

Saimaan ammattikorkeakoulu  
Tekniikka Lappeenranta  
Tietotekniikka  
Ohjelmistotekniikka

Ilkka Jyrkinen

## **Vikailmoitusjärjestelmä**

Opinnäytetyö 2012

## **Tiivistelmä**

Ilkka Jyrkinen

Vikailmoitusjärjestelmä, 43 sivua

Saimaan ammattikorkeakoulu

Tekniikka Lappeenranta

Tietotekniikka

Ohjelmistotekniikka

Opinnäytetyö 2012

Ohjaajat: lehtori Martti Ylä-Jussila, Saimaan ammattikorkeakoulu  
tuotantopäällikkö, Kari Kattainen, Imatran Työstöasennus Oy

Tämän opinnäytetyön tarkoituksena oli määritellä, suunnitella, toteuttaa ja käyttöönottaa Imatran Työstöasennus Oy:n konepajan koneille vikailmoitusjärjestelmä. Vikailmoitusjärjestelmän tavoite on nopeuttaa vikailmoitusten kirjaamista ja käsittelyä.

Työ aloitettiin selvittämällä järjestelmän vaatimukset ja opiskelemalla mahdollisia toteutustapoja. Vaatimuksien pohjalta päätettiin, että järjestelmä on lähiverkossa toimiva sivusto, joka toteutettiin ASP.NET-tekniikalla ja C#-ohjelmointikielellä.

Työn tuloksena saatiin kehitettyä järjestelmä, joka vastasi asiakkaan vaatimuksia ja on otettu käyttöön toukokuussa 2012.

Asiasanat: ASP, .NET, C#, vikailmoitusjärjestelmä

## **Abstract**

Ilkka Jyrkinen

Fault reporting system, 43 pages

Saimaa University of Applied Sciences

Information technology

Software Engineering

Bachelor's Thesis 2012

Instructors: lector Martti Ylä-Jussila, Saimaa University of Applied Sciences  
production manager Kari Kattainen, Imatran Työstöasennus Oy.

The objective of this thesis was to define, design, implement and put to use a fault reporting system for the machines in the machine hall of Imatran Työstöasennus Oy. The goal of the fault reporting system is to speed up the recording and processing of fault reports.

The work began by finding out the requirements for the system, and by studying possible solutions. Based on the requirements, it was decided to create a web-page working in the local area network environment, which was developed by using ASP.NET technology and the programming language was C#.

As a result, a system was developed, which fills the requirements of the customer and the system has been put to use in May 2012.

Keywords: ASP, .NET, C#, fault reporting system

## Sisältö

Käytetyt termit ja lyhenteet.....	5
1 Johdanto .....	6
1.1 Taustaa .....	6
1.2 Tavoite.....	7
2 Menetelmät ja tekniikat .....	8
2.1 Vesiputousmalli .....	8
2.2 Ketterä kehittämismalli .....	8
2.3 Kehittämismallien vaiheet.....	9
2.4 Relaatiotietokanta.....	12
2.5 UML.....	15
2.6 ASP.NET .....	22
3 Työkalut .....	23
3.1 Microsoft Visual Studio 2010 .....	23
3.2 Microsoft SQL Server Management Studio 2008 .....	24
3.3 Microsoft Windows Server 2008 R2 .....	25
4 Vikailmoitusjärjestelmä-projektin vaiheet ja organisaatio .....	26
4.1 Organisaatio .....	26
4.2 Esiselvitys.....	26
4.3 Projektisuunnitelma .....	27
4.4 Toiminnallinen määrittely .....	28
4.5 Toteutus .....	28
4.6 Testaus.....	28
4.7 Käyttöönotto .....	28
5 Vikailmoitusjärjestelmä .....	29
5.1 Sisäänkirjautuminen .....	29
5.2 Vikailmoituksen jättäminen .....	30
5.3 Vikalistaus .....	31
5.4 Vikailmoituksen tilan muuttaminen .....	32
5.5 Raportointi .....	33
5.6 Koneiden hallinta.....	34
5.7 Korjaajien hallinta .....	35
5.8 Postituslistan hallinta .....	36
5.9 Käyttäjätunnuksien hallinta.....	37
5.10 Käyttäjähallinta .....	38
5.11 Tietokanta.....	39
5.12 Käyttöönottokaavio .....	40
6 Yhteenvedo ja pohdinta .....	41
Kuvat.....	42
Taulukot.....	43
Lähteet.....	43

## Käytetyt termit ja lyhenteet

<b>Termi</b>	<b>Selvitys</b>
ASP.NET	Active Server Pages, Web-ohjelmistokehys.
CNC-sorvin	Computerized Numerical Control, Tietokoneistettu numeerisesti ohjattu sorvin.
C#	Oliopohjainen ohjelmointikieli, joka on kehitetty .NET-sovelluskehystä varten.
JOT	Juuri Oikeaan Tarpeeseen, menetelmä laatuksennuksien vähentämiseen.
Konepaja	Metallituotteiden valmistukseen tai korjaukseen erikoistunut yritys.
MD5-tiiviste	128-bittinen tiiviste, joka esitetään 32-merkkisenä 16-järjestelmään koodatussa muodossa.
.NET	Ohjelmistokomponenttikirjasto.
Relaatiotietokanta	Tietokanta, jossa tieto on tallennettu tauluihin. Tiedon lisäksi tietokantaan tallennetaan taulujen väliset yhteydet.
SQL	Structured Query Language. Standardoitu tietokannan kyselykieli.
SMTP	Simple Mail Transfer Protocol. Käytetään viestien välittämiseen sähköpostipalvelimien kesken.
StarUML	Ohjelma, jolla pystyy luomaan UML 2.0 -standardin mukaisia diagrammeja.
Vikailmoitusjärjestelmä	Järjestelmä vikailmoitusten kirjaamiseen, käsittelyyn ja raportointiin.

# 1 Johdanto

Nykyaikainen konepajateollisuus toimii pääosin niin, että suuret kone- ja laitevalmistajat vastaavat lopputuotteen suunnittelusta ja kokoonpanosta. Nämä suuret konepajat tilaavat tarvittavat osat pienemmiltä erikoistuneilta konepajoilta.

Koko ketju toimii JOT-perustaisesti, josta johtuen osia ei valmisteta varastoon. Tämän takia alihankkijoina toimivilta konepajoilta vaaditaan tehokasta ja laadukasta toimintaa. Tuotantokoneiden on oltava hyvässä kunnossa ja ilmenneet viat korjattava mahdollisimman nopeasti.

## 1.1 Taustaa

Veljekset Esko ja Pekka Matikainen perustivat Imatran Työstöasennus Oy:n vuonna 1980. Yritys valmistaa teräksestä vaativia koneistettuja komponentteja suurille kone- ja laitevalmistajille.

Imatran Työstöasennus aloitti vanhoissa Imatran Moottorin tiloissa, jolloin työntekijöitä yrityksellä oli kolme ja koneita oli kaksi, manuaalijyrsin ja pylväsporakone. Yritys on lamavuotta lukuun ottamatta kasvanut perustamisesta lähtien. Vuonna 1988 yritys investoi uuteen halliin ja vuonna 1999 hankittiin ensimmäinen CNC-sorvi, joka on vielä nykyäänkin yrityksellä käytössä.

Vuosituhaten vaihtuessa yritys työllisti noin kymmenen henkeä, jolloin yritys laajensi taas toimitilojaan. Vuonna 2002 valmistuneen laajennuksen johdosta yritys tuplasi hallitilansa noin tuhanteen neliometriin. Kuitenkin yrityksen kasvusta johtuen vuonna 2006 käynnistettiin uusi projekti, jonka myötä haettiin yritykselle uusia toimitiloja. Tämän johdosta yrityksen toimitilat sijaitsevat nykyään Ruokolahdella, ja toimitilat ovat nyt noin 3000 neliometriä.

Vuonna 2007 ja 2008 yritys teki uusia konehankintoja, joita olivat muun muassa 9-akselinen monitoimisorvin ja toinen vaakakarainen koneistuskeskus, joka liitettiin FM-järjestelmään. Tällöin hankintoihin kuului myös 3D-koordinaattimittakone, jota käytetään laadunvarmistukseen.

Nykyään Imatran Työstöasennuksen toimitusjohtajana toimii Esko Matikaisen poika, Tommi Matikainen. Yrityksessä työskentelee noin 35 henkilöä ja CNC-työstökoneita on yhteensä 14 kappaletta. Vuonna 2011 yrityksen liikevaihto oli noin 4,5 miljoonaa euroa ja sen pääasiakkaisiin kuuluu Wärtsilä Oyj, Sandvik Mining and Construction Oy ja Ponsse Oyj.

## **1.2 Tavoite**

Tämän opinnäytetyön tarkoitus on suunnitella, toteuttaa, testata ja käyttöönottaa sähköinen vikailmoitusjärjestelmä konepajan koneille. Vanhaa sähköistä järjestelmää ei ole, joten tällä hetkellä kaikki vikailmoitukset kirjataan paperilomakkeille.

Manuaalisen vikailmoitusjärjestelmän ongelmia ovat muun muassa hidas tiedon eteneminen, vikailmoitusten katoaminen sekä niiden unohtuminen. Sähköisellä järjestelmällä pyritään poistamaan kyseiset ongelmat.

Uuden järjestelmän myötä tavoitellaan myös suurempaa vikailmoituksen täyttöprosenttia, helppoa vikailmoituksen tilan seurantaa sekä raportoinnin mahdollistamista.

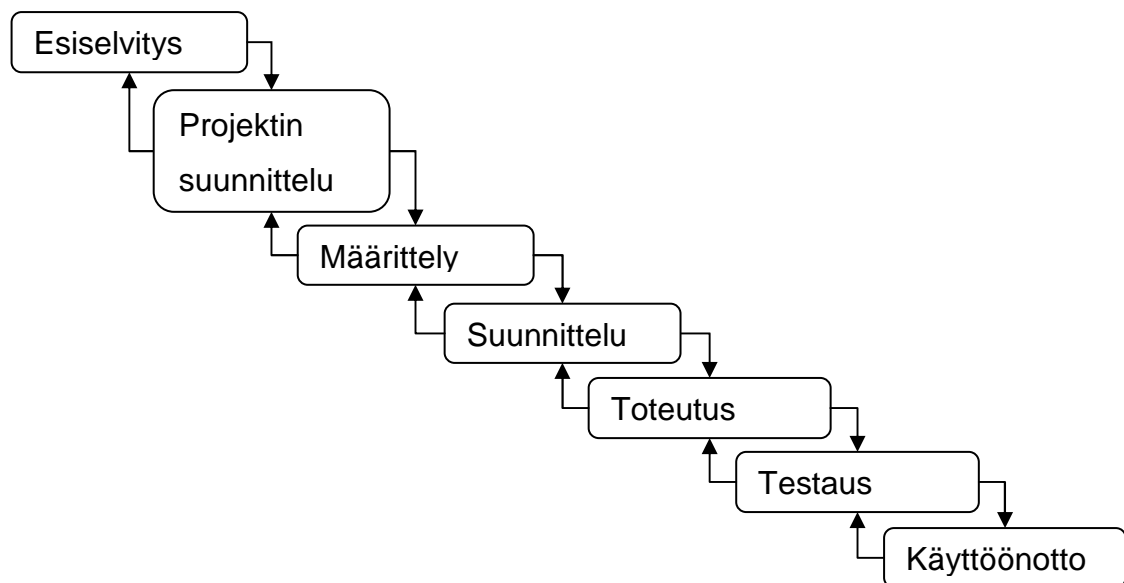
## 2 Menetelmät ja tekniikat

Tietojärjestelmän kehittämismalli määrittää projektissa käytettävän järjestelmän kehittämistavan. Kehittämismalli vaikuttaa projektin joustavuuteen ja se valitaan yleensä projektin suuruuden, tarpeiden ja luonteen mukaisesti.

Projektissa käytettävät tekniikat määräytyvät pitkälti järjestelmän vaatimuksista. Näiden vaatimuksien johdosta selviää muun muassa tietokannan tarpeellisuus ja osa käytettävistä työkaluista.

### 2.1 Vesiputousmalli

Vesiputousmallissa dokumentoidaan huolellisesti jokainen projektin vaihe ja pyritään etenemään kaavamaisesti, kunnes projekti on viety loppuun vaihe vaiheelta (Kuva 2.1). Tämä rajoittaa toteutuksessa esiintyviin muutostarpeisiin joustamista, mutta projektin läpivienti tapahtuu laajoissa projekteissa paljon selkeämmin (1, s.9–12).



Kuva 2.1 Vesiputousmalli (1, s.11).

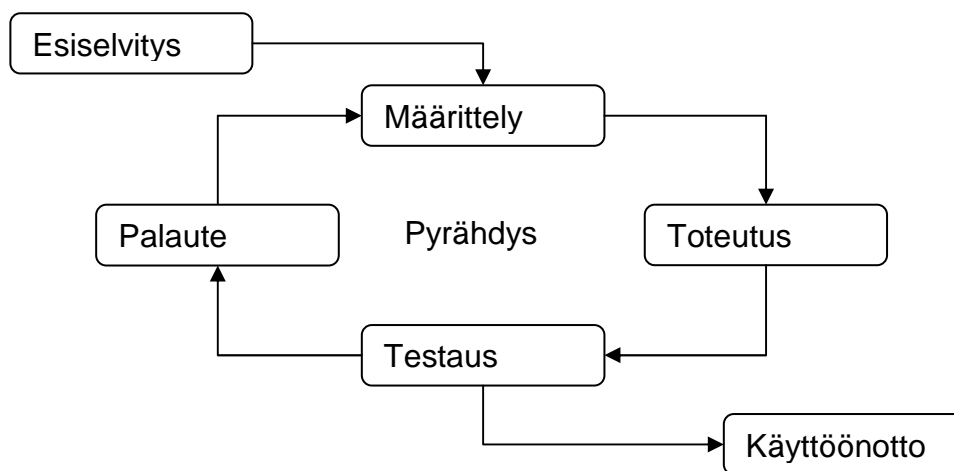
### 2.2 Ketterä kehittämismalli

Yleisimpiä ketteristä kehittämismalleista ovat eXtreme Programming ja Scrum. EXtreme Programming on niin sanottu testauslähtöinen ohjelmistokehitysmalli,



jossa yksikkötestit luodaan jo ennen koodin kirjoittamista. Scrum perustuu projektin pilkkomiseen pieniin osiin. Tämän jälkeen projektin toteuttaminen suoritetaan pyrähdyksissä. Näiden pyrähdysten alussa valitaan toteutettavaksi projektista osia, jotka arvioidaan olevan sopivia tälle aikajaksolle (1, s.14–25).

Ketterässä kehittämismallissa pyritään nopeisiin kehityssykleihin eli pyrähdysiin ja näin ollen alussa tehtävään suunnitteluun ei käytetä paljoa aikaa (Kuva 2.2). Pyrähdykset ovat lyhyitä ja niiden välissä katsotaan aina, mitä on saatu aikaiseksi, onko suunta oikea ja tarvitseeko muutoksia tehdä. Dokumentaatio elää toteutuksen edetessä koko ajan, ja tämä mahdollistaa näin ollen myös muutokset järjestelmään projektin ollessa jo pitkällä (1, s.14).



Kuva 2.2 Ketterä kehittämismalli (1, s.13)

### 2.3 Kehittämismallien vaiheet

Ohjelmiston kehittämismallit jakaantuvat vaiheisiin, joiden sisältö vaihtelee vain vähän kehitysmallin tyypin mukaan.

#### Esiselvitys

Esiselvityksellä pyritään hahmottelemaan projektin tarpeellisuus, riskit, toteutuvuus ja tavoitteet. Tässä tulee esille myös järjestelmälle asetetut vaatimukset, jotka asiakas määrittää (2, s.83).

## **Projektin suunnittelu**

Projektin suunnittelussa selvitetään projektissa käytettävät resurssit, arvioidaan työmäärä ja jaetaan työt projektiin osallistuvien kesken. Projektisuunnitelmaa päivitetään jatkuvasti projektin edetessä, seuraten projektin etenemistä ja kustannuksia (2, s.619–621).

## **Määrittely**

Määrittelyssä luodaan toiminnallinen määrittelydokumentti, jossa kuvataan järjestelmän toiminnalliset ja ei-toiminnalliset vaatimukset. Määrittelyssä laaditaan erilaisia malleja, joilla kuvataan järjestelmän sisältöä, toimintoja ja käyttötapauksia.

Järjestelmän mallintamisessa luodaan abstraktit järjestelmämallit, jotka perustuvat nykyään lähes poikkeuksetta UML-mallinnuskieleen. Näillä malleilla kuvataan järjestelmän sisältö, mutta ei oteta kantaa toteutukseen (2, s.119).

Käyttötapausmallilla kuvataan toimintoja, jotka tarvitsevat käyttäjältä syötteen. Tämän mallin yksinkertaisuudesta johtuen, sen yhteydessä kerrotaan sanallisesti tarkempi kuvaus käyttötapauksen kulusta (2, s.124–125).

Ei-toiminnalliset vaatimukset ovat rajoitteita palveluihin ja toimintoihin, jotka liittyvät ajoituksiin, kehitysprosesseihin ja standardeihin. Nämä vaatimukset koskevat koko järjestelmää (2, s.85).

## **Suunnittelu**

Tässä vaiheessa suunnitellaan, kuinka järjestelmä toteutetaan. Olennaisia asioita ovat muun muassa järjestelmän komponentit, yhteydet ja arkkitehtuuri.

Järjestelmän suunnittelussa suunnitellaan järjestelmän komponentit ja yhteydet asiakkaan vaatimuksien mukaisesti. Näissä suunnitelmissa yleensä huomioidaan myös mahdolliset toteutusvaiheen ongelmatilanteet (2, s.177).

Arkkitehtuurisuunnitelmassa luodaan kerroksittainen rajapintamalli järjestelmästä, josta nähdään, millä kerroksella mikäkin järjestelmän osa toimii (2, s.167–169).

### **Toteutus**

Toteutuksessa järjestelmä ohjelmoidaan suunnittelun pohjalta (2, s.177).

Yleensä toteutusvaiheessa yksikkötestausta suorittaa kyseistä järjestelmän osaa toteuttava henkilö. Joissain tapauksessa toteuttajalle valitaan testauspari, joka suunnittelee ja suorittaa kyseisen järjestelmän osan testauksen (2, s.210–211).

### **Testaus**

Järjestelmätestauksen päämääränä on näyttää asiakkaalle, että järjestelmä toimii ja on käyttöönotettavassa kunnossa. Hyväksymistestauksessa testausta suorittavat loppukäyttäjät ja mahdollisesti sellaiset henkilöt, jotka eivät osallistu järjestelmän kehittämiseen.

Tässä vaiheessa järjestelmää testataan toiminnallisen määrittelydokumentin pohjalta, koska testaajat eivät tiedä, kuinka järjestelmä on rakennettu (2, s.224–230).

### **Palaute**

Palautteen tekee asiakas testauksen pohjalta. Palautteen pohjalta määrittelyä joko muokataan seuraavaan pyrähdykseen tai jatketaan nykyisen määrittelyn pohjalta, kunnes järjestelmä todetaan olevan käyttöönotettavassa tilassa.

### **Käyttöönotto**

Käyttöönotossa järjestelmä asennetaan asiakkaan tiloihin ja otetaan käyttöön. Käyttöönoton yhteydessä suoritetaan myös mahdollinen järjestelmän käyttökoulutus.

## 2.4 Relaatietietokanta

Relaatietietokannassa tietueet säilytetään tauluissa. Tietue yksilöidään uniikilla avaimella, jota voidaan käyttää myös taulujen välisiin yhteyksiin. Näillä taulujen välisillä yhteyksillä pyritään saattamaan tietokanta normaalimuotoon, jossa sama tieto ei esiinny useassa paikassa ja on tehokkaasti saatavissa.

Relaatietietokantoja hallinnoidaan SQL-kieltä käyttäen. SQL soveltuu tietokannan tietojen valitsemiseen, lisäämiseen, poistamiseen ja muokkaamiseen (3, s.3).

### Taulu

Tietokannassa tieto tallennetaan tauluihin, joissa on aina vähintään yksi kenttä johon yksittäiset tiedot tallennetaan. Taulun luominen tietokantaan tapahtuu CREATE TABLE -komentoa käyttäen seuraavasti.

```
CREATE TABLE Kone (kone_id INT NOT NULL, nimi
VARCHAR(40) NOT NULL, aktiivinen BIT NOT NULL)
```

jossa *Kone* on taulun nimi, joka luodaan, ja *kone\_id*, *nimi* ja *aktiivinen* ovat taulun kenttiä. Kentän jälkeen määritellään mahdolliset kentän rajoitteet (Taulukko 2.1) ja kentän tietotyyppi (Taulukko 2.2)(3, s.19–20).

Rajoite	Selite
NOT NULL	Taulun kenttä ei voi olla tyhjä.
PRIMARY KEY	Primary key määrittää taulun tietueen perusavaimeen kuuluvat kentät. Perusavain on tietueen yksilöivä avain, joka voi esiintyä taulun tietueissa vain kerran.
UNIQUE	Unique määrittää, että kaikki taulun tietyissä kentässä ei voi esiintyä samaa tietoa useammin kuin kerran.

Taulukko 2.1 Kenttien rajoitteita (3, s.248–259).

Tietotyyppi	Käyttötarkoitus
BIT	Bitti. Käytetään kyllä/ei tiedon tallentamiseen.
DATETIME	Päivämäärä ja aika. Käytetään päivämäärän ja ajan tallentamiseen.
DECIMAL	Desimaaliluku. Käytetään desimaalilukujen tallentamiseen.
IDENTITY	Laskuri. Käytetään taulun tietueiden yksilöistämiseen. Luo automaattisesti taulun tietueelle yksilöllisen numeron.
INT	Kokonaisluku. Käytetään kokonaislukujen tallentamiseen.
VARCHAR	Teksti. Käytetään minkä tahansa tiedon tallentamiseksi.

Taulukko 2.2 Kenttien tietotyyppejä (3, s.72–73).

### Tietueen lisääminen tauluun

Yksittäistä tauluun tallennettua riviä tietokannassa kutsutaan tietueeksi. Tietueen lisääminen tauluun tapahtuu INSERT-komentoa käyttäen seuraavasti.

```
INSERT INTO Kone (kone_id, nimi, aktiivinen)
VALUES (7, 'Höylä', 1)
```

jossa *Kone* on tietokannan taulu, johon tietue lisätään ja sen jälkeisissä suluissa ovat taulun kentät, joihin halutaan määrittää tietoa. VALUES-avainsanan jälkeen suluissa määritellään kenttiin asetettavat arvot siinä järjestyksessä missä, taulun kentät on aikaisemmin lueteltu (3, s.231–233).

### Taulussa olevan tietueen valitseminen

Tietueen valitseminen taulusta tapahtuu SELECT-komentoa käyttäen seuraavasti.

```
SELECT * FROM Kone WHERE kone_id = 7
```

SELECT-avainsanan jälkeen käytetty \*-merkki tarkoittaa, että taulun tietueista valitaan kaikki tiedot. *Kone* on tietokannan taulu, josta valinta tapahtuu ja WHERE-avainsanan jälkeen asetetaan ehdot, jonka mukaan tietueet valitaan. Tässä tapauksessa valinta tapahtuu koneen yksilöllisen tunnuksen *kone\_id* mukaan, jolloin tietokannasta tulee valituksi korkeintaan yksi tietue (3, s.85–90).

### Tietueen päivittäminen

Taulussa olevan tietueen päivittäminen tapahtuu UPDATE-komentoa käyttäen seuraavasti.

```
UPDATE Kone SET aktiivinen = 0 WHERE kone_id = 7
```

jossa *Kone* on tietokannan taulu, jossa tiedon päivittäminen tapahtuu. SET-avainsanan jälkeen määritellään kentät ja niiden uudet arvot. Tässä tapauksessa kentän aktiivinen arvo muutetaan arvoksi 0, joka kyseisessä järjestelmässä tarkoittaa, että kone ei ole aktiivinen. WHERE-avainsanan jälkeen määritellään ehdot, joiden mukaan päivittäminen tapahtuu. Kyseisessä tapauksessa tietue, jonka yksilöllinen tunnus *kone\_id* on 7, päivitetään uudella arvolla (3, s.242–246).

### Tietueen poistaminen taulusta

Tietueen poistaminen taulusta tapahtuu DELETE-komentoa käyttäen seuraavasti.

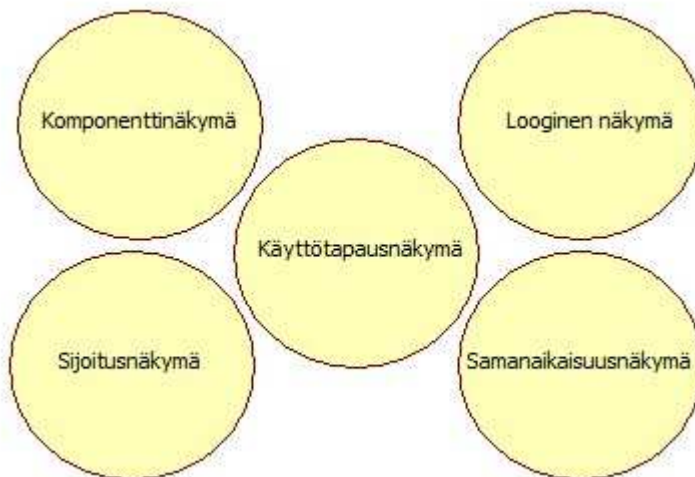
```
DELETE FROM Kone WHERE kone_id = 7
```

jossa *Kone* on tietokannan taulu, josta tietueita halutaan poistaa ja WHERE-avainsanan jälkeen määritellään ehdot, joiden mukaan taulusta poistetaan tietueet. Tässä tapauksessa käytetään koneen yksilöllistä tunnusta *kone\_id* rivien poistamiseen, jolloin tietokannasta poistuu korkeintaan yksi tietue (3, s.238–242).

## 2.5 UML

UML (Unified Modeling Language) on mallinnuskieli, jolla kuvataan järjestelmän toiminnallisuuksia eri näkökulmista (4, s.1–2). UML:n uusin versio on 2.4.1, joka julkaistiin elokuussa 2011. Uusimmassa versiossa kaaviotyyppejä on kolmetoista.

Kehitettävästä järjestelmästä on luotava useita malleja eri näkökulmista, että malleilla pystytään kattamaan kaikki järjestelmän toiminnalliset vaatimukset. Nämä mallit myös ohjaavat järjestelmän kehitystä vaatimusmäärittelyn jälkeisissä vaiheissa (Kuva 2.3)(4, s.238).

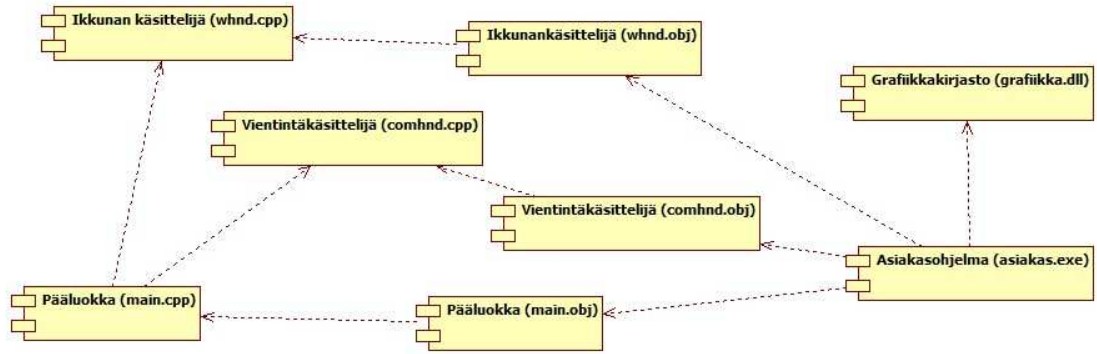


Kuva 2.3 UML:n näkymät

Esimerkkikuvat ovat tehty käyttäen StarUML-ohjelmaa. Allaolevien kaavioiden lisäksi on vielä ajoituskaavio ja vuorovaikutuskaavio. Ajoituskaaviossa kuvataan toimintojen ajoituksia ja vuorovaikutuksia. Vuorovaikutuskaavio annetaan yleiskäsitys vuorovaikutuksesta ja kontrollivirrasta (5).

### **Komponenttikaavio**

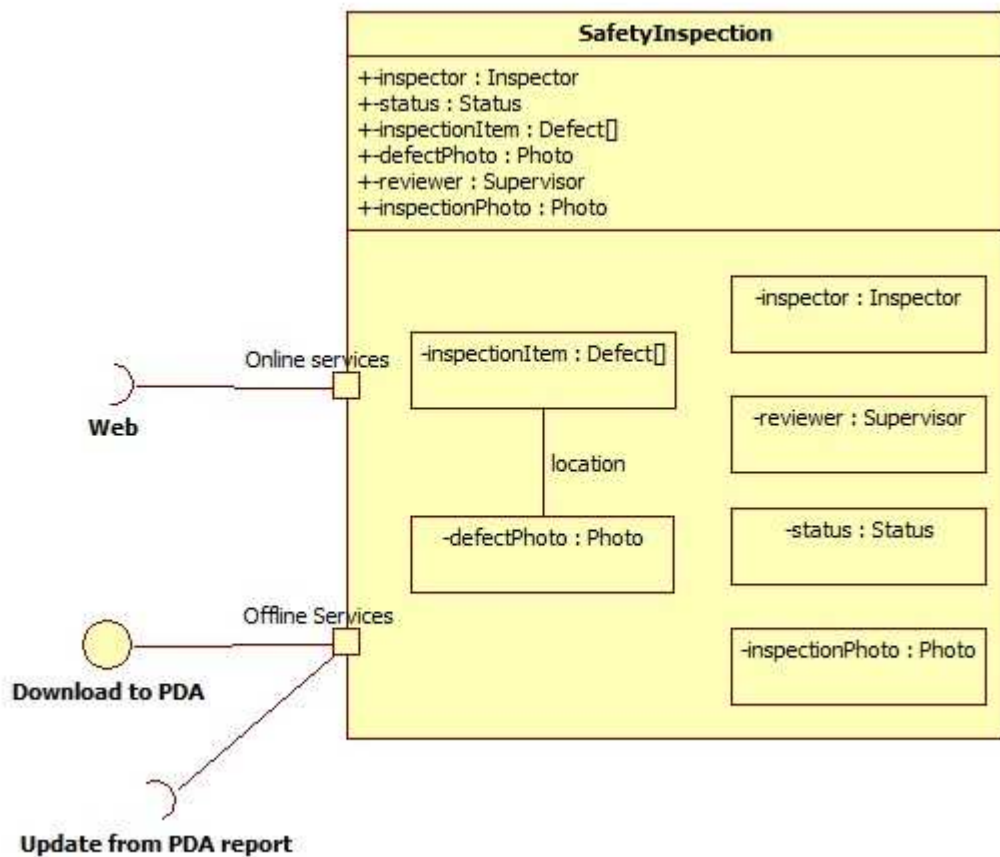
Komponenttikaavio kuvaa loogisen- ja komponenttinäkymän väliset yhteydet näyttämällä lähdekoodin fyysisen rakenteen koodikomponentteina (Kuva 2.4)(4, s.20).



Kuva 2.4 Komponenttikaavio (4, s.20).

### Koostekaavio

Koostekaavio kuvaa luokan sisäistä rakennetta (Kuva 2.5)(5).



Kuva 2.5 Koostekaavio (5).



## Käyttötapauskaavio

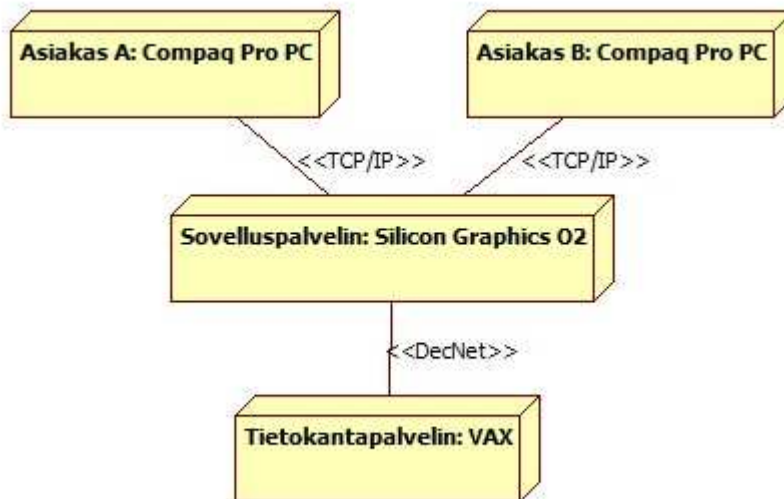
Käyttötapauskaavio kuvaa järjestelmän yhden osan toimintaa. Tällä kaaviolla ei oteta kantaa toteutukseen (Kuva 2.6)(4, s.15).



Kuva 2.6 Käyttötapauskaavio (4, s.15).

## Käyttöönottokaavio

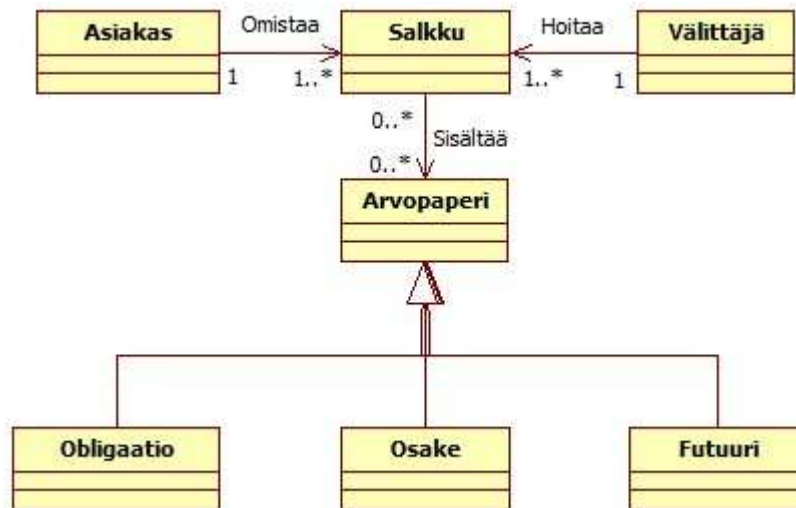
Käyttöönottokaavio kuvaa järjestelmän fyysistä kokoonpanoa, näyttäen järjestelmään kuuluvat laitteet ja ohjelmiston sijoittuminen niille (Kuva 2.7)(4, s.21).



Kuva 2.7 Käyttöönottokaavio (4, s.21).

## Luokkakaavio

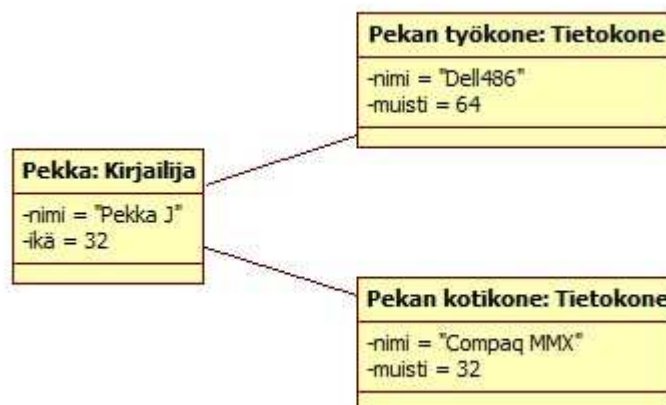
Luokkakaavio kuvaa yleensä vain yhtä osaa järjestelmän luokkarakenteesta. Järjestelmässä voi olla useita luokkakaavioita (Kuva 2.8)(4, s.15).



Kuva 2.8 Luokkakaavio (4, s.16).

## Oliokaavio

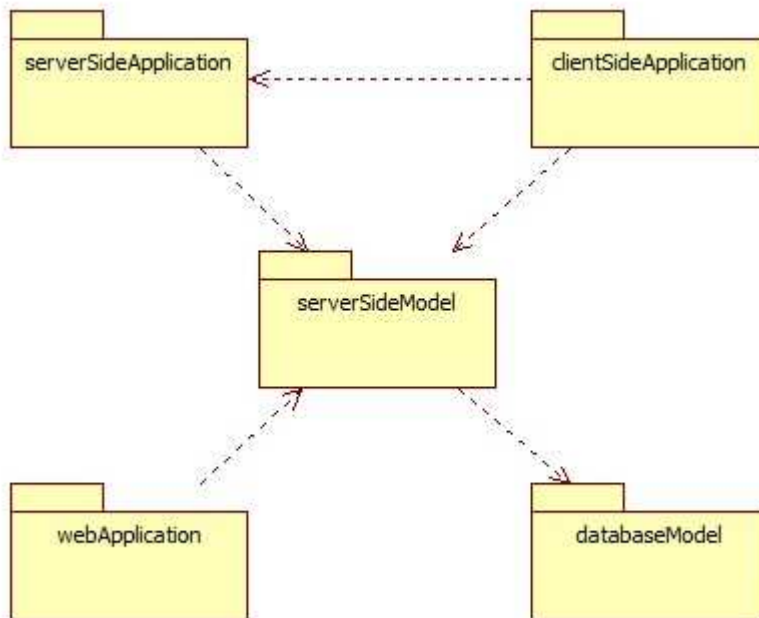
Oliokaavio näyttää esimerkin tilasta, jossa luokat voivat olla tietyllä hetkellä (Kuva 2.9)(4, s.17).



Kuva 2.9 Oliokaavio (4, s.16)

## Pakettikaavio

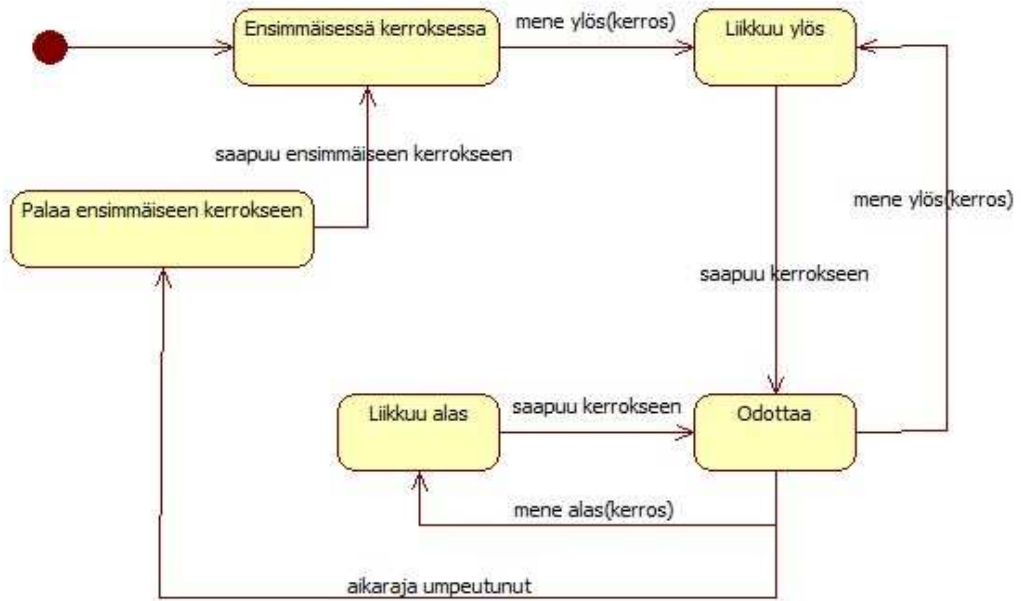
Pakettikaavio näyttää rakenteen ja riippuvuudet alijärjestelmien ja moduulejen välillä (Kuva 2.10)(5).



Kuva 2.10 Pakettikaavio (5).

## Tilakaavio

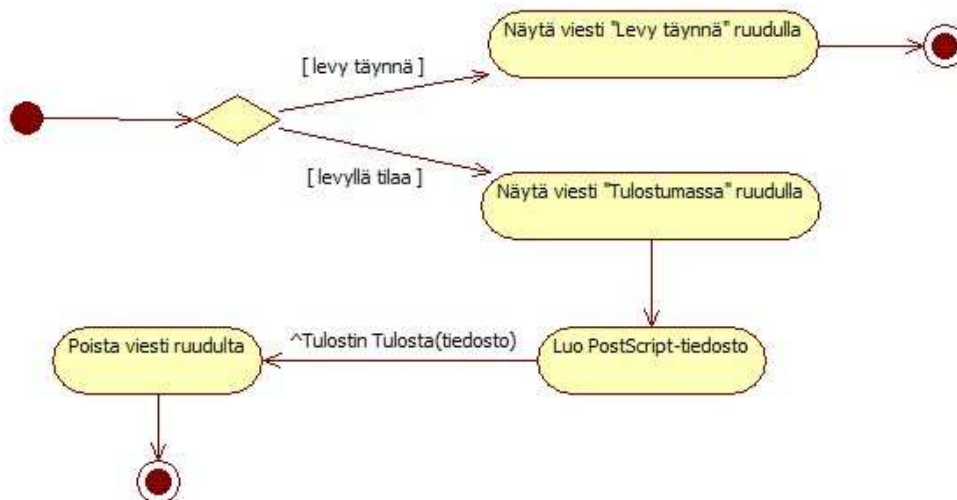
Tilakaavio näyttää tilasiirtymiä aiheuttavat tapahtumat sekä olioiden mahdolliset tilat (Kuva 2.11)(4, s.17).



Kuva 2.11 Tilakaavio (4, s.17).

### Toimintokaavio

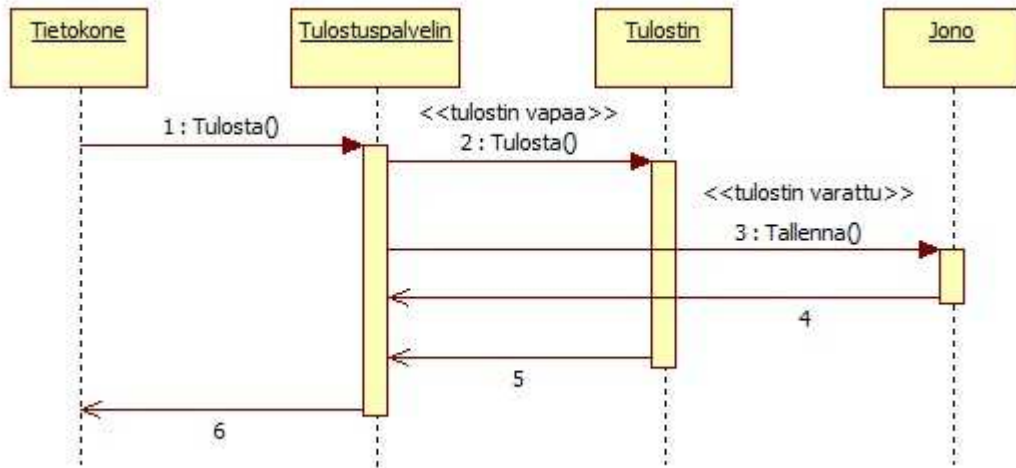
Toimintokaaviossa kuvataan tietyn operaation tapahtumien kulku aikajärjestyksessä (Kuva 2.12)(4, s.19–20).



Kuva 2.12 Toimintokaavio (4, s.19).

## Viestiyhteyskaavio

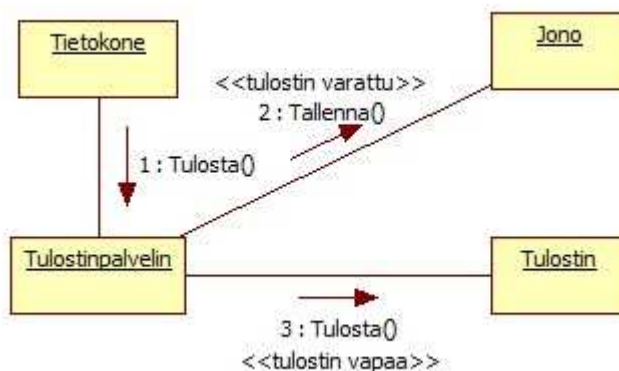
Viestiyhteyskaavio näyttää sarjan viestejä olioiden välillä, kuvaten niiden dynaamista yhteistyötä aikajanalla (Kuva 2.13)(4, s.18).



Kuva 2.13 Viestiyhteyskaavio (4, s.18).

## Yhteistyökaavio

Yhteistyökaavio näyttää sarjan viestejä olioiden välillä, kuvaten niiden dynaamista yhteistyötä ja niiden välisiä suhteita (Kuva 2.14)(4, s.18).

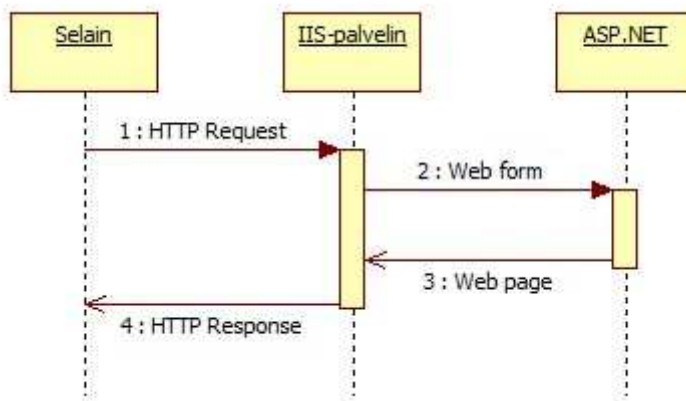


Kuva 2.14 Yhteistyökaavio (4, s.19).

## 2.6 ASP.NET

ASP.NET on web-ohjelmistokehys, jonka avulla voidaan rakentaa dynaamisia verkkolomakkeita. ASP.NET:ssä jokainen palvelimelle tuleva HTTP pyyntö käsitellään erikseen ja näytettävä verkkosivu luodaan käyttäjän tekemien valintojen mukaisesti (6, s.3–6).

Verkkolomake lähetetään IIS-palvelimelle HTTP pyyntönä, joka lähetetään edelleen ASP.NET ajonaikaiseen käsittelyyn. Käsittelyn tuloksena saadaan verkkosivu, joka palautetaan selaimelle näytettäväksi (Kuva 2.15).



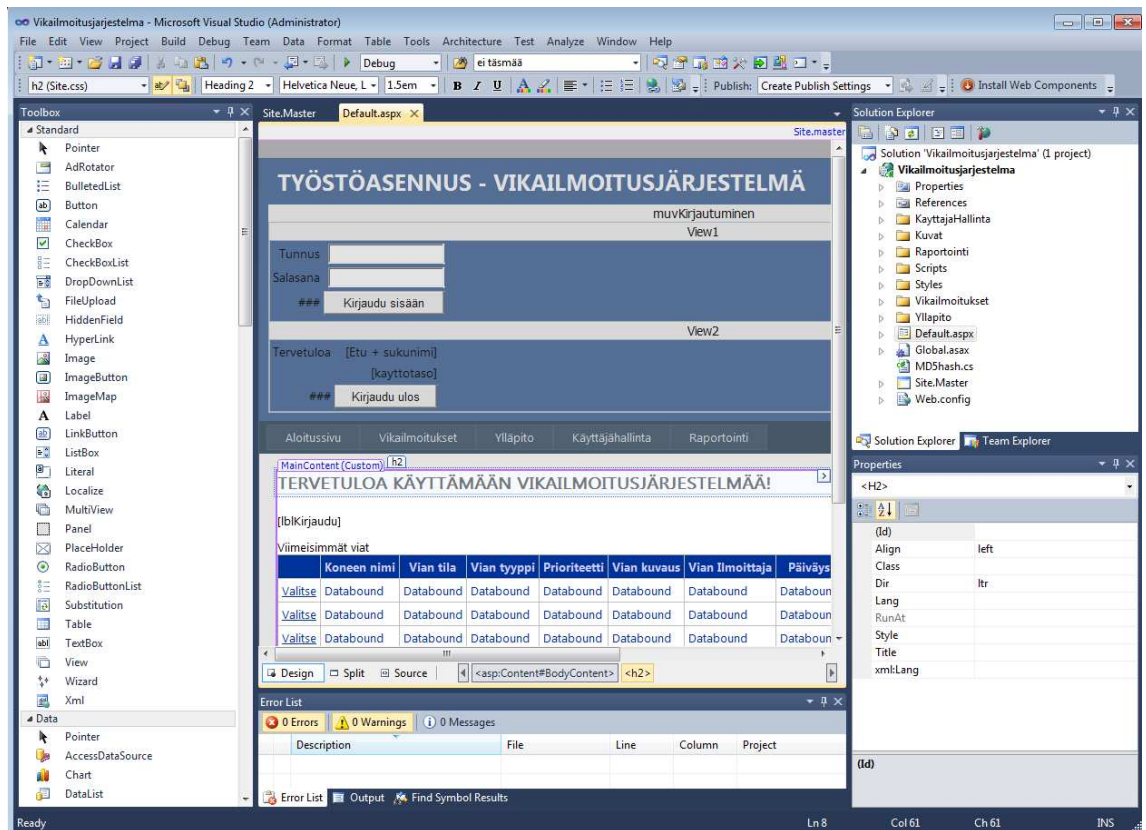
Kuva 2.15 Verkkolomakkeen toiminta (7, s.5).

## 3 Työkalut

Tässä luvussa käydään läpi järjestelmän toteutuksessa käytetyt ohjelmat.

### 3.1 Microsoft Visual Studio 2010

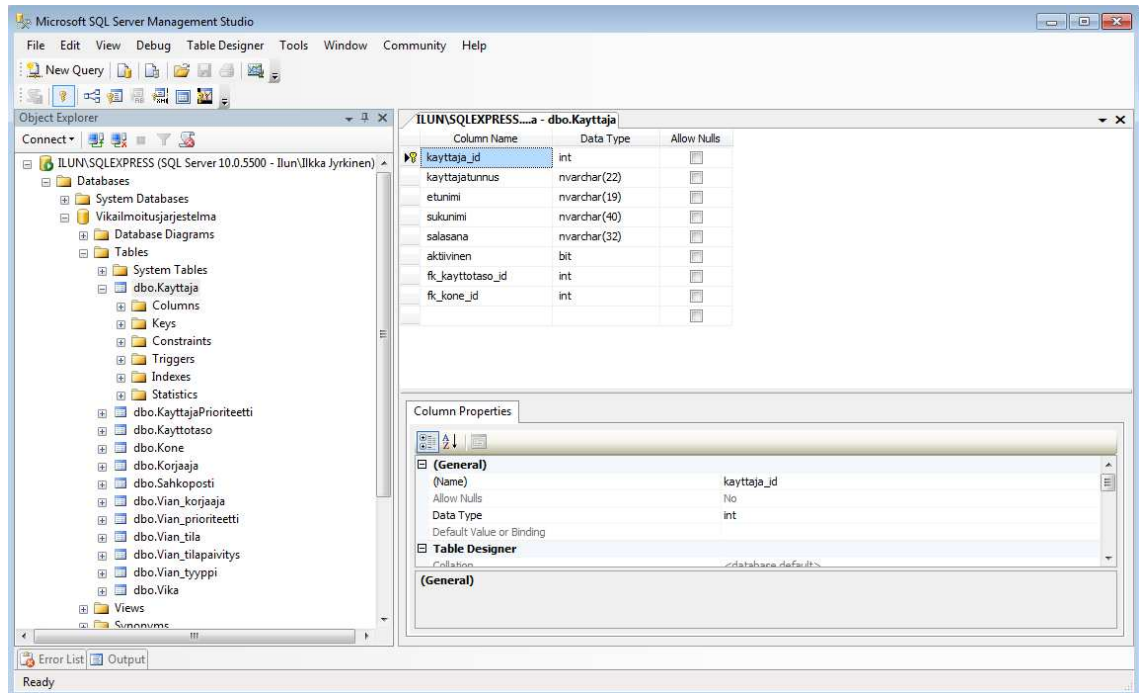
Visual studio on Microsoftin kehittämä integroitu kehitysympäristö, jolla voidaan kehittää muun muassa erilaisia Windows-, verkko- ja mobiilisovelluksia (8)(Kuva 3.1).



Kuva 3.1 Microsoft Visual Studio 2010. Kuvassa näkyy käyttöliittymän suunnittelunäkymä Microsoft Visual Studio 2010 -kehitysympäristöstä.

### 3.2 Microsoft SQL Server Management Studio 2008

SQL Server Management Studio on Microsoftin kehittämä tietokannan hallintatyökalu Microsoft SQL Servereille, joka tarjoaa graafisia työkaluja tietokannan kehittämiseen (9)(Kuva 3.2).

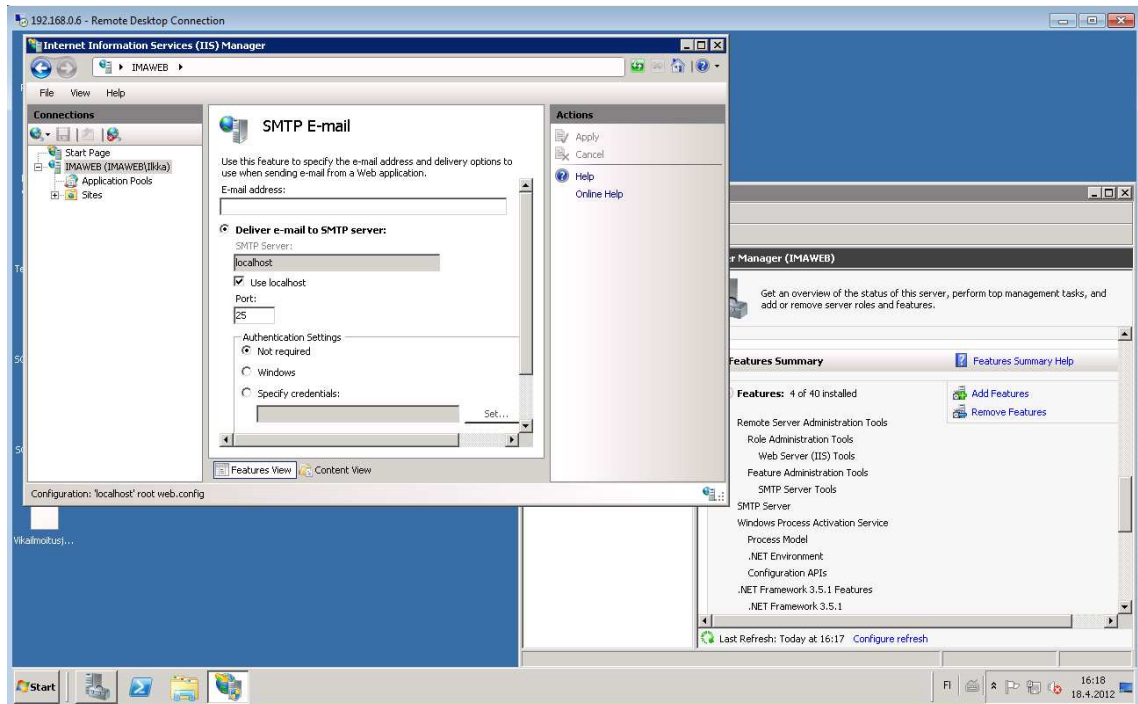


Kuva 3.2 Microsoft SQL Server Management Studio 2008. Kuvassa näkyy tietokannassa olevia tauluja sekä tietokannan taulun muokkaamiseen käytettävä näkymä.



### 3.3 Microsoft Windows Server 2008 R2

Windows Server 2008 R2 -käyttöjärjestelmä tarjoaa alustan IIS-palvelinohjelmistokokonaisuudelle mahdollistaen .NET 4 -versiota käyttävien sivustojen isännöimisen (Kuva 3.3).



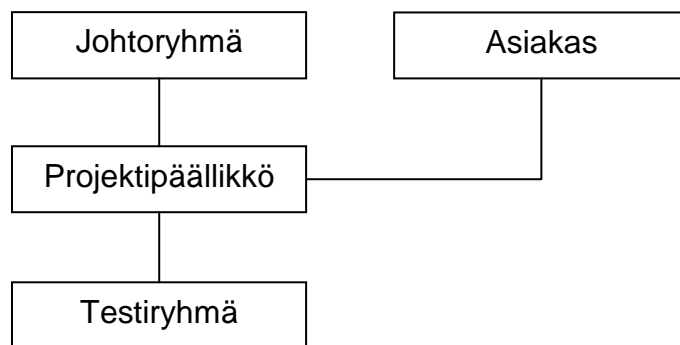
Kuva 3.3 Microsoft Windows Server 2008 R2. Kuvassa on näkymä IIS-hallintasivustosta löytyvistä SMTP-asetuksista sekä osa Server Manager -työkalusta.

## 4 Vikailmoitusjärjestelmä-projektin vaiheet ja organisaatio

Tässä opinnäytetyössä käytiin läpi ohjelmiston kehittämismallin kaikki vaiheet esiselvityksestä järjestelmän käyttöönottoon.

### 4.1 Organisaatio

Kuvassa 4.1 on nähtävissä projektin organisaatio.



Kuva 4.1 Projektin organisaatiokaavio.

Projektiin liittyvät henkilöt ja toimenkuvat:

- Martti Ylä-Jussila, opinnäytetyönohjaaja
- Kari Kattainen, asiakas/ohjaaja
- Tommi Matikainen, asiakas
- Ilkka Jyrkinen, projektipäällikkö
- Teemu Särmölä, testaaja
- Tatu Ylämäki, testaaja

Projektin edetessä ohjausryhmän palaverit käytiin projektipäällikön, johtoryhmän ja asiakkaan kesken. Testiryhmä testasi järjestelmää ennen järjestelmän käyttöönottoa.

### 4.2 Esiselvitys

Aloituspalaverin pohjalta saatiin käsitys asiakkaan tuoteideasta ja nykyisestä järjestelmästä. Tämän lisäksi käytiin myös läpi projektin tavoittelemat hyödyt ja järjestelmän vaatimukset.

Tuoteideana oli toteuttaa järjestelmä, jolla konepajan koneille ilmenevät viat kirjataan sähköiseen muotoon.

Sähköistä järjestelmää ei ollut. Vikailmoitusten kirjaaminen toimi täysin paperilomakkeiden pohjalta.

Sähköisellä vikailmoitusjärjestelmällä tavoiteltiin ajankäytön vähentämistä vikailmoituksen siirtymisessä ja sen käsittelyssä. Tämän lisäksi sähköisellä järjestelmällä mahdollistettiin raportointi. Raportoinnin avulla pystytään seuraamaan mahdollisia vikojen toistuvuuksia ja korjaustapoja, jonka johdosta korjausvaiheen ajankäyttöä pystytään mahdollisesti pienentämään tai jopa kokonaan ennaltaehkäisemään.

Esiselvityksen yhteydessä selvisi myös järjestelmän vaatimukset, jotka muodostuivat asiakkaan tarpeiden ja resurssien mukaisesti.

#### **4.3 Projektisuunnitelma**

Projektisuunnitelmassa asetettiin projektille päättämisehdot ja tavoitteet, päätettiin projektin seurannasta, asetettiin standardit ja määräykset. Tässä vaiheessa myös luotiin projektille aikataulu, jota päivitettiin seurannan yhteydessä.

Tavoitteena oli saada määritellyksi ja toteutetuksi kaikilta ominaisuuksiltaan toimiva vikailmoitusjärjestelmä, joka on kauttaaltaan yksinkertainen käyttää ja helppo ylläpitää.

Projekti päättyi, kun järjestelmä oli hyväksytysti käyttöön otettu sekä dokumentit toimitettu.

Asiakkaan kanssa pidettiin palavereja aina, jos asiat kaipasivat tarkempaa määrittelyä, testausta, hyväksyntää tai muuta neuvottelua. Opinnäytetyön ohjausryhmän kanssa palavereja pidettiin noin kolmen viikon välein.

Sivuston ulkoasun tyylinä käytettiin Microsoftin tyyliä selkeyden ja helppokäyttöisyyden vuoksi.

Tekijällä on tekijänoikeudet, jos asiakas ei niitä lunasta. Asiakkaalla on käyttöoikeus tuotteen koekäytössä hyväksytyyn versioon.

#### **4.4 Toiminnallinen määrittely**

Järjestelmän pienen koon vuoksi toiminnallinen määrittely oli lähinnä suuntaa antava dokumentti, jota päivitettiin järjestelmän kehittyessä ja suurimmat muutokset tulivat palavereiden yhteydessä.

Suunnitteludokumenttia ei kirjoitettu, koska järjestelmän katsottiin olevan niin pieni, että suunnittelu pystyttiin tekemään toteutuksen kanssa rinnakkain.

#### **4.5 Toteutus**

Projektin pienen koon vuoksi toteutus ja testaus suoritettiin ketterällä kehittämismallilla, jolloin saatiin järjestelmä kehitettyä osa osalta ja nähtiin helposti suunta, johon oltiin menossa. Järjestelmä toteutettiin pääasiassa toiminnallisen määrittelydokumentin pohjalta. Joissakin tapauksissa jouduttiin keskustelemaan asiakkaan kanssa, koska valittu toteutustapa tuntui huonolta jonkin toiminnallisuuden toteuttamiseksi.

#### **4.6 Testaus**

Toteuttaja testasi järjestelmän alustavasti toteutusvaiheessa. Ennen järjestelmän käyttöönottoa testausta suoritti asiakas, toteuttaja ja kaksi Saimaan ammattikorkeakoulun testauskurssilaista, Teemu Särmölä ja Tatu Ylämäki.

#### **4.7 Käyttöönotto**

Ennen käyttöönottoa asiakas järjesti pienimuotoisen koulutuksen työntekijöille, jossa jokainen työntekijä loi vikailmoituksen harjoitusmielessä.

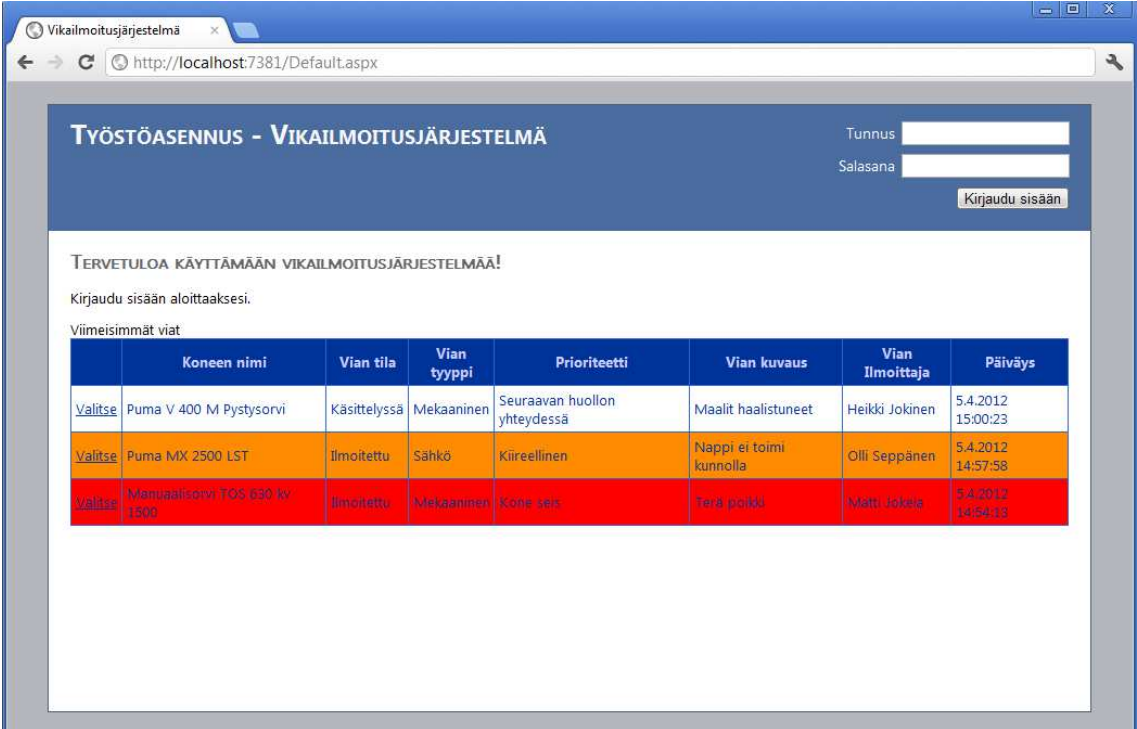
## 5 Vikailmoitusjärjestelmä

Tässä osiossa esitellään Vikailmoitusjärjestelmän käyttöliittymät ja tietokanta.

### 5.1 Sisäänkirjautuminen

Sisäänkirjautuminen tapahtuu syöttämällä käyttäjätunnus ja salasana Aloitussivun oikeaan yläreunaan. Jos käyttäjä ei ole kirjautunut sisään, on käytettävissä vain Aloitussivu (Kuva 5.1).

Jos käyttäjän tunnus on sama kuin salasana, voidaan kirjautua kirjoittamalla pelkästään tunnus.



TYÖSTÖASENNUS - VIKAILMOITUSJÄRJESTELMÄ

Tunnus

Salasana

TERVETULOA KÄYTTÄMÄÄN VIKAILMOITUSJÄRJESTELMÄÄ!

Kirjaudu sisään aloittaaksesi.

Viimeisimmät viat

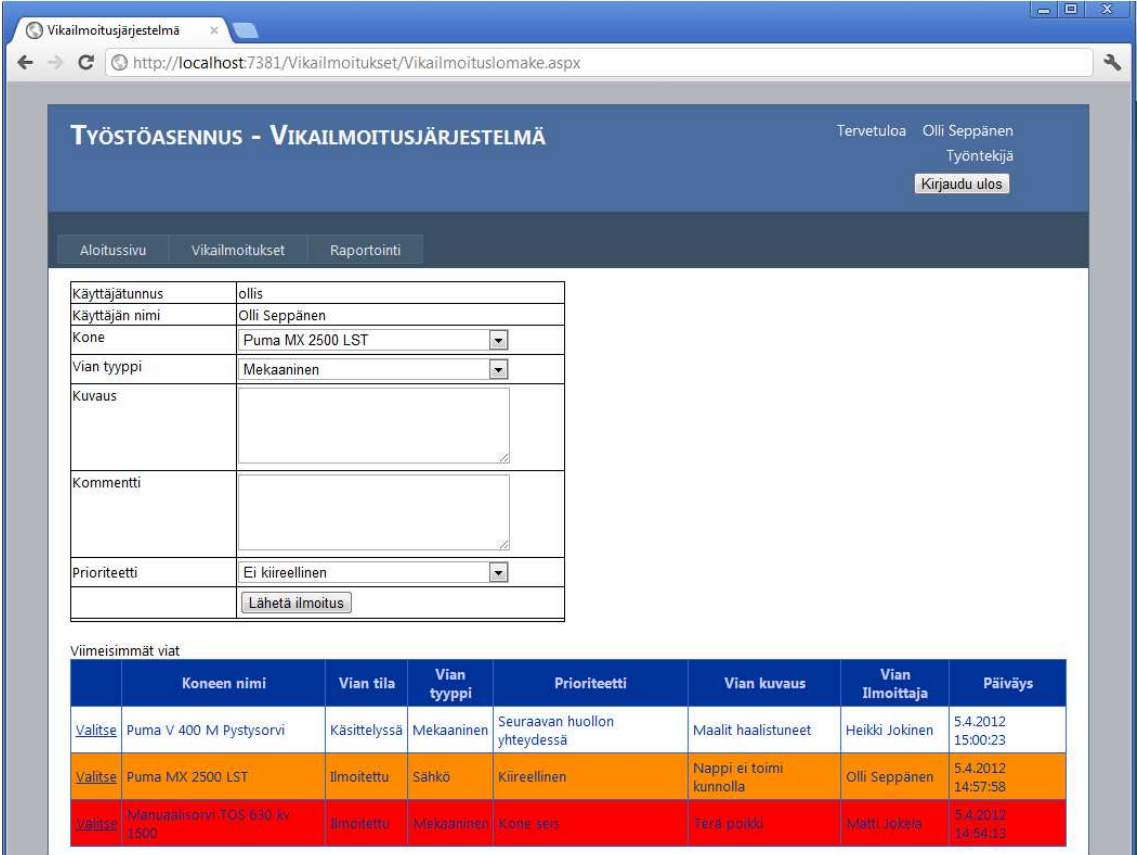
	Koneen nimi	Vian tila	Vian tyyppi	Prioriteetti	Vian kuvaus	Vian Ilmoittaja	Päiväys
Valitse	Puma V 400 M Pystysorvi	Käsittelyssä	Mekaaninen	Seuraavan huollon yhteydessä	Maalit haalistuneet	Heikki Jokinen	5.4.2012 15:00:23
Valitse	Puma MX 2500 LST	Ilmoitettu	Sähkö	Kiireellinen	Nappi ei toimi kunnolla	Olli Seppänen	5.4.2012 14:57:58
Valitse	Manuaalisorvi TOS 630 kv 1500	Ilmoitettu	Mekaaninen	Kone seis	Tera poikki	Matti Jokela	5.4.2012 14:54:33

Kuva 5.1 Aloitussivu. Aloitussivulla listataan viimeisimmät viat ja mahdollistetaan sisäänkirjautuminen.

## 5.2 Vikailmoituksen jättäminen

Koneita käyttävä työntekijä ohjataan suoraan Vikailmoituslomake-sivustolle sisäänkirjautumisen jälkeen. Lomakkeessa on automaattisesti täyttyvinä kenttinä Käyttäjätunnus, Käyttäjän nimi ja Kone, jotka määräytyvät tietokannassa olevien tietojen mukaan (Kuva 5.2). Tekstikentistä Kuvaus-kenttä on pakollinen ja Kommentti-kenttä vapaaehtoinen.

Lomakkeen alapuolelta löytyy myös viimeisimpien vikojen listaus, josta käyttäjä voi katsoa, onko vika jo mahdollisesti listattu.



The screenshot shows a web browser window with the URL `http://localhost:7381/Vikailmoitukset/Vikailmoituslomake.aspx`. The page title is "TYÖSTÖASENNUS - VIKAILMOITUSJÄRJESTELMÄ". The user is logged in as "Olli Seppänen" (Työntekijä). The page has three tabs: "Aloitussivu", "Vikailmoitukset", and "Raportointi".

The main form contains the following fields:

Käyttäjätunnus	ollis
Käyttäjän nimi	Olli Seppänen
Kone	Puma MX 2500 LST
Vian tyyppi	Mekaaninen
Kuvaus	<input type="text"/>
Kommentti	<input type="text"/>
Prioriteetti	Ei kiireellinen
<input type="button" value="Lähetä ilmoitus"/>	

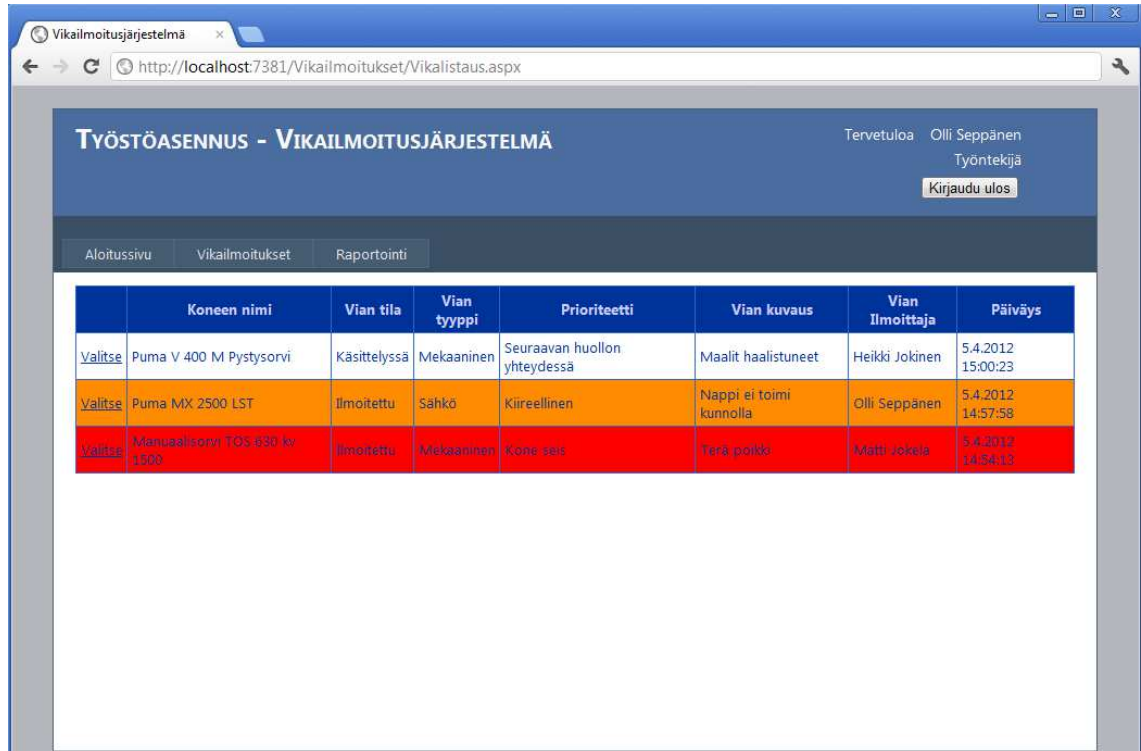
Below the form is a table titled "Viimeisimmät viat" (Recent faults):

	Koneen nimi	Vian tila	Vian tyyppi	Prioriteetti	Vian kuvaus	Vian Ilmoittaja	Päiväys
<a href="#">Valitse</a>	Puma V 400 M Pystysorvi	Käsittelyssä	Mekaaninen	Seuraavan huollon yhteydessä	Maalit haalistuneet	Heikki Jokinen	5.4.2012 15:00:23
<a href="#">Valitse</a>	Puma MX 2500 LST	Ilmoitettu	Sähkö	Kiireellinen	Nappi ei toimi kunnolla	Olli Seppänen	5.4.2012 14:57:58
<a href="#">Valitse</a>	Manuaalisorvi TOS 630 kv 1500	Ilmoitettu	Mekaaninen	Kone seis	Tera poikki	Matti Jokela	5.4.2012 14:54:13

Kuva 5.2 Vikailmoituslomake. Kuvassa näkyy Vikailmoituslomake ja listaus viimeisimmistä vioista.

### 5.3 Vikalistaus

Vikalistaus on nähtävissä Aloitus-, Vikailmoitukset- ja Vikailmoituslomakesivulla (Kuva 5.3). Listasta voidaan valita vika käsiteltäväksi.



	Koneen nimi	Vian tila	Vian tyyppi	Prioriteetti	Vian kuvaus	Vian Ilmoittaja	Päiväys
<a href="#">Valitse</a>	Puma V 400 M Pystysorvi	Käsittelyssä	Mekaaninen	Seuraavan huollon yhteydessä	Maalit haalistuneet	Heikki Jokinen	5.4.2012 15:00:23
<a href="#">Valitse</a>	Puma MX 2500 LST	Ilmoitettu	Sähkö	Kiireellinen	Nappi ei toimi kunnolla	Olli Seppänen	5.4.2012 14:57:58
<a href="#">Valitse</a>	Manuaalisorvi TOS 830 kv 1500	Ilmoitettu	Mekaaninen	Kone seis	Terä poikki	Matti Jokela	5.4.2012 14:56:13

Kuva 5.3 Vikalistaus.

## 5.4 Vikailmoituksen tilan muuttaminen

Vikailmoituksen tilaa pystytään muuttamaan vian Katselmointi-sivulla. Vialle pystytään asettamaan korjaaja ja vaihtamaan vian tilaa (Kuva 5.4).

Kun vika kuitataan korjatuksi, häviää se vikalistauksista ja on nähtävissä vain raportoinnin kautta.

Vian kuvausta, tyyppiä ja prioriteettia pystytään muokkaamaan Muokkaa-nappia painamalla, jonka jälkeen näkymä hieman muuttuu. Vikailmoitus voidaan tarvittaessa poistaa Muokkaa-näkymän kautta.

The screenshot shows a web browser window with the URL `http://localhost:7381/Vikailmoitukset/Vika.aspx?vika_id=2`. The page title is 'TYÖSTÖASENNUS - VIKAILMOITUSJÄRJESTELMÄ'. The user is logged in as 'Jukka Holttinen', the administrator. The navigation menu includes 'Aloitussivu', 'Vikailmoitukset', 'Ylläpito', 'Käyttäjähallinta', and 'Raportointi'. The main content area has a search bar 'Hae vika tunnuksella' and a 'Hae vika' button. Below is a form for editing the ticket details:

Vian tunnus:	2
Kuvaus:	Maalit haalistuneet
Vian tyyppi:	Mekaaninen
Vian prioriteetti:	Seuraavan huollon yhteydessä
Koneen nimi:	Puma V 400 M Pystysorvi

Buttons:

Form fields:

- Tila:  (dropdown)
- Vian korjaaja:  (dropdown)
- Kommentti:

Buttons: ,

Vian tilapäivitykset

	Päiväys	Vian tila	Käyttäjä	Kommentti
<a href="#">Poista</a>	5.4.2012 15:00:23	Käsittelyssä	Jukka Holttinen	Maalit tilattu
<a href="#">Poista</a>	5.4.2012 14:56:40	Ilmoitettu	Heikki Jokinen	

Vian korjaajat

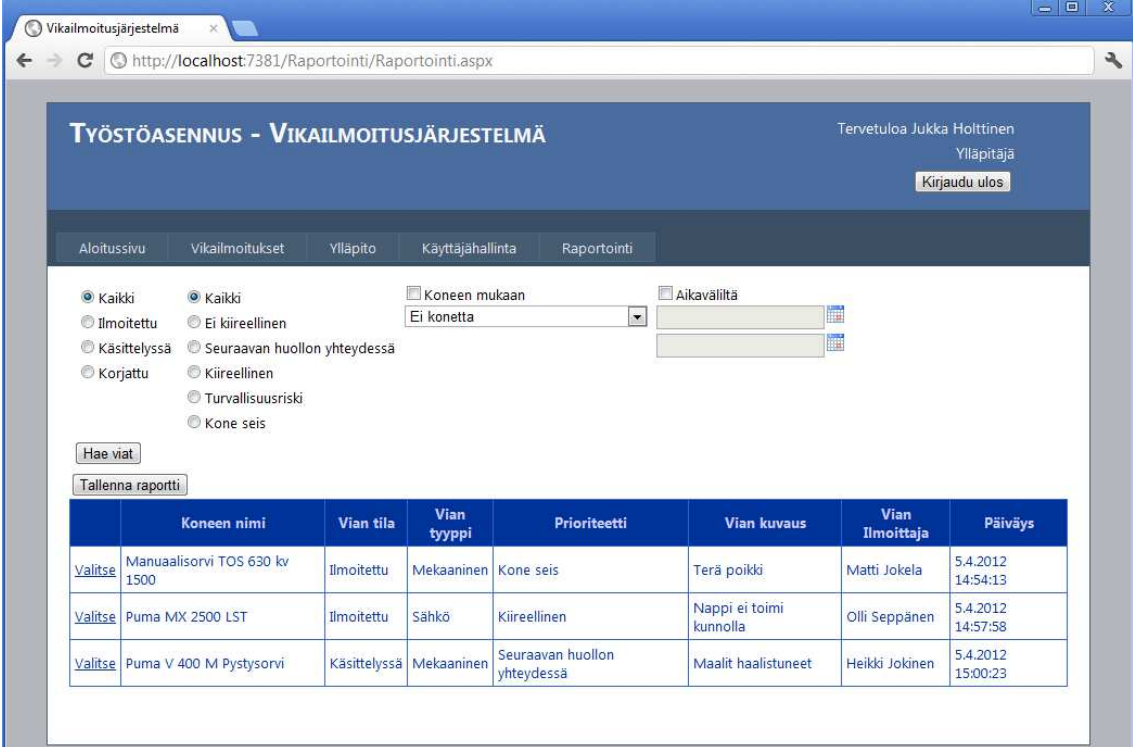
	Korjaaja
<a href="#">Poista</a>	Konepalvelu Oy

Kuva 5.4 Vian käsittely. Kuvassa näkyy vian tilapäivitykset, vikaan liitetty korjaaja, sekä mahdollisuus lisätä tilapäivitys ja korjaaja.



## 5.5 Raportointi

Ilmaantuneista vioista voidaan luoda raportti käyttäen erilaisia hakukriteerejä. Mahdollisia kriteerejä on vian tila, prioriteetti, kone ja aikaväli (Kuva 5.5). Raportti voidaan tallentaa Excel-asiakirjaksi ja tämän kautta myös tulostaa.



The screenshot shows a web browser window with the URL `http://localhost:7381/Raportointi/Raportointi.aspx`. The page title is "TYÖSTÖASENNUS - VIKAILMOITUSJÄRJESTELMÄ". The user is identified as "Tervetuloa Jukka Holttinen" and "Ylläpitäjä". There is a "Kirjaudu ulos" button. The navigation menu includes "Aloitussivu", "Vikailmoitukset", "Ylläpito", "Käyttäjähallinta", and "Raportointi".

The reporting interface includes the following filters:

- Machine name:  Kaikki,  Ilmoitettu,  Käsitteilyssä,  Korjattu
- Status:  Kaikki,  Ei kiireellinen,  Seuraavan huollon yhteydessä,  Kiireellinen,  Turvallisuusriski,  Kone seis
- Machine type:  Koneen mukaan,
- Time range:  Aikaväliltä

Buttons: "Hae viat", "Tallenna raportti"

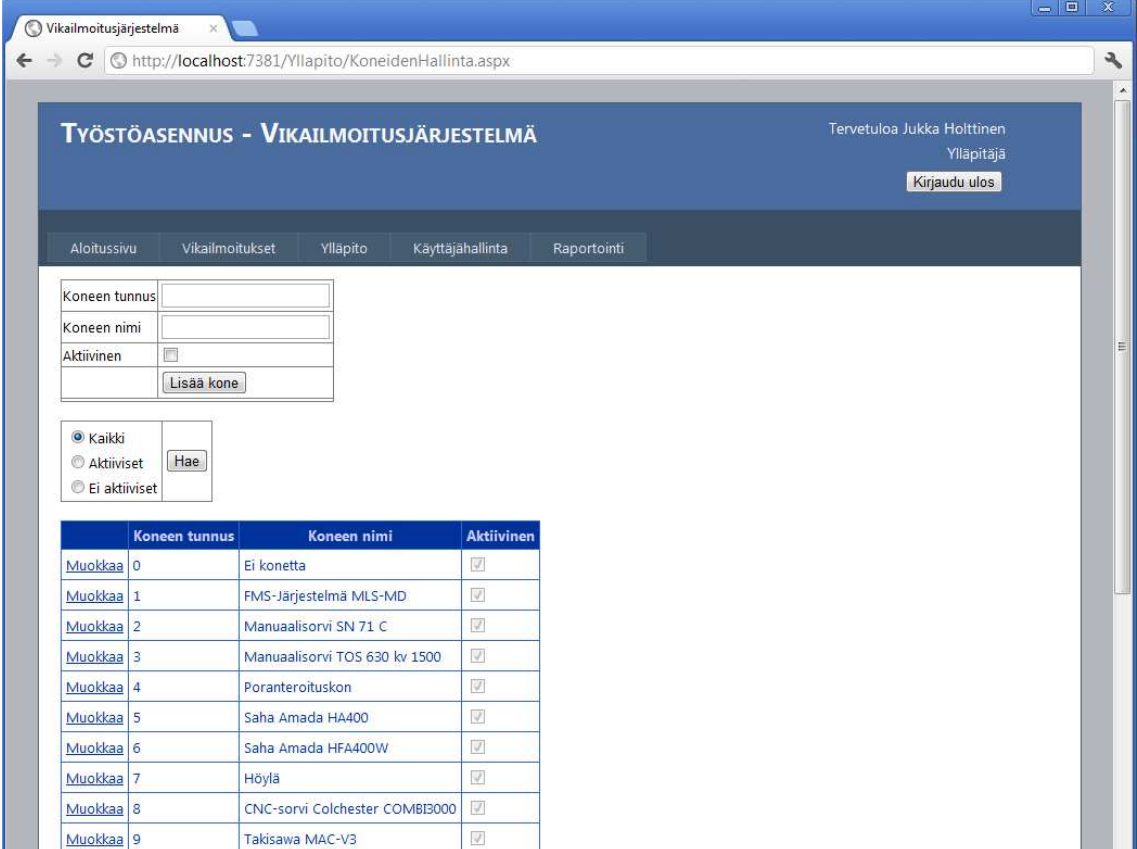
	Koneen nimi	Vian tila	Vian tyyppi	Prioriteetti	Vian kuvaus	Vian Ilmoittaja	Päiväys
<a href="#">Valitse</a>	Manuaalisorvi TOS 630 kv 1500	Ilmoitettu	Mekaaninen	Kone seis	Terä poikki	Matti Jokela	5.4.2012 14:54:13
<a href="#">Valitse</a>	Puma MX 2500 LST	Ilmoitettu	Sähkö	Kiireellinen	Nappi ei toimi kunnolla	Olli Seppänen	5.4.2012 14:57:58
<a href="#">Valitse</a>	Puma V 400 M Pystysorvi	Käsitteilyssä	Mekaaninen	Seuraavan huollon yhteydessä	Maalit haalistuneet	Heikki Jokinen	5.4.2012 15:00:23

Kuva 5.5 Raportointi. Kuvassa näkyy mahdolliset hakukriteerit ja vikalistaus hakukriteereiden perusteella.

## 5.6 Koneiden hallinta

Koneiden Hallinta-sivulla voidaan lisätä uusia sekä hallita jo tietokannassa olevia koneita (Kuva 5.6).

Koneiden aktiivinen tila vaikuttaa koneen näkyvyyteen vikailmoituslomakkeen alavetovalikossa.



The screenshot shows a web browser window with the URL `http://localhost:7381/Yllapito/KoneidenHallinta.aspx`. The page title is "TYÖSTÖASENNUS - VIKAILMOITUSJÄRJESTELMÄ" and the user is logged in as "Jukka Holttinen, Ylläpitäjä". The navigation menu includes "Aloitussivu", "Vikailmoitukset", "Ylläpito", "Käyttäjähallinta", and "Raportointi".

The main content area contains a form for adding a new machine with the following fields:

- Koneen tunnus:
- Koneen nimi:
- Aktiivinen:
- 

Below the form is a search filter with radio buttons for "Kaikki" (selected), "Aktiiviset", and "Ei aktiiviset", and a "Hae" button.

The table below lists the machines in the database:

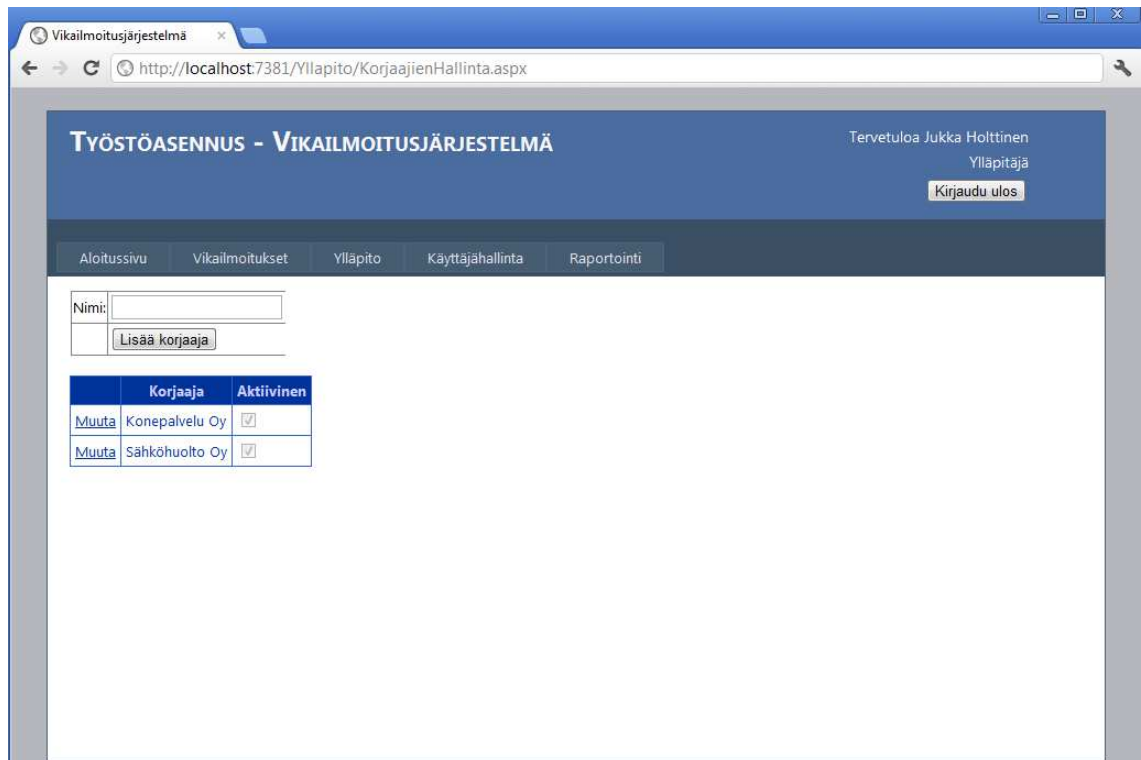
	Koneen tunnus	Koneen nimi	Aktiivinen
<a href="#">Muokkaa</a>	0	Ei konetta	<input checked="" type="checkbox"/>
<a href="#">Muokkaa</a>	1	FMS-Järjestelmä MLS-MD	<input checked="" type="checkbox"/>
<a href="#">Muokkaa</a>	2	Manuaalisorvi SN 71 C	<input checked="" type="checkbox"/>
<a href="#">Muokkaa</a>	3	Manuaalisorvi TOS 630 kv 1500	<input checked="" type="checkbox"/>
<a href="#">Muokkaa</a>	4	Poranteroituskon	<input checked="" type="checkbox"/>
<a href="#">Muokkaa</a>	5	Saha Amada HA400	<input checked="" type="checkbox"/>
<a href="#">Muokkaa</a>	6	Saha Amada HFA400W	<input checked="" type="checkbox"/>
<a href="#">Muokkaa</a>	7	Höylä	<input checked="" type="checkbox"/>
<a href="#">Muokkaa</a>	8	CNC-sorvi Colchester COMBB000	<input checked="" type="checkbox"/>
<a href="#">Muokkaa</a>	9	Takisawa MAC-V3	<input checked="" type="checkbox"/>

Kuva 5.6 Koneiden hallinta. Kuvassa näkyy koneen luomiseen käytettävä lomake ja lista koneista.

## 5.7 Korjaajien hallinta

Korjaajien Hallinta-sivulla voidaan lisätä uusia sekä muokata vanhoja korjaajia (Kuva 5.7).

Korjaajan aktiivinen tila vaikuttaa korjaajan näkyvyyteen vian katselmointisivulla valittavien korjaajien alasvetovalikossa.

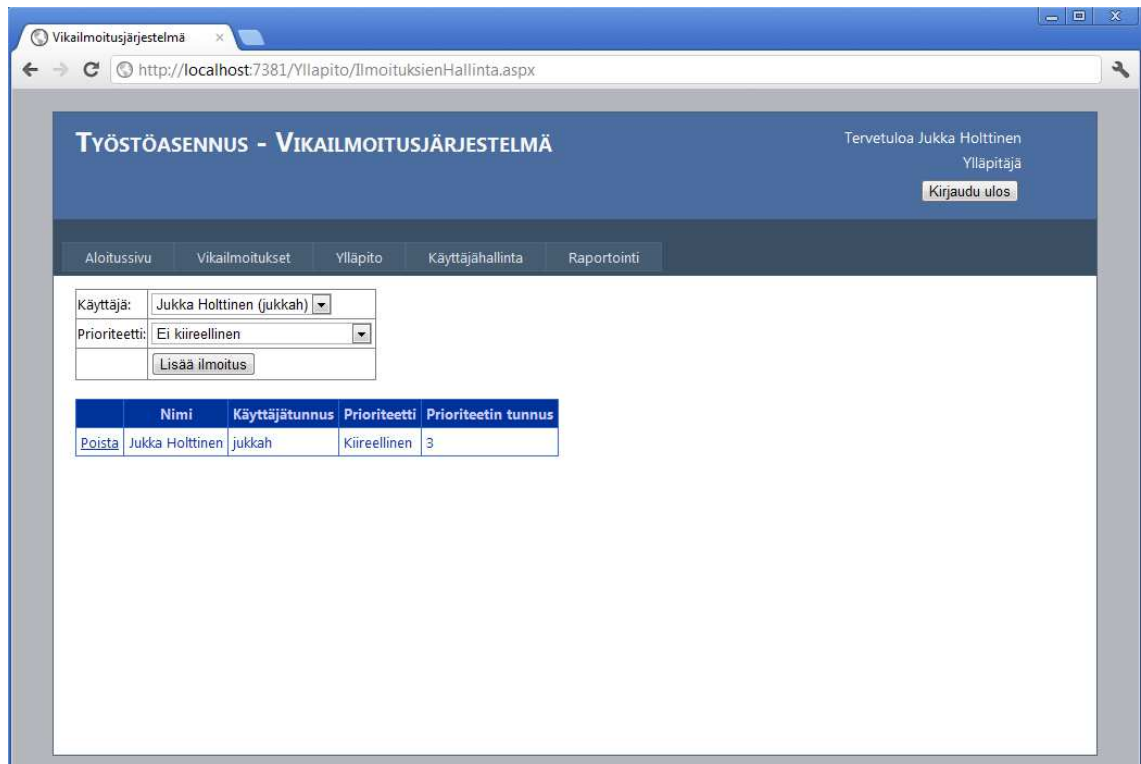


Kuva 5.7 Korjaajien hallinta. Kuvassa näkyy korjaajan luomiseen käytettävä lomake ja lista korjaajista.

## 5.8 Postituslistan hallinta

Postituslistan Hallinta-sivulla voidaan lisätä ja poistaa henkilöitä postituslistalta vian prioriteetin mukaan (Kuva 5.8). Alasvetovalikossa näkyvät vain henkilöt, joilla on sähköposti määritettynä.

Postituslistaa käytetään, kun vikailmoitus jätetään käsiteltäväksi. Postituslistalla olevat henkilöt saavat sähköposti-ilmoituksen, kun tietyn prioriteetin vika lisätään järjestelmään.



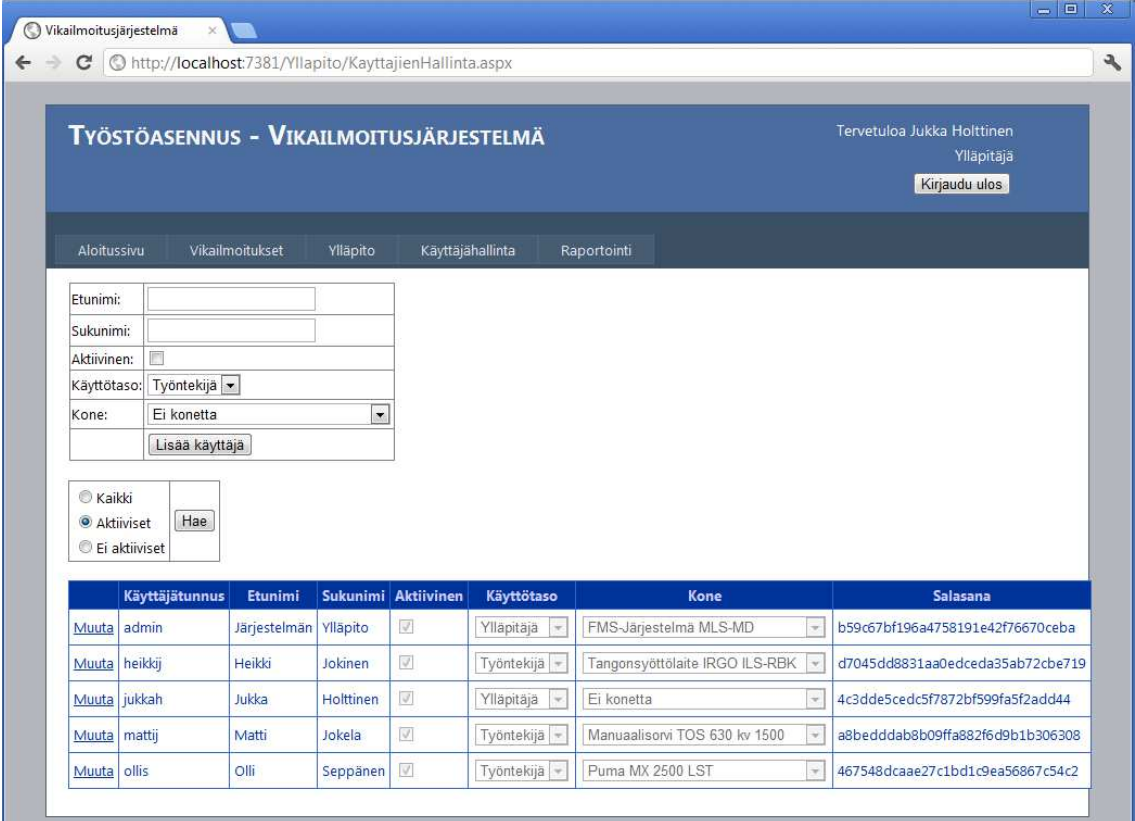
Kuva 5.8 Postituslistan hallinta. Kuvassa näkyy postituslista ja työkalu käyttäjän lisäämiselle postituslistalle prioriteetin mukaan.

## 5.9 Käyttäjätunnuksien hallinta

Käyttäjätunnuksien hallintasivulla voidaan luoda uusia ja muokata vanhoja käyttäjätunnuksia (Kuva 5.9).

Käyttäjätunnuksien salasanat ovat MD5-tiivistettyjä. Tämä estää ylläpitäjiä näkemästä käyttäjien mahdolliset henkilökohtaiset salasanat.

Käyttäjätunnuksen aktiivinen tila määrittelee, voidaanko käyttäjätunnuksella kirjautua sisään.



The screenshot shows a web browser window with the URL `http://localhost:7381/Yllapito/KayttajienHallinta.aspx`. The page title is "TYÖSTÖASENNUS - VIKAILMOITUSJÄRJESTELMÄ". The user is logged in as "Jukka Holttinen, Ylläpitäjä". The navigation menu includes "Aloitussivu", "Vikailmoitukset", "Ylläpito", "Käyttäjähallinta", and "Raportointi".

The form for creating a new user includes the following fields:

- Etunimi:
- Sukunimi:
- Aktiivinen:
- Käyttötaso: Työntekijä (dropdown)
- Kone: Ei konetta (dropdown)
- Lisää käyttäjä (button)

Below the form is a search section with radio buttons for "Kaikki", "Aktiiviset", and "Ei aktiiviset", and a "Hae" button.

The table below lists the existing users:

	Käyttäjätunnus	Etunimi	Sukunimi	Aktiivinen	Käyttötaso	Kone	Salasana
<a href="#">Muuta</a>	admin	Järjestelmän	Ylläpito	<input checked="" type="checkbox"/>	Ylläpitäjä	FMS-Järjestelmä MLS-MD	b59c67bf196a4758191e42f76670ceba
<a href="#">Muuta</a>	heikkj	Heikki	Jokinen	<input checked="" type="checkbox"/>	Työntekijä	Tangonsyöttölaite IRGO ILS-RBK	d7045dd8831aa0edceda35ab72cb719
<a href="#">Muuta</a>	jukkah	Jukka	Holttinen	<input checked="" type="checkbox"/>	Ylläpitäjä	Ei konetta	4c3dde5cedc5f7872bf599fa5f2add44
<a href="#">Muuta</a>	mattij	Matti	Jokela	<input checked="" type="checkbox"/>	Työntekijä	Manuaalisovi TOS 630 kv 1500	a8bedddab8b09ffa882f6d9b1b306308
<a href="#">Muuta</a>	ollis	Olli	Seppänen	<input checked="" type="checkbox"/>	Työntekijä	Puma MX 2500 LST	467548dcaae27c1bd1c9ea56867c54c2

Kuva 5.9 Käyttäjätunnuksien hallinta. Kuvassa näkyy lomake käyttäjätunnuksen luomiselle ja listaus käyttäjätunnuksista.

## 5.10 Käyttäjähallinta

Käyttäjähallinnan kautta Ylläpitäjä- ja Työnjohto-tason tunnuksilla voidaan hallita omia tietoja, kuten nimeä ja oletuskonetta, muuttaa salasanaa sekä lisätä ja poistaa sähköpostiosoitteita tunnukselta (Kuva 5.10).

Vikailmoitusjärjestelmä

http://localhost:7381/KayttajaHallinta/KayttajaHallinta.aspx

**TYÖSTÖASENNUS - VIKAILMOITUSJÄRJESTELMÄ**

Tervetuloa Jukka Holtttinen  
Ylläpitäjä  
[Kirjaudu ulos](#)

Aloitussivu Vikailmoitukset Ylläpito **Käyttäjähallinta** Raportointi

Etunimi: Jukka  
Sukunimi: Holtttinen  
Kone: Ei konetta  
[Tallenna](#)

Vanha salasana:   
Uusi salasana:   
Salasana uudestaan:   
[Tallenna](#)

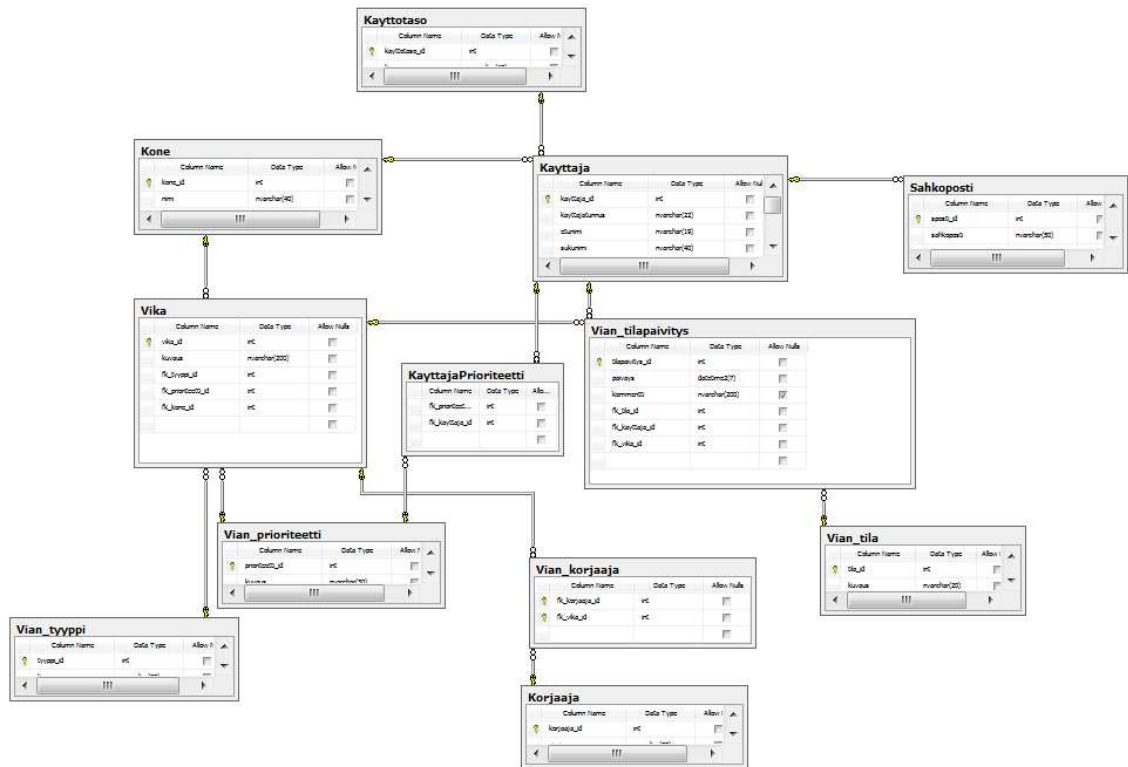
[Lisää sähköposti](#)

Sähköpostiosoitteet  
Ei asetettuja sähköpostiosoitteita

Kuva 5.10 Käyttäjähallinta.

## 5.11 Tietokanta

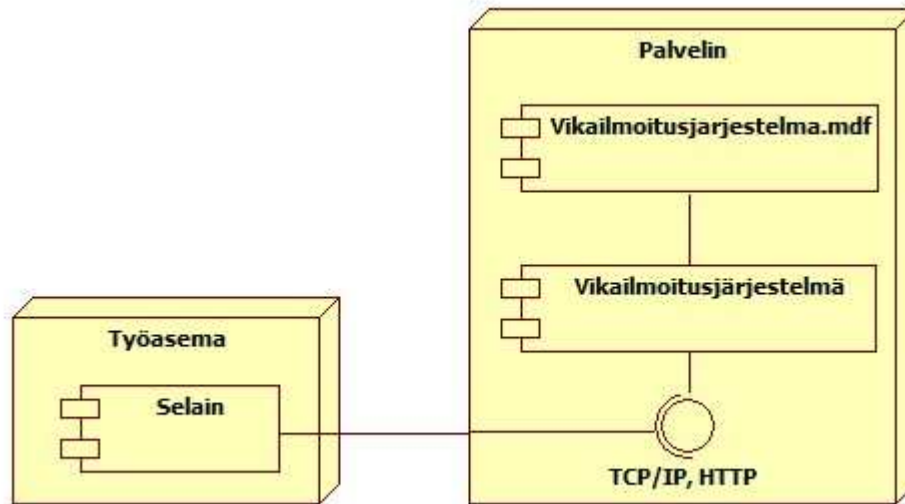
Järjestelmän tietokanta luotiin SQL Server Management Studioossa tietokantakaaviota muokkaamalla (Kuva 5.11). Tätä kaaviota muokkaamalla voidaan luoda tietokantaan tauluja ilman SQL-kielen kirjoittamista.



Kuva 5.11 Microsoft SQL Server Management Studio 2008, tietokantakaavio. Kuvassa näkyy vikailmoitusjärjestelmässä käytetyn tietokannan tietokantakaavio.

## 5.12 Käyttöönottoakaavio

Vikailmoitusjärjestelmä pyörii erillisellä palvelimella, johon otetaan yhteys työasemilta selaimella lähiverkon kautta (Kuva 5.12).



Kuva 5.12 Vikailmoitusjärjestelmän käyttöönottoakaavio.



## 6 Yhteenveto ja pohdinta

Opinnäytetyön tavoitteena oli toteuttaa asiakkaan vaatimuksien mukainen vikailmoitusjärjestelmä. Tärkein ei-toiminnallinen vaatimus oli luoda käyttöliittymät niin yksinkertaisiksi, että ne ovat helppoja oppia. Nämä tavoitteet saavutettiin joustavan lähestymistavan avulla, jossa mukauduttiin käyttäjien tarpeisiin projektin edetessä.

Projekti eteni lähes ongelmitta, mutta itse opiskelulle olisi pitänyt varata enemmän aikaa, koska minulla oli vain vähän aiempaa kokemusta ASP.NET -tekniikasta, mutta silti ongelmista selvittiin erittäin vähällä.

Työtavat ja ratkaisut toimivat suurelta osalta ongelmitta, mutta itse toteutusvaiheessa työpari olisi varmasti tuonut toteutukseen uusia näkökulmia ja toteutustapoja, joita ei välttämättä yksin työskennellessä tule ajatelleeksi.

Projektin aikana opin projektin etenemisestä paljon. Eduksi voisi sanoa, että sain viedä projektin alusta loppuun itse ja nähdä kaikki projektin vaiheet ja oppia projektin edetessä. Tämän lisäksi laajensin käsitystäni järjestelmän toteutukseen käytetyistä tekniikoista, ja toteutusvaiheessa tuli esiin uusia toteutustapoja.

Järjestelmä toteutettiin alusta loppuun suunnitelman mukaisesti, mutta jatkokehitysmahdollisuuksia on esimerkiksi raportoinnin puolella, jossa graafiset raportit voisivat olla eduksi asiakkaalle. Järjestelmään on myös helppo toteuttaa uusia ominaisuuksia selkeän toteutuksen ansiosta.

Tässä vaiheessa järjestelmän hyötyjä on vielä vaikea arvioida tarkasti. Alussa arvioitiin paperilomakkeen täyttöön ja viemiseen kuluvaksi ajaksi jopa viisi minuuttia. Tämä aika pystyttiin lyhentämään järjestelmän avulla yhteen minuuttiin. Tämän ajansäästön lisäksi mahdollistettiin helppo vian korjauksen seuranta sekä vikojen raportointi, jonka avulla käyttäjät on helppo pitää ajan tasalla konepajan koneilla olevista vioista ja niiden toistuvuuksista. Näiden asioiden ansiosta tehostunut valvonta lyhentää myös vika-aikoja.

## **Kuvat**

Kuva 2.1 Vesiputousmalli s.8

Kuva 2.2 Ketterä kehittämismalli s.9

Kuva 2.3 UML:n näkymät s.15

Kuva 2.4 Komponenttikaavio s.16

Kuva 2.5 Koostekaavio s.16

Kuva 2.6 Käyttötapauskaavio s.17

Kuva 2.7 Käyttöönottoakaavio s.17

Kuva 2.8 Luokkakaavio s.18

Kuva 2.9 Oliokaavio s.18

Kuva 2.10 Pakettikaavio s.19

Kuva 2.11 Tilakaavio s.20

Kuva 2.12 Toimintokaavio s.20

Kuva 2.13 Viestiyhteyksikaavio s.21

Kuva 2.14 Yhteistyökaavio s.21

Kuva 2.15 Verkkolomakkeen toiminta s.22

Kuva 3.1 Microsoft Visual Studio 2010 s.23

Kuva 3.2 Microsoft SQL Server Management Studio 2008 s.24

Kuva 3.3 Microsoft Windows Server 2008 R2 s.25

Kuva 4.1 Projektin organisaatiokaavio s.26

Kuva 5.1 Aloitus sivu s.29

Kuva 5.2 Vikailmoituslomake s.30

Kuva 5.3 Vikalistaus s.31

Kuva 5.4 Vian käsittely s.32

Kuva 5.5 Raportointi s.33

Kuva 5.6 Koneiden hallinta s.34

Kuva 5.7 Korjaajien hallinta s.35

Kuva 5.8 Postituslistan hallinta s.36

Kuva 5.9 Käyttäjätunnusten hallinta s.37

Kuva 5.10 Käyttäjähallinta s.38

Kuva 5.11 Microsoft SQL Server Management Studio 2008, tietokantakaavio s.39

Kuva 5.12 Vikailmoitusjärjestelmän käyttöönottokaavio s.40

## Taulukot

Taulukko 2.1 Kenttien rajoitteita s.12

Taulukko 2.2 Kenttien tietotyyppiä s.13

## Lähteet

1. Dooley, J. 2011. Software Development and Professional Practice. Apress. United States of America.
2. Sommerville, I. 2011. Software Engineering. Pearson. United States of America.
3. Groff, J., Weinberg, P. & Oppel, A. 2010. The Complete Reference SQL, 3rd Edition. The McGraw-Hill Companies. United States.
4. Eriksson, H. & Penker, M. 2000. UML. Gummerus. Jyväskylä.
5. Visual Paradigm. UML Modeling. <http://www.visual-paradigm.com/VPGallery/diagrams/> Luettu 30.5.2012
6. Esposito, D. 2011. Programming Microsoft ASP.NET 4. Microsoft Press. United States of America.
7. Bochicchio, D., Mostarda, S., & Stanctis, M.D. 2011. ASP.NET 4.0 in Practice. Manning. United States of America.
8. Wikipedia 2012. Microsoft Visual Studio. [http://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio), Luettu 10.4.2012
9. Wikipedia 2012. SQL Server Management Studio. [http://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server#SQL\\_Server\\_Management\\_Studio](http://en.wikipedia.org/wiki/Microsoft_SQL_Server#SQL_Server_Management_Studio), Luettu 10.4.2012