomestari Markku Reiniharjun mukaan järjestelmän ylläpito ja lainausten hoito on työlästä ja vaatisi yhden henkilön kokopäiväisen työpanoksen, jotta se toimisi jouhevasti. Oman näkemäni ja kuulemani perusteella ohjelma ja koko varausjärjestelmä kaipaisi kipeästi päivitystä. Reiniharju kertoo että nykyisen järjestelmän tekijät ovat jo aikoja sitten valmistuneet koulustamme, eivätkä he ilmeisesti enää hoida ohjelman päivitystä. Multimediatauotannon vastuuopettaja, Arto Kutvosen kanssa olemme jo aiemmin visioineet, että selainpohjainen järjestelmä pitäisi toteuttaa siten, että sitä pystyy ja osaa muutkin kuin sen tekijä päivittää eli että järjestelmän tulevaisuus ei jää nojaamaan yhden ihmisen tai ryhmän varaan.