



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη εφαρμογής για την επικοινωνία με τη διαδικτυακή υπηρεσία XML-RPC
Thesis Title	Application development for communication with the web service XML-RPC
Όνοματεπώνυμο Φοιτητή	Μαρινέλα Κύρκο
Πατρώνυμο	Βασίλης
Αριθμός Μητρώου	ΜΠΠΛ/ 10017
Επιβλέπων	Χρήστος Δουληγέρης, Καθηγητής
Συνεπιβλέπων	Βασίλης Μενεκλής Διδάκτορας

Ημερομηνία Παράδοσης **Μάιος 2013**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Χρήστος Δουληγέρης
ΚαθηγητήςΔέσποινα Πολέμη
ΚαθηγήτριαΠαναγιώτης
Κοτζανικολάου
Λέκτορας

ΠΕΡΙΛΗΨΗ

Η παρούσα μεταπτυχιακή διατριβή ασχολείται με το σχεδιασμό, την ανάλυση και την ανάπτυξη μιας εφαρμογής με χάρτες της Google. Χρησιμοποιούνται κυρίως δύο τεχνολογίες το Codeigniter και το XML-RPC με σκοπό την επικοινωνία με μια διαδικτυακή υπηρεσία και την παρουσίαση δεδομένων από μια βάση δεδομένων. Βασικός στόχος είναι η εμφάνιση πραγματικών δεδομένων στο χάρτη, σε πραγματικό χρόνο.

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν οι τεχνολογίες Codeigniter, Xml-Rpc, Php, Html, Javascript, JQuery, GoogleMapApplication. Η ανάλυση και ο σχεδιασμός βασίστηκαν σε αντικειμενοστρεφείς τεχνικές (uml διαγράμματα).

ABSTRACT

This master thesis deals with the design, analysis and development of an application with Google maps. Twotechnologies are mainly used, namely Codeigniter and XML-RPC, to communicate with a web service and to present data from a database. The main objective is the appearance of real data on a map, in real time.

For the application development technologies were used were Codeigniter, Xml-Rpc, Php, Html, Javascript, JQuery, Google Map Application. The analysis and the design were based on object oriented techniques (uml diagrams).

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα μεταπτυχιακή διατριβή είναι η προσπάθεια αρκετών μηνών και θα ήταν αδύνατο να εκπονηθεί χωρίς τη βοήθεια κάποιων ανθρώπων τους οποίους και θα ήθελα να ευχαριστήσω.

Αρχικά, τον Επιβλέποντα Καθηγητή κ. Χρήστο Δουληγέρι, Καθηγητή του τμήματος Πληροφορικής του Μεταπτυχιακού προγράμματος του πανεπιστημίου Πειραιά, ο οποίος με τίμησε με την εμπιστοσύνη του και μου έδωσε τη δυνατότητα να ασχοληθώ με αυτό το θέμα. Ιδιαίτερα ευχαριστώ στο διδάκτορα κ. Βασίλειο Μενεκλή για την απεριόριστη υπομονή του, τις συμβουλές και καθοδηγήσεις του καθ' όλη τη διάρκεια της ενασχόλησής μου με την εργασία, καθώς με τις γνώσεις και την εμπειρία του διευκόλυνε σε μεγάλο βαθμό την εκπόνησή της.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου για την αμέριστη υποστήριξή τους σε όλη τη διάρκεια των φοιτητικών μου χρόνων και για τις πολλές θυσίες που έκαναν προκειμένου να φτάσω έως εδώ. Επίσης, θέλω να ευχαριστήσω τους μεταπτυχιακούς φοιτητές του τμήματος Μαρία Κλεισά, Δημήτρη Κουγιουμτζή και Αποστόλη Χριστοδούλου για την ενθάρρυνση τους, τις συμβουλές τους, την βοήθεια τους και όλες τις ωραίες στιγμές που έχουμε περάσει μαζί.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη.....	3
Ευχαριστίες.....	4
Κατάλογος σχημάτων	7
Κατάλογος πινάκων.....	8
Γλωσσάρι.....	9
Κεφάλαιο 1- Εισαγωγή.....	10
1.1.Σύντομη αναφορά στην ανάπτυξη της κατάστασης της τεχνολογίας του Διαδικτύου...	10
1.2.Ανάλυση θέματοςεργασίας.....	10
1.3.Δομή κεφαλαίων.....	11
Κεφάλαιο 2 – Βιβλιογραφική Έρευνα.....	11
Κεφάλαιο 3 – Τεχνολογίες που χρησιμοποιήθηκαν.....	22
3.1.PHP.....	22
3.2.Πλαίσιο CodeIgniter.....	23
3.2.1.Χαρακτηριστικά CodeIgniter.....	24
3.2.2.Σύγκριση CodeIgniter με άλλες πλατφόρμες.....	25
3.3.Μοντέλο-Άποψη-Ελεγκτή (MVC).....	26
3.3.1.Παραδείγματα Μοντέλου-Άποψης-Ελεγκτή.....	27
3.3.2.ΠλεονεκτήματαMVC.....	28
3.4.Υπηρεσίες Ιστού.....	29
3.5.XML-RPC.....	33
3.5.1.Αποστολή XML-RPC αίτησης.....	34
3.5.2.Διαμόρφωση απάντησης.....	35
3.6.GoogleMap.....	35
3.6.1.Παγκόσμιο Γεωδαισιπικό Σύστημα Αναφοράς 1984-WGS84.....	36
3.6.2.Στοιχεία απεικόνισης GoogleMapAPI.....	37
3.6.3.Ανάλυση GoogleMapAPI.....	37
3.6.4.Παράδειγμα.....	38
3.7.JQuery.....	40
3.8.HTML.....	42
3.9.CSS.....	43
Κεφάλαιο 4 – Αρχιτεκτονική.....	44
4.1.Ενοποιημένη γλώσσα σχεδιασμού (UnifiedModelingLanguage) UML.....	45
4.2.Σύλληψη απαιτήσεων.....	45
4.3.Ανάλυση και σχεδιασμός.....	46
4.4.Παρουσίαση εφαρμογής.....	
Κεφάλαιο 5 – Συμπεράσματα.....	52

Βιβλιογραφία.....	54
Παράρτημα Κώδικα.....	57

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Ανάπτυξη εφαρμογής για την επικοινωνία με τη διαδικτυακή υπηρεσία XML-RPC7

Σχήμα 1. Σχεδιάγραμμα λειτουργίας PHP	22
Σχήμα 2. Δομή αρχείων του CodeIgniter.....	24
Σχήμα 3. Δομή συστήματος MVC προτύπου.....	27
Σχήμα 4. Μοντέλο των WebServices.....	30
Σχήμα 5. Αρχιτεκτονική Webservices.....	31
Σχήμα 6. Βασικές τεχνολογίες των web services.....	32
Σχήμα 7. Ορισμός συστήματος συντεταγμένων WGS84.....	36
Σχήμα 8. Googlemap, χάρτης με επίκεντρο την Αθήνα Ελλάδα.....	39
Σχήμα 9. Διάγραμμα περιπτώσεων χρήσης.....	39
Σχήμα 10. Διάγραμμα τάξεων.....	39
Σχήμα 11. Διάγραμμα αντικειμένων για τη σχέση του χρήστη και του μεσίτη με την εφαρμογή...	39
Σχήμα 12. Διάγραμμα αντικειμένων για το πρότυπο «Μοντέλο-Άποψη-Ελεγκτής».....	39
Σχήμα 13. Διάγραμμα συνεργασίας για τη διαδικασία ανάκτησης δεδομένων μέσω του webservice..... ...39	39
Σχήμα 14. Διάγραμμα συνεργασίας για το πρότυπο «Μοντέλο-Άποψη-Ελεγκτής».....	39
Σχήμα 15. Διάγραμμα σειράς για τη διαδικασία ανάκτησης δεδομένων μέσω του webservice..... ...39	39
Σχήμα 16. Διάγραμμα σειράς για το πρότυπο «Μοντέλο-Άποψη-Ελεγκτής».....	39
Σχήμα 17. Διάγραμμα δραστηριοτήτων.....	39
Σχήμα 18. Διάγραμμα εξαρτημάτων.....	39
Σχήμα 19. Αρχική σελίδα εφαρμογής.....	39
Σχήμα 20. Άνοιγμα παραθύρου πληροφοριών.....	39
Σχήμα 21. Εμφάνιση διαθέσιμων προς ενοικίαση ακινήτων.....	39

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1. Συγκριτική ανάλυση των παραμέτρων μεταξύ των τριώνπλαισίων.....	25
Πίνακας 2. Βασικά βήματα για την ανάπτυξη εφαρμογών με webservices.....	32

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΓΛΩΣΣΑΡΙ

Googlemapapplication: εφαρμογές με χάρτες της Google

OpensourcePHPframeworks: ανοιχτού κώδικα πλαίσια PHP

Model-View-Controller, MVC: Μοντέλο-Άποψη-Ελεγκτής

Request: αίτηση

Server: διακομιστής

Response: απάντηση

Markers: δείκτες

HyperTextpreprocessor: ΥπερκείμενοΠροεπεξεργαστή

Script: σενάριο

Client: πελάτης

Interface: διεπαφή

Free and Open Source Software Conference, FrOSCon:

ΕλεύθεροκαιΑνοικτούΛογισμικούΣυνέδριο

Web: ιστός

WebServices: Διαδικτυακές Υπηρεσίες

Software interface: διεπαφή λογισμικού

WebServicesDescriptionLanguage, WSDL: Περιγραφή Γλώσσας των Υπηρεσιών Ιστού

UniversalDescription, DiscoveryandIntegration, UDDI:

ΟικουμενικήΠεριγραφή,ΑνακάλυψηκαιΕνσωμάτωση

WorldWideWebConsortium, W3C: Κοινοπραξία του Παγκόσμιου Ιστού

Representational State Transfer,Rest: Αντιπροσωπευτική Μεταφορά Κατάστασης

Simple Object Access Protocol, SOAP: Απλό Πρωτόκολλο Πρόσβασης Αντικειμένου

Remote Procedure Call, RPC: Κλήση απομακρυσμένης διαδικασίας

Application Programming Interface , API : διεπαφήπρογραμματισμούεφαρμογών

International Earth Rotation Service, IERS: ΔιεθνήςΥπηρεσίαΠεριστροφήςτηςΓης

Institute for Environment and Sustainability, IES: ΙνστιτούτογιατοΠεριβάλλονκαιτηνΑειφορία

Earth-Centered- Earth-Fixed,ECEF: Γη-Κεντραρισμένη-Γη-Σταθερή

1. ΕΙΣΑΓΩΓΗ

1.1. Σύντομη αναφορά στην ανάπτυξη της κατάστασης της τεχνολογίας του Διαδικτύου

Κάνοντας μια μικρή αποτίμηση της κατάστασης κάθε έκφανσης της καθημερινότητας, γρήγορα μπορεί κανείς να καταλήξει στο αδιαμφισβήτητο συμπέρασμα ότι οι υπολογιστές και η πληροφορική έχουν διεισδύσει καθοριστικά και αμετάκλητα σε αναρίθμητες πτυχές της ζωής. Τόσο το εργασιακό, όσο και το οικιακό περιβάλλον, περιπλέκονται και αλληλοεξαρτώνται από την ύπαρξη και λειτουργία πληροφοριακών τεχνολογιών και εφαρμογών. Πρωτεύοντα ρόλο σε αυτή την εισβολή της πληροφοριακής τεχνολογίας στην εποχή μας, διαδραματίζει το Διαδίκτυο (Internet), που ίσως αποτελεί την επικρατέστερη και πιο εξέχουσα μορφή διείσδυσης της κοινωνίας της πληροφορικής.

Η επιρροή και χρησιμότητα του Διαδικτύου στην εποχή που διανύουμε, είναι κάτι περισσότερο από προφανές. Καθημερινά, ποικίλες ενέργειες τόσο στο χώρο της εργασίας, αλλά ακόμα και στον ελεύθερο χρόνο του καθενός, πραγματοποιείται στο διαδίκτυο. Χαρακτηριστικό είναι ότι πλέον πολλές καθημερινές ανάγκες και συνήθειες, έχουν μεταφερθεί από τον υλικό κόσμο, στον εικονικό κόσμο του διαδικτύου. Αγορές, πληρωμές λογαριασμών, αναζήτηση για κάθε είδος πληροφόρηση αποτελούν μόνο ορισμένες καθημερινές ανάγκες στις οποίες το Διαδίκτυο έχει παρέμβει. Ανάγκες που πλέον γίνονται πολύ πιο εύκολα, γρήγορα αλλά και πιο οικονομικά, χάρις στο Διαδίκτυο. Αναπτύχθηκαν πολλές τεχνολογίες, στις οποίες επάνω δομήθηκε η αναγνωρισιμότητα και η επικράτηση του Διαδικτύου στις μέρες μας. Εν τέλει με μια μικρή ομάδα αυτών των τεχνολογιών θα ασχοληθεί και η παρούσα μεταπτυχιακή διατριβή, για μια προσπάθεια ανάλυσης και κατανόησης της σημερινής κατάστασης της κοινωνίας της πληροφορικής. (Καλαποδάς Γεώργιος, 2005-2006.)

1.2. Θεματολογία

Η παρούσα εργασία ασχολείται με τη δημιουργία μιας εφαρμογής με τη χρήση χαρτών της google (googlemapapplication) ενσωματωμένη σε μια σελίδα, η οποία θα παρέχει πληροφορίες για πωλήσεις-ενοικιάσεις ακινήτων στους επισκέπτες της.

Χρησιμοποιήθηκε για την υλοποίησή της ένα πλαίσιο ανοιχτού κώδικα PHP (OpensourcePHPframeworks), το Codeigniter, το οποίο είναι βασισμένο στο αρχιτεκτονικό πρότυπο Μοντέλο-Άποψη-Ελεγκτής (Model-View-Controller, MVC). Το πρότυπο αυτό βοηθά να διατηρείται οργανωμένος ο κώδικας ώστε να είναι κατανοητός και πιο εύκολα συντηρούμενος.

Το Μοντέλο είναι το κομμάτι που διαχειρίζεται λειτουργίες πάνω σε δεδομένα της βάσης. Η Άποψη είναι το τμήμα που ασχολείται με το τι φαίνεται στο χρήστη και ο Ελεγκτής ελέγχει λειτουργίες του χρήστη, τις αναγνωρίζει και τις μεταφέρει στο κατάλληλο σημείο του κώδικα που θα επιτελέσει μια συγκεκριμένη λειτουργία. Επιπλέον, δέχεται τα δεδομένα που προκύπτουν από τα Μοντέλα και τα μεταφέρει στην κατάλληλη Άποψη.

Αναλυτικότερα, για τις ανάγκες της παρούσας πτυχιακής δε δημιουργήθηκε βάση δεδομένων που θα περιέχει τα δεδομένα των ακινήτων, αλλά ως πηγή δεδομένων χρησιμοποιήθηκε βάση μίας κτηματομεσιτικής πύλης, με τη βοήθεια μιας διαδικτυακής υπηρεσίας (webservice). Για την εκτέλεση αυτής της διαδικασίας έγινε χρήση της τεχνολογίας XML-RPC, η οποία έχει κύριο χαρακτηριστικό της την ομοιόμορφη μορφή και ανταλλαγή δεδομένων. Το Codeigniter παρέχει την τεχνολογία αυτή με την ιδιότητα της κλάσης που επιτρέπει την αποστολή αίτησης (request) σε άλλο διακομιστή (server), ή τη δημιουργία ενός διακομιστή XML-RPC που ναδέχεται αιτήσεις. Με αυτή την ικανότητα δημιουργήθηκε ένα XML-RPCrequest το οποίο συνδέθηκε με τη βάση της πύλης και καλούσε τα δεδομένα της, αν η σύνδεση και το ερώτημα ήταν σωστά στημένα τότε, ο server επέστρεφε το response δηλαδή την απάντηση με τα δεδομένα.

Στην συνέχεια με τη βοήθεια του GoogleApplication και τη χρήση της βιβλιοθήκης της Javascript , JQuery εμφανίστηκε το GoogleMap.

Σκοπός της εργασίας είναι να ενσωματωθούν τα δεδομένα που λήφθηκαν με τη βοήθεια της τεχνολογίας XML-RPC σε πραγματικό χρόνο στο χάρτη τηςGoogle. Τα δεδομένα αυτά μπορεί να ήταν συντεταγμένες των περιοχών που βρίσκονταν τα ακίνητα, τιμή, κατηγορία, τηλέφωνο και ούτω καθ' εξής. Έγινε επιλογή των δεδομένων που κρίθηκαν απαραίτητα για την εφαρμογή, πιο συγκεκριμένα στο χάρτη υπάρχουν markers (δείκτες) που «τραβούν» τις συντεταγμένες από τα δεδομένα της βάση. Οι markers λειτουργούν σαν κουμπιά που ο χρήστης μπορεί να πατήσει και να ανοίξει ένα παράθυρο (infowindow) με πληροφορίες των ακινήτων όπως κατηγορία, τιμή κ.τ.λ. Επιπρόσθετα έχουν δημιουργηθεί δύο κουμπιά πώληση/ενοικίαση (sale/rent) . Ο χρήστης έχει τη δυνατότητα να επιλέξει ένα από τα κουμπιά κάθε φορά με αποτέλεσμα να εμφανίζονται στο χάρτη οι markers με τις αντίστοιχες αγγελίες.

1.3. Δομή Κεφαλαίων

Στο 1^ο κεφάλαιο πραγματοποιείται μια σύντομη αναφορά στην ανάπτυξη της κατάστασης της τεχνολογίας καθώς και μια πρώτη εισαγωγή στο θέμα με το οποίοασχολείται η εργασία.

Στο 2^ο κεφάλαιο γίνεται μια εκτενή βιβλιογραφική αναφορά, πιο συγκεκριμένα αναφορές σε εργασίες με θέματα που σχετίζονται με την παρούσα μεταπτυχιακή διατριβή καθώς και μεταξύ τους σύγκριση.

Στο 3^ο κεφάλαιο αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν όπως το πρότυπο MVC, το PHPframeworksCodeigniter, η τεχνολογία XML-RPC κ.α

Στο 4^ο κεφάλαιο περιγράφεται η σχεδίαση της αρχιτεκτονικής της εργασίας με τη βοήθεια των διαγραμμάτων UML.

Στο 5^ο κεφάλαιο εξάγονται συμπεράσματα της κατάστασης της εφαρμογής που αναπτύχθηκε και πιθανές βελτιώσεις και εξελίξεις της.

2. ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΡΕΥΝΑ

Χρήση των χαρτών της Google ως διεπαφή για κατάλογο βιβλιοθήκης

Ο Michael Vandenburg (Michael Vandenburg, 2007.) της δημόσιας βιβλιοθήκης Kingston, από το Kingston του Καναδά, ασχολήθηκε με τη δημιουργία μιας εφαρμογής που είχε σκοπό να επιτρέψει στους χρήστες την αναζήτηση υλικών της βιβλιοθήκης, με γεωγραφικό θέμα σαν τίτλο αναζήτησης και σαν κύριο σημείο διεπαφής, τους χάρτες της Google. Η προσέγγιση είναι περιγραφική, με τη διαδικασία ανάπτυξης (development process) για την απόδειξη της ιδέας, τοποθετημένη στα ευρύτερα πλαίσια της ανάπτυξης εφαρμογών στη βιβλιοθήκη υποδοχής και την εμφάνιση των εργαλείων του Web 2.0. Η εργασία δείχνει ότι είναι δυνατή η χρήση των Εφαρμογών Διεπαφής Προγραμματισμού (Applications Programming Interface, APIs) που παρέχεται από μεγάλες επιχειρήσεις διαδικτύου όπως η Google, για τη δημιουργία χαρτών βασισμένων σε εργαλεία πλοήγησης με σκοπό την πρόσβαση σε βιβλιογραφικές πληροφορίες.

Τον Μάιο του 2005 η δημόσια βιβλιοθήκη Kingston Frontenac (Kingston Frontenac Public Library, KFPL) αποφάσισε να κάνει χρήση του Amazon API. Αντικατέστησε το σύστημα που βασιζόταν στο χαρτί με το αίτημα απευθείας σύνδεσης (online request) για αγορά (request for purchase, RFP) web εφαρμογής που αναπτύσσεται στο σπίτι. Η εφαρμογή RFP επέτρεψε στους χρήστες να εκτελέσουν μια απλή αναζήτηση στο Amazon για τον τίτλο που ζητούσαν αντί να τους επιβάλλει να συμπληρώσουν δώδεκα πεδία με βιβλιογραφικά δεδομένα σε τυποποιημένη μορφή web. Με την χρήση του API ήταν σε θέση να παρουσιαστούν τα αποτελέσματα του Amazon απευθείας στους χρήστες μέσα από την ιστοσελίδα της βιβλιοθήκης. Με την ύπαρξη ενός κουμπιού δίπλα σε κάθε αποτέλεσμα μπορούσε ο χρήστης κάνοντας κλικ να ζητήσει ένα αντίγραφο της αγοράς του τίτλου που είχε κάνει. Η ανάπτυξη της KFPL's RFP εφαρμογής έγινε με τη χρήση της Python και την εφαρμογή ανάπτυξης πλατφόρμας Zope. Η εφαρμογή κυκλοφόρησε στις αρχές του 2006.

Στη συνέχεια ενισχυμένοι από την επιτυχία της RFP εφαρμογής, το προσωπικό της βιβλιοθήκης αποφάσισε να «επενδύσει» και στους χάρτες της Google. Άρχισε λοιπόν να στήνεται η εφαρμογή η οποία στην απλούστερη μορφή της αποτελούταν από μια ιστοσελίδα στο πρότυπο του HousingMaps (www.housingmaps.com), με ένα χάρτη της Google στην αριστερή πλευρά της οθόνης και τις βιβλιογραφικές εγγραφές να φορτώνονται στη δεξιά πλευρά. Για τη δυναμική φόρτωση των δεδομένων χωρίς να απαιτείται η ξαναφόρτωση της σελίδας η KFPL έκανε χρήση του Mike West's DataRequestor JavaScript wrapper.

Για την ανάκτηση και την απεικόνιση των βιβλιογραφικών αρχείων χρησιμοποιήθηκε ένα προγενέστερο έργο της KFPL. Το 2002 ο βιβλιοθηκονόμος της βάσης δεδομένων δημιούργησε μια εφαρμογή που επέτρεπε στο προσωπικό να δημιουργεί και να διατηρεί λίστες βιβλίων (booklist) στην ιστοσελίδα της βιβλιοθήκης. Το έργο αναπτύχθηκε με την ιδέα ότι η αντιγραφή τμημάτων της βιβλιογραφικής βάσης δεδομένων της βιβλιοθήκης στην ιστοσελίδα θα ήταν ένα λάθος, και ότι θα ήταν προτιμότερο να τραβηχτούν αυτές οι πληροφορίες απευθείας από τη βάση δεδομένων της βιβλιοθήκης. Η εφαρμογή αναπτύχθηκε σε μια πλατφόρμα LAMP, χρησιμοποιώντας YAZ διακομιστή για την ανάκτηση των αρχείων από τη βάση δεδομένων μέσω του πρωτοκόλλου Z39.50. Για το backend της ιστοσελίδας, το προσωπικό του KFPL διέθετε με μια σειρά από εργαλεία που τους επέτρεπε να δημιουργήσουν μια λίστα βιβλίων και να αναζητήσουν στη βάση δεδομένων της βιβλιοθήκης εγγραφές για συμπλήρωση. Τα αποτελέσματα της αναζήτησης συνοδεύεται από συνδέσμους για την προσθήκη ατομικών εγγραφών στο Booklist. Μόνο το μοναδικό αναγνωριστικό για ένα επιλεγμένο αρχείο αποθηκεύεται στη βάση δεδομένων του Booklist, και όταν μια λίστα βιβλίων ανακτάται στο δημόσιο δικτυακό τόπο, γίνονται μια σειρά από αιτήσεις Z39.50 με σκοπό να παρουσιαστούν στους χρήστες τα βιβλιογραφικά αρχεία που απαιτούνται για να παρουσιαστεί η λίστα. Με μικρές τροποποιήσεις το εργαλείο αυτό θα χρησιμοποιηθεί για την εμφάνιση βιβλιογραφικών αρχείων μέσω του googlemap. Αντί οι χρήστες να πληκτρολογούν έναν όρο αναζήτησης σε μια φόρμα για την ανάκτηση αποτελεσμάτων, θα κάνουν κλικ στο χάρτη.

Δημιουργήθηκε ο χάρτης και τοποθετήθηκαν δείκτες (markers) για κάθε χώρα. Κάνοντας κλικ στους δείκτες θα γινόταν μια αναζήτηση για τις βιβλιογραφικές εγγραφές που σχετίζονται με την εκάστοτε χώρα. Για την τοποθέτηση των markers είναι απαραίτητα το γεωγραφικό μήκος και πλάτος τα οποία δόθηκαν από την CIA's World Factbook. Οι συντεταγμένες φορτώθηκαν σε ένα XML αρχείο που μπορούσε το googlemap να «διαβάσει». Το επόμενο βήμα ήταν να μετατραπεί αυτό σε μια συμβολοσειρά αναζήτησης που θα μπορούσε να σταλεί στο διακομιστή Z39.50 για την ανάκτηση ενός συνόλου εγγραφών. Στη συνέχεια θεωρήθηκε λάθος να ανακτηθούν όλα τα αρχεία που σχετίζονταν με μια χώρα και αποφασίστηκε να περιοριστεί η αναζήτηση σε αρχεία που σχετίζονται με την ιστορία, τα ταξίδια και την περιγραφή. Στο πλαίσιο κειμένου που θα εμφανίζονταν κάνοντας κλικ στους δείκτες ο χρήστης είχε την επιλογή να κάνει αναζήτηση στους καταλόγους της βιβλιοθήκης για αρχεία. Το εργαλείο Mike West's DataRequestor AJAX έκανε σχετικά απλό το να λάβει τις συμβολοσειρές αναζήτησης από τα πλαίσια κειμένου και να τα χρησιμοποιήσει για την φόρτωση ενός συνόλου καταλόγων με αποτελέσματα στη δεξιά πλευρά της οθόνης χωρίς να χρειαστεί να ξαναφορτωθεί η σελίδα. Η εφαρμογή θεωρήθηκε πλήρης τον Δεκέμβριο του 2006.

Σύγκριση με τη Μεταπτυχιακή Διατριβή

Στη μεταπτυχιακή διατριβή και στην παρούσα εργασία παρατηρούνται ορισμένες ομοιότητες και διαφορές. Ξεκινώντας με τις ομοιότητες, το κοινό σημείο που εντοπίζεται είναι στη χρήση μιας συγκεκριμένης τεχνολογίας, τους χάρτες της google και ο τρόπος διαχείρισης αυτής της τεχνολογίας. Αναλυτικότερα, όπως αναφέρθηκε και παραπάνω η βιβλιοθήκη δημιούργησε μια ιστοσελίδα όπου και ενσωμάτωσε ένα χάρτη της google, παρόμοια ενέργεια ακολουθήθηκε και στην μεταπτυχιακή διατριβή. Στη συνέχεια εξίσου και οι δύο εφαρμογές έκαναν χρήση των δεικτών γνωστών ως markers. Για την εμφάνιση των markers και οι δυο πλευρές ήταν απαραίτητο να δώσουν το γεωγραφικό πλάτος και το γεωγραφικό μήκος των περιοχών που επιθυμούσαν να συμπεριλάβουν στο χάρτη. Επιπρόσθετα παρόμοια λειτούργησαν δίνοντας λειτουργία στους δείκτες τους. Έτσι κάνοντας κλικ στο κάθε marker ανοιγόταν ένα πλαίσιο κειμένου (infowindow). Φρόντισαν και οι δύο πλευρές να εμπλουτίσουν τα infowindows με πληροφορίες σχετικές με την θέση που βρίσκονταν ο κάθε marker. Πληροφορίες που οι εφαρμογές τραβούσαν από μια βάση δεδομένων. Για την ανάκτηση των πληροφοριών αυτών έγινε χρήση πρωτοκόλλων υπεύθυνων για την αναζήτηση και την ανάκτηση πληροφοριών σε απομακρυσμένη βάση δεδομένων.

Σε ότι αφορά στις διαφορές, η πρώτη διαφορά είναι απλή και σχετίζεται με τα δεδομένα. Η μεταπτυχιακή διατριβή χρησιμοποιεί δεδομένα που πάρθηκαν από τη βάση δεδομένων μιας κτηματομεσιτικής πύλης ενώ η παραπάνω εργασία από τη βάση δεδομένων μιας βιβλιοθήκης. Δεύτερον η εργασία αυτή ακολούθησε το πρότυπο Housingmap της Google, με αποτέλεσμα ο χάρτης να βρίσκεται αριστερά της ιστοσελίδας και στη δεξιά πλευρά της να φορτώνονται πληροφορίες. Στη μεταπτυχιακή διατριβή οι πληροφορίες περιορίζονται στο περιεχόμενο των infowindows. Μια επιπλέον διαφορά είναι ότι χρησιμοποιήθηκαν διαφορετικά πρωτόκολλα για την ανάκτηση των πληροφοριών από τις εκάστοτε βάσεις. Στη μεταπτυχιακή διατριβή χρησιμοποιήθηκε το πρωτόκολλο XML-RPC ενώ στη συγκεκριμένη εργασία το πρωτόκολλο Z39.50. Επίσης στο infowindow της εφαρμογής των KFPL υπήρχε η επιλογή αναζήτησης αρχείων της βιβλιοθήκης. Έγινε και χρήση του εργαλείου Mike West's DataRequestor AJAX για να λάβει τις συμβολοσειρές αναζήτησης από τα πλαίσια κειμένου και να τα χρησιμοποιήσει για τη φόρτωση ενός συνόλου καταλόγων με αποτελέσματα στη δεξιά πλευρά της οθόνης χωρίς να χρειαστεί να ξαναφορτωθεί η σελίδα. Ένα εργαλείο που δεν ήταν απαραίτητο στη μεταπτυχιακή διατριβή μιας και δεν υπήρξε αυτή η λειτουργικότητα. Μια σημαντική διαφορά ήταν το γεγονός του ότι η μεταπτυχιακή διατριβή τραβούσε δυναμικά τις συντεταγμένες από απομακρυσμένη βάση ενώ η άλλη πλευρά φόρτωνε τις συντεταγμένες από ένα αρχείοxml.

Παρουσίαση των εργαλείων γραφήματος της Google και των χαρτών της Google σε μαθήματα οπτικοποίησης δεδομένων

Ο YingZhu καθηγητής στο πανεπιστήμιο της Γεωργίας (Georgia State) του τμήματος Επιστήμης των Υπολογιστών προτείνει τη χρήση των εργαλείων γραφήματος της Google (Google Chart Tools, <https://developers.google.com/chart>) καθώς και των χαρτών της Google (Google Maps API, <https://developers.google.com/maps>) για την οπτικοποίηση δεδομένων, θεωρώντας ότι παρέχουν περισσότερα οφέλη (YingZhu, 2012.).

Στο πανεπιστήμιο τα μαθήματα οπτικοποίησης δεδομένων όπως η οπτικοποίηση πληροφοριών (Information Visualization), οπτικοποίηση δεδομένων (Data Visualization), Οπτική ανάλυση (Visual Analytics) κ.α. αποσκοπούν κυρίως στο να διδάξουν τις θεωρίες ώστε να βοηθήσουν τους φοιτητές να μάθουν τα εργαλεία για την κατασκευή οπτικοποίησης δεδομένων.

Τα εργαλεία όμως που χρησιμοποιούνται έχουν ελλείψεις. Για παράδειγμα, οι φοιτητές μπορούν να πάρουν μόνο προσωρινές άδειες για εμπορικά εργαλεία όπως τα Tableau, Spotfire, Matlab και Mathematica. Επίσης στις περισσότερες περιπτώσεις, δε μπορούν να συνεχίσουν να χρησιμοποιούν αυτά τα εργαλεία μετά το τέλος της τάξης. Δωρεάν ή εργαλεία ανοικτού κώδικα συχνά στερούνται τεκμηρίωσης, σεμιναρίων (tutorials) ή δειγμάτων. Επιπλέον αυτά τα εργαλεία απαιτούν εξειδικευμένη αγορά. Οι εκπαιδευτές προτιμούν τη χρήση ευέλικτων εργαλείων, ευπροσάρμοστων, σύγχρονων/σταθερών (stable up-to-date), ανεξαρτήτων πλατφόρμας (cross-platform) και οι φοιτητές επιθυμούν εργαλεία σχετικά εύκολα στο να τα μάθουν, καλά τεκμηριωμένα και εμπορεύσιμα στο εγγύς μέλλον. Με αποτέλεσμα οι φοιτητές να έχουν λιγότερα κίνητρα και να επενδύουν πολύ λίγο χρόνο στην εκμάθησή τους. Λύση για τα προβλήματα αυτά ο YingZhu θεωρεί πως είναι οι σύγχρονες τεχνολογίες που αναφέρθηκαν.

Αναλυτικότερα, ο εκπαιδευτής δίδαξε το μάθημα της οπτικοποίησης των δεδομένων το φθινόπωρο του 2011 σε είκοσι φοιτητές. Το μάθημα επικεντρώθηκε στη 2D οπτικοποίηση δεδομένων και αποτελούταν από δύο μέρη το θεωρητικό και το πρακτικό. Οι φοιτητές έπρεπε να ολοκληρώσουν τέσσερις εργασίες θεωρητικού επιπέδου που σκοπό είχαν να εφαρμόσουν τις θεωρίες οπτικοποίησης σε παραδείγματα πραγματικού κόσμου και τέσσερις πρακτικές εργασίες που στόχος ήταν η πρακτική εμπειρία που θα αποκτούσαν από την κατασκευή διαφορετικών τύπων δεδομένων οπτικοποίησης. Το μάθημα και κατ' επέκταση και οι εργασίες στηρίχθηκαν στις τεχνολογίες: Εργαλεία γραφήματος της Google, Χάρτες της Google και τη γλώσσα προγραμματισμού Processing.

Στην πρώτη εργασία οι φοιτητές χρησιμοποίησαν το εργαλείο Tableau για την ανάλυση δύο συνόλων με δεδομένα και στη συνέχεια ανέφεραν τα πορίσματά και τις εμπειρίες τους. Στόχος ήταν να μάθουν οπτικές τεχνικές ανάλυσης μέσω εμπορικών εργαλείων οπτικοποίησης δεδομένων. Στη δεύτερη εργασία, δημιούργησαν δύο ταμπλό από διαδραστικά γραφήματα με τα εργαλεία γραφήματος της Google. Η πρώτη πηγή δεδομένων ήταν ένα υπολογιστικό φύλλο της Google που περιείχε δεδομένα χρονολογικών σειρών, τα δεδομένα αυτά προήλθαν από την ιστοσελίδα της Εφημερίδας Στατιστικής Εκπαίδευσης (Journal of Statistics Education website, www.amstat.org/publications/jse/v19n1/depaulo.pdf). Η δεύτερη πηγή ήταν δεδομένα της επιλογής των μαθητών. Στο τρίτο έργο, οι μαθητές χρησιμοποίησαν τους χάρτες της Google (Google Maps API) για να δημιουργήσουν ένα διαδραστικό χάρτη που απεικόνιζε πάνω από 400.000 γεωγραφικά κωδικοποιημένα άρθρα της Wikipedia, τα οποία ήταν αποθηκευμένα σε ένα συγχωνευμένο (fusion) πίνακα της Google. Το πρόγραμμα επέτρεπε στους χρήστες να διεξάγουν χωρικά ερωτήματα ώστε να λαμβάνουν οδηγίες προς οποιαδήποτε τοποθεσία άρθρων της Wikipedia στο χάρτη. Στο τέταρτο έργο, οι μαθητές χρησιμοποίησαν τη γλώσσα προγραμματισμού Processing για να σχεδιάσουν ένα διαδραστικό γράφημα από τα πιο πολυσύχναστα διεθνή δρομολόγια από το Hartsfield προς το διεθνές αεροδρόμιο Jackson της Atlanta.

Σύγκριση με τη Μεταπτυχιακή Διατριβή

Οι εργασίες που ανέθεσε ο καθηγητής YingZhu στους φοιτητές τους παρουσιάζουν ορισμένες ομοιότητες και διαφορές με τη μεταπτυχιακή διατριβή. Οι ομοιότητες που μπορούν να διακριθούν είναι πρώτον η κοινή προσπάθεια οπτικοποίησης δεδομένων 2D. Δεύτερον χρησιμοποιούνται τεχνολογίες της Google και πιο συγκεκριμένα οι χάρτες της Google. Και οι δύο περιπτώσεις εμφανίζουν ένα διαδραστικό χάρτη με τη προσθήκη δεικτών (markers) για την προβολή πληροφοριών.

Οι διαφορές που γίνονται αντιληπτές είναι αρχικά οι διάφοροι τρόποι οπτικοποίησης των δεδομένων όπως με τα εργαλεία γραφήματος της Google, τους χάρτες της Google και τη γλώσσα προγραμματισμού Processing. Στην μεταπτυχιακή διατριβή ο τρόπος οπτικοποίησης περιορίζεται στη χρήση των χαρτών της Google. Στη συνέχεια μια επιπλέον διαφορά είναι η προβολή διαφορετικών θεμάτων, οι φοιτητές ενσωμάτωσαν στο χάρτη άρθρα της Wikipedia ενώ στην μεταπτυχιακή διατριβή προβάλλονται πληροφορίες από μια κτηματομεσιτική πύλη. Επιπρόσθετα στις εργασίες που έπρεπε να εκπονήσουν οι φοιτητές του YingZhu έπρεπε να έχει ο χρήστης την επιλογή να θέσει κάποιο χωρικό ερώτημα κάτι που στην περίπτωση της μεταπτυχιακής διατριβής δεν υπάρχει. Ακόμα μια σημαντική διαφορά είναι ότι το γεωγραφικό μήκος και το γεωγραφικό πλάτος στη μεταπτυχιακή διατριβή φορτώνονται με τη βοήθεια του πρωτοκόλλου XMLRPC από μια βάση δεδομένων ενώ οι φοιτητές τραβούσαν τις συντεταγμένες από ένα συγχωνευμένο (fusion) πίνακα της Google.

Χρησιμοποιώντας τους χάρτες της Google με τη χαρτογράφηση εθνικών οδών (HIGHWAY MAPPING)

Με την εν λόγω εργασία (JamesD. Teresco (n.d).) ασχολήθηκε ο επίκουρος Καθηγητής της Επιστήμης των Υπολογιστών του Κολεγίου της Σιένα James D. Teresco . Σκοπός της είναι να δείξει πως χρησιμοποιούνται τα δεδομένα από έναν αυτοκινητόδρομο που λαμβάνονται από το έργο της χαρτογράφησης της εθνικής οδού (Clinched Highway Mapping, CHM), για να διερευνηθούν γραφικά δεδομένα και αλγόριθμοι και να εμφανιστούν τα αποτελέσματα με την χρήση των χαρτών της Google (Googlemap).

Το έργο CHM έχει συγκεντρώσει πληροφορίες σχετικά με τα δρομολόγια των αυτοκινητοδρόμων στη Βόρεια Αμερική και την Ευρώπη. Το πρόγραμμα επιτρέπει στους ταξιδιώτες να παρακολουθούν τα τμήματα του οδικού δικτύου που έχουν ταξιδέψει (ή, στην ορολογία, του έργου " Clinched "), να δουν αναπαραστάσεις από τα ταξίδια τους, υπό μορφή χάρτη, και να συγκρίνουν την έκταση των ταξιδιών τους με εκείνα των άλλων χρηστών του έργου. Τα δεδομένα που περιγράφουν τις εθνικές οδούς πρέπει να είναι αρκετά λεπτομερή για να επιτρέπουν στους χρήστες να καθορίσουν τα ταξίδια με ένα λογικό επίπεδο ακρίβειας, αλλά αρκετά τραχιά για να είναι σε θέση να παρουσιάσουν (σε μορφή λίστας και σε μορφή χάρτη) τα σημεία τέλους των τμημάτων του δρόμου ώστε να είναι διαθέσιμα να σημειωθούν ως clinched (available to be marked as clinched). Κάθε διαδρομή έχει προστεθεί στο έργο με τον ορισμό των συντεταγμένων του γεωγραφικού πλάτους και του γεωγραφικού μήκους των τελικών σημείων της διαδρομής και στις μεγάλες διασταυρώσεις πάρθηκαν τα σημεία κατά μήκος της διαδρομής. Οι εθελοντές του προγράμματος σχεδίασαν τον αυτοκινητόδρομο με λεπτομέρειες χρησιμοποιώντας κύριες πηγές δεδομένων όπως ημερολόγιο διαδρομών (route logs) από κρατικές υπηρεσίες μεταφοράς, δεδομένα χαρτών από το πρόγραμμα ανοιχτού δρόμου χάρτες (OpenStreetMap) και δορυφορικές εικόνες της κυβέρνησης.

Τα δεδομένα CHM του συστήματος του αυτοκινητόδρομου χρησιμοποιούνται για να παράγουν ένα γράφημα που αντιπροσωπεύει την εν λόγω εθνική οδό. Ένα πρόγραμμα προεπεξεργασίας διαβάζει το σύνολο των σημείων για κάθε διαδρομή και τις ταιριάζει με τις κοινές (shared) διασταυρώσεις με βάση τις συντεταγμένες. Τα σημεία (όλων των διαδρομών σε ένα σύστημα) είναι οι κορυφές του γραφήματος και τα τμήματα του δρόμου που συνδέουν γειτονικά σημεία είναι οι γραφικές ακμές. Κάθε κορυφή χαρακτηρίζεται με μια συμβολοσειρά βασισμένη στο όνομα των δρόμων και το όνομα των σημείων και αποθηκεύει τις συντεταγμένες

της. Οι άκρες χαρακτηρίζονται από το όνομα της διαδρομής που συνδέει τα τελικά σημεία. Οι αποστάσεις (μήκη ακμής) υπολογίζονται εύκολα από τις συντεταγμένες.

Τα συστήματα της εθνικής οδού που είναι διαθέσιμα στο έργο CHM ποικίλουν σε μέγεθος, δηλαδή τα δεδομένα του γραφήματος μπορεί να παραχθούν σε μια ποικιλία κλιμάκων. Για παράδειγμα, οι Διακρατικές εθνικές οδοί στη Χαβάη έχουν δημιουργήσει ένα γράφημα με 47 κορυφές και 48 ακμές, ενώ οι Διακρατικές εθνικές οδοί της Νέας Υόρκης των ΗΠΑ, και τα συστήματα κρατικών εθνικών οδών συνδυάζονται για να σχηματίσουν ένα γράφημα με 7265 κορυφές και 8416 ακμές.

Αυτά τα αρχεία γραφικών δεδομένων μπορούν να χρησιμοποιηθούν ως εισαγόμενα δεδομένα (input) για παραδείγματα τάξεων ή εκχωρήσεις (assignments) σε δομές δεδομένων και αλγορίθμων πορείας (course). Τα στοιχεία αυτά χρησιμοποιήθηκαν σε ένα εργαστήριο ανάθεσης για δομές δεδομένων πορείας στο Mount Holyoke College, όπου οι μαθητές έπρεπε να οικοδομήσουν μια γραφική δομή που αντιπροσωπεύει ένα σύστημα εθνικών οδών και στη συνέχεια, εκτέλεσαν τον αλγόριθμο του Dijkstra για να υπολογίσουν τη συντομότερη διαδρομή μεταξύ δύο συγκεκριμένων σημείων. Το πρόγραμμα των φοιτητών δημιουργεί αρχεία εξόδου του καταλόγου των τμημάτων του δρόμου κατά μήκος της συντομότερης διαδρομής. Αυτά στη συνέχεια φορτώθηκαν σε ένα course web server όπου χρησιμοποιήθηκε το Google Maps API για την απεικόνιση των αποτελεσμάτων. Οι μαθητές είχαν τη δυνατότητα να αναπτύξουν και να διορθώσουν τα προγράμματά τους χρησιμοποιώντας μικρά σύνολα δεδομένων, όπως τις διακρατικές οδούς της Χαβάης ενώ στη συνέχεια, χρησιμοποιήθηκαν τα προγράμματα αυτά για να υπολογίσουν πολύ πιο σύνθετες διαδρομές με μεγαλύτερα σύνολα δεδομένων.

Σύγκριση με τη Μεταπτυχιακή Διατριβή

Η παραπάνω εργασία και η μεταπτυχιακή διατριβή έχουν ομοιότητα μόνο στη χρήση των τεχνολογιών της Google και πιο συγκεκριμένα των χαρτών της Google. Οι δύο εργασίες διαφοροποιούνται αρκετά ως προς την διαχείριση αυτής της τεχνολογίας. Όσον αφορά τη συγκεκριμένη εργασία, στο κολέγιο Mount Holyoke οι φοιτητές έπρεπε οι οικοδομήσουν μια γραφική δομή που αντιπροσωπεύει ένα σύστημα εθνικών οδών. Για να πραγματοποιηθεί αυτό χρησιμοποιήθηκαν δεδομένα από έναν αυτοκινητόδρομο, δεδομένα που λήφθηκαν από το έργο της χαρτογράφησης εθνικής οδού (Clnched Highway Mapping, CHM). Κάθε διαδρομή προστέθηκε στο έργο με τον ορισμό των συντεταγμένων του γεωγραφικού πλάτους και του γεωγραφικού μήκους των τελικών σημείων της διαδρομής. Ένα πρόγραμμα προεπεξεργασίας διαβάζε το σύνολο των σημείων για κάθε διαδρομή και τις ταίριαζε με τις κοινές (shared) διασταυρώσεις με βάση τις συντεταγμένες. Τα σημεία (όλων των διαδρομών σε ένα σύστημα) ήταν οι κορυφές του γραφήματος και τα τμήματα του δρόμου που συνδέουν γειτονικά σημεία ήταν οι γραφικές ακμές. Μετά την δημιουργία του γραφήματος οι φοιτητές εκτέλεσαν τον αλγόριθμο του Dijkstra που είναι υπεύθυνος για τον υπολογισμό της συντομότερης διαδρομής μεταξύ δύο συγκεκριμένων σημείων. Τα αποτελέσματα της συντομότερης διαδρομής φορτώθηκαν σε ένα course web server και για την απεικόνισή τους χρησιμοποιήθηκε το Google Maps API.

Αντίθετα η χρήση του Googlemapapi στη μεταπτυχιακή διατριβή είχε διαφορετική χρήση. Αναλυτικότερα, η εφαρμογή παρέχει πληροφορίες στους χρήστες όσον αφορά τις πωλήσεις και τις ενοικιάσεις ακινήτων, ως πηγή δεδομένων χρησιμοποιήθηκε η διαδικτυακή υπηρεσία (webservice) μίας κτηματομεσιτικής πύλης. Για την εκτέλεση αυτής της διαδικασίας έγινε χρήση της τεχνολογίας XML-RPC, τεχνολογία υπεύθυνη για την ανταλλαγή δεδομένων. Σκοπός της εργασίας είναι να ενσωματωθούν τα δεδομένα που λήφθηκαν με τη βοήθεια της τεχνολογίας XML-RPC σε πραγματικό χρόνο στο GoogleMapAppi. Έγινε επιλογή των δεδομένων που κρίθηκαν απαραίτητα για την εφαρμογή και στη συνέχεια προστέθηκαν στο χάρτη markers (δείκτες) που «τραβούν» τις συντεταγμένες από τα δεδομένα της βάση. Οι markers λειτουργούν σαν κουμπιά που ο χρήστης μπορεί να πατήσει και να ανοίξει ένα παράθυρο (infowindow) με πληροφορίες των ακινήτων όπως κατηγορία, τιμή κ.τ.λ. Επιπλέον υπάρχουν δύο κουμπιά πώληση/ενοικίαση (sale/rent) δίπλα από τον χάρτη όπου ο χρήστης έχει τη δυνατότητα να

επιλέξει όποιο κουμπί επιθυμεί με αποτέλεσμα να εμφανίζονται στο χάρτη οι markers με τις αντίστοιχες αγγελίες.

Αναβάθμιση δεδομένων ανάλυσης TCABR και συστημάτων απόκτησης για την εξ αποστάσεως συμμετοχή χρησιμοποιώντας Java, XML, RPC και ένα σύγχρονο client / server, επικοινωνία / έλεγχος ταυτότητας

Τοέργοαυτό (LarssonandSprague, 2004.)

έχει υποστηριχθεί από το Εθνικό Συμβούλιο για την Επιστημονική και Τεχνολογική Ανάπτυξη (National Council for Scientific and Technological Development, CNPq) και το Ίδρυμα Ερευνών του κράτους του Σάο Πάολο (FAPESP). Αναλυτικότερα, σήμερα η επιστήμη έχει τη συμμετοχή πολλών επιστημόνων από διαφορετικές χώρες σε κοινά ερευνητικά προγράμματα με διαφορετικά εργαστήρια που εργάζονται από κοινού. Κάθε εργαστήριο έχει τον δικό του τρόπο για τον έλεγχο των πειραμάτων και την ανάκτηση των δεδομένων. Οι επιστήμονες, σε αυτό το πολύπλοκο περιβάλλον, πρέπει να μάθουν διαφορετικά προγράμματα προσέγγισης, και κωδικούς για την πρόσβαση στα πειραματικά δεδομένα και να κάνουν την ανάλυσή τους. Αυτό σημαίνει ότι θα πρέπει να προσαρμοστούν σε κάθε πειραματικό περιβάλλον. Υπάρχουν δύο επιπλέον προβλήματα: ο έλεγχος και το σύστημα απόκτησης δεδομένων που είναι κλειστά πίσω από ένα τείχος προστασίας ενώ η ανάλυση των δεδομένων είναι, σε όλες σχεδόν τις περιπτώσεις, ένα ιδιόκτητο σύστημα που αναπτύχθηκε από το εργαστήριο. Η TCABR tokamak (Tokamak Chauffage Alféh Brésilien είναι μια συσκευή που χρησιμοποιεί ένα μαγνητικό πεδίο για να περιορίζει το πλάσμα, μία από τις τέσσερις θεμελιώδεις καταστάσεις της ύλης, σε σχήμα σπείρας) έχει εγκατασταθεί στο Εργαστήριο Φυσικής του Πλάσματος (Plasma Physics) του Ινστιτούτου Φυσικής του Πανεπιστημίου του Σάο Πάολο της Βραζιλίας και είναι ένα από τα εργαστήρια που συμμετέχουν στο Συντονισμό Έρευνας του Έργου (Co-ordinating Research Project, CRP) στο «Κοινό Κέντρο Ερευνών με χρήση του μικρού τύπου tokamak" του Διεθνούς Οργανισμού Ατομικής Ενέργειας (International Atomic Energy Agency, IAEA). Η εξ αποστάσεως συμμετοχή διαδραματίζει έναν σημαντικό ρόλο στη σχέση μεταξύ των επιστημόνων. Αναπτύσσουν και εφαρμόζουν εξ αποστάσεως συμμετοχή τεχνολογιών που διατηρούν τον έλεγχο και τις τεχνικές απόκτησης δεδομένων και επιτρέπουν μια «διαφανή» πρόσβαση στα πειραματικά δεδομένα. Στόχος της εξ αποστάσεως συμμετοχής είναι να κρύψει όλη την πολυπλοκότητα των μεθόδων που αφορούν την αποθήκευση και την ανάκτηση των δεδομένων από τα εργαστήρια αλλά και το πώς ελέγχεται το πείραμα από τους τελικούς χρήστες. Αναπτύχθηκε ένα διακομιστής / πελάτης (server/client) σύστημα με απλές λειτουργίες που λειτουργεί ως γέφυρα μεταξύ του τελικού χρήστη και της βάσης δεδομένων του TCABR.

Το σύστημα απόκτησης δεδομένων TCABR (TCAqs) αναφέρεται σε όλες τις συσκευές, υπολογιστές και συστήματα πληροφορικής που υποστηρίζουν και καταγράφουν την δραστηριότητα TCABR tokamak. Τα TCAqs βασίζονται στο πρότυπο οργάνων VME και λειτουργικό σύστημα Motorola 68040 και Motorola 68060 CPU and OS9. Εφαρμόζουν ένα crate ACTA (Advanced Computing Architecture Telecom) [5] με επεξεργαστή Intel Core 2 Duo και λειτουργικό σύστημα Linux. Όσον αφορά την απόκτηση δεδομένων, είναι διαθέσιμες η ανάλυση από αναλογική σε ψηφιακή των (Analog to Digital Converter, ADCs modules) ενοτήτων των 12-bit.

Το λειτουργικό σύστημα Linux χρησιμοποιείται σε επιτραπέζιους υπολογιστές για τον έλεγχο και το σύστημα απόκτησης δεδομένων. Τα TCAqs είναι επικεντρωμένα γύρω από ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων (η MySQL έχει επιλεγεί για την TCABR), που επιτρέπει σε ένα χρήστη να αποθηκεύει, να χειραγωγεί και να ανακτά τα δεδομένα από ένα πίνακα δομών, που περιέχει όλες τις απαραίτητες πληροφορίες για τις δραστηριότητες του TCABR. Όλο το λογισμικό που απαιτείται για τον έλεγχο των μονάδων συλλογής δεδομένων έχει αναπτυχθεί σε τοπικό επίπεδο. Για λόγους μείωσης των σφαλμάτων μετάδοσης, την αύξηση της αποτελεσματικότητας της αποθήκευσης και προκειμένου να παρέχουν ταχύτερη πρόσβαση δεδομένων, τα δεδομένα συμπιέζονται για την ευκολότερη διαχείριση του όγκου. Η

μέθοδος που επιλέχθηκε ήταν μια προσαρμογή των Lempel-Ziv-Welch (LZW), ενός αλγορίθμου συμπίεσης χωρίς απώλεια δεδομένων .

Πριν από την εφαρμογή της εξ αποστάσεως συμμετοχής των τεχνολογιών και προκειμένου να υπάρχει πρόσβαση στα δεδομένα τα οποία είναι σε συμπιεσμένη μορφή και στις πληροφορίες που βρίσκονται στη βάση δεδομένων, γράφτηκε το λογισμικό Γραφικά Διεπαφής Χρήστη (Graphics User Interface, GUI) με τη χρήση της γλώσσας C + +. Αυτό το λογισμικό λειτουργεί ως γέφυρα μεταξύ των δεδομένων και τον τελικό χρήστη, επιτρέποντας την επεξεργασία δεδομένων, την μεταφορά και την αποθήκευση. Αυτό το σύστημα βρίσκεται πίσω από ένα τείχος προστασίας.

Το σύστημα TCAqs σχεδιάστηκε παρέχοντας ασφάλεια για τον έλεγχο του πειράματος και την πρόσβαση των δεδομένων που διαχωρίζει τη περιοχή tokamak, τον έλεγχο και την αίθουσα συστήματος απόκτησης δεδομένων από το παγκόσμιο δίκτυο με ένα τείχος προστασίας. Αυτό το τείχος προστασίας δεν επιτρέπει την πρόσβαση από το παγκόσμιο δίκτυο, αλλά είναι δυνατόν να σταλούν πληροφορίες και δεδομένα από το τοπικό δίκτυο στον κόσμο. Το λογισμικό που έχει αναπτυχθεί, επιτρέπει στους χρήστες να εργάζονται μόνο στην αίθουσα συστήματος ελέγχου και απόκτησης δεδομένων. Για την πρόσβαση της εξ αποστάσεως συμμετοχής, η TCAqs έχει επανασχεδιαστεί με την προσθήκη εγκαταστάσεων αντί της πλήρους αλλαγής του συστήματος.

Το νέο σύστημα βασίζεται στην σχέση ενός διακομιστή (server) και ενός πελάτη (client). Έχει αναπτυχθεί με μια ανεξάρτητη πλατφόρμα αρχιτεκτονικής, και έχει αντικειμενοστρεφή εφαρμογή της γλώσσας προγραμματισμού Java. Το νέο σύστημα TCAqs βρίσκεται σε αρχικό στάδιο και μπορεί να διαιρεθεί σε τρία μέρη: πρώτον το σύστημα διακομιστή (TCABRServer), δεύτερον τη μονάδα ελέγχου ταυτότητας και ασφάλειας και τρίτον το σύστημα πελάτη (TCABRClient).

Σύστημα διακομιστή (TCABRServer)

Το σύστημα διακομιστή (TCABRServer) λειτουργεί ως στρώμα διεπαφής μεταξύ της βάσης δεδομένων και των πελατών. Αυτό αποτελείται από τρεις ενότητες: την τεχνολογία XML-RPC, το λογισμικό TCABRServer και τη βιβλιοθήκη TCABRServer. Αναλυτικότερα, όσον αφορά την τεχνολογία XML-RPC αυτή είναι μια απομακρυσμένη διαδικασία κλήσης που επιτρέπει να εκτελούνται λειτουργίες σε έναν απομακρυσμένο διακομιστή και να λαμβάνονται τιμές από αυτόν. Το XML-RPC είναι μια απομακρυσμένη κλήση διαδικασίας που χρησιμοποιεί το HTTP για τη μεταφορά και το XML για την κωδικοποίηση. Έχε σχεδιαστεί να είναι όσο το δυνατόν απλούστερο και να επιτρέπει σε πολύπλοκες δομές δεδομένων να διαβιβάζονται, να επεξεργάζονται και να επιστρέφονται. Στην αναβάθμιση TCAqs χρησιμοποιείται το Apache XML-RPC. Πρόκειται για μια XML-RPC εφαρμογή της Java, ένα δημοφιλές πρωτόκολλο που χρησιμοποιεί XML μέσω HTTP. Ο πρωταρχικός λόγος για την επιλογή XML-RPC είναι ότι είναι κατάλληλο για χρήση σε ακαδημαϊκές εγκαταστάσεις όπου υπάρχουν διάφορες πλατφόρμες υπολογιστών. Σε αυτό το στάδιο της ανάπτυξης, η αναβάθμιση των TCAqs δεν χρειάζεται την πολυπλοκότητα άλλων μεθόδων και η μεταφορά των μηνυμάτων δεν χρειάζεται να είναι σε δυαδική μορφή. Το λογισμικό TCABRServer είναι ένας κώδικας Java που τρέχει στο μηχάνημα του διακομιστή και υλοποιεί το XML-RPC. Ενώ η βιβλιοθήκη TCABRServer είναι μια ομάδα από κώδικες της Java που διαβάζουν τους TCABR shot αριθμούς, βρίσκουν δεδομένα από τον διακομιστή αποθήκευσης δεδομένων, αποσυμπιέζουν τα ζητούμενα δεδομένα και μεταφέρουν τις αναγκαίες πληροφορίες από αυτά τα δεδομένα. Το λογισμικό TCABRServer χρησιμοποιεί αυτή τη βιβλιοθήκη όταν οι πελάτες ζητούν πληροφορίες.

Μονάδα ελέγχου ταυτότητας και ασφάλειας

Σε αυτό το πρώιμο στάδιο της αναβάθμισης των TCAqs ο έλεγχος της ταυτότητας δημιουργήθηκε κάνοντας μια εξουσιοδότηση στην διεύθυνση IP του πελάτη στο διακομιστή της μηχανής όπου τρέχει το TCABRServer. Η διεύθυνση IP του πελάτη θέτει άδεια πρόσβασης στη βάση δεδομένων του διακομιστή της μηχανής.

Σύστημα πελάτη (TCABRClient)

Το σύστημα πελάτη (TCABRClient) είναι το μέρος για τους τελικούς χρήστες της αναβάθμισης TCAqs. Είναι μια ομάδα κωδικών της Java που σχηματίζουν μια βιβλιοθήκη που μπορεί να χρησιμοποιηθεί σε μια πλατφόρμα που έχει εγκατεστημένη την εικονική μηχανή της Java (JavaVirtualMachine, JVM) ή το λογισμικό που είναι το front-end αυτής της βιβλιοθήκης. Η βιβλιοθήκη TCABRClient αποτελείται από τρία είδη λειτουργιών: Πρώτον την σύνδεση με το TCABRServer, αυτή η λειτουργία ανοίγει μια σύνδεση του πελάτη με τον απομακρυσμένο διακομιστή. Δεύτερον την εύρεση του αριθμού στη βάση δεδομένων του διακομιστή, αυτή η λειτουργία προετοιμάζει τον πελάτη να επεξεργαστεί τα δεδομένα που ανήκουν σε ένα συγκεκριμένο πλάνο ενδιαφέροντος. Τρίτον το χειρισμό των δεδομένων (data manipulation, η διαδικασία ταξινόμησης, αναδιάταξης, μετακίνησης των ερευνητικών δεδομένων χωρίς να αλλάζουν), αυτό είναι μια ομάδα λειτουργιών που δουλεύει με δεδομένα που ανήκουν σε ένα συγκεκριμένο πλάνο ενδιαφέροντος. Αυτές οι λειτουργίες στέλνουν αιτήματα στο TCABRServer και επιστρέφονται πληροφορίες ή δεδομένα.

Σύγκριση με τη Μεταπτυχιακή Διατριβή

Η μεταπτυχιακή διατριβή και η παραπάνω εργασία βασίζονται σε μια κοινή τεχνολογία την XML-RPC για την ανταλλαγή δεδομένων. Όσον αφορά την παραπάνω εργασία, για την υποστήριξη ενός κοινού ερευνητικού προγράμματος με την χρήση της εξ αποστάσεως συμμετοχής των τεχνολογιών αναβαθμίστηκε το σύστημα απόκτησης και ανάλυσης δεδομένων TCABR. Η αρχιτεκτονική του νέου συστήματος χρησιμοποιεί ως περιβάλλον προγραμματισμού τη γλώσσα Java. Αφού οι παράμετροι της εφαρμογής και το υλικό (hardware) σε ένα κοινό πείραμα είναι ένα συγκρότημα με μεγάλη μεταβλητότητα στοιχείων, απαιτήσεων και προσδιορισμένων λύσεων πρέπει να είναι ευέλικτα και αρθρωτά (modular) ανεξάρτητα από το λειτουργικό σύστημα και την αρχιτεκτονική του υπολογιστή. Για να περιγραφούν και να οργανωθούν οι πληροφορίες όλων των εξαρτημάτων και των συνδέσεων τους, τα συστήματα αναπτύχθηκαν χρησιμοποιώντας την τεχνολογία eXtensibleMarkupLanguage (XML). Η επικοινωνία μεταξύ των πελατών και των εξυπηρετητών πραγματοποιείται με την κλήση απομακρυσμένης διαδικασίας (RPC) με βάση την XML (XML-RPC τεχνολογία). Η ολοκλήρωση της γλώσσας προγραμματισμού java, της XML και της τεχνολογίας XML-RPC επιτρέπει να αναπτυχθεί εύκολα ένα πρότυπο στρώμα πρόσβασης δεδομένων και επικοινωνίας μεταξύ χρηστών και εργαστηρίων χρησιμοποιώντας κοινές βιβλιοθήκες λογισμικού και εφαρμογές ιστού (Web). Οι βιβλιοθήκες επιτρέπουν την ανάκτηση δεδομένων χρησιμοποιώντας τις ίδιες μεθόδους για όλους τους χρήστες των εργαστηρίων με κοινή συνεργασία και η εφαρμογή Web επιτρέπει μια απλή πρόσβασή με το GUI.

Η μεταπτυχιακή διατριβή ασχολείται με την δημιουργία μιας εφαρμογής με τη χρήση χαρτών της google (googlemapapplication), η οποία παρέχει πληροφορίες για πωλήσεις-ενοικιάσεις ακινήτων στους επισκέπτες. Η διαδικασία έχει ως εξής : με την βοήθεια της τεχνολογίας XML-RPC ανακτώνται δεδομένα από μια διαδικτυακή υπηρεσία μιας κτηματομεσιτικής πύλης, δημιουργήθηκε δηλαδή ένα XML-RPCrequest το οποίο συνδέθηκε με τη βάση της πύλης και καλούσε τα δεδομένα της. Στη συνέχεια τα δεδομένα αυτά εμφανίστηκαν σε ένα Googlemap με τη μορφή ενός infowindow στους ενσωματωμένους δείκτες του χάρτη. Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε ένα πλαίσιο ανοιχτού κώδικα PHP (OpensourcePHPframeworks), το Codeigniter, το οποίο είναι βασισμένο στο αρχιτεκτονικό πρότυπο Μοντέλο-Άποψη-Ελεγκτής (Model-View-Controller, MVC).

Εξαιρετική Ανάπτυξη: Διανομή και Διαμόρφωση 450 Φορητών υπολογιστών σπουδαστών σε πέντε ώρες

Η πρώτη χρονιά που το πανεπιστήμιο Ντρου (Drew) παρείχε φορητούς υπολογιστές με WindowsXP στους φοιτητές του, ήταν τη σχολική χρονιά 2002-2003 (W.P. deSá, 2010.). Δεν υπήρχε Ενεργός Τομέας Καταλόγου (ActiveDirectorydomain, ADd, υπηρεσία καταλόγου που παρέχει μια κεντρική θέση για τη διαχείριση δικτύου και την ασφάλεια) στην πανεπιστημιούπολη και έπρεπε οι φοιτητές να χρησιμοποιούν τοπικούς λογαριασμούς στους φορητούς υπολογιστές τους. Οι φοιτητές ήταν σε θέση να πιστοποιήσουν την ταυτότητα τους στο Novell eDirectory (είναι μια σειρά από πρότυπα δικτύωσης υπολογιστών που καλύπτουν ηλεκτρονικές υπηρεσίες καταλόγου, X.500- συμβατή με την υπηρεσία καταλόγου των προϊόντων λογισμικού που κυκλοφόρησε για την κεντρική διαχείριση πρόσβασης πόρων σε πολλούς servers και υπολογιστές μέσα σε ένα δεδομένο δίκτυο) μέσω του Novell Client (σταθμοί εργασίας λογισμικού που παρέχον NetWare -λειτουργικό σύστημα δικτύου H/Y - σύνδεση στα Windows), αλλά αυτό τους παρείχε πρόσβαση μόνο στους πόρους του δικτύου. Οι φορητοί υπολογιστές είχαν ρυθμιστεί με ένα γενικό λογαριασμό, ονόματι χρήστη Ντρου (Drew User), στο οποίο μπορούσαν να συνδεθούν από προεπιλογή. Αυτό προκάλεσε πολλά προβλήματα καθ' όλη τη διάρκεια του σχολικού έτους. Ο λογαριασμός του χρήστη Ντρου δεν είχε κωδικό, αφήνοντας τους υπολογιστές χωρίς ασφάλεια. Οι φοιτητές πρόσθεταν έναν κωδικό πρόσβασης στον λογαριασμό του χρήστη Ντρου, αλλάζοντας μερικές φορές το κωδικό πρόσβασης του λογαριασμού του Διαχειριστή και αφού ξεχνούσαν και τους δύο κωδικούς, κλείδωναν τις μηχανές τους. Μερικοί φοιτητές δημιουργήσουν ένα νέο λογαριασμό, διαγράφοντας τον λογαριασμό του χρήστη Ντρου, χρησιμοποιώντας αντίθετα ένα λογαριασμό που δεν ήταν προσαρμοσμένος στο προφίλ του Mozilla (ανοικτού κώδικα φυλλομετρητής του παγκόσμιου ιστού) και των άλλων απαραίτητων ρυθμίσεων.

Μόλις τοποθετήθηκε μια δομή ADd και διαμορφώθηκε για να λειτουργεί με την υπάρχουσα δομή των υπηρεσιών καταλόγου της Novell (Novell Directory Services, NDS), ήταν σε θέση να συνδεθούν από μόνοι τους οι φοιτητές. Έγινε μια ενημερωμένη έκδοση ρύθμισης παραμέτρων για τους φορητούς υπολογιστές χρησιμοποιώντας το βοηθητικό πρόγραμμα προετοιμασίας συστήματος της Microsoft Sysprep (μία τεχνολογία που μπορεί να χρησιμοποιηθεί με άλλα εργαλεία ανάπτυξης για την εγκατάσταση λειτουργικών συστημάτων Microsoft Windows με ελάχιστη παρέμβαση από έναν διαχειριστή ή τεχνικό). Οι φοιτητές συνδέθηκαν μια φορά με το ίδιο όνομα χρήστη και τον ίδιο κωδικό πρόσβασης στο δίκτυο Novell και στους φορητούς τους υπολογιστές. Τα διαπιστευτήρια θα αποκρύπτονταν, επιτρέποντας στους φοιτητές να συνεχίσουν να το χρησιμοποιούν και εκτός του δικτύου της πανεπιστημιούπολης. Με το να είναι στον τομέα (domain), το τμήμα εξυπηρέτησης πελατών είχε περισσότερες μεθόδους για να φτιαχτεί ένας φορητός υπολογιστής όπου δεν ήταν διαθέσιμος ο λογαριασμός του φοιτητή και ο λογαριασμός του διαχειριστή. Οι φοιτητές δεν θα μπορούσαν πλέον να «κλειδώσουν» τις μηχανές τους τόσο εύκολα.

Όταν ετοιμαζόταν η εικόνα των φορητών υπολογιστών για το σχολικό έτος 2003-2004 ένα νέο πρόβλημα αναγνωρίστηκε. Όταν σε ένα μηχάνημα είχε τοποθετηθεί η σύγχρονη διαμόρφωση από το τμήμα εξυπηρέτησης ένας φοιτητής/υπάλληλος ήταν εκεί να καθοδηγήσει τον υπολογιστή μέσα στη μίνι οργάνωση που «έτρεχε» αφού πρώτα είχε φορτωθεί στη μνήμη, τερματίζοντας τις εργασίες του Sysprep. Ενώ η μίνι εγκατάσταση είναι σχετικά φιλική προς τον χρήστη και αυτοεξηγούμενη δεν γινόταν να χρησιμοποιηθεί για την τάξη που ερχόταν. Οι φοιτητές δεν θα ονόμαζαν τους υπολογιστές τους σωστά, θα συνδέονταν στο domain λανθασμένα ή δεν θα μπορούσαν να είναι οι διαχειριστές στις δικές τους μηχανές. Εξετάστηκαν κάποιες εναλλακτικές, αρχικά να χρησιμοποιηθούν φοιτητές/υπάλληλοι για να προσαρμόσουν το κάθε φορητό υπολογιστή σε ένα συγκεκριμένο φοιτητή. Στα πακέτα «commercialimagingsoftware» απαιτούνταν να γνωρίζουν προκαταβολικά σε ποιόν φοιτητή άνηκε ο κάθε φορητός υπολογιστής. Βάσει της εφοδιαστικής (logistics) διαδικασίας διανομής, αυτό δεν ήταν αποδεκτή επιλογή. Εξετάστηκαν μέθοδοι να βελτιωθεί η διαδικασία, όπως το να εμφανίζονται οι φοιτητές με μια προεπιλεγμένη σειρά, τοποθετώντας τους φορητούς υπολογιστές στην ίδια σειρά ανάλογα με την ώρα που θα έφταναν. Όλες αυτές οι πιθανότητες εξετάστηκαν προσεκτικά και θεωρήθηκαν μη πρακτικές.

Τελικά αποφασίστηκε να σχεδιαστεί μια μέθοδος ανάπτυξης που θα αναπαριστά τα χαρακτηριστικά του Sysprep και θα είναι αυτοματοποιημένη ώστε να απαιτείται όσο το δυνατόν λιγότερη παρέμβαση του χρήστη. Θα πρέπει να είναι αρκετά ευέλικτη και προσαρμοσμένη στις ανάγκες του εκάστοτε φοιτητή.

Κατά τη διάρκεια του καλοκαιριού, πριν την έναρξη της σχολικής χρονιάς γίνεται μια κατά παραγγελία διαμόρφωση του φορητού υπολογιστή. Το πρότυπο (standard) λογισμικό του Ντρου τοποθετείται σ' αυτή τη διαμόρφωση καθώς είναι ο κυρίαρχος πελάτης της ανάπτυξης (eXtreme Deployment client, το eXtreme Deployment είναι ένα σύστημα που έχει κτιστεί στην κορυφή διάφορων ανοιχτού κώδικα και εμπορικών τεχνολογιών. Η κύρια έμφαση του σχεδιασμού για το έργο ήταν η ταχεία ανάπτυξη και η ευκολία τροποποίησης για την υποστήριξη όλων των τυπικών διαμορφώσεων Ντρου για τον Η/Υ). Αυτή η διαμόρφωση δίνεται στον πωλητή ο οποίος εφοδίαζε όλους τους φορητούς υπολογιστές μ' αυτή. Κατά την διάρκεια του «προσανατολισμού» των νέων φοιτητών, οι εισερχόμενοι φοιτητές λαμβάνουν έναν υπολογιστή και τους δίνεται μια χρονική περίοδος κατά την οποία μπορούν να τους παραλάβουν. Στην διανομή πιστοποιείται η καταλληλότητά τους και δίνεται ένας φορητός υπολογιστής και αξεσουάρ. Όταν σε έναν φοιτητή δίνεται ένας φορητός υπολογιστής, ο σειριακός αριθμός και η ταυτότητα του φοιτητή εισάγονται σε μια δικτυακή εφαρμογή από τους φοιτητές/υπαλλήλους. Αυτό αναθέτει το φορητό υπολογιστή σε utrack , δημιουργεί έναν λογαριασμό για τον φορητό υπολογιστή στην πανεπιστημιούπολη ADd και συσχετίζει το notebook με τον λογαριασμό AD (Active Directory) και τον ιδιοκτήτη στη βάση δεδομένων της ανάπτυξης.

Ο φοιτητής ξεπακετάρει τον φορητό υπολογιστή του, τον συνδέει με το δίκτυο της πανεπιστημιούπολης που υπάρχει στο δωμάτιο του και το φορτώνει στη μνήμη. Ο φορητός υπολογιστής αυτόματα συνδέεται με έναν τοπικό λογαριασμό, που ονομάζεται Deploy, η οποία διαχειρίζεται την Κυρίαρχη Ανάπτυξη εκκίνησης (eXtreme Deployment bootstrap). Η διαδικασία extremedeployment έρχεται σε επαφή με τον διακομιστή της ανάπτυξης, ελέγχοντας αν υπάρχουν νεότερες εκδοχές των στρώματων σ' αυτό το μηχάνημα. Αν υπάρχουν νεότερες εκδοχές, τα σύγχρονα στρώματα κατεβάζονται και τοποθετούνται.

Στη συνέχεια ένα τοπικό στιγμιότυπο (instance) του HTTP διακομιστή Apache ξεκινάει και ανοίγεται σε ένα παράθυρο προγράμματος περιήγησης σε πλήρη οθόνη. Ο πελάτης αποκτά την πληροφορία του στοιχείου της ετικέτας (assettag) από την SMBIOS –σύστημα διαχείρισης BIOS (βασικό σύστημα εισόδου/εξόδου) που ορίζει δομές δεδομένων σε ένα BIOS που επιτρέπει σε ένα χρήση ή εφαρμογή να αποθηκεύει και να ανακτά πληροφορίες για τον εν λόγω Η/Υ-του μηχανήματος και τη μεταδίδει στον διακομιστή ανάπτυξης σε ένα αίτημα XML-RPC. Η πληροφορία που εισάγεται στον διακομιστή ανάπτυξης από το σενάριο (script) επιστρέφεται δίνοντας στο πελάτη ό,τι χρειάζεται για την προσαρμογή της μηχανής. Μόλις ο φοιτητής ελέγξει την εγκυρότητα της πληροφορίας, ο υπολογιστής μετονομάζεται κατάλληλα.

Ένα λογισμικό δημιουργίας SID «τρέχει» για να αλλάξει το SID (security identifier-αναγνωριστικό ασφαλείας) του συστήματος και ολοκληρώνεται με μια επανεκκίνηση. Μετά τη επανεκκίνηση του υπολογιστή ο φοιτητής εισχωρεί τα διαπιστευτήρια του για να συνδέσει τον υπολογιστή με το λογαριασμό του Active Directory domain που δημιουργήθηκε για αυτόν. Μετά από μια δεύτερη και τελική επανεκκίνηση για την ολοκλήρωση της διαδικασίας συμμετοχής του υπολογιστή στο domain, ο φοιτητής προστίθεται στην ομάδα των τοπικών διαχειριστών. Στο λογαριασμό των τοπικών διαχειριστών δίνεται ένας τυχαίος κωδικός και αυτός ο κωδικός αποθηκεύεται στο διακομιστή ανάπτυξης σε μια ασφαλή βάση δεδομένων. Η αυτόματη σύνδεση αχρηστεύεται όπως και ο κωδικός πρόσβασης και το Deploy αποσυνδέεται αφήνοντας στον χρήστη ένα πλήρως ανεπτυγμένο μηχάνημα. Την πρώτη φορά που ο φοιτητής συνδέεται στο μηχάνημα ένα προφίλ Mozilla δημιουργείται για αυτόν με προσαρμοσμένες ρυθμίσεις email και κατάλληλους σελιδοδείκτες Ντρου. Το σενάριο δημιουργού του Mozilla είναι μέρος του προεπιλεγμένου (default) προφίλ χρήστη των Windows και επομένως εκτελείται επίσης για οποιονδήποτε επόμενο χρήστη που συνδέεται.

Σύγκριση με τη Μεταπτυχιακή Διατριβή

Στην παραπάνω εργασία και στη μεταπτυχιακή διατριβή η σύγκριση θα βασιστεί στην τεχνολογία που χρησιμοποιείται και στις δύο περιπτώσεις, την XML-RPC. Η διαδικασία που ακολουθήθηκε στο πανεπιστήμιο Ντρου αφορούσε την σχεδίαση μιας μεθόδου ανάπτυξης που θα αναπαριστά τα χαρακτηριστικά του Sysprep, ώστε να παρέχεται στους φοιτητές ασφαλή και συνεχόμενη πρόσβαση στο δίκτυο της πανεπιστημιούπολης και την κατάλληλη προσαρμογή των φορητών υπολογιστών. Οι φοιτητές θα ήταν διαχειριστές των μηχανών τους.

Διαμορφώθηκαν λοιπόν οι υπολογιστές των φοιτητών με βάση το σύστημα eXtreme Deployment. Στη συνέχεια οι Η/Υ συνδέονται με το δίκτυο της πανεπιστημιούπολης και αυτόματα συνδέονται με έναν τοπικό λογαριασμό, που ονομάζεται Deploy. Πραγματοποιείται μια τοπική παράκληση (instance) του Apache και ανοίγεται σε ένα παράθυρο προγράμματος περιήγησης σε πλήρη οθόνη. Ο πελάτης αποκτά την πληροφορία του στοιχείου της ετικέτας (assettag) από την SMBIOS του μηχανήματος και τη μεταδίδει στον διακομιστή ανάπτυξης σε ένα αίτημα XML-RPC. Η πληροφορία που εισάγεται στον διακομιστή ανάπτυξης από το script επιστρέφεται δίνοντας στο πελάτη ότι χρειάζεται για την προσαρμογή της μηχανής. Μόλις ο φοιτητής ελέγξει την εγκυρότητα της πληροφορίας, ο υπολογιστής μετονομάζεται κατάλληλα.

Ένα SID generator utility «τρέχει» για να αλλάξει το SID του συστήματος και ολοκληρώνεται με μια επανεκκίνηση. Μετά τη επανεκκίνηση του υπολογιστή ο φοιτητής εισχωρεί τα διαπιστευτήρια του για να συνδέσει τον υπολογιστή με το λογαριασμό του Active Directory domain που δημιουργήθηκε για αυτόν. Παρατηρείται στην εργασία αυτή η απαραίτητη χρήση της τεχνολογίας XML-RPC που είναι υπεύθυνη για την ανάκτηση πληροφοριών. Η λειτουργία αυτή εκτελείται και στη μεταπτυχιακή διατριβή με τη δημιουργία ενός XML-RPCrequest το οποίο συνδέθηκε με τη βάση μιας πύλης και καλούσε τα δεδομένα της. Στην παραπάνω εργασία χρησιμοποιήθηκαν τεχνολογίες που δεν υπάρχουν στην μεταπτυχιακή διατριβή όπως το σύστημα eXtremeDeployment. Μια διαφορά αποτελεί και ο τρόπος χρήσης της τεχνολογίας XML-RPC. Στο πανεπιστήμιο Ντρου ήταν απαραίτητη η χρήση της για την ανάκτηση πληροφοριών απαραίτητων για την προσαρμογή της μηχανής των φοιτητών σε αντίθεση με τη μεταπτυχιακή διατριβή που χρησιμοποίησε την τεχνολογία αυτή για την προβολή δεδομένων σε έναν ιστότοπο όπως αναλυτικότερα αναφέρθηκε στην προηγούμενη σύγκριση εργασιών.

Στο κεφάλαιο αυτό, πραγματοποιήθηκε σύγκριση της εν λόγω πτυχιακής εργασίας με εργασίες οι οποίες χρησιμοποιούν κοινές τεχνολογίες όπως το Googlemap και το πρότυπο XML-RPC. Η διαδικασία αυτή είναι ενδιαφέρουσα και σημαντική καθώς διαπιστώνεται πως αξιοποίησε ο καθένας τις τεχνολογίες αυτές αλλά και το θέμα στο οποίο τις προσάρμοσαν. Δίνοντας έτσι το πλεονέκτημα να μπορεί κάποιος να υιοθετήσει να προσαρμόσει ακόμα και να εξελίξει κάποιες από τις παραπάνω ιδέες εκτός από τις περαιτέρω γνώσεις που αποκτά. Στην συνέχεια ακολουθεί μια αναλυτική αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν για την εκτέλεση της εφαρμογής.

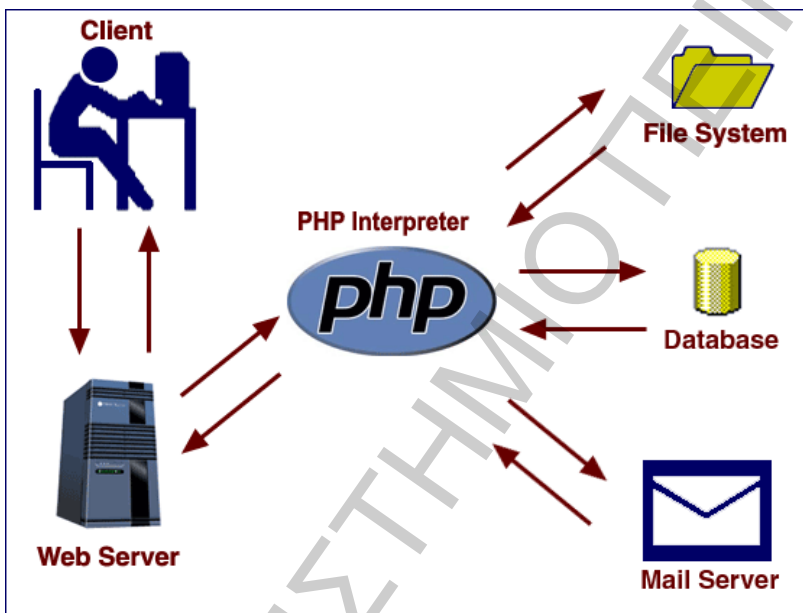
3. ΤΕΧΝΟΛΟΓΙΕΣ

3.1. PHP

Η **PHP** είναι μια γλώσσα προγραμματισμού που σχεδιάστηκε για τη δημιουργία δυναμικών σελίδων στο διαδίκτυο και είναι επισήμως γνωστή ως: **Υπερκείμενο Προεπεξεργαστή (HyperTextprocessor)**.

Είναι μια server-side (εκτελείται στον διακομιστή) γλώσσα σεναρίων (scripting) που γράφεται συνήθως πλαισιωμένη από **HTML**, για μορφοποίηση των αποτελεσμάτων. Αντίθετα από μια συνηθισμένη HTML σελίδα, η σελίδα PHP δε στέλνεται άμεσα σε έναν πελάτη (client), άντ' αυτού πρώτα αναλύεται και μετά αποστέλλεται το παραγόμενο αποτέλεσμα. Τα στοιχεία HTML στον πηγαίο κώδικα μένουν ως έχουν, αλλά ο PHP κώδικας ερμηνεύεται και εκτελείται.

Ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να δημιουργήσει εικόνες, να διαβάσει και να γράψει αρχεία, να συνδεθεί με απομακρυσμένους υπολογιστές, κ.ο.κ. Σε γενικές γραμμές οι δυνατότητες που δίνει είναι απεριόριστες.



Σχήμα 1. Σχεδιάγραμμα λειτουργίας PHP.

Η PHP χρησιμοποιεί μια μίξη από διερμηνευση (interpretation) και μεταγλώττιση (compilation) έτσι ώστε να μπορέσει να δώσει στους προγραμματιστές τον καλύτερο δυνατό συνδυασμό απόδοσης και ευελιξίας. Στο παρασκήνιο, η PHP μεταγλωττίζει το script σε μια σειρά από εντολές (instructions), που είναι γνωστές με τον όρο opcodes, οι οποίες εντολές εκτελούνται μία-μία μέχρι να τελειώσει το script. Αυτή η συνεχής μεταγλώττιση μπορεί να φαίνεται ως απώλεια χρόνου, αλλά δεν είναι καθόλου κακή καθώς δε χρειάζεται να κάνουμε συνέχεια εμείς τη μεταγλώττιση των scripts όταν γίνονται κάποιες αλλαγές σ' αυτά.

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο RasmusLerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό σενάριο με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Το 1997 οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, έτσι η PHP έφθασε στην έκδοση η οποία θύμιζε περισσότερο τη σημερινή μορφή της. (Νώτης Χρήστος, 2011.)

Για την κατανόηση της PHP ακολουθεί ένα παράδειγμα:

```
<html>
  <body>
    <script type="text/javascript">
      mypointsLat =newArray();
      mypointsLong = new Array();
    </script>
    <?phpforeach ($listings as $listing):?>
      <script type="text/javascript">
        mypointsLong[<?php echo $listing["key"]; ?>] = <?php echo $listing["lng"]; ?>;
        mypointsLat[<?php echo $listing["key"]; ?>] = <?php echo $listing["lat"]; ?>;
      </script>
    <?phpendforeach;?>
  </body>
</html>
```

Αρχικά με την βοήθεια της Javascript ορίζουμε δύο μεταβλητές ως πίνακες. Στην συνέχεια με τη χρήση της εντολής `foreach` της PHP δημιουργούμε μια επανάληψη (loop) για να διαβάσουμε κάθε στοιχείο από τη λίστα με τα δεδομένα που επιθυμούμε να «τραβήξουμε» από τη βάση. Στις μεταβλητές που ορίσαμε με την εντολή της PHP `echo` ορίζουμε τι θα εμφανίζει η μεταβλητή αυτή και συνεχίζει παρόμοια η διαδικασία.

Τα πλεονεκτήματα της PHP είναι πολλά, το κυριότερο από αυτά είναι ότι προσφέρεται εντελώς δωρεάν. Επιπρόσθετα προσφέρει ασφάλεια που βοηθά στην πρόληψη κακόβουλων επιθέσεων. Η PHP διαθέτει ένα μεγάλο αριθμό από βιβλιοθήκες και επεκτάσεις. Ακόμα ένα σημαντικό πλεονέκτημα είναι πως είναι φιλικά σχεδιασμένη προς το χρήστη. Παρέχει υψηλή συμβατότητα με κορυφαία λειτουργικά συστήματα και διακομιστές web, δίνοντας έτσι τη δυνατότητα να μπορεί να εύκολα να αναπτυχθεί σε πολλές διαφορετικές πλατφόρμες όπως Linux, Windows κτλ. και επιπλέον αρκετές εργασίες διαδικτύου μπορούν εύκολα να εκτελεστούν χρησιμοποιώντας PHP (MaestroSolutionsLTD, 2004-2013.).

3.2. Πλαίσιο CodeIgniter

Το CodeIgniter είναι ένα πλαίσιο ανοικτού κώδικα που χρησιμοποιείται για τη δημιουργία δυναμικών ιστοσελίδων με PHP. Σκοπός του είναι να επιτρέψει στους προγραμματιστές να αναπτύξουν έργα (projects) πολύ πιο γρήγορα από τη σύνταξη κώδικα από το μηδέν. Παρέχει ένα πλούσιο σύνολο βιβλιοθηκών για τις κοινές εργασίες και μια απλή διεπαφή (interface) και λογική δομή για την πρόσβαση σε αυτές τις βιβλιοθήκες. Η πρώτη δημόσια έκδοση του CodeIgniter κυκλοφόρησε στις 28/02/2006 (Ιωάννης Σταυγιουνουδάκης (n.d.)). Η τελευταία σταθερή έκδοση 2.1.2 κυκλοφόρησε στις 29/06/2012.

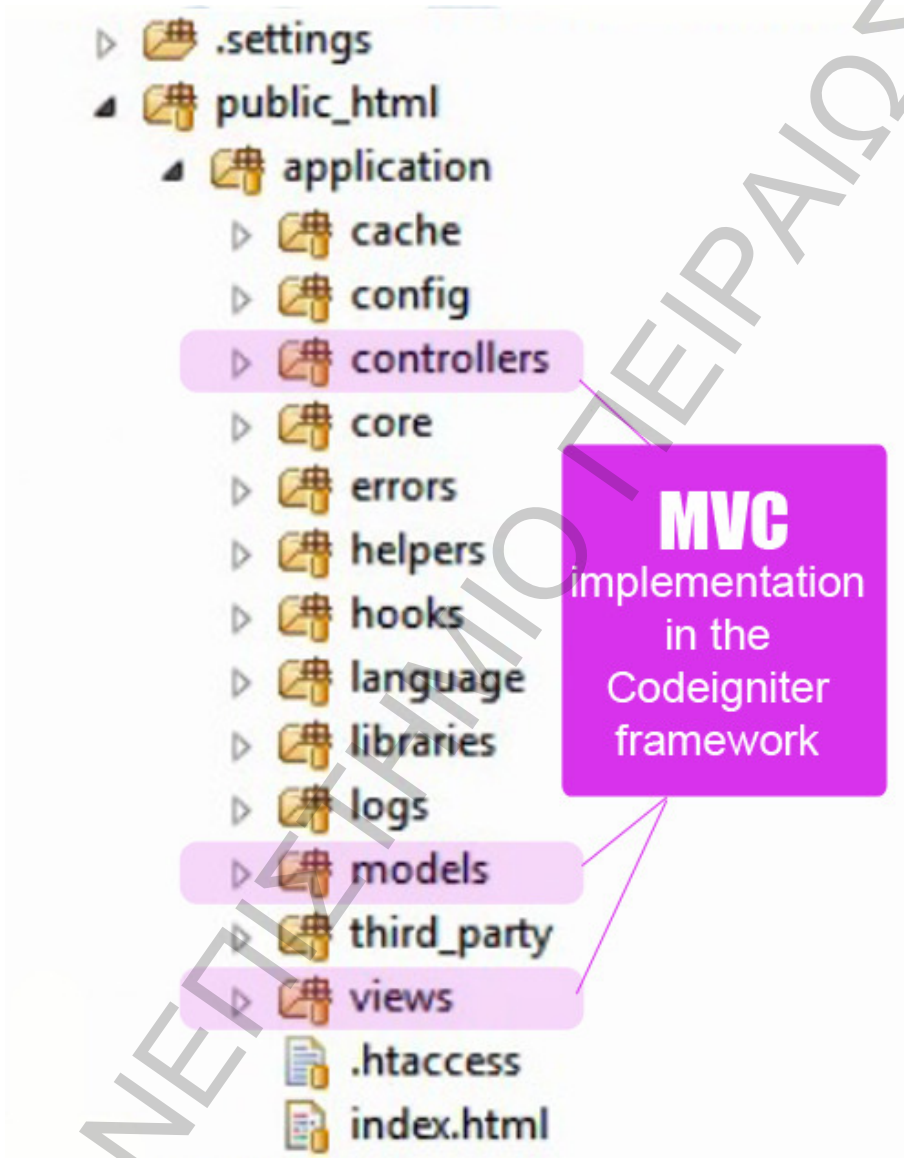
Η ανάπτυξη του CodeIgniter βασίζεται στο πρότυπο «**Μοντέλο-Αποψη-Ελεγκτής**» (**Model-View-Controller, MVC**). Ενώ οι κλάσεις **άποψη** και **ελεγκτής** είναι απαραίτητα μέρος για την ανάπτυξη μιας εφαρμογής χρησιμοποιώντας CodeIgniter, η κλάση **μοντέλο** είναι προαιρετική.

Όπως αναφέρθηκε προηγουμένως το CodeIgniter είναι γνωστό για την ταχύτητά του σε σύγκριση με άλλα πλαίσια PHP. Σε μια γενική ανάλυση των PHP πλαισίων, ο δημιουργός της PHP, Rasmus Lerdorf μίλησε στο Ελεύθερο και Ανοικτού Λογισμικού Συνέδριο (FreeandOpenSourceSoftwareConference, FrOSCon) τον Αύγουστο του 2008, σημειώνοντας ότι του άρεσε το CodeIgniter "γιατί είναι γρηγορότερο, ευκολότερο και λιγότερο σαν ένα πλαίσιο."

Το πλαίσιο παρέχει πλήρη υποστήριξη για την PHP 4 και 5, καθώς και διάφορες βάσεις δεδομένων, πρότυπα (templates), φόρμες (forms) και κρυφή μνήμη (cache), μαζί με τις

Ανάπτυξη εφαρμογής για την επικοινωνία με τη διαδικτυακή υπηρεσία XML-RPC25

βιβλιοθήκες για μια ποικιλία λειτουργιών. Στο αρχείο του CodeIgniter υπάρχει ο φάκελος system που περιέχει τον βασικό κώδικα του CodeIgniter. Ο φάκελος application που περιέχει το application και όλα όσα συνδέονται με αυτό, όπως το μοντέλο, η άποψη και ο ελεγκτής. Στο αρχείο υπάρχουν αρκετοί υποφάκελοι που περιέχουν τις βιβλιοθήκες και άλλα αρχεία (MihaiCurteanu, 2010.)



Σχήμα 2. Δομή αρχείων του CodeIgniter

3.2.1. Χαρακτηριστικά του CodeIgniter

Το CodeIgniter είναι application framework. Είναι ένα εργαλείο για σχεδιαστές ιστοσελίδων (webdevelopers) που δημιουργούν εφαρμογές PHP. Στόχος είναι να δημιουργηθούν προγράμματα πολύ πιο γρήγορα από το να γράφει κανείς κώδικα από την αρχή.

Τα κύρια χαρακτηριστικά του είναι ότι :

Παρέχετε δωρεάν. Έχει χορηγηθεί άδεια ελεύθερης χρήσης λογισμικού ανοιχτού κώδικα από την Apache: εξυπηρετητής ιστού /BSD: λειτουργικό σύστημα

Ανάπτυξη εφαρμογής για την επικοινωνία με τη διαδικτυακή υπηρεσία XML-RPC26

Αληθινά ελαφρύ. Το σύστημα πυρήνων (coresystem) απαιτεί μόνο μερικές μικρές βιβλιοθήκες σε αντίθεση με άλλα πλαίσια εφαρμογής (applicationframeworks) που απαιτούν περισσότερους πόρους. Έτσι το σύστημα βάσεων είναι εύκαμπτο και γρήγορο.

Παράγει καθαρά URL. Τα URL που παράγει το CodeIgniter είναι καθαρά και φιλικά στις μηχανές αναζήτησης.

Προσφέρει μια πλούσια βιβλιοθήκη. Οι αναγκαίοι στόχοι μιας εφαρμογής όπως σύνδεση με μια βάση, αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου, επικύρωση δεδομένων (validation), διαχείριση εικόνων κ.τ.λ. καλύπτονται με μια σειρά από πλούσιες βιβλιοθήκες που υπάρχουν στο «πακέτο» του πλαισίου.

Εύκολη επέκταση. Επεκτείνεται εύκολα μέσω των βιβλιοθηκών, κάποιων helpers και επιπλέον δεν απαιτείται templateEngine (λογισμικό που έχει σχεδιαστεί για να επεξεργάζεται πρότυπα Ιστού και πληροφορίες περιεχομένου για την παραγωγή του web εγγράφων). Δεν είναι υποχρεωτική η χρήση όλων των δυνατοτήτων και όλων των βιβλιοθηκών που παρέχει.

Το CodeIgniter είναι τεκμηριωμένο. Η τεκμηρίωση είναι τόσο σημαντική όσο και ο κώδικας. Το πλαίσιο αποτελείται από έναν καθαρό κώδικα και μια λεπτομερή τεκμηρίωση, άλλο ένα θετικό για το συγκεκριμένο πλαίσιο.

Έχει μια φιλική κοινότητα χρηστών. Το φόρουμ του CodeIgniter μπορεί να φανεί χρήσιμο στους νέους και μη προγραμματιστές λύνοντας απορίες ή μοιράζοντας ιδέες (Ιωάννης Σταυγιουνουδάκης (n.d.)).

3.2.2. Σύγκριση CodeIgniter με άλλες πλατφόρμες

Παρακάτω παρουσιάζεται μια σύγκριση ανάμεσα στα δύο πιο δημοφιλή πλαίσια ανοικτού κώδικα PHP, CodeIgniter και CakePHP και το νεοφερμένο, Kohana.

Χαρακτηριστικά	CodeIgniter 1.6.1	CakePHP	Kohana 2.1.1
Άδεια	Apache/BSD-style		new BSD
PHP Συμβατότητα	4.3.2+ and 5		5.1.3+
Υποστηριζόμενες Βάσεις Δεδομένων	MySQL (4.1+) MySQLi MS SQL PostgreSQL SQLite Oracle ODBC	MySQL SQLite PostgreSQL* * Partial	MySQL PostgreSQL SQLite
Τεκμηρίωση	Εξαιρετική	Καλή	Περιορισμένη
Διαθεσιμότητα tutorials και παραδειγμάτων	Καλή	Καλή	Περιορισμένη
MVC	Αυστηρό	Αυστηρό	Αυστηρό
Modular / HMVC	Τρίτο μέρος βιβλιοθηκών	Τρίτο μέρος βιβλιοθηκών	Ντόπιο (native)
Συμβάσεις	Εύκαμπτες	Αυστηρές	Εύκαμπτες
Διαμόρφωση	PHP αρχεία	PHP αρχεία	PHP αρχεία
Επικύρωση	Καλή	Επικύρωση & ασφάλεια	Καλή

Caching	Περιορισμένο	Πολύ καλό	Μέτριο
Συνεδρίαση	Καλή	Καλή	Εξαιρετική
Logging / Debugging	Καλό	Καλό	Εξαιρετικό
Templating	Native PHP	Θέματα, σχεδιαγράμματα, Απόψεις και Elemente	Native PHP
Βοηθοί	Καλοί	Εξαιρετικοί	Καλοί
JavaScript / AJAX	N/A	N/A	N/A

Πίνακας1. Συγκριτική ανάλυση των παραμέτρων μεταξύ των τριών πλαισίων (MihaiCurteanu, 2010.)

3.3. Μοντέλο-Άποψη-Ελεγκτής (MVC)

Το **CodeIgniter** βασίστηκε όπως αναφέρθηκε παραπάνω σε μια έννοια γνωστή ως «**Μοντέλο-Άποψη-Ελεγκτής**». Αυτή είναι μια πολύ λογική προσέγγιση ως προς τη διαίρεση του κώδικα, χωρίζει τη λογική της εφαρμογής από τη λογική της παρουσίασης.

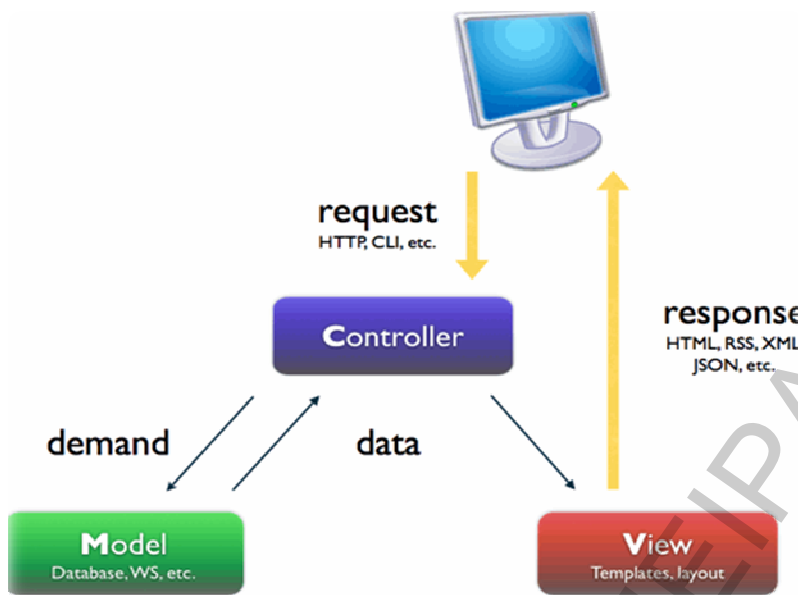
Το πρότυπο MVC διασπά τη λειτουργικότητα σε τρεις διακριτές κατηγορίες :

Μοντέλο: Αντιπροσωπεύει τις δομές δεδομένων. Χαρακτηριστικά οι κλάσεις των μοντέλων περιέχουν τις λειτουργίες που βοηθούν κάποιον να ανακτήσει, να παρεμβάλει και να ενημερώσει τη βάση δεδομένων του.

Άποψη: Είναι οι πληροφορίες που παρουσιάζονται στον χρήστη. Αποτελείται από τα παράθυρα, τα στοιχεία ελέγχου και άλλα στοιχεία, τα οποία ο χρήστης βλέπει και μπορεί να αλληλεπιδράσει μαζί τους. Μια άποψη θα μπορούσε να είναι μια ολοκληρωμένη ιστοσελίδα αλλά στο CodeIgniter μπορεί επίσης να είναι ένα μέρος σελίδων όπως μια επιγραφή, ένα μενού κ.α.

Ελεγκτής: Συνδέει το Μοντέλο και την Άποψη μεταξύ τους, αποτελεί τη λογική με την οποία αποφασίζεται το πώς διαχειρίζεται η εφαρμογή τις εισαγωγές του χρήστη. Είναι υπεύθυνος για την επεξεργασία ενός HTTP για να παραχθεί μια ιστοσελίδα.

Η δομή του συστήματος MVC απεικονίζεται στο Σχήμα 3.



Σχήμα 3. Δομή συστήματος προτύπου MVC.

Ο στόχος του MVC είναι η εξασθένιση της σύζευξης μεταξύ των παραπάνω συστατικών της εφαρμογής. Κάθε αντικείμενο που δημιουργείται θα πρέπει να ανήκει νοητά σε μια από τις τρεις παραπάνω ενότητες, και να περιέχει όσο το δυνατόν λιγότερα στοιχεία από τις άλλες δύο. Για παράδειγμα, ένα αντικείμενο που αντιπροσωπεύει ένα κουμπί δε θα πρέπει να περιλαμβάνει και τον κώδικα επεξεργασίας των δεδομένων, που συμβαίνει όταν πατάμε αυτό το κουμπί. Επίσης ένα αντικείμενο που αντιπροσωπεύει ένα τραπεζικό λογαριασμό δε θα πρέπει να περιλαμβάνει και τον κώδικα σχεδίασης μιας φόρμας που περιέχει τις συναλλαγές του.

Το MVC βοηθά να εξασφαλίσουμε τη μέγιστη δυνατή επαναχρησιμοποίηση του κώδικά. Μια κλάση που υλοποιεί ένα γενικής φύσεως κουμπί μπορεί να χρησιμοποιηθεί σε οποιαδήποτε εφαρμογή. Σε αντίθεση, μια κλάση που υλοποιεί ένα κουμπί το οποίο εκτελεί συγκεκριμένους υπολογισμούς όταν το πατάμε, μπορεί να χρησιμοποιηθεί μόνο στην εφαρμογή για την οποία το δημιουργήσαμε (Αναστάσιος Πιτόπουλος, 2011, Ιωάννης Σταυγιουνουδάκης .(n.d.)).

3.3.1. Παραδείγματα «Μοντέλο-Αποψη-Ελεγκτής»

Παρακάτω είναι ένα παράδειγμα ενός μοντέλου που χρησιμοποιείται για ένα σύστημα blogging.

```
class Blogmodel extends Model {
  var $content = "";
  function Blogmodel(){
    // Call the Model constructor
    parent::Model();
  }
  function insert_entry(){
    $this->content = $_POST['content'];
    $this->db->insert('entries', $this);
  }
  function update_entry(){
    $this->content = $_POST['content'];
```

Ανάπτυξη εφαρμογής για την επικοινωνία με τη διαδικτυακή υπηρεσία XML-RPC29

```

    $this->db->update('entries', $this, array('id' => $_POST['id']));
}
}

```

Στο παρακάτω παράδειγμα η κλάση άποψη περιέχει τον απαραίτητο κώδικα για να φορτωθούν ατομικά δεδομένα όπως το googlemap αλλά και να δημιουργηθούν divs κ.τ.λ. με τη χρήση της HTML.

```

<html>
<head>
<script type="text/javascript"src="http://maps.googleapis.com/maps/api/js?
sensor=false"></script>
<body onload="initialize()">
<div id="map_canvas" class="map_diast"></div>
</body>
</html>

```

Παράδειγμαελεγκτήναφορτώνειμιαάποψη.

```

class Xmlrpc_client1 extends CI_Controller {
public function __construct()
{
    $this->load->view("mapm",$this->data);
}
}

```

Οι υλοποιήσεις του MVC παρατηρούνται σε εφαρμογές web, όπου οι απόψεις είναι σε κώδικα HTML ή XHTML, που παράγεται από την εφαρμογή. Οελεγκτήςσυχνά λαμβάνει εισόδους μέσω GET ή POST και αποφασίζει τι να κάνει, παραδίδοντας τα στοιχεία με βάση το μοντέλο το οποίο περιέχει ειδικούς κανόνες και διαδικασίες για τη διαχείριση και επεξεργασία των δεδομένων (MihaiCurteanu, 2010.).

3.3.2. Πλεονεκτήματα του MVC

Αν και με την πρώτη ματιά η προσέγγιση MVC φαίνεται πολύ επίπονη, δεν είναι απαραίτητα έτσι. Ο

ταν μια τροποποίηση είναι αναγκαία για την εφαρμογή, απαιτείται η κατανόηση της αλλαγής, το οποίο με τη σειρά του σημαίνει ότι ο προγραμματιστής πρέπει να ξέρει ακριβώς ποιο μέρος της εφαρμογής χρειάζεται να τροποποιηθεί. Το κύριο πλεονέκτημα της χρήσης του MVC είναι ότι, εφόσον έχει εφαρμοστεί σωστά, η αλλαγή ενός μέρους του συστήματος δεν έχει επίδραση πάνω σε άλλο μέρος αυτού του συστήματος. Έτσι, για παράδειγμα, μία μεταβολή στην άποψη όπως η προσθήκη μιας επιπλέον βιβλιοθήκης δεν απαιτεί αλλαγή στον ελεγκτή. Έτσι, όλες οι λειτουργίες είναι έγκλειστες και ο κώδικας διατηρείται καθαρός.

Τα πλεονεκτήματα της χρήσης ενός MVC είναι πολλά και πολύ διαφορετικά. Μεταξύ αυτών αναφέρονται : Τα περισσότερα πράγματα υλοποιούνται με λιγότερο κώδικα. Έχει σχεδιαστεί ακριβώς για να αύξηση την παραγωγικότητα και τη συντήρηση των εφαρμογών. Το MVC επιτρέπει την επαναχρησιμοποίηση κώδικα με αποτέλεσμα την αύξηση της παραγωγικότητας. Επιπλέον βοηθά στην τυποποίηση του κώδικα σε ένα έργο όπου εμπλέκονται πολλοί προγραμματιστές, όπως συμβαίνει στις περισσότερες περιπτώσεις, με τη δημιουργία ενός ενιαίου συστήματος που όλοι οι εμπλεκόμενοι πρέπει να προσαρμοστούν. Επίσης διαχωρίζει τον PHP κώδικα από τον HTML, αυτό έχει σαν αποτέλεσμα έναν κώδικα καθαρότερο(MihaiCurteanu, 2010.).

3.4. Υπηρεσίες Ιστού (WebServices)

Τα web services είναι μια τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Ένα web service είναι μια διεπαφή λογισμικού (software interface) που περιγράφει μια συλλογή από λειτουργίες οι οποίες μπορούν να προσεγγιστούν από το δίκτυο μέσω πρότυπων μηνυμάτων XML. Χρησιμοποιεί πρότυπα βασισμένα στη γλώσσα XML για να περιγράψει μία λειτουργία (operation) προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή. Μια ομάδα από web services οι οποίες αλληλεπιδρούν μεταξύ τους καθορίζει μια εφαρμογή web services.

Τα δεδομένα των webservices είναι μορφοποιημένα για μεταφορά χρησιμοποιώντας XML, βελτιώνοντας ή εξαλείφοντας το marshalling (συλλογή στοιχείων και η μετατροπή τους σε μια τυποποιημένη μορφή πριν τη μετάδοση τους μέσω ενός δικτύου), το unmarshalling (το αντίθετο του marshalling) και άλλες απαιτήσεις σχετικά με τη μετάφραση που συνήθως προγραμματίζονταν από τον ίδιο τον προγραμματιστή. Μεταφέρονται χρησιμοποιώντας προτυποποιημένα πρωτόκολλα όπως το HTTP ή το SMTP, τα οποία έχουν δημοσιευμένα καλά καθορισμένα πρότυπα. Επιπλέον η προς έκθεση υπηρεσία τους είναι καλά καθορισμένη χρησιμοποιώντας ένα γνωστό και αποδεκτό μηχανισμό, τη Γλώσσα Περιγραφής των Υπηρεσιών Ιστού (Web Services Description Language, WSDL). Οι υπηρεσίες ανευρίσκονται χρησιμοποιώντας ένα καλά καθορισμένο πρότυπο, την Οικουμενική Περιγραφή Ανακάλυψη και Ενσωμάτωση (Universal Description, Discovery and Integration, UDDI). Το UDDI είναι μια πλατφόρμα για την περιγραφή των υπηρεσιών (Θεοχάρης Δημητρίου, 2007.).

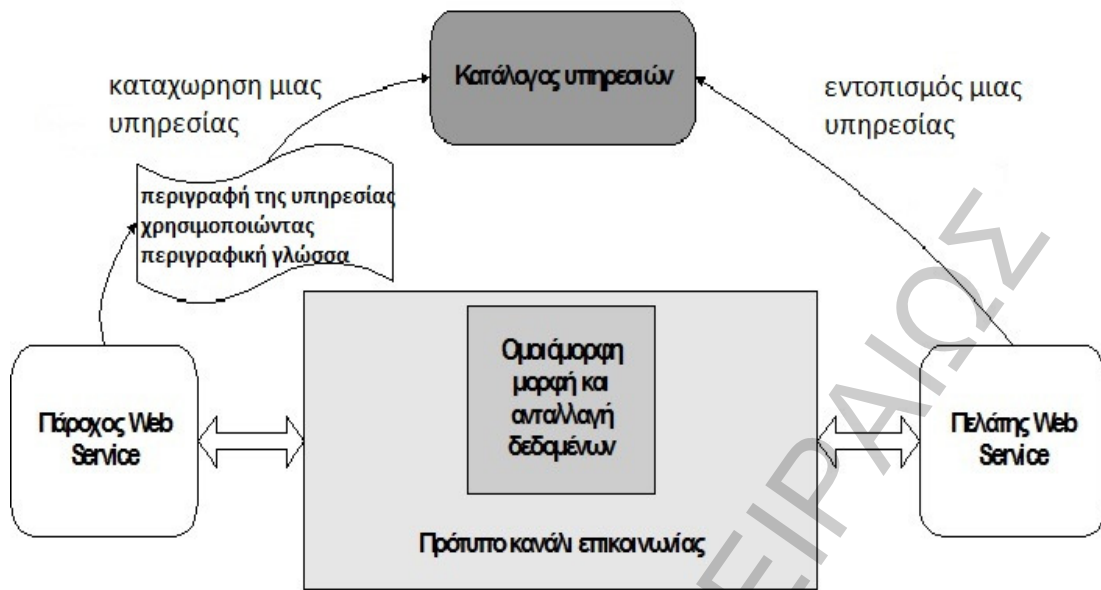
Η Κοινοπραξία του Παγκόσμιου Ιστού (WorldWideWebConsortium, W3C) ορίζει μια διαδικτυακή υπηρεσία (webservice) ως ένα σύστημα λογισμικού σχεδιασμένο να υποστηρίζει μια λειτουργική αλληλεπίδραση μηχανής-προς-μηχανή μέσω ενός δικτύου. Διαθέτει μια διεπαφή που περιγράφεται σε επεξεργάσιμη μηχανογραφικά μορφή (WSDL). Τα συστήματα αλληλεπιδρούν με τη διαδικτυακή υπηρεσία με τρόπο που προβλέπεται από την περιγραφή της, χρησιμοποιώντας μηνύματα Απλού Πρωτοκόλλου Πρόσβασης Αντικειμένου (Simple Object Access Protocol, SOAP), που μεταφέρονται χρησιμοποιώντας HTTP με σύνταξη XML και προτυποποίηση, σε συνδυασμό με πρότυπα που σχετίζονται με το διαδίκτυο.

Το W3C αναφέρει: «Μπορούμε να προσδιορίσουμε δύο σημαντικές κατηγορίες των υπηρεσιών web. Υπηρεσίες web συμβατές με Αντιπροσωπευτική Μεταφορά Κατάστασης (Representational State Transfer, Rest), στις οποίες ο πρωταρχικός σκοπός της υπηρεσίας είναι να χειριστεί αναπαραστάσεις πόρων XML που προέρχονται από το web χρησιμοποιώντας ένα ενιαίο σύνολο λειτουργιών και αυθαίρετες Υπηρεσίες web, όπου η υπηρεσία μπορεί να εκθέσει ένα αυθαίρετο σύνολο πράξεων. »

Οι υπηρεσίες web είναι ένα σύνολο εργαλείων που μπορούν να χρησιμοποιηθούν σε μια σειρά από τρόπους. Οι τρεις πιο συχνές μορφές χρήσης είναι η Κλήση απομακρυσμένης διαδικασίας (Remote Procedure Call, RPC) SOAP και REST (Γεράσιμος Μηνιάτης, 2011.).

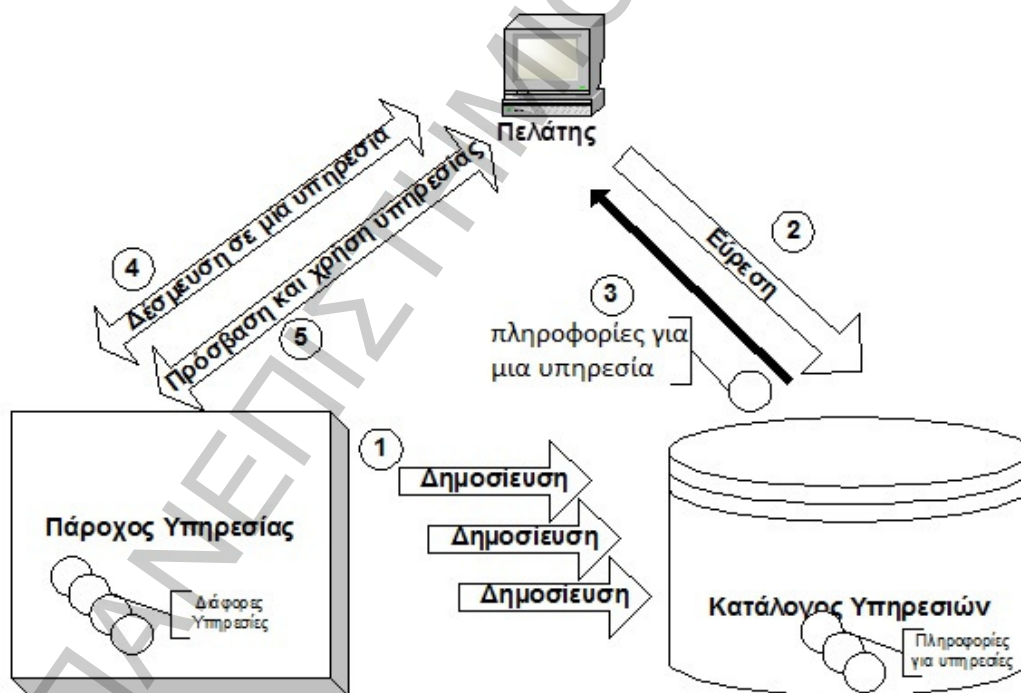
Το Μοντέλο των web services

Το μοντέλο των web services ακολουθεί το παράδειγμα δημοσίευση (publish), εύρεση (find) και σύνδεση (bind). Στο πρώτο βήμα, ο προμηθευτής της υπηρεσίας δημοσιεύει την υπηρεσία σε ένα κατάλογο υπηρεσιών. Στο δεύτερο βήμα, ο πελάτης ο οποίος ψάχνει για μια υπηρεσία που να καλύπτει τις απαιτήσεις του, την αναζητεί στον κατάλογο. Αφού επιτυχημένα βρει πολλαπλές υπηρεσίες επιλέγει μία βάσει των προτιμήσεών του. Τότε μεταφορτώνει την περιγραφή της υπηρεσίας και συνδέεται (δεσμεύεται) με αυτήν ώστε να μπορέσει να καλέσει και να εκτελέσει την υπηρεσία.



Σχήμα 4. Το Μοντέλο των WebServices.

Αρχιτεκτονική των Web Services



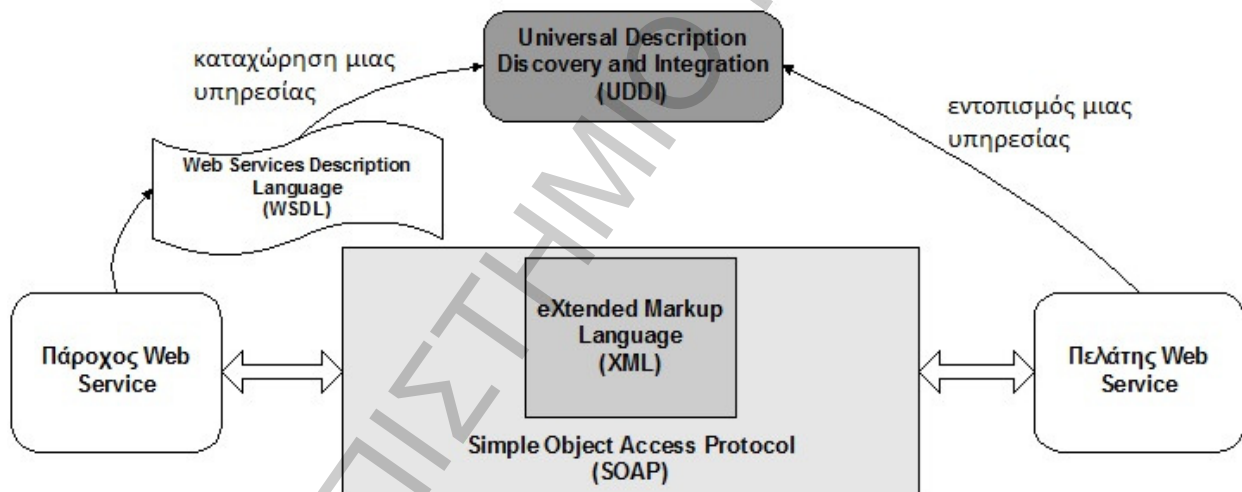
Σχήμα 5. Αρχιτεκτονική των WebServices.

Μια βασική αρχιτεκτονική για webservises πρέπει να παρέχει έναν πρότυπο τρόπο για επικοινωνία και ένα ομοίωμορφο μηχανισμό για περιγραφή και ανταλλαγή των δεδομένων. Μια πρότυπη περιγραφική γλώσσα (metalinguage) για να περιγράψει τις υπηρεσίες. Επίσης πρέπει να περιέχει ένα μηχανισμό για να καταχωρούνται και να εντοπίζονται οι εφαρμογές που βασίζονται σε webservises.

Ανάπτυξη εφαρμογής για την επικοινωνία με τη διαδικτυακή υπηρεσία XML-RPC32

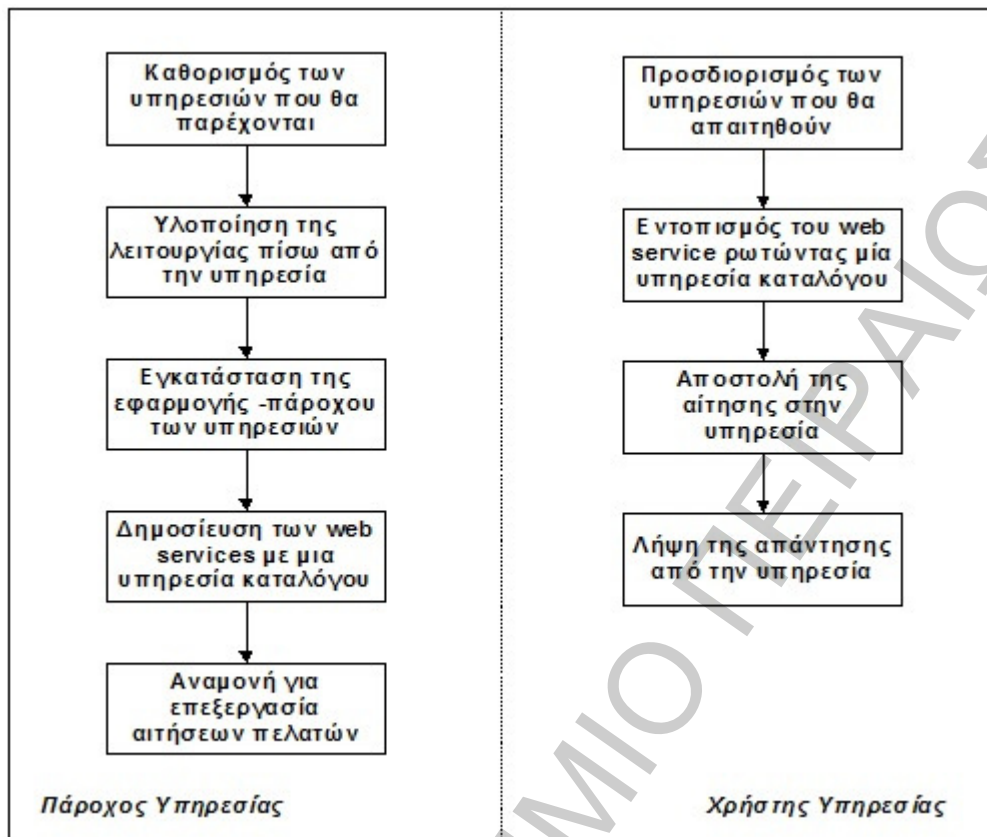
Βασικές Τεχνολογίες των Web Services

Οι βασικές τεχνολογίες που χρησιμοποιούν οι webservices είναι: η **XML** με κύριο χαρακτηριστικό την ομοιόμορφη μορφή και ανταλλαγή δεδομένων. Η Extended Markup Language (XML) είναι μια μέτα-γλώσσα (περιγραφική γλώσσα) η οποία έχει καλά καθορισμένη σύνταξη και σημασιολογία. Τα «αυτοπεριγραφικά» χαρακτηριστικά της XML την κάνουν ένα απλό, αλλά δυνατό, μηχανισμό για τη σύλληψη και την ανταλλαγή των στοιχείων μεταξύ των διαφορετικών εφαρμογών. Η **SOAP** που παρέχει ένα πρότυπο κανάλι επικοινωνίας το οποίο χρησιμοποιείται για επικοινωνία μεταξύ μιας εφαρμογής – προμηθευτή web service και μιας εφαρμογής-πελάτη. Η απλότητα του SOAP είναι το ότι δεν καθορίζει κανένα νέο πρωτόκολλο μεταφοράς. Η πρότυπη περιγραφική γλώσσα για την περιγραφή των παρεχόμενων υπηρεσιών, **WSDL** είναι μια γλώσσα βασισμένη στην XML και χρησιμοποιεί ένα ειδικό σύνολο ετικετών (tags) για να περιγράψει ένα web service. Όταν ένα πρόγραμμα-πελάτη συνδέεται με μια υπηρεσία ιστού μπορεί να διαβάσει το αρχείο WSDL για να προσδιορίσει τις είδους λειτουργίες είναι διαθέσιμες στο διακομιστή. Το πρόγραμμα-πελάτη μπορεί στη συνέχεια να χρησιμοποιήσει το πρωτόκολλο SOAP για να καλέσει μια από τις λειτουργίες που αναφέρονται στο αρχείο WSDL. Επιπλέον χρησιμοποιείται και η τεχνολογία UDDI (Universal Description Discovery and Integration) που είναι ένα μητρώο ανεξάρτητο πλατφόρμας, βασισμένο στην XML, ανεπτυγμένο για να στεγάσει τις επιχειρήσεις από όλο τον κόσμο, σε λίστα στο διαδίκτυο, περιέχοντας επίσης ένα μηχανισμό για την καταγραφή και τον εντοπισμό των εφαρμογών των υπηρεσιών ιστού.



Σχήμα 6. Βασικές τεχνολογίες των webservices.

Βασικά βήματα για την ανάπτυξη εφαρμογών με web services



Πίνακας 2.Βασικά βήματα για την ανάπτυξη εφαρμογών με webservices.

Τα παραπάνω βήματα για την ανάπτυξη εφαρμογών με την υπηρεσία ιστού, θα είναι τα ίδια σε όλες τις εφαρμογές ανεξαρτήτου τεχνολογίας και γλώσσας προγραμματισμού η οποία θα χρησιμοποιηθεί για την υλοποίηση των web services.

Τα οφέλη των WebServices είναι η αλληλεπίδραση μεταξύ υπηρεσιών σε οποιαδήποτε πλατφόρμα και οποιαδήποτε γλώσσα προγραμματισμού, η χαλαρή συνδεσιμότητα μεταξύ εφαρμογών καθώς και η προσαρμογή ήδη υπαρχουσών εφαρμογών στις μεταβαλλόμενες επιχειρησιακές συνθήκες και ανάγκες των πελατών.

Τα WebServices έχουν κάποια πλεονεκτήματα σε σχέση με παλαιότερες κατανεμημένες τεχνολογίες. Πιο συγκεκριμένα, διαθέτουν απλότητα στο πρωτοκόλλο επικοινωνίας και ευκολότερο χειρισμό των δεδομένων. Επιπρόσθετα, διαθέτουν απλότητα στην υποδομή αλλά και ευκολία στην επικοινωνία. Τα WebServices τα χαρακτηρίζουν η διαλειτουργικότητα και η ευκολία ανάπτυξης νέων εφαρμογών (Θεοχάρης Δημητρίου, 2007.).

3.5. XML-RPC

Το **XML-RPC** είναι ένα πρωτόκολλο με μια σειρά από εφαρμογές που επιτρέπουν στο λογισμικό που τρέχει σε αυτό να λειτουργεί σε διαφορετικά περιβάλλοντα να πραγματοποιούνται κλήσεις διαδικασίας μέσω του Διαδικτύου. Είναι μια απομακρυσμένη κλήση διαδικασίας που χρησιμοποιεί την HTTP για τη μεταφορά του αιτήματος και την XML για την κωδικοποίησή του. Η τιμή που επιστρέφεται είναι επίσης σε μορφή XML. Η XML-RPC έχει σχεδιαστεί για να είναι όσο το δυνατόν απλούστερη, ενώ επιτρέπει πολύπλοκες δομές δεδομένων που πρέπει να διαβιβάζονται, επεξεργάζονται και να επιστρέφονται. Οι εφαρμογές XML-RPC είναι συμβατές σε όλα τα λειτουργικά συστήματα, γλώσσες προγραμματισμού,

Ανάπτυξη εφαρμογής για την επικοινωνία με τη διαδικτυακή υπηρεσία XML-RPC 34

δυναμικά και στατικά περιβάλλοντα, open source , για Perl, Python, Java, Frontier, C/C+ +, Lisp, PHP, Microsoft. NET κ.α (ScriptingNewsInc, 2004-2011.).

Πολύ απλά είναι ένας τρόπος ώστε δύο υπολογιστές να επικοινωνούν μέσω του διαδικτύου με τη χρήση XML. Ένας υπολογιστής, τον οποίο θα ονομάσουμε πελάτη, στέλνει μια αίτηση XML-RPC σε έναν άλλο υπολογιστή, τον οποίο θα ονομάσουμε διακομιστή (server) . Μόλις ο διακομιστής λαμβάνει και επεξεργάζεται το αίτημα στέλνει πίσω μια απάντηση στον πελάτη (EllisLabInc, 2006-2012).

Στρατηγικές/Στόχοι

Τείχος Προστασίας: Ο στόχος αυτού του πρωτοκόλλου είναι να θέσει μια συμβατή βάση διαφορετικά περιβάλλοντα, καμία νέαισχύς δεν παρέχεται πέρα από τις δυνατότητες της Κοινής Πύλης Διεπαφής (Common Gateway Interface, CGI).- μια προδιαγραφή για τη μεταφορά πληροφοριών μεταξύ ενός World Wide Web server και ενός προγράμματος CGI. Ένα πρόγραμμα CGI είναι κάθε πρόγραμμα που έχει σχεδιαστεί για να δέχεται και να επιστρέφει τα δεδομένα που συμμορφώνεται με την προδιαγραφή CGI- Το τείχος προστασίας λογισμικού μπορεί να «κοιτάξει» για POST των οποίων ο τύπος κειμένου είναι κείμενο /xml.

Εύρεση: Ήταν επιθυμητή μια καθαρή, επεκτάσιμη μορφή (format) που να είναι πολύ απλή. Θα πρέπει να είναι δυνατόν για ένα κωδικοποιητή (coder) HTML να είναι σε θέση να δει ένα αρχείο που περιέχει μια XML-RPC κλήση διαδικασίας, να κατανοήσει τι κάνει, και να είναι σε θέση να την τροποποιήσει και να δουλέψουμε την πρώτη ή τη δεύτερη προσπάθεια.

Εύκολο στην εφαρμογή: Επιθυμητή ήταν επίσης η εύκολη εφαρμογή του πρωτοκόλλου που θα μπορούσε γρήγορα να προσαρμοστεί για να τρέξει σε άλλα περιβάλλοντα ή σε άλλα λειτουργικά συστήματα (ScriptingNewsInc, 2004-2011).

Η XML-RPC είναι κλάση του CodeIgniter που επιτρέπει την αποστολή αίτησης (request) σε άλλο διακομιστή (server), ή τη δημιουργία XML-RPC διακομιστή που να δέχεται αιτήσεις. Όπως και οι περισσότερες άλλες κλάσεις στο CodeIgniter, οι XML-RPC και XML-RPCS κλάσεις αρχικοποιούνται στον Ελεγκτή χρησιμοποιώντας τη συνάρτηση: `$this->load->library.`

Για να φορτωθεί η XML-RPC κλάση θα χρησιμοποιηθεί:

```
$this->load->library('xmlrpc');
```

3.5.1. Αποστολή Αίτησης XML-RPC

Για να γίνει αποστολή αίτηματος σε ένα διακομιστή XML-RPC θα πρέπει να καθοριστούν η διεύθυνση URL του διακομιστή, η μέθοδος για το διακομιστή που επιθυμεί κανείς να καλέσει και τα δεδομένα αίτησης.

Παρακάτω παρουσιάζεται ένα παράδειγμα αποστολής αίτηματος (request) (τμήμα από τον κώδικα της εφαρμογής).

```
$this->load->library('xmlrpc');
$this->xmlrpc->server('http://webservices.sp-orfeas.com/mobileDeviceSearch/v1_0', 80);
$this->xmlrpc->method('sg_mobile.search');
```

```
$struct = array(
    'language' => 'gr',
    'listingType' => $listingType,
    'category' => 'residential',
    'limit' => '5',
    'pricelow' => 500,
    'pricehigh' => 800,
```

```

        'livingAreaLow'=>50,
    );

    $request = array (
        array( "8c785e5bd1b2ce408bc2b140d", 'string'),
        array( $struct, 'struct')
    );

$this->xmlrpc->request($request);

if ( ! $this->xmlrpc->send_request() )
{
    echo $this->xmlrpc->display_error();
}

```

Ο παραπάνω κώδικας αρχικοποιεί την κλάση XML-RPC, θέτει το URL του server και την επιθυμητή μέθοδο ('sg_mobile.search'). Η αίτηση (στην συγκεκριμένη περίπτωση language, category κ.α.) τοποθετείται σε έναν πίνακα για τη μεταφορά, και τη μεταγλώττιση με βάση τη συνάρτηση της αίτησης. Τέλος, αποστέλλεται η πλήρης αίτηση. Εάν η send_request() μέθοδος επιστρέψει λάθος θα εμφανιστεί το μήνυμα σφάλματος που στέλνεται πίσω από το XML-RPC Server.

Ανατομία Αίτησης

Μια XML-RPC αίτηση είναι απλά δεδομένα που επιθυμεί κανείς να στείλει στο XML-RPC server. Κάθε κομμάτι των δεδομένων σε μία αίτηση αναφέρεται ως μια παράμετρος του αιτήματος. Το παραπάνω παράδειγμα έχει πολλές παραμέτρους όπως το language, το category κ.α. Όταν ο XML-RPC server λαμβάνει το αίτημα, θα ψάξει για αυτές τις παραμέτρους.

Οι παράμετροι της αίτησης πρέπει να τοποθετηθούν σε έναν πίνακα για τη μεταφορά. Κάθε παράμετρος μπορεί να είναι ένας από τους πεπτά τύπους δεδομένων (strings, αριθμός, ημερομηνία, κ.λπ.). Εάν οι παράμετροι είναι κάτι άλλο από strings θα πρέπει να συμπεριληφθούν στον τύπο δεδομένων του πίνακα του αιτήματος.

Ακολουθεί ένα παράδειγμα ενός πίνακα με τρεις παραμέτρους:

```

$request = array('language', 'category', 'www.some-site.com');
$this->xmlrpc->request($request);

```

Εάν χρησιμοποιηθούν τύποι δεδομένων, εκτός από strings, ή αν υπάρχουν πολλά διαφορετικά είδη δεδομένων, θα πρέπει τοποθετηθεί κάθε παράμετρος στο δικό της πίνακα, με το είδος των δεδομένων στη δεύτερη θέση:

```

$request = array (
    array('language', 'string'),
    array('category', 'string'),
    array(FALSE, 'boolean'),
    array(12345, 'int')
);
$this->xmlrpc->request($request);

```

3.5.2. Διαμόρφωση απάντησης

Παρόμοια με τις αιτήσεις, οι απαντήσεις πρέπει να διαμορφωθούν ως πίνακας. Ωστόσο, σε αντίθεση με τις αιτήσεις, μια απάντηση είναι ένας πίνακας που περιέχει ένα μόνο στοιχείο. Αυτό το

στοιχείο μπορεί να είναι ένας πίνακας με πολλούς πρόσθετους πίνακες, αλλά μπορεί να υπάρχει μόνο ένας κύριος index πίνακας. Με άλλα λόγια, το βασικό πρότυπο είναι το εξής:

```
$response = array('ResponseData', 'array');
```

Η απάντηση, ωστόσο, περιέχει συνήθως πολλαπλά κομμάτια πληροφορίας. Προκειμένου να επιτευχθεί αυτό, πρέπει να τοποθετηθεί η απάντηση στον δικό της πίνακα, έτσι ώστε ο κύριος πίνακας να εξακολουθεί να περιέχει ένα κομμάτι δεδομένων. Ακολουθεί ένα παράδειγμα που δείχνει πώς αυτό μπορεί να επιτευχθεί:

```
$response = array (
    array(
        'param1' => array('language', 'string'),
        'param2' => array('category', 'string'),
        'param3' => array(123435, 'int'),
    ),
    'struct'
);
```

Ο παραπάνω πίνακας έχει διαμορφωθεί ως ένα struct. Αυτός είναι ο πιο κοινός τύπος δεδομένων για απαντήσεις.

Όπως και με αιτήσεις, η απάντηση μπορεί να είναι ένα από τους επτά τύπους δεδομένων που προαναφέρθηκαν (EllisLabInc, 2006-2012).

3.6. GOOGLE MAP API

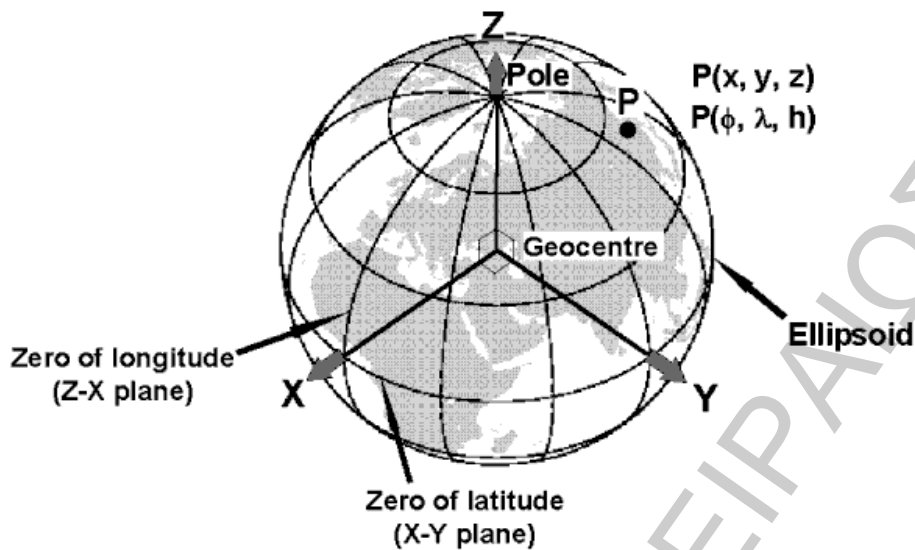
Το Google Map API είναι μια διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface – API) της Google το οποίο δίνει τη δυνατότητα ενσωμάτωσης ενός δυναμικού χάρτη σε ιστοσελίδες με τη χρήση Javascript. Το API παρέχει έναν αριθμό εργαλείων για το χειρισμό χαρτών και την προσθήκη περιεχομένου σε αυτούς μέσω διαφόρων υπηρεσιών, επιτρέποντας έτσι τη δημιουργία εύρωστων εφαρμογών που εκμεταλλεύονται γεωγραφικές πληροφορίες. Η επίσημη ιστοσελίδα του Google Map είναι: <http://maps.google.com>

Είναι μια δωρεάν υπηρεσία, διαθέσιμη για οποιοδήποτε ιστότοπο που είναι ελεύθερος για το κοινό. Το μόνο που χρειάζεται είναι ένα κλειδί API, (API key) που μπορεί να ζητηθεί εφόσον υπάρχει λογαριασμός στη Google και αυτομάτως συνδέεται με το λογαριασμό.

Βασικό στοιχείο οποιασδήποτε Google Map API εφαρμογής είναι ο ίδιος ο χάρτης, ο οποίος φορτώνεται και αρχικοποιείται με συγκεκριμένες διαστάσεις (ύψος και πλάτος), κέντρο (γεωγραφικό πλάτος, γεωγραφικό μήκος), επίπεδο εστίασης και τύπο (κανονικό, δορυφορικό, υβριδικό) (Δήμητρα Μανδουράρη (n.d.)).

3.6.1. Παγκόσμιο Γεωδαιτικό Σύστημα Αναφοράς 1984 -WGS 84

Το WGS84 είναι ένα συμβατικό γήινο σύστημα αναφοράς. Ο ορισμός WGS84 υπακούει στα κριτήρια που έχουν ορισθεί από τη Διεθνή Υπηρεσία Περιστροφής της Γης (International Earth Rotation Service, IERS).



Σχήμα 7. Ορισμός συστήματος συντεταγμένων WGS84.

Το WGS84 ως αρχή ορίζει το κέντρο μάζας της γης. Ο άξονας Z έχει τη διεύθυνση του IES(Ινστιτούτο για το Περιβάλλον και την Αειφορία) Πόλου Αναφοράς. Αυτή αναφέρεται στη διεύθυνση του Συμβατικού Γήινου Πόλου την εποχή 1984,0 με μια ακρίβεια 0,005". Ο άξονας X ορίζεται από την τομή IERS του μεσημβρινού αναφοράς και του επιπέδου που περνά από την αρχή των αξόνων, κάθετα στον άξονα Z. Ο μεσημβρινός αναφοράς συμπίπτει με το μηδενικό μεσημβρινό την εποχή 1984,0 και με ακρίβεια 0,005". Ο άξονας Y συμπληρώνει το δεξιόστροφο, Γη-Κεντραρισμένη-Γη-Σταθερή (Earth-Centered- Earth-Fixed, ECEF) ορθογώνιο σύστημα συντεταγμένων. (NIMA WGS84 Update Committee, 1997).

Το ελλειψοειδές αναφοράς WGS84 ορίζεται από τις ακόλουθες παραμέτρους:

Μεγάλος ημιάξονας $a = 6378137 \text{ m} \pm 2 \text{ m}$.

Παγκόσμια Βαρυτιμετρική σταθερά $GM = 3986005 \cdot 10^8 \text{ m}^3/\text{s}^2$.

Γωνιακή ταχύτητα της Γης $\omega = 72921158 \cdot 10^{-11} \text{ rad/s}$.

Κανονικοποιημένος συντελεστής του δυναμικού έλξης της γης:

$C2,0 = -484.16685 \cdot 10^{-6}$. (Δεληκαράγλου, 2005)

Το ελλειψοειδές WGS84 είναι πρακτικά ίδιο με το GRS-80 (Geodetic Reference System, 1980).

Για την αρχική υλοποίηση WGS84 χρησιμοποιήθηκε ένα πλήθος από σταθμούς Doppler.

3.6.2. Στοιχεία Απεικόνισης GoogleMapsAPI

Τα δεδομένα που διατίθενται χρησιμοποιούν το Παγκόσμιο Γεωδαιτικό Σύστημα Αναφοράς WGS84 με απεικόνιση σε ορθή Μερκατορική (χαρτογραφική) προβολή. Σημειώνουμε ότι το σύστημα αυτό χρησιμοποιεί εγγενώς το σύστημα GPS.

Οι εικόνες μπορεί να προέρχονται από δορυφορικές λήψεις ή αεροφωτογραφίες. Η διακριτική τους ικανότητα ποικίλει από περιοχή σε περιοχή με μέγεθος εικονοστοιχείου από δεκάδες μέτρα μέχρι ένα μέτρο περίπου για τις δορυφορικές λήψεις και μέχρι λίγα εκατοστά για τις αεροφωτογραφίες. Η λήψη ανάλογα με την περιοχή μπορεί να είναι χρονικά ενήμερη, συνήθως όμως οι εικόνες είναι ως τέσσερα χρόνια παλαιότερες. Η Ελλάδα απεικονίζεται μόνο σε δορυφορικές λήψεις, οι οποίες οριοθετούνται χρονικά μεταξύ των ετών 2002-2008. Ως παράδειγμα δίνεται η ανάλυση εικόνων των ακόλουθων περιοχών:

Η.Π.Α. : 1 m (εκτός από τις Πολιτείες της Αλάσκας και της Χαβάης που έχουν χαμηλότερη ανάλυση).

Τσεχία, Σλοβακία : 0,2 m

Ουγγαρία, Βαλκάνια : 2,5 m (μέση ανάλυση)

Παγκόσμια : γενικά 15 m (μερικές περιοχές όπως η Ανταρκτική έχουν χαμηλότερη ανάλυση), αλλά εξαρτώνται από την ποιότητα της δορυφορικής εικόνας/ αεροφωτογραφίας.

Οι ημερομηνίες των εικόνων ποικίλουν. Οι περισσότερες εικόνες αστικών περιοχών χρονολογούνται από το 2004 και μετά, ενώ η πλειοψηφία των εικόνων των μεγαλύτερων αστικών περιοχών τις Ελλάδας έχουν ενημερωθεί πρόσφατα.

3.6.3. Ανάλυση GoogleMapAPI

Η ανάλυση των δεδομένων που διαθέτει η Google γίνεται μέσω της διασύνδεσης GoogleMapsApplicationProgramInterface (API). Πρόκειται για ένα σύνολο κλάσεων με ιδιότητες (properties), συμβάντα (events) και μεθόδους (methods) δομημένα με τρόπο που κάνουν εφικτή την πρόσβαση στα δεδομένα και να επιτρέπουν την ανάκτησή τους, καθώς και τον εμπλουτισμό τους πριν την απεικόνιση (Νικόλαος Ευστάθιος, 2009.).

Παρακάτω παρουσιάζονται τα χαρακτηριστικά των κυριότερων κλάσεων που χρησιμοποιήθηκαν στην πτυχιακή και παρέχονται από τη διεπαφή Google Maps API:

Γεγονότα (Events)

Το Google Map API περιέχει δηλώσεις γεγονότων (events) τα οποία εμφανίζονται μέσω των φυλλομετρητών ιστοσελίδων (web browsers) πάνω σε αντικείμενα του Google Map. Ας σημειωθεί ωστόσο ότι τα γεγονότα αυτά είναι διαφορετικά από τα πρότυπα γεγονότα πάνω στο Μοντέλο Αντικειμενικού Εγγράφου (Document Object Model, DOM), το οποίο είναι μια προγραμματιστική διεπαφή που επιτρέπει σε προγράμματα και εκτελέσιμα σενάρια δράσης (scripts) την ενημέρωση του περιεχομένου και του στυλ όλου του εγγράφου με δυναμικό τρόπο. Έτσι το Google Map API δημιουργεί και διαχειρίζεται δικά του γεγονότα ενώ το Μοντέλο Αντικειμένου Εγγράφου δημιουργεί άλλα γεγονότα και τα επεξεργάζεται σύμφωνα με το μοντέλο αντικειμένου που χρησιμοποιεί ο συγκεκριμένος φυλλομετρητής. Επιπλέον, το Google Map API έχει το πλεονέκτημα ότι προσφέρει μηχανισμούς που ακούνε (listen) και απαντάνε (respond) στα γεγονότα DOM ανεξάρτητα από τις ιδιαιτερότητες του κάθε φυλλομετρητή. Παράδειγμα γεγονότων του Google Map API είναι το κλικ του ποντικιού, το διπλό κλικ, η κίνηση του ποντικιού τα οποία σηματοδοτούν την έναρξη κάποιων διαδικασιών που περιγράφονται στον κώδικα Javascript. Εξάλλου η Javascript είναι μια γλώσσα σεναρίων (scripting) που «καθοδηγεί γεγονότα» («event driven») εξελισσόμενα στους φυλλομετρητές.

Στην εργασία αυτή έγινε χρήση των Γεγονότων (Events). Πιο συγκεκριμένα στην διαδικασία φόρτωσης του χάρτη της Google γράφτηκε μια JavaScript συνάρτηση για τη δημιουργία ενός αντικειμένου "χάρτη". Για την αρχικοποίηση του αντικειμένου χρησιμοποιήθηκε το συμβάν φόρτωσης (onload event) της ετικέτας σώματος. Επιπλέον χρησιμοποιήθηκε και ο μηχανισμός που προσφέρουν οι χάρτες της Google το addListener για να μπορεί ο χρήστης κάνοντας κλικ στους δείκτες (markers) που υπάρχουν στο χάρτη να ανοίγει ένα παράθυρο πληροφοριών.

Επικαλύψεις (Overlays)

Τα overlays είναι αντικείμενα πάνω στο χάρτη που αντιστοιχούν σε συγκεκριμένες συντεταγμένες και γι αυτό μετακινούνται ανάλογα όταν μετακινείται ή αλλάζει εστίαση ο χάρτης. Χρησιμοποιούνται για το σχεδιασμό και την επίδειξη σημείων, γραμμών ή ολόκληρων περιοχών. Το GoogleMapAPI διαθέτει πολλά είδη overlays όπως τα σημεία, τις γραμμές, τις περιοχές, τον χάρτη και το παράθυρο πληροφοριών (infowindow). Παρακάτω παρουσιάζονται οι επικαλύψεις που χρησιμοποιήθηκαν στην εργασία.

Σημεία

Αναπαριστώνται με τη χρήση δεικτών (markers) πάνω στο χάρτη και συχνά προβάλλουν μια σχετική εικόνα στο σημείο που επιδεικνύουν. Τα markers είναι σχεδιασμένα να είναι διαδραστικά, δηλαδή επάνω τους πραγματοποιούνται γεγονότα, όπως ένα κλικ ή δεξί κλικ, και συχνά προβάλλεται παράλληλα με αυτά ένα παράθυρο πληροφοριών για το σημείο. Μια επιπλέον ιδιότητά τους είναι ότι μπορεί να είναι συρόμενα (draggable) από ένα σημείο σε άλλο στο χάρτη. Επίσης είναι δυνατή η επεξεργασία και η αλλαγή της εικόνας (icon) που αντιστοιχεί σε κάθε δείκτη. Η εικόνα αποτελείται από μια εικόνα στο παρασκήνιο και μια άλλη ως σκιά τοποθετημένες σε συγκεκριμένες θέσεις μεταξύ τους. Παρότι οι δείκτες έχουν μεγάλη χρησιμότητα, η προσθήκη πολλών από αυτούς σε έναν χάρτη Google μπορεί να έχει αρνητική επίδραση στην ταχύτητα απεικόνισης του χάρτη (maprendering) ειδικά σε συγκεκριμένα επίπεδα εστίασης. Λύση σε αυτό δίνει ο Διαχειριστής Δεικτών (MarkerManager), ο οποίος έχει την ικανότητα να παρακολουθεί ποιοι δείκτες είναι ορατοί σε ποια επίπεδα εστίασης και να στέλνει μόνο τους απαραίτητους στο χάρτη για σχεδιαστικούς σκοπούς. Δηλαδή προσθέτει ή αφαιρεί δυναμικά δείκτες από το χάρτη με αποτέλεσμα την αύξηση της ταχύτητας απεικόνισης του χάρτη και της μείωσης των άχρηστων οπτικών στοιχείων.

Στην εφαρμογή υπάρχουν πάνω στο χάρτη δείκτες σε διάφορες περιοχές της χώρας. Η χρήση δεικτών είναι ένα κύριο κομμάτι της εργασίας καθώς οι απαραίτητες συντεταγμένες που χρειάζεται να οριστούν για να εμφανισθούν στο χάρτη, φορτώνονται μέσω μιας διαδικτυακής υπηρεσίας (webservice). Οι δείκτες είναι διαδραστικοί δηλαδή μπορεί ο χρήστης να κάνει κλικ πάνω τους και να ανοίξει ένα παράθυρο πληροφοριών.

Το παράθυρο πληροφοριών (infowindow)

Αποτελεί ένα ειδικό overlay. Προστίθεται στο χάρτη αυτόματα και σε κάθε χάρτη αντιστοιχεί ένα μόνο αντικείμενο παραθύρου πληροφοριών. Το παράθυρο πληροφοριών που ανοίγεται με ένα κλικ πάνω στους markers, εμφανίζει πληροφορίες για ακίνητα που είναι προς πώληση ή προς ενοικίαση. Τα δεδομένα αυτά φορτώνονται από τη βάση μιας κτηματομεσιτικής πύλης (Δήμητρα Μανδουράρη (n.d.)).

Σύμπλεγμα δεικτών (markercluster)

Οι χάρτες της Google διαθέτουν την βιβλιοθήκη συμπλέγματος δεικτών (markercluster) που δημιουργεί και διαχειρίζεται ανά επίπεδο zoom μεγάλες ποσότητες δεικτών, με αποτέλεσμα να παρέχει έναν πιο αποδοτικό χάρτη. Στην εργασία έγινε χρήση του εργαλείου αυτού ώστε να είναι ευδιάκριτη η ποσότητα των δεικτών ανά περιοχή (GoogleProjectHosting, (n.d.)).

3.6.4. Παράδειγμα

Ακολουθεί παράδειγμα για την κατανόηση της λειτουργίας του GoogleMapAPI, Η ακόλουθη ιστοσελίδα εμφανίζει ένα χάρτη με επίκεντρο την Αθήνα, Ελλάδα:



Σχήμα 8. Googlemap, χάρτης με επίκεντρο την Αθήνα, Ελλάδα

```

<!DOCTYPE html>
<html>
<head>

<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?sensor=false">
</script>

<script type="text/javascript">

function initialize() {

var athina = new google.maps.LatLng(37.98994,23.729038);
var mapOptions = {
center: athina,
zoom: 10,
mapTypeId: google.maps.MapTypeId.ROADMAP
};

var map = new
google.maps.Map(document.getElementById("map_canvas"),
mapOptions);
}

</script>

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

```

Ανάπτυξη εφαρμογής για την επικοινωνία με τη διαδικτυακή υπηρεσία XML-RPC41

```
<div id="map_canvas" class="map_diast"></div>
</body>
</html>
```

Αρχικά, γίνεται δήλωση της εφαρμογής ως HTML5 χρησιμοποιώντας την `<!DOCTYPE html>` δήλωση. Στη συνέχεια ακολουθεί η προσθήκη του Maps API JavaScript χρησιμοποιώντας την ετικέτα `script`. Επιπλέον δημιουργείται ένα `div` στοιχείο που ονομάζεται "map_canvas" για να «κρατήσει» το χάρτη καθώς και ένα αντικείμενο JavaScript για να «κρατήσει» τις ιδιότητες του χάρτη. Γράφεται μια συνάρτηση JavaScript για τη δημιουργία ενός αντικειμένου "χάρτη" και αρχικοποιείται το αντικείμενο "χάρτης" από το συμβάν φόρτωσης (onload) της ετικέτας σώματος. Το χαρακτηριστικό Onload της ετικέτας σώματος είναι ένα παράδειγμα χειρισμού συμβάντων (GoogleDevelopers, 2013.).

3.7. JQUERY

Η jQuery είναι μια JavaScript βιβλιοθήκη συμβατή με όλους του περιηγητές, που δίνει έμφαση στην αλληλεπίδραση μεταξύ JavaScript και HTML. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον John Resig και πλέον χρησιμοποιείται σε περισσότερο από το 27% των 10.000 δημοφιλέστερων διαδικτυακών τόπων, καθιστώντας την ως την πιο δημοφιλή βιβλιοθήκη JavaScript σήμερα.

Η βιβλιοθήκη jQuery είναι δωρεάν, ανοιχτού κώδικα λογισμικό υπό διπλή άδεια χρήσης MIT License και GNU General Public License, Version 2. Η σύνταξή της έχει ως στόχο την εύκολη πλοήγηση κόμβων ενός εγγράφου, επιλέγοντας στοιχεία του Document Object Model (DOM), δημιουργώντας animations, διαχειρίζοντας γεγονότα (events), και αναπτύσσοντας διαδικτυακές εφαρμογές AJAX. Η jQuery παρέχει επίσης τη δυνατότητα στους προγραμματιστές να δημιουργήσουν plugins στην ίδια τη βιβλιοθήκη, με τέτοιο τρόπο ώστε να την επεκτείνουν. Με την παροχή αυτής της δυνατότητας, οι προγραμματιστές μπορούν να δημιουργήσουν χαμηλού επιπέδου αλληλεπιδράσεις και animations, προχωρημένα εφέ και υψηλού επιπέδου widgets. Το παραπάνω συμβάλλει δραστικά στη δημιουργία ισχυρών και δυναμικών ιστοσελίδων στα πρότυπα του Web 2.0.

Για να ενσωματώσει κανείς τη βιβλιοθήκη αυτή στην ιστοσελίδα του, αρκεί μόνο να κατεβάσει την βιβλιοθήκη- η λήψη του jQuery πραγματοποιείται από την ιστοσελίδα www.jquery.com- και να συνδέσει το κεντρικό script με το έγγραφο HTML, γράφοντας την παρακάτω γραμμή στον κώδικα του εγγράφου HTML:

```
<script type="text/javascript" src="jQuery.js"></script>
```

Το παραπάνω μπορεί επίσης να ενσωματωθεί και μέσω του GoogleAJAXLibrariesAPI:

```
<script type="text/javascript" src="http://www.google.com/jsapi"></script>
```

```
<script>google.load("jquery", "1.3.2");</script>
```

Μόλις η βιβλιοθήκη jQuery ενσωματωθεί στο έγγραφο HTML, μπορούμε να χρησιμοποιήσουμε τις λειτουργίες της. Η jQuery έχει δύο είδη βασικών αλληλεπιδράσεων. Το πρώτο είδος είναι μέσω της ειδικής συνάρτησης `$`, η οποία είναι ο κατασκευαστής ενός αντικειμένου τύπου jQuery. Αυτές οι συναρτήσεις, συχνά αποκαλούνται "εντολές", και είναι chainable – ελιτρεύουν πάντα αντικείμενο jQuery. Το δεύτερο είδος είναι μέσω των συναρτήσεων του στυλ `$`-πρόθεμα. Αυτές οι συναρτήσεις αποκαλούνται βοηθητικές λειτουργίες και δε λειτουργούν με τοκαθεαυτάντικείμενο jQuery.

Μια τυπική ροή εργασίας για επεξεργασία πολλαπλών κόμβων του DOM, αρχίζει με τη συνάρτηση `$` η οποία καλείται με κάποιον CSS selector (πρότυπα που χρησιμοποιούνται για να επιλέξουν το στοιχείο που επιθυμούν να μορφοποιήσουν), που έχει ως αποτέλεσμα ένα jQuery αντικείμενο το οποίο παραπέμπει σε κανένα ή και πολλά αντικείμενα DOM του HTML εγγράφου. Αυτά τα αντικείμενα ή κόμβοι, μπορούν να επεξεργαστούν και να παραποιηθούν, μέσω των συναρτήσεων και των μεθόδων της βιβλιοθήκης (Ανάργυρος Ακριβός, 2010.).

Βασική σύνταξη : `$(selector).action()`

Ένα \$ υπογράφει για τον καθορισμό / πρόσβαση σε jQuery

Ένα (selector) για να θέτει "ερώτημα σε (ή να βρίσκει)" στοιχεία HTML

Ένα jQuery action() για να εκτελούνται τα στοιχεία

Παραδείγματα:

`$(this).hide()`-κρύβει το τρέχον στοιχείο.

`$("p").hide()`-κρύβει όλα τα στοιχεία <p>.

`$(".test").hide()`-κρύβει όλα τα στοιχεία με class = "test".

`$("#test").hide()`- κρύβει το στοιχείο με id = "test".

(RefsnesData, 1999-2013.)

Στο ακόλουθο παράδειγμα γίνεται η αρχικοποίηση του χάρτη και δημιουργείται ένα αντικείμενο με επιλογές του χάρτη (Map options object) που περιέχει μεταβλητές αρχικοποίησης του.

```
function initialize(){
var athina= new google.maps.LatLng(37.98994,23.729038)
var mapOptions={
center: athina,
zoom:10,
mapTypeId:google.maps.MapTypeId.ROADMAP
};
var map=new google.maps.Map(document.getElementById("map_canvas"),
mapOptions);
}
```

Τα πλεονεκτήματα του jquery είναι πρώτον ότι προσπαθεί να απλουστεύσει τον προγραμματισμό σε javascript, προσφέροντας απλούς μηχανισμούς και εντολές μέσω του framework της. Δεύτερον περιέχει πλήρη και αναλυτική τεκμηρίωση, μάλιστα υποστηρίζεται και από μια ενεργή κοινότητα. Επιπλέον το βασικό πακέτο της JQUERY έχει μικρό μέγεθος, μόλις 20kb. Η λογική με την οποία είναι φτιαγμένη είναι απλή, γεγονός που καθιστά εύκολη την επέκτασή/τροποποίησή της (iDesiner, 2013.).

3.8. HTML

Η HTML εφευρέθηκε το 1990 από τον Tim Berners-Lee. Σκοπός ήταν να καταστεί ευκολότερη για τους επιστήμονες σε διαφορετικά πανεπιστήμια να αποκτήσουν πρόσβαση ο ένας στα έγγραφα της έρευνας του άλλου. Το έργο έγινε μεγαλύτερη επιτυχία από ό, τι ο Tim Berners-Lee είχε ποτέ φανταστεί. Με την επινόηση της HTML τέθηκαν τα θεμέλια για το διαδίκτυο όπως το γνωρίζουμε σήμερα (HTML.net.(Anon, n.d)). Η HTML είναι μια συντομογραφία του "HyperText Mark-up Language".

Hyper είναι η μέθοδος με την οποία μπορεί κάποιος να κινηθεί στο διαδίκτυο, κάνοντας κλικ στο ειδικό κείμενο που ονομάζεται υπερ-συνδέσμος που θα φέρει την επόμενη σελίδα. Το γεγονός ότι είναι *υπερ* σημαίνει απλά ότι δεν είναι γραμμική - δηλαδή, υπάρχει δυνατότητα να πλοηγηθείς από σημείο σε σημείο στο διαδίκτυο απλά κάνοντας κλικ σε συνδέσμους.

Text είναι αγγλικοί χαρακτήρες που χρησιμοποιούνται για να δημιουργήσουν λέξεις.

Markup είναι ό, τι κάνουν οι ετικέτες HTML μέσα στο κείμενο για να αλλάξει η εμφάνισή του. Μπορεί να επισημανθεί ως ένας ορισμένος τύπος κειμένου, για παράδειγμα κείμενο με πλάγιους χαρακτήρες. Είναι παρόμοια διαδικασία με τα προγράμματα επεξεργασίας κειμένου.

Language, η HTML είναι μια γλώσσα, καθώς έχει λέξεις και σύνταξη, όπως οποιαδήποτε άλλη γλώσσα. Χρησιμοποιεί πολλές αγγλικές λέξεις.

Η HTML αποτελείται από μια σειρά σύντομων κωδικών που πληκτρολογούνται σε ένα αρχείο κειμένου από τον προγραμματιστή, αυτές είναι οι ετικέτες. Το κείμενο στη συνέχεια αποθηκεύεται ως αρχείο HTML, και προβάλλεται μέσω ενός προγράμματος περιήγησης, όπως ο Internet Explorer ή ο FireFox. Το πρόγραμμα περιήγησης διαβάζει το αρχείο και μετατρέπει το κείμενο σε μια ορατή μορφή (RossShannon, 2000-2013.). Για τη δημιουργία σελίδων σε HTML συχνά χρησιμοποιείται ένα απλός επεξεργαστής κειμένου, το σημειωματάριο, το οποίο βρίσκεται συνήθως στο μενού έναρξης σύμφωνα με τα προγράμματα στα βοηθήματα. Αλλά χρησιμοποιούνται και επεξεργαστές γραφικών όπως το Dreamweaver.

Όταν γράφει κάποιος σε γλώσσα HTML, προσθέτει "ετικέτες" στο κείμενο, προκειμένου να δημιουργηθεί η δομή. Οι ετικέτες (Tags) δίνουν δομή σε ένα HTML έγγραφο και ενημερώνουν το φυλλομετρητή (browser) για τον τρόπο εμφάνισης της ιστοσελίδας (JillJeffersGoodell, 1999-2004.). Γενικά, τα στοιχεία αποτελούνται από μια ετικέτα αρχής, κάποιο περιεχόμενο, καθώς και μια ετικέτα τέλους. Όλα τα Tags έχουν την ίδια μορφή: ξεκινούν με το σύμβολο λιγότερο "<" και τέλος με το σύμβολο μεγαλύτερο ">". Σε γενικές γραμμές, υπάρχουν δύο είδη ετικετών, το άνοιγμα tags: <html> και το κλείσιμο: </html>. Η μόνη διαφορά ανάμεσα σε ένα tag ανοίγματος και ένα tag κλεισίματος είναι η κάθετος προς τα εμπρός "/". Μπορεί να υπάρχει περιεχόμενο, ανάμεσα σε ένα tag ανοίγματος και ένα tag κλεισίματος.

Η HTML είναι μια γλώσσα, η οποία καθιστά δυνατή την παρουσίαση πληροφοριών (π.χ. επιστημονική έρευνα) στο διαδίκτυο. Αυτό που βλέπετε όταν προβάλλετε μια σελίδα στο Διαδίκτυο είναι η ερμηνεία του προγράμματος περιήγησης της HTML. Για να δει κάποιος τον κώδικα HTML μιας σελίδας στο Διαδίκτυο, αρκεί απλά να κάνει κλικ στο "view" στο μενού στην κορυφή του browser και επιλέξει το "source" (HTML.net.(Anon, n.d)).

3.9. CSS

Το CSS είναι μια συντομογραφία για το Cascading Style Sheets. Τα φύλλα στυλ είναι απλά αρχεία κειμένου (. Css), που αποτελούνται από γραμμές κώδικα που υποδεικνύουν στα προγράμματα περιήγησης πως να εμφανιστεί μια σελίδα HTML. Δίνουν στο σχεδιαστή περισσότερο έλεγχο πάνω στην εμφάνιση μιας ιστοσελίδας, επιτρέποντας να προσδιοριστούν ειδικά στυλ όπως οι γραμματοσειρές στη σελίδα. Με τη χρήση CSS θα μπορούσε κανείς να διαχωρίσει το περιεχόμενο HTML από την εμφάνισή του ή να διακρίνει το στυλ από τη δομή. Το CSS είναι ένα εργαλείο επεξεργασίας κειμένου για το διαδίκτυο.

Οι προδιαγραφές της CSS τηρούνται από το διεθνή οργανισμό προτύπων W3C (World Wide Web Consortium). Η σύνταξη θεωρείται απλή καθώς χρησιμοποιεί συγκεκριμένες λέξεις κλειδιά για να προσδιορίσει τα ονόματα διαφόρων ιδιοτήτων (χρώμα, μέγεθος, διάστιχο, κλπ). Ένα έγγραφο CSS αποτελείται από μια λίστα κανόνων. Ο κάθε κανόνας απευθύνεται σε ένα ή και περισσότερα στοιχεία. Ο κάθε κανόνας αποτελείται από μια λίστα προσδιορισμού ιδιοτήτων στην οποία καθορίζονται οι ιδιότητες δομής και μορφοποίησης των στοιχείων για τα οποία απευθύνεται ο κανόνας αυτός (Ανάργυρος Ακριβός, 2010.).

Τα φύλλα στυλ μπορεί να υπάρχουν στο εσωτερικό του εγγράφου HTML ή εξωτερικά ή να επισυνάπτεται ως ξεχωριστό αρχείο. Ένα εσωτερικό φύλλο στυλ επηρεάζει μόνο το έγγραφο HTML στο οποίο είναι ενσωματωμένο. Ενώ ένα εξωτερικό φύλλο στυλ, είναι ανεξάρτητο και μπορεί να συνδέεται με ένα ή πολλά έγγραφα HTML. Με εξωτερικά φύλλα στυλ, υπάρχει δυνατότητα διαμόρφωσης της εμφάνισης μιας ιστοσελίδας με την αλλαγή ενός μόνο αρχείου.

Ένας από τους λόγους που δημιουργήθηκε το CSS ήταν να διαχωρίσει το περιεχόμενο HTML από την εμφάνισή του, ή το ύφος από τη δομή. Με αυτόν τον τρόπο, υπάρχει μόνο ένα αρχείο να ελέγχει την εμφάνιση μιας ολόκληρης ιστοσελίδας. Αυτό εξοικονομεί χρόνο και επιτρέπει την ευκολότερη συντήρηση του δικτυακού τόπου. Για παράδειγμα, η αλλαγή του στυλ

της γραμματοσειράς σε όλες τις συνδέσεις σε μια ιστοσελίδα από «Verdana» σε «Courier», θα χρειαζόταν την τροποποίηση όλων των ετικετών της γραμματοσειράς μέσα σε κάθε σελίδα HTML. Με το CSS αρχείο η αλλαγή πραγματοποιείται μία φορά. Επιπλέον οι πιθανότητες λάθους μειώνονται, εξοικονομώντας ακόμη περισσότερο χρόνο. Ως ξεχωριστό αρχείο, ένα φύλλο στυλ μπορεί εύκολα να τροποποιηθεί για διαφορετικούς τύπους προγραμμάτων περιήγησης, προσφέροντας προσβασιμότητα στον παγκόσμιο ιστό για όλους τους τύπους χρηστών.

Τα πλεονεκτήματα του CSS είναι η ακριβής μορφοποίηση καθώς και ο διαχωρισμός του περιεχομένου HTML από την εμφάνιση, με αποτέλεσμα να εξοικονομείται χρόνος και να γίνεται ευκολότερη η συντήρηση της ιστοσελίδας. Ένα ακόμη επιπλέον πλεονέκτημα είναι η προσβασιμότητα στον παγκόσμιο ιστό (TuyetNquyen, 2005.).

Ανακεφαλαιώνοντας, σε αυτό το κομμάτι της πτυχιακής παρουσιάστηκαν λεπτομερώς οι τεχνολογίες που χρησιμοποιήθηκαν για την δημιουργία της εφαρμογής. Ιδιαίτερη προσοχή επικεντρώθηκε στο πλαίσιο ανοικτού κώδικα που χρησιμοποιείται για τη δημιουργία δυναμικών ιστοσελίδων με PHP, το CodeIgniter. Το πλαίσιο CodeIgniter είναι βασισμένο στο πρότυπο «Μοντέλο-Άποψη-Ελεγκτής», το οποίο χωρίζει τη λογική της εφαρμογής από τη λογική της παρουσίασης. Ιδιαίτερη έμφαση δόθηκε στην τεχνολογία XML-RPC υπεύθυνη για την ανάκτηση πληροφοριών καθώς αποτελεί και την κύρια λειτουργία της εφαρμογής. Εν τέλει έγινε παρουσίαση των χαρτών της Google που ενσωματώθηκαν σε μια σελίδα, η οποία παρέχει πληροφορίες στους επισκέπτες με την βοήθεια των δεικτών και των παραθύρων πληροφοριών που προστέθηκαν στον χάρτη. Στο επόμενο κεφάλαιο αναλύεται η αρχιτεκτονική της εργασίας με την βοήθεια των διαγραμμάτων UML.

4. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΡΓΑΣΙΑΣ

4.1. Ενοποιημένη γλώσσα σχεδιασμού (unified modeling language) UML

Η UML είναι μια γλώσσα μοντελοποίησης για την ανάλυση και το σχεδιασμό η οποία προέκυψε από επιρροές της μεθοδολογίας Booch, της OMT (Object Modeling Technique) και της OOSE (Object Oriented Software Engineering) (Μαρία Βίρβου (n.d.)). Δεν είναι ολόκληρη μεθοδολογία. Η γλώσσα UML αναπτύχθηκε από τους κύριους συντελεστές των μεθοδολογιών που προαναφέρθηκαν δηλ. από τον G. Booch, τον J. Rumbaugh και τον I. Jacobson. Η γλώσσα UML περιέχει στοιχεία και από άλλες μεθοδολογίες όπως την Odell για ταξινόμηση, την Fusion για περιγραφές και λειτουργίες, την Meyer για προϋποθέσεις και αποτελέσματα κ.τ.λ.

Οι Booch, Rumbaugh και Jacobson υποστηρίζουν ρητώς ότι: η UML είναι γλώσσα μοντελοποίησης και όχι μεθοδολογία. Η μεθοδολογία αποτελείται από μια γλώσσα μοντελοποίησης και μια διαδικασία. Η UML ορίζει μόνο τη γλώσσα μοντελοποίησης και όχι τη διαδικασία. Η γλώσσα μοντελοποίησης βοηθάει στην περιγραφή του σχεδιασμού. Η διαδικασία ορίζει τον τρόπο δημιουργίας του σχεδιασμού.

Η UML παρέχει το συμβολισμό για ανάλυση και σχεδιασμό. Μερικά στοιχεία του συμβολισμού (π.χ. τάξεις, συσχετισμοί, αθροίσματα, κληρονομικότητα) εισάγονται στην ανάλυση. Άλλα στοιχεία του συμβολισμού (π.χ. ιδιότητες) εισάγονται στο σχεδιασμό.

Η UML ορίζει δύο εργαλεία: ένα συμβολισμό και ένα μεταμοντέλο. Ο συμβολισμός ορίζει τα διαγράμματα της UML. Το μεταμοντέλο είναι ο ορισμός του συμβολισμού (Liberty J., 1998). Π.χ. μπορούμε να χρησιμοποιήσουμε το συμβολισμό της UML για να ορίσουμε τον ίδιο το συμβολισμό.

Η σχεδίαση γίνεται για να μπορέσει κανείς να κατανοήσει το σύστημα που αναπτύσσει. Έτσι δημιουργώντας σχέδια πρέπει να επιτυγχάνονται τέσσερις στόχοι (Διομήδης Σπινέλλης, 2008.). Ο πρώτος είναι η οπτική αναπαράσταση του συστήματος που υπάρχει ή πρόκειται να κατασκευαστεί. Ο δεύτερος είναι ο προσδιορισμός της δομής και της συμπεριφοράς του συστήματος. Ο τρίτος η δημιουργία ενός προτύπου για να βασιστεί η κατασκευή του συστήματος και ο τέταρτος η τεκμηρίωση των αποφάσεων που λήφθηκαν.

Η UML περιλαμβάνει τρία βασικά στοιχεία τις οντότητες, τις σχέσεις και τα διαγράμματα.

ΔΙΑΓΡΑΜΜΑΤΑ UML

Η UML ορίζει εννέα είδη διαγραμμάτων για να αναπαραστήσει τις διαφορετικές απόψεις μοντελοποίησης. Τα **διαγράμματα περιπτώσεων χρήσης** (*use case diagram*), αναπαριστούν τις λειτουργίες ενός συστήματος από την οπτική γωνία του χρήστη. Τα διαγράμματα δομής που αποτελούνται από το **διάγραμμα τάξεων** που αναπαριστούν τη στατική δομή όσον αφορά στις τάξεις και τις σχέσεις τους και τα **διαγράμματα αντικειμένων** τα οποία αναπαριστούν αντικείμενα και τις σχέσεις τους. Αντιστοιχούν σε απλοποιημένα διαγράμματα συνεργασίας που δεν αναπαριστούν μετάδοση μηνυμάτων. Ένα ακόμα είδος που ορίζει η UML είναι τα **διαγράμματα συμπεριφοράς** τα οποία περιλαμβάνουν τα **διάγραμμα καταστάσεων** (*statechart diagram*), αναπαριστούν τη συμπεριφορά της τάξης όσον αφορά στις καταστάσεις της και το **διάγραμμα δραστηριοτήτων** (*activity diagram*), σε αυτήν την περίπτωση η αναπαράσταση αφορά τη συμπεριφορά μιας λειτουργίας ως σύνολο ενεργειών. Τα **διαγράμματα αλληλεπίδρασης** τα οποία χωρίζονται σε **διάγραμμα σειράς** (*sequence diagram*) που με τη σειρά τους αποτελούν την χρονική αναπαράσταση των αντικειμένων και των αλληλεπιδράσεών τους αλλά και τα **διάγραμμα συνεργασίας** (*collaboration diagram*) τα οποία αναπαριστούν τα αντικείμενα, τις συνδέσεις και τις αλληλεπιδράσεις. Εν τέλει τα διαγράμματα ολοκληρώνονται με τα **διαγράμματα δομής υλοποίησης** τα οποία αποτελούνται από τα **διάγραμμα εξαρτημάτων** (*component diagram*) τα οποία αναπαριστούν τα φυσικά εξαρτήματα μιας εφαρμογής και τα **διαγράμματα διανομής** (*deployment diagram*) που αναπαριστούν τη διανομή των εξαρτημάτων σε συγκεκριμένα τεμάχια του hardware (υλικού).

4.2. Σύλληψη απαιτήσεων

Κάθε πληροφοριακό σύστημα πρέπει να πληροί κάποιες απαιτήσεις ώστε το αποτέλεσμα του σχεδιασμού του να το καθιστά περισσότερο λειτουργικό και αποτελεσματικό. Οι βασικότερες απαιτήσεις που πρέπει να οριστούν σε μια εφαρμογή που σκοπό έχει την ανάκτηση δεδομένων από μια απομακρυσμένη βάση, όπως στο παράδειγμά μας, αφορούν κυρίως τη λειτουργικότητα της εφαρμογής. Οι λειτουργικές απαιτήσεις αποσκοπούν στη βελτίωση και την ανάπτυξη της εφαρμογής. Σημαντικότερη λειτουργία είναι η σωστή αποστολή αιτήματος (request) από την εφαρμογή και απάντηση του συστήματος, καθώς και η ακριβείς προβολή των δεδομένων της βάσης στο παράθυρο πληροφοριών (infowindow) της εφαρμογής.

4.3. Ανάλυση και Σχεδιασμός

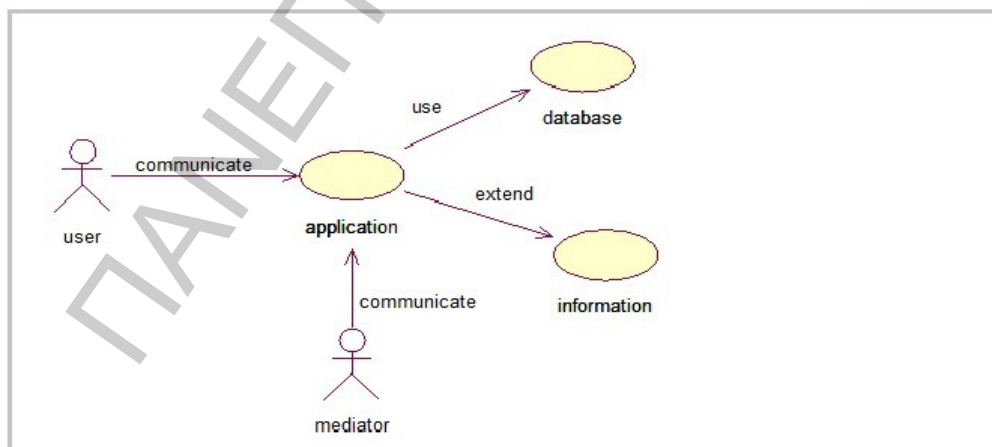
Διάγραμμα περιπτώσεων χρήσης

Οι περιπτώσεις χρήσης εφευρέθηκαν από τον Ivan Jacobson (Μαρία Βίρβου (n.d.)). Περιγράφουν τη συμπεριφορά ενός συστήματος από την οπτική γωνία ενός χρήστη. Επιτρέπουν τον ορισμό των ορίων του συστήματος και του περιβάλλοντος.

Μία περίπτωση χρήσης αντιστοιχεί σε ένα συγκεκριμένο είδος χρήσης του συστήματος. Είναι μια εικόνα της λειτουργικότητας ενός συστήματος το οποίο ενεργοποιείται για να ανταποκριθεί σε έναν εξωτερικό **ενεργοποιό** (actor). Το μοντέλο περιπτώσεων χρήσης περιλαμβάνει τις ίδιες τις περιπτώσεις χρήσης και τους ενεργοποιούς (actors).

Το σύνολο της λειτουργικότητας ενός δεδομένου συστήματος καθορίζεται διαμέσου της μελέτης των **λειτουργικών απαιτήσεων κάθε ενεργοποιού**. Αυτές εκφράζονται με τις περιπτώσεις χρήσης σε μορφή οικογενειακών διεπαφών. Οι ενεργοποιοί συμβολίζονται με ανθρωπάκια που ενεργοποιούν τις περιπτώσεις χρήσης. Οι περιπτώσεις χρήσης συμβολίζονται με ελλείψεις. Ένας ενεργοποιός αναπαριστά ένα ρόλο που παίζεται από ένα άτομο ή πράγμα και αλληλεπιδρά με το σύστημα.

Το διάγραμμα περιπτώσεων χρήσης αναπαριστά τις λειτουργίες του συστήματος από την οπτική γωνία του χρήστη. Παρακάτω περιγράφονται οι ενεργοποιοί του συστήματος, οι χρήστες (users) και ο μεσίτης (mediator). Οι περιπτώσεις χρήσης προέκυψαν από τον ορισμό του προβλήματος και τη μελέτη του συστήματος. Η διαδικασία συνοπτικά έχει ως εξής: πρώτη περίπτωση, ο ενεργοποιός χρήστη εισέρχεται στην εφαρμογή(application) και μπορεί να βρει τις πληροφορίες(information) που επιθυμεί. Τις πληροφορίες αυτές η εφαρμογή τις «τραβάει» μέσω του webservice από την βάση (database) της κτηματομεσιτικής πύλης. Στη δεύτερη περίπτωση, ο ενεργοποιός μεσίτης έχει αντίστοιχη δυνατότητα με τον χρήστη αλλά και τη δυνατότητα προσθήκης πληροφοριών στην εφαρμογή.



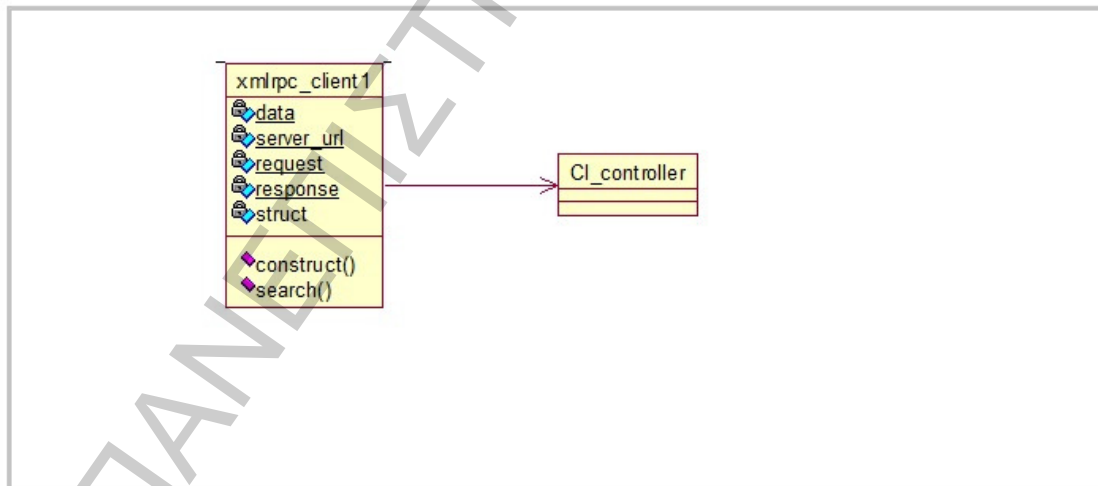
Σχήμα 9. Διάγραμμα περιπτώσεων χρήσης.

Διάγραμμα τάξεων

Οι τάξεις αναπαρίστανται ως ορθογώνια τα οποία περιέχουν διαχωρισμούς. Το πρώτο τμήμα του ορθογωνίου περιέχει το όνομα της τάξης, το δεύτερο περιέχει τα χαρακτηριστικά της τάξης το τρίτο τις λειτουργίες της.

Στο διάγραμμα τάξεων αναπαρίστανται μια στατική δομή των τάξεων και των σχέσεων τους. Στο διάγραμμα που ακολουθεί, οι τάξεις είναι: `xmlrpc_client1`, `CI_controller`. Οι σχέσεις που καθορίζουν τις τάξεις είναι: η κλάση `xmlrpc_client1` επεκτείνει (`extends`) το `CI_controller`. Αναλυτικότερα, η τάξη `xmlrpc_client1` αποτελείται από τη μέθοδο `construct()` και τη μεταβλητή `data`. Η μέθοδος αυτή είναι δημόσια (`public`) και εκεί δημιουργείται το στιγμιότυπο (`instance`) της κλάσης. Επίσης έχει την ιδιότητα όταν προστίθεται κάτι να ισχύει και για τις υπόλοιπες μεθόδους. Η μεταβλητή `data` παίρνει την τιμή του `baseurl`, που επιστρέφει τη διεύθυνση URL της τοποθεσίας, όπως ορίζεται στο αρχείο ρυθμίσεων και την φορτώνει στην κλάση. Στη μέθοδο `construct()` φορτώνονται επίσης, το αρχείο `URLhelper` το οποίο επιστρέφει το URL της σελίδας, αλλά και η βιβλιοθήκη της `Javascript` για τη λειτουργία του αρχείου `Jquery` που περιλαμβάνεται στην εφαρμογή. Τα αρχεία αυτά φορτώνονται στην κλάση με τη βοήθεια του `this` που αναφέρεται στην τρέχουσα κλάση π.χ. `$this->load->helper('url')`. Η επόμενη μέθοδος είναι η `search` στην οποία εκτελείται όλη η διαδικασία δημιουργίας και αποστολής του XML-RPC αιτήματος (`request`) και η επιστροφή της απάντησης (`response`). Για τη διαδικασία αυτή χρησιμοποιήθηκαν αρκετές μεταβλητές: η μεταβλητή `server_url` που φορτώνει το `site_url`. Μια ακόμη μεταβλητή που χρησιμοποιείται είναι η `struct`, η οποία περιλαμβάνει έναν πίνακα με δεδομένα που θα σταλούν στο διακομιστή XML-RPC. Η μεταβλητή `request` που δημιουργεί το αίτημα και τοποθετεί σε έναν πίνακα για μεταφορά, περιέχει και τον απαραίτητο κωδικό πρόσβασης για να επιτραπεί η σύνδεση στη βάση δεδομένων. Στη συνέχεια ορίζεται η μεταβλητή `response` για να εμφανιστεί η απάντηση που στέλνει πίσω ο XML-RPC Server. Στην μέθοδο αυτή φορτώνονται και οι απαραίτητες βιβλιοθήκες όπως η βιβλιοθήκη XML-RPC κ.α. με τη χρήση του `this` αναφέρεται στη συγκεκριμένη μέθοδο.

Στα παρακάτω διαγράμματα θα υπάρξουν και παραδείγματα που εστιάζουν στη λειτουργικότητα του προτύπου MVC που χρησιμοποιεί το `codeigniter`, παρουσιάζονται οι λειτουργίες των κλάσεων ελεγκτής και άποψη. Το διάγραμμα αυτό ισχύει και για το παράδειγμα του MVC.



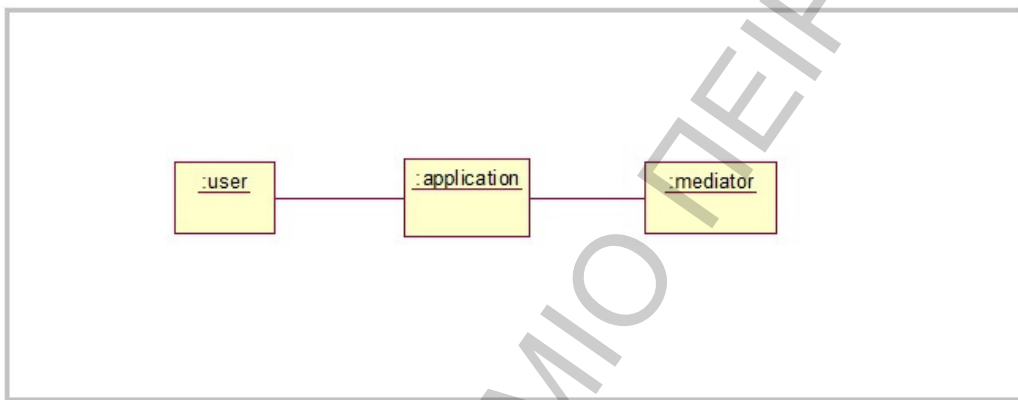
Σχήμα 10. Διάγραμμα τάξεων.

Διάγραμμα αντικειμένων

Κάθε αντικείμενο αναπαρίσταται με ένα ορθογώνιο, το οποίο περιέχει είτε το όνομα του αντικειμένου, είτε το όνομα και την τάξη του αντικειμένου (χωρισμένα με άνω και κάτω τελεία), ή μόνο την τάξη του αντικειμένου (σε αυτή την περίπτωση το αντικείμενο θεωρείται ότι είναι ανώνυμο). Το όνομα από μόνο του αντιστοιχεί σε ένα ελλειπές μοντέλο, στο οποίο η τάξη του αντικειμένου δεν έχει ακόμα καθοριστεί. Η τάξη από μόνη της αποφεύγει την εισαγωγή μη απαραίτητων ονομάτων στα διαγράμματα, ενώ επιτρέπει την έκφραση γενικών μηχανισμών που είναι έγκυροι για πολλά αντικείμενα.

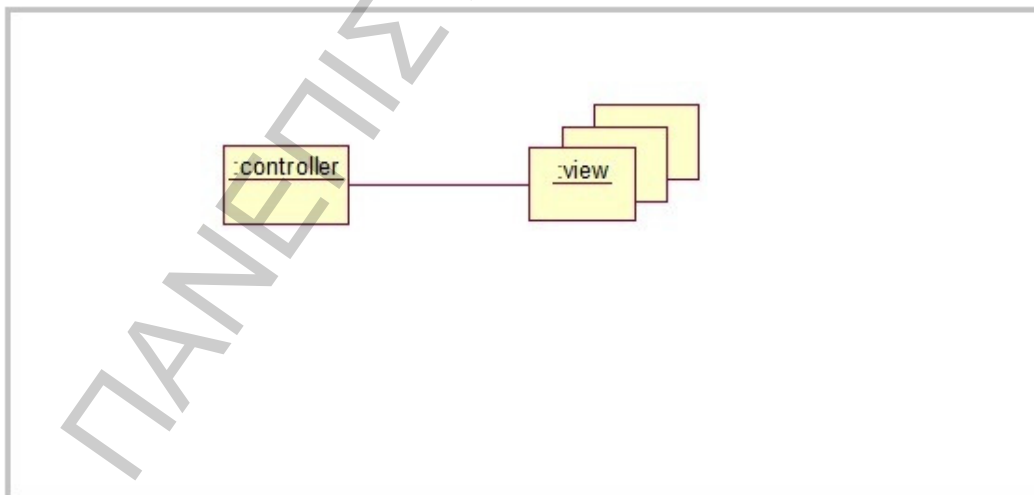
Τα αντικείμενα συνδέονται μέσω συνδέσμων, οι οποίοι είναι στιγμιότυπα των συσχετισμών μεταξύ των τάξεων των αντικειμένων που μελετώνται.

Τα διαγράμματα αντικειμένων αναπαριστούν τα αντικείμενα και τις σχέσεις τους, αντιστοιχούν σε απλοποιημένα διαγράμματα συνεργασίας και δεν αναπαριστούν μετάδοση μηνυμάτων. Στη περίπτωση μας παριστάνεται μια απλή σχέση του χρήστη και του μεσίτη με την εφαρμογή.



Σχήμα 11. Διάγραμμα αντικειμένων για τη σχέση του χρήστη και του μεσίτη με την εφαρμογή.

Το παρακάτω διάγραμμα αντικειμένων απεικονίζει την απλή σχέση Ελεγκτή (controller)-Άποψης (view). Στην εφαρμογή υπάρχει ένας ελεγκτής που φορτώνει μια άποψη.



Σχήμα 12. Διάγραμμα αντικειμένων για το πρότυπο «Μοντέλο-Άποψη-Ελεγκτής».

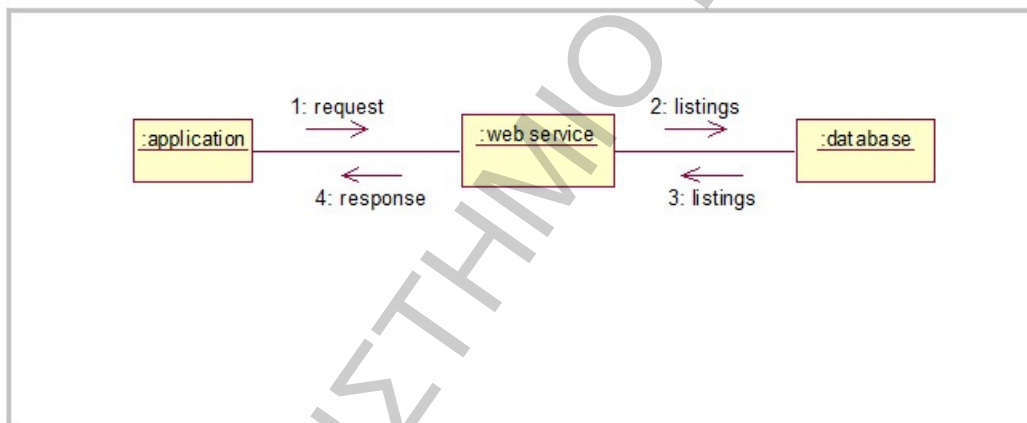
Διάγραμμα συνεργασίας

Τα διαγράμματα συνεργασίας απεικονίζουν τις αλληλεπιδράσεις ανάμεσα στα αντικείμενα, χρησιμοποιώντας μία στατική χωρική δομή, που διευκολύνει την αναπαράσταση της συνεργασίας ενός συνόλου αντικειμένων. Τα διαγράμματα συνεργασίας εκφράζουν τόσο το περιβάλλον ενός συνόλου αντικειμένων (μέσω των αντικειμένων και των συνδέσμων) όσο και την αλληλεπίδραση ανάμεσα σε αυτά τα αντικείμενα (αναπαριστώντας εκπομπές μηνυμάτων). Αυτά τα διαγράμματα είναι μία επέκταση των διαγραμμάτων αντικειμένων.

Το περιεχόμενο μίας αλληλεπίδρασης περιλαμβάνει τα ορίσματα, τις τοπικές μεταβλητές που δημιουργήθηκαν κατά τη διάρκεια της εκτέλεσης, και τους συνδέσμους ανάμεσα στα αντικείμενα που συμμετέχουν στην αλληλεπίδραση.

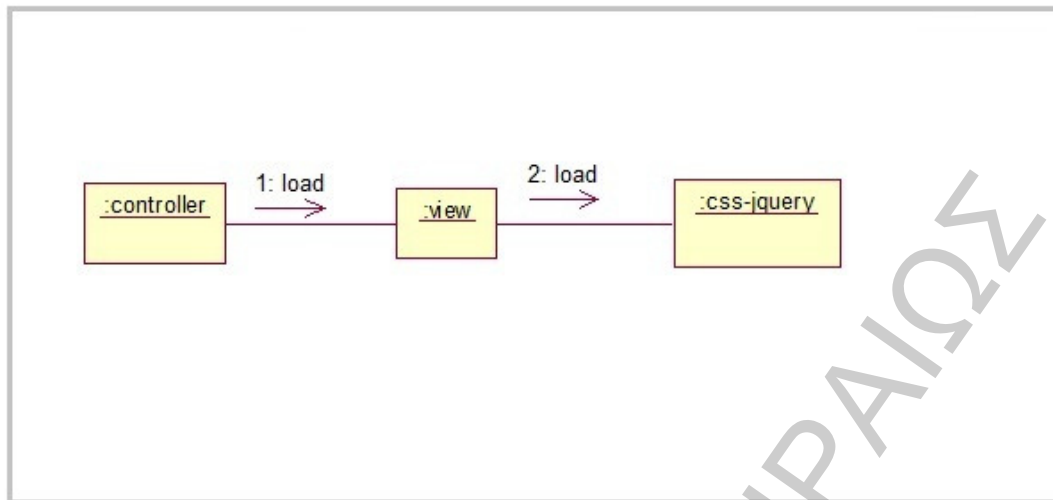
Μία αλληλεπίδραση εκτελείται από ένα σύνολο αντικειμένων που συνεργάζονται ανταλλάσσοντας μηνύματα. Αυτά τα μηνύματα εμφανίζονται κατά μήκος των συνδέσμων που συνδέουν τα αντικείμενα, χρησιμοποιώντας βέλη που δείχνουν προς τον παραλήπτη του μηνύματος.

Τα διαγράμματα συνεργασίας δείχνουν τις αλληλεπιδράσεις ανάμεσα στα αντικείμενα και ταυτόχρονα τις δομικές σχέσεις που διευκολύνουν αυτές τις αλληλεπιδράσεις. Πρώτη ενέργεια είναι η δημιουργία του request στο application, η δεύτερη ενέργεια είναι η αποστολή του request που περιλαμβάνει τη λίστα με τα δεδομένα που θέλουμε να επιστραφούν (listings) από την βάση, οι επόμενες ενέργειες είναι η αποστολή της λίστας με τα δεδομένα από τη βάση και η τελική απάντηση (response) που φτάνει στην εφαρμογή μέσω του webservice.



Σχήμα 13. Διάγραμμα συνεργασίας για τη διαδικασία ανάκτησης δεδομένων μέσω του webservice.

Στο διάγραμμα συνεργασίας η πρώτη ενέργεια είναι η φόρτωση του view από τον controller και η δεύτερη πράξη είναι η φόρτωση των αρχείων css – jquery από το view.



Σχήμα 14. Διάγραμμα συνεργασίας για το πρότυπο «Μοντέλο-Άποψη-Ελεγκτής».

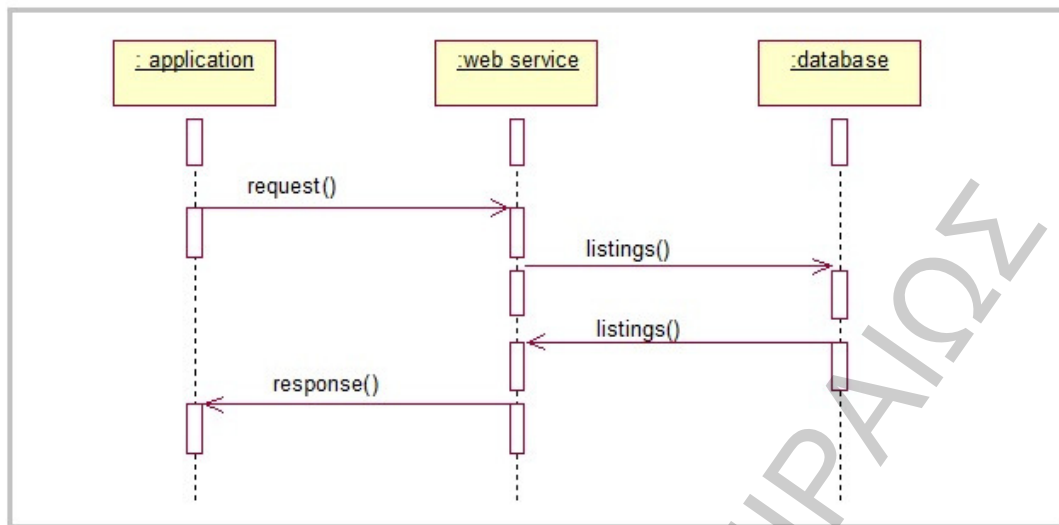
Διάγραμμα σειράς

Τα διαγράμματα σειράς αναπαριστούν αλληλεπιδράσεις ανάμεσα στα αντικείμενα από μία χρονική άποψη. Σε αντίθεση με τα διαγράμματα συνεργασίας, το περιβάλλον των αντικειμένων δεν αναπαρίσταται σαφώς. Η αναπαράσταση επικεντρώνεται στην έκφραση των αλληλεπιδράσεων. Το διάγραμμα σειράς αποτελεί τη χρονική αναπαράσταση των αντικειμένων και των αλληλεπιδράσεων τους.

Σε ένα διάγραμμα σειράς αναπαριστάται μία αλληλεπίδραση ανάμεσα σε αντικείμενα, που επικεντρώνεται στη χρονολογία της εκπομπής μηνυμάτων. Ο συμβολισμός προέρχεται από το Διάγραμμα Σειράς Μηνυμάτων Αντικειμένων της Ομάδας Προτύπου της Siemens. Ένα αντικείμενο αναπαρίσταται με ένα ορθογώνιο και μία κάθετη γραμμή, που καλείται γραμμή ζωής του αντικειμένου.

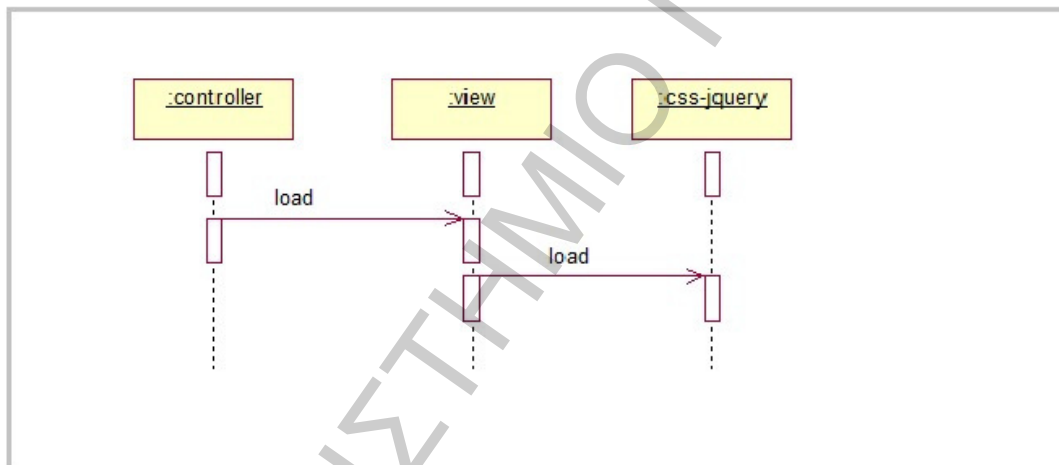
Τα αντικείμενα επικοινωνούν ανταλλάσσοντας μηνύματα, τα οποία αναπαριστώνται με οριζόντια βέλη σχεδιασμένα από τον αποστολέα του μηνύματος προς τον παραλήπτη του μηνύματος. Η σειρά αποστολής μηνύματος καθορίζεται από τη θέση του μηνύματος στον κάθετο άξονα. Ο κάθετος άξονας μπορεί να περιγράφεται με ετικέτες για να εκφράζει ακριβώς τους χρονικούς περιορισμούς, όπως για παράδειγμα στην περίπτωση σχεδιασμού ενός συστήματος πραγματικού χρόνου.

Μέσω του ακόλουθου διαγράμματος είναι εμφανείς οι διάφορες λειτουργίες στις οποίες υπόκεινται τα δεδομένα με χρονική σειρά.



Σχήμα 15. Διάγραμμα σειράς για τη διαδικασία ανάκτησης δεδομένων μέσω του webservice.

Στην συγκεκριμένη περίπτωση ισχύουν τα ίδια με τα παραπάνω.



Σχήμα 16. Διάγραμμα σειράς για το πρότυπο «Μοντέλο-Αποψη-Ελεγκτής».

Διάγραμμα δραστηριοτήτων

Ένα διάγραμμα δραστηριοτήτων είναι μία παραλλαγή των διαγραμμάτων καταστάσεων οργανωμένο σύμφωνα με ενέργειες, και στοχεύει κυρίως στην αναπαράσταση της εσωτερικής συμπεριφοράς μίας μεθόδου (την υλοποίηση μίας λειτουργίας) ή μίας περίπτωσης χρήσης.

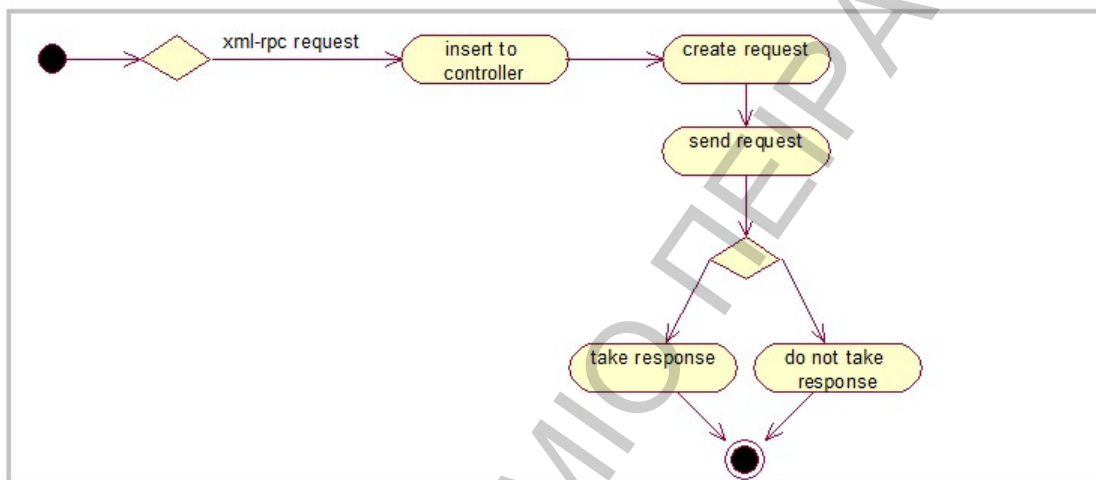
Αναπαράσταση Δραστηριοτήτων:

Ένα διάγραμμα δραστηριοτήτων αναπαριστά την κατάσταση εκτέλεσης ενός μηχανισμού ως μία σειρά βημάτων που ομαδοποιούνται σειριακά σαν παράλληλες διακλαδώσεις ροής ελέγχου. Ένα διάγραμμα δραστηριοτήτων μπορεί επίσης να αναπαραστήσει αυτή την ακολουθία βημάτων. Παρόλα αυτά, δεδομένης της διαδικαστικής φύσης της υλοποίησης των λειτουργιών – στην οποία τα περισσότερα γεγονότα απλά ανταποκρίνονται στοτέλος της προηγούμενης δραστηριότητας – δεν είναι απαραίτητο να διαχωρίσουμε καταστάσεις, δραστηριότητες, και γεγονότα συστηματικά. Γι' αυτό είναι χρήσιμο να έχουμε μία απλοποιημένη αναπαράσταση για την άμεση αναπαράσταση δραστηριοτήτων. Σε αυτό το περιβάλλον, μία δραστηριότητα αναπαρίσταται σαν στερεότυπο μίας κατάστασης. Μία

δραστηριότητα αναπαρίσταται με ένα στρογγυλεμένο ορθογώνιο, με τον ίδιο τρόπο όπως και οι καταστάσεις, αλλά περισσότερο οβάλ στην εμφάνιση.

Κάθε δραστηριότητα αναπαριστά μία συγκεκριμένη κατάσταση στην εκτέλεση της περικλείουσας μεθόδου. Οι δραστηριότητες συνδέονται με αυτόματες μεταβάσεις, που αναπαριστώνται με βέλη, με τον ίδιο τρόπο όπως οι μεταβάσεις στα διαγράμματα καταστάσεων. Όταν μία δραστηριότητα τερματίζει, η μετάβαση ενεργοποιείται και η επόμενη δραστηριότητα αρχίζει. Οι δραστηριότητες δεν έχουν εσωτερικές μεταβάσεις ή μεταβάσεις που ενεργοποιούνται από γεγονότα.

Στην UML μία απόφαση αναπαρίσταται με ένα ρόμβο με πολλές μεταβάσεις να ξεκινούν από αυτό. Το διάγραμμα δραστηριοτήτων αναπαριστά τη συμπεριφορά μιας λειτουργίας ως σύνολο ενεργειών. Στο συγκεκριμένο παράδειγμα γίνεται η ανάλυση της δημιουργίας ενός αιτήματος XML-RPC.



Σχήμα 17. Διάγραμμα δραστηριοτήτων.

Διάγραμμα εξαρτημάτων

Τα διαγράμματα εξαρτημάτων περιγράφουν τμήματα λογισμικού και τη σχέση τους με το περιβάλλον υλοποίησης. Υποδηλώνουν τις επιλογές που γίνονται κατά τη χρονική στιγμή υλοποίησης.

Τα εξαρτήματα αναπαριστούν όλα τα είδη των στοιχείων που αναφέρονται στην τμηματοποίηση των εφαρμογών λογισμικού. Μεταξύ άλλων, αυτά μπορεί να είναι απλά αρχεία ή βιβλιοθήκες που «διαβάζονται» και χρησιμοποιούνται δυναμικά.

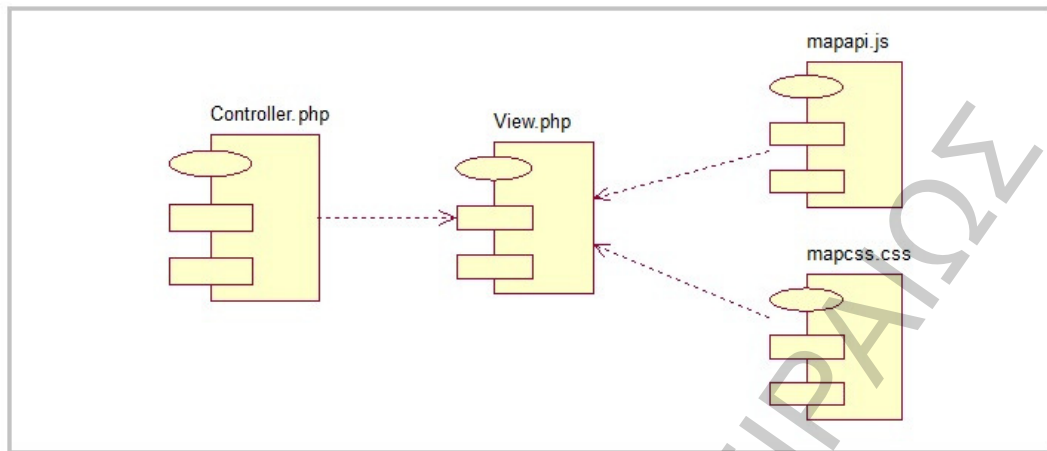
Εξ' ορισμού, κάθε τάξη στο λογικό μοντέλο έχει έναν προσδιορισμό και ένα κυρίως σώμα. Ο προσδιορισμός περιέχει το σύστημα διεπαφής της τάξης, η οποία αναπαρίσταται με το συμβολισμό της UML με ένα μικρό κύκλο που συνδέεται μέσω μιας γραμμής με το σώμα της τάξης. Το σώμα περιέχει την υλοποίηση της ίδιας της τάξης και αναπαρίσταται με το ίδιο το σύμβολο του εξαρτήματος. Το διάγραμμα εξαρτημάτων αναπαριστά τα φυσικά εξαρτήματα μιας εφαρμογής.

Οι σχέσεις εξάρτησης χρησιμοποιούνται στα διαγράμματα εξαρτημάτων για να δηλώσουν ότι ένα εξάρτημα αναφέρεται σε υπηρεσίες που προσφέρονται από άλλα εξαρτήματα. Αυτός ο τύπος εξάρτησης ανακλά επιλογές υλοποίησης. Μία σχέση εξάρτησης αναπαρίσταται με ένα βέλος με διακεκομμένη γραμμή σχεδιασμένο από τον πελάτη προς τον προμηθευτή. Σε ένα διάγραμμα εξαρτημάτων, οι σχέσεις εξάρτησης γενικά αναπαριστούν εξαρτήσεις μεταγλώττισης.

Στο παρακάτω διάγραμμα περιγράφονται οι σχέσεις εξάρτησης των αρχείων της εφαρμογής. Το αρχείο View.php φορτώνει το αρχείο marcss.css το οποίο είναι απαραίτητο για την μορφοποίηση της ιστοσελίδας. Επιπρόσθετα φορτώνει και το αρχείο marpari.js που περιέχει τον κώδικα για την εμφάνιση του χάρτη, των δεικτών (markers) και των παραθύρων

Ανάπτυξη εφαρμογής για την επικοινωνία με τη διαδικτυακή υπηρεσία XML-RPC53

πληροφορίας (infowindow). Στη συνέχεια το αρχείο Controller.php φορτώνει με την σειρά του το View.php το οποίο είναι υπεύθυνο για τις πληροφορίες που παρουσιάζονται στον χρήστη.

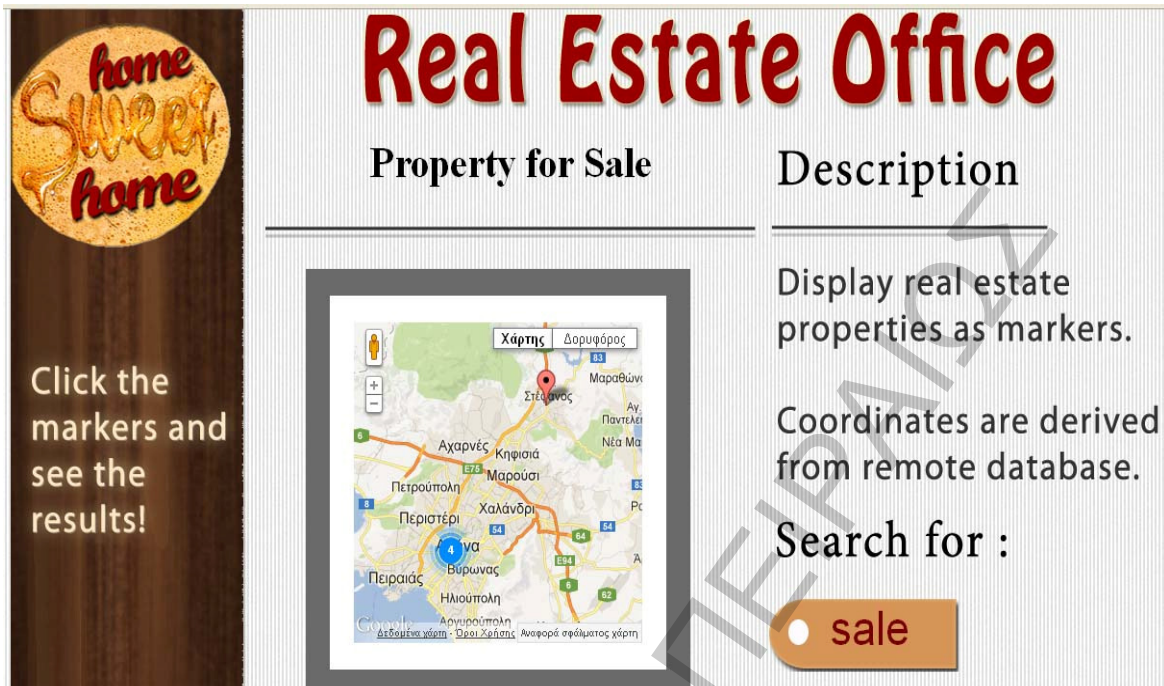


Σχήμα 18. Διάγραμμα εξαρτημάτων.

Συνοψίζοντας στο συγκεκριμένο κεφάλαιο έγινε ανάλυση της αρχιτεκτονικής της εφαρμογής. Η αρχιτεκτονική πραγματοποιήθηκε με την χρήση της Ενοποιημένης Γλώσσας Σχεδιασμού (unified modeling language) UML η οποία είναι μια γλώσσα μοντελοποίησης για ανάλυση και σχεδιασμό. Η γλώσσα αυτή ορίζει εννέα διαγράμματα όπως αναφέρθηκε και παραπάνω με κύρια τα διάγραμμα περιπτώσεων χρήσης (*use case diagram*) και τα διάγραμμα κλάσεων (*class diagram*). Τα περισσότερα από αυτά διαγράμματα σχεδιάστηκαν για την ανάλυση της εργασίας εστιάζοντας σε δύο τομείς: πρώτον στην κύρια λειτουργία της εφαρμογής η οποία είναι η χρήση της τεχνολογίας XML-RPC για την δημιουργία ενός αιτήματος με σκοπό την ανάκτηση πληροφοριών και δεύτερον στην λειτουργία του προτύπου Μοντέλο-Αποψη-Ελεγκτής στο οποίο βασίστηκε η δημιουργία της εφαρμογής. Στο κεφάλαιο που ακολουθεί θα πραγματοποιηθεί μια συνοπτική παρουσίαση της εργασίας καθώς και τα συμπεράσματα που προκύπτουν αλλά και αναφορές σε πιθανές μελλοντικές βελτιώσεις.

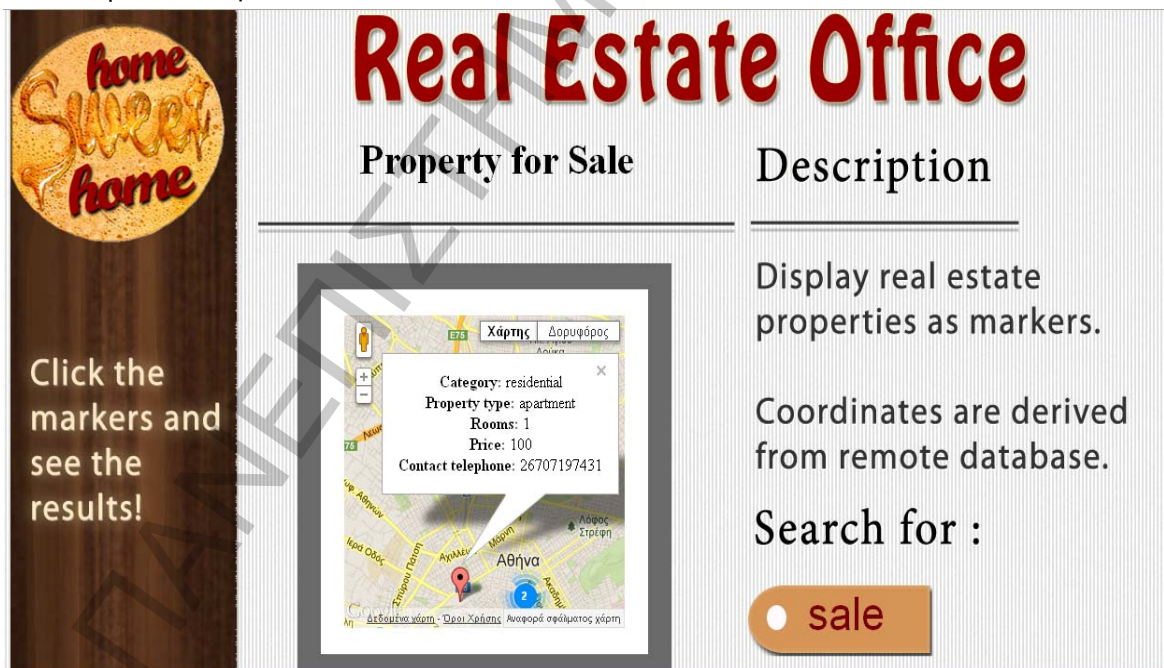
4.4. Παρουσίαση Εφαρμογής

Αρχική σελίδα εφαρμογής. Ο χάρτης φορτώνεται με συγκεκριμένες διαστάσεις και τύπο (κανονικός). Με το άνοιγμα της εφαρμογής εμφανίζονται το σύμπλεγμα δεικτών (markercluster) και οι δείκτες για την κατηγορία πώληση (sale). Επιπλέον έχει οριστεί στο χάρτη να προσαρμόζει αυτόματα το μέγιστο δυνατό επίπεδο εστίασης που να εμφανίζει όλους τους δείκτες και markercluster της εφαρμογής ανά κατηγορία.



Σχήμα 19. Αρχική σελίδα εφαρμογής.

Κάνοντας κλικ ο χρήστης πάνω σε ένα δείκτη ανοίγεται ένα παράθυρο πληροφοριών (infowindow) με πληροφορίες για την πώληση ακινήτου. Οι πληροφορίες μπορεί να είναι η τιμή του ακινήτου, τα δωμάτια από τα οποία αποτελείται κ.α.



Σχήμα 20. Άνοιγμα παραθύρου πληροφοριών.

Ο χρήστης έχει την δυνατότητα να επιλέξει ανάμεσα στις κατηγορίες πώληση (sale) και ενοικίαση (rent). Παρακάτω ο χρήστης έχει επιλέξει την κατηγορία rentόπως φαίνεται και από

τον τίτλο «PropertyforRent» και στη συνέχεια έχει ανοίξει ένα παράθυρο πληροφοριών που αντιστοιχεί σε ένα δείκτη της περιοχής της επιλογής του.

Property for Rent

Description

Click the markers and see the results!

Category: residential
Property type: apartment
Rooms: 1
Price: 100000
Contact telephone: 26707197431

Display real estate properties as markers.

Coordinates are derived from remote database.

Search for :

sale

rent

Copyright © 2013

Σχήμα 21. Εμφάνιση διαθέσιμων προς ενοίκιαση ακινήτων.

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην πτυχιακή αυτή εργασία περιγράφηκε η ανάλυση, ο σχεδιασμός και η υλοποίηση σύγχρονων τεχνολογιών για την προβολή περιεχομένου. Απώτερος σκοπός της εφαρμογής είναι πρώτον να παρουσιαστεί έμπρακτα πώς χρησιμοποιούνται και ποιες είναι οι δυνατότητες των τεχνολογιών αυτών και δεύτερον η παροχή πληροφοριών στους χρήστες.

Η εκπόνηση της μεταπτυχιακής διατριβής βασίστηκε σε σχετικά καινούριες τεχνολογίες οι οποίες προσφέρουν αρκετά πλεονεκτήματα, όπως παρατηρήθηκε κατά τη διάρκεια ανάπτυξης του κώδικα, σε έναν προγραμματιστή. Οι κύριες τεχνολογίες που χρησιμοποιήθηκαν είναι το πλαίσιο ανοιχτού κώδικα, CodeIgniter. Το CodeIgniter στηρίζεται στη γλώσσα προγραμματισμού PHP και είναι βασισμένο στο πρότυπο «Μοντέλο-Άποψη-Ελεγκτής». Χαρακτηριστικό του είναι η γρήγορη δημιουργία εφαρμογών αφού δεν απαιτείται η γραφή πολλών γραμμών κώδικα και αυτό γιατί παρέχει πολλές βιβλιοθήκες που καλύπτουν κοινές εργασίες. Σημαντικό πλεονέκτημα του πλαισίου αυτού είναι η κατανοητή τεκμηρίωσή του. Ένας αρχάριος προγραμματιστής έχει τη δυνατότητα να αναπτύξει βήμα-βήμα μια μικρή εφαρμογή.

Το βασικότερο πλεονέκτημα του CodeIgniter είναι το πρότυπο στο οποίο βασίζεται. Το πρότυπο «Μοντέλο-Άποψη-Ελεγκτής» χωρίζει τη λογική της εφαρμογής από τη λογική της παρουσίασης. Έχει ως ιδιότητα τη διαίρεση του κώδικα στα μέρη που αναφέρει το όνομά του, στο μοντέλο-αντιπροσωπεύει τις δομές δεδομένων, στην άποψη- οι πληροφορίες που παρουσιάζονται στο χρήστη και τον ελεγκτή- συνδέει το μοντέλο και την άποψη μεταξύ τους κ.α. Αυτό έχει σαν αποτέλεσμα ο κώδικας να είναι οργανωμένος και «καθαρός». Επιπλέον εξυπηρετεί τον προγραμματιστή στη γρήγορη και εύκολη διόρθωση των λαθών αλλά και στην επαναχρησιμοποίηση μέρους του κώδικα. Κατανοητό είναι πως το πλαίσιο CodeIgniter σε συνδυασμό με το πρότυπο «Μοντέλο-Άποψη-Ελεγκτής» εξασφαλίζει παραγωγικό χρόνο. Μια ακόμα σημαντική τεχνολογία που χρησιμοποιήθηκε ήταν το πρωτόκολλο XML-RPC που πραγματοποιεί απομακρυσμένες κλήσεις διαδικασίας μέσω διαδικτύου. Το πρωτόκολλο αυτό με το να επιτρέπει να λειτουργούν μαζί ανόμοια συστήματα και το γεγονός ότι δεν έχει καμία σχέση με τις γλώσσες προγραμματισμού, επιτρέποντας να επικοινωνούν προγράμματα τα οποία είναι γραμμένα σε οποιαδήποτε γλώσσα, καταφέρνει να διευκολύνει αυτού του είδους τις εργασίες αφού διαθέτει ευκολία στην επικοινωνία και το χειρισμό των δεδομένων.

Η ανάκτηση δεδομένων από υπολογιστή σε υπολογιστή είναι μια διαδικασία χρήσιμη και γρήγορη. Στην πτυχιακή χρησιμοποιήθηκε η τεχνολογία XML-RPC για την ανάκτηση πληροφοριών μέσω διαδικτυακών υπηρεσιών. Αυτό είναι και το κυριότερο πλεονέκτημα καθώς χρησιμοποιώντας την τεχνολογία αυτή δεν χρειάζεται κανείς να δημιουργήσει βάση δεδομένων και πόσο μάλλον να την επεξεργαστεί. Επιπρόσθετα χρήση έγινε και της τεχνολογίας των χαρτών της Google. Με την προσθήκη λίγων γραμμών κώδικα στην εφαρμογή, η Google σου παρέχει την δυνατότητα εμφάνισης ενός δυναμικού χάρτη και με το σύνολο των εργαλείων που διαθέτει επιτρέπει διάφορες ενέργειες όπως η προσθήκη δεικτών, παραθύρων πληροφορίας κ.α. Η Google επιτρέπει και τη μορφοποίηση του χάρτη, μπορεί δηλαδή κάποιος να ορίσει συγκεκριμένες συντεταγμένες αλλά και το γεωγραφικό πλάτος και μήκος. Επίσης δίνει τη δυνατότητα επιλογής του τύπου του χάρτη, όπως αν θα είναι δορυφορικός ή κανονικός. Όλες αυτές οι επιλογές προδιαθέτουν τη δημιουργία εύρωστων εφαρμογών. Οι χάρτες της Google παρέχουν μια αναλυτική τεκμηρίωση καθώς και μια σειρά από παραδείγματα για την κατανόηση των εργαλείων και των δυνατοτήτων τους δίνοντας έμπνευση για τη δημιουργία νέων εφαρμογών. Η διεπαφή αυτή προσφέρει την δυνατότητα δημιουργίας απλών αλλά και σύνθετων εφαρμογών. Μια σημαντική τεχνολογία καθώς οι εφαρμογές με χάρτες είναι ιδιαίτερα κατανοητές και εύχρηστες για τους χρήστες.

Η ενασχόληση με μια μεταπτυχιακή διατριβή αγγίζει διάφορους τομείς από τις τεχνολογίες που θα χρησιμοποιηθούν για τη δημιουργία της εφαρμογής μέχρι τον τρόπο που θα γραφτεί η τεκμηρίωση. Η διαδικασία αυτή αποτελεί μια καλή ευκαιρία για την αποκόμιση γνώσεων. Οι τεχνολογίες που χρησιμοποιήθηκαν στην εργασία αυτή απαιτούσαν αρκετό χρόνο για την εκμάθηση και την κατανόησή τους, όμως όπως αναφέρθηκε και παραπάνω η ανάπτυξη προγραμμάτων σε CodeIgniter ή η χρήση του πρωτοκόλλου XML-RPC είναι σημαντικά εργαλεία λόγω των δυνατοτήτων που προσφέρουν. Επιπλέον είναι σύγχρονα και η γνώση τους είναι ένα

εφόδιο για την αγορά εργασίας. Όσον αφορά την τεκμηρίωση, η αναζήτηση πληροφοριών για τις τεχνολογίες βοηθά στην καλύτερη κατανόηση τους, η σύγκριση με άλλες πτυχιακές εργασίες δίνει τη δυνατότητα να αντιληφθεί κανείς διάφορους τρόπους χρήσης των κοινών τεχνολογιών ακόμα και ο συγκεκριμένος τρόπος γραφής της τεκμηρίωσης αποτελεί μια επιπλέον γνώση. Συμπερασματικά πέραν του ότι αποτελεί μια πηγή γνώσεων, αξιοσημείωτο είναι ότι η κατανόηση των τεχνολογιών αυτών έχει σαν αποτέλεσμα την εμβάθυνση στη λογική του προγραμματισμού.

Περιορισμοί στη δημιουργία της εφαρμογής

Κατά τη διάρκεια της εκπόνησης της πτυχιακής εργασίας υπήρξαν ορισμένοι περιορισμοί. Αρχικά, το επίπεδο γνώσεων του προγραμματισμού δεν ήταν αρκετό για τις απαιτήσεις που χρειάζονταν για τη δημιουργία της εφαρμογής. Αναγκαία κρίθηκε η μελέτη τεχνολογιών και γλωσσών προγραμματισμού σχεδόν από τα αρχικά τους στάδια. Το γεγονός αυτό οδήγησε στο να καθυστερήσει η ολοκλήρωση της εφαρμογής αλλά και να γίνει μια προσαρμογή όσον αφορά τα ζητούμενά της. Εξαιτίας του περιορισμού αυτού δεν μπόρεσε να υλοποιηθεί ο δυναμικός τρόπος αλλαγής των κριτηρίων όπως του εμβαδού, της τιμής κτλ. Επιπλέον περιορισμός ήταν το γεγονός ότι δεν έγινε χρήση του πιο πρόσφατου πρωτοκόλλου JSONRPC - μιας «ελαφριάς» κλήσης απομακρυσμένης διαδικασίας. Η προδιαγραφή (specification) ορίζει διάφορες δομές δεδομένων και κανόνες γύρω από την επεξεργασία τους. Είναι αγνωστικιστής μεταφοράς (transport agnostic) δηλαδή οι έννοιες μπορούν να χρησιμοποιηθούν στο πλαίσιο της ίδιας διαδικασίας, πάνω από υποδοχές (sockets), μέσω HTTP, ή να περνούν διάφορα μηνύματα σε περιβάλλοντα. Χρησιμοποιεί JSON (RFC 4627), για τη μορφή των δεδομένων (MattMorley of MPCMTechologiesLLC, 2013.) - αλλά τουXMLRPC επειδή με το συγκεκριμένο πρωτόκολλο ήταν υλοποιημένο το web service με το οποίο επικοινωνούσε η εφαρμογή. Επιπλέον δεν χρησιμοποιήθηκαν οι τεχνολογίες AJAX / JSON . Η χρήση της τεχνικής AJAX επιτρέπει σε μια ιστοσελίδα να ανανεώνεται ασύγχρονα (asynchroneously) ανταλλάσσοντας στο παρασκήνιο μικρού όγκου δεδομένα με τον server επιτρέποντας να ανανεώνονται μέρη της ιστοσελίδας (ένα div για παράδειγμα), χωρίς να ανανεώνεται ολόκληρη η σελίδα (Άκης Καργιοφύλλης, 2013.). Το JSON από την άλλη πλευρά είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Οι τεχνολογίες αυτές δεν ήταν απαραίτητες για την υλοποίηση της εφαρμογής (Google, Εισαγωγή στο Json, 2013.).

Μελλοντικές βελτιώσεις εφαρμογής

Η εφαρμογή μπορεί να επεκταθεί και να βελτιωθεί με διάφορους τρόπους. Μια πιθανή βελτίωση αφορά στη δημιουργία του αιτήματος. Στην εφαρμογή τα δεδομένα που είναι επιθυμητά να ανακτηθούν καλούνται στατικά, ορίζονται δηλαδή ποια δεδομένα θα ανακτηθούν, ενώ θα μπορούσαν να καλούνται δυναμικά. Πιο συγκεκριμένα ορίζεται στον ελεγκτή ότι τα δεδομένα που χρειάζονται να επιστραφούν με την βοήθεια των webservicessείναι για παράδειγμα η γλώσσα στην οποία είναι επιθυμητό να επιστραφούν τα δεδομένα, μια συγκεκριμένη κατηγορία ακινήτου κτλ. Αυτή η τακτική θα μπορούσε να αποφευχθεί και δυναμικά να επιστρέφονταν για παράδειγμα κάθε κατηγορία ακινήτου.

Στη συνέχεια θα μπορούσε ο χρήστης να κάνει μια αναζήτηση σύμφωνα με τα χαρακτηριστικά που ζητά να έχει το ακίνητο που επιθυμεί να αγοράσει ή να νοικιάσει. Δηλαδή στην κεντρική σελίδα που θα είχε πρόσβαση ο χρήστης θα μπορούσαν να υπάρχουν κουμπιά με επιλογές όπως αν το ακίνητο που επιθυμούν να αναζητήσουν να είναι διαμέρισμα, γκαρσονιέρα, μεζονέτα κτλ. Ακόμη να υπάρχει η επιλογή ορισμού της περιοχής, δηλαδή η αναζήτηση να πραγματοποιείται για παράδειγμα για την περιοχή της Αττικής και πιο συγκεκριμένα για την περιοχή του Πειραιά. Επιπλέον για την καλύτερη εξυπηρέτηση του χρήστη σημαντικό θα ήταν να ορίζετε και η τιμή στην οποία είναι επιθυμητή να κυμανθεί η αγορά ή η ενοικίαση των ακινήτων.

Επιπρόσθετη βελτίωση θα ήταν η προσθήκη ασφάλειας στο πρόγραμμα καθώς οι σελίδες είναι επιρρεπείς σε κινδύνους. Το Codeigniter παρέχει την δυνατότητα προσθήκης ασφάλειας σε μια σελίδα με την κλάση Security η οποία παρέχει μεθόδους ασφαλείας που βοηθούν στην δημιουργία μιας ασφαλής εφαρμογής κα αυτό το επιτυγχάνει επεξεργάζοντας τα δεδομένα εισόδου. Το Codeigniter έρχεται με ένα CrossSiteScriptingHack φίλτρο πρόληψης το οποίο μπορεί να τρέξει αυτόματα φιλτράροντας όλα τα POST και COOKIE δεδομένα που αντιμετωπίζει ή μπορεί να τρέξει βάσει του είδους των δεδομένων (EllisLabInc, 2013.).

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Ακριβός Ανάργυρος, 2010. Κατασκευή portlet συναρμολόγησης και εξατομικευμένης διαμόρφωσης προϊόντων πληροφοριακού τύπου [pdf]
Availableat:<<http://nefeli.lib.teicrete.gr/browse/stef/epp/2010/AkrivosAnargyros/attached-document-1271055707-261759-10785/Akrivos2010.pdf>>
- Βίρβου Μαρία (n.d). Τεχνολογία Λογισμικού [pdf] Availableat:
<<http://gunet2.cs.unipi.gr/eclass/modules/document/document.php?course=TME119>>
- ΔημητρίουΘεοχάρης, 2007. Web Services και SOAP [pdf] Available at:
<<http://pdplab.it.uom.gr/project/soap/Downloads/WebservicesTheory.pdf> >
- Εισαγωγή στο Json(Anon, n.d.) [online] Available at: <<http://www.json.org/json-el.html>> [Accessed 16 March 2013].
- Ευστάθιος Νικόλαος, 2009. Υπηρεσίες Ιστού στον Αγροτικό Τομέα [pdf]
Availableat:<http://dSPACE.aua.gr/xmlui/bitstream/handle/10329/475/Eustathiou_N.pdf?sequence=1>
- ΚαργιοφύλληςΆκης, 2013. WLEARN. [online] Available at:
<<http://www.wlearn.gr/index.php/ajax-intro2>> [Accessed 16 March 2013].
- ΚαλαποδάςΓεώργιος, 2005-2006. ΑνακάλυψηWebServicesσεσυστήματαομοτίμων (Peer-to-Peer) [pdf]Availableat:
<[http://nemertes.lis.upatras.gr/jspui/bitstream/10889/526/1/Nimertis_Kalapodas\(m\).pdf](http://nemertes.lis.upatras.gr/jspui/bitstream/10889/526/1/Nimertis_Kalapodas(m).pdf)>
- Μανδουράρη Δήμητρα (n.d.) Πληροφοριακό σύστημα για την καθοδήγηση χρηστών με GIS[pdf]
Availableat:<<http://nefeli.lib.teicrete.gr/browse2/stef/epp/2011/MandourariDimitra/attached-document-1308739738-773291-16209/Mandourari2011.pdf>>
- Μηνιατής Γεράσιμος, 2011. Η τεχνολογία των WebServices [pdf] Availableat:
<http://ifestos.teilar.gr/index.php?option=com_docman&task=doc_view&gid=69>
- Νώτης Χρήστος, 2011. Δημιουργία τουριστικού οδηγού για τον δήμο Πάργας με την χρήση ενός webcontentmanagement (Joomla) [pdf] Availableat:
<http://ifestos.teilar.gr/index.php?option=com_docman&task=doc_view&gid=37>
- Πιτόπουλος Αναστάσιος, 2011. Δημιουργία mobileapplicationiOS (iPhone/iPad) για την οργάνωση οικιακών εξόδων, ακολουθώντας πρακτικά πρότυπα προγραμματισμού [pdf]
Availableat:<http://ifestos.teilar.gr/index.php?option=com_docman&task=doc_view&gid=72>
- Σπινέλλης Διομήδης, 2008. Ανάλυση και σχεδίαση με UML. [online] Available at:<<http://www.dmst.aueb.gr/dds/ism/oo/indexw.htm>>
- ΣταυγιουνουδάκηςΙωάννης (n.d.)
ΗλεκτρονικόλειτουργικόπαιγνίδιμεχρήσηMVCPHPFramework[pdf] Availableat:
<<http://nefeli.lib.teicrete.gr/browse/stef/epp/2011/Stavgianoudakisloannis/attached-document-1308313029-601074-2294/stavgianoudakis.pdf>>
- Computer Notes (Anon, n.d.)[online] Available at:<<http://ecomputernotes.com/fundamental/disk-operating-system/what-is-marshalling>>
- Curteanu Mihai, 2010. Using the Model-View-Controller for Creating Applications for Project Management [pdf] Available at: <<http://www.opensourcejournal.ro/2010-Volume02/number04/paper011-fullpaper.pdf>>
- EllisLab Inc, 2006-2012. CodeIgniter User Guide Version 2.1.3. [online] Available at:
<<http://ellislab.com/codeigniter/user-guide/libraries/security.html>>[Accessed 16 March 2013].
- EllisLab Inc, 2006-2012 CodeIgniter User Guide Version 2.1.3. [online] Available at:
<<http://ellislab.com/codeigniter/user-guide/libraries/xmlrpc.html>>
- Ανάπτυξη εφαρμογής για την επικοινωνία με τη διαδικτυακή υπηρεσία XML-RPC60

- Goodell Jeffers Jill, 1999-2004. What is HTML anyway? .[online] Available at: <<http://www.goodellgroup.com/tutorial/chapter1.html>>
- Google Developers, 2013. Google Maps JavaScript API v3. [online] Available at: <<https://developers.google.com/maps/documentation/javascript/>>
- Google Project Hosting, (n.d.) [online] Available at: <<http://google-maps-utility-library-v3.googlecode.com/svn/trunk/markerclusterer/docs/reference.html>>
- HTML.net. (Anon, n.) [online] Available at: <<http://www.html.net/tutorials/html/lesson2.php>>
- HTML.net. (Anon, n.d.) [online] Available at: <<http://www.html.net/tutorials/html/lesson3.php>>
- iDesiner, 2013. iDesiner:Εισαγωγή στην JQuery. [online] Available at: <http://www.idesigner.gr/>
- Larsson and Sprague, 2004. ACM DL:eXtreme Deployment: Distributing and Configuring 450 Student Laptops in Five Hours [pdf] Available at: <<http://dl.acm.org/results.cfm?h=1&cfid=296509066&cftoken=52563714>>
- Maestro Solutions LTD, 2004-2013. Web design library. [online] Available at: <<http://www.webdesign.org/web-programming/php/advantages-of-php-programming.21905.html>>
- Mprley Matt of MPCM Technologies LLC, 2005-2013. JSON-RPC 2.0 Specification.[online] Available at: <<http://www.jsonrpc.org/specification>> [Accessed 16 March 2013].>
- Nquyen Tuyet, 2005. Cascading Style Sheets.[online] Available at: <<http://web.wellesley.edu/Computing/Dreamweaver/CSS/cssMain.html>>
- Refsnes Data, 1999-2013. W3schools.com. [online] Available at: <http://www.w3schools.com/jquery/jquery_syntax.asp>
- Scripting News Inc, 2004-2011.XML-RPC.Com Home.[online] Available at: <<http://xmlrpc.scripting.com/default.html>>
- Scripting News Inc, 2004-2011.XML-RPC.Com Specification.[online] Available at: <<http://xmlrpc.scripting.com/spec.html>>
- Shannon Ross, 2000-2013. HTML Source: HTML tutorials [online] Available at: <<http://www.yourhtmlsource.com/starthere/whatishtml.html>>
- Teresco D. James (n.d). ACM DL:Using the Google maps api with highway mapping [pdf] Available at:<<http://dl.acm.org/results.cfm?h=1&cfid=296509066&cftoken=52563714>>
- Vandenburg Michael, 2007. Emerald Article: Using Google Maps as an interface for the library catalogue [pdf] Available at:<<http://www.emeraldinsight.com/>>
- W.P. de Sá, 2010. Elsevier:Upgrading a TCABR data analysis and acquisition system for remote participation using Java, XML, RCP and modern client/server communication/authentication [pdf] Available at:<<http://www.sciencedirect.com/>>
- Zhu Ying, 2012. IEE Xplore: Introducing Google Chart Tools and Google Maps API in Data Visualization Courses[pdf] Available at:<<http://ieeexplore.ieee.org/xpl/ProdList.jsp?product=ASPP&publicationType=periodical&startLetter=a>>

ΠΑΡΑΡΤΗΜΑΚΩΔΙΚΑ

ΑρχείοController

```
<?php

class Xmlrpc_client1 extends CI_Controller {

public function __construct()
    {
        parent::__construct();

        $this->load->helper('url');
        $this->data["baseUrl"] = base_url();
        $this->load->library('javascript');

    }

function search( $listingType = "sale" )
{
    $this->load->helper('url');
    $server_url = site_url('xmlrpc_server');

    $this->load->library('xmlrpc');
    $this->xmlrpc->server('http://webservices.sp-
orfeas.com/mobileDeviceSearch/v1_0', 80);
    $this->xmlrpc->method('sg_mobile.search');

    $struct = array(
        'language'=> 'gr',
        'listingType'=> $listingType,
        'category'=>'residential',
        'limit'=>'5',
        'pricelow' => 500,
        'pricehigh' => 800,
        'livingAreaLow'=>50,

    );

    $request = array (
        array( "8c785e5bd1b2ce408bc2b140d", 'string'),
        array( $struct, 'struct')
    );
}
```

```
$this->xmlrpc->request($request);

if ( ! $this->xmlrpc->send_request() )
{
    echo $this->xmlrpc->display_error();
}
else
{

    $response = $this->xmlrpc->display_response();

    foreach ( $response["listings"] as $listingKey => $listingValue )
    {

        $this->data["listings"][$listingKey]["lng"] = $listingValue["longitude"];
        $this->data["listings"][$listingKey]["lat"] = $listingValue["latitude"];
        $this->data["listings"][$listingKey]["cat"] = $listingValue["category"];
        $this->data["listings"][$listingKey]["price"] = $listingValue["price"];
        $this->data["listings"][$listingKey]["rooms"] = $listingValue["rooms"];
        $this->data["listings"][$listingKey]["tel"] = $listingValue["contactTelephone"];
        $this->data["listings"][$listingKey]["type"] = $listingValue["propertyType"];
        $this->data["listings"][$listingKey]["key"] = $listingKey;
    }

    $this->load->view("mapm",$this->data);

}

}
```

ΑρχείοView

```
<html>
<head>

<meta http-equiv="content-type" content="text/html; charset=Windows-1253" />
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />

<link rel="stylesheet" type="text/css" href="<?php echo $baseUrl; ?>css/mapcss.css" />

<script type="text/javascript"src="http://maps.googleapis.com/maps/api/js?sensor=false">
</script>

<script type="text/javascript" src="<?php echo $baseUrl; ?>js/jquery-1.8.1.min.js"></script>
<script type="text/javascript" src="<?php echo $baseUrl; ?>js/cluster.js"></script>
<script type="text/javascript" src="<?php echo $baseUrl; ?>js/Fluster2.js"></script>
<script type="text/javascript" src="<?php echo $baseUrl; ?>js/Fluster2Cluster.js"></script>
<script type="text/javascript" src="<?php echo $baseUrl; ?>js/Fluster2.packed.js"></script>
<script type="text/javascript" src="<?php echo $baseUrl; ?>js/Fluster2ClusterMarker.js"></script>
<script type="text/javascript" src="<?php echo $baseUrl; ?>js/Fluster2ProjectionOverlay.js"></script>

<script type="text/javascript" src="<?php echo $baseUrl; ?>js/mapapi.js"></script>

</head>

<body onload="initialize()">


<script type="text/javascript">
mypointsLat = new Array();
mypointsLong = new Array();
cat = new Array();
price = new Array();
rooms = new Array();
tel = new Array();
type = new Array();
</script>

<div id="map_canvas" class="map_diast"></div>

<?php foreach ($listings as $listing):?>
```

```
<script type="text/javascript">
```

```
mypointsLong[<?php echo $listing["key"]; ?>] = <?php echo $listing["lng"]; ?>;  
mypointsLat[<?php echo $listing["key"]; ?>] = <?php echo $listing["lat"]; ?>;  
cat[<?php echo $listing["key"]; ?>] = "<?php echo $listing["cat"]; ?>";  
price[<?php echo $listing["key"]; ?>] = <?php echo $listing["price"]; ?>;  
rooms[<?php echo $listing["key"]; ?>] = <?php echo $listing["rooms"]; ?>;  
tel[<?php echo $listing["key"]; ?>] = <?php echo $listing["tel"]; ?>;  
type[<?php echo $listing["key"]; ?>] = "<?php echo $listing["type"]; ?>";  
</script>
```

```
<?php endforeach;?>
```

```
<div class="links">
```

```
<a href="<?php echo $baseUrl; ?>index.php/xmlrpc_client1/search/sale">Sale</a>  
</div>
```

```
<div class="link">
```

```
<a href="<?php echo $baseUrl; ?>index.php/xmlrpc_client1/search/rent">Rent</a>  
</div>
```

```
</body>
```

```
</html>
```

ΑρχείοJquery

```
function initialize() {  
  
    var athina = new google.maps.LatLng(37.98994,23.729038);  
    var mapOptions = {  
        center: athina,  
        zoom: 10,  
        mapTypeId: google.maps.MapTypeId.ROADMAP  
    }  
    var map = new google.maps.Map(document.getElementById("map_canvas"),  
    mapOptions);  
  
    var infowindow = new google.maps.InfoWindow();  
  
    var fluster = new Fluster2(map);  
  
    var bounds = new google.maps.LatLngBounds();  
  
    for (i=0; i<mypointsLong.length; i++)  
    {  
        var temp = new google.maps.LatLng(mypointsLat[i],mypointsLong[i]);  
        markers = new google.maps.Marker ({  
            position: temp  
        });  
  
        markers.setMap(map);  
        fluster.addMarker(markers);  
        bounds.extend(temp);  
        map.fitBounds(bounds);  
  
        fluster.initialize();  
  
        google.maps.event.addListener(markers, 'click', (function(markers, i) {  
            return function() {  
                infowindow.setContent("<b>Category: </b>"+cat[i]+"<br><b> Property type:  
</b>"+type[i]+"<br><b>Rooms: </b>"+rooms[i]+  
                "<br><b>Price:</b>"+price[i]+"<br><b>Contact telephone: </b>"+tel[i]);  
  
                infowindow.open(map, markers);  
            })(markers, i);  
        });  
    }  
}
```

```
}
```

ΑρχείοCSS

```
html {  
height: 100%  
}  
  
body {  
height: 100%;  
margin: 5px;  
padding: 0;  
}  
.map_diast {  
width: 300px;  
height: 300px;  
top: -455px;  
right: -265px;  
}  
  
.links {  
position:absolute;  
top: 527px;  
right: 185px;  
}  
  
.link {  
position:absolute;  
top: 595px;  
right: 185px;  
}  
  
a {  
display: block;  
width: 120px;  
font: bold 40px/1 sans-serif;  
color:#B22222;  
text-decoration:none;  
}  
  
a:hover {  
color:#E9BE75;  
}
```

```
.hide {  
display: none; }
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ