

Huoltokoneiden hallintajärjestelmän käyttöliittymäsuunnittelu

Janne Pulkkinen

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

2010



Tietojenkäsittelyn koulutusohjelma

<p>Tekijät Janne Pulkkinen</p>	<p>Ryhmä tai aloitusvuosi T6ti</p>
<p>Opinnäytetyön nimi Huoltokoneiden hallintajärjestelmän käyttöliittymäsuunnittelu</p>	<p>Sivu- ja liitesivumäärä 39 + 14</p>
<p>Ohjaaja tai ohjaajat Heikki Hietala</p>	
<p>Opinnäytetyön tavoitteena oli toteuttaa selainpohjainen käyttöliittymä, jonka tärkeimpinä ominaisuuksina olivat helppo opittavuus, virhemahdollisuuksien minimointi ja käyttäjän opastaminen. Käyttöliittymä suunniteltiin käytettäväksi selainpohjaisessa tietojärjestelmässä. Opinnäytetyössä pyrittiin luomaan ulkoasultaan yhdenmukainen ja helppokäyttöinen käyttöliittymä.</p> <p>Käyttöliittymä suunniteltiin huoltokoneiden hallintaprosessia varten tarkoitettuun tietojärjestelmään. Opinnäytetyössä kohdeyritykseksi oli valittu keskisuuri rautakauppa. Käyttöliittymä tai tietojärjestelmä ei tule oikeasti minkään yrityksen käyttöön.</p> <p>Opinnäytetyössä lähdettiin liikkeelle järjestelmän loppukäyttäjien profiloinnista ja järjestelmän toiminnallisesta määrittelystä. Loppukäyttäjiä profiloitiin luomalla fiktiivisiä käyttäjäpersoonia, jotka sopivat ominaisuuksiltaan järjestelmän käyttäjiksi. Järjestelmän toiminnallisuutta määriteltiin käyttäjäpersoonien avulla toteutettujen käyttötapauksien pohjalta. Käyttöliittymä toteutettiin käyttötapauksien mukaisesti hyödyntäen Sari A. Laakson kehittämää simulointipohjaista käyttöliittymäsuunnittelumallia. Käyttöliittymän toteutuksessa käytettiin HTML- merkkäuskieltä ja CSS- tyylikieltä.</p> <p>Opinnäytetyössä hyödynnettiin käyttöliittymäsuunnittelussa apuna käytettäviä heuristiikkoja, sekä ihmisen ja tietokoneen vuorovaikutuksen tutkimuksista saatuja tuloksia.</p> <p>Opinnäytetyössä onnistuttiin luomaan tavoitteiden mukainen käyttöliittymä. Käyttöliittymä simuloitiin käyttötapauksien pohjalta siten, että se ei sisällä turhia käyttöliittymäkomponentteja. Yhtenäisen ulkoasun ja toimintalogiikan ansiosta käyttöliittymä on helposti opittava ja se ohjaa käyttäjää järjestelmän virheettömään käyttöön.</p>	
<p>Asiasanat Käytettävyys, Selainpohjainen käyttöliittymä, HTML, CSS</p>	

Business information technology

<p>Authors Janne Pulkkinen</p>	<p>Group or year of entry T6ti</p>
<p>The title of thesis User interface design for service machinery management software</p>	<p>Number of pages and appendices 39 + 14</p>
<p>Supervisors Heikki Hietala</p>	
<p>The aim of the thesis was to implement web- based user interface, which main features were easy learnability, minimization of error opportunities and user guidance. User interface was designed for use in a web-based software. This thesis aimed to create a visually consistent and easy to use interface.</p> <p>User interface was designed for service machinery management software. The target company of the thesis was chosen to be a medium- sized hardware store. User interface or system will not be in a use of any company.</p> <p>The process of the thesis begins with profiling of the end users and functional specifications of the system. End users were profiled by creating fictitious user personas to fit the characteristics of the system users. Functionality of the system was defined through use case scenarios implemented by user personas. User interface was implemented in accordance with results discovered in use case scenarios utilizing simulation-based user interface design developed by Sari A. Laakso. The actual user interface was constructed with HTML- markup language and CSS- style sheet language.</p> <p>Heuristics of user interface design and results from human-computer interaction researches were utilized in this thesis.</p> <p>User interface that is consistent with the thesis objectives was successfully implemented in this project. User interface was designed with actual use cases so that it doesn't contain any useless components. Because of consistent appearance and functionality user interface is easy to learn and it guides user to a correct way of using the system.</p>	
<p>Key words Usability, Web-based user interface, HTML, CSS</p>	

Sisällys

1	Johdanto.....	1
1.1	Työn aihe, tavoite ja menetelmät.....	1
1.2	Työn rakenne ja rajaukset.....	2
1.3	Keskeiset käsitteet.....	2
2	Käytettävyys.....	3
2.1	Ihminen tietojärjestelmän käyttäjänä.....	3
2.1.1	Fyysiset ominaisuudet ja näköaisti.....	4
2.1.2	Muisti ja oppiminen.....	5
2.2	Käytettävyyden heuristiikat.....	6
2.2.1	Yhdenmukainen käyttöliittymä.....	7
2.2.2	Virheiden välttäminen ja virheilmoitukset.....	7
3	Käyttöliittymäsuunnittelu.....	8
3.1	Loppukäyttäjien profilointi.....	9
3.1.1	Käyttäjäpersoonat.....	9
3.1.2	Käyttötapaukset.....	10
3.2	Käyttöliittymän sommittelu.....	10
3.3	Tekstin, kuvien ja värien käyttö.....	10
4	Käyttöliittymän toteutus.....	11
4.1	Tavoite.....	11
4.2	Järjestelmän toiminnallinen määrittely.....	12
4.3	Käyttäjäpersoonat.....	12
4.4	Järjestelmän käyttötapaukset.....	14
4.5	Käyttöliittymän simulointi.....	16
4.5.1	Sommittelu käyttötapausten pohjalta.....	16
4.5.2	Visuaalinen ulkoasu.....	32
4.5.3	Virheilmoitukset.....	34
5	Pohdinta.....	36
5.1	Yhteenveto ja johtopäätökset.....	36
5.2	Jatkokehitysmahdollisuudet ja tulevaisuuden näkymät.....	38
6	Lähteet.....	39

7 Liitteet.....	40
Liite 1. Järjestelmään kirjautumis näkymä.....	40
Liite 2. Järjestelmään kirjautumis näkymän lähdekoodi.....	41
Liite 3. Järjestelmän päävalikko näkymä	42
Liite 4. Järjestelmän päävalikko näkymän lähdekoodi	43
Liite 5. Huoltolähetteen lisäys näkymä.....	44
Liite 6. Huoltolähetteen lisäys näkymän lähdekoodi.....	45
Liite 7. Huoltolähetteen tiedot näkymä.....	47
Liite 8. Huoltolähetteen tiedot näkymän lähdekoodi.....	48
Liite 9. Statuksen valinta näkymä	49
Liite 10. Statuksen valinta näkymän lähdekoodi.....	51

Kuvioluettelo

Kuvio 1. Järjestelmään kirjautuminen	16
Kuvio 2. Järjestelmän päävalikko.....	17
Kuvio 3. Asiakastietojen syöttäminen.....	17
Kuvio 4. Tuotetietojen hakeminen tietokannasta.....	18
Kuvio 5. Hakutulosten näyttäminen.....	18
Kuvio 6. Viankuvauksen syöttäminen.....	19
Kuvio 7. Huoltolähetteen tulostaminen.....	19
Kuvio 8. Päävalikkoon palaaminen.....	20
Kuvio 9. Järjestelmästä uloskirjautuminen.....	20
Kuvio 10. Vahvistus uloskirjautumisesta.....	20
Kuvio 11. Paluu takaisin muokkaamaan lähetteen tietoja.....	21
Kuvio 12. Tietojen muokkaus.....	21
Kuvio 13. Muutosten tallentaminen.....	22
Kuvio 14. Muokattu huoltolähete.....	22
Kuvio 15. Hakunäkymän avaaminen.....	23
Kuvio 16. Avoimien huoltojen hakunäkymä.....	23
Kuvio 17. Huoltolähetteen valitseminen listalta.....	24
Kuvio 18. Huoltolähetteen tiedot.....	24
Kuvio 19. Siirtyminen päävalikosta hakutoimintoon.....	25
Kuvio 20. Avoimien huoltojen hakunäkymä.....	25

Kuvio 21. Hakutulosten näyttäminen.....	26
Kuvio 22. Statuksen muokkaaminen.....	26
Kuvio 23. Statuksen vaihtaminen.....	27
Kuvio 24. Muokattu status.....	27
Kuvio 25. Huoltolähetteen poistaminen.....	28
Kuvio 26. Poistamistoiminnon valinta.....	29
Kuvio 27. Vahvistus lähetteen poistamisesta.....	29
Kuvio 28. Hakukriteerin valinta.....	30
Kuvio 29. Poistetun lähetteen palauttaminen.....	31
Kuvio 30. Statuksen ja päivämäärän valinta.....	31
Kuvio 31. Vahvistus huoltolähetteen palauttamisesta.....	32
Kuvio 32. Käyttöliittymän fontit.....	32
Kuvio 33. Käyttöliittymän värit.....	33
Kuvio 34. Päävalikon linkkien tyyli.....	34
Kuvio 35. Virhe sisäänkirjautumisessa.....	35
Kuvio 36. Puutteelliset asiakastiedot.....	35
Kuvio 37. Virheellinen hakukriteeri.....	36

1 Johdanto

Nykypäivänä melkein jokaisella suuremmalla yrityksellä on käytössään yksi tai useita erilaisia tietojärjestelmiä. Järjestelmät ovat kehittyneet vuosien saatossa yksinkertaista merkkipohjaisista järjestelmistä kaiken toiminnan kattaviksi massiivisiksi toiminnanohjausjärjestelmiksi. Järjestelmien laajeneminen on asettanut haasteita niiden käytettävyydelle. Järjestelmistä tulee helposti monimutkaisia ja vaikeita käyttää. Vielä nykyäänkin käytettävyyteen panostetaan liian vähän. Käytettävyys suunnittelussa tehty virhe saattaa ensi silmäykseltä tuntua sivustakatsojalle mitätömältä, mutta päivittäin järjestelmän parissa työskenteleville ihmisille tilanne on toinen.

Opinnäytetyötä on innoittanut tekijän kiinnostus käyttöliittymäsuunnittelua kohtaan ja henkilökohtaiset kokemukset tietojärjestelmien käytöstä. Hyvin usein törmää tietojärjestelmiin, joiden käyttöliittymä ei palvele järjestelmän käyttötarkoitusta parhaalla mahdollisella tavalla. Monelle käyttöliittymäsuunnittelijalle omien taitojen näyttäminen hienon visuaalisen ulkoasun ja ”ammattimaisen” lopputuloksen muodossa menee käytettävyyden edelle. Tämä näkyy loppukäyttäjille monimutkaisina ratkaisuuina, jotka turhaan vaikeuttavat järjestelmän käytettävyyttä. Monissa käyttöliittymissä on painikkeita ja tekstikenttiä, joiden tarkoitusta ei oikein kukaan tunnu tietävän.

1.1 Työn aihe, tavoite ja menetelmät

Opinnäytetyön tarkoituksena on suunnitella selainpohjainen käyttöliittymä huoltokoneiden hallintaa varten tarkoitettuun tietojärjestelmään. Järjestelmä ei tule oikeasti minkään yrityksen käyttöön. Tässä opinnäytetyössä se suunnitellaan keskisuurelle rautakaupalle, jonka takuuhuolto määrä on noin 300-500 laitetta vuodessa. Järjestelmän tarkoitus on helpottaa rautakaupan huoltokoneiden käsittelyprosessia. Lähtötilanteessa kaupalla on käytössään huoltoliikkeeseen faksattava paperinen huoltolähete. Tämän käytännön kanssa on syntynyt ongelmia kun varastoon jää lojumaan laitteita, joista kukaan ei oikein tiedä ovatko ne lähdössä huoltoon vai tulossa huollosta ja milloin laitteet on lähetetty tai vastaanotettu. Uusi järjestelmä tarjoaa henkilökunnalle ja asiakkaille ajankohtaista tietoa huollon statuksesta ja laitteen olinpaikasta.

Järjestelmän käyttöliittymä toteutetaan työssä käsiteltävien käytettävyyden heuristiikkojen mukaisesti ja sen tärkeimpinä tavoitteina ovat helppo opittavuus, virheiden teko mahdollisuuksien minimointi ja käyttäjän opastaminen.

Aihepiiriksi on valittu rautakaupan käyttöön tuleva huoltokoneiden hallintajärjestelmä, koska järjestelmän loppukäyttäjäkunta koostuu sekalaisen tietokoneen käyttökokemuksen omaavista ihmisistä, joten käytettävyys muodostuu tärkeäksi tekijäksi tällaisessa lähtötilanteessa.

Käyttöliittymä suunnitellaan ja toteutetaan käytettäväksi web- selainpohjaisessa ympäristössä hyödyntäen HTML- merkkäuskieltä ja CSS- tyylikieltä. Selainpohjainen käyttöliittymä on valittu, koska se vastaa parhaiten loppukäyttäjien tottumuksia. Käyttöliittymä suunnitellaan simuloinnin avulla siten, että se rakennetaan vaihe kerrallaan käyttötapausten pohjalta. Tällöin käyttöliittymään ei tule ylimääräisiä elementtejä, koska kaikki käyttöliittymän osat palvelevat jotain käyttötapauksissa ilmenevää tarvetta.

1.2 Työn rakenne ja rajaukset

Työn alussa esitellään opinnäytetyön viitekehys, joka on jaettu kahteen päälukuun. Ensimmäisessä luvussa käsitellään käytettävyyden teoriaa ja ihmistä tietojärjestelmän käyttäjänä. Toisessa luvussa käsitellään käyttöliittymäsuunnittelun teoriaa.

Empiirinen osa koostuu käyttöliittymän suunnittelun ja toteutuksen prosessikuvauksista. Suunnittelu vaiheessa profiloidaan järjestelmän loppukäyttäjät ja järjestelmällä tehtäviä toimenpiteitä. Käyttöliittymän toteutus ja simulointi tehdään näiden määritysten pohjalta.

Opinnäytetyön yhteenvedossa pohditaan päästiinkö tavoitteisiin ja käydään mahdolliset kehitysehdotukset läpi. Opinnäytetyö käsittelee käyttöliittymän käytettävyyttä, eikä ota kantaa järjestelmän tekniseen toimivuuteen muuten kuin käyttöliittymän simulointia varten tehdyn toiminnallisen määrittelyn osalta. Opinnäytetyössä toteutetaan siis järjestelmän selainpohjainen käyttöliittymä, mutta ei koko järjestelmää.

1.3 Keskeiset käsitteet

Käyttöliittymä on ohjelman tai tuotteen osa, jonka avulla sitä käytetään. Normaalissa tietojärjestelmässä käyttöliittymä tarkoittaa ruudulla näkyvää visuaalista käyttöliittymää, sekä näppäimistöä ja hiirtä.

Käytettävyys on tuotteen tai laitteen ominaisuus, jonka avulla määritellään kuinka helppoa ja nopeasti opittavaa sen käyttö on.

HTML HyperText Markup Language on internetsivujen muodostamiseen käytetty merkkuskieli. Kielen avulla määritellään internetsivujen rakennetta ja sivuilla käytettäviä komponentteja.

CSS Cascading Style Sheets on internetsivujen yhteydessä käytettävä tyyliarkki, jossa määritellään sivuilla käytettävä visuaalinen ulkoasu.

2 Käytettävyys

Jakob Nielsen määrittelee käytettävyyden osaksi tuotteen hyödyllisyyttä ISO-9241 standardissa. (ISO 9241-11, Ergonomic requirements for office work with visual display terminals (VDTs) 1998, part 11). Käytettävyys voidaan nähdä tietojärjestelmän ominaisuutena. Se kuvaa kuinka helposti ja tehokkaasti käyttäjä pystyy suorittamaan haluamansa tehtävän.

Tietojärjestelmän käytettävyys rakentuu järjestelmän opittavuudesta, muistettavuudesta, tehokkuudesta, virheettömyydestä ja miellyttävyydestä käyttäjä. Käytettävyys ammentaa oppinsa kognitiivisesta psykologiasta ja ihminen-tietokone-vuorovaikutus tutkimuksista saaduista tuloksista. (Sinkkonen, Kuoppala, Parkkinen 2006, 11.)

2.1 Ihminen tietojärjestelmän käyttäjänä

Suunnitellessa tietojärjestelmää tai mitä tahansa tuotetta, joka on tarkoitettu ihmisen käytettäväksi, on otettava huomioon ihmisten fyysiset ja henkiset kyvyt. Ihmisten aistit, ajatustoiminta, päättelymekanismit ja fyysiset ominaisuudet määrittelevät pitkälti miten kykenemme tietojärjestelmää käyttämään. Ihminen on monimutkainen olento, jolla on paljon aisteja ja erilaisia tapoja omaksua asioita. Aistit, ajatustoiminta ja erilaiset päättelymekanismit ovat käytettävyyden kannalta ratkaisevassa asemassa. (Kuutti 2003, 22-23.)

Myös kulttuurilla on oma vaikutuksensa ihmisen käyttäytymiseen. Kulttuurina voidaan pitää lähes kaikkea ihmisyhteisön tuottamaa informaatiota kuten kieliä, tapoja, tottumuksia ja normeja. (Kuutti 2003, 22-23.) Ihmisen biologiset ominaisuudet ovat lähtökohtaisesti hyvin samanlaisia, mutta myös kulttuurilla on suuri vaikutus siihen miten hyödynämme ominaisuuksiamme. Esimerkiksi kiinalaisella ja suomalaisella käyttäjällä voi olla samanlaiset valmiudet havainnointiin ja lukemiseen, mutta kulttuurista johtuen suomalainen käyttäjä etsii päävalikkoa vasemmalta puolelta sivua ja kiinalainen oikealta.

Kulttuuriset tekijät yhdessä biologisten ominaisuuksien kanssa ohjaavat ihmisen käyttäytymistä erilaisissa tilanteissa, myös tietojärjestelmien käytössä. (Sinkkonen ym. 2006, 27-28.)

2.1.1 Fyysiset ominaisuudet ja näköaisti

Suurinta osaa normaaleista tietojärjestelmistä pystyy käyttämään pelkästään hiiren ja näppäimistön avulla. Tällaisissa järjestelmissä käyttäjän fyysiset ominaisuudet eivät ole niin ratkaisevassa roolissa kuin niin sanotuissa sulautetuissa järjestelmissä, joissa järjestelmää käytetään yhdessä jonkun laitteen kanssa. Pääsääntöisesti tietojärjestelmien käyttö onnistuu keneltä tahansa normaaleilla fyysisillä ominaisuuksilla varustetulta henkilöltä. Fyysisten ominaisuuksien tärkeys korostuu kuitenkin siinä tapauksessa, jos järjestelmää suunnitellaan jollekin fyysisiä rajoitteita omaavalle kohderyhmälle kuten lapsille tai liikuntarajoitteisille. Nuorimmilla lapsilla lihaksiston ja hermoston koordinaatio ei ole välttämättä vielä kehittynyt täysin ja jotkut toiminnot voivat tuntua hankalilta. (Kuutti 2003, 47-48.)

Tietojärjestelmien käytön suhteen ihmisen tärkein aisti on luonnollisesti näkö. Tästä johtuen suurin osa käyttöliittymistä on visuaalisia. Näköaisti koostuu fysiologisesta näkemisestä ja näköhavainnon tulkinnasta aivoissa. Ihmisen silmässä on soluja, jotka aistivat valoa ja värejä eritavalla. Käyttöliittymien suunnittelun suhteen on tärkeä tietää miten ihminen pystyy havaitsemaan väriyhdistelmiä, yksityiskohtia ja tulkitsemaan havaintojaan. Värejä havainnoivista soluista vain noin 2 % on herkkiä sinisen valon aaltopituudelle. Tästä johtuen sinisten yksityiskohtien ja kahden eri sinisen sävyn erottaminen toisistaan on ihmissilmälle huomattavasti vaikeampaa kuin muiden värien sävyjen. Ihmisen aivoille on tyypillistä ryhmitellä yhteenkuuluviksi näkemiään asioita. Tätä kannattaa hyödyntää käyttöliittymäsuunnittelussa. Toisiaan lähellä olevat, samanlaiset ja väreiltään yhtenäiset kohteet mielletään helposti yhdeksi loogiseksi kokonaisuudeksi (Kuutti 2003, 25-28.)

Ihmisen havainnointikyvyn rajoitukset kannattaa ottaa suunnittelutyössä huomioon. Ihmisen keskittäessä katseensa esimerkiksi tekstiin, pystyy hän havaitsemaan selkeästi noin 12 merkkiä eteenpäin ja 4 taaksepäin kulttuurin mukaisen lukusuunnan mukaisesti. Tämä kannattaa pitää mielessä tekstin sommittelussa ja määrässä.

(Sinkkonen ym. 2006, 57-59.)

2.1.2 Muisti ja oppiminen

Ihmisen muisti jaetaan yleensä pitkä- ja lyhytkestoiseen muistiin. Pitkäkestoissa muistissa on varastoituna jo opittuja tietoja ja taitoja, kun taas lyhytkestoinen muisti on eräänlainen työmuisti, jota ihminen tarvitsee suorittaessaan erilaisia askareita. Esimerkiksi kokatessa pitää muistaa mitä aineksia ruokaan tulee ja missä järjestyksessä ne lisätään. Pitää myös muistaa mitä aineita ruokaan on jo lisännyt ja mitä lisätään seuraavaksi. (Kuutti 2003, 47-48.)

Lyhytkestoisen muistin kapasiteetti on varsin pieni. Lyhytkestoisessa muistissa pysyvät vain ne muutamat asiat joita aivot käsittelevät sillä hetkellä. Lyhytkestoista muistia tarvitaan esimerkiksi matemaattisten ongelmien ratkomiseen tai vastaavaa ajatustyötä vaativien toimintojen suorittamiseen. Lyhytkestoisen muistin kapasiteetti on keskimäärin kolmesta seitsemään asiaa. Tämä on hyvä muistaa käyttöliittymäsuunnittelussa, jottei muistia rasiteta liikaa. (Sinkkonen ym. 2006, 149-150.)

Pitkäkestoisen muistin avulla ihminen pystyy muistamaan vuosia taaksepäin tapahtuneita asioita, koska sen kapasiteetti on suuri ja kestoikä pitkä. Pitkäkestoiseen muistiin tallentuvat kaikki tiedot, taidot ja muistot, joita ihminen on elämänsä aikana oppinut ja kokenut. Esimerkiksi pienenä lapsena opeteltu pyörällä ajamisen taito pysyy ihmisen muistissa aina eläkepäiville asti. Pitkäkestoinen muisti ei tyhjenny samalla tavalla kuin lyhytkestoinen muisti. Opitut asiat pysyvät muistissa, mutta eivät ole aina saatavilla. Pitkäkestoiseen muistiin tallentunut asia saattaa unohtua pitkäksi aikaa ja muistua taas mieleen myöhemmin yleensä johonkin aikaisempaan tapahtumahetkeen liittyvän tunnetilan tai aistimuksen avulla. (Sinkkonen ym. 2006, 153-155.)

Muistiin tallennettujen tietojen ja taitojen uudelleen hyödyntäminen vaatii niiden oppimista. Taitojen oppiminen voi tapahtua kokemusten kautta tai tietoisesti opiskelemalla. Yksi tehokas oppimisen väline on toistaminen. Sellainen taito, jota ei käytetä kovin usein, pysyy muistissa pitkään, jos sitä on opiskeluvaiheessa toistettu useampi kerta. Oppimisprosessissa ihminen pystyy käyttämään kaikkia aistejaan. Tästä johtuen oppimistyylyissä on paljon yksilöllisiä eroja. Jotkut oppivat parhaiten näkemäänsä, jotkut kuulemaansa ja toiset tekemäänsä. Oppiminen on käyttöliittymäsuunnittelussa todella tärkeässä asemassa. Jopa todella yksinkertainen käyttöliittymä, jonka toimintalogiikka on useammille käyttäjille selkeä jo ensimmäisellä käyttökerralla, on joskus vaatinut opettelua. Ihmiset toimivat käyttötilanteissa aikaisemman kokemusmaailman tuoman käsityksen mukaisesti. Aikaisempi kokemusmaailma on kuitenkin aina opittua, eikä synnyntäistä tietoa. (Kuutti 2003, 41-42.)

2.2 Käytettävyyden heuristiikat

Käytettävyyden arvioinnin perustaksi on kehittynyt niin sanottuja heuristiikkoja. Nämä ovat yleisesti tunnettuja sääntöjä ja normeja, joita käytettävyydeltään hyvän tietojärjestelmän tulisi noudattaa. Heuristiikkoja on monenlaisiin käyttötarkoituksiin sopivia. Yleispäteviä heuristiikkoja, jotka sopivat melkein kaikenlaisten käyttöliittymien kanssa sovellettavaksi, sekä erikoiskäyttöön räätälöityjä.

Heuristiikkoja ovat luoneet useat aikojen saatossa käyttöliittymien parissa työskennelleet ihmiset. Laajalti käyttöliittymien suunnittelun yhteydessä käytetään Jakob Nielsenin luomaa 10-heuristiikan listaa. Nielsenin näkemyksen mukaan käytettävyydeltään hyvän käyttöliittymän pitää sisältää seuraavat ominaisuudet:

- **Järjestelmän tilan näkyvyys.** Käyttäjälle tulee tarjota ajankohtaista tietoa järjestelmän tilasta ja käyttäjän sijainnista järjestelmässä.
- **Järjestelmän ja tosielämän vastaavuus.** Käyttäjää tulee puhutella selkeällä kielellä. Hankalasti ymmärrettävää atk- sanastoa tulee välttää.
- **Kontrolli ja vapaus.** Järjestelmässä on oltava selkeät poistumistiet, joita hyödyntämällä käyttäjä pääsee poistumaan helposti, jos hän on tehnyt virheellisen valinnan.
- **Käyttöliittymän yhtenäisyys.** Käyttöliittymän toimintalogiikkaa ja ulkoasu tulee olla samanlainen koko järjestelmässä.
- **Virheiden estäminen.** Käyttöliittymän tulee ohjata käyttäjää virheettömään käyttöön.
- **Tunnistaminen ennen muistamista.** Käyttäjän muistia ei kannata rasittaa liikkaa. Käyttöliittymä komponenttien tarkoitus tulisi olla helposti pääteltävissä. Käyttäjää ei saa pakottaa muistamaan asioita järjestelmän siirtyessä uuteen näkymään.
- **Käytön tehokkuus ja joustavuus.** Järjestelmän käytön tulisi olla tehokasta ja joustavaa käyttäjän osaamistasosta riippumatta. Järjestelmän tulisi tarjota mahdollisuudet pikavalintojen luomiselle usein käytettyjä toimintoja varten.
- **Esteettinen ja minimaalinen ulkoasu.** Käyttöliittymässä tulee olla vain tarvittavat käyttöliittymä komponentit. Käytettävät värit ja fontit täytyy olla selkeitä ja helppolukuisia.

- **Virhetilanteiden tunnistaminen, ilmoittaminen ja korjaaminen.** Järjestelmässä käytettävät virheilmoitukset ovat selkeitä ja tarjoavat ratkaisun ongelmaan.
- **Opastus ja ohjeistus.** Ohjeiden tulee olla helposti ja nopeasti löydettävissä. Niiden pitää olla tarpeeksi lyhyitä ja helposti ymmärrettäviä.

(Kuutti 2003, 47-48.)

2.2.1 Yhdenmukainen käyttöliittymä

Hyvän käytettävyyden saavuttamiseksi koko järjestelmän toimintalogiikka tulisi olla yhtenäinen. Hyvin toteutetussa järjestelmässä käyttäjä voi tulla uuteen toimintoon tai järjestelmän osaan ja pystyä käyttämään sitä aikaisemmin omaksutun toimintalogiikan perusteella. Samojen toimintojen tulisi toimia samalla tavalla kaikissa järjestelmän osissa. Epäjohdonmukaisuuksia tulisi välttää, koska ne aiheuttavat ongelmia käytön suhteen. Erityisesti oikopoluissa ja näppäinyhdistelmissä täytyy olla erityisen varovainen. Mikäli tietty näppäinyhdistelmä tallentaa jossain järjestelmän osassa ja poistaa toisessa niin virheitä syntyy varmasti. (Kuutti 2003, 55-56.)

Yhdenmukaisuus on tärkeää myös visuaalisessa ulkoasussa ja komponenttien sijoittelussa. Tekstikenttien, valikkojen ja painikkeiden sijoittelussa säilytetty yhdenmukaisuus auttaa käyttäjää omaksumaan järjestelmän toimintalogiikan ilman, että hänen tarvitsee jokaisessa uudessa osiossa erikseen omaksua uudenlainen ulkoasu ja komponenttien sijainti. (Kuutti 2003, 55.)

2.2.2 Virheiden välttäminen ja virheilmoitukset

Käyttöliittymän suunnittelussa tulisi pyrkiä minimoimaan käyttäjän virheenteko mahdollisuudet. Järjestelmässä saattaa olla toimintoja, jotka ovat hyvinkin virhealttiita. Hyvällä suunnittelulla näistä seuraavia ongelmatilanteita pystytään kuitenkin välttämään. Yleisimpiä ongelmatilanteita aiheuttavat yksinkertaiset näppäilyvirheet, joita voidaan estää esimerkiksi tarjoamalla käyttäjälle tiedoston tai toiminnan valintaa listalta sen sijaan, että käyttäjä joutuisi itse kirjoittamaan tiedoston nimen. Ongelmatilanteita saattavat aiheuttaa myös järjestelmät, jotka toimivat tietyssä tilassa eri tavalla kuin normaalisti. Esimerkiksi Microsoft Wordin ollessa tekstinlisäystilassa, se lisää kirjoitetun tekstin vanhan tekstin väliin, kun taas korvaustilassa se kirjoittaa olemassa olevan tekstin päälle korvaten sen uudella tekstillä. (Kuutti 2003, 62-64.)

Vaikka tietojärjestelmän suunnittelussa olisi onnistuttu minimoimaan virheen teko mahdollisuudet, niin silti järjestelmän käytössä saattaa tapahtua jotain odottamatonta joka johtaa virhetilanteeseen. Käytettävyyden kannalta näiden virhetilanteiden tehokas hoitaminen on erittäin tärkeää. Ongelmatilanteissa käyttäjää tulee ohjata virheilmoituksilla, joiden tärkein tehtävä on olla selkokielisiä ja helposti ymmärrettäviä ohjeita tilanteen selvittämiseksi. Pelkkä ilmoitus toiminnon epäonnistumisesta ei riitä. Virheilmoitukset eivät saa olla liian ympäröiväitä vaan niiden pitää pureutua tarkasti juuri siihen ongelmaan johon ne vastaavat. Hyvässä virheilmoituksessa on selkeästi ilmoitettu mistä virhe johtui, mitä on tehtävissä sen korjaamiseksi ja miten sama virhe voidaan jatkossa välttää. Käyttöliittymäsuunnittelussa tulisi pyrkiä siihen, että virhetilanteista pystytään aina toipumaan, joidenkin suurempien järjestelmien kanssa käytetään vikakoodeja, joiden avulla tukipalvelussa voidaan selvittää ongelman syytä. Vikakoodit tulisi sijoittaa virheilmoitusten loppuun, etteivät ne häiritse virheilmoituksen asiasisällön ymmärrettävyyttä. (Kuutti 2003, 61-62.)

3 Käyttöliittymäsuunnittelu

Käyttöliittymällä tarkoitetaan kaikkia toimintoja ja välineitä, joiden avulla ihminen käyttää tietojärjestelmää. Itse näytöllä näkyvän tietojärjestelmän lisäksi käyttöliittymään kuuluvat sen käyttöön tarvittavat fyysiset laitteet, kuten näppäimistö ja hiiri.

Käyttöliittymäsuunnittelussa pyritään luomaan tarvittavat työkalut tietojärjestelmän käyttöä varten. Käyttöliittymäsuunnittelussa tehtyihin ratkaisuihin vaikuttavat paljolti järjestelmän loppukäyttäjien taitotaso ja tarpeet, sekä järjestelmän käyttötarkoitus ja toiminnallisuus. Mikäli suunnitellaan atk-ammattilaisten käyttöön tarkoitettua tietojärjestelmää, ovat käyttöliittymä ratkaisut erilaisia kuin suuremman kohderyhmän tarpeisiin räätälöidyissä järjestelmissä. (Kallio 1992, 49-51.)

Simulointipohjainen käyttöliittymäsuunnittelu (*Goal-Derived Design, GDD*) on Helsingin Yliopiston tietojenkäsittelytieteenlaitoksen professorin Sari A Laakson kehittämä työkalu käyttöliittymien suunnitteluun. Sen tarkoituksena on lähteä rakentamaan käyttöliittymää vastaamaan käyttäjien tarpeita konkreettisten käyttötapausten pohjalta vaihe kerrallaan. Suunnittelumallissa ei synny turhia toimintoja tai käyttöliittymän osia, koska kaikki osat on suunniteltu vastaamaan jotain tiettyä tarvetta tai käyttötarkoitusta.

Simulointiprosessissa selvitetään käyttöliittymässä tarvittavat toiminnot ja tietosisältö, sekä järjestelmän visuaalinen ulkoasu ja toimintalogiikka.

(<http://www.interacta.fi/suunnitteluvaihe.html>)

3.1 Loppukäyttäjien profilointi

Tietojärjestelmän loppukäyttäjien profilointi on tärkeää, jos suunnitellaan ryhmälle jonka tarpeet ja tottumukset eivät ole selvillä, ei päästä parhaaseen mahdolliseen lopputulokseen. Käyttöliittymän toteuttamiseen tarvittava informaatio on suoraan johdettavissa tiedosta mitä käyttäjistä on tarjolla. Ideaalitulanteessa voitaisiin toteuttaa täysin käyttäjälle sopiva käyttöliittymä, mikäli käytettävissä olisi täydellisen tarkka kuvaus käyttäjän ominaisuuksista ja vaatimuksista. Käytännössä näin ei kuitenkaan ole. Käyttäjän mallintaminen on hankalaa ja kallista, lisäksi käyttäjän täydellinen mallintaminen on lähestulkoon mahdotonta. (Kuutti 2003, 117-118.)

Loppukäyttäjien profilointiin on kehitetty monenlaisia tekniikoita. Yksinkertaisimpien tekniikoiden avulla pystytään tekemään kevyitä mallinnuksia loppukäyttäjistä keräämällä ylimalkaisesti vain tiettyjä käyttäjiä koskevia tietoja. Kehittyneempien suunnittelumenetelmien avulla voidaan puolestaan laatia tarkkoja määrittämiä, joista saadaan jalostettua toteutettavan järjestelmän määrittely eri vaiheiden kautta. (Kuutti 2003, 117-118.)

3.1.1 Käyttäjäpersoonat

Yksi tapa profiloida loppukäyttäjiä on luoda kuvitteellisia käyttäjäpersoonia. Käyttäjäpersoonille määritellään ominaisuuksia mahdollisimman tarkasti. Heidät nimetään, määritellään luonteenpiirteitä, ominaisuuksia, harrastuksia jne. Käyttäjäpersoonasta tehdään järjestelmän käyttäjäksi mahdollisimman luonnollisesti sopiva.

Suunnittelutyön kannalta on järkevämpää käyttää mielikuvituksellisia hahmoja kuin oikeita henkilöitä, koska oikeilla ihmisillä on erilaisia käyttäytymistapoja ja ominaisuuksia, jotka saattavat hämätä suunnittelutyötä. Kuvitteellisille käyttäjäpersoonille on määritelty ominaisuuksia, jotka vastaavat paremmin koko käyttäjäkuntaa yleisellä tasolla. (Kuutti 2003, 122-123.)

3.1.2 Käyttötapaukset

Käyttötapaukset ovat mallinnuksia tehtävistä, joita käyttäjä suorittaa järjestelmää hyödyntäen. Käyttötapauksissa on kuvattu vaihe-vaiheelta toiminnot, jotka käyttäjä suorittaa päästäkseen haluamaansa lopputulokseen.

Käyttötapaus on nopea ja kustannustehokas menetelmä, mutta se ei anna niin laajaa käsitystä käyttäjän tarpeista kuin muut menetelmät. Tästä johtuen se sopii parhaiten kevyempien järjestelmien kanssa käytettäväksi. Käyttötapauksia voidaan kuvata joko tekstinä tai kaavioina. Käyttötapaus tarkastelee järjestelmän toimintaa käyttäjän näkökulmasta. (Kuutti 2003, 132-134.)

3.2 Käyttöliittymän sommittelu

”Käyttöliittymän visuaalisen ulkoasun sommittelulla tarkoitetaan yleensä yhden kokonaisuuden, tyypillisesti näytön sisällön sijoittelua.” (Kuutti 2003, 91.) Sommittelu tulisi toteuttaa siten, että käyttäjän huomio kiinnittyy oikeisiin asioihin oikeassa järjestyksessä. Länsimaisessa kulttuurissa ihmiset lukevat vasemmalta oikealle ja ylhäältä alas. Tämän ohjesäännön mukaisesti olisi hyvä edetä myös käyttöliittymä suunnittelussa. Käyttäjää tulisi opastaa normaalin luku-suunnan mukaisessa etenemisessä, koska se on ihmiselle luonnollisin tapa omaksua informaatiota. Harhaan johtavat visuaaliset ärsykkeet saattavat suistaa käyttäjän tältä reitiltä hidastaen etenemistä ja rasittaen havainnointikykyä. (Kuutti 2003, 91-92.)

Oikeanlaisella visuaalisella hierarkialla pystytään ohjaamaan käyttäjää. Tärkeät asiat tulee olla keskeisillä paikoilla ja helposti erotettavissa. Loogisesti yhteen kuuluvat kokonaisuudet kuuluvat myös visuaalisesti yhteen. Painikkeiden ja opasteiden tulisi kuvata käyttöliittymän toimintojen ja osien välisiä suhteita selkeästi. (Krug 2006, 31-32.)

3.3 Tekstin, kuvien ja värien käyttö

Selkeässä käyttöliittymässä on käytetty sopivassa suhteessa sekä kuvia, että tekstiä ikään kuin tukemaan toisiaan. Pelkkää tekstiä sisältävä käyttöliittymä saattaa helposti muodostua vaikeasti tulkittavaksi ja hidastaa käyttöä. Tekstin korvaaminen kuvilla ei myöskään ole suositeltavaa, koska joidenkin kuvien merkitys esitetyssä asiayhteydessä saattaa jäädä epäselväksi kun ei ole tekstiä selventämässä kuvan tai symbolin merkitystä. (Kuutti 2003, 98-100.)

Käyttöliittymän teksteissä kannattaa käyttää mahdollisimman selkeää ja helppolukuista fonttia. Erilaisia fontteja tulisi käyttää varoen, ettei käyttöliittymästä tule vaikealukuinen. Selainpohjaisessa käyttöliittymässä yleinen käytäntö on yhtenäiset fontit pääotsikoille, alaotsikoille ja runkotekstille. (Sinkkonen ym. 2006, 105-110.) Selainpohjaisessa käyttöliittymässä kannattaa käyttää niin sanottuja ”web-safe” fontteja eli yleisiä fontteja, joiden lukemiseen pystyvät kaikki web-selaimet ja ympäristöt. Standardeja web- fontteja ovat: Verdana, Georgia, Times New Roman, Times, Arial, Helvetica, Tahoma, Comic Sans MS, Trebuchet ja Courier. (Collison 2006, 59-60.)

Värejä voidaan käyttää käyttöliittymässä muun muassa opastamisen apuvälineenä. Esimerkiksi liikennevaloista tutut punainen ja vihreä toimivat hyvin viestimään jonkun asian onnistumisesta tai epäonnistumisesta. Voimakkaiden vastavärien käyttöä vierekkäin tulisi välttää etenkin tekstin ja taustan väreinä, koska se vaikeuttaa lukemista ja rasittaa silmiä. (Kuutti 2003, 100-102.) Loogisten kokonaisuuksien selkeyttämiseksi kannattaa käyttää värejä. Otsikoiden ja alaotsikoiden ollessa samanvärisiä käyttäjän on helpompi ymmärtää niiden kuuluvan samaan ryhmään. (Sinkkonen ym. 2006, 114-115.)

4 Käyttöliittymän toteutus

4.1 Tavoite

Käyttöliittymäsuunnittelussa toteutetaan selainpohjainen käyttöliittymä viitekehyksessä käsiteltyjen käytettävyysteorioiden pohjalta. Tavoitteena on luoda käyttäjille visuaalisesti tuttu toimintaympäristö, jonka oppiminen on helppoa ja nopeaa.

Tietojen esittämisessä ja ulkoasun sommitelussa pyritään siihen, että käyttäjälle näytetään vain hänen tarvitsemansa tiedot oikeissa paikoissa. Käyttöliittymässä ei pyritä näyttävään visuaaliseen ulkoasuun vaan mahdollisimman selkeään lopputulokseen. Käyttöliittymän sommittelulla, sekä fonttien ja värien käytöllä pyritään helpottamaan järjestelmän käytettävyyttä selkeyttämällä loogisia kokonaisuuksia ja opastamalla käyttäjää järjestelmän virheettömään käyttöön.

4.2 Järjestelmän toiminnallinen määrittely

Järjestelmän tarkoituksena on toimia rautakaupan henkilökunnan apuna huoltokoneiden hallintaa varten. Järjestelmä varastoi huoltoprosessiin liittyviä asiakas- ja laitetietoja tietokantaan, josta ne ovat helposti luettavissa ja päivitettävissä.

Rautakaupalla on huoltosopimus takuunalaisista laitteista huoltoliikkeen kanssa. Asiakkaalla tulee olla esittää ostokuitti tai takuutodistus laitteesta ennen huoltolähetteen tekemistä. Huoltoliike tunnistaa takuunalaiset laitteet ja näkee tarkemmat tuotetiedot laitteen tuotenumeron perusteella.

Tuotenumero on pakollinen huoltolähetettä tehdessä. Järjestelmän perus toimintalogiikka on seuraavanlainen: 1. Asiakas tuo viallisen koneen liikkeeseen, jonka jälkeen myyjä syöttää järjestelmään asiakkaan- ja laitteen tiedot. 2. Tietojen syöttämisen jälkeen myyjä tallentaa huoltolähetteen. Lähetete tallentuu tietokantaan statuksella ”lähetetty huoltoon”. Tallentamisesta lähtee automaattinen lähetete huoltoliikkeeseen sähköpostilla. Tämän jälkeen kone on valmis toimitettavaksi huoltoliikkeeseen toimenpiteitä varten. 3. Myymälän vastaanottaa huollosta palautuneen koneen, avaa järjestelmästä avoimen huoltolähetteen, kuittaa laitteen vastaanotetuksi ja ilmoittaa asiakkaalle sähköpostilla tai puhelimella huoltokoneen olevan valmis ja noudettavissa myymälästä. Tässä vaiheessa huoltolähetete saa statuksen ”palautunut huollosta, odottaa noutoa”. 4. Asiakas noutaa laitteen ja myyjä kuittaa huoltolähetteen valmiiksi, jolloin statukseksi muuttuu ”valmis, noudettu”.

Järjestelmän toimintaan liittyy myös tietojen muokkausta ja poistamista koskevia toimenpiteitä. Niihin palataan tarkemmin luvussa 5.3 käyttötapaukset

4.3 Käyttäjäpersoonat

Toteutettavan järjestelmän loppukäyttäjä kunta koostuu monen ikäisistä, erilaisen koulutus- ja työhistorian omaavista ihmisistä. Kyseiselle käyttäjäkunnalle on vaikea määrittää mitään yhteistä ominaisuutta, luonteenpiirrettä tai tietojärjestelmien käyttö taitotasoa. Tämän takia tässä projektissa on päädytty käyttämään käyttäjäpersoonia loppukäyttäjien profilointiin. Luotuja käyttäjäpersoonia hyödynnetään myöhemmin myös käyttöliittymän simuloinnissa.

Käyttäjäpersoonaa 1

Nimi: Jenni

Ikä: 19- vuotta

Koulutustausta: ylioppilas

Harrastukset: lenkkeily ja kuntosalit

Luonne: sosiaalinen ja ulospäin suuntautunut

Tietokoneen käyttötaito- ja historia: Käyttää tietokonetta lähinnä internetin selailuun ja sosiaaliseen mediaan. Selainpohjainen käyttöliittymä on tuttu, mutta kokemusta varsinaisista tietojärjestelmistä ei ole.

Käyttäjäpersoonaa 2

Nimi: Matti

Ikä: 53- vuotta

Koulutustausta: Lvi-asentaja

Harrastukset: ulkoilu ja television katselu

Luonne: ahkera ja hieman vanhanaikainen

Tietokoneen käyttötaito- ja historia: Ei ole nuoruudessaan käyttänyt tietokoneita. Opetellut vanhemmalla iällä internetin ja sähköpostin käytön alkeet. Keskittymiskyky ei ole kovin korkea, turhautuu helposti ja käyttäminen on hidasta.

Käyttäjäpersoonaa 3

Nimi: Liisa

Ikä: 44- vuotta

Koulutustausta: Sihteeri

Harrastukset: koirien kanssa lenkkeily

Luonne: ujo ja hiljainen

Tietokoneen käyttötaito- ja historia: On aikaisemmin urallaan käyttänyt merkkipohjaisia tietojärjestelmiä, sähköpostia ja internetiä jonkun verran. Käyttäjä on tottunut merkkipohjaisten järjestelmien käyttölogiikkaan.

Käyttäjäpersoona 4

Nimi: Harri

Ikä: 24- vuotta

Koulutustausta: Opiskelee tradenomiksi

Harrastukset: jääkiekko

Luonne: sosiaalinen ja kunnianhimoinen

Tietokoneen käyttötaito- ja historia: Oma paljon kokemusta internetin käytöstä, sähköpostista ja yleisimmistä toimisto-ohjelmista.

4.4 Järjestelmän käyttötapaukset

Käyttötapauksien avulla mallinnetaan kaikki järjestelmän avulla tehtävät toimenpiteet. Huoltokoneiden hallintajärjestelmän avulla käyttäjän tulee pystyä luomaan uusi huoltotapahtuma, muokkaamaan tietoja, muuttamaan lähetteen statusta ja poistamaan huoltolähete.

Käyttötapaus 1: Uuden huoltolähetteen luominen.

Vaihe 1: myyjä kirjautuu järjestelmään.

Vaihe 2: myyjä avaa uuden huoltolähetteen.

Vaihe 3: myyjä syöttää asiakkaan tiedot ja hakee tietokannasta huollettavan laitteen.

Vaihe 4: myyjä tallentaa huoltolähetteen.

Vaihe 5: myyjä tulostaa huoltolähetteen.

Vaihe 6: myyjä palaa takaisin päävalikkoon.

Vaihe 7: myyjä kirjautuu ulos järjestelmästä.

Käyttötapaus 2: Huoltolähetteen tietojen muokkaaminen juuri luodusta lähetteestä.

Vaihe 1: luotuaan huoltolähetteen myyjä huomaa syöttäneensä asiakkaan nimen väärin.

Vaihe 2: myyjä muokkaa lähetteen tietoja.

Vaihe 3: myyjä tallentaa muokatun lähetteen.

Vaihe 4: myyjä palaa takaisin päävalikkoon.

Vaihe 5: myyjä kirjautuu ulos järjestelmästä.

Käyttötapaus 3: Huoltolähetteen tietojen hakeminen.

Vaihe 1: myyjä kirjautuu järjestelmään.

Vaihe 2: myyjä avaa haun.

Vaihe 3: myyjä tekee haun.

Vaihe 4: myyjä tarkastelee haun tuloksia.

Vaihe 5: myyjä palaa takaisin päävalikkoon.

Vaihe 6: myyjä kirjautuu ulos järjestelmästä.

Käyttötapaus 4: Huoltolähetteen statuksen muuttaminen.

Vaihe 1: myyjä kirjautuu järjestelmään.

Vaihe 2: myyjä avaa haun.

Vaihe 3: myyjä hakee muokattavan lähetteen tiedot.

Vaihe 4: myyjä muokkaa lähetteen statusta.

Vaihe 5: myyjä tallentaa lähetteen.

Vaihe 6: myyjä palaa takaisin päävalikkoon.

Vaihe 7: myyjä kirjautuu ulos järjestelmästä.

Käyttötapaus 5: Huoltolähetteen poistaminen.

Vaihe 1: myyjä kirjautuu järjestelmään.

Vaihe 2: myyjä avaa haun.

Vaihe 3: myyjä tekee haun.

Vaihe 4: myyjä poistaa haluamansa lähetteen.

Vaihe 5: myyjä tallentaa tiedot.

Vaihe 6: myyjä palaa takaisin päävalikkoon.

Vaihe 7: myyjä kirjautuu ulos järjestelmästä.

Käyttötapaus 6: Poistetun huoltolähetteen palauttaminen.

Vaihe 1: myyjä kirjautuu järjestelmään.

Vaihe 2: myyjä avaa haun.

Vaihe 3: myyjä tekee haun.

Vaihe 4: myyjä valitsee poistetun huoltolähetteen.

Vaihe 5: myyjä palauttaa lähetteen aktiiviseksi.

Vaihe 6: myyjä palaa takaisin päävalikkoon.

Vaihe 7: myyjä kirjautuu ulos järjestelmästä.

4.5 Käyttöliittymän simulointi

Käyttöliittymän simulointi toteutetaan käyttötapauksien pohjalta hyödyntäen aikaisemmin määriteltyjä käyttäjäpersoonia. Ensiksi tehdään simulointi, jonka tarkoituksena on sijoitella kaikki tarvittavat otsikot, opasteet, painikkeet ja tekstikentät käyttöliittymän eri osiin. Tässä vaiheessa ei vielä kiinnitetä huomiota tekstin muotoiluun tai väreihin. Ennen kuin mietitään värien ja tekstin muotoilua, täytyy käyttöliittymässä olla toiminnallisuuden kannalta kaikki oleelliset elementit. Tästä johtuen komponenttien sommittelu toteutetaan ennen visuaalista ulkoasua.

Käyttöliittymä toteutetaan HTML- koodauksella ajettavaksi suoraan selaimeen. Tehdyt ulkoasun muotoiluratkaisut määritellään erikseen CSS-tiedostoihin. Käyttöliittymän simulointi prosessi kuvataan näyttökuvien, toimintaselosteiden ja käyttäjän toimenpiteitä osoittavien nuolten avulla. Opinnäytetyön luettavuuden kannalta, jokaista hiiren klikkausta ja käyttäjän tekemää tiedonsyöttöä ei esitetä erillisinä näyttökuvina, vaan toiminnot simuloidaan vaihe kerrallaan. Järjestelmän käyttötapaukset alkavat järjestelmään kirjautumisella ja loppuvat ulos kirjautumiseen. Nämä toiminnot simuloidaan ainoastaan ensimmäisessä käyttötapauksessa.

4.5.1 Sommittelu käyttötapauksen pohjalta

Käyttötapaus 1: Uuden huolto lähetteen luominen

Ensimmäisessä käyttötapauksessa Käyttäjäpersoonana Liisa kirjautuu järjestelmään tehdäkseen uuden huoltolähetteen viallisesta porakoneesta. Lopuksi hän tulostaa asiakkaalle oman kappaleen läheteestä, palaa takaisin järjestelmän päävalikkoon ja kirjautuu ulos järjestelmästä.

Kuvio 1. Järjestelmään kirjautuminen

Kirjaudu järjestelmään

The image shows a login form with the following elements:

- Label: **Käyttäjätunnus:** followed by a text input field containing "Liisa".
- Label: **Salasana:** followed by a password input field containing "****".
- Button: **Kirjaudu sisään**.

Annotations in a box on the right point to these elements:

- Arrow pointing to the username field: **Liisa syöttää kenttiin käyttäjätunnuksen ja salasanan.**
- Arrow pointing to the password field: **Kirjautuu järjestelmään painamalla kirjautu sisään painiketta.**
- Arrow pointing to the login button: **Kirjautuu järjestelmään painamalla kirjautu sisään painiketta.**

Järjestelmä vaatii käyttäjältä sisään kirjautumisen. Otsikot, tekstikentät ja painike on sijoitettu vasempaan yläreunaan allekkain normaalin lukusuunnan mukaiseen etenemisjärjestykseen. Käyttäjän ei tarvitse hiirellä siirtää kursoria tekstikenttään syöttääkseen tietoja, koska kursori keskittyy automaattisesti käyttäjätunnus kenttään, johon ensimmäinen syötettävä tieto tulee. Kirjautumissivu ei tarvitse erillisiä opastetekstejä, koska sen toimintalogiikka on varsin yksiselitteinen ja ylimääräiset opastetekstit kiinnittäisivät käyttäjän huomion ja hidastaisivat turhaan järjestelmään kirjautumista.

Kuvio 2. Järjestelmän päävalikko



Päävalikko ja linkit järjestelmän eri osiin on sijoitettu vasempaan reunaan, jonne käyttäjän huomio luonnollisesti keskittyy ensimmäiseksi. Järjestelmään kirjautuessa käyttäjä tietää jo minkä toiminnon hän haluaa suorittaa järjestelmän avulla. Tämän takia on selkeämpää esitellä kaikki linkit suoraan etusivulla kuin esimerkiksi laittaa ne pudotusvalikon alle.

Kuvio 3. Asiakastietojen syöttäminen



Uutta huoltolähetettä luodessa järjestelmään syötetään ensimmäiseksi asiakasta koskevat tiedot. Sivun avautuessa kursori keskittyy ensimmäiseen tekstikenttään.

Kuvio 4. Tuotetietojen hakeminen tietokannasta

Uusi huoltolähete

Asiakastiedot

Sukunimi:	<input type="text" value="Malli"/>	Etinimi:	<input type="text" value="Asiakas"/>
Katuosoite:	<input type="text" value="Valimotie 16"/>	Postinumero:	<input type="text" value="00380"/>
Postitoimipaikka:	<input type="text" value="HELSINKI"/>	Puhelinnumero:	<input type="text" value="05055555"/>
Sähköposti:	<input type="text" value="M_Asiakas@gmail.com"/>		

Tuotetiedot

Syötä huollettavan tuotteen nimi tai koodi allaoleviin tekstikenttiin.

Tuotenimi:	<input type="text"/>	
Tuotekoodi:	<input type="text" value="12345678"/>	<input type="button" value="Hae tietokannasta"/>

Liisa syöttää hakukriteeriksi tuote koodin, jonka hän on katsonut asiakkaan kuitista. Liisa painaa hae tietokannasta painiketta.

Kaikki takuunalaiset laitteet löytyvät tuotetietokannasta. Järjestelmässä haetaan laitteita joko nimen tai koodin perusteella. Opastetekstin käyttö on tässä kohtaa perusteltua, ettei käyttäjän tarvitse jäädä turhaan miettimään hakukriteerien käyttöä. Ilman opastetekstiä saa helposti vaikutelman, että hakukriteeriksi on syötettävä molemmat tiedot. Asiakas- ja tuotetietoja koskevat alueet on erotettu toisistaan poikkiviivalla. Tämä helpottaa käyttäjää hahmottamaan yhteenkuuluvat loogiset kokonaisuudet

Kuvio 5. Hakutulosten näyttäminen

Tuotetiedot

Syötä huollettavan tuotteen nimi tai koodi allaoleviin tekstikenttiin.

Tuotenimi:	<input type="text"/>	
Tuotekoodi:	<input type="text" value="12345678"/>	<input type="button" value="Hae tietokannasta"/>

Haun tulokset:

<input type="text" value="Tuotenimi: Porakone, Tuotekoodi: 12345678"/>
--

Liisa valitsee tuotteen listalta ja painaa valitse tuote näppäintä.

Käyttöliittymässä haetaan huollettava laite ensin tietokannasta listalle, jossa hakutulokset näkyvät. Listalta käyttäjä valitsee oikean laitteen ja vahvistaa valinnan painamalla valitse tuote painiketta. Käyttäjän ei annetta kirjoittaa laitetietoja suoraan huoltolähteeseen. Tällä tavalla vältetään mahdollisuus virheellisten laitetietojen syöttämiseen.

Kuvio 6. Viankuvauksen syöttäminen

Valitse tuote

Valittu tuote: Porakone, tuotekoodi 1234578

Vian Kuvaus:

Poran laturi on viallinen.

Tallenna lähete

Järjestelmä antaa vahvistuksen huollettavasta laitteesta valittu tuote kohtaan. Liusa kirjoittaa vian kuvauksen ja painaa tallenna lähete painiketta.

Käyttäjälle ilmoitetaan tuotteen onnistuneesta valinnasta näyttämällä se tekstillä, joka tulee näkyviin käyttäjän painettua valitse tuote painiketta. Vahvistus näytetään suoraan käyttöliittymässä eikä erillisessä ponnahdusikkunassa, koska se aiheuttaa ylimääräisiä hiiren klikkauksia ja saattaa vangita käyttäjän huomion uuteen avautuvaan ikkunaan.

Kuvio 7. Huoltolähetteen tulostaminen

Huoltolähetteen tiedot

Huoltolähetteen numero: 987654

Status: Lähetetty huoltoon 20.11.2010

Asiakkaan tiedot

Sukunimi: Malli Etunimi: Asiakas
Katuosoite: Valimotie 16 Postinumero ja toimipaikka: 00380 HELSINKI
Puhelinnumero: 05055555 Sähköposti: M_asiakas@gmail.com

Tuotetiedot

Tuotenimi: Porakone
Tuotekoodi: 12345678

Vian seloste

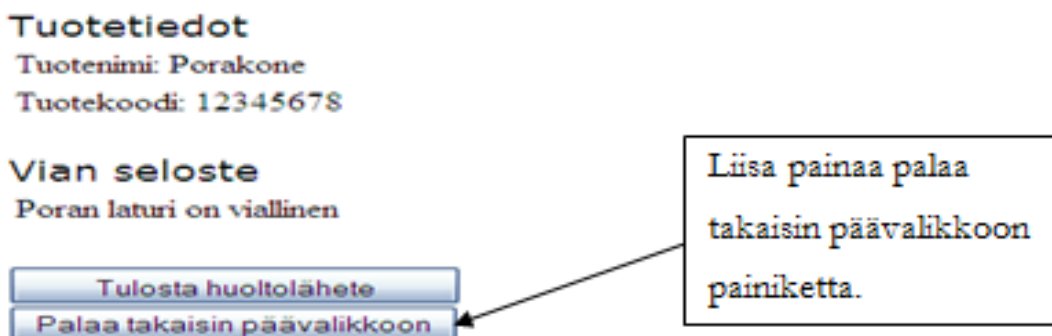
Poran laturi on viallinen

Tulosta huoltolähete

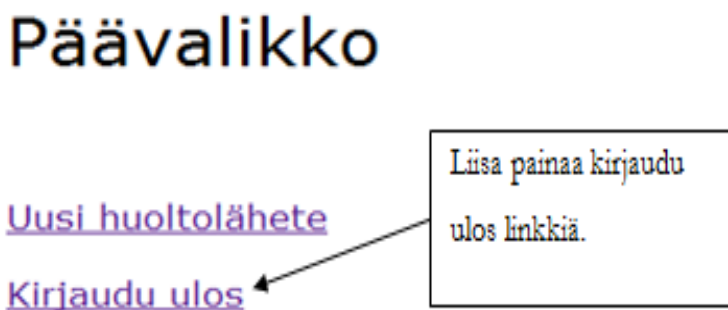
Liusa painaa tulosta lähete painiketta.

Huoltolähetteen tiedot näkymässä on esillä kaikki lähetettä koskeva informaatio. Näkymä on identtinen tietojen syöttämiseen tarkoitettujen näkymien kanssa. Ainoana erona on tekstikenttien puuttuminen. Tiedot näytetään suoraan sivun rungossa, jotta käyttäjä ei erehdy luulemaan yhteenveto näkymää muokkaus näkymäksi ja korjaamaan lähetteen tietoja tähän kohtaan. Tulostusta varten ei aukea enää uutta näkymää vaan järjestelmä lähettää suoraan käytössä olevalle kirjoittimelle käskyn sivun tietojen tulostamisesta. Windowsin omat toiminnot ilmoittavat käyttäjälle tulostuksen onnistumisesta, joten järjestelmän ei tarvitse sitä erikseen tehdä.

Kuvio 8. Päävalikkoon palaaminen



Kuvio 9. Järjestelmästä ulos kirjautuminen



Kuvio 10. Vahvistus ulos kirjautumisesta

Olet kirjautunut ulos järjestelmästä, voit sulkea selaimen!

Selaimesta aukeaa uusi sivu, jossa ilmoitetaan ulos kirjautumisen onnistumisesta ja kehoitetaan sulkemaan selain.

Käyttötapaus 2: Huoltolähetteen tietojen muokkaaminen

Luotuaan uuden huoltolähetteen Liisa huomaa, että on kirjoittanut asiakkaan Sukunimen väärin. Huoltolähetteen tiedot näkymästä hän palaa muokkaamaan tietoja ja tallentaa muokatun lähetteen.

Kuvio 11. Paluu takaisin muokkaamaan lähetteen tietoja

Huoltolähetteen tiedot

Huoltolähetteen numero: 987654

Status: Lähetetty huoltoon 20.11.2010

Asiakkaan tiedot

Sukunimi: Malli Etunimi: Asiakas
Katuosoite: Valimotie 16 Postinumero ja toimipaikka: 00380 HELSINKI
Puhelinnumero: 05055555 Sähköposti: M_asiakas@gmail.com

Tuotetiedot

Tuotenimi: Porakone
Tuotekoodi: 12345678

Vian seloste

Poran laturi on viallinen

[Tulosta huoltolähetteen tiedot](#)
[Palaa takaisin päävalikkoon](#)
[Muokkaa lähetteen tietoja](#)

Liisa painaa muokkaa lähetteen tietoja painiketta.

Kuvio 12. Tietojen muokkaus

Voit muokata asiakastietoja ja vian kuvausta kirjoittamalla suoraan tekstikenttiin. Tuotetietoja voit muokata hakemalla uuden laitteen tiedot kannasta. Paina tallenna lähetteen tietoja painiketta sivun alareunasta tallentaaksesi muutokset.

Asiakastiedot

Sukunimi: Etunimi:
Katuosoite: Postinumero:
Postitoimipaikka: Puhelinnumero:
Sähköposti:

Liisa syöttää asiakkaan sukunimen uudestaan.

Tietojen muokkaus näkymä on samanlainen kuin alkuperäinen näkymä, jossa tietoja syötettiin järjestelmään. Vanhat tiedot säilyvät pohjana, jotta käyttäjän ei tarvitse syöttää niitä enää uudestaan vaan hän voi muuttaa vain haluamansa tiedot. Käyttäjää opastetaan muutosten tekemisessä otsikon alla olevan opastetekstin avulla.

Kuvio 13. Muutosten tallentaminen

Valittu tuote: Porakone, tuotekoodi 1234578

Vian Kurvaus:

Poran laturi on viallinen.

Lüisa painaa Tallenna lähete painiketta ja palaa takaisin huoltolähetteen tiedot näkymään.

Kuvio 14. Muokattu huoltolähete

Huoltolähetteen tiedot

Huoltolähetteen numero: 987654

Status: Lähetetty huoltoon 20.11.2010

Asiakkaan tiedot

Sukunimi: Esimerkki Etunimi: Asiakas
Katuosoite: Valimotie 16 Postinumero ja toimipaikka: 00380 HELSINKI
Puhelinnumero: 05055555 Sähköposti: M_asiakas@gmail.com

Tuotetiedot

Tuotenimi: Porakone
Tuotekoodi: 12345678

Vian seloste

Poran laturi on viallinen

-
-
-

Käyttötapaus 3: Huoltolähteen tietojen hakeminen

Asiakas on tiedustellut käyttäjäpersoona Harrilta viivästyneestä huollosta. Harri avaa asiakkaan huoltolähteen, tarkastaa sen statuksen ja palaa takaisin päävalikkoon.

Kuvio 15. Haku näkymän avaaminen

Päävalikko

[Uusi huoltolähete](#)
[Hae avoimia huoltoja](#)
[Kirjaudu ulos](#)

Harri klikkaa hae avoimia huoltoja linkkiä.

Kuvio 16. Avoimien huoltojen hakunäkymä

Hae avoimia huoltoja

Syötä huoltolähteen numero, tuotekoodi tai asiakkaan sukunimi hakukriteeriksi.

Huoltolähteen numero:

Tuotekoodi:

Sukunimi:

Harri syöttää asiakkaan antaman huoltolähteen numeron hakuehdoksi ja painaa hae lähteitä painiketta.

Hae lähteitä

Avoimien huoltolähteiden haku toiminto on samankaltainen tuotetietojen hakutoiminnon kanssa. Käyttäjää opastetaan valitsemaan käytettävä hakukriteeri opastetekstillä. Kahta huoltolähetettä ei voi olla samalla numerolla, joten huoltolähteen numeroa kannattaa käyttää ensisijaisena hakuehtona, jos se vain on saatavilla. Käyttäjää ohjataan tähän keskittämällä kursori huoltolähteen numero tekstikenttään.

Kuvio 17. Huoltolähetteen valitseminen listalta

Hae avoimia huoltoja

Syötä huoltolähetteen numero, tuotekoodi tai asiakkaan sukunimi hakukriteeriksi.

Huoltolähetteen numero:

Tuotekoodi:

Sukunimi:

Hann tulokset

Huoltolähetteen numero: 987654, Sukunimi: Malli, Etunimi: Asiakas, Tuotenimi: Porakone

Harri valitsee huoltolähetteen ja painaa näytä tiedot painiketta.

Käyttäjä valitsee oikean huoltolähetteen listalta ja painaa näytä tiedot painiketta. Näkymässä käytetään normaalia leveämpää listaa hakutulosten selkeän esittämisen vuoksi. Mikäli hakutuloksia on useita ja yhden lähetteen tiedot eivät mahdu yhdelle riville, tulee listasta helposti vaikea lukuinen.

Kuvio 18. Huoltolähetteen tiedot

Huoltolähetteen numero: 987654

Status: Lähetetty huoltoon 20.11.2010

Asiakkaan tiedot

Sukunimi: Malli Etunimi: Asiakas
Katuosoite: Valimotie 16 Postinumero ja toimipaikka: 00380 HELSINKI
Puhelinnumero: 05055555 Sähköposti: M_asiakas@gmail.com

Tuotetiedot

Tuotenimi: Porakone
Tuotekoodi: 12345678

Vian seloste

Poran laturi on viallinen

Harri tarkastaa huollon statuksen ja palaa takaisin päävalikkoon.

Käyttötapaus 4: Huoltolähetteen statuksen muuttaminen

Käyttäjäpersoona Jenni on työvuoronsa alussa vastaanottanut huollosta palautuneen laitteen. Hän ilmoittaa asiakkaalle puhelimitse laitteen olevan noudettavissa ja muuttaa järjestelmään huoltolähetteen statukseksi ”palautunut huollosta, odottaa noutoa”.

Kuvio 19. Siirtyminen päävalikosta hakutoimintoon

Päävalikko

Tervetuloa, huoltokoneiden hallintajärjestelmään!

Voit luoda uuden huoltolähetteen uusi huoltolähetete linkistä.

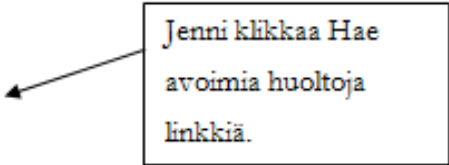
Mikäli haluat muokata vanhan huoltolähetteen tietoja klikkaa hae avoimia huoltoja linkkiä.

[Uusi huoltolähetete](#)

[Hae avoimia huoltoja](#)

[Kirjaudu ulos](#)

Jenni klikkaa Hae avoimia huoltoja linkkiä.



Päävalikkoon on lisätty opasteteksti selventämään linkkien toimintaa. Opastetekstin avulla käyttäjälle kerrotaan, että kaikki lähetteen muokkaamiseen liittyvät toiminnot ovat hae avoimia huoltoja linkin alla, mikäli käyttäjä avaa järjestelmän aikoakseen poistaa vanhan huoltolähteen hän saattaa jäädä ihmettelemään poistotoiminnon puuttumista päävalikosta.

Kuvio 20. Avoimien huoltojen hakunäkymä

Hae avoimia huoltoja

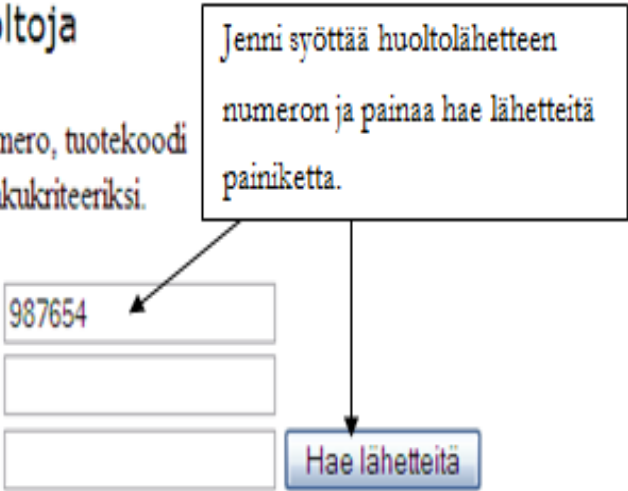
Syötä huoltolähetteen numero, tuotekoodi tai asiakkaan sukunimi hakukriteeriksi.

Huoltolähetteen numero:

Tuotekoodi:

Sukunimi:

Jenni syöttää huoltolähetteen numeron ja painaa hae lähetteitä painiketta.



Hae lähetteitä

Kuvio 21. Hakutulosten näyttäminen

Huoltolähteen numero:

Tuotekoodi:

Sukunimi:

Hain tulokset

Huoltolähteen numero: 987654, Sukunimi: Malli, Etunimi: Asiakas, Tuotenimi: Porakone

Jenni valitsee listalta huoltolähteen ja painaa näytä tiedot painiketta.

Kuvio 22. Statuksen muokkaaminen

Huoltolähteen tiedot

Huoltolähteen numero: 987654

Status: Lähetetty huoltoon 20.11.2010 [Vaihda statusta](#)

Asiakkaan tiedot

Sukunimi: Malli Etunimi: Asiakas
Katuosoite: Valimotie 16 Postinumero ja toimipaikka: 00380 HELSINKI
Puhelinnumero: 05055555 Sähköposti: M_asiakas@gmail.com

Tuotetiedot

Tuotenimi: Porakone
Tuotekoodi: 12345678

Vian seloste

Poran laturi on viallinen

Jenni klikkaa Vaihda status-
ta linkkiä.

Kuvio 23. Statuksen vaihtaminen

Valitse huoltolähetteen status

Lähetetty huoltoon
 Palannut huollosta, odottaa noutoa
 Valmis, noudettu

Vahvista

Jenni valitsee statuksen ja painaa vahvista painiketta.

Vaihda statusta linkki avaa ponnahdusikkunan, josta huoltolähetteen status valitaan. Vahvista painike sulkee ponnahdusikkunan ja status päivittyy huoltolähetteeseen. Statuksen vaihtaminen on sijoitettu erillisen ponnahdusikkunan taakse, koska on haluttu säilyttää selkeä looginen kokonaisuus. Kaikki statukseen liittyvä tieto ja sen muokkaustoiminnot ovat samassa kohdassa.

Mikäli toiminto olisi suoraan huoltolähetteen tiedot näkymässä vaikka valmiina listana ja valintapainikkeena tulisi näkymästä helposti sekava. Vaihtaessaan huoltolähetteen statusta käyttäjä ei varsinaisesti muokkaa lähetteen tietoja, vaan muuttaa ainoastaan sen tilaa. Tämän takia statuksen vaihtaminen ja tietojen muokkaustoiminto on eroteltu toisistaan.

Kuvio 24. Muokattu status

Huoltolähetteen numero: 987654

Status: Palannut huollosta, odottaa noutoa 20.11.2010 [Vaihda statusta](#)

Asiakkaan tiedot

Sukunimi: Malli Etunimi: Asiakas
Katuosoite: Valimotie 16 Postinumero ja toimipaikka: 00380 HELSINKI
Puhelinnumero: 05055555 Sähköposti: M_asiakas@gmail.com

Tuotetiedot

Tuotenimi: Porakone
Tuotekoodi: 12345678

Vian seloste

Poran laturi on viallinen

Tulosta huoltolähette
Palaa takaisin päävalikkoon
Muokkaa lähetteen tietoja

Status päivittyy huoltolähetteen tiedot näkymään. Päivämäärä päivittyy automaattisesti aina kun lähete saa uuden statuksen.

Käyttötapaus 5: Huoltolähetteen poistaminen

Käyttäjäpersoonana Matti kirjautuu järjestelmään poistaakseen turhaksi osottautuneen huoltolähetteen. Matti hakee poistettavan huoltolähetteen tiedot edellisen samalla tavalla kuin edellisessä käyttötapauksessa.

Kuvio 25. Huoltolähetteen poistaminen

Huoltolähetteen tiedot

Huoltolähetteen numero: 987654

Status: Lähetetty huoltoon 20.11.2010 [vaihda statusta / Poista lähete](#)

Asiakkaan tiedot

Sukunimi: Malli Etunimi: Asiakas
Katuosoite: Valimotie 16 Postinumero ja toimipaikka: 00380 HELSINKI
Puhelinnumero: 05055555 Sähköposti: M_asiakas@gmail.com

Tuotetiedot

Tuotenimi: Porakone
Tuotekoodi: 12345678

Vian seloste

Poran laturi on viallinen

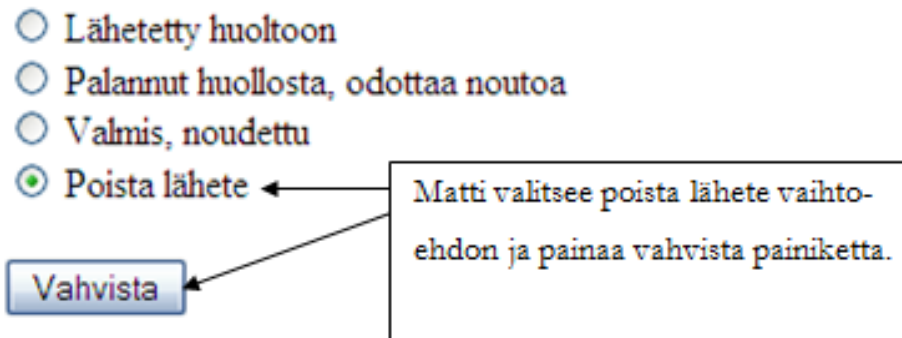
[Tulosta huoltolähete](#)
[Palaa takaisin päävalikkoon](#)
[Muokkaa lähetteen tietoja](#)

Matti klikkaa vaihda statusta
/ Poista lähete linkkiä.

Poisto toiminto on sijoitettu statuksen vaihtamisen kanssa saman linkin alle. Linkin nimeä on muokattu, jotta se opastaisi käyttäjää paremmin.

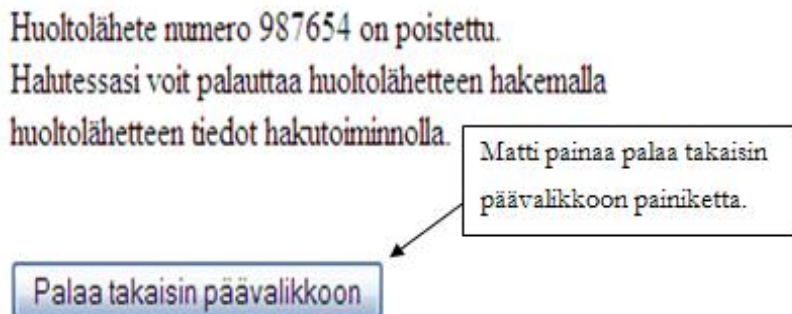
Kuvio 26. Poistamistoiminnon valinta

Valitse huoltolähteen status



Käyttäjältä ei kysytä enää varmistusta lähetteen poistamiseksi. Poisto toimintoa suorittaessaan hän on jo päättänyt poistaa lähetteen, joten varmistuksen kysyminen on vain turhaa klikkailua. Käytettävyyden kannalta on järkevämpää tarjota ratkaisua, jonka avulla käyttäjä pystyy myöhemmin halutessaan palauttamaan poistetut tiedot.

Kuvio 27. Vahvistus lähetteen poistamisesta



Käyttäjälle näytetään vahvistus huoltolähteen poistamisesta. Opasteteksti kertoo käyttäjälle miten poistetun lähetteen voi palauttaa.

Käyttötapaus 6: Poistetun huoltolähetteen palauttaminen

Käyttäjäpersoonana Matti on vahingossa poistanut väärän huoltolähetteen. Laite on lähetetty huoltoon muutama päivä aikaisemmin kuin lähete on poistettu. Matti palauttaa lähetteen aktiiviseksi ja muokkaa huoltolähetteen statuksen päivämäärän oikeaksi.

Kuvio 28. Hakukriteerin valinta

Hae avoimia huoltoja

Syötä huoltolähetteen numero, tuotekoodi tai asiakkaan sukunimi hakukriteeriksi.

Näytä tulokset:
 Vain avoimet Kaikki

Huoltolähetteen numero:

Tuotekoodi:

Sukunimi:

Haun tulokset:

Matti valitsee näytettäväksi kaikki huoltolähteet. Matti painaa hae lähetteitä painiketta.

Haku toiminnossa käyttäjä voi valita näytetäänkö kaikki huoltolähteet vai ainoastaan avoimena olevat. Avoimien läheteiden näyttäminen on valittu oletusarvoksi, koska hakutoimintoa käytetään useinmiten avoimien läheteiden hakemiseen. Mikäli käyttäjä ohittaa kohdan ja tekee haun niin hakutuloksina ei näytetä poistettuja tai valmiita huoltolähteitä.

Kuvio 29. Poistetun lähetteen palauttaminen

Huoltolähetteen tiedot

Huoltolähetteen numero: 987654

Status: Poistettu 20.11.2010 [Palauta lähete](#)

Asiakkaan tiedot
Sukunimi: Malli Etunimi: Asiakas
Katuosoite: Valimotie 16 Postinumero ja toimipaikka: 00380 HELSINKI
Puhelinnumero: 05055555 Sähköposti: M_asiakas@gmail.com

Tuotetiedot
Tuotenimi: Porakone
Tuotekoodi: 12345678

Vian seloste
Poran laturi on viallinen

[Tulosta huoltolähete](#)
[Palaa takaisin päävalikkoon](#)
[Muokkaa lähetteen tietoja](#)

Matti painaa palauta lähete linkkiä.

Huoltolähetteen tiedot näkymässä statuksen vaihto linkki muuttuu automaattisesti palauta lähete linkiksi, mikäli huoltolähetteen statuksena on poistettu.

Kuvio 30. Statuksen ja päivämäärän valinta

Valitse huoltolähetteen status

Lähetetty huoltoon
 Palannut huollosta, odottaa noutoa
 Valmis, noudettu
 Poista lähete

Valitse päivämäärä

2010/12/13

<--	<	December 2010				>	-->
28	29	30	1	2	3	4	
5	6	7	8	9	10	11	
12	13	14	15	16	17	18	
19	20	21	22	23	24	25	
26	27	28	29	30	31	1	
2	3	4	5	6	7	8	

[Vahvista](#)

Matti valitsee oikean statuksen palautettavalle läheteelle ja päivämäärän jolloin laite alun perin lähetettiin huoltoon. Matti painaa vahvista painiketta.

Kuten käyttötapauksesta selviää huoltolähetteen palauttamisessa tai statuksen muuttamisessa voi ilmetä tarvetta muuttaa myös päivämäärää. Päivämäärän valitsemiseen käyttöliittymään on liitetty kalenteri, josta käyttäjä voi suoraan valita oikean päivämäärän. Tällä ratkaisulla yritetään minimoida riski väärän päivämäärä tiedon syöttämiselle. Mikäli käyttäjän pitäisi muistaa oikea päivämäärä ja kirjoittaa se itse tekstikenttään, olisi virheen teko mahdollisuus paljon suurempi. Kaikissa statuksen vaihtoon liittyvissä toiminnoissa ei tarvitse muuttaa päivämäärää ollenkaan. Tästä johtuen kalenterin oletusarvoksi on valittu nykyhetki.

Kuvio 31. Vahvistus huoltolähetteen palauttamisesta

Huoltolähete numero 987654 on palautettu aktiiviseksi.



4.5.2 Visuaalinen ulkoasu

Käyttöliittymässä käytetään kolmea erilaista fonttia ja fonttikokoa. Pääotsikoille, alaotsikoille ja normaalille tekstile on omat fonttinsa. Toisistaan eroavien fonttien avulla käyttäjän on helpompaa hahmottaa kuuluuko teksti runkotekstiin vai otsikoihin.

Kuvio 32. Käyttöliittymän fontit

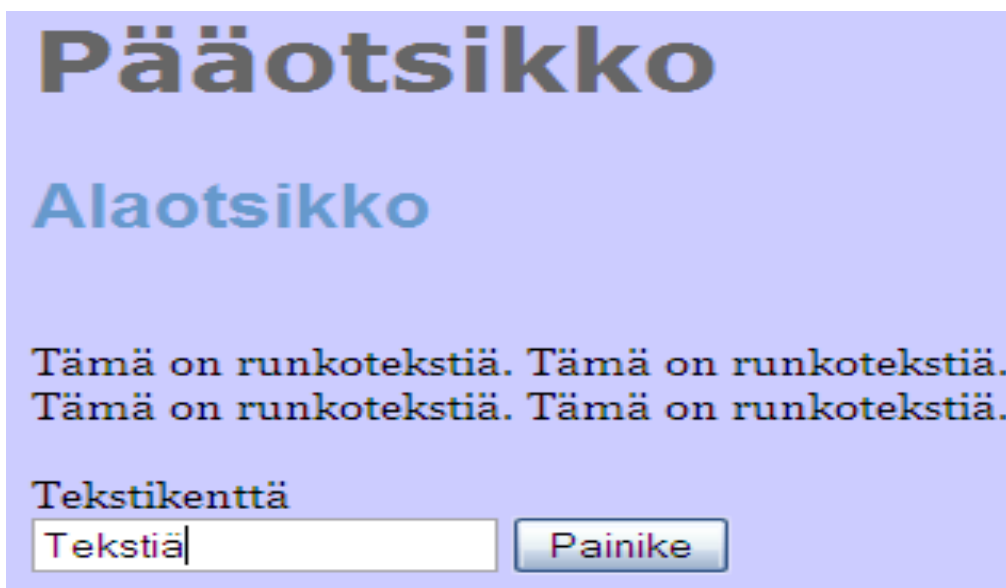
Pääotsikko

Alaotsikko

Tämä on runkotekstiä. Tämä on runkotekstiä.
Tämä on runkotekstiä. Tämä on runkotekstiä.

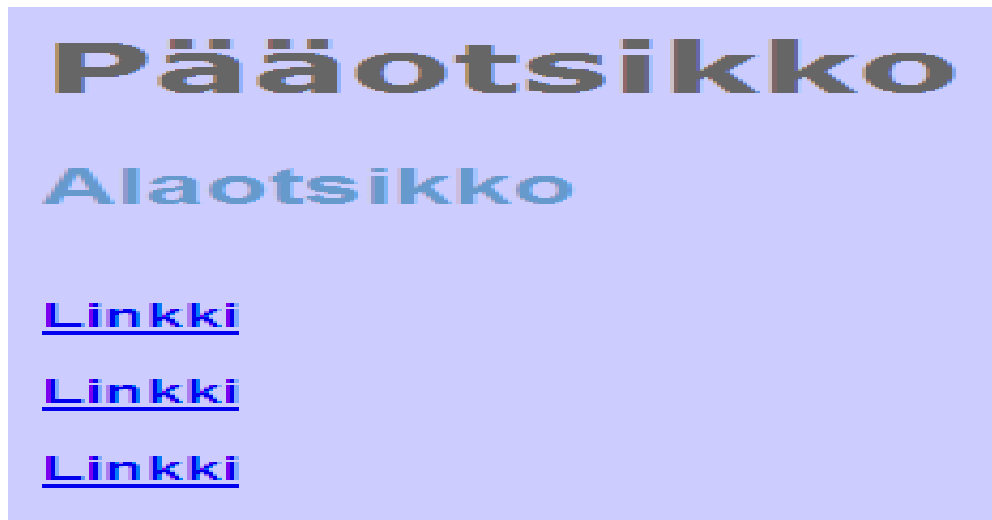
Runkoteksteissä käytetään Georgia fonttia. Fontti on selkeä lukuinen ja se on eritoten suunniteltu ruudulta luettavaksi. Pääotsikoissa käytetään Verdana fonttia, joka toimii hyvin myös suuremmalla koolla. Alaotsikoissa käytettävä fontti on Helvetica. Se eroaa Georgiasta tarpeeksi selkeästi, jotta sitä ei sotketa runkotekstiin. Helvetica muistuttaa hieman pääotsikoissa käytettävää Verdana fonttia, mutta ei kuitenkaan sotkeudu siihen. Tästä johtuen myös alaotsikot on helpompi mieltää kuuluvaksi otsikkoryhmään. Kaikki käytettävät fontit ovat web-ystävällisiä ja ne toimivat kaikissa yleisissä selaimissa ja käyttöympäristöissä. Käytettävät fonttikoot ovat: Pääotsikot 35 px, alaotsikot 25 px ja runkoteksti 15 px.

Kuvio 33. Käyttöliittymän värit



Väreillä pyritään luomaan helposti luettava kokonaisuus joka ei rasita silmiä. Käyttöliittymään on valittu saman sävvyisiä värejä, jotka kuitenkin erottuvat toisistaan. Kaikissa käyttöliittymän osissa käytetään yhtenäistä värimaailmaa.

Kuvio 34. Päävalikon linkkien tyyli

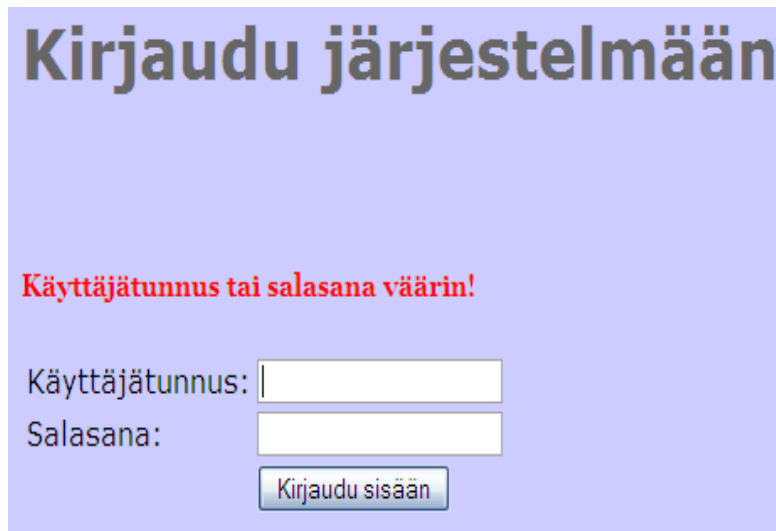


Linkeissä säilytetään perinteinen internetsivuilla paljon käytetty tyyli, jossa linkin väri on sininen ja se on alleviivattu. Tämä tyyli on todella laajasti käytetty ja varmasti tuttu monelle hie-
man vähemmänkin internetiä käyttäneelle henkilölle.

4.5.3 Virheilmoitukset

Käytettävissä virheilmoituksissa on yhtenäinen logiikka kaikissa käyttöliittymän osissa. Käyttöliittymän virheilmoitukset näytetään suoraan käyttöliittymän osassa, jossa virhe tapahtuu. Kaikki virheilmoitukset ovat tekstimuodossa ja ne on korostettu muusta tekstistä punaisella fontilla. Virheilmoitusten tavoitteena on olla mahdollisimman selkeitä ja informatiivisia, joten käyttöliittymässä on päädytty käyttämään pelkästään tekstimuotoisia virheilmoituksia korostamatta niitä symboleilla tai kuvilla. Virheilmoituksilla pyritään paitsi kertomaan käyttäjälle tapahtuneesta virheestä, myös opastamaan virhetilanteesta eteenpäin. Seuraavassa on näyttökuvain kuvattu muutamia järjestelmän yleisimmistä virhetilanteista ja virheilmoituksista.

Kuvio 35. Virhe sisäänkirjautumisessa



Kirjaudu järjestelmään

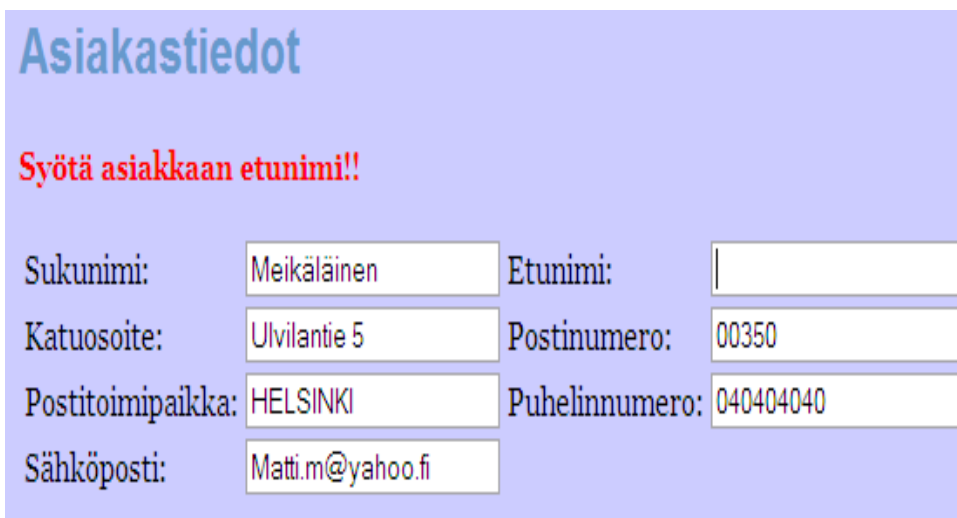
Käyttäjätunnus tai salasana väärin!

Käyttäjätunnus:

Salasana:

Mikäli käyttäjä syöttää salasanan tai käyttäjätunnuksen väärin käyttöliittymä antaa virheilmoituksen tekstikenttien yläpuolella ja kursori keskittyy automaattisesti käyttäjätunnus kohtaan. Virheen syy on niin yksiselitteinen, että virheilmoituksessa ei tarvita erillisiä ohjeita virheestä selviämistä varten.

Kuvio 36. Puutteelliset asiakastiedot



Asiakastiedot

Syötä asiakkaan etunimi!!

Sukunimi:	<input type="text" value="Meikäläinen"/>	Etunimi:	<input type="text"/>
Katuosoite:	<input type="text" value="Ulvilantie 5"/>	Postinumero:	<input type="text" value="00350"/>
Postitoimipaikka:	<input type="text" value="HELSINKI"/>	Puhelinnumero:	<input type="text" value="040404040"/>
Sähköposti:	<input type="text" value="Matti.m@yahoo.fi"/>		

Mikäli käyttäjä on jättänyt jonkun pakollisen tiedon syöttämättä ja yrittää tallentaa huoltolähetettä, järjestelmä antaa virheilmoituksen puuttuvasta tiedosta ja keskittää kursorin automaattisesti oikeaan tekstikenttään

Kuvio 37. Virheellinen hakukriteeri



Tuotetiedot

Syötä huollettavan tuotteen nimi tai koodi allaoleviin tekstikenttiin.

Tuotekoodi on virheellinen!! Tarkista koodi tai hae laitetta tuotenimen perusteella.

Tuotenimi:

Tuotekoodi:

Käyttäjän syöttäessä virheellisen hakukriteerin järjestelmä antaa virheilmoituksen hakutoiminnon yläpuolelle ja ohjaa kursorin automaattisesti tekstikenttään, johon virheellinen tieto on syötetty. Virheilmoituksessa opastetaan käyttäjää ensin tarkistamaan hakukriteerin oikeellisuus tai käyttämään vaihtoehtoisia hakukriteeriä.

5 Pohdinta

5.1 Yhteenveto ja johtopäätökset

Opinnäytetyössä suunniteltiin rautakaupalle huoltokoneiden hallintaan tarkoitettuun tietojärjestelmään selainpohjainen käyttöliittymä. Käyttöliittymän tärkeimpinä tavoitteina olivat helppo opittavuus ja toimintalogiikka, joka ohjaa käyttäjää virheettömään käyttöön.

Opinnäytetyössä lähdettiin liikkeelle loppukäyttäjien profiloinnista ja käyttötapausten suunnittelusta. Käyttötapausten varten luotiin fiktiivisiä käyttäjäpersoonia, jotka vastaavat mahdollisimman hyvin järjestelmän loppukäyttäjiiä. Käyttötapaukset suunniteltiin järjestelmän toiminnallisen määrittelyn pohjalta. Käyttötapaukset suunniteltiin sellaisiksi, että kaikki järjestelmällä tehtävät toimenpiteet tulivat katetuksi. Suunnitteluvaiheessa huomattiin, että oikeiden käyttötapausten suunnittelu on varsin tehokas tapa hahmottaa järjestelmän toimintalogiikkaa ja visioita käyttöliittymän ulkoasusta. Käyttäjäpersoonien käyttäminen ei ole hyvän käyttöliittymän suunnittelun kannalta välttämättä pakollista, mutta se antaa hieman laajampaa perspektiiviä suunnittelutyöhön.

Käyttötapausten suunnittelussa ja simuloinnissa on helpompaa ajatella toteutuksen toimivuutta käyttäjäpersoonien kautta. Pystyykö näillä ominaisuuksilla varustettu henkilö suoriutumaan tällaisesta tehtävästä ja miten hän mahdollisesti reagoi, jos jokin menee pieleen? Loppukäyttäjäprofiloinnissa ja käyttötapausten suunnittelussa onnistuttiin luomaan järjestelmän toiminnallista määrittelyä vastaavat ratkaisut.

Käyttöliittymän simulointi tehtiin vaihe kerrallaan käyttötapausten pohjalta suoraan käyttöliittymään eli selaimen. Tähän ratkaisuun päädyttiin, koska kyseessä on aika kevyt ja toimintalogikaltaan selkeä järjestelmä. Tästä johtuen käyttöliittymä oli helposti hahmoteltavissa suoraan selaimen. Paperille piirretyillä käyttöliittymäsuunnitelmilla ei tässä tapauksessa olisi saavutettu merkittävää hyötyä. Simuloinnin ja käyttötapausten avulla onnistuttiin rakentamaan käyttöliittymä, jossa kaikki käyttöliittymän komponentit on luotu palvelemaan jotain tarkoitusta, eikä käyttöliittymässä ole turhia elementtejä.

Opinnäytetyössä suunniteltiin pelkästään käyttöliittymä eikä toteutettu koko tietojärjestelmää. Tästä seikasta ja järjestelmän keveydestä johtuen käytettävyydestä ei käsitelty tässä opinnäytetyössä. Käytettävyydesti olisi ollut hankala järjestää ilman valmista toimivaa järjestelmää ja pelkillä käyttöliittymä näkymillä tehtynä siitä saatava hyöty olisi kärsinyt liikaa.

Opinnäytetyössä näytettiin, että käyttötapausten pohjalta tehdyllä simulointipohjaisella käyttöliittymäsuunnittelulla pystytään toteuttamaan selkeitä ja käytettävyydeltään hyviä käyttöliittymiä. Huolellisesti simuloitu käyttöliittymä muodostuu käyttötarkoitusta vastaavaksi ja varsinkin pienten järjestelmien kanssa käytettävyydestä tuskien saavutetaan merkittäviä tuloksia.

Opinnäytetyössä onnistuttiin luomaan käyttöliittymä joka noudattaa ulkoasultaan ja toimintalogikaltaan yhtenäistä kaavaa kaikissa käyttöliittymän osissa. Tämä tekee käyttöliittymästä helposti opittavan. Käyttöliittymässä on käytetty opastetekstejä, virheilmoituksia ja yhtenäisiä loogisia kokonaisuuksia, jotka ohjaavat käyttäjää järjestelmän oikeanlaiseen käyttöön ja auttaa selviämään mahdollisista virhetilanteista.

5.2 Jatkokehitysmahdollisuudet ja tulevaisuuden näkymät

Selainpohjaiset käyttöliittymät tuntuvat olevan yleinen trendi nykypäivän ohjelmistosuunnittelussa. Joillain aloilla ne ovat ja standardeja.

Kirjoittajan mielestä selainpohjaisten käyttöliittymien suunnittelu ei kuitenkaan vastaa käyttäjien tarpeita useissa tapauksissa. Joidenkin tietojärjestelmien kohdalla ei ole otettu kunnolla huomioon selainpohjaisen ympäristön tuomia mahdollisuuksia ja rajoituksia, joten käyttöliittymät ovat puutteellisia ja toiminnot kömpelöitä. Toisessa ääripäässä suunnittelussa on lähetty toteuttamaan näyttävää ulkoasua ja käytettävyys on jäänyt taka-alalle. Kummassakin tapauksessa järjestelmän käytettävyys kärsii ja tietojärjestelmä ei ole paras mahdollinen työkalu käyttötarkoitukseensa.

Kirjoittajan näkökulmasta tulevaisuuden näkymät käytettävyysuunnittelulle vakavasti otettavana ohjelmistosuunnitteluprosessin osana ovat hyvät. Mobiilikäytön yleistyessä käytettävyyteen on pakon edestä panostettava enemmän. Järjestelmien on taivuttava matkapuhelimien näytöille sulavasti ilman pitkiä latausaikoja. Pienet näytöt ja perinteisestä näppäimistöä ja hiirestä poikkeavat käyttövälineet asettavat mielenkiintoisia haasteita käyttöliittymäsuunnittelulle.

Opinäytetyössä toteutettua huoltokoneiden hallintajärjestelmää voisi tulevaisuudessa kehittää mobiilikäyttöön paremmin sopivaksi. Yhdenmukaisen käyttöliittymän konseptia voisi myös laajentaa ja kehittää järjestelmää kattamaan muitakin yrityksen toimintoja. Parhaassa tapauksessa saavutettaisiin samat käytettävyyden tunnusmerkit täyttävä, kaiken toiminnan kattava toiminnanohjausjärjestelmä. Tällaisessa tilanteessa saavutettaisiin merkittäviä etuja esimerkiksi henkilökunnan kouluttamisessa.

Lähteet

Aitken, P. 2000. Internet-ohjelmointi Visual Basic. IT press. Helsinki.

Collison, S. 2006. Beginning CSS Web development – From novice to professional. Apress. New York.

Guide – Käyttöliittymäsuunnittelu. Luettavissa: <http://www.interacta.fi/suunnitteluvaihe.html>
Luettu 14.12.2010.

ISO 9241-11. Ergonomic requirements for office work with visual display terminals (VDTs). 1998. part 11.

Kallio, T. 1993. Käyttöliittymät. Suomen Atk-kustannus Oy. Espoo.

Krug, S. 2006. Älä pakota minua ajattelemaan. 2.painos. Readme.fi. Helsinki.

Kuutti, W. 2003. Käytettävyys, suunnittelu ja arviointi. Talentum. Helsinki.


Pohjonen, R. 2002. Tietojärjestelmien kehittäminen. Docendo Finland Oy. Jyväskylä.

Sinkkonen, I., Kuoppala, H., Parkkinen, J., Vastamäki, R. 2006. Psychology of usability. Edita Publishing oy. Helsinki.

Matti Vuori, Jouni Kivistö-Rahnasto & Sirra Toivonen, VTT valmistustekniikka. Hyvä käyttöliittymäsuunnittelu lähtee käytön tarpeista. Luettavissa:
<http://www.vtt.fi/inf/julkaisut/muut/1998/av7-98.pdf>. Luettu 14.9.2010

Liitteet

Liite 1. Järjestelmään kirjautumis
näkymä



The image shows a login interface with a light blue header bar containing the text 'Kirjaudu sisään' and a small square icon. The main area has a light purple background with the title 'Kirjaudu järjestelmään' in large, bold, dark grey font. Below the title are two white input fields: the first is labeled 'Käyttäjätunnus:' and the second is labeled 'Salasana:'. A blue button with a gradient and the text 'Kirjaudu sisään' is positioned below the password field.

Liite 2. Järjestelmään kirjautumis
näytön lähdekoodi

```
<html>
<head>
<title>Kirjaudu sisään</title>
<style type="text/css">
  <!--
    body { font-family: Georgia;
           color: #000000;font-size:15px;
           background-color: #CCCCFF;
           }
    h1 { font-family: Verdana;
         color: #666666; font-size:35px;
         }
    h2 { font-family: Helvetica;
         color: #6699CC; font-size:25px;
         }
    h3 {
         font-family: Georgia;
         color: #FF0000;font-size:15px;
         }
  -->
</style>

</head>
<body onload="MyForm.ktunnus.focus()">
<h1> kirjaudu järjestelmään</h1>
<br>
<br>
<br>

<form method="LINK" action="paavalikko.html" name="MyForm">
<table border="0">

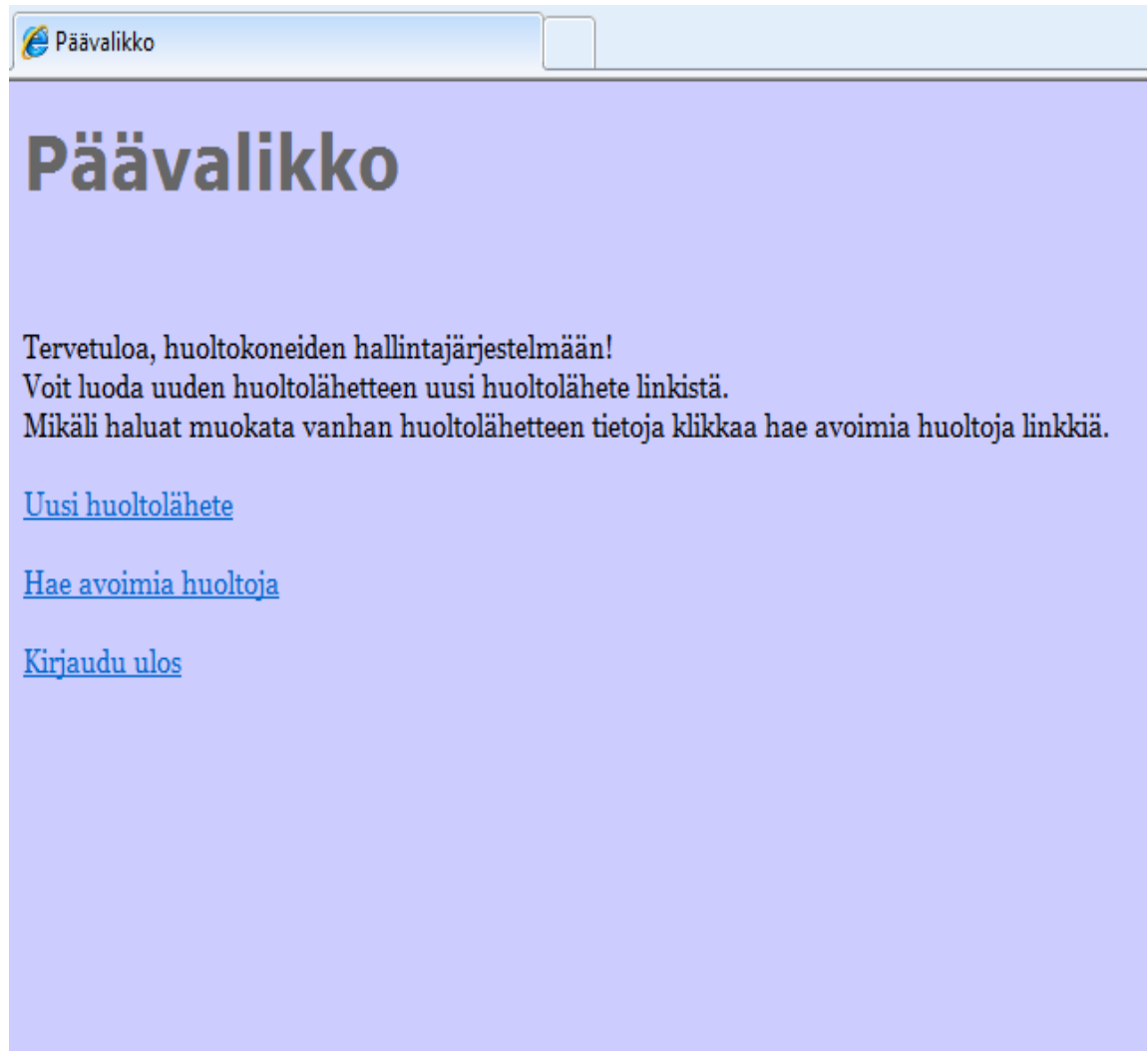
<tr>
<td>käyttäjätunnus:</td>
<td><input type="text" name="ktunnus"/></td>
</tr>

<tr>
<td>salasana:</td>
<td><input type="text" name="salasana"/></td>
</tr>

<tr>
<td></td>
<td> <INPUT TYPE="submit" VALUE="Kirjaudu sisään"></td>
</tr>

</form>
</table>
```


Liite 3. Järjestelmän päävalikko näkymä



Liite 4. Järjestelmän päävalikko
näkökuvan lähdekoodi

```
<html>
<head>
<title>Päävalikko</title>
<style type="text/css">
  <!--
    body { font-family: Georgia;
           color: #000000;font-size:15px;
           background-color: #CCCCFF;
           }

    h1 { font-family: verdana;
         color: #666666; font-size:35px;
         }

    h2 { font-family: Helvetica;
         color: #6699CC; font-size:25px;
         }

    h4 { border: 2px solid red;
         background-color: beige;
         padding: 5px;
         text-align: center;
         }

    h5 {
         font-family: Georgia;
         color: #FF0000;font-size:15px;
         }

    ul { list-style: upper-roman inside; }
  -->
</style>
</head>
<body>
<h1> Päävalikko </h1>
<br>
<br>
Tervetuloa, huoltokoneiden hallintajärjestelmään! <br>
voit luoda uuden huoltolähetteen uusi huoltolähete linkistä. <br>
Mikäli haluat muokata vanhan huoltolähetteen
tietoja klikkaa hae avoimia huoltoja linkkiä
<br>
<br>
<a href="uusihuolto.html">Uusi huoltolähete</a>
<br>
<br>
<a href="haku.html">Hae avoimia huoltoja</a>
<br>
<br>
<a href="logoutvahvistus.html">kirjaudu ulos</a>

</body>
</html>
```

Uusi huoltolähetä

Uusi huoltolähetä

Asiakastiedot

Sukunimi: Etunimi:

Katuosoite: Postinumero:

Postitoimipaikka: Puhelinnumero:

Sähköposti:

Tuotetiedot

Syötä huollettavan tuotteen nimi tai koodi allaoleviin tekstikenttiin.

Tuotenimi:

Tuotekoodi:

Haun tulokset:

Valittu tuote:

Vian Kuvaus:

Liite 6. Huoltolähetteen lisäys
näkymän lähdekoodi

```
<html>
<head>
<title>uusi huoltolähetete</title>

<style type="text/css">
  <!--
    body { font-family: Georgia;
           color: #000000;font-size:15px;
           background-color: #CCCCFF;

           }

    h1 { font-family: verdana;
         color: #666666; font-size:35px;

         }

    h2 { font-family: Helvetica;
         color: #6699CC; font-size:25px;

         }

    h3 {
         font-family: Georgia;
         color: #FF0000;font-size:15px;
         }
  -->
</style>
</head>

<body>

<h1> uusi huoltolähetete</h1>
<h2> Asiakastiedot</h2>

<form method="LINK" action="tuotetiedot.html" name="MyForm">
<br>
<table border="0">

<tr>
<td>Sukunimi:</td>
<td><input type="text" name="snimi" /></td>
<td>Etunimi:</td>
<td><input type="text" name="Enimi" /></td>
</tr>

<tr>
<td>katuosoite:</td>
<td><input type="text" name="kosoite" /></td>
<td>Postinumero:</td>
<td><input type="text" name="ponumero" /></td>
</tr>

<tr>
<td>Postitoimipaikka:</td>
<td><input type="text" name="potoimipaikka" /></td>
<td>Puhelinnumero:</td>
<td><input type="text" name="punumero" /></td>
</tr>

<tr>
<td>sähköposti:</td>
<td><input type="text" name="sähköposti" /></td>
<td></td>
<td></td>
</tr>
</table>
```

```

<hr>
<h2> Tuotetiedot</h2>
Syötä huollettavan tuotteen nimi tai koodi<br>
allaoleviin tekstikenttiin.
<br>
<br>
<table border="0">
<tr>
<td>Tuotenimi:</td>
<td><input type="text" name="Tnimi"/></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuotekoodi:</td>
<td><input type="text" name="Tkoodi"/></td>
<td><INPUT TYPE="submit" VALUE="Hae tietokannasta"></td>
<td></td>
</tr>
</table>

<table border="0">
<br>
<tr>
<td>Haun tulokset:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><textarea rows="4" cols="40">
</textarea></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><INPUT TYPE="submit" VALUE="Valitse tuote"></td>
<td></td>
<td></td>
<td></td>
</tr>
</table>

<hr>
<table border="0">
<tr>
<td>valittu tuote: </td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vian kuvaus:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><textarea rows="4" cols="40">
</textarea></td>
<td></td>
<td></td>
<td></td>
</tr>
</table>
</table>
<br>
<INPUT TYPE="submit" VALUE="Tallenna lähete">

</form>
</body>
</html>

```

 Lähetteen tiedot

Huoltolähetteen tiedot

Huoltolähetteen numero: 987654

Status: Lähetetty huoltoon 20.11.2010 [Vaihda statusta](#) / [Poista lähete](#)

Asiakkaan tiedot

Sukunimi: Malli Etunimi: Asiakas
Katuosoite: Valimotie 16 Postinumero ja toimipaikka: 00380 HELSINKI
Puhelinnumero: 05055555 Sähköposti: M_asiakas@gmail.com

Tuotetiedot

Tuotenimi: Porakone
Tuotekoodi: 12345678

Vian seloste

Poran laturi on viallinen

Liite 8. Huoltolähetteen tiedot
näkyvän lähdekoodi

```
<html>
<head>
<title>Lähetteen tiedot</title>

<SCRIPT LANGUAGE='JAVASCRIPT' TYPE='TEXT/JAVASCRIPT'>

<!--
var popupwindow=null;
function popup(mypage,myname,w,h,pos,infocus){

if (pos == 'random')
{LeftPosition=(screen.width)?Math.floor(Math.random()*(screen.width-w))
:100;TopPosition=(screen.height)?Math.floor(Math.random()*((screen.height-h)-75)):100;}
else{LeftPosition=(screen.width)?(screen.width-w)/2:100;TopPosition=(screen.height)?
(screen.height-h)/2:100;}settings='width='+ w + ',height='+ h + ',top=' + TopPosition +
',left=' + LeftPosition + ',scrollbars=no,location=no,directories=no,status=no,menubar=
no,toolbar=no,resizable=no';popupwindow=window.open('',myname,settings);
if(infocus=='front'){popupwindow.focus();popupwindow.location=mypage;}
if(infocus=='back'){popupwindow.blur();popupwindow.location=mypage;popupwindow.blur();}
}
// -->
</script>

<style type="text/css">
<!--
body { font-family: Georgia;
color: #000000;font-size:20px;
background-color: #CCCCFF;

}

h1 { font-family: verdana;
color: #666666; font-size:35px;

}

h2 { font-family: Helvetica;
color: #6699CC; font-size:25px;

}

h3 {
font-family: Georgia;
color: #FF0000;font-size:15px;
}
-->
</style>
</head>

<h1> Huoltolähetteen tiedot</h1>
<br>
Huoltolähetteen numero: 987654
<br>
<br>
Status: Lähetetty huoltoon 20.11.2010 <a style="font-size:12px;color:#000066;font-family:verdana;" href="javascript:popup
('http://jannepul.brinkster.net/popup.html','pagename','440','280','center','front');">vaihda statusta / Poista lähete</a>

<br>

<h2>Asiakkaan tiedot</h2>

<form method="LINK" action="muokkaus.html" name="MyForm">

<table border="0">
<tr>
<td>Sukunimi: Malli</td>
<td></td>
</tr>
</table>
```

```

<td>Etunimi: Asiakas</td>
<td></td>
</tr>

<tr>
<td>Katuosoite: Valimotie 16</td>
<td></td>
<td>Postinumero ja toimipaikka: 00380 HELSINKI</td>
<td></td>
</tr>

<tr>
<td>Puhelinnumero: 05055555</td>
<td></td>
<td>Sähköposti: M_asiakas@gmail.com</td>
<td></td>
</tr>
</table>

<h2> Tuotetiedot</h2>

<table border="0">
<tr>
<td>Tuotenimi: Porakone</td>
<td></td>
<td></td>
<td></td>
</tr>

<tr>
<td>Tuotekoodi: 12345678</td>
<td></td>
<td></td>
<td></td>
</tr>
</table>

<h2> vian seloste</h2>

<table border="0">
<tr>
<td>Poran laturi on viallinen</td>
<td></td>
<td></td>
<td></td>
</tr>
</table>

<br>
<INPUT TYPE="submit" VALUE="Tulosta huoltolähete" style="height: 25px; width: 195px">
<br>
<INPUT TYPE="submit" VALUE="Palaa takaisin päävalikkoon" style="height: 25px; width: 195px">
<br>
<INPUT TYPE="submit" VALUE="Muokkaa lähetteen tietoja" style="height: 25px; width: 195px">
<br>

</form>
</body>
</html>

```


 Status

Valitse huoltolähetteen status

- Lähetetty huoltoon
- Palannut huollosta, odottaa noutoa
- Valmis, noudettu
- Poista lähete

Valitse päivämäärä

<<<	<	February 2011			>	>>>
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	1	2	3	4	5
6	7	8	9	10	11	12

Liite 10. Statuksen valinta näkymän

lähdekoodi

```
<html>
<head>

<style type="text/css">
  <!--
    body { font-family: Georgia;
          color: #000000;font-size:15px;
          background-color: #CCCCFF;

          }

    h1 { font-family: verdana;
        color: #666666; font-size:35px;

        }

    h2 { font-family: Helvetica;
        color: #6699CC; font-size:25px;

        }

    h3 {

        font-family: Georgia;
        color: #FF0000;font-size:15px;
        }
  -->
</style>
</head>
<title>Status</title>

<body>
Valitse huoltolähetteen status
<br>
<br>
<input type="radio" name="status" value="Lähetetty huoltoon"> Lähetetty huoltoon
<br>
<input type="radio" name="status" value="Palannut huollosta, odottaa noutoa"> Palannut huollosta, odottaa noutoa
<br>
<input type="radio" name="status" value="Valmis, noudettu"> Valmis, noudettu
<br>
<input type="radio" name="status" value="Poista lähete"> Poista lähete
<br>
<br>
valitse päivämäärä

<SCRIPT language="JavaScript" type="text/javascript">
<!--
function free_date_picker(obj_name, layer_name, text_to ,type, hide_self, language)
{
  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  // IF str < lng, add leading 0 to complete the str
  function leading_zeros(str, lng)
  {
    // param validation
    if(lng == null)
      lng = 2;

    var str = str.toString();

    while(str.length < lng)
      str = '0' + str;

    return str;
  }
  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  // Nbr days for this month
```

```

    return dt;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
function return_month_name(date)
{
    switch(language){
        case("en"):
            var month = new Array('January', 'February', 'March', 'April', 'May', 'June',
                'July', 'August', 'September', 'October', 'November', 'December');
            break;
        case("fr"):
        default:
            var month = new Array('Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin',
                'Juillet', 'Août', 'Septembre', 'Octobre', 'Novembre', 'Décembre');
    }

    return month[date.getMonth()];
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
this.setto = function(eng_date)
{
    // input text
    if(type == 1)
        document.getElementById(text_to).value = eng_date;
    // span, div ...
    if(type == 2)
        document.getElementById(text_to).innerHTML = eng_date;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
this.write_calendar = function(date)
{
    var nb_days_last_month = 0;
    var cnt_days = 0;
    var days_month = 0;
    var str = new String();
    var style = new String();
    var next_month = new Date();
    var last_month = new Date();
    var today = new Date();

    var date = new Date(date);
    var last_year = new Date(date);
    var next_year = new Date(date);

    date.setDate(1);
    next_month = returnNextMonth(date);
    last_month = returnLastMonth(date);
    nb_days_last_month = returnDays(last_month);
    cnt_days = -date.getDay()+1;
    days_month = returnDays(date);
    str = '';

    next_year.setYear(date.getFullYear() + 1);
    last_year.setYear(date.getFullYear() - 1);

    str += '<table cellspacing="0" cellpadding="0" border="1" width="250" bgcolor="white">';
    str += '<tr>';

    str += ' <td colspan="1" align="left"><a href="" + "javascript:" +
        obj_name + ".write_calendar(' + eval(last_year) + ')" + ";">---</a></td>';
}

```

```

str += ' <td colspan="1" align="left"><a href="' + "javascript:" +
obj_name + ".write_calendar('" + eval(last_month) + "')" + '"></a></td>';

str += ' <td colspan="3" align="center">'
str += return_month_name(date) + ' ' + date.getFullYear();
str += ' </td>';

str += ' <td colspan="1" align="right"><a href="' + "javascript:" +
obj_name + ".write_calendar('" + eval(next_month) + "')" + '"></a></td>';

str += ' <td colspan="1" align="right"><a href="' + "javascript:" +
obj_name + ".write_calendar('" + eval(next_year) + "')" + '"></a></td>';
str += ' </tr>';

var cnt_semaines = 0;
while(cnt_semaines < 6){
    cnt_semaines ++;

    str += ' <tr>';

    for(var cnt = 0; cnt < 7; cnt++){

        if(cnt_days > days_month){
            day = cnt_days - days_month;
            month = next_month.getMonth() + 1;
            year = next_month.getFullYear();

            style = "font-size:10px;";
        }

        else if(cnt_days < 1){
            day = cnt_days + nb_days_last_month;
            month = last_month.getMonth() + 1;
            year = last_month.getFullYear();

            style = "font-size:10px;";
        }

        else{
            var d1 = new Date();
            var d2 = new Date();

            d1.setMonth(date.getMonth());
            d1.setYear(date.getFullYear());
            d1.setDate(cnt_days);

            day = cnt_days;
            month = date.getMonth() + 1;
            year = date.getFullYear();

            if(d1.getFullYear() == d2.getFullYear() && d1.getMonth() == d2.getMonth() &&
            d1.getDate() == d2.getDate() ){
                style = "font-size:18px;font-color:red";
            }
            else{
                style = "font-size:12px;";
            }
        }

        dstr = year + '/' + leading_zeros(month) + '/' + leading_zeros(day);
        str += ' <td width="14%">';
    }
}

```

```

    day = cnt_days;
    month = date.getMonth() + 1;
    year = date.getFullYear();

    if(d1.getFullYear() == d2.getFullYear() && d1.getMonth() == d2.getMonth() &&
       d1.getDate() == d2.getDate() ){
        style = "font-size:18px;font-color:red";
    }
    else{
        style = "font-size:12px;";
    }
}

dstr = year + '/' + leading_zeros(month) + '/' + leading_zeros(day);

str += ' <td width="14%">|';
str += ' <span style="' + style + '">';
str += ' <a href="javascript:' + obj_name + '.setto(\'' + dstr + '\')">';
str += day;
str += ' </a>';
str += ' </font>';
str += ' </td>';

    cnt_days ++;
}

str += ' </tr>';
}

str += ' </table>';
document.getElementById(layer_name).innerHTML = str;
}

this.show = function()
{
    if(hide_self)
        document.getElementById(layer_name).style.visibility = 'visible';
}

this.hide = function()
{
    if(hide_self)
        document.getElementById(layer_name).style.visibility = 'hidden';
}

////////////////////////////////////
// create layer and show calendar
document.write('<div id="' + layer_name + '" onmouseout="' + obj_name + '.hide();">...</div>');
this.write_calendar(new Date());
}
</SCRIPT>

<table border=0>
<tr>
<td width=33%><input id='bbb'></td>
</tr>
<tr>
<td><script>var b = new free_date_picker('b', 'date_picker_b', 'bbb', 1, false, 'en');
</script></div></td>
</tr>
</table>

<br>
<INPUT TYPE="submit" VALUE="vahvista"
</body>
</html>

```

