



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	Development of an application for communication using the XML-RPC web service
Όνοματεπώνυμο Φοιτητή	ΚΛΕΙΣΣΑ ΜΑΡΙΑ
Πατρώνυμο	ΙΩΑΝΝΗΣ
Αριθμός Μητρώου	ΜΠΠΛ/ 10012
Επιβλέπων	Χρήστος Δουληγέρης, Καθηγητής
Συνεπιβλέπων	Βασίλης Μενεκλής Διδάκτορας

Ημερομηνία Παράδοσης **Οκτώβριος 2013**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Χρήστος Δουληγέρης
Καθηγητής

Δέσποινα Πολέμη Επίκουρη
Καθηγήτρια

Χαράλαμπος
Κωνσταντόπουλος
Επίκουρος Καθηγητής

ΠΕΡΙΛΗΨΗ

Η παρούσα μεταπτυχιακή διατριβή ασχολείται με το σχεδιασμό, την ανάλυση και την ανάπτυξη μιας εφαρμογής με τη χρήση ενός πρόσθετου παρουσίασης διαφανειών (cycle plugin). Χρησιμοποιούνται κυρίως δύο τεχνολογίες το Codeigniter και το XML-RPC με σκοπό την παρουσίαση δεδομένων από μια βάση δεδομένων. Βασικός στόχος είναι η εμφάνιση πραγματικών δεδομένων στο plugin, σε πραγματικό χρόνο.

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

ABSTRACT

This paper deals with the design, analysis and development of an application using a plugin for presentation (cycle plugin). Two technologies are mainly used: Codeigniter and XML-RPC to present data from a database. The main objective is the appearance of the real data on the plugin, in real time.

For the application we used the following development technologies: Codeigniter, Xml-Rpc, Php, Html, Javascript, JQuery, and cycle plugin. The analysis and design is based on object oriented techniques (uml diagrams).

ΕΥΧΑΡΙΣΤΙΕΣ

Η εκπόνηση της παρούσας μεταπτυχιακής διατριβής δεν θα ολοκληρωνόταν χωρίς την βοήθεια κάποιων ανθρώπων, τους οποίους θα ήθελα να ευχαριστήσω.

Θα ήθελα καταρχάς να ευχαριστήσω τον επιβλέποντα καθηγητή, κύριο Χρήστο Δουληγέρη για την ευκαιρία που μου έδωσε να εκπονήσω τη μεταπτυχιακή μου διατριβή πάνω σε ένα τόσο ενδιαφέρον αντικείμενο όπως είναι η εφαρμογή με προβολή εικόνων(slideshow).

Ένα μεγάλο ευχαριστώ στον διδάκτορα κύριο Βασίλειο Μενεκλή για την πολύτιμη βοήθειά του και τις πολλές ώρες που μου αφιέρωσε από το χρόνο του αλλά και την υποστήριξη και την υπομονή του καθ' όλη τη διάρκεια της εκπόνησης της μεταπτυχιακής μου διατριβής.

Επίσης θα ήθελα να ευχαριστώ τους συμφοιτητές μου Κύρκο Μαρινέλλα, Κουγιουμτζή Δημήτρη και Χριστοδούλου Απόστολο για την ψυχολογική αλλά και πρακτική υποστήριξη στην προσπάθεια εκπόνησης της μεταπτυχιακής διατριβής.

Και τέλος την οικογένειά μου για την πολύτιμη βοήθεια που μου έχουν προσφέρει σε όλη τη διάρκεια των σπουδών μου.

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη.....	3
Ευχαριστίες.....	4
Κατάλογος σχημάτων	6
Κατάλογος πινάκων.....	7
Γλωσσάρι.....	8
Κεφάλαιο 1- Εισαγωγή.....	9
1.1 Σύνοψη αναφορά στην ιστορία του διαδικτύου.....	9
1.2 Ανάλυση θέματος εργασίας.....	9
1.3 Δομή κεφαλαίων.....	9
Κεφάλαιο 2 – Βιβλιογραφική Έρευνα.....	11
Κεφάλαιο 3 – Μεθοδολογία Ανάπτυξης.....	18
3.1 Το πλαίσιο ανάπτυξης της εφαρμογής.....	18
3.2 Ανάπτυξη της εφαρμογής.....	18
3.3 Τεχνολογίες.....	18
3.3.1 Βάσεις δεδομένων.....	18
3.3.2 Html.....	19
3.3.3 Css.....	19
3.3.4 Php.....	20
3.3.5 Codeigniter.....	22
3.3.6 Javascript.....	23
3.3.7 JQuery.....	23
3.3.8 Plugin.....	23
3.3.9 jQueryCyclePlugin	24
3.3.10 Xml-Rpc.....	24
3.3.11 Web services.....	27
Κεφάλαιο 4 – Αρχιτεκτονική Εργασίας.....	28
4.1 Σύνοψη παρουσίαση της RUP.....	28
4.2 Διαγράμματα UML.....	28
4.3.Ανάλυση και σχεδιασμός.....	29
4.4.Παρουσίαση της εφαρμογής.....	35
Κεφάλαιο 5 – Συμπεράσματα.....	38
5.1 Περιορισμοί.....	38
5.2 Βελτιώσεις	39
Βιβλιογραφία.....	39
Παράρτημα Κώδικα.....	43

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1. Αρχιτεκτονική του συστήματος	16
Σχήμα 2. Σχεδιάγραμμα λειτουργίας PHP.....	20
Σχήμα 3. Σχηματική παρουσίαση του πρότυπου MVC.....	23
Σχήμα 4. Η λειτουργία του xmlhttp.....	26
Σχήμα 5. Η αρχιτεκτονική του Webservice.....	28
Σχήμα 6. Διάγραμμα περιπτώσεων χρήσης της εφαρμογής.....	29
Σχήμα 7. Διάγραμμα περιπτώσεων χρήσης του προγράμ.....	30
Σχήμα 8. Διάγραμμα τάξεων του προγράμματος.....	30
Σχήμα 9. Διάγραμμα αντικειμένων του προγράμματος.....	31
Σχήμα 10. Διάγραμμα συνεργασίας της εφαρμογής.....	32
Σχήμα 11. Διάγραμμα συνεργασίας του προγράμματος.....	32
Σχήμα 12. Διάγραμμα σειράς του προγράμματος.....	33
Σχήμα 13. Διάγραμμα δραστηριοτήτων της εφαρμογής.....	34
Σχήμα 14. Διάγραμμα δραστηριοτήτων του προγράμματος.....	34
Σχήμα 15. Διάγραμμα εξαρτημάτων του προγράμματος.....	35
Σχήμα 16. Αρχική σελίδα της εφαρμογής.....	36
Σχήμα 17. Πλοήγηση του χρήστη με αλλαγή σελίδων.....	36
Σχήμα 18. Προβολή των δεδομένων προς ενοικίαση.....	37
Σχήμα 19. Μετάβαση του χρήστη σε επόμενη παρουσίαση.....	37

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1. Βάσεις Δεδομένων υποστηριζόμενες από την ρηρ.....21

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΓΛΩΣΣΑΡΙ

jQuerycycleplugin: πρόσθετο προβολής διαφανειών

plugincarousel:Πρόσθετο καρουσέλ

Floating: Κυμαινόμενο

Collage: Κολάζ

Diagonal:Διαγώνιος

Mixed:Μικτές

Tile:Πλακάκια

Slideshow:Παρουσίαση

Extremedeployment:Ακραία ανάπτυξη

Webbrowser:Πρόγραμμα περιήγησης στον ιστό

Attributes:Χαρακτηριστικά

Cache:προσωρινή μνήμη

Webdevelopers:Προγραμματιστές του ιστού

Framework:Πλαίσιο

Request:Αίτημα

Webservices:Υπηρεσίες Ιστού:

Suppliers:προμηθευτές

Users:Χρήστες

Managers:Διευθυντές

World Wide Web – WWW:Παγκόσμιος ιστός

1. ΕΙΣΑΓΩΓΗ

1.1 Σύντομη αναφορά στην ιστορία του διαδικτύου

Στις μέρες μας ο όρος πληροφορική είναι πλέον ευρέως γνωστός, καθώς αποτελεί αναπόσπαστο κομμάτι της καθημερινότητας. Μια πληθώρα ενεργειών και λειτουργιών πλέον πραγματοποιείται μέσω ηλεκτρονικών υπολογιστών, κινητών τηλεφώνων κ.α. Η πληροφορική και κατ' επέκταση η «πληροφορία» έχει εισέλθει σε κάθε σπίτι.

Πολύ σημαντικές είναι οι εξελίξεις στα δίκτυα των υπολογιστών και ιδιαίτερα η αλματώδης ανάπτυξη του Διαδικτύου (Internet) που επέφερε αλλαγές στους τρόπους συλλογής, επεξεργασίας και διανομής της πληροφορίας. Μία από τις εφαρμογές του, ο Παγκόσμιος Ιστός (WorldWide Web - WWW) είχε ως αποτέλεσμα να το κάνει εξαιρετικά δημοφιλές με αποτέλεσμα σήμερα οι χρήστες του να ανέρχονται σε δισεκατομμύρια και να αυξάνονται συνεχώς.

Το κυριότερο που προσφέρει ο Ιστός είναι η πρόσβαση σε απομακρυσμένη πληροφορία κάθε είδους με έναν εύχρηστο τρόπο για το χρήστη, ενώ συγχρόνως αποτελεί μια πολύ σημαντική (αν όχι τη σημαντικότερη) πηγή δεδομένων για κάθε τομέα της ανθρώπινης δραστηριότητας.

Έτσι, με την ολοένα αυξανόμενη ποσότητα πληροφοριών και την ανάγκη του χρήστη για πρόσβαση σε αυτές τις πληροφορίες χρειάζονται εύχρηστοι και γρήγοροι τρόποι εύρεσης και ανάκτησης τους. Οι Βάσεις Δεδομένων αποτελούν έναν τέτοιο τρόπο, καθώς τα δεδομένα βρίσκονται αποθηκευμένα και κατηγοριοποιημένα, με αποτέλεσμα την γρήγορη εύρεση και ανάκτησή τους. (ΕΙΣΑΓΩΓΗ ΣΤΟ INTERNET, Δεκέμβριος 1997, ΚΕ.Δ. - Πανεπιστήμιο Θεσσαλίας)

Μια βασική ανάγκη είναι η εύρεση κατοικίας και η εύκολη πρόσβαση σε αυτά τα δεδομένα είναι αναγκαία.

Ένα μεσιτικό γραφείο αποτελεί μια γρήγορη και πρακτική λύση για τον πελάτη που ενδιαφέρεται για ακίνητα αλλά και για τον ιδιοκτήτη, καθώς περιέχει ποικίλες πληροφορίες σχετικά με τα ακίνητα. Ένα μεσιτικό γραφείο σε ηλεκτρονική μορφή παρέχει και όλα τα πλεονεκτήματα της τεχνολογίας. Η τεχνολογία παρέχει τη δυνατότητα αποθήκευσης μεγάλου όγκου δεδομένων και τη διατήρησή τους σε μια βάση. Με αποτέλεσμα, από την πλευρά του πελάτη, να παρέχει την γρήγορη και εύκολη πρόσβαση, σε μια πληθώρα δεδομένων, ανάλογα με τις ανάγκες του. Επιπρόσθετα τα δεδομένα όντας σε ηλεκτρονική μορφή μπορούν να παρουσιάζουν λεπτομέρειες, όπως εικόνες των ακινήτων, ακριβή τοποθεσία, τετραγωνικά μέτρα κλπ.

1.2 Ανάλυση θέματος εργασίας

Η παρούσα μεταπτυχιακή διατριβή περιγράφει την ανάλυση, το σχεδιασμό και την υλοποίηση μιας εφαρμογής η οποία αναλαμβάνει την παρουσίαση των δεδομένων από μια κτηματομεσιτική διαδικτυακή πύλη. Η ανάκτηση των δεδομένων γίνεται με τη χρήση διαδικτυακών υπηρεσιών (webservices). Η παρουσίαση των δεδομένων γίνεται με τη χρήση του JQuerycycleplugin, από μια διαδικτυακή βάση. Βασικός στόχος είναι η εμφάνιση αποτελεσμάτων της αναζήτησης σε πραγματικό χρόνο.

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

Η ανάπτυξη της εφαρμογής έγινε με το codeigniter και το JQuerycycleplugin. Τα διαγράμματα έγιναν με το IBM Rational Rose Enterprise Edition. Η ιστοσελίδα δημιουργήθηκε με html, Php και css.

1.3 Δομή κεφαλαίων

Η δομή έχει την ακόλουθη μορφή:

Στο 1^ο κεφάλαιο περιγράφεται μια συνοπτική αναφορά της εργασίας και η δομή των κεφαλαίων.

Στο 2^ο κεφάλαιο περιγράφονται διάφορες εργασίες που είναι σχετίζονται θεματικά με την παρούσα.

Στο 3^ο κεφάλαιο περιγράφεται το κίνητρο της εργασίας αυτής, πού βασίστηκε και τι θα εξυπηρετεί η εφαρμογή που θα αναπτυχθεί καθώς και με ποια μεθοδολογία και με ποιες τεχνολογίες θα γίνει η ανάπτυξή της.

Στο 4^ο κεφάλαιο γίνεται η ανάλυση απαιτήσεων και περιγράφονται οι περιπτώσεις χρήσεις για την εφαρμογή καθώς και το εγχειρίδιο χρήσης, εγκατάστασης και συστήματος.

Στο 5^ο κεφάλαιο γίνεται μία ανασκόπηση των όσων έγιναν στην εργασία και αναφέρονται τρόποι βελτίωσης και επέκτασης της εφαρμογής.

2. ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΡΕΥΝΑ

Στο κεφάλαιο αυτό γίνεται μια αναφορά εργασιών οι οποίες χρησιμοποιούν τεχνολογίες παρόμοιες με αυτές της παρούσας μεταπτυχιακής διατριβής. Οι κυριότερες τεχνολογίες που αναφέρονται είναι οι XML-RPC και το `plygincarousel`, τα οποία αποτελούν δύο από τις βασικές λειτουργίες της μεταπτυχιακής διατριβής.

Η εργασία των Cooper, deBruijn, Spence και Witkowski (2006) των πανεπιστημίων ImperialCollege, London και UniversityofManchester του Ηνωμένου Βασιλείου σχετίζεται με τον τρόπο παρουσίασης μιας συλλογής εικόνων.

Αναλύονται διάφοροι τρόποι κινούμενης και στατικής παρουσίασης και συγκρίνονται με βάση ένα συγκεκριμένο κριτήριο, την εύρεση μιας εικόνας που έχει αποφασιστεί από πριν. Μέσα στους τρόπους κινούμενης παρουσίασης αναφέρεται και η προβολή σε καρουσέλ.

Με αυτήν την εργασία γίνεται αντιληπτό το πόσο σημαντικός είναι ο τρόπος παρουσίασης ενός δεδομένου έτσι ώστε να έχει το επιθυμητό αποτέλεσμα στον χρήστη.

Η παρούσα μεταπτυχιακή διατριβή όσον αφορά το χρήστη βασίζεται κυρίως σε αυτό το κομμάτι. Καθώς το περιβάλλον διεπαφής του χρήστη αποτελείται από μια παρουσίαση εικόνων με σκοπό την εύρεση μιας «εικόνας-στόχου».

Οι επόμενες εργασίες που αναλύονται σχετίζονται κατά βάση με την τεχνολογία XML-RPC, η οποία αποτελεί μια κλήση απομακρυσμένης διαδικασίας.

Αναλυτικότερα, η εργασία του Allman (2003) του NASA Glenn Research Center / BBN Technologies παρουσιάζει μια έρευνα στην οποία γίνεται σύγκριση ανάμεσα στο XML-RPC και στο `java.net package`. Αναλύονται οι τρόποι σύνταξης της κάθε τεχνολογίας, καθώς και τα πλεονεκτήματα και τα μειονεκτήματα του κάθε κώδικα. Η ανάλυση των τεχνολογιών αυτών μας παρέχει πληροφορίες για τη δομή και τον τρόπο σύνταξης μιας κλήσης απομακρυσμένης διαδικασίας.

Η εργασία των Duarte, Santos, Fernandes, Neto, Pereira και Varandas (2008) από το Κέντρο Πυρηνικής Σύντηξης στη Λισαβόνα, αναλύει τον τρόπο τμηματοποίησης των συστημάτων των πυρηνικών αντιδραστήρων με σκοπό τις διεργασίες σε συγκεκριμένα σημεία του συστήματος. Ο τρόπος που επιτυγχάνεται αυτό είναι με την κλήση απομακρυσμένης διαδικασίας, XML-RPC η οποία επιτρέπει την διέλευση και την ανάκτηση διαφορετικών τύπων δεδομένων. Με αυτή την εργασία, βλέπουμε την σημαντική συνεισφορά της XML-RPC σε τομείς που απαιτούν ακρίβεια και αποφυγή λαθών.

Η εργασία των Larsson και Sprague (2004) αναφέρεται σε ένα σύστημα που δημιούργησε το πανεπιστήμιο Drew με σκοπό την εύκολη πρόσβαση των υπολογιστών των φοιτητών στο δίκτυο του πανεπιστημίου. Και σε αυτήν την εργασία η XML-RPC επιλέχθηκε ως ένα πρωτόκολλο πελάτη / εξυπηρετητή, διότι οι XMLRPC βιβλιοθήκες είναι διαθέσιμες για μια ποικιλία γλωσσών.

Και τέλος, η εργασία των Ardagna, Damiani, DeCapitanidi Vimercati, και Samarati (2006) από το Τμήμα Τεχνολογιών Πληροφορικής του Πανεπιστημίου του Μιλάνου αναφέρεται στην ασφαλή πρόσβαση σε υπηρεσίες. Η εφαρμογή αυτή είναι σημαντική στις μέρες μας, λόγω του μεγάλου όγκου πληροφοριών που βρίσκονται στο διαδίκτυο. Και πάλι η XML έρχεται να παίξει ένα σημαντικό ρόλο σε αυτόν τον τομέα. Έχουν αναπτυχθεί διάφορες γλώσσες ασφαλείας που βασίζονται στην xml, όπως η Security Assertion Markup Language (SAML). Η γλώσσα XACML έχει σχεδιαστεί για να εκφράσει τις πολιτικές εξουσιοδότησης σε XML.

Μια σύγκριση μεταξύ στατικής και κινούμενης παρουσίασης μιας συλλογής εικόνων

Η συγκεκριμένη εργασία αναπτύχθηκε από τους Cooper, deBruijn, Spence και Witkowski (2006) των πανεπιστημίων Imperial College, London και University of Manchester του Ηνωμένου Βασιλείου.

Σκοπός της εργασίας τους είναι η παρουσίαση, η σύγκριση και η εύρεση του αποδοτικότερου τρόπου προβολής εικόνων με σκοπό την εύρεση μιας εικόνας-στόχου. Στις μέρες μας πολλά προσωπικά αλλά και επαγγελματικά θέματα παρουσιάζονται μέσα από συλλογές εικόνων. Ένα κύριο χαρακτηριστικό είναι ο χρόνος εύρεσης μιας εικόνας-στόχου. Εικόνα-στόχος ορίζεται μια εικόνα που αποτελεί το ζητούμενο μιας αναζήτησης.

Με το πέρασμα του χρόνου, δημιουργήθηκαν νέοι τρόποι παρουσίασης με διαφορετική λογική και χρόνο παρουσίασης, όπως το καρουσέλ, το κολάζ (Collage) κ.α. Τα διάφορα μοντέλα παρουσίασης συγκρίνονται για το ποιο είναι αποδοτικότερο στην εύρεση μια εικόνας-στόχου, σύμφωνα με την περιοχή αναζήτησης και το χρόνο παρουσίασης της συλλογής.

Οι τρόποι παρουσίασης που συγκρίνονται όσον αφορά την κινούμενη παρουσίαση είναι το κυμαινόμενο (floating), το κολάζ (collage), το καρουσέλ (carousel) και ο διαγώνιος (diagonal). Ο σκοπός του πειράματος ήταν η παρακολούθηση κάποιων κινούμενων εικόνων από τα άτομα, τα οποία πρέπει στη συνέχεια να αναφέρουν αν μέσα στις εικόνες είδαν την εικόνα-στόχο. Η έρευνα αποκάλυψε ότι οι συμμετέχοντες ήταν σε θέση να εστιάσουν και να παρακολουθήσουν όλες τις εικόνες έως ότου εντοπιστεί ο στόχος.

Σε ένα δεύτερο πείραμα παρουσιάστηκαν στατικές εικόνες με τη μορφή μικτό (mixed), πλακίδια (tile) και συρόμενη εμφάνιση (slide-show). Ζητήθηκε από 30 συμμετέχοντες, στους οποίους έδειξαν για πρώτη φορά μια εικόνα-στόχο, μετά την παρουσίαση, να πουν αν η εικόνα στόχος ήταν παρούσα ή απύουσα. Τους ζητήθηκε επίσης να βαθμολογήσουν τη δυσκολία που αντιμετώπισαν.

Στις συγκρίσεις που έγιναν παρουσιάζονται αποτελέσματα παλιότερων και νέων πειραμάτων. Οι διαφορές στα αποτελέσματα μεταξύ στατικών και κινούμενων παρουσιάσεων εικόνων που αναλύθηκαν, έδειξαν ότι οι στατικές συλλογές έχουν μεγαλύτερο ποσοστό εύρεσης της εικόνας-στόχου, σε σχέση με τις κινούμενες συλλογές. Το συμπέρασμα που βγαίνει από την παραπάνω έρευνα είναι ότι οι στατικές συλλογές είναι προτιμότερες από τις κινούμενες, όσον αφορά την επιτυχία αναγνώρισης της εικόνας αλλά και την προτίμηση των χρηστών.

Στην εργασία αυτή παρατηρούμε ότι ο τρόπος παρουσίασης μιας συλλογής εικόνων αποτελεί ενδιαφέρον πεδίο έρευνας. Αυτό είναι αντιληπτό, καθώς ο τομέας της πληροφορικής βασίζεται κατά κόρον στην προβολή των πληροφοριών μέσω εικόνων. Μερικά παραδείγματα που μας κάνουν να αντιληφθούμε την σημασία της εικόνας στην πληροφορική αναλύονται παρακάτω.

Οι μηχανές αναζήτησης παρέχουν την επιλογή αναζήτησης μόνο εικόνας, για παράδειγμα η αναζήτηση μέσω Google. Αυτό αποδεικνύει πόσο βασικές είναι οι εικόνες αλλά και πόσο εξυπηρετούν το χρήστη στην όποια αναζήτησή του. Τα διάφορα μέσα κοινωνικής δικτύωσης βασίζονται κυρίως στην προβολή εικόνας και υποστηρίζουν ειδικές ρυθμίσεις στην προβολή εικόνων διαφημίσεων.

Όλα τα παραπάνω παρουσιάζουν ότι ο τρόπος προβολής τους και τα αποτελέσματα που έχει στους χρήστες είναι ένα κομμάτι που αξίζει να αναλυθεί.

Μια αξιολόγηση της XML-RPC *

Ο Mark Allman (2003) του NASA Glenn Research Center / BBN Technologies παρουσιάζει μια έρευνα στην οποία γίνεται σύγκριση ανάμεσα στο XML-RPC και στο java.net package. Το xml-rpc είναι ένα πρωτόκολλο Κλήσης Απομακρυσμένης Διαδικασίας που λειτουργεί μέσω του διαδικτύου. Ένα μήνυμα xml-rpc είναι ένα αίτημα HTTP-POST. Στη συγκεκριμένη εργασία παρουσιάζονται οι διαφορές στη σύνταξη του κώδικα καθώς και τα πλεονεκτήματα και τα μειονεκτήματα της κάθε μεθόδου.

Στην περίπτωση του `java.net code`, χρησιμοποιούνται 3 διαφορετικές κλάσεις. Μια για τον πελάτη, μια για τον διακομιστή και μια τρίτη για τις διάφορες ρουτίνες δικτύωσης που χρησιμοποιούν και ο πελάτης και ο διακομιστής.

Στην περίπτωση του XML-RPC, το τμήμα δεδομένων είναι ένα XML αίτημα που τρέχει μια ιδιαίτερη μέθοδο με τις δεδομένες παραμέτρους ή μια XML απόκριση που κωδικοποιεί τα αποτελέσματα της απομακρυσμένης μεθόδου κλήσης.

Το πρώτο πράγμα που παρατηρούμε είναι ότι ο κώδικας `java.net` αντιπροσωπεύει περισσότερη δουλειά για τον σχεδιαστή από την εφαρμογή του κώδικα XML-RPC επειδή ο προγραμματιστής πρέπει να υλοποιήσει όλες τις λεπτομέρειες των πράξεων.

Από την άλλη πλευρά, ο κώδικας XML-RPC είναι σύντομος και εύκολος να κατανοηθεί, ακόμη και από κάποιον που δεν έχει μελετήσει την XML-RPC. Επίσης αυτό που παρατηρούμε είναι ότι και στις δυο περιπτώσεις που εξετάστηκαν εμφανίζεται μια συγκεκριμένη μέθοδος, η `LogTransform()`.

Ένας άλλος τρόπος σύγκρισης των δύο εκδόσεων, είναι η μέτρηση των γραμμών του κώδικα. Τα συμπεράσματα που προκύπτουν είναι ότι η εφαρμογή `java.net` είναι περίπου 50% πιο σύνθετη στην κωδικοποίηση και τη διατήρηση σε σύγκριση με τον κώδικα XMLRPC σύμφωνα με μια ποικιλία μετρήσεων.

Με βάση τις συγκρίσεις προκύπτει ότι η XML-RPC παρέχει τη δυνατότητα στον προγραμματιστή να κατανοήσει καλύτερα τον κώδικα αλλά και να είναι πιο «τακτοποιημένος». Επίσης ο χρόνος που απαιτείται για τις μικρές συναλλαγές είναι περίπου ο ίδιος ανεξάρτητα από το υποκείμενο σύστημα επικοινωνίας τόσο στο τοπικό δίκτυο όσο και στο διαδίκτυο (Internet). Ο κωδικός `java.net` εκτελείται μέχρι μίας τάξης μεγέθους καλύτερα από ό,τι ο XML-RPC κώδικας, όταν μεταφέρεται ένα σημαντικό ποσό των δεδομένων. Μια άλλη παρατήρηση είναι ότι ο XML-RPC κώδικας παρ' όλες τις συμπίεσεις που υφίσταται παραμένει μεγαλύτερος από το πακέτο `java.net`.

Όσο οι τεχνολογίες αναπτύσσονται, η ανάγκη εύρεσης νέων τρόπων επικοινωνίας των συστημάτων κρίνεται επιτακτική. Η ροή της πληροφορίας όντας αμφίδρομη χρήζει αποδοτικούς και γρήγορους τρόπους μετάβασης. Η χρήση του μοντέλου διακομιστής- πελάτης και οι κλήσεις απομακρυσμένης διαδικασίας είναι απαραίτητα όταν τα δεδομένα που διαχειριζόμαστε βρίσκονται σε διαφορετικό `server` αλλά και πιθανόν σε διαφορετική μορφή.

Με αυτό τον τρόπο η ανταλλαγή πληροφοριών γίνεται εύκολα και γρήγορα σε πραγματικό χρόνο. Τα δεδομένα βρίσκονται αποθηκευμένα σε μια βάση. Ο χρήστης κάνει αναζήτηση των δεδομένων που θέλει μέσα από το σύστημα του και επιθυμεί να επιστραφούν αποτελέσματα με βάση την αναζήτησή του.

Αυτό επιτυγχάνεται μέσω των κλήσεων απομακρυσμένης διαδικασίας, καθώς ο πελάτης στέλνει ένα αίτημα με τα κριτήρια της αναζήτησης του χρήστη και ο διακομιστής ύστερα από ελέγχους επιστρέφει τα αποτελέσματα που ταιριάζουν με τα κριτήρια. Με την διαδικασία XML-RPC αυτό είναι εφικτό και σχετικά προσβάσιμο από μη επαγγελματίες προγραμματιστές με αποτέλεσμα η χρήση τέτοιων συστημάτων να αποτελεί μια πολύ έξυπνη λύση.

FireCalc: Ένα πλαίσιο βασισμένο στην XML για την κατανεμημένη ανάλυση των δεδομένων

Η συγκεκριμένη εργασία από τους Duarte, Santos, Fernandes, Neto, Pereira και Varandas (2008) από το Κέντρο Πυρηνικής Σύντηξης στη Λισαβόνα,

αναλύει τον τρόπο τμηματοποίησης των συστημάτων των πυρηνικών αντιδραστήρων με σκοπό τις διεργασίες σε συγκεκριμένα σημεία του συστήματος.

Η συνεχής τεχνολογική εξέλιξη καθιστά δύσκολο να προβλεφθεί ποια θα είναι η καλύτερη διαθέσιμη λύση κατά τη διάρκεια ζωής μιας μηχανής και ειδικά των αντιδραστήρων.

Τα αρθρωτά συστήματα επιτρέπουν να γίνονται αλλαγές σε κάποια τμήματα του συστήματος, χωρίς να επηρεάζεται όλο το σύστημα.

Η χρήση τεχνολογιών ανοιχτού κώδικα αποτελεί μια καλή λύση σε αυτό το πρόβλημα.

Με βάση τα παραπάνω αναπτύχθηκε ένα πλαίσιο που ονομάζεται FireCalc. Το πλαίσιο αυτό έπρεπε να είναι γραμμένο σε μια γλώσσα που να υποστηρίζει την τμηματοποίηση του συστήματος. Η δομή της γλώσσας XML είναι αρκετά ευέλικτη για να επιτρέψει ανεξάρτητες

λύσεις στην πρόσβαση δεδομένων. Η XML είναι μια τυποποιημένη μορφή κειμένου που χρησιμοποιείται ευρέως για την ανταλλαγή δεδομένων.

Το αρχείο περιγραφής χωρίζεται σε δύο μέρη, την κεφαλίδα και τον κώδικα. Η κεφαλίδα περιέχει πληροφορίες σχετικά με τον διακομιστή FireCalc με τον οποίο θα συνδεθεί, και τα δεδομένα εισόδου και εξόδου. Η XML έχει σχεδιαστεί για να υποστηρίζει πρωτόκολλα δεδομένων απομακρυσμένης πρόσβασης.

Στο συγκεκριμένο παράδειγμα ο πελάτης και ο διακομιστής έχουν προγραμματιστεί στη γλώσσα Java. Οι κλήσεις απομακρυσμένης διαδικασίας επιτρέπουν την εκτέλεση των λειτουργιών σε έναν απομακρυσμένο διακομιστή και τη λήψη δεδομένων από αυτόν, χάρη στην κλήση απομακρυσμένης διαδικασίας, XML-RPC η οποία επιτρέπει την διέλευση και την ανάκτηση διαφορετικών τύπων δεδομένων.

Η εφαρμογή ήταν απαραίτητο να ακολουθήσει αυτή τη δομή:

- Πελάτης (Client): Υπεύθυνος για τη συλλογή τοπικών δεδομένων, καθώς και την αποστολή με την περιγραφή XML στο διακομιστή και την αποθήκευση των αποτελεσμάτων σε τοπικό επίπεδο.
- Διακομιστής (Server): Ανακτά δεδομένα από το απομακρυσμένο διακομιστή δεδομένων και καλεί τον πυρήνα ανάλυσης του λογισμικού για τους ζητούμενους υπολογισμούς.
- Διακομιστής Δεδομένων (Data server): Αυτό το επίπεδο είναι υπεύθυνο για την πρόσβαση στην απομακρυσμένη βάση δεδομένων. Εάν απαιτείται και άδεια, μπορεί να σώσει τα αποτελέσματα εξ αποστάσεως.
- Ασφάλεια: πιστοποιεί τους χρήστες στο σύστημα και διαχειρίζεται τα προνόμιά τους για πρόσβαση σε δεδομένα, εκτελεί τον κώδικα και αποθηκεύει τα αποτελέσματα εξ αποστάσεως.

Η εφαρμογή FireCalc θα αποστείλει το αρχείο σε έναν απομακρυσμένο διακομιστή χρησιμοποιώντας ένα πυρήνα Scilab. Οι διάφορες μεταβλητές εισόδου ορίζονται στο <remoteData> και η μεταβλητή εξόδου σε <outputData>. Ο κύριος Scilab κώδικας για τον υπολογισμό zeff είναι στη <mainCode> του XMLelement, μέρος του οποίου είχε παραλειφθεί, διότι θα ήταν υπερβολικά μεγάλος.

Η χρήση της τεχνολογίας xmlhttp κάνει το FireCalc ένα πολύ ευέλικτο πλαίσιο για την ανάπτυξη μιας σπονδυλωτής ανάλυσης δεδομένων και των υπολογιστικών συστημάτων. Το γεγονός ότι η xmlhttp επιτρέπει διαφορετικές γλώσσες, καθιστά δυνατόν να συγκριθούν τα αποτελέσματα από διαφορετικά σενάρια για το ίδιο το περιβάλλον. Το πλαίσιο FireCalc επιτρέπει στους χρήστες να δρομολογήσουν εύκολα τις εργασίες υπολογισμού δεδομένων στον τοπικό υπολογιστή τους καθώς και σε απομακρυσμένα μηχανήματα ή δίκτυα.

Η παραπάνω εργασία αναφέρεται στην ανάγκη τμηματοποίησης του κώδικα που διέπει τα συστήματα των πυρηνικών αντιδραστήρων. Τέτοια συστήματα είναι απαραίτητο να είναι ακριβή και να αποτρέπουν την ύπαρξη λαθών στον κώδικα καθώς οποιοδήποτε λάθος μπορεί να αποβεί μοιραίο. Η χρήση λοιπόν ενός προγράμματος που θα χρησιμοποιεί τον κώδικα τμηματικά αποτελεί μια αποδοτική λύση σε τέτοιου είδους συστήματα.

Με αυτόν τον τρόπο καθίσταται ευκολότερη η εύρεση αλλά και η διόρθωση λαθών καθώς δεν είναι απαραίτητη η αλλαγή όλου του συστήματος αλλά μόνο η διόρθωση του τμήματος που αφορά το λάθος. Αυτό έχει σαν αποτέλεσμα τη γρήγορη και εύκολη χρήση του συστήματος.

Με βάση τα παραπάνω γίνεται αντιληπτό πως η χρήση του συστήματος FireCalc και του xmlhttp που τμηματοποιούν τον κώδικα είναι πολύ σημαντική. Οι κλήσεις απομακρυσμένης διαδικασίας επιτρέπουν την εκτέλεση των λειτουργιών σε έναν απομακρυσμένο διακομιστή και τη λήψη δεδομένων από αυτόν χάρη στην XMLRPC η οποία επιτρέπει την διέλευση και την ανάκτηση διαφορετικών τύπων δεδομένων.

Η χρήση της XMLRPC και η τμηματοποίηση του κώδικα βοηθάει στην εύκολη διόρθωση, επεξεργασία και αλλαγή ενός ή πολλών τμημάτων του, χωρίς να επηρεάζεται ολόκληρο το σύστημα. Αυτό έχει σαν αποτέλεσμα το σύστημα να είναι πιο αποδοτικό και εύχρηστο.

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

Η ακόλουθη εργασία από τους Larsson και Sprague (2004) αναφέρεται σε ένα σύστημα που δημιούργησε το πανεπιστήμιο Drew με σκοπό την εύκολη πρόσβαση των υπολογιστών των φοιτητών στο δίκτυο του πανεπιστημίου.

Το πανεπιστήμιο Drew δημιούργησε μια τεχνολογία, την eXtremeDeployment, η οποία είναι προ-εγκατεστημένη στους φορητούς υπολογιστές από τον προμηθευτή. Είναι χτισμένη σε τεχνολογίες ανοιχτού κώδικα και προσδιορίζει τους υπολογιστές από τις ενεργές ετικέτες BIOS. Μέσω αυτής της διαδικασίας το πανεπιστήμιο Drew είναι σε θέση να παρέχει άμεση εξατομίκευση των φοιτητικών υπολογιστών χωρίς να τους προδιαθέσει σε συγκεκριμένους φοιτητές. Η διανομή ηλεκτρονικών υπολογιστών με τη χρήση της eXtremeDeployment ήταν μια επιτυχία, με την εγγραφή πάνω από 450 υπολογιστών στον τομέα του πανεπιστημίου σε πέντε ώρες.

Το πανεπιστήμιο Drew ήταν από τα πρώτα που εισήγαγε τους φορητούς υπολογιστές και τα email σε πανεπιστήμιο. Εφαρμόζοντας το NovelleDirectory μέσω του οποίου φοιτητές, διδάσκοντες και προσωπικό έχουν μια μόνο σύνδεση που παρέχει πρόσβαση σε όλους τους δικτυωμένους πόρους. Κατά τη διάρκεια εφαρμογής αυτού του προγράμματος παρουσιάστηκαν κάποια προβλήματα, όπως η δυσκολία απομνημόνευσης των κωδικών σύνδεσης στο σύστημα αλλά και το μπλοκάρισμα. Μετά από διάφορες λύσεις που αποδείχθηκαν μη αποτελεσματικές, αποφασίστηκε ότι ήταν αναγκαίο να δημιουργηθεί ένα δικό τους σύστημα. Το eXtremeDeployment που δημιουργήθηκε είναι χτισμένο πάνω σε αρκετές τεχνολογίες ανοιχτού κώδικα αλλά και σε εμπορικές.

Ο διακομιστής εγκατάστασης είναι υπεύθυνος για την παροχή πρόσβασης στη βάση δεδομένων. Οι πελάτες επικοινωνούν με το διακομιστή στέλνοντας Xml-rpc μηνύματα μέσω μιας ασφαλούς Http σύνδεσης. Η XML-RPC επιλέχθηκε ως ένα πρωτόκολλο πελάτη / εξυπηρετητή, διότι οι XML-RPC βιβλιοθήκες είναι διαθέσιμες για μια ποικιλία γλωσσών. Επίσης επιλέχθηκε επειδή έχει ένα πλεονέκτημα απλότητας πάνω από ανταγωνιστικές υπηρεσίες ιστού όπως το πρωτόκολλο SOAP. Ο διακομιστής XML-RPC εφαρμόζεται στην γλώσσα προγραμματισμού Php και τρέχει στον διακομιστή ιστού Apache. Η γλώσσα Php κρίθηκε κατάλληλη για την υλοποίηση του έργου αυτού, γιατί έδωσε την δυνατότητα αξιοποίησης των υπαρχόντων διαδικασιών ανάπτυξης. Επίσης έγινε ανάπτυξη της βάσης δεδομένων σε Microsoft Sql Server. Η πρόσβαση στη βάση δεδομένων από το Διακομιστή ανάπτυξης PHPscripts επιτεύχθηκε μέσω ODBC. Το eXtremeDeployment βασίζεται στην γλώσσα html.

Μια αρχιτεκτονική για υπηρεσίες ιστού για την εφαρμογή των πολιτικών ελέγχου πρόσβασης

Εργασία

αυτή αναφέρεται στην ασφαλή πρόσβαση σε υπηρεσίες ιστού από τους Agostino Ardagna, Damiani, DeCapitanidi Vimercati, και Samarati (2006) από το Τμήμα Τεχνολογιών Πληροφορικής του Πανεπιστημίου του Μιλάνο.

Οι υπηρεσίες ιστού αντιπροσωπεύουν μια πρόκληση και μια ευκαιρία για τους οργανισμούς που επιθυμούν να εκθέσουν προϊόντα και προσφορές υπηρεσιών μέσω του Διαδικτύου. Ωστόσο, κάθε οργανισμός που χρησιμοποιεί XML και υπηρεσίες διαδικτύου πρέπει να εξασφαλίζει ότι μόνο οι σωστοί χρήστες που στέλνουν το κατάλληλο περιεχόμενο XML μπορούν να έχουν πρόσβαση στις Υπηρεσίες Ιστού τους. Η εργασία αυτή είναι μια προσπάθεια να κατανοήσουμε τις βασικές έννοιες για την εξασφάλιση των υπηρεσιών ιστού και τις απαιτήσεις για την εφαρμογή ασφαλών υπηρεσιών ιστού.

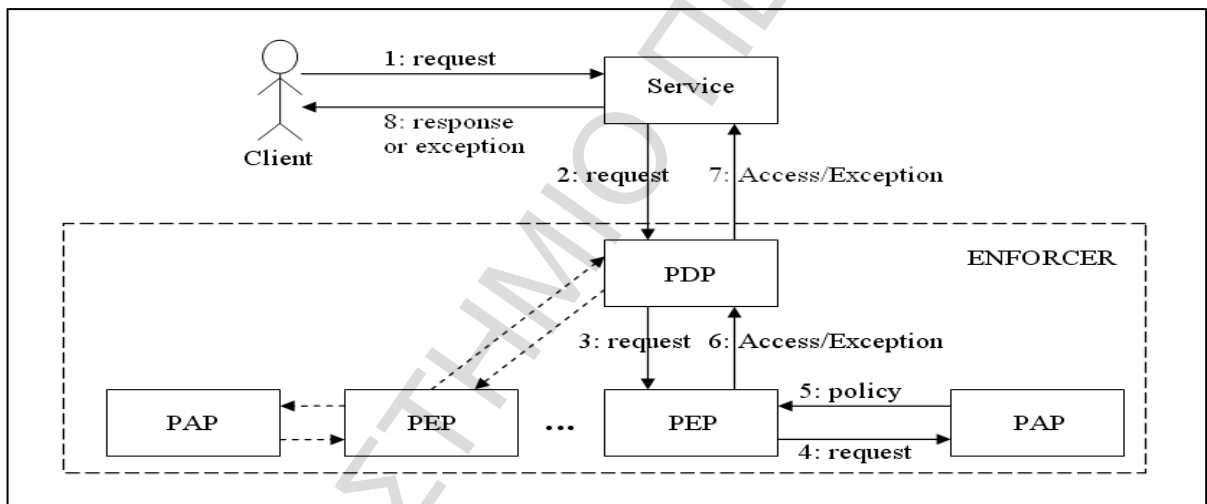
Η πρόσβαση στις πληροφορίες του διαδικτύου έχει γίνει μια βασική απαίτηση της σύγχρονης οικονομίας. Οι υπηρεσίες του ιστού που έχουν αναπτυχθεί γι' αυτό το σκοπό επιτρέπουν την ανταλλαγή πληροφοριών μέσω του δικτύου βασιζόμενες κυρίως σε τέσσερα πρότυπα : την eXtensible Markup Language (XML) , το Simple Object Access Protocol (SOAP) , το Web Services Description Language (WSDL), και το Universal Discovery, Description and Integration (UDDI). Παρόλο που η XML χρησιμοποιείται ευρέως, υπάρχουν δύο βασικά θέματα που πρέπει να αντιμετωπιστούν: αν περιορίζεται η πρόσβαση σε μια υπηρεσία XML σε εξουσιοδοτημένους χρήστες, και δεύτερον αν προστατεύεται η ακεραιότητα και το απόρρητο των μηνυμάτων XML που ανταλλάσσονται σε ένα περιβάλλον υπηρεσιών ιστού.

Πρόσφατα έχουν αναπτυχθεί διάφορες γλώσσες ασφαλείας που βασίζονται στην xml, όπως η SecurityAssertionMarkupLanguage (SAML) που έχει σχεδιαστεί για να επιτρέπει την ασφαλή μεμονωμένη είσοδο χρήση, για εφαρμογές μέσα στους οργανισμούς και στις εταιρίες και υποστηρίζει πολλούς διαφορετικούς μηχανισμούς πιστοποίησης.

Μια άλλη σημαντική πτυχή που πρέπει να ληφθεί υπόψη για την εξασφάλιση των υπηρεσιών Web είναι ο έλεγχος πρόσβασης του οποίου η λύση απαιτεί διερεύνηση των πολιτικών για τον καθορισμό των κανόνων ελέγχου πρόσβασης μαζί με μια γλώσσα για την έκφραση τους.

Για παράδειγμα η γλώσσα XACML έχει σχεδιαστεί για να εκφράσει τις πολιτικές εξουσιοδότησης σε XML. Το πλαίσιο πολιτικής της υπηρεσίας ιστού παρέχει ένα γενικό μοντέλο και μια ευέλικτη και επεκτάσιμη γραμματική για την περιγραφή και την επικοινωνία των πολιτικών από μια υπηρεσία Web. Μια πολιτική είναι μια συλλογή ενός ή περισσότερων ισχυρισμών που αντιπροσωπεύουν μια διαφορετική προτίμηση, απαίτηση, ικανότητα ή άλλες ιδιότητες που πρέπει να πληρούνται για την απόκτηση πρόσβασης.

Η αρχιτεκτονική για την εφαρμογή των πολιτικών ελέγχου πρόσβασης ικανοποιεί τις ακόλουθες απαιτήσεις: Τμηματοποίηση, Πολιτικές ανεξαρτησίας της γλώσσας, Επεκτασιμότητα, Επαναχρησιμοποίηση, Υψηλή απόδοση, Ανεξαρτησία γλώσσας προγραμματισμού. Η αρχιτεκτονική περιέχει έναν «εφαρμοστή» που περιλαμβάνει τον υπολογισμό των δικαιωμάτων πρόσβασης στις επιμέρους υπηρεσίες του ιστού και επιστρέφει για κάθε αίτηση αν παρέχεται πρόσβαση ή όχι.



Σχήμα 1. Αρχιτεκτονική του συστήματος

Στο Σχήμα 1 παρατηρούμε την αρχιτεκτονική του συστήματος. Αρχικά η μονάδα απόφασης πολιτικής PDP (PolicyDecisionPoint) λαμβάνει ένα αίτημα πρόσβασης και επιστρέφει ένα «ναι» ή ένα «όχι».

Η μονάδα αξιολόγησης πολιτικής PEP (PolicyEvaluationPoint) αλληλεπιδρά με την PAP και αξιολογεί το αίτημα σύμφωνα με τις υπάρχουσες πολιτικές και επιστρέφει την τελική απόφαση στην ενότητα PDP.

Η μονάδα διοίκησης πολιτικής PAP (PolicyAdministrationPoint) ξαναδοκιμάζει τις πολιτικές που εφαρμόζονται σε ένα αίτημα και επιστρέφει την πρόσβαση τους στη μονάδα PEP. Στη συνέχεια περιγράφεται πως επεξεργάζονται οι αιτήσεις πρόσβασης που υποβάλλονται από τους πελάτες.

Ένα από τα πιο σημαντικά χαρακτηριστικά αυτής της ενότητας PAP είναι ότι μπορεί να αποτρέψει την εισαγωγή άκυρων πολιτικών και την ενημέρωση εργασιών που δημιουργούν άκυρες πολιτικές. Η μονάδα PEP αντιλαμβάνεται την εφαρμογή των πολιτικών που επέστρεψε η μονάδα PAP. Η αίτηση πρόσβασης χορηγείται, εφόσον τουλάχιστον μία πολιτική ικανοποιείται.

Η μονάδα PDP είναι η διεπαφή μεταξύ των υπηρεσιών και του εξυπηρετητή.

Όταν υπάρχει μια εξαίρεση κατά την αξιολόγηση ενός αιτήματος πρόσβασης που υποβάλλεται από έναν πελάτη, το σύστημα δεν θα πρέπει να καλύπτει αποκλειστικά και μόνο τις εξαιρέσεις, αλλά και να επικοινωνεί για την εξαίρεση με τον πελάτη. Εάν δεν επιτρέπεται η

πρόσβαση στην αίτηση, οδιαχειριστής της αίτησης πρέπει να αναφέρει ένα σφάλμα που υποδεικνύει τον λόγο για τον οποίο έχει απορριφθεί αίτηση πρόσβασης που υποβλήθηκε.

Κλείνοντας παρατηρούμε ότι η ασφάλεια αποτελεί σήμερα μία από τις κύριες ανησυχίες και πολλές πρωτοβουλίες βρίσκονται σε εξέλιξη με στόχο την επίτευξη μιας τυποποιημένης μορφής για την υποστήριξη της ακεραιότητας, της εμπιστευτικότητας, και ελέγχου πρόσβασης για υπηρεσίες XML Web.

Το θέμα που αναλύεται στην παραπάνω εργασία αφορά την πολιτική ελέγχου πρόσβασης σε υπηρεσίες ιστού. Με την εξέλιξη της τεχνολογίας, η ύπαρξη ασφάλειας θεωρείται απαραίτητη. Η χρήση της eXtensible Markup Language (XML) σε μια πληθώρα εφαρμογών του ιστού έδωσε την αφορμή δημιουργίας της γλώσσας XACML που έχει σχεδιαστεί για να εκφράσει τις πολιτικές εξουσιοδότησης σε XML.

Ο τρόπος που λειτουργεί η αρχιτεκτονική, περιέχει έναν «εφαρμοστή» που ελέγχει τα δικαιώματα πρόσβασης στις επιμέρους υπηρεσίες του ιστού και επιστρέφει για κάθε αίτηση αν παρέχεται πρόσβαση ή όχι. Ο τρόπος που λειτουργεί βασίζεται στην δημιουργία αιτημάτων και απαντήσεων από έναν πελάτη σε ένα διακομιστή.

Στο κεφάλαιο αυτό έγινε παρουσίαση διαφόρων εργασιών που χρησιμοποιούν ανάλογες τεχνολογίες με τη μεταπτυχιακή διατριβή.

Οι τεχνολογίες που χρησιμοποιούνται είναι η XML-RPC και το plugin carousel για την παρουσίαση εικόνων.

Με βάση την παραπάνω ανάλυση παρατηρούμε τον διαφορετικό τρόπο με τον οποίο εφαρμόζονται αυτές οι τεχνολογίες και την ποικιλία εφαρμογών τους.

Αυτό μας παρακινεί να αντιληφθούμε το ευρύ φάσμα εφαρμογής αλλά και τρόπους που μπορούν να συμπεριληφθούν στην ίδια τη μεταπτυχιακή διατριβή.

Στη συνέχεια αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν στη μεταπτυχιακή διατριβή.

3. ΜΕΘΟΔΟΛΟΓΙΑ ΑΝΑΠΤΥΞΗΣ

3.1 Το πλαίσιο ανάπτυξης της εφαρμογής

Τα τελευταία χρόνια υπάρχει αυξημένο ενδιαφέρον στον τρόπο παρουσίασης και διάδοσης της πληροφορίας. Με την ανάπτυξη νέων τεχνολογιών, η επίτευξη της παρουσίασης και διάδοσης των πληροφοριών είναι εφικτή με πολλούς και πρωτότυπους τρόπους. Για να είναι επιτυχημένη μια ιστοσελίδα πρέπει να παρέχει στο χρήστη, αρχικά τα δεδομένα τα οποία του είναι χρήσιμα, να υπάρχει ενδιαφέρον τρόπος παρουσίασης τους και τέλος να υπάρχει διάδραση.

Η πληροφορία που βρίσκεται αποθηκευμένη σε βάσεις δεδομένων μπορεί να χρησιμοποιηθεί και να επεξεργαστεί με σκοπό την προβολή της σε διάφορους ιστότοπους. Ο τρόπος παρουσίασης διαφέρει από ιστότοπο σε ιστότοπο, με βάση τα δεδομένα. Η παρουσίαση των δεδομένων μέσω μιας εφαρμογής εναλλαγής δεδομένων, είναι ένας εύχρηστος και ενδιαφέρον τρόπος.

Ο χρήστης έχει τη δυνατότητα να πάρει όλες τις πληροφορίες που του χρειάζονται αλλά και να συμμετέχει στον τρόπο με τον οποίο θα του παρουσιαστούν καθώς με τα κουμπιά πλοήγησης και τις σελίδες, μπορεί να μεταβεί στα δεδομένα που τον ενδιαφέρουν. Επίσης μια εφαρμογή αυτόματης εναλλαγής δεδομένων κρατάει αμείωτο το ενδιαφέρον του χρήστη.

3.2 Ανάπτυξη της εφαρμογής

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν αντικειμενοστρεφείς τεχνολογίες. Υπάρχουν διάφορες μεθοδολογίες και προσεγγίσεις για τον κύκλο ζωής μιας τέτοιας εφαρμογής. Μία μεθοδολογία ορίζει την διαδικασία ανάπτυξης του λογισμικού, στην οποία συμπεριλαμβάνονται τα βήματα, οι πρακτικές και οι τεχνικές που πρέπει να ακολουθηθούν. Στο μοντέλο του καταρράκτη (waterfallmodel) όλη η ανάλυση και ο σχεδιασμός γίνεται πριν την υλοποίηση (συγγραφή του κώδικα. (Ian Sommerville, 2011, Softwareengineering)

Στην Ενοποιημένη Διαδικασία Rational (RationalUnifiedProcess) πρώτα γίνεται η ανάλυση απαιτήσεων και οι δραστηριότητες ανάλυσης και σχεδιασμού για ολόκληρο το σύστημα και στη συνέχεια το σύστημα κατασκευάζεται με επαναληπτικό τρόπο. (Werner Heijstek , 2010, TheRationalUnifiedProcess)

Ακολουθήθηκαν ορισμένες από τις πρακτικές που αυτές προτείνουν. Οι κυριότερες από αυτές είναι:

Επαναδιατύπωση του κώδικα (refactoring) – Αυτή η τεχνική χρησιμοποιείται για την αλλαγή της εσωτερικής δομής του κώδικα, ώστε να είναι πιο εύκολη η κατανόηση και διαχείρισή του.

Διαγράμματα UML- Τα διαγράμματα UML χρησιμοποιούνται σε όλες τις φάσεις του κύκλου ζωής του λογισμικού, από την ανάλυση και τον σχεδιασμό μέχρι την τεκμηρίωση.

Επαναληπτική ανάπτυξη (iterative development) – Επιτρέπει τη διασαφήνιση των απαιτήσεων και την βελτίωση του σχεδιασμού, καθώς και τη γρήγορη ανάδραση από τους χρήστες.

Σχεδιασμός οδηγούμενος από τους χρήστες (userdrivendesign) – Το λογισμικό πρέπει να ικανοποιεί τις συγκεκριμένες απαιτήσεις των χρηστών.

3.3 Τεχνολογίες

3.3.1 Βάσεις δεδομένων

Με τον όρο βάση δεδομένων εννοείται μία συλλογή από συστηματικά οργανωμένα (formatted) σχετιζόμενα δεδομένα. Ένας τηλεφωνικός κατάλογος, για παράδειγμα, θεωρείται βάση δεδομένων, καθώς αποθηκεύει και οργανώνει σχετιζόμενα τμήματα πληροφορίας, όπως είναι το όνομα και ο αριθμός τηλεφώνου.

Ωστόσο, στον κόσμο των υπολογιστών, με τον όρο βάση δεδομένων αναφερόμαστε σε μια συλλογή σχετιζόμενων δεδομένων τμημάτων πληροφορίας ηλεκτρονικά αποθηκευμένων. Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει βάσει του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων της σε προγράμματα ή συλλογές προγραμμάτων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων.

Η ηλεκτρονική βάση δεδομένων χρησιμοποιεί ιδιαίτερου τύπου λογισμικό προκειμένου να οργανώσει την αποθήκευση των δεδομένων της. Το διακριτό αυτό λογισμικό είναι γνωστό ως Σύστημα διαχείρισης βάσης δεδομένων συντομευμένα (DBMS).

(Βάση δεδομένων, 2013, Βικιπαίδεια)

3.3.2 Html

Η HTML είναι το ακρωνύμιο των λέξεων HyperTextMarkupLanguage (γλώσσα μορφοποίησης υπερκειμένου) και είναι η βασική γλώσσα δόμησης σελίδων του WorldWide Web (ή απλά ιστού: Web). Είναι μία γλώσσα προγραμματισμού. Χρησιμοποιείται για να επισημάνει ένα τμήμα κειμένου και να το κάνει να εμφανίζεται καλύτερα. Επιτρέπει την ενσωμάτωση ήχου και εικόνων στις σελίδες του ιστού.

Αρχικά είχε κατασκευασθεί με σκοπό μόνο την μορφοποίηση κειμένου, αλλά μεγάλωσε και ενσωμάτωσε σχεδιαστικές τεχνικές κ.α.

Η γλώσσα χρησιμοποιεί ένα αριθμό από ετικέτες - tags για την μορφοποίηση κειμένου, για την δημιουργία συνδέσμων (links) μετάβασης ανάμεσα στις σελίδες, για την εισαγωγή εικόνων, ήχου κ.α. Όταν ένας Web Browser ανοίγει ένα αρχείο HTML τα στοιχεία (tags) μεταφράζονται σε κατάλληλα χαρακτηριστικά με αποτελέσματα στην εμφάνιση και στην λειτουργικότητα της συγκεκριμένης σελίδας. (Τι είναι η html, 2004, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης)

Στο παρακάτω τμήμα κώδικα δίνεται ένα παράδειγμα html:

```
<table border="0"
style="border-width: 0px;

border-style: none;

border-collapse: separate;
background-color: #E6F6F6;">

<tr>
<td width='30%' valign="top" height="50">
```

Στο παραπάνω παράδειγμα ορίζεται ένας πίνακας με το tag <table> καθώς και η μορφοποίηση του με την εντολή style. Στη συνέχεια ορίζονται τα κελιά και οι γραμμές του πίνακα με τα tags <tr> και <td>.

3.3.3 Css

Το CSS είναι μια απλή γλώσσα που μας βοηθάει να ορίσουμε με σαφήνεια και ιδιαίτερη ευελιξία τον τρόπο με τον οποίο θα εμφανίζονται τα διάφορα στοιχεία στην ιστοσελίδα μας. (CSS, 2012, Βικιπαίδεια)

Τα πλεονεκτήματα που έχει η χρήση CSS έναντι της μορφοποίησης μέσω HTML attributes είναι:

- Πολύ μεγαλύτερη ευελιξία. Το CSS κατέστησε εφικτές μορφοποιήσεις οι οποίες ήταν αδύνατες ή πολύ δύσκολες με την κλασική HTML.
- Ευκολότερη συντήρηση των ιστοσελίδων. Η εμφάνιση ενός ολόκληρου site μπορεί να ελέγχεται από ένα μόνο εξωτερικό αρχείο CSS. Έτσι, κάθε αλλαγή στο στυλ της ιστοσελίδας μπορεί να γίνεται με μια μοναδική αλλαγή σε αυτό το αρχείο, αντί για την επεξεργασία πολλών σημείων σε κάθε σελίδα που υπάρχει στην εφαρμογή.
- Μικρότερο μέγεθος αρχείου, δεδομένου ότι ο κάθε κανόνας μορφοποίησης γράφεται μόνο μια φορά και όχι σε κάθε σημείο που εφαρμόζεται.
- Καλύτερο SEO (Searchengineoptimization). Οι μηχανές αναζήτησης δεν

«μπερδεύονται» ανάμεσα σε περιεχόμενο και τη μορφοποίηση του, αλλά έχουν πρόσβαση στο περιεχόμενο σκέτο, οπότε είναι πολύ ευκολότερο να το καταγράψουν και να το αρχειοθετήσουν (indexing).

- Γρηγορότερες σελίδες. Όταν χρησιμοποιούμε εξωτερικό αρχείο CSS, ο browser την πρώτη φορά που θα φορτώσει κάποια σελίδα της εφαρμογής μας το αποθηκεύει στην cache μνήμη, οπότε δεν χρειάζεται να το κατεβάσει ξανά κάθε φορά που κατεβάζει ο χρήστης του κάποια άλλη σελίδα της εφαρμογής μας.

Στο παρακάτω τμήμα κώδικα δίνεται ένα παράδειγμα css:

```
.main
{
margin:auto;
padding-left:20px;
width:300px;
height: 450px;
}
```

Με τη σύνταξη `.main` ορίζουμε τη μορφοποίηση της κλάσης `main` που είναι ορισμένη στο `php` έγγραφο. Οι εντολές που βρίσκονται στα άγκιστρα `{ }` ορίζουν διάφορες λειτουργίες για την `main`, όπως το πλάτος, το ύψος κλπ.

3.3.4 Php

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων ιστού με δυναμικό περιεχόμενο.

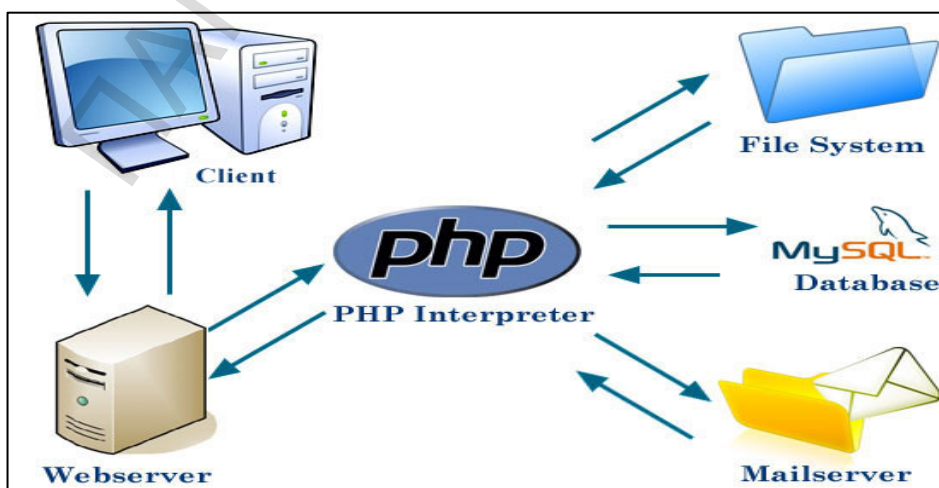
Η PHP, όπου τα αρχικά σημαίνουν Hypertext Pre Processor, είναι μια γλώσσα συγγραφής σεναρίων (scripting language) που ενσωματώνεται μέσα στον κώδικα της HTML και εκτελείται στην πλευρά του server (server-side scripting).

Το μεγαλύτερο μέρος της σύνταξής της, η PHP το έχει δανειστεί από την C, την Java και την Perl και διαθέτει και μερικά δικά της μοναδικά χαρακτηριστικά. Ο σκοπός της γλώσσας είναι να δώσει τη δυνατότητα στους web developers να δημιουργούν δυναμικά παραγόμενες ιστοσελίδες.

Για την επεξεργασία και ανάπτυξη δυναμικών ιστοσελίδων, θα πρέπει να χρησιμοποιήσουν και να κατανοήσουν διάφορες τεχνολογίες. Υπάρχουν τρία βασικά συστατικά για τη δημιουργία δυναμικών ιστοσελίδων:

- Ένας διακομιστής web
- Μια γλώσσα προγραμματισμού από την πλευρά του server
- Μια βάση δεδομένων

Στο παρακάτω σχήμα βλέπουμε τη σχέση των τριών αυτών συστατικών της δυναμικής ιστοσελίδας που αναφέρονται παραπάνω. (PHP & MYSQL TUTORIAL - PREVIEW 2013, Bogotobogo © 2013)



Σχήμα 2. Σχεδιάγραμμα λειτουργίας Php

Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML. Ο κώδικας της PHP περικλείεται με ειδικές ετικέτες-tags αρχής και τέλους για να μπορούμε να εισερχόμαστε και να εξερχόμαστε από την κατάσταση PHP.

(PHP,2013, Βικιπαίδεια)

Ίσως το δυνατότερο και πιο σημαντικό χαρακτηριστικό της PHP είναι η υποστήριξη που παρέχει σε μια ευρεία γκάμα από βάσεις δεδομένων. Έτσι, το να δημιουργήσουμε μια ιστοσελίδα που να παρέχει υποστήριξη σε βάσεις δεδομένων είναι απίστευτα απλό. Υποστηρίζει τις εξής βάσεις δεδομένων :

Adabas D	dBase	Empress	FilePro	Informix	InterBase	mSQL
MySQL	Oracle	PostgreSQL	Solid	Sybase	Velocis	Unix dbm

Πίνακας 1. Βάσεις δεδομένων υποστηριζόμενες από την php

Η ιδέα για την δημιουργία της PHP ελήφθη το φθινόπωρο του 1994 από τον *Rasmus Lerdorf*. Οι πρώτες ανεπίσημες εκδόσεις (versions) της PHP χρησιμοποιήθηκαν στην αρχική του σελίδα (home page) για να μπορεί να παρακολουθεί αυτούς που έμπαιναν στην σελίδα. Η πρώτη έκδοση που δόθηκε για χρήση στο κοινό ήταν διαθέσιμη στις αρχές του 1995 με το όνομα *Personal Home Page Tools*.

Αποτελείτο από μια πολύ απλοϊκή μηχανή ανάλυσης (parser engine) η οποία καταλάβαινε λίγες μόνο ειδικές μακροεντολές (macros) και έναν αριθμό από utilities που βρίσκονταν σε κοινή χρήση στις home pages εκείνη την εποχή. Ένα guestbook, ένας μετρητής (counter) και κάποιο άλλο υλικό. Ο αναλυτής (parser) ξαναγράφηκε στα μέσα του 1995 και ονομάστηκε *PHP/FI Version 2*.

Το όνομα FI προέρχεται από ένα άλλο πακέτο που είχε γράψει ο Rasmus και το οποίο διερμήνευε (interpreted) τα δεδομένα από φόρμες της HTML. Συνδύασε τα εργαλεία scripts της Personal Home Page με τον *Form Interpreter* και πρόσθεσε υποστήριξη για mSQL. Έτσι γεννήθηκε η *PHP/FI*, η οποία αναπτύχθηκε αλματωδώς και διάφοροι χρήστες άρχισαν να συνεισφέρουν κώδικα σ' αυτήν.

Υπολογίζεται ότι μέχρι τα τέλη του 1996, η PHP/FI χρησιμοποιείτο σε τουλάχιστον 15.000 web sites σ' όλον τον κόσμο και στα μέσα του 1997 αυτός ο αριθμός είχε ξεπεράσει τις 50.000. Στα μέσα του 1997 είχαμε επίσης μια αλλαγή στην ανάπτυξη της PHP.

Σταμάτησε να αποτελεί το αγαπημένο αντικείμενο του Rasmus και έγινε ο στόχος μιας πιο καλά οργανωμένης ομαδικής εργασίας. Ο αναλυτής (parser) ξαναγράφηκε από την αρχή από τους Zeev Suraski και Andi Gutmans και αυτός ο νέος parser αποτέλεσε τη βάση για την PHP Version 3. Ένα μεγάλο μέρος του utility code μεταφέρθηκε από την PHP/FI στην PHP3 και ένα μεγάλο μέρος του ξαναγράφηκε από την αρχή.

Σήμερα, η PHP/FI και η PHP3 έρχονται μ' έναν αριθμό εμπορικών προϊόντων όπως ο web server C2 StrongHold και το RedHat Linux. Σύμφωνα με μια συντηρητική εκτίμηση, η PHP χρησιμοποιείται από περισσότερα από 150.000 sites σ' όλον τον κόσμο.

(Η Γλώσσα Προγραμματισμού PHP Αναλυτικά, 2005, ΚΕΝΤΡΟ ΠΛΗ.ΝΕ.Τ.

ΠΕΡ. ΕΝΟΤΗΤΑΣ ΦΛΩΡΙΝΑΣ)

Στο παρακάτω τμήμα κώδικα δίνεται ένα παράδειγμα php:

```
<?phpforeach($listingsas$listing):?>
<divclass="a" onmouseover="this.style.background='#FAFAD2';"
onmouseout="this.style.background='#E6F6F6';">
<imgsrc="<?php echo $listing["photoUrl"];?>" class="thumbnail"/>
<p><?php echo $listing["categ"]; ?>
<?php echo $listing["litype"]; ?>
<?php echo $listing["livarea"].'M&sup2;'; ?></p>
<h3><?php echo '&euro; '.$listing["pric"]; ?><br>
<?php echo 'Rooms: '.$listing["room"]; ?><br>
```

```
<?php echo 'Type: '.$listing["proptype"]; ?></h3>
  <h4><?php echo 'Id: '.$listing["id"]; ?><br>
<?php echo 'Contact Name: '.$listing["name"]; ?><br>
  <?php echo 'Email: '.$listing["mail"]; ?></h4>
</div>
```

Στο παραπάνω παράδειγμα προσπελούνται όλα τα στοιχεία *listings* του πίνακα array με την εντολή `<?phpforeach ($listingsas $listing):?>`.

Στη συνέχεια με το tag *div* η οποία ορίζει ένα τμήμα στο HTML έγγραφο. Ένα τμήμα ομαδοποιεί τα περιεχόμενα που είναι μέσα στις ετικέτες `<div>...</div>` ώστε να μπορούμε να ορίζουμε ένα ενιαίο στυλ (χρώμα, γραμματοσειρά, στοίχιση) με την ιδιότητα *style*, ορίζουμε τη μορφοποίηση της κλάσης *a*.

Στο επόμενο τμήμα κώδικα παρατηρείται η εντολή *echo* που αποτελεί τη έξοδο ενός ή περισσοτέρων γραμματοσειρών. Η *echo* δεν είναι λειτουργία (function), είναι ένα κατασκευάσμα γλώσσας. Με την εντολή `<?php echo` λοιπόν, εμφανίζονται τα δεδομένα που ορίζουμε κάθε φορά, από τα *listings* που προαναφέρθηκαν.

3.3.5 Codeigniter

Ο τρόπος επικοινωνίας με την βάση διαφέρει ανάλογα με τη γλώσσα προγραμματισμού που χρησιμοποιείται. Στην περίπτωση της παρούσας εργασίας η εφαρμογή έγινε σε *codeigniter*.

Το *Codeigniter* είναι ένα framework (πλαίσιο) ανοιχτού λογισμικού για την κατασκευή δυναμικών ιστοσελίδων με PHP. Στόχος του είναι να δώσει τη δυνατότητα να αναπτυχθούν έργα, γρηγορότερα απ' ότι αν ο κώδικας γραφόταν από το μηδέν. Αυτό το πετυχαίνει παρέχοντας ένα πλούσιο σύνολο βιβλιοθηκών για τις εργασίες που συνήθως απαιτούνται, καθώς και ένα απλό περιβάλλον διεπαφής και μια λογική δομή για πρόσβαση στις βιβλιοθήκες του. Το *codeigniter* επιτρέπει στον προγραμματιστή να επικεντρωθεί στο έργο του, ελαχιστοποιώντας το μέγεθος του κώδικα που απαιτείται για μια εργασία.

M-V-C Πρότυπο.

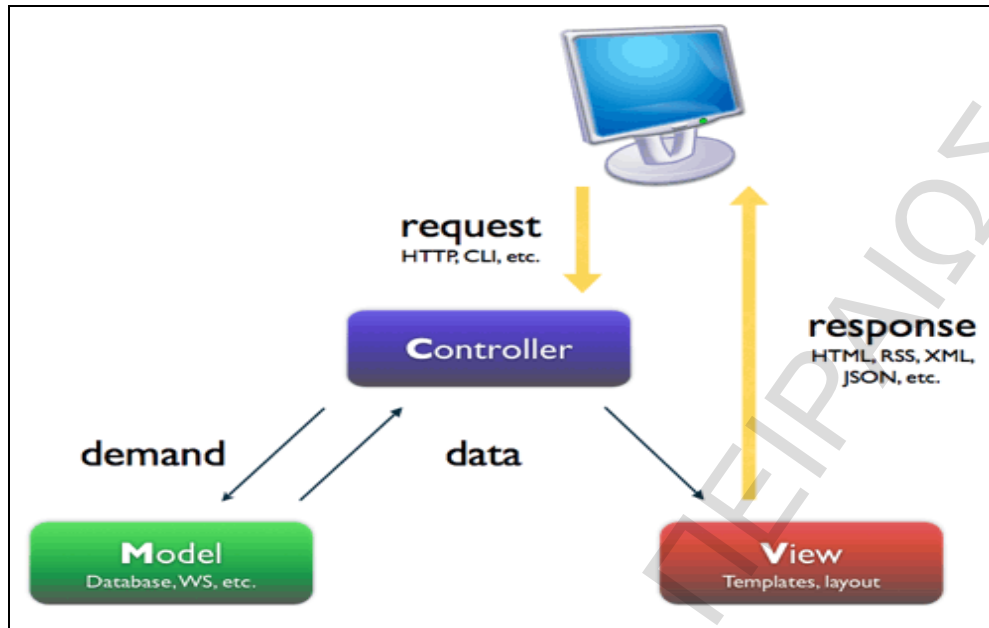
(Greek *codeigniter* framework, 2010-2013, Computerman)

Το πλαίσιο *Codeigniter* χρησιμοποιεί το πρότυπο ανάπτυξης MVC (model-view-controller). Το MVC είναι ένα κλασικό σχεδιαστικό πρότυπο που χρησιμοποιείται συχνά από εφαρμογές που χρειάζονται την δυνατότητα να διατηρούν πολλαπλές διαφορετικές παρουσιάσεις/όψεις των ίδιων δεδομένων. Το πρότυπο MVC βασίζεται στον σαφή διαχωρισμό των αντικειμένων σε τρεις κατηγορίες – τα μοντέλα για να διατηρούν τα δεδομένα (model), τις όψεις (view) για την παρουσίαση όλων ή μέρους των δεδομένων και τους ελεγκτές (controllers) για να διαχειρίζονται τις ενέργειες που επηρεάζουν τα μοντέλα ή τις όψεις.

Εξαιτίας αυτού του διαχωρισμού, πολλές όψεις και πολλοί ελεγκτές μπορούν να αλληλεπιδρούν με το ίδιο μοντέλο. Το πρότυπο αυτό, που δημιουργήθηκε αρχικά για να καλύψει εφαρμογές σε προγράμματα πελάτη (client-side), προωθεί το διαχωρισμό του κώδικα σε τρεις καλά διακριτές ενότητες:

1. **Model (Μοντέλο):** η απεικόνιση των πληροφοριών τις οποίες διαχειρίζεται η εφαρμογή – διαφορετικού τύπου κάθε φορά, ανάλογα με το πεδίο δραστηριοποίησης (domain). Το μοντέλο είναι στην ουσία το κομμάτι εκείνο που ασχολείται με τη λογική της εφαρμογής (application logic). Η λογική της εφαρμογής είναι αυτή που δίνει νόημα σε ανεπεξέργαστα δεδομένα (π.χ. προσδιορίζει ότι ένα σύνολο αριθμών αποτελεί στην πραγματικότητα τα ποσά εσόδων και εξόδων μιας εταιρίας). Πολλές εφαρμογές χρησιμοποιούν έναν μόνιμο μηχανισμό αποθήκευσης, όπως συστήματα βάσεων δεδομένων ή αρχεία, για την αποθήκευση δεδομένων. Το MVC δεν προσδιορίζει ρητά τέτοιους διακριτούς μηχανισμούς διαχείρισης πόρων, επειδή θεωρείται ότι αποτελούν επί μέρους τμήματα που περικλείονται στο Μοντέλο.
2. **View (Όψη):** το κομμάτι που ασχολείται με το τι είναι ορατό στον τελικό χρήστη και αναλαμβάνει να δημιουργήσει τα στοιχεία διεπαφής (UI) που θα δώσουν στο χρήστη τη δυνατότητα να αλληλεπιδράσει με το Μοντέλο. Τυπικά, στις Web εφαρμογές, αυτά τα στοιχεία περιλαμβάνουν HTML σελίδες που περιέχουν φόρμες ή εκθέτουν πληροφορία.
3. **Controller (Ελεγκτής):** ο κώδικας που αναλαμβάνει να διασυνδέσει τα δύο παραπάνω επίπεδα, καθορίζοντας το ποιες λειτουργίες γίνονται στο επίπεδο του

Μοντέλου ως απόκριση σε ενέργειες που προκαλεί ο χρήστης μέσω της διεπαφής. Τυπικά, όταν ένας χρήστης υποβάλλει τα στοιχεία μιας φόρμας στο επίπεδο View, ο Controller μεταφέρει τον έλεγχο εκτέλεσης σε κάποια κατάλληλη μέθοδο κάποιας κλάσης του επιπέδου Model που θα οδηγήσει στη μόνιμη αποθήκευσή τους.



Σχήμα 3. Σχηματική παρουσίαση του πρότυπου MVC

Από τα παραπάνω είναι προφανές ότι ο κύριος διαχωρισμός που επιτυγχάνει το MVC είναι ανάμεσα στο ορατό στον τελικό χρήστη τμήμα της εφαρμογής και σε αυτό που περιέχει τη λογική της εφαρμογής. Ενώ το view και ο controller είναι απαραίτητα μέρη της ανάπτυξης στο πλαίσιο του codeigniter, το κομμάτι του model είναι προαιρετικό.

3.3.6 Javascript

Η JavaScript είναι γλώσσα προγραμματισμού ηλεκτρονικών υπολογιστών, η οποία έχει σαν σκοπό την παραγωγή δυναμικού περιεχομένου και την εκτέλεση κώδικα στην πλευρά του πελάτη (client-side) σε ιστοσελίδες. Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript. (JavaScript,2013, Βικιπαίδεια)

Στο παρακάτω τμήμα κώδικα δίνεται ένα παράδειγμα JavaScript:

```
<script type="text/javascript" src="<?php echo $baseUrl;?>js/jquery.min.js"></script>
<script type="text/javascript" src="<?php echo $baseUrl;?>js/jquery.cycle.all.js"></script>
<script type="text/javascript" src="<?php echo $baseUrl;?>js/slideshow.js"></script>
```

Οι παραπάνω εντολές έχουν σαν σκοπό την επέκταση των html εγγράφων με εξαιρετικά ενεργό και δραστικό τρόπο. Τα αρχεία που ενσωματώνονται στο έγγραφο αυτό εξυπηρετούν τη λειτουργία του slideshow.

3.3.7 JQuery

Η JQuery είναι μια βιβλιοθήκη με συναρτήσεις γραμμένες σε Javascript. Το πλεονέκτημα της JQuery είναι ο συμπαγής κώδικας και η ευκολία στην εκμάθηση αλλά και στη συγγραφή.

Απαραίτητες γνώσεις που πρέπει να κατέχει κάποιος πριν αρχίσει να ασχολείται με την JQuery είναι η Html, η Css, και η Javascript.

Η jQuery είναι μια βιβλιοθήκη (framework) JavaScript που χρησιμοποιείται από προγραμματιστές για τη ταχεία ανάπτυξη ιστοσελίδων και διαδικτυακών εφαρμογών που χρειάζονται μεγάλη ευχρηστικότητα και διαδραστικότητα (interactivity). Η jQuery πρωτοεμφανίστηκε τον Ιανουάριο του 2006 στο BarCamp από τον John Resig. Πρόκειται για μια βιβλιοθήκη Javascript ανοιχτού κώδικα, υπό τις άδειες MIT License και την GNU General Public License.

Μερικά από τα πλεονεκτήματα της JQuery είναι ότι ακολουθεί την αρχή KISS (KeepItSimpleStupid): Η βιβλιοθήκη JQuery προσπαθεί να υπεραπλουστεύσει τον προγραμματισμό σε Javascript προσφέροντας πραγματικά απλούς μηχανισμούς και εντολές μέσω του framework της.

Παρέχει πλήρη και αναλυτικότερη τεκμηρίωση που συμπληρώνεται από την εκτεταμένη παρουσία ηλεκτρονικών βοηθημάτων. Εκτός από την πολυκαλοδουλεμένη τεκμηρίωσή της JQuery, οι ενδιαφερόμενοι μπορούν να ανατρέξουν και στην σελίδα του VisualJQuery όπου μπορούν να βρουν μία εναλλακτική αλλά πολύ βολική, από άποψη δομής, τεκμηρίωση.

Υποστηρίζεται από μία πάρα πολύ ενεργή κοινότητα: Όπως για τα περισσότερα open source έργα λογισμικού, έτσι και για το JQuery η ύπαρξη μιας κατά το μέγιστο δυνατό ενεργής κοινότητας αποτελεί τον ακρογωνιαίο λίθο για την ανάπτυξη και ευημερία του.

Έχει μικρό μέγεθος: Το γεγονός ότι το βασικό πακέτο της JQuery είναι μόλις 20Kb αφενός επιβεβαιώνει την πρώτη παρατήρηση, ότι δηλαδή η φιλοσοφία της έγκειται στην απλότητα και αφετέρου κάνει πολύ εύκολη την κατανόηση της αρχιτεκτονικής της.

Έχει ποικιλία χαρακτηριστικών: Η JQuery δίνει τη δυνατότητα στον χρήστη να χρησιμοποιήσει σχεδόν το σύνολο των δυνατοτήτων που προσφέρει η γλώσσα JavaScript. Από απλά χαρακτηριστικά που σχετίζονται με βασικές λειτουργίες εμφάνισης / απόκρυψης, ως Ajax κλήσεις και σύνθετα εφέ.

Παρέχει τη δυνατότητα επεκτασιμότητας: Η λογική με την οποία είναι φτιαγμένη η JQuery είναι απλή, πράγμα που αντικατοπτρίζεται και στον ίδιο της τον κώδικα. Αυτό κάνει πολύ εύκολη την επέκτασή / τροποποίησή της με τη χρήση plugins. (ΕισαγωγήστηνJQuery, 2013, iDesigner)

3.3.8Plugin

Ως plug-in, επίσης plugin (ελλ. άρθρωμα), ορίζεται ένα σύστημα συστατικών κάποιου λογισμικού που προσθέτει ιδιαίτερες δυνατότητες σε ένα μεγαλύτερο λογισμικό. Τα plug-in χρησιμοποιούνται ιδιαίτερα σε φυλλομετρητές για την προβολή video, την ανίχνευση ιών, και την επίδειξη νέων τύπων αρχείων. Πολύ γνωστά παραδείγματα plug-in είναι το Adobe FlashPlayer και το QuickTime.

Τα plug-in είναι εξειδικευμένες μορφές αυτού που αποκαλείται add-on ή πρόσθετο και περιλαμβάνει επεκτάσεις ή οπτικά θέματα. Οι διάφορες διαδικτυακές εφαρμογές υποστηρίζουν plug-in για πολλούς λόγους, ανάμεσα στους οποίους περιλαμβάνονται η δυνατότητα άλλων προγραμματιστών να προγραμματίσουν επιπλέον δυνατότητες κάποιας εφαρμογής, η υποστήριξη εύκολης πρόσθεσης νέων χαρακτηριστικών, η μείωση του μεγέθους του πυρήνα μιας εφαρμογής και ο διαχωρισμός του πηγαίου κώδικα από της εφαρμογή σε περίπτωση ασύμβατων αδειών. (Plug-in, 2012, Βικιπαίδεια)

3.3.9jQueryCyclePlugin

Το jQueryCyclePlugin είναι ένα πρόσθετο προβολής σελίδων που υποστηρίζει πολλούς διαφορετικούς τύπους εφέ μετάβασης. Υποστηρίζει αυτόματο σταμάτημα, αυτόματη προσαρμογή, κουμπιά ενεργοποίησης κ.α. Επίσης υποστηρίζει, αλλά δεν απαιτεί, το easingplugin. (jQueryPlugins (bymalsup), 2013, GitHub, Inc.)

Το plugin παρέχει μια μέθοδο που ονομάζεται κύκλος (cycle) η οποία ενεργοποιείται σε ένα περιεχόμενο στοιχείο. Κάθε στοιχείο-παιδί του περιεχομένου γίνεται μια σελίδα «slide». Οι επιλογές ελέγχουν το πώς και το πότε θα αλλάζουν οι σελίδες «slides».

Με λίγα λόγια το jQueryCyclePlugin παίρνει κάθε αξιόπιστη ομάδα εμφωλευμένων html στοιχείων και τα μετατρέπει σε μια προβολή διαφανειών (slideshow) χρησιμοποιώντας διάφορα εφέ jQuery.

Στο παρακάτω τμήμα κώδικα δίνεται ένα παράδειγμαjQueryCyclePlugin:

```
$(function() {

    $('#slideshow').cycle({
fx: 'scrollHorz',
```



```

timeout: 7000,
prev: '#prev',
next: '#next',
pager: '#nav',
pagerAnchorBuilder: pagerFactory
});

function pagerFactory(idx, slide) {
var s = idx > 5 ? ' style="display:none" : '';
return '<li'+s+'><a href="#"'+(idx+1)+'</a></li>';
};

});

```

Στον παραπάνω κώδικα ορίζεται όλη η λειτουργία του slideshow. Το plugin παρέχει μια μέθοδο που ονομάζεται κύκλος (cycle), όπως προαναφέρθηκε. Μέσα σε αυτήν την μέθοδο ορίζονται οι διάφορες παράμετροι της λειτουργίας του slideshow, όπως ο τρόπος μετάβασης των slides: *fx: 'scrollHorz'*, ο χρόνος αλλαγής τους: *timeout: 7000*, ο οποίος αντιστοιχεί σε 7 sec, κ.α.

3.3.10 Xml-Rpc

Το xml-rpc είναι ένα πρωτόκολλο Κλήσης Απομακρυσμένης Διαδικασίας που λειτουργεί μέσω του διαδικτύου. Ένα μήνυμα xml-rpc είναι ένα HTTP-POST αίτημα. Το σώμα του αιτήματος είναι σε xml. Μια διαδικασία εκτελείται στον διακομιστή (server) και η τιμή που επιστρέφει είναι επίσης σε μορφή xml. Οι παράμετροι των διαδικασιών μπορεί να είναι αριθμοί, ημερομηνίες κλπ. Μπορούν επίσης να είναι σύνθετες καταγραφές ή δομημένες λίστες.

Το XML-RPC βασίζεται στο μοντέλο πελάτη-διακομιστή και στο πρωτόκολλο HTTP. Ο πελάτης στέλνει το όνομα και τις παραμέτρους της απομακρυσμένης διαδικασίας σε αίτημα μορφής XML request, και ο διακομιστής απαντά με παραμέτρους επιστροφής πάλι σε μορφή XML. Το XML-RPC δεν αντιλαμβάνεται αντικείμενα, ούτε μπορεί να επεκτείνει το λεξιλόγιο XML. Το XML-RPC επιτρέπει την ολοκλήρωση ετερογενών εφαρμογών που είναι γραμμένες σε διαφορετικές γλώσσες, τεχνολογίες ή πλατφόρμες και ανταλλάσσουν απλές δομές δεδομένων. (XML-RPC και Java, 1994 – 2009, Pdrlab)

Το XML-RPC αποτελείται από τρία τμήματα:

- Μοντέλο δεδομένων
Το σύνολο των επιτρεπόμενων τύπων δεδομένων για τις παραμέτρους των διαδικασιών και τα σφάλματα.
- Δομές αιτημάτων
Πρόκειται για εντολές HTTP POST που περιέχουν ονόματα διαδικασιών και τιμές παραμέτρων κλήσης.
- Δομές απαντήσεων
Πρόκειται για απαντήσεις HTTP που περιέχουν τιμές παραμέτρων επιστροφής ή σφάλματα.

Το Μοντέλο δεδομένων XML-RPC ορίζει τους ακόλουθους τύπους δεδομένων: int, double, Boolean, string, dateTime.iso8601, base64. Οι βασικοί τύποι πάντα περικλείονται από στοιχεία *value*. Μόνο τα strings μπορεί να περικλείονται σε στοιχεία *value* χωρίς τα στοιχεία *string*. Οι βασικοί τύποι μπορούν να συνδυαστούν σε σύνθετους τύπους, array και struct. Το array παριστά ακολουθιακά αποθηκευμένη πληροφορία, ενώ το struct παριστά ζεύγη ονόματος-τιμής.

Οι τιμές στο τύπο array περικλείονται από στοιχεία value, array, data.

Τα αιτήματα XML-RPC είναι συνδυασμός περιεχομένου XML και κεφαλίδων HTTP. Το περιεχόμενο XML παρέχει πληροφορία για τη διαδικασία που καλείται και τις παραμέτρους που απαιτούνται, ενώ οι κεφαλίδες HTTP παρέχουν ένα κέλυφος για την μεταβίβαση του περιεχομένου μέσω του Ιστού.

Κάθε αίτημα περιέχει μόνο ένα κείμενο XML, με αρχικό στοιχείο το *methodCall*. Το *methodCall* περιέχει ένα στοιχείο *methodName* και ένα στοιχείο *params*.

Το *methodName* περιέχει το όνομα της καλούμενης διαδικασίας, ενώ το *params* περιέχει τη λίστα των παραμέτρων με τις τιμές τους, δηλαδή στοιχεία *value*.

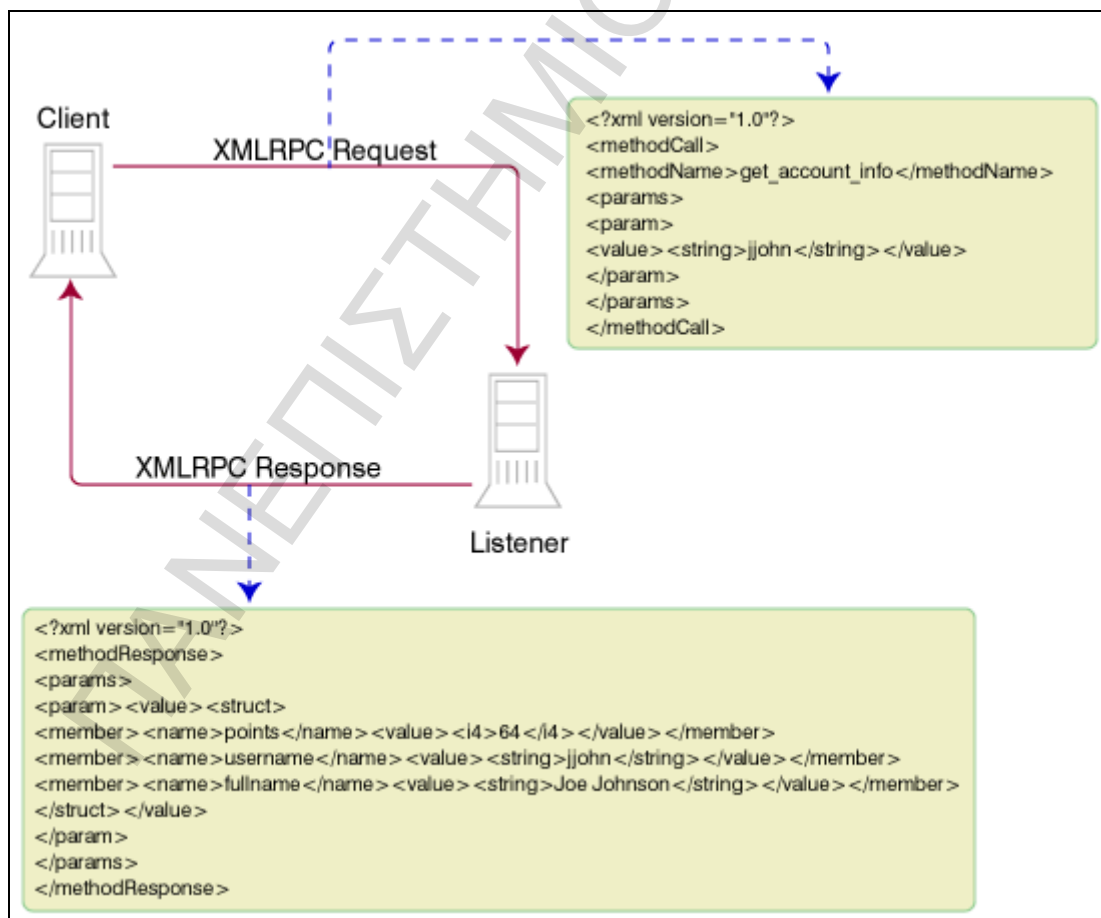
Η απάντηση XML-RPC περιέχει μόνο μια παράμετρο. Όμως με το δεδομένο ότι η παράμετρος επιστροφής μπορεί να είναι array ή struct, μπορούμε να επιστρέψουμε πολλαπλές τιμές. Η απάντηση πρέπει να περιέχει οπωσδήποτε μια παράμετρο. Στην απλούστερη περίπτωση επιστρέφουμε μια τιμή "επιτυχούς εκτέλεσης", πχ μια τιμή boolean true.

Όπως και τα αιτήματα, οι απαντήσεις XML-RPC έχουν κεφαλίδες HTTP. Οι απαντήσεις χρησιμοποιούν τον κωδικό 200 OK, ακόμη και αν η απάντηση περιέχει μήνυμα σφάλματος.

Μόλις η απάντηση παραληφθεί από τον πελάτη η σύνδεση XML-RPC κλείνει. Κάθε νέο αίτημα απαιτεί ξεχωριστή σύνδεση XML-RPC.

Ένας τύπος απάντησης XML-RPC είναι τα σφάλματα. Αν παρουσιαστεί κάποιο πρόβλημα στην επεξεργασία του αιτήματος από τον διακομιστή, το στοιχείο *methodResponse* θα περιέχει ένα στοιχείο σφάλματος αντί του στοιχείου της παραμέτρου. Το στοιχείο σφάλματος έχει και αυτό μια μοναδική τιμή, η οποία δείχνει τι πηγή στραβά.

Η XML - RPC λειτουργεί κωδικοποιώντας τις αιτήσεις RPC σε XML και την αποστολή τους σε μια τυπική σύνδεση HTTP σε ένα κομμάτι server ή ακροατή. Ο ακροατής αποκωδικοποιεί η XML, εκτελεί την απαιτούμενη διαδικασία, και στη συνέχεια τα πακέτα ανεβάζουν τα αποτελέσματα σε XML και τα στέλνουν πίσω στον πελάτη. Ο πελάτης αποκωδικοποιεί την XML, μετατρέπει τα αποτελέσματα σε στάνταρ γλωσσικούς τύπους δεδομένων, και συνεχίζει την εκτέλεση. Το σχήμα 1 είναι ένα διάγραμμα που δείχνει μια πραγματική XML - RPC συνομιλία μεταξύ ενός πελάτη και έναν ακροατή ο οποίος επιστρέφει τα αποτελέσματα της διαδικασίας αυτής.



Σχήμα 4. Η λειτουργία του xmlrpc

Στο παρακάτω τμήμα κώδικα δίνεται ένα παράδειγμα xml-rpc:

```

$this->xmlrpc->request($request);
    $image_url='localhost/codecarousel3/photos1/noimage.jpg';
        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]["key"] = $listingKey;
            if ($listingValue['mainImageURL'] != "")
                {$this->data["listings"][$listingKey]["photoUrl"] =
                $listingValue['mainImageURL'];
            }
        }
    }

```

Η βιβλιοθήκη Codeigniter περιλαμβάνει την κλάση xml-rpc που έχει την ανωτέρω σύνταξη. Όπως και οι άλλες κλάσεις σε Codeigniter, οι xml-rpc και xml-rpcs κλάσεις αρχικοποιούνται στον controller με την εντολή *\$this->load->library ()*.

Στο παράδειγμα ορίζεται το αίτημα xmlrpcrequest. Η επόμενη εντολή, ορίζει την τιμή της μεταβλητής image_url, που είναι ένα μονοπάτι (path) μιας εικόνας.

Στη συνέχεια γίνεται ένας έλεγχος αποστολής request. Σε περίπτωση αποτυχίας εμφανίζεται μήνυμα λάθους, ενώ σε περίπτωση επιτυχίας, για κάθε στοιχείο της λίστας του πίνακα που επιστρέφεται, ορίζονται διάφορες μεταβλητές που παίρνουν αυτά τα στοιχεία.

3.3.11 Υπηρεσίες Ιστού

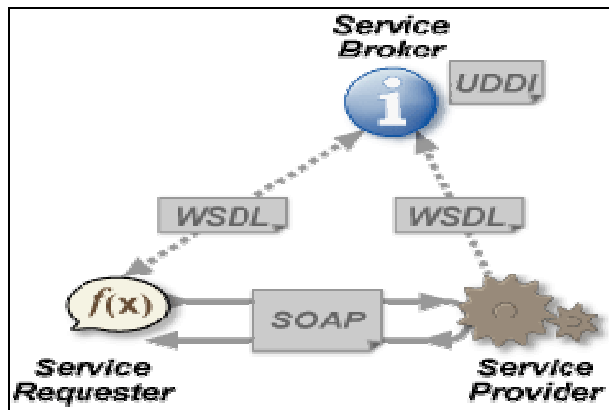
Webservice είναι μια μέθοδος επικοινωνίας μεταξύ δύο ηλεκτρονικών συσκευών μέσω του WorldWideWeb (WWW). Ένα Webservice είναι μια λειτουργία λογισμικού που παρέχεται σε μια διεύθυνση δικτύου και μπορεί να χαρακτηριστεί σαν μια υπηρεσία που είναι πάντα ενεργή με την έννοια του βοηθητικού προγραμματισμού.

Τα Webservices είναι μια τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους, ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Ένα Webservice είναι μια διεπαφή λογισμικού (softwareinterface) 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 μια λειτουργία (operation) προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή. Μια ομάδα από Webservices οι οποίες αλληλεπιδρούν μεταξύ τους, καθορίζει μια εφαρμογή Webservice.

Τα webservices εκθέτουν χρήσιμη λειτουργικότητα σε χρήστες του διαδικτύου μέσα από ένα πρότυπο δικτυακό πρωτόκολλο. Στις περισσότερες περιπτώσεις αυτό το πρωτόκολλο είναι το SOAP (SimpleObject Access Protocol).

Τα webservices παρέχουν ένα τρόπο να περιγράψουν τις διεπαφές τους με αρκετή λεπτομέρεια ώστε να επιτρέψουν στο χρήστη τους να χτίσει μια εφαρμογή πελάτη η οποία να επικοινωνήσει μαζί τους. Η περιγραφή συνήθως παρέχεται σε ένα έγγραφο XML το οποίο ονομάζεται έγγραφο WSDL (Web ServicesDescriptionLanguage). Τα webservices καταχωρούνται ώστε οι δυνητικοί χρήστες να μπορούν να τα βρουν εύκολα. Αυτό γίνεται με το UDDI (Universal Discovery Description and Integration).

Τα webservices λοιπόν αποτελούν μία αρχιτεκτονική καταμεμημένων συστημάτων κατασκευασμένα από πολλά διαφορετικά υπολογιστικά συστήματα τα οποία επικοινωνούν μέσω του δικτύου ώστε να δημιουργήσουν ένα σύστημα. Αποτελούνται από ένα σύνολο από πρότυπα τα οποία επιτρέπουν στους υπεύθυνους για την ανάπτυξη (προγραμματιστές - developers) να υλοποιήσουν καταμεμημένες εφαρμογές (χρησιμοποιώντας διαφορετικά εργαλεία από διαφορετικούς προμηθευτές) ώστε να κατασκευάσουν εφαρμογές που χρησιμοποιούν ένα συνδυασμό από ενότητες λογισμικού (softwaremodules) οι οποίες καλούνται από συστήματα που ανήκουν σε διαφορετικά τμήματα ενός οργανισμού ή σε διαφορετικούς οργανισμούς. (Web service, 2013, Wikipedia)



Σχήμα 5. Η αρχιτεκτονική του webservice

Όπως είδαμε παραπάνω, η δημιουργία μιας εφαρμογής περιλαμβάνει πληθώρα γλωσσών προγραμματισμού οι οποίες διαφέρουν από εφαρμογή σε εφαρμογή. Η πραγματοποίηση μιας εφαρμογής όμως δεν εξαρτάται μόνο από τις γλώσσες προγραμματισμού αλλά και από τον προσδιορισμό και την ανάλυση των απαιτήσεων.

Σε κάθε στάδιο υλοποίησης οι απαιτήσεις διαφέρουν. Ένας τρόπος ελέγχου της υλοποίησης της εφαρμογής είναι το αντικειμενοστρεφές μοντέλο RationalUnifiedProcess.

4. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΡΓΑΣΙΑΣ

4.1 Σύντομη παρουσίαση της RUP

Το αντικειμενοστρεφές μοντέλο RationalUnifiedProcess έχει αναπτυχθεί από τους δημιουργούς της αντικειμενοστρεφούς γλώσσας μοντελοποίησης UML, τους Booch, Rumbaugh και Jacobson (WernerHeijstek, 2010, TheRationalUnifiedProcess)

Βασίζεται στο μοντέλο του καταρράκτη, αλλά θεωρεί ότι η ανάλυση απαιτήσεων, ο σχεδιασμός, η υλοποίηση και ο έλεγχος δεν συμπίπτουν με χρονικές φάσεις αλλά αντιπροσωπεύουν τμήματα διαδικασίας τα οποία λαμβάνουν χώρα σε διάφορες χρονικές φάσεις.

Η διαδικασία RationalUnifiedProcess αποτελείται από ένα σύνολο οδηγιών σχετικά με τις τεχνικές και οργανωτικές απόψεις της ανάπτυξης λογισμικού. Η διαδικασία αυτή αφορά κυρίως στην Ανάλυση Απαιτήσεων και στο Σχεδιασμό.

Η διαδικασία RationalUnifiedProcess είναι δομημένη σε δύο διαστάσεις:

- 1) Χρόνο - Χωρισμός του κύκλου ζωής σε φάσεις και επαναλήψεις και
- 2) Τμήματα διαδικασίας - Καλά ορισμένες εργασίες.

Η δόμηση έργου σύμφωνα με τη διάσταση των τμημάτων διαδικασίας περιλαμβάνει τις ακόλουθες δραστηριότητες:

- 1) Σύλληψη απαιτήσεων (Requirementscapture): Μια αφήγηση του τι πρέπει να κάνει το σύστημα.
- 2) Ανάλυση και σχεδιασμός (Analysisanddesign): Μια περιγραφή του πώς θα υλοποιηθεί το σύστημα.
- 3) Υλοποίηση (Implementation): Η παραγωγή του κώδικα.
- 4) Έλεγχος (Test): Η επαλήθευση του συστήματος.

4.2 Διαγράμματα UML

Η UML ορίζει 9 είδη διαγραμμάτων για να αναπαραστήσει τις διάφορες απόψεις μοντελοποίησης.

Μερικά από αυτά είναι, τα Διαγράμματα τάξεων (ClassDiagrams), στα οποία αναπαρίσταται μια στατική δομή των τάξεων και των σχέσεων τους, τα Διαγράμματα αντικειμένων (ObjectDiagrams) τα οποία αναπαριστούν τα αντικείμενα και τις σχέσεις τους, τα Διαγράμματα περιπτώσεων χρήσης (Usecasediagrams) τα οποία περιγράφουν τη

συμπεριφορά ενός συστήματος από την οπτική γωνία ενός χρήστη, τα Διαγράμματα σειράς (Sequencediagrams) τα οποία αναπαριστούν αλληλεπιδράσεις ανάμεσα στα αντικείμενα από μία χρονική άποψη και άλλα τα οποία αναφέρονται αναλυτικότερα παρακάτω.

4.3 Ανάλυση και σχεδιασμός

Στην παρούσα εργασία παρουσιάζονται τα διαγράμματα σε ό,τι αφορά στην εφαρμογή αλλά και τμήματα του Codeigniter που μπορούν να αναπαρασταθούν διαγραμματικά.

Διάγραμμα περιπτώσεων χρήσης

Οι περιπτώσεις χρήσης εφευρέθηκαν από τον IvanJacobson(MartinFowler,2006,Εισαγωγή στη UML Συνοπτικός οδηγός της πρότυπης γλώσσας μοντελοποίησης αντικειμένων).

Περιγράφουν τη συμπεριφορά ενός συστήματος από την οπτική γωνία ενός χρήστη.

Επιτρέπουν τον ορισμό των ορίων του συστήματος και του περιβάλλοντος.

Μία περίπτωση χρήστη 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).

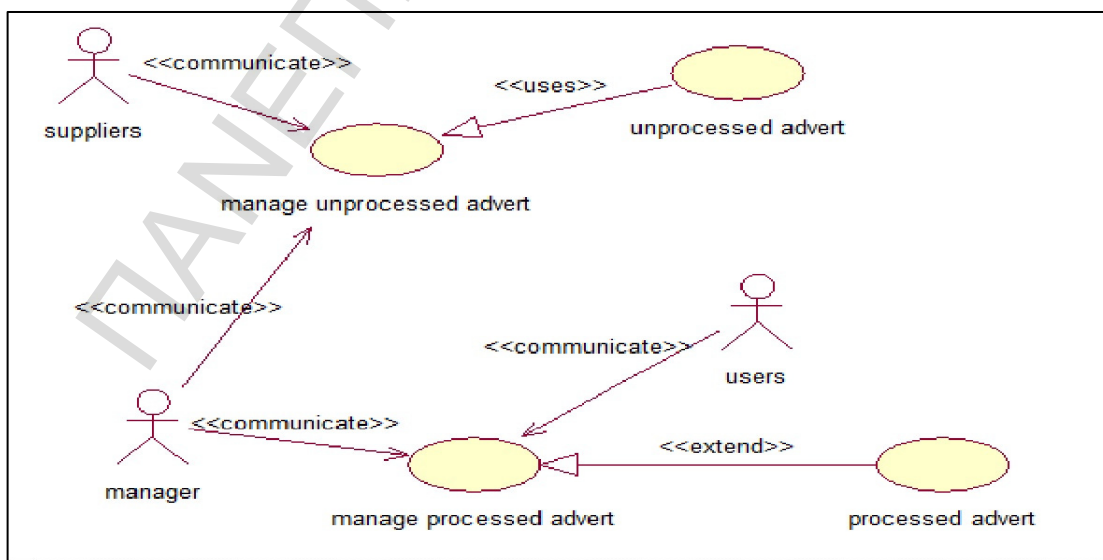
Το μοντέλο περιπτώσεων χρήσης περιλαμβάνει :

- 1) Τις ίδιες τις περιπτώσεις χρήσης
- 2) 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.

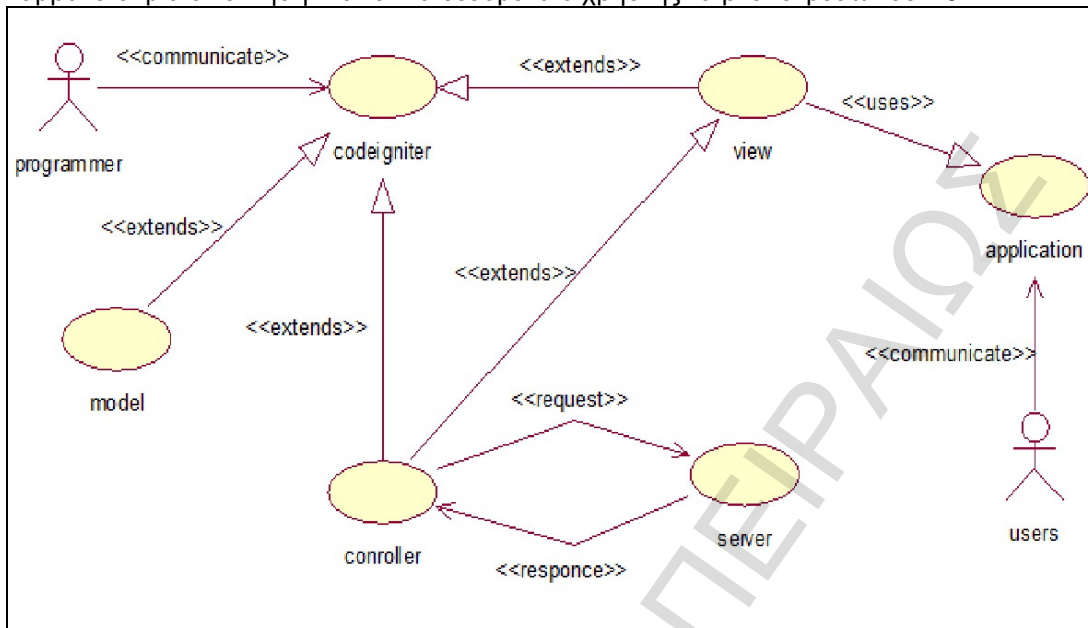
Το διάγραμμα περιπτώσεων χρήσης αναπαριστά τις λειτουργίες του συστήματος από την οπτική γωνία του χρήστη. Περιγράφονται οι ενεργοποιοί του συστήματος, οι suppliers, οι users και οι managers. Οι περιπτώσεις χρήσης προέκυψαν από τον ορισμό του προβλήματος και τη μελέτη του συστήματος. supplier/suppliers που εφοδιάζουν το σύστημα με μη επεξεργασμένα δεδομένα. Στη συνέχεια ο manager/managers επεξεργάζεται τα δεδομένα και τα «ανεβάζει» στο σύστημα. Τέλος ο user/users έχει πρόσβαση στα επεξεργασμένα δεδομένα.

Στο παρακάτω διάγραμμα περιπτώσεων χρήσης αναπαρίσταται η εφαρμογή, όπου ο supplier/suppliers εφοδιάζουν το σύστημα με μη επεξεργασμένα δεδομένα. Στη συνέχεια ο manager/managers επεξεργάζεται τα δεδομένα και τα «ανεβάζει» στο σύστημα. Τέλος ο user/users έχει πρόσβαση στα επεξεργασμένα δεδομένα.



Σχήμα 6. Διάγραμμα περιπτώσεων χρήσης της εφαρμογής

Στο παρακάτω διάγραμμα περιπτώσεων χρήσης αναπαρίσταται ο Codeigniter. Ο προγραμματιστής επικοινωνεί με την εφαρμογή, η οποία περιλαμβάνει το μοντέλο, την προβολή και τον ελεγκτή. Μέσω του ελεγκτή στέλνεται ένα αίτημα στον διακομιστή και λαμβάνεται μια απάντηση. Τα τελικά δεδομένα ο χρήστης τα βλέπει μέσω του view.



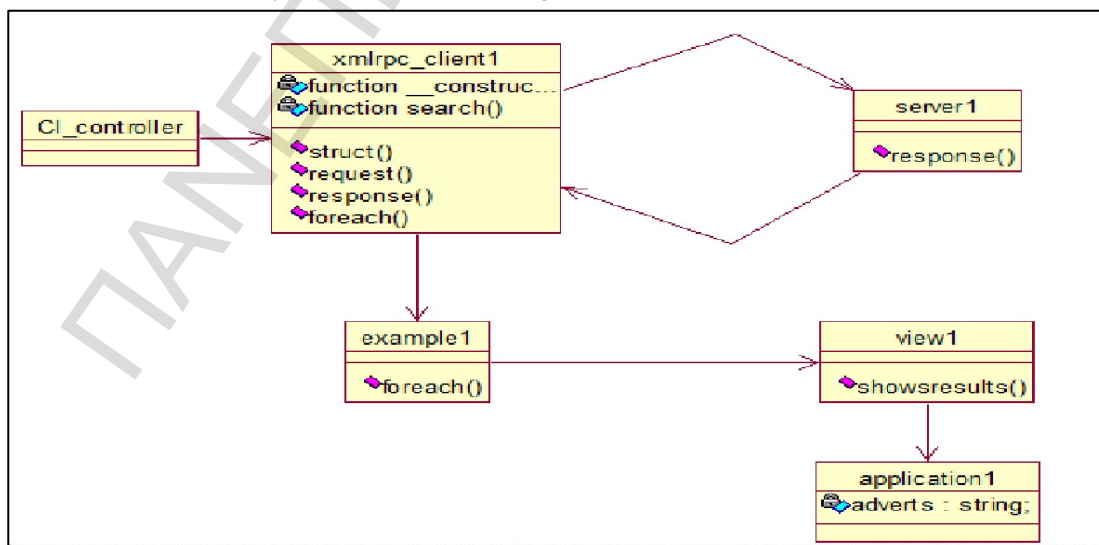
Σχήμα 7. Διάγραμμα περιπτώσεων χρήσης του προγράμματος

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 και το τρίτο τις λειτουργίες της. Στο διάγραμμα τάξεων αναπαρίσταται μια στατική δομή των τάξεων και των σχέσεών τους. Το διάγραμμα των τάξεων ενός συστήματος είναι ένα διάγραμμα δομής που περιέχει τις κλάσεις μαζί με τους αντίστοιχους δεσμούς εξάρτησης, γενίκευσης και σύνδεσης. Έτσι ένα διάγραμμα κλάσεων μπορεί να απεικονίσει τη χρήση της κληρονομικότητας στο σχεδιασμό με τη χρήση δεσμών γενίκευσης.

Στο παρακάτω διάγραμμα τάξεων αναπαρίσταται ο Codeigniter. Η κλάση controller μέσω του xmlhttp_client1 επικοινωνεί με τον server αλλά και με το view. Τα αποτελέσματα που εμφανίζονται στο view μέσω του example1 είναι ορατά στην εφαρμογή.

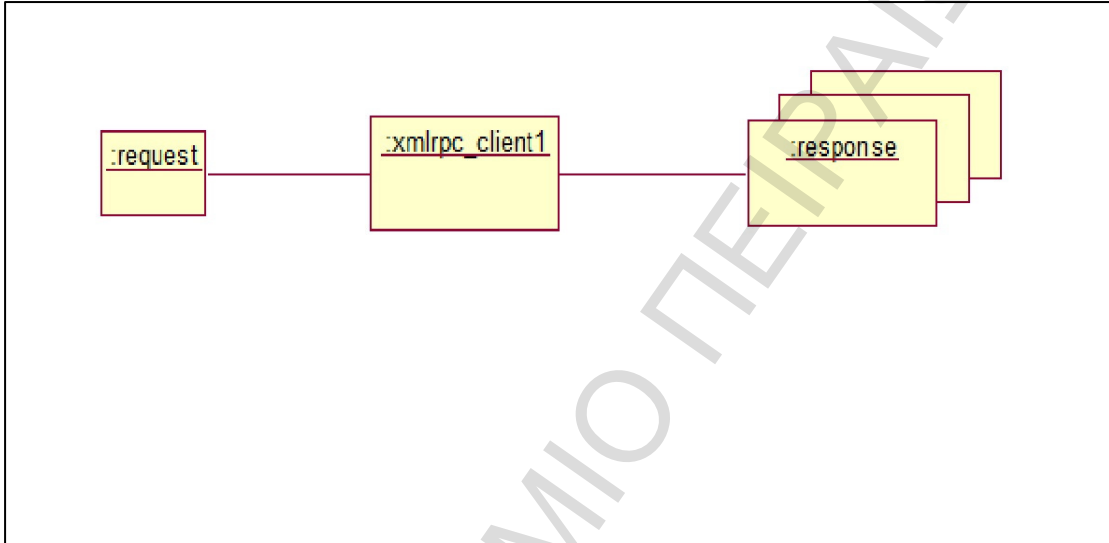
Σχήμα 8. Διάγραμμα τάξεων του προγράμματος



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 διαγράμματα αντικειμένων αναπαριστούν τα αντικείμενα και τις σχέσεις τους και αντιστοιχούν σε απλοποιημένα διαγράμματα συνεργασίας αλλά δεν αναπαριστούν μετάδοση μηνυμάτων.



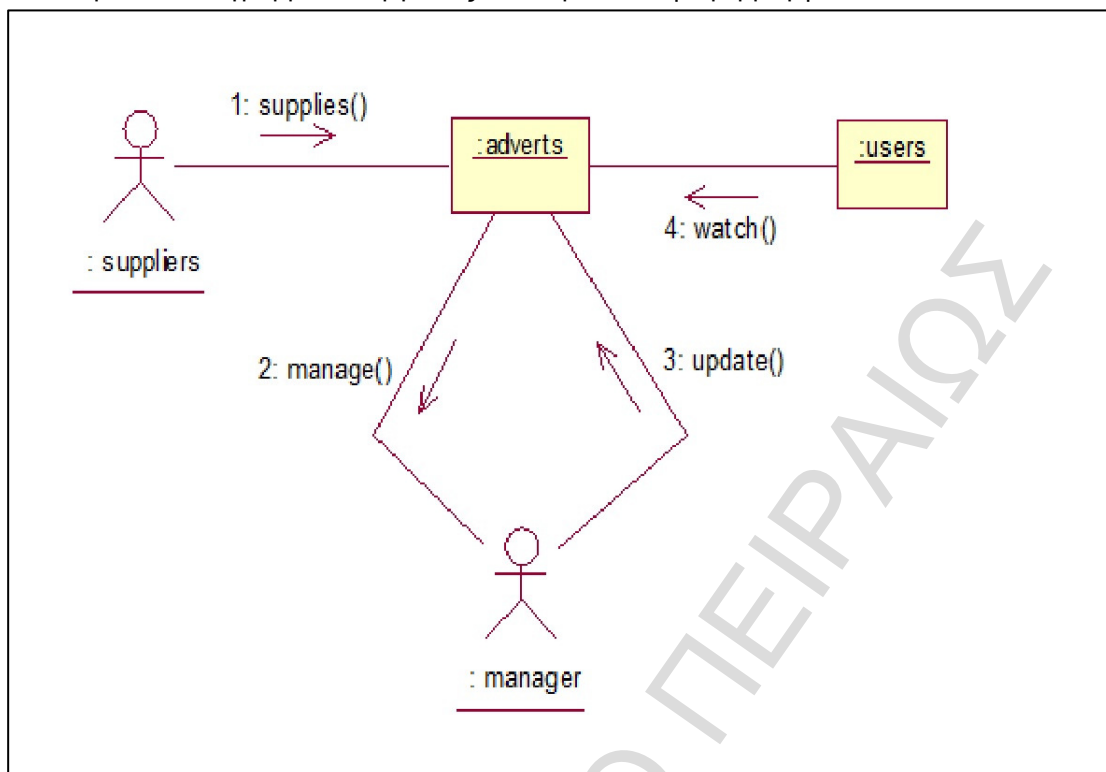
Σχήμα 9. Διάγραμμα αντικειμένων του προγράμματος

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.

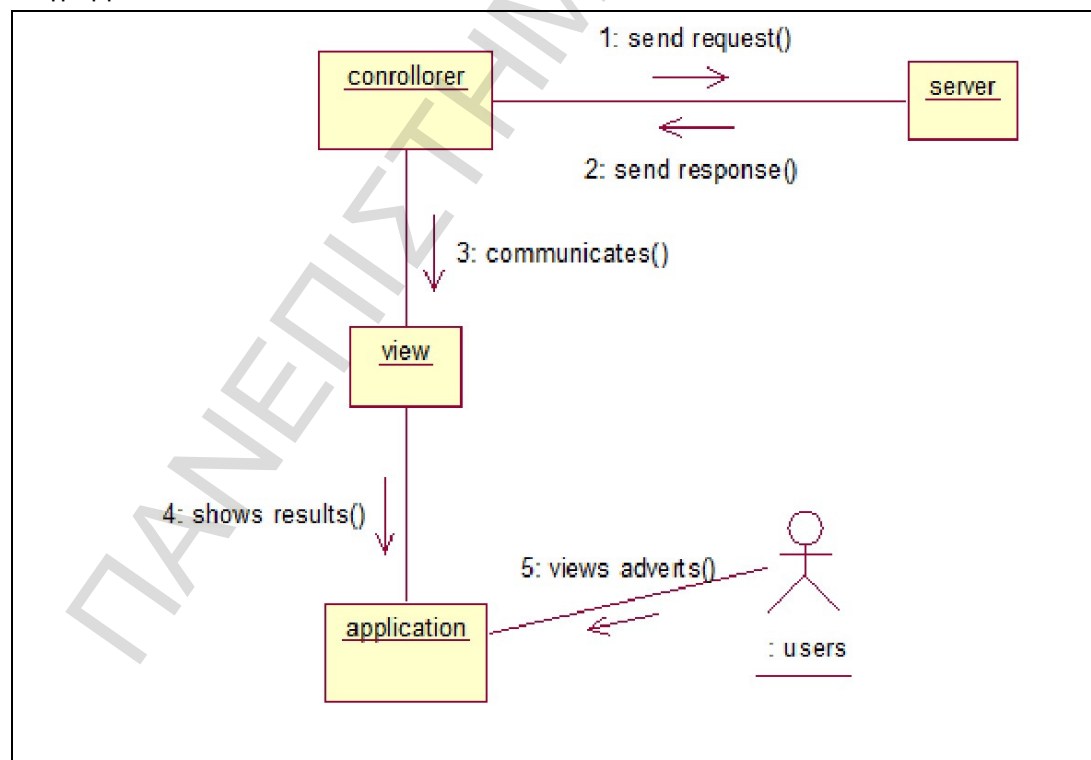
Ένα διάγραμμα συνεργασίας αναπαριστά τα αντικείμενα, τις συνδέσεις και τις αλληλεπιδράσεις. Πρώτη ενέργεια είναι η παροχή δεδομένων από τον supplier, η δεύτερη ενέργεια είναι η επεξεργασία από τον manager, η τρίτη ενέργεια είναι το ανέβασμα των δεδομένων από τον manager και η τέταρτη και τελευταία είναι η προβολή των δεδομένων στον user.

Στο παρακάτω διάγραμμα συνεργασίας αναπαρίστανται η Εφαρμογή.



Σχήμα 10. Διάγραμμα συνεργασίας της εφαρμογής

Αντίστοιχη αναπαράσταση της λειτουργίας του Codeigniter φαίνεται στο παρακάτω διάγραμμα.



Σχήμα 11. Διάγραμμα συνεργασίας του προγράμματος

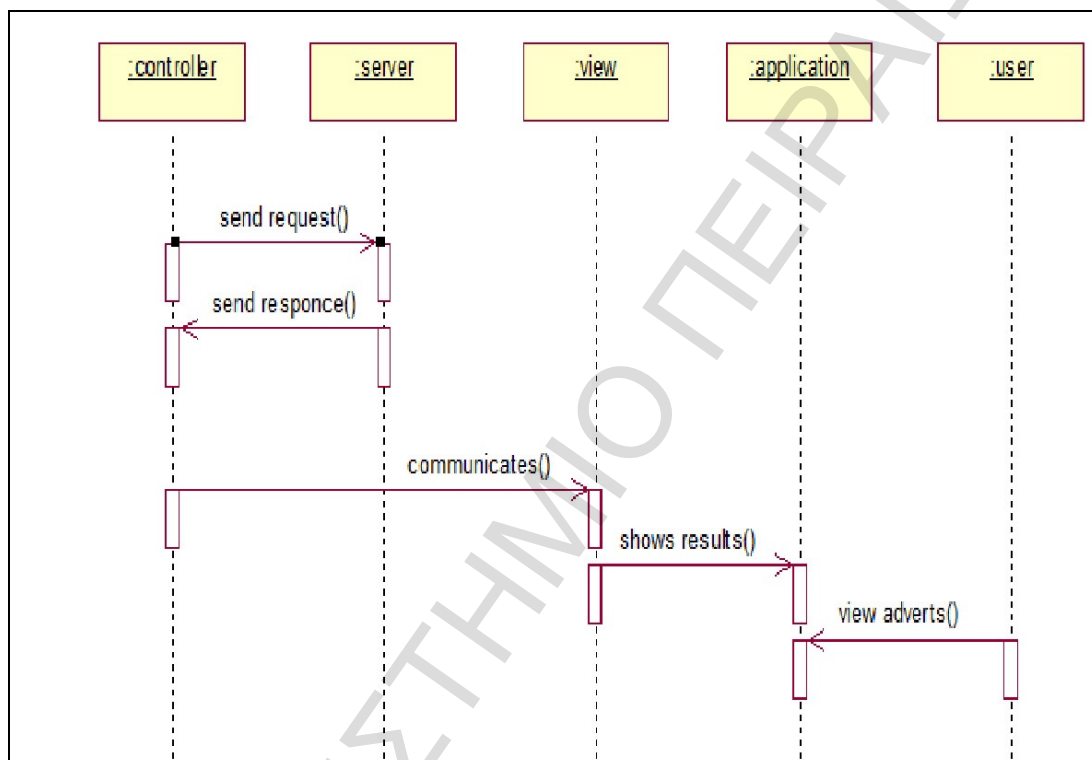
Διάγραμμα σειράς

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

Codeigniter

Πάλι παρουσιάζεται η λειτουργία του Codeigniter με βάση τη χρονική αναπαράσταση και την αλληλεπίδραση μεταξύ τους.



Σχήμα 12. Διάγραμμα σειράς του προγράμματος

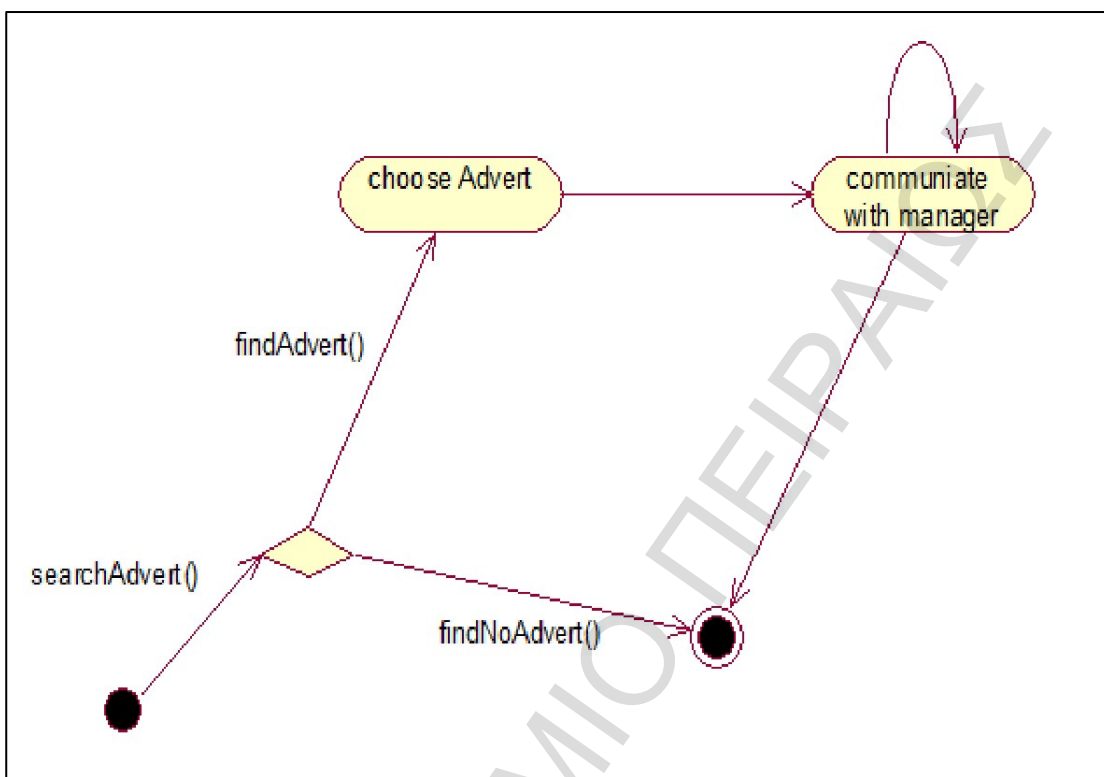
Διάγραμμα δραστηριοτήτων

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

Ένα διάγραμμα δραστηριοτήτων αναπαριστά την κατάσταση εκτέλεσης ενός μηχανισμού σαν 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 στο τέλος 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.

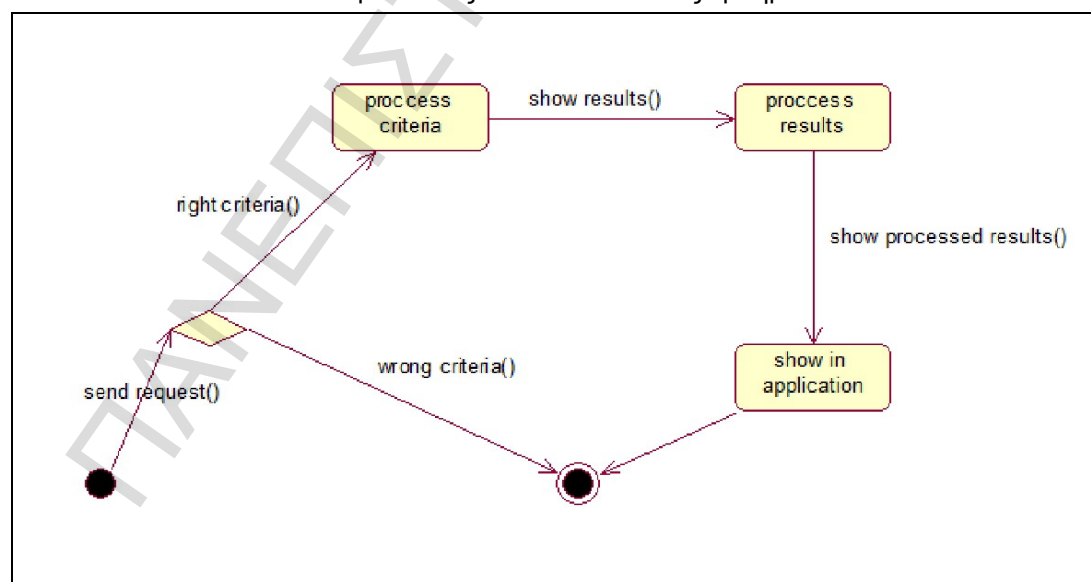
Το διάγραμμα δραστηριοτήτων αναπαριστά τη συμπεριφορά μιας λειτουργίας ως σύνολο ενεργειών. Στο συγκεκριμένο παράδειγμα γίνεται η ανάλυση εύρεσης αγγελίας (advert) από ένα χρήστη.

Στο παρακάτω διάγραμμα δραστηριοτήτων αναπαρίσταται η Εφαρμογή.



Σχήμα 13. Διάγραμμα δραστηριοτήτων της εφαρμογής

Σε αυτό το διάγραμμα φαίνεται η διαδικασία κατά την οποία στέλνεται το αίτημα από τον controller στον server και οι περιπτώσεις σωστών και λάθος κριτηρίων.



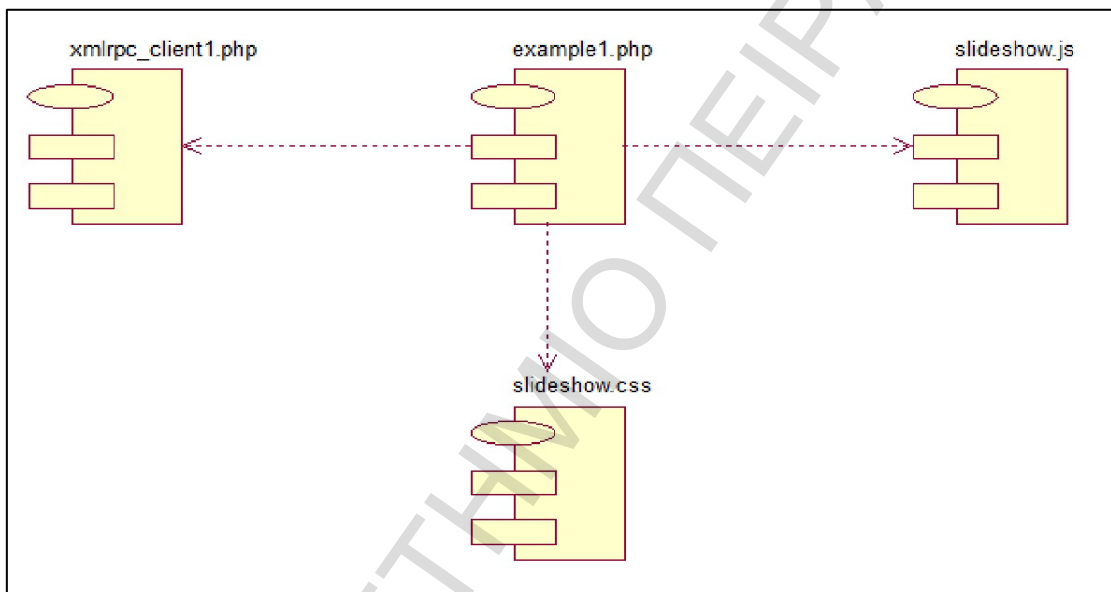
Σχήμα 14. Διάγραμμα δραστηριοτήτων του προγράμματος

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 τοσυμβολισμό 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 τουεξαρτήματος.

Το διάγραμμα εξαρτημάτων αναπαριστά τα φυσικά εξαρτήματα μιας εφαρμογής. Στο παρακάτω διάγραμμα παρουσιάζονται τα φυσικά εξαρτήματα που αφορούν το Codeigniter.



Σχήμα 15. Διάγραμμα εξαρτημάτων του προγράμματος

Με τη βοήθεια του αντικειμενοστρεφούς μοντέλου rationalrose και των διαγραμμάτων υμπαλοποσαφηνίζεται ο τρόπος υλοποίησης της εφαρμογής καθώς τμηματοποιείται και είναι πιο σωστά καθορισμένες οι λειτουργίες κάθε τμήματος. Αυτό έχει σαν αποτέλεσμα την αποφυγή λαθών αλλά και την ευκολότερη πρόληψη τους κατά τη διάρκεια της υλοποίησης.

Στη συνέχεια ακολουθεί η παρουσίαση της υλοποίησης της εφαρμογής.

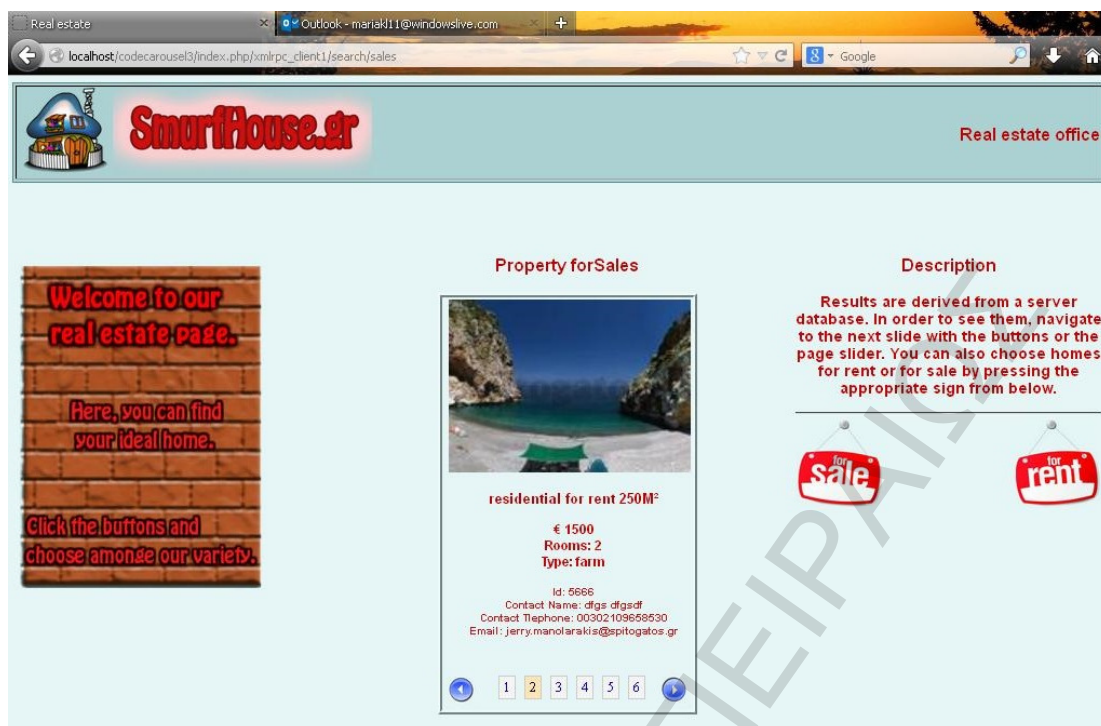
4.4 Παρουσίαση της εφαρμογής

Στο σημείο αυτό θα παρουσιαστεί η εφαρμογή στη μορφή που την βλέπει ο χρήστης.

Αρχική σελίδα εφαρμογής.

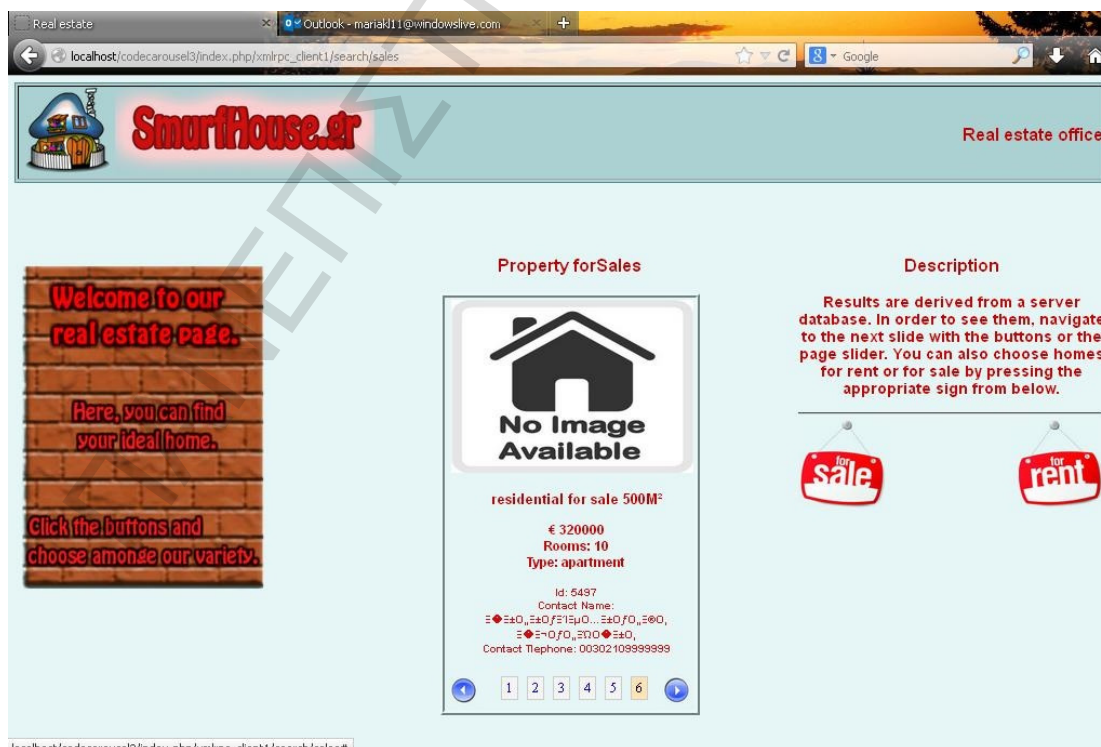
Στην αρχική σελίδα φορτώνεται το πρόσθετο παρουσίασης εικόνων όπου υπάρχουν σαν προκαθορισμένη επιλογή τα δεδομένα που αφορούν τις πωλήσεις ακινήτων. Αυτό είναι εμφανές από τη διεύθυνση url όπου αναγράφεται το sale αλλά και από τον τίτλο πάνω από το πρόσθετο.

Ο χρήστης στην αρχική σελίδα έχει τη δυνατότητα να δει τα δεδομένα που πληρούν τα κριτήρια που έχουν δοθεί στον controller.



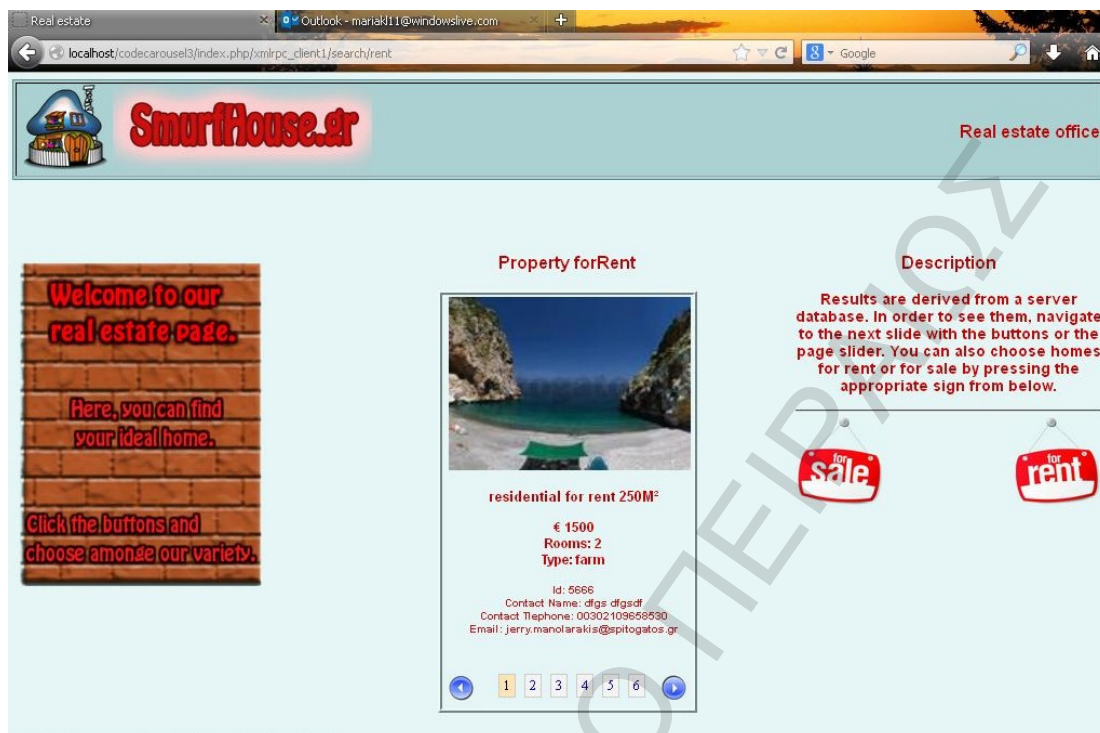
Σχήμα 16. Αρχική σελίδα της εφαρμογής

Η προβολή γίνεται μέσω της παρουσίασης εικόνων (slideshow) όπου σύμφωνα με τις ρυθμίσεις που έχουν γίνει στο plugin. Γίνεται αυτόματη εναλλαγή των εικόνων κάθε 7 δευτερόλεπτα. Παρόλα αυτά ο χρήστης έχει τη δυνατότητα να πλοηγηθεί με τη χρήση κουμπιών και αριθμημένων σελίδων που βρίσκονται στο κάτω μέρος του πρόσθετου. Τα κουμπιά δίνουν την επιλογή να δει την προηγούμενη ή την επόμενη παρουσίαση από αυτή που βρίσκεται, ενώ οι σελίδες δίνουν την επιλογή να μεταβεί στην παρουσίαση που επιθυμεί αυτόματα.



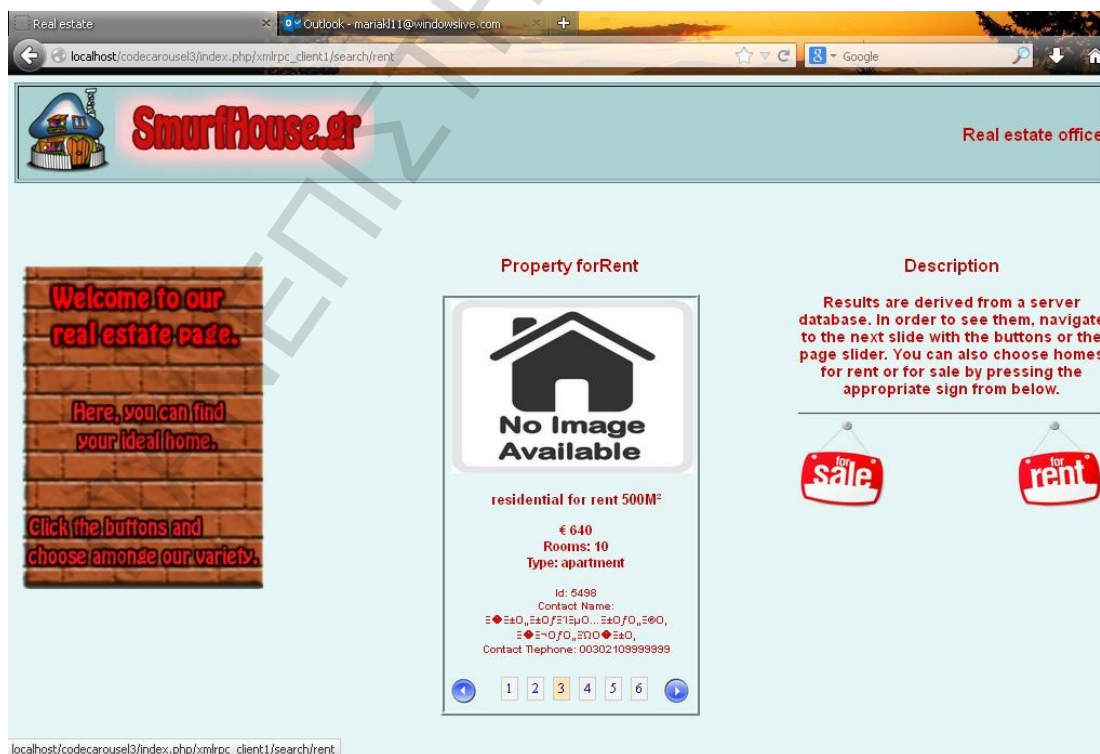
Σχήμα 17. Πλοήγηση του χρήστη με αλλαγή σελίδων

Στα δεξιά υπάρχουν δύο επιπλέον κουμπιά που δίνουν στο χρήστη τη δυνατότητα να επιλέξει ποια ακίνητα θα δει σύμφωνα με το αν είναι προς πώληση ή προς ενοικίαση. Με το πάτημα του forrentαυτόματα, αλλάζουν τα δεδομένα του πρόσθετου.



Σχήμα 18. Προβολή των δεδομένων προς ενοικίαση

Και σε αυτή την περίπτωση ο χρήστης μπορεί να πλοηγηθεί με τα κουμπιά και τις αριθμημένες σελίδες.



Σχήμα 19. Μετάβαση του χρήστη σε επόμενη παρουσίαση

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Η παρούσα μεταπτυχιακή διατριβή περιγράφει την ανάλυση, το σχεδιασμό και την υλοποίηση μιας εφαρμογής η οποία αναλαμβάνει την παρουσίαση των δεδομένων από μια κτηματομεσιτική διαδικτυακή πύλη. Η ανάκτηση των δεδομένων γίνεται με τη χρήση διαδικτυακών υπηρεσιών (webservices). Η παρουσίαση των δεδομένων γίνεται με τη χρήση του JQuerycycleplugin, από μια διαδικτυακή βάση. Βασικός στόχος είναι η εμφάνιση αποτελεσμάτων της αναζήτησης, σε πραγματικό χρόνο.

Απώτερος σκοπός ήταν η κατανόηση και η υλοποίηση μιας κλήσης απομακρυσμένης διαδικασίας.

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν νέες τεχνολογίες, όπως το πλαίσιο ανοιχτού λογισμικού Codeigniter που παρέχει ταχεία ανάπτυξη εφαρμογών ιστού και χρησιμοποιείται για την κατασκευή δυναμικών ιστοσελίδων με PHP. Ο στόχος του είναι να επιτρέψει στους προγραμματιστές να αναπτύξουν εφαρμογές πολύ πιο γρήγορα από ό, τι αν έγραφαν τον κώδικα από το μηδέν, παρέχοντας ένα πλούσιο σύνολο των βιβλιοθηκών για τα πιο κοινά των καθηκόντων που απαιτούνται, καθώς και ένα απλό περιβάλλον εργασίας.

Ο Codeigniter χρησιμοποιεί το πρότυπο ανάπτυξης MVC (model-view-controller).

Ενώ η προβολή και ο ελεγκτής είναι ένα απαραίτητα μέρη της ανάπτυξης στο πλαίσιο CodeIgniter, το μοντέλο είναι προαιρετικό.

Η συγγραφή κώδικα σε τμήματα παρέχει πολλές δυνατότητες, όπως η διατήρηση του κώδικα πιο ξεκάθαρου και ευανάγνωστου, η ευκολία διόρθωσης των λαθών σε κάθε τμήμα του κώδικα καθώς και η κατανόηση της λειτουργίας του κάθε τμήματος.

Για την δημιουργία της εφαρμογής χρησιμοποιήθηκαν οι γλώσσες Html, PHP και Xml-Rpc όπου με αυτό τον τρόπο υλοποιήθηκε η κλήση απομακρυσμένης διαδικασίας.

Μέσω της xml-rpc έγινε η σύνδεση με μια διαδικτυακή βάση δεδομένων, καθώς και η διαδικασία αποστολής αιτήματος από τον πελάτη και η αποστολής απάντησης από το διακομιστή. Με αυτόν τον τρόπο συλλέχθηκαν τα δεδομένα που χρησιμοποιήθηκαν στην εφαρμογή.

Τα αποτελέσματα ήταν εμφανή στο χρήστη μέσω του πρόσθετου JQuery. Το JQuerycycleplugin είναι ένα πρόσθετο προβολής σελίδων που υποστηρίζει πολλούς διαφορετικούς τύπους εφέ μετάβασης. Υποστηρίζει αυτόματο σταμάτημα, αυτόματη προσαρμογή, κουμπιά ενεργοποίησης κ.α. Μέσω της εισαγωγής των βιβλιοθηκών που παρέχει αλλά και της συγγραφής κώδικα σε JavaScript, διαμορφώθηκε ο τρόπος προβολής των σελίδων.

Πιο αναλυτικά, υπάρχει δυνατότητα να οριστεί ο τρόπος μετάβασης από τη μια σελίδα στην άλλη, ο χρόνος αυτόματης εναλλαγής σελίδων, ο αριθμός σελίδων αλλά και η ύπαρξη κουμπιών για εναλλακτική μετάβαση μέσω αυτών. Επιπρόσθετα μέσω του CSS, που είναι μια απλή γλώσσα που μας βοηθάει να ορίσουμε με σαφήνεια και ιδιαίτερη ευελιξία τον τρόπο με τον οποίο θα εμφανίζονται τα διάφορα στοιχεία στην ιστοσελίδα μας, γίνεται η μορφοποίηση του περιεχομένου που προβάλλεται στο χρήστη.

5.1 Περιορισμοί

Περιορισμός ήταν το γεγονός ότι δεν έγινε χρήση του πιο πρόσφατου πρωτοκόλλου JSONRPC - μια «ελαφριά» κλήση απομακρυσμένης διαδικασίας. Η προδιαγραφή (specification) ορίζει διάφορες δομές δεδομένων και κανόνες γύρω από την επεξεργασία τους. Είναι αγνωστικιστής μεταφοράς (transportagnostic) δηλαδή οι έννοιες μπορούν να χρησιμοποιηθούν στο πλαίσιο της ίδιας διαδικασίας, πάνω από υποδοχές (sockets), μέσω HTTP, ή να περνάει διάφορα μηνύματα σε περιβάλλοντα. Χρησιμοποιεί JSON (RFC 4627), για τη μορφή των δεδομένων αλλά του XMLRPC επειδή με το συγκεκριμένο πρωτόκολλο ήταν υλοποιημένο το webservice με το οποίο επικοινωνούσε η εφαρμογή. (Matt Morley of MPCMTechologies LLC, 2013. JSON-RPC 2.0 Specification)

Επιπλέον δεν χρησιμοποιήθηκαν οι τεχνολογίες AJAX / JSON. Η χρήση της τεχνικής AJAX επιτρέπει σε μια ιστοσελίδα να ανανεώνεται ασύγχρονα (asynchronously) 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. WLEARN)

Το JSON από την άλλη πλευρά είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Οι τεχνολογίες αυτές δεν ήταν απαραίτητες για την υλοποίηση της εφαρμογής. (Google, 2013, Εισαγωγή στο Json)

5.2 Βελτιώσεις

Η παρούσα μεταπτυχιακή διατριβή θα μπορούσε να επεκταθεί και να υποστεί βελτιώσεις. Αρχικά θα μπορούσε να υπάρχει η δυνατότητα εισόδου του εκάστοτε χρήστη στην εφαρμογή. Αυτό θα μπορούσε να επιτευχθεί με την βοήθεια της γλώσσας php. Αρχικά πρέπει να δημιουργηθεί μια βάση δεδομένων όπου θα κρατάει το όνομα χρήστη και των κωδικό εισόδου.

Στη συνέχεια, αφότου γίνει η σύνδεση της εφαρμογής με τη βάση, έρχεται το σημείο που ο χρήστης θα κάνει εγγραφή στην εφαρμογή. Τα νέα δεδομένα θα καταχωρούνται στη βάση και έτσι κάθε φορά που ο χρήστης θα κάνει login θα γίνεται έλεγχος μέσω της διαδικασίας select, για το αν τα στοιχεία είναι καταχωρημένα στη βάση. Αν είναι θα εμφανίζεται μήνυμα επιτυχούς σύνδεσης, αν όχι θα εμφανίζεται μήνυμα ανεπιτυχούς σύνδεσης. Επίσης θα υπάρχει επιλογή αποσύνδεσης.

Με αυτό τον τρόπο ο χρήστης που θα είναι συνδεδεμένος στην εφαρμογή θα μπορούσε να έχει πρόσθετες επιλογές. Μια επιλογή θα ήταν η προβολή συναφούς περιεχομένου με βάση τις πρόσφατες αναζητήσεις του. Επίσης θα μπορούσε να δημιουργεί μια λίστα με το περιεχόμενο που τον ενδιαφέρει με σκοπό την προβολή του αργότερα. Αυτή η επιλογή θα έκανε την εφαρμογή πιο αποδοτική και θα βοηθούσε τον χρήστη στη διαδικασία αναζήτησης περιεχομένου.

Μια επιπλέον λειτουργικότητα που θα μπορούσε να προστεθεί είναι η ύπαρξη επιπλέον κριτηρίων αναζήτησης από το χρήστη. Η υπάρχουσα εφαρμογή παρέχει τη δυνατότητα επιλογής ενοικίασης ή πώλησης ακινήτου στο χρήστη. Υπάρχουν δύο αντίστοιχα κουμπιά, που με το πάτημα τους εμφανίζονται είτε τα ακίνητα που είναι προς πώληση είτε προς ενοικίαση. Αντίστοιχα θα μπορούσε να γίνεται αναζήτηση με βάση κάποια επιπλέον κριτήρια, όπως τον τύπο ακινήτου, την τιμή και άλλα.

Ο τρόπος που θα μπορούσε να επιτευχθεί αυτό, όπως και στην περίπτωση της αναζήτησης με βάση την ενοικίαση ή την πώληση, θα ήταν με την βοήθεια μιας μεταβλητής που θα κρατούσε τα δεδομένα που επιστρέφει η κλήση απομακρυσμένης διαδικασίας. Στη συνέχεια θα δίνεται η επιλογή στο χρήστη να διαλέξει μέσα από μια λίστα δεδομένων του κάθε κριτηρίου (selectbox) και ανάλογα την επιλογή του θα εμφανίζονται τα δεδομένα που πληρούν τις προϋποθέσεις αναζήτησης.

Η διαδικασία κριτηρίων μας δίνει την επιλογή μεμονωμένης αλλά και σύνθετης αναζήτησης καθώς έχουμε την επιλογή να επιλέξουμε περισσότερα από ένα κριτήρια που επιθυμούμε. Με αυτόν τον τρόπο η εφαρμογή θα ήταν πιο λειτουργική και εύχρηστη καθώς ο χρήστης θα έβλεπε μόνο τα δεδομένα που τον ενδιαφέρουν σύμφωνα με την αναζήτηση του και έτσι θα ήταν πιο εύκολο να επιλέξει αυτό που πληροί τις ανάγκες του.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Πανεπιστήμιο Θεσσαλίας, 1997, Η ιστορία του Internet [online]
Available at :<<http://www2.uth.gr/main/help/help-desk/internet/internet3.html>>
[Accessed 4 October 2012].

K. Cooper, O. de Bruijn, R. Spence and M.Witkowski,2006. A Comparison of Static and Moving Presentation Modes for Image Collections [pdf]

Available at:

<http://www.academia.edu/2465119/A_comparison_of_static_and_moving_presentation_modes_for_image_collections>

United Kingdom: Academia Education

[Accessed 4 October 2012].

M. Allman, 2003. An Evaluation of XML-RPC* [pdf]

Available at: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.6666>> CiteSeerX

[Accessed 29 January 2013]

A.S. Duarte, J.H. Santos, H. Fernandes, A. Neto, T. Pereira and C.A.F. Varandas, 2008. FireCalc: An XML-based framework for distributed data analysis [pdf]

Available at: <<http://www.sciencedirect.com/science/article/pii/S0920379607005017>>

Elsevier

[Accessed 3 October 2012]

E. Axel Larsson and R. Sprague, 2004. eXtreme Deployment: Distributing and Configuring 450 Student Laptops in Five Hours [pdf]

Available at: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.83.7624>>

CiteSeerX

[Accessed 3 October 2012]

C. A. Ardagna, E. Damiani, S. De Capitani di Vimercati, and P. Samarati, 2006, Elsevier Article: A Web Service Architecture for Enforcing Access Control Policies [pdf]

Available at: <<http://www.sciencedirect.com/science/article/pii/S1571066105052102>>

ScienceDirect

[Accessed 14 January 2013]

I. Sommerville, 2011, Sommerville : Software engineering [pdf]

Available at: <[http://www.alexu.edu.eg/Courses/CS_Advanced%20S.E/Software%20Engineering%20\(9th%20Edition\)%20by%20Ian%20Sommerville.pdf](http://www.alexu.edu.eg/Courses/CS_Advanced%20S.E/Software%20Engineering%20(9th%20Edition)%20by%20Ian%20Sommerville.pdf)>

[Accessed 26 January 2013]

W. Heijstek , 2010, Leiden Institute of Advanced Computer Science: The Rational Unified Process [pdf]

Available at:

<http://www.liacs.nl/~chaudron/se2010/2010_09_14_SE_RUP_final_WH.pdf>

[Accessed 26 January 2013]

ΒΙΚΙΠΑΙΔΕΙΑ , 2013, Βικιπαίδεια: Βάση δεδομένων [online]

Available at:

<http://el.wikipedia.org/wiki/%CE%92%CE%AC%CF%83%CE%B7_%CE%B4%CE%B5%CE%B4%CE%BF%CE%BC%CE%AD%CE%BD%CF%89%CE%BD>

[Accessed 05 February 2013]

ΕΡΓΑΣΤΗΡΙΟ ΕΦΑΡΜΟΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΑ ΜΜΕ , 2004, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης: Τι είναι η html [online]

Available at:

<<http://pacific.jour.auth.gr/html/>>

[Accessed 05 February 2013]

ΒΙΚΙΠΑΙΔΕΙΑ , 2013, Βικιπαίδεια: CSS [online]

Available at:

<<http://el.wikipedia.org/wiki/CSS>>

[Accessed 05 February 2013]

ΒΙΚΙΠΑΙΔΕΙΑ , 2013, Βικιπαίδεια: PHP [online]

Availableat:
<<http://el.wikipedia.org/wiki/PHP>>
[Accessed 07 February 2013]

ΚΕΝΤΡΟ ΠΛΗ.ΝΕ.Τ.ΠΕΡ. ΕΝΟΤΗΤΑΣ ΦΛΩΡΙΝΑΣ , 2005, ΚΕΝΤΡΟ ΠΛΗ.ΝΕ.Τ.
Ν. ΦΛΩΡΙΝΑΣ: ΗΓΛΩΣΣΑ Προγραμματισμού PHP Αναλυτικά [online]
Availableat:
<<http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-Php-Analytical.html>>
[Accessed 07 February 2013]

ΒΙΚΙΠΑΙΔΕΙΑ , 2013, Βικιπαίδεια: JavaScript [online]
Availableat:
<<http://el.wikipedia.org/wiki/JavaScript>>
[Accessed 10 February 2013]

iDesigner, 2013, iDesigner: Εισαγωγή στην jQuery [online]
Availableat:
<<http://www.idesigner.gr/%CE%B5%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE-%CF%83%CF%84%CE%B7%CE%BD-jquery/>>
[Accessed 10 February 2013]

ΒΙΚΙΠΑΙΔΕΙΑ , 2012, Βικιπαίδεια: Plug-in [online]
Availableat:
<<http://el.wikipedia.org/wiki/Plug-in>>
[Accessed 20 February 2013]
GitHub, Inc, 2013, Mike Alsup: jQuery Plugins (by malsup [online]
Availableat:
<<http://jquery.malsup.com/cycle/>>
[Accessed 20 February 2013]

Computerman, Inc, 2010-2013, Computerman: Greek codeigniter framework [online]
Availableat:
<<http://tutorials.computerman.gr/eisagogi-sto-codeigniter-framework/>>
[Accessed 20 February 2013]

dplab, Inc, 2010-2013, dplab: XML-RPC και Java [online]
Availableat:
<<http://pdplab.it.uom.gr/teaching/sunjava/xml-rpc.html>>
[Accessed 20 February 2013]

Wikipedia, 2013, Wikipedia: Web service [online]
Availableat:
<https://en.wikipedia.org/wiki/Web_service>
[Accessed 20 February 2013]

W.Heijstek , 2010, Leiden Institute of Advanced Computer Science: The Rational Unified Process [pdf]
Availableat:
<http://www.liacs.nl/~chaudron/se2010/2010_09_14_SE_RUP_final_WH.pdf>
[Accessed 26 January 2013]

M. Morley of MPCM Technologies LLC, 2013. JSON-RPC 2.0 Specification.[online] Available at: <<http://www.jsonrpc.org/specification>>
[Accessed 16 March 2013].)

A.Καργιοφύλλης, 2013. WLEARN. [online] Available at: <<http://www.wlearn.gr/index.php/ajax-intro2>>

[Accessed 16 March 2013].

Google, Εισαγωγή στοJson, 2013.[online] Available at: <<http://www.json.org/json-el.html>>
[Accessed 16 March 2013]

ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ

Αρχείο Controller

```
<?php
class Xmlrpc_client1 extends CI_Controller {

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

        $this->load->helper(array('form', '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'=>'6',
            'latitudeLow'=> '0.123457',
            'latitudeHigh'=>'30.123457',
            'priceLow'=>'10',
            'priceHigh'=>'1000000',
            'livingAreaLow'=>'10',
            'livingAreaHigh'=>'50000',
            'offset'=>'0',
            'region'=>'100',
            'longitudeLow'=>'0.678902',
            'longitudeHigh'=>'20.678902',
            'sortBy'=>'rankingScore',
            'sortOrder'=>'desc',

        );

        $request = array (
            array( "d3c10e1840823cb941c0a1924", 'string'),
            array( $struct, 'struct'),
        );
    }
}
```

```

    );

    $this->xmlrpc->request($request);
$image_url='localhost/codecarousel3/photos1/noimage.jpg';
    if ( ! $this->xmlrpc->send_request()
    {
        echo $this->xmlrpc->display_error();

    }

    else
    {

        $response = $this->xmlrpc->display_response();
        //echo "<pre>";
        //print_r($response["listings"]);
        //die();
        foreach ( $response["listings"] as $listingKey => $listingValue)
        {

            $this->data["listings"][$listingKey]["key"] = $listingKey;

            if ($listingValue['mainImageURL'] != "")
            {$this->data["listings"][$listingKey]["photoUrl"] =
$image_url.$listingValue['mainImageURL'];
            }
            else{
                $this->data["listings"][$listingKey]["photoUrl"] = base_url() .
"photos1/noimage.jpg";
            }
            $this->data["listings"][$listingKey]["categ"] =
$image_url.$listingValue['category'];
            $this->data["listings"][$listingKey]["litype"] =
$image_url.$listingValue['listingType'];
            $this->data["listings"][$listingKey]["livarea"] =
$image_url.$listingValue['livingArea'];
            $this->data["listings"][$listingKey]["pric"] =
$image_url.$listingValue['price'];
            $this->data["listings"][$listingKey]["room"] =
$image_url.$listingValue['rooms'];
            $this->data["listings"][$listingKey]["proptype"] =
$image_url.$listingValue['propertyType'];
            $this->data["listings"][$listingKey]["id"] =
$image_url.$listingValue['listingID'];
            $this->data["listings"][$listingKey]["name"] =
$image_url.$listingValue['contactName'];
            $this->data["listings"][$listingKey]["tele"] =
$image_url.$listingValue['contactTelephone'];
            $this->data["listings"][$listingKey]["mail"] =
$image_url.$listingValue['contactEmail'];
        }

        $this->data["listingType"] = ucfirst($listingType);

        $this->load->view("example1",$this->data);
        //echo "<pre>";
        //print_r($this->data);
    }
}

```

```
        //echo "</pre>";
    }
}
}
```

Αρχείο View

```
<!DOCTYPE html>

<html>

<head class="head">

<meta http-equiv="content-type" content="text/html; charset=Windows-1253" />

<meta http-equiv="Content-Style-Type" content="text/css" />

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

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

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

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

<title> Real estate</title>

<table border="1" width='100%' valign="top" height="100"
style="border-width: 2px;
        border-style:solid ;
        border-color:#528E90;
        border-collapse: separate;
        background-color: #ACD1D2;">

<tr>

<td>
```

```
<imgsrc="<?php echo $baseUrl;?>photos1/SmurfHouse.jpg" class="smurf" />

<imgsrc="<?php echo $baseUrl;?>photos1/logo.jpg" class="stroumf" />

<h2>
Real estate office</h2>

</td>

</tr>

</table>

</head>

<body onload="initialize()">

<table border="0"
style="border-width: 0px;
border-style: none;
border-collapse: separate;
background-color: #E6F6F6;">

<tr>

<td width='30%' valign="top" height="50">

<br>

<br>

<imgsrc="<?php echo $baseUrl;?>photos1/wellcome1.jpg" class="wellcome" />

</td>

<td width='40%' valign="top" height="50" >

<div class="main">
```

```
<h3 style="font-size:20px; color: #c00000; padding-top:60px;">Property for<?php echo
$listingType ?>

</h3>

<div id="carousel" class="carousel">

<div id="slideshow" class="pics" >

<?phpforeach ($listings as $listing):?>

<div class="a" onmouseover="this.style.background='#FAFAD2';"
onmouseout="this.style.background='#E6F6F6';">

    <imgsrc="<?php echo $listing["photoUrl"];?>" class="thumbnail"/>

    <p><?php echo $listing["categ"]; ?>

    <?php echo $listing["litype"]; ?>

    <?php echo $listing["livarea"].'M&sup2;'; ?></p>

    <h3><?php echo '&euro; '.$listing["pric"]; ?><br>

    <?php echo 'Rooms: '.$listing["room"]; ?><br>

<?php echo 'Type: '.$listing["proptype"]; ?></h3>

    <h4><?php echo 'Id: '.$listing["id"]; ?><br>

<?php echo 'Contact Name: '.$listing["name"]; ?><br>

    <?php echo 'Contact Telephone: '.$listing["tele"]; ?><br>

    <?php echo 'Email: '.$listing["mail"]; ?></h4>

</div>

<?phpendforeach;?>

</div>

<div id="demos">
```

```
<div>
<div id="prev">
    <a href name="prev" value="Prev" title="previous">
        <imgsrc= "<?php echo $baseUrl;?>photos1/back-button1.jpg" width="45"
height="45"/>
    </a>
</div>
<div id="next">
    <a href name="next" value="Next" title="next">
        <imgsrc= "<?php echo $baseUrl;?>photos1/next_button1.jpg" width="45"
height="45"/>
    </a>
</div>
    <ul id="nav"></ul>
    <br>
</div>
</div>
</div>
</div>
</td>
<td width='30%' valign="top" align="right" height="50" style="padding-left:20px;">
<h3 style="font-size:20px; color: #c00000; padding-top:60px;">Description</h3>
<h3 style="font-size:17px; text-align:center; top:1px;">
Results are derived
```


from a server database.

In order to see them,

navigate to the next

slide with the buttons

or the page slider.

You can also choose homes

for rent or for sale

by pressing the appropriate

sign from below. </h3>

```
<hr>
```

```
<div class="sale">
```

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

```
<img id="img1"
```

```
src= "<?php echo $baseUrl; ?>photos1/sale.jpg" width="100" height="100"/>
```

```
</a>
```

```
</div>
```

```
<div class="rent">
```

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

```
<img id="img2"
```

```
src= "<?php echo $baseUrl; ?>photos1/rent.jpg" width="100" height="100"/>
```

```
        </a>
    </div>
</td>
</tr>
</table>
</body>
</html>
```

ΑρχείοJquery

```
// slideshow.js

$(function() {

    $('#slideshow').cycle({
    fx: 'scrollHorz',
    timeout: 7000,
    prev: '#prev',
    next: '#next',
    pager: '#nav',
    pagerAnchorBuilder: pagerFactory
    });

    function pagerFactory(idx, slide) {
    var s = idx>5? ' style="display:none"' : "";
```

```
return '<li'+s+'><a href="#">'+(idx+1)+'</a></li>';  
  
};  
  
});
```

Αρχείο CSS

```
#nav {padding-left:60px; padding-top:10px}  
  
#nav li { float: left; list-style: none}  
  
#nav a { margin: 5px; padding: 3px 5px; border: 2px solid #ccc; background: #F5F5F5;  
text-decoration: none }  
  
#navli.activeSlide a { background:#FFE4B5; color: black }  
  
#nav a:focus { outline: none; }  
  
body  
{  
background-color:#E6F6F6;  
}  
  
.main  
{  
  
margin:auto;  
  
padding-left:20px;  
  
width:300px;  
  
height: 450px;
```

```
}  
  
.carousel  
{  
border-width: 5px ;  
border-style: ridge;  
border-color:#CBEEEE;  
}  
  
.pics  
{  
height: 415px;  
}  
  
.smurf  
{  
width:100px;  
height:100px;  
padding-left:5px;  
float:left;  
}
```

```
.stroumf  
{  
width:300px;  
height:100px;  
padding-left:5px;  
}
```

```
.wellcome  
{  
width:280px;  
height:380px;  
padding-top:50px;  
padding-left:5px;  
float:left;  
}
```

```
.sale  
{  
  
float:left;  
}
```

```
.rent
{
float:right;
}

.thumbnail
{
padding-left:5px;
width:280px;
height:200px;
}

#prev
{
text-decoration:none;
float:left;
}
```

```
#next
{
text-decoration:none;
float:right;
}

.a
{
width:300px;
float:center;
padding:2px;
}

p
{
font-family:Arial;
font-weight:bold;
font-size:16px;
color:#c00000;
text-align:center;
}
```

```
h2
{
padding-top:25px;
font-family:Arial;
font-size:20px;
color: #c00000;
float:right;
}

h3
{
font-family:Arial;
font-size:15px;
color: #CA0202;
text-align:center;
}

h4
{
font-family:Arial;
font-size:13px;
color: #D10C0C;
text-align:center;
```



```
}
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ