

# Weather Service Client

ASE-6010 Harjoitustyö 1

Mikael Eriksson  
mikael.eriksson@student.tut.fi  
218605

## Sisällysluettelo

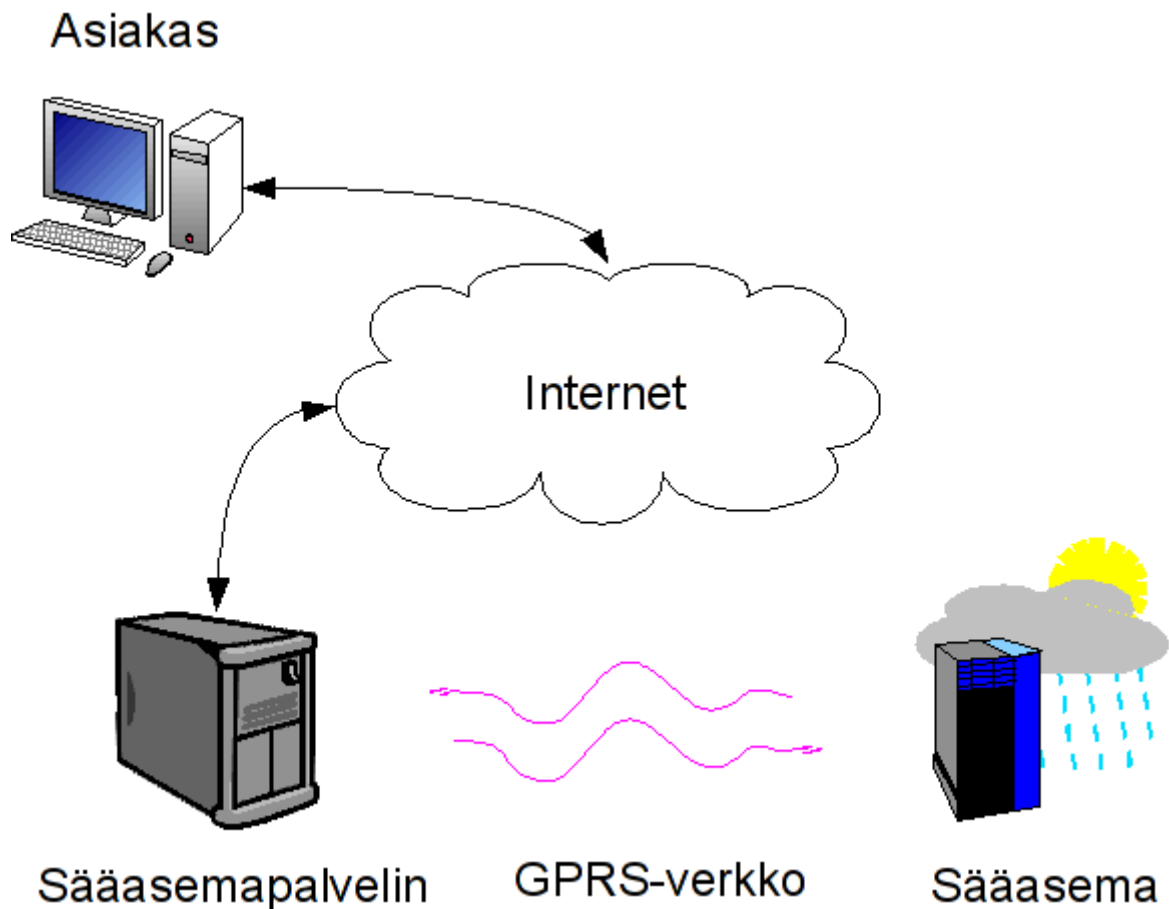
1.	Sääasema .....	2
1.1	Järjestelmän rakenne .....	2
1.2	Sääaseman tarjoama mittaustieto .....	2
2.	Client-sovellus .....	3
2.1	Yhteys sääasemaan .....	3
2.2	Mittaustietojen säilöminen .....	3
2.3	Käyttöliittymä .....	4
3.	Tietoturva-analyysi .....	4

## 1. Sääasema

### 1.1 Järjestelmän rakenne

"PK-yrityksesi on saanut tehtäväkseen toteuttaa asiakkaalle sääaseman käyttöliittymäohjelmiston. Sääasemasta oleva dokumentaatio on huonoa ja siitä on saatavilla vain [Webservice-kuvaus](#) (WSDL).

Asiakasyritys ei ole antanut valmista käyttöliittymäkuvausta, eikä vaadi tiettyä toteutustapaa. Ainoa vaatimus on, että käyttöliittymän tulee kyetä näyttämään ainakin kyseisen päivän historiatieto sekä tämän hetkinen säätila."



Kuva 1. Järjestelmän rakenne.

Kuvassa 1 on esitetty järjestelmän rakenne ja toiminta. Tarkoituksena on suunnitella ja toteuttaa kuvassa ylhäällä vasemmalla esitetyn asiakkaan käyttöliittymäohjelmisto.

### 1.2 Sääaseman tarjoama mittaustieto

Systeemitekniikan laitoksen katolla sijaitseva Vaisala ROSA –sääasema tuottaa erinäistä tietoa vallitsevasta säätilasta. Tehtävää varten annettu rajapinta tarjoaa sääaseman GPS-koordinaatit ja aikaleiman, mutta varsinaiseen säätilaan liittyen ainoastaan lämpötilan suureen ja yksikön. Muita sääaseman mittaamia suureita, kuten sademäärää tai tuulen nopeutta ei saada ulos käyttäen annettua rajapintaa.

Rajapinta esittää mittaustiedon WSMeasurement luokan olioiden avulla, jonka sisältö näkyy alla olevassa kuvassa 2.

```
public class WSMMeasurement implements java.io.Serializable {  
    private java.lang.String gps;  
    private java.lang.String timestamp;  
    private java.lang.String unit;  
    private double value;
```

Kuva 2. Mittaustiedon tallentamiseksi käytetty luokka

Sääasemalta on mahdollista kysyä 1-100 viimeisintä mittaustietoa. Response-tyyppinen vastaus sisältää mittaustiedot WSMMeasurement-objektien taulukkona, kuva 3.

```
public class Response implements java.io.Serializable {  
    private fi.tut.rd.ain.schemas.WSMMeasurement[] measurements;
```

Kuva 3. Response-luokan sisältö

## 2. Client-sovellus

### 2.1 Yhteys sääasemaan

Ohjelman käynnistyessä sovellus luo WSClient-tyyppisen olion joka ottaa yhteyden sääasemaan ja hakee 100 viimeisintä mittaustietoa. Olio tallentaa saadut mittaustiedot tietokantaan ja nukkuu 60 sekuntia. Tämän jälkeen olio hakee minuutin välein vain yhden uuden mittaustiedon sääasemalta ja lisää sen tietokantaan. Tietokanta tarkastaa itse, onko lisätty alkio jo valmiiksi säilössä. Yhteys sääasemaan luodaan omaan säikeeseensä ja se jatkaa toimintaansa ohjelman sulkemiseen saakka.

### 2.2 Mittaustietojen säilöminen

Vaatimusten mukaan käyttöliittymän on kyettävä näyttämään kyseisen päivän historiatieto sekä tämän hetken säätila. Sääasema päivittää tietoja noin 1,5 minuutin välein, joten vuorokaudessa mittaustietoa kertyy alle 1000 objektia. Sääasema puolestaan tarjoaa maksimissaan vai 100 viimeisintä mittaustietoa, joten sitä vanhempi mittaustieto on säilöttävä lokaalisti. Toteutuksessa päädyttiin säilöämään 1000 viimeisintä mittaustieto-objektia linkitettyyn listaan, jolloin käytettävissä on mittaustiedot noin 25 tunnin ajalta taaksepäin. Tiedon säilömistä varten tehtiin oma DataBase-luokka, joka tarjoaa ulospäin alla olevan kuvan mukaiset metodit.

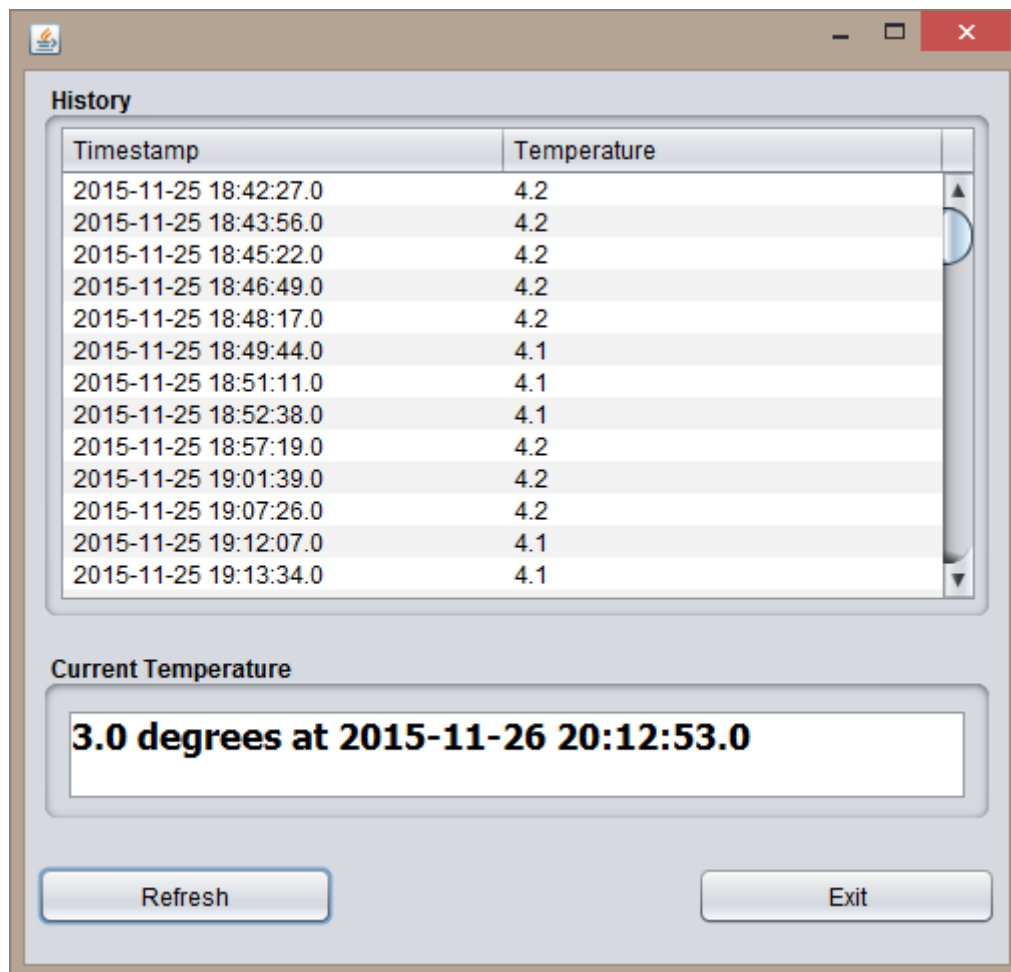
```
public class DataBase {  
    public void updateList(WSMMeasurement[] array)  
    public Object getCurrent()  
    public LinkedList getList()
```

Kuva 4. Mittaustietojen säilömiseen käytett luokka

Ensimmäinen metodi updateList ottaa argumenttina WSMMeasurement-objektien taulukon ja lisää uudet taulukon alkiot tietosäiliöön.

## 2.3 Käyttöliittymä

Käyttöliittymän ulkoasu on esitetty kuvassa 5.



Kuva 5. Käyttöliittymän ulkoasu

History-taulukko näyttää kaikki lokaalisti tallessa olevat mittaustiedot; taulukon selaaminen tapahtuu vierityspalkilla. Current Temperature kertoo viimeisimmän sääasemalta saadun mittaustiedon ja ajankohdan. Näkymän päivittäminen tapahtuu Refresh-nappia painamalla, joka ohjaa ainoastaan näkymää. Uusimmat tiedot sääasemalta päivittyvät tietosäiliöön automaattisesti minuutin välein. Exit-nappi tekee sen mitä lupaa - sulkee ohjelman.

## 3. Tietoturva-analyysi

Ohjelma ottaa yhteyden ennalta määrättyyn palvelimeen, joten riskiä kommunikoida vieraan palvelimen kanssa vahingossa ei sinällään ole. Riski yhteyden kaappaamiseen on kuitenkin olemassa, eikä tätä ole huomioitu ohjelmassa millään muotoa.

Palvelimelta saadaan vastauksena taulukollinen olioita, mutta Java ei ole hyvin tarkka siitä, mitä olioita taulukkoon voidaan tallentaa. On siis mahdollista, että vastauksena saatu taulukko on täynnä aivan muun tyyppisiä olioita, kuin mitä palvelimelta on tilattu. Riskiä voitaisiin pyrkiä pienentämään suorittamalla tarkastus, jossa taulukon alkiot käydään läpi ja tarkastetaan niiden laillisuus. Tässä tapauksessa sitä ei kuitenkaan tehdä. Käytännössä ohjelma on yksi iso tietoturva-aukko.

## **Liitteet**

WSClient.java

DataBase.java

StartGUI.java