

KEMI-TORNION AMMATTIKORKEAKOULU

Avainhallintasovelluksen käyttötapausten suunnittelu

Sami Sirén

Tietojenkäsittelyn koulutusohjelman opinnäytetyö
Web-palveluiden sisällöntuotannon ja tietoturvan suuntautumisvaihtoehto
Tradenomi

TORNIO 2011

TIIVISTELMÄ

Sirén, Sami. 2011. Avainhallintasovelluksen käyttötapausten suunnittelu. Opinnäytetyö. Kemi-Tornion ammattikorkeakoulu. Kaupan ja kulttuurin toimiala. Sivuja 66. Liitteet 1.

Opinnäytetyön tavoitteena oli suunnitella ja dokumentoida avainhallintasovelluksen uuden version käyttötapaukset ohjelmoijalle, joka kirjoittaa sovelluksen uuden version. Opinnäytetyö oli hankkeistettu ja sen toimeksiantajana oli Turvaykköset Oy.

Tutkimuksessa käytettiin konstruktiiivista sekä kvalitatiivista tutkimusmenetelmää. Lisäksi tutkimuksessa käytettiin haastatteluja sekä aivoriiheä tiedonkeruumenetelmänä. Suunnittelussa käytettiin hyväksi vanhaa avainhallintasovellusta sekä yksityiskohdissa top-down-menetelmää. Tutkimusaineistona opinnäytetyön tekijällä oli aiempi tietämys UML-kuvauskielestä sekä alan kirjallisuus ja internetaineistot. Työvälineinä oli kannettavassa tietokoneessa Microsoft Office -toimisto-ohjelmisto sekä UML-kaavioiden piirtoon Dia-sovellus.

Opinnäytetyön tuloksena syntyi käyttötapaukset ja käyttötapauskaaviot ohjelmoijalle, joka niiden perusteella on jo aloittanut avainhallintasovelluksen uuden version toteuttamisen suunnittelemalla tietokantarakenteen luovutettujen käyttötapausten perusteella.

Asiasanat: ohjelmistokehitys, UML, verkkoliiketoiminta

ABSTRACT

Sirén, Sami. 2011. Designing use cases of key control software. Bachelor's Thesis. Kemi-Tornio University of Applied Sciences. Business and Culture. Pages 66. Appendix 1.

The goal of this thesis was to design and document use cases and diagrams of a new version of key control software. The use cases are given to a programmer who will write the new version of the software. The commission was given by Turvaykköset Oy.

The research was carried out using constructive and qualitative research methodology. Interviews and think tank were used as data acquisition methods. The old version of the key control software was used in designing and top-down approach in details. As research material were used earlier knowledge of Unified Modeling Language and field-specific literature in addition to Internet sources. Laptop computer with Microsoft Office software and Dia for UML diagrams were used as tools.

The results of this thesis were use cases and diagrams to the programmer, who has already started producing the software by designing the database structure based on use cases.

Keywords: program development, UML, e-commerce

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO	5
1.1	Työn lähtökohta, tavoite ja rajaus	5
1.2	Tutkimusmenetelmä ja sisällön kuvaus	5
1.3	Yritysesittely	7
1.4	Teoreettinen viitekehys	7
1.5	Työvälineet	9
1.6	Käsitteet	10
2	UML	11
2.1	Kuvaus	11
2.2	Käyttötapausmallinnus	12
2.3	Avainhallintasovelluksen suunnittelu	15
3	AVAINHALLINTA –SOVELLUS	17
3.1	Perustieto	17
4	TUTKIMUS	20
4.1	Alustava tutkimus	20
4.2	Kaaviot	21
5	POHDINNAT	22
	LÄHTEET	24
	LIITTEET	27

1 JOHDANTO

1.1 Työn lähtökohta, tavoite ja rajaus

Turvaykköset Oy on aikoinaan ostanut avainhallintasovelluksen rungon Finnturva Oy:ltä ja kehittänyt sitä siltä pohjalta paremmin vastaamaan tarpeitaan.

Sovelluksella hallitaan kiinteistön avaintietokantaa, missä mikäkin avain on ja kuka sen on kuitannut. Sovelluksella tulostetaan kuittauslomakkeet, kun avaimia luovutetaan tai vastaanotetaan. Sovelluksesta voidaan myös tulostaa raportteja tietokannan sisällöstä. Sovellus on niin sanottu online-sovellus eli sitä käytetään internetsivujen kautta ja se tukee useita yhtäaikaisia käyttäjiä. Sovellus on kuitenkin kirjoitettu nyt jo vanhaksi jääneellä ohjelmointikielellä ja tietokantarakenne on jääne vanhasta järjestelmästä eikä arvioiden mukaan kykene kauaa vastaamaan käyttötarpeen asettamiin vaatimuksiin. Esille on tullut myös haluttuja ominaisuuksia, joita ei olisi helppoa tai mahdollista toteuttaa vanhassa ohjelmointiympäristössä. ASP-ohjelmointikielen (Active Server Pages) tuki loppuneen myös kielen uuden version ASP.NETin vallatessa enemmän tilaa parempine ominaisuuksineen. Sovellukseen on tarkoitus myöhemmin kehittää verkkoyhteyksiä toisiin sovelluksiin ja järjestelmiin, joita Turvaykkösillä ja yhteistyökumppaneilla on käytössä, mikä otetaan huomioon jo ohjelman suunnitteluvaiheessa.

Lähtökohtana on kehittää Turvaykköset Oy:lle nykyaikainen ja vaatimukset täyttävä uusi versio avainhallintasovelluksesta, joka olisi rakenteellisesti helposti laajennettavissa tulevaisuudessa. Tässä työssä tavoitteena on suunnitella ja luoda käyttötapaaukset uuden sovelluksen kirjoittamista varten.

1.2 Tutkimusmenetelmä ja sisällön kuvaus

Vaatimusmäärittelyn ja käyttötapausten tutkimusmenetelmä on kvalitatiivinen sekä konstruktiiivinen. Vaatimusmäärittelyn suunnittelussa työmenetelminä käytetään

aivoriiehiä sekä käyttäjien haastattelua. Myös muiden vastaavien avainhallintasovellusten ominaisuuksiin perehdytään ja haetaan sitä kautta ideoita toimivista ratkaisuista.

Hirsjärvi, Remes ja Sajavaara (2001, 152-155) kuvaavat laadullisen tutkimuksen tarkoituksiksi tosiseikkojen paljastamisen tai löytämisen. Lähtökohtana on todellisen elämän kuvaaminen. Tyypilliset kvalitatiivisen tutkimuksen piirteet, kuten haastattelut ja induktiivinen analyysi erottuvat tässä tutkimustyössä selkeästi. Konstruktiivinen eli soveltava tutkimus toteutuu käyttötapausten suunnittelussa ja toteutuksessa, sillä siinä analyysillä saadut tulokset arvioidaan ja niiden pohjalta pyritään kehittämään ja rakentamaan toimiva uusi innovaatio.

Aivoriiehiessä kokoontuvat sovelluksen pääkäyttäjät sekä ohjelmoija antamassa teknistä näkökulmaa kehitystyöhön. Aivoriiehiessä keskustellaan ja ideoidaan uuteen versioon tarpeelliseksi huomattuja ominaisuuksia sekä rajataan sovelluksen toiminta halutuksi eli päätetään mitä sovellukselta odotetaan ja mitä sillä on pystyttävä tekemään. Toisin sanoen luodaan vaatimusmäärittely. Aivoriiehiessä käydään läpi myös kaikki uudet ajatukset sovelluksen toiminnan laajentamisesta. Aivoriiehin tuloksena saadaan listaus uusista toteutettavista ominaisuuksista sekä poistettavat ominaisuudet.

Haastattelu sisältää niin asiakaskäyttäjien kuin Turvaykkösten toimipaikoissa työskentelevien avainhallintakäyttäjien haastatteluja. Haastattelu pidetään vapaamuotoisena keskusteluna ja siihen valmistaudutaan laatimalla ehdotuksia uusista ominaisuuksista sekä kysymyksiä, joilla voidaan selvittää tarkemmin, mitä osa-alueita sovelluksessa on tehty huonosti aiemmin tai mikä on täysin turha tietyn käyttäjän kannalta. Tuloksena pitäisi tulla listaukset hyvistä ja huonoista ominaisuuksista. Tulokset käydään läpi vielä uudessa tapaamisessa ja arvioidaan niiden käyttömahdollisuudet.

Toisiin sovelluksiin tutustutaan messuilla tai henkilökohtaisesti pyydetyillä tuotesittelyillä tai ottamalla sovellus koekäyttöön. Näillä keinoin sovelluksesta saa nopeasti kuvan, mitä ominaisuuksia siinä on uutta verrattuna muihin vastaaviin sovelluksiin.

Näiden aineistojen pohjalta rakennetaan sovelluksen käyttötapaukset eli kerrotaan, minkälaiset toiminnot sovelluksessa pitää olla käyttäjän näkökulmasta.

1.3 Yritysesittely

Turvaykköset Oy on perustettu vuonna 2007 alun perin nimellä Suomen Ykkösturva Oy ja yritys muodostui yhdentoista erillisen lukkoliikkeen yhteensulautumisesta. Ennen yhdistymistä kyseiset lukitusalan yritykset olivat kehittäneet markkinointiyhteenliittymän, jossa Turvaykköset oli ketju- ja brandinimenä. Turvaykköset-nimen yritys sai virallisesti käyttöön vuonna 2009. Toimipisteitä Turvaykkösillä on kaksitoista kymmenellä paikkakunnalla ja henkilöstöä noin 125 henkilöä. Liikevaihto oli vuonna 2009 noin 20 miljoonaa euroa. Yrityksen omistajuus jakautuu siten, että Eläke-vakuutusyhtiö Varmalla on 20 prosenttia ja yrityksen johdolla ja esimiehillä 80 prosenttia. Lukitusliiketoiminnassa Turvaykkösten markkinaosuuden arvioidaan Suomessa olevan noin 10 prosenttia. Yrityksen toimialaan kuuluu lukitustekniikka, kameravalvonta-, kulunvalvonta-, hälytys- ja turvajärjestelmät, automaattiovet ja -portit, kalterointi sekä laminointi. Yritys suorittaa niin suunnittelun kuin asennustyötkin. Näiden lisäksi yritys tarjoaa pääasiassa taloyhtiöille ja isännöitsijöille Turvaykköset NAKS™ -palvelua, johon kuuluu mm. avainhallinta tähän työhön liittyvällä sovelluksella.

1.4 Teoreettinen viitekehys

Verkkoliiketoiminta tai sähköinen liiketoiminta terminä tarkoittaa kaikkea sitä liiketoimintaa, jossa tietoverkkoja käytetään hyväksi kaikenlaisen tuottavuuden saamiseksi tai luomiseksi (Casals 2003). Verkkoliiketoimintaan lasketaan mukaan mm. online-sovellukset, verkkokauppa sekä myös verkossa tapahtuva markkinointi.

Verkkoliiketoiminta on lukitusosalalla varsin uusi tekijä. Viimeisen kymmenen vuoden aikana tietotekniikan mukaantulo on helpottanut laskutusta, kassatoimintaa, varasto- ja muuta kirjanpitoa. Työtilausten läpivienti tarjouksesta ja tilauksesta laskutukseen asti toimii tietokoneiden ja tietoverkkojen avulla täysin sähköisessä muodossa. Lukitustekniikassa tietoverkkojen käyttö on lisääntynyt sähkölukkojen ja kulunvalvonnan myötä. Kulunvalvonta voi toimia tietoverkkojen kautta ja valvontatieto

kulkee nykyään suojattuna vaikka internetin kautta valvomoon. Murtohälytysjärjestelmät siirtävät hälytyksen vartiointiliikkeeseen tai muuhun vastaavaan valvomoon usein puhelin- tai GSM-verkon kautta, mutta myös tässä internet on tullut mukaan valikoimaan. Kameravalvonnassakin on nykyään mahdollista seurata kohdetta internetistä käsin tunnistautumalla suoraan kamerassa olevaan internetpalvelimeen. Vaikka sinänsä tietoverkkoja käytetään monessakin tekniikassa hyödyksi, ei alan tuotteita tai palveluita yleensä mielletä verkkoliiketoimintaan liittyviksi. Avainhallinta kuitenkin sovelluksena käyttää tietoverkkoa käyttöliittymässään ja tarjoaa sitä kautta hyödykkeen asiakkaalle. Aiemmin avainhallinta on ollut vain vihkoon tehtyjä merkintöjä ja kuittauksia avainkohtaisesti tai sarjoituskaavio jossain kansiossa. Nyt se kuitenkin on täysin sähköistettynä ja kaikkien käytettävissä tietoverkon kautta ja tarjoaa aina reaaliaikaisen tiedon avaimen sijainnista ja historiasta.

Tietoverkkoja hyödynnettäessä liiketoiminnassa voidaan tietoa tallentaa ja siirtää nopeasti ja varmasti paikasta toiseen. Tiedon siirtämisessä ja tallentamisessa täytyy usein olla tarkkana, ettei kriittistä tietoa vuoda järjestelmän ulkopuolisille. Menetelmiä tiedon sekä tiedonsiirron salaamiseen on kehitetty, joten siirto tapahtuu tarpeen vaatiessa tietoturvallisuutta menettämättä. Tietoa voidaan helposti tallentaa ja analysoida nopeasti myös suurestakin määrästä edellyttäen, että se kaikki on digitaalisessa ja luettavassa muodossa. Tietoverkkojen myötä kaikki tuo tieto voidaan myös keskittää palvelimeen siten, ettei tarvitse olla fyysisesti lähellä voidakseen käyttää sitä, niin kuin tilanne oli vielä 1980-luvulla. Palvelimessa olevaa tietoa voidaan käsitellä etänä sopivia työkaluja käyttäen ja parhaimmassa tapauksessa tietoa voi käsitellä moni käyttäjä yhtä aikaa. Näiden tekijöiden myötä tiedonsiirtoon ja tallentamiseen liittyvissä toimissa tulee huomioida tietoturva. Sen tärkeimmät ydintekijät ovat luottamuksellisuus, eheys ja saatavuus (ISO/IEC 27002 2005). Luottamuksellisuudella tarkoitetaan, että tiedon pitää pysyä vain valtuutettujen henkilöiden tai järjestelmien saatavilla. Eheydellä tarkoitetaan, ettei tieto saa muuttua tai sitä ei saa muutettua huomaamatta missään vaiheessa. Saatavuudella tarkoitetaan, että tiedon tulee olla saatavilla aina, kun sitä tarvitaan, ilman kohtuuttomia viiveitä.

Sovelluksen suunnittelussa on myös kyseessä käytettävyyden suunnittelu. Sovelluksen käytettävyyden kymmenen päänyrkkisääntöä ovat Nielsenin mukaan (2005):

- järjestelmän tilan näkyvyys
- järjestelmän ja oikean maailman vastaavuus
- käyttäjän vapaus liikkua sovelluksessa
- yhtenäisyys ja johdonmukaisuus
- virheiden ennaltaehkäisy
- muistettavuus
- joustavuus ja tehokkuus
- esteettinen suunnittelu
- virheiden tunnistaminen ja korjaaminen
- apu ja dokumentaatio.

Vaikka suurimman osan käytettävyyden suunnitteluun suorittaa sovelluksen ohjelman tekijä, täytyy se ottaa huomioon jo sovelluksen käyttötapauksia suunnitellessa. Tuotteen tulee olla mieluisa ja helppo käyttää, ja mielellään sen opetteluun ei kulu liikaa aikaa, vaikka ei osaisikaan tietotekniikkaa paljon. Sovelluksen toiminnan tulee olla johdonmukaista ja tehokasta sekä virheiden ennaltaehkäisyyn tulee kiinnittää huomiota. Samalla tietoturvallisuus ei saa vaarantua sovellusta käytettäessä.

1.5 Työvälineet

Teknistä välineistöä edustavat opinnäytetyön tekijän oma kannettava tietokone Acer TravelMate 5520 sekä Turvaykkösten kannettava tietokone HP ProBook 6545b. Molemmissa käyttöjärjestelmänä on Microsoft Windows sekä sisältävät Microsoft Office -toimisto-ohjelmistopakettin, jota käytetään työn kaikissa tekstinkäsittelytehtävissä sekä graafisissa kuvaajissa. Kuvankaappausten ja kuvien muokkaukseen käytetään ilmaista IrfanView-sovellusta. Käyttötapauskaavioiden piirtämiseen käytetään ilmaista Dia-sovellusta. Tulostamiseen on käytettävissä Canon LBP2900 lasertulostin. Avainhallintatiimin palavereja varten käytettävissä on Turvaykkösten toimipaikkojen neuvotteluhuoneet Helsingin Oulunkylässä sekä Kouvolassa. Etäpalaveriinkin käytetään Skype-sovellusta.

1.6 Käsitteet

Active Server Pages eli ASP, tai myös Classic ASP, on Microsoftin ensimmäinen dynaaminen palvelinskriptikieli, joka julkaistiin noin vuonna 1998. Palvelinskripti suoritetaan web-palvelimessa ja sen tuottaa pelkkää tekstiä tai html-kieltä, jonka internet-selain sitten tulkitsee. (Microsoft Active Server Pages 1997.)

Enterprise Resource Planning eli toiminnanohjausjärjestelmä on integroitu tietojärjestelmä, jolla voidaan mm. hoitaa yrityksen laskutus, kirjanpito, kassa ja palkanlaskenta (DL Prime ERP 2011).

Https-yhteys (HyperText Transfer Protocol Secure) on IETF:n (The Internet Engineering Task Force) luoma yhteysprotokolla RFC2818, jossa http-protokollan viestit lähetetään salattuna internetyhteyden läpi (Rescorla 2000).

Intranet on yrityksen sisäinen lähiverkko, johon vain yrityksen henkilökunnalla on käyttöoikeus (Mensola 1997).

Käyttötapauskaavio on yksi UML-kielen kaaviotyypin ilmentymä, jolla kuvataan graafisesti jokin sovelluksen toiminto käyttäjän näkökulmasta (Eriksson & Penker 2000, 13).

Ohjelmointirajapinta (Application Programming Interface, API) on sovelluksen ominaisuus, jonka avulla kyseinen sovellus voi keskustella toisten sovellusten kanssa etukäteen sovittujen sääntöjen puitteissa (PC Magazine Encyclopedia 1998-2011a).

Online-sovellus tarkoittaa tietokoneohjelmaa, jota käytetään tietoverkon, kuten internet, yli joko internetselainta tai erillistä asiakasohjelmaa käyttäen (Tietotekniikan termitalkoot 2001).

Tagi (engl. Tag – merkki, lappu) tarkoittaa kulunvalvonnassa käytettävää yleensä rfid-tekniikalla toteutettua tunnistetta, jolla saadaan sähköisesti kulkuoikeus käyttämällä sitä etälukijassa (PC Magazine Encyclopedia 1998-2011b).

Top-Down-menetelmä tarkoittaa suunnittelussa tapaa ensin suunnitella kokonaiskuva ja sitten jakaa se osa-alueisiin ja suunnitella ne tarkemmin (Chatfield & Johnson 2000).

UML eli Unified Modeling Language on Object Management Groupin (OMG) vuonna 1997 standardoima graafinen mallinnuskieli, jolla voidaan kuvata sovelluksen rakennetta, käyttäytymistä ja vuorovaikutusta kaavioiden avulla (OMG 2011).

2 UML

2.1 Kuvaus

Unified Modeling Language (yhtenäinen mallinnuskieli) on graafinen kuvauskieli, jolla määritellään, visualisoidaan, rakennetaan ja dokumentoidaan järjestelmiä tai prosesseja. Sen historia juontuu tarpeeseen kehittää yhtenäinen menetelmä olio-ohjelmointiprojekteihin. 1970-luvun lopulla C++:n ja SmallTalkin yleistyttyä oliosuuntaus tuli suosituksi ja yleistyi, joten tarvittiin työmenetelmiä, jotka tukivat ohjelmistokehitystä. 1980-luvulla suosittuja oliopohjaisia mallinnustekniikoita olivat muiden muassa: Booch, Object Modeling Technique (OMT), OOSE, Objectory, Fusion ja Coad/Yourdon. Nämä erosivat kukin hieman toisistaan ja kullakin oli oma tapansa ja symbolinsa kuvata oliot ja mallit. Tästä johtuen ohjelmistokehittäjien täytyi ennen kaikkea valita projektiin parhaiten sopiva työmenetelmä. Käytännössä tämä johti siihen, että kokeneet kehittäjät ottivat yhden mallin pohjaksi ja lisäsivät siihen joitain muiden menetelmien hyväksi havaitsemiaan ominaisuuksia. (Eriksson & Penker 2000, 3-4.)

UML:n kehityksen nähdään alkaneen, kun Booch-menetelmän kehittänyt Grady Booch ja toinen OMT:n pääkehittäjä James Rumbaugh alkoivat kehittää Unified Method:ia vuonna 1994. Joukkoon liittyi vuonna 1995 vielä Ivar Jacobson, joka oli kehittänyt OOSE:n ja Objectoryn. Siinä vaiheessa he huomasivat kehittäneensä standardiksi soveltuvan mallinnuskielen menetelmän sijaan ja nimesivät sen Unified Modeling Language:ksi. He julkaisivat useita esijulkaisuja UML:sta ja palautteen perusteella parannettuaan työtään julkaisivat version 1.0 tammikuussa vuonna 1997. Tällä hetkellä uusin versio on versio 2.3 ja se on julkaistu toukokuussa 2010. Eriksson ja Penker (2000, 4-5) kuvaavat kirjassaan, että kehittäjiensä mukaan UML:n tavoitteet ovat seuraavat:

- Järjestelmien mallintaminen oliopohjaisesti.
- Käsitteellisten ja fyysisten asioiden välisen yhteyden muodostaminen.
- Monimutkaisten, toimintakriittisten järjestelmien skaalautuvuusongelmien ratkaisu.

- Sellaisen mallinnuskielen kehittäminen, jota sekä ihmiset että koneet voivat käsitellä.

Mallinnuskielen uusin versio 2.3 koostuu neljästätoista kaaviotyypistä, jotka jaetaan kahteen kategoriaan: rakennetta kuvaaviin kaavioihin ja käytöstä kuvaaviin kaavioihin.

Rakennetta kuvaavat kaaviot ovat

- luokkakaavio (class diagram)
- oliokaavio (object diagram)
- komponenttikaavio (component diagram)
- yhdistelmärakennekaavio (composite structure diagram)
- sijoittelu- tai käyttöönottokaavio (deployment diagram)
- pakettikaavio (package diagram)
- profiilikaavio (profile diagram).

Käytöstä kuvaavat kaaviot ovat

- aktiviteettikaavio (activity diagram)
- käyttötapauskaavio (use case diagram)
- tilakonekaavio (state machine diagram)
- sekvenssikaavio (sequence diagram)
- kommunikaatiokaavio (communication diagram)
- kokoava vuorovaikutuskaavio (interaction overview diagram)
- ajoitustuskaavio (timing diagram).

Kaavioissa käytetään mallinnuselementtejä, kuten luokkia, olioita sekä viestejä, kuten assosiaatioita, riippuvuuksia ja yleistyksiä. Kaikki nämä vastaavat yleisiä olio-ohjelmoinnin käsitteitä. (Eriksson & Penker 2000, 11.)

2.2 Käyttötapausmallinnus

Käyttötapausmallinnuksen kehitti Ivar Jacobson OOSE- ja Objectory-työmenetelmien perusteella. Käyttötapausmallinnus on tekniikka, jolla kuvataan järjestelmän toimintoja.

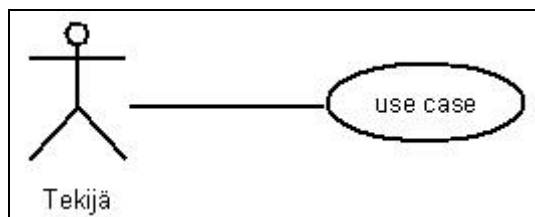
Yksittäinen käyttötapaus kuvaa järjestelmän yhtä toimintoa. Käyttötapaukset sisältävät ulkoisia toimijoita (actors) ja järjestelmän tarjoamia käyttötapauksia (use case) sekä näiden välisiä yhteyksiä. Käyttötapaukset kuvaavat toiminnon vain ulkoisen toimijan näkökulmasta, eikä ota lainkaan kantaa, kuinka toiminto toteutetaan.

Seuraavassa kuvataan uusimman UML version 2.3 määrittelyn mukaiset kuvaukset ja piirrosmerkit (OMG 2010).

Toimija tai tekijä on järjestelmän käyttäjä. Yleensä sillä tarkoitetaan järjestelmän ulkoista tekijää, kuten esimerkiksi ihmistä, konetta tai toista järjestelmää. Toimijalla voidaan kuvata myös useaa käyttäjää ja vaihtoehtoisesti yksi käyttäjä voi ilmentyä monessa eri toimijassa. Sitä kuvataan yksinkertaisella tikku-ukolla.

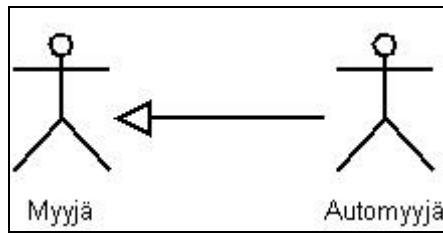
Käyttötapaus kuvataan ellipsinä, jonka nimi lukee käyttötapauksen sisällä tai alapuolella. Käyttötapaus kuvaa yhtä toimintoa toimijan näkökulmasta ja se on yleensä yhteydessä toimijaan. Nimeämiseen on suositeltavaa käyttää verbimuotoa.

Viestiassosiaatio on nimitys toimijan ja käyttötapauksen väliselle yhteydelle. Se on usein yksi yhteen -tyyppinen yhteys eikä sillä ole suuntaa eli se kommunikoi molempiin suuntiin. Sitä kuvataan pelkällä yhtenäisellä viivalla. Nämä on kuvattu kuvassa 1.

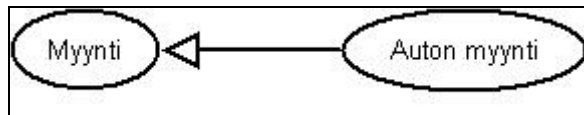


Kuva 1. Tekijän ja käyttötapauksen välinen assosiaatio

Yleistyssuhde kahden entiteetin eli olion välillä kuvaa yleisen olion ja erikoistuneen olion välistä suhdetta. Erikoistunut olio sisältää yleisen olion yhden erikoistuneen muodon. Suhde voidaan piirtää toimijoiden (kuva 2) tai käyttötapausten (kuva 3) välille. Piirtäminen tapahtuu yhtenäisellä viivalla oliosta toiseen ja yleisen olion päässä viivaa on ontto kolmio.

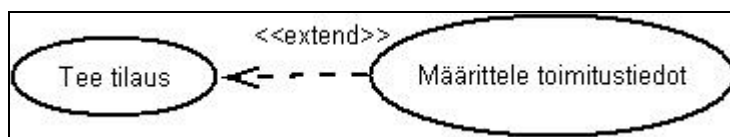


Kuva 2. Toimijoiden suhde



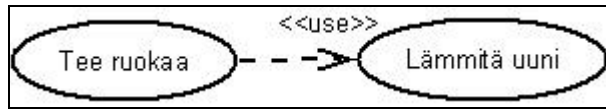
Kuva 3. Käyttötapausten suhde

Laajennussuhde on suhde, jossa erikoiskäyttötapaus lisää peruskäyttötapaukseen jotain toimintoja. Peruskäyttötapaus toimii itsenäisenäkin, mutta erikoiskäyttötapaus on riippuvainen peruskäyttötapauksesta ja lisää siihen vain jotain toimintoja. Erikoiskäyttötapausten suorittaminen ei ole pakollista, vaan se lisää mahdollisia toimintoja. Esimerkiksi peruskäyttötapaus **Tee tilaus** voisi sisältää erikoiskäyttötapausten **Määrittele toimitustiedot**. Tilauksia voidaan tehdä ilman toimitustietojakin, mutta tarvittaessa voidaan määrittellä nekin. Piirtäminen tapahtuu katkoviivanuolella, jossa nuolenpäähän liitetään kaksoiskulmasulkeissa sana extend (kuva 4).

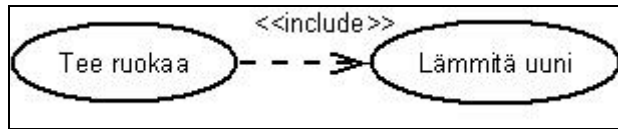


Kuva 4. Laajennussuhde

Käytösuhde on suhde, jossa yksi käyttötapaus käyttää toista siten, että erikoistunut käyttötapaus sisältää yleisen käyttötapausten toiminnot. Tätä sanotaan perimiseksi. Käytösuhde vastaa täysin riippuvuussuhdetta include. Esimerkissä käyttötapaus **Tee ruokaa** sisältää aina käyttötapausten lämmitä uuni, järjestelmänä esimerkissä on uuni. Tällöin ruuan tekeminen vaatii aina uunin lämmittämisen. Käytösuhteen piirtäminen tapahtuu katkoviivanuolella, jossa nuolenpäähän liitetään kaksoiskulmasulkeissa sana use (kuva 5). Riippuvuussuhteessa piirtäminen tapahtuu katkoviivanuolella, jossa nuolenpäähän liitetään kaksoiskulmasulkeissa sana include (kuva 6).



Kuva 5. Käytösuhde



Kuva 6. Riippuvuussuhde

2.3 Avainhallintasovelluksen suunnittelu

Ensimmäisessä palaverissa sovelluksen ohjelmoija pyysi saada sovelluksesta toimintakartoituksen, josta selviää, mitä toimintoja sovelluksella toteutetaan. Dobing ja Parsons (2005) kertovat UML-kirjoissa usein neuvottavan käytettäväksi pelkkiä käyttötapauskaavioita vaatimusmäärittelyn tekemiseen asiakkaan kanssa, joten sen käyttöä ehdotettiin sekä vanhan sovelluksen kartoittamiseen että uuden version suunnittelemiseen.

UML on kielenä standardoitu ja käyttötapauskaavio on riittävän yksinkertainen ja selkeä kuvaamaan sovelluksen käyttötapauskaavioita. Käyttötapausmäärittelyn katsottiin soveltuvan erinomaisesti verkkosovelluksen suunnitteluun. Sovelluksen rakenne on modulaarinen eli koostuu osista ja kukin toteuttaa jonkin toiminnon eli käyttötapauskaavioita. Käyttötapauskaaviolla on myös aika reilusti liikkumavaraa, jota kohdennetaan sanallisella kertomuksella käyttötapauskaavioita.

Muita kaavioita ei katsottu tarpeellisiksi. Rakenne on ohjelmoijan vapaasti muotoiltavissa eikä rakennetta kuvaavia kaavioita tarvittu lainkaan. Rakenteen perusteet tulee vanhasta sovelluksesta eli verkon yli tapahtuvasta tietokannan ylläpidosta yksilöllisellä käyttöliittymällä. Käytöstä kuvaavista kaavioista parhaiten tähän soveltui käyttötapauskaavio ja -määrittelyt. Vaihtoehtoisista sekvenssikaavioista ei olisi ollut tarkoituksenmukainen, koska sovelluksen suunnittelussa ei ole menty kovin syvästi olioiden suunnitteluun asti ja täten sekvenssien suunnittelu olisi ollut hyvin hankalaa.

Interaktiota kuvaavia kaavioita ei tarvittu lainkaan, koska suurin osa käyttötapauksista oli perusteeltaan käyttäjän syöttämän tiedon tallentamista tietokantaan.

3 AVAINHALLINTA –SOVELLUS

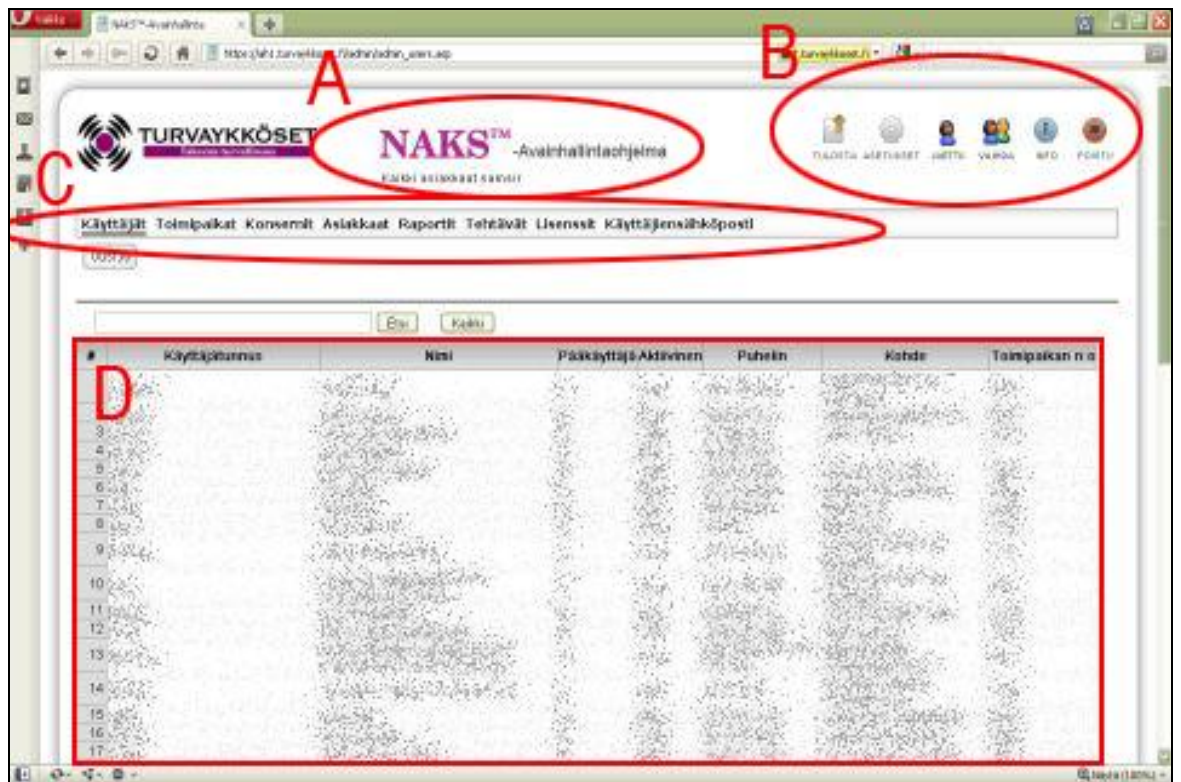
3.1 Perustieto

Sovelluksella ylläpidetään ajantasaista tietokantaa kohteista, kohteiden avaimista ja niiden liikkeistä. Sovelluksen kohdeasiakkaita ovat isännöitsijät, kiinteistösihteerit tai vastaavat henkilöt, jotka tekevät kiinteistöihin uusien avainten tilauksia tai hallinnoivat muuten kiinteistön avaimistoa ja lukostoa.

Sovellus on kirjoitettu Classic ASP -kielellä ja sovellusta suoritetaan Windows Server 2003 –palvelimessa. Tällä hetkellä kirjautumisen käyttäjätunnistuksesta vastaa RSA:n SecurID-palvelu, joka on kahden tekijän tunnistusmenetelmä. Käyttäjä syöttää oman käyttäjätunnuksensa sekä kahdesta osasta muodostuvan RSA-koodin: ensin oma pin-koodi ja sitten SecurID-avaimenperässä näkyvä kuusinumeroisen luku. Onnistuneen käyttäjätunnistuksen jälkeen RSA:n kirjautumispalvelin siirtää käyttäjän tunnuksilleen oikeutettuun etusivuun.

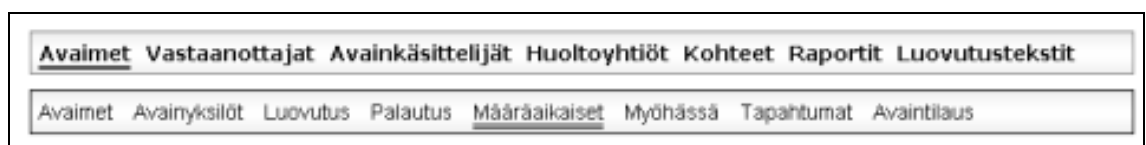
Perusnäköymän osat ovat seuraavat (kuva 7):

- A. Kirjautumistiedot, joista näkee millä tunnuksella on kirjaututtu järjestelmään ja onko se liitetty johonkin tiettyyn asiakkaaseen ja kohteeseen. Pääkäyttäjällä tai toimipaikkakäyttäjällä ei liitosta ole.
- B. Käyttäjävalikko, jonka sisältö vaihtelee hieman käyttäjätyyppin mukaan.
- C. Ohjelmavalikko, jonka sisältö vaihtelee enemmän käyttäjätyyppin mukaan. Alleviivaus kertoo aktiivisen valikon.
- D. Sisältörivit, jossa näkyy aina kulloisenkin toiminnon tai raportin sisältö.



Kuva 7. Sovelluksen perusnäkyä selaimessa

Käyttäjien perusvalikot eroavat toisistaan jonkin verran. Kuvassa 8 tavallisen asiakaskäyttäjän valikko perustuu lähinnä tietojen selaamiseen ja avainhallintaan, kuten avainten luovutukseen ja palautukseen. Käyttäjä voi lisätä vastaanottajia, avainkäsittelijöitä, huoltoyrityksiä ja -miehiä, mutta ei muita. Avaimet-valikko on sen verran laaja, että sille on luotu oma valikkorivi perusvalikon alle.



Kuva 8. Käyttäjän valikko

Kuvassa 9 toimipaikkakäyttäjän valikossa on tietojen selaamisen ja hallinnan lisäksi avaintilausten hoito ja raportit.



Kuva 9. Toimipaikkakäyttäjän valikko

Kuvassa 10 pääkäyttäjän valikossa voi hallinnoida käyttäjiä, toimipaikkoja, konserneja, asiakkaita ja ottaa raportteja. Tehtävät-valikosta näkee aktivoimattomat käyttäjät ja pääkäyttäjällä voi aktivoinnin tehdä.



Kuva 10. Pääkäyttäjän valikko

4 TUTKIMUS

4.1 Alustava tutkimus

Uuden ohjelmaversioon tekeminen alkoi yrityksen hallituksen päätöksestä kehittää avainhallinnasta uudempi versio, johon voitaisiin toteuttaa halutut ominaisuudet, joita ei aiempaan ohjelmaversioon voinut tai kannattanut tehdä. Aiemmin oli syntynyt tarve toteuttaa sovellus uudella ohjelmatekniikalla ja nyt oli syntynyt halu toteuttaa sovelluksesta uusi versio ja tässä avainhallintatiimille annettiin vapaat kädet sidotun budjetin rajoissa. Sovelluksen uuden version kehittäminen oli ollut avainhallintatiimin ajatuksena jo pitkään, mutta rahoituksen puutteessa on se jäänyt toteuttamatta.

Ensimmäisessä suunnittelupalaverissa ohjelmointityöstä vastaava Timo Kauranen kertoi valinneensa uuden sovelluksen ohjelmointiympäristöksi Microsoftin .Net -framework:in ja ohjelmointikieleksi C#:n. Valinta oli helppo, sillä aiemman ohjelman osat ovat kirjoitettu ASP-kielellä (Active Server Pages) ja ovat kohtalaisen helppo siirtää suurimmalta osin uuden kielen pariin, eikä suuria muutoksia tarvitse tehdä niiltä osin. Vanhan sovelluksen osia voidaan käyttää hyödyksi uuden version runkona.

Seuraavana oli vanhan sovellusversion käyttötapausten kartoittaminen. Asia toteutettiin käyttämällä demosovellusta, joka on käytössä olevan version identtinen kopio. Se toimii kokeellisena alustana, jossa testataan jokainen kehitysidea, ennen kuin se siirretään käytössä olevaan versioon. Tällä vältetään riskittömästi vakavien virheiden aiheuttamat sotkut käytössä olevassa sovelluksessa. Samalla voitiin testata monia toimintoja, joita ei normaalissa päivittäisessä avainhallinnassa välttämättä tule käytettyä, kuten esimerkiksi tietojen poistot. Jokainen sovelluksen ominaisuus kokeiltiin käytännössä ja näin ne kartoitettiin järjestyksessä ja kirjattiin ylös uuden version suunnitelman pohjaksi.

Uusi järjestelmä päätettiin rakentaa siten, että vanha tietokanta voidaan replikoida uuteen versioon ilman täydellistä uudelleen kirjoitusta. Siirtymisen pitäisi tapahtua riittävän nopeasti, eikä järjestelmään tarvitse teettää käyttökatkosta. Tietokannan replikointi voidaan toteuttaa vaikka viikonlopun aikana, jolloin käyttö on vähäisintä. Täten siirtymäajasta saadaan nopeampi ja käyttäjäystävällisempi, vaikka samalla

päätimmekin käyttää tarpeen vaatiessa vaikka kahta järjestelmää rinnakkain, kunnes uusi ohjelmaversio on täydellisesti käynnistetty ja tietokanta uuden version käyttöön replikoitu.

Sovelluksen kehittäjän tapa tehdä muutokset ja kehitykset ensin demoversioon ennen varsinaisen käyttöversion muokkaamista päätettiin ottaa käyttöön myös uuden version kanssa. Tarkemmin ottaen uusi versio tehdään pikkuhiljaa niinsanotuksi demoversioksi, jonka kehitys on nähtävissä koko ohjelmointityön aikana. Näin uusien ominaisuuksien testaus pääsee käyntiin heti, kun ominaisuus on lisätty demoversioon.

Sovelluksen kehityksessä otetaan myös huomioon se mahdollisuus, että sovellus voi tulevaisuudessa käyttää hyödyksi toisten tietojärjestelmien verkkopalveluja. Tarkoituksena on kehittää menetelmiä, joilla lukostotiedostot voidaan lukea automaattisesti järjestelmän muistiin, jolloin nykyisin täysin manuaalista työtä vaativa osuus jäisi kokonaan pois tai työn määrä vähenisi huomattavasti. Tällä hetkellä näitä verkkopalveluja ei vielä ole saatavilla, mutta neuvotteluja halutaan käydä asian eteenpäin viemiseksi.

4.2 Kaaviot

Sovelluksen kehittämisessä huomasi kaavioiden tarjoavan aika vähän tarkempaa tietoa, mutta antavat nopeasti silmäyksellä kuvan sovelluksen vaativuudesta. Monet yhteydet käyttötapauksesta toiseen viittaavat vaikeampaan rakenteeseen. Itse suunnittelin ensin karkeasti sovelluksen rakenteen piirroksena, sitten toiminnan tekstinä ja käyttökuvausena ja lopuksi muodostin kaavion kyseisestä toiminnosta. Liitteessä yksi on esimerkkinä käyttötapausselostus ja -kaavio, jossa kuvataan eräät asiakkaalle mahdolliset käyttötapaukset.

5 POHDINNAT

Sovelluksen kehittämisessä mukana olo näin alkuvaiheesta lähtien on hyvin mielenkiintoista. Kokemukseni sovelluksen suunnittelusta on ollut aiemmin kovin vähäistä, mutta nyt aloitettu projekti tulee kestämaan pitkään ja sisältää kehitystyössä kaikkea paitsi koodin kirjoittamista. Sovelluksen suunnittelun jälkeen alkaa toteutusvaihe, joka kestää ohjelmoijan mukaan mahdollisesti kuukausia. Sovelluksesta kirjoitetaan aluksi runko käyttäen vanhasta versiosta hyödyksi lähdekoodia, jos se vain on mahdollista. Tässä huomioidaan palaverissa sovitut poisjätetyt ominaisuudet, joiden koodia ei käännetä ollenkaan uuteen ympäristöön. Muilta osin koodi käännetään uuden sovelluksen käyttöön niin pitkälti kuin mahdollista.

Tässä projektissa oli hyvää se, että alkuasetelmana voitiin käyttää vanhaa, jo käytössä olevaa sovellusta. Sovelluksen runko oli helppo suunnitella pohjautuen tietoon ja käytäntöön siitä, mitä avainhallintatyö sisältää tai pitäisi sisältää. Käyttötapausten suunnittelu antaa erilaisen näkymän sovelluksen toimintaan ja suunnitellessa syntyy näkemys siitä, minkälainen sovelluksesta pitää tulla. Käyttötapaukset eivät kuitenkaan määrittele sovelluksen teknistä toteutusta mitenkään ja siinä jää sovelluksen kirjoittajalle valinnan vapaus, kuinka toteuttaa käyttötapausten mukainen toiminta tai ominaisuus. Tässä kuitenkin on kuultu ohjelmoijaa, jotta tiedämme lähtökohtaisesti, mitä voidaan toteuttaa helposti ja mitä ei. Omat kokemukset uudesta ohjelmointiympäristöstä sekä -kielestä auttaa näkemään mahdollisuuksia toteuttaa jotain ominaisuuksia. Esimerkkinä .NET-ympäristö sisältää automaattisia taulukkojen käsittelyn toimintoja ilman, että tarvitsee itse kirjoittaa niitä, kuten lajittelu kentän mukaan. Tietoisuus olemassa olevista helpottavista teknisistä ominaisuuksista auttaa suunnittelemaan helppokäyttöisempiä sovelluksia, kun tietää suunnitellessa niiden toteuttamisen olevan mahdollista. Tämän työn suunnittelussa olenkin hieman pitänyt mielessä, ettei budjetti välttämättä tule olemaan ääretön.

Työmenetelmät osoittautuivat hyväksi. Vanhan sovellusversion koekäyttö ja käyttötapausten kartoitus osoittautui erittäin tärkeäksi. Niiden avulla sain sovelluksen kokonaistarkoituksesta hyvän kuvan ja osasin mielestäni ohjata sen kehitystä oikeaan suuntaan, jotta uudesta versiosta saadaan paras hyöty yritykselle tulevaisuudessa. Uusien ominaisuuksien kartoittamiseen haastatteluja olisi saanut olla useampi sekä

kirjallinen kysely mahdollisista toiveista olisi saattanut tuoda jotain uutta mieleen. Kaikki ei välttämättä tule mieleen juuri haastattelutilanteessa ja ajallisten resurssien vähydestä johtuen haastatteluja ei voitu järjestää enempää. Yhteydenpito tiimin kesken oli melko vähäistä, mutta ei liian. Keskustelu tapahtui lähinnä sähköpostilla siten, että kun minulla syntyi mielestäni tarpeeksi kysymyksiä, lähetin useamman kysymyksen kerralla toiselle pääkäyttäjälle tai ohjelmoijalle. Suuremmat epäselvyydet selvitimme puhelimitse.

LÄHTEET

Painetut

Eriksson, Hans-Erik & Penker, Magnus 2000. UML. Edita/IT Press, Helsinki.

Hirsjärvi, Sirkka & Remes, Pirkko & Sajavaara, Paula 2001. Tutki ja kirjoita. 6.-7. painos. Tammi, Helsinki.

ISO/IEC 27002 2005. Informaatioteknologia. Turvallisuus. Tietoturvallisuuden hallintaa koskeva menettelyohje [Information technology - Security techniques - Code of practice for information security management]. International Organization for Standardization, Geneva.

Painamattomat

Candela, Pierre (toim.) 2003. What is HyperText. Luettu 30.10.2010.

<<http://www.w3.org/WhatIs.html>>

Casals, Kare 26.5.2003. Verkkoliiketoiminta. Luettu 30.10.2010.

<http://www.e-finland.org/ukk/_/kasitteet/verkkoliiketoiminta>

Chatfield, Carl & Johnson, Timothy 2000. Microsoft Project 2000 Trainer. Luettu 12.4.2011.

<http://cna.mikkeliyamk.fi/Public/KosonenH/Projektinhallinta/project/project_luku03.pdf>

DL Prime ERP. Luettu 8.4.2011.

<<http://www.dlsoftware.com/document.aspx?docID=49>>

Dobing Brian & Parsons Jeffrey 2005. Current Practices in the Use of UML. Teoksessa Akoka Jacky ym. (toim.) Perspectives in Conceptual Modeling, ER 2005

Workshops AOIS, BP-UML, CoMoGIS, eCOMO, and QoIS Klagenfurt, Austria, October 24–28. Berlin: Springer-Verlag, 3.

<<http://books.google.fi/books?id=-b181PShYsUC&printsec=frontcover#v=onepage&q&f=false>>

Mensola, Sami 1997. Intranet. Luettu 4.4.2011.

<<http://www.netlab.tkk.fi/opetus/s38116/1997/esitelmat/41748f/>>

Microsoft Active Server Pages 1997. Frequently Asked Questions. Luettu 1.4.2011.

<<http://msdn.microsoft.com/en-us/library/ms972347.aspx>>

Miten tilaat avainhallintapalvelun käyttösi?. Luettu 27.10.2010.

<<http://www.turvaykkoset.fi/avainhallintatilaus>>

Nielsen, Jakob 2005. Ten Usability Heuristics. Luettu 31.10.2010.

<http://www.useit.com/papers/heuristic/heuristic_list.html>

OMG 2010. OMG Unified Modeling Language™ (OMG UML), Superstructure. Luettu 1.12.2010.

<<http://www.omg.org/spec/UML/2.3/Superstructure/PDF/>>

OMG 2011. Introduction To OMG's Unified Modeling Language™ (UML®) . Luettu 6.4.2011.

<http://www.omg.org/gettingstarted/what_is_uml.htm>

PC Magazine Encyclopedia 1998-2011a. Definition of: API. Luettu 4.4.2011.

<http://www.pcmag.com/encyclopedia_term/0,2542,t=application+programming+interface&i=37856,00.asp>

PC Magazine Encyclopedia 1998-2011b. Definition of: RFID tag. Luettu 8.4.2011.

<http://www.pcmag.com/encyclopedia_term/0,2542,t=RFID+tag&i=56955,00.asp>

Rescorla, Eric 2000. Http over TLS. Luettu 4.4.2011.

<<http://www.ietf.org/rfc/rfc2818.txt>>

Tietotekniikan termitalkoot 2001. Hakemistot. Online. Luettu 5.4.2011.

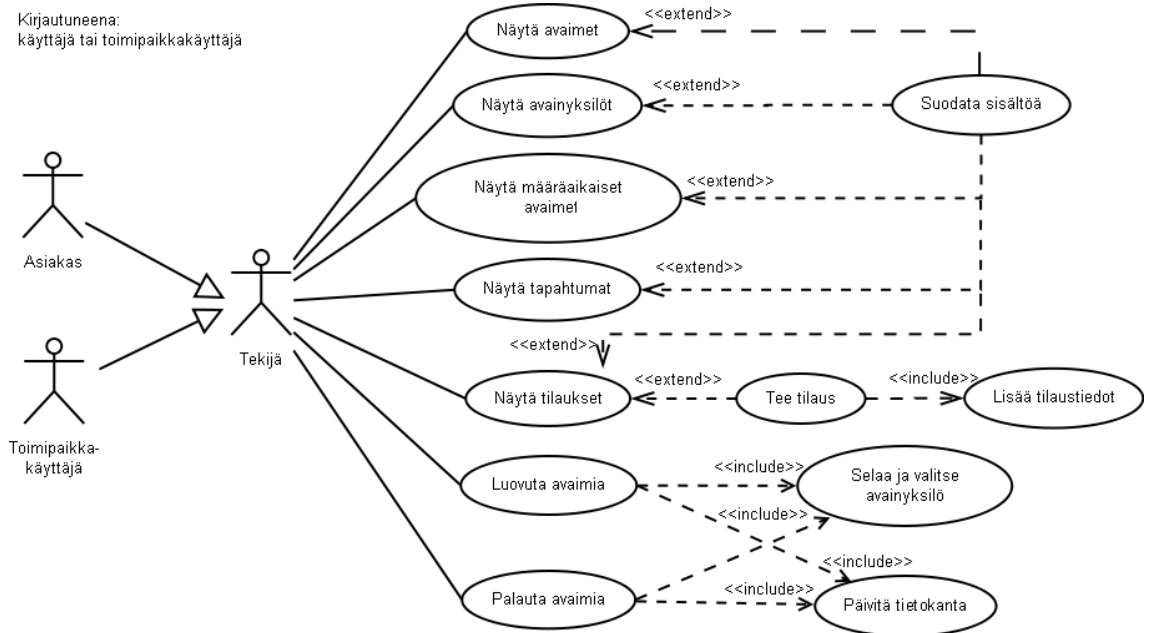
<http://www.tsk.fi/tsk/termitalkoot/hakemistot-267.html?page=get_id&id=ID0144&vocabulary_code=TSKTT>

Turvaykköset 2009-2010. Luettu 19.9.2010.

<<http://turvaykkoset.fi>>

Käyttäjätaso

Avaimet



- Näytä avaimet

Yhteenvedo: Käyttäjä listaa kaikki avaimet kohteesta riippumatta.

Alkuasetelma: Kirjautuneena järjestelmään käyttäjänä.

Kuvaus: Sovellus listaa oletuksena kaikki avaimet. Haussa voidaan hakea millä tahansa hakusanalla mistä tahansa kentästä [*Poikkeama: ei osumia*], kaikille avaimille (=ei suodatusta) on oma nappinsa.

Toiminnot: Klikattaessa mitä tahansa avainriviä hiiren vasemmalla avataan avainkortti (ei muokattavissa)(-*kortilla tarkoitetaan tässä ja jatkossa kyseisen kohteen kaikki tiedot sisältävää popup-ikkunaa*). Klikkaamalla hiiren oikealla tulee tooltip-ikkuna, josta voi valita kyseiselle avaimelle Luovuta, Palauta, Luovuta edelleen, Tila, Suodatus (Kolumneissa nimi, osoite, kohde). (Luovutus/palautus-toiminto luovuttaa kerralla kaikki avaimet esim AS1/1, AS1/2, AS1/3 jne).

Kolumnit: Lukstorivi, Nimi, Leimaus, Varastossa/ Määrä , Varastopaikka, Tyyppi, Osoite, Kohde.

Poikkeamat: Ei osumia – ei näytetä listauksessa sisältörivejä, vain otsikot.

- Näytä avainyksilöt

Yhteenvedo: Käyttäjä listaa kaikki avainyksilöt kohteesta riippumatta.

Alkuasetelma: Kirjautuneena järjestelmään käyttäjänä.

Kuvaus: Sovellus listaa kaikki avainyksilöt. Haussa voidaan hakea millä tahansa hakusanalla mistä tahansa kentästä [*Poikkeama: ei osumia*], kaikille avainyksilöille (=ei suodatusta) on oma nappinsa.

Toiminnot: Klikattaessa mitä tahansa avainriviä hiiren vasemmalla näytetään kyseisen avainyksilön avainyksilökortti (muokattavissa yksilön tila: käytössä/kadonnut/viottunut/poistettu). Avainyksilökortissa näytetään linkki (nappi) avainyksilön tapahtumiin, josta pääsee takaisin avainyksilöön Takaisin-valinnalla. Jos avaimella on haltija ja sitä tai yritys-kolumnia klikataan hiiren vasemmalla, avataan vastaanottajakortti. Klikkaamalla avainrivillä hiiren oikella tulee tooltip-ikkuna, josta voi valita kyseiselle avaimelle Luovuta, Palauta, Luovuta edelleen, Määräaikaiselle avaimelle määräajan jatko, Tila, Suodatus (Kolumneissa avain, osoite, haltija, yritys, kohde).

Kolumnit: Avain, Leima, Haltija, Yritys, Kohde, Osoite, Tyyppi, Tunniste, PalautusPvm.

Poikkeamat: Ei osumia – ei näytetä listauksessa sisältörivejä, vain otsikot.