



TILAVARAUSJÄRJESTELMÄN SUUNNITTELU

Raisa Koikkalainen

**Opinnäytetyö
Toukokuu 2008**

Liiketalous



**JYVÄSKYLÄN
AMMATTIKORKEAKOULU**

Tekijä(t) KOIKKALAINEN, Raisa	Julkaisun laji Opinnäytetyö	
	Sivumäärä 78	Julkaisun kieli Suomi
	Luottamuksellisuus <input type="checkbox"/> Salainen _____ saakka	
Työn nimi TILAVARAUSJÄRJESTELMÄN SUUNNITTELU		
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma		
Työn ohjaaja(t) KARHULAHTI, Mika		
Toimeksiantaja(t) Jyväskylän seutukunta ry		
<p>Tiivistelmä</p> <p>Tutkimuksen tavoitteena oli suunnitella tilavarausjärjestelmä Jyväskylän käyttöön. Keskeisenä lähtökohtana oli käytettävyyden huomioon ottaminen suunnitteluprosessissa. Tutkimuksessa noudatettiin käyttäjäkeskeisen suunnittelun periaatteita. Tutkimus rajattiin koskemaan järjestelmän vaatimusmäärittelyä ja analyysivaihetta.</p> <p>Teoriaosuudessa kuvattiin käytettävyydestutkimuksen käsitteitä ja periaatteita. Keskeisiä käsitteitä olivat käytettävyys, käyttäjäkeskeinen suunnittelu ja käytettävyyssuunnittelu. Teoriaosuudessa käsiteltiin lyhyesti myös kognitiivisen psykologian teorioita. Tietojärjestelmien kehittämistä tarkasteltiin oliokeskeisen ohjelmistotuotannon näkökulmasta.</p> <p>Tutkimusmenetelmänä oli laadullinen tutkimus. Tutkimusaineistoa kerättiin osallistuvan havainnoinnin ja haastattelujen avulla. Käyttäjät olivat tutkimuksessa keskeisessä asemassa. Tutkimuksen tuloksina saatiin kuvaukset käyttäjistä, tehtävistä ja käyttöympäristöstä sekä kartoitukset käyttäjien tarpeista ja vaatimuksista. Käyttäjävaatimusten perusteella määritettiin käytettävyystavoitteet. Tietojärjestelmän kehittäminen aloitettiin järjestelmän vaatimusmäärittelyllä, joka tehtiin käyttäjävaatimusten perusteella. Järjestelmän analyysivaihe toteutettiin oliokeskeisen ohjelmistotuotannon periaatteilla. Analyysissa määriteltiin käsitelmä, oliomalli ja tietokannan rakenne. Käyttöliittymän määrittely tehtiin operaatioanalyysillä. Järjestelmäkuvaukset tehtiin UML-mallinnuksen avulla.</p> <p>Käyttäjäkeskeinen suunnittelu oli tutkimuksessa keskeisessä asemassa. Käytettävyystavoitteet otettiin huomioon järjestelmän kehitysprosessissa. Tietorakennetta muokattiin käyttäjien toiveiden perusteella. Jatkotutkimuksessa käytettävyys voidaan ottaa huomioon myös käyttöliittymän suunnittelussa ja toteutuksessa.</p>		
Avainsanat (asiasanat) käytettävyys, käyttäjäkeskeinen suunnittelu, käytettävyyssuunnittelu, tietojärjestelmän kehittäminen, ohjelmistotuotanto, tilavarausjärjestelmä		
Muut tiedot		

Author(s) KOIKKALAINEN, Raisa	Type of Publication Bachelor's Thesis	
	Pages 78	Language Finnish
	Confidential <input type="checkbox"/> Until	
Title ROOM BOOKING SYSTEM DEVELOPMENT		
Degree Programme Business Information Systems		
Tutor(s) KARHULAHTI, Mika		
Assigned by Jyväskylä Settlement		
Abstract <p>The aim of this thesis was to develop a room booking system for Jyväskylä settlement. Usability was the main focus of the development process with user-centered design as a guideline. The object of the study was defined to follow requirements specification and systems analysis.</p> <p>The theoretical part illuminates the concepts and principles of usability research. Fundamental concepts are usability, user-centered design, and usability engineering. The theories of cognitive psychology are discussed in brief. The information systems development is dealt with emphasis on object-oriented programming.</p> <p>The research method was qualitative research. The research material was collected through participant observation and interviews. The research was user-centered. The study yielded descriptions of the users, their tasks and the working environment. Further a survey of user needs and requirements was accomplished. Usability goals were set on the basis of user requirements. The software process started with systems requirements specification based on user requirements. The system analysis was based on rules of object-oriented programming. The data model, the object model and database structure model of the system were defined. The user interface specification was performed with operation analysis. The system diagrams were modelled with UML.</p> <p>User-centered design was the main aspect of the study. Usability goals were taken into account during the system's development process. The data model was adapted to user wishes. In future development, usability can be taken into account in the process of user interface design and implementation.</p>		
Keywords usability, user-centered design, usability engineering, information systems development, software engineering, room booking system		
Miscellaneous		

SISÄLTÖ

1	JOHDANTO.....	4
2	TAUSTATEORIA	6
2.1	Käytettävyystudkimus	6
2.1.1	Mihin käytettävyyttä tarvitaan?	6
2.1.2	Mitä on käytettävyys?	6
2.1.3	Käytettävyystekijät ja niiden mittaaminen	7
2.1.4	Ihminen tiedon käsittelijänä.....	10
2.1.5	Käyttäjäkeskeinen suunnittelu	14
2.1.6	Käytettävyyssuunnittelu	14
2.1.7	Käyttöliittymän suunnittelu	18
2.1.8	Heuristiset säännöt.....	18
2.1.9	Käytettävyyden arviointi	24
2.2	Tietojärjestelmän kehittäminen	27
2.2.1	Käsitteiden määrittelyä	27
2.2.2	Tietojärjestelmän kehittämisvaiheet	28
2.2.3	Elinkaarimallit	29
2.2.4	Oliokeskeinen lähestymistapa ja mallintaminen	29
2.2.5	OMT++	31
3	TUTKIMUSASETELMA.....	34
3.1	Tutkimuksen tavoitteet ja rajaukset.....	34
3.2	Tutkimusmenetelmät.....	34
3.3	Tutkimuskysymykset.....	35
4	TUTKIMUKSEN TOTEUTUS	37
4.1	Käyttäjäkeskeinen suunnittelu	37
4.2	Tietojärjestelmän kehittäminen	38

5	TUTKIMUKSEN TULOKSET JA ANALYYSI.....	39
5.1	Käyttäjakeskeinen suunnittelu	39
5.1.1	Käyttäjäkuvaukset	39
5.1.2	Nykyisen tehtävän kuvaus	40
5.1.3	Käyttöympäristön kuvaus	41
5.1.4	Käyttäjien tarpeet ja vaatimukset	42
5.1.5	Käytettävyystavoitteet	43
5.2	Tietojärjestelmän kehittäminen	45
5.3	Tutkimuskysymyksiin vastaaminen.....	46
6	JOHTOPÄÄTÖKSET JA POHDINTA	47
	LÄHTEET	49
	LIITTEET	51
	Liite 1. Haastattelukysymykset	51
	Liite 2. Vaatimusmäärittely	52
	Liite 3. ER-kaavio	60
	Liite 4. Luokkakaavio.....	61
	Liite 5. Taulukuvaus.....	62
	Liite 6. Tietokantakaavio	74
	Liite 7. Tehtävälista	75
	Liite 8. Dialogikaavio.....	78

TAULUKOT

TAULUKKO 1. Käytettävyyssuunnittelun elinkaarimalli.....	16
--	----

KUVIOT

KUVIO 1. Hahmolait	12
KUVIO 2. OMT++ -menetelmän vaiheet Jaaksin mukaan	31
KUVIO 3. Käyttöliittymän määrittelyn vaiheet	32

1 JOHDANTO

Tietotekniikan käyttö tehostaa työntekoa ja parantaa sen tuottavuutta. Tietotekniikka myös helpottaa työntekoa vähentämällä toistuvia rutiinisuurituksia ja tekemällä siitä mielekkäämpää ja haastavampaa. Näiden tavoitteiden toteutuminen ei kuitenkaan ole aina itsestään selvää, sillä sovellukselta saattaa puuttua yksi tärkeä ominaisuus, käytettävyys. Pelkkä täydellisesti toimiva tekninen järjestelmä ei riitä, vaan sille täytyy rakentaa hyvä rajapinta käyttäjän ja sovelluksen välille, hyvä käyttöliittymä. Tässä tehtävässä auttaa käytettävyystutkimus, jonka teoriat ja menetelmät ovat keskeisessä asemassa myös tässä opinnäytetyössä.

Opinnäytetyön tarkoitus on suunnitella tilavarausjärjestelmä Jyvälän käyttöön. Olen toiminut Jyväskylässä soitonopettajana useita vuosia ja tein siellä myös tietojenkäsittelyn työharjoitteluni. Tehtäviini kuului erilaisten lomakepohjien suunnittelua, opetustyössä avustamista ja atk-tukihenkilön tehtäviä. Aloitin tilavarausjärjestelmän suunnittelun harjoitteluaikanani ja jatkan projektia nyt opinnäytetyönä.

Jyvälä on Jyväskylässä toimiva yleishyödyllinen yhdistys, viralliselta nimeltään Jyvälän setlementti ry. Jyvälä on toiminut vuodesta 1940 ja se on monipuolinen palveluntuottaja ja kansalaisjärjestö. Jyvälä järjestää toimintaa, joka tukee kansalaisaktiivisuutta ja erityisesti tietoyhteiskuntavalmiuksien kehittymistä. Toimialoja ovat yleissivistävä koulutus, lapsi- ja nuorisotyö sekä sosiaalinen työ. Keskeisiä toimintoja ovat kansalaisopisto, päiväkotitoimi ja nuorisotyö. Muita toimintoja ovat koululaisten iltapäivätoiminta sekä erilaiset ihmisten sosiaalista hyvinvointia tukevat projektit ja hankkeet. Myös maahanmuuttajat ja seniorit ovat keskeisellä sijalla Jyvälän toiminnassa. Päätoimipaikkana on vanha Nisulan kartanorakennus, jonka lisäksi kurssi- ja muuta toimintaa on pienemmissä toimipisteissä ja eri kouluilla. (Huovila 2008.)

Jyväskylässä on työsuhteisia työntekijöitä runsas sata henkeä ja henkilötötyövuosia on noin neljäkymmentä. Kansalaisopiston kursseilla on vuosittain yli 2000 opiskelijaa ja opetustunteja on noin 3500. Jyvälän Nuorten kerhotoimintaan osallistuu vuosittain noin 300 lasta ja nuorta. Toimintaa kuvaavien perinteisten taloudellisten ja määrällisten mittareiden rinnalle on nykyään noussut tarve mitata toimintaa myös sosiaalisesta ja

yhteiskunnallisesta sekä laadullisesta näkökulmasta. Setlementeissä onkin käynnistymässä sosiaalisen tilinpidon hanke, jossa etsitään indikaattoreita sosiaalisen ja yhteiskunnallisen sekä laadullisen tuloksen mittaamisen. (Huovila 2008.)

Jyvälän toimistotyöntekijöitä ovat vakituksessa työsuhteessa oleva kokoaikainen toimistonhoitaja ja koulutussuunnittelija sekä muutama määrä- ja osa-aikaisessa työsuhteessa oleva toimisto- ja projektityöntekijä. Kurssien ja muiden tapahtumien tilavaraukset tehdään taulukkolaskentaohjelman lomakkeille, jotka tulostetaan ja kiinnitetään ilmoitustaululle. Käytännössä tämä vaatii paljon käsityötä ja kopiointia ja on hidasta ja virhealtista. Joka viikko joudutaan kirjoittamaan uudet lomakkeet eri tiloja varten, koska käytössä ei ole ohjelmaa, jolla pystyisi tekemään tilavaraukset viikoittain toistuville tapahtumille koko lukukaudeksi. Opinnäytetyön tarkoitus on suunnitella tilavarausjärjestelmä, joka omalta osaltaan helpottaisi, nopeuttaisi ja tehostaisi toimistotyöntekijöiden työtä sekä tekisi siitä entistä mielekkäämpää poistamalla rutiiniluonteista työtä. Keskeisenä tavoitteena on suunnitella järjestelmä, jolla on hyvä käytettävyys.

2 TAUSTATEORIA

Tässä luvussa käsitellään ensin käytettävyytustutkimusta ja määritellään siihen kuuluvia keskeisiä käsitteitä. Luvussa luodaan myös lyhyt katsaus kognitiivisen psykologian teorioihin. Seuraavaksi kuvataan käytettävyyssuunnittelua ja sen periaatteita sekä käytettävyyden arviointimenetelmiä. Luvun lopussa käsitellään ohjelmistotuotantoa painottuen erityisesti oliokeskeisiin menetelmiin.

2.1 Käytettävyytustutkimus

2.1.1 Mihin käytettävyyttä tarvitaan?

Käytettävyytustutkimuksella on pitkä historia. Käytettävyyttä on arvioitu jo vuosikymmenien ajan sekä erilaisten teollisuustuotteiden että tietokoneiden käytön yhteydessä. Jokainen meistä on törmännyt joskus tuotteisiin, joita on kömpelö tai hankala käyttää. Samoin tietokoneen käytössä voi syntyä vaikeuksia ja ongelmia, jotka hidastavat työntekoa sekä herättävät ärtymystä ja turhautumista. Tästä syystä heräsi pyrkimys helpottaa ja tehostaa laitteiden ja ohjelmien käyttöä. Käytettävyydellä todettiin olevan myös kaupallista merkitystä, kun huomattiin, että parempi käytettävyys säästää sekä aikaa että rahaa. Helppokäyttöisyydestä tuli myyntivaltti ja tietokoneita mainostettiin korostamalla, miten yksinkertaista ja helppoa jopa lasten on niitä käyttää. Puhuttiin tietokoneiden käyttäjystävällisyydestä. (Faulkner 2000, 2–8.) Nykyään käytetään laajempaa käsitettä käytettävyys, jonka eri määritelmiä esitellään seuraavassa luvussa.

2.1.2 Mitä on käytettävyys?

Sovellusohjelman käytettävyys (*usability*) kuvaa käyttöliittymän laatua eli sitä, kuinka hyvin sovellus täyttää käyttäjän tarpeet. Käyttäjän tarpeet ovat sekä työtehtävään liittyviä että puhtaasti inhimillisiä eli havaitsemisprosesseihin, muistiin, ajatteluun ja toimintaan liittyviä. Käyttöliittymä (*user interface*) on se tietokonesovelluksen osa, jonka avulla käyttäjä ja sovellus ovat vuorovaikutuksessa keskenään. Käyttöliittymä on nykyään useimmiten graafinen ja koostuu ikkunoista, valikoista, kuvakkeista ja ohjauspainikkeista. (Kalimo 1996, 7–8.)

ISO 9241-11 -käyttöliittymästandardissa (Näyttöpäätteillä tehtävän toimistotyön ergonomiset vaatimukset. Osa 11: Käytettävyyden määrittely ja arviointi) käytettävyys määritellään tuottavuudeksi, tehokkuudeksi ja tyytyväisyydeksi, joilla määrätty käyttäjät voivat saavuttaa määrättyjä tavoitteita tietyissä ympäristöissä (Faulkner 2000, 7; SFS-standardien luettelo 2008). Tuottavuudella (*effectiveness*) tarkoitetaan sitä, että käyttäjä pystyy suorittamaan aikomansa tehtävän; suoritukseen kulunutta aikaa ei oteta huomioon. Tehokkuutta (*efficiency*) mitataan sillä, miten nopeasti käyttäjä suoriutuu tehtävästä. Tyytyväisyys (*satisfaction*) tarkoittaa sitä, miten hyvin käyttäjä hyväksyy järjestelmän, miten miellyttäväksi hän kokee sen käytön ja pitääkö hän enemmän kyseisestä vai jostakin toisesta järjestelmästä. (Faulkner 2000, 7–8.)

Käytettävyydellä tarkoitetaan myös sitä, miten hyvin tietty sovellus sopii tietyn tehtäväkokonaisuuden suorittamiseen tietyssä ympäristössä ja millaisia henkisiä ja fyysisiä ponnisteluja sen käyttö vaatii (Koivunen & Nieminen 1996, 12). Käytettävyys voidaan määritellä menetelmä- ja teoriakentäksi, jonka kautta käyttäjän ja laitteen yhteistointaa pyritään saamaan tehokkaammaksi ja käyttäjän kannalta miellyttävämmäksi (Sinkkonen, Kuoppala, Parkkinen & Vastamäki 2006, 17).

Hyvä käyttöliittymä ja käytettävyys vähentää tietotekniikan kustannuksia lisäämällä työn tuottavuutta ja pienentämällä koulutus- ja tukikustannuksia (Kalimo 1996, 8). Käytettävyys edistää myös sovelluksen laatuvaatimusten täyttymistä vastaamalla oikealla tavalla käyttäjien todellisiin vaatimuksiin ja tarpeisiin. Laadukas tietojärjestelmä toimii työntekijän odotusten ja työskentelytapojen mukaisesti eikä kuormita häntä tarpeettomasti. (Koivunen & Nieminen 1996, 15–16.)

Käytettävyystutkimus hyödyntää kognitiivisen psykologian sekä ihmisen ja tietokoneen välisen vuorovaikutuksen tutkimusta. Viimeksi mainitusta tutkimusalasta käytetään englanninkielistä nimitystä Human–Computer Interaction eli HCI. HCI:n tavoitteena on kehittää tai parantaa järjestelmien turvallisuutta, hyödyllisyyttä, tehokkuutta ja käytettävyyttä. (Preece 1994, 7, 14.)

2.1.3 Käytettävyystekijät ja niiden mittaaminen

Käytettävyuden määritelmän (ISO 9241-11) mukaan käytettävyuden tekijät ovat tuottavuus, tehokkuus ja tyytyväisyys. Kuinka sitten pystytään arvioimaan, täyttääkö jokin

tuote käytettävyysskriteerit vai ei? Mistä tiedetään, pystyykö käyttäjä suorittamaan aikomansa tehtävän (tuottavuus), suorittaako käyttäjä tehtävän nopeasti vähillä ponnistuksilla (tehokkuus) ja onko käyttäjä tuotteeseen tyytyväinen (tyytyväisyys)? (Faulkner 2000, 114.)

Nielsenin (1993, 26) mukaan käytettävyydellä on viisi ominaisuutta tai tekijää, joiden avulla käytettävyyttä voidaan arvioida, mitata ja parantaa. Käytettävyystekijät ja niiden merkitykset ovat seuraavat:

1. Opittavuus (Learnability)

Järjestelmän tulisi olla niin helppokäyttöinen, että käyttäjä voi nopeasti aloittaa työkentelyn sen parissa.

2. Tehokkuus (Efficiency)

Järjestelmän käytön tulisi olla tehokasta, jolloin on mahdollista saavuttaa korkea tuottavuuden taso.

3. Muistettavuus (Memorability)

Järjestelmän tulisi olla helposti muistettava, jolloin käyttäjä pystyy palauttamaan mieleensä sen käytön pitkänkin ajan kuluttua eikä hänen tarvitse opetella käyttöä aina uudelleen.

4. Virheiden määrä (Errors)

Järjestelmän käytössä tulisi syntyä mahdollisimman vähän virheitä, ja käyttäjän tulisi selviytyä virhetilanteista helposti. Katastrofaalisia virheitä ei saa tapahtua.

5. Tyytyväisyys (Satisfaction)

Järjestelmän käytön tulisi olla miellyttävää, niin että käyttäjät kokevat olonsa tyytyväiseksi ja pitävät sen käytöstä.

Shneiderman (1998, 15) on esittänyt vastaavat tekijät ja kuvannut, miten niillä voidaan mitata käytettävyyttä. Tekijät ovat:

1. Oppimiseen kuluva aika

Kuinka kauan tavanomaisilta käyttäjiltä kuluu aikaa oppia tehtävien suorittamiseen vaadittavat toimenpiteet?

2. Suorituksen nopeus

Kuinka kauan tehtävien suorittamiseen kuluu aikaa?

3. Käyttäjien tekemien virheiden määrä

Miten paljon ja millaisia virheitä käyttäjät tekevät suorittaessaan tehtäviä?

4. Muistissa säilyttäminen

Kuinka hyvin käyttäjät muistavat oppimansa tunnin, päivän tai viikon kuluttua?

5. Subjektiivinen tyytyväisyys

Kuinka paljon käyttäjät pitävät järjestelmän eri ominaisuuksien käytöstä?

Faulknerin mukaan ISO 9241-11 -määritelmä ei kerro, kuinka tuottava, kuinka tehokas ja kuinka tyydyttävä järjestelmän tulee olla ollakseen käytettävä. Siksi käytettävyystekijöihin pitää liittää mitattavia ominaisuuksia. Tällaisia ovat (Faulkner 2000, 117–119):

Tuottavuus

- onnistuneiden ja epäonnistuneiden yritysten suhde tehtävän suorituksessa
- eri komentojen ja toimintojen käytön esiintymistiheys; onko järjestelmässä turhia toimintoja, joita käyttäjät eivät käytä, ja onko käyttö tehokasta
- käytön ongelmatilanteiden mittaaminen
- työn tuotoksen laatu.

Tehokkuus

- tehtävän suoritukseen kulunut aika
- suorituksessa tarvittujen toimenpiteiden lukumäärä
- käyttöohjeiden lukemiseen kulunut aika
- online-ohjeen käyttämiseen kulunut aika
- virheiden käsittelyyn kulunut aika.

Käyttäjän tyytyväisyys

- käyttäjän suhtautuminen järjestelmään mitattavissa olevalla asteikolla.

Näiden ISO:n määritelmässä olevien tekijöiden lisäksi Faulkner (2000, 120–124) liittää käytettävyyteen opittavuuden ja joustavuuden:

Opittavuus

- järjestelmän käytön oppimiseen kulunut aika
- määrätyn suoritustason saavuttamiseen kulunut aika
- vaadittujen taitojen hankkimisessa havaitut vaikeudet
- järjestelmän käytön muistamiseen kulunut aika
- uudelleenoppimiseen kulunut aika
- virheilmoitusten esiintymistiheys.

Joustavuus

- järjestelmän on tarjottava vaihtoehtoisia käyttötapoja; käyttäjän on voitava käyttää järjestelmää itselleen sopivimmalla tavalla kulloisessakin yhteydessä.

2.1.4 Ihminen tiedon käsittelijänä

Ihmisen mielessä tapahtuvaa tietojenkäsittelyä eli kognitiivisia toimintoja tutkiva tieteenala on nimeltään kognitiivinen psykologia. Keskeisiä kognitiivisia toimintoja ovat havaitseminen, tarkkaavaisuus, muisti, oppiminen, ajattelu ja ongelmanratkaisu.

(Preece 1994, 39.) Seuraavaksi tarkastellaan lähemmin havaitsemista ja muistin toimintaa.

Havaitseminen

Jostain kohteesta tai tapahtumasta ihmiselle syntyvä havainto ei ole todellisuuden kopia vaan ihmisen oma tulkinta siitä. Tulkintaan vaikuttavat mielentilat, tunteet, muistot ja odotukset. Havaituille asioille luodaan merkityksiä yhdistelemällä havaintoja mielessä oleviin asioihin ja menneisiin kokemuksiin. Havainnot muokkaavat ihmisen mielikuvia ja maailmankuvaa ja päinvastoin. Jotta havainto onnistuisi, sen kohteen on erotuttava taustasta ja oltava ihmiselle merkityksellinen. Konstruktivistisen havaintoteorian mukaan maailmankuva muodostuu ympäristöstä saatavasta tiedosta, joka yh-

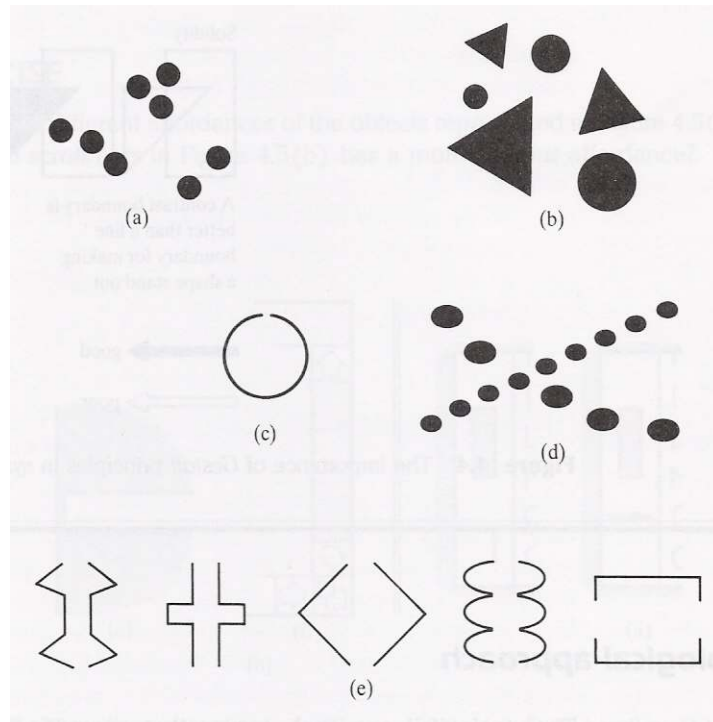
distyy ihmisen aiemmin tallentamaan tietoon ja asiayhteyteen. Ekologisen havaintoteorian mukaan taas tieto on ympäristössä valmiina poimittavaksi eikä havaintoa tarvitse konstruoida mielessä. (Sinkkonen ym. 2006, 80–83.)

Hahmolait

Ihmisen havaintojärjestelmä ryhmittelee yksittäiset aistiärsykkeet isommiksi kokonaisuuksiksi. Odotukset ja kohteen tuttuus vaikuttavat siihen, kuinka nopeasti kohde tunnistetaan ja luokitellaan. Aistiärsykkeiden yhdistelmistä hahmotetaan yleensä yksinkertaisin ja tutuin. Näitä ilmeisesti synnynnäisiä hahmotustapoja kuvaavat hahmolait. (Sinkkonen ym. 2006, 89–91.) Hahmolakeja ovat seuraavat (kirjaimet viittaavat kuvion 1 kuviin):

- Läheisyys (a): kaksi visuaalista ärsykettä, jotka sijaitsevat lähellä toisiaan, mielletään yhteenkuuluviksi.
- Samanlaisuus (b): kaksi samanlaista visuaalista ärsykettä mielletään yhteen tai samaan ryhmään kuuluviksi.
- Sulkeutuvuus (c): jos visuaaliset ärsykkeet ikään kuin sulkevat sisäänsä jonkin alueen, katsoja näkee sen alueena ja ärsykkeet siihen kuuluvina rajoina.
- Jatkuvuus (d): jos viivat leikkaavat toisiaan, katsoja jakaa kokonaisuuden selkeästi jatkuviin osiin.
- Symmetria (e): symmetrisesti rajatut alueet mielletään yhtenäisiksi kuvioiksi.
- Tuttuus: tutut ja merkitykselliset alueet nähdään kuviona.
- Valiomuotoisuus: kuviot ymmärretään mahdollisimman yksinkertaisina.
- Yhteinen liike: kohteet, jotka liikkuvat samaan suuntaan samalla nopeudella, kuuluvat yhteen ryhmään tai kohteeseen.
- Yhteenliittyminen: kohteet, jotka ovat kiinni toisissaan, kuuluvat yhteen ryhmään tai kohteeseen.

(Sinkkonen ym. 2006, 89–91; Preece 1994, 79–80.)



KUVIO 1. Hahmolait (Preece 1994, 79).

Hahmolait tulee ottaa huomioon käyttöliittymän suunnittelussa, jotta niissä olevat asiat hahmottuisivat käyttäjälle suunnittelijan tarkoittamalla tavalla. Käyttöliittymässä voidaan käyttää esimerkiksi läheisyyden ja samankaltaisuuden lakeja, joilla ilmaistaan yhteenkuuluvat asiat. (Sinkkonen ym. 2006, 91.)

Tarkkaavaisuus

Ihminen pystyy kiinnittämään huomion kerrallaan vain rajalliseen määrään aisti-informaatiota. Prosessoitavaksi tuleva tieto valikoituu tarkkaavaisuuden avulla. Ihmisen tietojenkäsittelykapasiteetin rajallisuus pakottaa aisti-informaation jatkuvaan valikointiin. Valikointi voi olla tietoista, jolloin puhutaan keskittymisestä tai valikoivasta tarkkaavaisuudesta. Huomio voi myös ohjautua automaattisesti ennakkokäsitysten ja odotusten mukaan, jolloin on kyse rutiinitoiminnasta. Suuntautumisrefleksissä huomio kiinnittyy johonkin uuteen, silmiinpistävään tai yllättävään kohteeseen. Käyttäjää voidaan auttaa kiinnittämään huomio oikeisiin asioihin esittämällä tieto sopivassa järjestyksessä ja oikeassa paikassa sekä ryhmittelemällä se semanttisesti ja hierarkkisesti. Monitehtäväympäristössä huomio joudutaan siirtämään nopeasti tehtävästä toiseen tai jakamaan se usealle kohteelle samanaikaisesti, mikä yleensä haittaa tehtävien suorit-

tamista. Tarkkaavaisuuden jakaminen on helpompaa, jos taitotaso on korkea ja toiminta on automatisoitunutta. (Sinkkonen ym. 2006, 97–102.)

Muisti

Muistin hallinnassa on kolme aliprosessia. Tiedot on ensin talletettava muistiin ja liitettävä siellä oikeisiin asioihin, toiseksi säilytettävä muistissa sekä kolmanneksi haettava ja löydettävä muistista. Muistit jaetaan lyhyt- ja pitkäkestoisiin muisteihin. Lyhytkestoisia muisteja ovat *sensorinen muisti*, joka varastoi aistihavainnon hyvin lyhyeksi ajaksi, sekä *työmuisti*, jossa tietoa säilytetään lyhyen aikaa ja käsitellään aktiivisesti kerraten. Työmuistin kapasiteetti on 3–5 mieltämysyksikköä, joka on yksittäisistä asioista muodostunut mielekäs kokonaisuus. Työmuistissa säilyy suurempia ja useampia mieltämysyksiköjä, jos ne ovat ihmiselle mielekkäästi rakentuneita. Pitkäkestoiseen *säilömuistiin* tallentuvat ihmisen tiedot, taidot ja kokemukset. Säilömuisti on pysyvää, mutta muistiin tallentunut tieto usein muuttuu ajan kuluessa. Tiedon tallentamisessa säilömuistiin auttaa tiedon organisointi, toisto ja asioiden yhdistely aiempaan tietopohjaan. (Sinkkonen ym. 2006, 168–176, 193.)

Käsiteverkot, skeemat ja mentaalimallit

Yksinkertaisin tiedon varastointirakenne on *käsiteverkko* eli käsitteiden muodostama hierarkkinen verkosto. Käsitteiden väliset suhteet voivat vaihdella ihmisillä hyvin paljon, eli käsiteverkosto on erilainen eri ihmisillä. Tuotteen käsitteellisen mallin on tärkeää vastata käyttäjän käsitemaailmaa suhteineen, hierarkioineen ja käsitteiden nimineen. Käsiteluokista on kehitetty myös yleisempiä tietorakenteita, joista yleisin on skeema. *Skeema* on järjestäytynyt ja jäsennelty informaatiokokonaisuus, joka sisältää yleistä tietoa kohteen ominaisuuksista. Skeemaverkko on ihmisen käsitys kyseessä olevasta asiakokonaisuudesta ja siinä olevien asioiden suhteista toisiinsa. Asiat liittyvät yhteen assosiaatioiden avulla. Mitä tärkeämpi jokin asiakokonaisuus on ihmiselle, sitä tiheämpi ja laajempi on skeemaverkko. Assosiaation voimakkuus riippuu myös aihealueen ennakolta virittymisestä. (Sinkkonen ym. 2006, 181–185.) *Mentaalimallit* ovat todellisuuden vastineita eli edustuksia ihmisen mielessä. Mentaalimallien avulla käyttäjä pyrkii selittämään tuotteen toimintatavan tai rakenteen. Ne ovat yleensä tiedostamattomia ja ei-kielellisiä. Mentaalimallit ovat todellisen maailman analogioita, dynaamisia, sirpaleisia ja tyypillisesti epätäydellisiä, kun taas skeemat ovat tietomaailmamme kuvauksia, staattisempia ja ristiriidattomia. (Sinkkonen ym. 2006, 207–211.)

2.1.5 Käyttäjäkeskeinen suunnittelu

Käyttäjäkeskeisen suunnittelun (*user-centered design*) lähtökohtana ovat käyttäjät ja heidän tarpeensa (Pettersson & Riihiäho 1996, 36). Gouldin ja Lewisin (1985, 300) mukaan jokaisen ihmisten käyttöön suunnitellun järjestelmän tulisi olla helposti opittava ja muistettava sekä hyödyllinen, eli sen tulee sisältää juuri niitä toimintoja, joita ihmiset työssään tarvitsevat. Järjestelmän käytön tulisi olla helppoa ja miellyttävää. He esittävät käyttäjäkeskeistä suunnittelua varten seuraavat kolme periaatetta:

- **Huomion keskittäminen käyttäjiin ja heidän tehtäviinsä alusta lähtien**
Suunnittelijoiden on perehdyttävä jo suunnittelun alkuvaiheessa käyttäjien ominaisuuksiin ja heidän työtehtäviinsä, joiden suorittamista tulevan järjestelmän tulee tukea.
- **Empiirinen mittaaminen**
Käyttäjien tulee kokeilla varhaisessa vaiheessa järjestelmän käyttöä simulaatioiden ja prototyyppien avulla, ja heidän suoritustaan ja reaktioitaan tulee tarkkailla ja analysoida.
- **Iteratiivinen suunnittelu**
Testauksessa esiin tulleet ongelmat tulee korjata. Suunnittelun tulee olla iteratiivista, eli suunnittelun, testauksen ja mittaamisen sekä uudelleensuunnittelun tulee seurata toisiaan sykleissä niin kauan kuin on tarpeen.

2.1.6 Käytettävyyssuunnittelu

Käyttäjäkeskeisen suunnittelun kanssa läheinen käsite on käytettävyyssuunnittelu tai käytettävyystekniikka (*usability engineering*). Käsitteitä käytetään kirjallisuudessa lähes toistensa synonyymeina. Faulknerin (2000, 12) mukaan käytettävyys, käytettävyyssuunnittelu ja käytettävyyden arviointi muodostavat yhdessä käyttäjäkeskeisen suunnittelun lähestymistavan. Preecen teoksessa on esitetty seuraava Tyldesleyn (1988) määritelmä: ”Käytettävyyssuunnittelu on prosessi, jonka avulla tuotteen käytettävyys määritellään kvantitatiivisesti ja etukäteen. Valmiista tuotteesta voidaan siten osoittaa, onko vaadittu käytettävyyden taso saavutettu.” (Preece 1994, 650.)

Faulkner tarkoittaa käytettävyyssuunnittelulla ohjelmistotuotannon lähestymistapaa, jossa käyttäjät ovat mukana alusta alkaen ja tuotteen tehokkuus taataan käytettävyyss-määrittelyn ja -mittauksen avulla. Suunnitteluprosessi on iteratiivinen, eli siihen kuuluu sykleissä toistuva suunnittelu, arviointi ja sen pohjalta tehty uudelleensuunnittelu. Käytettävyyssuunnittelulla tarkoitetaan koko ohjelmistotuotannon prosessia, jossa pyritään osoittamaan, että valmis tuote todella vastaa asiakkaan tarpeisiin ja toiveisiin. (Faulkner 2000, 12–13.)

Käytettävyyssuunnittelun avulla saadaan konkreettista tietoa käyttäjien tarpeista, heidän työskentelytavoistaan, käyttöympäristöstä ja työskentelyn aikana esiin tulevista ongelmista. Käytettävyyssuunnittelussa pyritään selvittämään tehtävät, joiden suorittamista sovelluksen tulee tukea, sekä vaatimukset, joita käyttäjät ja käyttöympäristö asettavat sovellukselle. Samoin selvitetään, miten nopeasti käyttäjä oppii sovelluksen käytön ja onko käyttö riittävän sujuvaa ja vaivatonta. Edelleen tarkastellaan, millaisia virheitä käytössä tapahtuu ja kuinka käyttäjä selviää niistä. Selvitetään myös, kuinka paljon sovelluksen käyttö kuormittaa käyttäjää ja miten tyytyväisiä käyttäjät ovat sovellukseen. (Koivunen & Nieminen 1996, 12.)

Käytettävyyssuunnittelun eri vaiheista on esitetty erilaisia elinkaarimalleja, joissa käydään vaihe vaiheelta läpi koko ohjelmistosuunnittelun prosessi käytettävyyden kannalta. Taulukossa 1 on esitetty Faulknerin (2000, 15) käytettävyyssuunnittelun elinkaarimalli.

TAULUKKO 1. Käytettävyyssuunnittelun elinkaarimalli (Faulkner 2000, 15)

Tehtävä	Saatu informaatio
Käyttäjän tunteminen	Käyttäjän ominaisuuksien kuvaus Käyttäjän taustan kuvaus
Tehtävän tunteminen	Käyttäjän nykyisen tehtävän kuvaus Tehtäväanalyysi
Käyttäjävaatimusten kerääminen	Käyttäjävaatimukset
Käytettävyystavoitteiden asettaminen	Käytettävyyismäärittely
Suunnitteluprosessi	Suunnittelu
Ohjeiden ja heuristiikan soveltaminen	Palaute suunnittelua varten
Prototyyppien teko	Prototyyppi käyttäjätestausta varten
Käyttjäarviointi	Palaute uudelleensuunnittelua varten
Uudelleensuunnittelu ja käyttjäarviointi	Valmis tuote
Käyttjäarviointi ja raportointi	Palaute tulevaisuuden järjestelmiä varten

Käyttäjistä, heidän tehtävistään ja työympäristöstään saadaan tietoa keskusteluilla, havainnoinnilla, haastatteluilla ja kyselyillä. Samoja menetelmiä käyttäen saadaan tietoa käyttäjien vaatimuksista ja käyttäjien antamasta palautteesta. (Faulkner 2000, 30–36.) Seuraavaksi kuvataan yksityiskohtaisemmin käytettävyyssuunnittelun ensimmäisiä vaiheita.

Käyttäjän ominaisuudet

Käyttäjien tuntemus on tärkeää, sillä esimerkiksi käyttökokemus ja osaaminen vaikuttavat siihen, miten käyttäjä käyttää sovellusta. Aloittelija käyttää järjestelmää eri tavalla kuin asiantuntija, ja kumpikin on otettava huomioon käyttöliittymän suunnittelussa. Aloittelijoita varten on oltava yksinkertainen ja selkeä käyttötapa, kun taas asiantuntija haluaa käyttää toimintaa nopeuttavia pikakomentoja ja lisätoimintoja. Käyttäjät eroavat toisistaan myös iän ja sukupuolen sekä vaikeammin eroteltavien tekijöiden kuten avaruudellisen muistin, päättelykyvyn ja vallitsevan oppimistyylin perusteella. Ihmiset ovat myös kyvyiltään erilaisia. Toiset pystyvät työskentelemään nope-

ammin ja tehokkaammin kuin toiset. Samoin asenteet ja mieltymykset vaikuttavat: osa käyttäjistä on vastahakoisia, kun taas jotkut ovat innokkaita tehokäyttäjiä. (Nielsen 1993, 41–48.) Käyttäjistä on tiedettävä työkokemus, koulutustaso, ikä ja tietokoneen käyttökokemus, jotta voitaisiin ennakoida käytön oppimisen vaikeuksia. On myös otettava huomioon, onko käyttäjillä aikaa tai tilaisuutta käytön opiskeluun. Tarvittaessa käyttöliittymä on tehtävä riittävän yksinkertaiseksi, jotta käytön oppii nopeasti. (Nielsen 1993, 74–75.)

Käyttäjän tehtävät

Aluksi on selvitettävä käyttäjän tavoitteet ja nykyinen toiminta tehtävän suorittamiseksi. Samoin on selvitettävä käyttäjän mentaalimalli tehtävästä, jotta käyttöliittymä saataisiin vastaamaan käyttäjän mielikuvia. Myös nykyisen toiminnan heikot kohdat, kuten missä tulee epäonnistumisia, missä kuluu liikaa aikaa ja mikä on hankalaa käyttää, on selvitettävä, jotta uuteen järjestelmään voitaisiin tehdä parannuksia. Käyttäjiä on pyydetty kuvaamaan työskentelyään ja siinä ilmenneitä ongelmia. Käyttäjiltä on myös kysyttävä, mistä he pitävät nykyisessä toiminnassa ja mistä eivät ja mitä parannusehdotuksia ja toivomuksia heillä on tulevan järjestelmän suhteen. (Nielsen 1993, 75–77.)

Käyttäjävaatimukset

Vaatimusten määrittely on aloitettava käyttöympäristöön perehtymisellä, mikä onnistuu parhaiten havainnoimalla työyhteisön toimintaa. Seuraavaksi on tunnistettava järjestelmän tavoite ja nykyisen järjestelmän ongelmat. Vaatimusmäärittelyyn sisältyy organisaation tarpeiden, järjestelmän toiminnallisten vaatimusten ja teknisten rajoitteiden sekä käyttäjän vaatimusten määrittely. Käyttäjävaatimukset ja järjestelmän toiminnallisuus määritellään käyttäjäkeskeisestä näkökulmasta. Käyttäjävaatimusten määrittely perustuu käyttäjäkuvauksiin ja tehtäväanalyysiin. (Faulkner 2000, 92–96.)

Käytettävyystavoitteet

Käytettävyystavoitteita määriteltäessä täytyy ensinnäkin valita, mitkä käytettävyystekijöistä ovat keskeisimpiä käsillä olevassa suunnitteluprojektissa. Jonkin järjestelmän kohdalla voi olla tärkeää, että sen käyttö on helposti opittavissa, koska työntekijät vaihtuvat usein. Jos taas järjestelmää käytetään harvoin, on tärkeää, että sen käyttö palautuu helposti mieleen. Käytettävyystekijöille määritellään sitten erilaisia suoritus-

tasoja, joista määritellään alin hyväksyttävä, tavoiteltava ja nykyinen taso. Samoin voidaan määritellä teoreettisesti paras mahdollinen taso, vaikka tätä arvoa ei käytännössä saavutettaisikaan. Esimerkkinä voidaan tarkastella käyttäjien tekemien virheiden määrää tuntia kohden. Tavoitetasoksi voidaan määritellä kaksi virhettä, minimitasoksi viisi virhettä ja teoreettiseksi optimitilanteeksi ei yhtään virhettä tuntia kohden. (Nielsen 1993, 80–81.)

2.1.7 Käyttöliittymän suunnittelu

Ihminen ja tietokonejärjestelmä ovat vuorovaikutuksessa keskenään käyttöliittymän avulla. Ihmisen ja tietokoneen välisen vuorovaikutuksen (HCI) tutkimus pohjautuu moneen eri tieteenalaan, joita ovat muun muassa tietojenkäsittelytiede, kognitiivinen psykologia ja ergonomia (Preece 1994, 37). Käyttöliittymän suunnittelu vaatii edellä mainittujen tieteenalojen tuntemusta ja niiden lisäksi tietoa graafisen suunnittelun periaatteista. Käyttäjakeskeisen suunnittelun keskipisteessä ovat käyttäjät ja heidän tarpeensa. Käyttöliittymän suunnittelun apuna voidaan käyttää heuristisia sääntöjä, joita esitellään tarkemmin seuraavassa luvussa. Käyttöliittymän teknistä määrittelyä kuvataan luvussa 2.2.5.

2.1.8 Heuristiset säännöt

Käyttöliittymän suunnittelussa voidaan käyttää apuna erilaisia käytettävyyssperiaatteita eli heuristiikkoja. Näitä ovat Nielsenin ja Molichin kymmenen heuristista sääntöä ja Shneidermanin kahdeksan kultaista sääntöä. Myös Normanin seitsemän periaatetta voidaan lukea heuristisiin sääntöihin. (Faulkner 2000, 188.) Periaatteet ovat keskenään melko samanlaisia ja käsittelevät samoja kysymyksiä. Kirjallisuudessa viitataan useimmin Nielsenin heuristisiin sääntöihin. Seuraavaksi tarkastellaan eri heuristiikkoja lähemmin.

2.1.8.1 Nielsenin ja Molichin kymmenen heuristista sääntöä

Nielsen (1993, 20) on esittänyt teoksessaan kymmenen heuristista sääntöä, joita voidaan käyttää apuna käyttöliittymää suunniteltaessa. Säännöt ovat:

1. Yksinkertainen ja luonnollinen dialogi
2. Käyttäjän kielen puhuminen
3. Käyttäjän muistikuorman minimointi
4. Yhdenmukaisuus
5. Palaute
6. Selkeästi merkityt poistumistiet
7. Oikopolut
8. Hyvät virheilmoitukset
9. Virhetilanteiden estäminen
10. Ohje ja dokumentaatio.

Seuraavaksi esitellään Nielsenin ja Molichin heuristisia sääntöjä tarkemmin ja kuvataan niiden soveltamista käyttöliittymän suunnitteluun (Nielsen 1993, 115–152).

1. Yksinkertainen ja luonnollinen dialogi

Käyttöliittymän tulisi olla mahdollisimman yksinkertainen, koska kaikki ylimääräiset ominaisuudet ja liika informaatio haittaavat tärkeiden asioiden löytymistä. Käyttöliittymän tulisi myös soveltua käyttäjän työtehtävään mahdollisimman luonnollisella tavalla. Järjestelmän ja käyttäjän käyttämien käsitteiden tulisi vastata toisiaan. Käyttöliittymässä tulisi olla vain ja ainoastaan käyttäjän tarvitsema informaatio loogisessa ja sopivassa järjestyksessä. Hyvä graafinen suunnittelu auttaa yksinkertaisen ja luonnollisen käyttöliittymän toteuttamisessa. Hahmolait (esimerkiksi läheisyys, samanlaisuus ja sulkeutuvuus) tulisi ottaa huomioon näyttöjä suunniteltaessa, jotta käyttäjän olisi helpompi ymmärtää käyttöliittymän rakenne ja komponenttien väliset suhteet. Ikkunarakenteessa on hyvä sijoittaa kaikki käyttäjän tarvitsema keskeinen tieto yhteen ja samaan ikkunaan ja vähemmän tärkeä tieto lisäikkunoihin. Värien käytössä on noudatettava kohtuullisuutta niin määrän kuin värikylläisyyden suhteen, sillä liian runsas ja räikeiden värien käyttö herättää levottomuutta. Värisokeat käyttäjät on otettava huomioon siten, että järjestelmää pystyy käyttämään myös ilman värien erottelukykä.

2. Käyttäjän kielen puhuminen

Käyttöliittymän terminologian tulisi perustua käyttäjien omaan kieleen eikä järjestelmän tekniseen sanastoon. Dialogien tulisi myös mielellään olla käyttäjän omalla äidinkielellä. Käsitteistön ja sanaston tulisi olla tuttua käyttäjälle ja perustua hänen

omaan maailmaansa ja mielikuviinsa. Käyttöliittymässä voi olla myös metaforia, joiden avulla järjestelmän käsitteitä voidaan linkittää johonkin arkielämän tuttuun käsitteeseen.

3. Käyttäjän muistikuorman minimointi

Ihmisen muisti toimii siten, että hänen on paljon helpompi tunnistaa jokin asia annetuista vaihtoehtoista kuin muistaa sama asia ilman annettuja vihjeitä. Tämän vuoksi muistamista auttavat käyttöliittymässä olevat valikot, listat ja esitetyt tekstikentät. Järjestelmässä tulisi olla kauttaaltaan yhtenäinen ja yleispätevä säännöstö, mikä helpottaa myös siirtymistä sovelluksesta toiseen ja uusien järjestelmien oppimista.

4. Yhdenmukaisuus

Samalla komennolla tai toimenpiteellä tulisi aina olla sama vaikutus järjestelmän toimintaan. Saman informaation tulisi sijaita aina samassa paikassa näyttöikkunaa ja samalla tavalla muotoiltuna, jotta käyttäjä löytäisi ja tunnistaisi sen helposti. Käyttöliittymästandardien avulla voidaan pitää järjestelmän dialogit yhdenmukaisina.

5. Palaute

Käyttäjän tulisi saada järjestelmältä palaute tekemistään toimenpiteistä ja antamistaan syötteistä. Palaute tulee antaa yhtä lailla onnistuneista kuin epäonnistuneista toiminnoista. Samoin järjestelmän tulee antaa varoitusviesti, jos käyttäjä aikoo epähuomiossa tehdä jonkin peruuttamattoman ja vaarallisen toiminnon, esimerkiksi tiedoston tuhoamisen. Myös jostakin kauan kestävästä toiminnosta, esimerkiksi tiedoston latauksesta, tulee antaa ilmoitus ja arvio jäljellä olevasta ajasta sekä näyttää palkki, joka kuvaa toiminnon edistymistä mahdollisesti prosenttilukujen kera. Järjestelmän virheitä tai ”kaatumisesta” tulee myös antaa ilmoitus eikä jättää käyttäjää arvelemaan, mikä nyt meni pieleen.

6. Selkeästi merkityt poistumistiet

Järjestelmän kaikissa näytöissä ja tiloissa tulisi olla selkeästi esillä peruuta-painike tai jokin muu paluumahdollisuus aikaisempaan tilaan. Kumoa-toiminto on myös hyödyllinen, kun käyttäjä haluaa peruuttaa jonkin tekemänsä muutoksen. Näiden paluutoimintojen ansiosta käyttäjä uskaltaa paremmin kokeilla järjestelmän eri toimintoja ja oppii samalla uutta, koska hän luottaa siihen, että voi aina palata edelliseen tilaan ja

kumota tekemänsä muutokset. Samoin käyttäjän on helpompi selvittää tekemistään virheistä, kun järjestelmä antaa mahdollisuuden kumota ne.

7. Oikopolut

Kokeneille käyttäjille tulisi antaa mahdollisuus käyttää toimintaa nopeuttavia oikopolkuja. Tällaisia ovat pika- ja toimintonäppäimet, objektien kaksoisklikkaukset sekä pikavalikot ja -kuvakkeet. Lista viimeksi käytetyistä komennoista tai tiedostoista nopeuttaa niiden hakua uudelleen käytettäväksi. Valmiiksi annetut oletusarvot nopeuttavat järjestelmän käyttöä ja oppimista, kun käyttäjä ei joudu pohtimaan alusta lähtien kaikkien arvojen asettamista.

8. Hyvät virheilmoitukset

Virheilmoitusten tulee olla selkokielisiä, helposti ymmärrettäviä ja täsmällisiä. Niiden pitäisi auttaa käyttäjää ratkaisemaan ongelmatilanne antamalla ohjeita jatkotoimenpiteistä. Järjestelmä voi myös esittää virheellisen syötön jälkeen arvauksen, mitä käyttäjä itse asiassa tarkoitti sanoa. Virheilmoitusten tulisi olla luonteeltaan rakentavia ja kohteliaita eikä pelotella ja syyllistää käyttäjää. Järjestelmän tulisi myös mahdollistaa hyvä virheistä palautuminen antamalla käyttäjän kumota tekemiään virheitä.

9. Virhetilanteiden estäminen

Järjestelmän tulisi ensisijaisesti estää virhetilanteiden syntyminen. Esimerkiksi kirjoitusvirheistä johtuvia virheitä voidaan välttää antamalla käyttäjän valita listasta oikea vaihtoehto, jolloin hänen ei tarvitse itse kirjoittaa tarvittavaa tietoa. Peruuttamattomia virheitä voidaan vähentää pyytämällä käyttäjältä vahvistus aiottuun toimintoon, esimerkiksi tiedoston tuhoamiseen.

10. Ohje ja dokumentaatio

Järjestelmää tulisi voida käyttää ilman ohjekirjoja, mutta lisä- ja erikoistoimintoja varten ohje on tarpeen. Edistynyt käyttäjä haluaa lisätä järjestelmän tuntemustaan ohjeen avulla. Nykyään käytössä on useimmiten ohjelmaan kuuluva online-ohje ja -dokumentaatio, eikä erillisiä opaskirjoja tarvita. Online-avustuksen etuja on myös sen tilannekohtaisuus, eli ohje löytyy heti ongelmakohdassa. Hyvään ohjeeseen kuuluu sisällysluettelo ja hakemisto, ja sen tulee sisältää järjestelmän oman terminologian lisäksi käyttäjän tehtävien mukaista sanastoa. Sen tulee myös sisältää runsaasti synonyymeja eri käsitteille.

2.1.8.2 Shneidermanin kahdeksan kultaista sääntöä

Shneiderman (1998, 74–75) esittää käyttöliittymän suunnittelun taustaksi seuraavat kahdeksan periaatetta:

1. Pyri yhdenmukaisuuteen

Toimintojen järjestyksen tulisi olla yhdenmukainen samanlaisissa tilanteissa. Kehotteissa, menuissa ja ohjeissa tulisi käyttää samanlaista terminologiaa. Väriytyksen, layoutin, isojen kirjainten käytön ja fontin tulisi olla yhdenmukainen kautta koko käyttöliittymän.

2. Salli tottuneen käyttäjän käyttää oikopolkuja

Tottunut käyttäjä haluaa nopeuttaa toimintaa ja vähentää tarvittavien toimenpiteiden määrää. Hän arvostaa mahdollisuutta käyttää lyhenteitä, erikoisnäppäimiä ja makroja.

3. Anna informatiivista palautetta

Kaikista käyttäjän toiminnoista tulisi antaa palautetta. Usein toistuvista ja pienistä toimenpiteistä annettu palaute voi olla vähäistä, harvemmin toistuvista ja suurista toimenpiteistä tuntuvampaa.

4. Suunnittele dialogit niin, että tehtävät viedään päätökseen

Peräkkäiset toimenpiteet pitäisi järjestää ryhmiin, joilla on alku, keskikohta ja loppu. Käyttäjä saa tyytyväisyyden tunteen loppuunviedystä tehtävästä.

5. Estä virheiden synty ja mahdollista helppo virheiden käsittely

Suunnittele järjestelmä sellaiseksi, että käyttäjä ei voi tehdä vakavia virheitä. Jos käyttäjä tekee virheen, järjestelmän tulisi huomata se ja antaa yksinkertainen, rakentava ja täsmällinen ohje virheen korjaamiseksi.

6. Mahdollista toimintojen helppo kumoaminen

Toimintojen tulisi olla kumottavissa olevia. Käyttäjä tietää silloin, että virheet voidaan kumota, ja uskaltaa kokeilla ennestään tuntemattomia toimintoja.

7. Tue sisäistä kontrollia

Kokeneet käyttäjät haluavat tuntea valvovansa järjestelmän toimintaa ja että järjestelmä vastaa heidän toimintoihinsa, ei päinvastoin.

8. Vähennä lyhytkestoisen muistin kuormitusta

Lyhytkestoisen muistin rajoitukset edellyttävät, että näyttöjen tulee olla yksinkertaisia, siirtymistä ikkunoiden välillä tulee vähentää ja käyttäjä saa online-opastusta.

2.1.8.3 Normanin seitsemän periaatetta

Norman tiivistää suunnittelun tavoitteiksi sen, että (1) käyttäjä ymmärtää, mitä pitää tehdä, ja (2) osaa sanoa, mitä tapahtuu. Suunnittelun tulisi hyödyntää ihmisten ja maailman luonnollisia ominaisuuksia, suhteita ja rajoituksia. Sovelluksen käytössä tulisi tarvita mahdollisimman vähän ohjeita ja selityksiä. Suunnittelussa voidaan käyttää ”seitsemää periaatetta, joiden avulla voidaan muuttaa vaikeat tehtävät yksinkertaisiksi”. (Norman 1988, 188–200.) Periaatteet ovat:

1. Käytä sekä maailmassa olevaa että päänsisäistä tietoa

Ihmiset oppivat helpommin ja tuntevat olonsa mukavammaksi, jos tehtävän suorituksessa tarvittava tieto on saatavissa ulkopuolisesta lähteestä eikä kaikkea tarvitse muistaa. Kun tieto sisäistetään, tehtävän suorituksesta tulee kuitenkin nopeampaa ja tehokkaampaa. Oppimista helpottaa, jos käyttäjälle on muodostunut hyvä käsitteellinen malli järjestelmän toiminnasta.

2. Yksinkertaista tehtävien rakennetta

Tehtävien tulisi olla rakenteeltaan yksinkertaisia, mikä vähentää suunnittelua ja ongelmanratkaisua. Suunnittelijan tulee ottaa huomioon lyhyt- ja pitkäkestoisen muistin ja tarkkaavaisuuden rajoitukset.

3. Tee asiat näkyviksi

Tee käyttäjälle selväksi, mitä toimintoja on mahdollista tehdä ja miten ne suoritetaan. Toimintaa arvioitaessa tee käyttäjälle selväksi, mitä vaikutuksia hänen toimillaan on. Järjestelmän tulisi tarjota aikomusten kanssa yhteen sopivia toimintoja. Järjestelmän tilan tulee olla selkeästi havaittavissa ja tulkittavissa.

4. Tee asioiden yhteydet selviksi

Varmista, että käyttäjä ymmärtää yhteydet, jotka ovat aikomusten ja mahdollisten toimintojen välillä, toimintojen ja niiden vaikutusten välillä, järjestelmän todellisen ja havaitun tilan välillä sekä järjestelmän havaitun tilan ja käyttäjän tarpeiden, aikomusten ja odotusten välillä. Järjestelmän tulee antaa käyttäjälle selkeä palaute hänen tekemistään toimista.

5. Hyödynnä sekä luonnollisten että keinotekkoisten rajoitusten voima

Käytä rajoituksia siten, että käyttäjä kokee, että on vain yksi mahdollinen toimintatapa. Näin käyttäjää johdatetaan toimimaan oikein.

6. Varaudu virhetilanteisiin

Varaudu siihen, että käyttäjä tekee virheitä. Anna käyttäjälle mahdollisuus palautua virhetilanteista ja selvitä hänelle, mitä tapahtui. Tee toimintojen kumoaminen helpoksi ja vaikeuta peruuttamattomien toimintojen tekemistä.

7. Kun kaikki muu epäonnistuu, käytä standardeja

Kun suunnittelussa ei löydy sopivia yhteyksiä kohteiden välille ja kohdataan muitakin vaikeuksia, kannattaa turvautua standardisoiuihin ratkaisuihin. Standardeja voidaan soveltaa toimintojen, layoutin ja näyttöjen suunnittelussa.

2.1.9 Käytettävyyden arviointi

Käytettävyyden arviointia tarvitaan iteratiivisen suunnitteluprosessin eri vaiheissa, kun testataan suunnitelmia ja halutaan käyttäjiltä arvioita järjestelmästä prototyyppien avulla. Suunnitteluperiaatteet ja arviointikriteerit ovat oikeastaan saman asian kaksi eri puolta: suunnitteluperiaatteet ilmaisevat, kuinka järjestelmä pitäisi toteuttaa, ja arvioinnissa selvitetään, kuinka hyvin periaatteita on noudatettu. (Erkiö, Hölttä, Koivunen, Nieminen & Riihiahho 1996, 83.)

Arvioinnin tuloksena saadaan luettelo virheistä ja ongelmista. Virhe tarkoittaa suunnitteluperiaatteiden rikkomista ja ongelma käyttäjän hankalaksi kokemaa kohtaa käyttöliittymässä. (Erkiö ym. 1996, 83.) Käytettävyysongelmat voidaan luokitella niiden vakavuuden mukaan seuraavasti (Nielsen 1993, 103):

0 = ei käytettävyysoongelma

1 = kosmeettinen ongelma – ei tarvitse korjata, ellei ole ylimääräistä aikaa

2 = vähäinen ongelma – korjaamiselle tulee antaa alhainen prioriteetti

3 = suuri ongelma – tärkeää korjata, korjaamiselle annettava korkea prioriteetti

4 = katastrofaalinen ongelma – pakko korjata ennen järjestelmän käyttöönottoa.

Käytettävyyden arvioinnista on eniten hyötyä suunnitteluprosessille, kun arviointeja tehdään riittävän ajoissa. Silloin arvioinnin tuottama palaute auttaa suunnittelijoita löytämään oikeita ratkaisuja, joilla voi parantaa ja varmistaa järjestelmän käytettävyyttä. Tällöin säästyy myös kustannuksia, sillä varhaisessa vaiheessa tehtävät muutokset on helpompi ja halvempi toteuttaa kuin valmiiseen järjestelmään tehtävät korjaukset. (Erkiö ym. 1996, 85.)

Käytettävyyden arviointi voi olla formatiivista tai summatiivista. *Formatiivinen arviointi* tukee suunnitteluprosessia, ja sen avulla pyritään löytämään ja korjaamaan käytettävyysoongelmia. Formatiivinen arviointi on luonteeltaan iteratiivista ja käyttäjakeskeistä, ja sen tulisi käynnistyä mahdollisimman varhaisessa vaiheessa suunnitteluprosessia. Käyttäjien mielipiteitä järjestelmästä kysytään prototyyppien avulla. Formattiivisella arvioinnilla saadaan kvalitatiivista tietoa järjestelmän ominaisuuksista. *Summatiivinen arviointi* kohdistuu valmiiseen järjestelmään, ja sen avulla vertaillaan lopputulosta asetettuihin tavoitteisiin tai kilpaileviin järjestelmiin. Summatiivinen arviointi mittaa järjestelmän käytettävyyttä ja suorituskkyä ja tuottaa kvantitatiivista tietoa. (Erkiö ym. 1996, 90; Faulkner 2000, 138–139.)

Käytettävyyden arviointimenetelmät voidaan jakaa kolmeen eri ryhmään (Erkiö ym. 1996, 87). Arviointimenetelmiä ovat

- käyttäjäpohjaiset menetelmät
- asiantuntija-arviot
- malli- eli teoriapohjaiset menetelmät.

Menetelmät poikkeavat toisistaan muun muassa objektiivisuutensa perusteella. Teoriapohjaiset menetelmät ovat selvimmin objektiivisia, käyttäjien haastatteluun perus-

tuvat arvioinnit taas selvästi subjektiivisia. Asiantuntija-arviot sijoittuvat näiden välille. Tyypillistä on monen menetelmän yhteiskäyttö. (Erkiö ym. 1996, 87–88.)

Käyttäjäpohjaisia arviointimenetelmiä ovat käyttäjien toiminnan havainnointi, haastattelut ja käyttäjäkyselyt. *Havainnoinnissa* tarkastellaan käyttäjien normaalia työskentelyä oikeiden tehtävien parissa oikeassa työympäristössä. Voidaan havainnoida joko nykyisen järjestelmän käyttöä, suunniteltavan järjestelmän käyttöä prototyyppien avulla tai valmiin järjestelmän käyttöä. Havainnointimenetelmä voi vaihdella vapaamuotoisesta havainnoinnista tarkasti määrättyyn klassiseen koejärjestelyyn. *Haastatte- luilla ja kyselyillä* voidaan selvittää käyttäjien tyytyväisyyttä ja mielipiteitä. Niillä saadaan myös tietoa käytettävyysongelmista ja kehitysehdotuksista. (Erkiö ym. 1996, 89, 91, 102.)

Asiantuntijoiden suorittamia arviointeja ovat heuristinen arviointi, yhdenmukaisuuden tarkistus, ryhmäläpikäynti ja kognitiivinen läpikäynti. *Heuristisessa arvioinnissa* käydään läpi käyttöliittymän eri osat ja tarkistetaan, onko käyttöliittymän suunnitteluperiaatteita eli heuristiikkoja noudatettu. Käyttöliittymää voidaan arvioida joko paperiversion, prototyyppien tai valmiin järjestelmän avulla. *Yhdenmukaisuuden tarkistuksessa* järjestelmän käyttöliittymää verrataan muihin saman valmistajan ohjelmiin ja katsotaan, ovatko ne keskenään yhdenmukaisia. Tarkoitus on tutkia, noudattaako käyttöliittymä kyseisen valmistajan käyttöliittymästandardia. *Ryhmäläpikäyntiin* osallistuu käytettävyyssasiantuntijan lisäksi joukko käyttäjiä ja suunnittelijoita. Käyttöliittymää käydään läpi näyttö näytöltä, kukin ryhmänjäsen tekee muistiinpanoja ja lopuksi keskustellaan ja tuodaan esille jokaisen mielipiteet. Ryhmäläpikäynti voidaan tehdä jo suunnittelun alkuvaiheessa käyttäen käyttöliittymän paperiversioita. *Kognitiivisessa läpikäynnissä* käydään esimerkiksi paperiversioiden avulla läpi kaikki sovelluksen tukevat tehtävät. Kunkin tehtävän osalta pohditaan, miten käyttöliittymä tukee käyttäjää sovelluksen oppimisessa, tehtävän vaatimien tavoitteiden asettelussa ja tarvittavien toimintojen löytämisessä. (Erkiö ym. 1996, 108–112.)

Malli- eli teoriapohjaisissa menetelmissä luodaan malli, joka kuvaa ja jäsentää järjestelmän ominaisuuksia ja käyttäjän toimintoja. Tärkeänä tavoitteena on helpottaa käyttäjän toiminnan ymmärtämistä. Käytettävyyden arviointi vaatii mallia, jonka avulla voidaan ennustaa käyttäjän suoritusaikaa, virheiden määrää ja oppimisvaikeuksia. Yleisimmin käytetty on GOMS-malli. (Erkiö ym. 1996, 114.)

Käytettävyydestä koostuu yleensä muutamasta toisiaan täydentävästä arviointimenetelmästä. Keskeisenä menetelmänä on käyttäjän toiminnan havainnointi. Lisäksi käytetään alku- ja loppuhaastatteluja sekä -kyselyitä. Myös heuristinen analyysi ja visuaalinen läpikäynti kuuluvat usein testiin. (Erkiö ym. 1996, 95.) Tyypillisen käytettävyydestin rakenne on seuraava:

- heuristinen analyysi
- alkuhaastattelut ja -kyselyt
- visuaalinen läpikäynti
- käyttäjän toiminnan havainnointi
- loppuhaastattelut ja -kyselyt.

Visuaalisessa läpikäynnissä tarkastellaan, miten hyvin käyttäjät seuraavat käyttöliittymän välittämiä visuaalisia viestejä. Käyttäjää pyydetään kertomaan, mihin hänen huomionsa kiinnittyy ja miten hän ymmärtää ja tulkitsee eri viestit, kuten esimerkiksi värikoodit. Menetelmä antaa myös tietoa termien ymmärtämisestä ja käyttäjien itse käyttämistä käsitteistä. (Erkiö ym. 1996, 96.)

2.2 Tietojärjestelmän kehittäminen

2.2.1 Käsitteiden määrittelyä

Haikalan ja Märijärven teoksessa on esitetty seuraavanlainen Naurin ja Randellin (1969) määritelmä: ohjelmistotuotanto (*software engineering*) tarkoittaa kestävien suunnitteluperiaatteiden vakiinnuttamista ja käyttämistä, jotta voitaisiin taloudellisesti kehittää ohjelmistoja, jotka ovat luotettavia ja toimivat tehokkaasti oikeissa tietokoneissa. Ohjelmistotuotanto käsittää kaikki ohjelmiston tuotantoprosessiin liittyvät osa-alueet, jotka ovat laatuja järjestelmä, projektinhallinta, dokumentointi, tuotteenhallinta, laadunvarmistus, testaus, määrittely, suunnittelu, toteutus, käyttöönotto ja ylläpito. (Haikala & Märijärvi 2004, 16.)

Tietojärjestelmä (*information system*) taas viittaa ohjelmistoja laajempaan kokonaisuuteen. Se on ihmisistä, tietojenkäsittelylaitteista, tiedonsiirtolaitteista ja ohjelmista koostuva järjestelmä, jonka tarkoitus on tietoja käsittelemällä tehostaa tai helpottaa

jotakin toimintaa tai tehdä toiminta mahdolliseksi. Tietojärjestelmän käsitteeseen sisältyvät myös tietojenkäsittelyn ympäristön organisaationaaliset, sosiaaliset ja inhimilliset ulottuvuudet. Ohjelmistoprosessilla (*software process*) ymmärretään kokonaisuutta, jolla ohjataan tietojärjestelmän kehittämistä idean syntyhetkestä aina järjestelmästä luopumiseen saakka. Ohjelmistoprosessi kattaa siis koko järjestelmän elinkaarren. (Pohjonen 2002, 5–6, 21.)

2.2.2 Tietojärjestelmän kehittämisvaiheet

Tietojärjestelmän kehittämisvaiheisiin kuuluu ensin esitutkimus, jonka tarkoitus on selvittää, onko tietojärjestelmän rakentaminen ylipäättään mahdollista ja mielekasta. Kehittämisratkaisun kannalle päätynyttä esitutkimusta seuraa vaatimusmäärittely, analyysi, suunnittelu, toteutus ja testaus. Järjestelmän käyttöönottoa seuraa vielä ylläpito-vaihe. (Pohjonen 2002, 26–27.) Seuraavaksi tarkastellaan näitä vaihteita lähemmin.

Vaatimusmäärittelyssä kartoitetaan järjestelmän tulevien käyttäjien järjestelmälle asettamat toiminnalliset ja ei-toiminnalliset vaatimukset. Järjestelmän tekniseen toteutukseen ei oteta vielä kantaa. Toiminnalliset vaatimukset määrittelevät sen, mitä järjestelmän odotetaan tekevän eli mitä käyttäjien tehtäviä sen tulee tukea. Ei-toiminnalliset vaatimukset koskevat toiminnan reunaehdoja, kuten esimerkiksi teknisiä ominaisuuksia ja käytettävyyttä. Käyttäjävaatimukset voidaan kerätä haastattelemalla ja havainnoimalla käyttäjiä ja työympäristöä. Vaatimusmäärittelyn dokumentointiin tulee sisältyä kuvaus kehittämishankkeen toimeksiannosta, nykytilanteesta, kohdejärjestelmästä, tavoitteista ja kaikista vaatimuksista. (Pohjonen 2002, 28–31.)

Järjestelmäanalyysissa tehdään vaatimusmäärittelyn pohjalta järjestelmän toiminnallinen määrittely. Tämä tehdään toteutusriippumattomalla tavalla. Tarkoituksena on selvittää, mitä järjestelmän tulee tehdä, ja luoda kuvaus järjestelmän toiminnasta yleisellä tasolla. Toiminnallisen määrittelyn tulee sisältää myös kuvaukset muun muassa käyttäjistä, jokaisesta toiminnasta, käsiteltävistä tiedoista ja tietokannoista. (Pohjonen 2002, 31–32.)

Suunnitteluvaiheessa pohditaan, miten järjestelmä toteutetaan. Suunnittelussa tehdään järjestelmän tekninen määrittely toiminnallisen määrittelyn pohjalta. Suunnittelu jakautuu yleensä arkkitehtuuri- ja moduulisuunnitteluun. Arkkitehtuuri-suunnittelussa

määritellään järjestelmän yleinen rakenne, jonka jälkeen kokonaisuus jaetaan itse-
näisiin moduuleihin. Seuraavaksi on vuorossa toteutusvaihe, jossa ohjelmisto toteute-
taan jollain ohjelmointikielellä ja ohjelmamoduulit integroidaan toimivaksi kokonai-
suudeksi. Lopuksi seuraa testaus, käyttöönotto ja ylläpito. (Pohjonen 2002, 32–37.)

2.2.3 Elinkaarimallit

Tietojärjestelmän kehitysprosessia voidaan kuvata elinkaarimalleilla, joissa prosessin
eri vaiheet on koottu järjestelmälliseksi kokonaisuudeksi. Perinteisin ja yleisin on *ve-
siputousmalli*, jossa ohjelmistotuotanto kuvataan suoraviivaisesti etenevänä prosessina
esitutkimuksesta ylläpitoon. Suunnitteluprosessille tyypillinen iteratiivisuus ei tule
erityisen hyvin esiin. Asiakkaalle ei pystytä esittämään varsinaisia tuloksia kuin vasta
prosessin myöhäisessä vaiheessa. (Pohjonen 2002, 39–40.)

Prototyypilähestymistavassa järjestelmästä tuotetaan jo varhaisessa vaiheessa epä-
täydellinen prototyyppi, jonka asiakas saa arvioitavakseen. Prototyypissä on rajoite-
tusti toiminnallisia ominaisuuksia tai se voi olla pelkkä paperiversio (Nielsen 1993,
97). Prototyyppi voidaan tehdä jo vaatimusmäärittelyn jälkeen ja sitä parannellaan
saadun käyttäjäpalautteen perusteella. Tämä toistuu, kunnes ollaan valmiita etenemään
toteutusvaiheeseen. (Pohjonen 2002, 41.)

Spiraalimallin keskeinen ajatus on ohjelmistoprosessin iteratiivisuus. Toinen keskei-
nen piirre on prosessiin liittyvien riskien jatkuva analysoiminen ja prosessin uudelleen
ohjaaminen riskianalyysin tulosten mukaan. (Pohjonen 2002, 42.) Toinen syklinen
malli on *Evo-malli*, jonka ideana on rakentaa ensimmäisessä projektissa ydinjärjestel-
mä, jota sitten seuraavissa projekteissa kehitetään edelleen (Haikala & Märijärvi 2004,
45).

2.2.4 Oliokeskeinen lähestymistapa ja mallintaminen

Oliosuuntautunut, oliokeskeinen tai oliopohjainen (*object-oriented*) tietojärjestelmien
kehittäminen nousi 1990-luvulla keskeisimmäksi kehitystyön menetelmäksi. Oliolä-
hestymistavan juuret juontavat jo 1960-luvulle, mutta järjestelmien raskaus esti olio-
ohjelmoinnin yleistymisen, kunnes teknologia kehittyi tarpeeksi 1980-luvun lopussa.

Oliosuunnittelussa käytetään yleisesti *UML (Unified Modeling Language)* -mallinnuskieltä. (Pohjonen 2002, 144, 150.)

Vaatimusmäärittelyssä kuvataan järjestelmälle asetettavat vaatimukset tekstimuotoisten käyttötapausten eli *use case* -kuvausten avulla. Kukin yksittäinen use case kuvaa tietyn toiminnon tai toimintasarjan järjestelmän ja aktorien eli käyttäjäroolien välisenä vuorovaikutuksena. Graafinen use case -kaavio esittää koottuna järjestelmän use case:t, aktorit ja niiden väliset suhteet. (Pohjonen 2002, 152.)

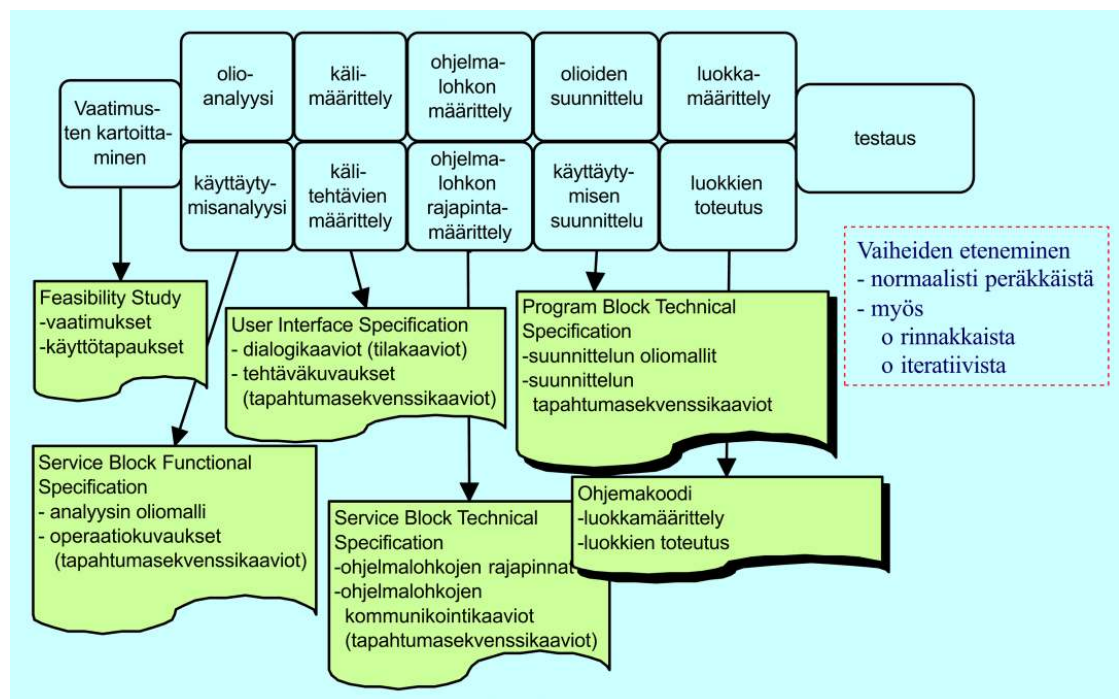
Analyysivaiheessa järjestelmää tarkastellaan sen staattisen rakenteen ja dynaamisen käyttäytymisen näkökulmista. Toteutukseen ei oteta vielä kantaa, vaan keskitytään järjestelmäratkaisun olennaisten osien hahmottamiseen ja kuvaamiseen. Oliomallintamisen keskeinen väline on luokkakaavio, joka kuvaa järjestelmän staattista rakennetta. Sekvenssikaaviot mallintavat järjestelmän dynaamista käyttäytymistä. Jokaisen käyttötapauksen pohjalta laaditaan sekvenssikaavio, joka kuvaa luokkien välistä kanssakäymistä. (Pohjonen 2002, 165, 167.)

Rakennettavasta järjestelmästä voidaan laatia analyysivaiheessa myös tietomalli, jossa kuvataan järjestelmän sisältämät tiedot, niiden staattinen rakenne sekä niiden väliset suhteet ja riippuvuudet. Yleisesti käytetty tietomallin kuvaustapa on ER-kaavio (*Entity–Relationship Diagram*), joka koostuu kahdesta peruskomponenttityypistä, kohteista ja niiden välisistä suhteista. ER-kaaviossa esitetään yleensä myös kohteiden tai suhteiden ominaisuudet eli attribootit, jolloin siitä voidaan käyttää nimitystä ERA-kaavio (*Entity–Relationship–Attribute Diagram*). (Pohjonen 2002, 120.) Kohteet ovat verrattavissa oliomallinnuksen luokkiin.

Suunnitteluvaihe aloitetaan käyttöliittymän suunnittelulla, jolloin keskeinen tehtävä on näyttöjen suunnittelu. Apuna voidaan käyttää protoilutyökaluja ja tietovirtakaavioita. Analyysivaiheen luokkakaavio täydennetään suunnittelun luokkakaavioksi ottamalla mukaan käyttöliittymän luokat ja täydentämällä muita luokkia. Samoin analyysivaiheen sekvenssikaavioita täydennetään uusilla luokilla. (Pohjonen 2002, 169–171.) Lopuksi ohjelmisto toteutetaan jollain oliokielellä ja testataan.

2.2.5 OMT++

Yksi esimerkki oliokeskeisistä menetelmistä on Suomessa kehitetty OMT++, joka perustuu OMT-, Fusion- ja OOSE-menetelmiin. OMT on lyhenne sanoista Object Modeling Technique. OMT++ -menetelmälle on ominaista käyttäliittymän keskeinen asema sovelluskehityksessä. Sovelluksen kehitysprosessi alkaa vaatimusten keruulla, jota seuraa määrittely, suunnittelu, ohjelmointi ja lopuksi testaus. Määrittelyvaiheessa tuotetaan määrittelyoliomalli, järjestelmän käyttäytymisen määrittely ja käyttäliittymän määrittely, jossa määritellään käyttäjälle näkyvä osuus järjestelmästä. Suunnitteluvaiheessa määrittelyoliomalli jalostuu suunnitteluoliomalliksi, luodaan tapahtuma-kaaviot ja suunnitellaan käyttäliittymäluokat. Ohjelmointi toteutetaan kirjoittamalla luokkamäärittelyt ja luokkien toteutus jollain ohjelmointikielellä. (Jaaksi & Pettersson 1996, 62–64.) OMT++ -menetelmän vaiheet esitetään kuviossa 2.



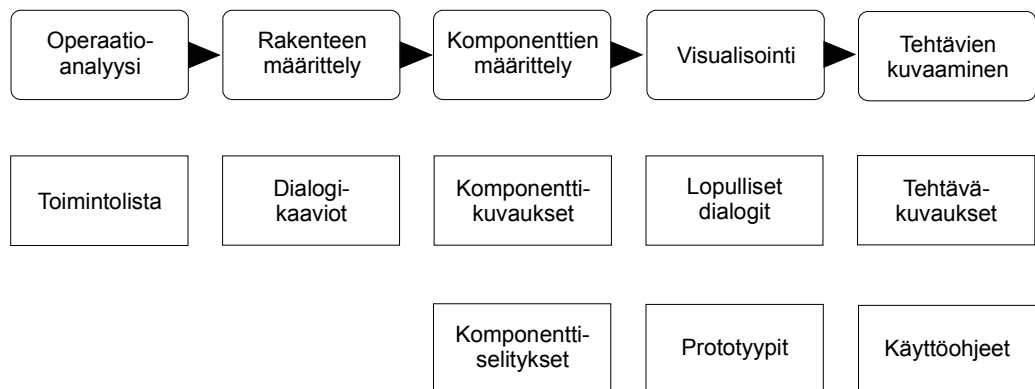
KUVIO 2. OMT++ -menetelmän vaiheet Jaaksin mukaan (Jaakkola 2002).

OMT++ sitoo käyttäliittymän määrittelyn, suunnittelun ja toteutuksen kiinteäksi osaksi ohjelmistoprosessia. Käyttäliittymän suunnittelu etenee rinnakkaisena prosessina muun ohjelmistosuunnittelun kanssa, jolloin voidaan varmistaa myös ohjelmiston käy-

tettävyys. Käyttöliittymän määrittelyn lähtökohtana on käyttäjän sovellukselle asettamien toiminnallisten vaatimusten analyysi. Tavoitteena on käyttöliittymä, jolla tehtävien suoritus on mahdollisimman tehokasta ja joustavaa. (Jaaksi & Pettersson 1996, 46–47, 64–65.)

Käyttöliittymän määrittelyä edeltää oliomäärittely, joka tuottaa määrittelyoliomallin, ja käyttäytymisanalyysi, jossa kuvataan kaikki järjestelmän operaatiot käyttäjän kannalta tarkasteltuna. Operaatio on jokin käyttäjän suorittama mielekäs tehtäväkokonaisuus, joka asettaa järjestelmän toiminnallisen vaatimuksen. (Jaaksi & Pettersson 1996, 69; Haikala & Märijärvi 2004, 383.)

Käyttöliittymän määrittely etenee käytännössä kuviossa 3 esitetyllä tavalla. Käyttöliittymän määrittely alkaa operaatioanalyysillä eli käyttäjälle tarjottavien tehtävien keräämisellä ja analysoinnilla. Seuraavaksi määritellään käyttöliittymän perusrakenne ja komponentit. Vasta sitten siirrytään käyttöliittymän visualisointiin ja lopulliseen ulkonäköön. Lopuksi käyttöliittymällä suoritettavat tehtävät dokumentoidaan esimerkiksi käyttöohjeessa. (Jaaksi & Pettersson 1996, 68.)



KUVIO 3. Käyttöliittymän määrittelyn vaiheet (Jaaksi & Pettersson 1996, 68).

Käyttöliittymän määrittelyn ensimmäinen vaihe on operaatioanalyysi, jossa operaatiot jaetaan pienemmiksi tehtäviksi. Kokoamalla yhteen kaikkien operaatioiden tehtävät saadaan lista käyttöliittymässä esiintyvistä tehtävistä. Käyttöliittymän rakenteen mää-

rittelyssä kuvataan dialogihierarkia dialogikaaviolla. Dialogihierarkia suunnitellaan käyttäjien tehtävien pohjalta, koska sen on tuettava niiden suorittamista. Dialogikaavio on kevyesti muokattu Harelin tilakaavio, jonka tilat ovat käyttöliittymän dialogeja ja tilasiirtymät siirtymiä dialogien välillä. Myös komponenttien määrittelyn lähtökohdina ovat käyttöliittymällä suoritettavat tehtävät. Kukin dialogi varustetaan sellaisilla komponenteilla, jotka ovat tarpeen tehtävien tehokkaan suorittamisen kannalta. (Jaaksi & Pettersson 1996, 70–72; Haikala & Märijärvi 2004, 384.)

Käyttöliittymän visualisointi tehdään vasta määrittelyn loppuvaiheessa, koska sitä tulee edeltää huolellinen tehtävä- ja rakennemäärittely. Liian aikaisin tehdyt prototyyppit vievät kehitysprosessia liikaa toteutusläheisiin yksityiskohtiin ja sattumanvaraisiin valintoihin. Tällöin käyttöliittymän suuret ja tärkeät linjat jäävät huomiotta, mikä johtaa helposti epäyhtenäisyyteen ja huonoon käytettävyyteen. (Jaaksi & Pettersson 1996, 78–79.)

3 TUTKIMUSASETELMA

3.1 Tutkimuksen tavoitteet ja rajaukset

Tutkimuksen tavoitteena on suunnitella tietokantapohjainen tilavarausjärjestelmä Jyväskylän käyttöön. Keskeisenä näkökulmana on käytettävyyden huomioon ottaminen suunnitteluprosessissa. Tavoitteena on siis suunnitella käytettävyydeltään hyvä järjestelmä, joka helpottaa, nopeuttaa ja tehostaa toimistotyöntekijöiden työtä. Tilavarausjärjestelmän tehtäviä ovat

- uuden tilavarauksen tekeminen
- tilavarauksen tietojen muuttaminen ja poistaminen
- tilavaraustietojen hakeminen
- tilavaraustietojen tulostaminen
- tietokannan muiden tietojen lisääminen, muuttaminen, poistaminen, hakeminen ja tulostaminen.

Järjestelmän tulevia käyttäjiä ovat kurssisihteeri ja toimistohenkilö, joiden tehtäviin kuuluu muun muassa kurssien ja tapahtumien tarvitsemien tilojen varaaminen ja järjestely. Tilavarausjärjestelmä asennetaan joko yhdelle koneelle paikallista käyttöä varten tai palvelimelle internetin tai intranetin kautta käytettäväksi. Tämä riippuu käyttäjien toiveista ja selviää haastattelujen jälkeen. Jyväskylässä on lähiverkko ja vähän aikaa sitten hankittu oma palvelin.

Opinnäytetyö rajataan koskemaan tilavarausjärjestelmän vaatimusmäärittelyä ja järjestelmäanalyysiä, jolloin suunnittelu- ja toteutusvaihe jää sen ulkopuolelle myöhemmin toteutettavaksi.

3.2 Tutkimusmenetelmät

Tutkimusmenetelmänä on *laadullinen eli kvalitatiivinen tutkimus*. Laadullinen tutkimus on luonteeltaan kokonaisvaltaista tiedon hankintaa, ja tutkimusaineisto koostaan luonnollisissa, todellisissa tilanteissa. Ihminen on keskeinen tiedon keruun väline.

Laadullisen tutkimuksen lähtökohtana ei ole teorian tai hypoteesien testaaminen vaan uuden tiedon löytäminen tarkastelemalla aineistoa monitahoisesti. Tavoitteena on tuoda esiin tutkittavien näkökulmat, ja tutkimusmetodeja ovat muun muassa teemahaastattelu, ryhmähaastattelu ja osallistuva havainnointi. Teemahaastattelussa pyritään aihepiiriin sisällä, mutta kysymysten muotoa ja järjestystä ei ole tarkkaan määritetty. Laadullisen tutkimuksen kohdejoukko valitaan tarkoituksenmukaisesti eikä satunnaisotannalla. Tutkimussuunnitelma muotoutuu tutkimuksen edetessä, ja tutkimus toteutetaan joustavasti tarvittaessa suunnitelmia muuttaen. (Hirsjärvi, Remes & Sajavaara 2003, 155, 195.)

Tilavarausjärjestelmän suunnitteluprosessi toteutetaan *tietojärjestelmän kehittämisen ja oliopohjaisen ohjelmistotuotannon menetelmillä*. Lähtökohtana on käyttäjäkeskeinen suunnittelu. Tietojärjestelmän kehittämisvaiheisiin kuuluvat vaatimusmäärittely, järjestelmäanalyysi, suunnittelu, toteutus ja testaus. Tässä työssä toteutetaan vaatimusmäärittely ja järjestelmäanalyysi. Eri vaiheissa luodaan kuvauksia käyttäjien ja järjestelmän toiminnasta oliopohjaisessa ohjelmistosuunnittelussa käytetyillä mallinustavoilla.

3.3 Tutkimuskysymykset

Tutkimus toteutetaan painottaen käytettävyyšnäkökulmaa. Tutkimuskysymykset ovat:

1. Millainen on tilaajan käyttöön hyvin sopiva tilavarausjärjestelmä?

Tutkimuksessa suunnitellaan tilavarausjärjestelmä Jyväskylän käyttöön. Aluksi selvitetään, mitkä ovat käyttäjien järjestelmään kohdistuvat vaatimukset. Etsitään kaikki esiin tulevat käyttötapaukset ja kuvataan ne. Vaatimusten pohjalta ratkaistaan, mitä järjestelmän tulee tehdä, ja luodaan kuvaukset järjestelmän toiminnasta.

2. Miten käytettävyys otetaan huomioon järjestelmän suunnittelussa?

Tavoitteena on suunnitella tilavarausjärjestelmä, jolla on hyvä käytettävyys. Käyttäjiä haastatteleamalla selvitetään, millaisia heidän tarpeensa ja tehtävänsä ovat sekä mitä toiveita ja odotuksia he kohdistavat järjestelmään. Samoin selvitetään, mitä ongelmia nykyiseen toimintaan liittyy ja millainen käyttöympäristö on. Suunnitteluprosessissa

pohditaan, millainen järjestelmä voisi helpottaa, nopeuttaa ja tehostaa käyttäjien työskentelyä. Mietitään, miten järjestelmästä saataisiin helppo- ja mukavakäyttöinen.

3. Millainen on järjestelmän tekninen toteutus?

Tutkimuksessa suunnitellaan alustavasti järjestelmän rakenne ja tekninen toteutus.

Mietitään, miten järjestelmä täyttää asetetut vaatimukset ja miten kehitetään tilavausjärjestelmä, joka vastaa hyvin Jyvälän tarpeisiin.

4 TUTKIMUKSEN TOTEUTUS

Tutkimuksen lähtökohtana oli käyttäjäkeskeinen suunnittelu, jossa käyttäjät ja heidän vaatimuksensa ovat keskeisessä asemassa. Käytettävyyssuunnittelun mukaisesti perehdyttiin ensin käyttäjien ominaisuuksiin, tehtäviin ja työympäristöön. Tämän kartoituksen pohjalta selvitettiin käyttäjien tarpeet ja vaatimukset ja määritettiin käytettävyystavoitteet. Tutkimusmenetelmänä oli laadullinen tutkimus, koska testattavana ei ollut mitään hypoteesia, vaan tarkoitus oli kerätä tietoa käyttäjistä ja työympäristöstä monitahoisen tarkastelun avulla. Laadullisen tutkimuksen menetelminä olivat havainnointi ja teemahaastattelu. Järjestelmän tekninen määrittely toteutettiin käyttäjävaatimusten perusteella. Määrittelyssä käytettiin tietojärjestelmän kehittämisen ja oliokeskeisen ohjelmistotuotannon menetelmiä. Oliokeskeisyys valittiin, koska se on nykyään keskeisin kehitysmenetelmä.

4.1 Käyttäjäkeskeinen suunnittelu

Käyttäjien ominaisuuksiin, tehtäviin ja työympäristöön perehdyttiin havainnoimalla ja haastatteleamalla käyttäjiä. Koska Jyväskylässä on tällä hetkellä vain kaksi henkilöä, jotka käyttävät tilavarausta, ei luonnollisestikaan tarvinnut pohtia käyttäjäryhmiin jakautumista ja kohdehenkilöiden valintaa. Havainnointi ja haastattelut oli helppo toteuttaa, koska tutkimuksen tekijä työskentelee samassa työpaikassa. Haastattelussa käytettiin teemahaastattelua, jossa kysymysten muoto ja järjestys ei ole tarkka, vaan eteneminen tapahtui osittain vapaamuotoisella keskustelulla. Haastattelun runko on esitetty liitteessä 1. Haastattelut nauhoitettiin ja purettiin tekstiksi. Havainnointi tapahtui jokapäiväisen työskentelyn ohessa toimistossa asioidessani ja tarvitessani tietoa tilavaruuksista. Käyttäjät myös esittelivät nykyisen järjestelmän käyttöä.

Käyttäjävaatimusten kerääminen tehtiin analysoimalla keskusteluja ja haastatteluja. Työympäristön ja nykyisen järjestelmän käytön havainnointi auttoi käyttäjävaatimusten keräämisessä. Käyttäjien haastatteluista ja havainnoinnista saadun tiedon perusteella määriteltiin myös käytettävyystavoitteet.

4.2 Tietojärjestelmän kehittäminen

Tilavarausjärjestelmän kehittämisessä käytettiin oliokeskeistä lähestymistapaa. Tutkimus rajattiin koskemaan vaatimusmäärittelyä ja järjestelmäanalyysiä. Järjestelmän vaatimusmäärittely tehtiin käyttäjävaatimusten perusteella. Vaatimusmäärittelyssä (ks. liite 2) kartoitettiin järjestelmän toiminnalliset ja ei-toiminnalliset vaatimukset sekä kuvattiin käyttötapaukset ja laadittiin use case -kaavio. Seuraavaksi tehtiin järjestelmäanalyysi eli järjestelmän toiminnallinen määrittely. Aluksi luotiin järjestelmän käsite- eli tietomalli ja tehtiin oliomäärittely. Tietomallia kuvattiin ER-kaaviolla (ks. liite 3) ja oliorakennetta luokkakaaviolla eli määrittelyoliomallilla (ks. liite 4). Käsiteltävistä tiedoista tehtiin taulukuvaukset (ks. liite 5) ja tietokannasta tehtiin kaavio (ks. liite 6). Käyttöliittymän määrittely aloitettiin OMT++ -menetelmän mukaisesti operaatioanalyysillä, jossa operaatiot jaettiin pienemmiksi tehtäviksi. Tehtävät listattiin (ks. liite 7) ja jaettiin dialogeihin, jonka jälkeen dialogien välinen hierarkia kuvattiin dialogikaaviolla (ks. liite 8). Järjestelmän mallinnuksessa käytettiin pääosin UML-mallinnuskieltä.

Työvälineinä käytettiin CorelDRAW 12 -vektorigrafiikkaohjelmaa, jolla tehtiin ER-kaavio, ja Microsoft Office Visio 2007 -kokeiluversiota, jolla tehtiin luokkakaavio, tietokantakaavio ja dialogikaavio. Muuten työvälineenä käytettiin Microsoft Word 2002 -tekstinkäsittelyohjelmaa.

5 TUTKIMUKSEN TULOKSET JA ANALYYSI

5.1 Käyttäjäkeskeinen suunnittelu

Tutkimuksen tavoitteena oli suunnitella tilavarausjärjestelmä Jyvälän käyttöön. Lähtökohtana oli käyttäjäkeskeinen suunnittelu ja pyrkimys hyvään käytettävyyteen. Tavoitteena oli suunnitella järjestelmä, joka olisi helposti opittava ja juuri käyttäjien tarpeisiin suunniteltu. Käytettävyyssuunnittelun muita tavoitteita on taata tuotteen tuotavuus, tehokkuus ja käyttäjien tyytyväisyys. Tutkimuksen aluksi perehdyttiin käyttäjien ominaisuuksiin, tehtäviin ja työympäristöön käyttäjien haastattelun ja havainnoinnin avulla. Näiden pohjalta kartoitettiin käyttäjien tarpeet ja vaatimukset ja määriteltiin käytettävyystavoitteet.

5.1.1 Käyttäjäkuvaukset

Jyvälän tilavarausjärjestelmän tulevia käyttäjiä on kaksi henkilöä, joista toinen on 43-vuotias koulutussuunnittelija ja toinen 25-vuotias toimistotyöntekijä. Käyttäjien ominaisuudet selvitettiin teemahaastattelulla, jonka kysymysrunko on esitetty liitteessä 1. Haastattelut tehtiin maalisi- ja huhtikuussa 2008.

Käyttäjä A

Käyttäjä A on 43-vuotias naispuolinen kansalaisopiston koulutussuunnittelija. Työtehtäviin kuuluu kurssien ja koulutusten suunnittelua, käytännön toteutusta ja arviointia. Myös asiakaspalvelu, neuvottelut opettajien kanssa ja luokkajärjestelyt kuuluvat A:n tehtäviin. Tilavarausjärjestelmän käyttö on lähinnä A:n vastuulla, koska hän suunnittelee kansalaisopiston kurssit ja kansalaisopisto on tilojen suurin käyttäjä. A on vakituinen työntekijä, kun taas työllistämishankkeiden kautta tuleva toimistohenkilökunta vaihtuu noin vuoden välein. A:lla on vaihteleva kokemus tietokoneiden käytöstä. Hän käytti jonkin verran tietokoneita 1990-luvun alussa, mutta koneet olivat silloin melko erilaisia kuin nykyään. Palattuaan työelämäänsä 2000-luvulla A kävi toimisto-ohjelmien peruskursseja ja on oppinut niiden käyttöä lisää työssä tekemällä. A arvioi itsensä keskitasoiseksi tietokoneen käyttäjäksi. Taidot ovat sellaisia, että niillä pärjää hyvin toimistotöissä. Suurempia ongelmia ei ole koneiden käytössä tullut vastaan, paitsi

erään koneen ja siinä olleen tietokannan ”kaatuminen” viime syksynä. Joitakin pienempiä ongelmia tulee silloin tällöin vastaan, mutta A ei koe niitä häiritsevinä. A pitää työskentelystä tietokoneella, kunhan työhön kuuluu muutakin. Pelkkää koneen ääressä istumista A ei haluaisi tehdä.

Käyttäjä B

Käyttäjä B on 25-vuotias naispuolinen toimistotyöntekijä, joka on työskennellyt talossa noin vuoden ajan. Työtehtäviin kuuluu keskeisenä osana asiakaspalvelu, joka tapahtuu puhelimitse tai toimistossa paikan päällä. Muita tehtäviä ovat laskutusten ja tilavarausten teko tietokoneella sekä kaikenlaiset muut toimistotyöt. B on käyttänyt tietokoneita pienestä pitäen. Koulussa oli erilaisia toimisto-ohjelmien kursseja. B arvioi itsensä hyväksi keskitason käyttäjäksi. Mitään suurempia ongelmia ei tietokoneen käytössä ole ollut, ja B pitää tietokoneella työskentelystä.

5.1.2 Nykyisen tehtävän kuvaus

Nykyinen tilavarausjärjestelmä on Quattro Pro -taulukkolaskentaohjelmalla tehty lomake. Kaikista Jyvälän tiloista on tehty lukujärjestyksen näköinen viikkotaulukko, jossa on ylhäällä tilan nimi, vaakasuorassa viikonpäivät ja pystysuorassa kellonajat. Kaikki tilat ovat samalla viikkosivulla, ja eri viikot ovat tiedoston alasivuilla. Tilavaraukset syötetään lomakkeelle kaikista Jyvälän tapahtumista, joita ovat kansalaisopiston kurssit, nuorten kerhotoiminta, projektit, palaverit ja muut tapahtumat. Tiedostossa on kerrallaan koko lukukausi, ja kurssit ja kerhot syötetään lomakkeelle lukukausien alussa. Syöttö on manuaalista ja tiedot siirretään kopioimalla viikosta toiseen. Seuraavan viikon tilavaraukset tulostetaan perjantaisin toimiston ilmoitustaululle tai lähetetään sähköpostitse kauempana olevaan toimipisteeseen.

Tilavaraustaulukosta on yhdellä koneella varsinainen päivitetty versio, jonka voi nähdä lähiverkon kautta myös toisilta koneilta. Varmuuskopio on toisella koneella, mutta se ei ole aina ajan tasalla. Taloon on hankittu hiljattain oma palvelin, joten järjestelmä voidaan tulevaisuudessa asentaa sinne.

Työnjako on sellainen, että käyttäjä A hoitaa kurssien syötön lukukausien alussa ja käyttäjä B huolehtii pääasiallisesti päivittäisistä muokkauksista ja lisäyksistä, tuloksista ja sähköpostilla lähettämisestä. Käyttäjä A käyttää järjestelmää lähes päivit-

täin, koska tietoihin ja tapahtumiin tulee muutoksia ja tarkistuksia. Aikaa kuluu noin yksi, korkeintaan kaksi tuntia viikossa, paitsi lukukausien alussa, jolloin kurssien syöttö kestää enemmän kuin yhden työpäivän. Myös käyttäjä B käyttää järjestelmää päivittäin lisätessään ja muokatessaan tapahtumia. Viikoittain kuluu aikaa kahdesta kolmeen tuntia.

Kumpikin käyttäjästä pitää nykyistä järjestelmää helppo- ja nopeakäyttöisenä ja selkeänä. Erityisesti he pitävät siitä, että kaikki tilat ovat samalla sivulla ja näkyvät nopeasti yhdellä silmäyksellä alasvierittämällä. Tilavarausten hallinta on helppoa nykyisellä systeemillä, koska tiloja on vain yksitoista. Ainoastaan lukukausien alussa tapahtuva varausten syöttö on työlästä ja hidasta. Virhemahdollisuuksia on siinä, että syöttö voi mennä väärälle riville tai väärään sarakkeeseen, mutta virheitä tulee harvoin. Ainoastaan se on ollut ongelmana, että joskus sarakkeita ja tietoa katoaa oudosti, mitä käyttäjät pitävät runsaasta kopiointista tai sitten ohjelman virheistä johtuvana. Samoin fontti saattaa muuttua ”itsestään” kopiointin myötä. Muuten käyttäjät ovat tyytyväisiä nykyjärjestelmään.

5.1.3 Käyttöympäristön kuvaus

Tilavarausjärjestelmää käytetään toimistotilassa, joka on pienehkö huone ja avointa tilaa. Työntekijöitä on tilassa kerrallaan yhdestä kolmeen henkilöä. Tila toimii myös asiakaspalvelutilana ja henkilökunnan läpikulkupaikkana. Tilassa on talon pääasiallinen tulostus- ja kopiokone, jonka käyttö lisää melua ja toimistossa asiointia. Kaksi puhelinta soivat usein. Käyttäjä B pitääkin tilaa ”melko hektisenä”, ja työ keskeytyy vähän väliä. Keskeytyksistä on jonkin verran haittaa työnteossa, mutta toisaalta työn monipuolisuus on mukavaa. Käyttäjä A:n mukaan toimisto on toisinaan paikka, jossa ei ole minkäänlaista työrauhaa. Se ei ole minkään keskittymistä vaativan työn ympäristö vilkkaimpina hetkinä, koska työ keskeytyy lukemattomia kertoja päivässä.

Talon tietokoneet ovat pääasiassa pari vuotta vanhoja, mutta osa on vanhempia. Koneissa on Windows XP -käyttöjärjestelmä. Ne ovat keskenään vertaisverkossa, mutta palvelin on juuri hankittu, joten lähiverkko tultaneen muuttamaan asiakas-palvelin-malliseksi.

5.1.4 Käyttäjien tarpeet ja vaatimukset

Käyttäjien vaatimuksista keskeisin on tilavarausten tekeminen ja muokkaaminen sekä varaustietojen tulostaminen. Käyttäjät olivat pääosin tyytyväisiä nykyiseen järjestelmään, mutta käyttäjä A totesi sen käytön vievän melko paljon aikaa lukukausien alussa. Samoin he totesivat käytön olevan jonkin verran virhealtista. Automatisoidumpi järjestelmä nopeuttaisi varausten syöttöä ja varaustietojen hakemista, kunhan järjestelmä olisi riittävän helppo- ja nopeakäyttöinen. Käyttäjä A:lla oli negatiivisia kokemuksia kaupungin koulujen tilavarausjärjestelmän käytöstä, jota hän piti hitaana ja hankalana. Joka tapauksessa käyttöä nopeuttaisi, jos kurssien varaukset voisi tehdä kerralla koko lukukaudeksi eikä tietoja tarvitsisi kopioida viikosta toiseen.

Käyttäjä A toivoi, että tulevassa järjestelmässä olisi syöttörajoituksia, mikä karsisi virhemahdollisuuksia. Syöttörajoituksilla esimerkiksi kellonaikaa ei voisi syöttää väärässä muodossa. Samoin hän toivoi, että järjestelmä lisäisi kurssien perään automaattisesti 15 minuutin siirtymäajan, jotta ehdittäisiin tekemään tarvittavat järjestelyt ennen uuden kurssin alkamista. Yhden kurssin oppituntien välisen tauon A laskee ”automaattisesti omassa päässään”, joten sitä järjestelmän ei tarvitsisi huomioida.

Toisena uutena asiana käyttäjä A toivoi tilojen vuokrien tallentamista varausten yhteyteen. Kaikki tiloja käyttävät yksiköt, kuten kansalaisopisto, nuorisotoiminta ja projektit, maksavat vuokraa Jyväskylää ylläpitävälle setlementille. Nykyään tilojen vuokrat ja käytön tuntimäärät tallennetaan omalle lomakkeelle, josta vuokrat lasketaan kerran kuussa. Jos tilavarausjärjestelmässä olisi mahdollisuus tallentaa ja tulostaa myös tilojen vuokrat, säästytäisiin vuokrienlaskennan isosta, erillisestä työvaiheesta.

Käyttäjä A toivoi myös, että kurssien ja muiden tapahtumien tarvitsemat varusteet näkyisivät tulostetussa tilavarauslomakkeessa. Tilojen järjestelyt tehdään niiden pohjalta, ja olisi kätevää, jos varusteet olisivat mukana muistilistana. Tosin lomakkeesta saattaisi tulla silloin sekava ja täyteen ahdettu. Asian ratkaisua jäätin pohtimaan.

Käyttäjät suhtautuvat melko myönteisesti ehdotukseen internetin kautta tapahtuvasta järjestelmän käytöstä. Tosin tarvetta ei ole toistaiseksi tullut selainpohjaiselle käytölle, mutta ei ajatusta pidetä huononakaan. Joskus käyttäjä A:lta on kysytty kotona ollessa sähköpostitse tietoja tilavarauksista, mutta muuten tarvetta on ollut hyvin vähän. Muut

toimipisteet saavat koko lukukauden tilavaraukset sähköpostissa viikoittain päivitetävänä versiona, joten niissäkään ei ole tarvetta nettikäyttöön. Tosin ei järjestelmän nettiselailusta olisi haittaakaan muille työntekijöille. Käyttäjä B mainitsee, ettei kaikille tulisi kuitenkaan antaa oikeuksia varausten tekoon, jotta varaukset eivät menisi sekaisin. Hyvänä puolena käyttäjä A pitää sitä, että internetpalvelimelle tallennettuna järjestelmä säilyisi varmemmassa tallessa.

Käyttäjät ovat kokeilleet myös Microsoftin Outlook-ohjelman kalenteritoimintoja tilavarausten tekoon. Ohjelman käytössä he pitivät kömpelönä sitä, että tiloja ei saa näkyviin samalle sivulle vaan joutuu navigoimaan sivujen välillä. Käyttäjät suhtautuivat pääosin myönteisesti Outlookin käyttöön, mutta käyttökokemusta oli vasta vähän. Käyttäjä B oli käyttänyt aiemmin samantapaista ajanvarausjärjestelmää, jonka hyvänä puolena hän piti erityisesti ohjelman ulkonäköä eli sen miellyttävyyttä silmälle.

Käyttäjien tarpeet ja vaatimukset ovat koottuna seuraavat:

- tilavarausten tekeminen, muokkaaminen ja poistaminen
- tilavarausten tulostaminen
- varausten syötön nopeuttaminen automatisoiduilla toiminnoilla
- helppokäyttöisyys
- syöttörajoitusten käyttö virheiden välttämiseksi
- 15 minuutin siirtymäajan lisääminen kurssien välille automaattisesti
- vuokrien tallentaminen, laskeminen ja tulostaminen
- varusteiden näkyminen tilavarausten tulosteissa
- internetin kautta tapahtuva käyttäminen tietyin varauksin.

5.1.5 Käytettävyystavoitteet

Tilavarausjärjestelmän keskeinen tavoite on tuottavuus, millä tarkoitetaan sitä, että käyttäjä pystyy suorittamaan aikomansa tehtävän eli pystyy lisäämään, muokkaamaan ja poistamaan tilavarauksia. Toiseksi järjestelmän tulee olla helppokäyttöinen, koska toimistotilan levottomuus ja puhelimien soiminen aiheuttaa työhön jatkuvia keskeytyksiä. Tehtävät vaihtuvat nopeasti, kun käyttäjät joutuvat välillä asiakaspalveluun tai

auttamaan kurssitilojen järjestelyssä. Käyttäjän on pystyttävä helposti jatkamaan aloittamaansa tehtävää siitä, mihin jäi ennen keskeytystä. Järjestelmän tilan tulee olla selkeästi havaittavissa, ettei käyttäjä joudu pohtimaan, mitä olikaan juuri tekemässä.

Järjestelmän tulee olla myös helposti opittavissa, sillä työllistämisvaroin palkattu toimistohenkilökunta vaihtuu melko nopeasti, noin vuoden välein. Samoin käyttäjien osaamistaso saattaa vaihdella. Nykyisten käyttäjien taitotaso on hyvä, mutta on mahdollista, että tulevien käyttäjien joukossa on myös huonompia osajia, joilla on vähäinen kokemus tietokoneen käytöstä.

Järjestelmän tulee estää virheiden syntyä esimerkiksi syöttörajoitusten avulla. Kirjoitusvirheitä voitaisiin vähentää myös valikkolistojen käytöllä, jolloin käyttäjän ei tarvitsisi itse kirjoittaa tietoja vaan ainoastaan valita oikea tieto luettelosta. Tapahtuneista virheistä tulee antaa selkeä virheilmoitus ja ohje virhetilanteesta palautumiseen.

Järjestelmän käytön on oltava miellyttävää. Tähän vaikuttaa dialogien selkeys ja yhdenmukaisuus, navigoinnin helppous, hyvä graafinen suunnittelu, esteettisyys, käyttäjän muistikuorman pienentäminen ja oikopolkujen käytön mahdollisuus. Käyttäjän on tunnettava hallitsevansa järjestelmän käytön ja hänen tulee saada järjestelmältä palautte tekemistään toimenpiteistä.

Järjestelmän on myös sovellettava erilaisille käyttäjille, niin vasta-alkajille kuin edistyneille käyttäjille. Käyttäjien mahdollinen värisokeus on otettava huomioon värienkäyttöä suunniteltaessa. Ikääntyneet tai huononäköiset käyttäjät on otettava huomioon oikeanlaisten ja riittävän suurten kirjasimien käytöllä. Päätteettömiä fontteja pidetään näytöllä helpompilukuisina kuin päätteellisiä (Sinkkonen ym. 2006, 124).

Keskeiset käytettävyystavoitteet ovat koottuna seuraavat:

- tuottavuus
- helppokäyttöisyys
- helppo opittavuus
- virhetilanteiden välttäminen
- miellyttävyys
- soveltuvuus erilaisille käyttäjille.

5.2 Tietojärjestelmän kehittäminen

Tietojärjestelmän kehittäminen rajattiin koskemaan vaatimusmäärittelyä ja järjestelmäanalyysia. Lähtökohtana oli oliokeskeinen ohjelmistotuotanto ja OMT++ -menetelmä. Vaatimusmäärittely tehtiin käyttäjävaatimusten perusteella. Järjestelmän tehtäviksi määriteltiin

- tilavarausten tekeminen tietokantaan
- tilavarausten muokkaaminen ja poistaminen
- halutun tilan ja ajankohdan varaustilanteen hakeminen tietokannasta
- varaustilanteiden tulostaminen.

Vaatimusmäärittely on esitetty liitteessä 2. Aluksi luetteloitiin järjestelmän toiminnalliset ja ei-toiminnalliset vaatimukset. Ei-toiminnallisissa vaatimuksissa kiinnitettiin huomiota järjestelmän käytettävyyteen. Vaatimukset kuvattiin käyttötapauksina eli use caseina, jotka yhdistettiin use case -kaavioksi. Use case -kaavio tehtiin UML-mallinnuksen mukaisesti. Järjestelmän tekniseen toteutukseen ei vielä otettu kantaa.

Järjestelmäanalyysissa määriteltiin järjestelmän staattinen rakenne. Aluksi tehtiin järjestelmän tietomalli, jossa kuvattiin järjestelmän sisältämät tiedot, niiden staattinen rakenne ja niiden väliset suhteet. Tietomalli kuvattiin ER-kaavion avulla, joka on liitteessä 3. Tietomallin kanssa rinnakkain tehtiin olioanalyysi, jossa etsittiin olioluokat, niiden attribuutit ja luokkien väliset suhteet. Määrittelyvaiheen luokkakaaviossa ei ole vielä luokkien operaatioita. Luokkakaavio on UML:n mukainen ja se on esitetty liitteessä 4. Seuraavaksi määriteltiin tietokannan rakenne, joka esitettiin taulukuvauksien ja tietokantakaavion avulla. Taulukuvaukset on esitetty liitteessä 5 ja tietokantakaavio liitteessä 6. Järjestelmän dynaaminen käyttäytyminen olisi voitu mallintaa sekvenssi-kaavioiden avulla, mutta tämä jäi ajanpuutteen vuoksi tekemättä.

OMT++ -menetelmän määrittelyvaiheessa tehdään myös käyttöliittymän määrittely. Tutkimuksessa toteutettiin operaatioanalyysi eli käyttäjälle tarjottavien operaatioiden kerääminen ja jakaminen pienemmiksi tehtäviksi. Tuloksena saatiin lista käyttöliitty-

mässä esiintyvistä tehtävistä, jotka jaettiin tutkimuksessa valmiiksi eri dialogeihin. Dialogihierarkia kuvattiin dialogikaaviolla, joka on muunnos tilakaaviosta. Operaatioanalyysin tuloksena syntynyt tehtävälista esitetään liitteessä 7 ja dialogikaavio on liitteessä 8. Tulostustoimintoja ei otettu mukaan tehtävälistaan eikä dialogeihin, koska niiden toteuttaminen oli vielä avointa. Käyttöliittymän komponenttimäärittelyä ja dialogien visualisointia ei toteutettu tässä tutkimuksessa. Näin myös prototyyppien tekeminen jäi tutkimuksen ulkopuolelle.

5.3 Tutkimuskysymyksiin vastaaminen

1. Millainen on tilaajan käyttöön hyvin sopiva tilavarausjärjestelmä?

Kysymykseen saatiin vastaus käyttäjäkeskeisen suunnittelun avulla. Käyttäjiä haastatteleamalla ja havainnoimalla sekä keskustelujen avulla saatiin selville käyttäjien vaatimukset, joiden pohjalta tehtiin järjestelmän vaatimusmäärittely.

2. Miten käytettävyys otetaan huomioon järjestelmän suunnittelussa?

Käytettävyys otettiin huomioon käytettävyysuunnittelun avulla. Käyttäjien haastattelujen ja havainnoinnin perusteella saatiin selville käyttäjien tarpeet, toiveet ja odotukset. Käyttäjien tehtäväkuvauksella selvitettiin nykyinen toiminta ja sen ongelmat. Käyttöympäristön kuvauksella kartoitettiin työtilasta johtuvat ongelmat ja tarpeet. Kuvausten perusteella määriteltiin käytettävyystavoitteet, joihin käyttöliittymän suunnittelussa tulisi pyrkiä.

3. Millainen on järjestelmän tekninen toteutus?

Tutkimuksessa tehtiin järjestelmän vaatimusmäärittelyn lisäksi järjestelmäanalyysi. Tietomallissa otettiin huomioon käyttäjien järjestelmään kohdistuvia toiveita, kuten tilavuokrien, varusteiden ja kurssien ylimenoajan mukaanotto järjestelmään. Näin pyrittiin kehittämään tilavarausjärjestelmä, joka vastaisi hyvin Jyväskylän tarpeisiin.

6 JOHTOPÄÄTÖKSET JA POHDINTA

Käytettävyyystutkimus on hyvin laaja-alainen tieteenala. Sen piiriin kuuluu ihmisen ja tietokoneen välisen vuorovaikutuksen tutkimus, tietojenkäsittelytiede, kognitiivinen psykologia, graafinen suunnittelu, ergonomia, sosiologia ja etnografia. Käytettävyyss-tutkimus on siis varsin poikkitieteellistä, ja sitä voidaan lähestyä monesta näkökulmasta. Käytettävyysskirjallisuus on laajaa ja monipuolista, mutta käsitteiden käyttö ja määrittely vaihtelevat teoksesta toiseen. Käsitteitä käytettävyyys, käyttäjäkeskeinen suunnittelu ja käytettävyyssuunnittelu käytetään osittain päällekkäisinä ja toistensa synonyymeina. Tutkimusala on vasta noin 40 vuoden ikäinen, mikä selittää osaltaan käsitteiden käytön kirjavuutta. Terminologian vaihtelu aiheutti jonkin verran vaikeuksia tutkimuksen taustateorian hahmotuksessa.

Käytettävyyystutkimuksen lähestymistapa voi olla joko ”pehmeä” ja laadullinen tai ”kova” ja määrällinen. Laadullisessa tutkimuksessa tutkimuskohdetta tarkastellaan kokonaisvaltaisesti todellisissa tilanteissa. Ihmisten havainnointi ja haastattelu ovat keskeisiä aineistonkeruuvälineitä, ja tutkija osallistuu kiinteästi tutkimusprosessiin. Määrällisessä tutkimuksessa taas pyritään löytämään tutkimuskohteista ominaisuuksia, joita voidaan mitata objektiivisesti. Tutkija on ulkopuolinen tarkkailija.

Tässä työssä käytettävyyystutkimus toteutettiin käyttäen apuna laadullista tutkimusmenetelmää. Menetelmä sopi hyvin tutkimukseen, koska käyttäjiä oli vain kaksi eikä tarkoituksena ollut saada aikaan objektiivisesti mitattuja tuloksia. Mitään käytettävyystestejä, joissa mitataan suoritusajoja tai lasketaan tehtyjen virheiden määrää, ei ollut aikomus toteuttaa. Aineistoa kerättiin haastattelun ja havainnoinnin avulla ja tuloksia tarkasteltiin subjektiivisella arvioinnilla. Käytettävyyssnäkökulma saatiin hyvin mukaan, kun nykyisessä järjestelmässä oli sopivasti puutteita, joiden poistaminen voitiin ottaa tavoitteeksi kehitettävässä järjestelmässä. Samoin hyvä perehtyminen käyttäjiin ja käyttöympäristöön toi tietoa käytön ongelmakohdista, jotka järjestelmän kehittämisessä tuli ottaa huomioon.

Järjestelmän kehittämisprojekti oli yllättävän työläs. Kuten Pohjonen (2002, 17) esittää teoksessaan, valitettava totuus on, että suuri osa projekteista ei koskaan valmistu tai ylittää niille varatut ajalliset, taloudelliset tai henkilöresurssit. Tässäkin tutkimuk-

sessä tavoitteita täytyi karsia, ja tutkimus rajautui vaatimusmäärittelyyn ja järjestelmänanalyysiin. Samoin prototyyppien teko jäi pois, vaikka niiden avulla olisi saatu käyttäjiltä arvokasta palautetta jatkokehitystä ajatellen. Tutkimuksella saatiin kuitenkin luotua hyvä perusta järjestelmän suunnittelu- ja toteutusvaihetta varten. Käyttäjien vaatimukset ja tarpeet tulivat hyvin esiin.

Käytettävyys on noussut tänä keväänä julkisuudessa kuumaksi kysymykseksi esimerkiksi pankin tietojärjestelmän muutoksen takia. Siinä käytettävyystavoitteiden toteutuminen epäonnistui kaikilta osin. Myös tekstinkäsittelyohjelma saattaa häiritä ohjelman käyttäjää korjaus- ja täydennysehdotuksillaan, kuten Holvas kuvaa pakinassaan (28.4.2008). Oletusasetuksilla voisi ottaa pois nuo häiritsevät toiminnot, mutta se vaatii käyttäjältä aikaa, vaivaa ja erityistaitoja. Holvas valittaakin, että ”vähitellen on syntynyt kokonainen ideologia, jonka mukaan käyttäjien tulee opetella kuuliaisesti väliin neen tarpeet”. Siis juuri päinvastoin, kuin mihin käytettävyystutkimuksessa pyritään!

Tietojärjestelmien kehittäjien ja ohjelmistotuotannon ammattilaisten tulee pyrkiä jatkossa parempiin tuloksiin järjestelmän toiminnan ja käytettävyyden kannalta. Käyttäjakeskeinen suunnittelu on tulevaisuudessa yhä tärkeämpää, koska tietojärjestelmiä käytetään yhä enemmän ja useammassa kohteissa. Ohjelmasovellukset ovat vallanneet kodinkoneet, autot ja viihde-elektroniikan. Käyttö laajenee myös kaikkiin väestöryhmiin, kuten ikääntyneisiin, maahanmuuttajiin tai vammaisiin, joiden erityistarpeet on otettava huomioon. Laadukas ohjelmistotuotanto ja käytettävyyden huomioiminen on tärkeää, sillä järjestelmien haavoittuvuus keskeisillä yhteiskunnan alueilla, kuten tietoliikenneyhteyksien ylläpidossa, sähköntuotannossa, teollisuuden prosesseissa, liikenteessä ja sairaaloissa, voi muodostaa vakavan vaaran yhteiskunnan toiminnalle.

LÄHTEET

- Erkiö, H., Hölttä, T., Koivunen, M.-R., Nieminen, M. & Riihiahho S. 1996. Käytettävyyden arviointitavat. Teoksessa Graafisen käyttöliittymän suunnittelu. Toim. A. Kalimo. Espoo: Suomen ATK-kustannus, 83–140.
- Faulkner, X. 2000. Usability Engineering. Basingstoke: Palgrave.
- Gould, J. D. & Lewis, C. 1985. Designing for Usability: Key Principles and What Designers Think. Communications of the ACM 28, 3, 300–311. Viitattu 27.4.2008. [Http://www.research.ibm.com/compsci/spotlight/hci/p300-gould.pdf](http://www.research.ibm.com/compsci/spotlight/hci/p300-gould.pdf).
- Haikala, I. & Märijärvi, J. 2004. Ohjelmistotuotanto. 10., uud. p. Helsinki: Talentum.
- Hirsjärvi, S., Remes, P. & Sajavaara, P. 2003. Tutki ja kirjoita. 6.–9. p. Helsinki: Kustannusosakeyhtiö Tammi.
- Holvas, J. 2008. Välineen ehdoilla. Helsingin Sanomat 28.4.2008, D1.
- Huovila, H. 2008. Toiminnanjohtaja, Jyväskylän settlementti ry. Haastattelu 10.4.2008.
- Jaakkola, H. 2002. Oliomenetelmistä (OMT++). Viitattu 4.5.2008. [Http://www.pori.tut.fi/~hj/for-guests/public/otumen/luku19.pdf](http://www.pori.tut.fi/~hj/for-guests/public/otumen/luku19.pdf).
- Jaaksi, A. & Pettersson, M. 1996. Käyttöliittymän suunnitteluprosessi. Teoksessa Graafisen käyttöliittymän suunnittelu. Toim. A. Kalimo. Espoo: Suomen ATK-kustannus, 46–82.
- Kalimo, A. 1996. Johdanto: Käytettävyyden merkitys. Teoksessa Graafisen käyttöliittymän suunnittelu. Toim. A. Kalimo. Espoo: Suomen ATK-kustannus, 7–11.
- Koivunen, M.-R. & Nieminen, M. 1996. Ohjelmiston käytettävyys. Teoksessa Graafisen käyttöliittymän suunnittelu. Toim. A. Kalimo. Espoo: Suomen ATK-kustannus, 12–25.
- Nielsen, J. 1993. Usability Engineering. San Diego, CA: Academic Press.
- Norman, D. A. 1988. The Psychology of Everyday Things. New York: Basic Books.
- Pettersson, M. & Riihiahho, S. 1996. Käyttöliittymän suunnitteluperiaatteet. Teoksessa Graafisen käyttöliittymän suunnittelu. Toim. A. Kalimo. Espoo: Suomen ATK-kustannus, 35–45.
- Pohjonen, R. 2002. Tietojärjestelmien kehittäminen. 2. p. Jyväskylä: Docendo.
- Preece, J. 1994. Human–Computer Interaction. Harlow: Addison-Wesley.
- SFS-standardien luettelo 2008. Viitattu 29.4.2008. Suomen Standardisoimisliitto. [Http://www.sfs.fi/luettelo/sfs.php](http://www.sfs.fi/luettelo/sfs.php).

Shneiderman, B. 1998. Designing the User Interface. 3rd ed. Reading, MA: Addison Wesley.

Sinkkonen, I., Kuoppala, H., Parkkinen, J. & Vastamäki, R. 2006. Käytettävyyden psykologia. 3., uud. p. Helsinki: Edita Publishing.

LIITTEET

Liite 1. Haastattelukysymykset

- 1 Mikä on ammattinimikkeesi Jyväskylässä? Mitä erilaisia asioita työtehtäviisi kuuluu?
- 2 Millainen kokemus sinulla on tietokoneen käytöstä? Miten kauan olet käyttänyt tietokonetta? Millaiseksi arvioisit itsesi tietokoneen käyttäjänä? Onko sinulla ollut ongelmia tietokoneen käytössä? Pidätkö työskentelystä tietokoneella?
- 3 Mihin tilavarausjärjestelmää käytetään Jyväskylässä? Kerro erilaisista käyttökohteista ja käyttötarpeista.
- 4 Ketkä kaikki tilavarausjärjestelmää käyttävät?
- 5 Miten usein käytät tilavarausjärjestelmää? Kuinka paljon sen käyttöön kuluu aikaa päivässä/viikossa?
- 6 Missä tilassa tilavarausjärjestelmän käyttö tapahtuu? Millainen käyttöympäristö on? Onko tila rauhallinen, onko häiriötekijöitä, onko työssä keskeytyksiä, millainen tietokone käytössäsi on?
- 7 Millainen nykyinen tilavarausjärjestelmä on? Miten sitä käytetään? Mitä eri toimintoja siinä on? Kuvaile tarkkaan, miten teet tilavarauksia.
- 8 Onko nykyinen järjestelmä helppokäyttöinen? Oletko siihen tyytyväinen? Onko käytössä ongelmia tai puutteita? Onko sitä nopea käyttää? Syntyykö käytössä virheitä? Onko käyttö miellyttävää?
- 9 Mitä parannuksia toivoisit nykyiseen järjestelmään?
- 10 Millainen tilavarausjärjestelmä olisi mielestäsi hyvä? Mitä toivomuksia asettaisit uudelle järjestelmälle? Mitä tehtäviä järjestelmän pitäisi osata suorittaa? Mitkä ovat omat käyttötarpeesi järjestelmän suhteen?
- 11 Käyttäisitkö tilavarausjärjestelmää mielelläsi myös muualla kuin vakituisessa työpisteessäsi? Olisiko tarvetta käyttää sitä myös toisilta koneilta tai kotoa ja matkoilta käsin? Olisiko internetissä toimiva järjestelmä mielestäsi hyvä?
- 12 Oletko käyttänyt muita tilavarausjärjestelmiä? Jos olet, kerro kokemuksistasi.
- 13 Kiitos haastattelusta!

Liite 2. Vaatimusmäärittely

YLEISKUVAUS

Toimeksianto

Tehtävänä on laatia tilavarausjärjestelmä, jonka avulla voidaan tehdä tilavarauksia kansalaisopiston kursseja ja muita tapahtumia varten. Tilaa-jana on Jyväskylän setlementti ry, jonka kansalaisopisto- ja muuta toimintaa järjestetään päärakennuksen lisäksi muutamassa muussa toimipisteessä.

Jyväskylässä on tällä hetkellä käytössä ainoastaan taulukkolaskentaohjelman avulla ylläpidettävä tilavarausjärjestelmä. Tiedot syötetään lomakkeille viikoittain käsin ilman automaattisia toimintoja.

Käyttäjät

Tilavarausjärjestelmän kohdekäyttäjiä ovat Jyväskylän kurssien ja muiden tapahtumien tilavarauksista vastuussa olevat henkilöt. Käyttäjiä tulee olemaan kahdesta kolmeen toimistohenkilöä.

Tehtävät

Tilavarausjärjestelmän tehtäviä ovat:

- tilavarausten tekeminen tietokantaan
- halutun tilan ja ajankohdan varaustilanteen hakeminen tietokannasta
- halutun tilan ja ajankohdan varaustilanteen tulostaminen
- varaustietojen muuttaminen ja poistaminen.

Ympäristö

Tilavarausjärjestelmä asennetaan joko yhdelle tietokoneelle paikallisesti käytettäväksi tai palvelimelle internetin tai intranetin kautta käytettäväksi. Talossa on käytössä melko uusi laitekanta, ja käyttäjien koneissa on Windows XP -käyttöjärjestelmä. Jyväskylässä on käytössä lähiverkko ja vähän aikaa sitten hankittu oma palvelin.

VAATIMUKSET

Toiminnalliset vaatimukset

Nro	Kuvaus	Luokka
1	Käyttäjä voi lisätä tilavarouksia	Pakollinen
2	Käyttäjä voi poistaa tilavarouksia	Pakollinen
3	Käyttäjä voi muuttaa tilavarouksia	Pakollinen
4	Käyttäjä voi hakea tietokannasta haluamansa tilan ja ajankohdan varaustilanteen	Pakollinen
5	Käyttäjä voi tulostaa raportteja luokkien ja muiden tilojen varaustilanteesta päivittäin, viikoittain ja luku-kausittain	Pakollinen

Ei-toiminnalliset vaatimukset

Nro	Kuvaus	Luokka
1	Järjestelmän tulee selvittää hyvin sekä ohjelman että käyttäjän aiheuttamista virhetilanteista	Pakollinen
2	Järjestelmän tulee olla riittävän helppokäyttöinen, niin että aloittelija oppii sen käytön 15 minuutissa	Pakollinen
3	Tietokannan varmuuskopioinnin on onnistuttava polttavalla CD-aseamalla	Pakollinen

Rajoitteet

1	Järjestelmään kirjataan vain Jyväskylän omassa hallinnassa olevien tilojen varauksia, ei kouluista vuokrattavia tiloja	
---	--	--

KÄYTTÖTAPAUKSET

Use case -kaavio



Use case 1: Uuden tilavarauksen tekeminen

Aktorit:	Toimistohenkilö
Frekvenssi:	Tarvittaessa (useita kertoja päivässä), kun käyttäjä haluaa tehdä uuden tilavarauksen.
Esiehdot:	Järjestelmä on toiminnassa ja tietokanta on avoinna.
Kuvaus:	Käyttäjä avaa varauksentekolomakkeen ja valitsee haluamansa tilan ja ajankohdan. Järjestelmän hakee kyseisen tilan ja ajankohdan varaustilanteen. Käyttäjä valitsee vapaan ajan ja syöttää varaustiedot lomakkeelle. Käyttäjä tallentaa tiedot tietokantaan tallennuspainikkeella. Järjestelmä antaa vahvistuksen varauksen kirjaamisesta.
Lopputulokset:	Tilavaraus on kirjattu tietokantaan.
Poikkeukset:	<ol style="list-style-type: none">1. Käyttäjä on syöttänyt väärentyyppisen tiedon kenttään, eikä varauksen teko onnistu. Järjestelmä huomauttaa käyttäjälle asiasta.2. Tietokantaa ei saada avattua, kun varausta yritetään tallentaa. Järjestelmä tiedottaa käyttäjälle asiasta.

Use case 2: Tilavarauksen tietojen muuttaminen

Aktorit:	Toimistohenkilö
Frekvenssi:	Tarvittaessa (useita kertoja päivässä), kun käyttäjä haluaa tehdä muutoksia tilavaraukseen.
Esiehdot:	Järjestelmä on toiminnassa, tietokanta on avoinna ja päivittävien tietojen kohde on olemassa.
Kuvaus:	Käyttäjä hakee halutun tilavarauksen ja avaa sen tiedot muokkaustilaan. Käyttäjä tekee lomakkeelle tarvittavat muutokset ja tallentaa tiedot. Järjestelmä antaa vahvistuksen varauksen päivittämisestä.
Lopputulokset:	Tilavarauksen uudet tiedot on kirjattu tietokantaan.
Poikkeukset:	<ol style="list-style-type: none">1. Käyttäjä on syöttänyt väärentyyppisen tiedon kenttään, eikä varauksen teko onnistu. Järjestelmä huomauttaa käyttäjälle asiasta.2. Tietokantaa ei saada avattua, kun varauksen tietoja yritetään tallentaa. Järjestelmä tiedottaa käyttäjälle asiasta.

Use case 3: Tilavarauksen poistaminen

Aktorit:	Toimistohenkilö
Frekvenssi:	Tarvittaessa (useita kertoja päivässä), kun käyttäjä haluaa poistaa tilavarauksen.
Esiehdot:	Järjestelmä on toiminnassa, tietokanta on avoinna ja poistettavien tietojen kohde on olemassa.
Kuvaus:	Käyttäjä hakee halutun kohteen ja valitsee tilavarauksen poistamisen. Käyttäjä voi myös poistaa kerralla useita tietoja. Ennen poistamista järjestelmä kysyy varmistuksen. Käyttäjä vahvistaa poiston tai peruu sen. Järjestelmä antaa vahvistuksen varauksen/varausten poistamisesta.
Lopputulokset:	Tilavaraus/-varaukset on poistettu tietokannasta.
Poikkeukset:	1. Tietokantaa ei saada avattua, kun varauksen tietoja yritetään poistaa. Järjestelmä tiedottaa käyttäjälle asiasta.

Use case 4: Tilavarauksen tietojen hakeminen

Aktorit:	Toimistohenkilö
Frekvenssi:	Tarvittaessa (useita kertoja päivässä), kun käyttäjä haluaa hakea tilavarausten tietoja.
Esiehdot:	Järjestelmä on toiminnassa ja tietokanta on avoinna.
Kuvaus:	Käyttäjä avaa hakulomakkeen ja syöttää varauksen hakuehdot. Järjestelmä listaa kohteet, jotka täyttävät annetut hakuehdot. Käyttäjä valitsee listasta haluamansa yksittäisen kohteen jatkotoimenpiteitä varten.
Lopputulokset:	Järjestelmä tuottaa listauksen käyttäjän haluamista kohteista.
Poikkeukset:	1. Tietokantaa ei saada avattua, kun tietoja yritetään hakea. Järjestelmä tiedottaa käyttäjälle asiasta. 2. Annettujen hakuehtojen perusteella ei löydy yhtään kohdetta. Järjestelmä tiedottaa käyttäjälle asiasta.

Use case 5: Tilavarausraporttien tulostaminen

Aktorit:	Toimistohenkilö
Frekvenssi:	Tarvittaessa (useita kertoja päivässä), kun käyttäjä haluaa tulostaa tilavaraustilanteen.
Esiehdot:	Järjestelmä on toiminnassa ja tietokanta on avoinna.
Kuvaus:	Käyttäjä avaa tulostusikkunan ja valitsee haluamansa raporttityypin. Käyttäjä voi valita tilakohtaisen raportin joko yhden päivän, viikon tai koko lukukauden ajalta.
Lopputulokset:	Järjestelmä tulostaa raportin tilavarauksista näytölle tai kirjoittimelle.
Poikkeukset:	<ol style="list-style-type: none">1. Tietokantaa ei saada avattua, kun raporttia yritetään tulostaa. Järjestelmä tiedottaa käyttäjälle asiasta.2. Tulostaminen kirjoittimelle ei onnistu asennus- tai verkko-ongelmien vuoksi.

Use case 6: Ylläpitotietojen lisääminen

Aktorit:	Toimistohenkilö
Frekvenssi:	Tarvittaessa (lähinnä lukukausien alussa), kun ylläpitotietoja lisätään.
Esiehdot:	Järjestelmä on toiminnassa ja tietokanta on avoinna.
Kuvaus:	Käyttäjä avaa haluamansa lomakkeen (rakennusten, tilojen, hintaryhmien, yksiköiden, vastuuhenkilöiden, tapahtumien, tapahtumatyyppien tai varusteiden lisääminen) ja syöttää tarvittavat tiedot. Käyttäjä tallentaa tiedot tietokantaan tallennuspainikkeella.
Lopputulokset:	Tiedot on lisätty tietokantaan.
Poikkeukset:	<ol style="list-style-type: none">1. Tietokantaa ei saada avattua, kun tietoja yritetään tallentaa. Järjestelmä tiedottaa käyttäjälle asiasta.

Use case 7: Ylläpitotietojen muuttaminen

Aktorit:	Toimistohenkilö
Frekvenssi:	Tarvittaessa (lähinnä lukukausien alussa), kun ylläpitotietoja muutetaan.
Esiehdot:	Järjestelmä on toiminnassa ja tietokanta on avoinna.
Kuvaus:	Käyttäjä avaa haluamansa lomakkeen (rakennukset, tilat, hintaryhmät, yksiköt, vastuuhenkilöt, tapahtumat, tapahtumatyypit tai varusteet), valitsee halutun kohteen ja syöttää uudet tiedot. Käyttäjä tallentaa tiedot tietokantaan tallennuspainikkeella.
Lopputulokset:	Tiedot on lisätty tietokantaan.
Poikkeukset:	1. Tietokantaa ei saada avattua, kun tietoja yritetään tallentaa. Järjestelmä tiedottaa käyttäjälle asiasta.

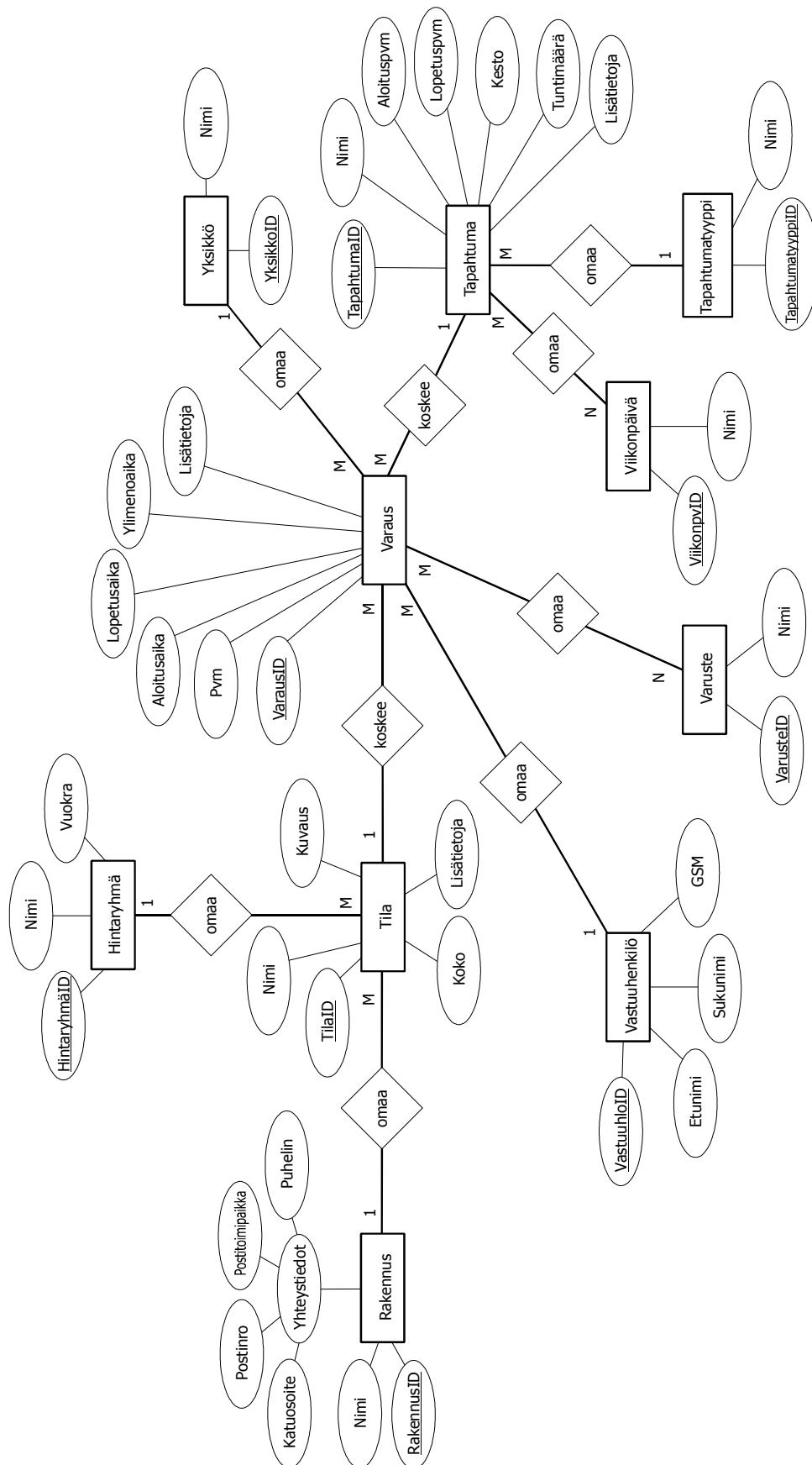
Use case 8: Ylläpitotietojen poistaminen

Aktorit:	Toimistohenkilö
Frekvenssi:	Tarvittaessa (lähinnä lukukausien alussa), kun ylläpitotietoja poistetaan.
Esiehdot:	Järjestelmä on toiminnassa ja tietokanta on avoinna.
Kuvaus:	Käyttäjä avaa haluamansa lomakkeen (rakennusten, tilojen, hintaryhmien, yksiköiden, vastuuhenkilöiden, tapahtumien, tapahtumatyyppien tai varusteiden poistaminen), valitsee halutun kohteen ja poistaa sen. Käyttäjä voi myös poistaa kerralla kaikki tietueet. Ennen poistamista järjestelmä kysyy varmistuksen. Käyttäjä vahvistaa poiston tai peruuttaa sen.
Lopputulokset:	Tiedot on poistettu tietokannasta.
Poikkeukset:	1. Tietokantaa ei saada avattua, kun tietoja yritetään poistaa. Järjestelmä tiedottaa käyttäjälle asiasta.

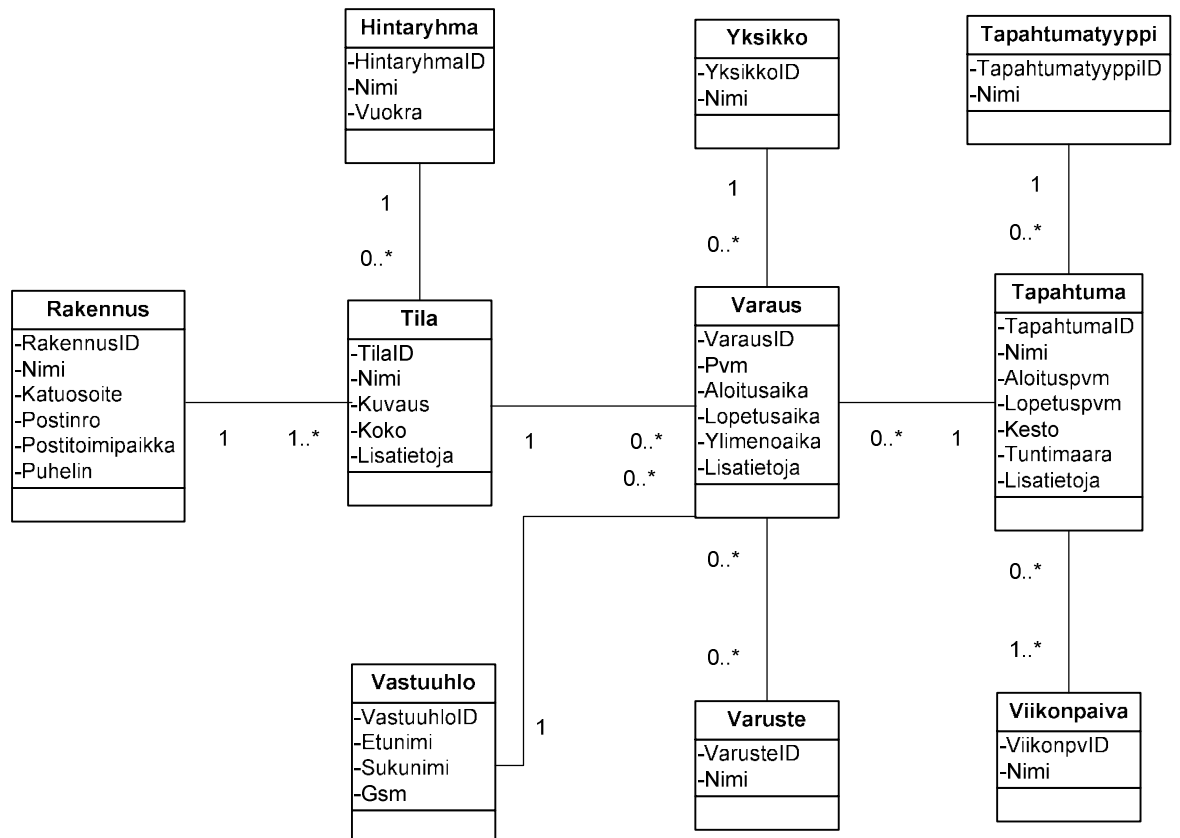
Use case 9: Ylläpitotietojen hakeminen

Aktorit:	Toimistohenkilö
Frekvenssi:	Tarvittaessa (lähinnä lukukausien alussa), kun käyttäjä haluaa hakea ylläpitotietoja.
Esiehdot:	Järjestelmä on toiminnassa ja tietokanta on avoinna.
Kuvaus:	Käyttäjä avaa hakulomakkeen ja syöttää tietojen hakuehdot. Järjestelmä listaa kohteet, jotka täyttävät annetut hakuehdot. Käyttäjä valitsee listasta haluamansa yksittäisen kohteen jatkotoimenpiteitä varten.
Lopputulokset:	Järjestelmä tuottaa listauksen käyttäjän haluamista kohteista.
Poikkeukset:	<ol style="list-style-type: none">1. Tietokantaa ei saada avattua, kun tietoja yritetään hakea. Järjestelmä tiedottaa käyttäjälle asiasta.2. Annettujen hakuehtojen perusteella ei löydy yhtään kohdetta. Järjestelmä tiedottaa käyttäjälle asiasta.

Liite 3. ER-kaavio



Liite 4. Luokkakaavio



Liite 5. Taulukuvaus

1 Rakennus

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
RakennusID	Rakennuksen tunniste	NOT NULL	INTEGER
Nimi	Rakennuksen nimi	NOT NULL	VARCHAR(50)
Katuosoite	Rakennuksen katuosoite		VARCHAR(50)
Postinro	Postinumero		VARCHAR(10)
Postitoimipaikka	Postitoimipaikka		VARCHAR(30)
Puhelin	Rakennuksen puhelinnumero		VARCHAR(20)

Indeksit

Indeksoidut kentät	Tyyppi
RakennusID	UNIQUE INDEX
Nimi	UNIQUE INDEX

Rajoitteet

Rajoite (Constraint)	Kenttä
PRIMARY KEY	RakennusID

2 Tila

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
TilaID	Tilan tunniste	NOT NULL	INTEGER
Nimi	Tilan nimi	NOT NULL	VARCHAR(50)
RakennusID	Rakennuksen tunniste	NOT NULL	INTEGER
HintaryhmaID	Hintaryhmän tunniste	NOT NULL	INTEGER
Kuvaus	Tilan kuvaus		VARCHAR(200)
Koko	Tilan pinta-ala tai henkilö- määrä		VARCHAR(50)
Lisätietoja	Muut tilaa koskevat tiedot		TEXT

Indeksit

Indeksoidut kentät	Tyyppi
TilaID	UNIQUE INDEX
Nimi	UNIQUE INDEX

Rajoitteet

Rajoite (Constraint)	Kenttä
PRIMARY KEY	TilaID
FOREIGN KEY	RakennusID REFERENCES Rakennus (RakennusID)
FOREIGN KEY	HintaryhmaID REFERENCES Hintaryhmä (HintaryhmaID)

3 Hintaryhma

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
HintaryhmaID	Hintaryhmän tunniste	NOT NULL	INTEGER
Nimi	Hintaryhmän nimi	NOT NULL	VARCHAR(30)
Vuokra	Vuokran suuruus euroina	NOT NULL	FLOAT

Indeksit

Indeksoidut kentät	Tyyppi
HintaryhmaID	UNIQUE INDEX
Nimi	UNIQUE INDEX

Rajoitteet

Rajoite (Constraint)	Kenttä
PRIMARY KEY	HintaryhmaID

4 Varaus

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
VarausID	Varauksen tunniste	NOT NULL	INTEGER
YksikkoID	Yksikön tunniste	NOT NULL	INTEGER
TapahtumaID	Tapahtuman tunniste	NOT NULL	INTEGER
TilaID	Tilan tunniste	NOT NULL	INTEGER
VastuuhloID	Vastuuhenkilön tunniste	NOT NULL	INTEGER
Pvm	Varauksen päivämäärä	NOT NULL	DATE
Aloitusaika	Varauksen aloitusaika	NOT NULL	TIME
Lopetusaika	Varauksen lopetusaika	NOT NULL	TIME
Ylimenoaika	Varauksen vaatima 15 min ylimenoaika	NOT NULL	TRUE/FALSE
Lisätietoja	Muut varausta koskevat tiedot		TEXT

Indeksit

Indeksoidut kentät	Tyyppi
VarausID	UNIQUE INDEX

Rajoitteet

Rajoite (Constraint)	Kenttä
PRIMARY KEY	VarausID
FOREIGN KEY	YksikkoID REFERENCES Yksikko (YksikkoID)
FOREIGN KEY	TapahtumaID REFERENCES Tapahtuma (TapahtumaID)
FOREIGN KEY	TilaID REFERENCES Tila (TilaID)
FOREIGN KEY	VastuuhloID REFERENCES Vastuuhenkilo (VastuuhloID)

5 Yksikko

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
YksikkoID	Yksikön tunniste	NOT NULL	INTEGER
Nimi	Yksikön nimi	NOT NULL	VARCHAR(50)

Indeksit

Indeksoidut kentät	Tyyppi
YksikkoID	UNIQUE INDEX
Nimi	UNIQUE INDEX

Rajoitteet

Rajoite (Constraint)	Kenttä
PRIMARY KEY	YksikkoID

6 Tapahtuma

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
TapahtumaID	Tapahtuman tunniste	NOT NULL	INTEGER
Nimi	Tapahtuman nimi	NOT NULL	VARCHAR(50)
TapahtumatyyppiID	Tapahtumatyypin tunniste	NOT NULL	INTEGER
Aloituspvm	Tapahtuman aloituspvm		DATE
Lopetuspvm	Tapahtuman lopetuspvm		DATE
Kesto	Tapahtuman kertojen lukumäärä		INTEGER
Tuntimaara	Tapahtuman kesto oppitunteina		FLOAT
Lisätietoja	Tapahtuman vaatimat tarvikkeet ja järjestelyt		TEXT

Indeksit

Indeksoidut kentät	Tyyppi
TapahtumaID	UNIQUE INDEX
Nimi	UNIQUE INDEX

Rajoitteet

Rajoite (Constraint)	Kenttä
PRIMARY KEY	TapahtumaID
FOREIGN KEY	TapahtumatyyppiID REFERENCES Tapahtumatyyppi (TapahtumatyyppiID)

7 Tapahtumatyyppi

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
TapahtumatyyppiID	Tapahtumatyyppin tunniste	NOT NULL	INTEGER
Nimi	Tapahtumatyyppin nimi	NOT NULL	VARCHAR(50)

Indeksit

Indeksoidut kentät	Tyyppi
TapahtumatyyppiID	UNIQUE INDEX
Nimi	UNIQUE INDEX

Rajoitteet

Rajoite (Constraint)	Kenttä
PRIMARY KEY	TapahtumatyyppiID

8 Vastuuhenkilö

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
VastuuhloID	Vastuuhenkilön tunniste	NOT NULL	INTEGER
Etunimi	Vastuuhenkilön etunimi	NOT NULL	VARCHAR(30)
Sukunimi	Vastuuhenkilön sukunimi	NOT NULL	VARCHAR(50)
Gsm	Vastuuhenkilön puhelinnumero		VARCHAR(20)

Indeksit

Indeksoidut kentät	Tyyppi
VastuuhloID	UNIQUE INDEX

Rajoitteet

Rajoite (Constraint)	Kenttä
PRIMARY KEY	VastuuhloID

9 Varuste

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
VarusteID	Varusteen tunniste	NOT NULL	INTEGER
Nimi	Varusteen nimi	NOT NULL	VARCHAR(50)

Indeksit

Indeksoidut kentät	Tyyppi
VarusteID	UNIQUE INDEX
Nimi	UNIQUE INDEX

Rajoitteet

Rajoite (Constraint)	Kenttä
PRIMARY KEY	VarusteID

10 Viikonpaiva

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
ViikonpvID	Viikonpäivän tunniste	NOT NULL	INTEGER
Nimi	Viikonpäivän nimi	NOT NULL	VARCHAR(15)

Indeksit

Indeksoidut kentät	Tyyppi
ViikonpvID	UNIQUE INDEX
Nimi	UNIQUE INDEX

Rajoitteet

Rajoite (Constraint)	Kenttä
PRIMARY KEY	ViikonpvID

11 Tapahtuman_viikonpv

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
TapahtumaID	Tapahtuman tunniste	NOT NULL	INTEGER
ViikonpvID	Viikonpäivän tunniste	NOT NULL	INTEGER

Indeksit

Indeksoidut kentät	Tyyppi
VarausID	
ViikonpvID	

Rajoitteet

Rajoite (Constraint)	Kenttä
FOREIGN KEY	VarausID REFERENCES Varaus (VarausID)
FOREIGN KEY	ViikonpvID REFERENCES Viikonpaiva (ViikonpvID)

12 Varauksen_varuste

Kenttä	Kuvaus	Pakollisuus	Tietotyyppi
VarausID	Varauksen tunniste	NOT NULL	INTEGER
VarusteID	Varusteen tunniste	NOT NULL	INTEGER

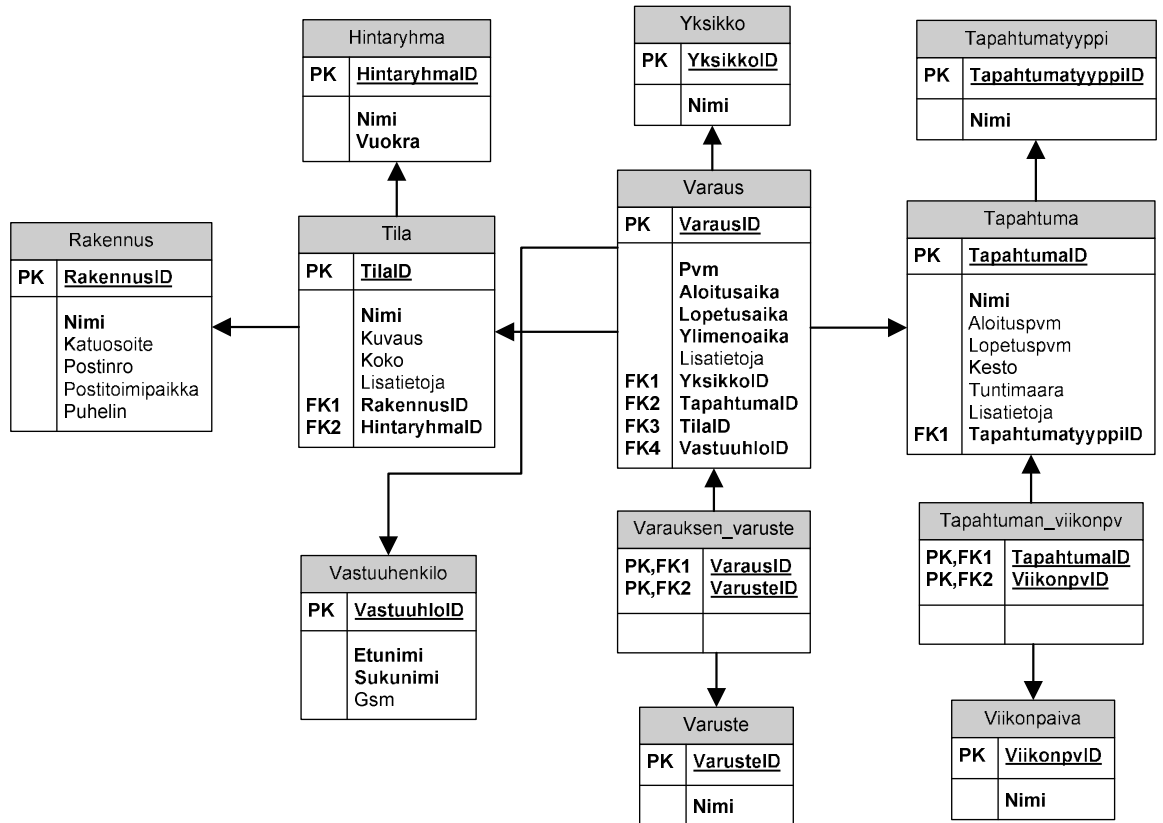
Indeksit

Indeksoidut kentät	Tyyppi
VarausID	
VarusteID	

Rajoitteet

Rajoite (Constraint)	Kenttä
FOREIGN KEY	VarausID REFERENCES Varaus (VarausID)
FOREIGN KEY	VarusteID REFERENCES Varuste (VarusteID)

Liite 6. Tietokantakaavio



Liite 7. Tehtävälista

Pääikkuna	
1	Valitse rakennus
2	Valitse tila
3	Valitse päivämäärä

Varaukset	
1	Hae varaukset
2	Valitse varaus

Varaus	
1	Lisää varaus
2	Muuta varauksen tietoja
3	Poista varaus

Rakennukset ja tilat	
1	Hae rakennukset
2	Valitse rakennus
3	Hae tilat
4	Valitse tila
5	Hae hintaryhmät
6	Valitse hintaryhmä

Rakennus	
1	Lisää rakennus
2	Muuta rakennuksen tietoja
3	Poista rakennus

Tila	
1	Lisää tila
2	Muuta tilan tietoja
3	Poista tila

Hintaryhmä	
1	Lisää hintaryhmä
2	Muuta hintaryhmän tietoja
3	Poista hintaryhmä

Yksiköt	
1	Hae yksiköt
2	Lisää yksikkö
3	Valitse yksikkö
4	Muuta yksikön tietoja
5	Poista yksikkö

Tapahtumatyypit ja tapahtumat	
1	Hae tapahtumatyypit
2	Valitse tapahtumatyyppe
3	Hae tapahtumat
4	Valitse tapahtuma

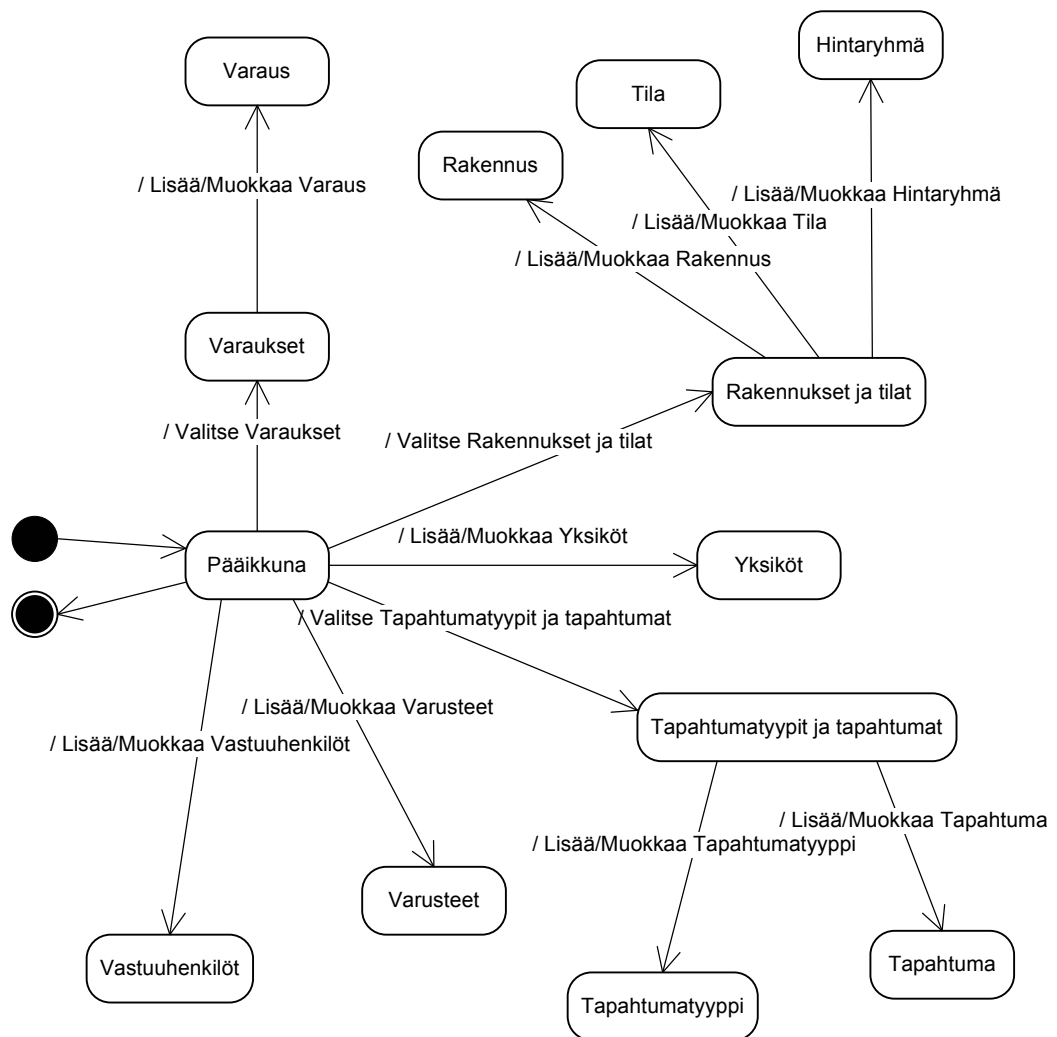
Tapahtumatyyppi	
1	Lisää tapahtumatyyppi
2	Muuta tapahtumatyypin tietoja
3	Poista tapahtumatyyppi

Tapahtuma	
1	Lisää tapahtuma
2	Muuta tapahtuman tietoja
3	Poista tapahtuma

Vastuuhenkilöt	
1	Hae vastuuhenkilöt
2	Lisää vastuuhenkilö
3	Valitse vastuuhenkilö
4	Muuta vastuuhenkilön tietoja
5	Poista vastuuhenkilö

Varusteet	
1	Hae varusteet
2	Lisää varuste
3	Valitse varuste
4	Muuta varusteen tietoja
5	Poista varuste

Liite 8. Dialogikaavio



Kuviosta on selkeyden vuoksi jätetty pois paluusuuntainen navigointi ikkunoiden välillä.