**PA International Web presence**

PA International produces Geo-location software and data logging units (*devices*). The department for data logging and geo location is fully developed, but the department for data presentation and unit monitoring is falling behind.  
  
The PA International has requested a project with a new website and a web-service for

1. device registration and upload of logged data, location and status, and
2. monitoring/tracking of device status and readings

The system should be able to handle a large number of simultaneous devices (several thousand), as well as a large number of simultaneous web page and service users (several hundreds).

## Devices

There are two types of devices; geo-stationary and dynamic. The geo-stationary devices have a fixed location and are often not equipped with a GPS received (are unaware of its own location). The dynamic devices change position and are always equipped with a GPS received. All devices communicate with the server. Each device has a unique ID (GUID, MACID, ...). Each device has a location, indicated by a GPS coordinate (Latitude, Longitude, Elevation?). The geographical datum is WGS84 (Earth).

## Server

The server refers to one or more physical servers equipped with load-balancing software and routers (bandwidth and load-balancing is not a part of this project, it is considered that sufficient bandwidth is available). Some considerations about the requirements for the server may be included for completeness.

The devices communicate with the server, and the server stores the data in a database and exposes the device data via a website and a web service.

The server groups devices based on owner and administrator. This means that a given login only give access to a subsection of the total number of devices.

## Website

All websites access must be authenticated. It should be considered what the costs (performance) are of using HTTPS (TSL) as opposed to normal HTTP.

The website should be written as Java servlets and JSP, with use of JavaScript (JQuery) and AJAX for client side performance.

### Users

The website must be able display information about the devices; their location (google maps), their status (operational, error, …), their history (status log), their data (graphical (Javascript) or tables (XSLT)), etc. The website must have an application look-and-feel (i.e. perceived performance for the client is an issue), and should therefore rely heavily on Javascript and AJAX. It is allowed to have these technologies as a requirement for the client browsers (no mobile device browser support). If mobile devices is to be supported it will be a new project.

### Administrators

The administrator website is used for administrating devices (adding new devices, assigning ownership, location, ID, …) and creating new zones (see google maps).

## Google maps

The google map must be used to display the devices, their position, status and last reading (where applicable). The map must be fluently zoom-able. Furthermore, for dynamic devices, google maps must be used to track the device’s movements over a period of time. Runtime updates of device status and location must be supported (AJAX).

Zones may be created indicating a geographical area on google maps (mathematically a polygon), which allows for status read back and control of the devices within this zone as one, e.g. if the entire zone is translucent green then all devices are functioning, if it is translucent red then one or more devices in that zone has reported an error.

Zones may be nested so a zone can contain other zones.

If a device has a location within a given zone it belongs to this zone, unless the zone relationship is manually overridden (special circumstances may dictate that a device should not be part of any zone, or should be part of a specific zone regardless of its geographical location). Dynamic devices can therefore jump from zone to zone.

Restrictions may be imposed to simplify the design, e.g. no overlapping zones, no mixed content (either a zone consists of devices or other zones, not both), limited number of zones or devices are permitted in a single zone (a maximum of N devices or zones are allowed in a zone. If this number is exceeded the system must create a new zone by intelligent splitting), etc.

## Database

The persistence layer is a GIS-aware relational database (PostgreSQL with PostGIS, Oracle 11g, Microsoft SQL Server 2008, ...). This database is considered installed, configured and maintained and is not a part of the project. Some considerations about the requirements for the database may be included for completeness.

## Web service

An XML language for information exchange and an XML schema to validate these must be developed or adopted. In order to use current industry standards on geographic information candidates for adoption are the Google [KML](http://en.wikipedia.org/wiki/Keyhole_Markup_Language) XML language or the OpenGIS [GML](http://en.wikipedia.org/wiki/Geography_Markup_Language) XML language.

The API for the web service should be REST-like.

* A correctly formatted GET request with a geographical coordinate should return the zone that a given point is within.
* A correctly formatted HTTP PUT request should add a new device (devices should be able to self-register, but the device implementation is not part of the project definition).

There should also be “methods” for administrating devices and ownership, creating, modifying and deleting zones, retrieving device and zone data, etc.

Relevant technologies (a subset of these will probably suffice):  
Java Servlets, JSP, XML, XML Schema, JDOM, XSLT, JavaScript, HTML, CSS, REST, JQuery, Java

* The map client must use AJAX to load preempted data and to request the current data at a user defined interval.