



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Toni Taivassalo

CMDB

Konfiguraatiohallintatietokanta

Tekniikka ja liikenne
2014

TIIVISTELMÄ

Tekijä	Toni Taivassalo
Opinnäytetyön nimi	Konfiguraatiohallintatietokanta
Vuosi	2014
Kieli	Suomi
Sivumäärä	47 + 1 liite
Ohjaaja	Pirjo Prosi

Opinnäytetyö tehtiin ABB Oy:lle, joka on globaali johtava sähkövoima- ja automaatioteknologiayhtymä. Opinnäytetyön aiheena oli suunnitella tietokanta ja tehdä tietokannalle käyttöliittymä. Tietokannan tarkoitus oli saada kaikki tarvittava työntekijöihin liittyvä tieto saataville samasta paikasta. Eli käytännössä CMDB (configuration management database), konfiguraatiohallintatietokanta.

ABB:llä on käytössä kolmen eri ympäristön palvelimia, kehitys, testi ja tuotanto. Tietokanta luotiin testiympäristöön jossa on käytössä SQL Server 2005. Tietokannan luomiseksi käytettiin SQL Server Management Studio 2008. Tietoja saatiin eri palveluntarjoajilta ja niistä valittiin CMDB:hen sopivat tiedot ja tuotiin ne tietokantaan. Kun tietokantaan saatiin tiedot tuotua, alettiin suunnittelemaan käyttöliittymää. Aluksi käytettiin raportointipalvelinta henkilöiden tietojen hakemiseen jonka oli ABB:n konsultti luonut. Kehitin itse tietokannalle käyttöliittymän käyttäen C#-ohjelmointikieltä ja Visual Studio 2010.

Tietokannasta tuli selkeä ja helppokäyttöinen. Käyttöliittymästä tuli siisti ja siitä näkee nopeasti tarpeelliset tiedot ja myöskin on mahdollista katsoa kattavasti tietoa tietyistä käyttäjistä.

ABSTRACT

Author	Toni Taivassalo
Title	CMDB
Year	2014
Language	Finnish
Pages	47 + 1 Appendice
Name of Supervisor	Pirjo Prosi

This thesis was done for ABB Oy, ABB is a global leader in power and automation technologies. The objective of the thesis was to design a database and user interface. The purpose for the database was to get all the necessary information about employees to one place, in practice to CMDB, configuration management database.

ABB is using three different environment servers, development, test and production. The database was first created in a test environment, which was using SQL server 2005. For the database creation SQL Server Management Studio 2008 was used. Data was gathered from different service providers and the most suitable data was selected to be brought to the database. At first a reporting server was used to search the users from the database, which is really good for creating larger reports. The interface for the database was created using the C# programming language and Visual Studio 2010.

The database was created and it was quite clear and easy to use. The interface is neat and a lot of information can be obtained of a searched person.

KÄYTETYT LYHENTEET

CMDB	Configuration management database, konfiguraatietietokanta joka sisältää kaiken oleellisen tiedon yrityksen IT-infrastruktuurista.
MS SQL	Microsoftin kehittämä SQL-palvelin.
SQL	Structured query language, IBM:n kehittämä kyselykieli relaatiotietokantoihin.
C#	Microsoftin kehittämä ohjelmointikieli.
HPSM	Hewlett-Packard Service Manager.
IDM	Identity management, identiteetin hallintaan käytetty rajapinta.
ER-kaavio	Entity-relationship model, tietokannan tietomalli.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	KUVIO- JA TAULUKKOLUETTELO	7
2	JOHDANTO.....	9
3	ABB OY	10
4	TYÖN TARKOITUS	11
5	TYÖVÄLINEET	12
	5.1 SQL.....	12
	5.2 MS SQL Server.....	12
	5.3 Microsoft server management studio.....	12
	5.4 C# 12	
	5.5 Visual Studio.....	12
6	TIETOKANNAT.....	13
	6.1 Mikä on tietokanta?.....	13
	6.2 Erityyppiset tietokannat	13
	6.3 Tietomallit.....	13
	6.4 Tietokannan suunnittelu.....	13
	6.4.1 Suunnittelun merkitys	13
	6.4.2 Tietokannan suunnittelun tavoitteet	14
	6.4.3 Tietokannan suunnittelun vaiheet	14
7	DATAN KERUU	16
	7.1 Vaatimusmäärittely	18
	7.2 Tietojen siirron suunnittelu	18
	7.2.1 Henkilön tiedot.....	18
	7.2.2 Työasemat	20
	7.2.3 Puhelimet ja liittymät	21
	7.2.4 Keikat	22
8	TIETOKANNAN FYYSINEN SUUNNITTELU JA LUONTI	24
	8.1 Tietokannan asetusten määrittely.....	24
	8.2 Taulujen luonti	25

8.2.1	ER-kaavio.....	25
8.2.2	PersonData	26
8.2.3	Workstation	27
8.2.4	Mobile	28
8.2.5	EppDevice	29
8.2.6	Incident.....	30
8.2.7	Interraction	31
8.3	Tietojen tuonti Excel-taulukosta	31
8.4	Tietojen päivittäminen	32
8.5	Näkymät	32
9	KÄYTTÖLIITTYMÄ	33
9.1	Määrittely.....	33
9.2	Suunnittelu	33
9.3	Toteutus.....	34
9.3.1	DataGrid	34
9.3.2	Search kenttä	34
9.3.3	Välilehdet	35
9.3.4	Kirjautuminen	38
9.3.5	Admin-näkymä.....	40
9.3.6	Näkymät	42
9.4	Testaus	44
10	JOHTOPÄÄTÖKSET JA POHDINTA	46
	LÄHTEET.....	47
	LIITE.....	48

1 KUVIO- JA TAULUKKOLUETTELO

Kuva 1.	Kuva tunnuksestani IDM:ssä	s. 19
Kuva 2.	Tietokannan luonti	s. 24
Kuva 3.	Tietokannan luonti	s.25
Kuva 4.	ER-kaavio	s.26
Kuva 5.	PersonData-taulu	s.27
Kuva 6.	Workstation-taulu	s.28
Kuva 7.	Mobile-taulu	s.29
Kuva 8.	EppDevice-taulu	s.29
Kuva 9.	Incident-taulu	s.30
Kuva 10.	Interraction-taulu	s.31
Kuva 11.	Käyttötapakaavio	s.33
Kuva 12.	DataGrid ja hakukenttä	s.35
Kuva 13.	Henkilön perustiedot	s.35
Kuva 14.	Henkilön työaseman tiedot	s.36
Kuva 15.	Työaseman admin-tunnusten haku	s.36
Kuva 16.	Henkilön puhelin ja liittymätiedot	s.37
Kuva 17.	Henkilön tunnustiedot SAPista	s.37
Kuva 18.	Henkilön keikat HPSM-keikkajärjestelmästä	s.38
Kuva 19.	Login-taulu tietokannassa	s.39

Kuva 20.	Login-taulu	s.40
Kuva 21.	Admin-näkymä	s.41
Kuva 22.	Uuden tunnuksen luonti	s.41
Kuva 23.	UAM-roolin näkymä	s.42
Kuva 24.	SD-näkymä	s.43
Kuva 25.	WS-näkymä	s.44

2 JOHDANTO

Opinnäytetyö tehtiin ABB Oy:lle Vaasassa, ABB tarvitsi työkalun, josta voisi nähdä kaikki henkilöihin liittyvät tiedot samasta tietokannasta yhden työkalun kautta. Kaikki tieto oli kyllä saatavilla, mutta tietoa haetaan monella eri työkalulla monesta eri tietokannasta sekä joissain paikoissa tiedot ovat puutteellisia. Täten päädyttiin siihen tulokseen, että olisi tarve kehittää konfiguraatiohallintatietokanta (CMDB).

Alunperin oli tarkoituksena, että käyttöliittymä tilataan ulkopuoliselta yritykseltä. Osana opinnäytetyötä kuitenkin kehitettiin tietokannalle käyttöliittymä, jolla voidaan suorittaa hakuja tietokannasta. Käyttöliittymä ohjelmoitiin C#-ohjelmointikielellä

Konfiguraatiohallintatietokannassa on yleensä kuvattu kaikki IT:hen liittyvät komponentit ja niiden suhteet. Konfiguraatiohallintatietokanta on tärkeä osa isojen yritysten IT:tä, koska se helpottaa ylläpitoa ja seuraamista. Tässä työssä oli tarkoitus lähteä liikkeelle siitä, että saadaan ensisijaisesti kaikki henkilöihin liittyvät tiedot hyvin esille, joita vasten voi sitten tehdä hakuja. Myöhemmin lisätään esim. serveri- ja IP-tasoisia tietoja.

Tietokanta luodaan virtuaalipalvelimelle jossa on käytössä Microsoft SQL Server 2005. Tietokanta luodaan käyttäen Microsoft Server Management Studio 2008. Käyttöliittymä luodaan Visual Studio 2010 käyttäen.

3 ABB OY

ABB on perustettu tammikuussa 1988 yhdistämällä ruotsalaisen Asean ja sveitsiläisen Brown Boverin sähkötekniiset liiketoiminnot. Nykyään ABB on johtava sähkövoima- ja automaatioteknologiayhtymä, joka toimii yli 100 maassa. Suomessa ABB toimii 30 paikkakunnalla. ABB:llä on viisi eri divisioonaa: sähkövoimatuotteet, sähkövoimajärjestelmät, sähkökäytöt ja kappaletavara-automaatio, pienjännitetuotteet ja prosessiautomaatio./1/

ABB on yksi Suomen suurimmista teollisista työnantajista, pääkaupunkiseudulla suurin. ABB työllistää globaalisti yli 145 000 henkilöä, joista Suomessa on noin 6 600. Tehdaskeskittymät Suomessa ABB:lla sijaitsee Helsingissä, Vaasassa sekä Porvoossa. /1/

4 TYÖN TARKOITUS

Työn tavoitteena on saada ensisijaisesti service deskin tiimille helppokäyttöinen työkalu, jota he voivat käyttää samalla, kun ottavat ongelmatapauksia vastaan puhelimitse. Tällä hetkellä tietoa haetaan monesta eri paikasta usean eri ohjelman avulla. Puheluiden yhteydessä useaan eri ohjelmaan hakujen tekeminen vie liikaa aikaa.

Suunnitellaan ja luodaan tietokanta, johon kasataan tiedot ABB:n työntekijään liittyen, kuten organisaatio-, työasema-, puhelin-, tunnus- ja keikkatiedot. Tietokannalle tarvittaisiin käyttöliittymä, josta saisi yhdellä haulilla haettua tiedot tietystä henkilöstä esille ja nähtäisiin kaikki oleellinen yhdellä silmäyksellä. Tämä parantaisi service deskin tehokkuutta ja kommunikointia, koska he voisivat puheluiden yhteydessä jo nopeasti katsoa henkilön tietoja, liittyen laitteisiin ja tunnuksiin.

5 TYÖVÄLINEET

5.1 SQL

SQL eli Structured Query Language on lähes kaikkien relaatiotietokantatuotteiden tukema tietokannan määrittely- ja käsittelykieli. SQL-kielillä tehdään taulujen määrittely, tietojen haku, päivitys ja poisto sekä valtuusmääritykset ja tapahtumankäsittelyn ohjausta. SQL:ää voidaan käyttää vuorovaikutteisesti työasemalla tai upotettuna ohjelmointikieleen./3/

5.2 MS SQL Server

Microsoft SQL Server on relaatiotietokannan hallintajärjestelmä jonka on kehittänyt Microsoft.

5.3 Microsoft server management studio

Microsoftin kehittämä tuote, jota käytetään MS SQL Serverin hallintaan.

5.4 C#

C# on .NET-konseptia varten kehitetty ohjelmointikieli, jossa on yhdistetty C++ tehokkuus ja Java-kielen helppokäyttöisyys. Kehittäjä on Microsoft.

5.5 Visual Studio

Microsoftin ohjelmankehitysympäristö, jossa voi käyttää useita ohjelmakieliä. Visual Studiolla voidaan kehittää esimerkiksi Windows-, web- ja mobiilisovelluksia.

6 TIETOKANNAT

6.1 Mikä on tietokanta?

Tietokanta on loogisesti yhteenkuuluvien tallennettujen tietojen joukko, jota voidaan helposti käsitellä tietokantakielellä, kuten SQL. Tietokannassa olevia tietoja hallinnoi erityinen ohjelmisto, tietokannan hallintajärjestelmä eli TKHJ (Database Management System, DBMS). Tunnettuja esimerkkejä näistä ovat Oracle, DB2, Microsoft SQL Server, MySQL ja Access. /3/

6.2 Erityyppiset tietokannat

Aiemmin käytettiin rakenteeltaan verkkomallisia ja hierarkkisia tietokannan hallinta järjestelmiä. Nykyään käytetään valtaosin SQL-pohjaisia relaatiotietokantoja. Relaatiotietokannat ovat helpompia käyttää ja muuttaa kuin perinteisemmät hierarkkiset ja verkkomalliset tietokannat. On myös olemassa oliotietokantoja. Ne eivät rajoitu perinteisiin tietotyyppeihin tai kyselykieliin, vaan niissä luodaan monimutkaisia olioita ja niille sovellettavia operaatioita. Oliokannat soveltuvat hyvin olio-ohjelmoinnin jatkeeksi. /3/

6.3 Tietomallit

Tietomalli on kuvausmenetelmä ja joukko sääntöjä, jolla voidaan kuvata tietorakenteita ja niiden välisiä yhteyksiä. Tietomalleja ovat käsitemalli, oliomalli ja relaatiomalli. /3/

6.4 Tietokannan suunnittelu

6.4.1 Suunnittelun merkitys

Tietokantojen suunnittelu on tärkeää monesta syystä. Nykyään yhä suurempi osa tiedoista tallennetaan tietokantoihin on tärkeää, että tiedot voidaan esittää tiedon hakijalle nopeasti ja varmasti. Tietokanta muodostaa nykyaikaisten sovellusten perustan. Vaikka käyttäjälle näkyy lähinnä vain käyttöliittymä ja raportit, on tärkeää suunnitella ja rakentaa käytettävä tietokanta hyvin. Hyvin suunniteltu tietokanta helpottaa ohjelmointia. Tietokannan suunnittelu käsittää muutakin kuin

tietokannan mallinnuksen. Tietokannan mallinnus on pääasiassa tietokannan kuvaamista jollakin kuvaustekniikalla, kun taas tietokannan suunnittelu käsittää laajan kirjon asioita vaatimusmäärittelystä tietokannan mallinnukseen ja fyysiseen suunnitteluun. /3/

6.4.2 Tietokannan suunnittelun tavoitteet

Hyvän tietokannan rakenteen keskeisiä ominaisuuksia ovat:

- Kattavuus: sisältää kaikki järjestelmissä tai kyselyissä tarvittavat tiedot ja yhteydet
- Selkeys ja ymmärrettävyys: yksinkertainen rakenne
- Muutosjoustavuus: laajennettavuus minimoiden nykyisten ohjelmien muutokset
- Yleiskäyttöisyys: soveltuvuus erilaisiin ympäristöihin ja eri asiakkaille tarvitsematta muuttaa tietokannan rakennetta
- Ohjelmointimukavuus: selkeät tietorakenteet, sarakkeilla kiinteä merkitys, sarakkeen merkitys ei saisi riippua toisesta sarakkeesta
- Tehokkuus: riittävä vastausaika tapahtumille ja riittävän tehokkaat eräajot./3/

6.4.3 Tietokannan suunnittelun vaiheet

Tietokannan suunnittelu koostuu tietyistä vaiheista. Joskus jotkin vaiheet ovat hyvin nopeita tai jopa ohitettavia, mutta kannattaa joka tapauksessa miettiä, onko tietokantasuunnittelussa otettu nämä huomioon.

Vaiheisiin kuuluu käsiteanalyysi, tarveanalyysi, normalisointi, tietokannan toteutus, suorituskyvyn viritys.

Käsiteanalyysissä suunnitellaan tietokantaa loogisella tasolla ja muodostetaan käsitemalli. Käsiteanalyysillä mallinnetaan kohdealuetta ensin melko karkealla tasolla ja sitten tarkentaen. Laaditaan myös käsitemallirunko, joka kuvaa kohdealueen käsitteet, mutta ei vielä kaikkia tietoja.

Tietotarveanalyysissä käsitelmää tarkennetaan sovelluksen tietotarpeiden perusteella ja lisätään kaikki puuttuva tieto.

Normalisointivaiheessa tarkistetaan ettei mallissa ole turhaa toisteisuutta.

Normalisoinnin jälkeen siirrytään tuotekohtaiseen suunnitteluun, jolloin on oltava tiedossa tietokannan hallintajärjestelmä. Muunnetaan käsitelmä tiettyjen sääntöjen mukaan taulukaavioksi ja samalla määritellään perusindeksointi. Tässä fyysiseen suunnitteluun kuuluvassa vaiheessa voidaan lisäksi määrittää mm. taulujen tilavaraukset, sijoittelut ja mahdollinen osiointi. Lopuksi viritetään suorituskyky kohdalleen, mikä tarkoittaa lähinnä indeksointia./3/

7 DATAN KERUU

Pidimme esimiehen ja Notkia IT:ssä työskentelevän konsultin, jolla on aiempia kokemuksia CMDB:n luonnista, kanssa palaverin CMDB-projektin aloituksesta. Palaverissa mietittiin, mitä tietoja olisi järkevä alkaa kasaamaan tietokantaan ja yhteyshenkilöitä, joihin olla yhteydessä ja selvittää mistä datan voisi saada käyttöön ja millä tavoin.

Konsultilla oli hyviä näkemyksiä mitä CMDB:ssä tulisi olla, ja kunnollisessa CMDB:ssä olisi lopulta koko firman IT-infrastruktuuri mallinnettuna. Mutta päädyimme siihen, että aluksi tarkoituksena on saada jokaisen henkilön tiedot kantaan, koska ensimmäisenä on tarkoitus saada service deskin porukalle käyttöön sovellus, jolla he voivat puheluiden yhteydessä nopeasti tarkastella yhdestä paikasta kaikki olennainen henkilöön liittyvä data, tarvitsematta kysellä sitä henkilöltä suoraan. Seuraavia tietoja saatiin kasattua, jotka olisi tarpeellista tietää ja saatiin selville myös joka osa-alueen yhteyshenkilö, jonka kautta voi aloittaa tietojen kasaamisen:

Henkilön perustiedot:

- käyttäjätiedot
- kustannuspaikkatiedot
- organisaatiotiedot
- paikkatiedot
- esimies ja esimiehen henkilönnumero.

Tunnustiedot:

- tunnukset
 - SAP(ympäristöt ESP, EST, ESD, BP1, BT1, BD1)
 - AD
 - sähköposti.

Henkilön perustiedot ja tunnustiedot saadaan suoraan IDM:n tietokannasta. AD-tunnusta lukuunottamatta, joka kannattaa hakea suoraan AD:sta.

Työasemat:

- laitetunnus
- omistaja
- laitetyyppi
- hankinta-/sopimuspäivämäärä
- sarjanumero
- muisti
- kustannuspaikkatiedot
- kuka kirjautunut viimeksi
- merkki
- käyttöjärjestelmä.

Työasematiedot löytyvät Webshopista sekä Snowsta, saatiin yhteyshenkilö, joka on Snow-työkalun ylläpitäjä, johon voidaan olla yhteydessä ja kysellä saisiko oikeudet Snow-työkaluun tai jopa suoraan tietokantaan.

Puhelimet:

- malli
- käyttäjä
- imei
- sarjanumero
- lainan alku- ja loppupäivämäärä.

Liittymä:

- puhelinnumero
- omistaja
- mobiili data
- yhteystyyppi
- sim kortin numero
- laskutusnumero ja osoite
- kustannuspaikka

- status.

Puhelin- ja liittymätiedot tulevat Elisan kautta ja ABB:n puhelinpalvelun esimiehen voidaan olla yhteydessä, ja löytää tapa saada tiedot puhelimista ja liittymistä.

HPSM keikat:

- keikkanumero
- kategoria
- kuka avannut
- status
- kenelle osoitettu
- kuka sulkenut
- sulkemisaika.

HPSM-tietokantaan on mahdollista saada suoraan yhteys, pitää jutella asiasta yhteyshenkilön kanssa ja pyytää tunnukset tietokantaan.

7.1 Vaatimusmäärittely

Tietokanta tulee suunnitella selkeäksi, yksinkertaisella rakenteella. Selkeät tietorakenteet sekä sarakkeiden kiinteä merkitys takaa että ohjelmointi helpottuu käytettäessä tietokantaa. Tehokkuus on myös tärkeä, että saadaan haut tehtyä kantaa vastaan tarpeeksi nopeasti. Tietokannan olisi myös hyvä olla sellainen, että kun sen kerran suunnittelee, voi sitä käyttää jatkossa myös eri ympäristöissä, ilman että tietokannan rakennetta tarvitsee muuttaa.

7.2 Tietojen siirron suunnittelu

7.2.1 Henkilön tiedot

Henkilöiden tiedot, kuten nimi-, organisaatio-, esimies- ja SAP-tunnustiedot tuodaan suoraan IDM:n tietokannasta. IDM-lyhenne tulee identity managementista, joka on identiteetin hallintaan käytettävä sovellus. IDM:ää käytetään rajapintana

esim. AD-tunnuksiin sekä SAP-tunnuksiin, ja sitä kautta voidaan hallita esimerkiksi hyväksyntäketjuja roolien tai ryhmien hyväksymiseen.

Jokainen ABB:lle tuleva työntekijä löytyy IDM:stä (**Kuva 1.**).

The screenshot displays a user profile for 'Taivassalo Toni' in an IDM system. The profile is organized into several sections: 'General Attributes', 'Account Attributes', 'Assigned Roles', 'Assigned Privileges', 'Assigned Groups', and 'Settings'. The 'General Attributes' section includes fields for Unique ID, Display Name, First Name, Last Name, Primary E-Mail, Personnel Area, Personnel Area Text, Cost Center, Cost Center Text, Personnel Number, and ABB GED Global Employee Number. The 'Account Attributes' section includes fields for Valid From, Valid Through, Primary Telephone Number, Primary Mobile Number, and Department. The 'Manager' section at the bottom shows 'SAARIKKO KLAUS' as the manager.

Kuva 1. Kuva tunnuksestani IDM:ssä.

IDM:n tietokantaan saatiin oikeudet, jotta sinne voidaan ottaa yhteys suoraan ja hakea sieltä SQL-haulla tarvittavat tiedot. IDM:n taulurakenne on todella sekava ja sieltä tietoa hakeakseen joutuu käyttämään väliaikaisia tauluja haussa, jotta sieltä saa järkevästi tietoa ulos.

Otetaan henkilön tiedoista oleellimmat sarakkeet, kuten:

- etunimi
- sukunimi
- henkilönumero
- sähköposti
- yksikkö
- kustannuspaikka
- esimiehen nimi
- esimiehen henkilönumero
- AD-tunnus
- SAP-tunnus.

7.2.2 Työasemat

Työasemien tiedot löytyvät Snow-työkalusta, joka on käyttäjäystävällinen SAM(software asset manager)-työkalu lisenssien ja laitteiden hallintaan. Snow mahdollistaa ohjelmistojen, työasemien sekä sopimusten hallinnan. Lisäksi siinä on mukana raportointityökalu, jolla voi luoda raportteja./2/

Snow-raportointityökalu on tosi helppo käyttää ja sieltä saa vaivatta ajettua raportteja ulos. Mukaan otettavia sarakkeita voi valita valmiina olevasta listasta. Työkalusta saa ajastettua ajon, joka tietynä aikana ajaa raportin ja lähettää sen määritettyyn sähköpostiosoitteeseen. Raportti tulee Excel-muodossa.

Saatiin opinnäytetyötä varten oikeudet Snow-raportointityökaluun, josta pääsee ajamaan raportteja työasemiin liittyen. Suoraan tietokantaan ei saada yhteyttä, joten raportti pitää ajaa aina Snow-työkalun kautta. Valittiin seuraavat sarakkeet työasemiin liittyen:

- tietokoneen tunnus
- organisaatio
- kaupunki
- asset type
- main type
- valmistaja
- tietokoneen malli
- sarjanumero
- käyttöjärjestelmä
- arkkitehtuuri
- muisti
- käyttäjän nimi
- käyttäjän ID
- viimeksi logannut käyttäjä
- lainan alku- ja loppupäivä.

7.2.3 Puhelimet ja liittymät

Puhelin- ja liittymätietoja hallitsee Elisa, ne tiedot on saatavilla Elisan omasta raportointityökalusta. Sieltä löytyy valmiiksi määritellyt raportit, josta saa ulos tiedot puhelimista ja liittymistä, molemmista on oma raportti.

Raportit saa ajettua ulos Excel-muodossa, mutta raporteja ei voi muokata ennen niiden ajamista, vaan pitää sitten Excelissä tehdä muokkaukset raporteihin. Raporteja ei ole mahdollista ajastaa, vaan ne pitää aina käydä ajamassa ja tallentamassa Elisan raportointijärjestelmän kautta.

Raportti liittymistä sisältää paljon tietoa jotka eivät ole tarpeellisia CMDB:tä varten, joten raporttia joutuu karsimaan aika paljon. Valitaan raportista seuraavat sarakkeet tietokantaan:

- henkilönumero
- omistajan nimi
- puhelinnumero
- yhteystyyppi
- mobiili data
- sim kortin numero
- laskutus numero
- laskutusosoite
- kustannuspaikka
- status.

Raportti puhelimista sisältää valmiiksi jo kaiken aika oleellisen. Valitaan seuraavat sarakkeet tietokantaan:

- puhelinnumero
- sopimus
- omistajan nimi
- laite
- IMEI

- vuokran alkupäivämäärä
- vuokran loppumispäivämäärä
- laskutusosoite
- kustannuspaikka.

7.2.4 Keikat

Keikkoja hallitaan ABB:llä HP(Hewlett-Packard) service managerilla eli HPSM:llä. HPSM on keikkojen hallintajärjestelmä, joka mahdollistaa organisaation kaikkien keikkojen koordinoinnin, ylläpidon ja toteutuksen yhdessä järjestelmässä. HPSM:n kautta keikkoja on helppo hallinnoida ja seurata niiden kulkua.

Keikkoja voi luoda HPSM:n itsepalvelun kautta, johon ABB:läisillä on oikeudet ja sitä kautta osoittaa keikka oikealle ratkaisuryhmälle. Keikat voi myös toimittaa service deskin kautta, jotka ottavat keikkoja vastaan puhelimitse ja sähköpostitse ja sitten luovat keikat HPSM-järjestelmään oikealle ratkaisija ryhmälle.

Saatiin lukuoikeus HPSM:n tietokantaan, jossa on kaikki ABB:n tiedot. Tietokanta on todella sekava, taulut on nimetty ihmeellisellä logiikalla, ja tauluja on satoja. Tarvittavat tiedot löydettiin tauluista PROBSUMMARYM1 ja INCIDENTSM1. Saadaan suoraan yhteys CMDB-tietokannasta näihin tauluihin ja haettua sieltä seuraavat tiedot incidenteistä ja interratioista:

Valitaan tarvittavat incidentteihin liittyvät sarakkeet PROBSUMMARYM1-taulusta:

- incident numero
- interratio numero
- kategoria
- kuka avannut keikan
- kenelle keikka osoitettu
- keikan tila
- sulkemisaika
- keikan sulkija

- sulkukoodi
- toteuttajan nimi.

Valitaan tarvittavat interractioneihin liittyvät sarakkeet INCIDENTSM1-taulusta:

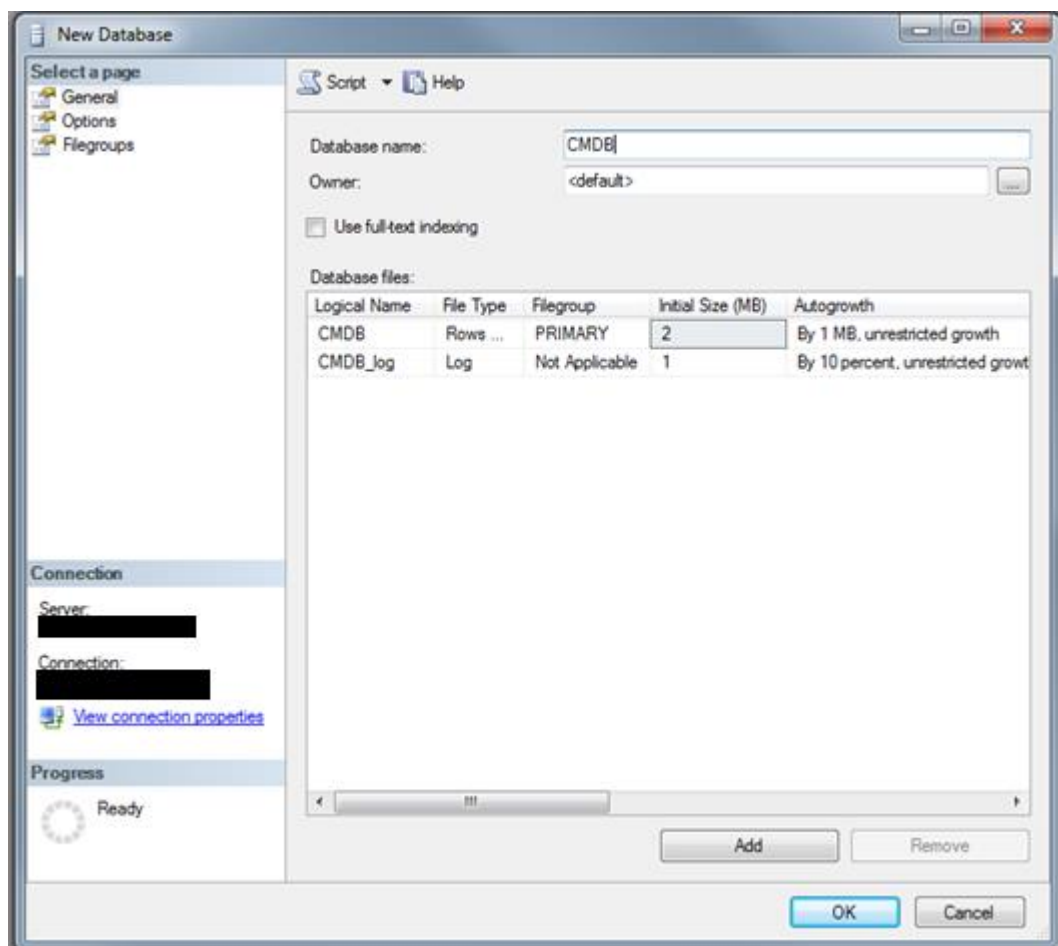
- interraction numero
- keikan kontakti henkilö
- interraction avausaika
- interraction avaaja
- ryhmä johon interraction liittyy
- interraction omistajan nimi
- sulkemiskoodi
- sulkemisaika
- kategoria.

8 TIETOKANNAN FYYSINEN SUUNNITTELU JA LUONTI

Tietokannan luontiin käytetään Microsoftin SQL Server Management Studio 2008. Otetaan yhteys serveriin, ja sitten luoda tietokanta. Asetetaan tietokannan nimeksi CMDB. Serveri jolle tietokanta luotiin käyttää SQL Server 2005:sta.

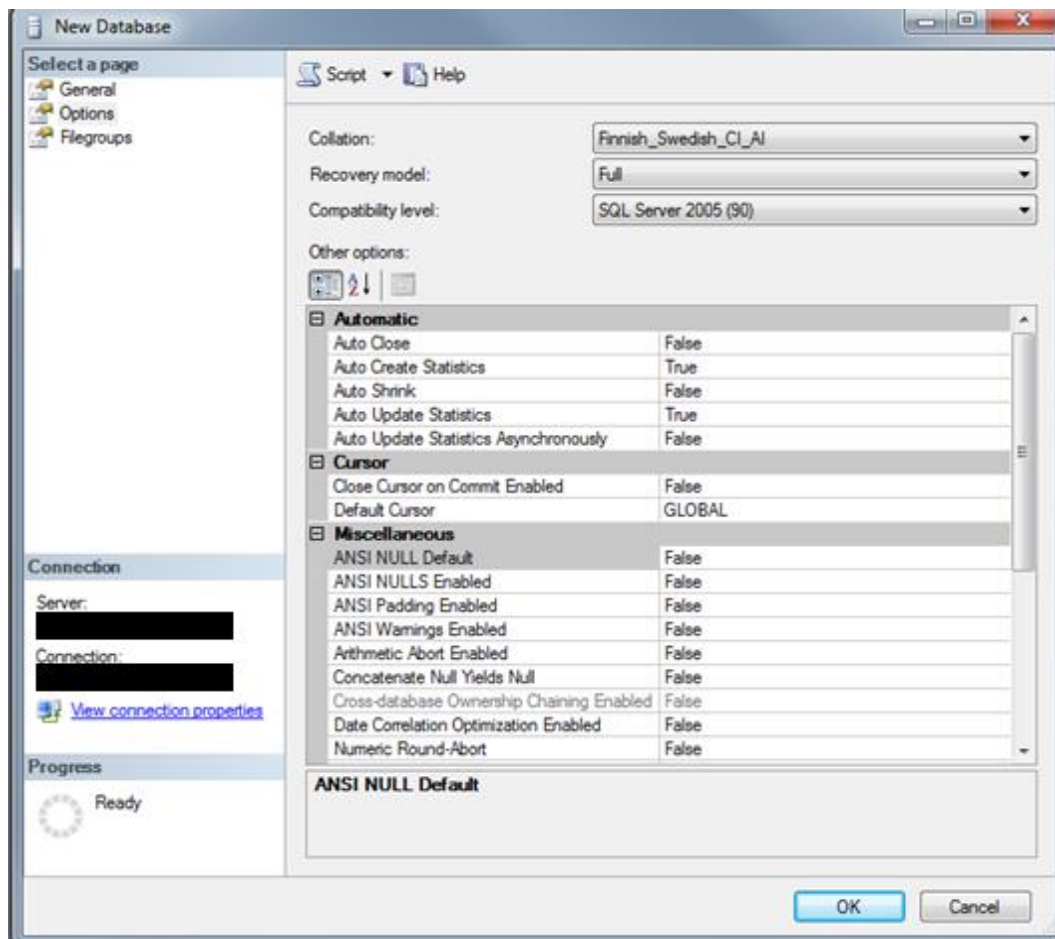
8.1 Tietokannan asetusten määrittely

Luodaan tietokanta, valitaan New Database painike ja asetetaan tietokannalle nimi, tässä tapauksessa valittiin nimeksi CMDB. Omistajaksi tulee automaattisesti AD-tunnus joka tietokannan luo (**Kuva 2.**).



Kuva 2. Tietokannan luonti.

Mennään seuraavaksi “Options”-välilehdelle ja valitaan sieltä collationiksi Finnish_Swedish_CI_AI. Tämä collaatio tarkoittaa sitä, että pohjoismaiset erikoismerkit näkyvät oikein tietokannassa (**Kuva 3.**).



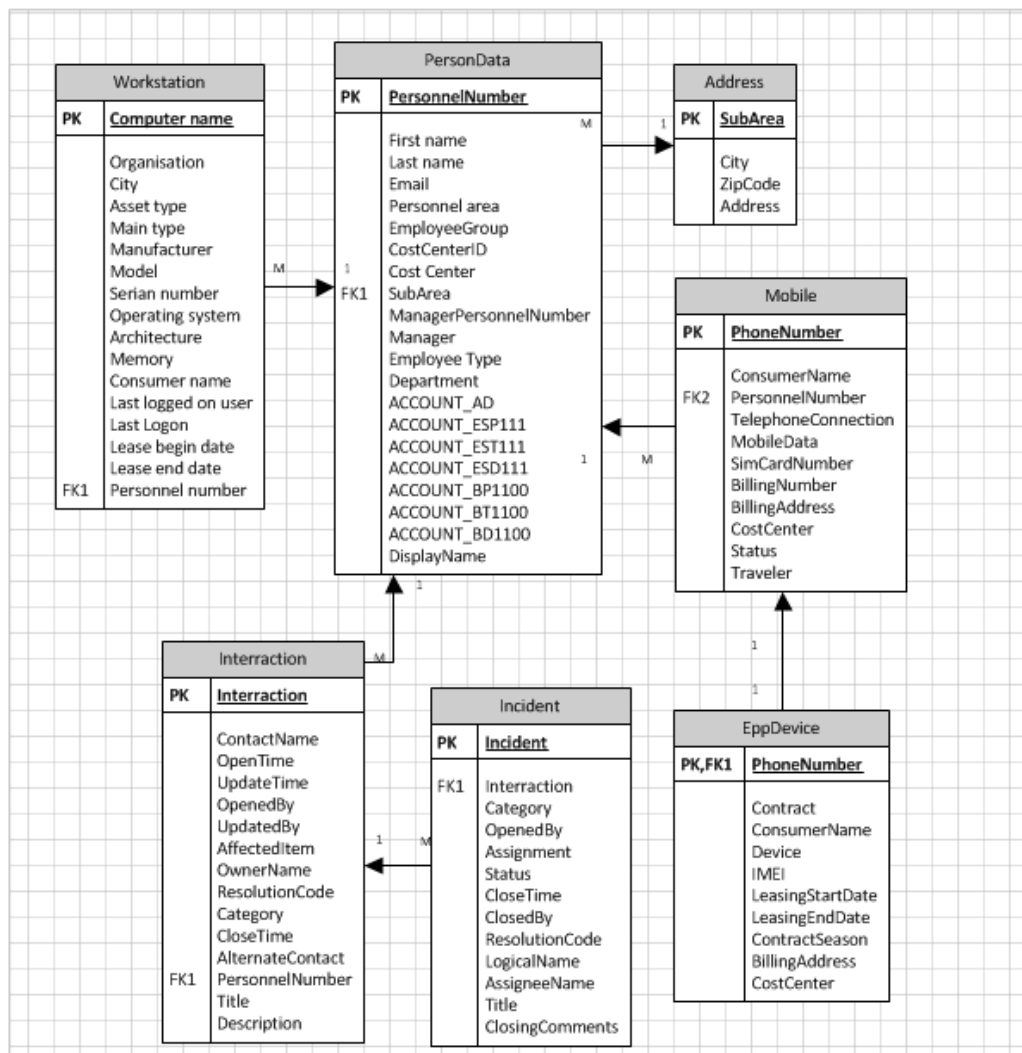
Kuva 3. Tietokannan luonti.

8.2 Taulujen luonti

Mietittiin taulun rakenne, sekä jaettiin eri paikoista saatavat tietueet omiin tauluihin. Jokainen tauluista on yhdistettävissä toiseen tauluun jonkun sarakkeen avulla.

8.2.1 ER-kaavio

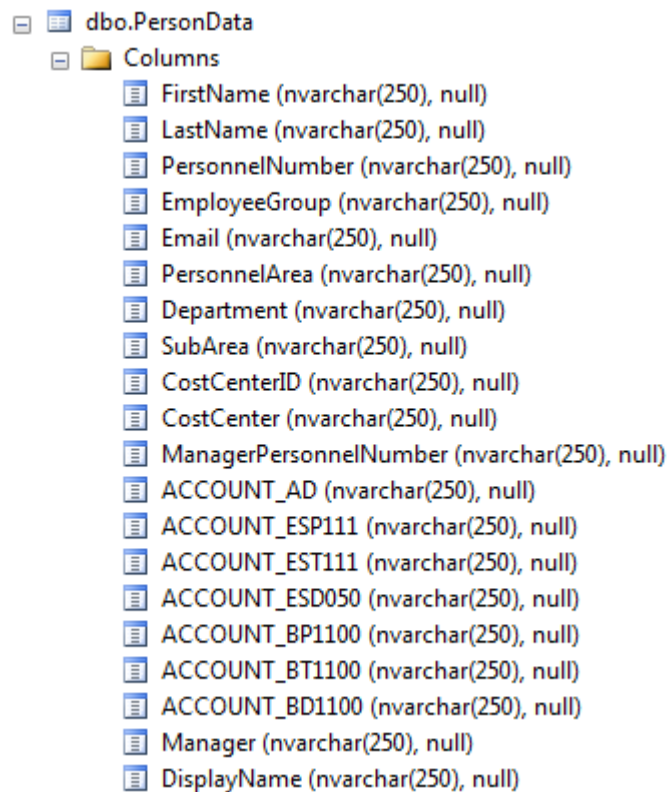
Luotiin ER-kaavio sen perusteella, mitä dataa on saatavilla ja mikä niistä on tähän tarkoitukseen oleellista (**Kuva 4.**).



Kuva 4. ER-kaavio

8.2.2 PersonData

Luotiin taulu ABB:llä työskentelevien henkilöiden perustiedoista ja tunnuksista joita heillä on käytössä (**Kuva 5.**):

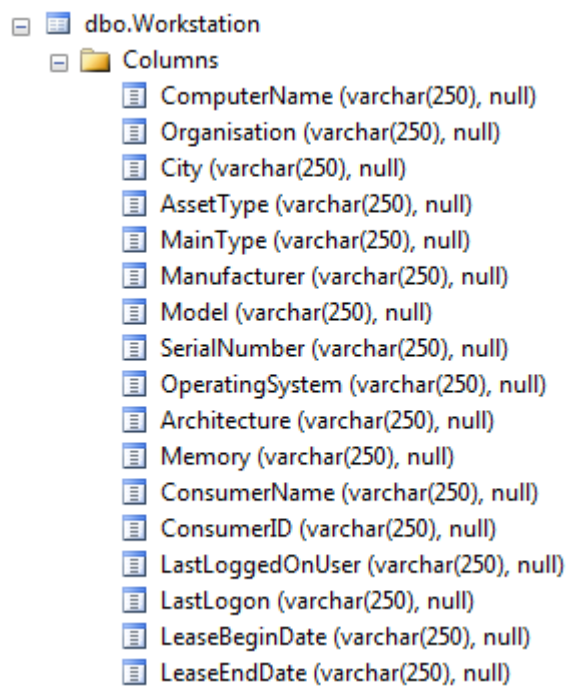


Kuva 5. PersonData-taulu.

Taulun tiedot tuodaan IDM:n tietokannasta SQL-lauseella. Tuotavat tietueet mietittiin aiemmin ja koottiin seuraava lause, joka tuo tarvittavat tiedot PersonData-tauluun. Kts. Liite 1. PersonData-taulun SQL-lause.

8.2.3 Workstation

Työasematiedot tuodaan Snow-työkalulla saatavasta raportista, joka listaa kaikki Suomen ABB:n työntekijöillä käytössä olevat työasemat ja niiden tiedot. Raporttiin karsittiin tarvittavat tiedot ja luotiin Excel-sarakkeita vastaava taulu tietokantaan (**Kuva 6.**).



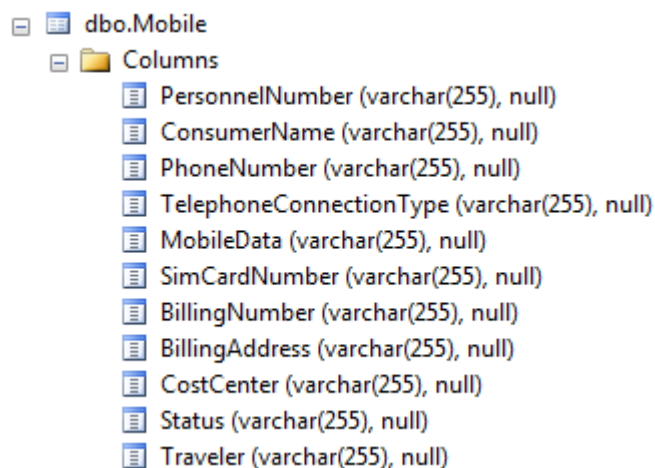
Kuva 6. Workstation-taulu.

Tiedot työasemista saadaan ulos .xls-tiedosto Snow-työkalusta ja tiedot voidaan tuoda tietokantaan .csv-tiedostona.

Taulussa on henkilönnumero tieto saatavilla, joten tiedot on helppo yhdistää PersonData-tauluun.

8.2.4 Mobile

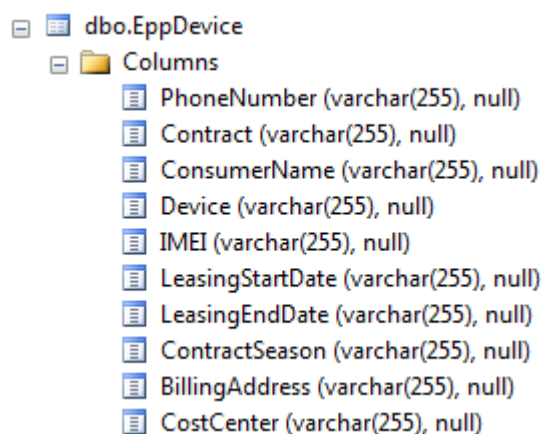
Puhelinliittymätiedot saadaan Elisan kautta valmiilla raportilla ajettua ulos, raportti tulee .xls-tiedostona, ja sitä ei voi ennalta muuttaa. Raporttiin tulee kaikki mahdolliset tiedot mukaan ja sitä pitää sitten Excelillä muokata. Karsitaan tietoja pois raportista ja jätetään aiemmin mietityt tietueet ja luodaan niiden mukaisesti taulu tietokantaan (**Kuva 7.**).



Kuva 7. Mobile-taulu.

8.2.5 EppDevice

Mobiililaitetiedot saadaan myös Elisa-raportointityökalun kautta. Kuten liittymäraporttissakin, niin myös mobiililaitteille on ennalta määritelty raportti jota ei voi muokata ennen kuin sen ajaa ulos, joten muutokset raportille pitää tehdä joka kerta manuaalisesti. Raportti tulee .xls-muodossa. Muokataan raportti ajon jälkeen ja jätetään siihen oleelliset tiedot (**Kuva 8.**).



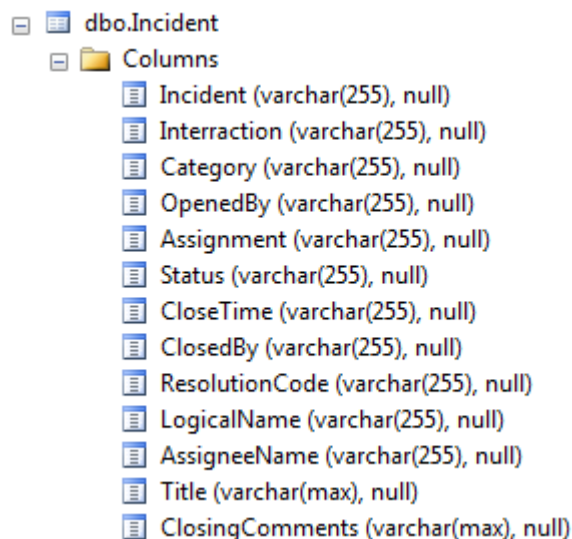
Kuva 8. EppDevice-taulu.

8.2.6 Incident

Incident tiedot voidaan hakea suoraan HPSM:n tietokannasta, koska siihen saatiin lukuoikeudet. Oikeiden tietojen löytäminen oli kuitenkin vaikeaa, koska tietokannassa on todella paljon tauluja ja niiden nimeämislogiikka on outo. Lopulta selvisi, että tarvittavat tiedot löytyvät taulusta PROBSUMMARYM1. Taulussa on paljon sarakkeita ja niistä suurin osa on CMDB:n tarkoitukseen turhia. Luotiin haku tietokantaan, jolla haetaan tarvittavat tiedot ja tuodaan ne omaan incident tauluun CMDB-tietokannassa. Haku on seuraava:

```
INSERT INTO CMDB.DBO.Incident(Incident, Interraction, Category, OpenedBy, Assignment,
Status, CloseTime, ClosedBy, ResolutionCode, LogicalName, AssigneeName, Title, Clos-
ingComments)
select NUMBER, INCIDENT_ID,CATEGORY, OPENED_BY, ASSIGNMENT, STATUS,
CLOSE_TIME, CLOSED_BY, RESOLUTION_CODE, LOGICAL_NAME, AS-
SIGNEE_NAME, BRIEF_DESCRIPTION, CLOSING_COMMENTS from
HPSM.SM70Prod.dbo.PROBSUMMARYM1
where ASSIGNMENT like 'NEU FI%'
```

Ja luodaan taulu vastaamaan hakua (**Kuva 9.**).



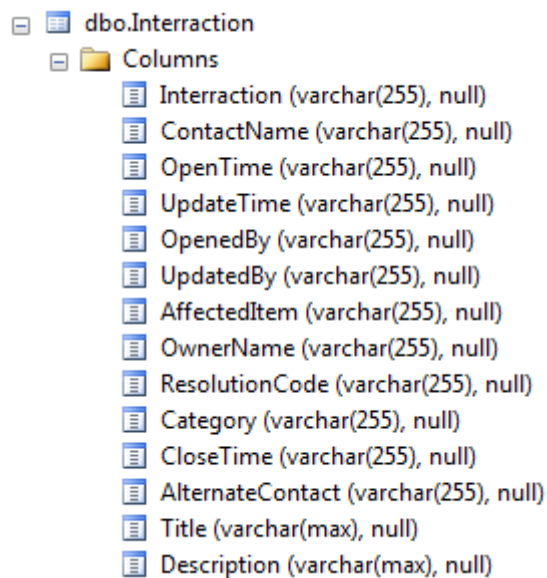
Kuva 9. Incident-taulu.

8.2.7 Interraction

Interraction tauluun saadaan myös tiedot HPSM:n tietokannasta. Tiedot löytyivät taulusta INCIDENTSM1. Mietittiin tarvittavat tiedot taulusta ja luotiin haku, jolla saadaan tuotua tiedot omaan tauluun:

```
INSERT INTO CMDB.DBO.Interraction(Interraction, ContactName, OpenTime, UpdateTime,
OpenedBy, UpdatedBy, AffectedItem, OwnerName, ResolutionCode, Category, CloseTime, Al-
ternateContact, Title, Description)
SELECT INCIDENT_ID, CONTACT_NAME, OPEN_TIME, UPDATE_TIME,
OPENED_BY, UPDATED_BY, AFFECTED_ITEM, OWNER_NAME, RESOLU-
TION_CODE, CATEGORY, CLOSE_TIME, ALTERNATE_CONTACT, TITLE, DESCRIP-
TION from HPSM.SM70Prod.dbo.INCIDENTSM1
where AFFECTED_ITEM like 'NEU FI%'
```

Ja luotiin tietokantaan taulu (**Kuva 10.**).



Kuva 10. Interraction-taulu.

8.3 Tietojen tuonti Excel-tilukosta

Raportti työasemista ja mobiililiittymistä sekä laitteista voidaan tuoda tietokannan tauluun SQL-lauseella. Ensinn pitää ottaa huomioon, että taulun sarakkeet vastaavat

Excel-sarakkeita, sitten pitää muuttaa .xls-tiedosto .csv-muotoon, jotta tiedon tuonti onnistuu. Esimerkki .csv-tiedoston tuonnista tietokannan tauluun:

```
bulk insert [dbo].[Workstation]
from '\\server\CMDB\Snow_raportti_20130801.csv'
with (fieldterminator = ';', rowterminator = '\n')
go
```

Bulk insert – valitaan taulu, jonne halutaan tiedot tuoda.

From – Polku, jossa tiedosto josta tiedot tuodaan sijaitsee.

With – Voidaan ottaa parametreja mukaan, jotta logiikka osaa pätkiä tiedot oikean kokosiin palasiin ja asettaa ne oikeaan kohtaan. Fieldterminator-muuttujaan valitaan ”;”, niin lause tietää, että ”;”-merkin jälkeen vaihdetaan saraketta ja rowterminator-muuttujaan asetetaan ”\n”, jotta lause tietää, että ”\n”-merkki tarkoittaa rivinvaihtoa. Täten saadaan tiedot oikeille sarakkeille ja riville.

8.4 Tietojen päivittäminen

PersonData, Incident ja Interraction tiedot muuttuvat aika usein, joten ne pitäisi päivittää päivittäin, ehkä jopa parikin kertaa päivässä. Ja kun tiedot haetaan suoraan toisesta SQL-kannasta, niin päivityksen noille tauluille voisi ajastaa suoraan serveriltä ajettavaksi automaattisesti. Luodaan serverille .SQL-tiedosto, jonne laitetaan SQL-haut, joilla saadaan päivitettyä tiedot tauluihin. Sitten voidaan ajastaa serverin ajoihin, ajetaan konsolista tietyin väliajoin tiedosto. Ajastetaan se aluksi hakemaan nuo tiedot kerran päivässä, joka aamu klo 7:00.

Tiedot työasemista sekä puhelimista ja liittymistä voidaan päivittää kerran viikossa. Tiedot eivät muutu oikein useasti, joten viikon päivitysväli sopii niihin hyvin.

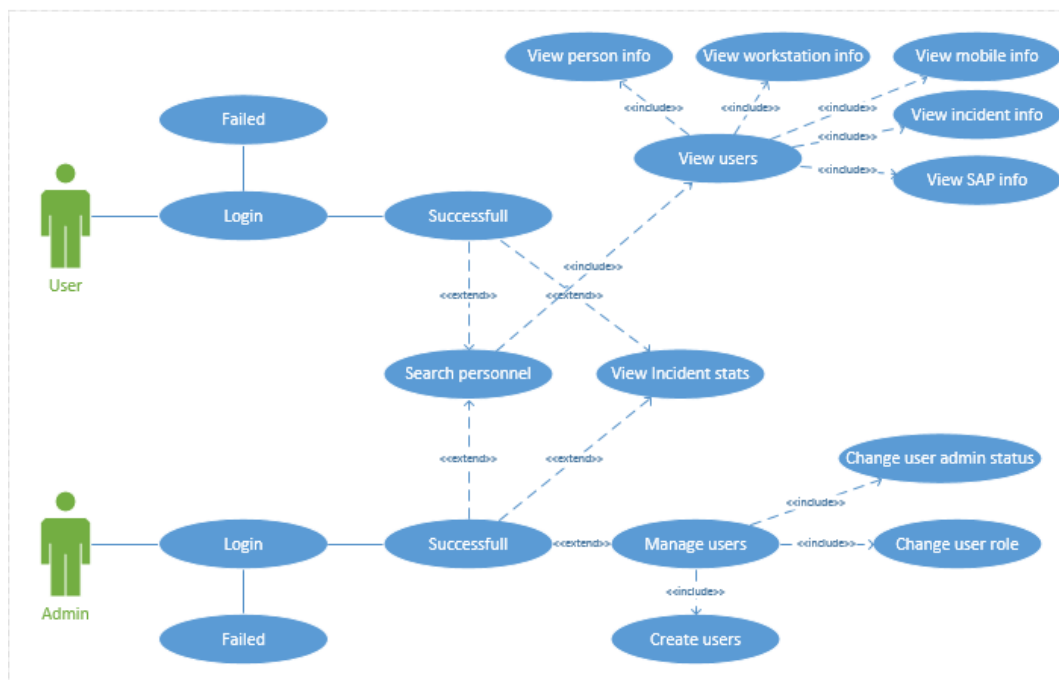
8.5 Näkymät

Luotiin datagridiä varten yksi iso näkymä johon yhdistettiin henkilöntietoihin kaikki henkilöllä olevat laitteet.

9 KÄYTTÖLIITTYMÄ

9.1 Määrittely

Käyttöliittymällä on tarkoitus tehdä hakuja tietokantaa vasten ja näyttää tietoja käyttäjälle. Ohjelmalle tulee kirjoitusoikeudet vain tietokannan tauluun, jossa määritellään tunnukset joilla on oikeudet käyttää ohjelmaa. Tuonne tauluun voidaan ohjelman kautta lisätä käyttäjiä.



Kuva 11. Käyttötapakaavio.

9.2 Suunnittelu

Aluksi tietokantaan pystyi tekemään hakuja Microsoftin raportointipalvelun kautta, johon ABB:n konsultti oli yhdistänyt CMDDB-tietokannan. Tuo raportointipalvelu on kuitenkin tarkoitettu lähinnä isojen raporttien luomiseen, se toimii hitaasti yksittäisten hakujen osalta, joten tähän tarkoitukseen se ei ollut oikein sopiva muuta kuin väliaikaiseksi ratkaisuksi.

Ohjelmointikieleksi valitsin C#-kielen ja projektityypiksi Windows Forms -projektin, koska siitä löytyy valmiiksi paljon komponentteja, joista on helppo kasata esim. yhteys tietokantaan.

Aloin miettimään, millainen olisi hyvä käyttöliittymä henkilöiden tietojen katseluun. Tietoja tulee näkyville paljon, mutta kuitenkin ohjelman koko olisi hyvä pitää mahdollisimman pienenä. Ohjelmasta tulisi nähdä henkilön perustiedot(nimi, sähköposti, työpaikan osoitetiedot, yksikkötieto, kustannuspaikka), työasemasta tiedot, työpuhelimesta tiedot, SAPin tunnustiedot, service deskille jätetyt keikat.

9.3 Toteutus

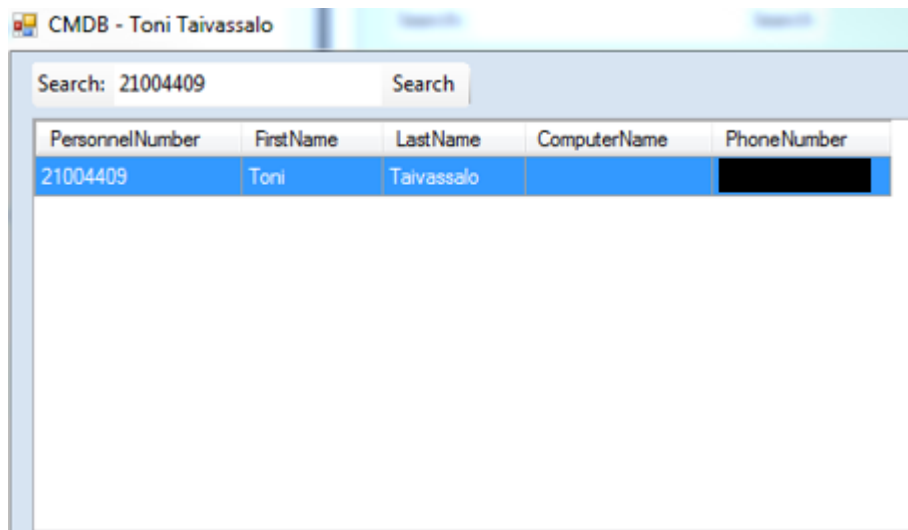
9.3.1 DataGrid

Aloitin ohjelman kokoamisen datagridistä, eli komponentti johon tulee näkyville tulokset mitä haetaan tietokannasta. Huomasin, että en voi suoraan käyttää datagridissä tauluja joita olin luonut tietokantaan, koska tietoja pitäisi yhdistellä, joten tulin siihen tulokseen, että minun pitää luoda näkymiä tietokantaan.

Näkymään pitää saada yhdistettyä henkilön tiedot, työasemat, puhelimentiedot sekä tunnustiedot, ne voidaan sitten kaikki hakea yhdellä haulla.

9.3.2 Search kenttä

Search-kenttä tulee automaattisesti datagridin mukana, siihen voi määritellä, että millä tiedoilla hakuja voi suorittaa. Lisäsin tuohon haku kenttään ominaisuudet, että hakuja voi suorittaa henkilön nimellä, ABB:n henkilönumerolla, työaseman laitetunnuksella tai puhelinnumerolla. Lisääminen onnistuu lisäämällä SQL-lause objektille, jotka se sitten käy läpi ja katsoo löytyykö haluttavaa tietoa (**Kuva 12.**).

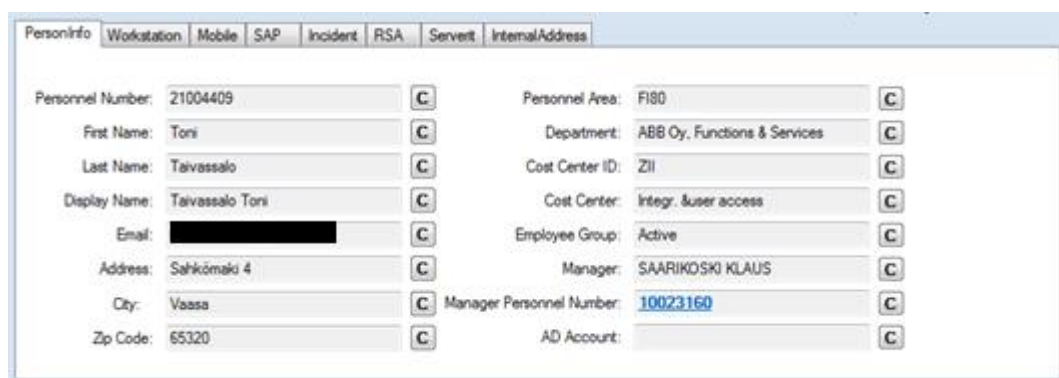


Kuva 12. DataGrid ja haku kenttä.

Muokkasin datagridiin näkyville vain tiedot, joita voi käyttää haussa.

9.3.3 Välilehdet

Kun datagridilta valitaan haluttu rivi, täyttää ohjelma sen jälkeen valitun henkilön tiedot välilehdille.



Kuva 13. Henkilön perustiedot.

PersonInfo-välilehti näyttää henkilön perustiedot, johon kuuluu työpisteen osoitetiedot sekä yksikön tiedot jossa henkilö työskentelee ja missä tehtävissä. Esimies näkyy myös välilehdellä ja Manager Personnel Number kohdassa olevasta numerosta jos painaa, niin ohjelma hakee ruudulle esimiehen tiedot (**Kuva 13.**).

The screenshot shows a software interface with several tabs: PersonInfo, Workstation, Mobile, SAP, Incident, RSA, Server, and InternalAddress. The 'Workstation' tab is active. It displays the following information:

- Workstation: FI-L-7002904
- Computer Name: FI-L-7002904
- Serial Number: PK1RT76
- Main Type: High Performance Laptop
- Architecture: 64-bit
- Consumer Name: Taivassalo, Toni Kristian
- Consumer ID: 11002774
- Lease End Date: 04.03.2018
- Lease Begin Date: (empty)

Additional information on the right side includes:

- City: VAASA
- Organisation: ABB Oy/FIABB15 Functions & Servic

Below this information is a section titled 'Admins' containing a list of administrators (currently showing 'Administrator'), a 'Read File' button, a 'Clear' button, and a 'Last logged by' field. A message box states 'The command completed successfully.' and there is a link to 'Open log file'.

Kuva 14. Henkilön työaseman tiedot.

Workstation-välilehti näyttää henkilön työaseman tiedot, mikäli henkilön nimiltä sellainen löytyy. Mitään tarkempia laitetietoja tähän ei ole otettu (**Kuva 14.**).

This is a close-up view of the 'Admins' section from the previous screenshot. It shows a list box containing 'Administrator' and a redacted area below it. To the right of the list box are two buttons: 'Read File' and 'Clear'. Below the buttons is the text 'Last logged by : ' followed by a redacted area. At the bottom of the section, a message box displays 'The command completed successfully.'

Kuva 15. Työaseman admin-tunnusten haku.

Workstation-välilehdeltä löytyy laatikko johon voidaan hakea työasemalle viimeksi kirjautunut tunnus sekä työasemalta löytyvät paikalliset admin-tunnukset (**Kuva 15.**).

Ohjelma etsii ensin oikean log-tiedoston ja sen jälkeen lukee sieltä tiedot työasemassa olevista admin-tunnuksista sekä katsoo mikä tunnus on viimeksi kirjautunut koneelle. Log-tiedostot on muotoa työaseman nimi.log, eli tässä tapauksessa ohjelma ottaa valitusta työasemasta Computer Name -tiedon, ja etsii sen avulla oikean tiedoston.

Tiedostoja kansiossa, jossa log-tiedostoja ylläpidetään on tuhansia. Ja log-tiedosto sisältää paljon tietoa työasemasta, kuten mitä komponentteja työasema sisältää, tiedot koneesta tai kuka kirjautunut koneelle ja milloin. Halutaan kaivaa sieltä vain

admin-tiedot sekä viimeksi kirjautunut henkilö ja jos saatavilla, niin milloin kirjaututtu(ei näy kaikille koneille).

Monesti service deski tarvitsee tietoa kuka on viimeksi kirjautunut koneelle. Esimerkiksi tapauksissa joissa halutaan tietää kenellä jokin kone on käytössä tai ollut viimeksi käytössä. Tunnuksen avulla voidaan sitten tunnistaa henkilö.

Ohjelmaan on asetettu aloituspiste ja lopetuspiste, eli kohdat jotka se etsii tiedostosta ja sitten niiden väliin osuvat rivit tulostetaan välilehdelle.

The screenshot shows a software interface with several tabs: PersonInfo, Workstation, Mobile, SAP, Incident, RSA, Servent, and InternalAddress. The 'Mobile' tab is active. It displays a 'Device' dropdown menu. Below it, there are fields for 'Phone Number' (redacted), 'Status' (Aktiivinen), and 'IMEI'. To the right, there are fields for 'Mobile Consumer Name' (TAIVASSALO, TONI), 'Leasing Start Date', and 'Leasing End Date'. A small 'C' icon is visible next to the Phone Number field.

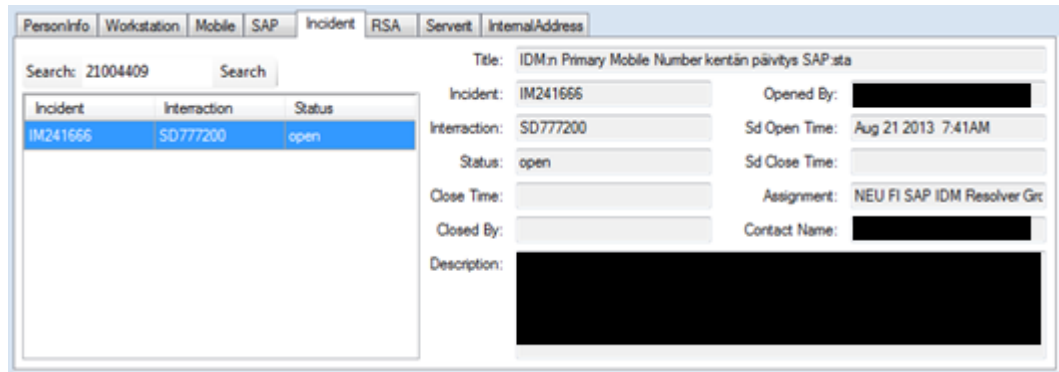
Kuva 16. Henkilön puhelin- ja liittymätiedot.

Mobile-välilehdeltä löytyy henkilön nimille merkitty puhelinmalli, sekä liittymän tietoja. Huom. esimerkiksi minulla on käytössä lainapuhelin, joten se ei ole minun nimille merkitty, joten minulta näkyy vain liittymän tiedot (**Kuva 16.**).

The screenshot shows the same software interface as in Kuva 16, but with the 'SAP' tab active. It displays a list of accounts with their IDs and status icons. The accounts are: ACCOUNT ESP111, ACCOUNT EST111, ACCOUNT ESD050, ACCOUNT BP1100, ACCOUNT BT1100, and ACCOUNT BD1100. Each account ID is followed by a redacted field and a small 'C' icon.

Kuva 17. Henkilön tunnistiedot SAPista.

SAP-välilehdeltä näkee SAP-tunnukset eri ympäristöihin, mikäli kenttä on tyhjä, ei henkilöllä ole tunnusta (**Kuva 17.**).



Kuva 18. Henkilön keikat HPSM-keikkajärjestelmästä.

Incident-välilehdeltä näkyy henkilön keikkoja joita on jätetty HPSM-keikkajärjestelmään. Ohjelma näyttää kaikki henkilöllä olevat avoimet keikat sekä viisi viimeksi suljettua keikkaa. Incidenciteillä on käytössä oma hakukenttä. Kun datagridilta valitaan rivi, ohjelma poimii rivin henkilöltä henkilönumeron ja lisää sen automaattisesti Incident välilehden search-kenttään ja suorittaa haun (**Kuva 18.**).

9.3.4 Kirjautuminen

Seuraavaksi piti miettiä, miten kirjautuminen ohjelmaan tapahtuu. Aluksi meinattiin tehdä ihan tavallisen kirjautumisen, että henkilöille jotka käyttävät ohjelmaa luodaan tunnus ja salasana ja sitten kirjautuvat niitä käyttäen sisään ja kirjautumistietoja säilytettäisiin tietokannassa. Ongelmaksi tuli, että on jo tällä hetkellä niin paljon eri paikkoja mihin täytyy muistaa tunnukset ja salasanat, ettei se olisi kovin hyvä vaihtoehto.

Alettiin tutkimaan, miten henkilön käyttämää Active Directory -tunnusta voisi käyttää hyväksi kirjautumisessa, saisiko sen ohjelmallisesti haettua tunnuksen, jolla ohjelman käyttäjä on kirjautunut koneelle, ja sitten ylläpidettäisiin tietokannassa vain Active Directory -tunnuksia, joilla olisi oikeudet kirjautua. Hetken asiaa tutkittuani löysin, että tunnuksen saa selville seuraavalla koodilla:

```
string kayttaja= WindowsIdentity.GetCurrent().Name.ToString();
```

Tunnus tulee muotoon DOMAIN\AD-TUNNUS, eli pitää vielä katkaista tuo lause ja ottaa vain tuo loppuosa siitä muistiin. Lauseen katkaiseminen ”\” merkin kohdalta onnistuu lauseella

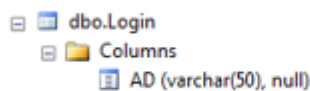
```
string[] split = kayttaja.Split("\");
```

Ja katkaisun jälkeen voidaan kirjoittaa se takaisin muistiin kayttaja-muuttujaan

```
kayttaja = split[1];
```

Split –muuttujan nollakohtaan eli split[0] kirjoittuu tässä tapauksessa DOMAIN ja sitten tuohon split[1] kohtaan kirjoitetaan AD-TUNNUS.

Luodaan seuraavaksi tietokantaan Login-taulu, johon luotiin vain AD-sarake. Sarakkeessa ylläpidetään Active Directory -tunnuksia, joilla on oikeus käynnistää ohjelma.



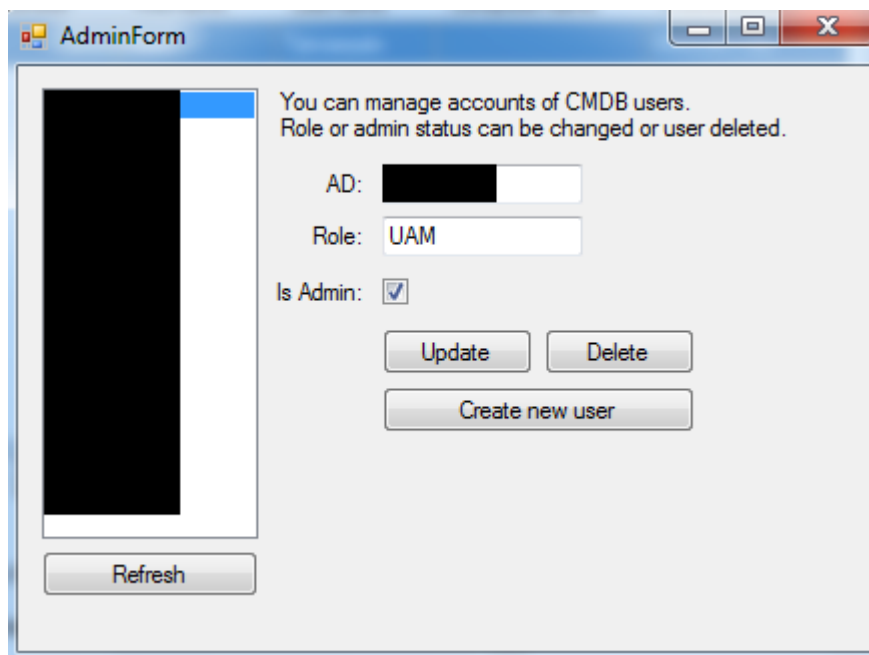
Kuva 19. Login-taulu tietokannassa.

Tehdään ohjelmaan tarkistus, joka katsoo löytyykö ohjelmaa avaavan henkilön AD-tunnus tietokannasta (**Kuva 19.**). Otetaan ensin henkilön Active Directory -tunnus ylös sitten käytetään sitä SQL-lausekkeessa seuraavasti:

```
string query = "SELECT AD FROM Login WHERE AD=@adminId";
```

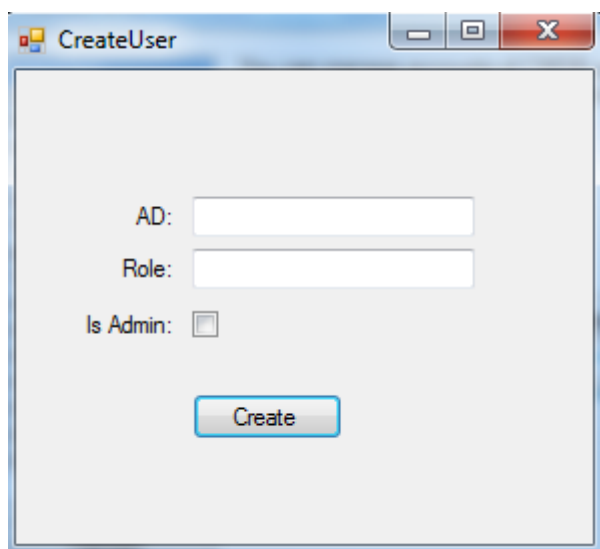
Luetaan tietokantaa ja jos oikea tunnus löytyi sieltä, niin kirjoitetaan se muistiin muuttujaan adUser.

```
while (dr.Read())
{
    adUser = dr.GetString(0);
}
```

Kuva 21. Admin-näkymä.

Admin-näkymässä voi muokata tunnuksia joilla on jo oikeudet kantaan, lista vasemmassa reunassa, tunnuksia voi myös poistaa ja listan päivittää refresh napista. Create new user -nappia painamalla avautuu jälleen uusi näkymä, jossa voi luoda uuden tunnuksen.



Kuva 22. Uuden tunnuksen luonti.

Tällä hetkellä tunnuksen luonnissa ei ole mitään tarkistusta esim. roolin suhteen, että siihen voisi kirjoittaa vain käytössä olevia rooleja, vaan siihen voisi kirjoittaa mitä vain, eli uutta tunnusta luodessa pitäisi tietää, mitä rooleja on käytössä. Myöhemmin tuohon voisi ylläpitää listan, josta valitaan käytössä olevista rooleista henkilölle sopivin.

9.3.6 Näkymät

Ohjelmasta voi seurata omien keikkojen statuksia sekä tiimin keikkajonoa. Nämä näkymät päivittyvät sen mukaan, mikä rooli henkilöllä on käytössä.

The screenshot displays the UAM user interface. At the top, it shows the logged-on user as 'Admin - UAM' with a redacted email address. There are 'Exit' and 'Admin' buttons. Below this, the interface is divided into two columns of statistics:

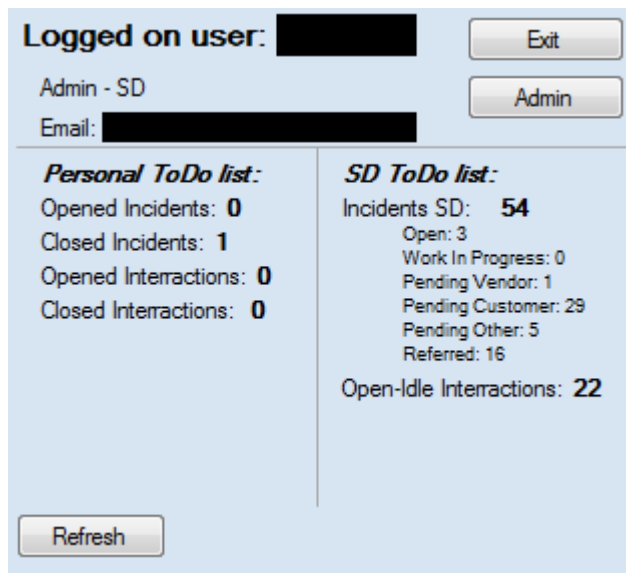
Personal ToDo list:	UAM ToDo list:
Opened Incidents: 0	Incidents UAM: 174
Closed Incidents: 1	Open: 103
Incidents: 35	Work In Progress: 8
Open: 0	Pending Vendor: 32
Work In Progress: 0	Pending Customer: 14
Pending Vendor: 25	Pending Other: 11
Pending Customer: 3	Referred: 1
Pending Other: 7	
Referred: 0	
Open Tasks: 0	Total opened and closed:
Closed Tasks: 0	Opened UAM Incidents: 0
	Closed UAM Incidents: 2
	Summary: -2

At the bottom left, there is a 'Refresh' button.

Kuva 23. UAM-roolin näkymä.

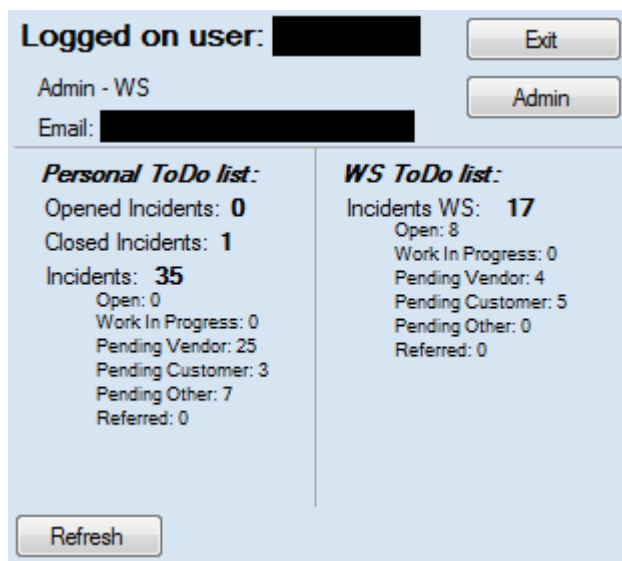
Ensiksi ylhäällä näkyy tunnus jolla ohjelma on avattu, eli Active Directory -tunnus, sama jolla kirjaudutaan koneelle. Sen alla näkyy onko henkilö admin vai pelkkä käyttäjä sekä hänen roolinsa. Sitten on kirjoitettu ylös henkilön sähköposti. Sen alta alkaa tiedot henkilön ja tiimin keikoista, personal To-Do list listaa henkilöllä HPSM:ssä olevat keikat, tarkastelee montako keikkaa on avattu ja suljettu viimeisen 12 tunnin aikana. Käyttäjällä tällä hetkellä olevat avoimet keikat ilmoitetaan Incidents kohdassa ja sen jälkeen lajitellaan ne vielä statuksen mukaan, eli tuosta voi seurata hyvin onko jossain keikassa vielä työ kesken(work in progress) tai jos keikalle on tullut uusi viesti (referred). UAM-näkymässä näkyy

myös taskit, taskit on keikkoja, joilla tehdään muutoksia eri SAP-rooleihin. Oikealla on UAM-todo list, jossa näkyy koko tiimille merkityt keikat, kokonaismäärä sekä sitten lajiteltuna eri statuksen mukaan. Ja niiden alla on total opened and closed, josta näkee viimeisen 12 tunnin aikana jonoon avatut ja suljetut keikat.



Kuva 24. SD-näkymä.

Service deskin henkilökohtainen jono näyttää itseasiassa vain avatut ja suljetut incidentit ja interractions. Service deskin henkilöiden keikat ovat aina yhteisessä jonossa eli keikat näkyy tuossa SD-ToDo list -kohdan alla, joten ne eivät tarvitse henkilökohtaista seuranta. Open-Idle interractions tarkoittaa keikkoja, jotka odottavat että ne siirretään eri ratkaisijaryhmille.



Kuva 25. WS-näkymä.

Työasematiimillä näkyy henkilökohtainen keikkajono jaettuna keikkojen statuksen mukaan, sekä koko tiimin yhteinen keikkajono jaettuna statuksen mukaan.

9.4 Testaus

Saatiin service deskin tiimistä yhden henkilön testailemaan ohjelmaa. Suurimmaksi osaksi hakujen osalta ohjelma toimi niin kuin pitikin. Työasemien tiedoista löydettiin yksi ongelma, työaseman leasing aika näytti väärää tietoa. Asiaa tutkittua huomattiin, että tuo kenttä oli virheellisesti yhdistynyt puhelimen leasing aikaan ja tästä syystä näytti väärin.

Aluksi ohjelma kaatuili välillä, kun haun jälkeen vaihteli liian nopeasti henkilöä jonka tietoja halusi katsella. Esimerkiksi jos haki henkilöä sukunimellä ja hakuun tuli useampi saman sukunimen omaavaa henkilöä ja heidän välillään vaihteli liian nopeasti. Tämä ongelma saatiin korjattua muokkaamalla datagridin asetuksia.

Myös HPSM-tilastoissa oli aluksi ongelmaa, tilastot näyttivät väärää lukuja SD:n(service deskin) roolilla. Ongelma löytyi lopulta tietokanta-hauista, joka korjattiin.

Haussa on edelleen pieni ongelma, että siinä on pakko käyttää skandinaavisia kirjaimia, mikäli henkilön nimessä sellaisia esiintyy. Tuohon ongelmaan ei ole vielä ratkaisua keksitty.

10 JOHTOPÄÄTÖKSET JA POHDINTA

Tavoitteena oli suunnitella ja toteuttaa tietokanta, jossa ylläpidetään kaikkia henkilöihin liittyvää tietoa, joka on sitten helposti saatavilla. Alunperin oli myös sovittu, että suunnitellaan käyttöliittymä tietokannalle.

Projekti onnistui aika hyvin, vaikka se ei aina edennyt ihan niin kuin oli suunniteltu ja haasteitakin tuli vastaan.

Haasteita oli jonkin verran tietojen kasaamisessa eri järjestelmistä, kun huomattiin että puhelinten ja liittymien osilta data olikin osittain puutteellista, mutta selvittiin niistäkin osittain ja jossain vaiheessa tulevaisuudessa noita pitää vielä parannella.

Projektista tuli hieman laajempi mitä aluksi ajateltiin. Käyttöliittymä piti alunperin tilata ulkoiselta toimittajalta, mutta päädyinkin sen ohjelmoimaan itse, joka olikin tosi hyvä asia, sai näyttää työnantajalle, että osaa ohjelmoida ja samalla itse oppi paljon uutta.

Käyttöliittymän perusta suunniteltiin aluksi mahdollisimman yksinkertaiseksi sekä helppokäyttöiseksi, jota sitten laajennettiin sen mukaan mitä ideoita tuli ja minulla osaamista riitti. Vieläkin ohjelmaa kehitetään sen mukaan kuin tarvetta on.

Työ oli omasta mielestäni tosi mielenkiintoinen ja sopivasti haastava, varsinkin käyttöliittymän suunnittelu oli tosi mielenkiintoista ja palkitsevaa kun sai itse suunnitella ja toteuttaa sitä sen mukaan mitä sai ideoita.

Olen kiitollinen ABB:lle, että sain mahdollisuuden tehdä tämän työn heille.

LÄHTEET

/1/ ABB OY. Yritys. Viitattu 3.10.2013.

<http://www.abb.fi/cawp/fiabb251/657dfdcf6e344cc7c1256b20003149ae.aspx>

/2/ Snow Learn more. Viitattu 4.10.2013. [http://snowsoftware.com/web-](http://snowsoftware.com/web-site1/1.0.1.0/12/2/)

[site1/1.0.1.0/12/2/](http://snowsoftware.com/web-site1/1.0.1.0/12/2/)

/3/ Hovi, A., Huotari, J. & Lahdenmäki, T. 2005. Tietokantojen suunnittelu & indeksointi. Porvoo. WS Bookwell


```

update @usr set ACCOUNT_AD = avalue from IDM.mxmc_db.dbo.idmv_value_ext where at-
trname='XX_AD_SAMACCOUNTNAME' and usrkey=mskey
update @usr set ACCOUNT_ESP111 = avalue from IDM.mxmc_db.dbo.idmv_value_ext where
attrname='ACCOUNTESP111' and usrkey=mskey
update @usr set ACCOUNT_EST111 = avalue from IDM.mxmc_db.dbo.idmv_value_ext where
attrname='ACCOUNTTEST111' and usrkey=mskey
update @usr set ACCOUNT_ESD050 = avalue from IDM.mxmc_db.dbo.idmv_value_ext where
attrname='ACCOUNTESD050' and usrkey=mskey
update @usr set ACCOUNT_BP1100 = avalue from IDM.mxmc_db.dbo.idmv_value_ext where
attrname='ACCOUNTBP1100' and usrkey=mskey
update @usr set ACCOUNT_BT1100 = avalue from IDM.mxmc_db.dbo.idmv_value_ext where
attrname='ACCOUNTBT1100' and usrkey=mskey
update @usr set ACCOUNT_BD1100 = avalue from IDM.mxmc_db.dbo.idmv_value_ext where
attrname='ACCOUNTBD1100' and usrkey=mskey
update @usr set Manager = avalue from IDM.mxmc_db.dbo.idmv_value_ext where at-
trname='displayname' and mskey=
(select mcothersmskey from IDM.mxmc_db.dbo.idmv_link_ext where mcattrname='mx_man-
ager' and mcthismskey=usrkey)
update @usr set DisplayName = avalue from IDM.mxmc_db.dbo.idmv_value_ext where at-
trname='DISPLAYNAME' and usrkey=mskey

```

```

Insert into CMDB.DBO.PersonData(FirstName, LastName, PersonnelNumber, EmployeeGroup,
Email, PersonnelArea, Department, SubArea, CostCenterID, CostCenter, ManagerPersonnel-
Number, ACCOUNT_AD, ACCOUNT_ESP111, ACCOUNT_EST111, ACCOUNT_ESD050,
ACCOUNT_BP1100, ACCOUNT_BT1100, ACCOUNT_BD1100, Manager, DisplayName)
select FirstName, LastName, PersonnelNumber, EmployeeGroup, Email, PersonnelArea, De-
partment, SubArea, CostCenterID, CostCenter, ManagerPersonnelNumber, ACCOUNT_AD,
ACCOUNT_ESP111, ACCOUNT_EST111, ACCOUNT_ESD050, ACCOUNT_BP1100, AC-
COUNT_BT1100, ACCOUNT_BD1100, Manager, DisplayName from @usr where Personnel-
Number <> 'null'

```