



## Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

### Μεταπτυχιακή Διατριβή

Τίτλος	<b>Δημιουργία διαδικτυακής εφαρμογής με χρήση Web Services σε χωρική βάση δεδομένων με τις τιμές αερίων ρύπων στους σταθμούς μέτρησης στην Ελλάδα.</b>
Title	<b>Development of a web application that adds functionality to a spatial database by using web services.</b>
Όνοματεπώνυμο Φοιτητή	<b>Παύλος Αγγελόπουλος</b>
Πατρώνυμο	<b>Ξενοφών</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ/ 10016</b>
Επιβλέποντες	<b>Ιωάννης Θεοδωρίδης, Καθηγητής</b> <b>Δρ. Ευθύμιος Αλέπης</b>

Ημερομηνία Παράδοσης **Νοέμβριος 2013**

### **Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

κ. Θεοδωρίδης Ιωάννης,  
Καθηγητής

Όνομα Επώνυμο  
Βαθμίδα

(υπογραφή)

κ. Αλέπης Ευθύμιος,  
Λέκτορας

Όνομα Επώνυμο  
Βαθμίδα

(υπογραφή)

κ. Βίρβου Μαρία,  
Καθηγήτρια

Όνομα Επώνυμο  
Βαθμίδα

## Περίληψη

Η παρούσα μεταπτυχιακή διατριβή αφορά το σχεδιασμό και την ανάπτυξη μιας διαδικτυακής εφαρμογής η οποία προσθέτει λειτουργικότητα σε μια χωρική βάση δεδομένων η οποία δημιουργήθηκε για αυτό το σκοπό. Όλα τα λογισμικά που χρησιμοποιήθηκαν για την εκπόνηση της διατριβής είναι Ελεύθερα Λογισμικά / Λογισμικά Ανοιχτού Κώδικα.

Πιο συγκεκριμένα, κατασκευάστηκε μια βάση δεδομένων στην οποία περιέχονται δεδομένα σχετικά με τις ημερήσιες μέσες τιμές μέτρησης 7 ατμοσφαιρικών ρύπων σε 30 διαφορετικούς σταθμούς μέτρησης στην Ελλάδα. Η βάση κατασκευάστηκε με το PostgreSQL, ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων που προσθέτει υποστήριξη για γεωγραφικά αντικείμενα (επέκταση PostGIS) σε αντικείμενο-σχεσιακή βάση δεδομένων.

Στη συνέχεια, χρησιμοποιήθηκε ο Geoserver, μέσω του οποίου οπτικοποιούνται και διαμοιράζονται τα δεδομένα μέσω των υποστηριζόμενων υπηρεσιών Web Map Service, Web Feature Service.

Η διεπαφή της εφαρμογής με το χρήστη, έχει κατασκευαστεί με HTML5 και JavaScript. Επίσης, με JavaScript γίνονται τα αιτήματα στο Geoserver και δημιουργείται ο δυναμικός χάρτης με τους σταθμούς.

Ο χρήστης, όταν επισκέπτεται την σελίδα βλέπει σε δυναμικό χάρτη τις θέσεις των σταθμών μέτρησης αερίων ρύπων σε όλη την Ελλάδα και έχει τη δυνατότητα κάνοντας κλικ επάνω σε κάποιον από αυτούς να δει τα δεδομένα που τον αφορούν. Μπορεί να επιλέξει ως υπόβαθρο ένα από τα Open Street Map, Google Map και Google Satellite.

Τέλος, του δίνεται η δυνατότητα να αναζητήσει τη τιμή οποιουδήποτε ρύπου σε όποιο σταθμό μετρήθηκε την ημέρα που τον ενδιαφέρει.

## Abstract

This thesis concerns the design and development of a web application that adds functionality to a spatial database which was created for this purpose. All software used for the development of this thesis is free software / open source software.

More specifically, database contains the daily average values of seven atmospheric pollutants measured by 30 different measurement stations in Greece. Database has been created with PostgreSQL, a database management system that adds support for geographic objects (PostGIS extension) to an object-relational database.

Geoserver was used for data sharing and map creation through Web Map Service and Web Feature Service

User Interface was created with HTML5 and JavaScript code. Also, JavaScript is used for Geoserver requests and dynamic map creation.

Any user visiting the web page sees on a map the positions of stations pollutants throughout Greece and by clicking on any of them can see his data. He can also choose as a base map Open Street Map, Google Map or Google Satellite.

Finally, user can also query the value of any pollutant, in every station measured, the day of interest.

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>1. Εισαγωγή</b>	<b>10</b>
<b>1.1 Αντικείμενο</b>	<b>10</b>
<b>1.2 Στόχοι</b>	<b>10</b>
<b>1.3 Ευχαριστίες</b>	<b>10</b>
<b>2. Γενικά</b>	<b>11</b>
<b>2.1 Σχετικά με την ατμοσφαιρική ρύπανση</b>	<b>11</b>
<b>2.2 Κύριοι αέριοι ρύποι</b>	<b>12</b>
<b>2.2.1 Οξειδία του Αζώτου (NO<sub>x</sub>)</b>	<b>12</b>
<b>2.2.2 Όζον (O<sub>3</sub>)</b>	<b>13</b>
<b>2.2.3 Διοξείδιο του θείου (SO<sub>2</sub>)</b>	<b>13</b>
<b>2.2.4 Βενζόλιο (C<sub>6</sub>H<sub>6</sub>)</b>	<b>14</b>
<b>2.2.5 Αιθάλη / Καπνός (Smoke)</b>	<b>14</b>
<b>2.3 Πηγές δεδομένων</b>	<b>14</b>
<b>3. Ανασκόπηση πεδίου</b>	<b>17</b>
<b>3.1 Αντίστοιχες εφαρμογές στην Ελλάδα</b>	<b>17</b>
<b>3.2 Αντίστοιχες εφαρμογές διεθνώς</b>	<b>24</b>
<b>4. Παρουσίαση και χρήση της εφαρμογής</b>	<b>30</b>

<b>5. Η αρχιτεκτονική της εφαρμογής .....</b>	<b>49</b>
<b>5.1 Λογισμικά – τεχνολογίες που χρησιμοποιήθηκαν .....</b>	<b>49</b>
<b>5.1.1 PostgreSQL .....</b>	<b>49</b>
<b>5.1.2 GeoServer .....</b>	<b>51</b>
<b>5.1.3 HTML5 – JavaScript – AJAX .....</b>	<b>52</b>
<b>5.2 Υλοποίηση της εφαρμογής .....</b>	<b>56</b>
<b>5.3 Διαγράμματα UML .....</b>	<b>76</b>
<b>6. Συμπεράσματα .....</b>	<b>78</b>
<b>7. Βιβλιογραφία – Πηγές .....</b>	<b>79</b>
<b>8. Παράρτημα .....</b>	<b>81</b>
<b>8.1 Κώδικας του αρχείου AerioidRypoiProject.html .....</b>	<b>81</b>
<b>8.2 Κώδικας του αρχείου AerioidRypoiProject.js .....</b>	<b>85</b>
<b>8.3 Κώδικας του αρχείου AerioidRypoiProject.css .....</b>	<b>99</b>
<b>8.4 Κώδικας του αρχείου objSort.js .....</b>	<b>102</b>
<b>8.5 Πίνακας αρχείων πρωτογενών δεδομένων .....</b>	<b>104</b>
<b>8.6 Εικόνα Π1. Τμήμα του αρχείου μετρήσεων .....</b>	<b>112</b>

**ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ**

Εικόνα 1. Νέφος τύπου Λονδίνου .....	11
Εικόνα 2. Νέφος τύπου Los Angeles .....	12
Εικόνα 3. Αρχική σελίδα της εφαρμογής AirDMS .....	17
Εικόνα 3α. Air-DMS Διάγραμμα ρύπου – ανέμου .....	19
Εικόνα 3β. Air-DMS Διάγραμμα ρύπου σε όλους τους σταθμούς .....	19
Εικόνα 3γ. Air-DMS Γράφημα δείκτη ρύπανσης .....	20
Εικόνα 3δ. Air-DMS Χάρτης με δείκτη ρύπανσης .....	20
Εικόνα 3ε. Air-DMS Χάρτης με δείκτη ρύπανσης και άνεμο .....	21
Εικόνα 4. EAP-3 Αρχική οθόνη εφαρμογής .....	22
Εικόνα 4α. EAP-3 Δεδομένα μετρήσεων σταθμού Ποντοκώμης .....	22
Εικόνα 4β. EAP-3 Δείκτες μετρήσεων σταθμού Ποντοκώμης .....	23
Εικόνα 4γ. EAP-3 Χαρακτηριστικά σταθμού Ποντοκώμης .....	23
Εικόνα 5. Αρχική οθόνη της εφαρμογής της E.E. ....	24
Εικόνα 5α. Δεδομένα του σταθμού Κοζάνης .....	25
Εικόνα 5β. Δεδομένα της προηγούμενης ημέρας του σταθμού Κοζάνης .....	25
Εικόνες 6α, 6β, 6γ. Η εφαρμογή State of the Air .....	26
Εικόνες 7α, 7β, 7γ, 7δ. Η εφαρμογή obsAIRve .....	27
Εικόνες 8α, 8β, 8γ. Η εφαρμογή Beijing Air Quality .....	28
Εικόνες 9α, 9β, 9γ, 9δ. Η εφαρμογή Air Quality China .....	28
Εικόνα 10α. Η αρχική οθόνη της εφαρμογής Visibility .....	29
Εικόνες 10β, 10γ. Η εφαρμογή Visibility εν λειτουργία .....	29
Εικόνα 11. Η αρχική σελίδα της εφαρμογής .....	30
Εικόνα 12. Οι επιλογές του χάρτη .....	31
Εικόνα 13. Χάρτης Google Map .....	32
Εικόνα 14. Χάρτης Google Satellite .....	33
Εικόνα 15. Υπηρεσία WFS για το σταθμό Ν.Σμύρνης .....	34
Εικόνα 16. Η φόρμα αναζήτησης .....	35
Εικόνες 17α, 17β. Αναζήτηση σταθμού μέτρησης .....	35
Εικόνα 18. Αναζήτηση ρύπου .....	36
Εικόνα 19. Φιλτράρισμα ρύπων με βάση τον επιλεγμένο σταθμό .....	36
Εικόνα 20. Φιλτράρισμα σταθμών με βάση τον επιλεγμένο ρύπο .....	37

Εικόνα 21α, 21β. Ελλιπής είσοδος από το χρήστη .....	37
Εικόνα 22. Εκτέλεση ερωτήματος χωρίς αποτέλεσμα .....	37
Εικόνα 23. Εκτέλεση ερωτήματος για το σταθμό 2 του Πειραιά .....	38
Εικόνα 24. Εκτέλεση ερωτήματος για το σταθμό Αγίας Σοφίας στη Θεσσαλονίκη .....	39
Εικόνα 25. Εκτέλεση ερωτήματος για το σταθμό 1 της Πάτρας .....	40
Εικόνα 26. Εκτέλεση ερωτήματος για το σταθμό του Βόλου .....	41
Εικόνα 27. Εκτέλεση ερωτήματος για το σταθμό της Λάρισας .....	42
Εικόνα 28. Εκτέλεση ερωτήματος για το σταθμό των Ιωαννίνων .....	43
Εικόνα 29. Εκτέλεση ερωτήματος για το σταθμό Ηρακλείου Κρήτης .....	44
Εικόνα 30. Εκτέλεση ερωτήματος για το σταθμό του Γεωπονικού Πανεπιστημίου .....	45
Εικόνα 31. Εκτέλεση ερωτήματος για το σταθμό του Αλιάρτου .....	46
Εικόνα 32. Εκτέλεση ερωτήματος για το σταθμό της Ελευσίνας .....	47
Εικόνα 33. Εκτέλεση ερωτήματος για το σταθμό Περιστερίου Αττικής .....	48
Εικόνα 34. Διαχρονικά η υποστήριξη HTML5 από τους κυριότερους browser .....	53
Εικόνα 35. Διαφορά στην αρχιτεκτονική κλασσικού – AJAX μοντέλου σε Web εφαρμογές .....	55
Εικόνα 36. Διαφορά στην ροή κλασσικού(synchronous) – AJAX (asynchronous) μοντέλου σε Web εφαρμογές .....	55
Εικόνα 37. Αρχιτεκτονική της εφαρμογής .....	56
Εικόνα 38. Η Βάση Δεδομένων της εφαρμογής .....	56
Εικόνα 39. Τα αρχεία στον GeoServer .....	56
Εικόνα 40. Διάγραμμα Οντοτήτων – Συσχετίσεων .....	59
Εικόνα 41. Σχεσιακό Μοντέλο .....	59
Εικόνα 42α. GeoServer – Δημιουργία νέου Workspace .....	60
Εικόνα 42β. GeoServer – Δημιουργία νέου Workspace .....	60
Εικόνα 43. GeoServer – Ρυθμίσεις παραμέτρων του νέου Workspace .....	61
Εικόνα 44. GeoServer – Δημιουργία νέου Data Store .....	62
Εικόνα 45. GeoServer – Επιλογή τύπου δεδομένων .....	62
Εικόνα 46. GeoServer – Ρυθμίσεις παραμέτρων σύνδεσης στη βάση δεδομένων .....	63
Εικόνα 47α. Add new layer .....	64
Εικόνα 47β. Add new layer .....	64
Εικόνα 48α. Ρύθμιση παραμέτρων της καρτέλας Data για το layer stathmoi .....	65
Εικόνα 48β. Ρύθμιση παραμέτρων της καρτέλας Data για το layer stathmoi .....	66
Εικόνα 49. Ρύθμιση παραμέτρων της καρτέλας Publishing για το layer stathmoi .....	67



<b>Εικόνα 50. Ρύθμιση παραμέτρων της καρτέλας Dimension για το layer stathmoi .....</b>	<b>68</b>
<b>Εικόνα 51. Ρύθμιση παραμέτρων της καρτέλας Tile Caching για το layer stathmoi .....</b>	<b>68</b>
<b>Εικόνα 52. Layer Preview από το GeoServer για το layer stathmoi .....</b>	<b>69</b>
<b>Εικόνα 53. Διάγραμμα περιπτώσεων χρήσης (Use Case Diagram) .....</b>	<b>76</b>
<b>Εικόνα 54. Διάγραμμα της διεπαφής .....</b>	<b>77</b>
<b>Εικόνα 55. Διάγραμμα της κλάσης stathmos .....</b>	<b>77</b>
<b>Εικόνα 56. Διάγραμμα της κλάσης gyros .....</b>	<b>77</b>
<b>Εικόνα 57. Διάγραμμα της κλάσης metrshh .....</b>	<b>77</b>

## 1. Εισαγωγή

### 1.1 Αντικείμενο

Η διατριβή αυτή εκπονήθηκε στα πλαίσια του Μεταπτυχιακού Προγράμματος Σπουδών «ΠΛΗΡΟΦΟΡΙΚΗ» του Πανεπιστημίου Πειραιώς και εμπίπτει στα γνωστικά αντικείμενα των μαθημάτων της Γεωπληροφορικής και της Περιβαλλοντικής Πληροφορικής. Αφορά στο σχεδιασμό και κατασκευή μιας διαδικτυακής εφαρμογής η οποία προσθέτει λειτουργικότητα σε μια διαδικτυακή χωρική Βάση Δεδομένων που υλοποιήθηκε και περιέχει δεδομένα σχετικά με τους σταθμούς μέτρησης των ατμοσφαιρικών ρύπων στην Ελλάδα και τους ρύπους που αυτοί μετρούν καθημερινά. Πιο συγκεκριμένα, τα δεδομένα αφορούν σε τριάντα σταθμούς μέτρησης, επτά αέριους ρύπους και τις μέσες ημερήσιες τιμές τους για την χρονική περίοδο 2000-2011.

Ο χρήστης της εφαρμογής με την επίσκεψη του στην ιστοσελίδα, βλέπει σε χάρτη τη θέση των σταθμών, επιλέγοντας ως υπόβαθρο Open Street Map (OSM), Google Map ή Google Satellite. Επίσης, μπορεί να δει τις πληροφορίες που περιέχονται στη Βάση Δεδομένων για όποιο σταθμό ενδιαφέρεται κάνοντας κλικ επάνω του. Οι παραπάνω υπηρεσίες υλοποιήθηκαν σύμφωνα με τα standard Web Map Service (WMS) και Web Feature Service (WFS).

Τέλος, μπορεί να αναζητήσει την μέση ημερήσια τιμή μέτρησης κάθε ρύπου, σε όποιο σταθμό αυτός μετρήθηκε, την ημερομηνία που επιθυμεί. Αυτό πραγματοποιείται μέσω αιτημάτων που γίνονται στο Geoserver με την τεχνική AJAX και επεξεργασία των αποκρίσεων με JavaScript.

### 1.2 Στόχοι

Στόχος της συγκεκριμένης εφαρμογής, αποτελεί η εύκολη πρόσβαση κάθε ενδιαφερόμενου σε δημόσια – ανοιχτή πληροφορία, ώστε να είναι δυνατή η αξιοποίηση της. Για τη δημιουργία της χρησιμοποιήθηκαν δεδομένα τα οποία διατίθενται ελεύθερα από το Υπουργείο Περιβάλλοντος κ Κλιματικής Αλλαγής αλλά βρίσκονται σε μορφή που τα καθιστά δύσκολα αξιοποιήσιμα. Επίσης, σε όλα τα στάδια υλοποίησης της εφαρμογής χρησιμοποιήθηκαν ελεύθερα λογισμικά.

### 1.3 Ευχαριστίες

Νιώθω την ανάγκη να ευχαριστήσω όλους τους καθηγητές και του συμφοιτητές μου στο Μεταπτυχιακό Πρόγραμμα Σπουδών για την συμπαράσταση και τη βοήθεια τους.

Ιδιαίτερη μνεία πρέπει να γίνει για τους καθηγητές κ.κ. Ευθύμιο Αλέπη και Ιωάννη Θεοδωρίδη αφού χωρίς την υποστήριξη και βοήθεια τους η διατριβή αυτή δεν θα είχε ποτέ ολοκληρωθεί. Η συμβολή τους υπήρξε καθοριστική σε όλα τα στάδια εκπόνησης της, ενώ ήταν πάντα διαθέσιμοι να προσφέρουν τη βοήθεια τους, όποτε ζητήθηκε.

Τέλος, είναι αυτονόητο πως για οποιαδήποτε παράλειψη ή λάθος της παρούσας διατριβής η ευθύνη βαρύνει αποκλειστικά εμένα.

## 2. Γενικά

### 2.1 Σχετικά με την ατμοσφαιρική ρύπανση

Ατμοσφαιρική ρύπανση καλείται η παρουσία στην ατμόσφαιρα κάθε είδους ουσιών, σε συγκέντρωση ή διάρκεια τέτοια, που μπορούν να προκαλέσουν αρνητικές επιπτώσεις στην υγεία των ζωντανών οργανισμών και στα οικοσυστήματα, ή γενικότερα να καταστήσουν το περιβάλλον ακατάλληλο για τις επιθυμητές χρήσεις του.

Οι ρύποι που εκλύονται στην ατμόσφαιρα από τις ανθρώπινες δραστηριότητες, ιδίως μετά τη Βιομηχανική Επανάσταση, έρχονται να προστεθούν σε αυτούς που εκπέμπονται από φυσικές πηγές δημιουργώντας τις προϋποθέσεις για αλλοίωση της σύστασης της. Ιδιαίτερα τις τελευταίες δεκαετίες, η ανθρωπογενής επίδραση αυξάνεται με εκθετικούς ρυθμούς, επιβαρύνοντας περαιτέρω την ποιότητα του ατμοσφαιρικού αέρα. Παρά το γεγονός ότι η βασική σύσταση της ατμόσφαιρας σε άζωτο και οξυγόνο - το 99% του ατμοσφαιρικού αέρα - δεν μεταβάλλεται, η αύξηση συγκεντρώσεων των αερίων με μικρή συμμετοχή στην σύσταση της υποβαθμίζει αισθητά την ποιότητα ζωής στη γη. Άλλωστε, η ανθρώπινη υγεία και ευεξία, η ευστάθεια του κλίματος, το φυσικό και ανθρωπογενές περιβάλλον, έχει αποδειχθεί πως είναι πολύ ευαίσθητα σε αλλαγές αερίων των οποίων η συγκέντρωση στην ατμόσφαιρα δεν υπερβαίνει το 0.04%

Κάτω από συγκεκριμένες συνθήκες, η ατμοσφαιρική ρύπανση μπορεί να φτάσει σε μη ανεκτά επίπεδα διαβίωσης. Σε αυτήν την περίπτωση λέγεται ότι υπάρχει «Νέφος», το οποίο παρουσιάζεται με δύο μορφές:

Νέφος καπνομίχλης, σχηματίζεται όταν υπάρχει υψηλή συγκέντρωση των ρύπων μονοξειδίου του άνθρακα, διοξειδίου του θείου και αιωρούμενα σωματίδια, σε συνδυασμό με σχετικά χαμηλή θερμοκρασία και μεγάλη σχετική υγρασία. Το φαινόμενο είναι εντονότερο κατά τους χειμερινούς μήνες και κυρίως τις πρωινές ώρες, κατά τις οποίες επικρατούν οι παραπάνω καιρικές συνθήκες. Στη βιβλιογραφία συναντάται και ως ατμοσφαιρική ρύπανση «τύπου Λονδίνου», καθώς πρώτη φορά παρουσιάστηκε στην πρωτεύουσα της Αγγλίας, με σοβαρότερο επεισόδιο το 1952, οπότε πέθαναν εκατοντάδες άνθρωποι. Προκαλείται σχεδόν αποκλειστικά από την κατανάλωση καυσίμων υλών, ειδικά κάρβουνου, σε στάσιμες πηγές όπως είναι οι σταθμοί παραγωγής ενέργειας και τα χυτήρια.



Εικόνα 1. Νέφος τύπου Λονδίνου

Αντίθετα, το φωτοχημικό νέφος εμφανίζεται στις υψηλές θερμοκρασίες με μεγάλη ηλιοφάνεια σε ένταση και διάρκεια και μικρή σχετική υγρασία όταν παράλληλα υπάρχουν υψηλή συγκέντρωση οξειδίων του αζώτου, υδρογονανθράκων, και δευτερογενών παραγώγων τους. Ονομάζεται και ρύπανση «τύπου Λος Άντζελες», καθώς εκεί εμφανίστηκε πρώτη φορά το 1943.

Αν και υπάρχει συμβολή από στάσιμες πηγές, η φωτοχημική αιθαλομίχλη συνδέεται κυρίως με εκπομπές από τροχοφόρα.



Εικόνα 2. Νέφος τύπου Los Angeles

## 2.2 Κύριοι αέριοι ρύποι

Παρουσιάζονται συνοπτικά οι κύριοι αέριοι ρύποι με τα χαρακτηριστικά τους, τις ανθρωπογενείς πηγές εκπομπής τους και τις επιδράσεις τους στην υγεία και το περιβάλλον.

### 2.2.1 Οξειδία του Αζώτου (NO<sub>x</sub>)

Από τα επτά γνωστά οξειδία του αζώτου (NO, NO<sub>2</sub>, NO<sub>3</sub>, N<sub>2</sub>O, N<sub>2</sub>O<sub>3</sub>, N<sub>2</sub>O<sub>4</sub>, και N<sub>2</sub>O<sub>5</sub>) μόνο δύο είναι εκείνα που κατέχουν σπουδαίο ρόλο στα προβλήματα της ατμοσφαιρικής ρύπανσης: το μονοξείδιο και το διοξείδιο του αζώτου. Για το λόγο αυτό έχει επικρατήσει, ο όρος «οξειδία του αζώτου (NO<sub>x</sub>)», να χρησιμοποιείται για να δηλώσει μόνο τα δύο αυτά οξειδία. Το NO είναι αέριο, άχρωμο και άοσμο ενώ το NO<sub>2</sub> είναι αέριο με καφέ χρώμα, ισχυρό οξειδωτικό διαλυτό στο νερό, με οξεία και ερεθιστική οσμή.

Τα οξειδία του αζώτου παράγονται από τη χρήση καυσίμων, κυρίως σε αυτοκίνητα αλλά και σε βιομηχανικούς καυστήρες και σε σταθμούς παραγωγής ηλεκτρικής ενέργειας. Η παραγωγή των NO<sub>x</sub> γίνεται είτε από την οξειδωση του ατμοσφαιρικού αζώτου κατά τη διάρκεια της καύσης, είτε κατά τη οξειδωση των αζωτούχων ενώσεων που περιέχονται στα καύσιμα. Το μεγαλύτερο μέρος από τις εκπομπές NO<sub>x</sub> γίνεται με τη μορφή του NO, το οποίο όμως οξειδώνεται γρήγορα προς NO<sub>2</sub>.

Το διοξείδιο του αζώτου, υπό την επίδραση της ηλιακής ακτινοβολίας, αντιδρά με υδρογονάνθρακες και οδηγεί στην παραγωγή όζοντος και φωτοχημικού νέφους ενώ έχει συμβολή στην όξινη βροχή. Το διοξείδιο του αζώτου, σε μεγάλες συγκεντρώσεις, είναι ερεθιστικό για τον ανθρώπινο οργανισμό. Πιο συγκεκριμένα, στα παιδιά προκαλεί αναπνευστικές ασθένειες ενώ στους ασθματικούς έντονα προβλήματα αναπνοής. Σε περιπτώσεις ψηλών συγκεντρώσεων, τα άτομα που επηρεάζονται προτρέπονται να περιορίζουν τις μετακινήσεις τους σε εξωτερικούς χώρους στον ελάχιστο δυνατό βαθμό. Λόγω της μετατροπής του σε οξύ προκαλεί διάβρωση στα μέταλλα και τα υλικά, ενώ είναι τοξικό και για τη βλάστηση. Τέλος, προκαλεί μείωση της ορατότητας της ατμόσφαιρας καθώς μέσα από φωτοχημικές αντιδράσεις δημιουργεί ένα καστανοκίτρινο νέφος στον ουρανό των πόλεων.

### 2.2.2 Όζον (O<sub>3</sub>)

Το όζον (O<sub>3</sub>) δεν εκπέμπεται κατευθείαν στην ατμόσφαιρα αλλά παράγεται μετά από μια σειρά αντιδράσεων. Ο συνδυασμός των οξειδίων του αζώτου, των διαφόρων υδρογονανθράκων και του ηλιακού φωτός είναι δυνατό να εκκινήσει μια σειρά πολύπλοκων χημικών αντιδράσεων που σαν προϊόντα έχουν μια σειρά από δευτερογενείς ρύπους, ο κυριότερος από τους οποίους είναι το όζον. Αποτελεί το βασικό ρύπο της φωτοχημικής ρύπανσης των πόλεων και γι' αυτό χρησιμοποιείται σαν δείκτης της. Στην ανώτερη ατμόσφαιρα (στρατόσφαιρα) το όζον έχει ευεργετικό ρόλο γιατί απορροφά τις υπεριώδεις ακτινοβολίες (UV), προστατεύοντας έτσι τη γη από τις βλαβερές ακτίνες του ήλιου.

Το O<sub>3</sub> είναι αέριο άχρωμο, βαρύτερο του αέρα με δριμεία οσμή. Επίσης, είναι ισχυρότατο οξειδωτικό. Διαλύεται δύσκολα στο νερό γι' αυτό και μπορεί να διεισδύσει μέχρι τους πνεύμονες με όλες τις αρνητικές συνέπειες για την υγεία των ανθρώπων. Οι εποχικές διακυμάνσεις στα επίπεδα του έχουν άμεση σχέση με τις αντιδράσεις σχηματισμού του και ιδιαίτερα με την απαιτούμενη ηλιακή ακτινοβολία. Οι μέγιστες συγκεντρώσεις παρουσιάζονται, την θερμή περίοδο του έτους όπου τόσο η ένταση της ηλιακής ακτινοβολίας όσο και η διάρκεια της ημέρας είναι μεγαλύτερες.

Το όζον σχηματίζεται στην κατώτερη ατμόσφαιρα (τροπόσφαιρα) ως αποτέλεσμα χημικών αντιδράσεων μεταξύ του οξυγόνου, πτητικών οργανικών ενώσεων (VOCs), και οξειδίων του αζώτου (NO<sub>x</sub>) με τη βοήθεια της ηλιακής ακτινοβολίας. Πηγές εκπομπής αυτών των πρόδρομων ουσιών του όζοντος (VOCs και NO<sub>x</sub>) είναι τα οχήματα, τα χημικά εργοστάσια, τα χημικά διαλυτικά και τα βενζινάδικα.

Το O<sub>3</sub> είναι εξαιρετικά τοξικό και μπορεί να προκαλέσει από αίσθημα ξηρότητας στο λαιμό και ερεθισμό στους οφθαλμούς έως ζάλη και εμετούς. Ακόμα, σε υψηλές συγκεντρώσεις στον ατμοσφαιρικό αέρα μπορεί να ερεθίσει το αναπνευστικό σύστημα, προκαλώντας βήχα και πόνο στο στήθος, φλεγμονή στους πνεύμονες και πιθανή επιδεικτικότητα σε μολύνσεις του αναπνευστικού. Τέλος, το όζον θεωρείται πως έχει τις δυσμενέστερες επιδράσεις στα φυτά, καθώς η παρουσία του έχει αρνητικές επιπτώσεις στις αγροτικές καλλιέργειες, δασική και άλλη βλάστηση.

### 2.2.3 Διοξείδιο του θείου (SO<sub>2</sub>)

Το διοξείδιο του θείου είναι αέριο, άχρωμο, άοσμο σε χαμηλές συγκεντρώσεις αλλά με έντονη ερεθιστική οσμή σε πολύ υψηλές συγκεντρώσεις.

Το 80% των ανθρωπογενών εκπομπών διοξειδίου του θείου προέρχεται από την καύση ορυκτών καυσίμων από σταθερές πηγές (βιομηχανία, θέρμανση). Από αυτό, το 85% αποτελεί εκπομπές από σταθμούς παραγωγής ηλεκτρικής ενέργειας. Σημαντικές πηγές αποτελούν επίσης τα διυλιστήρια πετρελαίου, τα εργοστάσια επεξεργασίας χαλκού, η χημική βιομηχανία και η χαρτοβιομηχανία. Ακόμα, άλλες πηγές εκπομπής SO<sub>2</sub> αλλά σε μικρότερο βαθμό, αποτελούν οι κεντρικές θερμάνσεις κατοικιών και τα πετρελαιοκίνητα αυτοκίνητα που χρησιμοποιούν καύσιμο με υψηλή περιεκτικότητα σε θείο.

Μακροχρόνια έκθεση στο διοξείδιο του θείου μπορεί να προκαλέσει αναπνευστικά προβλήματα, να τροποποιήσει τον αμυντικό μηχανισμό των πνευμόνων και να επιδεινώσει τυχόν υπάρχουσες καρδιαγγειακές παθήσεις. Άτομα με καρδιαγγειακές, χρόνιες πνευμονολογικές παθήσεις καθώς και μικρά παιδιά και ηλικιωμένοι είναι ιδιαίτερα ευπαθή σε τέτοιες συνθήκες. Υψηλές συγκεντρώσεις SO<sub>2</sub> στην ατμόσφαιρα συμβάλλουν επίσης στη μείωση της ορατότητας, στην αύξηση της οξύτητας των λιμνών και των ποταμών και προκαλούν αλλοιώσεις στη βλάστηση. Τέλος, έχει αρνητική επίδραση στα δομικά υλικά όπως το μάρμαρο και τα μέταλλα.

### 2.2.4 Βενζόλιο (C<sub>6</sub>H<sub>6</sub>)

Το βενζόλιο είναι ένας αρωματικός υδρογονάνθρακας που είναι φυσικό συστατικό του αργού πετρελαίου και ένα από τα πιο θεμελιώδη πετροχημικά. Πρόκειται για πτητική οργανική ένωση (VOC) η οποία αποτελεί ένα από τα δευτερεύοντα συστατικά της βενζίνης. Στις συνηθισμένες συνθήκες (θερμοκρασία 25°C και πίεση 1 atm), το βενζόλιο είναι ένα υγρό άχρωμο, με ιδιάζουσα γλυκιά μυρωδιά, πολύ εύφλεκτο και πτητικό. Διαλύεται ελάχιστα στο νερό και καίγεται με ζωηρή φλόγα και καπνό. Αναμιγνύεται εύκολα σε οποιαδήποτε αναλογία με τον αιθέρα και το οινόπνευμα. Παρόλο που εμφανίζει μηδενική διπολική ροπή, διαλύει πολλές οργανικές ενώσεις, όπως λίπη, λάδια, καουτσούκ, αλλά και ανόργανες ουσίες, όπως θείο, φωσφόρο και ιώδιο. Οι ατμοί με τον ατμοσφαιρικό αέρα αποτελούν εκρηκτικό μείγμα.

Πηγές εκπομπής βενζολίου στον ατμοσφαιρικό αέρα, είναι τα πρατήρια βενζίνης και τα αυτοκίνητα διανομής της, καθώς επίσης και όλες οι μηχανές που χρησιμοποιούν βενζίνη σαν καύσιμο.

Το βενζόλιο είναι ιδιαίτερα τοξική χημική ένωση η οποία θεωρείται καρκινογόνος για τον άνθρωπο. Όταν εισπνέεται σε μεγάλες ποσότητες μπορεί να προκαλέσει ζάλη, ταχυκαρδία, πονοκεφάλους, σύγχυση, αναισθησία, ακόμα και το θάνατο. Επίσης, όταν βρίσκεται σε μεγάλες συγκεντρώσεις στα τρόφιμα μπορεί να προκαλέσει ερεθισμό, τάση για εμετό και σπασμούς. Μακροχρόνια έκθεση έχει σημαντικές επιπτώσεις στην υγεία του ανθρώπου και κυρίως στο αίμα. Καταστρέφει το μυελό των οστών και μπορεί να προκαλέσει την εμφάνιση αναιμίας. Επίσης, μπορεί να προκαλέσει υπερβολική αιμορραγία και να μειώσει την ικανότητα του ανοσοποιητικού συστήματος αυξάνοντας τις πιθανότητες μόλυνσης.

### 2.2.5 Αιθάλη / Καπνός (Smoke)

Σαν καπνός αναφέρονται τα μικρά σωματίδια τα οποία προέρχονται από ατελείς καύσεις και αποτελούνται κυρίως από άνθρακα και άλλα καύσιμα υλικά. Το μέγεθός τους είναι σχετικά μικρό, μέχρι 1 μm, αλλά είναι ορατά λόγω της μεγάλης ποσότητας τους. Σε μεγάλες συγκεντρώσεις στον ατμοσφαιρικό αέρα, προκαλεί δυσκολία στην αναπνοή και μείωση της ορατότητας

## 2.3 Πηγές δεδομένων

Όλα τα δεδομένα που χρησιμοποιήθηκαν για τη δημιουργία της εφαρμογής προέρχονται από την ιστοσελίδα του Υπουργείου Περιβάλλοντος και Κλιματικής Αλλαγής (<http://www.ypeka.gr/Default.aspx?tabid=495&language=el-GR>). Ο επισκέπτης μπορεί σε αυτή τη διαδικτυακή τοποθεσία να κατεβάσει τα δεδομένα μετρήσεων για κάθε έτος που τον ενδιαφέρει.

Για την εφαρμογή αξιοποιήθηκαν τα δεδομένα της περιόδου 2000 – 2011. Για κάθε χρόνο, διατίθεται ένα zip αρχείο το οποίο περιέχει φακέλους με τον καθένα από αυτούς να αντιστοιχεί σε έναν από τους σταθμούς μέτρησης. Κάθε φάκελος περιέχει τόσα αρχεία όσοι και οι ρύποι που μετρήθηκαν από το συγκεκριμένο σταθμό το συγκεκριμένο έτος. Τα αρχεία είναι σε μορφή \*.dat, και η ονομασία τους είναι σύνθετη, αποτελούμενη από τρία μέρη και δίνεται σύμφωνα με τον παρακάτω κανόνα:

Το πρώτο συνθετικό αφορά το μετρούμενο ρύπο και είναι ένα εκ των :

NO# για το μονοξειδίο του Αζώτου

NO2 για το διοξείδιο του Αζώτου

NOX για τα οξειδία του Αζώτου (οι μετρήσεις αφορούν τα έτη 2000 έως 2002)

O3# για το Όζον

SO2 για το Διοξείδιο του Θείου

Benz για το Βενζόλιο (οι μετρήσεις αφορούν τα έτη 2004 έως 2011)

Δημιουργία διαδικτυακής εφαρμογής με χρήση Web Services

Smoke για την Αιθάλη (οι μετρήσεις αφορούν τα έτη 2000 έως 2005 εκτός του 2003).

Το δεύτερο συνθετικό του ονόματος αφορά στον κωδικό του σταθμού μέτρησης και είναι ένα από τα παρακάτω(σε παρένθεση τα έτη κατά τα οποία λειτουργούσε ο σταθμός στην περίοδο 2000-2011) :

AGS για το σταθμό πλατείας Αγίας Σοφίας στη Θεσσαλονίκη (2001-2011)

APT για το σταθμό του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης (2001-2011)

KOD για το σταθμό στο Κορδελιό Θεσσαλονίκης (2001-2011)

PAO για το σταθμό στο Πανόραμα Θεσσαλονίκης (2001-2011)

KAL για το σταθμό στην Καλαμαριά Θεσσαλονίκης (2001-2010)

SIN για το σταθμό στην Σίνδο Θεσσαλονίκης (2001-2011)

NEO για το σταθμό στην Νεοχωρούδα Θεσσαλονίκης (2001-2011 εκτός του 2010)

PAI για τον πρώτο από τους δύο σταθμούς στην Πάτρα (2001-2011 εκτός 2005,2006)

PAII για τον δεύτερο σταθμό της Πάτρας (2001-2011 εκτός 2004,2005,2006)

VOL για τον σταθμό μέτρησης αερίων ρύπων στο Βόλο (2001-2008)

LAR για το σταθμό της Λάρισας (2001,2002,2007,2008,2009,2011)

IRA για το σταθμό Ηρακλείου Κρήτης (2001,2002,2003)

IOA για το σταθμό Ιωαννίνων (2008-2011)

ATH για το σταθμό της οδού Αθηνάς στο κέντρο της Αθήνας (2000-2011)

GEO για το σταθμό στο Γεωπονικό Πανεπιστήμιο της Αθήνας (2000-2011)

LIO για το σταθμό στα Λιόσια Αττικής (2000-2011)

LYK για το σταθμό στη Λυκόβρυση Αττικής (2000-2011)

MAR για το σταθμό στο Μαρούσι Αττικής (2000-2011)

SMY για το σταθμό στη Νέα Σμύρνη Αττικής (2000-2011)

PAT για το σταθμό της οδού Πατησίων στο κέντρο της Αθήνας (2000-2011)

PIR για τον ένα από τους δύο σταθμούς στον Πειραιά (2000-2011)

PER για το σταθμό στο Περιστερί Αττικής (2000-2011)

AGP για το σταθμό στην Αγία Παρασκευή Αττικής (2000-2011)

ELE για το σταθμό στην Ελευσίνα Αττικής (2000-2011)

THR για το σταθμό στην περιοχή Θρακομακεδόνες Αττικής (2000-2011)

PAN για το σταθμό στην περιοχή Ζωγράφου Αττικής (2000-2008)

GOU για το σταθμό στην περιοχή Γουδή Αττικής (2000-2010)

BIO για το δεύτερο σταθμό του Πειραιά (2000-2008)

GAL για το σταθμό στο Γαλάτσι Αττικής (2000-2005)

ALI για το σταθμό στον Αλιάρτο (2001-2011 εκτός 2003,2004)

Το τρίτο συνθετικό αναφέρεται στο έτος κατά το οποίο έγιναν οι μετρήσεις. Συνολικά για την περίοδο 2000-2011 υπάρχουν 1510 αρχεία μετρήσεων από τους παραπάνω 30 σταθμούς για τους 7 ρύπους. Η τυπική μορφή των αρχείων φαίνεται στην Εικόνα Π1 του Παραρτήματος. Κάθε γραμμή περιλαμβάνει την ημερομηνία και 24 ωριαίες μετρήσεις. Για τις ώρες που δεν έγινε μέτρηση, υπάρχει η τιμή -9999. Στην παρούσα εργασία, έχουν υπολογιστεί οι ημερήσιοι μέσοι όροι από τις ωριαίες μετρήσεις. Η επεξεργασία των αρχείων μετρήσεων έγινε με κώδικα java και θα αναλυθεί παρακάτω, στο αντίστοιχο κεφάλαιο.

Επίσης, από την ιστοσελίδα του ΥΠΕΚΑ προέκυψαν και τα δεδομένα που αφορούν στους σταθμούς μέτρησης, δηλαδή η θέση τους και τα χαρακτηριστικά τους, όπως παρατίθενται παρακάτω:

ΟΝΟΜΑ	ΚΩΔ.	ΓΕΩΓΡΑΦΙΚΟ ΜΗΚΟΣ	ΓΕΩΓΡΑΦΙΚΟ ΠΛΑΤΟΣ	ΥΨΟΜΕΤΡΟ	ΧΑΡΑΚΤΗΡΙΣΜΟΣ
ΠΛ.ΑΓΙΑΣ_ΣΟΦΙΑΣ	AGS	22,945289	40,633772	27	Αστικός-Κυκλοφορίας
Α.Π.Θ.	APT	22,956703	40,633572	55	Αστικός-Υποβάθρου
ΚΟΡΔΕΛΙΟ	KOD	22,893439	40,673547	30	Αστικός-Βιομηχανικός
ΠΑΝΟΡΑΜΑ	PAO	23,031681	40,588931	363	Περιαστικός-Υποβάθρου
ΚΑΛΑΜΑΡΙΑ	KAL	22,959303	40,578917	60	Περιαστικός-Υποβάθρου
ΣΙΝΔΟΣ	SIN	22,802367	40,657842	14	Αστικός-Βιομηχανικός
ΝΕΟΧΩΡΟΥΔΑ	NEO	22,876231	40,739842	229	Περιαστικός-Υποβάθρου
ΠΑΤΡΑ-1	PAI	21,738431	38,253097	16	Αστικός-Κυκλοφορίας
ΠΑΤΡΑ-2	PAII	21,735897	38,245975	19	Αστικός-Κυκλοφορίας
ΒΟΛΟΣ	VOL	22,943056	39,366389	31	Αστικός-Υποβάθρου
ΛΑΡΙΣΑ	LAR	22,453333	39,667500	15	Αστικός-Κυκλοφορίας
ΗΡΑΚΛΕΙΟ	IRA	25,080000	35,332500	10	Αστικός-Κυκλοφορίας
ΙΩΑΝΝΙΝΑ	IOA	20,850000	39,620000	485	Αστικός-Υποβάθρου
ΑΘΗΝΑΣ	ATH	23,726842	37,978203	100	Αστικός-Κυκλοφορίας
ΑΡΙΣΤΟΤΕΛΟΥΣ	ARI	23,727628	37,988028	95	Αστικός-Κυκλοφορίας
ΓΕΩΠΟΝΙΚΗ	GEO	23,706789	37,983625	40	Περιαστικός-Βιομηχανικός
ΛΙΟΣΙΑ	LIO	23,697842	38,076814	165	Περιαστικός-Υποβάθρου
ΛΥΚΟΒΡΥΣΗ	LYK	23,776886	38,069633	210	Περιαστικός
ΜΑΡΟΥΣΙ	MAR	23,787358	38,030839	170	Αστικός-Κυκλοφορίας
ΝΕΑ_ΣΜΥΡΝΗ	SMY	23,713008	37,931994	50	Αστικός-Υποβάθρου
ΠΑΤΗΣΙΩΝ	PAT	23,733047	37,999458	105	Αστικός-Κυκλοφορίας
ΠΕΙΡΑΙΑΣ_I	PIR	23,647511	37,943286	20	Αστικός-Κυκλοφορίας
ΠΕΡΙΣΤΕΡΙ	PER	23,688356	38,020808	80	Αστικός-Υποβάθρου
ΑΓ.ΠΑΡΑΣΚΕΥΗ	AGP	23,819417	37,995108	290	Περιαστικός-Υποβάθρου
ΕΛΕΥΣΙΝΑ	ELE	23,538447	38,051350	20	Περιαστικός-Βιομηχανικός
ΘΡΑΚΟΜΑΚΕΔΟΝΕΣ	THR	23,758183	38,143522	550	Περιαστικός-Υποβάθρου
ΚΟΡΩΠΙ	KOR	23,879022	37,901306	140	Περιαστικός-Υποβάθρου
ΖΩΓΡΑΦΟΥ	PAN	23,786728	37,969603	245	Περιαστικός-Υποβάθρου
ΓΟΥΔΗ	GOU	23,767431	37,984144	155	Αστικός-Κυκλοφορίας
ΠΕΙΡΑΙΑΣ_II	BIO	23,652686	37,941969	25	Αστικός-Υποβάθρου
ΓΑΛΑΤΣΙ	GAL	23,748206	38,020286	154	Περιαστικού-Υποβάθρου
ΟΙΝΟΦΥΤΑ	OIN	23,638914	38,306219	100	Περιαστικός-Βιομηχανικός
ΑΛΙΑΡΤΟΣ	ALI	23,110267	38,375247	110	Υποβάθρου

#### Χαρακτηριστικά των σταθμών μέτρησης

Να σημειωθεί πως η γεωγραφική θέση των σταθμών (λ, φ) διατίθεται στο γεωκεντρικό Σύστημα Αναφοράς WGS84 και είναι σε μοίρες σε μορφή λεπτών και δευτερολέπτων. Έτσι, έγινε επεξεργασία τους ώστε να μετατραπούν σε μοίρες με δεκαδική μορφή σύμφωνα με την αναλογία  $1^\circ = 60' = 3600''$



### 3. Ανασκόπηση πεδίου

#### 3.1 Αντίστοιχες εφαρμογές στην Ελλάδα

Μετά από εκτεταμένη έρευνα στο διαδίκτυο δεν βρέθηκε καμία εφαρμογή αντίστοιχης λειτουργίας. Για τον Ελλαδικό χώρο, υπάρχει μόνο μία εφαρμογή που λέγεται Air-DMS (<http://www.air-quality.gr/>), η οποία όμως δεν παρέχει τις τιμές μέτρησης των αερίων ρύπων, αλλά γραφήματα σχετικά με αυτές. Η Εικόνα 3 αναφέρεται σε αυτήν.

**Αρχική** | Ατμοσφαιρική Ρύπανση | Χρήσιμα | Συνδέσεις | Επικοινωνία

**AirDMS**  
Data Management System

Διαχείριση AirDMS
Δυνατότητες γενικά
Γραφικές αναφορές
Δυναμικές παρουσιάσεις
Σχετικά

#### AirDMS Data Management System

Το AirDMS είναι ένα ολοκληρωμένο σύστημα διαχείρισης περιβαλλοντικών δεδομένων. Δίνεται η δυνατότητα πρόσβασης στα δεδομένα μέσω της ιστοσελίδας σας, ενώ διασφαλίζεται διαβαθμισμένη πρόσβαση, ανάλογα με τα διακαιώματα που καθορίζονται για τους χρήστες. Η παρουσίαση των δεδομένων γίνεται με απλό και κατανοητό τρόπο, μέσω γραφικών παραστάσεων και περιβαλλοντικών δεικτών. Η διαχείριση του site σας γίνεται με **πρόγραμμα από τον υπολογιστή σας**. Μέσω του προγράμματος διαχείρισης γίνεται των αποστολή δεδομένων στο internet, ενώ οι τεχνικοί των σταθμών μπορούν να εισάγουν αναφορές κατάστασης των σταθμών. Οι αναφορές και τα δεδομένα σταθμού (τρέχοντα δεδομένα, ιστορικά δεδομένα, δυναμικές γραφικές παραστάσεις και αναφορές κατάστασης σταθμών) μπορούν να είναι προσβάσιμες στο κοινό ή για το καθένα από αυτά να δωθεί πρόσβαση μόνο σε εξουσιοδοτημένους χρήστες.

© 2009 [www.air-quality.gr](http://www.air-quality.gr)

**Εικόνα 3. Αρχική σελίδα της εφαρμογής AirDMS**

Όσον αφορά τις δυνατότητες του προγράμματος, τα παρακάτω παρατίθενται όπως αναφέρονται στο υπομενού Δυνατότητες γενικά της αρχικής σελίδας της εφαρμογής :

*«Ο χρήστης έχει τη δυνατότητα πρόσβασης στους σταθμούς μέσω της ιστοσελίδας ενώ μπορεί να πάρει αναφορές σε πίνακες ή γραφικές παραστάσεις*

- *Βοηθητικά εργαλεία Service:*  
Πρόσβαση στους σταθμούς μέσω της ιστοσελίδας (Remote Control).  
Εμφάνιση διαγνωστικών παραμέτρων των αναλυτών.  
Δυναμική αναπαράσταση σφαλμάτων αναλυτών.
- *Διαθέσιμες αναφορές:*

*Αναφορές σταθμού – Ημερήσιες, Εβδομαδιαίες, Μηνιαίες, Περιοδικές.*

*Αναφορές ενός ρύπου σε όλους τους διαθέσιμους σταθμούς.*

*Αναφορές δεικτών - ρύπανσης, δυσφορίας, αίσθηση ψύχους κ.ά.*

*Αναφορές βαθμονομήσεων.*

*Αναφορές σφαλμάτων.*

- *Δυναμικές αναφορές:*

*Τρέχουσες μετρήσεις ενός ή και περισσότερων σταθμών.*

*Τελευταία λήψη δεδομένων για κάθε σταθμό.*

*Δυναμικός χάρτης με τις θέσεις των σταθμών και τις τιμές του δείκτη ρύπανσης.*

*Δυναμικός χάρτης με τις θέσεις των σταθμών και τα διανύσματα του ανέμου.*

*Δυναμική αναπαράσταση περιβαλλοντικών δεικτών.*

*Δυναμική αναπαράσταση μετεωρολογικών παραμέτρων.*

- *Ανάλυση:*

*Ροδοδιάγραμμα ανέμου.*

*Ροδοδιάγραμμα ρύπανσης.*

*Ημερήσιο διάγραμμα διανυσμάτων ανέμου μαζί με το ρύπο.*

*Ομαδικό γράφημα ρύπου σε όλους τους σταθμούς.*

*Γράφημα ρύπου με το Εθνικό όριο.*

*Συγκριτικό γράφημα παραμέτρων σε δύο σταθμούς.*

*Εξαγωγή δεδομένων σε xml,xls,pdf format. »*

Η συγκεκριμένη εφαρμογή, σταμάτησε να ενημερώνεται με δεδομένα από την 30/11/2009 και όσα δεδομένα αναφέρονται σε μεταγενέστερη ημερομηνία έχουν τυχαίες τιμές. Αυτό προκύπτει αφού μετά από ανανεώσεις της ιστοσελίδας προκύπτουν εντελώς διαφορετικά αποτελέσματα. Επίσης, χρησιμοποιεί μετρήσεις από 8 διαφορετικούς σταθμούς, σε πόλεις που δεν περιλαμβάνονται στον κατάλογο του ΥΠΕΚΑ ως τόποι εγκατάστασης σταθμών μέτρησης αερίων ρύπων, ενώ δεν υπάρχει πουθενά αναφορά για την προέλευση των δεδομένων ή τις μεθόδους που ακολουθήθηκαν για τη συλλογή τους.

Κατά τη λειτουργία της, μόνο πέντε γραφήματα είναι δυνατόν να κατασκευαστούν, τουλάχιστον μέσω πρόσβασης από το διαδίκτυο. Η συγκεκριμένη εφαρμογή δεν είναι καθαρά διαδικτυακή, καθώς όπως αναφέρεται απαιτείται να εγκατασταθεί κάποιο ειδικό λογισμικό στον υπολογιστή – πελάτη. Έτσι, τα μόνα γραφήματα που μπορούν να δουν οι απλοί (μη εξουσιοδοτημένοι) χρήστες είναι τα εξής:

Διάγραμμα ρύπου – ανέμου

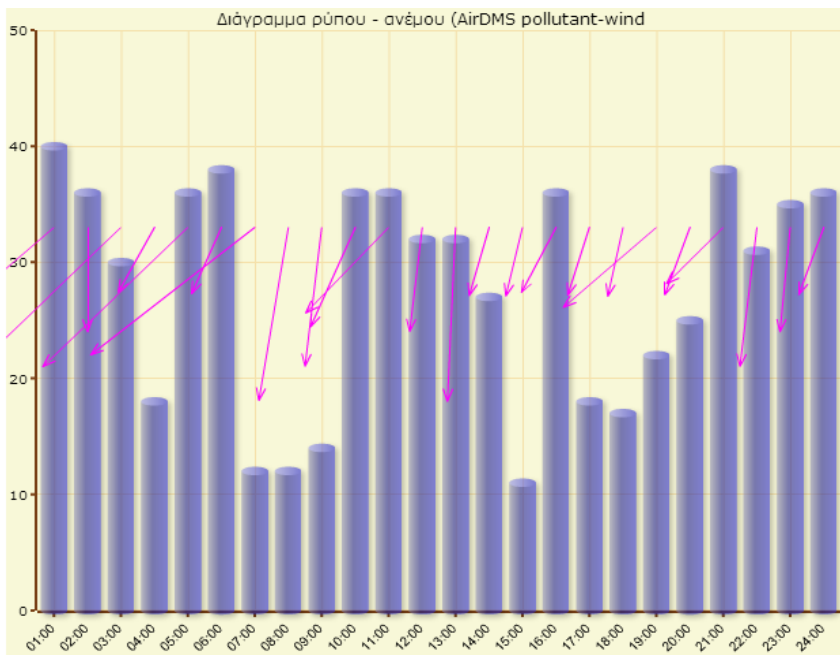
Διάγραμμα ενός ρύπου σε όλους τους σταθμούς

Γραφική απεικόνιση του δείκτη ρύπανσης

Χάρτη με απεικόνιση του δείκτη ρύπανσης

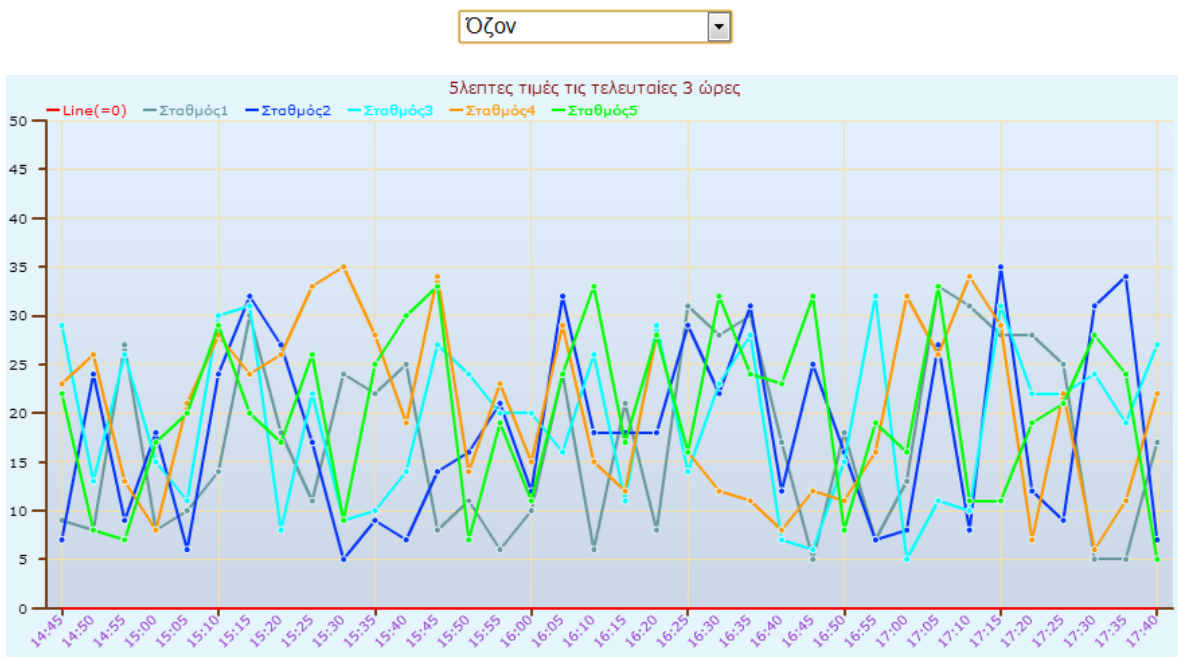
Χάρτη με απεικόνιση του δείκτη ρύπανσης και του ανέμου

Στις εικόνες 3α, 3β, 3γ, 3δ, 3ε παρουσιάζονται τα γραφήματα, όπως δημιουργήθηκαν από την εφαρμογή Air-DMS.



© 2009 www.air-quality.gr

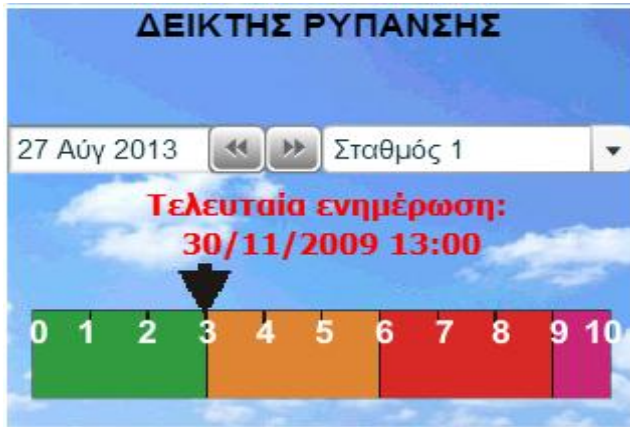
Εικόνα 3α. Air-DMS Διάγραμμα ρύπου – ανέμου



\*\*\* Τυχαίες τιμές με αυτόματη ανανέωση κάθε 1 min.

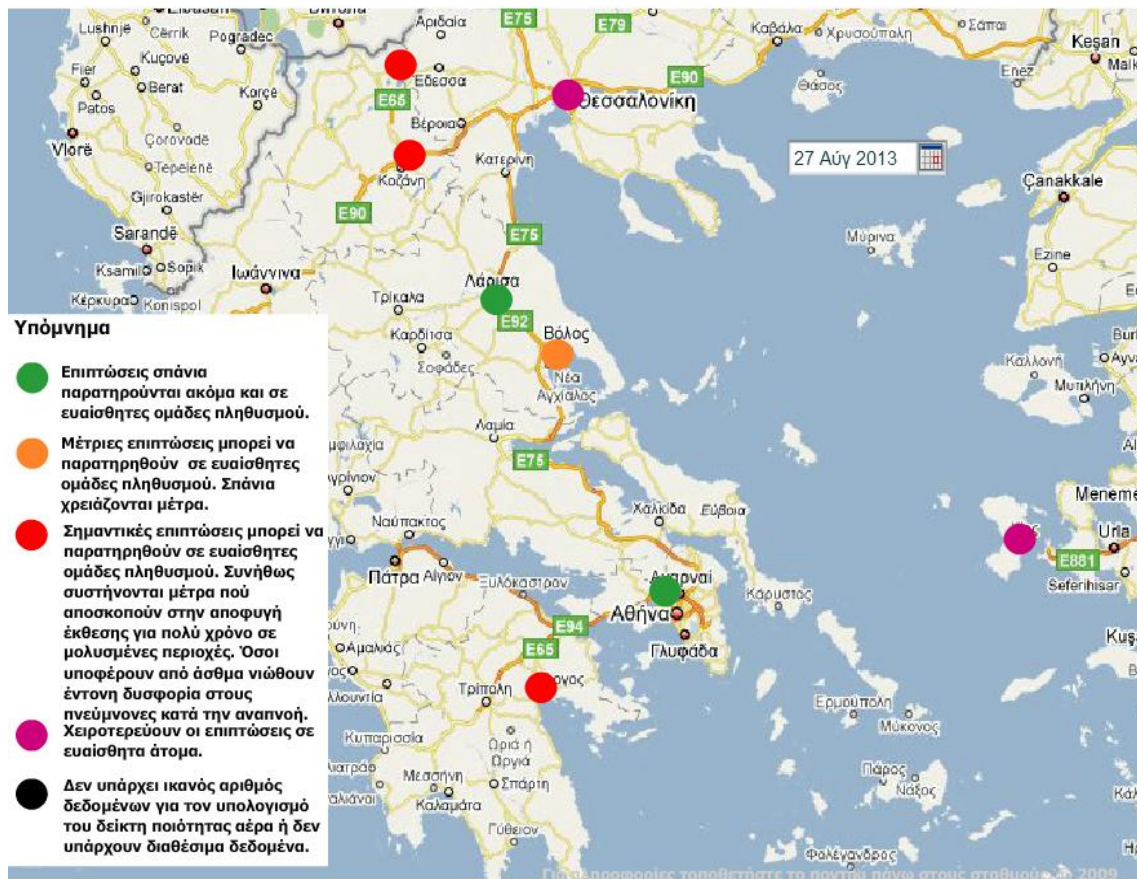
© 2009 www.air-quality.gr

Εικόνα 3β. Air-DMS Διάγραμμα ρύπου σε όλους τους σταθμούς

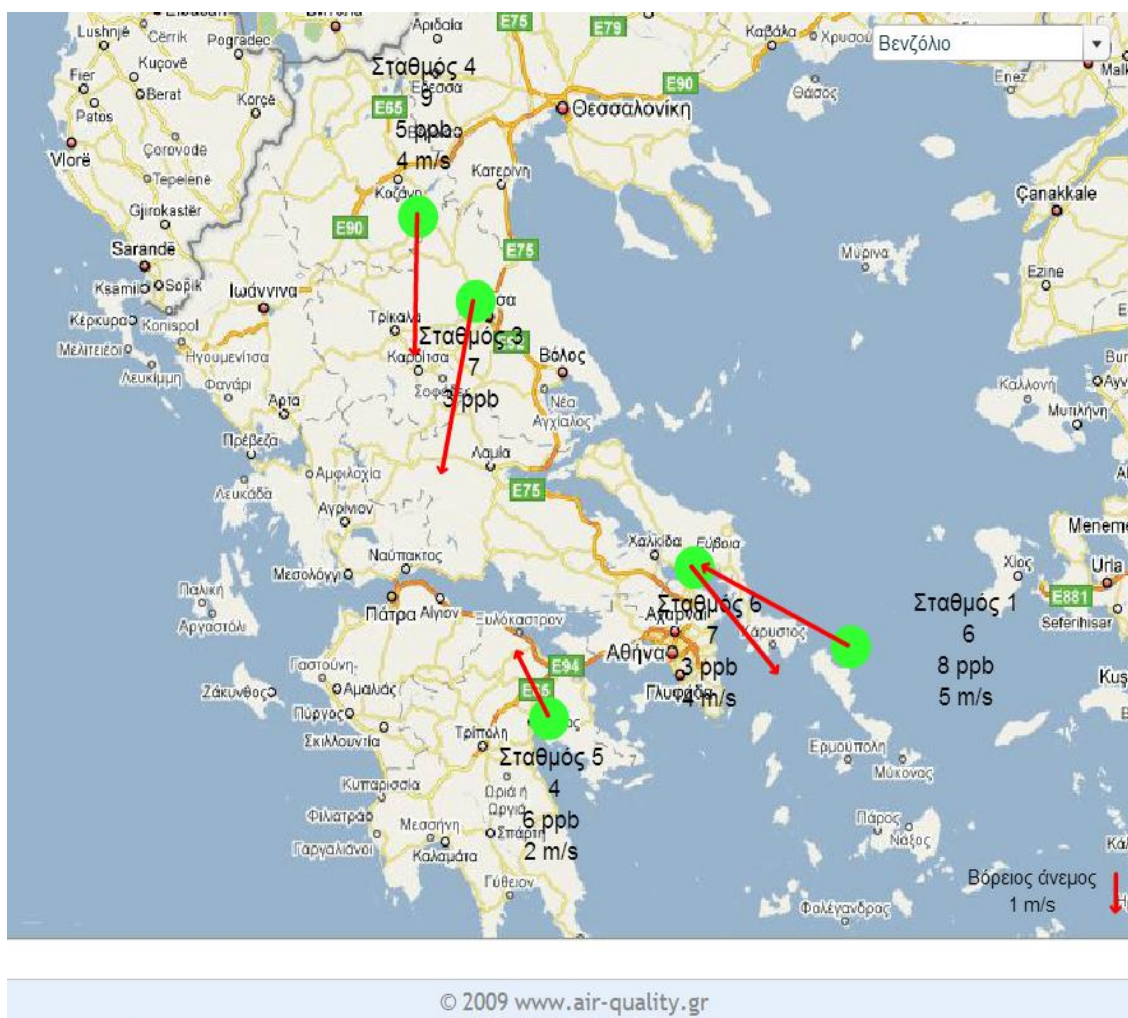


© 2009 www.air-quality.gr

Εικόνα 3γ. Air-DMS Γράφημα δείκτη ρύπανσης



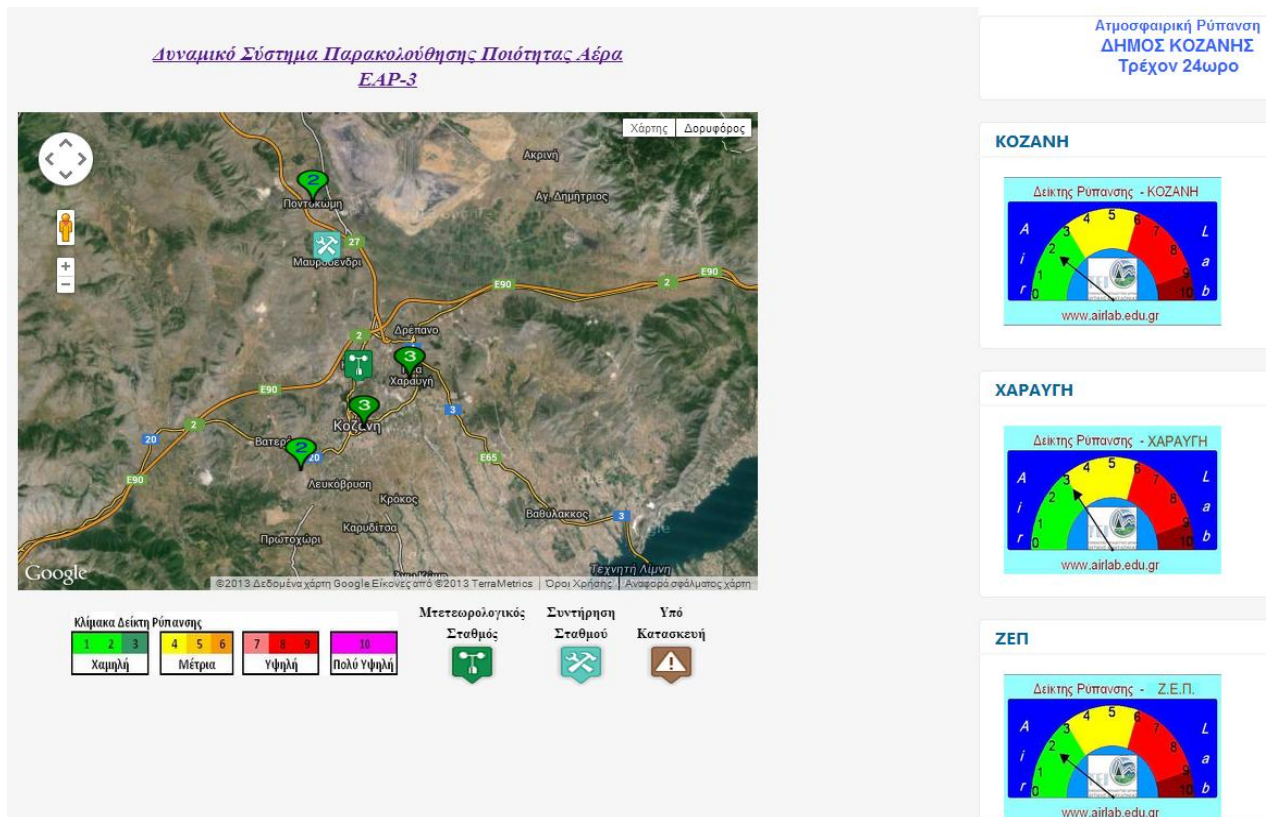
Εικόνα 3δ. Air-DMS Χάρτης με δείκτη ρύπανσης



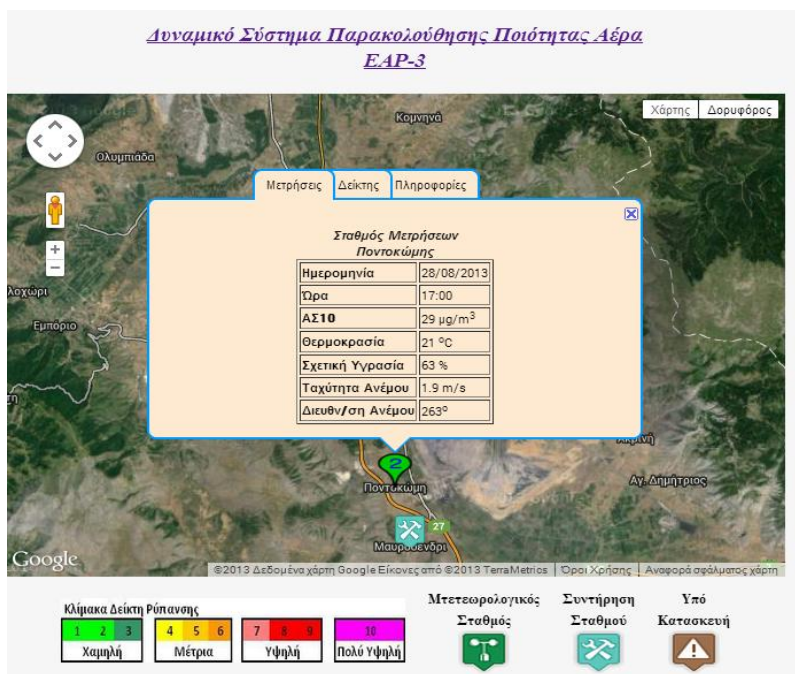
**Εικόνα 3ε. Air-DMS Χάρτης με δείκτη ρύπανσης και άνεμο**

Μία άλλη πολύ αξιόλογη διαδικτυακή εφαρμογή, έχει αναπτυχθεί από το Εργαστήριο Ατμοσφαιρικής Ρύπανσης και Περιβαλλοντικής Φυσικής (Ε.Α.Ρ.-ΠΕ.ΦΥ.) το οποίο αποτελεί διατμηματικό εργαστήριο της Σχολής Τεχνολογικών Εφαρμογών του ΤΕΙ Δυτικής Μακεδονίας ([http://www.airlab.edu.gr/index.php?option=com\\_wrapper&view=wrapper&Itemid=165&lang=el](http://www.airlab.edu.gr/index.php?option=com_wrapper&view=wrapper&Itemid=165&lang=el)). Ονομάζεται ΕΑΡ-3 και δίνει στον επισκέπτη τη δυνατότητα να πληροφορηθεί σε πραγματικό χρόνο για τις τιμές μέτρησης των μικροσωματιδίων PM10 (σωματίδια μεγέθους <math><10\mu\text{m}</math>), και άλλα χαρακτηριστικά του ατμοσφαιρικού αέρα, όπως θερμοκρασία, σχετική υγρασία, ταχύτητα και διεύθυνση ανέμου, στους δήμους Κοζάνης και Αμυνταίου. Τα δεδομένα, προέρχονται από τέσσερεις σταθμούς μέτρησης στην ευρύτερη περιοχή του Δήμου Κοζάνης. Ο επισκέπτης της ιστοσελίδας βλέπει σε χάρτη τη θέση των σταθμών οι οποίοι χρωματίζονται ανάλογα με την ποιότητα του ατμοσφαιρικού αέρα που μετρούν, ώστε να έχει άμεσα μια εποπτική εικόνα της κατάστασης. Επίσης, κάνοντας κλικ σε κάποιον από τους σταθμούς μπορεί να δει όλα δεδομένα έχει καταγράψει καθώς και τα χαρακτηριστικά του.

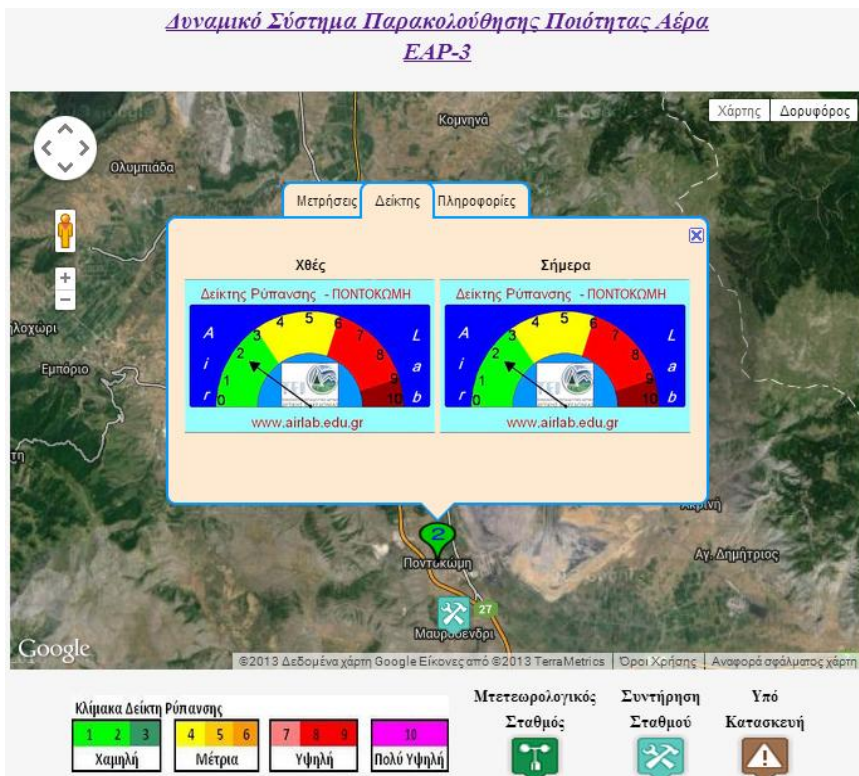
Η τεκμηρίωση της εφαρμογής είναι πληρέστατη, καθώς αναφέρει αναλυτικά τον εξοπλισμό και τη μεθοδολογία που χρησιμοποιήθηκε για τη συλλογή των δεδομένων, και τον τρόπο επεξεργασία και διάθεσης τους. Στην εικόνα 4 παρουσιάζεται αρχική οθόνη της εφαρμογής και στις εικόνες 4α, 4β, 4γ τα δεδομένα που έχει καταγράψει ο σταθμός



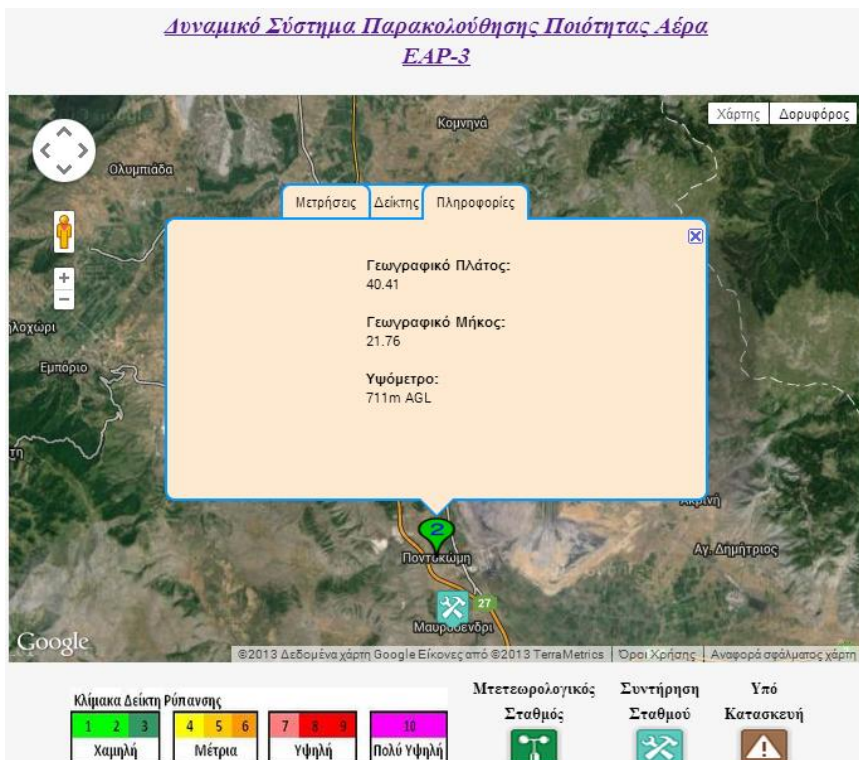
Εικόνα 4. EAP-3 Αρχική οθόνη εφαρμογής



Εικόνα 4α. EAP-3 Δεδομένα μετρήσεων σταθμού Ποντοκώμης



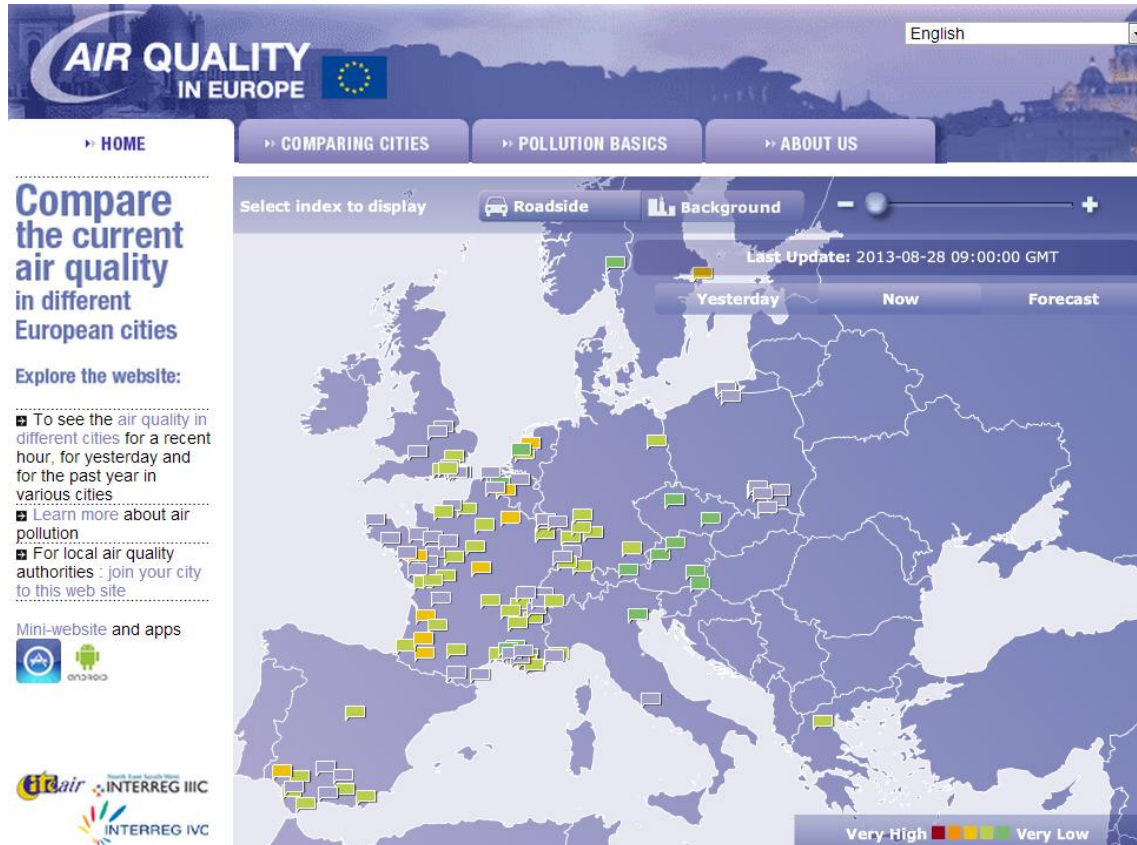
Εικόνα 4β. EAP-3 Δείκτες μετρήσεων σταθμού Ποντοκώμης



Εικόνα 4γ. EAP-3 Χαρακτηριστικά σταθμού Ποντοκώμης

### 3.2 Αντίστοιχες εφαρμογές διεθνώς

Η Ευρωπαϊκή Ένωση έχει αναπτύξει μια διαδικτυακή εφαρμογή η οποία ενημερώνει τους πολίτες των χωρών που την απαρτίζουν για την ποιότητα του ατμοσφαιρικού αέρα ορισμένων πόλεων (<http://www.airqualitynow.eu/index.php>). Η Εικόνα 5 δείχνει τη συγκεκριμένη ιστοσελίδα. Τα δεδομένα τα διαθέτουν οι φορείς κάθε πόλης και οι χρήστες ενημερώνονται με απλή επίσκεψη στη σελίδα. Εκεί, βλέπουν σε χάρτη όλη την Ευρωπαϊκή ήπειρο με τις πόλεις που συμμετέχουν να έχουν χρωματιστές ταμπέλες, όπου το χρώμα αντιπροσωπεύει τα επίπεδα ρύπανσης. Τα δεδομένα ενημερώνονται ανά τακτά χρονικά διαστήματα, ώστε να προσομοιάζει λειτουργία πραγματικού χρόνου.



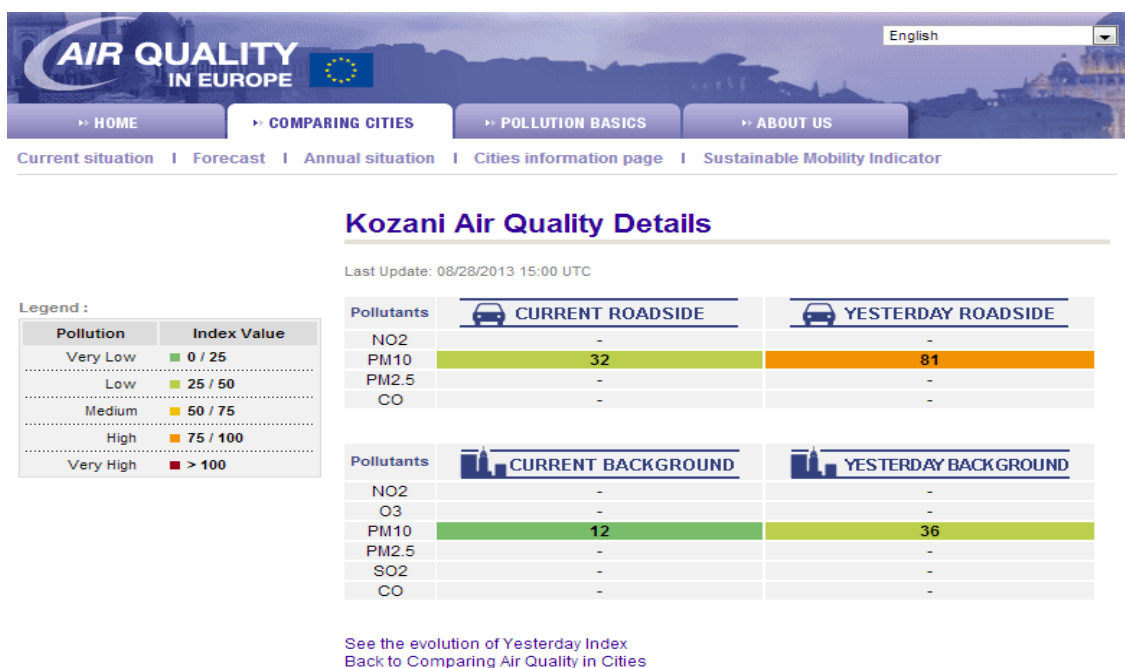
Εικόνα 5. Αρχική οθόνη της εφαρμογής της Ε.Ε.

Ο χρήστης μπορεί να δει τις μετρήσεις της συγκεκριμένης ημέρας μαζί με την ώρα ενημέρωσης ή τις μετρήσεις της προηγούμενης ημέρας. Για την προηγούμενη ημέρα, μπορεί να δει τις μετρήσεις ανά ώρα. Επίσης, η εφαρμογή του δίνει τη δυνατότητα να δει την πρόγνωση της ποιότητας του ατμοσφαιρικού αέρα για την επόμενη ημέρα, σε περίπτωση φυσικά που η πόλη διαθέτει τα στοιχεία. Γενικά, η εφαρμογή οπτικοποιεί ό,τι δεδομένα παρέχει κάθε συμμετέχουσα πόλη.

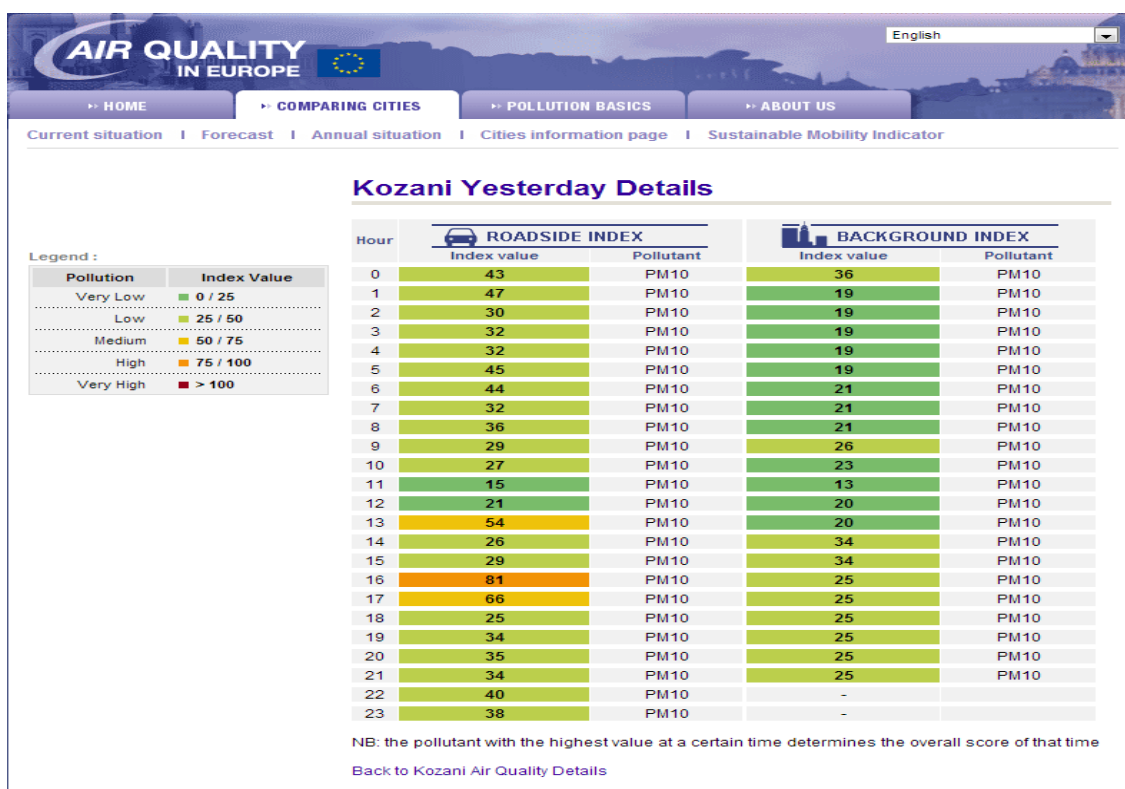
Όσον αφορά την Ελλάδα, η εφαρμογή περιέχει δεδομένα μόνο για την πόλη της Κοζάνης και μόνο για έναν ρύπο, τα μικροσωματίδια PM10 (σωματίδια μεγέθους <math><10\mu\text{m}</math>). Χαρακτηριστικό είναι το γεγονός πως μετράται η τιμή του κάθε ρύπου τόσο στους οδικούς άξονες, περιοχές ιδιαίτερα επιβαρυμένες από την κίνηση οχημάτων, όσο και στο ευρύτερο περιβάλλον. Τα δεδομένα διατίθενται από το Εργαστήριο Ατμοσφαιρικής Ρύπανσης και Περιβαλλοντικής Φυσικής (Ε.Α.Ρ.-ΠΕ.ΦΥ.) και είναι τα ίδια με την εφαρμογή ΕΑΡ-3.

Οι Εικόνες 5α, 5β απεικονίζουν τα δεδομένα της εφαρμογής για το σταθμό της Κοζάνης.





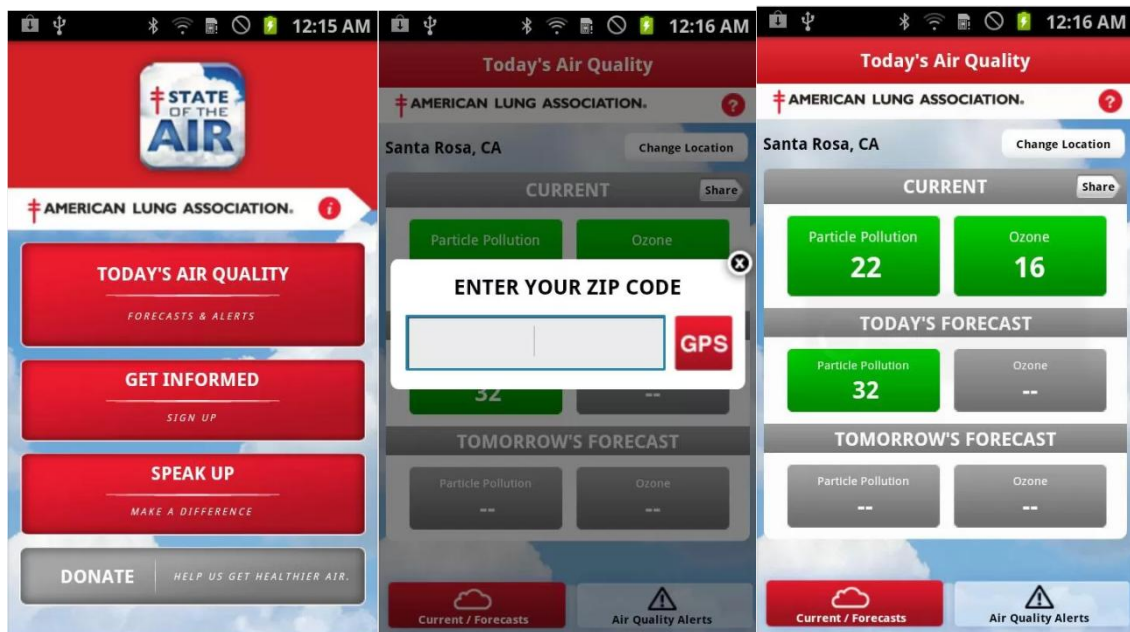
Εικόνα 5α. Δεδομένα του σταθμού Κοζάνης



Εικόνα 5β. Δεδομένα της προηγούμενης ημέρας του σταθμού Κοζάνης

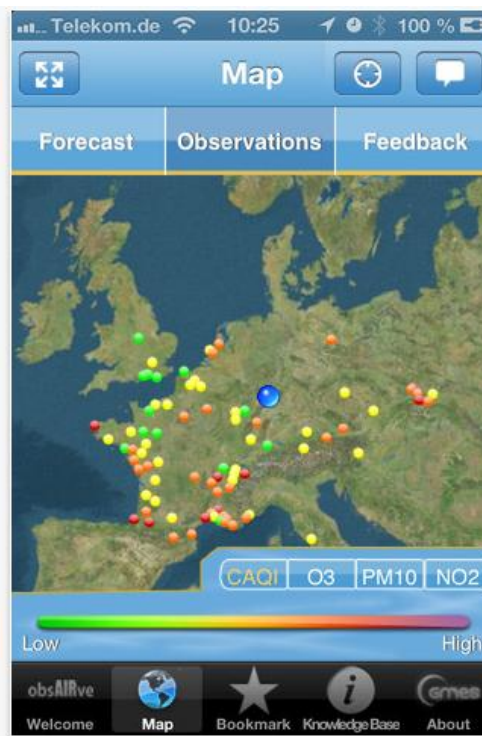
Τέλος, με την επικράτηση των smartphones την τελευταία περίοδο, έχουν δημιουργηθεί κάποιες εφαρμογές για τα λειτουργικά Android και iPhone OS με σκοπό να ενημερώνουν τους χρήστες για την ποιότητα του ατμοσφαιρικού αέρα μέσω κινητού τηλεφώνου.

Πιο συγκεκριμένα, η American Lung Association έχει αναπτύξει μία εφαρμογή η οποία ονομάζεται State of the Air, (<http://www.lung.org/healthy-air/outdoor/state-of-the-air/app.html>), συμβατή τόσο με Android όσο και με iOS λειτουργικά συστήματα η οποία ενημερώνει τους χρήστες για τα επίπεδα ρύπανσης του ατμοσφαιρικού αέρα στις πόλεις των Η.Π.Α. Ο χρήστης, καλείται να δώσει το ταχυδρομικό κωδικό της περιοχής ή να ενεργοποιήσει τη γεωαναφορά στο τηλέφωνο του και ενημερώνεται άμεσα για τα επίπεδα τιμών όζοντος και μικροσωματιδίων στον ατμοσφαιρικό αέρα της περιοχής του. Η εφαρμογή χρησιμοποιεί δεδομένα που διατίθενται από το U.S. Environmental Protection Agency (EPA), και εκτός από τις μετρημένες τιμές των ρύπων, έχει και τη δυνατότητα πρόγνωσης για τις τιμές της επόμενης ημέρας. Οι εικόνες 6α, 6β, 6γ δείχνουν την εφαρμογή.

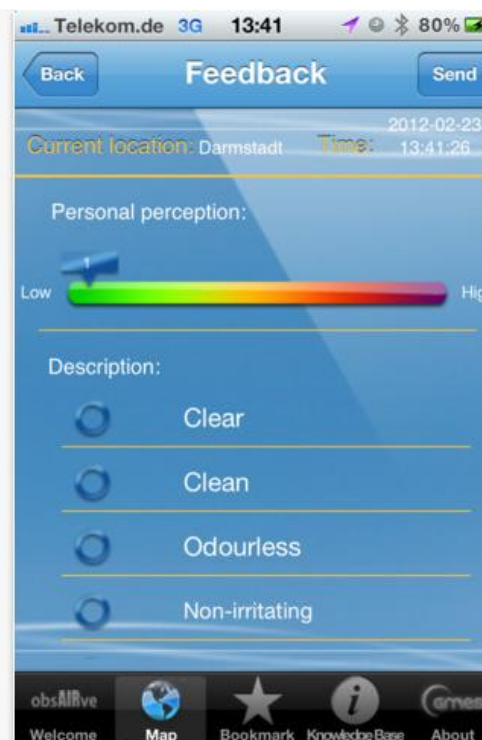


Εικόνες 6α, 6β, 6γ. Η εφαρμογή State of the Air

Μία άλλη εφαρμογή με αντίστοιχες δυνατότητες αλλά διαθέσιμη μόνο για λειτουργικά iOS είναι η ObsAIRve (<https://itunes.apple.com/de/app/obsairve/id490939811?mt=8>). Η συγκεκριμένη, παρέχεται από την Ευρωπαϊκή Επιτροπή και αποτελεί μέρος του προγράμματος GMES (Global Monitoring for Environment and Security). Με αυτή οι χρήστες είναι σε θέση να πληροφορηθούν τα επίπεδα Όζοντος, διοξειδίου του Αζώτου και μικροσωματιδίων στον ατμοσφαιρικό αέρα των ευρωπαϊκών πόλεων που συμμετέχουν στο πρόγραμμα. Εκτός από τις μετρήσεις, σχεδόν πραγματικού χρόνου, καθώς οι ενδείξεις ανανεώνονται ανά τακτά χρονικά διαστήματα, η εφαρμογή παρέχει επίσης τη δυνατότητα πρόβλεψης της ατμοσφαιρικής ρύπανσης για το προσεχές 24ωρο. Στις Εικόνες 7α, 7β, 7γ, 7δ παρακάτω παρουσιάζεται η εφαρμογή η οποία αναμένεται στο μέλλον να διατεθεί και για το λειτουργικό σύστημα Android.



Εικόνες 7α, 7β. Η εφαρμογή obsAIRve



Εικόνες 7γ, 7δ. Η εφαρμογή obsAIRve

Ακόμα, έχουν αναπτυχθεί δύο εφαρμογές για λειτουργικό Android οι οποίες μετρούν την ποιότητα του ατμοσφαιρικού αέρα στην Κίνα. Όπως είναι γνωστό, η ασιατική χώρα αντιμετωπίζει τεράστιο πρόβλημα ατμοσφαιρικής ρύπανσης και αυτές οι εφαρμογές φιλοδοξούν να ενημερώσουν τους πολίτες για την ποιότητα του αέρα σε πραγματικό χρόνο.

Η πρώτη εφαρμογή ονομάζεται Beijing Air Quality, είναι διαθέσιμη στο Android Market (<https://play.google.com/store/apps/details?id=com.longcatlabs.beijingairquality&hl=el>) και παρέχει δεδομένα μόνο για το Πεκίνο. Τα δεδομένα αφορούν στο δείκτη ρύπανσης, συλλέγονται από την πρεσβεία των Η.Π.Α. στην πόλη, και ανανεώνονται κάθε μία ώρα. Η εφαρμογή παρουσιάζει σε γραφήματα 24ώρου και 7ημέρου τις τιμές ρύπανσης. Οι Εικόνες 8α, 8β και 8γ δείχνουν την εφαρμογή



Εικόνες 8α, 8β, 8γ. Η εφαρμογή Beijing Air Quality

Η δεύτερη εφαρμογή για την Κίνα ονομάζεται Air Quality China (<https://play.google.com/store/apps/details?id=com.cas.airquality&hl=el>) όμως δεν αναφέρει τίποτα σχετικά με την προέλευση των δεδομένων ή την μεθοδολογία.

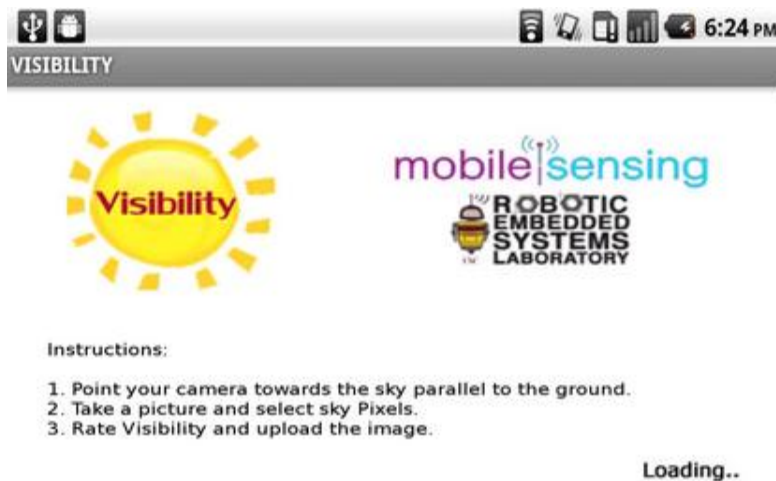


Εικόνες 9α, 9β, 9γ, 9δ. Η εφαρμογή Air Quality China

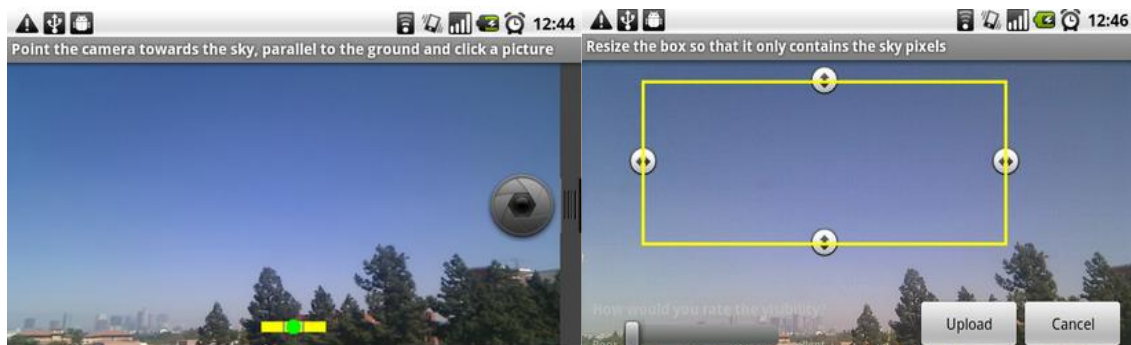
Τέλος, μία εντελώς διαφορετική εφαρμογή αναπτύχθηκε από τον καθηγητή – ερευνητή του University of Southern California, Guarav Sukhatme για smartphones με λειτουργικό Android. Η συγκεκριμένη, βρίσκεται ακόμα σε πειραματικό στάδιο, ονομάζεται Visibility (<http://robotics.usc.edu/~mobilesensing/Projects/AirVisibilityMonitoring>) και η διαφορά της έγκειται στον τρόπο που αποφαινεται για την ατμοσφαιρική ρύπανση.

Πιο αναλυτικά, η εφαρμογή δεν αξιοποιεί δεδομένα από σταθμούς μέτρησης αερίων ρύπων αλλά εκτιμά την ποιότητα του ατμοσφαιρικού αέρα μέσω φωτογραφιών που βγάζει ο χρήστης και ανάλυσης τους μέσω πολύπλοκων αλγορίθμων. Ουσιαστικά, συγκρίνεται η φωτογραφία του χρήστη με εικόνες καθαρής ατμόσφαιρας και η ρύπανση προκύπτει από την διαφορά τους.

Το μεγάλο πλεονέκτημα αυτής της προσέγγισης είναι το γεγονός ότι δεν χρησιμοποιεί δεδομένα σταθμών, οι οποίοι λόγω της χωρικής κατανομής τους δεν προσφέρουν αξιόπιστα δεδομένα για περιοχές που βρίσκονται σε απόσταση από αυτούς. Επίσης, η εκτίμηση αφορά τη στιγμή που λήφθηκε η εικόνα, κάτι που σημαίνει πως είναι απολύτως πραγματικού χρόνου, χωρίς καμία καθυστέρηση. Το μειονέκτημα είναι πως δεν προσφέρει καμία πληροφορία για τις συγκεντρώσεις άχρωμων αερίων όπως το Όζον ή τα οξείδια του Αζώτου. Οι Εικόνες 10α, 10β, 10γ δείχνουν την λειτουργία της εφαρμογής.



Εικόνα 10α. Η αρχική οθόνη της εφαρμογής Visibility

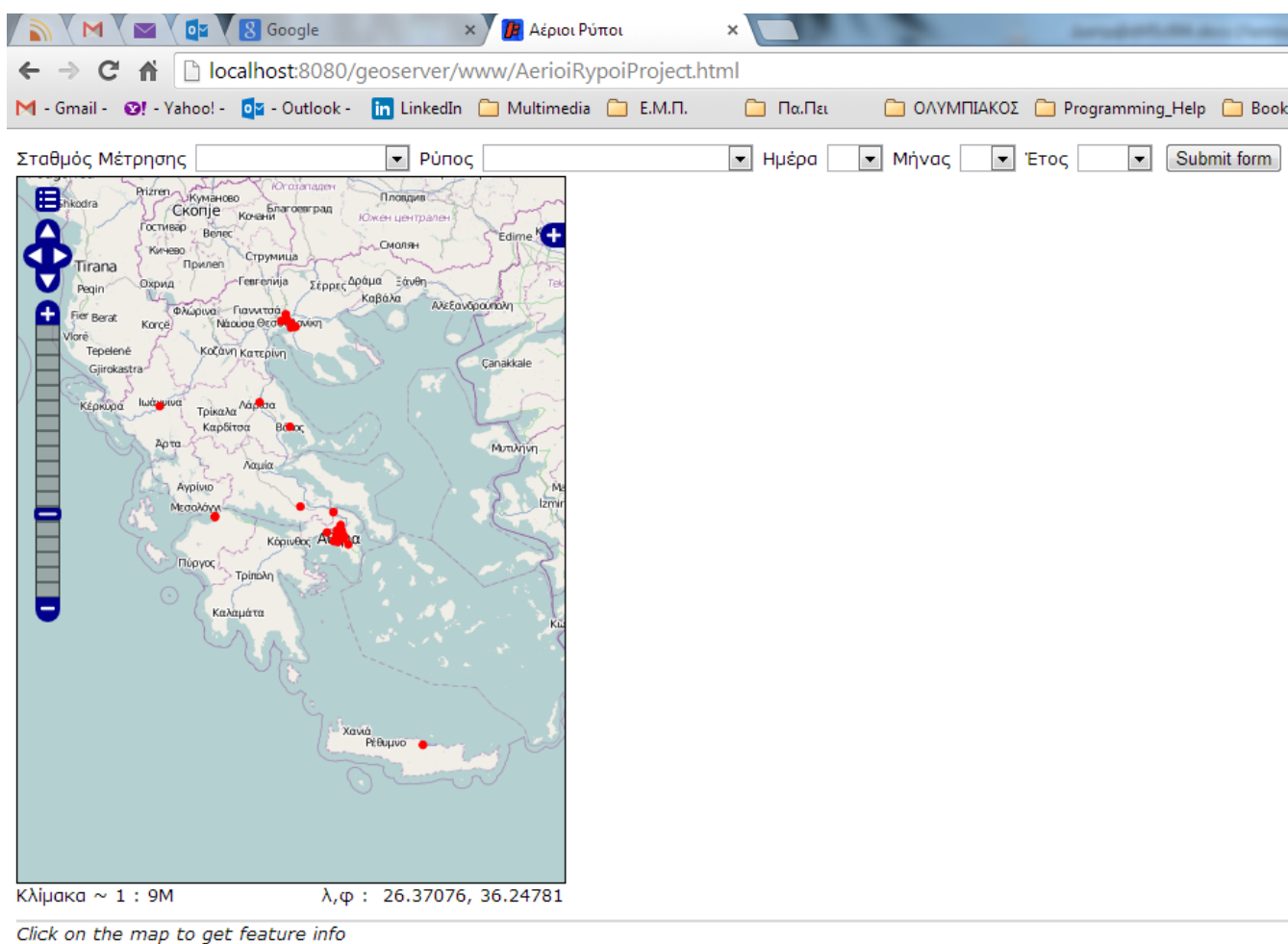


Εικόνες 10β, 10γ. Η εφαρμογή Visibility εν λειτουργία

#### 4. Παρουσίαση και χρήση της εφαρμογής

Ο χρήστης – επισκέπτης της εφαρμογής το μόνο που χρειάζεται είναι να διαθέτει περιηγητή διαδικτύου και να πληκτρολογήσει το url. Ο χρήστης, με την επίσκεψη του στην ιστοσελίδα βλέπει ένα χάρτη με τους σταθμούς μέτρησης των αερίων ρύπων πανελλαδικά. Αυτό, υλοποιείται μέσω της υπηρεσίας WMS (Web Map Service). Ο χάρτης είναι δυναμικός, δηλαδή επιτρέπει λειτουργίες όπως pan ή zoom. Επίσης, το μέγεθος του χάρτη (πλάτος/ύψος) μπορεί να ρυθμιστεί κατά τη βούληση του χρήστη. Κατά τη φόρτωση της ιστοσελίδας ο χάρτης ρυθμίζεται να απεικονίζει στο κατάλληλο zoom το σύνολο του Ελλαδικού χώρου.

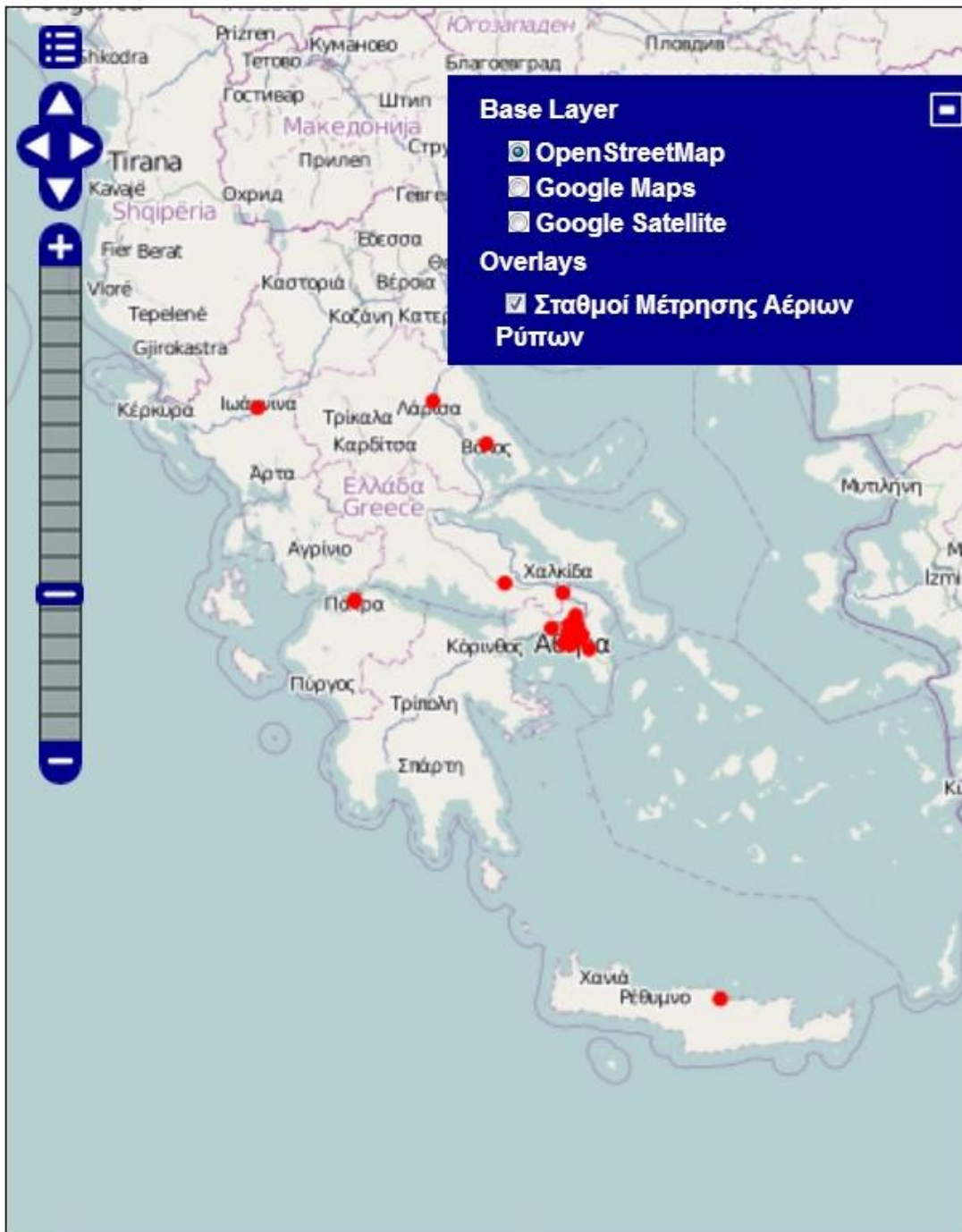
Η εικόνα 11 δείχνει την εφαρμογή όπως τη βλέπει ο επισκέπτης της ιστοσελίδας



**Εικόνα 11. Η αρχική σελίδα της εφαρμογής**

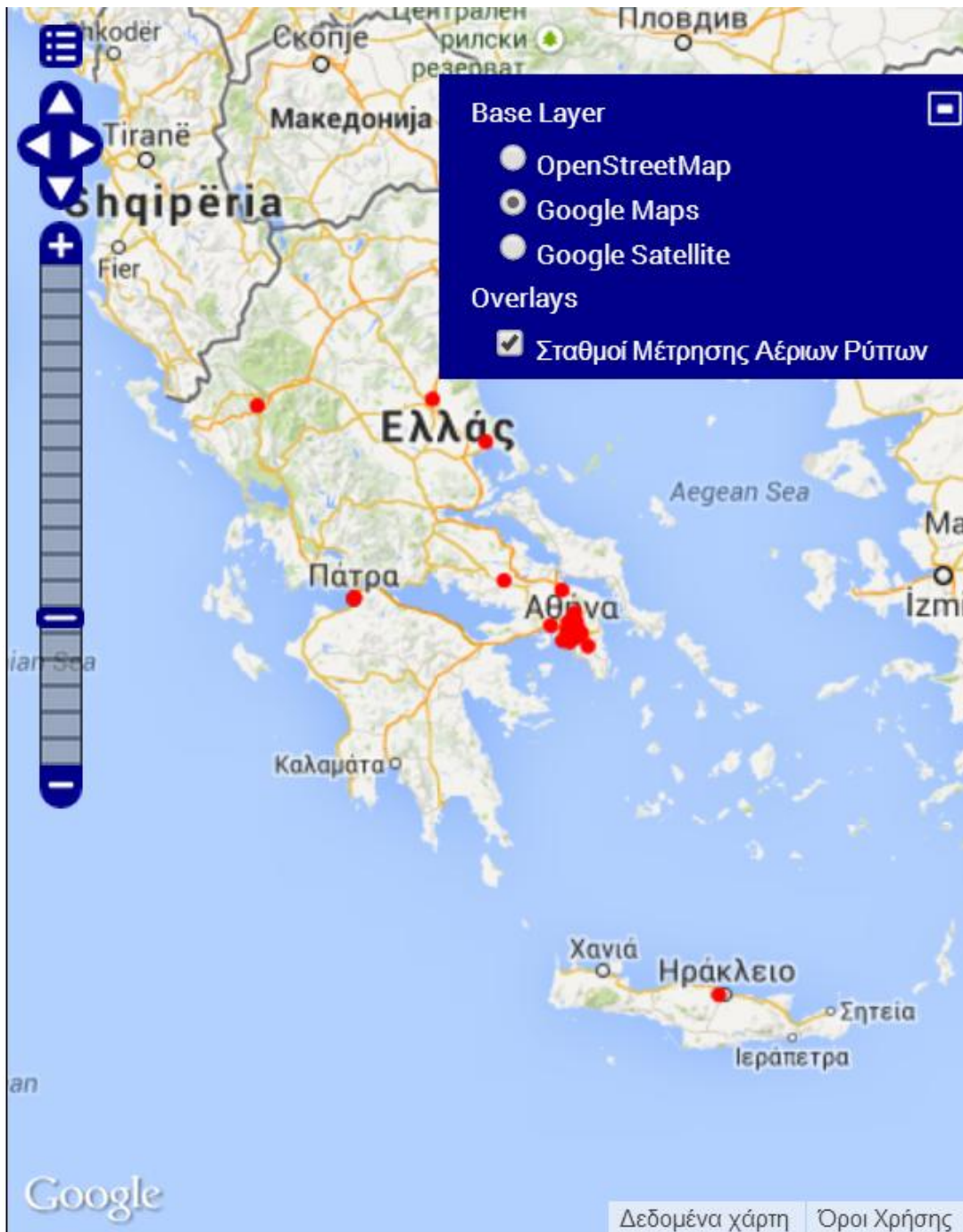
Στο κάτω μέρος του χάρτη, αναγράφεται η κλίμακα θέασης καθώς και οι συντεταγμένες λ,φ στο σύστημα WGS84 της θέσης του ποντικιού. Η προεπιλογή υποβάθρου είναι χάρτης OSM (Open Street Map), αλλά ο χρήστης μπορεί να επιλέξει αν θέλει και τους χάρτες Google (Google Map ή Google Satellite). Επίσης, μπορεί αν θέλει να αποκρύψει τις θέσεις των σταθμών από το χάρτη, αποεπιλέγοντας το συγκεκριμένο layer.

Η εικόνα 12 δείχνει τις επιλογές υποβάθρου – layer του χάρτη.



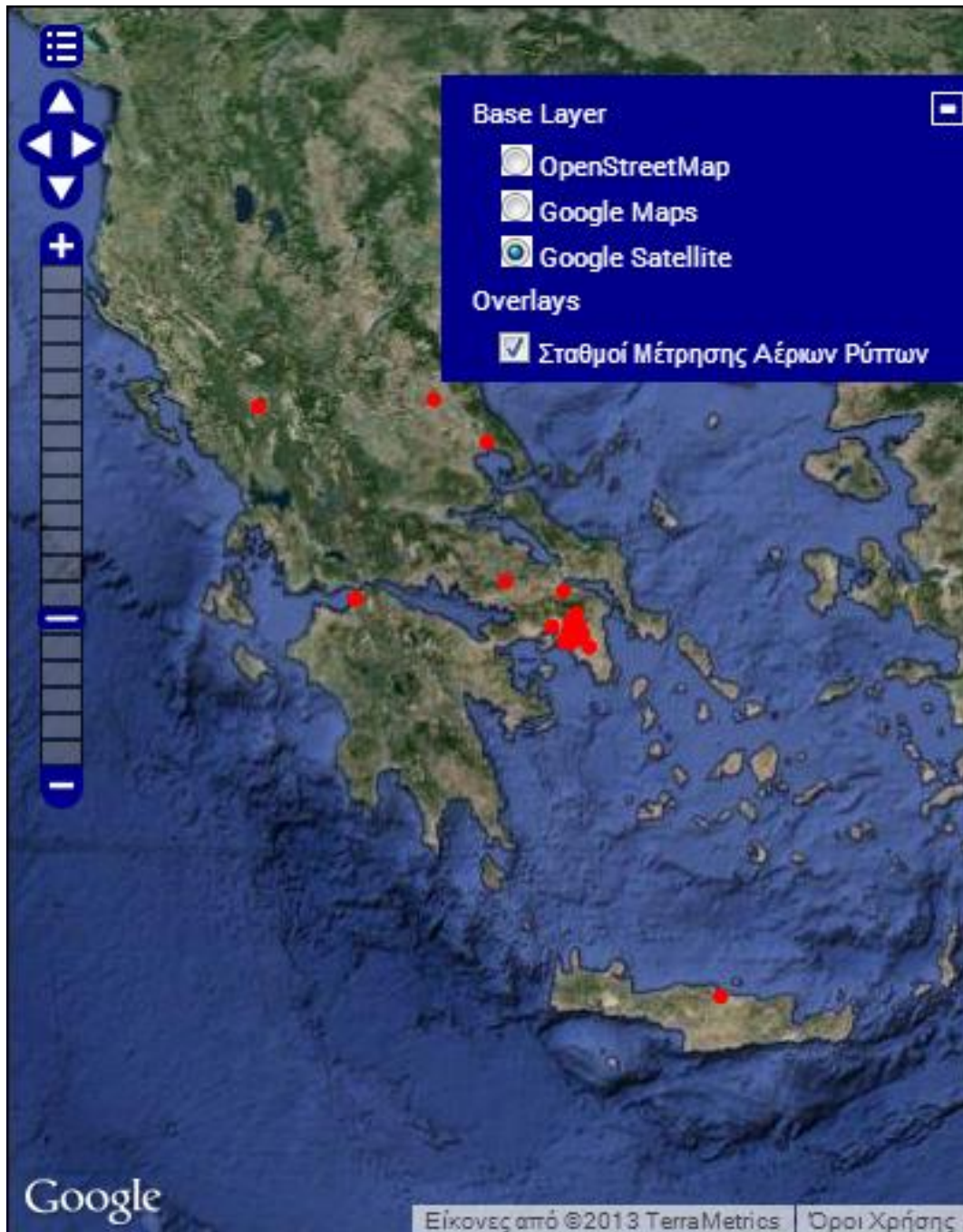
Εικόνα 12. Οι επιλογές του χάρτη

Στις εικόνες 13,14 φαίνεται ο χάρτης με υπόβαθρα Google Map και Google Satellite.



Εικόνα 13. Χάρτης Google Map

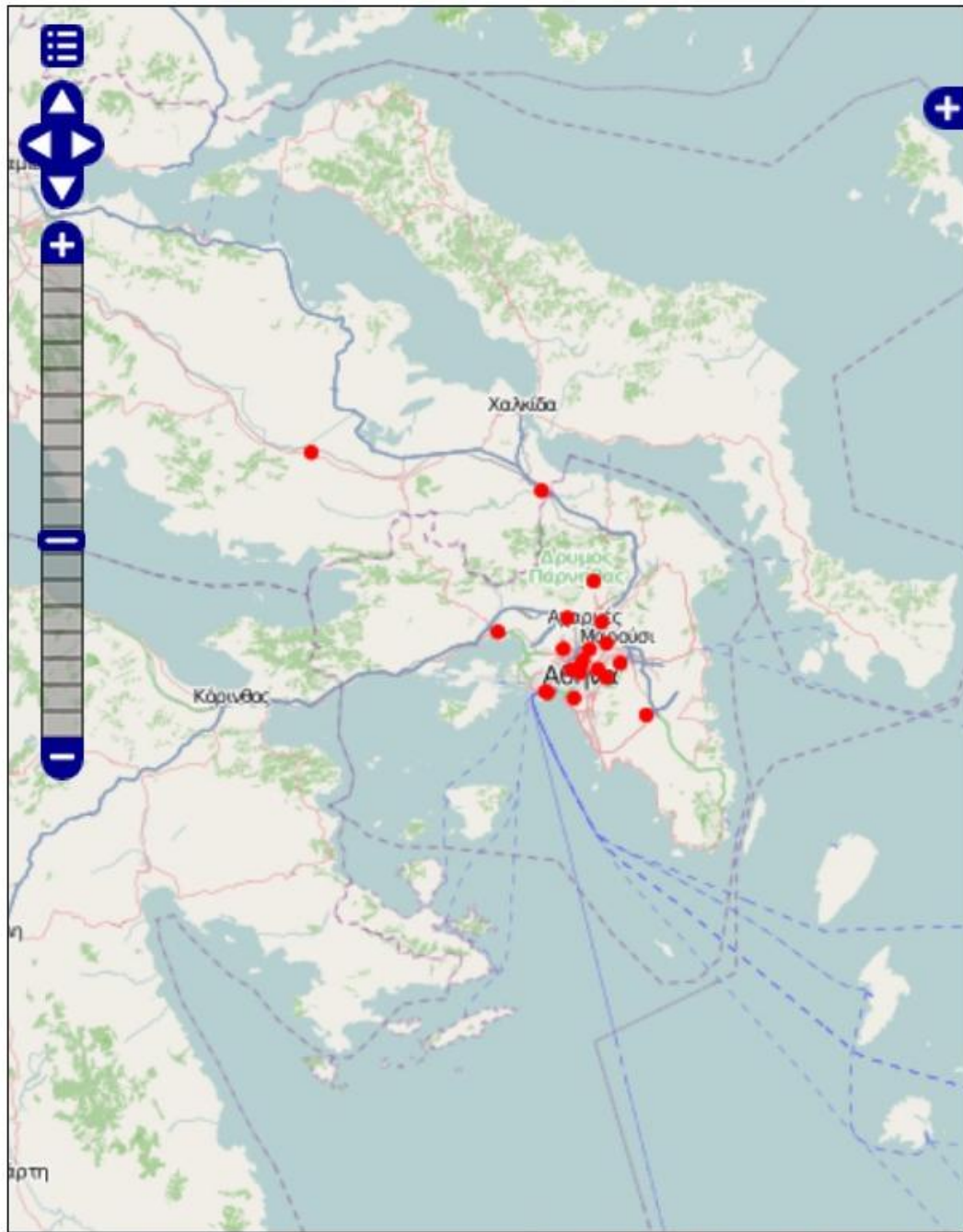




Εικόνα 14. Χάρτης Google Satellite

Όταν ο χρήστης κάνει κλικ πάνω σε κάποιον σταθμό, τότε μπορεί να δει τα χαρακτηριστικά του, όπως έχουν καταχωρηθεί στη Βάση Δεδομένων. Αυτό, καθίσταται εφικτό μέσω της υλοποιούμενης υπηρεσίας WFS (Web Feature Service).

Η Εικόνα 15 δείχνει την εν λόγω υπηρεσία για το σταθμό της Νέας Σμύρνης Αττικής.



Κλίμακα ~ 1 : 2M

λ,φ : 24.10941, 37.63365

**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.20	20	NEA_ΣΜΥΡΝΗ	50	Αστικός-Υποβάθρου	SMY

Εικόνα 15. Υπηρεσία WFS για το σταθμό Ν.Σμύρνης

Ο κύριος σκοπός της εφαρμογής, είναι να δώσει στο χρήστη τη δυνατότητα να αναζητήσει την μέση ημερήσια τιμή για κάθε έναν από τους επτά αερίους ρύπους, σε κάθε ένα από τους τριάντα σταθμούς μέτρησης, την ημερομηνία που τον ενδιαφέρει. Γι αυτό το λόγο έχει δημιουργηθεί μία φόρμα με τη μορφή μπάρας πάνω από το χάρτη, όπου ο χρήστης μπορεί να αναζητήσει τα στοιχεία που τον ενδιαφέρουν. Η φόρμα αυτή περιέχει:

- Το σταθμό μέτρησης
- Τον ρύπο μέτρησης
- Την ημέρα
- Τον μήνα
- Το έτος

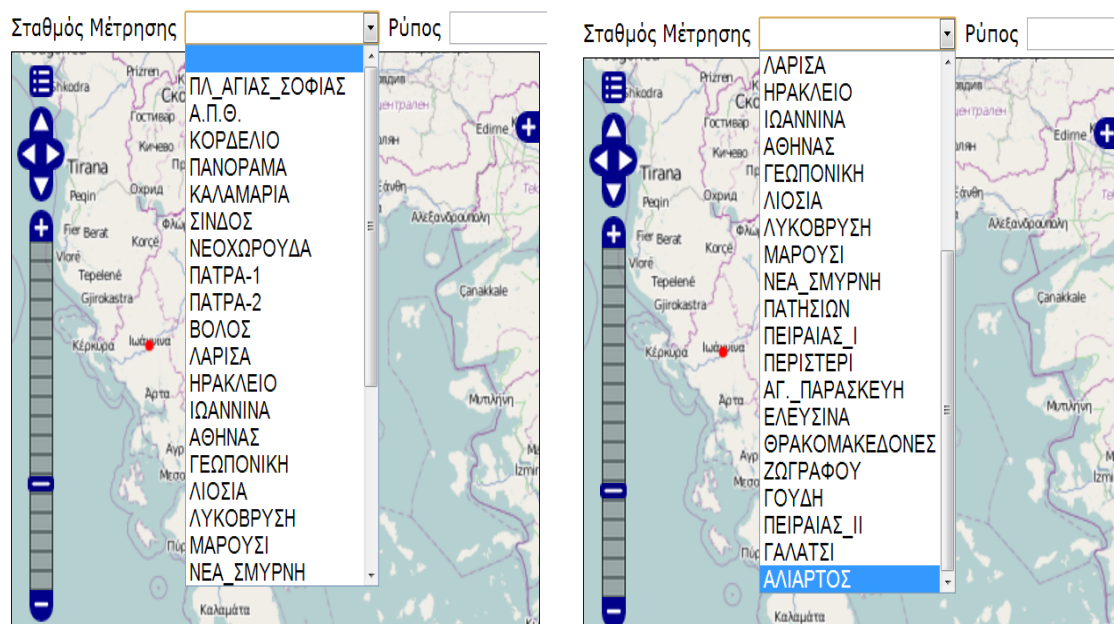
Στην εικόνα 16 παρουσιάζεται η φόρμα αναζήτησης.

Σταθμός Μέτρησης  Ρύπος  Ημέρα  Μήνας  Έτος

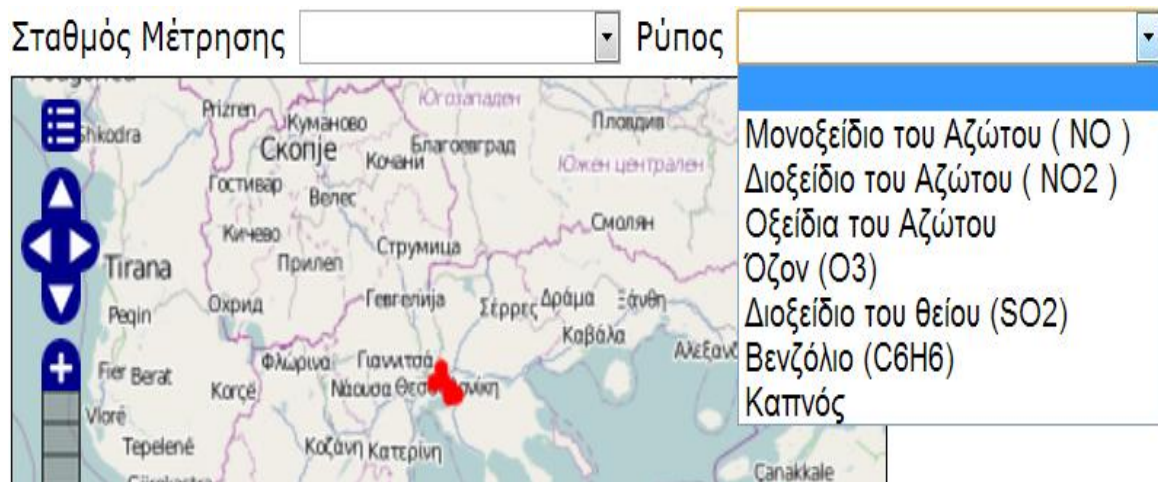
**Εικόνα 16. Η φόρμα αναζήτησης**

Όλα τα στοιχεία που εμφανίζονται στη φόρμα (Σταθμοί, Ρύποι, Ημέρα, Μήνας, Έτος), δεν προκύπτουν από τον πηγαίο κώδικα JavaScript, αλλά αναζητούνται από τη βάση δεδομένων. Αυτό, έγινε διότι κάνει την εφαρμογή δυναμική, υπό την έννοια ότι αν προστεθούν και άλλα δεδομένα στη βάση σε μεταγενέστερο χρόνο, ο χρήστης θα μπορεί να τα αναζητήσει, δηλαδή θα μπορούν να ενσωματωθούν στην λειτουργία της εφαρμογής.

Οι Εικόνες 17α, 17β δείχνουν τις δυνατές επιλογές αναζήτησης σταθμού από το χρήστη, ενώ η Εικόνα 18 την αναζήτηση ρύπου.



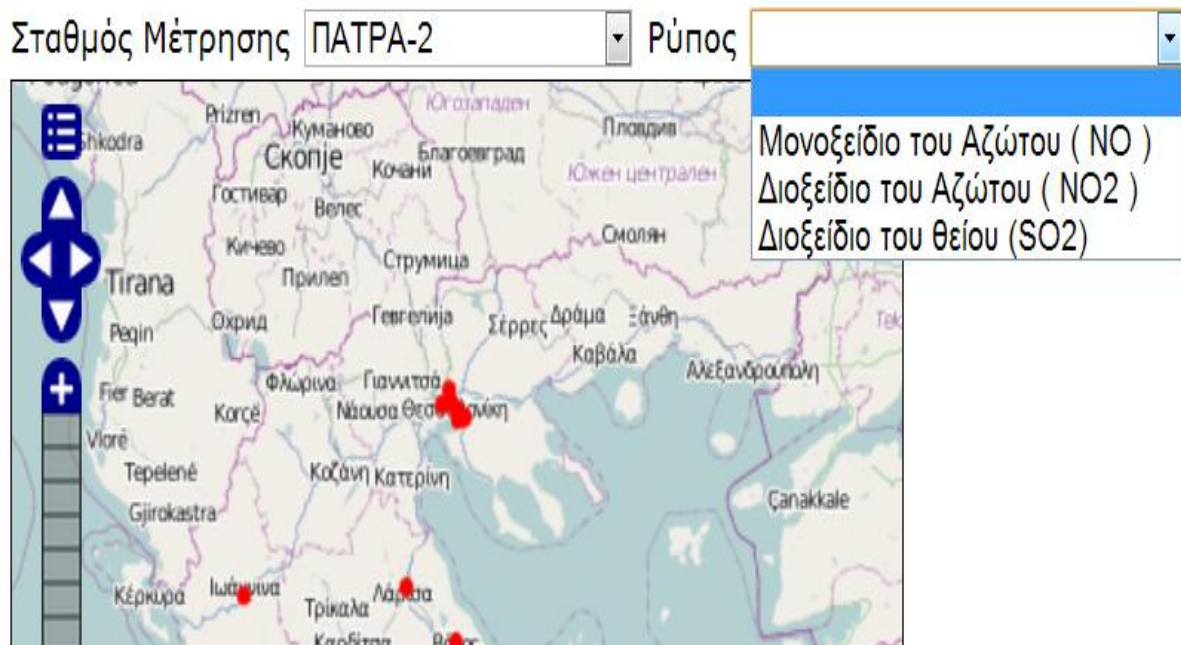
**Εικόνες 17α, 17β. Αναζήτηση σταθμού μέτρησης**



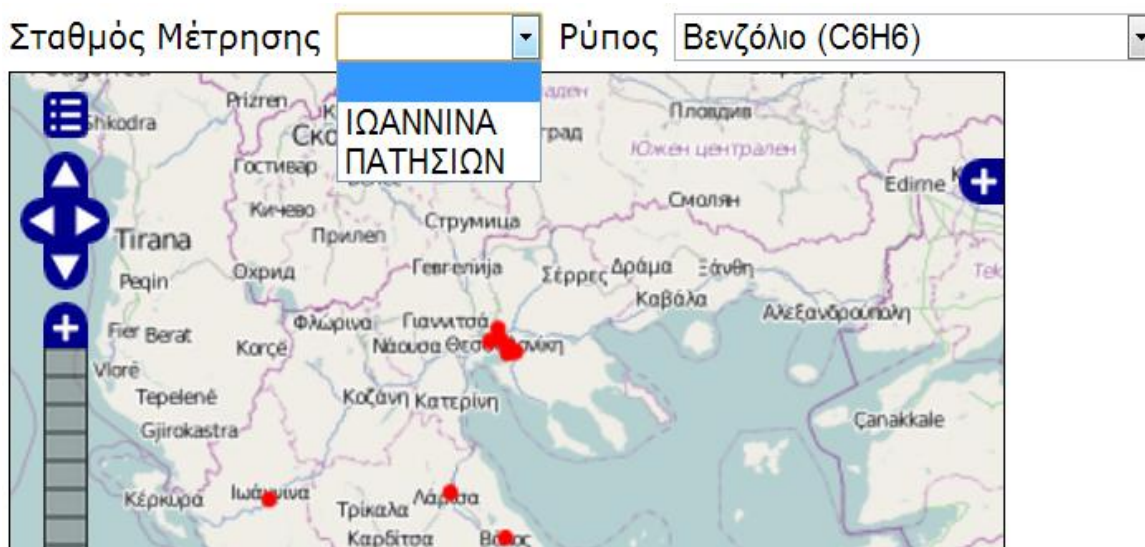
**Εικόνα 18. Αναζήτηση ρύπου**

Οι παραπάνω εικόνες είναι για την περίπτωση που δεν έχει γίνει ήδη επιλογή ρύπου(17α, 17β) ή επιλογή σταθμού (18). Πιο συγκεκριμένα, εάν ο χρήστης έχει ήδη επιλέξει σταθμό, η φόρμα θα του εμφανίσει μόνο τους ρύπους που μετράει ο συγκεκριμένος σταθμός. Αντίστοιχα, αν έχει ήδη επιλέξει ρύπο, στη φόρμα θα εμφανιστούν προς επιλογή μόνο οι σταθμοί που μετρούν το συγκεκριμένο ρύπο. Η φόρμα δηλαδή είναι δυναμική και δημιουργείται ανάλογα με τις επιλογές σταθμού ή ρύπου που θα κάνει ο χρήστης.

Στις Εικόνες 19, 20 φαίνεται το φιλτράρισμα που κάνει η εφαρμογή ανάλογα την επιλογή του χρήστη.

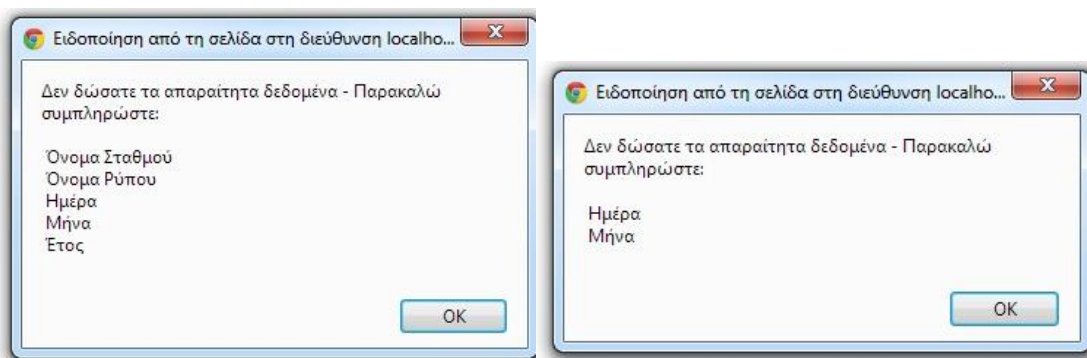


**Εικόνα 19. Φιλτράρισμα ρύπων με βάση τον επιλεγμένο σταθμό**



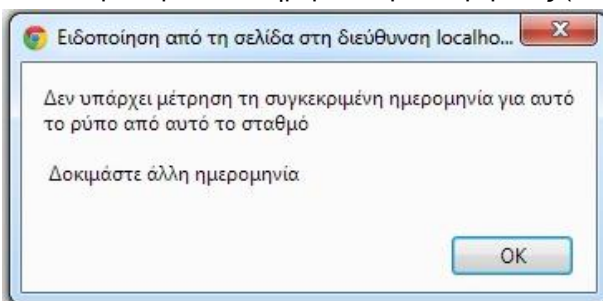
Εικόνα 20. Φιλτράρισμα σταθμών με βάση τον επιλεγμένο ρύπο

Για να αναζητήσει ο χρήστης την τιμή μέτρησης, πρέπει να συμπληρώσει όλα τα στοιχεία της φόρμας. Αν δεν το κάνει και κάποια δεν συμπληρωθούν, τότε ένα παράθυρο τον ενημερώνει για τις ελλείψεις (Εικόνες 21α, 21β)



Εικόνα 21α, 21β. Ελλιπής είσοδος από το χρήστη

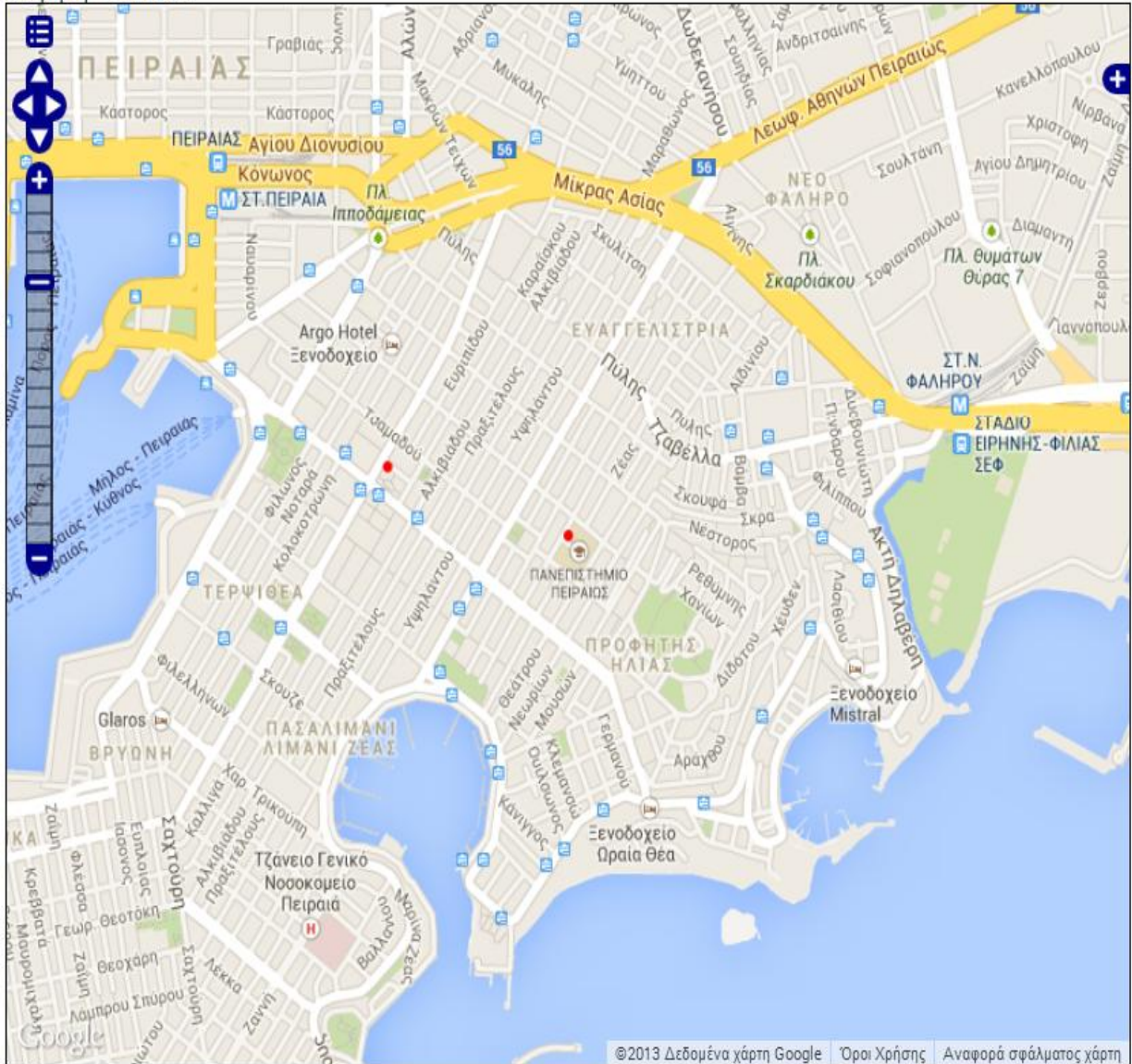
Αντίστοιχα, στην περίπτωση που τα έχει συμπληρώσει όλα αλλά δεν διατίθεται μέτρηση για το σταθμό και το ρύπο που επέλεξε στην συγκεκριμένη ημερομηνία (πολλές φορές στην περίοδο 2000-2011 ορισμένοι σταθμοί τέθηκαν εκτός λειτουργίας για κάποιο διάστημα), ένα άλλο παράθυρο τον ενημερώνει για το γεγονός (Εικόνα 22)



Εικόνα 22. Εκτέλεση ερωτήματος χωρίς αποτέλεσμα

Αν επιλεγούν έγκυρες τιμές εισόδου τότε ο χάρτης μετακινείται και κάνει zoom στον επιλεγμένο σταθμό ενώ εμφανίζεται και η τιμή μέτρησης του επιλεγμένου ρύπου την ημερομηνία που ζήτησε ο χρήστης. Οι Εικόνες 23 έως 33 δείχνουν στιγμιότυπα από επιτυχή αναζήτηση

Σταθμός Μέτρησης ΠΕΙΡΑΙΑΣ\_II    Ρύπος Μονοξειδίο του Αζώτου ( NO )    Ημέρα 10    Μήνας 1    Έτος 2001  
 Μέτρηση = 47.125



Κλίμακα ~ 1 : 17K

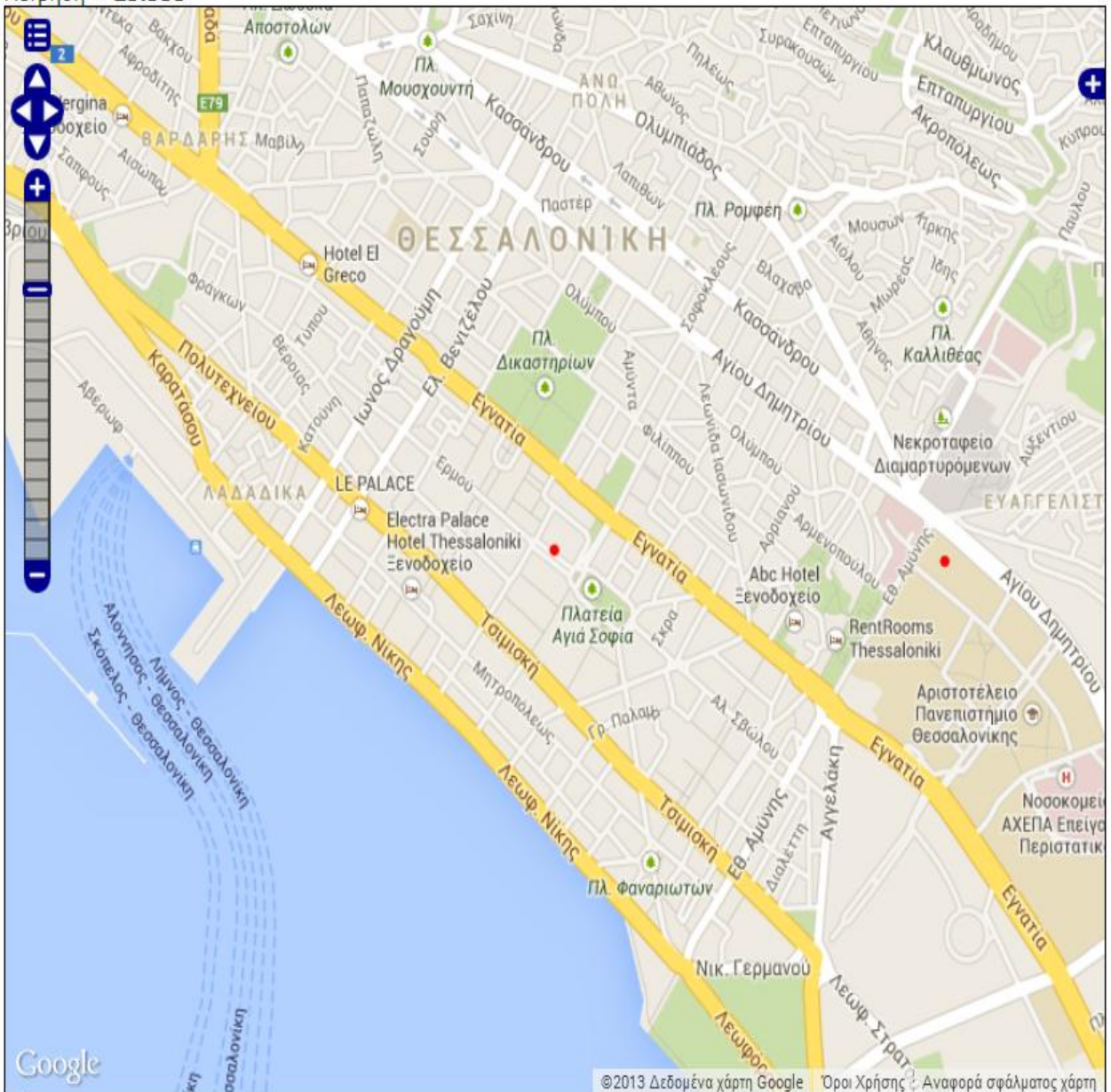
λ,φ : 23.65260, 37.94197

**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.30	30	ΠΕΙΡΑΙΑΣ_II	25	Αστικός-Υποβάθρου	BIO

**Εικόνα 23. Εκτέλεση ερωτήματος για το σταθμό 2 του Πειραιά**

Σταθμός Μέτρησης ΠΛ\_ΑΓΙΑΣ\_ΣΟΦΙΑΣ ▾ Ρύπος Διοξειδίο του Αζώτου ( NO2 ) ▾ Ημέρα 10 ▾ Μήνας 10 ▾ Έτος 2010  
 Μέτρηση = 25.333



Κλίμακα ~ 1 : 17K

λ,φ : 22.94512, 40.63383

**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.1	1	ΠΛ_ΑΓΙΑΣ_ΣΟΦΙΑΣ	27	Αστικός-Κυκλοφορίας	AGS

**Εικόνα 24. Εκτέλεση ερωτήματος για το σταθμό Αγίας Σοφίας στη Θεσσαλονίκη**

Σταθμός Μέτρησης ΠΑΤΡΑ-1 ▾ Ρύπος Όζον (O3) ▾ Ημέρα 3 ▾ Μήνας 3 ▾ Έτος 2003 ▾  
 Μέτρηση = 29.750



Κλίμακα ~ 1 : 17K

λ,φ : 21.73869, 38.25319

**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.88	ΠΑΤΡΑ-1	16	Αστικός-Κυκλοφορίας	ΡΑΙ	

**Εικόνα 25.** Εκτέλεση ερωτήματος για το σταθμό 1 της Πάτρας



Σταθμός Μέτρησης  Ρύπος  Ημέρα  Μήνας  Έτος   
 Μέτρηση = **6.125**



Κλίμακα ~ 1 : 17K

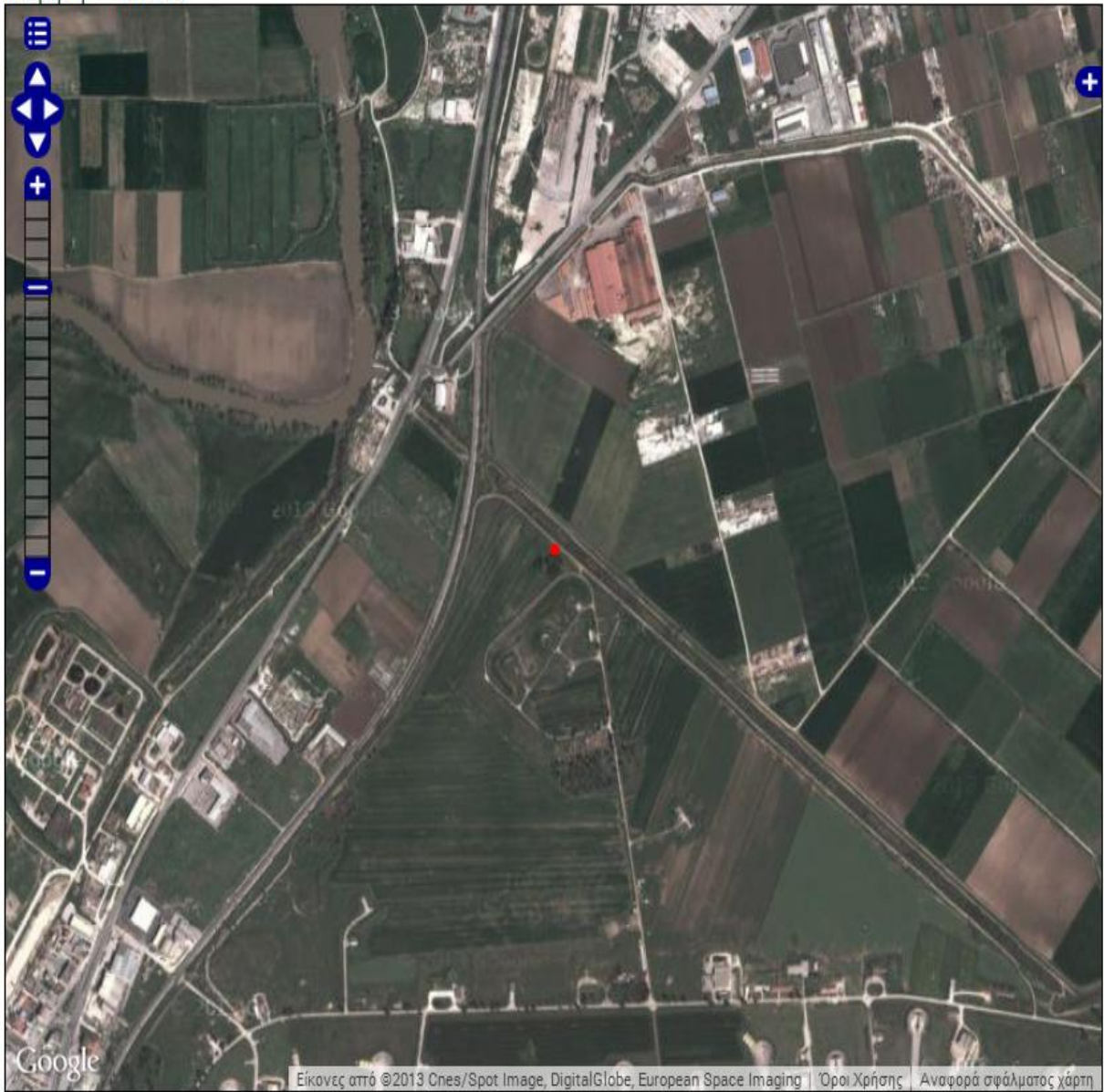
λ,φ : 22.94302, 39.36635

**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.10	10	ΒΟΛΟΣ	31	Αστικός-Υποβάθρου	VOL

**Εικόνα 26. Εκτέλεση ερωτήματος για το σταθμό του Βόλου**

Σταθμός Μέτρησης  Ρύπος  Ημέρα  Μήνας  Έτος   
 Μέτρηση = **16.292**



Κλίμακα ~ 1 : 17K

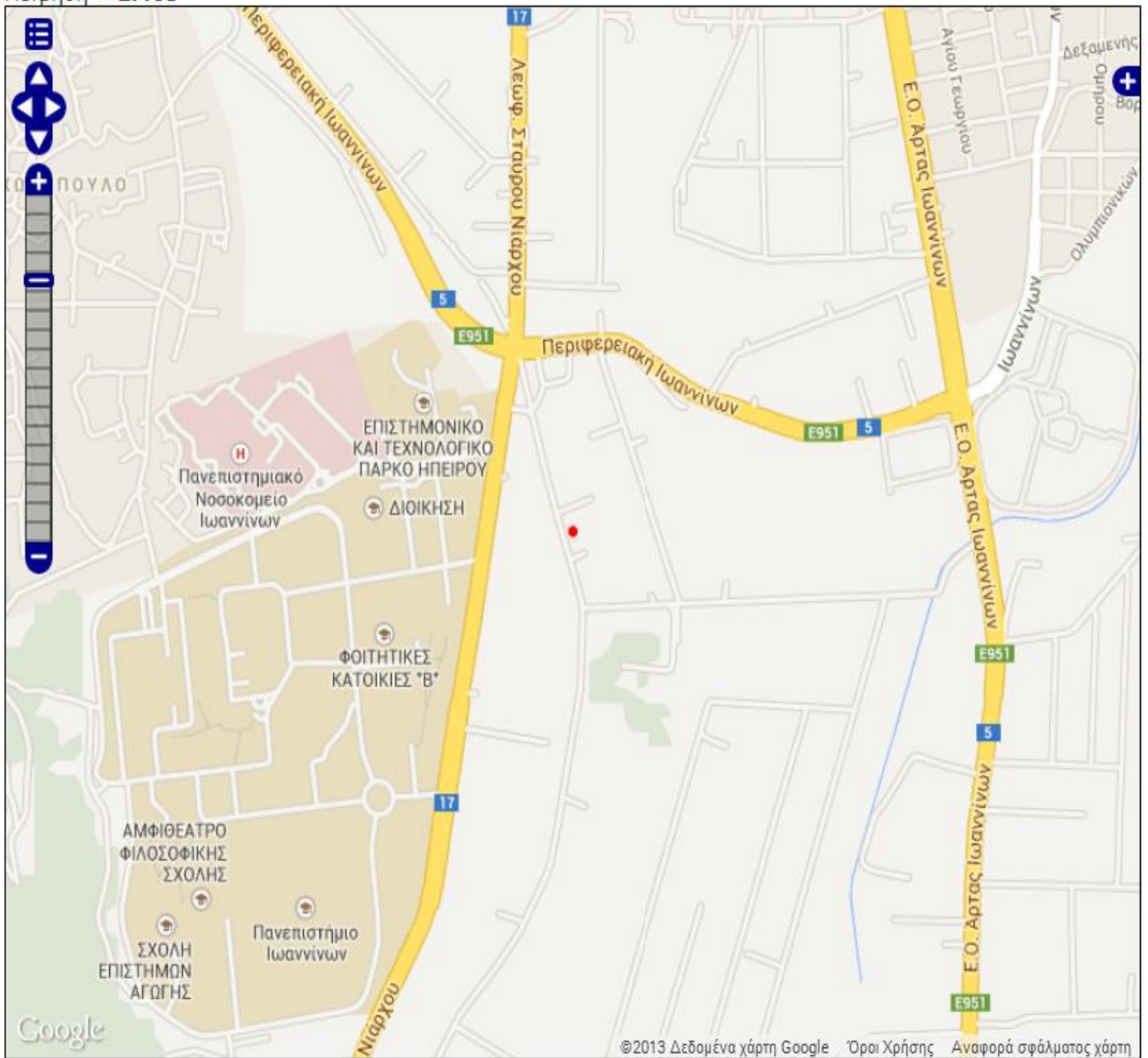
λ,φ : 22.45325, 39.66750

**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.11	11	ΛΑΡΙΣΑ	15	Αστικός-Κυκλοφορίας	LAR

**Εικόνα 27. Εκτέλεση ερωτήματος για το σταθμό της Λάρισας**

Σταθμός Μέτρησης  Ρύπος  Ημέρα  Μήνας  Έτος   
 Μέτρηση = 2.465



Κλίμακα ~ 1 : 17K

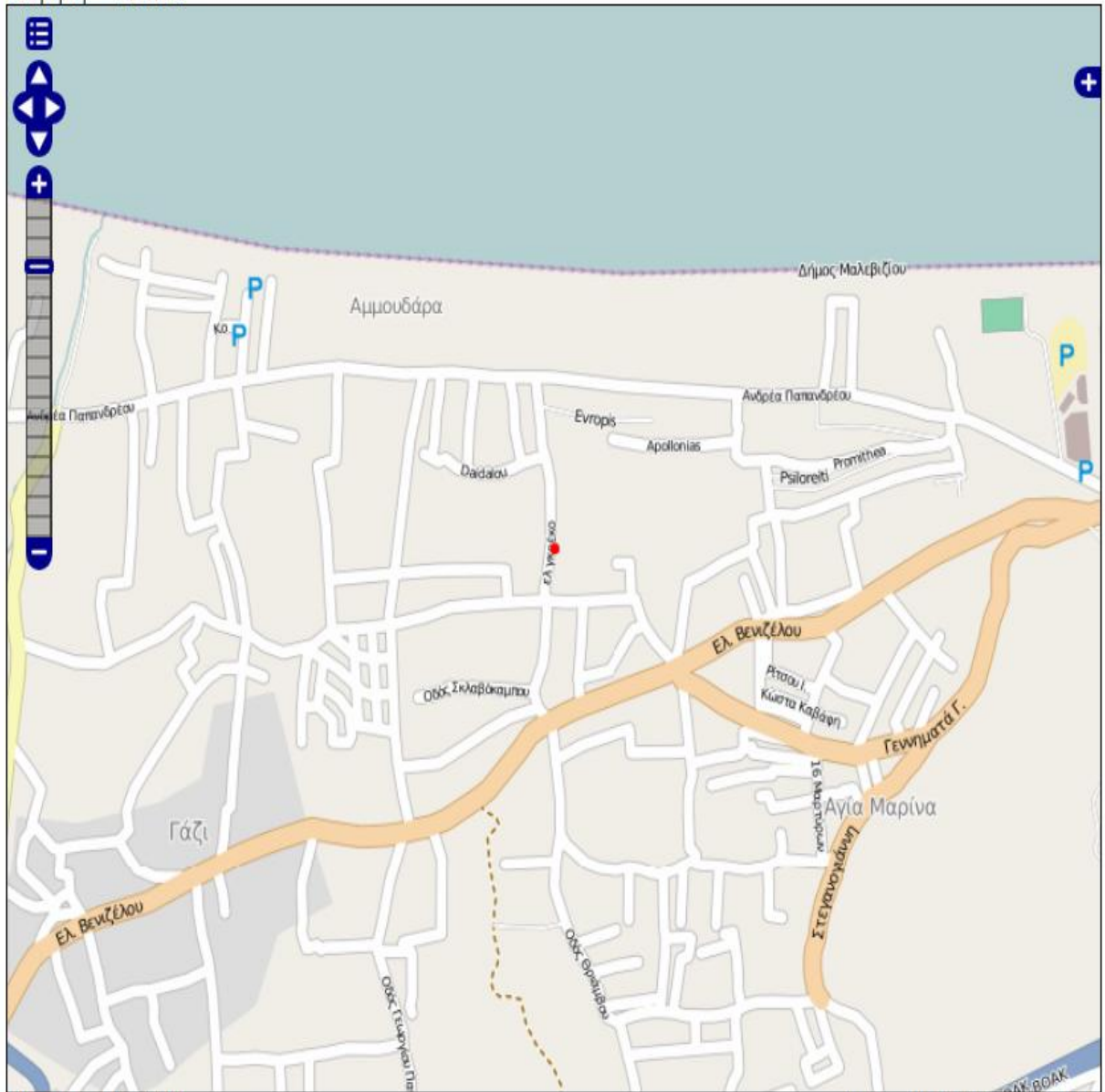
λ,φ : 20.84992, 39.61996

**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.13	13	ΙΩΑΝΝΙΝΑ	485	Αστικός-Υποβάθρου IOA	

Εικόνα 28. Εκτέλεση ερωτήματος για το σταθμό των Ιωαννίνων

Σταθμός Μέτρησης  Ρύπος  Ημέρα  Μήνας  Έτος   
 Μέτρηση = **71.625**



Κλίμακα ~ 1 : 17K

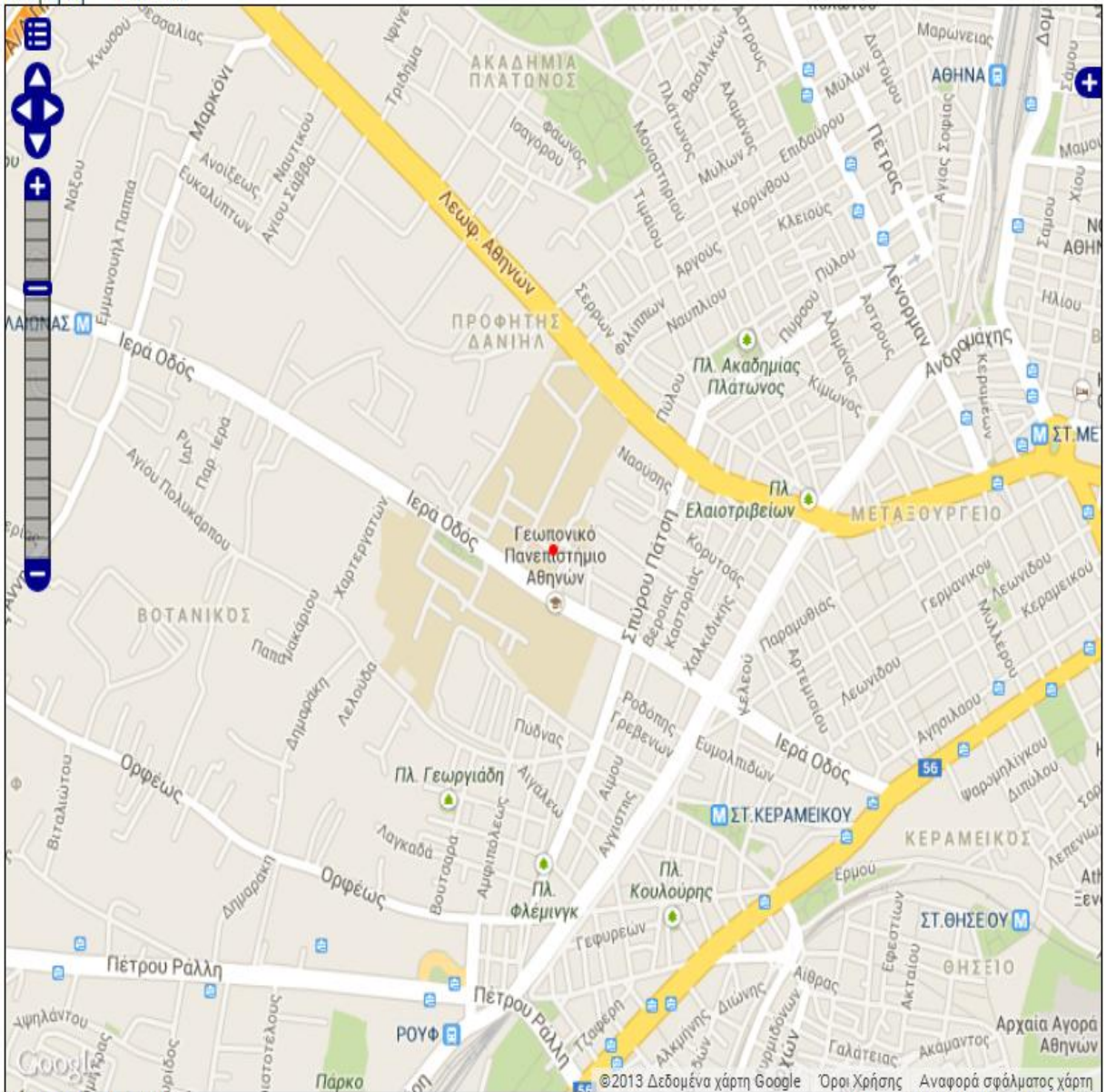
λ,φ : 25.08005, 35.33253

**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.12	12	ΗΡΑΚΛΕΙΟ	10	Αστικός-Κυκλοφορίας	IRA

**Εικόνα 29. Εκτέλεση ερωτήματος για το σταθμό Ηρακλείου Κρήτης**

Σταθμός Μέτρησης  Ρύπος  Ημέρα  Μήνας  Έτος   
 Μέτρηση = 18.250



Κλίμακα ~ 1 : 17K

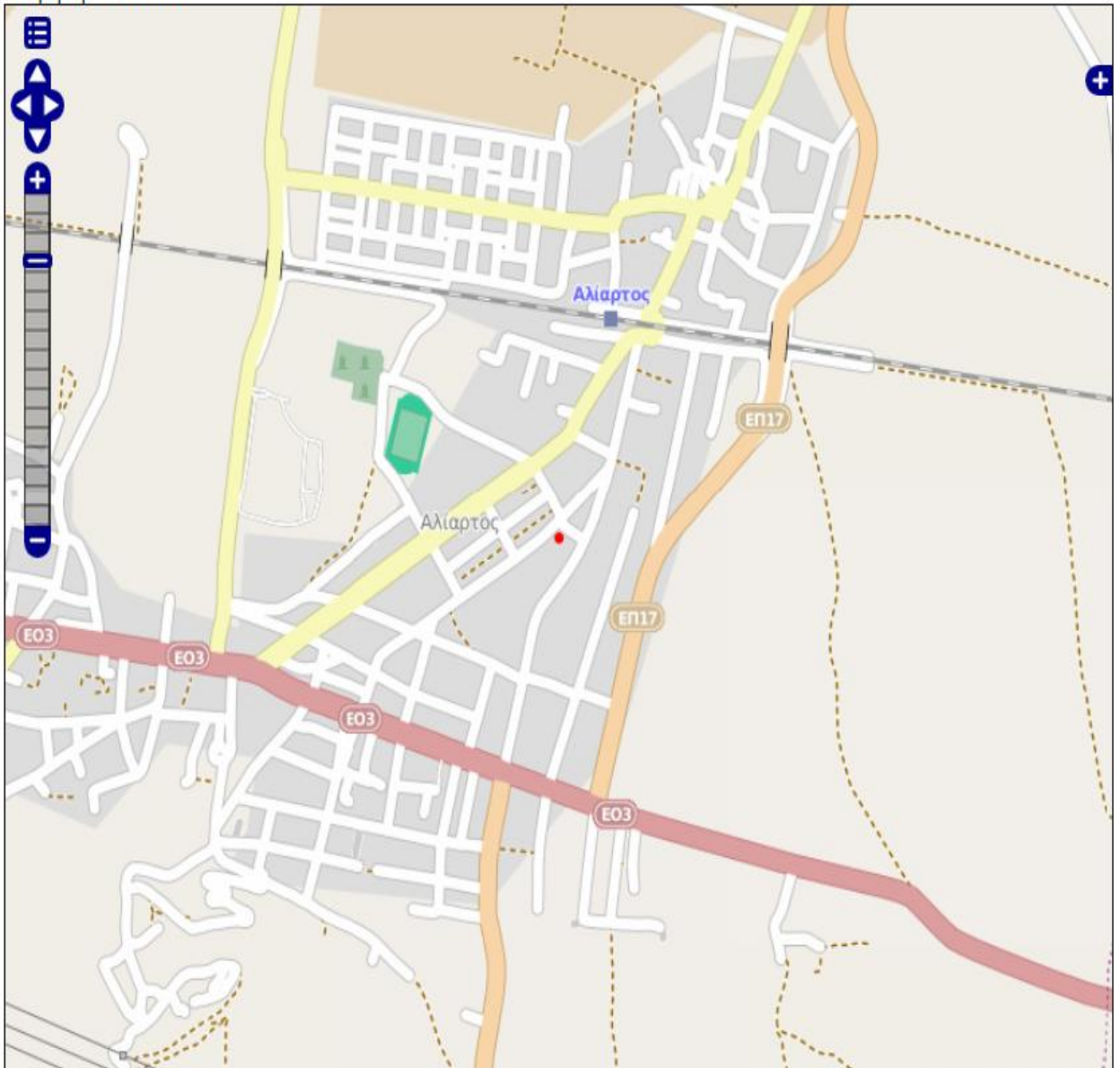
λ,φ : 23.70675, 37.98362

**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.16	16	ΓΕΩΠΟΝΙΚΗ	40	Περιστικός-Βιομηχανικός	GEO

**Εικόνα 30. Εκτέλεση ερωτήματος για το σταθμό του Γεωπονικού Πανεπιστημίου**

Σταθμός Μέτρησης  Ρύπος  Ημέρα  Μήνας  Έτος   
 Μέτρηση = **12.667**



Κλίμακα ~ 1 : 17K

λ,φ : 23.11023, 38.37528

**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.33	33	ΑΛΙΑΡΤΟΣ	110	Υποβάθρου	ALI

**Εικόνα 31. Εκτέλεση ερωτήματος για το σταθμό του Αλιάρτου**

Σταθμός Μέτρησης  Ρύπος  Ημέρα  Μήνας  Έτος   
 Μέτρηση = 34.583



Κλίμακα ~ 1 : 17K

λ,φ : 23.53845, 38.05141

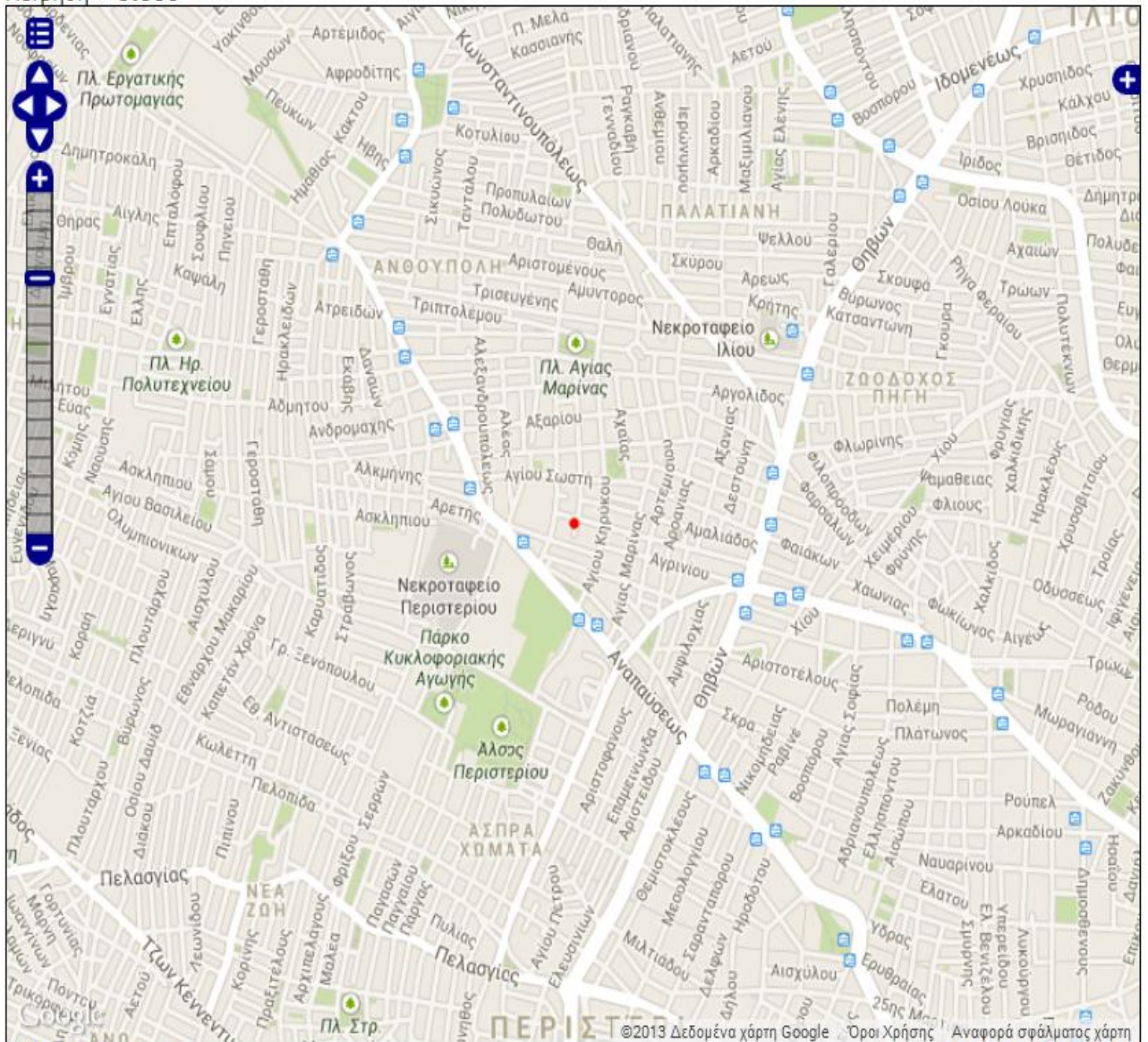
**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.25	25	ΕΛΕΥΣΙΝΑ	20	Περιστικός-Βιομηχανικός	ELE

**Εικόνα 32. Εκτέλεση ερωτήματος για το σταθμό της Ελευσίνας**

Σταθμός Μέτρησης  Ρύπος  Ημέρα  Μήνας  Έτος

Μέτρηση = **9.000**



**stathmoi**

fid	sta_id	sta_name	sta_h	sta_xar	sta_kwd
stathmoi.23	23	ΠΕΡΙΣΤΕΡΙ	80	Αστικός-Υποβάθρου	PER

**Εικόνα 33. Εκτέλεση ερωτήματος για το σταθμό Περιστερίου Αττικής**



## 5. Η αρχιτεκτονική της εφαρμογής

### 5.1 Λογισμικά – τεχνολογίες που χρησιμοποιήθηκαν

Όλα τα λογισμικά που χρησιμοποιήθηκαν για την εκπόνηση της συγκεκριμένης εργασίας, είναι Ελεύθερα Λογισμικά / Λογισμικά Ανοιχτού Κώδικα. Πρόκειται δηλαδή για λογισμικά τα οποία μπορούν να χρησιμοποιηθούν, αντιγραφούν, μελετηθούν, τροποποιηθούν και αναδιανεμηθούν χωρίς περιορισμό, σύμφωνα με τα όσα ορίζει το το Ίδρυμα Ελευθέρου Λογισμικού (Free Software Foundation, FSB). Παρακάτω ακολουθεί η περιγραφή και τα χαρακτηριστικά τους.

#### 5.1.1 PostgreSQL

Η PostgreSQL αποτελεί ένα ανοιχτού κώδικα Σύστημα Διαχείρισης αντικειμενο-σχεσιακής Βάσης Δεδομένων (ΣΔΒΔ) με πολλές και συνεχώς αυξανόμενες δυνατότητες. Η ανάπτυξη του ήδη διαρκεί πάνω από 20 χρόνια και βασίζεται σε μια αποδεδειγμένα καλή αρχιτεκτονική η οποία εξασφαλίζει την αξιοπιστία, την ακεραιότητα δεδομένων και την ορθή λειτουργία του. Τρέχει σε όλα τα βασικά λειτουργικά συστήματα, όπως Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), και Windows. Επίσης, είναι ACID συμβατό (ACID compliant), δηλαδή συμμορφώνεται με το παρακάτω σύνολο ιδιοτήτων:

- **Atomicity / Ατομικότητα** : Η Ατομικότητα απαιτεί η τροποποίηση που θα γίνει στην ΒΔ να τηρεί τον κανόνα όλα ή τίποτα. Κάθε ατομική συναλλαγή έχει το χαρακτηριστικό πως αν ένα μέρος της αποτύχει, αποτυγχάνει όλη η συναλλαγή και η ΒΔ μένει όπως ήταν πριν εκτελεστεί η συναλλαγή. Είναι πολύ κρίσιμο το ΣΔΒΔ να διατηρεί την ατομικότητα κάθε συναλλαγής ανεξάρτητα από το είδος της εφαρμογής, του ΣΔΒΔ, του λειτουργικού συστήματος ή αστοχίες του υλικού. Μια ατομική συναλλαγή δεν μπορεί να υποδιαιρεθεί, και πρέπει να επεξεργάζεται ολόκληρη ή καθόλου. Έτσι, οι χρήστες είναι εξαλείφουν την πιθανότητα μη ολοκληρωμένων ή ημιτελών συναλλαγών.
- **Consistency / Συνέπεια** : Η ιδιότητα της Συνέπειας διασφαλίζει ότι η ΒΔ διατηρείται σε μια συνεπή κατάσταση, δηλαδή ότι κάθε συναλλαγή θα οδηγεί την βάση δεδομένων από την μια συνεπή κατάσταση στην άλλη. Για παράδειγμα αν ένα πεδίο ενός πίνακα είναι μόνο για ακέραιους αριθμούς τότε το ΣΔΒΔ μπορεί είτε να απορρίψει απόπειρες για είσοδο δεκαδικών αριθμών είτε να τους στρογγυλοποιήσει. Και οι δυο αυτές ενέργειες διατηρούν την συνέπεια.
- **Isolation / Απομόνωση** : Η Απομόνωση αναφέρεται στην απαίτηση ότι καμία ενέργεια δεν μπορεί να γίνει σε δεδομένα τα οποία τροποποιούνται εκείνη την στιγμή από μια συναλλαγή η οποία δεν έχει ακόμα ολοκληρωθεί.
- **Durability / Μονιμότητα** : Η Μονιμότητα εγγυάται στον χρήστη το ΣΔΒΔ ότι αν τελειώσει μια συναλλαγή επιτυχώς τότε τα αποτελέσματα της δεν θα χαθούν. Οι αλλαγές που έχει κάνει η συναλλαγή δεν θα χαθούν ακόμα και αν κρσάρει το σύστημα.

Η PostgreSQL έχει ολοκληρωμένη υποστήριξη για foreign keys, joins, views, triggers, καθώς και για stored procedures σε διάφορες γλώσσες προγραμματισμού. Συμπεριλαμβάνει τους περισσότερους τύπους δεδομένων SQL:2008, συμπεριλαμβανομένων INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, και TIMESTAMP. Επίσης, υποστηρίζει αποθήκευση μεγάλων αντικειμένων (binary objects), όπως εικόνες, ήχοι ή video. Διαθέτει ακόμα περιβάλλοντα προγραμματισμού (APIs) για περισσότερες από δώδεκα γλώσσες προγραμματισμού όπως C, C++, Java, Perl, Python, Ruby, Tcl και άλλες, καθώς και υποστήριξη για την πλατφόρμα .NET και το πρότυπο ODBC.

Επιπρόσθετα, η PostgreSQL υλοποιεί εξεζητημένα χαρακτηριστικά όπως Multi-Version Concurrency Control (MVCC), point in time recovery, tablespaces, asynchronous replication, Δημιουργία διαδικτυακής εφαρμογής με χρήση Web Services

nested transactions (savepoints), online/hot backups, a sophisticated query planner/optimizer, write ahead logging for fault tolerance. Υποστηρίζει επίσης διεθνή σετ χαρακτήρων, Unicode, κωδικοποίηση χαρακτήρων σε πολλά bytes καθώς και δυνατότητα ταξινόμησης δεδομένων ανεξάρτητα από το locale. Έχει ολοκληρωμένη υποστήριξη για subqueries (συμπεριλαμβανομένων subselects μέσα από το FROM), read-committed και serializable transaction isolation levels.

Άλλα προηγμένα χαρακτηριστικά της PostgreSQL είναι τα table inheritance, rules systems και database events. Το table inheritance (κληρονομικότητα πινάκων) προσθέτει μια αντικειμενοστραφής διάσταση στην δημιουργία πινάκων, επιτρέποντας στους σχεδιαστές βάσεων δεδομένων να δημιουργούν νέους πίνακες από άλλους χρησιμοποιώντας τους ως βάση. Το rules system, (το οποίο επίσης καλείται the query rewrite system), επιτρέπει στον σχεδιαστή να δημιουργήσει κανόνες που ορίζουν συγκεκριμένες λειτουργίες για έναν πίνακα ή για ένα view. Το events system αποτελεί ένα interprocess communication system στο οποίο μηνύματα και events μπορούν να μεταδοθούν μεταξύ πελατών (clients) χρησιμοποιώντας τις LISTEN και NOTIFY εντολές, επιτρέποντας από την απλή peer to peer επικοινωνία ως ένα εξελιγμένο συντονισμό βασισμένο σε database events. Εφόσον τα notifications μπορεί να προέρχονται από triggers και stored procedures, οι clients μπορούν να επιβλέπουν λειτουργίες όπως: update, insert ή delete πινάκων όταν αυτά γίνονται.

Στην βασική βιβλιοθήκη συναρτήσεων της PostgreSQL συμπεριλαμβάνονται εκατοντάδες built-in συναρτήσεις οι οποίες καλύπτουν από βασικές μαθηματικές συναρτήσεις και διαχείριση συμβολοσειρών έως κρυπτογραφία. Triggers και stored procedures μπορούν να γράφουν σε c και να φορτωθούν μέσα στην βάση δεδομένων ως βιβλιοθήκη, επιτρέποντας μεγάλη ευελιξία στην επέκταση των δυνατοτήτων της βάσης. Παρομοίως η PostgreSQL περιλαμβάνει framework που επιτρέπει τον ορισμό και την δημιουργία custom data types καθώς και βοηθητικές συναρτήσεις και τελεστές (operators) που θα περιγράψουν την λειτουργία τους.

Η PostgreSQL αποτελεί ένα πλήρες σχεσιακό σύστημα που υποστηρίζει πολλαπλά σχήματα ανά database, ενώ ο κατάλογος (πληροφορίες σχετικά με τους πίνακες, στήλες, views, κ.α.) είναι διαθέσιμος μέσω του Information Schema, όπως ορίζεται στο SQL standard. Όσον αφορά στα data integrity χαρακτηριστικά, συμπεριλαμβάνονται primary keys, foreign keys με υποστήριξη restricting και cascading updates/deletes, check constraints, unique constraints, και not null constraints. Επίσης, έχει αρκετά προηγμένα χαρακτηριστικά όπως auto-increment columns μέσω sequences και LIMIT/OFFSET που επιτρέπουν την επιστροφή partial result sets. Όσον αφορά τα indexes υποστηρίζει compound, unique, partial, και functional indexes τα οποία μπορούν να χρησιμοποιήσουν οποιοδήποτε από τους B-tree, R-tree, hash, ή Gist αλγόριθμους.

Το GiST (Generalized Search Tree) indexing αποτελεί ένα προηγμένο σύστημα το οποίο συνδυάζει ένα μεγάλο εύρος από διαφορετικούς αλγόριθμους ταξινόμησης και αναζήτησης όπως B-tree, B+-tree, R-tree, partial sum trees, ranked B+-trees και αρκετούς ακόμα. Επίσης, διαθέτει interface το οποίο επιτρέπει τόσο την δημιουργία custom data types όσο και επεκτάσιμους τρόπους ερωτήσεων (query) για την αναζήτηση τους. Πάνω στο GiST και την PostgreSQL έχει στηριχθεί και η επέκταση PostGIS.

Το PostGIS είναι μία επέκταση της PostgreSQL για να υποστηρίζει χωρικά δεδομένα, σύμφωνα με το πρότυπο του OGC (Open Geospatial Consortium). Παρέχει ειδικούς τελεστές για τη σύνταξη ερωτημάτων, λειτουργίες συνάθροισης επάνω σε χωρικά δεδομένα καθώς και χωρικές συναρτήσεις. Επιτρέπει επίσης την ανάθεση προβολικών συστημάτων στα χωρικά δεδομένα. Τέλος, μπορεί να χρησιμοποιηθεί για να οπτικοποιηθούν τα δεδομένα μέσω ειδικών εφαρμογών όπως το QuantumGIS και ο GeoServer.

Η PostgreSQL μπορεί να διαχειριστεί εύκολα μεγάλους αριθμούς ταυτόχρονων χρηστών καθώς και μεγάλο όγκο δεδομένων. Υπάρχουν άλλωστε ενεργές εγκαταστάσεις σε περιβάλλοντα παραγωγής που διαχειρίζονται πάνω από 4 terabytes δεδομένων. Μερικές γενικές οριακές τιμές συμπεριλαμβάνονται στον παρακάτω πίνακα:

Limit	Value
Maximum Database Size	Unlimited
Maximum Table Size	32 TB
Maximum Row Size	1.6 TB
Maximum Field Size	1 GB
Maximum Rows per Table	Unlimited
Maximum Columns per Table	250 - 1600 depending on column types
Maximum Indexes per Table	Unlimited

Τέλος, ο πηγαίος κώδικας της PostgreSQL είναι διαθέσιμος κάτω από την ελεύθερη open source άδεια, BSD license. Αυτή η άδεια δίνει την δυνατότητα χρήσης, μετατροπής και διανομής της PostgreSQL σε οποιαδήποτε μορφή, ανοιχτού ή κλειστού κώδικα. Η PostgreSQL δεν είναι μόνο μια δυνατή βάση δεδομένων ικανή να τρέχει μέσα σε επιχειρήσεις, είναι μια πλατφόρμα ανάπτυξης πάνω στην οποία δύνανται να γίνει ανάπτυξη in-house, web ή εμπορικών εφαρμογών τα οποία χρειάζονται RDBMS.

### 5.1.2 GeoServer

Ο GeoServer είναι ένας open source Web Server που επιτρέπει στους χρήστες του να διαχειρίζονται και να μοιράζονται γεωχωρικά δεδομένα. Είναι γραμμένος στη γλώσσα προγραμματισμού Java και από το 2001 που ξεκίνησε το Geoserver Project αναπτύσσεται συνεχώς από μια ευρεία κοινότητα χρηστών και προγραμματιστών σε όλο τον κόσμο. Συμμορφώνεται πλήρως με τα πρότυπα του Open Geospatial Consortium (OGC) και προσφέρει υπηρεσίες σε χωρική πληροφορία χρησιμοποιώντας τα open standards.

Ο GeoServer χρησιμοποιείται για να αξιοποιηθούν δεδομένα στα οποία υπάρχει χωρική πληροφορία (vector ή raster) και υποστηρίζει μεγάλο πλήθος μορφών όπως:

- PostGIS
- Oracle Spatial
- ArcSDE
- DB2
- MySQL
- Shapefiles
- GeoTIFF
- GTOPO30
- ECW, MrSID
- JPEG2000

Στα χωρικά δεδομένα ο GeoServer προσφέρει υπηρεσίες όπως Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS) και άλλες. Υποστηρίζει ένα τεράστιο πλήθος από συστήματα αναφοράς, τόσο παγκόσμια όσο και τοπικά (γεωκεντρικά και γεωδαιτικά), ενώ επιτρέπει on-the-fly transformation από ένα σύστημα σε άλλο. Η υπηρεσία

WMS αφορά τη θέαση (προβολή) των χωρικών δεδομένων σε χάρτη και αυτό μπορεί να γίνει σε οποιαδήποτε από τις παρακάτω μορφές:

- AtomPub
- GIF
- GeoRSS
- GeoTiff / GeoTiff 8-bits
- JPEG
- KML compressed / KML network link / KML plain
- OpenLayers
- PDF
- PNG / PNG 8-bit
- SVG
- Tiff / Tiff 8-bits

Αντίστοιχα, η υπηρεσία WFS υποστηρίζει αναζήτηση και ανάκτηση των δεδομένων, με γεωγραφικά ή μη κριτήρια. Επίσης, ο GeoServer υλοποιεί και την υπηρεσία transactional Web Feature Service (WFS-T) η οποία επιτρέπει επίσης την δημιουργία, διαγραφή και τροποποίηση δεδομένων. Η υπηρεσία υλοποιείται στις παρακάτω μορφές:

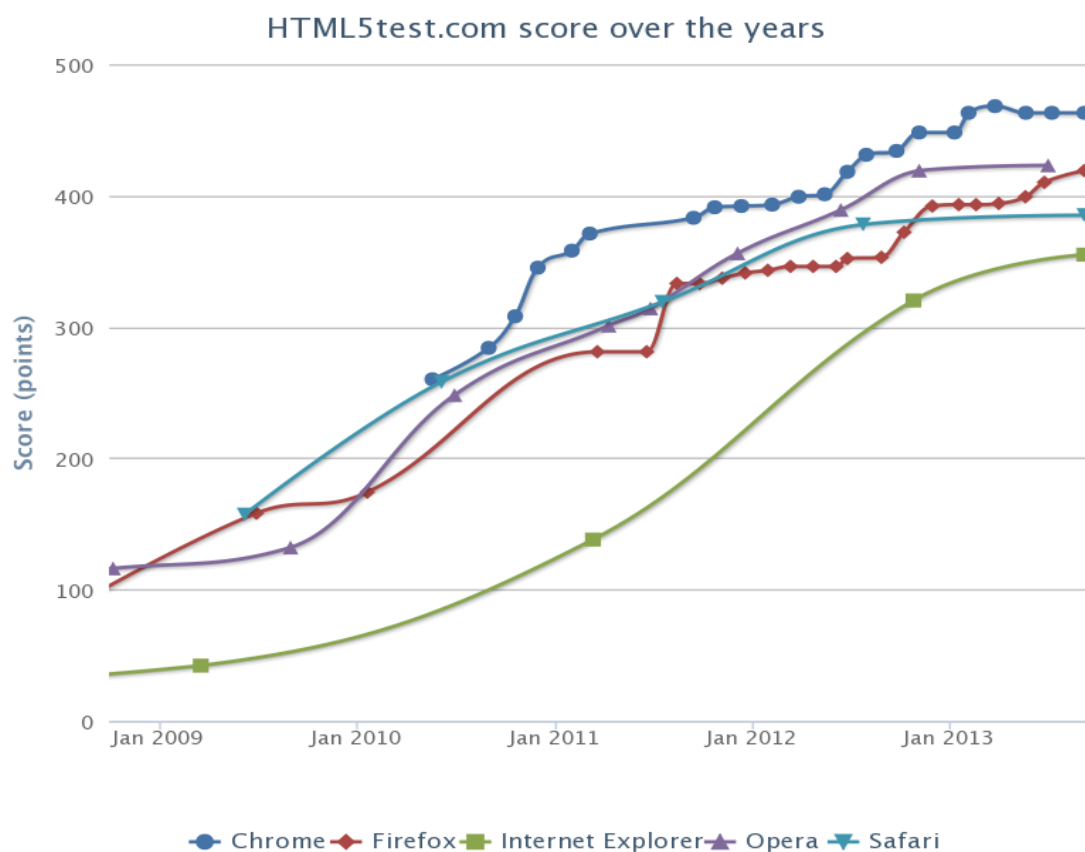
- CSV
- GML2 / GML2-GZIP / GML3.1 /GML3.2
- GeoJSON
- Shapefile
- application/gml+xml

Ο GeoServer είναι το λογισμικό που δέχεται τα αιτήματα του χρήστη της εφαρμογής και αποκρίνεται αναλόγως με τα περιεχόμενα της βάσης δεδομένων.

### **5.1.3 HTML5 – JavaScript – AJAX**

Η HTML (HyperText Markup Language) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες και η HTML5 είναι η εξέλιξη της, η οποία αναμένεται στο μέλλον να την αντικαταστήσει πλήρως. Γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες, (οι οποίες συνήθως λειτουργούν ανά ζεύγη για παράδειγμα <html> </html>), μέσα στο περιεχόμενο της ιστοσελίδας. Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ. Οι ιστοσελίδες ουσιαστικά είναι \*.html αρχεία, τα οποία οι browser τα διαβάζουν και ανάλογα μορφοποιούν την ιστοσελίδα. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει και να μορφοποιήσει το περιεχόμενο της σελίδας.

Η HTML5 αποτελεί το μελλοντικό πρότυπο στο σχεδιασμό των ιστοσελίδων όπως σχεδιάζεται και υλοποιείται από το World Wide Web Consortium (W3C) και το Web Hypertext Application Technology Working Group (WHATWG). Αποτελεί την 5η κατά σειρά βελτίωση της HTML και προσφέρει πολλά πλεονεκτήματα στους σχεδιαστές, όπως η μείωση της ανάγκης για plug-in και η υποστήριξη multimedia αντικειμένων. Ακόμα δεν έχει υιοθετηθεί πλήρως ως πρότυπη γλώσσα γραφής ιστοσελίδων (ακόμα χρησιμοποιείται η HTML 4.01), αλλά ήδη υποστηρίζεται από τους περισσότερους browser αφού ουσιαστικά η HTML5 αποτελεί υπερσύνολο της HTML. Στην Εικόνα 34 φαίνεται η υποστήριξη HTML5 στους πιο συνηθισμένους browser διαχρονικά.



**Εικόνα 34. Διαχρονικά η υποστήριξη HTML5 από τους κυριότερους browser**

Η HTML5 μαζί με τη JavaScript δίνει στους προγραμματιστές τη δυνατότητα να σχεδιάζουν και να αναπτύσσουν εφαρμογές οι οποίες τρέχουν σε όλες τις συσκευές, υπολογιστές, tablets ή Smartphones, ανεξαρτήτως μεγέθους και τύπου οθόνης ή Λειτουργικού Συστήματος.

Η JavaScript (JS) είναι διερμηνευόμενη (interpreted) γλώσσα προγραμματισμού για διαδικτυακές εφαρμογές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται. Πρόκειται για γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης.

Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστραφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού. Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι browsers (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets).

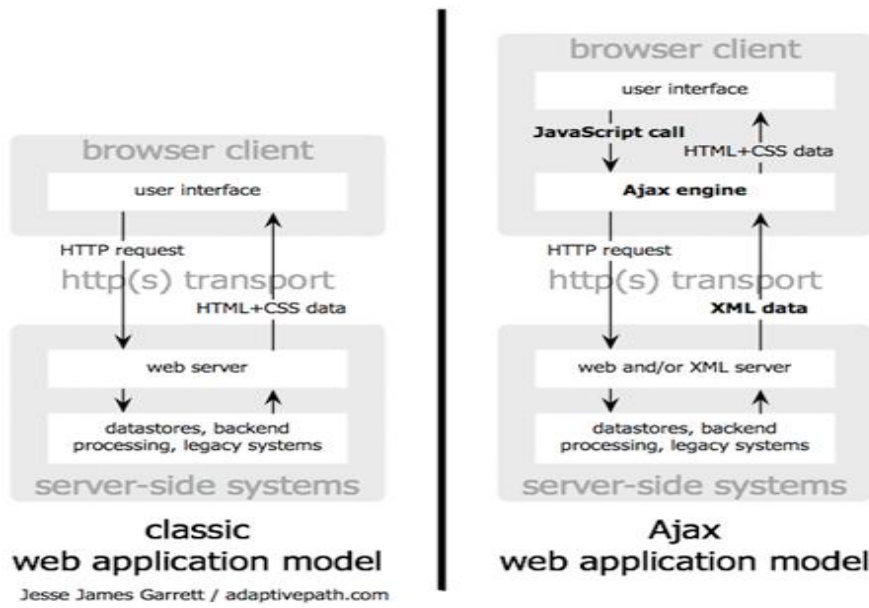
Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), δηλαδή το browser του χρήστη, και χαρακτηρίστηκε σαν client-side γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα JavaScript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή (server), αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού). Η χρήση της JavaScript στο διακομιστή εμφανίζεται πάλι σήμερα, με τη διάδοση του Node.js, ενός μοντέλου προγραμματισμού βασισμένο στα γεγονότα (events).

Η JavaScript δημιουργήθηκε από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java (καθώς και οι δύο είναι αντικειμενοστραφείς γλώσσες προγραμματισμού). LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά, με το πρόγραμμα περιήγησης Netscape Navigator 2.0, τον Σεπτέμβριο του 1995. Από LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου 1995, όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape έκδοσή 2.0B3. Έτσι, απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης. Η εταιρεία Microsoft ονόμασε την εφαρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων και την συμπεριέλαβε στο πρόγραμμα Internet Explorer 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996.

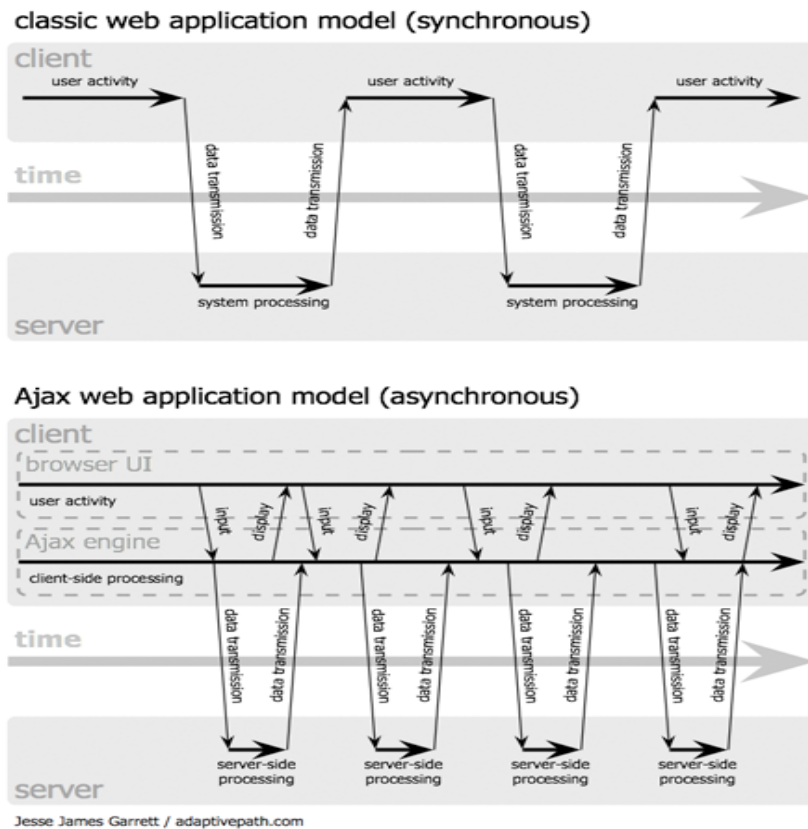
Η JavaScript είναι πλέον μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον παγκόσμιο ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές. Με τη χρήση της τεχνολογίας AJAX, η γλώσσα επέστρεψε στο προσκήνιο και έδωσε πολύ περισσότερες δυνατότητες στους προγραμματιστές.

Με τη τεχνική AJAX (Asynchronous JavaScript And Xml) οι διαδικτυακές εφαρμογές έχουν τη δυνατότητα να στείλουν ή να ανακτήσουν δεδομένα από το server στο παρασκήνιο χωρίς καμία παρέμβαση στην εμφάνιση ή τη συμπεριφορά της ιστοσελίδας. Αυτό συμβαίνει μέσω των αντικειμένων XMLHttpRequest και το μεγάλο πλεονέκτημα αυτής της τεχνικής είναι πως ελαχιστοποιεί τις περιπτώσεις που η ιστοσελίδα κολλάει και δεν ανταποκρίνεται όσο περιμένει την απάντησή από το server.

Στις Εικόνες 35 – 36 (<http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>) φαίνεται η διαφορά στην αρχιτεκτονική μιας διαδικτυακής εφαρμογής με την τεχνολογία AJAX σε σχέση με το κλασικό μοντέλο, καθώς και η διαφορά μεταξύ synchronous και asynchronous αιτημάτων στο διακομιστή.



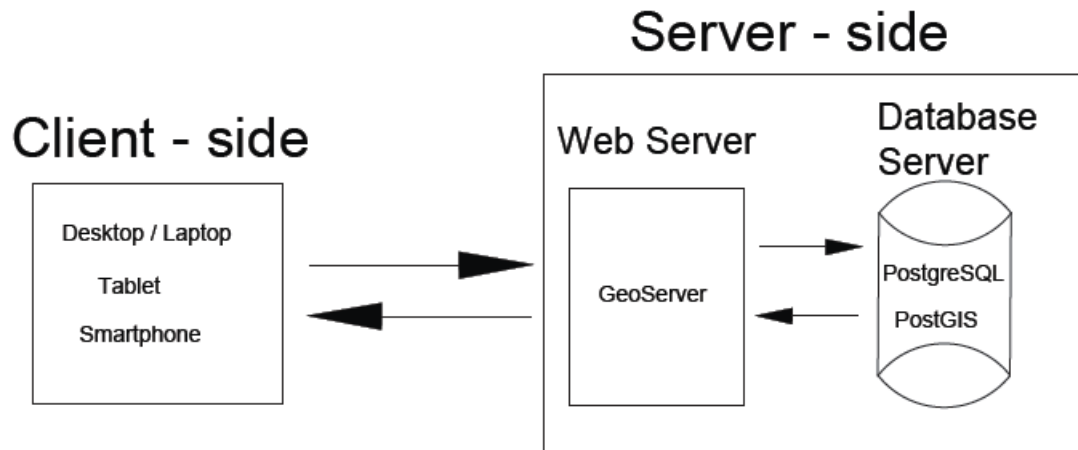
Εικόνα 35. Διαφορά στην αρχιτεκτονική κλασσικού – AJAX μοντέλου σε Web εφαρμογές



Εικόνα 36. Διαφορά στην ροή κλασσικού(synchronous) – AJAX (asynchronous) μοντέλου σε Web εφαρμογές

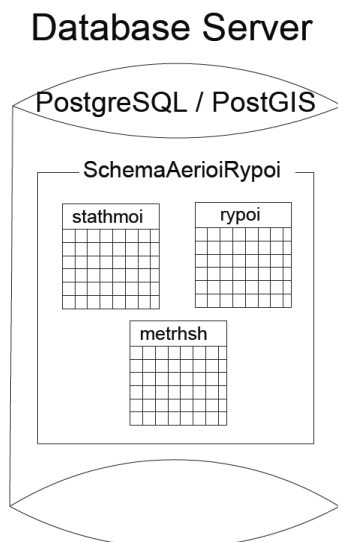
### 5.2 Υλοποίηση της εφαρμογής

Η υλοποίηση της εφαρμογής συνίσταται στο σχεδιασμό και κατασκευή της βάσης δεδομένων, στη διασύνδεση της με τον GeoServer, και στη συνέχεια τη δημιουργία της διεπαφής με το χρήστη (User Interface) καθώς και των αλγορίθμων που προσδίδουν τη λειτουργικότητα στην εφαρμογή. Στην Εικόνα 37 φαίνονται τα συστατικά μέρη της εφαρμογής.



Εικόνα 37. Αρχιτεκτονική της εφαρμογής

Η Βάση Δεδομένων αποτελείται από τρεις πίνακες που περιέχουν δεδομένα για τους σταθμούς, τους ρύπους και τις μετρήσεις, ενώ στον Geoserver περιέχονται τέσσερα αρχεία που αποτελούν τον κώδικα της εφαρμογής. Οι Εικόνες 38 και 39 δείχνουν σχηματικά τα παραπάνω.



Εικόνα 38. Η Βάση Δεδομένων της εφαρμογής



Εικόνα 39. Τα αρχεία στον GeoServer

Για την δημιουργία των πινάκων της Βάσης απαιτήθηκε επεξεργασία των πρωτογενών δεδομένων που παρέχονται από το Υ.ΠΕ.Κ.Α. Σε αυτό το στάδιο αρχικά χρησιμοποιήθηκε κώδικας Linux bash shell scripts ώστε να ομαδοποιηθούν τα 1510 \*.dat αρχεία που βρίσκονταν



κατανεμημένα σε 325 διαφορετικούς φακέλους. Αυτά τα αρχεία ήταν ομαδοποιημένα ως προς το σταθμό μέτρησης, το ρύπο μέτρησης και το έτος. Ο πίνακας Π1 του Παρατήματος παρουσιάζει τα αρχεία όπως λήφθηκαν από την ιστοσελίδα του υπουργείου.

Στη συνέχεια χρησιμοποιήθηκε κώδικας Java ούτως ώστε να ομαδοποιηθούν ανά σταθμό και ρύπο, συγχωνεύτηκαν δηλαδή τα αρχεία που αναφέρονταν στον ίδιο σταθμό και ρύπο ανεξάρτητα από το έτος που έγιναν οι μετρήσεις. Αυτό κατέστη δυνατόν χωρίς να χαθεί απαραίτητη πληροφορία αφού σε κάθε γραμμή αναφέρεται η ημερομηνία της μέτρησης, και έτσι το πλήθος των αρχείων δεδομένων έπεσε στον αριθμό 134. Ο Πίνακας Π2 του Παραρτήματος δείχνει την ομαδοποίηση ανά σταθμό και ο Πίνακας Π3 την ομαδοποίηση ανά ρύπο έτσι όπως προέκυψαν μετά την επεξεργασία των αρχείων με πρόγραμμα Java.

Κατόπιν, γράφτηκε άλλο πρόγραμμα Java με σκοπό να υπολογιστούν οι ημερήσιοι μέσοι όροι των μετρήσεων. Όπως έχει ήδη αναφερθεί, τα αρχεία περιέχουν 24 ωριαίες μετρήσεις για κάθε ημέρα, ενώ στις ώρες που δεν λειτουργούσε ο σταθμός – και άρα δεν υπάρχουν μετρήσεις – έχουν τιμή -9999.

Οι ορισμοί της Βάσης Δεδομένων ακολουθούν παρακάτω:

```
-- Database: "DbAerioiRypoi"
-- DROP DATABASE "DbAerioiRypoi";
CREATE DATABASE "DbAerioiRypoi"
  WITH OWNER = postgres
       ENCODING = 'UTF8'
       TABLESPACE = pg_default
       LC_COLLATE = 'Greek, Greece'
       LC_CTYPE = 'Greek, Greece'
       CONNECTION LIMIT = -1;

-- Schema: SchemaAerioiRypoi
-- DROP SCHEMA "SchemaAerioiRypoi";
CREATE SCHEMA "SchemaAerioiRypoi"
  AUTHORIZATION postgres;

-- Table: "SchemaAerioiRypoi".stathmoi
-- DROP TABLE "SchemaAerioiRypoi".stathmoi;
CREATE TABLE "SchemaAerioiRypoi".stathmoi
(
  sta_id integer NOT NULL,
  sta_name character varying(50),
  sta_geometry geometry,
  sta_h integer,
  sta_xar character varying(50),
  sta_kwd character varying(5),
  CONSTRAINT stathmoi_pkey PRIMARY KEY (sta_id )
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "SchemaAerioiRypoi".stathmoi
  OWNER TO postgres;
```

```

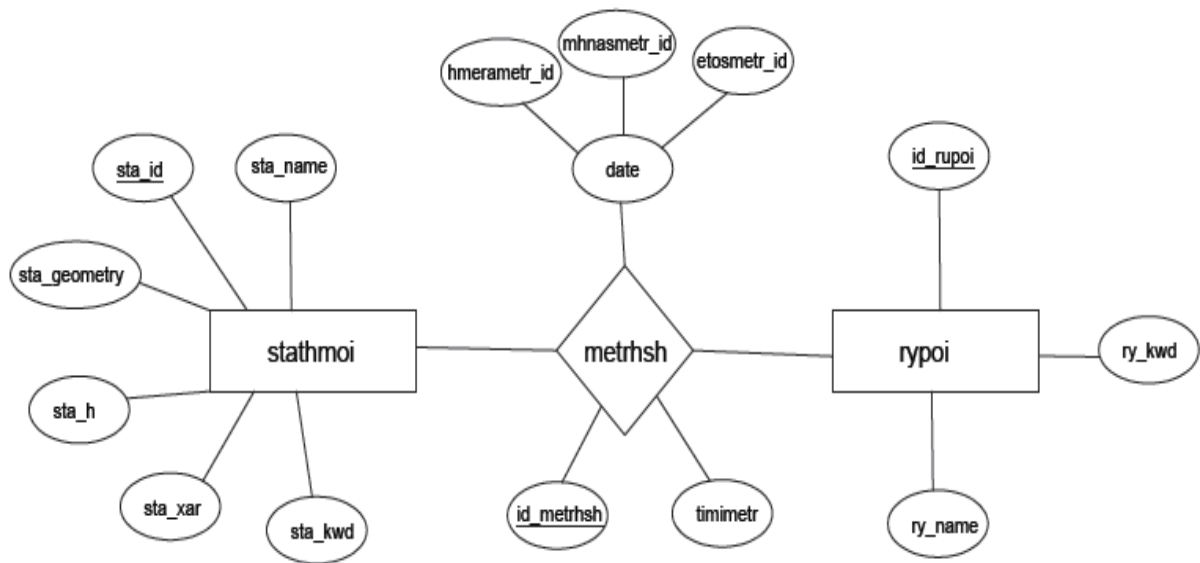
-- Table: "SchemaAerioiRypoi".rypoi
-- DROP TABLE "SchemaAerioiRypoi".rypoi;
CREATE TABLE "SchemaAerioiRypoi".rypoi
(
  id_rupoi integer NOT NULL,
  ry_kwd character varying(15) NOT NULL,
  ry_name character varying(45) NOT NULL,
  CONSTRAINT rypoi_pkey PRIMARY KEY (id_rupoi )
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "SchemaAerioiRypoi".rypoi
  OWNER TO postgres;

-- Table: "SchemaAerioiRypoi".metrhsh
-- DROP TABLE "SchemaAerioiRypoi".metrhsh;
CREATE TABLE "SchemaAerioiRypoi".metrhsh
(
  id_metrhsh integer NOT NULL,
  timimetr numeric(10,3) DEFAULT NULL::numeric,
  stamet_id integer NOT NULL,
  rymet_id integer NOT NULL,
  etosmetr_id integer NOT NULL,
  mhnametr_id integer NOT NULL,
  hmerametr_id integer NOT NULL,
  CONSTRAINT metrhsh_pkey PRIMARY KEY (id_metrhsh ),
  CONSTRAINT id_rupoi FOREIGN KEY (rymet_id)
    REFERENCES "SchemaAerioiRypoi".rypoi (id_rupoi) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT id_stathmoi FOREIGN KEY (stamet_id)
    REFERENCES "SchemaAerioiRypoi".stathmoi (sta_id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "SchemaAerioiRypoi".metrhsh
  OWNER TO postgres;

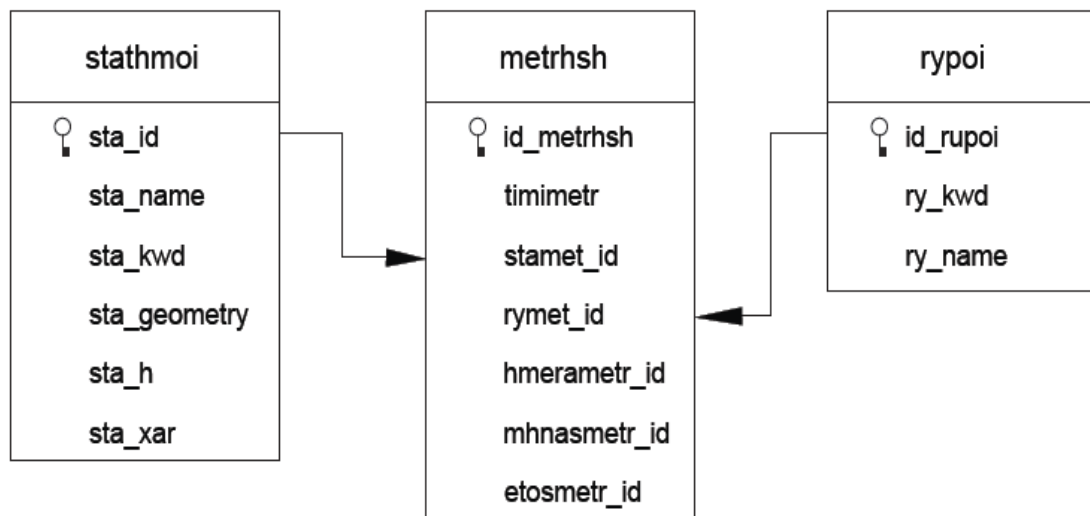
```

Όπως φαίνεται και από τους ορισμούς παραπάνω, δημιουργήθηκε μία χωρική βάση δεδομένων (με την επέκταση PostGIS στην PostgreSQL) η οποία περιέχει 3 πίνακες. Από αυτούς, μόνο ο πίνακας *stathmoi* περιέχει στήλη με χωρική πληροφορία, πρόκειται για τον τύπο *geometry* ο οποίος κρατάει τη θέση των σταθμών (φ,λ δηλαδή γεωγραφικό πλάτος και μήκος) στο σύστημα αναφοράς WGS84.

Παρακάτω, η Εικόνα 40 παρουσιάζει το Διάγραμμα Οντοτήτων – Συσχετίσεων της βάσης δεδομένων, ενώ η Εικόνα 41 δείχνει το Σχεσιακό Μοντέλο με μορφή πινάκων.



Εικόνα 40. Διάγραμμα Οντοτήτων – Συσχετίσεων



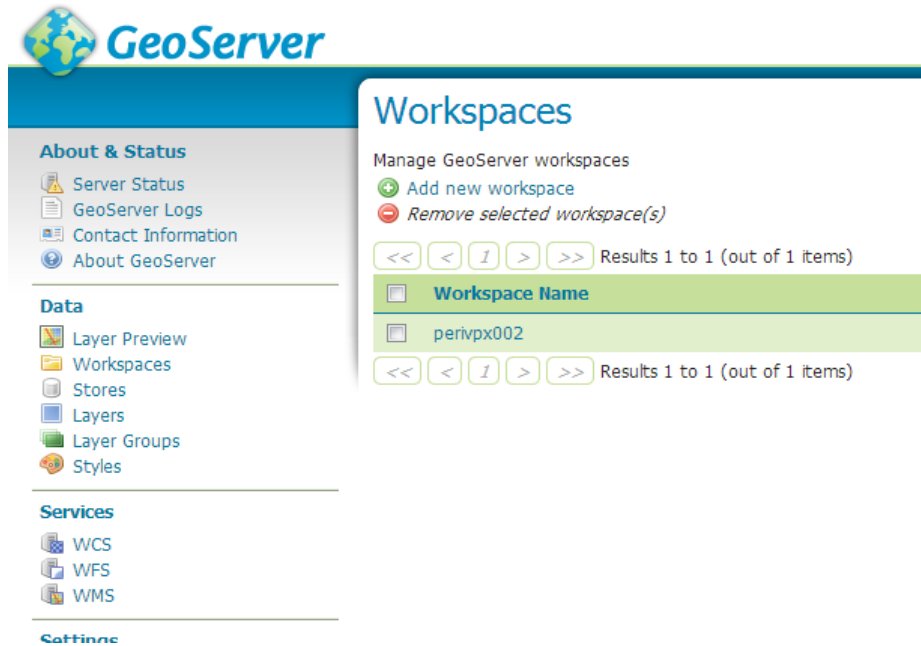
Εικόνα 41. Σχεσιακό Μοντέλο

Όπως φαίνεται και από τα παραπάνω διαγράμματα, ο πίνακας metrhsh περιέχει επίσης, εκτός από την τιμή μέτρησης και την ημερομηνία, τον κωδικό ρύπου και τον κωδικό του σταθμού. Πρόκειται δηλαδή για ξένα κλειδιά, τα οποία αναφέρονται στους πίνακες stathmoi και rypoi αντίστοιχα. Με πρόγραμμα Java κατασκευάστηκαν queries ώστε να γεμίσουν οι πίνακες με τις κατάλληλες τιμές που προέκυψαν μετά από επεξεργασία των δεδομένων.

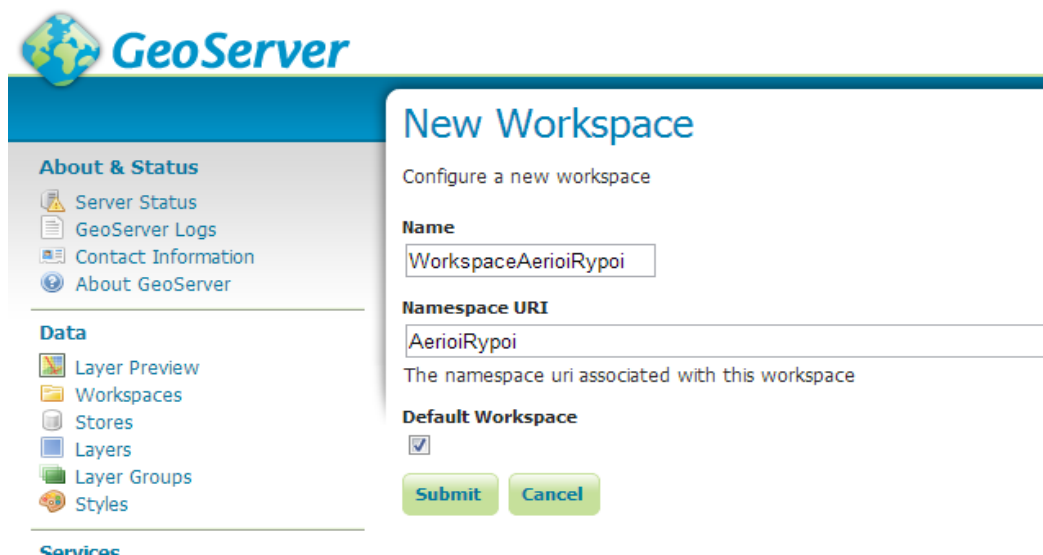
Στη συνέχεια, αφού ολοκληρώθηκε η βάση δεδομένων, συνδέθηκε με τον GeoServer. Με αυτό τον τρόπο γίνεται δυνατή η αξιοποίηση της βάσης από τη διαδικτυακή εφαρμογή, μέσω αιτημάτων που γίνονται στον GeoServer, είτε με την τεχνική AJAX είτε με queries γραμμένα στις γλώσσες CQL (Common Query Language) και ECQL(Extended Common Query Language).

Πρόκειται για γλώσσες που έχουν αναπτυχθεί από την OGC (Open Geospatial Consortium) σε συμφωνία με το πρότυπο WFS, και ουσιαστικά επιτρέπουν στον προγραμματιστή να κάνει αιτήματα στο Server, εφαρμόζοντας συγκεκριμένα φίλτρα στα δεδομένα. Παρακάτω, φαίνεται σε βήματα, ο τρόπος διασύνδεσης της βάσης δεδομένων με το Geoserver.

**Βήμα 1ο** : Αφού γίνει σύνδεση στο Geoserver (username = admin, password=geoserver οι default τιμές) πρέπει να οριστεί ένα νέο Workspace. Αυτό γίνεται επιλέγοντας **Workspaces** από την αντίστοιχη επιλογή στο μενού Data στα αριστερά και στη συνέχεια το κουμπί **Add new Workspace** όπως φαίνεται στις εικόνες 42α και 42β.



Εικόνα 42α. GeoServer – Δημιουργία νέου Workspace

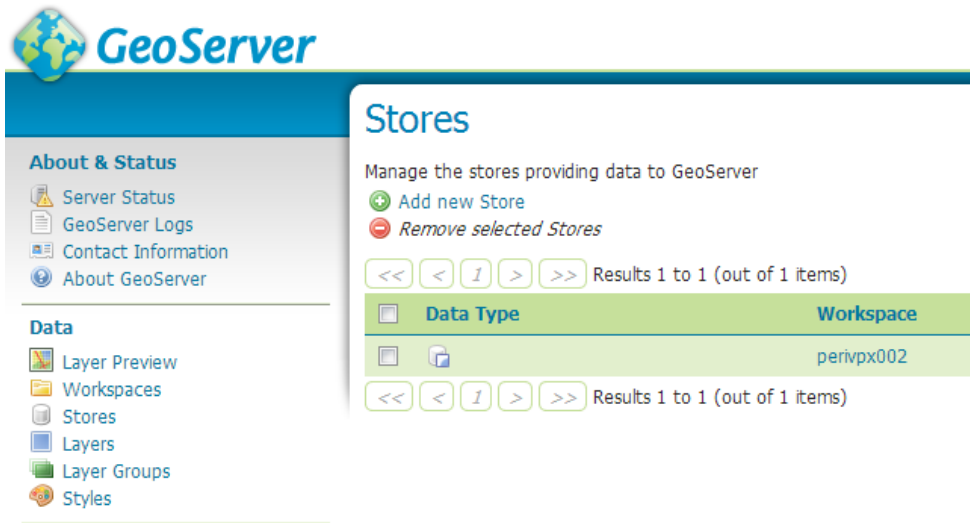


Εικόνα 42β. GeoServer – Δημιουργία νέου Workspace

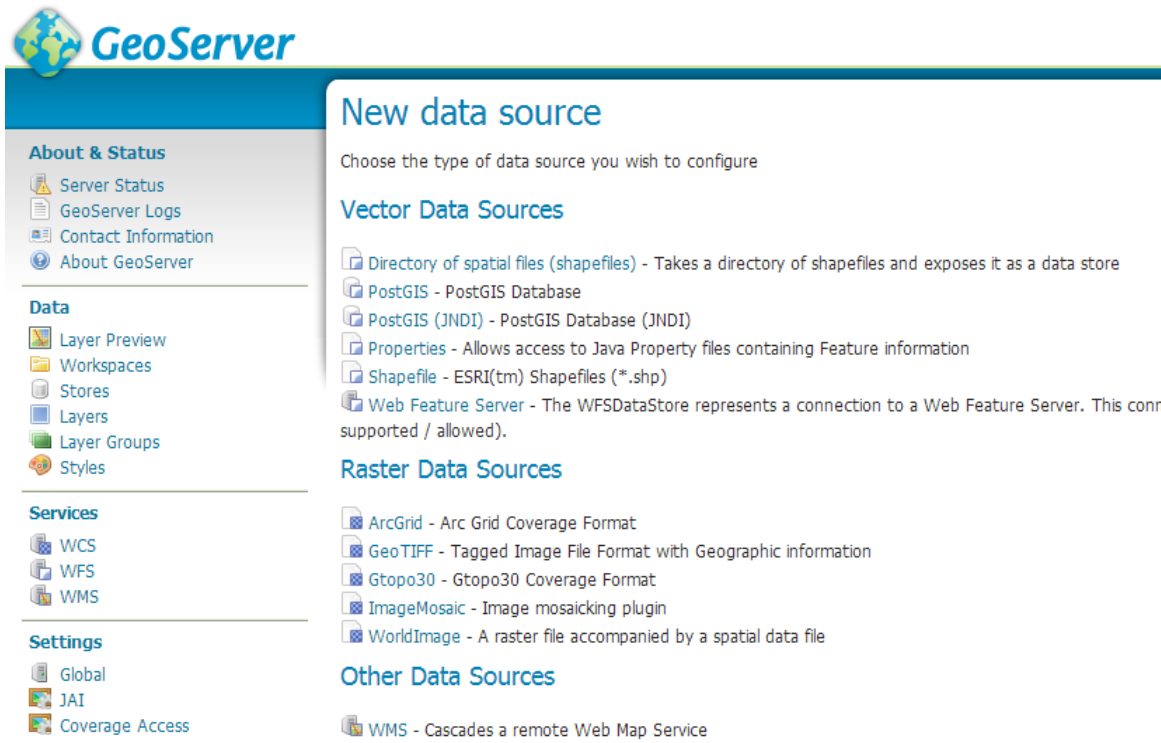
Στη συνέχεια, κάνοντας κλικ στο νέο Workspace που δημιουργήθηκε - ώστε να κάνει Edit - ο χρήστης μπορεί να το ενεργοποιήσει και να επιλέξει τις υπηρεσίες (Services) που επιθυμεί να υποστηρίζονται όπως WMS, WFS WCS (Εικόνα 43). Επίσης, σε αυτή τη σελίδα μπορεί να επιλέξει την κωδικοποίηση χαρακτήρων (UTF-8) αλλά και να προσθέσει τις προσωπικές του πληροφορίες. Αφού συμπληρώσει τα απαιτούμενα πεδία (τα περισσότερα είναι προαιρετικά) πατάει Save για να αποθηκεύσει τις αλλαγές.

Εικόνα 43. GeoServer – Ρυθμίσεις παραμέτρων του νέου Workspace

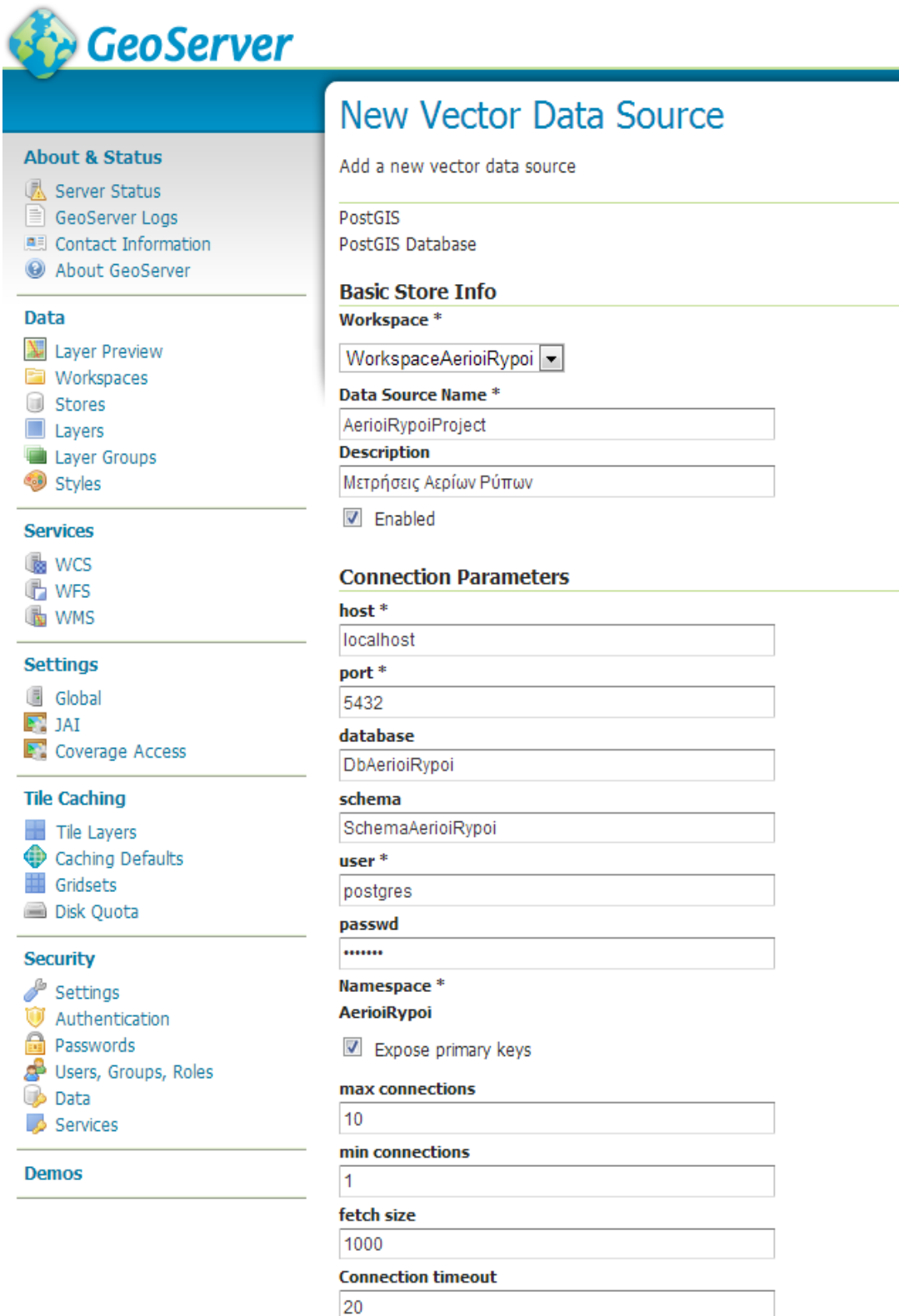
**Βήμα 2ο :** Στη συνέχεια, επιλέγοντας από το μενού Data την επιλογή **Stores**, με το κουμπί **Add new Store** (Εικόνα 44) ο χρήστης καλείται να επιλέξει την προέλευση των δεδομένων. Από τις διαθέσιμες επιλογές για τη συγκεκριμένη εργασία επιλέγει από την κατηγορία Vector Data Sources την επιλογή **PostGIS – PostGIS Database** (Εικόνα 45). Στη συνέχεια, όπως φαίνεται στην Εικόνα 46 καλείται να συμπληρώσει τις απαραίτητες παραμέτρους ώστε να συνδεθεί με τη βάση. Για την λειτουργία της εφαρμογής είναι απαραίτητο να ονομάσει το Workspace ως **WorkspaceAerioiRypoi**, και να επιλέξει την επιλογή **Expose primary keys** καθώς χρησιμοποιούνται οι τιμές αυτές από τον κώδικα JavaScript. Αφού συμπληρώσει τα απαραίτητα, πατάει Save.



Εικόνα 44. GeoServer – Δημιουργία νέου Data Store



Εικόνα 45. GeoServer – Επιλογή τύπου δεδομένων



**GeoServer**

## New Vector Data Source

Add a new vector data source

PostGIS  
PostGIS Database

---

### Basic Store Info

**Workspace \***  
WorkspaceAerioiRypoi

**Data Source Name \***  
AerioiRypoiProject

**Description**  
Μετρήσεις Αερίων Ρύπων

Enabled

---

### Connection Parameters

**host \***  
localhost

**port \***  
5432

**database**  
DbAerioiRypoi

**schema**  
SchemaAerioiRypoi

**user \***  
postgres

**passwd**  
\*\*\*\*\*

**Namespace \***  
AerioiRypoi

Expose primary keys

**max connections**  
10

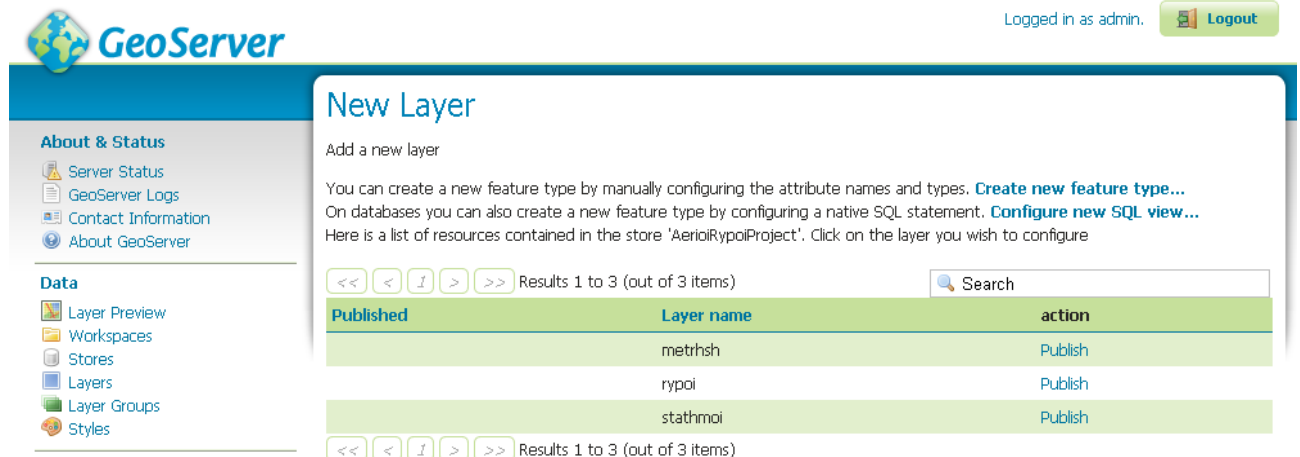
**min connections**  
1

**fetch size**  
1000

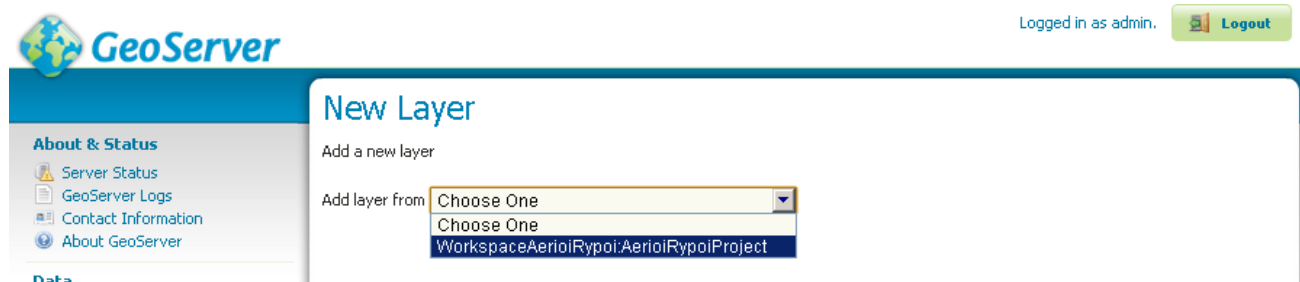
**Connection timeout**  
20

Εικόνα 46. GeoServer – Ρυθμίσεις παραμέτρων σύνδεσης στη βάση δεδομένων

**Βήμα 3ο** :Στη συνέχεια, ο χρήστης καλείται να επιλέξει ποιους από τους πίνακες της βάσης δεδομένων ενδιαφέρει να εισάγει στον GeoServer (Εικόνα 47α). Στην συγκεκριμένη περίπτωση και οι τρεις πίνακες (stathmoi, rypoi, metrhsh) πρέπει να γίνουν publish. Αυτό μπορεί να γίνει επίσης από την επιλογή **Add a New Resource** από το μενού Layers όπως φαίνεται στην Εικόνα 47β παρακάτω.



Εικόνα 47α. Add new layer



Εικόνα 47β. Add new layer

Στη συνέχεια, πρέπει να γίνει ρύθμιση παραμέτρων για κάθε layer. Αυτό αφορά κυρίως το Σύστημα Αναφοράς (Coordinate Reference System) και το Bounding Box και έχουν χρησιμότητα σε layers για τα οποία υπάρχει χωρική πληροφορία (συντεταγμένες). Όσον αφορά το πρώτο, ο GeoServer δίνει τη δυνατότητα επιλογής από ένα πολύ μεγάλο πλήθος διαφορετικών συστημάτων αναφοράς, τόσο για το εγγενές σύστημα (Native) όσο και για δεδηλωμένο (Declared), ώστε να είναι δυνατόν on-the-fly transformation. Στη συγκεκριμένη εργασία – για τον πίνακα stathmoi – επιλέγεται ως σύστημα αναφοράς το WGS84 με κωδικό EPSG:4326. Το Bounding Box αποτελεί ένα παραλληλόγραμμο μέσα στο οποίο περικλείονται όλα τα στοιχεία του layer, και ο GeoServer μπορεί να το υπολογίσει αυτόματα με την επιλογή **Compute from data**.

Σύστημα Αναφοράς και Bounding Box είναι απαραίτητο να οριστούν ακόμα και για πίνακες για τους οποίους δεν είναι διαθέσιμη χωρική πληροφορία για να μπορέσει ο GeoServer να κάνει publish τα layer. Σε αυτή την περίπτωση – όπως για τους πίνακες rypoi και metrhsh – ως σύστημα αναφοράς ορίστηκε και πάλι το WGS84 ενώ η επιλογή Compute from data δίνει άκυρες τιμές (min X,Y = -1, max X,Y =0). Στις Εικόνες 48α και 48β παρακάτω φαίνεται η ρύθμιση των παραμέτρων για το layer stathmoi της καρτέλας Data.





**About & Status**

- Server Status
- GeoServer Logs
- Contact Information
- About GeoServer

---

**Data**

- Layer Preview
- Workspaces
- Stores
- Layers
- Layer Groups
- Styles

---

**Services**

- WCS
- WFS
- WMS

---

**Settings**

- Global
- JAI
- Coverage Access

---

**Tile Caching**

- Tile Layers
- Caching Defaults
- Gridsets
- Disk Quota

---

**Security**

- Settings
- Authentication
- Passwords
- Users, Groups, Roles
- Data
- Services

---

**Demos**

## Edit Layer

Edit layer data and publishing

### WorkspaceAerioiRypoi:stathmoi

Configure the resource and publishing information for the current layer

Data

Publishing

Dimensions

Tile Caching

#### Basic Resource Info

**Name**

**Title**

**Abstract**

---

**Keywords**

**Current Keywords**

- features
- stathmoi

**New Keyword**

**Vocabulary**

---

**Metadata links**

No metadata links so far

Note only FGDC and TC211 metadata links show up in WMS 1.1.1 capabilities

Εικόνα 48α. Ρύθμιση παραμέτρων της καρτέλας Data για το layer stathmoi

## Coordinate Reference Systems

### Native SRS

EPSG:4326 [EPSG:WGS 84...](#)

### Declared SRS

EPSG:4326  [EPSG:WGS 84...](#)

### SRS handling

Force declared ▼

## Bounding Boxes

### Native Bounding Box

Min X	Min Y	Max X	Max Y
20,85	35,3325	25,08	40,739842

[Compute from data](#)

### Lat/Lon Bounding Box

Min X	Min Y	Max X	Max Y
20,85	35,3325	25,08	40,739842

[Compute from native bounds](#)

## Feature Type Details

Property	Type	Nullable	Min/Max Occurrences
sta_id	Integer	false	1/1
sta_name	String	true	0/1
sta_geometry	Geometry	true	0/1
sta_h	Integer	true	0/1
sta_xar	String	true	0/1
sta_kwd	String	true	0/1

[Reload feature type](#) ⚠ ...

### Εικόνα 48β. Ρύθμιση παραμέτρων της καρτέλας Data για το layer stathmoi

Στις καρτέλες Publishing (Εικόνα 49), Dimensions (Εικόνα 50) και Tile Caching(Εικόνα 51) διατηρούνται οι προεπιλεγμένες (default) τιμές.

## WorkspaceAerioiRypoi:stathmoi

Configure the resource and publishing information for the current layer

Data
Publishing
Dimensions
Tile Caching

### Edit Layer

---

**Name**

Enabled

Advertised

**HTTP Settings**

---

Response Cache Headers

**Cache Time (seconds)**

**WFS Settings**

---

**Per-Request Feature Limit**

**Maximum number of decimals**

**WMS Settings**

---

Queryable

**Default Style**

■

**Additional Styles**

Available Styles
burg
capitals
cite_lakes
dem
giant_polygon
grass
green
line
poi
point

Selected Styles

**Default Rendering Buffer**

**Default WMS Path**

**Authority URLs for this WMS Layer**

---

No authority URLs so far

### Layer Identifiers

Εικόνα 49. Ρύθμιση παραμέτρων της καρτέλας Publishing για το layer stathmoi

## WorkspaceAerioiRypoi:stathmoi

Configure the resource and publishing information for the current layer

Data
Publishing
Dimensions
Tile Caching

**Time**

Enabled *Cannot enable, no attribute of type Date found*

**Elevation**

Enabled

Save
Cancel

Εικόνα 50. Ρύθμιση παραμέτρων της καρτέλας Dimension για το layer stathmoi

## WorkspaceAerioiRypoi:stathmoi

Configure the resource and publishing information for the current layer

Data
Publishing
Dimensions
Tile Caching

**Tile cache configuration**

Create a cached layer for this layer

Enable tile caching for this layer

**Metatiling factors**  
 tiles wide by  tiles high

**Gutter size in pixels**

**Tile Image Formats**

image/png  
 image/png8  
 image/jpeg  
 image/gif

**Parameter Filters**

Create a separate cache for each STYLE  
 Create a separate cache for the TIME WMS parameter  
 Create a separate cache for the ELEVATION WMS parameter

**Available gridsets**

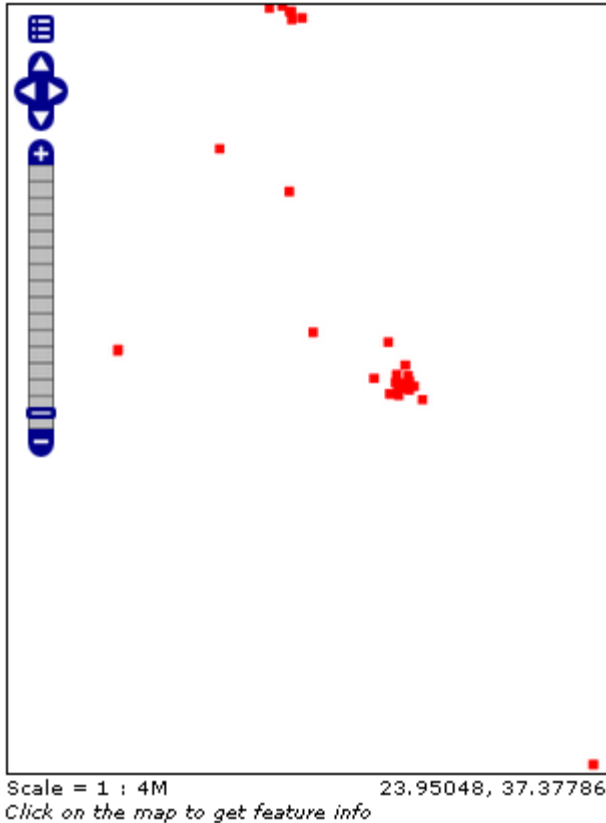
Gridset	Published zoom levels	Cached zoom levels	Grid subset bounds
EPSG:900913	Min / Max	Min / Max	Dynamic
EPSG:4326	Min / Max	Min / Max	Dynamic

Add grid subset:

Save
Cancel

Εικόνα 51. Ρύθμιση παραμέτρων της καρτέλας Tile Caching για το layer stathmoi

Μόλις ολοκληρωθεί η ένταξη του πίνακα stathmoi ως layer στον GeoServer, ο τελευταίος δίνει τη δυνατότητα προβολής του με την βιβλιοθήκη OpenLayers, από το μενού Layer Preview (Εικόνα 52) ή αποθήκευσης του τοπικά στον υπολογιστή σε πλήθος διαφορετικών format (π.χ. KML, GML, CSV κ.α.).



**Εικόνα 52. Layer Preview από το GeoServer για το layer stathmoi**

Η ίδια διαδικασία ακολουθείται για την εισαγωγή και των δύο άλλων πινάκων (gyroi, metrshsh) ως layer, παρά ο γεγονός πως για αυτούς δεν είναι δυνατή η οπτικοποίηση, εφόσον στερούνται χωρικής πληροφορίας. Αντίθετα, σε αυτά τα layers απευθύνονται τα αιτήματα που γίνονται στον GeoServer μέσω JavaScript.

Το τελικό στάδιο για την ανάπτυξη της εφαρμογής αποτελεί η δημιουργία της γραφικής διεπαφής χρήστη (User Interface) η οποία κατασκευάστηκε με HTML5 και CSS3 καθώς και των αλγορίθμων λειτουργίας (flow logic) που έγινε σε JavaScript. Πιο συγκεκριμένα, τα τέσσερα αρχεία που αποτελούν την εφαρμογή (AerioiRyroiProject.html, AerioiRyroiProject.css, AerioiRyroiProject.js, objSort.js) τοποθετήθηκαν όλα μαζί στο φάκελο δεδομένων του GeoServer ώστε να είναι προσβάσιμα από αυτόν. Για τυπική εγκατάσταση σε λειτουργικό Windows η θέση του φακέλου δεδομένων είναι: C:\Program Files\GeoServer2.2.1\data\_dir\www και εκεί ή σε κάποιο υποκατάλογο αυτού αποθηκεύονται τα αρχεία. Στο Παράρτημα, διατίθεται το σύνολο του κώδικα των εν λόγω αρχείων, ενώ εδώ θα παρατεθούν αποσπασματικά συγκεκριμένα τμήματα κώδικα με την απαραίτητη τεκμηρίωση

Το αρχείο AerioiRyroiProject.html είναι ένα τυπικό αρχείο το οποίο περιέχει τα συστατικά μέρη (tags) της ιστοσελίδας. Θα ήταν δυνατόν να εμπεριέχει και κώδικα JavaScript ή/και στοιχεία μορφοποίησης (δηλαδή τα περιεχόμενα του css αρχείου), αλλά επειδή θεωρείται ως βέλτιστη προγραμματιστική πρακτική ο διαχωρισμός της μορφοποίησης και της λογικής από τα στοιχεία της HTML προτιμήθηκε το καθένα να αποτελεί ξεχωριστό αρχείο. Έτσι, στο εν

λόγω αρχείο περιέχονται μόνο οι αναφορές στα αρχεία JavaScript και css μαζί με αναφορές σε εξωτερικές βιβλιοθήκες απαραίτητες για τη λειτουργία της εφαρμογής. Παρακάτω παρατίθενται οι αναφορές στα αρχεία και τις βιβλιοθήκες:

```
<head>

  <!-- Αρχεία της εφαρμογής -->

  <!-- Αρχείο μορφοποίησης css -->
  <link rel="stylesheet" href="AerioiRyroiProject.css">

  <!-- Αρχείο JavaScript με τον αλγόριθμο της εφαρμογής -->
  <script src="AerioiRyroiProject.js" defer="defer" type="text/javascript">
  </script>

  <!-- Αρχείο JavaScript για objectSorting -->
  <script type="text/javascript" src="objSort.js"></script>

  <!-- Εξωτερικές Βιβλιοθήκες -->

  <!-- OpenLayers -->
  <script src="http://openlayers.org/dev/OpenLayers.js"></script>

  <!-- OpenStreetMap base layer js -->
  <script src="http://www.openstreetmap.org/openlayers/OpenStreetMap.js">
  </script>

  <!-- Google Maps -->
  <script src="http://maps.google.com/maps/api/js?sensor=false"></script>

</head>
```

Τα αρχεία της εφαρμογής, θα πρέπει να βρίσκονται στον ίδιο φάκελο μαζί με το αρχείο HTML το οποίο τα καλεί. Η ιδιότητα `defer="defer"` είναι συμβατή με τη σύνταξη για XHTML (αποτελεί πιο αυστηρή σύνταξη HTML) και εξασφαλίζει πως το `script` δεν θα τρέξει μέχρι να φορτωθεί ολόκληρη η ιστοσελίδα – οπότε θα έχουν αποτιμηθεί οι εκφράσεις που είναι απαραίτητες για την ορθή λειτουργία του. Ακόμα, η βιβλιοθήκη `OpenLayers` διατίθεται για download οπότε θα ήταν δυνατόν να αποθηκευθεί στον ίδιο φάκελο με τα υπόλοιπα αρχεία και να καλείται από εκεί. Αυτή η λύση δεν προτιμήθηκε, καθώς αποκλείει μελλοντικές βελτιώσεις που μπορεί να προσφέρει ο οργανισμός που την αναπτύσσει.

Ακόμα, στο αρχείο `AerioiRyroiProject.html` ορίζονται τα δομικά στοιχεία της εφαρμογής, οργανωμένα σε τμήματα (`divs`). Τα βασικά τμήματα της ιστοσελίδας είναι τρία και αφορούν στις θέσεις όπου εμφανίζονται :

1. Η φόρμα αναζήτησης των τιμών μέτρησης των αερίων ρύπων,
2. Ο χάρτης
3. Η κλίμακα θέασης του χάρτη και οι συντεταγμένες του ποντικιού (το καθένα σε ξεχωριστό `div` εμφωλευμένα στο μητρικό).

Εκτός από τα τρία βασικά, υπάρχουν άλλα δύο τμήματα τα οποία δημιουργούνται δυναμικά μετά από ενέργειες του χρήστη. Το πρώτο αφορά τις ρυθμίσεις του χάρτη, αν πατηθεί το αντίστοιχο κουμπί πάνω από την μπάρα του zoom, ενώ το δεύτερο εμφανίζει τα δεδομένα των σταθμών όταν εκτελείται η υπηρεσία WFS. Παρακάτω, ακολουθούν οι ορισμοί των τριών βασικών τμημάτων της ιστοσελίδας.

#### 1. Φόρμα αναζήτησης των τιμών μέτρησης των αερίων ρύπων.

```
<div id="my">
  <form name="myform">

    <label for="st">Σταθμός Μέτρησης</label>
    <select id="st" onchange="findOther( this.value,'ry', Stathmoi,
                                      'sta_id', 'pinRyp', Rypoi,
                                      'id_rupoi', 'ry_name')" >

    </select>

    <label for="ry">Ρύπος</label>
    <select id="ry" onchange="findOther(this.value, 'st', Rypoi,
                                      'id_rupoi', 'pinStat',
                                      Stathmoi, 'sta_id',
                                      'sta_name')">

    </select>

    <label for="hmera">Ημέρα</label>
    <select id="hmera"></select>

    <label for="mhnas">Μήνας</label>
    <select id="mhnas"></select>

    <label for="etos">Έτος</label>
    <select id="etos"></select>

    <input type="button" onclick="formSubmit()" value="Submit form">

  </form>
</div>
```

Όπως φαίνεται, η φόρμα αποτελείται από 5 στοιχεία επιλογής (select tags), μαζί με τις ετικέτες (label tags) του καθένα. Αυτά αφορούν την είσοδο σταθμού(id=st), ρύπου(id=ry), ημέρας(id=hmera), μήνα(id=mhnas) και έτους(id=etos). Όσον αφορά την επιλογή σταθμού και ρύπου, σε κάθε αλλαγή της τιμής τους (attribute onchange), καλείται η συνάρτηση findOther η οποία ορίζεται στο JavaScript αρχείο AerioiRypoiProject.js. Στο ίδιο αρχείο καθορίζονται μετά από αιτήματα στον GeoServer και οι πιθανές τιμές εισόδου για τα συγκεκριμένα στοιχεία και θα αναλυθούν παρακάτω. Τη φόρμα ολοκληρώνει ένα κουμπί submit, με το πάτημα του οποίου (attribute onclick) καλείται η συνάρτηση formSubmit η οποία ορίζεται επίσης στο ίδιο JavaScript αρχείο.

## 2. Χάρτης

```
<div id="map">
  
</div>
```

Όπως φαίνεται εδώ, δημιουργείται το τμήμα που θα τοποθετηθεί ο χάρτης και το οποίο περιέχει επίσης μία εικόνα (id="options") η οποία παρέχεται από τον GeoServer.

## 3. Κλίμακα θέασης και συντεταγμένες

```
<div id="wrapper">
  <div id="location"> location </div>
  <div id="lf"> λ,φ : </div>
  <div id="scale"></div>
</div>
```

Εδώ, ορίζεται ένα τμήμα (id="wrapper") το οποίο περιέχει τρία άλλα τμήματα μέσα του, το ένα για να αναγράφεται η κλίμακα (id="scale"), ενώ το δεύτερο και το τρίτο αφορούν στις συντεταγμένες που ο χρήστης τοποθετεί το ποντίκι. Το ένα (id="lf"), έχει χρησιμότητα ετικέτας, δηλαδή αναγράφει λ,φ: , ενώ το άλλο (id="location") φιλοξενεί τις αριθμητικές τιμές των συντεταγμένων. Όσο ο χρήστης δεν έχει τοποθετήσει ακόμα το ποντίκι μέσα στο χάρτη, εκεί αναγράφεται η λέξη location, ενώ στην συνέχεια τα περιεχόμενα του τμήματος αλλάζουν δυναμικά συνεχώς ανάλογα την κίνηση του ποντικιού.

Ο αλγόριθμος λειτουργίας της εφαρμογής περιέχεται στο JavaScript αρχείο AerioiRypoiProject.js. Εκεί έχουν οριστεί και υλοποιηθεί οι απαραίτητες μεταβλητές, συναρτήσεις και αντικείμενα που χρειάζεται η εφαρμογή. Πιο συγκεκριμένα, στην αρχή ορίζονται κάποιες καθολικές (global) μεταβλητές, οι οποίες είναι απαραίτητες για την εκτέλεση και χρησιμοποιούνται από περισσότερες των μία συναρτήσεις. Τέτοιες είναι τα διάφορα συστήματα προβολής, καθώς και κάποιες αλφαριθμητικές σταθερές οι οποίες χρησιμοποιούνται για τα αιτήματα στον GeoServer και πρέπει να είναι σύμφωνες με τις ρυθμίσεις που έγιναν σε αυτόν (π.χ. το όνομα του Namespace, των Layer κ.α.). Επίσης ορίζονται δύο Arrays (Rypoi, Stathmoi) στα οποία θα αποθηκευθούν αργότερα τα αντικείμενα τύπου rypos και stathmos.

```
var urlBase = "http://localhost:8080/geoserver/";
var layerNamespace = "WorkspaceAerioiRypoi";
var urlWfs = "/wfs?service=wfs&version=2.0.0";
var urlWfs1 = "/wfs?service=wfs&version=1.0.0";
var urlGetFeature = "&request=GetFeature";

var urlTypeNameRypoi = "&typeName=" + layerNamespace + ":rypoi";
var Rypoi = new Array();
var urlLayerRypoi = urlBase + layerNamespace + urlWfs + urlGetFeature
  + urlTypeNameRypoi;

var urlTypeNameStathmoi = "&typeName=" + layerNamespace + ":stathmoi";
var Stathmoi = new Array();
var urlLayerStathmoi = urlBase + layerNamespace + urlWfs + urlGetFeature
  + urlTypeNameStathmoi;
```



```
var urlTypeNameMetrhsh = "&typeName=" + layerNamespace + ":metrhsh";
var Metrsh = new Array();
var urlLayerMetrhsh = urlBase + layerNamespace + urlWfs1 + urlGetFeature +
urlTypeNameMetrhsh;
```

Στη συνέχεια, ορίζονται οι συναρτήσεις που είναι απαραίτητες για την εφαρμογή. Μόλις φορτωθεί η σελίδα σε κάποιον Web browser καλείται και τρέχει η συνάρτηση `init()`, η οποία είναι η βασική συνάρτηση που καλεί τις υπόλοιπες. Η συγκεκριμένη, αφού κάνει τις απαιτούμενες ρυθμίσεις που αφορούν στο χάρτη, (προσθήκη των `layer`, `zoom`, `map extend` κ.α.) καλεί την συνάρτηση `getdata`. Ο σκοπός αυτής της συνάρτησης είναι να πάρει όλες τις απαραίτητες τιμές της βάσης δεδομένων (που τώρα έχει κάνει `publish` ως `layer` ο `GeoServer`), να φτιάξει αντικείμενα `rypos` και `stathmos` και να τα αποθηκεύσει σε `Arrays` ούτως ώστε να μπορούν στην συνέχεια να χρησιμοποιηθούν άμεσα από τις υπόλοιπες συναρτήσεις. Δεδομένου ότι πρόκειται ουσιαστικά για την ίδια διαδικασία - δηλαδή αιτήματα στον `GeoServer` και επεξεργασία των αποκρίσεων - η οποία πρέπει να γίνει σε δύο διαφορετικά `Layers` και λόγω των πλεονεκτημάτων που προσφέρει η τμηματοποίηση του κώδικα με την προσθήκη συναρτήσεων, προτιμήθηκε η κατασκευή της συνάρτησης `getdata` με παραμέτρους ώστε να είναι επαναχρησιμοποιήσιμη. Έτσι, καλείται δύο φορές, μία με ορίσματα που αφορούν στο `layer` `rypoi` και μία για το `layer` `stathmoi`. Οι παράμετροι κλήσης της συνάρτησης είναι οι εξής:

- α) Το `url` στο οποίο θα γίνουν τα αιτήματα,
- β) ένα `Array` στο οποίο θα αποθηκευθούν τα αντικείμενα που θα κατασκευαστούν ανάλογα με τις αποκρίσεις του `GeoServer`,
- γ) και μια μέθοδος κατασκευής (`constructor`) των αντικειμένων αυτών.

Το `url` είναι η διεύθυνση στην οποία απευθύνονται τα αιτήματα στον `GeoServer` η οποία είναι αποθηκευμένη σε αλφαριθμητική σταθερά (`string`) και χτίζεται ανάλογα το αίτημα. Κάθε αίτημα απαιτεί διαφορετικό `url` οπότε είναι απαραίτητο να δίδεται ως παράμετρος ανάλογα την κλήση της συνάρτησης. Στην εφαρμογή χρησιμοποιούνται τρία `url`, για κάθε ένα από τα `layer` `rypoi(urlLayerRypoi)`, `stathmoi(urlLayerStathmoi)` και `metrhsh(urlLayerMetrhsh)`. Ο σχηματισμός της συμβολοσειράς που κρατάει το `url` όπως και η δημιουργία των δύο `Arrays` που θα χρησιμοποιηθούν για την κλήση γίνεται από τη συνάρτηση `init()` όπως παρουσιάστηκαν στο τμήμα κώδικα παραπάνω. Στη συνέχεια παρουσιάζονται οι κατασκευαστές των αντικειμένων οι οποίοι παίρνουν ως παράμετρο ένα πίνακα `a` με τις τιμές που θα τοποθετηθούν σε αυτά ως ιδιότητες.

```
function rypos(a) {
    this.id_rypoi = a[0]; // id του ρύπου στη Βάση Δεδομένων
    this.ry_kwd = a[1]; // Κωδικός του ρύπου
    this.ry_name = a[2]; // Όνομα του ρύπου
    this.pinStat = new Array(); // Πίνακας που περιέχει τα id των σταθμών
    // που μετράνε το συγκεκριμένο ρύπο
}

function stathmos(a) {
    this.sta_id = a[0]; // id σταθμού
    this.sta_name = a[1]; // όνομα
    this.sta_geometry = a[2]; // γεωμετρία
    this.sta_h = a[3]; // υψόμετρο
    this.sta_xar = a[4]; // χαρακτηρισμός σταθμού
    this.sta_kwd = a[5]; // κωδικός σταθμού
    this.pinRyp = new Array(); // Πίνακας που περιέχει τα id των ρύπων
    // που μετρούνται από το συγκεκριμένο σταθμό
}
```

Παρακάτω φαίνεται η κλήση της συνάρτησης `getdata` για το `layer rypoi` με τις ανάλογες παραμέτρους.

```
getdata(urlLayerRypoi, Rypoi, rypos);
```

Όπως έχει ήδη αναφερθεί, η συνάρτηση αυτή κάνει αιτήματα στον GeoServer και αξιοποιεί τις αποκρίσεις του φτιάχνοντας αντικείμενα. Η απάντηση των αιτημάτων AJAX στην ουσία είναι δεδομένα δομημένα σε μορφή XML (Extensible Markup Language) τα οποία επεξεργάζεται η συνάρτηση `dataprocess` που καλείται μέσα στην `getdata`. Η συγκεκριμένη συνάρτηση δέχεται ως παραμέτρους τα δεδομένα μορφής XML (την απόκριση του GeoServer σε κάθε αίτημα) καθώς και το όνομα του πεδίου του οποίου η τιμή ενδιαφέρει και επιστρέφει αυτήν τη τιμή. Ουσιαστικά, η `dataprocess` αποδομεί την απόκριση του GeoServer με επεξεργασία του XML μέσα σε ένα επαναληπτικό βρόχο ώστε να πάρει τις τιμές των πεδίων που θα αποθηκευθούν ως ιδιότητες των αντικειμένων.

Έτσι, μετά την ολοκλήρωση της εκτέλεσης της συνάρτησης `getdata` έχουν γεμίσει τα `Arrays` με αντικείμενα που αντιστοιχούν στα δεδομένα του GeoServer για τα `layer rypoi` και `stathmoi`. Στη συνέχεια χρησιμοποιήθηκε η μέθοδος `objSort` η οποία βρίσκεται στο αρχείο JavaScript `objSort.js` το οποίο δημιούργησε ο Thomas Frank και διατίθεται ελεύθερα στο διαδίκτυο. Η μέθοδος αυτή ταξινομεί τα περιεχόμενα ενός `Array` αντικειμένων με βάση κάποια ιδιότητα τους. Χρησιμοποιείται για να ταξινομηθούν οι λίστες `Rypoi` και `Stathmoi` με βάση το `id` τους. Κατόπιν, αφαιρούνται από το `Array Stathmoi` οι σταθμοί για τους οποίους δεν διατίθενται μετρήσεις και καλείται η συνάρτηση `getmetrhsh`.

Όπως φαίνεται παραπάνω, κάθε αντικείμενο `rypos` και `stathmos` έχει ως πεδίο ένα `Array` (`pinStat` και `pinRyp` αντίστοιχα) στο οποίο κρατούνται για τους μεν ρύπους οι σταθμοί οι οποίοι τον μετράνε, για τους δε σταθμούς οι ρύποι τους οποίους αυτοί μετρούν. Αυτά τα `Arrays` αντικειμένων γεμίζουν από τη συνάρτηση `getmetrhsh`, έτσι ώστε να είναι άμεσα διαθέσιμη αυτή η πληροφορία από τις υπόλοιπες συναρτήσεις της εφαρμογής αργότερα χωρίς να απαιτούνται αιτήματα στον GeoServer. Αντίθετα, όλα τα αιτήματα γίνονται από αυτή τη συνάρτηση μέσα σε δύο επαναληπτικούς βρόχους. Το `url` που χρησιμοποιείται για αυτά είναι διαφορετικό για κάθε αντικείμενο και προκύπτει από συνένωση του βασικού `url` για το `layer metrhsh` (`urlLayerMetrhsh`) με την προσθήκη του κατάλληλου κάθε φορά φίλτρου. Τα φίλτρα δημιουργούνται με βάση τη γλώσσα CQL (Common Query Language) η οποία υποστηρίζεται από το GeoServer για την εκτέλεση ερωτημάτων στα `layers`.

Πιο συγκεκριμένα, εκτελείται μία επαναληπτική διαδικασία για κάθε αντικείμενο που περιέχεται μέσα στο `Array Stathmoi` (το οποίο περιέχει τα δεδομένα για όλους τους σταθμούς) οπότε ανακτά το `id` κάθε σταθμού. Στη συνέχεια εκτελείται ένας δεύτερος βρόχος μέσα στον προαναφερθέν και ανακτάται το `id` κάθε ρύπου. Έτσι, μπορεί να κατασκευαστεί ένα αίτημα προς τον GeoServer με χρήση φίλτρου CQL το οποίο ρωτάει αν το `layer metrhsh` περιέχει εγγραφή που αφορά το σταθμό και το ρύπο με τα συγκεκριμένα `id`.

Σε αυτό το σημείο πρέπει να αναφερθεί πως η συγκεκριμένη διαδικασία απαιτεί αρκετό χρόνο αφού ουσιαστικά εκτελούνται  $m \cdot n$  αιτήματα ( $m = \text{ο αριθμός των σταθμών}$ ,  $n = \text{ο αριθμός των ρύπων}$ ) ενώ δεν είναι `asynchronous` διότι πρέπει να έχει εκτελεστεί και ολοκληρωθεί πριν από την κλήση των επόμενων συναρτήσεων. Αυτό θα ήταν δυνατόν να αποφευχθεί αν ο GeoServer υποστήριζε φίλτρο ανάλογο της εντολής `SELECT DISTINCT` της γλώσσας SQL για τα `layers` κάτι όμως που δεν συμβαίνει.

Έτσι, πραγματοποιούνται όλα τα αιτήματα και η απόκριση ελέγχεται από τη συνάρτηση `SendReq` η οποία επιστρέφει `true` στην περίπτωση που ο συγκεκριμένος σταθμός μετράει το συγκεκριμένο ρύπο. Σε αυτή την περίπτωση προστίθεται στο `Array pinRyp` αυτού του σταθμού ο κωδικός αυτού του ρύπου και στο `Array pinStat` του συγκεκριμένου ρύπου ο κωδικός αυτού του σταθμού.

Μετά την ολοκλήρωση εκτέλεσης της συνάρτησης `getmetrhsh` η εφαρμογή έχει συγκεντρώσει όλα τα απαραίτητα δεδομένα για να απαντά στα αιτήματα του χρήστη χωρίς χρονοτριβή. Οι υπόλοιπες συναρτήσεις που έχουν δημιουργηθεί αφορούν στη διαλειτουργικότητα μεταξύ του User Interface και της εφαρμογής δηλαδή στην διασύνδεση των επιλογών του χρήστη με τα δεδομένα της εφαρμογής.

Η πρώτη από αυτές τις συναρτήσεις είναι η `fillsel` η οποία γεμίζει τα πεδία επιλογής του χρήστη (δηλαδή τα `select tags` του HTML αρχείου) για τους σταθμούς μέτρησης και του ρύπου. Η συγκεκριμένη παίρνει ως παραμέτρους:

- α) ένα `string` το οποίο αντιστοιχεί στο `id` του `tag` του αρχείου HTML,
- β) ένα `Array` που περιέχει τα αντικείμενα επιλογής,
- γ) την ιδιότητα του αντικειμένου που θα εμφανίζεται στο χρήστη,
- δ) την ιδιότητα του αντικειμένου με βάση την οποία θα ταυτοποιείται.

Παρακάτω φαίνεται η κλήση της συνάρτησης από το αρχείο JavaScript για επιλογή σταθμού και ρύπου.

```
fillsel("st", Stathmoi, "sta_name", "sta_id");
fillsel("ry", Rypoi, "ry_name", "id_rupoi");
```

Αντίστοιχα, έχουν δημιουργηθεί και οι συναρτήσεις `setdate` και `setdate2` οι οποίες εμφανίζουν προς επιλογή τα στοιχεία ημερομηνίας (ημέρα και μήνα η `setdate`, έτος η `setdate2`).

Όταν ο χρήστης επιλέξει κάποιο σταθμό ή ρύπο οι διαθέσιμες επιλογές του για ρύπο ή σταθμό αντίστοιχα περιορίζονται, δεδομένου ότι δεν μετρούνται όλοι οι ρύποι σε όλους τους σταθμούς. Έτσι η συνάρτηση `findOther` ενεργοποιείται κάθε φορά που γίνεται μία επιλογή σταθμού ή ρύπου ώστε να φιλτράρει τις διαθέσιμες επιλογές του άλλου αντικειμένου. Η συνάρτηση αυτή καλείται από το αρχείο HTML αφού δίνεται ως τιμή στην ιδιότητα `onchange` του `select tag`. Καλείται κάθε φορά που γίνεται μεταβολή από το χρήστη στον επιλεγμένο σταθμό ή ρύπο και ανάλογα περιορίζει τις επιλογές για ρύπο ή σταθμό. Με αυτό τον τρόπο εξασφαλίζεται πως ο χρήστης δεν θα κάνει επιλογές που δεν υποστηρίζονται από τη βάση δεδομένων. Η συνάρτηση καλείται με οχτώ παραμέτρους:

```
<select id="st" onchange="findOther(this.value, 'ry', Stathmoi, 'sta_id',
                                'pinRyp', Rypoi, 'id_rupoi', 'ry_name')">
</select>
```

- α) την επιλογή (`id`) του αντικειμένου που έγινε πριν κληθεί η συνάρτηση
- β) το `id` του `select` στο οποίο πρόκειται να περιοριστούν (φιλτραριστούν) οι διαθέσιμες επιλογές
- γ) το `Array` στο οποίο περιέχονται τα αντικείμενα από τα οποία επέλεξε ο χρήστης
- δ) την ιδιότητα του επιλεγμένου αντικειμένου με την οποία θα αναζητηθεί στο `Array`
- ε) το `Array` που περιέχεται στο επιλεγμένο αντικείμενο, με τις διαθέσιμες επιλογές για το άλλο
- ζ) το `Array` όπου περιέχονται τα αντικείμενα για τα οποία αναζητούνται οι διαθέσιμες επιλογές
- η) το `id` του αντικειμένου που αναζητείται
- θ) το όνομα της ιδιότητας που θα εμφανιστεί στο `select`

Η συνάρτηση απαιτεί μεγάλο πλήθος παραμέτρων ώστε να μπορεί να χρησιμοποιηθεί τόσο για το φιλτράρισμα των διαθέσιμων σταθμών ανάλογα τον επιλεγμένο ρύπο, όσο και για το φιλτράρισμα των διαθέσιμων ρύπων ανάλογα τον επιλεγμένο σταθμό. Ο μεγάλος αριθμός παραμέτρων καθιστά την συνάρτηση πιο ευέλικτη και πιο εύκολα επαναχρησιμοποιήσιμη.

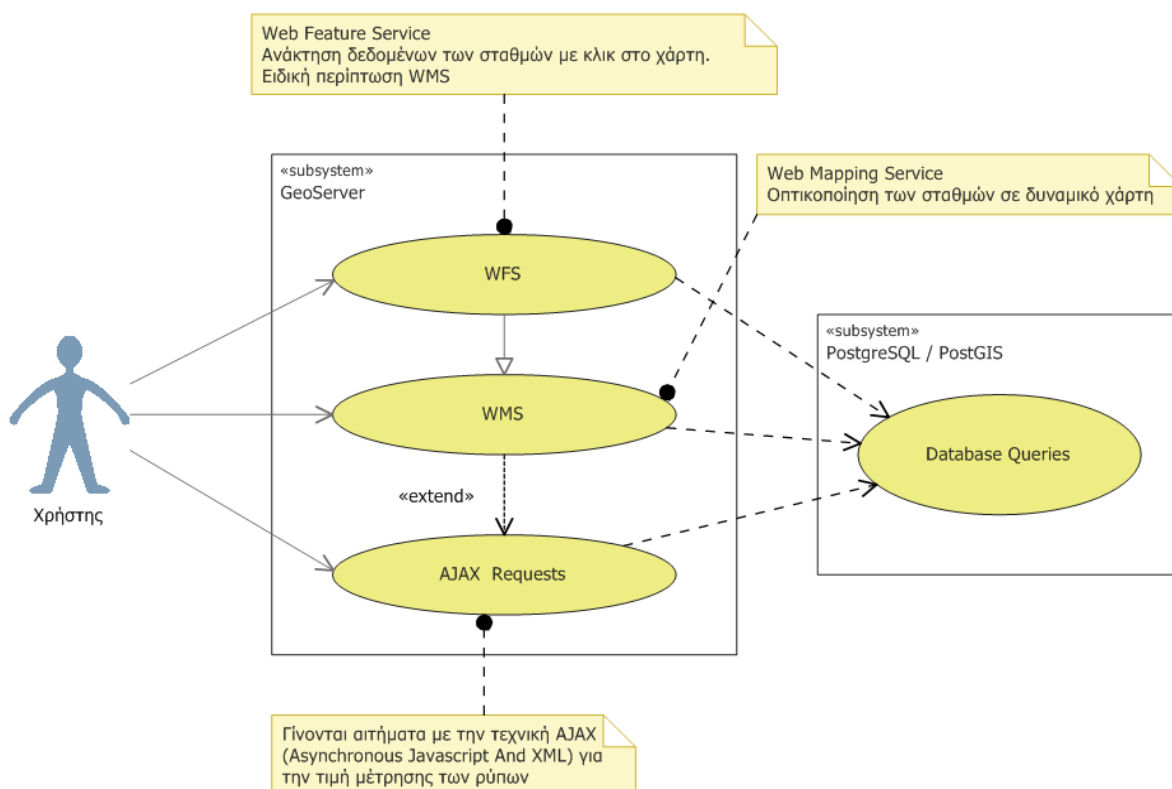
Η τελευταία βασική συνάρτηση της εφαρμογής είναι η `formSubmit()` και αυτό που κάνει είναι να συγκεντρώνει τις επιλογές του χρήστη για σταθμό, ρύπο, ημέρα, μήνα και έτος, στη συνέχεια να κάνει αίτημα στον GeoServer και αφού λάβει απόκριση να την εμφανίζει στο χρήστη. Πιο συγκεκριμένα, η `formSubmit` δίνεται ως ιδιότητα `onclick` στο κουμπί της φόρμας που πατάει ο χρήστης και ενεργοποιείται μόλις αυτός πατήσει το κουμπί.

```
<input type="button" onclick="formSubmit()" value="Submit form">
```

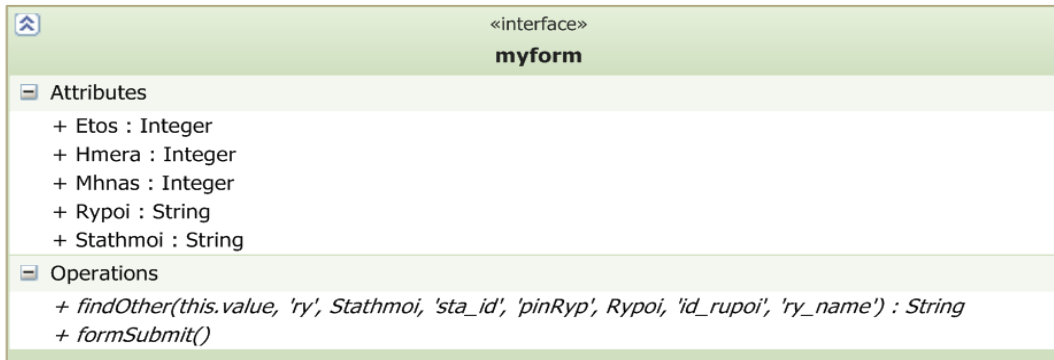
Τότε συγκεντρώνει όλες τις επιλεγμένες τιμές του χρήστη σχετικά με το σταθμό, το ρύπο και την ημερομηνία της μέτρησης και είτε κατασκευάζει το url με το οποίο θα αποστείλει το αίτημα στον GeoServer. Το αίτημα γίνεται στο layer `metrsh` και η απόκριση υφίσταται επεξεργασία ούτως ώστε να ενημερώσει το χρήστη για την τιμή μέτρησης του ρύπου ή διαφορετικά, σε περίπτωση που δεν έχει συμπληρωθεί σωστά η φόρμα αναζήτησης, για τα λάθη του χρήστη.

### 5.3 Διαγράμματα UML

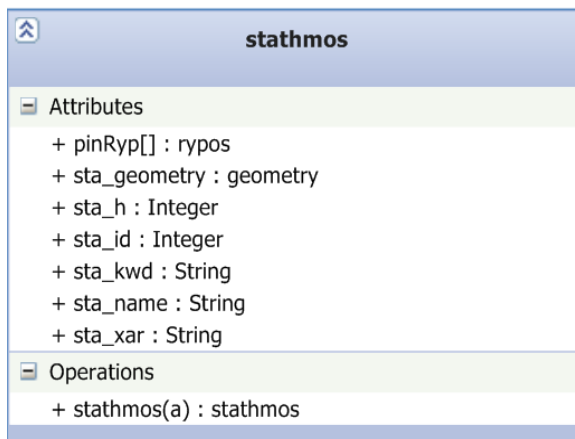
Παρακάτω φαίνονται τα διαγράμματα UML της εφαρμογής. Πιο συγκεκριμένα, παρουσιάζονται διαγράμματα των περιπτώσεων χρήσης, της διεπαφής (δηλαδή της μπάρας αναζήτησης), καθώς και των κλάσεων της εφαρμογής.



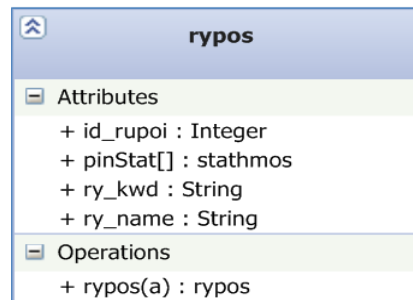
Εικόνα 53. Διάγραμμα περιπτώσεων χρήσης (Use Case Diagram)



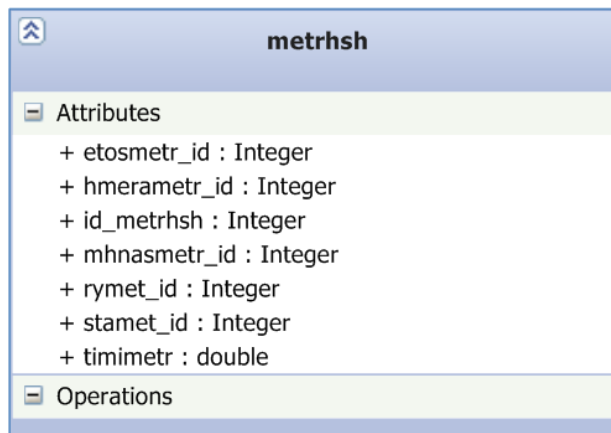
Εικόνα 54. Διάγραμμα της διεπαφής



Εικόνα 55. Διάγραμμα της κλάσης stathmos



Εικόνα 56. Διάγραμμα της κλάσης rypos



Εικόνα 57. Διάγραμμα της κλάσης metrhsh

Στα παραπάνω διαγράμματα φαίνεται καθαρά η δομή των αντικειμένων που χρησιμοποιούνται στην εφαρμογή, καθώς και η ιδιότητες με τις μεθόδους τους. Αντίστοιχα, για το interface φαίνονται τα πεδία που πρέπει να συμπληρώσει ο χρήστης μαζί με τις μεθόδους που υποστηρίζονται.

## 6. Συμπεράσματα

Η συγκεκριμένη εφαρμογή χρησιμοποιεί την τεχνική AJAX για να κάνει αιτήματα σε κάποια layer που έχουν γίνει publish από τον GeoServer και να ενημερώσει σχετικά το χρήστη. Τα layers προέρχονται από μια βάση δεδομένων PostgreSQL με την επέκταση PostGIS και αφορούν σε πίνακες που περιέχουν τόσο χωρική όσο και αμιγώς περιγραφική πληροφορία. Όλη η εφαρμογή αναπτύχθηκε με τη χρήση JavaScript και HTML5 ενώ όλα τα λογισμικά που αποτελούν μέρος της ή συνέβαλαν στην ανάπτυξη της είναι Ελεύθερα Λογισμικά.

Η ίδια εφαρμογή θα μπορούσε να κατασκευαστεί με τη χρήση άλλων γλωσσών ή χρησιμοποιώντας διαφορετική βάση δεδομένων ή λογισμικό Server. Έγινε επιλογή όμως να χρησιμοποιηθούν ελεύθερα λογισμικά, δωρεάν διαθέσιμα σε οποιονδήποτε ενδιαφερόμενο. Τα συγκεκριμένα, δεν υστερούν σε τίποτα από τα εμπορικά λογισμικά, ενώ υπόκεινται σε συνεχείς βελτιώσεις και αναβαθμίσεις από πολυπληθείς κοινότητες προγραμματιστών και χρηστών διεθνώς. Παράλληλα, το γεγονός ότι είναι επίσης Λογισμικά Ανοιχτού Κώδικα τα καθιστά πιο εύκολα αξιοποιήσιμα από τρίτες εφαρμογές.

Επίσης, θα μπορούσε η ίδια η εφαρμογή να λειτουργεί με διαφορετικό τρόπο, όπως για παράδειγμα να προτιμηθεί η λύση των λιγότερων αιτημάτων, χωρίς να γίνεται φιλτράρισμα των επιλογών του χρήστη. Η συνάρτηση `getData` αποστέλλει στον GeoServer μεγάλο πλήθος αιτημάτων με σκοπό να καταγράψει ποιοι σταθμοί μετρούν ποιους ρύπους, πληροφορία η οποία θα αξιοποιηθεί μετά από την συνάρτηση `findOther` για να περιορίσει ανάλογα τις διαθέσιμες επιλογές του. Αυτή η αντιμετώπιση προτιμήθηκε διότι περιορίζει δραστικά τις αποτυχημένες αναζητήσεις, δεδομένου κυρίως του μεγάλου και ασυνεπούς dataset. Στον αντίποδα όμως, αυτός ο τρόπος λειτουργίας με τον μεγάλο αριθμό αιτημάτων προς το GeoServer προκαλεί σχετική καθυστέρηση στην φόρτωση της ιστοσελίδας.

Κατά την ανάπτυξη του αλγόριθμου δόθηκε μεγάλο βάρος στην τμηματοποίηση του κώδικα, λόγω των πολλών πλεονεκτημάτων που παρουσιάζει. Αυτά αφορούν κυρίως στη διαχείριση του κώδικα JavaScript καθώς και στην επεκτασιμότητα του με μελλοντικές προσθήκες. Έτσι, θα ήταν σχετικά εύκολο να αναπτυχθεί μία συνάρτηση η οποία θα προσθέσει επιπλέον λειτουργικότητα στην εφαρμογή. Μία τέτοια θα μπορούσε να είναι η ενημέρωση της βάσης με επιπλέον δεδομένα, ώστε να διευρύνεται το dataset με την πάροδο του χρόνου. Αυτό θα ήταν δυνατόν να γίνει αξιοποιώντας περαιτέρω τις δυνατότητες του GeoServer σχετικά με τα transactions που υποστηρίζονται. Επίσης, με το Geolocation API της JavaScript θα μπορούσε να δώσει στους χρήστες πιο εξατομικευμένη πληροφόρηση, αφού θα μπορούσε να προσδιορίσει τη γεωγραφική τους θέση (χρησιμοποιώντας τις δυνατότητες των σύγχρονων web browsers) και να τους δώσει τα δεδομένα των εγγύτερων σταθμών.

Συμπερασματικά, οι τεχνολογίες που χρησιμοποιούνται σε αυτή την εφαρμογή αποτελούν κάποιες από τις πλέον σύγχρονες λύσεις για την δημιουργία διαδικτυακών εφαρμογών σχετικές με τη διάθεση δεδομένων και υπηρεσιών πάνω σε αυτά. Η αξία άλλωστε των δεδομένων συναρτάται με την πληροφορία που προκύπτει από την επεξεργασία τους, οπότε είναι σημαντικό να εξασφαλιστεί η απρόσκοπτη πρόσβαση κάθε ενδιαφερόμενου σε αυτά. Πολύ περισσότερο σήμερα που η δημόσια διοίκηση υποχρεώνεται από ευρωπαϊκές οδηγίες να διαθέσει ελεύθερα δεδομένα υπάρχει πλούσιο υλικό προς αξιοποίηση. Έτσι, είναι δυνατόν με κατάλληλη επεξεργασία των δεδομένων και την προσθήκη λειτουργιών και υπηρεσιών σε αυτά να δημιουργηθεί προστιθέμενη αξία καθιστώντας τα ακόμα πιο χρήσιμα.

## **7. Βιβλιογραφία – Πηγές**

Θεοδωρίδης Γ.,

Σημειώσεις σε ΓΕΩΓΡΑΦΙΚΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ (GIS), Πειραιάς 2012

Καμπάση Α.,

Σημειώσεις για το μάθημα ΠΕΡΙΒΑΛΛΟΝΤΙΚΗ ΠΛΗΡΟΦΟΡΙΚΗ, Πειραιάς 2008

Θεοδωρίδης Γ.,

Σημειώσεις ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ, Πειραιάς 2011

Βόντας Μ., Ταμπάκης Π.,

Εργαστηριακή Διάλεξη στα Γεωγραφικά Πληροφοριακά Συστήματα – PostGIS Πειραιάς 2012

Στεφανάκης Ε. 2009,

Τεχνολογίες Δημοσιοποίησης Χαρτογραφικού Περιεχομένου στον Παγκόσμιο Ιστό. Αθήνα, ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ

Μελάς Δ., Αλεξανδροπούλου Α., Αμοιρίδης Β., Κακαρίδου Μ., Σουλακέλλης Ν.,

ΑΤΜΟΣΦΑΙΡΙΚΗ ΡΥΠΑΝΣΗ [Οδηγός εκπαιδευτικών], ΑΘΗΝΑ 2000

Janowicz, K., Schade, S., Broring, A., Kezler, C., Maue, P., Stasch, C. (2010),

Semantic Enablement for Spatial Data Infrastructures

Επίσημος διαδικτυακός τόπος της Εταιρείας Ελεύθερου Λογισμικού / Λογισμικού Ανοιχτού Κώδικα (ΕΛ/ΛΑΚ),

<http://www.ellak.gr/>

Επίσημος διαδικτυακός τόπος OGC,

<http://www.opengeospatial.org/>

Επίσημος διαδικτυακός τόπος του OpenLayers,

<http://openlayers.org>

Επίσημος διαδικτυακός τόπος του World Wide Web Schools,

<http://www.w3schools.com>

Επίσημος διαδικτυακός τόπος για το Σύστημα Διαχείρισης Βάσεων Δεδομένων PostgreSQL,

<http://www.postgresql.org/>

Επίσημος διαδικτυακός τόπος των κειμένων τεκμηρίωσης της επέκτασης PostGIS,

<http://postgis.refractory.net/documentation/>

Επίσημος διαδικτυακός τόπος του GeoServer,  
<http://geoserver.org>

OpenLayers WFS-T using a GeoServer hosted PostGIS layer,  
<http://www.gistutor.com/openlayers/22-advanced-openlayers-tutorials/47-openlayers-wfs-t-using-a-geoserver-hosted-postgis-layer.html>

Επίσημος διαδικτυακός τόπος του Οργανισμού Κτηματολογίου και Χαρτογραφίσεων Ελλάδος,  
<http://www.okxe.gr>

Επίσημος διαδικτυακός τόπος της Εθνικής Υποδομής Γεωχωρικών Πληροφοριών (ΕΥΓΕΠ),  
<http://www.inspire.okxe.gr>

Επίσημος διαδικτυακός τόπος της Federal Geographic Data Committee των Ηνωμένων Πολιτειών Αμερικής,  
<http://www.fgdc.gov/>

Guidance Note 7-3: EPSG Registry Developers Guide,  
<http://www.epsg.org/guides/docs/G7-3.pdf>

epsg projection 2100 - ggrs87 / greek grid  
<http://spatialreference.org/ref/epsg/2100/>

epsg projection 4326 - wgs 84  
<http://spatialreference.org/ref/epsg/4326/>



## 8. Παράρτημα

### 8.1 Κώδικας του αρχείου AerioiRypoiProject.html

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Αέριοι Ρύποι</title>
    <link rel="stylesheet" href="AerioiRypoiProject.css">

    <!-- objectSorting script -->
    <script type="text/javascript" src="objSort.js"></script>

    <!-- OpenLayers -->
    <script src="http://openlayers.org/dev/OpenLayers.js"></script>

    <!-- OpenStreetMap base layer js -->
    <script
src="http://www.openstreetmap.org/openlayers/OpenStreetMap.js"></script>

    <!-- Google Maps -->
    <script
src="http://maps.google.com/maps/api/js?sensor=false"></script>

    <!-- myscript -->
    <script      src="AerioiRypoiProject.js" defer="defer"
type="text/javascript"></script>

  </head>
  <body onload="init()">

    <div id="my">
      <form name="myform">

        <label for="st">Σταθμός Μέτρησης</label>
        <select id="st"
onchange="findOther(this.value,'ry',Stathmoi,'sta_id','pinRyp',Rypoi,'id_rupo
i','ry_name')"></select>

        <label for="ry">Ρύπος</label>
        <select id="ry"
onchange="findOther(this.value,'st',Rypoi,'id_rupoi','pinStat',Stathmoi,'sta_
id','sta_name')"></select>

        <label for="hmera">Ημέρα</label>
        <select id="hmera"></select>

        <label for="mhnas">Μήνας</label>

```

```

        <select id="mhns"></select>

        <label for="etos">Έτος</label>
        <select id="etos"></select>

        <input type="button" onclick="formSubmit()"
value="Submit form">
    </form>
</div>

<div id="res">
</div>

<div id="toolbar" style="display: none;">
    <ul>
        <li>
            <a>WMS version:</a>
            <select id="wmsVersionSelector"
onChange="setWMSVersion(value)">
                <option value="1.1.1">1.1.1</option>
                <option value="1.3.0">1.3.0</option>
            </select>
        </li>
        <li>
            <a>Tiling:</a>
            <select id="tilingModeSelector"
onChange="setTileMode(value)">
                <option value="untiled">Single
tile</option>
                <option value="tiled">Tiled</option>
            </select>
        </li>
        <li>
            <a>Transition effect:</a>
            <select id="transitionEffectSelector"
onChange="setTransitionMode(value)">
                <option value="">None</option>
                <option value="resize">Resize</option>
            </select>
        </li>
        <li>
            <a>Antialias:</a>
            <select id="antialiasSelector"
onChange="setAntialiasMode(value)">
                <option value="full">Full</option>
                <option value="text">Text
only</option>
                <option value="none">Disabled</option>
            </select>
        </li>
        <li>
            <a>Format:</a>

```

```

                                <select id="imageFormatSelector"
onchange="setImageFormat(value)">
                                <option value="image/png">PNG
24bit</option>
                                <option value="image/png8">PNG
8bit</option>
                                <option value="image/gif">GIF</option>
                                <option id="jpeg"
value="image/jpeg">JPEG</option>
                                </select>
                                </li>
                                <li>
                                <a>Styles:</a>
                                <select id="imageFormatSelector"
onchange="setStyle(value)">
                                <option value="">Default</option>
                                </select>
                                </li>
                                <li>
                                <a>Width/Height:</a>
                                <select id="widthSelector"
onchange="setWidth(value)">
                                <option value="auto">Auto</option>
                                <option value="600">600</option>
                                <option value="750">750</option>
                                <option value="950">950</option>
                                <option value="1000">1000</option>
                                <option value="1200">1200</option>
                                <option value="1400">1400</option>
                                <option value="1600">1600</option>
                                <option value="1900">1900</option>
                                </select>
                                <select id="heightSelector"
onchange="setHeight(value)">
                                <option value="auto">Auto</option>
                                <option value="300">300</option>
                                <option value="400">400</option>
                                <option value="500">500</option>
                                <option value="600">600</option>
                                <option value="700">700</option>
                                <option value="800">800</option>
                                <option value="900">900</option>
                                <option value="1000">1000</option>
                                </select>
                                </li>
                                <li>
                                <a>Filter:</a>
                                <select id="filterType">
                                <option value="cql">CQL</option>
                                <option value="ogc">OGC</option>
                                <option value="fid">FeatureID</option>
                                </select>

```

```
                <input type="text" size="80" id="filter"/>
                
                
            </li>
        </ul>
    </div>

    <div id="map">
        
    </div>

    <div id="wrapper">
        <div id="location">
            location
        </div>
        <div id="lf">
            λ,φ :
        </div>
        <div id="scale"></div>
    </div>

    <div id="nodelist">
        <em>Click on the map to get feature info</em>
    </div>
</body>
</html>
```

## 8.2 Κώδικας του αρχείου AerioiRypoiProject.js

```

////////////////////////////////////
//                               global variables                               //
////////////////////////////////////
var map;
var untyped;
var tiled;
var pureCoverage = false;

var WGS84;
var WGS84_google_mercator;
var greekgrid;

var urlBase = "http://localhost:8080/geoserver/";
var layerNamespace = "WorkspaceAerioiRypoi";
var urlWfs = "/wfs?service=wfs&version=2.0.0";
var urlWfs1 = "/wfs?service=wfs&version=1.0.0";
var urlGetFeature = "&request=GetFeature";

var urlTypeNameRypoi = "&typeName=" + layerNamespace + ":rypoi";
var Rypoi = new Array();
var urlLayerRypoi = urlBase + layerNamespace + urlWfs + urlGetFeature +
urlTypeNameRypoi;

var urlTypeNameStathmoi = "&typeName=" + layerNamespace + ":stathmoi";
var Stathmoi = new Array();
var urlLayerStathmoi = urlBase + layerNamespace + urlWfs + urlGetFeature +
urlTypeNameStathmoi;

var urlTypeNameMetrhsh = "&typeName=" + layerNamespace + ":metrhsh";
var Metrhsh = new Array();
var urlLayerMetrhsh = urlBase + layerNamespace + urlWfs1 + urlGetFeature +
urlTypeNameMetrhsh;

OpenLayers.IMAGE_RELOAD_ATTEMPTS = 5;
OpenLayers.DOTS_PER_INCH = 25.4 / 0.28;

////////////////////////////////////
//                               ARXH                               //
////////////////////////////////////

function init() {

    format = 'image/png';
    if (pureCoverage) {
        document.getElementById('filterType').disabled = true;
        document.getElementById('filter').disabled = true;
        document.getElementById('antialiasSelector').disabled = true;
        document.getElementById('updateFilterButton').disabled = true;
        document.getElementById('resetFilterButton').disabled = true;
    }
}

```

```
        document.getElementById('jpeg').selected = true;
    }

    var bounds = new OpenLayers.Bounds(20.8499984741211, 35.3324966430664,
25.0800018310547, 40.7398452758789);
    var options = {
        controls : [],
        maxExtent : bounds,
        maxResolution : 0.0211224555969238,
        projection : "EPSG:4326",
        displayProjection : "EPSG:4326",
        units : 'degrees'
    };

    map = new OpenLayers.Map('map', options);

    // projections
    WGS84 = new OpenLayers.Projection("EPSG:4326");
    // WGS84 projection
    WGS84_google_mercator = new OpenLayers.Projection("EPSG:900913");
    // WGS84 Google projection

    //base layers
    var openstreetmap = new OpenLayers.Layer.OSM();
    var google_maps = new OpenLayers.Layer.Google("Google Maps", {
        numZoomLevels : 20
    });
    var google_satellite = new OpenLayers.Layer.Google("Google Satellite",
{
        type : google.maps.MapTypeId.SATELLITE,
        numZoomLevels : 20
    });

    tiled = new OpenLayers.Layer.WMS("Σταθμοί Μέτρησης Αέριων Ρύπων",
"http://localhost:8080/geoserver/" + layerNamespace + "/wms", {
        LAYERS : layerNamespace + ':stathmoi',
        STYLES : '',
        format : format,
        transparent : 'true',
        tiled : true,
        tilesOrigin : map.maxExtent.left + ',' + map.maxExtent.bottom
    }, {
        buffer : 0,
        displayOutsideMaxExtent : true,
        isBaseLayer : false,
        yx : {
            'EPSG:4326' : true
        }
    });

    map.addLayers([tiled, openstreetmap, google_maps, google_satellite]);

    map.addControl(new OpenLayers.Control.LayerSwitcher());
```

```

map.addControl(new OpenLayers.Control.PanZoomBar({
    position : new OpenLayers.Pixel(2, 15)
}));
map.addControl(new OpenLayers.Control.Navigation());
map.addControl(new OpenLayers.Control.Scale($('scale')));
map.addControl(new OpenLayers.Control.MousePosition({
    element : $('location')
}));

var options = document.getElementById("options");
options.onclick = toggleControlPanel;

// GetFeatureInfo
map.events.register('click', map, function(e) {
    document.getElementById('nodelist').innerHTML = "Loading...
please wait...";
    var params = {
        REQUEST : "GetFeatureInfo",
        EXCEPTIONS : "application/vnd.ogc.se_xml",
        BBOX : map.getExtent().toBBOX(),
        SERVICE : "WMS",
        INFO_FORMAT : 'text/html',
        QUERY_LAYERS : map.layers[0].params.LAYERS,
        FEATURE_COUNT : 50,
        Layers : layerNamespace + ':stathmoi',
        WIDTH : map.size.w,
        HEIGHT : map.size.h,
        format : format,
        styles : map.layers[0].params.STYLES,
        srs : map.layers[0].params.SRS
    };

    // wms 1.3 h wms 1.1
    if (map.layers[0].params.VERSION == "1.3.0") {
        params.version = "1.3.0";
        params.j = parseInt(e.xy.x);
        params.i = parseInt(e.xy.y);
    } else {
        params.version = "1.1.1";
        params.x = parseInt(e.xy.x);
        params.y = parseInt(e.xy.y);
    }
    if (map.layers[0].params.CQL_FILTER != null) {
        params.cql_filter = map.layers[0].params.CQL_FILTER;
    }
    if (map.layers[0].params.FILTER != null) {
        params.filter = map.layers[0].params.FILTER;
    }
    if (map.layers[0].params.FEATUREID) {
        params.featureid = map.layers[0].params.FEATUREID;
    }
    OpenLayers.loadURL("http://localhost:8080/geoserver/"
    layerNamespace + "/wms", params, this, setHTML, setHTML);
}

```

```

        OpenLayers.Event.stop(e);
    });

    var mapextent = new OpenLayers.Bounds(20.8499984741211,
35.3324966430664, 25.0800018310547, 40.7398452758789).transform(WGS84,
WGS84_google_mercator);
    map.zoomToExtent(mapextent);

    getdata(urlLayerRypoi, Rypoi, rypos);
    Rypoi = Rypoi.objSort("id_rupoi");

    getdata(urlLayerStathmoi, Stathmoi, stathmos);
    Stathmoi = Stathmoi.objSort("sta_id");

    // Afairountai oi 3 stathmoi apo tous opoious de metroutai rypoi
    Stathmoi.splice(14, 1);
    Stathmoi.splice(25, 1);
    Stathmoi.splice(29, 1);

    getmetrhsh(Stathmoi, Rypoi);

    fillsel("st", Stathmoi, "sta_name", "sta_id");
    fillsel("ry", Rypoi, "ry_name", "id_rupoi");
    setdate("hmera", 31);
    setdate("mhnas", 12);
    setdate2("etos", 12);

} //end_init

//////////////////////////////////////
//                               functions                               //
//////////////////////////////////////

function rypos(a) {
    this.id_rupoi = a[0];
    this.ry_kwd = a[1];
    this.ry_name = a[2];
    this.pinStat = new Array();
}

function stathmos(a) {
    this.sta_id = a[0];
    this.sta_name = a[1];
    this.sta_geometry = a[2];
    this.sta_h = a[3];
    this.sta_xar = a[4];
    this.sta_kwd = a[5];
    this.pinRyp = new Array();
}

function metrhsh(a) {
    this.id_metrhsh = a[0];
    this.timimetr = a[1];
}

```



```

        this.stamet_id = a[2];
        this.rymet_id = a[3];
        this.etosmetr_id = a[4];
        this.mhnasmetr_id = a[5];
        this.hmerametr_id = a[6];
    }

    ///////////////////////////////////////////////////////////////////

function getdata(url, obj, func) {
    var pinakas = new Array();
    var request = new XMLHttpRequest();

    request.open("GET", url, false);
    request.onreadystatechange = function() {
        if (request.readyState == 4 && request.status == 200) {
            var tab = new Array();
            var metr = 0;
            var pxx = new func("");
            for (var property in pxx) {
                tab[metr] = property;
                metr++;
            }
            metr = null;
            pxx = null;
            var xmlDoc = request.responseXML;
            for (var i = 0; i < tab.length; i++) {
                pinakas[i] = dataprocess(xmlDoc, tab[i]);
            } //for loop
            for (var mg = 0; mg < pinakas[0].length; mg++)
                { //Ftiaxnontai antikeimena
                    var r = new Array();
                    for (var ms = 0; ms < tab.length; ms++) {
                        r[ms] = pinakas[ms][mg];
                    }
                    var ob = new func(r);
                    obj.push(ob);
                }
            } //end if ola ok
        };
        //ENDrequest.onload
        request.send(null);
    } //endfunction getdata

    ///////////////////////////////////////////////////////////////////

function dataprocess(xmlDoc, st) {
    var stfull = layerNamespace + ":" + st;
    var str1 = "&typeName=" + stfull;
    var xstr = new Array();
    var x = xmlDoc.getElementsByTagName(stfull);
    if (x.length == 0) {
        x = xmlDoc.getElementsByTagName(st);
    }
}

```



```

function fillsel(selname, Arr, s1, s2) {
    var myselect = document.getElementById(selname);
    myselect.options.length = 0;
    myselect.options[0] = new Option("", "", false, false);
    for (var i = 0; i < Arr.length; i++) {
        var obj = Arr[i];
        text = obj[s1];
        val = obj[s2];
        myselect.options[i + 1] = new Option(text, val, false, false);
    }
}

} //END fillsel

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

function setdate(selname, maxAr) { //fill mera,mhna

    var myselect = document.getElementById(selname);
    myselect.options[0] = new Option("", "", true, false);
    for (var i = 1; i <= maxAr; i++) {
        var ar = i;
        myselect.options[i] = new Option(ar, ar, false, false);
    }
}

} //END setdate

function setdate2(selname, maxAr) { //fill year

    var myselect = document.getElementById(selname);
    myselect.options[0] = new Option("", "", true, false);
    for (var i = 1; i <= maxAr; i++) {
        var ar = i + 1999;
        myselect.options[i] = new Option(ar, ar, false, false);
    }
}

} //END setdate2

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

function cleanStathmous(Arr, str) {
    var Stathmoi2 = new Array();
    var ii = 0;
    for (var i = 0; i < Arr.length; i++) {
        if (Stathmoi[i][str].length != 0) {
            Stathmoi2[ii] = Arr[i];
            ii++;
        }
    }
    return Stathmoi2;
}

} //END cleanStathmous

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```



```

////////////////////////////////////
function formSubmit() {

    var selSt = document.getElementById("st");
    var OpSt = selSt.options[selSt.selectedIndex].value;
    var SelRy = document.getElementById("ry");
    var OpRy = SelRy.options[SelRy.selectedIndex].value;
    var SelHmera = document.getElementById("hmera");
    var OpHm = SelHmera.options[SelHmera.selectedIndex].value;
    var SelMhnas = document.getElementById("mhnas");
    var OpMh = SelMhnas.options[SelMhnas.selectedIndex].value;
    var SelYear = document.getElementById("etos");
    var OpYe = SelYear.options[SelYear.selectedIndex].value;

    if (OpSt != "" && OpRy != "" && OpHm != "" && OpMh != "" && OpYe !=
"" ) {

        var filtro = "&cql_filter=stamet_id=" + OpSt + " AND rymet_id="
+ OpRy + " AND etosmetr_id=" + OpYe + " AND mhnasmetr_id=" + OpMh + " AND
hmerametr_id=" + OpHm + "&maxFeatures=1";
        var url = urlLayerMetrhsh + filtro;

        var request = new XMLHttpRequest();
        request.open("GET", url, false);
        request.onreadystatechange = function() {
            if (request.readyState == 4 && request.status == 200) {
                var xmlDoc = request.responseXML;
                var metrhsh = dataprocess(xmlDoc, "timimetr");

                //alert("metrhsh = "+ metrhsh);
                var res = document.getElementById("res");

                if (metrhsh == "") {
                    alert("Δεν υπάρχει μέτρηση τη συγκεκριμένη
ημερομηνία για αυτό το ρύπο από αυτό το σταθμό \n\n Δοκιμάστε άλλη
ημερομηνία");
                    res.innerHTML = "Δεν υπάρχει διαθέσιμη
μέτρηση ";
                } else {
                    res.innerHTML = " Μέτρηση = <b>" + metrhsh +
"</b>";
                }
            }
        } //end if
    } //end func
    request.send(null);

    var request = new XMLHttpRequest();
    var filtro = "&cql_filter=sta_id=" + OpSt;
    var url1 = urlLayerStathmoi + filtro;
    request.open("GET", url1, false);

```

```

        request.onreadystatechange = function() {
            if (request.readyState == 4 && request.status == 200) {
                var xmlDoc = request.responseXML;
                var sunt = dataprocess(xmlDoc, "gml:pos");
                if (sunt.length == 0) {
                    sunt = dataprocess(xmlDoc, "pos");
                }

                sunt1 = JSON.stringify(sunt);
                sunt = sunt1.split(" ");

                f = sunt[0];
                l = sunt[1];
                f = f.substring(2);
                l = l.substring(0, l.length - 2);

                sunt1 = new OpenLayers.LonLat(1, f);

                map.setCenter(sunt1.transform(WGS84,
WGS84_google_mercator), 15, true, true);

            }
        }
        request.send(null);

    } //endif ok
    else {
        var keimeno = "Δεν δώσατε τα απαραίτητα δεδομένα - Παρακαλώ
συμπληρώστε: \n";

        if (OpSt == "") {
            keimeno = keimeno + "\n Όνομα Σταθμού "
        }
        if (OpRy == "") {
            keimeno = keimeno + "\n Όνομα Ρύπου "
        }
        if (OpHm == "") {
            keimeno = keimeno + "\n Ημέρα "
        }
        if (OpMh == "") {
            keimeno = keimeno + "\n Μήνα "
        }
        if (OpYe == "") {
            keimeno = keimeno + "\n Έτος "
        }
        alert(keimeno);
    }

} //END formSubmit

////////////////////////////////////

```

```
////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////
function setHTML(response) {
    document.getElementById('nodelist').innerHTML = response.responseText;
};

////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////

function toggleControlPanel(event) {
    var toolbar = document.getElementById("toolbar");
    if (toolbar.style.display == "none") {
        toolbar.style.display = "block";
    } else {
        toolbar.style.display = "none";
    }
    event.stopPropagation();
    map.updateSize()
}

////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////

function setTileMode(tilingMode) {
    if (tilingMode == 'tiled') {
        untiled.setVisibility(false);
        tiled.setVisibility(true);
        map.setBaseLayer(tiled);
    } else {
        untiled.setVisibility(true);
        tiled.setVisibility(false);
        map.setBaseLayer(untiled);
    }
}

////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////

function setTransitionMode(transitionEffect) {
    if (transitionEffect === 'resize') {
        tiled.transitionEffect = transitionEffect;
        untiled.transitionEffect = transitionEffect;
    } else {
        tiled.transitionEffect = null;
        untiled.transitionEffect = null;
    }
}

////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////

function setImageFormat(mime) {
    // we may be switching format on setup
    if (tiled == null)
        return;

    tiled.mergeNewParams({
```

```

        format : mime
    });
    untiled.mergeNewParams({
        format : mime
    });
}

////////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////////

function setStyle(style) {

    if (tiled == null)
        return;

    tiled.mergeNewParams({
        styles : style
    });
    untiled.mergeNewParams({
        styles : style
    });
}

////////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////////

function setWMSVersion(wmsVersion) {
    // we may be switching style on setup
    if (wmsVersion == null)
        return;

    if (wmsVersion == "1.3.0") {
        origin = map.maxExtent.bottom + ',' + map.maxExtent.left;
    } else {
        origin = map.maxExtent.left + ',' + map.maxExtent.bottom;
    }

    tiled.mergeNewParams({
        version : wmsVersion,
        tilesOrigin : origin
    });
    untiled.mergeNewParams({
        version : wmsVersion
    });
}

////////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////////

function setAntialiasMode(mode) {
    tiled.mergeNewParams({
        format_options : 'antialias:' + mode
    });
    untiled.mergeNewParams({
        format_options : 'antialias:' + mode
    });
}

```



```
}  
  
////////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////////  
  
function setPalette(mode) {  
    if (mode == '') {  
        tiled.mergeNewParams({  
            palette : null  
        });  
        untiled.mergeNewParams({  
            palette : null  
        });  
    } else {  
        tiled.mergeNewParams({  
            palette : mode  
        });  
        untiled.mergeNewParams({  
            palette : mode  
        });  
    }  
}  
  
////////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////////  
  
function setWidth(size) {  
    var mapDiv = document.getElementById('map');  
    var wrapper = document.getElementById('wrapper');  
  
    if (size == "auto") {  
        mapDiv.style.width = null;  
        wrapper.style.width = null;  
    } else {  
        mapDiv.style.width = size + "px";  
        wrapper.style.width = size + "px";  
    }  
    map.updateSize();  
}  
  
////////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////////  
  
function setHeight(size) {  
    var mapDiv = document.getElementById('map');  
  
    if (size == "auto") {  
        mapDiv.style.height = null;  
    } else {  
        mapDiv.style.height = size + "px";  
    }  
    map.updateSize();  
}  
  
////////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////////
```

```
function updateFilter() {
    if (pureCoverage)
        return;

    var filterType = document.getElementById('filterType').value;
    var filter = document.getElementById('filter').value;

    var filterParams = {
        filter : null,
        cql_filter : null,
        featureId : null
    };
    if (OpenLayers.String.trim(filter) != "") {
        if (filterType == "cql")
            filterParams["cql_filter"] = filter;
        if (filterType == "ogc")
            filterParams["filter"] = filter;
        if (filterType == "fid")
            filterParams["featureId"] = filter;
    }
    mergeNewParams(filterParams);
}

////////////////////////////////// Συνάρτηση από το GeoServer ////////////////////////////////////

function resetFilter() {
    if (pureCoverage)
        return;

    document.getElementById('filter').value = "";
    updateFilter();
}

function mergeNewParams(params) {
    tiled.mergeNewParams(params);
    untiled.mergeNewParams(params);
}
```

### 8.3 Κώδικας του αρχείου AerioiRypoiProject.css

```
/* General settings */
body {
    font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
    font-size: small;
}
/* Toolbar styles */
#toolbar {
    position: relative;
    padding-bottom: 0.5em;
    display: none;
}

#toolbar ul {
    list-style: none;
    padding: 0;
    margin: 0;
}

#toolbar ul li {
    float: left;
    padding-right: 1em;
    padding-bottom: 0.5em;
}

#toolbar ul li a {
    font-weight: bold;
    font-size: smaller;
    vertical-align: middle;
    color: black;
    text-decoration: none;
}

#toolbar ul li a:hover {
    text-decoration: underline;
}

#toolbar ul li * {
    vertical-align: middle;
}

/* The map and the location bar */
#map {
    clear: both;
    position: relative;
    width: 600px;
    height: 512px;
    border: 1px solid black;
}
```

```

#wrapper {
    width: 400px;
}

#location {
    float: right;
}

#options {
    position: absolute;
    left: 13px;
    top: 7px;
    z-index: 3000;
}
/* ----- */
.olControllayerSwitcher {
    position: absolute;
    top: 25px;
    right: 0;
    width: 20em;
    font-family: sans-serif;
    font-weight: bold;
    margin-top: 3px;
    margin-left: 3px;
    margin-bottom: 3px;
    font-size: smaller;
    color: white;
    background-color: transparent;
}

.olControllayerSwitcher .layersDiv {
    padding-top: 5px;
    padding-left: 10px;
    padding-bottom: 5px;
    padding-right: 10px;
    /*background-color: "00BFFF";*/
}

.olControllayerSwitcher .layersDiv .baseLbl, .olControllayerSwitcher
.layersDiv .dataLbl {
    margin-top: 3px;
    margin-left: 3px;
    margin-bottom: 3px;
}

.olControllayerSwitcher .layersDiv .baseLayersDiv, .olControllayerSwitcher
.layersDiv .dataLayersDiv {
    padding-left: 10px;
}

.olControllayerSwitcher .maximizeDiv, .olControllayerSwitcher .minimizeDiv {
    width: 18px;
    height: 18px;
}

```

```
        top: 5px;
        right: 0;
        cursor: pointer;
    }
    /* ----- */
    table.featureInfo, table.featureInfo td, table.featureInfo th {
        border: 1px solid #ddd;
        border-collapse: collapse;
        margin: 0;
        padding: 0;
        font-size: 90%;
        padding: .2em .1em;
    }
    table.featureInfo th {
        padding: .2em .2em;
        font-weight: bold;
        background: #eee;
    }
    table.featureInfo td {
        background: #fff;
    }
    table.featureInfo tr.odd td {
        background: #eee;
    }
    table.featureInfo caption {
        text-align: left;
        font-size: 100%;
        font-weight: bold;
        padding: .2em .2em;
    }
    #lf {
        float: right;
        margin-right: 10px;
    }
    #myform{
        font-size: 100%;
        font-weight: bold;
        padding: .2em .2em;
        border: 13px solid #ddd;
        display: block;
    }
    div.lst {
        float: left;
        margin: 0 5px 0 5px;
    }
    #my{
        display: block;
    }
}
```

## 8.4 Κώδικας του αρχείου objSort.js

```

/*
objSort v 1.1
copyright 2006 Thomas Frank

This program is free software under the terms of the
GNU General Public License version 2 as published by the Free
Software Foundation. It is distributed without any warranty.
*/

tfObjSort={
  init:function(){
    Array.prototype.objSort=function(){
      tfObjSort.setThings(this);
      var a=arguments;
      var x=tfObjSort;
      x.a=[];x.d=[];
      for(var i=0;i<a.length;i++){
        if(typeof
a[i]=="string"){x.a.push(a[i]);x.d.push(1)};
        if(a[i]===-1){x.d[x.d.length-1]=-1}
      }
      return this.sort(tfObjSort.sorter);
    };
    Array.prototype.strSort=function(){
      tfObjSort.setThings(this);
      return this.sort(tfObjSort.charSorter)
    }
  },
  sorter:function(x,y){
    var a=tfObjSort.a
    var d=tfObjSort.d
    var r=0
    for(var i=0;i<a.length;i++){
      if(typeof x+typeof y!="objectobject"){return typeof
x=="object"?-1:1};
      var m=x[a[i]]; var n=y[a[i]];
      var t=typeof m+typeof n;
      if(t=="booleanboolean"){m*=-1;n*=-1}
      else
if(t.split("string").join("").split("number").join("")!=""){continue};
      r=m-n;
      if(isNaN(r)){r=tfObjSort.charSorter(m,n)};
      if(r!=0){return r*d[i]}
    }
    return r
  },
  charSorter:function(x,y){
    if(tfObjSort.ignoreCase){x=x.toLowerCase();y=y.toLowerCase()};
    var s=tfObjSort.chars;

```

```
        if(!s){return x>y?1:x<y?-1:0};

    x=x.split("");y=y.split("");l=x.length>y.length?y.length:x.length;
    var p=0;
    for(var i=0;i<l;i++){
        p=s.indexOf(x[i])-s.indexOf(y[i]);
        if(p!=0){break};
    };
    if(p==0){p=x.length-y.length};
    return p
},
setThings:function(x){
    this.ignoreCase=x.sortIgnoreCase;
    var s=x.sortCharOrder;
    if(!s){this.chars=false;return true};
    if(!s.sort){s=s.split(",");};
    var a="";
    for(var i=1;i<1024;i++){a+=String.fromCharCode(i)};
    for(var i=0;i<s.length;i++){
        z=s[i].split("");
        var m=z[0]; var n=z[1]; var o="";
        if(z[2]=="_"){o=n+m} else {o=m+n};
        a=a.split(m).join("").split(n).join(o);
    };
    this.chars=a
}
};
tfObjSort.init();
```

**8.5 Πίνακας αρχείων πρωτογενών δεδομένων**

BenzIOA2008.dat	NO#ARI2009.dat	NO#SMY2000.dat	NO2Oin2011.dat	O3#ELE2004.dat	SmokePAT2001.dat
BenzIOA2009.dat	NO#ARI2010.dat	NO#SMY2001.dat	NO2PAI2001.dat	O3#ELE2005.dat	SmokePAT2002.dat
BenzIOA2010.dat	NO#ARI2011.dat	NO#SMY2002.dat	NO2PAI2002.dat	O3#ELE2006.dat	SmokePAT2003.xls
BenzIOA2011.dat	NO#ATH2000.dat	NO#SMY2003.dat	NO2PAI2003.dat	O3#ELE2007.dat	SmokePAT2004.dat
BenzPAT2004.dat	NO#ATH2001.dat	NO#SMY2004.dat	NO2PAI2004.dat	O3#ELE2008.dat	SmokePAT2005.dat
BenzPAT2005.dat	NO#ATH2002.dat	NO#SMY2005.dat	NO2PAI2007.dat	O3#ELE2009.dat	SmokePER2000.dat
BenzPAT2006.dat	NO#ATH2003.dat	NO#SMY2006.dat	NO2PAI2008.dat	O3#ELE2010.dat	SmokePER2001.dat
BenzPAT2007.dat	NO#ATH2004.dat	NO#SMY2007.dat	NO2PAI2009.dat	O3#ELE2011.dat	SmokePER2002.dat
BenzPAT2009.dat	NO#ATH2005.dat	NO#SMY2008.dat	NO2PAI2010.dat	O3#GAL2000.dat	SmokePER2003.xls
BenzPAT2010.dat	NO#ATH2006.dat	NO#SMY2009.dat	NO2PAI2011.dat	O3#GAL2001.dat	SmokePER2004.dat
BenzPAT2011.dat	NO#ATH2007.dat	NO#SMY2010.dat	NO2PAI2001.dat	O3#GAL2002.dat	SmokeSMY2000.dat
CO#AGS2001.dat	NO#ATH2008.dat	NO#SMY2011.dat	NO2PAI2002.dat	O3#GAL2003.dat	SmokeSMY2001.dat
CO#AGS2002.dat	NO#ATH2009.dat	NO#THR2000.dat	NO2PAI2007.dat	O3#GAL2004.dat	SmokeSMY2002.dat
CO#AGS2003.dat	NO#ATH2010.dat	NO#THR2001.dat	NO2PAI2008.dat	O3#GAL2005.dat	SmokeSMY2003.xls
CO#AGS2007.dat	NO#ATH2011.dat	NO#THR2002.dat	NO2PAI2009.dat	O3#GEO2000.dat	SmokeSMY2004.dat
CO#AGS2008.dat	NO#BIO2000.dat	NO#THR2003.dat	NO2PAI2010.dat	O3#GEO2001.dat	SO2AGP2000.dat
CO#AGS2009.dat	NO#BIO2001.dat	NO#THR2004.dat	NO2PAN2000.dat	O3#GEO2002.dat	SO2AGP2001.dat
CO#AGS2010.dat	NO#BIO2002.dat	NO#THR2005.dat	NO2PAN2001.dat	O3#GEO2003.dat	SO2AGP2002.dat
CO#AGS2011.dat	NO#BIO2003.dat	NO#THR2006.dat	NO2PAN2002.dat	O3#GEO2004.dat	SO2AGP2003.dat
CO#ATH2000.dat	NO#BIO2004.dat	NO#THR2007.dat	NO2PAN2003.dat	O3#GEO2005.dat	SO2AGP2004.dat
CO#ATH2001.dat	NO#BIO2005.dat	NO#THR2008.dat	NO2PAN2004.dat	O3#GEO2006.dat	SO2AGP2005.dat
CO#ATH2002.dat	NO#BIO2006.dat	NO#THR2009.dat	NO2PAN2005.dat	O3#GEO2007.dat	SO2AGS2001.dat
CO#ATH2003.dat	NO#BIO2007.dat	NO#THR2010.dat	NO2PAN2006.dat	O3#GEO2008.dat	SO2AGS2002.dat
CO#ATH2004.dat	NO#ELE2000.dat	NO#THR2011.dat	NO2PAN2007.dat	O3#GEO2009.dat	SO2AGS2003.dat
CO#ATH2005.dat	NO#ELE2001.dat	NO#VOL2001.dat	NO2PAO2001.dat	O3#GEO2010.dat	SO2AGS2004.dat
CO#ATH2006.dat	NO#ELE2002.dat	NO#VOL2002.dat	NO2PAO2002.dat	O3#GEO2011.dat	SO2AGS2008.dat
CO#ATH2007.dat	NO#ELE2003.dat	NO#VOL2003.dat	NO2PAO2003.dat	O3#IOA2009.dat	SO2ALI2001.dat



CO#ATH2008.dat	NO#ELE2004.dat	NO#VOL2004.dat	NO2PAO2007.dat	O3#IOA2010.dat	SO2ALI2002.dat
CO#ATH2009.dat	NO#ELE2005.dat	NO#VOL2005.dat	NO2PAO2008.dat	O3#IOA2011.dat	SO2ALI2006.dat
CO#ATH2010.dat	NO#ELE2006.dat	NO#VOL2006.dat	NO2PAO2009.dat	O3#IRA2001.dat	SO2ALI2007.dat
CO#ATH2011.dat	NO#ELE2007.dat	NO2AGP2000.dat	NO2PAO2010.dat	O3#IRA2002.dat	SO2ALI2008.dat
CO#BIO2007.dat	NO#ELE2008.dat	NO2AGP2001.dat	NO2PAO2011.dat	O3#IRA2003.dat	SO2ALI2009.dat
CO#GEO2000.dat	NO#ELE2009.dat	NO2AGP2002.dat	NO2PAT2000.dat	O3#KAL2001.dat	SO2ALI2010.dat
CO#GEO2001.dat	NO#ELE2010.dat	NO2AGP2003.dat	NO2PAT2001.dat	O3#KAL2002.dat	SO2ALI2011.dat
CO#GEO2002.dat	NO#ELE2011.dat	NO2AGP2004.dat	NO2PAT2002.dat	O3#KAL2003.dat	SO2APT2001.dat
CO#GEO2003.dat	NO#GAL2000.dat	NO2AGP2005.dat	NO2PAT2003.dat	O3#KAL2004.dat	SO2APT2002.dat
CO#GEO2004.dat	NO#GAL2001.dat	NO2AGP2006.dat	NO2PAT2004.dat	O3#KAL2005.dat	SO2APT2003.dat
CO#GEO2005.dat	NO#GAL2002.dat	NO2AGP2007.dat	NO2PAT2005.dat	O3#KOD2001.dat	SO2APT2004.dat
CO#GEO2006.dat	NO#GAL2003.dat	NO2AGP2008.dat	NO2PAT2006.dat	O3#KOD2002.dat	SO2APT2007.dat
CO#GEO2007.dat	NO#GAL2004.dat	NO2AGP2009.dat	NO2PAT2007.dat	O3#KOD2003.dat	SO2APT2008.dat
CO#GEO2008.dat	NO#GAL2005.dat	NO2AGP2010.dat	NO2PAT2008.dat	O3#KOD2006.dat	SO2APT2009.dat
CO#GEO2009.dat	NO#GEO2000.dat	NO2AGP2011.dat	NO2PAT2009.dat	O3#KOD2007.dat	SO2ARI2000.dat
CO#GEO2010.dat	NO#GEO2001.dat	NO2AGS2001.dat	NO2PAT2010.dat	O3#KOD2008.dat	SO2ARI2001.dat
CO#GEO2011.dat	NO#GEO2002.dat	NO2AGS2002.dat	NO2PAT2011.dat	O3#KOD2009.dat	SO2ARI2002.dat
CO#IRA2001.dat	NO#GEO2003.dat	NO2AGS2003.dat	NO2PER2000.dat	O3#KOD2010.dat	SO2ARI2003.dat
CO#IRA2002.dat	NO#GEO2004.dat	NO2AGS2004.dat	NO2PER2001.dat	O3#KOD2011.dat	SO2ARI2004.dat
CO#IRA2003.dat	NO#GEO2005.dat	NO2AGS2007.dat	NO2PER2002.dat	O3#kor2008.dat	SO2ARI2005.dat
CO#KAL2001.dat	NO#GEO2006.dat	NO2AGS2008.dat	NO2PER2003.dat	O3#kor2009.dat	SO2ATH2000.dat
CO#KAL2002.dat	NO#GEO2007.dat	NO2AGS2009.dat	NO2PER2004.dat	O3#kor2010.dat	SO2ATH2001.dat
CO#KAL2003.dat	NO#GEO2008.dat	NO2AGS2010.dat	NO2PER2005.dat	O3#kor2011.dat	SO2ATH2002.dat
CO#KAL2004.dat	NO#GEO2009.dat	NO2AGS2011.dat	NO2PER2006.dat	O3#LAR2001.dat	SO2ATH2003.dat
CO#KAL2005.dat	NO#GEO2010.dat	NO2ALI2001.dat	NO2PER2007.dat	O3#LAR2002.dat	SO2ATH2004.dat
CO#KAL2008.dat	NO#GEO2011.dat	NO2ALI2005.dat	NO2PER2008.dat	O3#LAR2007.dat	SO2ATH2005.dat
CO#KAL2009.dat	NO#GOU2000.dat	NO2ALI2006.dat	NO2PER2009.dat	O3#LAR2008.dat	SO2ATH2006.dat
CO#KAL2010.dat	NO#GOU2001.dat	NO2ALI2007.dat	NO2PER2010.dat	O3#LAR2009.dat	SO2ATH2007.dat

CO#KAL2011.dat	NO#GOU2002.dat	NO2ALI2008.dat	NO2PER2011.dat	O3#LAR2011.dat	SO2ATH2008.dat
CO#KOD2001.dat	NO#GOU2003.dat	NO2ALI2009.dat	NO2PIR2000.dat	O3#LIO2000.dat	SO2ATH2009.dat
CO#KOD2002.dat	NO#GOU2004.dat	NO2ALI2010.dat	NO2PIR2001.dat	O3#LIO2001.dat	SO2ATH2010.dat
CO#KOD2003.dat	NO#GOU2005.dat	NO2ALI2011.dat	NO2PIR2002.dat	O3#LIO2002.dat	SO2ATH2011.dat
CO#KOD2004.dat	NO#GOU2006.dat	NO2APT2001.dat	NO2PIR2003.dat	O3#LIO2003.dat	SO2BIO2000.dat
CO#KOD2005.dat	NO#GOU2007.dat	NO2APT2002.dat	NO2PIR2004.dat	O3#LIO2004.dat	SO2BIO2001.dat
CO#KOD2008.dat	NO#GOU2008.dat	NO2APT2003.dat	NO2PIR2005.dat	O3#LIO2005.dat	SO2BIO2002.dat
CO#KOD2009.dat	NO#GOU2009.dat	NO2APT2006.dat	NO2PIR2006.dat	O3#LIO2006.dat	SO2BIO2003.dat
CO#KOD2010.dat	NO#IOA2008.dat	NO2APT2007.dat	NO2PIR2007.dat	O3#LIO2007.dat	SO2BIO2004.dat
CO#KOD2011.dat	NO#IOA2009.dat	NO2APT2008.dat	NO2PIR2008.dat	O3#LIO2008.dat	SO2BIO2005.dat
CO#LAR2001.dat	NO#IOA2010.dat	NO2APT2009.dat	NO2PIR2009.dat	O3#LIO2009.dat	SO2BIO2007.dat
CO#LAR2002.dat	NO#IRA2001.dat	NO2APT2010.dat	NO2PIR2010.dat	O3#LIO2010.dat	SO2ELE2000.dat
CO#LAR2007.dat	NO#KAL2001.dat	NO2APT2011.dat	NO2PIR2011.dat	O3#LIO2011.dat	SO2ELE2001.dat
CO#LAR2008.dat	NO#KAL2002.dat	NO2ARI2000.dat	NO2SIN2001.dat	O3#LYK2000.dat	SO2ELE2002.dat
CO#LAR2009.dat	NO#KAL2003.dat	NO2ARI2001.dat	NO2SIN2002.dat	O3#LYK2001.dat	SO2ELE2003.dat
CO#LAR2010.dat	NO#KAL2004.dat	NO2ARI2002.dat	NO2SIN2003.dat	O3#LYK2002.dat	SO2ELE2004.dat
CO#LAR2011.dat	NO#KAL2006.dat	NO2ARI2003.dat	NO2SIN2006.dat	O3#LYK2003.dat	SO2ELE2005.dat
CO#LYK2000.dat	NO#KAL2007.dat	NO2ARI2004.dat	NO2SIN2007.dat	O3#LYK2004.dat	SO2ELE2006.dat
CO#LYK2001.dat	NO#KAL2008.dat	NO2ARI2005.dat	NO2SIN2008.dat	O3#LYK2005.dat	SO2ELE2007.dat
CO#LYK2002.dat	NO#KAL2009.dat	NO2ARI2006.dat	NO2SIN2009.dat	O3#LYK2006.dat	SO2ELE2008.dat
CO#LYK2003.dat	NO#KAL2010.dat	NO2ARI2007.dat	NO2SIN2010.dat	O3#LYK2007.dat	SO2ELE2009.dat
CO#LYK2004.dat	NO#KOD2001.dat	NO2ARI2008.dat	NO2SIN2011.dat	O3#LYK2008.dat	SO2ELE2010.dat
CO#LYK2005.dat	NO#KOD2002.dat	NO2ARI2009.dat	NO2SMY2000.dat	O3#LYK2009.dat	SO2ELE2011.dat
CO#MAR2000.dat	NO#KOD2003.dat	NO2ARI2010.dat	NO2SMY2001.dat	O3#LYK2010.dat	SO2GAL2000.dat
CO#MAR2001.dat	NO#KOD2004.dat	NO2ARI2011.dat	NO2SMY2002.dat	O3#LYK2011.dat	SO2GAL2001.dat
CO#MAR2002.dat	NO#KOD2005.dat	NO2ATH2000.dat	NO2SMY2003.dat	O3#MAR2000.dat	SO2GAL2002.dat
CO#MAR2003.dat	NO#KOD2006.dat	NO2ATH2001.dat	NO2SMY2004.dat	O3#MAR2001.dat	SO2GAL2003.dat
CO#MAR2004.dat	NO#KOD2007.dat	NO2ATH2002.dat	NO2SMY2005.dat	O3#MAR2002.dat	SO2GAL2004.dat

CO#MAR2005.dat	NO#KOD2008.dat	NO2ATH2003.dat	NO2SMY2006.dat	O3#MAR2003.dat	SO2GAL2005.dat
CO#MAR2006.dat	NO#KOD2009.dat	NO2ATH2004.dat	NO2SMY2007.dat	O3#MAR2004.dat	SO2GEO2000.dat
CO#MAR2007.dat	NO#KOD2010.dat	NO2ATH2005.dat	NO2SMY2008.dat	O3#MAR2005.dat	SO2GEO2001.dat
CO#MAR2008.dat	NO#KOD2011.dat	NO2ATH2006.dat	NO2SMY2009.dat	O3#MAR2006.dat	SO2GEO2002.dat
CO#MAR2009.dat	NO#kor2008.dat	NO2ATH2007.dat	NO2SMY2010.dat	O3#MAR2007.dat	SO2GEO2003.dat
CO#MAR2010.dat	NO#kor2009.dat	NO2ATH2008.dat	NO2SMY2011.dat	O3#MAR2008.dat	SO2GEO2004.dat
CO#MAR2011.dat	NO#kor2010.dat	NO2ATH2009.dat	NO2THR2000.dat	O3#MAR2009.dat	SO2GEO2005.dat
CO#Oin2008.dat	NO#kor2011.dat	NO2ATH2010.dat	NO2THR2001.dat	O3#MAR2010.dat	SO2GEO2006.dat
CO#Oin2009.dat	NO#LAR2001.dat	NO2ATH2011.dat	NO2THR2002.dat	O3#MAR2011.dat	SO2GEO2007.dat
CO#PAI2001.dat	NO#LAR2002.dat	NO2BIO2000.dat	NO2THR2003.dat	O3#NEO2001.dat	SO2GEO2008.dat
CO#PAI2002.dat	NO#LAR2008.dat	NO2BIO2001.dat	NO2THR2004.dat	O3#NEO2002.dat	SO2GEO2009.dat
CO#PAI2003.dat	NO#LAR2009.dat	NO2BIO2002.dat	NO2THR2005.dat	O3#NEO2003.dat	SO2GEO2010.dat
CO#PAI2004.dat	NO#LAR2011.dat	NO2BIO2003.dat	NO2THR2006.dat	O3#NEO2005.dat	SO2IRA2001.dat
CO#PAI2007.dat	NO#LIO2000.dat	NO2BIO2004.dat	NO2THR2007.dat	O3#NEO2006.dat	SO2IRA2002.dat
CO#PAI2008.dat	NO#LIO2001.dat	NO2BIO2005.dat	NO2THR2008.dat	O3#NEO2007.dat	SO2IRA2003.dat
CO#PAI2009.dat	NO#LIO2002.dat	NO2BIO2006.dat	NO2THR2009.dat	O3#NEO2008.dat	SO2KAL2001.dat
CO#PAI2010.dat	NO#LIO2003.dat	NO2BIO2007.dat	NO2THR2010.dat	O3#NEO2009.dat	SO2KAL2002.dat
CO#PAI2011.dat	NO#LIO2004.dat	NO2ELE2000.dat	NO2THR2011.dat	O3#NEO2011.dat	SO2KAL2003.dat
CO#PAII2001.dat	NO#LIO2005.dat	NO2ELE2001.dat	NO2VOL2001.dat	O3#Oin2008.dat	SO2KOD2001.dat
CO#PAII2002.dat	NO#LIO2006.dat	NO2ELE2002.dat	NO2VOL2002.dat	O3#Oin2009.dat	SO2KOD2002.dat
CO#PAII2003.dat	NO#LIO2007.dat	NO2ELE2003.dat	NO2VOL2003.dat	O3#Oin2010.dat	SO2KOD2003.dat
CO#PAII2007.dat	NO#LIO2008.dat	NO2ELE2004.dat	NO2VOL2004.dat	O3#Oin2011.dat	SO2KOD2006.dat
CO#PAII2008.dat	NO#LIO2009.dat	NO2ELE2005.dat	NO2VOL2005.dat	O3#PAI2001.dat	SO2KOD2007.dat
CO#PAII2009.dat	NO#LIO2010.dat	NO2ELE2006.dat	NO2VOL2006.dat	O3#PAI2002.dat	SO2KOD2008.dat
CO#PAII2010.dat	NO#LIO2011.dat	NO2ELE2007.dat	NOXAGP2000.dat	O3#PAI2003.dat	SO2KOD2009.dat
CO#PAII2011.dat	NO#LYK2000.dat	NO2ELE2008.dat	NOXAGP2001.dat	O3#PAI2007.dat	SO2KOD2010.dat
CO#PAT2000.dat	NO#LYK2001.dat	NO2ELE2009.dat	NOXAGP2002.dat	O3#PAI2008.dat	SO2KOD2011.dat
CO#PAT2001.dat	NO#LYK2002.dat	NO2ELE2010.dat	NOXARI2000.dat	O3#PAI2009.dat	SO2kor2008.dat

CO#PAT2002.dat	NO#LYK2003.dat	NO2ELE2011.dat	NOXARI2001.dat	O3#PAI2010.dat	SO2kor2009.dat
CO#PAT2003.dat	NO#LYK2004.dat	NO2GAL2000.dat	NOXARI2002.dat	O3#PAI2011.dat	SO2kor2010.dat
CO#PAT2004.dat	NO#LYK2005.dat	NO2GAL2001.dat	NOXATH2000.dat	O3#PAN2000.dat	SO2LAR2001.dat
CO#PAT2005.dat	NO#LYK2006.dat	NO2GAL2002.dat	NOXATH2001.dat	O3#PAN2001.dat	SO2LAR2002.dat
CO#PAT2006.dat	NO#LYK2007.dat	NO2GAL2003.dat	NOXATH2002.dat	O3#PAN2002.dat	SO2LIO2000.dat
CO#PAT2007.dat	NO#LYK2008.dat	NO2GAL2004.dat	NOXBIO2000.dat	O3#PAN2003.dat	SO2LIO2001.dat
CO#PAT2008.dat	NO#LYK2009.dat	NO2GAL2005.dat	NOXBIO2001.dat	O3#PAN2004.dat	SO2LIO2002.dat
CO#PAT2009.dat	NO#LYK2010.dat	NO2GEO2000.dat	NOXBIO2002.dat	O3#PAN2005.dat	SO2LIO2003.dat
CO#PAT2010.dat	NO#LYK2011.dat	NO2GEO2001.dat	NOXELE2000.dat	O3#PAN2006.dat	SO2LIO2004.dat
CO#PAT2011.dat	NO#MAR2000.dat	NO2GEO2002.dat	NOXELE2001.dat	O3#PAN2007.dat	SO2LIO2005.dat
CO#PER2000.dat	NO#MAR2001.dat	NO2GEO2003.dat	NOXELE2002.dat	O3#PAO2001.dat	SO2LIO2006.dat
CO#PER2001.dat	NO#MAR2002.dat	NO2GEO2004.dat	NOXGAL2000.dat	O3#PAO2002.dat	SO2LIO2007.dat
CO#PER2002.dat	NO#MAR2003.dat	NO2GEO2005.dat	NOXGAL2001.dat	O3#PAO2003.dat	SO2LIO2008.dat
CO#PER2003.dat	NO#MAR2004.dat	NO2GEO2006.dat	NOXGAL2002.dat	O3#PAO2004.dat	SO2LIO2009.dat
CO#PER2004.dat	NO#MAR2005.dat	NO2GEO2007.dat	NOXGEO2000.dat	O3#PAO2005.dat	SO2MAR2000.dat
CO#PER2005.dat	NO#MAR2006.dat	NO2GEO2008.dat	NOXGEO2001.dat	O3#PAO2006.dat	SO2MAR2001.dat
CO#PER2006.dat	NO#MAR2007.dat	NO2GEO2009.dat	NOXGEO2002.dat	O3#PAO2008.dat	SO2MAR2002.dat
CO#PER2007.dat	NO#MAR2008.dat	NO2GEO2010.dat	NOXGOU2000.dat	O3#PAO2009.dat	SO2MAR2003.dat
CO#PER2008.dat	NO#MAR2009.dat	NO2GEO2011.dat	NOXGOU2001.dat	O3#PAO2010.dat	SO2MAR2004.dat
CO#PER2009.dat	NO#MAR2010.dat	NO2GOU2000.dat	NOXGOU2002.dat	O3#PAO2011.dat	SO2MAR2005.dat
CO#PER2010.dat	NO#MAR2011.dat	NO2GOU2001.dat	NOXLIO2000.dat	O3#PAT2000.dat	SO2Oin2008.dat
CO#PER2011.dat	NO#NEO2001.dat	NO2GOU2002.dat	NOXLIO2001.dat	O3#PAT2001.dat	SO2Oin2009.dat
CO#PIR2000.dat	NO#NEO2002.dat	NO2GOU2003.dat	NOXLIO2002.dat	O3#PAT2002.dat	SO2PAI2001.dat
CO#PIR2001.dat	NO#NEO2003.dat	NO2GOU2004.dat	NOXLYK2000.dat	O3#PAT2003.dat	SO2PAI2002.dat
CO#PIR2002.dat	NO#NEO2004.dat	NO2GOU2005.dat	NOXLYK2001.dat	O3#PAT2004.dat	SO2PAI2003.dat
CO#PIR2003.dat	NO#NEO2006.dat	NO2GOU2006.dat	NOXLYK2002.dat	O3#PAT2005.dat	SO2PAI2004.dat
CO#PIR2004.dat	NO#NEO2007.dat	NO2GOU2007.dat	NOXMAR2000.dat	O3#PAT2006.dat	SO2PAI2007.dat
CO#PIR2005.dat	NO#NEO2008.dat	NO2GOU2008.dat	NOXMAR2001.dat	O3#PAT2007.dat	SO2PAI2008.dat

CO#PIR2006.dat	NO#NEO2009.dat	NO2GOU2009.dat	NOXMAR2002.dat	O3#PAT2008.dat	SO2PAI2009.dat
CO#PIR2007.dat	NO#NEO2011.dat	NO2IOA2008.dat	NOXPAN2000.dat	O3#PAT2009.dat	SO2PAI2010.dat
CO#PIR2008.dat	NO#Oin2008.dat	NO2IOA2009.dat	NOXPAN2001.dat	O3#PAT2010.dat	SO2PAI2011.dat
CO#PIR2009.dat	NO#Oin2009.dat	NO2IOA2010.dat	NOXPAN2002.dat	O3#PAT2011.dat	SO2PAI2001.dat
CO#PIR2010.dat	NO#Oin2010.dat	NO2IRA2001.dat	NOXPAT2000.dat	O3#PER2000.dat	SO2PAI2002.dat
CO#PIR2011.dat	NO#Oin2011.dat	NO2KAL2001.dat	NOXPAT2001.dat	O3#PER2001.dat	SO2PAI2003.dat
CO#SIN2001.dat	NO#PAI2001.dat	NO2KAL2002.dat	NOXPAT2002.dat	O3#PER2002.dat	SO2PAI2007.dat
CO#SIN2002.dat	NO#PAI2002.dat	NO2KAL2003.dat	NOXPER2000.dat	O3#PER2003.dat	SO2PAI2008.dat
CO#SIN2003.dat	NO#PAI2003.dat	NO2KAL2004.dat	NOXPER2001.dat	O3#PER2004.dat	SO2PAI2009.dat
CO#SIN2006.dat	NO#PAI2004.dat	NO2KAL2006.dat	NOXPER2002.dat	O3#PER2005.dat	SO2PAI2010.dat
CO#SIN2007.dat	NO#PAI2007.dat	NO2KAL2007.dat	NOXPIR2000.dat	O3#PER2006.dat	SO2PAI2011.dat
CO#SIN2008.dat	NO#PAI2008.dat	NO2KAL2008.dat	NOXPIR2001.dat	O3#PER2007.dat	SO2PAN2000.dat
CO#SIN2009.dat	NO#PAI2009.dat	NO2KAL2009.dat	NOXPIR2002.dat	O3#PER2008.dat	SO2PAN2001.dat
CO#SIN2010.dat	NO#PAI2010.dat	NO2KAL2010.dat	NOXSMY2000.dat	O3#PER2009.dat	SO2PAN2002.dat
CO#SIN2011.dat	NO#PAI2011.dat	NO2KOD2001.dat	NOXSMY2001.dat	O3#PER2010.dat	SO2PAN2003.dat
CO#SMY2000.dat	NO#PAI2001.dat	NO2KOD2002.dat	NOXSMY2002.dat	O3#PER2011.dat	SO2PAN2004.dat
CO#SMY2001.dat	NO#PAI2002.dat	NO2KOD2003.dat	NOXTHR2000.dat	O3#PIR2000.dat	SO2PAN2005.dat
CO#SMY2002.dat	NO#PAI2007.dat	NO2KOD2004.dat	NOXTHR2001.dat	O3#PIR2001.dat	SO2PAT2000.dat
CO#SMY2003.dat	NO#PAI2008.dat	NO2KOD2005.dat	NOXTHR2002.dat	O3#PIR2002.dat	SO2PAT2001.dat
CO#SMY2004.dat	NO#PAI2009.dat	NO2KOD2006.dat	O3#AGP2000.dat	O3#PIR2003.dat	SO2PAT2002.dat
CO#SMY2005.dat	NO#PAI2010.dat	NO2KOD2007.dat	O3#AGP2001.dat	O3#PIR2004.dat	SO2PAT2003.dat
CO#SMY2006.dat	NO#PAN2000.dat	NO2KOD2008.dat	O3#AGP2002.dat	O3#PIR2005.dat	SO2PAT2004.dat
CO#SMY2007.dat	NO#PAN2001.dat	NO2KOD2009.dat	O3#AGP2003.dat	O3#PIR2006.dat	SO2PAT2005.dat
CO#SMY2008.dat	NO#PAN2002.dat	NO2KOD2010.dat	O3#AGP2004.dat	O3#PIR2007.dat	SO2PAT2006.dat
CO#SMY2009.dat	NO#PAN2003.dat	NO2KOD2011.dat	O3#AGP2005.dat	O3#PIR2008.dat	SO2PAT2007.dat
CO#SMY2010.dat	NO#PAN2004.dat	NO2kor2008.dat	O3#AGP2006.dat	O3#PIR2009.dat	SO2PAT2008.dat
CO#SMY2011.dat	NO#PAN2005.dat	NO2kor2009.dat	O3#AGP2007.dat	O3#PIR2010.dat	SO2PAT2009.dat
CO#VOL2001.dat	NO#PAN2006.dat	NO2kor2010.dat	O3#AGP2008.dat	O3#PIR2011.dat	SO2PAT2010.dat

CO#VOL2002.dat	NO#PAN2007.dat	NO2kor2011.dat	O3#AGP2009.dat	O3#SIN2001.dat	SO2PAT2011.dat
CO#VOL2003.dat	NO#PAO2001.dat	NO2LAR2001.dat	O3#AGP2010.dat	O3#SIN2002.dat	SO2PER2000.dat
CO#VOL2004.dat	NO#PAO2002.dat	NO2LAR2002.dat	O3#AGP2011.dat	O3#SIN2003.dat	SO2PER2001.dat
CO#VOL2005.dat	NO#PAO2003.dat	NO2LAR2008.dat	O3#AGS2001.dat	O3#SIN2004.dat	SO2PER2002.dat
CO#VOL2006.dat	NO#PAO2007.dat	NO2LAR2009.dat	O3#AGS2002.dat	O3#SIN2005.dat	SO2PER2003.dat
CO#VOL2007.dat	NO#PAO2008.dat	NO2LAR2011.dat	O3#AGS2003.dat	O3#SIN2006.dat	SO2PER2004.dat
CO#VOL2008.dat	NO#PAO2009.dat	NO2LIO2000.dat	O3#AGS2004.dat	O3#SIN2007.dat	SO2PER2005.dat
NO#AGP2000.dat	NO#PAO2010.dat	NO2LIO2001.dat	O3#AGS2007.dat	O3#SIN2008.dat	SO2PER2006.dat
NO#AGP2001.dat	NO#PAO2011.dat	NO2LIO2002.dat	O3#AGS2008.dat	O3#SIN2009.dat	SO2PER2007.dat
NO#AGP2002.dat	NO#PAT2000.dat	NO2LIO2003.dat	O3#AGS2009.dat	O3#SIN2010.dat	SO2PER2008.dat
NO#AGP2003.dat	NO#PAT2001.dat	NO2LIO2004.dat	O3#AGS2010.dat	O3#SIN2011.dat	SO2PER2009.dat
NO#AGP2004.dat	NO#PAT2002.dat	NO2LIO2005.dat	O3#AGS2011.dat	O3#SMY2000.dat	SO2PER2010.dat
NO#AGP2005.dat	NO#PAT2003.dat	NO2LIO2006.dat	O3#ALI2001.dat	O3#SMY2001.dat	SO2PER2011.dat
NO#AGP2006.dat	NO#PAT2004.dat	NO2LIO2007.dat	O3#ALI2002.dat	O3#SMY2002.dat	SO2PIR2000.dat
NO#AGP2007.dat	NO#PAT2005.dat	NO2LIO2008.dat	O3#ALI2007.dat	O3#SMY2003.dat	SO2PIR2001.dat
NO#AGP2008.dat	NO#PAT2006.dat	NO2LIO2009.dat	O3#ALI2008.dat	O3#SMY2004.dat	SO2PIR2002.dat
NO#AGP2009.dat	NO#PAT2007.dat	NO2LIO2010.dat	O3#ALI2009.dat	O3#SMY2005.dat	SO2PIR2003.dat
NO#AGP2010.dat	NO#PAT2008.dat	NO2LIO2011.dat	O3#ALI2010.dat	O3#SMY2006.dat	SO2PIR2004.dat
NO#AGP2011.dat	NO#PAT2009.dat	NO2LYK2000.dat	O3#ALI2011.dat	O3#SMY2007.dat	SO2PIR2005.dat
NO#AGS2001.dat	NO#PAT2010.dat	NO2LYK2001.dat	O3#APT2001.dat	O3#SMY2008.dat	SO2PIR2006.dat
NO#AGS2002.dat	NO#PAT2011.dat	NO2LYK2002.dat	O3#APT2002.dat	O3#SMY2009.dat	SO2PIR2007.dat
NO#AGS2003.dat	NO#PER2000.dat	NO2LYK2003.dat	O3#APT2003.dat	O3#SMY2010.dat	SO2PIR2008.dat
NO#AGS2004.dat	NO#PER2001.dat	NO2LYK2004.dat	O3#APT2004.dat	O3#SMY2011.dat	SO2PIR2009.dat
NO#AGS2007.dat	NO#PER2002.dat	NO2LYK2005.dat	O3#APT2005.dat	O3#THR2000.dat	SO2PIR2010.dat
NO#AGS2008.dat	NO#PER2003.dat	NO2LYK2006.dat	O3#APT2006.dat	O3#THR2001.dat	SO2PIR2011.dat
NO#AGS2009.dat	NO#PER2004.dat	NO2LYK2007.dat	O3#APT2007.dat	O3#THR2002.dat	SO2SIN2001.dat
NO#AGS2010.dat	NO#PER2005.dat	NO2LYK2008.dat	O3#APT2008.dat	O3#THR2003.dat	SO2SIN2002.dat
NO#AGS2011.dat	NO#PER2006.dat	NO2LYK2009.dat	O3#APT2009.dat	O3#THR2004.dat	SO2SIN2003.dat

NO#ALI2001.dat	NO#PER2007.dat	NO2LYK2010.dat	O3#APT2010.dat	O3#THR2005.dat	SO2SIN2004.dat
NO#ALI2005.dat	NO#PER2008.dat	NO2LYK2011.dat	O3#APT2011.dat	O3#THR2006.dat	SO2SIN2006.dat
NO#ALI2006.dat	NO#PER2009.dat	NO2MAR2000.dat	O3#ATH2000.dat	O3#THR2007.dat	SO2SIN2008.dat
NO#ALI2007.dat	NO#PER2010.dat	NO2MAR2001.dat	O3#ATH2001.dat	O3#THR2008.dat	SO2SIN2010.dat
NO#ALI2008.dat	NO#PER2011.dat	NO2MAR2002.dat	O3#ATH2002.dat	O3#THR2009.dat	SO2SIN2011.dat
NO#ALI2009.dat	NO#PIR2000.dat	NO2MAR2003.dat	O3#ATH2003.dat	O3#THR2010.dat	SO2SMY2000.dat
NO#ALI2010.dat	NO#PIR2001.dat	NO2MAR2004.dat	O3#ATH2004.dat	O3#THR2011.dat	SO2SMY2001.dat
NO#ALI2011.dat	NO#PIR2002.dat	NO2MAR2005.dat	O3#ATH2005.dat	O3#VOL2001.dat	SO2SMY2002.dat
NO#APT2001.dat	NO#PIR2003.dat	NO2MAR2006.dat	O3#ATH2006.dat	O3#VOL2002.dat	SO2SMY2003.dat
NO#APT2002.dat	NO#PIR2004.dat	NO2MAR2007.dat	O3#ATH2007.dat	O3#VOL2003.dat	SO2SMY2004.dat
NO#APT2003.dat	NO#PIR2005.dat	NO2MAR2008.dat	O3#ATH2008.dat	O3#VOL2004.dat	SO2SMY2005.dat
NO#APT2006.dat	NO#PIR2006.dat	NO2MAR2009.dat	O3#ATH2009.dat	O3#VOL2005.dat	SO2SMY2006.dat
NO#APT2007.dat	NO#PIR2007.dat	NO2MAR2010.dat	O3#ATH2010.dat	O3#VOL2006.dat	SO2SMY2007.dat
NO#APT2008.dat	NO#PIR2008.dat	NO2MAR2011.dat	O3#ATH2011.dat	O3#VOL2007.dat	SO2SMY2008.dat
NO#APT2009.dat	NO#PIR2009.dat	NO2NEO2001.dat	O3#BIO2000.dat	O3#VOL2008.dat	SO2SMY2009.dat
NO#APT2010.dat	NO#PIR2010.dat	NO2NEO2002.dat	O3#BIO2001.dat	SmokeARI2000.dat	SO2SMY2010.dat
NO#APT2011.dat	NO#PIR2011.dat	NO2NEO2003.dat	O3#BIO2002.dat	SmokeARI2001.dat	SO2VOL2001.dat
NO#ARI2000.dat	NO#SIN2001.dat	NO2NEO2004.dat	O3#BIO2003.dat	SmokeARI2002.dat	SO2VOL2002.dat
NO#ARI2001.dat	NO#SIN2002.dat	NO2NEO2006.dat	O3#BIO2004.dat	SmokeARI2003.xls	SO2VOL2003.dat
NO#ARI2002.dat	NO#SIN2003.dat	NO2NEO2007.dat	O3#BIO2005.dat	SmokeARI2004.dat	SO2VOL2004.dat
NO#ARI2003.dat	NO#SIN2006.dat	NO2NEO2008.dat	O3#BIO2006.dat	SmokeATH2000.dat	SO2VOL2005.dat
NO#ARI2004.dat	NO#SIN2007.dat	NO2NEO2009.dat	O3#BIO2007.dat	SmokeATH2001.dat	SO2VOL2006.dat
NO#ARI2005.dat	NO#SIN2008.dat	NO2NEO2011.dat	O3#ELE2000.dat	SmokeATH2002.dat	SO2VOL2007.dat
NO#ARI2006.dat	NO#SIN2009.dat	NO2Oin2008.dat	O3#ELE2001.dat	SmokeATH2003.xls	
NO#ARI2007.dat	NO#SIN2010.dat	NO2Oin2009.dat	O3#ELE2002.dat	SmokeATH2004.dat	
NO#ARI2008.dat	NO#SIN2011.dat	NO2Oin2010.dat	O3#ELE2003.dat	SmokePAT2000.dat	

### 8.6 Εικόνα Π1. Τμήμα του αρχείου μετρήσεων

