



Πανεπιστήμιο Θεσσαλίας

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

Ηλεκτρονική Διακυβέρνηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΥΡΙΑΚΟΥ ΚΑΤΣΑΡΗ

Επιβλέπουσα Καθηγήτρια : Α. Δασκαλοπούλου

Βόλος, 2007

.....
ΚΥΡΙΑΚΟΣ Μ. ΚΑΤΣΑΡΗΣ

Διπλωματούχος Μηχανικός Η/Υ, Τηλεπικοινωνιών και Δικτύων Π.Θ

Copyright © Κυριάκος Κατσαρής, 2007

Με επιφύλαξη παντός δικαιώματος. All rights reserved

Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης.

Οι απόψεις και τα προσωπικά συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και βαρύνουν αποκλειστικά και μόνο αυτόν.

Πρόλογος

Η διπλωματική εργασία που τεκμηριώνεται και αναλύεται στο παρόν έγγραφο έχει ως σκοπό την σχεδίαση και ανάπτυξη μιας Πύλης (Portal) εξυπηρέτησης πολιτών συμπεριλαμβανομένης όλων αυτών των στοιχείων που θα την καταστήσουν λειτουργική και αποτελεσματική. Επίσης υλοποιούνται και κάποιες υπηρεσίες ως παραδείγματα υπηρεσιών Ηλεκτρονικής Διακυβέρνησης. Η υλοποίηση όλων αυτών έγινε με την πλατφόρμα ανάπτυξης που προσφέρει η JAVA.

Ολοκληρώνοντας την διπλωματική εργασία θα ήθελα να ευχαριστήσω την κ.Δασκαλοπούλου για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον ζήτημα όπως είναι η Ηλεκτρονική Διακυβέρνηση και τα web services γενικότερα καθώς και για τις ιδέες και τα σχόλια της σχετικά με την ανάπτυξη της Πύλης.

Επίσης νιώθω την ανάγκη να ευχαριστήσω την οικογένεια μου για την διαρκή τους συμπαράσταση όλα αυτά τα χρόνια της φοίτησης μου στο Πολυτεχνείο καθώς και τους φίλους μου Βασίλη, Δημήτρη, Ηλία και Ορέστη για την ψυχολογική τους υποστήριξη όποτε και αν την χρειάστηκα .

Κυριάκος Κατσαρής

Πίνακας περιεχομένων

1 Εισαγωγή	8
1.1 Αντικείμενο Διπλωματικής	8
2 Περί της Ηλεκτρονικής Διακυβέρνησης	11
2.1 Περί Πύλης (Portal) Ηλεκτρονικής Διακυβέρνησης	13
2.2 Περιγραφή ιδιοτήτων Πύλης Ηλεκτρονικής Διακυβέρνησης.....	14
2.2.1 Χρήστες	14
2.2.2 Χαρακτηριστικά μιας Πύλης.....	15
2.2.3 Παραδείγματα portals Ηλεκτρονικής Διακυβέρνησης.....	17
2.3 Διαλειτουργικότητα	18
2.4 Υπηρεσίες Ιστού - Web Services	19
2.5 Οι τεχνολογίες των Web Services	21
2.6 Τεχνολογικά Πρότυπα και Μοντέλα.....	21
2.6.1 SOAP.....	21
2.6.2 WSDL (Web Services Description Language)	24
2.6.3 UDDI (Universal Description, Definition and Integration)	26
2.6.4 WSSD (Web Services Deployment Descriptor)	27
2.6.5 MVC (Model –View – Controller) Framework	27
2.7 Πλατφόρμες Ανάπτυξης Web based λογισμικού	27
3 Ανάλυση και Σχεδίαση συστήματος	29
3.1 Στόχοι.....	29
3.2 Όψεις.....	29
3.2.1 Υπηρεσιακή όψη	30
3.2.2 Πλατφορμική όψη	30
3.3 Επικοινωνία πρωτοκόλλου SOAP.....	32
3.4 Σχεδιαστικές αποφάσεις.....	32
3.5 Σχεδιασμός ροών ελέγχου και δεδομένων.....	33
3.5.1 Εισαγωγικά.....	33
3.5.2 Κλήση υπηρεσιών οργανισμών από το ΠΕΠ	34
3.5.3 Περιγραφή αρχιτεκτονικής εφαρμογών Οργανισμών και ΠΕΠ	41
3.5.3.1 Βάση δεδομένων Υπουργείου δημόσιας Τάξεως - Αστυνομία	41
3.5.3.2 Βάση δεδομένων Οργανισμού ΔΕΗ	42
3.5.3.2 Βάση δεδομένων των Δήμων	43
3.5.4 Δικτυακές υπηρεσίες Οργανισμών.	47
3.5.4.1 Δικτυακές υπηρεσίες υπουργείου δημοσίας τάξης – Αστυνομία.....	47
3.5.4.2 Δικτυακές υπηρεσίες Οργανισμού ΔΕΗ.....	48
3.5.4.3 Δικτυακές υπηρεσίες Δήμων.....	49
3.5.5 Περιγραφή αρχιτεκτονικής εφαρμογής ΠΕΠ	50
3.6 Γενική Περιγραφή της Πύλης Εξυπηρέτησης Πολιτών.....	51
3.6.1 Χρήστες	52
3.6.1.1 Χρήστες και Λειτουργίες του ΠΕΠ	52
3.6.1.2 Χαρακτηριστικές οθόνες πρόσβασης του πολίτη	61
3.6.1.3 Χαρακτηριστικές οθόνες πρόσβασης του διαχειριστή	66

4 Υλοποίηση	71
4.1 Σύστημα υλοποίησης της εφαρμογής.....	71
4.2 Πλατφόρμες και προγραμματιστικά εργαλεία	71
4.3 Λεπτομέρειες υλοποίησης	76
4.4 Τεχνικές λεπτομέρειες εφαρμογής	77
4.5 Δομή Απαραίτητων καταλόγων	79
4.6 Εγκατάσταση της εφαρμογής	81
5 Επίλογος	82
5.1 Σύνοψη	82
5.2 Συμπεράσματα	82
5.3 Μελλοντικές Επεκτάσεις	83
5.4 Προσωπική άποψη	84
6. Αναφορές	85
Παράρτημα Α.....	87
Παράρτημα Β.....	100

Ευρετήριο Εικόνων και Σχημάτων

Σχήμα 2.1 - Γραφική απεικόνιση των κατηγοριών	12
Σχήμα 2.2 - Μορφή e government.....	13
Σχήμα 2.3 - Χρήστες Πύλης	14
Σχήμα 2.4 - Αρχιτεκτονική e-GIF (πηγή e gif).....	19
Σχήμα 2.5 - Client – Server μορφή	19
Σχήμα 2.6 - Διαδικασία παροχής και εύρεσης υπηρεσιών.....	20
Σχήμα 2.7 - WSDL μορφή.....	24
Σχήμα 2.8 - Απεικόνιση n –tier μοντέλου (πηγή e gov.gr).....	28
Σχήμα 3.1 - Υπηρεσιακή όψη.....	30
Σχήμα 3.2 - Πλατφορμική όψη.....	31
Εικόνα 3.3- Οθόνη Χρηστικής όψης.....	33
Σχήμα 3.4 - Ροή Μεταδημότευσης.....	36
Σχήμα 3.5 - Ανοιγμα Οικογενειακής μερίδας.....	40
Σχήμα 3.6- Μεταφορά δεδομένων από Βάση δεδομένων στην Δικτυακή υπηρεσία.....	41
Σχήμα 3.7 - Διάγραμμα Βάσης δεδομένων Αστυνομικού Τμήματος	42
Σχήμα 3.8 - Διάγραμμα Βάσης δεδομένων Οργανισμού ΔΕΗ	43
Σχήμα 3.9 - Διάγραμμα Βάσης δεδομένων στοιχείων πολιτών σε Δήμους	44
Σχήμα 3.10 - Διάγραμμα Βάσης δεδομένων πρώην Δημοτών.....	45
Σχήμα 3.11 – Διάγραμμα Βάσης δεδομένων γάμων	46
Σχήμα 3.12 –Συνεργατικό Διάγραμμα –Αστυνομία.....	47
Σχήμα 3.13–Διαγράμματα Βάσεων δεδομένων,Κωδικός Συναλλαγής,Λογ. Χρήστη... ..	51
Σχήμα 3.14 –Περιπτώσεις Χρήσης του Πολίτη στην πύλη.....	52
Σχήμα 3.15 –Διαδικασία Login.....	53
Σχήμα 3.16 –Παράδειγμα απογορευμένης πρόσβασης.....	54
Σχήμα 3.17 –Διαδικασία Εγγραφής Χρήστη.....	55
Σχήμα 3.18 –Διαδικασία Υπηρεσιών.....	56
Σχήμα 3.19 –Διαδικασία Ψηφοφορίας.....	57
Σχήμα 3.20 –Περιπτώσεις χρήστη Διαχειριστή Συστήματος.....	58
Σχήμα 3.21 –Προσβάσιμες οθόνες απο Πολίτη.....	61
Εικόνα 3.22 - Οθόνη Αρχική Σελίδα Πύλης.....	62
Εικόνα 3.23 - Οθόνη Διαθέσιμων Υπηρεσιών	64
Εικόνα 3.24 – Οθόνη Register χρήστη	66
Σχήμα 3.25 – Προσβάσιμες οθόνες απο τον Διαχειριστή.....	67
Εικόνα 3.26 - Οθόνη Αρχική σελίδα διαχείρισης.....	68
Εικόνα 3.27 - Οθόνη Ενδεικτικών σελίδων Διαχείρισης.....	70
Σχήμα 4.1 – Στοιχεία που αποτελούν το EAR αρχείο.....	72
Παράρτημα Α	
Σχήμα 1 – Διάγραμμα Κλάσεων.....	110
Παράρτημα Β	
Οθόνες Περίπτωσης διαδικασιών απλού χρήστη.....	100
Οθόνες Περίπτωσης διαδικασιών Διαχειριστή συστήματος.....	108

Ευρετήριο Πινάκων

Πίνακας 2.1 μορφή SOAP μηνύματος	22
Πίνακας 2.2 αποστολή SOAP μηνύματος μέσω POST	23
Πίνακας 2.3 έκφραση λαθών στα SOAP μηνύματα	23
Πίνακας 2.4 WSDL	25
Πίνακας 2.5 UDDI	26
Πίνακας 1 – EJBs Αστυνομία	42
Πίνακας 2 – EJBs ΔΕΗ	43
Πίνακας 3 – EJBs , στοιχεία δημοτών	44
Πίνακας 4 – EJBs , στοιχεία παλιών δημοτών	45
Πίνακας 5 – EJBs , δήλωση γάμου	46

1 Εισαγωγή

1.1 Αντικείμενο Διπλωματικής

Η εξάπλωση του διαδικτύου τα τελευταία χρόνια οδήγησε σταδιακά τους φορείς της πολιτείας να παρέχουν ηλεκτρονικές υπηρεσίες προς όφελος των πολιτών και των επιχειρήσεων με την μορφή της Ηλεκτρονικής Διακυβέρνησης. Για την επίτευξη αυτής, απαιτούνται προηγμένες υπηρεσίες πλήρως διαδραστικές και εύχρηστες προς τους χρήστες και όχι μια απλή παροχή πληροφοριών. Η Ηλεκτρονική Διακυβέρνηση ανάλογα με το ποιός παρέχει την υπηρεσία σε ποιόν διακρίνεται σε Κυβέρνηση προς Πολίτη (G2C), Κυβέρνηση προς Επιχείρηση (G2B) και Κυβέρνηση προς Κυβέρνηση (G2G). Στην παρούσα διπλωματική θα εστιάσουμε στην Κυβέρνηση προς Πολίτη (G2C).

Για παράδειγμα, ένας πολίτης, προκειμένου να πραγματοποιήσει κάποια από τις υπηρεσίες-διαδικασίες που παρέχονται από τα ΚΕΠ πρέπει να προσκομίσει μια πληθώρα πιστοποιητικών, να συμπληρώσει τα απαιτούμενα έγγραφα και τέλος να περιμένει κάποιο χρονικό διάστημα ώστε να βεβαιωθεί η επιτυχής ή αποτυχή έκβαση της διαδικασίας. Αυτό οφείλεται κυρίως στην γραφειοκρατική δομή και οργάνωση που υπάρχει στις διάφορες κυβερνητικές και αυτοδιοικήτες υπηρεσίες.

Αναλυτικότερα :

- Σε πολλούς δήμους παρατηρείται το φαινόμενο της χειροκίνητης «διαχείρισης» πληροφοριών, λόγω έλλειψης πληροφοριακών συστημάτων οδηγώντας έτσι σε αργούς ρυθμούς ενημέρωσης αυτών και γενικότερα στην αργή πραγματοποίηση διαφόρων πράξεων.
- Πολλές φορές παρατηρείται η ύπαρξη ίδιων δεδομένων σε διαφορετικά έγγραφα αυξάνοντας έτσι την πιθανότητα ασυνέπειας κάποιου από αυτά κατά την τροποποίηση.
- Τέλος και πολύ σημαντικό αποτελεί η ανομοιογένεια μεταξύ αρχείων που κρατούν οι διάφοροι δήμοι για τις πληροφορίες τους. Κάποιοι δήμοι πιθανώς να έχουν ένα ολοκληρωμένο σύστημα διαχείρισης δεδομένων, άλλοι να κρατούν πληροφορίες σε αρχεία πχ τύπου Excel, Word τα οποία μπορεί και να βρίσκονται σε διαφορετικό λειτουργικό σύστημα, υπό την χρήση διαφορετικών προτύπων κτλ.

Με βάση αυτά είναι επιτακτική η ανάγκη ενός μηχανισμού ο οποίος θα διασυνδέσει τις διαφορετικές εφαρμογές σε ένα τέτοιο επίπεδο το οποίο θεωρεί τις λεπτομέρειες υλοποίησης των τοπικών πληροφοριακών συστημάτων σαν ένα μαύρο κουτί, λύνοντας έτσι τα προβλήματα ασυμβατότητας και διαλειτουργικότητας.

Η παρούσα διπλωματική εργασία έχει σαν στόχο την δημιουργία μίας Πύλης (portal) Εξυπηρέτησης των πολιτών, η οποία θα παρέχει εκτός από στατικές πληροφορίες (διατάξεις νόμων, περιγραφή διάφορων διαδικασιών, links υπουργείων, fora, δημοσκοπήση κτλ) επιπλέον την δυνατότητα πραγματοποίησης κάποιων υπηρεσιών, που κανονικά απαιτούν προσωπική παρουσία στα ΚΕΠ όπως αναφέρθηκε προηγουμένως. Οι υπηρεσίες που είναι διαθέσιμες στους πολίτες είναι μεταδημότευση και η καταχώρηση γάμου (άνοιγμα οικογενειακής μερίδας) στους δήμους.

Συγκεκριμένα οι πολίτες αφού πιστοποιηθούν από το σύστημα μπορούν να συμπληρώσουν τις αντίστοιχες φόρμες των υπηρεσιών και στην συνέχεια να ενημερώνονται άμεσα για την έκβαση της ενέργειας τους.

Σε περίπτωση επιτυχίας αποστέλλεται με e mail ο προσωπικός κωδικός συναλλαγής που αναγνωρίζει μοναδικά τον κάθε πολίτη καθώς και ο νέος αριθμός μερίδας του στον

δήμο.Επίσης δημιουργείται δυναμικά και ένα αρχείο pdf με τα ίδια δεδομένα διαθέσιμο για downloading το οποίο και πιστοποιεί την πράξη που πραγματοποιήθηκε.

Τέλος παρέχεται η δυνατότητα σε εξειδικευμένο προσωπικό(admin) να παραμετροποιεί και να παρακολουθεί τις υπηρεσίες και την εφαρμογή στο σύνολο της. Συγκεκριμένα υπάρχει η δυνατότητα διαχείρισης στις υπηρεσίες, στα μέλη της πύλης, στους κωδικούς συναλλαγής, του e mail, των logs και της δημοσκόπησης.

Αξίζει να αναφερθεί ό,τι στο θέμα της ασφάλειας της αποστολής δεδομένων τόσο από τους πολίτες όσο και από τους διαχειριστές συστήματος, χρησιμοποιείται το πρωτόκολλο SSL ενώ παράλληλα γίνεται καταγραφή των συναλλαγών σε logs ώστε να εξασφαλίζεται η «μη αποκύρξη».

Η υφιστάμενη ανταλλαγή δεδομένων ανάμεσα στους απαραίτητους φορείς που συμμετέχουν για την πραγμάτωση των υπηρεσιών αυτοματοποιείται με την βοήθεια των XML/SOAP και web services σε επίπεδο τεχνολογιών και της Service Oriented Architecture σε επίπεδο σχεδιασμού συστημάτων. Κάθε υπηρεσία που διατίθεται στους πολίτες αποτελεί μια ξεχωριστή ροή εργασίας η οποία αναλαμβάνει την επικοινωνία τόσο με τα web services όσο και με την Πύλη για την ορθή προβολή των αποτελεσμάτων.

Προδιαγραφές και απαιτήσεις

Οι προδιαγραφές και απαιτήσεις που αφορούν την ανάπτυξη της Πύλης Ηλεκτρονικής Διακυβέρνησης προσανατολισμένη στον Πολίτη, αποτελούν ένα υποσύνολο του εγγράφου ορισμού των απαιτήσεων και προδιαγραφών όπως ορίστηκε από τον δήμο Βόλου στα πλαίσια της προκύρξης έργου Ηλεκτρονικής Διακυβέρνησης (Φεβρουάριος 2007)

Θεσμικό πλαίσιο

Η αυτοματοποίηση των υπηρεσιών που υλοποιούνται από το σύστημα, ακολουθούν στο μέγιστο δυνατό βαθμό τα πρότυπα και τις ροές εργασίας που ορίζονται από τις διατάξεις λειτουργίας των ΚΕΠ και κατά επέκταση του υπουργείου Εσωτερικών στο οποίο ανήκουν διοικητικά.

Ισχύοντα τεχνολογικά πρότυπα και οδηγίες

Το σύστημα συμμορφώνεται με τις τεχνικές πολιτικές και τα πρότυπα που ορίζονται από το Ελληνικό Πλαίσιο Διαλειτουργικότητας Ηλεκτρονικής Διακυβέρνησης (e-Government Interoperability Framework). Παράλληλα, ο σχεδιασμός και η υλοποίηση των διαδικτυακών υπηρεσιών που ανταποκρίνονται στις απαιτήσεις του φυσικού προβλήματος βρίσκονται σε συνέπεια, στο μέγιστο δυνατό βαθμό με τις εξελίξεις στο σχεδιασμό πληροφοριακών συστημάτων Ηλεκτρονικής Διακυβέρνησης της Βρετανικής Κυβέρνησης, το οποίο και αποτελεί πρότυπο για την Ευρωπαϊκή Ένωση(e gif).

Τα συγκεκριμένα πρότυπα έχουν οριστεί από τον οργανισμό W3C (World Wide Web Consortium) και είναι «ανοικτά» δηλαδή εξασφαλίζουν την χρήση των διαδικτυακών υπηρεσιών ανεξάρτητα από την πλατφόρμα ανάπτυξης του πληροφοριακού συστήματος κάθε δήμου ή Οργανισμού γενικότερα

Θέματα υλοποίησης

Για την υλοποίηση της εφαρμογής στο σύνολο της χρησιμοποιήθηκε open source λογισμικό και τεχνολογίες που σχετίζονται με την Java (J2EE). Συγκεκριμένα ακολουθήθηκε το MVC μοντέλο ώστε να εξασφαλιστεί ο διαχωρισμός της προβολής (JSP σελίδες) από την επιχειρηματική λογική (servlets). Για την διασύνδεση με τις βάσεις δεδομένων χρησιμοποιήθηκε η τεχνολογία των EJBs τα οποία προσφέρουν την δυνατότητα δημιουργίας μεθόδων επιχειρηματικής λογικής απαλλάσσοντας τον προγραμματιστή από ζητήματα όπως ασφάλειας, συγχρονισμού κτλ .

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν τα ακόλουθα εργαλεία.

- Σύστημα Διαχείρισης Βάσης Δεδομένων Mysql
- Πλατφόρμα ανάπτυξης NET Beans
- Εξυπηρετητής εφαρμογών, JBoss
- Εξυπηρετητής υπηρεσιών Axis

2.Περί της Ηλεκτρονικής Διακυβέρνησης

«Ο δημόσιος τομέας χρειάζεται να προχωρήσει προς το να διαχειρίζεται την πληροφορία που κατέχει ως έναν επιχειρηματικό πόρο για να ωφελεί τους πολίτες και τις επιχειρήσεις και να βελτιώνει την αποτελεσματικότητα της ίδιας της διακυβέρνησης»

Cabinet Office,
Ηνωμένο Βασίλειο 2001

Η **Ηλεκτρονική Διακυβέρνηση** αποτελεί μια πολυδιάστατη έννοια μιας και επιδιώκει την εισαγωγή νέων τεχνολογιών και τεχνικών σε πολλούς, διαφορετικούς και όχι απόλυτα ορισμένους τομείς και υπηρεσίες οι οποίοι και συνθέτουν την «αφηρημένη» έννοια της διακυβέρνησης. Εάν επιδιώκαμε να δώσουμε έναν ορισμό της ηλεκτρονικής διακυβέρνησης θα μπορούσαμε να τη θεωρήσουμε ως “την εφαρμογή της τεχνολογίας της πληροφορίας και των επικοινωνιών, την οργανωτική αλλαγή και τις νέες ικανότητες που χρειάζονται στο δημόσιο τομέα με σκοπό την βελτίωση των δημόσιων υπηρεσιών, των δημοκρατικών διαδικασιών και την ενίσχυση της δημόσιας πολιτικής” (Information Society Directorate-General, 2004).

Συνοπτικά οι άμεσοι στόχοι-σκοποί της ηλεκτρονικής διακυβέρνησης είναι οι εξής:

Ένα ανοικτό και διεπόμενο από διαφάνεια δημόσιο τομέα: δημόσια διοίκηση κατανοητή και υπόλογη έναντι των πολιτών, ανοικτή στη δημοκρατική συμμετοχή και έλεγχο.

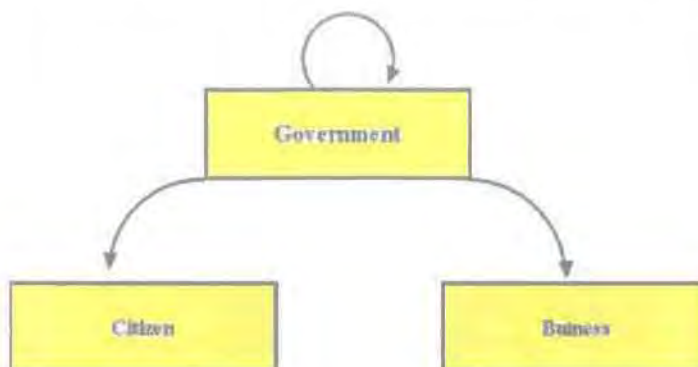
Δημόσιος τομέας στην υπηρεσία όλων. Δημόσιος τομέας με επίκεντρο τους χρήστες απευθύνεται σε όλους, δηλαδή να μην αποκλείεται κανένας από τις υπηρεσίες που παρέχονται και να γίνεται σεβαστή η προσωπικότητα με την παροχή εξατομικευμένων υπηρεσιών .

Παραγωγικός δημόσιος τομέας που να αξιοποιεί όσο το δυνατόν καλύτερα τα χρήματα των φορολογουμένων. Το αποτέλεσμα είναι να αναλώνεται λιγότερος χρόνος για την αναμονή στις ουρές, η δραστική μείωση των σφαλμάτων και η διάθεση περισσότερου χρόνου για επαγγελματική διαπροσωπική υπηρεσία, στοιχεία που καθιστούν πιο ικανοποιητική την εργασία των δημοσίων υπαλλήλων.

Δυνητικά προβλήματα για την περαιτέρω ανάπτυξη του e Government

- Χαμηλός βαθμός διείσδυσης του internet σε κάποιες χώρες
- Περιορισμένος βαθμός διάθεσης παρερχομένων υπηρεσιών
- Μικρή έως ελάχιστη η εξοικείωση πολιτών με την τεχνολογία και την κοινωνία της πληροφορίας.
- Έλλειψη εύκολης πρόσβασης σε άτομα με αναπηρία
- Καχυποψία και ανασφάλεια των πολιτών σε σχέση με την εξασφάλιση της ιδιωτικότητας τους

Θα μπορούσαμε να χωρίσουμε την ηλεκτρονική διακυβέρνηση στις εξής 3 κατηγορίες



Σχήμα 2.1 - Γραφική απεικόνιση των κατηγοριών

Government-to-Citizen (G2C) - Κυβέρνηση προς Πολίτες

Οι πολίτες μπορούν γρήγορα και άμεσα να έχουν πρόσβαση στις πληροφορίες από τις δημόσιες αρχές. Μπορούν να κατανοήσουν άμεσα πώς λαμβάνονται οι αποφάσεις και πώς δαπανώνται οι φόροι. Έτσι οι δημόσιες υπηρεσίες θα «αναγκαστούν» να είναι περισσότερο διαφανείς και ανοιχτοί σε επικρίσεις, πολεμώντας έτσι παράλληλα την διαφθορά και κατά επέκταση την απάτη.

Η χρήση των fora για ανταλλαγή απόψεων η δυνατότητα υποβολής άμεσων ερωτήσεων σε συνδυασμό με την ηλεκτρονική ψηφοφορία είναι αν μη τι άλλο ενίσχυση της δημοκρατικής διαδικασίας αυτή κάθε αυτής.

Government-to-Business (G2B) - Κυβέρνηση προς Επιχειρήσεις

Σημαντική πτυχή της ηλεκτρονικής διακυβέρνησης Government-to-Business είναι ότι οι παραγωγικότερες και υψηλότερης ποιότητας παρερχόμενες υπηρεσίες από τις δημόσιες διοικήσεις, επέφεραν αύξηση της παραγωγικότητας και της ανταγωνιστικότητας του ιδιωτικού τομέα, λόγω της μείωσης του κόστους λειτουργίας και συναλλαγής για τις επιχειρήσεις (χρόνος, πόροι). Μερικά παραδείγματα που έχουν άμεση προτεραιότητα, για την Ευρωπαϊκή Ένωση σύμφωνα με ανακοίνωση της αρμόδιας επιτροπής στο κοινοβούλιο (Βρυξέλλες 2003), είναι οι ηλεκτρονικές δημόσιες συμβάσεις, ηλεκτρονικά τελωνεία, δηλώσεις ειδικών φόρων κατανάλωσης, καθώς και η εξάλειψη της κυκλοφορίας εγγράφων σε χαρτί για την κοινωνική ασφάλιση.

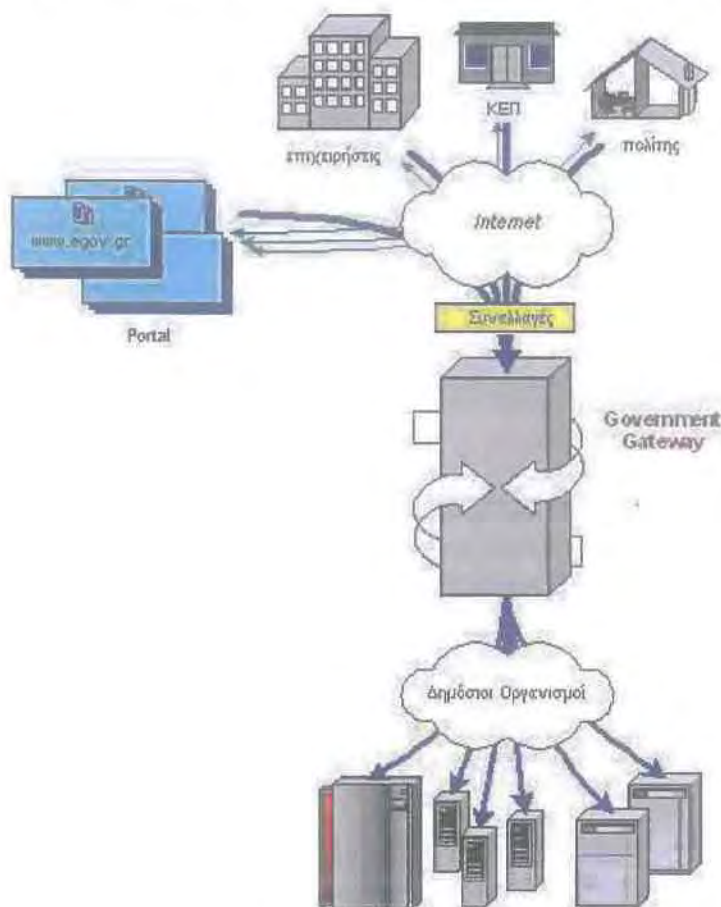
Είναι γεγονός πάντως ότι οι ηλεκτρονικές υπηρεσίες για τις επιχειρήσεις (δηλαδή η υποστήριξη της δυνατότητας διάδρασης και των συναλλαγών εκτός από την απλή παροχή πληροφοριών) έχουν εξελιχθεί περισσότερο από εκείνες για τους πολίτες

Government-to-Government (G2G) - Κυβέρνηση προς Κυβέρνηση

Η πολιτική για την καθιέρωση της ηλεκτρονικής διακυβέρνησης προβλέπει τους τρόπους που θα καταστήσουν δυνατή την οργανωμένη επικοινωνία μεταξύ περιφερειακών και τοπικών αρχών (πχ νομαρχίες-δήμοι, Υπουργεία μεταξύ τους) καθώς πλέον και των οργάνων της Ευρωπαϊκής Κοινότητας, παρότι πρέπει να

υπερπηδηθούν πολλά εμπόδια, όπως ενίοτε η πληθώρα ρυθμιστικών διατάξεων που δημιουργεί σύγχυση, ο κατακερματισμός των αρμοδιοτήτων μεταξύ των παραδοσιακών φορέων του δημοσίου τομέα και οι πεπατημένες διαδικασίες και τρόποι εργασίας. (Ευρωπαϊκό κοινοβούλιο, Βρυξέλλες 2003)

Μια γραφική απεικόνιση του e government είναι η εξής (πηγή www.egov.gr) :



Σχήμα 2.2 – Μορφή e government

2.1 Περι Πύλης (Portal) Ηλεκτρονικής Διακυβέρνησης

Τα portals που αφορούν το e government to citizen δεν είναι το ίδιο απλά όπως τα εμπορικά (commercial) ή άλλα portals του διαδικτύου. Περιστρέφονται γύρω από τον πολίτη προσφέροντας στατικές πληροφορίες αλλά και διαδραστικές υπηρεσίες που θα τον διευκολύνουν στις συναλλαγές με το κράτος. Τα portals είναι προσβάσιμα από όλους χωρίς να αποκλείονται ιδιαίτερες κοινωνικές ομάδες όπως τα άτομα με ειδικές ανάγκες.

Σύμφωνα με τους Layne and Lee (2001) χωρίζουμε τα portals σε δύο κατηγορίες :

- “thin”

Αποτελεί την απλούστερη μορφή ανάπτυξης government portal. Περιλαμβάνει στατικές πληροφορίες (πχ νομοθετικά πλαίσια), τρόπους επικοινωνίας, διάφορα links κτλ

- “thick”

Αποτελεί το επόμενο βήμα ανάπτυξης ενός Government portal. Περιλαμβάνει διαφορετικού είδους πληροφοριών, πρόσβαση σε βάσεις δεδομένων και χρήση πληθώρας δυναμικών υπηρεσιών. Παραδείγματος χάριν έκδοση πιστοποιητικών γέννησης, άδειες γάμου κτλ.

Προκειμένου ένα government portal να θεωρείται αξιόπιστο από τους πολίτες, πρέπει να χρησιμοποιεί την μέγιστη δυνατή τεχνολογία ασφάλειας, αποφεύγοντας τεχνικές που εκθέτουν την ιδιωτικότητα, όπως πχ τα cookies. Επιπλέον σημαντικός παράγοντας αξιοπιστίας αποτελεί και η αδιάλειπτη διαθεσιμότητα του συστήματος, ώστε να εξυπηρετεί τον χρήστη οποιαδήποτε χρονική στιγμή.

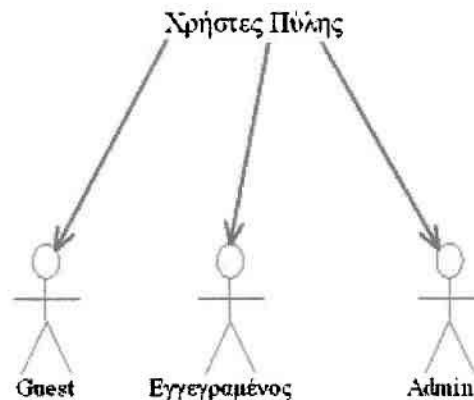
Επιπλέον για την καλύτερη «πλοήγηση» στο portal ο χρήστης να βρίσκει αυτό που ψάχνει με μέχρι δύο “clicks”, από το σημείο που βρίσκεται. Τέλος να υποστηρίζεται η παροχή δυνατότητας σχολίων από τους πολίτες καθώς και μηχανισμούς ανατροφοδότησης, με βάση αυτά. (Layne and Lee, 2001)

2.2 Περιγραφή ιδιοτήτων Πύλης Ηλεκτρονικής Διακυβέρνησης

Σε αυτή την ενότητα θα γίνει μια σύντομη αναφορά στις ιδιότητες που παρέχει μια Πύλη Ηλεκτρονικής Διακυβέρνησης προσανατολισμένη στον πολίτη.

2.2.1 Χρήστες

Υπάρχουν 3 «ειδών» χρήστες που σχετίζονται με την Πύλη. Ο guest, ο εγγεγραμμένος χρήστης και ο διαχειριστής (administrator), κάθε ένας από τους οποίους μπορεί να χρησιμοποιεί και να διαμορφώνει την Πύλη αντίστοιχα.



Σχήμα 2.3 – Χρήστες Πύλης

Guest : Ο πολίτης εισέρχεται στην Πύλη και επιδιώκει να βρει πληροφορίες για κάποιο θέμα που τον ενδιαφέρει. Οι πληροφορίες αυτές είναι σε στατική μορφή και είναι διαθέσιμες προς ανάγνωση και αποθήκευση σε όλους τους χρήστες χωρίς να απαιτείται οποιαδήποτε πιστοποίηση του. Οι πληροφορίες αυτές μπορεί να σχετίζονται με :

- Προκηρύξεις, διαγωνισμοί.
- Νόμοι, προεδρικά διατάγματα.
- Δελτία τύπου.
- Οδηγίες προς τον πολίτη για διεκπεραίωση κάποιας συναλλαγής.
- Παρουσίαση φορέα προς τους πολίτες και των εποπτευόμενων φορέων του.

- Links σε άλλα portals/sites ηλεκτρονικής διακυβέρνησης και πηγών ενδιαφερόντων πληροφοριών, σχετικών με το αντικείμενο του ίδιου του δημόσιου φορέα.
- Αναζήτηση πληροφοριών.
- Τηλέφωνα προσωπικού για το κοινό.
- Help desk πολιτών.
- Χάρτης του portal.

Εγγεγραμμένος Χρήστης : Ο Πολίτης προκειμένου να χρησιμοποιήσει κάποιες υπηρεσίες της Ηλεκτρονικής Διακυβέρνησης συμπληρώνει μια φόρμα προκειμένου να γίνει μέλος της Πύλης. Από τούδε και στο εξής θα κάνει login στην Πύλη με το προσωπικό όνομα χρήστη και κωδικό χρήστη που επέλεξε και θα είναι πλέον ικανός να χρησιμοποιεί τις υπηρεσίες .

Παραδείγματα υπηρεσιών μπορεί να είναι :

- Επεισόδια ζωής (απόκτηση ενός μωρού, αλλαγή διεύθυνσης, μετανάστευση, απόκτηση διπλώματος οδήγησης, αντιμετώπιση εγκλήματος).
- E-learning.
- Συμμετοχή στα κυβερνητικά δρώμενα με προσωπική ψήφο μέσω της «πύλης».
- Χώρος ανταλλαγής απόψεων με άλλους πολίτες.
- Υποβολή παραπόνων.
- Ηλεκτρονική συμπλήρωση φορολογικής δήλωσης και Φ.Π.Α.
- Οικονομικές συναλλαγές.

Διαχειριστής Συστήματος (Administrator) : Ο υπέρ χρήστης είναι επιφορτισμένος με την διαχείριση και ασφάλεια της Πύλης καθώς και με τον έλεγχο των συνεργαζόμενων διαδικτυακών υπηρεσιών.

2.2.2 Χαρακτηριστικά μιας Πύλης

- Πολυγλωσσική υποστήριξη. Το περιεχόμενο να είναι και σε άλλες γλώσσες πέρα της ελληνικής

Γνωρίζοντας ότι η Ελλάδα βρίσκεται εντός της Ευρωπαϊκής Ένωσης και συνεκτιμώντας την πληθώρα μεταναστών που δεν γνωρίζουν μόνο την ελληνική γλώσσα είναι επιτακτική η ανάγκη για μία πολυγλωσσική Πύλη που δεν θα απορρίπτει κανένα κομμάτι του πληθυσμού.

- Υπαρξη μηχανής αναζήτησης για γρήγορη και εύκολη αναζήτηση πληροφοριών από τους πολίτες
- Χρήση διαφόρων τεχνολογιών για πρόσβαση στην Πύλη πχ μέσω WAP κτλ

Γνωρίζοντας ότι οι περισσότεροι πλέον πολίτες κατέχουν κινητό τηλέφωνο γίνεται εύκολα αντιληπτή η ανάγκη για προβολή της Πύλης και σε μορφή WAP. Άλλωστε το μέλλον μάλλον συγκλίνει προς το mobile computing

- Δυνατότητα προσωποποίησης των σελίδων με βάση τις προτιμήσεις των χρηστών

Οι πολίτες να μπορούν εύκολα και άμεσα να παραμετροποιούν την Πύλη ώστε σε κάθε είσοδο τους σε αυτήν να βλέπουν συγκεκριμένες ειδήσεις κτλ. Η χρήση του RSS είναι επιβεβλημένη.

- Υπηρεσίες Ασφάλειας

Ίσως ένα από τα σημαντικότερα ζητήματα που αντιμετωπίζει η Ηλεκτρονική Διακυβέρνηση και αποτελεί ανασταλτικό παράγοντα για την περαιτέρω ανάπτυξη της.

Ο πολίτης από την μία να πειστεί ότι τα στοιχεία που δηλώνει κατά την συμπλήρωση μιας φόρμας δεν θα υποκλαπούν, παραποιηθούν κτλ από τρίτα πρόσωπα και αντίστοιχα από την πλευρά των κυβερνητικών οργανισμών ότι ο πολίτης είναι όντως αυτός που αιτήθηκε για την χρήση μιας υπηρεσίας. Εδώ μπορούν να χρησιμοποιηθούν πληθώρα διαθέσιμων τεχνολογιών ώστε να αυξηθεί στο μέγιστο δυνατό βαθμό η ασφάλεια των υπηρεσιών.

Παραδείγματα :

- Χρήση πιστοποιητικών από την πλευρά της Πύλης ορατά προς τους Πολίτες
- Χρήση τεχνολογίας SSL κατά την αποστολή δεδομένων (κατά την συμπλήρωση μιας φόρμας)
- Χρήση πιστοποιητικών από την πλευρά των Πολιτών τα οποία θα εκδίδονται από κάποιο δημόσιο φορέα και θα πιστοποιούν τον Πολίτη στην Πύλη
- Χρήση τεχνολογιών κρυπτογράφησης των SOAP μηνυμάτων που στέλνονται από τις διαδικτυακές υπηρεσίες
- Χρήση τεχνολογιών όπως υδατογράφηση για την πιστοποίηση εγγράφων

- Αδιάλειπτη και ασφαλής λειτουργία με υψηλή διαθεσιμότητα

Η πύλη είναι οργανωμένη με τέτοια αρχιτεκτονική και κατάλληλους μηχανισμούς (firewalls, back up servers κτλ) ώστε να αποτρέπει κακόβουλες επιθέσεις εναντίων της, μειώνοντας έτσι την αναξιοπιστία της προς τους πολίτες

- Ανοιχτή αρχιτεκτονική για εύκολη επέκταση του συστήματος
- Άμεση ενημέρωση των πολιτών με την χρήση κατάλληλων μηχανισμών για την πρόοδο και το αποτέλεσμα κάθε αίτησης του

Όταν αιτείται ο χρήστης κάποια υπηρεσία να γνωρίζει άμεσα για την έκβαση της πράξης του. Παραδείγματα αποτελούν η χρήση δυναμικών σελίδων που ενημερώνουν δυναμικά τον Πολίτη, αποστολή e mail με κάποιες πληροφορίες σε περίπτωση επιτυχίας, δημιουργία εγγράφων (πχ pdf) κτλ

- Δυνατότητα ψηφοφορίας μέσω ηλεκτρονικών δημοψηφισμάτων

Η πύλη περιέχει κάποιες ηλεκτρονικές ψηφοφορίες τις οποίες θα μπορεί να ψηφίσει κάθε πολίτης που επισκέπτεται την Πύλη. Οι ψηφοφορίες αυτές σχετίζονται περισσότερο με ένα θέμα που αφορά την ίδια την Πύλη (πόσο εύχρηστη πχ είναι) και δεν αποτελούν αποτέλεσμα επιστημονικής έρευνας.

- Δυνατότητα ανάδραση με την χρήση forum

Ο Πολίτης να έχει την δυνατότητα να ανταλλάσει απόψεις ερωτήματα και γενικά πληροφορίες με άλλους πολίτες (καθώς και υπεύθυνους της Πύλης) μέσω των fora

- Accessibility

Η πύλη να είναι σχεδιασμένη με τέτοιο τρόπο ώστε να μην αποκλείεται η χρήση της από πολίτες με ειδικές ανάγκες. Να ακολουθούνται όλοι οι κανόνες που ορίζονται από τους διεθνείς οργανισμούς.

Τέλος να αναφερθεί η αποφυγή της χρήσης Cookies η οποία μειώνει την αξιοπιστία της Πύλης.

Όλα αυτά τα στοιχεία έγινε προσπάθεια να υιοθετηθούν στο μέγιστο δυνατό βαθμό κατά την σχεδίαση της Πύλης Εξυπηρέτησης Πολιτών

2.2.3 Παραδείγματα portals Ηλεκτρονικής Διακυβέρνησης

Έχουν ήδη αναπτυχθεί κάποια portals ηλεκτρονικής διακυβέρνησης στο εξωτερικό τα οποία παρέχουν πληθώρα υπηρεσιών και αποτελούν πρότυπο για κάθε αντίστοιχη προσπάθεια. Ενδεικτικά θα αναφερθούν μερικά από αυτά και θα σχολιαστεί η ιδιαιτερότητα τους.

1)Portal Ηλεκτρονικής Διακυβέρνησης της Σιγκαπούρης

<http://www.ecitizen.gov.sg/>

Ο πολίτης έχει την δυνατότητα να πληρώσει μέσω του portal τους φόρους του, διάφορα πρόστιμα, να εκδώσει άδειες γάμων, να κάνει δωρεές κτλ Αυτό που έδωσε ώθηση για την ανάπτυξη της Ηλεκτρονικής Διακυβέρνησης στην Σιγκαπούρη ήταν η απόφαση της κυβέρνησης να «αναγκάζει» τους πολίτες να διαπεραιώνουν διάφορες υποθέσεις τους μόνο μέσω internet βάζοντας πρόστιμο σε όσους πηγαίνουν στις ίδιες τις υπηρεσίες. Αυτό διευκολύνεται και από την ύπαρξη τερματικών με πρόσβαση στο internet σε διάφορα σημεία της Σιγκαπούρης

2)Portal Ηλεκτρονικής Διακυβέρνησης της Μάλτας

<http://www.gov.mt/egovernment.asp>

Το portal της κυβέρνησης της Μάλτας πέρα από τις στατικές πληροφορίες διαθέτει και μια ξεχωριστή κατηγορία με πληθώρα υπηρεσιών (e services) διαθέσιμες προς τον πολίτη. Έχει ειδική κατηγορία για πιστοποιητικά όπου ο πολίτης μπορεί να κάνει αίτηση για πιστοποιητικά όπως γέννησης, θανάτου κτλ τα οποία και στέλνονται ταχυδρομικά. Έχει ειδικές επίσης κατηγορίες για ζητήματα υγείας, ασφάλειας, μετανάστευσης.

3)Portal Ηλεκτρονικής Διακυβέρνησης της Νέας Ζηλανδίας
<http://www.govt.nz/services/?treeid=374>

Το portal της κυβέρνησης της Νέας Ζηλανδίας πέρα από μια πληθώρα υπηρεσιών που παρέχει προς τους πολίτες αποτελεί και είσοδο προς τις δημόσιες υπηρεσίες με την μορφή του intranet. Γι αυτό και έχει αναπτύξει σύστημα αναγνώρισης τόσο για την ταυτοποίηση των πολιτών αλλά και για την πιστοποίηση των δημόσιων υπηρεσιών. Η ολοκλήρωση αυτού του εγχειρήματος προβλέπεται τον Ιούνιο του 2008 (<http://www.e.govt.nz/services/authentication>)

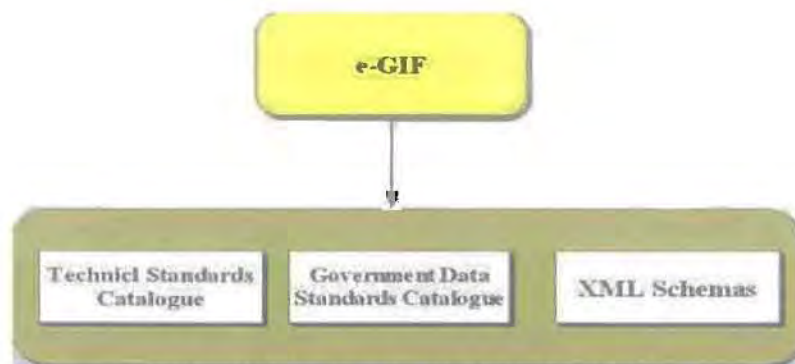
2.3 Διαλειτουργικότητα

Για την υλοποίηση του e government, επιτακτική είναι η ανάγκη της ομαλής και άμεσης διακίνησης πληροφοριών ανάμεσα στα πληροφοριακά συστήματα που την αποτελούν. Για να επιτευχθεί αυτό, απαιτείται η διαλειτουργικότητα, η οποία ορίζεται ως η δυνατότητα της επικοινωνίας και ανταλλαγής δεδομένων μεταξύ απομακρυσμένων συστημάτων ανεξαρτήτως (πλατφόρμας) λειτουργικού συστήματος, γλώσσας προγραμματισμού και τύπου πληροφοριακού συστήματος που χρησιμοποιείται.

Για αυτό το λόγο γίνονται, διεθνώς, προσπάθειες να τεθούν κάποιες συγκεκριμένες ανοιχτές προδιαγραφές διαλειτουργικότητας. Οι προδιαγραφές αυτές ορίζουν την βασική μορφή και υποδομή, δίνοντας στους διάφορους δημόσιους φορείς την δυνατότητα να επικεντρώνονται στην εξυπηρέτηση του πολίτη και των επιχειρήσεων μέσω των υπηρεσιών προστιθέμενης πληροφορίας που προσφέρουν.

Ένα τέτοιο πλαίσιο καθορίζει τις προδιαγραφές και νόρμες επικοινωνίας, αλλά αφήνει τους δημόσιους φορείς να καθορίσουν πώς να προσαρμόσουν τις επιχειρησιακές τους λειτουργίες έτσι ώστε να αυξήσουν την αποδοτικότητά τους επωφελούμενοι των ευκαιριών που προσφέρει μία αυξημένη διαλειτουργικότητα” (Electronic Business Systems, 2003).

Χαρακτηριστικό παράδειγμα αποτελεί η Βρετανική υπηρεσία e-GIF (e-Government Interoperability Framework) η οποία έχει ορίσει συγκεκριμένα πρότυπα τα οποία και έχουν υιοθετηθεί σε μεγάλο βαθμό από την Ευρωπαϊκή Ένωση.



Σχήμα 2.4 - Αρχιτεκτονική e-GIF (πηγή e gif)

Προκειμένου, λοιπόν, να επιτύχουμε την απαραίτητη διαλειτουργικότητα και ανεξαρτησία, που περιγράφηκε προηγουμένως, οδηγηθήκαμε στην ανάπτυξη της έννοιας των Υπηρεσιών Ιστού - Web services .

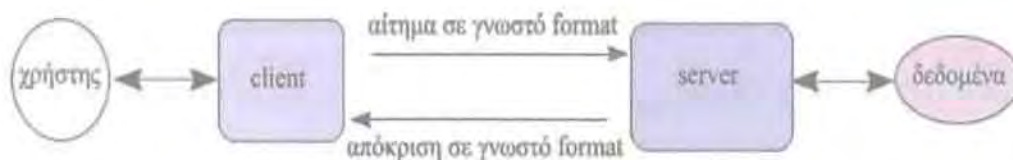
2.4 Υπηρεσίες Ιστού – Web Services

Μια υπηρεσία ιστού (web service) είναι μια εφαρμογή-οντότητα που δέχεται και επεξεργάζεται διάφορα αιτήματα από άλλες εφαρμογές σε ένα δίκτυο, όπως το διαδίκτυο. Συνήθως, καλείται να υπολογίσει κάτι, να αναζητήσει κάτι ή να επιστρέψει τιμές από μια βάση δεδομένων.

Μπορεί η όλη διαδικασία να περιγραφεί και ως ανταλλαγή μηνυμάτων μεταξύ μιας υπηρεσίας και των πελατών (clients) της. Ο πελάτης στέλνει ένα μήνυμα αιτήματος στην υπηρεσία και η υπηρεσία στέλνει ένα μήνυμα απάντησης πίσω στον πελάτη.

Η έννοια μιας υπηρεσίας ιστού είναι αρκετά διαφορετική από αυτήν μιας πύλης στο διαδίκτυο ή ενός ιστοχώρου, οι οποίοι απευθύνονται στους ανθρώπους και παρέχουν οπτικές πληροφορίες υπό μορφή κειμένου ή εικόνων. Σε μια υπηρεσία ιστού, η επικοινωνία γίνεται μεταξύ εφαρμογών, δηλαδή τον πελάτη και την υπηρεσία, οι οποίες και αντιστοιχούν σε μια web οντότητα(πχ ένα Portal) .

Για να περιγραφεί μια υπηρεσία ιστού, πρέπει να είναι γνωστό το περιεχόμενο και η δομή των μηνυμάτων που ανταλλάσσονται μεταξύ του πελάτη και της υπηρεσίας. Ένας πελάτης ξέρει πώς να υποβάλει ένα αίτημα και πώς να ερμηνεύσει την απάντηση που θα δεχτεί σαν επιστροφή από τον εξυπηρετή (server)



Σχήμα 2.5 - Client – Server μορφή

Προκειμένου όμως να γνωρίζουμε τις διαθέσιμες υπηρεσίες που διατίθενται, απαιτείται η ύπαρξη ενός προτύπου που θα καθορίζει την λειτουργικότητα της

υπηρεσίας, το πρωτόκολλο που χρησιμοποιείται και το URL που βρίσκεται. Ένα τέτοιο πρότυπο αποτελεί το WSDL και ο κατάλογος που περιέχει το σύνολο των πρότυπων αυτών αποτελεί το UDDI.

Σχηματικά η όλη διαδικασία παροχής και εύρεσης είναι η εξής (πηγή διπλωματική εργασία Αλέξανδρου Γκελμπέζη)



Σχήμα 2.6 - Διαδικασία παροχής και εύρεσης υπηρεσιών

Αναλυτικότερα, εάν επιθυμούσαμε να συμπτύξουμε τα **χαρακτηριστικά των υπηρεσιών ιστού** θα μπορούσαμε να αναφέρουμε τα εξής:

- Σχηματισμός ενός κατανεμημένου περιβάλλοντος. Αντικείμενα-υπηρεσίες προσπελάζονται μέσω κατάλληλων διεπαφών για την υλοποίηση λειτουργιών.
- Χρήση του N-tier¹ μοντέλου.
- Τα τμήματα τους είναι χαλαρά συνδεδεμένα: ένας πελάτης μπορεί να ρωτήσει τον ενδιάμεσο κατά τη διάρκεια εκτέλεσης και να «πάρει» τη διεπαφή προς μια υπηρεσία

2.5 Οι τεχνολογίες των Web Services

Η πιο διαδεδομένη προγραμματιστική προσέγγιση που χρησιμοποιείται στο υπηρεσιακό δίκτυο είναι τα XML web services απόγονος των τεχνολογιών των components και remote procedure calls (DCOM, CORBA).

Τα XML web services είναι στοιχειώδεις, αυτό-περιγραφόμενες, εφαρμογές που δημοσιοποιούνται, εντοπίζονται και καλούνται από οπουδήποτε στο Web.

Τα XML web services αποτελούν, ουσιαστικά, προγραμματιστικές βιβλιοθήκες. Πρόκειται δηλαδή για κάτι αντίστοιχο μιας αντικειμενοστραφούς κλάσης η οποία διαθέτει τις public μεθόδους της μέσω Internet και όχι μόνο στο συγκεκριμένο προγραμματιστικό περιβάλλον, με χρήση κοινά αποδεκτών προτύπων. Στα XML web services ανήκουν τα : WSDL, SOAP UDDI

Πριν έρθουν στην επιφάνεια τα XML Web services, υπήρξαν μια ποικιλία προτάσεων- πλατφορμών, βασισμένα σε XML, που σκοπό είχαν τη σύνδεση και αρμονική συνεργασία μεταξύ B2B (Business To Business) φορέων. Οι πλατφόρμες αυτές αποτελούν προπομπό των δικτυακών υπηρεσιών γιατί προσπάθησαν να επιτύχουν την αυτοματοποιημένη συνεργασία μεταξύ δύο ή παραπάνω ανεξάρτητων φορέων. Παρακάτω, αναφέρονται οι πιο διαδεδομένες:

- ebXML www.ebxml.com
- RosettaNet www.rosettanet.org
- Commerce XML www.cxml.org

Το σύνολο των πλατφορμών-προτάσεων που μόλις αναφέρθηκαν χρησιμοποιούν διαφορετικά μοντέλα και πρότυπα για την ανταλλαγή των μηνυμάτων και την οριοθέτηση των διαδικασιών. Η συνεργασία μεταξύ τους είναι αδύνατη, οπότε κάθε φορέας διαλέγει ποια πλατφόρμα να υιοθετήσει απορρίπτοντας ταυτόχρονα κάθε πιθανότητα για αυτοματοποιημένη συνεργασία με άλλη.

Η επιτυχία και η εξάπλωση των υπηρεσιών Ιστού θα εξαρτηθεί σε μεγάλο βαθμό από τον τρόπο ορισμού των πρωτοκόλλων και των τυποποιήσεων που τις αφορούν. Αν υπάρξουν πολλοί τύποι υπηρεσιών τότε η εξάπλωση τους θα σημάνει την ύπαρξη ενός πύργου της «Βαβέλ» όπου τελικά τα διάφορα τμήματα λογισμικού δεν θα επικοινωνούν.

2.6 Τεχνολογικά Πρότυπα και Μοντέλα

2.6.1 SOAP

Το SOAP είναι η τεχνολογία (RPC – remote Procedure Call) με την οποία μπορούμε να ανταλλάσσουμε μηνύματα, τα οποία έχουν μια xml δομή, μέσω του διαδικτύου.

Θεωρείται “stateless” διότι θυμάται μόνο την τωρινή session με τον πελάτη, αγνοεί την σημασιολογία των μηνυμάτων που μεταφέρει και επιδιώκει την επικοινωνία μεταξύ εφαρμογών με την χρήση αυτών.

XML

Η σύσταση της XML καθορίζει τη σύνταξη και τους κανόνες χρήσης ετικετών, για να συνθέσει τις πληροφορίες. Ο καθένας μπορεί να καθορίσει ένα λεξιλόγιο ετικετών και ιδιοτήτων των στοιχείων για να κτίσει τις πληροφορίες που τον ενδιαφέρουν. Μπορεί επίσης ρητά να καθοριστεί η δομή των εγγράφων, με τη χρήση Μεθόδου XML (XML Schema) ή Καθορισμού Τύπου Εγγράφου (DTD).

Χαρακτηριστικά του μοντέλου SOAP

- Έναν φάκελο (envelope) στον οποίο η εφαρμογή περιέχει οποιαδήποτε πληροφορία θα στείλει στην εφαρμογή στόχο.
- Μια πληθώρα συγκεκριμένων συμβάσεων που ορίζουν πώς οι πελάτες μπορούν να καλέσουν μια μέθοδο-υπηρεσία με την χρήση ενός μηνύματος SOAP και πώς αντίστοιχα αυτές ανταποκρίνονται στέλνοντας ένα άλλο μήνυμα SOAP.
- Μια περιγραφή για τον τρόπο μεταφοράς των μηνυμάτων SOAP πάνω από υπαρκτά πρωτόκολλα μεταφοράς (HTTP, SMTP)

Πίνακας 2.1 - Μορφή SOAP μηνύματος

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<doCheck soapenv:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
<arg0 xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">947-TI</arg0>
<arg1 xsi:type="soapenc:int"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">3</arg1>
</doCheck>
</soapenv:Body>
</soapenv:Envelope>
```

Το μήνυμα SOAP αποτελείται από τον φάκελο <Envelope> ο οποίος περιέχει την επικεφαλίδα <Header> και το κυρίως σώμα <Body>.

Η επικεφαλίδα περιέχει μεταδεδομένα τα οποία πολλές φορές αγνοούνται από διάφορες εφαρμογές. Τα μεταδεδομένα αυτά μπορεί να είναι ημερομηνία που στάλθηκε το μήνυμα, ο χρόνος που εκπνέει η εγκυρότητα αυτού κτλ. Το header είναι προαιρετικό. Το κυρίως σώμα είναι υποχρεωτικό και περιλαμβάνει όλη την χρήσιμη πληροφορία η οποία στέλνεται όπως προαναφέρθηκε σε XML.

Τα μηνύματα SOAP μπορούν να μεταφερθούν με την χρήση πρωτοκόλλων όπως HTTP, SMTP και FTP. Κατά κύριο λόγο προτιμάται το πρωτόκολλο HTTP λόγω της ταχύτητας απόκρισης του, ιδιαίτερα σε εφαρμογές που απαιτούν μια άμεση απάντηση. Τα δεδομένα μεταφέρονται όπως ακριβώς και οι ιστοσελίδες και δεν «κόβονται» από τυχόν τείχη ασφαλείας (firewalls).

Αναλυτικότερα το μήνυμα SOAP μεταφέρεται σαν ένα επιπλέον κομμάτι της HTTP αίτησης με την χρήση κάποιων επιπλέον ακόμα επικεφαλίδων απαραίτητων για την αναγνώριση του. Όπως γνωρίζουμε η POST και η GET χρησιμοποιούνται από το πρωτόκολλο HTTP για την αποστολή δεδομένων και επομένως και για την αποστολή SOAP μηνυμάτων.

Πίνακας 2.2 - Αποστολή SOAP μηνύματος μέσω POST

```
Content-Type: application/soap+xml; charset=utf-8
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<doCheckResponse
soapenv:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
<rpc:result xmlns:rpc="http://www.w3.org/2003/05/soap-rpc">return</rpc:result>
<return xsi:type="xsd:boolean">true</return>
</doCheckResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Επιπλέον ένα σύνολο κανόνων ορίζουν το πώς το πρότυπο SOAP χειρίζεται σε περίπτωση λαθών τόσο κατά την μεταφορά του μηνύματος όσο και κατά την προσπάθεια επεξεργασίας αυτού.

Στο μήνυμα λάθους που αποστέλλεται εντός του κυρίου σώματος περιλαμβάνεται η ετικέτα XML (tag) <fault> με υποστοιχεία τις ετικέτες : <faultcode> και <faultstring> στα οποία περιγράφεται το πού συνέβη το λάθος και πώς δημιουργήθηκε αυτό.

Υπάρχουν επιπλέον οι προαιρετικές ετικέτες <faultactor> και <detail> που δίνουν επιπλέον πληροφορίες για το λάθος ανάλογα με τον κωδικό που επιστρέφουν.

Πίνακας 2.3 - Έκφραση λαθών στα SOAP μηνύματα

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:st="http://www.skatestown.com/ws">
<env:Header>
<st:PublicServiceAnnouncement>
Skatestown's Web services will be unavailable after 5PM today
for a two hour maintenance window.
</st:PublicServiceAnnouncement>
</env:Header>
<env:Body>
<env:Fault>
<env:Code>
<env:Value>env:Sender</env:Value>
<env:Subcode>
<env:Value>st:InvalidPurchaseOrder</env:Value>
</env:Subcode>
</env:Code>
<env:Reason>
<env:Text xml:lang="en-US">
```



```

Your purchase order did not validate!
</env:Text>
</env:Reason>
<env:Detail>
<st:LineNumber>9</st:LineNumber>
<st:ColumnNumber>24</st:ColumnNumber>
</env:Detail>
</env:Fault>
</env:Body>
</env:Envelope>

```

Επιπλέον ορίζονται επεκτάσεις του προτύπου SOAP οι οποίες δίνουν ακόμα περισσότερες λειτουργίες όπως

- Ασφάλεια στην μεταφορά των μηνυμάτων με κρυπτογράφηση
- Συμπίεση και αποσυμπίεση των μηνυμάτων
- Δυνατότητα για χρήση από εξουσιοδοτημένους πελάτες
- Δυνατότητα ελέγχου αν τα μηνύματα ικανοποιούν συγκεκριμένα XML σχήματα.

2.6.2 WSDL (Web Services Description Language)

Η Web Services Description Language αποτελεί μια γλώσσα που και αυτή βασίζεται στην XML και σκοπό έχει τον καθορισμό- περιγραφή των διαδικτυακών υπηρεσιών ώστε να είναι προσβάσιμοι από τους πελάτες. Συγκεκριμένα περιγράφονται τα πρωτόκολλα που θα χρησιμοποιηθούν για την μεταφορά των μηνυμάτων, οι λειτουργίες που προσφέρονται από την υπηρεσία, η διεύθυνση της στο δίκτυο καθώς και το τί ορίσματα δέχεται και πως επιστρέφει τα δεδομένα.



Σχήμα 2.7 - WSDL μορφή

Πίνακας 2.4 - WSDL

```

<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions targetNamespace="http://localhost:8080/axis/services/AstinomikoTmima"
xmlns:apacheSOAP="http://xml.apache.org/xml-soap"
xmlns:impl="http://localhost:8080/axis/services/AstinomikoTmima"
xmlns:intf="http://localhost:8080/axis/services/AstinomikoTmima"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!--
WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)
-->
  <wsdl:message name="match_dataRequest">
    <wsdl:part name="in0" type="xsd:string" />
    <wsdl:part name="in1" type="xsd:string" />
    <wsdl:part name="in2" type="xsd:string" />
  </wsdl:message>
  <wsdl:message name="match_dataResponse">
    <wsdl:part name="match_dataReturn" type="xsd:int" />
  </wsdl:message>
  <wsdl:portType name="AstinomikoSessionLocal">
    <wsdl:operation name="match_data" parameterOrder="in0 in1 in2">
      <wsdl:input message="impl:match_dataRequest" name="match_dataRequest" />
      <wsdl:output message="impl:match_dataResponse" name="match_dataResponse" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="AstinomikoTmimaSoapBinding"
type="impl:AstinomikoSessionLocal">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  </wsdl:binding>
  <wsdl:operation name="match_data">
    <wsdlsoap:operation soapAction="" />
  </wsdl:operation>
  <wsdl:input name="match_dataRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://astinomiko" use="encoded" />
  </wsdl:input>
  <wsdl:output name="match_dataResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/axis/services/AstinomikoTmima" use="encoded" />
  </wsdl:output>
</wsdl:definitions>

```

2.6.3 UDDI (Universal Description, Definition and Integration)

Το πρωτόκολλο UDDI χρησιμοποιείται για την καταχώρηση διαδικτυακών υπηρεσιών. Προσφέρει την δυνατότητα ανακάλυψης υπηρεσιών (οι προγραμματιστές να μπορούν να βρίσκουν πληροφορίες για να αναπτύσσουν πελάτες που να αλληλεπιδρούν με τις υπηρεσίες αυτές).

Επομένως κάθε καταχώρηση στο UDDI περιέχει ένα αρχείο WSDL της κάθε υπηρεσίας (και την διεύθυνση αυτής) που είναι διαθέσιμη. Επιπλέον υπάρχουν και άλλες πληροφορίες για κάθε υπηρεσία όπως σχετικά με τον ιδιοκτήτη της.

Κατηγοριοποίηση του UDDI

- *Λευκές σελίδες* : Διαδικτυακές υπηρεσίες που προσφέρονται από μια συγκεκριμένη επιχείρηση
- *Κίτρινες σελίδες* : Αναζήτηση υπηρεσιών με βάση την κατηγορία που ανήκουν
- *Πράσινες σελίδες* : Περιγραφή του τρόπου που ένας πελάτης θα προσπελάσει μια συγκεκριμένη υπηρεσία

Πίνακας 2.5 - UDDI

```
<save_business generic="2.0" xmlns="urn:uddi-org:api_v2">
<authInfo>[authInfo value]</authInfo>
<businessEntity businessKey="55BB30D8-565A-4EF9-BA2E-83118AED644D">
<name>SkatesTown</name>
<description>UDDI business entity for SkatesTown</description>
<contacts>
...
</contacts>
<identifierBag>
<keyedReference keyName="DUNS"
keyValue="00-111-1111"
tModelKey="uuid:8609C81E-EE1F-4D5A-B202-3EB13AD01823"/>
</identifierBag>
<categoryBag>
<keyedReference keyName="Sporting and Athletic Goods Manufacturing"
keyValue="33992"
tModelKey="uuid:C0B9FE13-179F-413D-8A5B-5004DB8E5BB2"/>
<keyedReference keyName="New York"
keyValue="US-NY"
tModelKey="uuid:4E49A8D6-D5A2-4FC2-93A0-0411D8D19E88"/>
</categoryBag>
</businessEntity>
</save_business>
```

2.6.4 WSSD (Web Services Deployment Descriptor)

Το WSDD αποτελεί ένα XML έγγραφο το οποίο ακολουθεί μια συγκεκριμένη δομή με σκοπό την ανάπτυξη μιας δικτυακής υπηρεσίας σε έναν εξυπηρετητή (server). Το αρχείο αυτό χρησιμοποιείται και από τον Axis server (θα αναλυθεί παρακάτω) προκειμένου να δημιουργήσει το WSDL αρχείο.

2.6.5 MVC (Model –View – Controller) Framework

Κατά την υλοποίηση μεγάλων εφαρμογών απαιτείται η χρήση του MVC μοντέλου. Σύμφωνα με αυτό το πρόβλημα χωρίζεται σε 3 κατηγορίες. Η πρώτη κατηγορία είναι το μοντέλο (model) στο οποίο περιλαμβάνονται τα δεδομένα και η επιχειρηματική λογική. Η δεύτερη κατηγορία είναι η προβολή (view) το οποίο περιλαμβάνει την παρουσίαση της εφαρμογής στον χρήστη (interface,σελίδες κτλ). Τέλος έχουμε τον ελεγκτή (controller) ο οποίος ασχολείται με την επεξεργασία των δεδομένων που εισάγονται στο σύστημα και ουσιαστικά ενώνει τις δυο κατηγορίες που αναφέρθηκαν προηγουμένως. Χαρακτηριστικό παράδειγμα ενός απλού MVC μοντέλου αποτελούν οι jsp σελίδες (view) και των servlets (controller-model).

2.7 Πλατφόρμες Ανάπτυξης Web based λογισμικού

Τα τελευταία χρόνια η ανάπτυξη και εξέλιξη διαφόρων πλαισίων ανάπτυξης λογισμικού για Client-Server εφαρμογές οδήγησε δύο πολύ μεγάλες εταιρίες σε έναν ανεπίσημο πόλεμο με φόντο τις εφαρμοσμένες δικτυακές υπηρεσίες και πολυστρωματικών (n-tier) δικτυακών πληροφοριακών συστημάτων με τη χρήση δυναμικών (dynamic) portals.

Η Microsoft με την ενοποιημένη πλατφόρμα .NET και με τον SQL Server 2005 και η SUN, με το γνωστό προϊόν της (Java) με την αρχιτεκτονική J2EE και την υποστήριξη της Oracle παλεύουν να κατακτήσουν σημαντικό market share από την νέα μελλοντική φιλοσοφία συστημάτων : **web-based Information Systems**.

Το Internet πια, λειτουργεί σαν ένα αμφίδρομο, ενδιάμεσο στρώμα γεμάτο πληροφορία, η οποία είναι πραγματικά τεράστια μεταξύ των εκατομμύρια χρηστών ανά τον κόσμο. Τρία είναι τα σημαντικά σημεία τα οποία υποστηρίζαν και υποστηρίζουν αυτήν την φιλοσοφία

- Η ανάπτυξη δυναμικών-αμφίδρομων ιστοσελίδων (ASP, PHP, JSP) οι οποίες εξαλείφοντας στην ουσία την απλή στατική HTML μπορούν πια σε πολύ μεγάλο βαθμό να δημιουργήσουν μία ιδεατή και δυναμική client διεπαφή (νέου είδους δυναμικής σχεδίασης διεπαφών χρήστη) συνδυαζόμενες όλες μαζί σε έναν ενιαίο δικτυακό web server (portal)
- Η ανάπτυξη ισχυρών **web-based** Application servers οι οποίοι ως ενδιάμεσο στρώμα σε N-tier¹ αρχιτεκτονική αναλαμβάνουν την πολυδιάστατη επεξεργασία των δεδομένων που προέρχονται είτε από τα αμφίδρομα portals είτε από δικτυακές βάσεις δεδομένων είτε από το Διαδίκτυο και τέλος
- Η ανάπτυξη σύνθετων δικτυακών Βάσεων δεδομένων και η σύνδεσή τους σε ένα web interface (ODBC, JDBC), είτε για απλές εφαρμογές (Mysql) είτε για πιο σύνθετες (SQL Server, DB2, Oracle).

Το διαδίκτυο, λοιπόν, από μια απλή αποθήκη εικόνων και κειμένου κάποτε, εξελίσσεται ραγδαία σε ένα παροχέα υπηρεσιών, πληροφοριακού χαρακτήρα, που

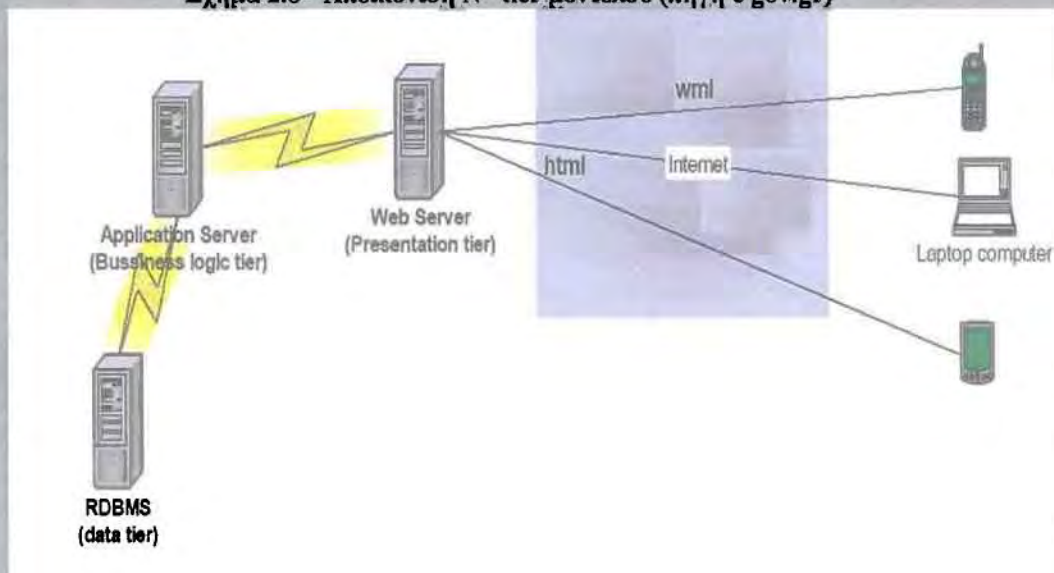
περιλαμβάνει για παράδειγμα κρατήσεις εισιτηρίων σε θέατρα, πλοία, πλήθος e-commerce και B2B (business to business) εφαρμογών. Αυτές οι νέες δυνατότητες που πλέον υπάρχουν, φαντάζουν ιδιαίτερα ελκυστικές από τους διάφορους φορείς-εταιρίες, με βάση την αυξημένη πιθανότητα κέρδους που οφείλεται στην προσθήκη, προστιθέμενης αξίας χρήσης, από τους διαδικτυακούς χρήστες. Το web banking και το e-government είναι κάποια χαρακτηριστικά παραδείγματα στο πώς η παροχή υπηρεσιών μέσω διαδικτύου έχει διεισδύσει σε όλες τις εκφάνσεις της κοινωνικής οργάνωσης.

N-tier¹ :

Το γενικό μοντέλο το οποίο ακολουθείται κυρίως κατά την ανάπτυξη εφαρμογών ηλεκτρονικού εμπορίου, e government κτλ είναι αυτό του 3-tier (ή n-tier στην γενική του μορφή). Πρόκειται για de facto τυποποίηση που υποστηρίζεται από πολλούς μεγάλους κατασκευαστές λογισμικού ανάπτυξης εφαρμογών (Microsoft, Sun). Σύμφωνα με το μοντέλο αυτό, η εφαρμογή διαχωρίζεται σε τρία βασικά επίπεδα

- Στο επίπεδο παρουσίασης (Presentation Layer), το οποίο συνιστά την διεπαφή (interface) της εφαρμογής με τον χρήστη
- Στο επίπεδο επιχειρηματικής λογικής (Business Layer ή Application Layer), στο οποίο υλοποιείται η «λογική» και οι λειτουργίες της εφαρμογής
- Στο επίπεδο δεδομένων (Data Layer), το επίπεδο στο οποίο συμβαίνει η αποθήκευση και διαχείριση δεδομένων.

Σχήμα 2.8- Απεικόνιση N-tier μοντέλου (πηγή e gov.gr)



3 Ανάλυση και Σχεδίαση συστήματος

Στο κεφάλαιο αυτό θα παρουσιαστεί αναλυτικά η σχεδίαση του συστήματος Ηλεκτρονικής Διακυβέρνησης για κάθε στοιχείο που την αποτελεί ξεχωριστά. Θα χρησιμοποιηθούν όροι και έννοιες που έχουν εξηγηθεί αναλυτικά σε προηγούμενη ενότητα .

3.1 Στόχοι

Όπως έχει ήδη αναφερθεί οι προδιαγραφές και οι στόχοι του συστήματος αποτελούν ένα υποσύνολο αυτών που είναι ορισμένες στο έγγραφο προκήρυξης έργου Ηλεκτρονικής Διακυβέρνησης του Δήμου Βόλου (Φεβρουάριος 2007)

Οι προδιαγραφές και οι στόχοι του συστήματος είναι οι εξής :

- Δημιουργία κεντρικού portal που θα δίνει στους πολίτες την δυνατότητα να επιλέγουν διάφορες υπηρεσίες
- Διαλειτουργικότητα και συμβατότητα συστήματος με διάφορες πλατφόρμες λογισμικού και πληροφοριακών συστημάτων
- Χρήση open source προγραμματιστικών γλωσσών και εργαλείων ώστε να επιτυγχάνεται η εύκολη επέκταση του συστήματος
- Υλοποίηση ενδιάμεσου στρώματος (middleware) το οποίο θα αναλαμβάνει να διασυνδέσει τα απομακρυσμένα συστήματα (πχ βάσεις δεδομένων των δήμων)
- Καταγραφή των συναλλαγών σε logs ώστε να εξασφαλίζεται η «μη αποκύρξη»
- Διαθεσιμότητα του συστήματος
- Επίδοση (χειρισμός λαθών από το σύστημα με επιστροφή κατάλληλων μηνυμάτων στους πολίτες)
- Ελάχιστες τεχνικές απαιτήσεις από τους πελάτες
- Ανάγκη μικρής ομάδας διαχείρισης-παραμετροποίησης του συστήματος
- Ασφάλεια των πράξεων των πολιτών

Στα επόμενα κεφάλαια γίνεται ανάλυση αυτών των απαιτήσεων καθώς και το πώς επιτυγχάνονται.

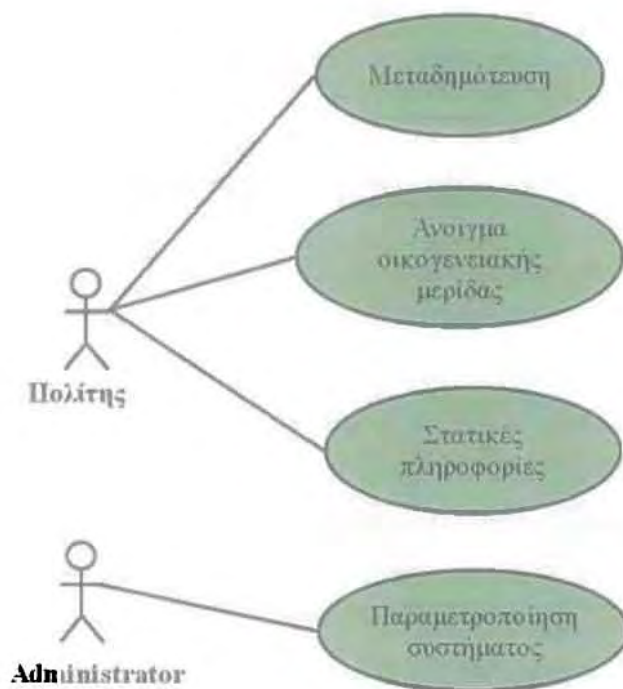
Τέλος αναφέρεται ότι το σύστημα που αναπτύσσεται στα πλαίσια μιας διπλωματικής εργασίας ενός εξαμήνου σίγουρα δεν μπορεί να αποτελεί ένα ολοκληρωμένο προϊόν αλλά παρά μόνο μια «ρεαλιστική» εφαρμογή, που εξετάζει διάφορες περιπτώσεις και πτυχές του προβλήματος της Ηλεκτρονικής Διακυβέρνησης, χωρίς να της καλύπτει και να της αναπτύσσει εξαντλητικά. Τέλος κάποιες παραδοχές που πιθανώς υπάρχουν θα αναφέρονται ρητά στο αντίστοιχο σημείο ανάλυσης του συστήματος.

3.2 Όψεις

Στην υποενότητα αυτή θα παρουσιαστούν οι διάφορες αρχιτεκτονικές λύσεις που αφορούν την ανάπτυξη του συστήματος .

3.2.1 Υπηρεσιακή όψη

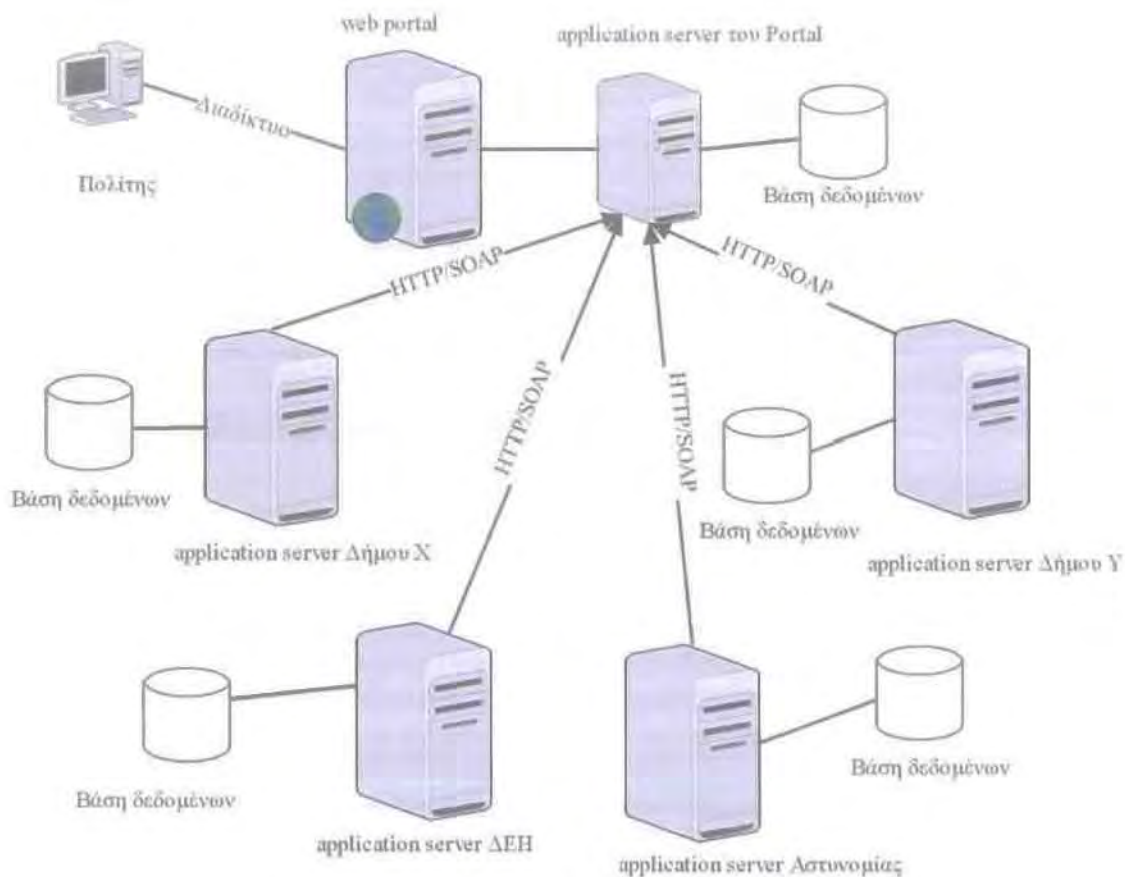
Η υπηρεσιακή όψη περιγράφει ποιες υπηρεσίες θα προσφέρει το λογισμικό που αναπτύσσεται και σε ποιες κατηγορίες χρηστών. Η περιγραφή της υπηρεσιακής αρχιτεκτονικής όψης γίνεται με διαγράμματα περιπτώσεων χρήσης (use case diagrams) της UML. Οι υπηρεσίες αυτές φαίνονται στο παρακάτω σχήμα:



Σχήμα 3.1 - Υπηρεσιακή οψη

3.2.2 Πλατφορμική όψη

Η **πλατφορμική** όψη περιγράφει την δομή – σχεδιασμό της πλατφόρμας hardware και software στην οποία θα λειτουργήσει το σύστημα. Το διάγραμμα φαίνεται παρακάτω :



Σχήμα 3.2 - Πλατφορμική όψη

Παρατίθενται οι παρακάτω «ορισμοί- συντομεύσεις », που από τούδε και στο εξής θα αναφέρονται με τις υπογραμμισμένες λέξεις

Οργανισμοί : ΔΕΗ , Αστυνομία , Δήμοι

ΠΕΠ : Πύλη Εξυπηρέτησης Πολιτών

Πύλη : Portal

Γενική εικόνα :

Θεωρείστε το υπουργείο δημόσιας τάξης – (Αστυνομία), την ΔΕΗ, και δύο δήμους έστω τον X και τον Y. Κάθε ένας από αυτούς διαθέτει από μια βάση δεδομένων. Έτσι το υπουργείο δημόσιας τάξης έχει έναν πίνακα ο οποίος περιέχει κάποια στοιχεία των πολιτών. Η ΔΕΗ έχει έναν πίνακα ο οποίος περιέχει δεδομένα σχετικά με τους πελάτες της και τέλος οι δήμοι (ο X και Y καθώς και όλοι οι δήμοι θεωρητικά που θα μπορούσαν να ενταχθούν στην ηλεκτρονική διακυβέρνηση) οι οποίοι έχουν τρεις πίνακες. Έναν ο οποίος κρατάει στοιχεία σχετικά με την μεταδημότευση πολιτών, έναν για την καταγραφή πολιτών που πλέον δεν ανήκουν στον δήμο (σύμφωνα με τον υπάρχον νόμο πρέπει να υπάρχει μια λογική «συνέχεια» στην κίνηση ενός πολίτη από έναν δήμο σε έναν άλλο) και τέλος ένας πίνακας στον οποίο καταχωρούνται στοιχεία για το άνοιγμα οικογενειακής μερίδας.

Τέλος θεωρείστε το ΠΕΠ το οποίο ουσιαστικά χωρίζεται σε δύο «κομμάτια». Σε αυτό που ασχολείται με το portal και την πρόσβαση εξουσιοδοτημένων χρηστών (περιλαμβάνει μια βάση δεδομένων με λογαριασμούς χρηστών) και σε αυτό που σχετίζεται με την διεκπεραίωση - υλοποίηση των υπηρεσιών. Στα πλαίσια του τελευταίου υπάρχει μια βάση δεδομένων στην οποία αποθηκεύονται πληροφορίες σχετικά με τις πράξεις του χρήστη - πολίτη.

Εκτός από τις βάσεις δεδομένων κάθε οργανισμός (δήμοι, ΔΕΗ, αστυνομία) καθώς και το ΠΕΠ διαθέτει έναν εξυπηρετητή (application server) ο οποίος διαθέτει το απαραίτητο λογισμικό για να απαντά σε αιτήσεις αναζήτησης, προσθήκης και διαγραφής των στοιχείων που περιέχει η βάση δεδομένων του. Οι αιτήσεις γίνονται μέσω του πρωτοκόλλου SOAP.

3.3 Επικοινωνία πρωτοκόλλου SOAP

Όπως φαίνεται και στο προηγούμενο σχήμα η επικοινωνία μεταξύ των οργανισμών και του ΠΕΠ γίνεται με το πρωτόκολλο SOAP. Θεωρώντας έναν οργανισμό και το ΠΕΠ ορίζουμε μια client - server επικοινωνία, με τον οργανισμό να παίζει τον ρόλο του εξυπηρετητή (server). Οι οργανισμοί και το ΠΕΠ διαθέτουν SOAP engines για την δημιουργία μηνυμάτων. Επιπλέον οι οργανισμοί διαθέτουν εξυπηρετητή δικτυακών υπηρεσιών (Web services Server) ο οποίος φιλοξενεί τις δικτυακές υπηρεσίες σύμφωνα με τα πρότυπα WSDL και WSDD. Οι υπηρεσίες αυτές θα χρησιμοποιηθούν από το ΠΕΠ για την υλοποίηση των διαδικασιών-υπηρεσιών που προσφέρει.

Περαιτέρω λεπτομέρειες για την δομή και αρχιτεκτονική κάθε οργανισμού ξεχωριστά, θα αναλυθούν σε παρακάτω ενότητα

3.4 Σχεδιαστικές αποφάσεις

- Το σύστημα ΠΕΠ είναι κεντριοποιημένο. Δηλαδή η εφαρμογή του ΠΕΠ είναι υπεύθυνη για να καλεί τις απαραίτητες δικτυακές υπηρεσίες που παρέχονται από τους οργανισμούς και να τις χρησιμοποιεί κατάλληλα για να υλοποιεί τις υπηρεσίες προς τον πολίτη. Αυτό μπορεί να είναι εν μέρει επικίνδυνο σε περίπτωση αστοχίας του server που φιλοξενεί το ΠΕΠ (θα μπορούσε βέβαια να αποφευχθεί κάτι τέτοιο με την χρήση εναλλακτικών back up server- και άλλων για εξισορρόπηση φορτίου) αλλά βοηθάει ιδιαίτερα στην απλοποίηση του προβλήματος.
- Κάθε δήμος που συνδέεται στο σύστημα ηλεκτρονικής διακυβέρνησης παρέχει τις ίδιες υπηρεσίες (σε σχέση με τα ονόματα, τις παραμέτρους που δέχεται, τα αποτελέσματα που στέλνει και την δομή των βάσεων δεδομένων που διαθέτει).
- Οι βάσεις δεδομένων των Οργανισμών είναι ανεξάρτητες οντότητες και προϋπήρχαν της υπό ανάπτυξης εφαρμογής (μπορούν να υλοποιηθούν τα σχήματα των βάσεων αυτόματα από τα EJBs αλλά θα θεωρήσουμε την πρώτη περίπτωση χωρίς βλάβη της γενικότητας)
- Δεν υπάρχει περιορισμός στην προγραμματιστική γλώσσα για την δημιουργία των υπηρεσιών από τους οργανισμούς. Ο κάθε οργανισμός θα μπορούσε να τις αναπτύξει θεωρητικά σε διαφορετική γλώσσα. Στην συγκεκριμένη περίπτωση έχει γίνει χρήση, όπως έχει αναφερθεί, της JAVA σε όλους.

- Ο τρόπος πρόσβασης στις βάσεις των δήμων μπορεί να ποικίλει και να είναι διαφορετικός χωρίς κανένα αντίκτυπο στην λειτουργία του ΠΕΠ. Στην συγκεκριμένη υλοποίηση έχουμε πρόσβαση μέσω EJBs
- Η υλοποίηση των υπηρεσιών μεταδημότευσης και άνοιγμα οικογενειακής μερίδας αφορά δύο δήμους. Με μικρές αλλαγές θα μπορούσε να κλιμακωθεί για όλους τους δήμους της Ελλάδας. Γι' αυτό και απαιτείται η κοινή μορφή των δικτυακών υπηρεσιών των δήμων.
- Σε διάφορους πίνακες που δημιουργήθηκαν δεν τοποθετήθηκαν όλα τα δυνατά πεδία (χάνοντας από την «ρεαλιστικότητα» της εφαρμογής) που πιθανώς να κρατούνται από τους δήμους παρά μόνο τα βασικά που απαιτούνται από την διαδικασία του ΚΕΠ για κάθε ενέργεια. Προφανώς επιπλέον προσθήκη πεδίων δεν θα επηρέαζε την λογική της εφαρμογής
- Χρήση συγκεκριμένων κωδικών σε περίπτωση λάθους (θα αναλυθούν διεξοδικά στην συνέχεια)
- Δημιουργία αρχείων log σε συγκεκριμένη μορφή (θα αναλυθεί στην συνέχεια)

3.5 Σχεδιασμός ροών ελέγχου και δεδομένων

Στην ενότητα αυτή θα παρουσιαστούν οι ροές ελέγχου και δεδομένων μεταξύ του ΠΕΠ και των διαφόρων υπηρεσιών που προσφέρονται από τους οργανισμούς. Για τον ορισμό των ροών χρησιμοποιήθηκε η γλώσσα μοντελοποίησης Unified Modeling Language (UML) και συγκεκριμένα τα Activity Diagrams της UML.

3.5.1 Εισαγωγικά

Στο μοντέλο που θα περιγραφεί θεωρούμε δύο περιπτώσεις :

Ι) Το ΠΕΠ παίζει τον ρόλο του client ενώ οι υπηρεσίες των οργανισμών τον ρόλο του server. Αναλυτικότερα οι ροές δεδομένων που έχουν σχεδιαστεί περιλαμβάνουν το εξής βασικό μοντέλο επικοινωνίας

- Για κάθε υπηρεσία-διαδικασία που έχει αναπτυχθεί στο ΠΕΠ και απαιτεί την κλήση μιας υπηρεσίας ενός οργανισμού θα στέλνεται το κατάλληλο μήνυμα (με τα σωστά ορίσματα που επιβάλλονται απο αυτό)
- Η υπηρεσία του (κάθε) οργανισμού λαμβάνει το μήνυμα, το επεξεργάζεται κάνει τις απαραίτητες ενέργειες (πρόσβαση στην βάση δεδομένων, υπολογισμούς κτλ) και επιστρέφει ένα μήνυμα-κωδικό επιτυχίας ή αποτυχίας της διαδικασίας
- Το μήνυμα αυτό λαμβάνεται από το ΠΕΠ και αφού επεξεργαστεί, δίνεται το αντίστοιχο μήνυμα επιτυχίας ή αποτυχίας μέσω μιας web σελίδας

II)Ο πολίτης μέσω του φυλλομετρητή του (browser), έχει εισαχθεί στο portal και πραγματοποιεί κάποια υπηρεσία-διαδικασία που προσφέρεται από αυτό. Σε αυτή την περίπτωση ο χρήστης θεωρείται σαν ο client και το ΠΕΠ σαν server. Αναλυτικότερα οι ροές δεδομένων που έχουν σχεδιαστεί περιλαμβάνουν το εξής βασικό μοντέλο επικοινωνίας

- Στις αντίστοιχες σελίδες του συστήματος ο χρήστης θα στέλνει τις απαραίτητες πληροφορίες που απαιτούνται για την πραγμάτωση μιας υπηρεσίας με την χρήση φορμών συμπλήρωσης
- Οι πληροφορίες αυτές στέλνονται στην εφαρμογή η οποία θα ενεργοποιήσει την αντίστοιχη ροή εργασίας που απαιτείται.
- Τα αποτελέσματα των υπολογισμών που πραγματοποιήθηκαν κατά την προηγούμενη φάση στέλνονται με την μορφή web σελίδων στον χρήστη

Το παραπάνω μοντέλο είναι σχεδιασμένο ώστε να ενεργεί σύγχρονα. Παραδείγματος χάριν, στην περίπτωση I) το ΠΕΠ πραγματοποιεί μια κλήση υπηρεσίας ενός οργανισμού και περιμένει έως ότου πάρει μήνυμα επιτυχίας ή αποτυχίας για να συνεχίσει την διαδικασία του.

Τέλος αξίζει να αναφερθεί ότι ο χωρισμός στις δύο αυτές περιπτώσεις είναι σε συμφωνία με το μοντέλο MVC (Model –View – Controller) όπου επιδιώκουμε τον χωρισμό του Business-logic από το Data και Presentation layer.

3.5.2 Κλήση υπηρεσιών οργανισμών από το ΠΕΠ

3.5.2.1 Ροή εργασίας : «Μεταδημότευση άγαμου ενήλικου σε δημοτολόγιο δήμου ή κοινότητας εκτός του τόπου καταγωγής του ίδιου ή των γονέων του»

Περίπτωση I)

Αρχικά γίνεται κλήση της υπηρεσίας του οργανισμού «Αστυνομία» προκειμένου να ελεγχθεί η φυσική ύπαρξη και αντιστοιχία των δεδομένων που δίνονται από τον πολίτη που επιθυμεί να κάνει την αίτηση για μεταδημότευση. Τα στοιχεία που ελέγχονται είναι ο αριθμός ταυτότητας σε σχέση με το όνομα και επίθετο του δημότη.

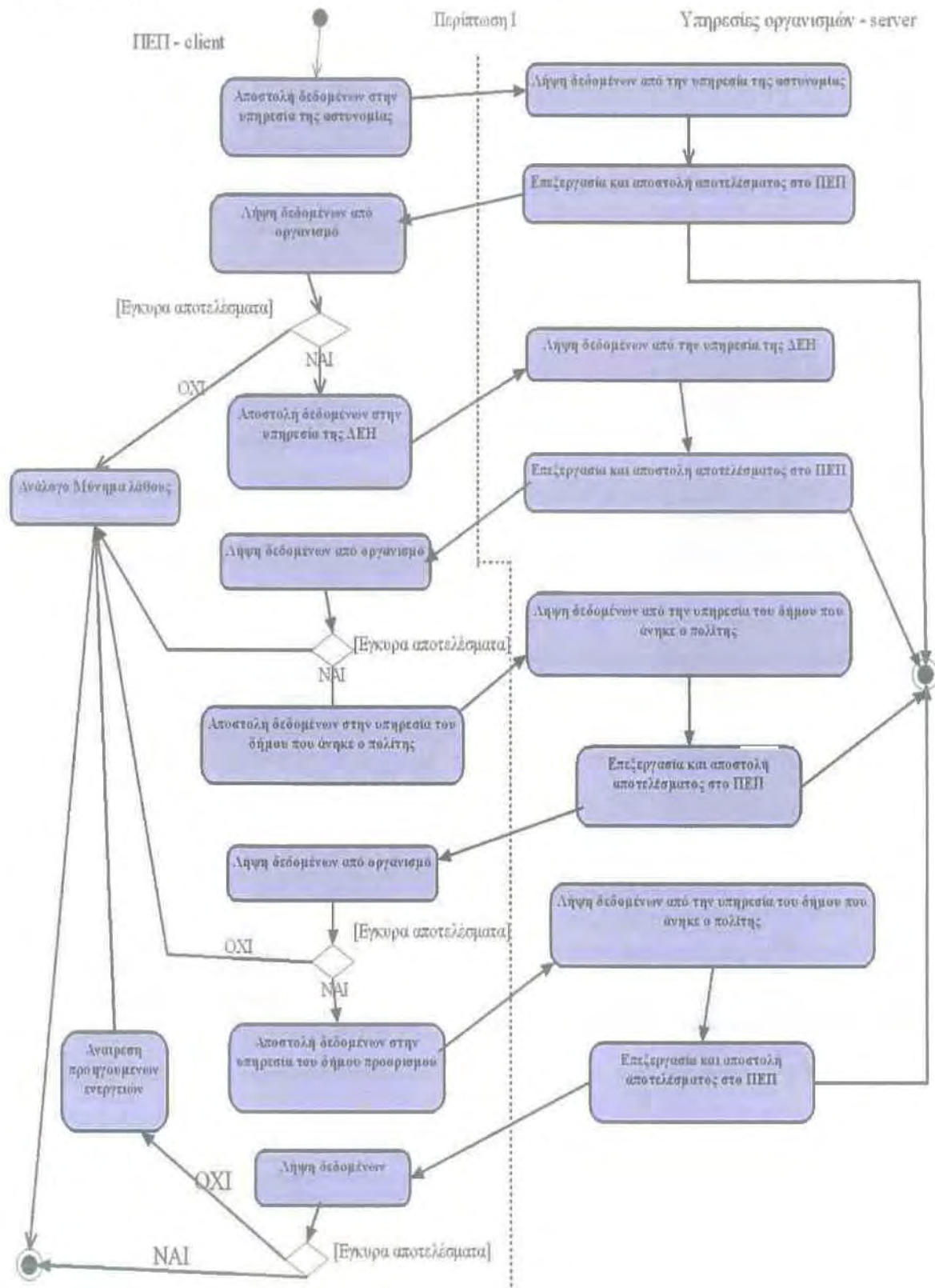
Στην συνέχεια εάν υπάρξει ορθή αντιστοιχία καλείται η υπηρεσία του οργανισμού ΔΕΗ. Σε αυτήν γίνεται έλεγχος της παραμονής του πολίτη στον δήμο προορισμού για τουλάχιστον δύο χρόνια από την ημέρα που κάνει την αίτηση μεταδημότευσης, σύμφωνα με τους ισχύοντες νόμους της μεταδημότευσης άγαμου πολίτη (όπως ορίζεται από την ροή εργασία του ΚΕΠ). Έπειτα και αφού έχουμε ορθή απάντηση από την υπηρεσία της ΔΕΗ καλείται η υπηρεσία διαγραφής από τον δήμο στον οποίο μέχρι τώρα άνηκε ο πολίτης. Και σε αυτή την περίπτωση γίνεται τυπικός έλεγχος για την ορθότητα-αντιστοιχία των δεδομένων που δίνονται από τον πολίτη κατά την συμπλήρωση της αίτησης. Εάν η διαγραφή είναι επιτυχής (ταυτόχρονα γίνεται εγγραφή στην βάση δεδομένων παλιών δημοτών ώστε να υπάρχει πάντα, σύμφωνα με το νομοθετικό πλαίσιο, λογική συνέχεια της μετακίνησης των πολιτών από δήμο σε δήμο (όπως συμβαίνει και στην περίπτωση θανάτου)) καλείται η υπηρεσία του δήμου

προορισμού προκειμένου να γίνει προσθήκη του πολίτη στην βάση δεδομένων του δήμου προορισμού. Σε περίπτωση αποτυχίας του τελευταίου έχουμε αναίρεση της προηγούμενης διαδικασίας (διαγραφή στον δήμο που άνηκε πριν).

Γενική η εικόνα ροής είναι η εξής :

Σχήμα 3.4 – Ροή Μεταδημότευσης

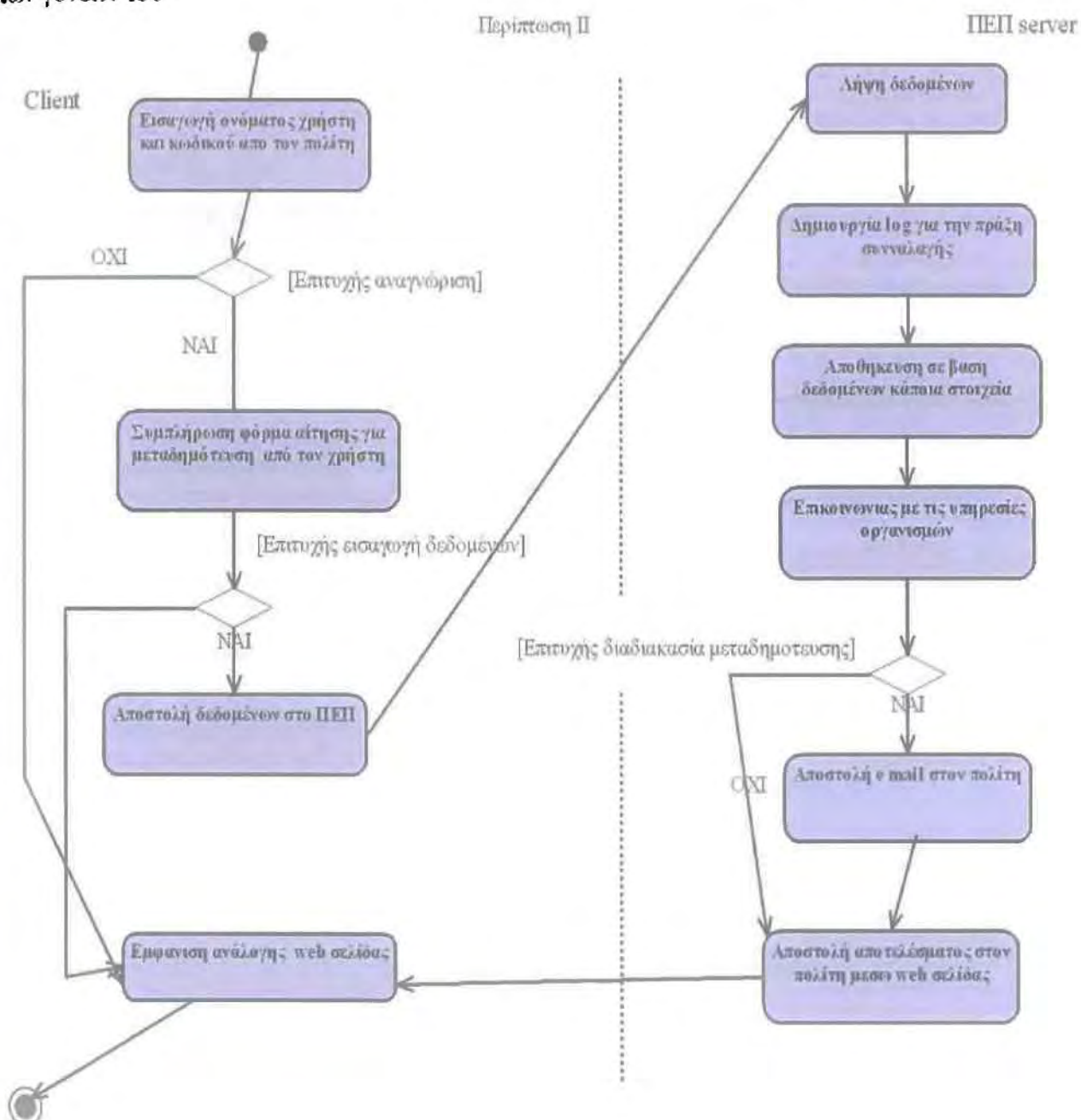
"Μεταδημότευση άγαμου ενήλικου σε δημοτολόγιο δήμου ή κοινότητας εκτός του τόπου καταγωγής του ίδιου ή των γονέων του"



Περίπτωση II)

Ο πολίτης, αφού αναγνωριστεί από το σύστημα (μέσω του ονόματος χρήστη και κωδικού) και μπει στην υπηρεσία αυτή, θα συμπληρώσει την απαραίτητη φόρμα. Αν ο έλεγχος ορθότητας των δεδομένων είναι επιτυχής τότε καλείται η ροή εργασίας που περιγράφηκε προηγουμένως. Επιπλέον σε περίπτωση επιτυχίας της διαδικασίας και για κάθε αίτηση που γίνεται κρατάται ένα αρχείο log καθώς επίσης στέλνεται ένα e mail στον πολίτη, για την επικύρωση των δεδομένων και τέλος κρατούνται κάποια δεδομένα του πολίτη σε μια βάση δεδομένων. Τέλος σε περίπτωση επιτυχίας δημιουργείται ένα αρχείο pdf που επικυρώνει την συναλλαγή και το οποίο μπορεί να κατεβάσει ο πολίτης. Σε κάθε περίπτωση, αποτυχίας ή επιτυχίας έχουμε τα κατάλληλα μηνύματα προς τον πολίτη.

"Μεταδημότευση άγαμου ενήλικου σε δημοτολόγιο δήμου ή κοινότητας εκτός του τόπου καταγωγής του ίδιου ή των γονέων του"



Ροή και των δύο περιπτώσεων (ενιαία μορφή) υπό την μορφή βημάτων και εναλλακτικών ροών

1. Login του πολίτη στην Πύλη
2. Συμπλήρωση φόρμας από τον πολίτη
3. Δημιουργία μοναδικού κωδικού συναλλαγής του πολίτη που αιτάται
4. Έναρξης εγγραφής στοιχείων στο ημερολόγιο (log)
5. Αποστολή μηνύματος κλήσης της υπηρεσίας «Αστυνομίας » με τα σωστά ορίσματα για ταυτοποίηση του πολίτη
6. Έλεγχος ορθότητας μηνύματος επιστροφής από Αστυνομία
7. Αποστολή μηνύματος κλήσης υπηρεσίας ΔΕΗ για ταυτοποίηση και έλεγχο κατοικίας του πολίτη για πάνω από δύο χρόνια
8. Έλεγχος ορθότητας μηνύματος επιστροφής από ΔΕΗ
9. Αποστολή μηνύματος κλήσης υπηρεσίας Δήμου για διαγραφή του πολίτη από τον δήμο που μέχρι τώρα διέμενε
10. Προσθήκη διαγραμμένου πολίτη σε αντίστοιχη βάση του δήμου
11. Έλεγχος ορθότητας μηνύματος υπηρεσίας Δήμου
12. Αποστολή μηνύματος κλήσης υπηρεσίας του Δήμου για προσθήκη του πολίτη στο νέο δήμο
13. Αποστολή e mail στον Πολίτη
14. Δημιουργία PDF αρχείου με τα στοιχεία της συναλλαγής
15. Εμφάνιση μηνύματος προς τον χρήστη
16. Κλείσιμο εγγραφής ημερολογίου
17. Η παρούσα ροή εργασίας τερματίζεται

Εναλλακτική ροή 1 : Σφάλμα στο βήμα 1

- 1.Μήνυμα λάθους για αποτυχημένη προσπάθεια εισαγωγής του πολίτη

Εναλλακτική ροή 2 : Σφάλμα στο βήμα 2

- 1.Εάν δεν συμπληρωθεί ορθά η φόρμα εμφανίζεται το λάθος που εντοπίστηκε.

Εναλλακτική ροή 3 : Σφάλμα στο βήμα 3

Εάν για κάποιο λόγο εμφανιστεί κάποιο πρόβλημα στην βάση δεδομένων ή στα SOAP μηνύματα που κρατά όλες τις συναλλαγές εμφανίζεται μήνυμα λάθους στον χρήστη με την αιτιολογία τεχνικού προβλήματος

Εναλλακτική ροή 4 : Σφάλμα στο βήμα 6

1. Λάθος ταυτοποίηση του πολίτη
2. Προβολή κατάλληλου μηνύματος στον πολίτη για το λάθος
- 3.Κλείσιμο εγγραφής ημερολογίου
- 4.Η ροή εργασίας τερματίζεται

Εναλλακτική ροή 5 : Σφάλμα στο βήμα 8

- 1.Λάθος ταυτοποίηση πελάτη
- 2.Προβολή κατάλληλου μηνύματος στον πολίτη για το λάθος
- 3.Κλείσιμο εγγραφής ημερολογίου
- 4.Η ροή εργασίας τερματίζεται

Εναλλακτική ροή 6 : Σφάλμα στο βήμα 8

1. Χρόνος κατοικίας στον δήμο λιγότερο απο 2 χρόνια
2. Προβολή κατάλληλου μηνύματος στον πολίτη για το λάθος
3. Κλείσιμο εγγραφής ημερολογίου
4. Η ροή εργασίας τερματίζεται

Εναλλακτική ροή 7 : Σφάλμα στο βήμα 12

- 1.Αναίρεση προηγούμενης διαδικασίας
- 2.Προβολή μηνύματος λάθους
- 3.Κλείσιμο εγγραφής ημερολογίου
- 4.Η ροή εργασίας τερματίζεται

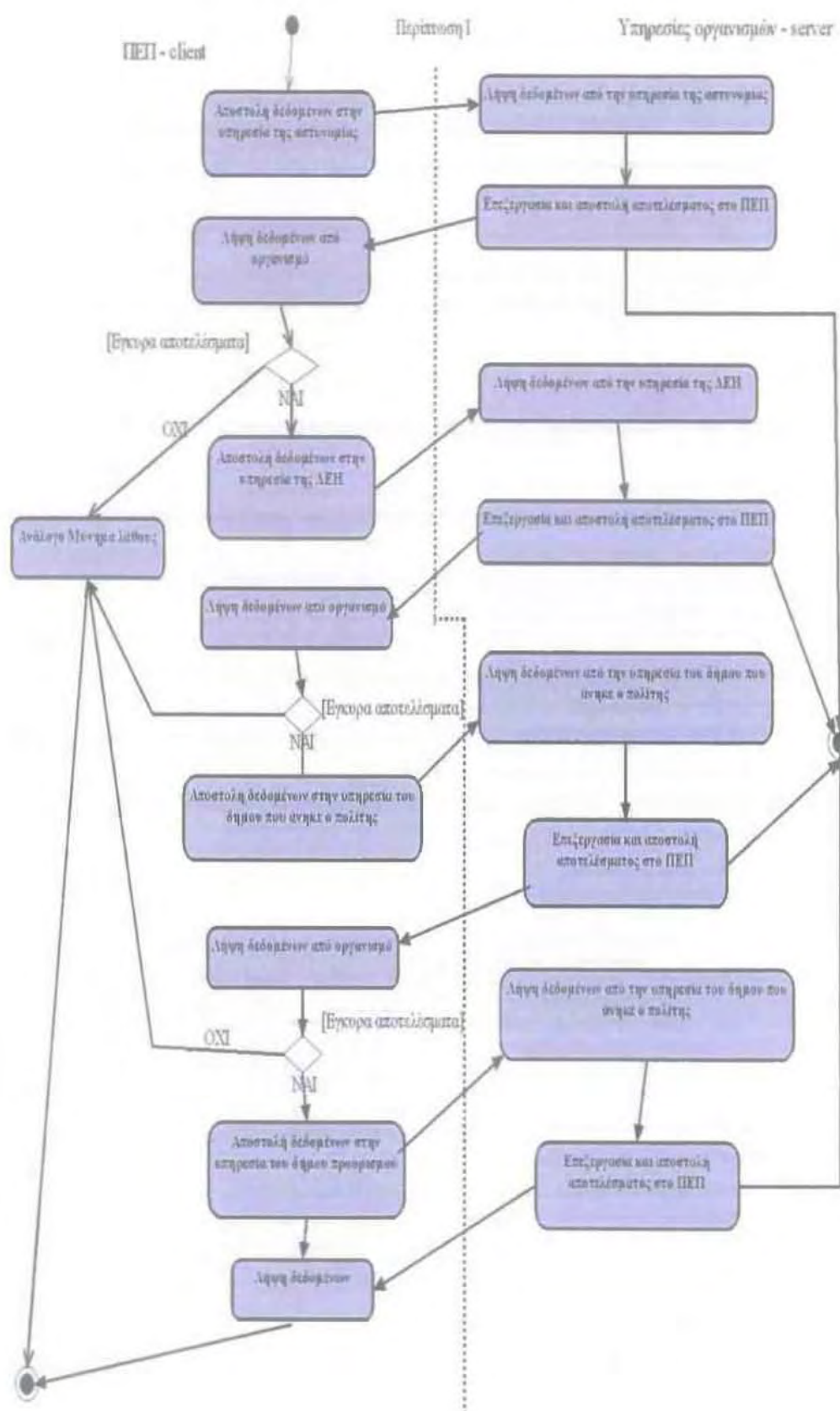
3.5.2.2 Ροή εργασίας : «Άνοιγμα Οικογενειακής Μερίδας στον δήμο που ανήκουμε »

Περίπτωση Ι)

Αρχικά γίνεται κλήση της υπηρεσίας του οργανισμού «Αστυνομία» προκειμένου να ελεγχθεί η φυσική ύπαρξη και αντιστοιχία των δεδομένων που δίνονται από τον πολίτη που επιθυμεί να κάνει την αίτηση για άνοιγμα οικογενειακής μερίδας. Τα στοιχεία που ελέγχονται είναι ο αριθμός ταυτότητας σε σχέση με το όνομα και επίθετο του δημότη. Στην συνέχεια εάν υπάρξει ορθή αντιστοιχία καλείται η υπηρεσία του οργανισμού ΔΕΗ. Αυτή τη φορά απλά θέλουμε μια επιβεβαίωση από την ΔΕΗ για την κατοικία του πολίτη (όπως ορίζεται από την ροή εργασίας του ΚΕΠ). Αφού πιστοποιηθούν τα στοιχεία τότε καλείται ο δήμος στον οποίο θέλει να ανοιχτεί οικογενειακή μερίδα. Προφανώς αυτός ο δήμος, είναι ο δήμος στον οποίο ήδη είχε κάποια μερίδα, επομένως άνηκε στον δήμο αυτό. Τότε αφού γίνει ο απαραίτητος έλεγχος των στοιχείων γίνεται διαγραφή του χρήστη από την υπάρχουσα βάση και τέλος έχουμε την τελική προσθήκη στην βάση δεδομένων όπου καταχωρούνται οι οικογενειακές μερίδες και επιστρέφεται ένας μοναδικός αριθμός της μερίδας που μόλις ανοίχτηκε.

Γενική εικόνα :

Σχήμα 3.5 – Άνοιγμα Οικογενειακής μερίδας
"Άνοιγμα Οικογενειακή Μερίδα: στον δήμο που ανήκουμε "



Περίπτωση II)

Ο πολίτης, αφού αναγνωριστεί από το σύστημα (μέσω του ονόματος χρήστη και κωδικού) και μπει στην υπηρεσία αυτή, θα συμπληρώσει την απαραίτητη φόρμα. Αν ο έλεγχος ορθότητας των δεδομένων είναι επιτυχής τότε καλείται η ροή εργασίας που περιγράφηκε προηγουμένως. Επιπλέον σε περίπτωση επιτυχίας της διαδικασίας και για κάθε αίτηση που γίνεται κρατάται ένα αρχείο log καθώς επίσης στέλνεται ένα e mail στον πολίτη, για την επικύρωση των δεδομένων και τέλος κρατούνται κάποια δεδομένα του πολίτη σε μια βάση δεδομένων. Σε κάθε περίπτωση, αποτυχίας ή επιτυχίας έχουμε τα κατάλληλα μηνύματα προς τον πολίτη.

Το σχήμα είναι ακριβώς ίδιο όπως αυτού της μεταδημότευσης στην αντίστοιχη περίπτωση .

Ροή και των δύο περιπτώσεων υπό την μορφή βημάτων και εναλλακτικών ροών :

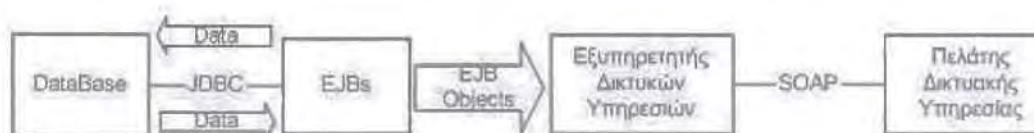
Ακριβώς όπως την προηγούμενη υπηρεσία

3.5.3 Περιγραφή αρχιτεκτονικής εφαρμογών Οργανισμών και ΠΕΠ

Οι εφαρμογές των οργανισμών περιλαμβάνουν πληθώρα τεχνολογιών. Θα δοθεί μια περίληψη παρουσιάζοντας τα βασικά συστατικά και στη συνέχεια θα παρουσιαστούν οι λεπτομέρειες αυτών.

Οι οργανισμοί διαθέτουν βάση δεδομένων, εξυπηρετητή εφαρμογών (application server), δικτυακό εξυπηρετητή (web server) και εξυπηρετητή δικτυακών υπηρεσιών (Axis server).

Η μεταφορά των δεδομένων προς το ΠΕΠ ακολουθεί την πορεία που φαίνεται στο παρακάτω σχήμα. Αναλυτικά τα EJBs αντικείμενα δημιουργούνται από τα δεδομένα των πινάκων στις βάσεις δεδομένων ή δημιουργούν δεδομένα και τα τοποθετούν στις βάσεις δεδομένων. Τα αντικείμενα ή γενικότερα τα αποτελέσματα διαφόρων υπολογισμών προωθούνται στον εξυπηρετητή δικτυακών υπηρεσιών ο οποίος αναλαμβάνει να τα μετατρέψει σε XML μορφή και να τα στείλει τους πελάτες (ΠΕΠ)

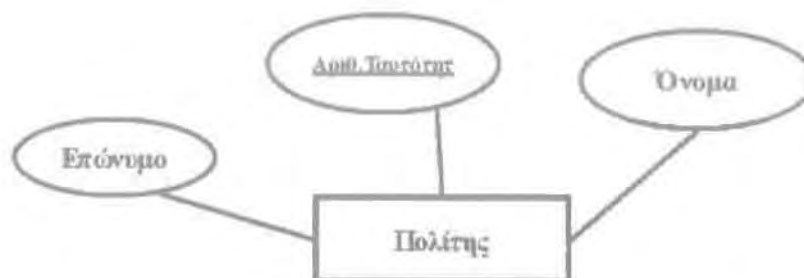


Σχήμα 3.6 - Μεταφορά δεδομένων από Βάση δεδομένων στην Δικτυακή υπηρεσία

Θα ξεκινήσουμε την ανάλυση από την βάση δεδομένων και τα EJBs των οργανισμών και ακολουθώντας την ροή του σχήματος θα συνεχίσουμε. Αφού τελειώσουμε με τους Οργανισμούς θα αναφερθούμε στην αρχιτεκτονική του ΠΕΠ το οποίο ίσως είναι και το πιο ενδιαφέρον από μορφή πολυπλοκότητας, μιας και περιλαμβάνει εκτός όλων των άλλων και διαχειριστική μορφή.

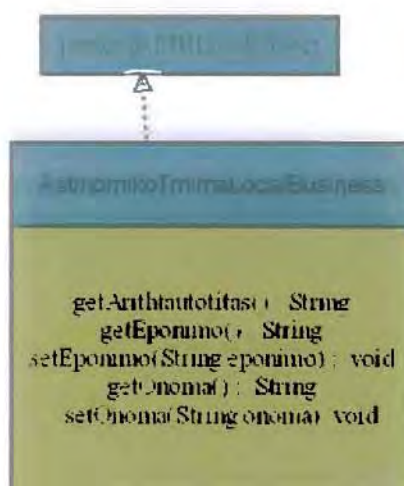
3.5.3.1 Βάση δεδομένων Υπουργείου δημόσιας Τάξεως – Αστυνομία

Ο πίνακα της βάσης δεδομένων που χρησιμοποιεί το υπουργείο δημοσίας τάξης-Αστυνομία φαίνεται στο παρακάτω σχήμα



Σχήμα 3.7 –Διάγραμμα Βάσης δεδομένων Αστυνομικού Τμήματος

Το υπουργείο δημόσιας Τάξης - Αστυνομία περιλαμβάνει έναν πίνακα στον οποίο καταχωρεί μόνο αυτά τα στοιχεία (θα μπορούσε να είναι το view μιας μεγαλύτερης βάσης) απλώς για να προσφέρει (σε άλλους φορείς) την ύπαρξη και βασική αντιστοιχία του αριθμού μιας ταυτότητας με κάποιο πολίτη. Τα πεδία είναι προφανής: Αριθμός ταυτότητας, Επώνυμο, Όνομα πολίτη.
Τα EJBs που δημιουργούνται από την αντίστοιχη βάση δεδομένων και αφορούν στοιχεία πολιτών είναι



Πίνακας 1 – EJBs Αστυνομία

3.5.3.2 Βάση δεδομένων Οργανισμού ΔΕΗ

Ο Οργανισμός ΔΕΗ έχει ένα πίνακα με 6 χαρακτηριστικά. Τον αριθμό Πελάτη, το οποίο αποτελεί και το πρωτεύων κλειδί, το Όνομα, το Επίθετο, τον Δήμο και το Έτος καταχώρησης. Στο πεδίο «δήμος» τοποθετείται ο δήμος στον οποίο κατοικεί μόνιμα ο πελάτης ενώ στο πεδίο «έτος Καταχώρησης» τοποθετείται το έτος από το οποίο ο πελάτης – πολίτης μένει στον δήμο αυτό. Ο πίνακας αυτός χρησιμοποιείται για την αποθήκευση των πελατών της ΔΕΗ προσφέροντας κυρίως μια «απόδειξη» του μόνιμου τόπου κατοικίας .



Σχήμα 3.8 – Διάγραμμα Βάσης δεδομένων Οργανισμού ΔΕΗ

Τα EJBs που δημιουργούνται από την αντίστοιχη βάση δεδομένων και αφορούν στοιχεία πελατών στην ΔΕΗ είναι

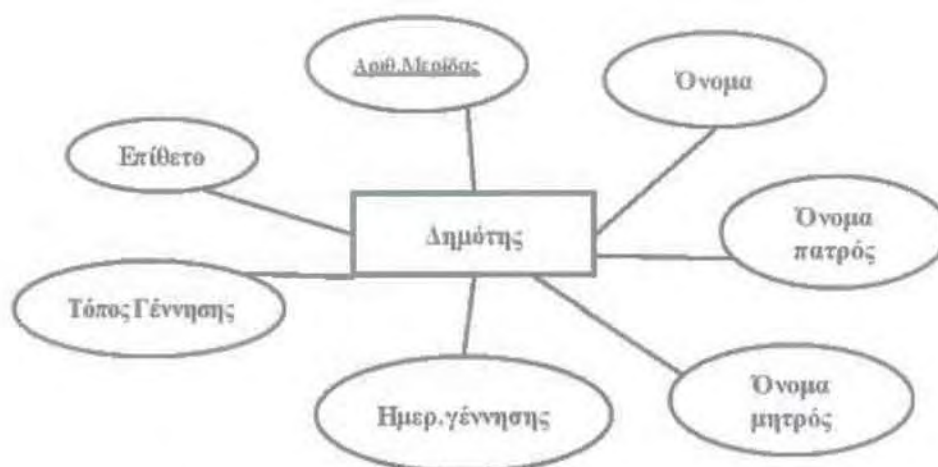


Πίνακας 2 – EJBs ΔΕΗ

3.5.3.2 Βάση δεδομένων των Δήμων

Η βάση δεδομένων του δήμου (και κάθε δήμου που θα ενταχθεί στο σύστημα Ηλεκτρονικής Διακυβέρνησης) αποτελείται από 3 πίνακες κάθε ένας από αυτούς υπάρχει για να ανταποκρίνεται στις ανάγκες όλων των δικτυακών υπηρεσιών που παρέχονται. Αναλυτικά στον πρώτο πίνακα έχουμε τα εξής πεδία : Αριθμός μερίδας, που αποτελεί και το πρωτεύων κλειδί, την ημερομηνία γέννησης, τον τόπο γέννησης, όνομα μητέρας, όνομα πατέρα και τέλος το όνομα και το επώνυμο του πολίτη. Ο

πίνακας αυτός χρησιμοποιείται προκειμένου να αποθηκεύονται τα στοιχεία του δημότη στον νέο δήμο που επιθυμεί την μεταδημότευση.



Σχήμα 3.9 – Διάγραμμα Βάσης δεδομένων στοιχείων πολιτών σε Δήμους

Εδώ αξίζει να τονιστεί ότι ανάλογα με τον δήμο θα μπορούσαν πιθανώς τα ονόματα της εκάστοτε βάσης να είναι διαφορετικά θα μπορούσαν να είναι όμως και ακριβώς τα ίδια. Επειδή όμως η εργασία αναπτύσσεται εντός του ίδιου υπολογιστή για τους δύο δήμους δόθηκαν διαφορετικά ονόματα στις βάσεις.

Τα EJBs που δημιουργούνται από την αντίστοιχη βάση δεδομένων και αφορούν στοιχεία των δημοτών είναι :



Πίνακας 3 – EJBs , στοιχεία δημοτών

Ο επόμενος πίνακας αναφέρεται στους παλαιούς δημότες και υπάρχει προκειμένου να υπάρχει λογική συνέχεια στις κινήσεις των δημοτών. Σχηματικά φαίνεται παρακάτω :



Σχήμα 3.10 – Διάγραμμα Βάσης δεδομένων πρώην Δημοτών

Προφανώς είναι ίδιος με το προηγούμενο με την εξής διαφορά. Υπάρχει ένα επιπλέον πεδίο με όνομα Aitiadiagrafis στο οποίο σημειώνεται ο λόγος για τον οποίο βρίσκεται μια εγγραφή σε αυτόν.

Τα EJBs που δημιουργούνται από την αντίστοιχη βάση δεδομένων και αφορούν στοιχεία των παλιών δημοτών είναι :



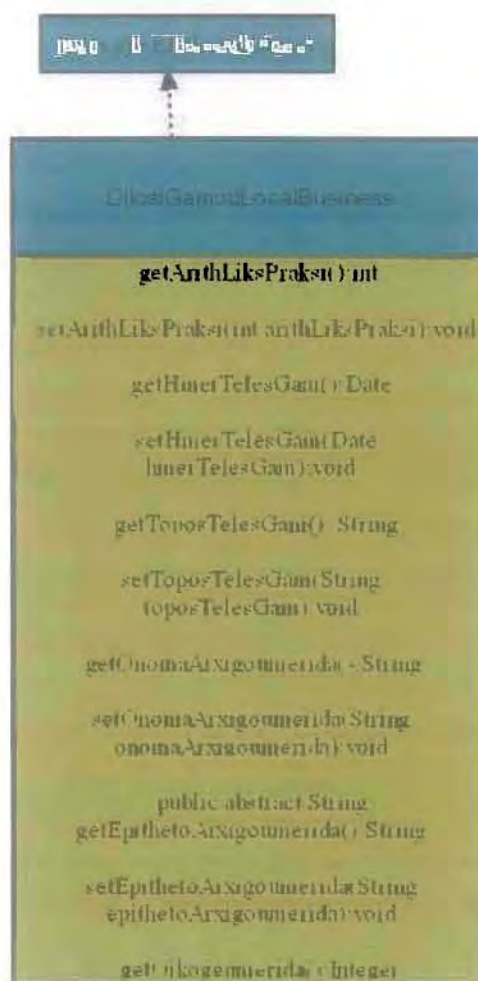
Πίνακας 4 – EJBs , στοιχεία παλιών δημοτών

Τέλος έχουμε και ακόμα έναν πίνακα ο οποίος σχετίζεται με το άνοιγμα οικογενειακής μερίδας στον δήμο.



Σχήμα 3.11 –Διάγραμμα Βάσης δεδομένων γάμων

Τα EJBs που δημιουργούνται από την αντίστοιχη βάση δεδομένων και αφορούν στοιχεία για την δήλωση γάμου είναι :



Πίνακας 5 – EJBs , δήλωση γάμου

3.5.3 Δικτυακές υπηρεσίες Οργανισμών

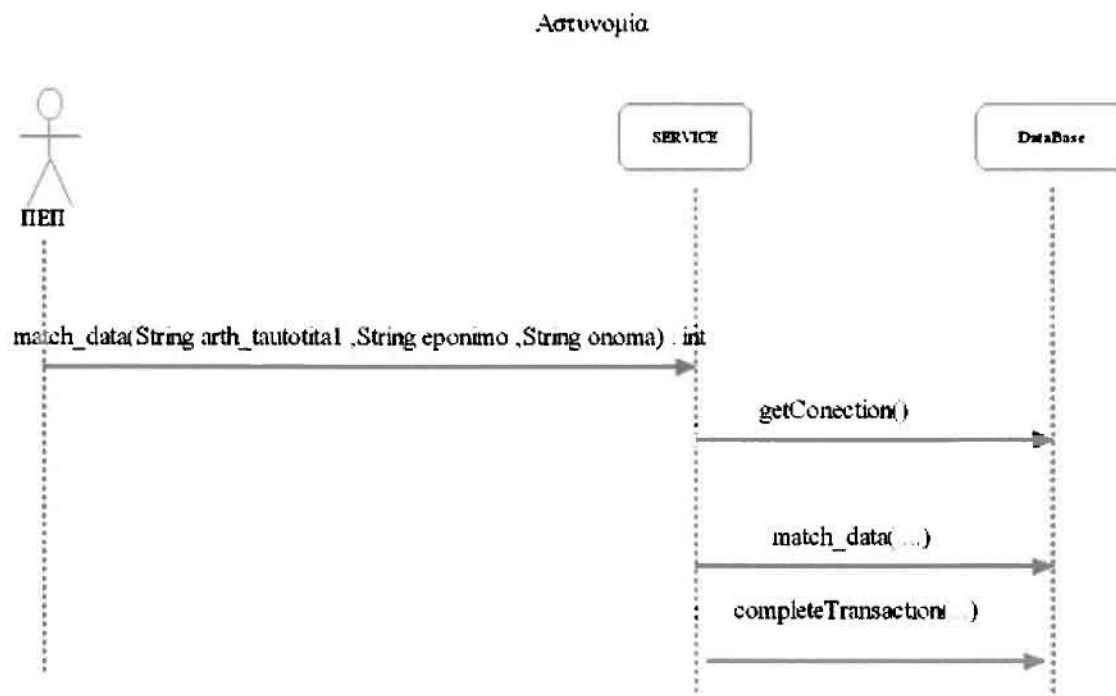
3.5.3.1 Δικτυακή υπηρεσία υπουργείου δημοσίας τάξης – Αστυνομία

Η μοναδική υπηρεσία που προσφέρει η Αστυνομία είναι η εξής :

I)match_data(String arth_tautotital, String eponimo ,String onoma) : int

Δέχεται σαν ορίσματα έναν αριθμό ταυτότητας, επώνυμο και όνομα και αναζητά στην βάση δεδομένων της αστυνομίας προκειμένου να διαπιστώσει την ύπαρξη της ταυτότητας αυτής και εν συνεχεία συγκρίνει εάν το όνομα και το επίθετο που έχει δοθεί, αντιστοιχεί με αυτό της βάσης.

Συνεργατικό Διάγραμμα



Σχήμα 3.12 –Συνεργατικό Διάγραμμα – Αστυνομία

Κωδικοί Επιστροφής

- Σε περίπτωση ορθής αντιστοιχίας επιστρέφεται ο κωδικός με αριθμό 400.
- Σε περίπτωση στην οποία υπάρχει μεν ο αριθμός ταυτότητας στην βάση αλλά δεν υπάρχει αντιστοιχία ονόματος και επιθέτου με αυτά που έχουν δοθεί τότε επιστρέφεται ο κωδικός 425.
- Σε περίπτωση που δεν υπάρχει καν στην βάση δεδομένων ένας τέτοιος αριθμός ταυτότητας τότε επιστρέφεται ο κωδικός 440.

- Σε περίπτωση που υπάρξει κάποιο απροσδόκητο πρόβλημα εμφανίζεται ο κωδικός 450.

3.5.3.2 Δικτυακές υπηρεσίες Οργανισμού ΔΕΗ

Ο οργανισμός ΔΕΗ προφέρει δύο υπηρεσίες ώστε να ανταποκρίνεται στις περισσότερες υπηρεσίες - διαδικασίες οι οποίες απαιτούν κάποιο στοιχείο απόδειξης μέσω της ΔΕΗ.

I) match_data(Integer getArithmospelati ,String epitheto ,String onoma,String dhmos) : int

Έχει ακριβώς την ίδια λογική που περιγράφηκε προηγουμένως, στην περίπτωση της αστυνομίας.

Κωδικοί Επιστροφής

- Σε περίπτωση ορθής αντιστοιχίας επιστρέφεται ο κωδικός με αριθμό 400.
- Σε περίπτωση στην οποία υπάρχει μεν ο αριθμός αυτός στην βάση αλλά δεν υπάρχει αντιστοιχία ονόματος και επιθέτου με αυτά που έχουν δοθεί τότε επιστρέφεται ο κωδικός 425.
- Σε περίπτωση που δεν υπάρχει καν στην βάση δεδομένων ένας τέτοιος αριθμός ταυτότητας τότε επιστρέφεται ο κωδικός 440.
- Σε περίπτωση που υπάρξει κάποιο απροσδόκητο πρόβλημα εμφανίζεται ο κωδικός 450.

Το συνεργατικό διάγραμμα της υπηρεσίας είναι αντίστοιχο με αυτό του σχήματος 3.12

II) check_time_passed(Integer getArithmospelati ,String epitheto ,String onoma,String dhmos,int how_time) : int

Καλώντας αυτή την υπηρεσία καλείται έμμεσα και η *match_data(...)*. Και αυτό διότι η σημασιολογία της *check_time_passed* είναι η εξής :

Αφού ελεγχθεί η ορθή αντιστοιχία του πολίτη με την βάση της ΔΕΗ ,σχετικά με το όνομα, επίθετο και δήμο, υπολογίζει τον χρόνο που έχει απέλθει από την χρονιά που γίνεται η αίτηση για μεταδημότευση με αυτή που είναι καταχωρημένη στην βάση σαν χρονιά εγγραφής.

Κωδικοί Επιστροφής

- Επιστρέφει 400 αν είναι πάνω από το χρόνο που δίνεται σαν όρισμα.
- Επιστρέφει 410 αν είναι κάτω από το χρόνο που δίνεται σαν όρισμα.
- Σε περίπτωση στην οποία υπάρχει μεν ο αριθμός αυτός στην βάση αλλά δεν υπάρχει αντιστοιχία ονόματος και επίθετος με αυτά που έχουν δοθεί τότε επιστρέφεται ο κωδικός 425.
- Σε περίπτωση που δεν υπάρχει καν στην βάση δεδομένων ένας τέτοιος αριθμός ταυτότητας τότε επιστρέφεται ο κωδικός 440.
- Σε περίπτωση που υπάρξει κάποιο απροσδόκητο πρόβλημα εμφανίζεται ο κωδικός 450.

Το συνεργατικό διάγραμμα της υπηρεσίας είναι αντίστοιχο με αυτό του σχήματος 3.12

3.5.3.3 Δικτυακές υπηρεσίες Δήμων

Οι δικτυακές υπηρεσίες των δήμων αποτελούν ίσως και τις πιο σημαντικές διότι εμπεριέχουν και την περισσότερη λειτουργικότητα που έχει σχέση με το σύστημα.

I) *add_to_municipality* (*String onoma,String epitheto,String onoma_patros,String onoma_mitros,java.sql.Date mera_gennisis,String topos_gennisis,Integer id*) : *int*

Αυτή η υπηρεσία σκοπό έχει την προσθήκη του πολίτη στην βάση δεδομένων που αφορά την μεταδημότευση. Εάν τελικά είναι επιτυχής η προσθήκη, επιστρέφεται ένας μοναδικός κωδικός ο οποίος είναι ο (αύξων) αριθμός μερίδας που πλέον έχει ο δημότης.

Κωδικοί Επιστροφής

- Επιστρέφει τον Αριθμό Μερίδας του νέου δημότη αν δεν έχει παρουσιαστεί κάποιο σφάλμα
- Επιστρέφει -1 σε περίπτωση που για κάποιο λόγο υπάρχει ήδη ένα τέτοιο στιγμιότυπο
- Σε περίπτωση που υπάρξει κάποιο απροσδόκητο πρόβλημα εμφανίζεται ο κωδικός 450.

Το συνεργατικό διάγραμμα της υπηρεσίας είναι αντίστοιχο με αυτό του σχήματος 3.12

II) *del_from_municipality* (*String onoma,String epitheto,String onoma_patros,String onoma_mitros,java.sql.Date mera_gennisis,String topos_gennisis,Integer arithMerida,String reason*) : *int*

Η υπηρεσία αυτή έχει σκοπό την διαγραφή κάποιου δημότη από τον δήμο στον οποίο βρίσκεται. Αφού γίνει πρώτα έλεγχος για την ύπαρξη του πολίτη αυτού στην βάση στην συνέχεια γίνεται η διαγραφή του. Μόλις γίνει η διαγραφή αυτή γίνεται προσθήκη (του πολίτη που διαγράφηκε) σε μία άλλη βάση δεδομένων η οποία είναι ίδια με την προηγούμενη με τις εξής δύο διαφορές. Καταρχήν η προσθήκη γίνεται με τον υπάρχων αριθμό Μερίδας που είχε ο δημότης (δεν υπάρχει περίπτωση διπλής ύπαρξης μιας και είναι παντοτινά μοναδικός) και τέλος υπάρχει μια επιπλέον εγγραφή στην οποία τοποθετείται ο λόγος για τον οποίο έγινε η διαγραφή από την κύρια, ισχύοντα, βάση δεδομένων. Οι λόγοι μπορεί να είναι η μεταδημότευση ή και ο θάνατος του δημότη.

Κωδικοί Επιστροφής

- Σε περίπτωση επιτυχής διαγραφής του δημότη επιστρέφεται 400.
- Σε περίπτωση λάθους κατά την διαγραφή επιστρέφεται -2.
- Σε περίπτωση λάθους κατά την προσθήκη στην βάση με τους διαγραφθέντες επιστρέφεται -3.
- Σε περίπτωση απροσδόκητου σφάλματος επιστρέφεται -1.

Το συνεργατικό διάγραμμα της υπηρεσίας είναι αντίστοιχο με αυτό του σχήματος 3.12

III) *add_marriage(String onoma,String epitheto, int arith_liksi_prak_gamou, java.sql.Date mera_tes_gamou,String topos_tes_gamou,Integer arithMerida) : int*

Η υπηρεσία έχει σκοπό το άνοιγμα οικογενειακής μερίδας για έναν δημότη. Πρώτα αναζητάτε ο πολίτης στην βάση δεδομένων του δήμου, αφού βρεθεί τότε διαγράφεται από αυτήν και γίνεται η προσθήκη στην νέα βάση επιστρέφοντας έναν νέο μοναδικό Αριθμό οικογενειακής Μερίδας

Κωδικοί Επιστροφής

- Σε περίπτωση επιτυχίας επιστρέφεται ο Αριθμός Οικογενειακής Μερίδας.
- Σε περίπτωση που παρουσιάστη σφάλμα επιστρέφεται -1.

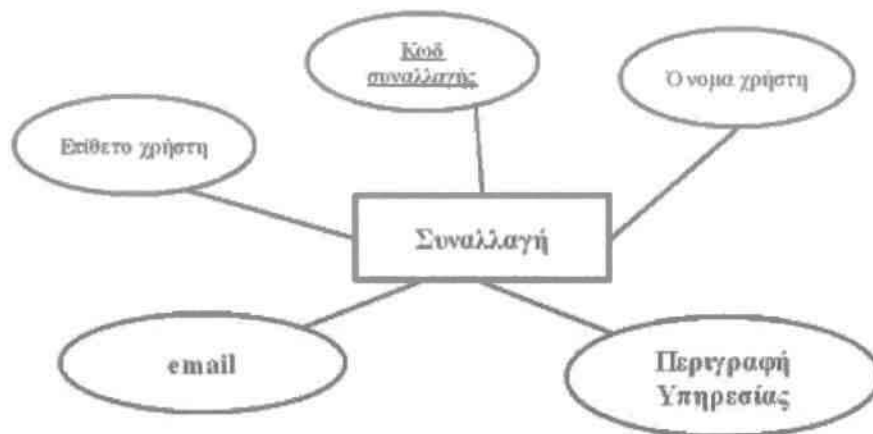
Το συνεργατικό διάγραμμα της υπηρεσίας είναι αντίστοιχο με αυτό του σχήματος 3.12

3.5.4 Περιγραφή αρχιτεκτονικής εφαρμογής ΠΕΠ

Το ΠΕΠ είναι η πιο σύνθετη εφαρμογή διότι περιλαμβάνει όλων των ειδών τις τεχνολογίες προκειμένου να επιτύχει την Ηλεκτρονική Διακυβέρνηση. Περιέχει SOAP engine προκειμένου να στέλνει και να δέχεται τα απαραίτητα μηνύματα , jsp και servlet engines, μιας και μέσω των servlets επιτυγχάνεται η επιχειρηματική λογική του συστήματος ενώ μέσω των jsp επιτυγχάνουμε την προβολή των αποτελεσμάτων μας. Τέλος υπάρχουν και δύο βάσεις δεδομένων ,μία στην οποία κρατάται ένα μοναδικός κωδικός συναλλαγής, το όνομα, το επίθετο και το e mail του δημότη που έκανε την αίτηση καθώς και μια περιγραφή για την υπηρεσία που αντιστοιχεί και τέλος μια βάση στην οποία υπάρχουν τα στοιχεία των δημοτών που είναι μέλη της Πύλης και αυτά είναι το username, password, όνομα, επίθετο και ένα πεδίο που αφορά τους διαχειριστές του συστήματος.

Οι βάσεις φαίνονται στο παρακάτω σχήμα :





Σχήμα 3.13 – Διαγράμματα Βάσεων δεδομένων,Κωδικός Συναλλαγής,Λογ. Χρήστη

Ο κωδικός συναλλαγής είναι μοναδικός για κάθε δοσοληψία κάποιου Πολίτη με την υπηρεσία αυτή και αποτελεί ένα τεκμήριο για την πράξη που πραγματοποιήθηκε από αυτόν.

Για κάθε πράξη που πραγματοποιείται είτε από ένα μέλος της Πύλης είτε από έναν Διαχειριστή συστήματος κρατούνται κάποια στοιχεία σε ειδικά αρχεία με την μορφή των logs.

Η εφαρμογή ΠΕΠ είναι επιφορτισμένη, όπως αναφέρθηκε και σε προηγούμενες ενότητες να συνδυάσει όλα τα αντικείμενα «επιχειρηματικής λογικής» (δηλ ανάκτηση δεδομένων από τις δικτυακές εφαρμογές και σύνθεση αυτών) και να τα παρουσιάσει στον πολίτη μέσω μιας Πύλης (portal) με την χρήση δυναμικών σελίδων ενημερώνοντας τον για κάθε πιθανή κατάσταση στην οποία βρίσκεται το σύστημα.

3.6 Γενική Περιγραφή της Πύλης Εξυπηρέτησης Πολιτών

Γενικές απαιτήσεις Πύλης :

Όπως περιγράφηκαν σε προηγούμενη υποενότητα (Γενική Περιγραφή ιδιοτήτων μιας Πύλης Ηλεκτρονικής Διακυβέρνησης)

Υλοποιημένες Απαιτήσεις :

- Υπηρεσίες Ασφάλειας
- Δυνατότητα ψηφοφορίας μέσω ηλεκτρονικών δημοψηφισμάτων.
- Άμεση ενημέρωση των πολιτών με την χρήση κατάλληλων μηχανισμών για την πρόοδο και το αποτέλεσμα κάθε αίτησης του.
- Αδιάλειπτη και ασφαλής λειτουργία με υψηλή διαθεσιμότητα.
- Ανοιχτή αρχιτεκτονική για εύκολη επέκταση του συστήματος.
- Παραμετροποίηση της Πύλης.

Οι υπόλοιπες απαιτήσεις όπως ύπαρξη forum, πολυγλωσσία, WAP κτλ υπάρχουν σαν σχεδιαστικές οντότητες στην πύλη αλλά δεν είναι υλοποιημένες

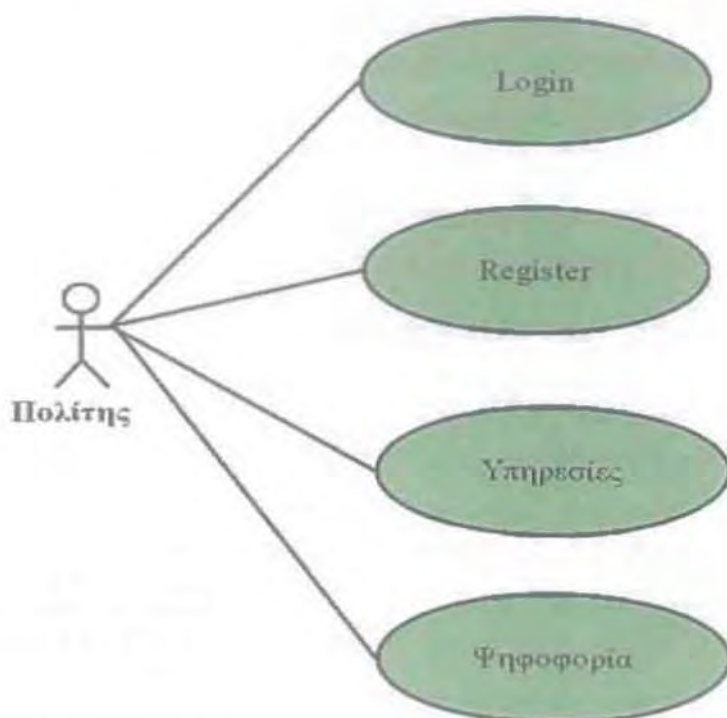
3.6.1 Χρήστες

Οι χρήστες χωρίζονται σε 3 κατηγορίες όπως αναφέρθηκε σε προηγούμενη ενότητα (Γενική Περιγραφή ιδιοτήτων μιας Πύλης Ηλεκτρονικής Διακυβέρνησης) έχοντας ακριβώς τους ρόλους που περιγράφονται σε αυτήν.

3.6.1.1 Χρήστες και Λειτουργίες του ΠΕΠ

Στην ενότητα αυτή θα αναλυθούν όλες οι λειτουργίες της Πύλης για κάθε είδους χρήστη. Ο διαχωρισμός θα γίνει μεταξύ των Πολιτών (εγγεγραμμένων και μη) και του Διαχειριστή συστήματος

Περίπτωση Ι)



Σχήμα 3.14 – Περιπτώσεις Χρήσης του Πολίτη στην πύλη

Login :

Ο χρήστης εισάγει το όνομα χρήστη και τον κωδικό χρήστη στην αντίστοιχη φόρμα. Το σύστημα αφού συνδεθεί με την κατάλληλη βάση δεδομένων και κάνει τους απαραίτητους ελέγχους έχουμε :

Σε περίπτωση επιτυχίας εμφανίζονται τα εξής :

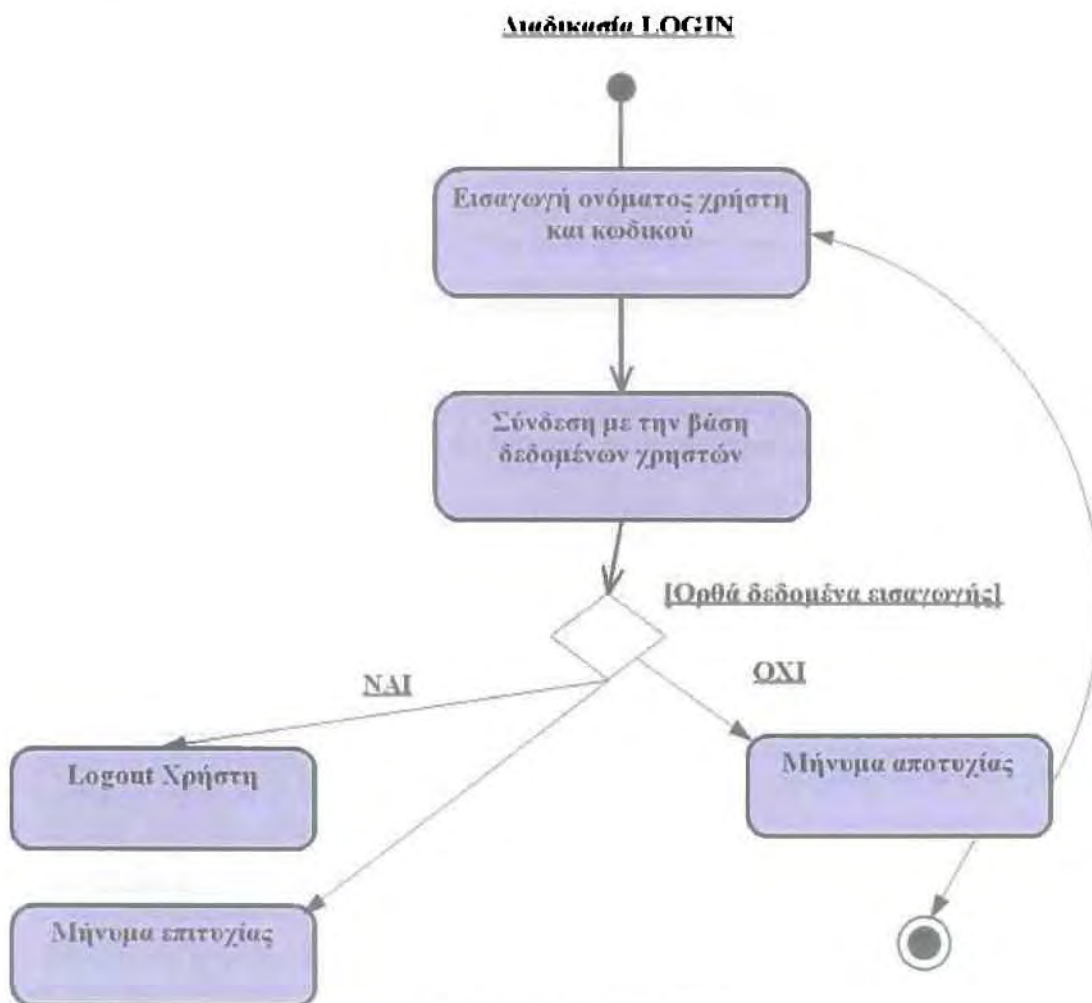
- Είστε login : "Όνομα"
- Logout Χρήστη

- Αλλαγή των στοιχείων μου (δεν είναι υλοποιημένο)

Όταν έχουμε ορθά εισαχθεί στο σύστημα τότε η αναγνώριση μας ισχύει για όλη την περιήγηση μας στην Πύλη έως ότου είτε κάνουμε εμείς Logout είτε παραμείνουμε ανενεργοί για 20 λεπτά.

Σε περίπτωση αποτυχίας εμφανίζονται

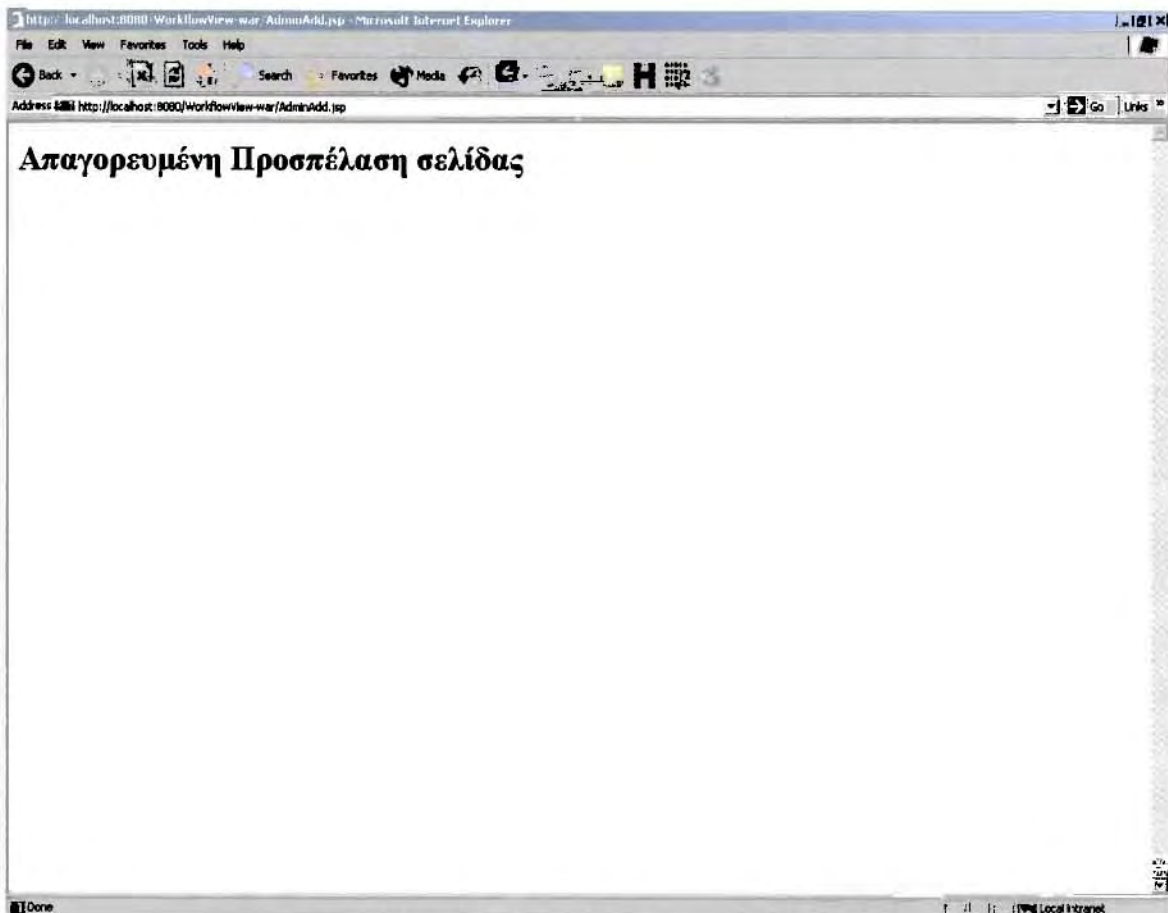
- Αποτυχημένη προσπάθεια εισαγωγής
- Ξέχασα τον κωδικό μου (δεν είναι υλοποιημένο)



Σχήμα 3.15 – Διαδικασία Login

Εάν επιλεγθεί το **logout** τότε από τούδε και στο εξής δεν έχουμε την ιδιότητα του μέλους και ενεργούμε σαν guest

Εάν γίνει προσπάθεια από έναν κακόβουλο χρήστη να προσπελάσει σελίδες που απαιτούν Login τότε θα εμφανιστεί κατάλληλο μήνυμα.



Σχήμα 3.16 – Παράδειγμα απογορευμένης πρόσβασης

Register :

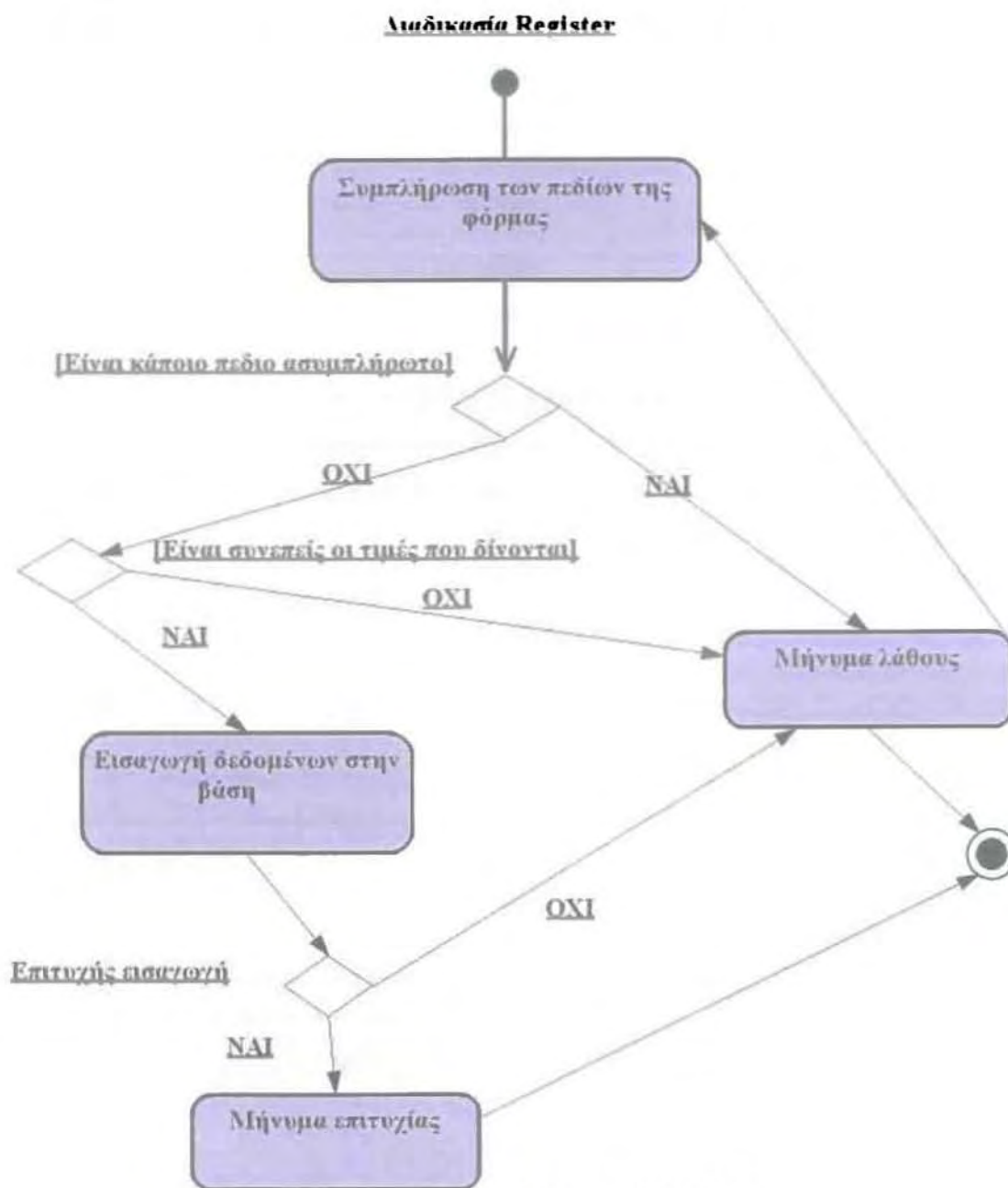
Ο χρήστης μπορεί να εισαχθεί στην Πύλη με την ιδιότητα του μέλους με την προϋπόθεση ότι πρώτα έχει συμπληρώσει την απαραίτητη φόρμα η οποία περιλαμβάνει τα εξής πεδία :

- Username
- Password
- Email
- Όνομα
- Επίθετο

Σε περίπτωση που κάποιο πεδίο δεν συμπληρωθεί ή δεν δοθεί ο συγκεκριμένος τύπος που απαιτείται (πχ συγκεκριμένη μορφή για email) τότε θα εμφανιστεί κατάλληλο μήνυμα που ενημερώνει τον χρήστη για το λάθος που έκανε.

Σε περίπτωση επιτυχίας ή αποτυχίας εμφανίζεται κατάλληλο μήνυμα που ενημερώνει τον χρήστη

Αφού τα δεδομένα που εισάγονται είναι τελικά ορθά τότε γίνεται σύνδεση με την αντίστοιχη βάση δεδομένων και εισάγονται τα δεδομένα σε αυτή.



Σχήμα 3.17 – Διαδικασία Εγγραφής Χρήστη

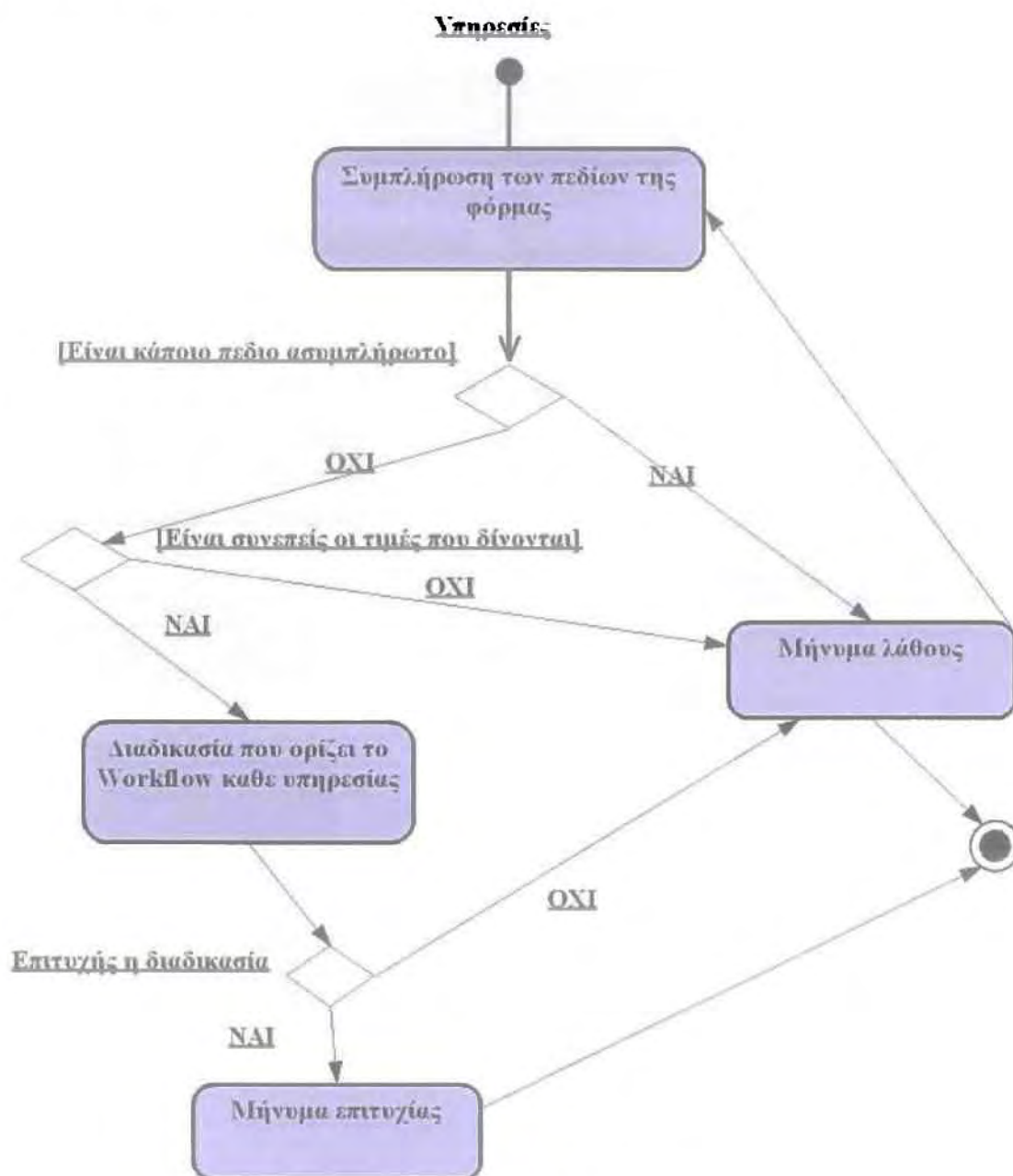
Υπηρεσίες :

Οι υπηρεσίες είναι αυτές όπως περιγράφηκαν διεξοδικά σε προηγούμενη ενότητα. Επομένως τα πεδία της φόρμας ποικίλουν από υπηρεσία σε υπηρεσία.

Και στις δύο υπηρεσίες έχουμε τις εξής ιδιότητες :

Για τα πεδία της φόρμας κάθε υπηρεσία γίνεται έλεγχος, αν έχουν συμπληρωθεί όλα και αν είναι με ορθούς τύπους (πχ σωστή μορφή του e-mail, αριθμούς όπου απαιτείται κτλ). Σε περίπτωση αποτυχίας της υπηρεσίας εμφανίζεται στον χρήστη μήνυμα λάθους που τον κατατοπίζει πλήρως για το πού εντοπίστηκε το λάθος ενώ σε περίπτωση επιτυχίας έχουμε αντίστοιχο μήνυμα .

Τέλος οι σελίδες που περιέχουν φόρμες καθώς και αυτές που εμφανίζουν το μήνυμα επιτυχίας βρίσκονται υπό την προστασία του πρωτοκόλλου SSL για την ασφάλεια των δεδομένων που στέλνει και δέχεται ο χρήστης.

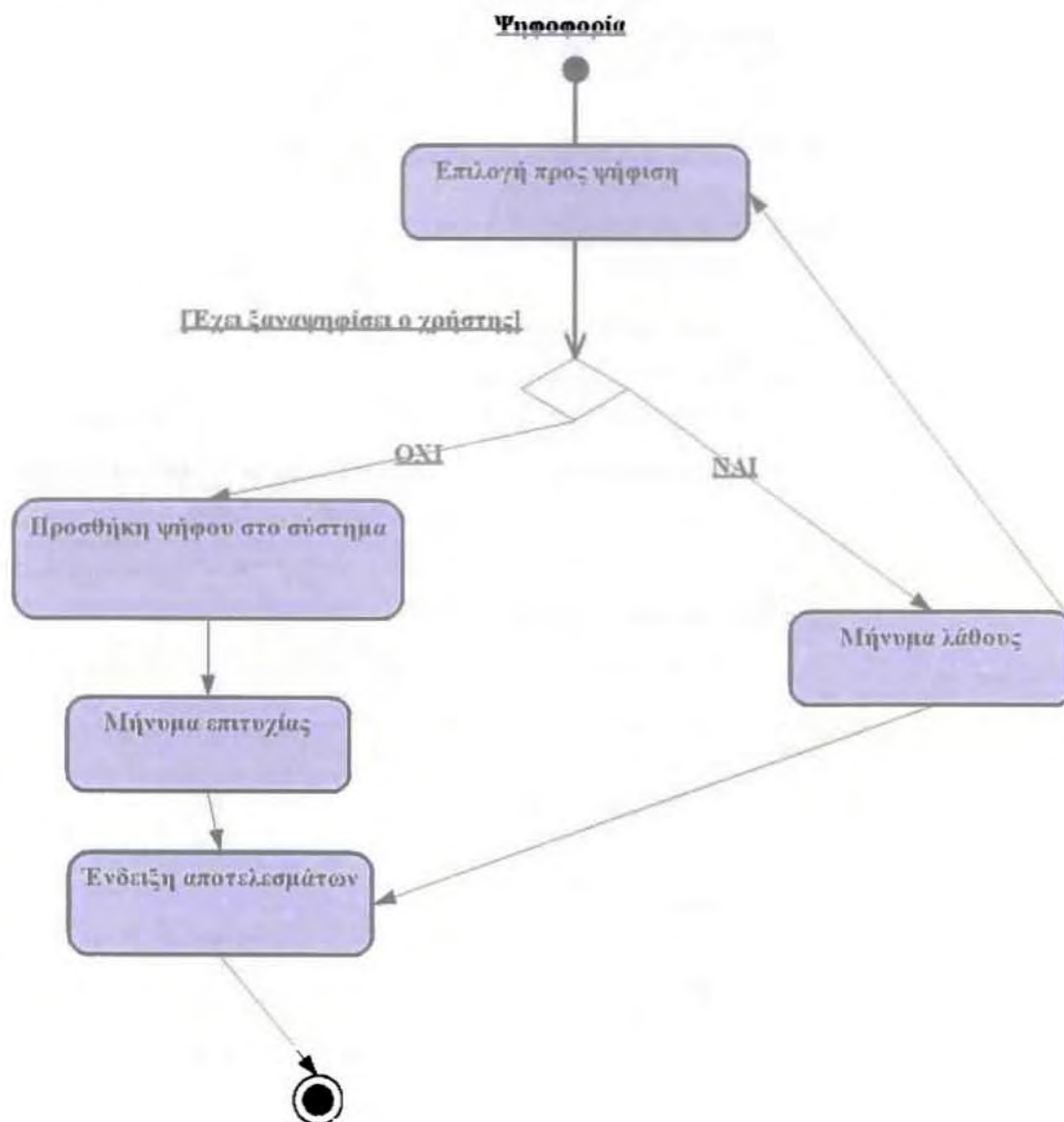


Σχήμα 3.18 – Διαδικασία Υπηρεσιών

Ψηφοφορία :

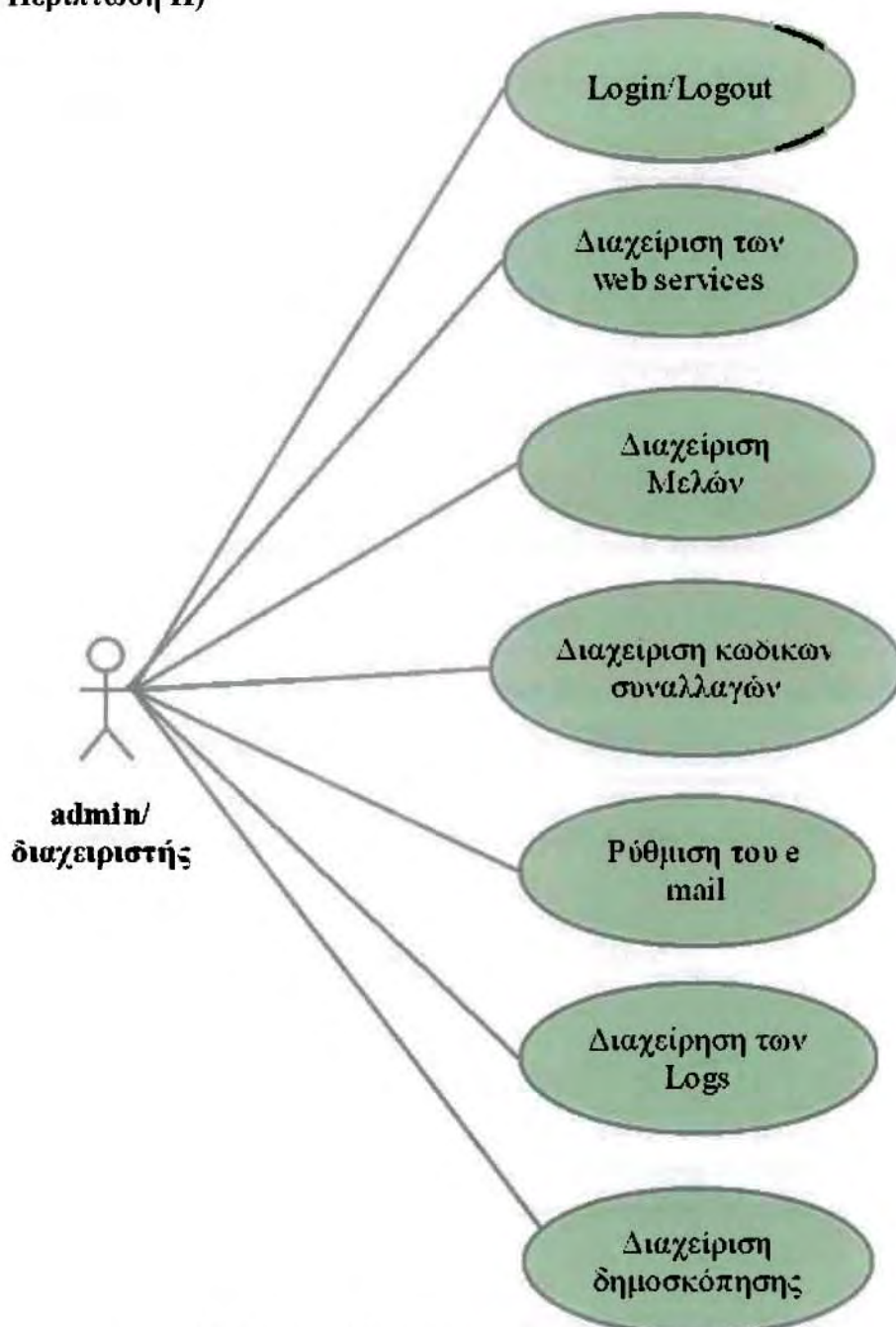
Οι χρήστες μπορούν να συμμετάσχουν στην διαδικασία ηλεκτρονικής ψηφοφορίας για μια δημοσκόπηση που λαμβάνει χώρα. Η ψηφοφορία αυτή δεν είναι επιστημονική επειδή δεν βασίζεται σε ασφαλής τεχνολογία για διπλοψηφίες.

Ο έλεγχος που γίνεται είναι απλά μέσω cookies.



Σχήμα 3.19 – Διαδικασία Ψηφοφορίας

Περίπτωση II)



Σχήμα 3.20 – Περιπτώσεις χρήσης Διαχειριστή Συστήματος

Από τούδε και στο εξής ο Διαχειριστής θα αναφέρεται ως **admin**.

Login: Όπως ο απλός χρήστης με την εξής διαφορά. Όταν γίνει ορθή αναγνώριση του χρήστη σαν admin/Διαχειριστή εμφανίζεται το εξής μήνυμα :

Είστε υπερχρήστης - admin Κονσόλα Διαχείρισης

Το «Κονσόλα Διαχείρισης» αποτελεί ένα link ώστε να μπορεί να εισαχθεί ο admin στην σελίδα διαχείρισης .

Διαχείριση των webservices

Η ενότητα αυτή χωρίζεται σε τρεις κατηγορίες/λειτουργίες

- Deploy / unDeploy Web services
- Παρακολούθηση SOAP μηνυμάτων
- Επισκόπηση των Web Services

Deploy / unDeploy Web services : Ο admin μπορεί συμπληρώνοντας μια φόρμα δύο πεδίων, ένα για την Διεύθυνση Υπηρεσίας και ένα με το Όνομα Αρχείου WSDD να ενεργοποιήσει ή να απενεργοποιήσει κάποια από τις υπηρεσίες που προσφέρουν οι οργανισμοί. Πάντα γίνεται έλεγχος για ορθότητα των δεδομένων που δίνονται και δίνεται κατάλληλο μήνυμα τόσο σε περίπτωση επιτυχίας ή αποτυχίας της πράξης. Δεν έχει ληφθεί μέριμνα για την ασφάλεια της διαδικασίας

Παρακολούθηση SOAP μηνυμάτων: Ο admin μπορεί να παρακολουθεί τα μηνύματα SOAP που ανταλλάσσονται μεταξύ των διαδικτυακών υπηρεσιών και του ΠΕΠ ώστε να μπορεί εύκολα να εντοπίσει τυχόν προβλήματα.

Επισκόπηση των Web Services: Παρουσιάζονται όλοι οι οργανισμοί που προσφέρουν τις δικτυακές τους υπηρεσίες στο ΠΕΠ μαζί με ένα link προς την σελίδα που παρουσιάζει τις υπηρεσίες που είναι on line αυτή την στιγμή ώστε ο admin να μπορεί να έχει επίγνωση αυτών.

Διαχείριση των Μελών

Η ενότητα αυτή χωρίζεται σε τρεις κατηγορίες/λειτουργίες :

- Διαγραφή account μέλους
- Προσθήκη Διαχειριστή Υπηρεσιών
- Προβολή στοιχείων κάποιου μέλους

Διαγραφή account μέλους : Ο admin δύναται να διαγράψει έναν χρήστη από την βάση δεδομένων που κρατούνται τα μέλη για διάφορους λόγους. Έτσι εισάγοντας το όνομα χρήστη στην κατάλληλη φόρμα διαγράφεται ο χρήστης. Γίνεται έλεγχος για την ορθότητα των δεδομένων (πχ ότι δεν είναι κενό το πεδίο) και στην συνέχεια το σύστημα πληροφορεί με κατάλληλο μήνυμα για την επιτυχία ή αποτυχία της πράξης αυτής. Η αποτυχία μπορεί να οφείλεται είτε σε αναντιστοιχία ονόματος χρήστη με την βάση είτε λόγω βλάβης της βάσης. Το μήνυμα είναι απολύτως διευκρινιστικό.

Προσθήκη Διαχειριστή Υπηρεσιών: Ο admin μπορεί να εισάγει και άλλους διαχειριστές της Πύλης συμπληρώνοντας τα κατάλληλα πεδία τα οποία είναι ίδια όπως στην περίπτωση του register.

Προβολή στοιχείων κάποιου μέλους : Ο admin μπορεί εισάγοντας το όνομα χρήστη στο κατάλληλο πεδίο της φόρμας να δει τα στοιχεία του μέλους αυτού. Και πάλι έχουμε έλεγχο για ορθότητα των δεδομένων που δίνονται.

Διαχείριση Κωδικών συναλλαγής

Η ενότητα αυτή χωρίζεται σε δυο κατηγορίες/λειτουργίες :

- Διαγραφή μιας συναλλαγής
- Προβολή στοιχείων κάποιας συναλλαγής

Όπως έχει αναφερθεί και σε προηγούμενη ενότητα για κάθε επιτυχημένη συναλλαγή ενός πολίτη με το ΠΕΠ δημιουργείται ένας μοναδικός αριθμός που την χαρακτηρίζει και αποτελεί απόδειξη για τους φορείς που εμπλέκονται.

Διαγραφή μιας συναλλαγής : Ο admin μπορεί να διαγράψει μία συναλλαγή από την βάση δεδομένων συμπληρώνοντας τον κωδικό συναλλαγής. Σε περίπτωση επιτυχίας της πράξης ή αποτυχίας εμφανίζεται κατάλληλο μήνυμα που ενημερώνει πλήρως.

Προβολή στοιχείων κάποιας συναλλαγής: Ο admin μπορεί να δει συμπληρώνοντας την αντίστοιχη φόρμα σε ποιον πολίτη ανήκει ο κωδικός μιας συγκεκριμένης συναλλαγής (όνομα, επίθετο, e mail και την υπηρεσία που χρησιμοποίησε) .Σε περίπτωση επιτυχίας ή αποτυχίας της πράξης εμφανίζεται κατάλληλο μήνυμα που ενημερώνει πλήρως.

Ρύθμιση του e mail

Η ενότητα αυτή αποτελείται από την λειτουργία :

- Ιδιότητες των e mail

Ιδιότητες των e mail : Ο admin μπορεί να αλλάξει τις παραμέτρους που σχετίζονται με την αποστολή ενός e mail συμπληρώνοντας την κατάλληλη φόρμα. Οι παράμετροι που μπορεί να αλλάξει είναι : ο αποστολέας, ο host server , η επικεφαλίδα και το κυρίως σώμα.
Γίνεται έλεγχος για το αν τα πεδία είναι κενά, αν έχουν δοθεί ορθά δεδομένα (σωστή μορφή e – mail) και σε περίπτωση αποτυχίας ή επιτυχίας εμφανίζεται κατάλληλο μήνυμα που ενημερώνει πλήρως.

Ρύθμιση των logs

Η ενότητα αυτή αποτελείται από την λειτουργία :

- Ιδιότητες των logs

Όπως έχει ήδη αναφερθεί όλες οι πράξεις που γίνονται καταγράφονται σε αρχεία logs.

Ιδιότητες των logs : Ο admin μπορεί συμπληρώνοντας την κατάλληλη φόρμα να καθορίσει τους καταλόγους όπου θα σώζονται τα logs για κάθε υπηρεσίας ξεχωριστά.

Διαχείριση Δημοσκοπήσης

Η ενότητα αυτή αποτελείται από την λειτουργία :

- Προσθήκη δημοσκοπήσης

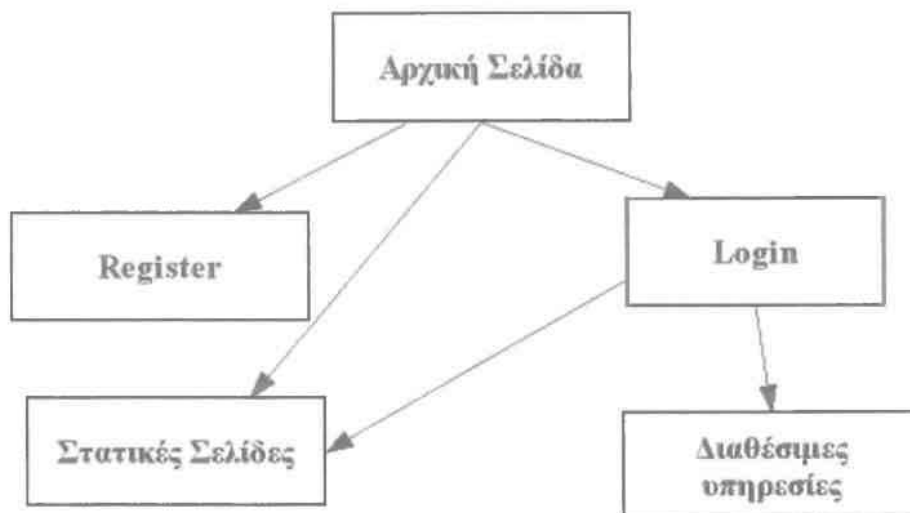
Προσθήκη δημοσκοπήσης : Ο admin συμπληρώνοντας την κατάλληλη φόρμα μπορεί να αλλάξει μια δημοσκοπήση

Όλες οι πράξεις που γίνονται από τον κάθε admin καταγράφονται σε logs.

3.6.1.2 Χαρακτηριστικές οθόνες πρόσβασης του πολίτη

Στην ενότητα αυτή εμφανίζονται οι οθόνες που θα χρησιμοποιεί ο απλός χρήστης κατά την περιήγηση του στην Πύλη Εξυπηρέτηση Πολιτών

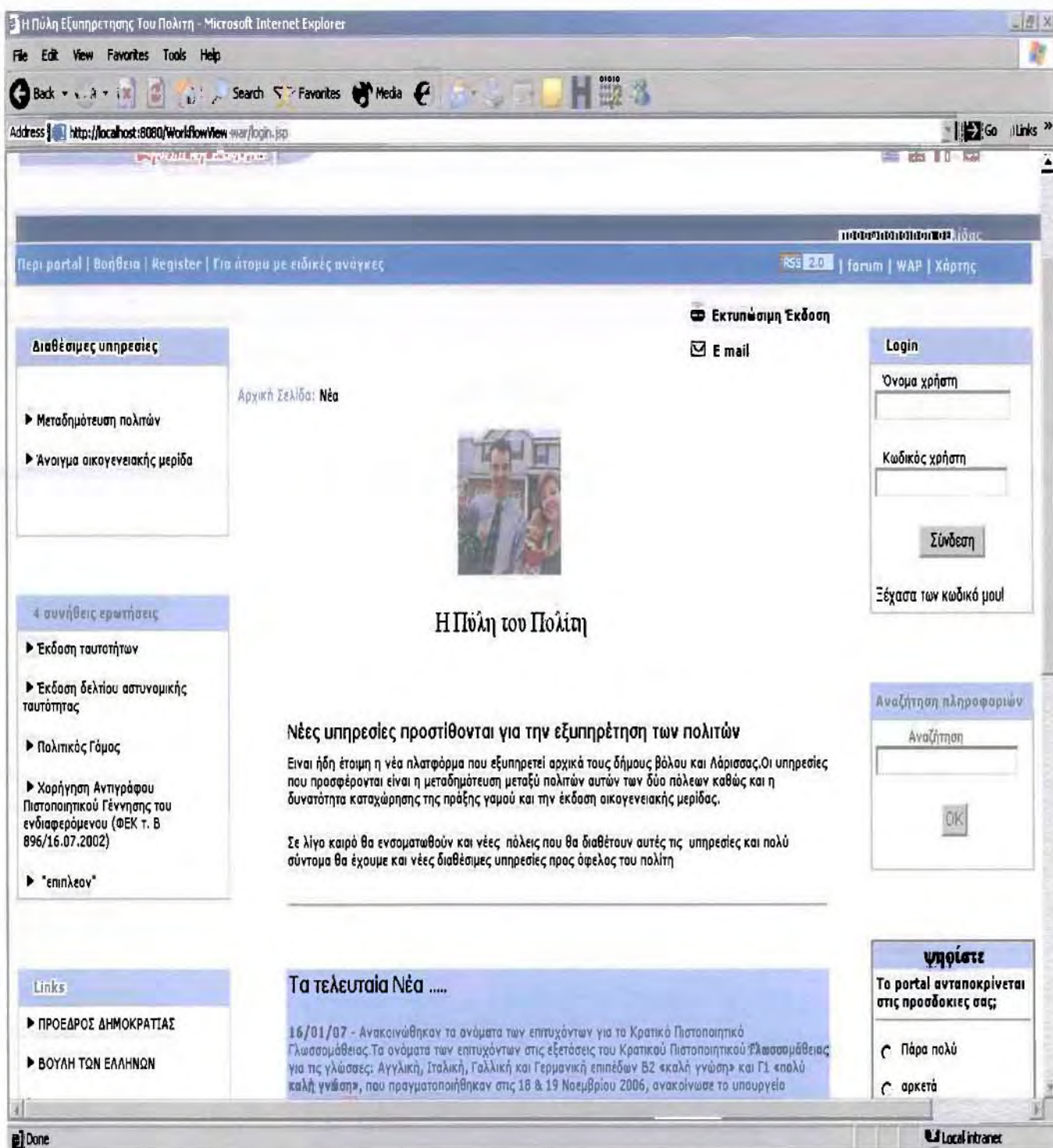
Στο σχήμα δίνεται το σύνολο των οθόνων αυτών :



Σχήμα 3.21 – Προσβάσιμες οθόνες απο Πολίτη

Αρχική σελίδα

Ο χρήστης μπορεί είτε να οδηγηθεί σε σελίδες με στατικό περιεχόμενο, είτε να κάνει Register ώστε να γίνει μέλος της Πύλης, είτε να κάνει Login



Εικόνα 3.22 Οθόνη Αρχική Σελίδα Πύλης

Login

Ο χρήστης όταν κάνει Login στην Πύλη μπορεί εκτός του να περιηγηθεί σε στατικές σελίδες να χρησιμοποιήσει και κάποια από τις Διαθέσιμες υπηρεσίες που υπάρχουν.

Διαθέσιμες υπηρεσίες

Αποτελούν τις υπηρεσίες που προσφέρει η Πύλη Εξυπηρέτηση Πολιτών και είναι οι «Άνοιγμα Οικογενειακής Μερίδας» και «Μεταδημότευση Πολιτών». Οι σελίδες αυτές προστατεύονται από το πρωτόκολλο SSL

Η Πύλη Εξυπηρέτησης Του Πολίτη - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media

Address <https://localhost:8443/WorkFlowView-war/LoginConf?choice=1>

Η Πύλη του Πολίτη

4 συνηθισμένες ερωτήσεις

- Έκδοση ταυτότητας
- Έκδοση δελτίου αστυνομικής ταυτότητας
- Πολιτικός Γάμος
- Χορήγηση Ανηγράφου Πιστοποιητικού Γέννησης του ενδιαφερόμενου (ΦΕΚ τ. Β 896/16.07.2002)

Links

- ΠΡΟΕΔΡΟΣ ΔΗΜΟΚΡΑΤΙΑΣ
- ΒΟΥΛΗ ΤΩΝ ΕΛΛΗΝΩΝ
- ΥΠΟΥΡΓΕΙΑ
- ΔΙΚΑΣΤΗΡΙΑ
- ΑΝΕΞΑΡΤΗΤΕΣ ΑΡΧΕΣ

forum

Αριθμός Ταυτότητας

Επώνυμο Δημότη

Όνομα Δημότη

Όνομα Πατρός

Όνομα Μητρός

Ημερομηνία Γέννησης(Έτος-Μήνας-Μέρα)

Τόπος Γέννησης

A.Φ.Μ

Δήμος που ανήκετε τώρα

Αριθμός μερίδας που είχατε στην δήμο

Δήμος Προορισμού μεταδημότευσης

*Αριθμός πελάτη ΔΕΗ

E mail

Σύνδεση

Είστε υπερχρήστης - admin
Κωνσάλα Διαχείρισης

Logout Χρήστη

Αναζήτηση πληροφοριών

Αναζήτηση

OK

Επακόδια Ζώνες

Οι τιμές αμολυβδης στο Νομό σας

Κάνε την αίτηση σου Online Τώρα

☒ Δηλώνω ότι δεν έχω τελέσει θρησκευτικό ή πολιτικό γάμο στην Ελλάδα ή το εξωτερικό.

Done Local intranet

Η Πύλη Εξυπηρέτησης Του Πολίτη - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <https://localhost:8443/WorkFlowView-war/LoginConf?choice=2> Go Links

Η Πύλη του Πολίτη

4 συνήθεις ερωτήσεις

- Έκδοση ταυτοτήτων
- Έκδοση δελτίου αστυνομικής ταυτότητας
- Πολυτικός Γάμος
- Χορήγηση Ανηγόραφου Πιστοποιητικού Γέννησης του ενδιαφερομένου (ΦΕΚ τ. Β 896/16.07.2002)

"Άνοιγμα Οικογενειακής Μερίδας στον δήμο καταγωγής"

Αριθμός Ταυτότητας	<input type="text"/>
Επώνυμο Δημότη	<input type="text"/>
Όνομα Δημότη	<input type="text"/>
Αριθμός ληξιαρχικής Πράξης Γάμου	<input type="text"/>
Τόπος Τέλεσης Γάμου	<input type="text"/>
Ημερομηνία Τέλεσης Γάμου(Έτος-Μήνας-Μέρα)	<input type="text"/>
A.Φ.Μ	<input type="text"/>
Δήμος κατοικίας	<input type="text" value="Βόλος"/>
Αριθμός μερίδας που είχατε στην δήμο	<input type="text"/>
Αριθμός πελάτη ΔΕΗ	<input type="text"/>
E mail	<input type="text"/>

☒ Δηλώνω γνωρίζοντας τις επιπτώσεις του νόμου ότι τα στοιχεία που δίνω δεν είναι ψευδή

Σύνδεση

Είστε υπερχρήστης - admin
Κονσόλα Διαχείρισης

Logout Χρήστη


Links

- ΠΡΟΕΔΡΟΣ ΔΗΜΟΚΡΑΤΙΑΣ
- ΒΟΥΛΗ ΤΩΝ ΕΛΛΗΝΩΝ
- ΥΠΟΥΡΓΕΙΑ
- ΔΙΚΑΣΤΗΡΙΑ
- ΑΝΕΞΑΡΤΗΤΕΣ ΑΡΧΕΣ


Αναζήτηση πληροφοριών

Αναζήτηση


OK




Επασόδια Ζωής



Οι τ.μέσ αμ.λυβ.δης
ONLINE



Κάνε την αίτησή σου Online
Τώρα >>



forum

Submit Query Reset

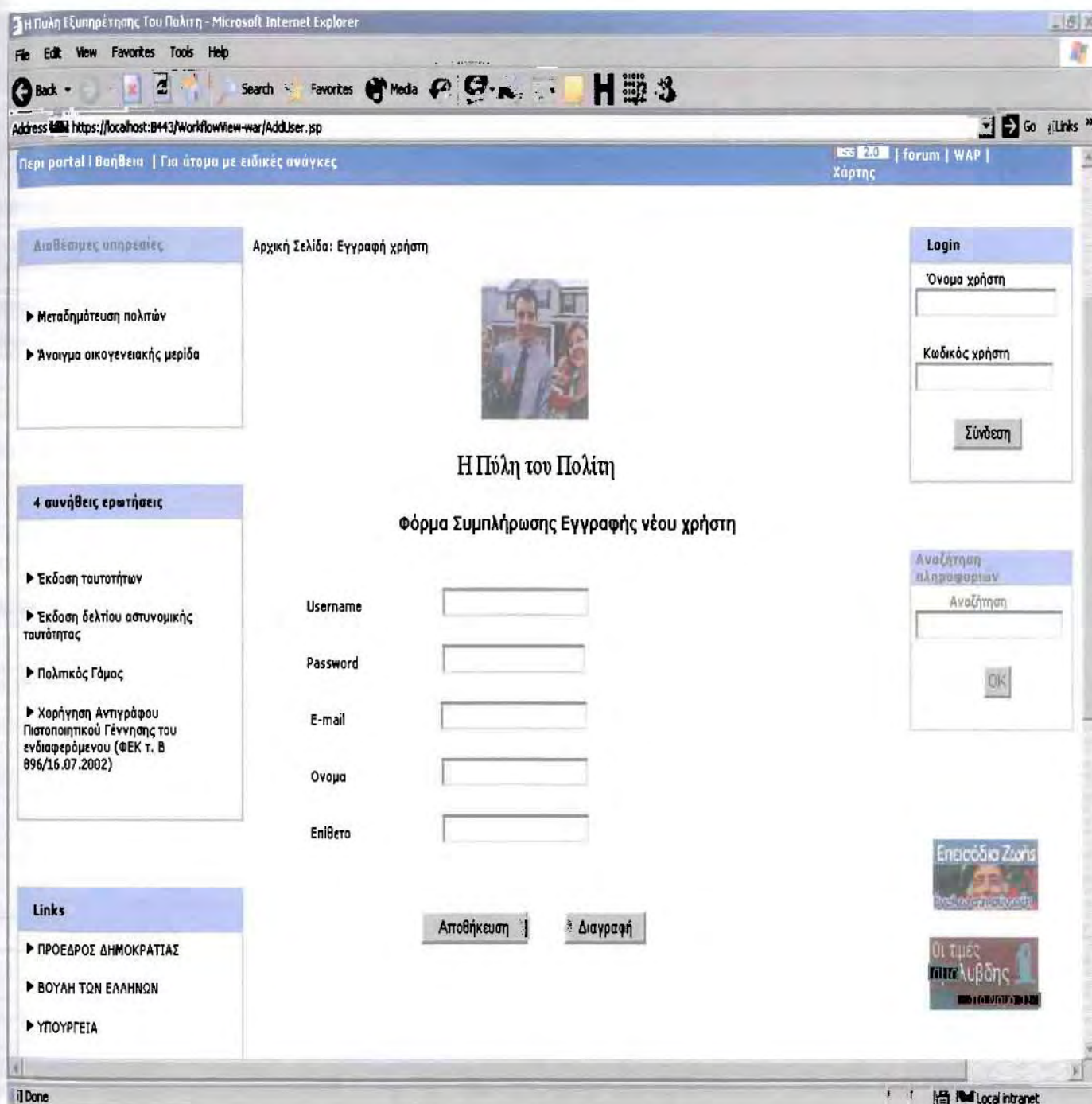
Εικόνα 3.23 – Οθόνη Διαθέσιμων Υπηρεσιών

Στατικές Σελίδες

Αποτελούν το σύνολο των στατικών σελίδων που μια Πύλη προσφέρει σαν πληροφορίες στους Πολίτες όπως αναφέρθηκε στην προηγούμενη ενότητα .Στην συγκεκριμένη περίπτωση της εφαρμογής αποτελούν απλά links χωρίς περιεχόμενο πληροφορίας

Register

Δυνατότητα που έχει ο πολίτης να εγγράφεται στην Πύλη και να μπορεί να αξιοποιήσει τις υπηρεσίες που προσφέρονται. Η σελίδα προστατεύεται με το πρωτόκολλο SSL

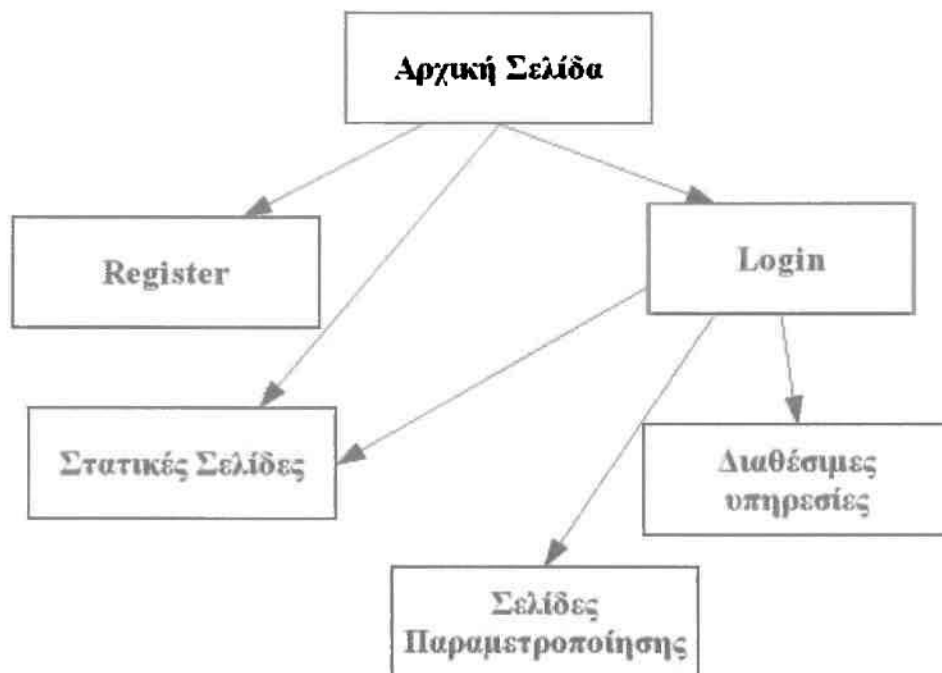


Εικόνα 3.24 – Οθόνη Register χρήστη

3.6.1.3 Χαρακτηριστικές οθόνες πρόσβασης του διαχειριστή

Εδώ εμφανίζονται οι οθόνες που θα χρησιμοποιεί ο admin για την παραμετροποίηση της Πύλης Εξυπηρέτηση Πολιτών

Στο σχήμα δίνεται το σύνολο των οθόνων αυτών :



Σχήμα 3.25 – Προσβάσιμες οθόνες από τον Διαχειριστή

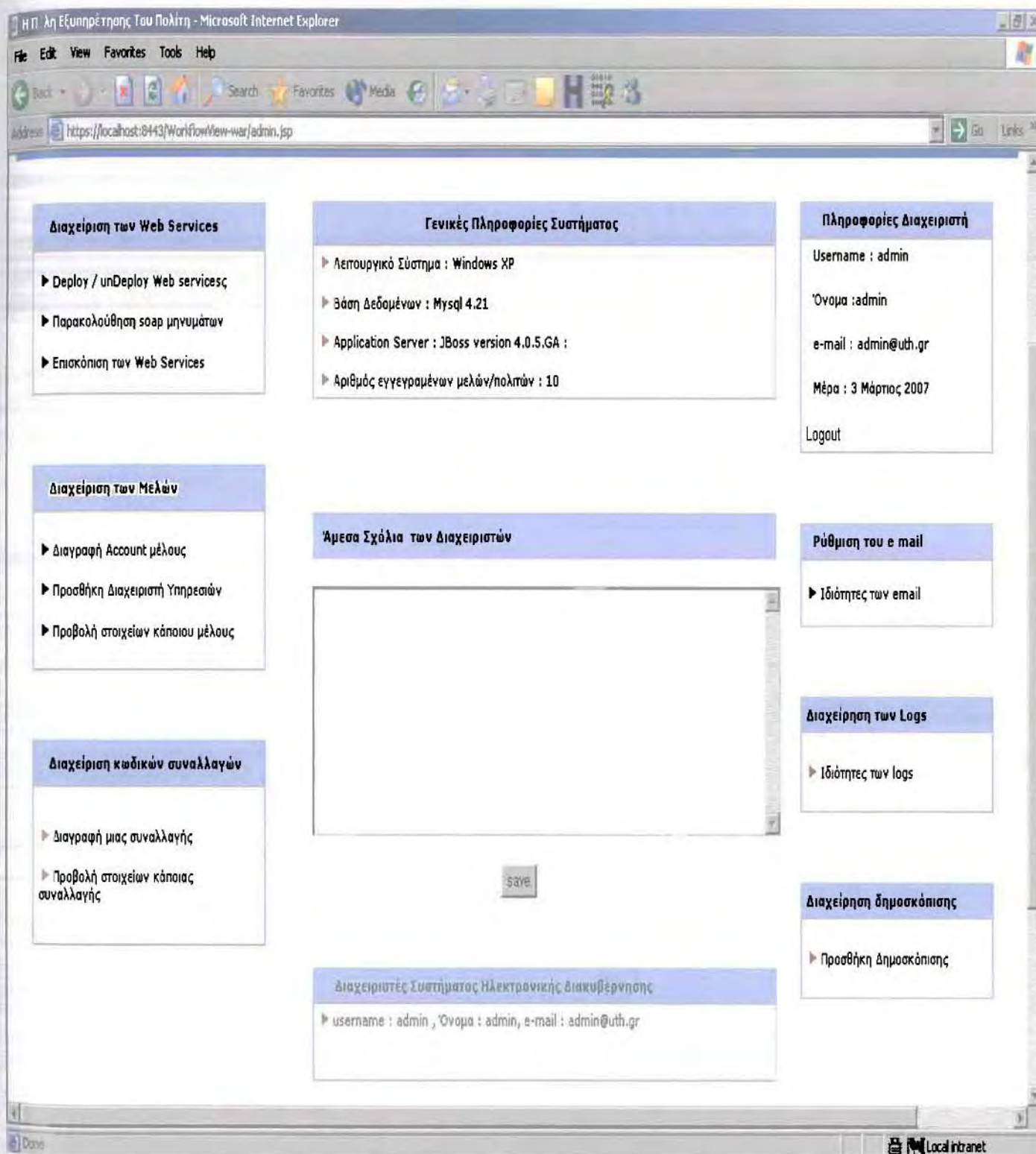
Εύκολα μπορούμε να παρατηρήσουμε ότι ο admin έχει πρόσβαση εκτός από τις σελίδες που έχει και ο Πολίτης και σε μία άλλη κατηγορία σελίδων που αφορούν την παραμετροποίηση της Πύλης. Στην συνέχεια θα δοθούν κάποια screenshots από αυτές τις σελίδες. Τέλος όλες οι σελίδες προστατεύονται από το πρωτόκολλο SSL και πως όλο το μοντέλο είναι single Thread για επιπλέον ασφάλεια.

Αρχική Σελίδα Διαχείρισης :

Στην σελίδα αυτή είναι συγκεντρωμένες όλες οι δυνατές «επιλογές» διαχείρισης που έχει ο admin. Αξίζει να αναφερθούν τα εξής :

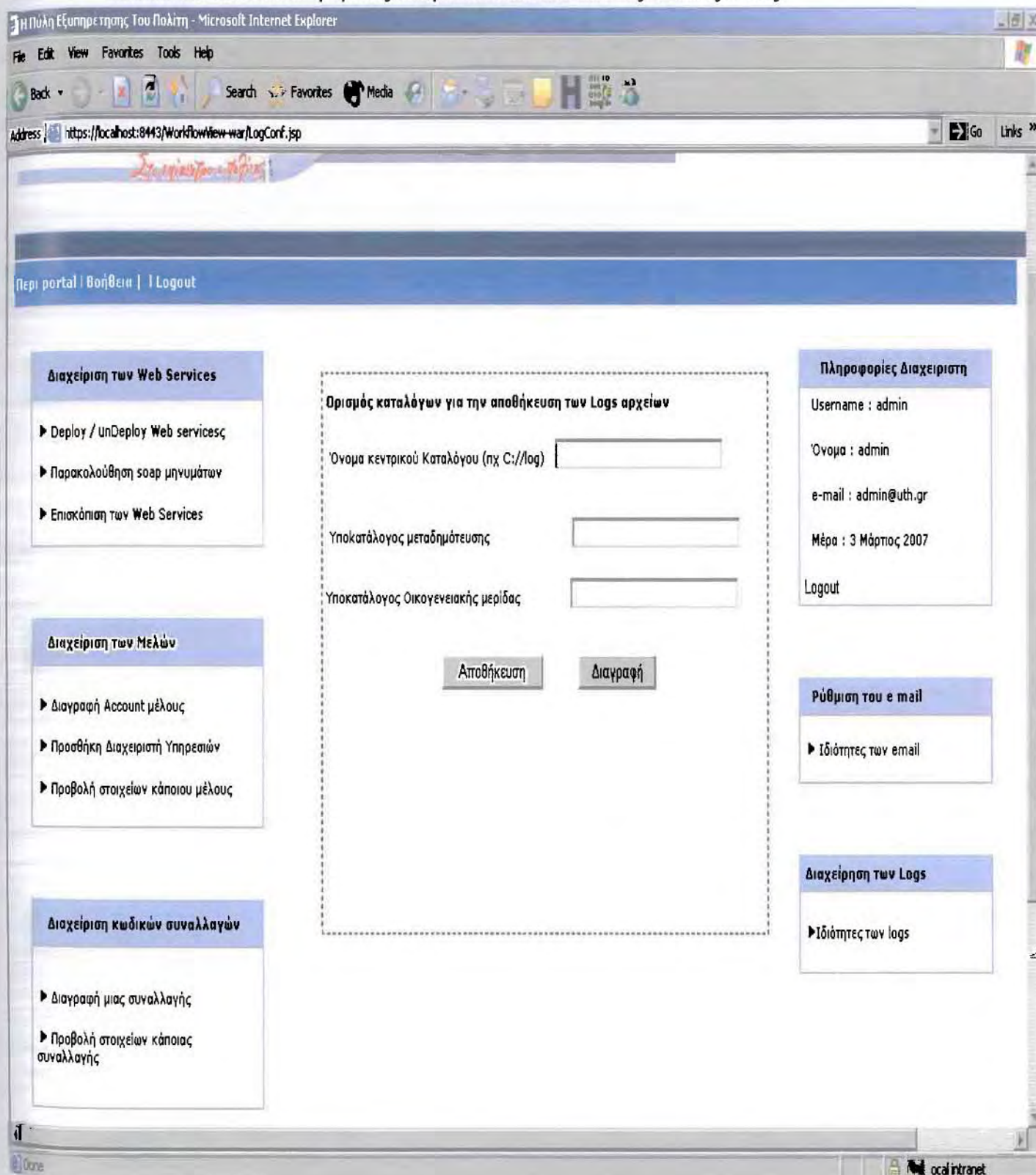
- Στο πάνω μέρος της σελίδας υπάρχει ένα κουτί που απεικονίζει πληροφορίες σχετικά με το σύστημα
- Στα δεξιά της σελίδας υπάρχει ένα κουτί που απεικονίζει πληροφορίες σχετικά με τον admin που είναι αυτή την στιγμή στο σύστημα
- Στο κέντρο της σελίδας υπάρχει ένα κουτί (text box) στο οποίο μπορούν οι admins να ανταλλάσουν σύντομα μηνύματα μεταξύ τους
- Στο κάτω μέρος της σελίδας υπάρχει ένα κουτί που αναφέρει όλους τους admin του συστήματος με κάποια στοιχεία αυτών

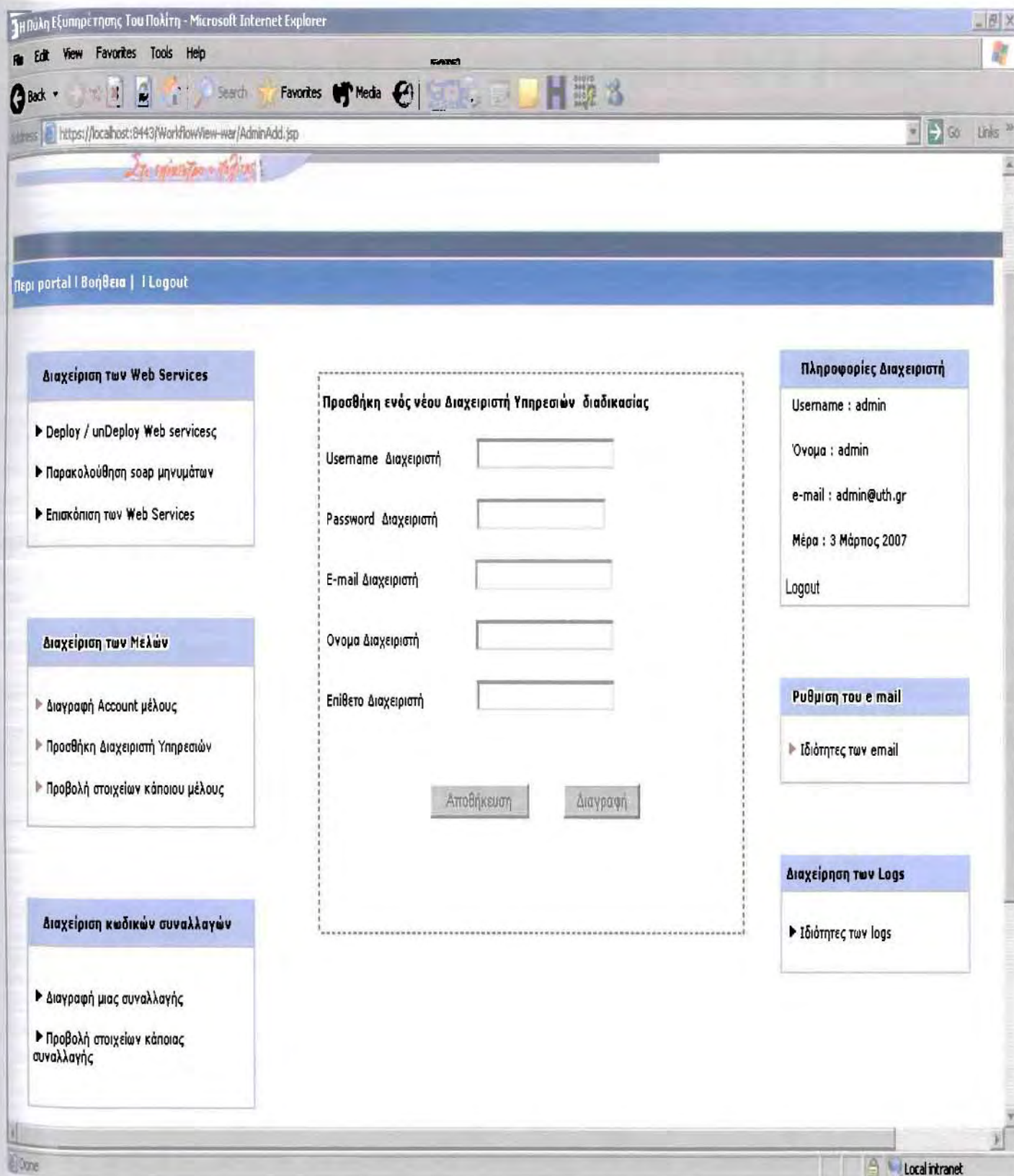
Όλα αυτά παρουσιάζονται με δυναμικό τρόπο.



Εικόνα 3.26 – Οθόνη Αρχική σελίδα διαχείρισης

Ενδεικτικά θα δοθούν μερικές ακόμα screenshots από τις σελίδες αυτές.





Εικόνα 3.27 – Οθόνη Ενδεικτικών σελίδων Διαχείρισης

4 Υλοποίηση

Στο κεφάλαιο αυτό θα παρουσιαστούν κάποιες λεπτομέρειες που αφορούν τα εργαλεία και τεχνολογίες ανάπτυξης της Πύλης και των υπηρεσιών και στην συνέχεια θα δοθεί μια περιγραφή πάνω σε σημαντικά σημεία της υλοποίησης.

4.1 Σύστημα υλοποίησης της εφαρμογής

Το σύστημα στο οποίο υλοποιήθηκε η εφαρμογή είναι το εξής:

- Λειτουργικό σύστημα Windows XP. Οι εφαρμογές μπορούν να εκτελεστούν σε οποιαδήποτε λειτουργικά υπάρχει έκδοση της java από την SUN Microsystems.
- Επεξεργαστής 1.4 GHz
- Περίπου 1 Gb στον σκληρό δίσκο
- Application server – χρήση του jboss (έκδοση 4.0.5.GA). Μπορεί να γίνει χρήση οποιουδήποτε application server που να υποστηρίζει Container Managed Persistence EJBs version 2.0.όπως οι : sun application server, Web Sphere, JRun. Οι application server εκτελούν και χρέη web server επομένως δεν υπάρχει λόγος για χρήση κάποιου ξεχωριστού.
- Χρήση Βάσης δεδομένων mysql Server 4.21. Μπορεί να χρησιμοποιηθεί οποιαδήποτε βάση δεδομένων για την οποία υπάρχει driver που να την υποστηρίζει η java (η πλειονότητα)
- Χρήση των Netbeans (έκδοση 5.5). Δεν είναι απαραίτητη η χρήση ενός τέτοιου εργαλείου (άλλο αντίστοιχο το JBuilder) αλλά σίγουρα αποτελεί μια διευκόλυνση για την παραγωγή μεγάλων προγραμμάτων

Προτείνεται επίσης η δημιουργία ενός κεντρικού φακέλου (πχ με όνομα egovernment_platform) το οποίο να βρίσκεται σε διαφορετικό partition από το λειτουργικό σύστημα όπου εκεί θα βρίσκονται όλες οι απαραίτητες κλάσεις και αντίστοιχες βιβλιοθήκες. Έτσι μπορεί να γίνει χρήση προγραμμάτων και βιβλιοθηκών από διάφορα λειτουργικά συστήματα εξασφαλίζοντας την αυτονομία της εφαρμογής .

4.2 Πλατφόρμες και προγραμματιστικά εργαλεία

Εξυπηρετητής εφαρμογών (Application Server)

Ο εξυπηρετητής εφαρμογών είναι επιφορτισμένος με το να φιλοξενεί τις εφαρμογές που έχουμε αναπτύξει. Συγκεκριμένα (για την συγκεκριμένη εφαρμογή) να υποστηρίζει Enterprise Java Beans (EJBs) και να μπορεί να εκτελεί και χρέη Web server, ώστε να μπορεί να ανταποκρίνεται σε αιτήσεις του πρωτοκόλλου HTTP υποστηρίζοντας servlets και jsp σελίδες.

Java 2

Στην εργασία χρησιμοποιήθηκε μόνο η γλώσσα προγραμματισμού Java καθώς και τεχνολογίες που παρέχονται με βάση αυτή όπως EJBs, JSP, servlets κτλ. Η έκδοση που χρησιμοποιήθηκε ήταν η JAVA 2 PLATFORM STANDARD EDITION DEVELOPMENT KIT 5.0 (περιλαμβάνει πληθώρα εργαλείων)

Περισσότερες πληροφορίες σχετικά με την μεταγλώττιση του πηγαίου κώδικα (.java) σε εκτελέσιμα (.class) αρχεία, την χρήση της virtual machine κτλ δεν θα αναφερθούν στην εργασία διότι ξεφεύγουν από τα πλαίσια αυτής.

WAR (Web Application Archive)

Το αρχείο .war είναι ένα αρχείο jar με συγκεκριμένη δομή το οποίο αντιλαμβάνεται ο δικτυακός εξυπηρετητής για την δημιουργία μιας δικτυακής εφαρμογής που βασίζεται στο HTTP πρωτόκολλο. Περιέχει στατικές και δυναμικές σελίδες (HTML, JSP) καθώς και κλάσεις, αρχεία jar και ένα αρχείο xml με όνομα web.xml

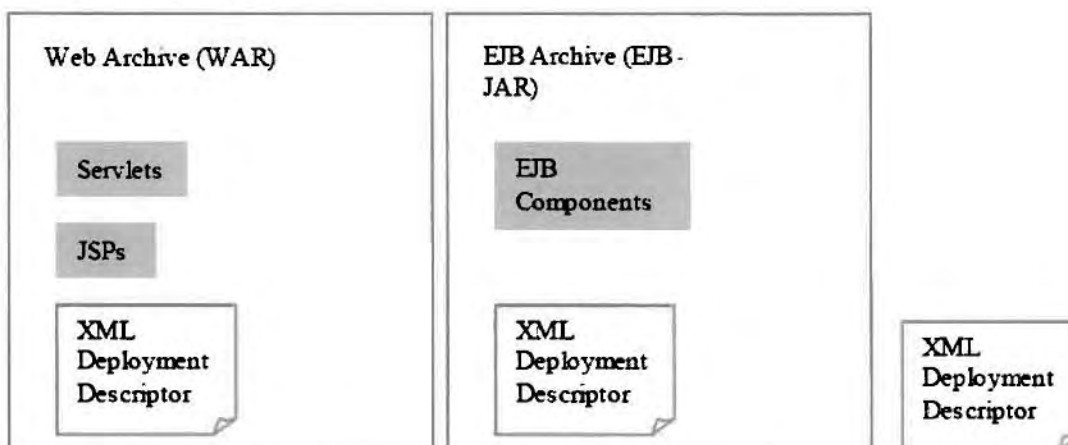
EJB-JAR

Το EJB αρχείο είναι ένα jar αρχείο το οποίο περιλαμβάνει κλάσεις και περιγραφές που είναι απαραίτητες για να το αναπτύξει ο application server.

EAR (Enterprise Application Archive)

Ένα αρχείο .ear αποτελεί μια δικτυακή εφαρμογή που χρησιμοποιεί και EJBs. Είναι στην ουσία ένα αρχείο jar το οποίο περιέχει στη ρίζα του το WAR και EJB αρχείο καθώς και κάποια αρχεία περιγραφής όπως το application.xml (στον φάκελο /META-INF)

Enterprise Archive (EAR)



Σχήμα 4.1 Στοιχεία που αποτελούν το EAR αρχείο

JDBC (Java Database Connectivity)

Η διασύνδεση μεταξύ εφαρμογών που αναπτύσσονται με την γλώσσα java και διαφόρων βάσεων δεδομένων επιτυγχάνεται μέσω ειδικών οδηγών που αναπτύσσονται συνήθως από τις κατασκευάστριες εταιρίες. Χαρακτηριστικό παράδειγμα που χρησιμοποιείται και στην εργασία είναι ο οδηγός `mysql-connector-java-5.0.4.jar` για σύνδεση με τον `mysql` server. Έτσι με βάση το API που δίνεται επιτρέπεται η δημιουργία πινάκων, SQL ερωτήσεων κτλ

JNDI (Java Naming and Directory Interface)

Το JNDI αποτελεί ένα πλαίσιο το οποίο καθορίζει τον τρόπο με τον οποίο γίνεται η αναζήτηση EJBs καθώς και των βάσεων δεδομένων σε μια enterprise εφαρμογή. Όταν δημιουργούμε EJBs καθορίζουμε στα αντίστοιχα αρχεία περιγραφής τα ονόματα τα οποία θα καταχωρηθούν στον JNDI server του εκάστοτε application server (πχ του JBoss) ώστε να έχουμε επιτυχή αναζήτηση.

JBoss

Στην εργασία χρησιμοποιήθηκε ο application server JBoss, ο οποίος παρέχεται δωρεάν στο διαδίκτυο. Ο JBoss είναι γραμμένος σε java και είναι αρκετά εύχρηστος και σταθερός. Ενσωματώνει επιπλέον έναν web server προσφέροντας έτσι μια ολοκληρωμένη λύση για την ανάπτυξη δικτυακών εφαρμογών.

Ενδεικτικά αναφέρεται η δομή του :

`/jboss/bin`

Σε αυτό το φάκελο περιλαμβάνονται όλα τα απαραίτητα αρχεία για την έναρξη και τερματισμό του JBoss

`/jboss/lib`

Περιλαμβάνει όλες τις βιβλιοθήκες με τις κλάσεις που χρησιμοποιεί ο JBoss.

`/jboss/server/default`

Στον φάκελο αυτό υπάρχουν οι υποφάκελοι `deploy`, `log` και `tmp`. Ο `deploy` περιλαμβάνει τις εφαρμογές (`.war`, `.jar`, `.ear`) καθώς και αρχεία που περιγράφουν τις συνδέσεις με τις βάσεις δεδομένων. Ο φάκελος `log` περιλαμβάνει αρχεία στα οποία καταγράφεται η κατάσταση του server κατά την εκκίνηση και την εύρεση λαθών.

Τέλος ο `temp` περιλαμβάνει προσωρινά αρχεία που δημιουργεί ο JBoss κατά την λειτουργία του.

Ανάπτυξη εφαρμογών

Η ανάπτυξη των εφαρμογών στον JBoss περιλαμβάνει απλά την αντιγραφή των αρχείων `.ear`, `.war`, `.jar` μέσα στο φάκελο `server/default/deploy`. Αντίστοιχα όπως αναφέρθηκε για την σύνδεση με μια βάση δεδομένων απαιτείται απλά η προσθήκη του κατάλληλου xml αρχείου σε αυτό τον φάκελο. Ο JBoss θα αντιληφθεί αμέσως τις αλλαγές που έγιναν και θα τις αναπτύξει άμεσα.

NETBEANS IDE 5.5

Το NETBEANS αποτελεί μια ολοκληρωμένη πλατφόρμα για την ανάπτυξη πολύπλοκων εφαρμογών. Περιλαμβάνει πληθώρα χρήσιμων προγραμμάτων και λειτουργιών διευκολύνοντας πολύ τους προγραμματιστές τόσο κατά την ανάπτυξη όσο και κατά την αποσφαλμάτωση των προγραμμάτων. Μερικές χρήσιμες δυνατότητες του είναι οι εξής :

- Έτοιμα πρότυπα για την δημιουργία διαδικτυακών εφαρμογών (.war .ear αρχεία)
- Αυτόματη δημιουργία EJBs από ένα σχήμα βάσης
- Ενσωμάτωση του εργαλείου ant
- Άμεσο deploy των εφαρμογών στον application – webserver που επιθυμούμε.
- Χρήση του JUnit για αποσφαλμάτωση
- Εύκολη δημιουργία γραφικών (interface)

Axis framework για δικτυακές υπηρεσίες (Web Services)

Ίσως το σημαντικότερο εργαλείο της εργασίας. Είναι αυτό που αποτελεί τον εξυπηρετητή των web services ή πιο απλά αποτελεί ένα πλαίσιο (framework) για την δημιουργία και εξυπηρέτηση των SOAP μηνυμάτων. Περιλαμβάνει επίσης και εργαλεία για την δημιουργία των απαραίτητων κλάσεων που θα δημιουργήσουν τα web services.

Αποτελείται από :

- Εργαλείο που δημιουργεί java κλάσεις από αρχεία WSDL και το αντίστροφο
- Εργαλείο για την παρακολούθηση SOAP μηνυμάτων
- Ένα servlet που δρα ως εξυπηρετητής (server) για τους application servers

Στην εργασία χρησιμοποιήθηκε το servlet που δρα ως εξυπηρετητής. Για να το επιτύχουμε αυτό απλά ανατιγράψουμε το περιεχόμενο του φακέλου C:\axis_name\webapps στον application server (στον φάκελο server/default/deploy) και να τον μετονομάσουμε σε axis.war. Ο application server θα το αναπτύξει άμεσα και θα έχουμε πρόσβαση στο servlet.

Για την δημιουργία των αρχείων WSDL μπορούμε να χρησιμοποιήσουμε είτε το Java2WSDL είτε να δημιουργήσουμε εμείς ένα WSDD αρχείο και με την εντολή **java -cp %AXISCLASSPATH% org.apache.axis.client.AdminClient file.wsdd** και να πάρουμε το αρχείο που επιθυμούμε. Προσοχή να δοθεί στο classpath (AXISCLASSPATH : περιλαμβάνει σχεδόν όλο τον φάκελο lib από τον κατάλογο axis\lib. Καλό είναι είτε να δημιουργούμε GLOBAL environment variables στα windows είτε ένα αρχείο batch που θα φορτώνουμε κάθε φορά. Αυτό επέλεξε και ο συγγραφέας σαν πιο βολικό.

Για την δημιουργία τώρα των stubs (αντικείμενα που δρουν ως proxy και επιτρέπουν στην εφαρμογή να καλεί απομακρυσμένες μεθόδους (remote calls) σαν να ήταν μέθοδοι τοπικών αντικειμένων) χρησιμοποιούμε το εργαλείο WSDL2Java με την εντολή **java -cp %AXISCLASSPATH% org.apache.axis.wsdl.WSDL2Java address_WSDL**. Θα δημιουργηθούν αυτόματα οι απαιτούμενες κλάσεις που θα χρησιμοποιηθούν στην εφαρμογή που θα της καλεί.

Σχετικά με τα EJBs

Τα (Enterprise Java Beans) EJBs είναι αντικείμενα που προσφέρονται για την διασύνδεση εφαρμογών με διάφορους πόρους όπως οι βάσεις δεδομένων. Προσφέρουν επίσης την δυνατότητα δημιουργίας μεθόδων επιχειρηματικής λογικής διευκολύνοντας περισσότερο την δημιουργία του MVC μοντέλου που αναφέρθηκε προηγουμένως.

Χωρίζονται σε 3 κατηγορίες.

- Session Beans
- Entity Beans
- Message Driven Beans

Τα session beans αντιπροσωπεύουν τις ενέργειες που εκτελεί ένας πελάτης σε μια εφαρμογή. Χωρίζονται στα stateful και stateless (ανάλογα άμα ενδιαφέρει να διατηρείται κάποια κατάσταση τους). Τα session bean έχουν διάρκεια ζωής που καθορίζεται από κάθε session του πελάτη και υπεύθυνο για αυτά, όπως και για όλα τα beans, είναι ο EJB Container.

Επίσης ένα πλεονέκτημα τους είναι ότι εξασφαλίζεται ο συγχρονισμός μεταξύ αυτών χωρίς τον φόβο deadlocks γράφοντας κώδικα σαν single threaded αντικείμενα. Ο EJB container είναι υπεύθυνος για όλα αυτά.

Θα μπορούσαμε να αποκαλέσουμε τα Sessions beans αντικείμενα «περισσότερο» επιχειρηματικής λογικής και είναι αυτά που συνήθως καλούν τα entity beans.

Τα entity beans χρησιμοποιούνται για την δημιουργία μιας αντικειμενοστραφούς προβολής μια βάσης δεδομένων. Δημιουργούνται αντικείμενα και σχέσεις αυτών για κάθε πίνακα μιας βάσης δεδομένων. Διαφέρουν από τα session bean διότι αποτελούν το μοντέλο (κατά το πρότυπο του MCV) και έχουν διάρκεια ζωής πολύ μεγαλύτερη(όσο υπάρχουν τα δεδομένα στη βάση). Μάλιστα σε περίπτωση αστοχίας του συστήματος μπορούν εύκολα να αναδομηθούν από μια υπάρχουσα βάση. Και αυτά χωρίζονται σε δυο κατηγορίες, σε BMP (Bean Managed Persistence) CMP (Container Managed Persistence). Η διαφορά τους είναι ότι στα BMP ο προγραμματιστής είναι υπεύθυνος για την ανάκτηση και αποθήκευση των δεδομένων σε μια βάση ενώ στα CMP την ευθύνη την έχει ο EJB Container.

Τέλος έχουμε τα **Message driven beans** που χρησιμοποιούνται για την ασύγχρονη επικοινωνία μεταξύ εφαρμογών (όχι αναγκαστικά J2EE τεχνολογίας) με την χρήση μηνυμάτων – EJBs.

Στην εργασία θα χρησιμοποιήσουμε αποκλειστικά τα CMP beans και κάθε αναφορά σε EJBs θα υπονοεί αυτά. Τα CMP υπάρχουν για να συνδέουν τις εφαρμογές με τις βάσεις δεδομένων και δεν περιλαμβάνουν κώδικα που να υλοποιεί τα transactions παρά μόνο σχέσεις μεταξύ των EJBs. Τα CMP για τις βάσεις που χρειάστηκαν στην εργασία δημιουργήθηκαν αυτόματα από το εργαλείο που υποστηρίζει η πλατφόρμα NETBEANS (που αναλύθηκε προηγουμένως).

Τέλος δημιουργείται και το αρχείο περιγραφής ejb-jar.xml. Είναι ένα xml αρχείο στο οποίο ορίζονται οι κλάσεις που δημιουργήθηκαν, οι σχέσεις μεταξύ τους, η ασφάλεια αυτών και άλλα στοιχεία που δεν θα αναλυθούν.

JSP - Servlets

Για να επιτύχουμε διαδραστικότητα μεταξύ του χρήστη με την δικτυακή εφαρμογή συνήθως απαιτείται η δημιουργία δυναμικών σελίδων. Οι σελίδες αυτές σε τεχνολογία της Java λέγονται JSP (Java server Pages).

Οι JSP σελίδες περιέχουν html κώδικα και σε κάποια σημεία ,στα οποία απαιτείται δυναμικά δεδομένα τοποθετείται κώδικας Java. Οι JSP σελίδες στην συνέχεια μετατρέπονται αυτόματα από το Web (ή application) server σε servlets .

Τα servlets είναι ειδικές κλάσεις (javax.servlet.http.HttpServlet) οι οποίες χρησιμοποιούνται από τον προγραμματιστή για να απαντούν σε αιτήσεις του πρωτοκόλλου HTTP προσφέροντας και άλλες δυνατότητες όπως πχ την δημιουργία Cookies.

Το αρνητικό με τα servlets είναι ο τρόπος με τον οποίο δίνεται η έξοδος προς τον χρήστη η οποία έξοδος, δηλαδή ο HTML κώδικας, γράφεται χειρωνακτικά από τον προγραμματιστή κάνοντας πολύ δύσκολο το debugging και γενικότερα την δημιουργία ευπαρουσίαστων σελίδων. Αυτό το πρόβλημα λύθηκε όπως προαναφέρθηκε με τις JSP σελίδες.

Με τα Servlets και τα JSPs καλύπτονται οι βασικές ανάγκες μιας δικτυακής εφαρμογής

4.3 Λεπτομέρειες υλοποίησης

Περιγραφή της Εφαρμογής της Πύλης και των υπηρεσιών :

Ο κεντρικός κατάλογος WorkflowView αποτελείται από τους εξής υποκαταλόγους

- *Commo* : Σε αυτό τον φάκελο περιλαμβάνονται όλα τα απαραίτητα αρχεία που είναι κοινά για τις δύο υπηρεσίες που έχουν αναπτυχθεί αλλά γενικότερα και για άλλες υπηρεσίες που θα εντάσσονταν στο σύστημα. Παραδείγματα αποτελούν τα αρχεία για δημιουργία PDF, αποστολή e mail, διαχείρισης των logs, form validations κτλ
- *servlet_OikogeniakiMerida*: Σε αυτόν τον φάκελο περιλαμβάνονται όλα τα αρχεία τα οποία είναι απαραίτητα για την δημιουργία της υπηρεσίας «Άνοιγμα οικογενειακής Μερίδας». Το κεντρικό servlet είναι αυτό που καθορίζει την ροή εργασίας και καλεί τις απαραίτητες κλάσεις είτε από τον κατάλογο Common είτε από τον κατάλογο που ανήκει.
- *servlet_metadimoteusi*: Σε αυτόν τον φάκελο περιλαμβάνονται όλα τα αρχεία τα οποία είναι απαραίτητα για την δημιουργία της υπηρεσίας «Μεταδημότευση πολιτών». Το κεντρικό servlet είναι αυτό που καθορίζει την ροή εργασίας και καλεί τις απαραίτητες κλάσεις είτε από τον κατάλογο Common είτε από τον κατάλογο που ανήκει
- *siteManagment*: Σε αυτόν τον φάκελο περιλαμβάνονται αρχεία που διαχειρίζονται την οργάνωση της Πύλης και αφορούν την διαδικασία login, logout, register, δημοσκόπηση κτλ

- *adminSources*: Σε αυτόν τον φάκελο περιλαμβάνονται όλα τα αρχεία που καθορίζουν την λειτουργικότητα του admin/Διαχειριστή της Πύλης και των υπηρεσιών που παραμετροποιεί.
- Οι φάκελοι της μορφής *Services_stubs.ipiresia* όπου στο κάθε φάκελο αντιστοιχούν κάποια αρχεία που είναι απαραίτητα για την «ενεργοποίηση-κάλεσμα» των διαδικτυακών υπηρεσιών με την τεχνολογία των soap μηνυμάτων.
- *Web pages*: Σε αυτόν τον φάκελο περιλαμβάνονται όλες οι δυναμικές σελίδες που συνθέτουν την λειτουργικότητα της Πύλης
- *Configuration files* : Αυτός ο φάκελος περιλαμβάνει το αρχείο web.xml το οποίο δημιουργείται αυτόματα από το IDE NETBEANS και είναι απαραίτητο για τον εξυπηρετητή.

Όλα αυτά τα στοιχεία «ενώνονται» όπως αναφέρθηκε και δημιουργούν το αρχείο WorkflowView.ear το οποίο και γίνεται deploy στον εξυπηρετητή (application server) και πλέον η web εφαρμογή είναι έτοιμη για προσπέλαση.

Περιγραφή της Εφαρμογής των Δήμων

- *Ipiresies*: Στον φάκελο αυτό υπάρχουν όλα τα απαραίτητα αρχεία τα οποία αποτελούν το business layer και περιέχουν τις μεθόδους αυτές που καλούνται από τις αντίστοιχες ροές δεδομένων της εφαρμογής της Πύλης.
- *Dilosi_Gamou*: Στον φάκελο αυτό περιλαμβάνονται όλα τα απαραίτητα αρχεία τα οποία αποτελούν το data layer σχετικά με την υπηρεσία Δήλωση Γάμου και τα οποία είναι υπεύθυνα για την πρόσβαση στις βάσεις δεδομένων (θυμηθείτε την τεχνολογία των EJBs)
- *Metadimoteusi_stoixeia*: Στον φάκελο αυτό περιλαμβάνονται όλα τα απαραίτητα αρχεία τα οποία αποτελούν το data layer σχετικά με την υπηρεσία Μεταδημότευση και τα οποία είναι υπεύθυνα για την πρόσβαση στις βάσεις δεδομένων (θυμηθείτε την τεχνολογία των EJBs)

Περιγραφή της Εφαρμογής Αστυνομία

- *Astinomiko_tmima*: Περιλαμβάνει τα αρχεία που εξασφαλίζουν τις υπηρεσίες της Αστυνομίας καθώς και την πρόσβαση στην βάση δεδομένων

Περιγραφή της Εφαρμογής ΔΕΗ

- *Deh*: Περιλαμβάνει τα αρχεία που εξασφαλίζουν τις υπηρεσίες της ΔΕΗ καθώς και την πρόσβαση στην βάση δεδομένων

4.4 Τεχνικές λεπτομέρειες εφαρμογής

Στην ενότητα αυτή θα αναφερθούν κάποια στοιχεία υλοποίησης τα οποία αξίζουν προσοχής.

Κωδικός συναλλαγής :

Ο κωδικός συναλλαγής που έχει αναφερθεί σε προηγούμενα κεφάλαια εξασφαλίζεται με τον εξής τρόπο. Γίνεται πρόσβαση στην αντίστοιχη βάση δεδομένων, παίρνεται το μέγιστο στοιχείο αυτής και αυξάνεται κατά ένα. Η υλοποίηση βασίζεται στα EJBs
Ο ίδιος τρόπος ισχύει για κάθε μοναδικό αριθμό όπου και αν απαιτείται.

PDF

Για την δημιουργία του PDF αρχείου χρησιμοποιείται η βιβλιοθήκη iText η οποία προσφέρεται ως freeware. Στην συνέχεια καλούνται διάφορες μέθοδοι οι οποίες και δημιουργούν το επιθυμητό αποτέλεσμα.

Logs

Τα αρχεία κειμένου αυτά, έχουν τις εξής ιδιότητες :

Το όνομα του αρχείου είναι της μορφής: 2_3_2007.txt δηλαδή για κάθε μέρα δημιουργείται ξεχωριστό αρχείο και κάθε υπηρεσία έχει το δικό της υπο-φάκελο με τα αρχεία αυτά.

Παράδειγμα : C:\log\Ipiresia_metadimoteusi, C:\log\Ipiresia_Gamou. Αυτά μπορούν να αλλάξουν όπως έχει αναφερθεί από τον admin

Η εσωτερική δομή κάθε αρχείου log είναι η εξής :

Ωρα [Κωδικός συναλλαγής Πελάτη X] [IP: X.X.X.X] μήνυμα κωδικού

Form Validation

Η ορθότητα μιας φόρμας προς συμπλήρωση επιτυγχάνεται με την χρήση της κλάσης FormValid η οποία περιλαμβάνει μεθόδους ελέγχου για περίπτωση NULL συμβολοσειρές, αριθμούς, μορφή ημερομηνίας, e mail κτλ.

Αξίζει να αναφερθεί η χρήση των Patterns που προσφέρει η JAVA για τον έλεγχο τόσο της έγκυρης μορφής ενός e mail καθώς και για την μορφή της ημερομηνίας.

Web pages

Οι περισσότερες (δυναμικές) σελίδες JSP που αποτελούν την Πύλη κάνουν χρήση της τεχνολογίας των Beans. Δηλαδή αποτελούν και μόνο το μέσο προβολής (μέσω των beans) των αποτελεσμάτων που δημιουργούνται και επεξεργάζονται από τα servlets. Υπάρχουν βέβαια και κάποιες δυναμικές σελίδες που κάνουν απευθείας πρόσβαση σε μια βάση δεδομένων και εμφανίζουν τα αποτελέσματα χωρίς ενδιάμεσους (Beans).

Ροές Εργασίας υπηρεσιών

Το servlet κάθε υπηρεσίας ξεκινάει με τον ίδιο τρόπο :

Αρχικά γίνεται αρχικοποίηση του servlet στο οποίο καθορίζονται κάποιοι παράμετροι όπως τα στοιχεία του e mail που θα σταλεί σε περίπτωση επιτυχίας καθώς και για το που θα σωθούν τα logs αρχεία. Οι παράμετροι αυτοί καθορίζονται από αντίστοιχα xml αρχεία τα οποία διαβάζονται (γίνεται parsing). Αυτά είναι :

XmlEmailValues.xml με το αντίστοιχο DTD για validation *EmailDTD.dtd*

XmlLogValues.xml με το αντίστοιχο DTD για validation *logDTD.dtd* .

Θα μπορούσε να ήταν και xml scheme αντί DTD αλλά όπως ορίζεται και από το e gif για μικρά αρχεία και λίγους τύπους περιγραφής δεδομένων τα DTD είναι πιο εύχρηστα. Στην συνέχεια στην αρχή κάθε αίτησης που εξυπηρετείται, γίνεται έλεγχος αν το αρχείο που καθορίζει τα logs έχει τροποποιηθεί, κοιτώντας την ημερομηνία τροποποίησης και εαν έχει όντως αλλάξει (από τον admin) τότε ξαναγίνεται parsing. Αντίστοιχα μόλις είναι έτοιμη η αποστολή του e mail και πάλι γίνεται έλεγχος για πιθανή τροποποίηση του αρχείου παραμετροποίησης του e mail.

Προκειμένου να επιτευχθεί συνέπεια των δεδομένων ακολουθείται η εξής τακτική :

Όταν ο admin πραγματοποιεί την διαδικασία της παραμετροποίησης για παράδειγμα του e mail τότε το αντίστοιχο xml αρχείο κλειδώνει (για ανάγνωση) έως ότου τελειώσει η εγγραφή σε αυτό. Σε περίπτωση αποτυχίας της διαδικασίας εγγραφής ξεκλειδώνει αυτόματα. Επιπλέον σε περίπτωση ανάγνωσης (επιτρέπεται η ταυτόχρονη ανάγνωση από πολλές αιτήσεις) κλειδώνεται η εγγραφή.

Εδώ να σημειώσουμε ότι υπάρχει η περίπτωση της λιμοκτονίας όταν πχ έρχονται συνεχώς αναγνώστες με τέτοιο ρυθμό με το οποίο ποτέ ένας εγγραφέας να μην μπορεί να κάνει εγγραφή. Η υλοποίηση γίνεται με βάση την τεχνική των lck αρχείων μέσω κάποιων κλάσεων και μεθόδων που παρέχει το development kit της JAVA και πιο συγκεκριμένα μέσω της FileChannel και Filelock.

Τέλος αναφέρεται ότι αυτός ο τρόπος υλοποίησης είναι μια σύμβαση και δεν αποτρέπει κάποια κακόβουλη εφαρμογή να εγγράψει στα αρχεία αυτά όταν είναι «κλειδωμένα»

Τέλος σημασία έχει ο συγχρονισμός μεταξύ διαφόρων αντικειμένων προκειμένου να επιτυγχάνεται ορθότητα και συνέπεια των δεδομένων κυρίως κατά την πρόσβαση σε βάσεις που απαιτούν εγγραφή σε αυτές κτλ. Αυτό επιτυγχάνεται ορίζοντας τις αναφορές των αντικειμένων ως Global μεταβλητές και στην συνέχεια συγχρονίζοντας αυτές μέσω του μηχανισμού Synchronized.

Αναφέρεται ό,τι όλες οι μεταβλητές που αναπτύσσονται εντός μιας τοπικής μεθόδου για παράδειγμα η doPost είναι μοναδικά αντίγραφα για κάθε αίτηση που εξυπηρετούν και είναι απαλλαγμένες από θέματα συγχρονισμού.

4.5 Δομή Απαραίτητων καταλόγων

Για την ορθή λειτουργία όλης της εφαρμογής είναι απαραίτητος ο κατάλογος ConfigurationFiles ο οποίος είναι στο ίδιο σχετικό επίπεδο με τον κατάλογο egovernment_platform που περιλαμβάνει τους καταλόγους με την εφαρμογή

Αναλυτικά ο κατάλογος ConfigurationFiles περιλαμβάνει :

- *Admin_Logs*

Εδώ υπάρχουν υποκατάλογοι ανάλογα με την κάθε υπηρεσία που μπορεί να παραμετροποιήσει ο admin και μέσα σε αυτήν σώζονται αρχεία (logs) στα οποία καταγράφονται : το όνομα του admin που κάνει την πράξη και κάθε λάθος ή επιτυχία πραγματοποιηθεί.

- *Libs*

Εδώ περιλαμβάνονται όλες οι εξωγενείς βιβλιοθήκες που είναι απαραίτητες για την λειτουργία της εφαρμογής

- *XMLs*

Εδώ υπάρχουν τα xml αρχεία μαζί με τα απαραίτητα DTD αρχεία

- *Deploy_Undeploy*

Εδώ υπάρχουν τα αρχεία WSDD με τα οποία μπορεί να κάνει deploy και undeploy δυναμικά ο admin διάφορες υπηρεσίες.

- *Logo*

Εδώ περιλαμβάνεται ένα αρχείο εικόνας που χρησιμοποιείται κατά την δημιουργία του PDF

- *Dimoskopisi*

Εδώ υπάρχουν κάποια αρχεία .dat τα οποία σχετίζονται με την δημοσκόπηση

Τέλος υπάρχουν και τα εξής αρχεία :

- *adminNotes.txt*

Εδώ καταχωρούνται οι άμεσες σημειώσεις που μπορούν να γράψουν οι admins και να είναι ορατές σε όλους τους άλλους (admins).

- *dimoi.txt*

Εδώ καταχωρούνται οι δήμοι με δίπλα τους ένα μοναδικό αριθμό που τους αντιστοιχεί.

- *Axis_Url.txt*

Εδώ υπάρχουν οι οργανισμοί και δίπλα το link με την σελίδα των υπηρεσιών που διαθέτουν. Το αρχείο διαβάζεται δυναμικά κάθε φορά και εμφανίζεται όταν απαιτηθεί από τον admin.

Αναλυτικά ο κατάλογος egovernment_platform περιλαμβάνει :

- *Astinomiko_tmima*

Εδώ υπάρχουν όλα τα απαραίτητα αρχεία που αφορούν την υπηρεσία της Αστυνομίας. Το αρχείο που γίνεται τελικά deploy στον application server είναι Astinomiko_tmima.jar

- *Deh*

Εδώ υπάρχουν όλα τα απαραίτητα αρχεία που αφορούν την υπηρεσία της ΔΕΗ. Το αρχείο που γίνεται τελικά deploy στον application server είναι deh.jar

- *DimosBolou*

Εδώ υπάρχουν όλα τα απαραίτητα αρχεία που αφορούν την υπηρεσία του δήμου Βόλου. Το αρχείο που γίνεται τελικά deploy στον application server είναι DimosBolou.jar

- *DimosLarissas*

Εδώ υπάρχουν όλα τα απαραίτητα αρχεία που αφορούν την υπηρεσία του δήμου Λάρισας. Το αρχείο που γίνεται τελικά deploy στον application server είναι DimosLarissas.jar

- *WorkflowView*

Εδώ υπάρχουν όλα τα απαραίτητα αρχεία που αφορούν την δημιουργία της Πύλης όπως αναφέρθηκαν στην αρχική ενότητα. Το αρχείο που γίνεται τελικά deploy στον application server είναι το WorkflowView.ear

4.6 Εγκατάσταση της εφαρμογής

Για την ορθή λειτουργία-εγκατάσταση γίνονται τα εξής βήματα

- Αντιγραφή των καταλόγων από το cd στο σκληρό δίσκο πχ C:
- Οι κατάλογοι έχουν την ίδια «σχετικότητα» όπως και στο cd
- Ο φάκελος jboss-4.0.5.GA που αποτελεί τον application server περιέχει ήδη (deploy) τα αρχεία των εφαρμογών. Αλλιώς ακολουθείται η εξής διαδικασία
 - 1) Είτε χειροκίνητα μέσω του εργαλείου ant
 - 2) Είτε μέσω κάποιας πλατφόρμας ανάπτυξης λογισμικού (πχ NETBEANS IDE)
- Εγκατάσταση του mysql-4.1.22-win32 προκειμένου να έχουμε το σωστό sql server και του mysql-query-browser-1.1.5-win.msi για να κάνουμε πλοήγηση και εισαγωγή δεδομένων στις βάσεις.

Σημείωση : Η βάση δεδομένων στην οποία έχουν πρόσβαση όλα τα EJBs είναι η mydata. Το username και password που χρησιμοποιούν κατά την πρόσβαση είναι root και java αντίστοιχα.

Μέσα στον φάκελο του application server υπάρχει και το πιστοποιητικό του SSL

5 Επίλογος

Στο κεφάλαιο αυτό γίνεται σύνοψη των αποτελεσμάτων της εργασίας δίνοντας κάποια συμπεράσματα καθώς και κάποιες αναφορές σε βελτιώσεις και μελλοντικές επεκτάσεις του συστήματος.

5.1 Σύνοψη

Στην παρούσα διπλωματική εργασία πραγματοποιήθηκε ο σχεδιασμός και η υλοποίηση μιας Πύλης ηλεκτρονικής Διακυβέρνησης προσανατολισμένη στον Πολίτη.

Αναπτύχθηκαν παράλληλα και κάποιες δικτυακές υπηρεσίες που προσφέρονται (σαν παραδείγματα) προς τον πολίτη καθώς και μια κονσόλα διαχείρισης της Πύλης από εξειδικευμένο προσωπικό.

Συγκεκριμένα αναπτύχθηκαν :

- Δυο δικτυακές υπηρεσίες : Μεταδημότευση και Άνοιγμα οικογενειακής Μερίδας
- Μηχανισμοί που αφορούν την μορφή και ορθή λειτουργία της Πύλης
- Ενδιάμεσο στρώμα (middleware) που αναλαμβάνει την επικοινωνία με τις δικτυακές υπηρεσίες
- Μηχανισμός Διαχείρισης της Πύλης

Επιπλέον χαρακτηριστικά του συστήματος :

- Προστασία από μη εξουσιοδοτημένη πρόσβαση σε σελίδες και υπηρεσίες της Πύλης
- Μηχανισμός ασφάλειας των δεδομένων που στέλνονται μέσω του πρωτοκόλλου SSL
- Δημιουργία μοναδικού κωδικού συναλλαγής για κάθε πράξη που πραγματοποιείται από τους πολίτες.
- Καταγραφή όλων των πράξεων που πραγματοποιούνται τόσο από τους Πολίτες όσο και από τον Διαχειριστή/admin σε logs
- Δημιουργία εγγράφων σε μορφή PDF που μπορεί να κατεβάσει ο πολίτης και πιστοποιούν την συναλλαγή του
- Εύκολη και απλή η παραμετροποίηση του συστήματος από τον Διαχειριστή/admin

5.2 Συμπεράσματα

Όπως αναφέρθηκε και στην προηγούμενη υποενότητα η διπλωματική εργασία κινήθηκε σε δύο πλαίσια. Στην δημιουργία μιας εύχρηστης και λειτουργικής Πύλης και στην ανάπτυξη διαδικτυακών υπηρεσιών. Για το πρώτο κομμάτι χρησιμοποιήθηκαν τεχνολογίες που προσδίδουν δυναμικότητα σε μία σελίδα (χρήση jsp και servlets ,EJBs,beans) ενώ για το δεύτερο κομμάτι όλοι αυτοί οι μηχανισμοί που είναι απαραίτητοι για την ανάπτυξη μιας διαλειτουργικής πλατφόρμας που είναι και η ουσία της Ηλεκτρονικής Διακυβέρνησης. Οι τεχνολογίες που χρησιμοποιήθηκαν είναι τα πρότυπα SOAP και WSDL.

5.3 Μελλοντικές Επεκτάσεις

Παρ'όλου που έγινε προσπάθεια να αναπτυχθεί μια Πύλη που να περιλαμβάνει εκείνα τα στοιχεία που την κάνουν ρεαλιστική, υπάρχει προφανώς πληθώρα ιδεών και λειτουργιών που θα μπορούσαν να προστεθούν.

Σχετικά με την Πύλη :

- Υλοποίηση των στοιχείων που τώρα υπάρχουν απλά σαν σχεδιαστικές οντότητες όπως WAP , προσωποποίηση σελίδας, RSS, Forum
- Θα μπορούσαν να εκδίδονται ειδικά πιστοποιητικά αναγνώρισης των χρηστών για την παροχή μεγαλύτερης ασφάλειας πέρα του ονόματος και κωδικού χρήστη

Σχετικά με της Υπηρεσίες :

- Θα μπορούσαν σταδιακά να προστίθενται νέες υπηρεσίες ή και να επεκτείνονται οι υπάρχουσες. Για παράδειγμα οι δύο υλοποιημένες υπηρεσίες αφορούν δύο δήμους τον Βόλο και την Λάρισα.
- Ανάπτυξη μιας επέκτασης SOAP τόσο στην πλευρά του ΠΕΠ όσο και από την πλευρά των δήμων η οποία θα συμπίπτει τα δεδομένα που αποστέλλονται και λαμβάνονται αντίστοιχα μειώνοντας έτσι τον χρόνο μεταφορά τους άρα και τον συνολικό χρόνο από την στιγμή της αίτησης του πελάτη μέχρι την λήψη της απάντησης
- Ανάπτυξη μηχανισμού σχετικά με το versioning του συστήματος. Δηλαδή η δυνατότητα μιας εφαρμογής να παρατηρεί τις αλλαγές που πραγματοποιούνται σε κάποιες δικτυακές υπηρεσίες, να προσαρμόζεται στις αλλαγές αυτές κρατώντας στο ακέραιο τις υπηρεσίες που προσφέρονταν σε παλαιότερες εκδόσεις του συστήματος
- Θα μπορούσε να υλοποιηθεί ένας μηχανισμός ασύγχρονης επικοινωνίας μεταξύ του ΠΕΠ και των δικτυακών εφαρμογών ώστε το ΠΕΠ να μπορεί να απαντάει άμεσα σε αιτήσεις οι οποίες χρήζουν γρήγορης απάντησης αφήνοντας άλλες λιγότερο σημαντικές για αργότερα. Όπως έχει αναφερθεί το μοντέλο που ισχύει στην υλοποίηση είναι το σύγχρονο.
- Χρήση μηχανισμών για περεταίρω ασφάλεια στην επικοινωνία μεταξύ του ΠΕΠ και των δικτυακών υπηρεσιών. Πέρα από το πρωτόκολλο SSL που χρησιμοποιήθηκε στην παρούσα εργασία υπάρχει πληθώρα (επιπλέον) τεχνολογιών που ειδικεύονται στην ασφάλεια XML αρχείων όπως για παράδειγμα XML Signature, XML Encryption), WS-Security, SAML (Security Assertion Markup Language). Τέλος θα μπορούσαν να χρησιμοποιηθούν και κάποιες επιπλέον ακόμα δυνατότητες που προσφέρονται όπως η δημιουργία ρόλων και εισαγωγή ομάδων χρηστών με συγκεκριμένα προνόμια.

5.4 Προσωπική άποψη

Σε όλο αυτό το κείμενο που προηγήθηκε έγινε μια προσπάθεια από τον συγγραφέα να αναλυθούν τόσο οι βασικές έννοιες της Ηλεκτρονικής Διακυβέρνησης όσο και κάποιες τεχνολογίες και τεχνικές που απαιτούνται προκειμένου να επιτευχθεί η υλοποίηση της. Έγινε προσπάθεια ώστε ο αναγνώστης του κειμένου να μπορέσει να αποκτήσει μια έστω και μερική εικόνα για όλα αυτά τα ζητήματα που αφορούν την σχεδίαση και ανάπτυξη της Ηλεκτρονικής Διακυβέρνησης αποτελώντας (ευχόμενος) αφετηρία για κάτι καινούργιο και πιο ολοκληρωμένο.

Τέλος νομίζω ότι αξίζει να προσεχθεί η πληθώρα εργαλείων και τεχνολογιών που είναι ήδη διαθέσιμες αλλά δεν έχουν αξιοποιηθεί σε εφαρμογές που να βελτιώνουν τις κοινωνίες και την ζωή των πολιτών γενικότερα. Το e government σίγουρα αποτελεί μια προσπάθεια να αξιοποιηθεί όλη η υπάρχουσα γνώση και τεχνολογία προς όφελος και του πολίτη. Όσο όμως δεν επενδύονται από το κράτος τα κατάλληλα κεφάλαια στα πανεπιστήμια με σκοπό την έρευνα και την προώθηση της τεχνολογίας αλλά αντίθετα αναλώνονται σε κινήσεις που κατακερματίζουν κάθε τέτοια προσπάθεια τόσο η κοινωνία μας θα παρα-μένει τεχνολογικά αναλφάβητη χάνοντας την νέα επανάσταση του 21 αιώνα, την τεχνολογική. Το πανεπιστήμιο, τελικά οφείλει να προάγει τον πολιτισμό και την γνώση, τόσο την τεχνολογική όσο και την ανθρωπιστική, προκειμένου να «μεταδώσει» την κριτική και ορθή σκέψη στους πολίτες που θα είναι πλέον ικανοί να αντιληφθούν τις θέσεις και αντιθέσεις της κοινωνίας μας. Επειδή μπορεί να κατηγορηθώ από τον αναγνώστη ότι ο επίλογος αυτός ξεφεύγει από τα πλαίσια της διπλωματικής εργασίας θα προβάλω το επιχείρημα ότι το πανεπιστήμιο αποτελούσε πάντα κέντρο όχι μόνο διδασκαλίας αλλά και χώρο πνευματικής ζύμωσης πολιτών, σκεπτόμενων ατόμων που ωθούν την κοινωνία προς τα εμπρός και όχι άτομα υπόλογα και φοβισμένα. Γι' αυτό αξίζει να δίνεται ιδιαίτερη προσοχή και ευαισθησία σε αυτό τον ζωντανό οργανισμό. Αυτός τέλος θα ήθελα να ήταν και ο επίλογος της ακαδημαϊκής μου ιδιότητας ως φοιτητής του τμήματος Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων.

Κυριάκος Μ. Κατσαρής

6. Αναφορές

1. Adamantios Dimitrios (2003) : «Evaluating e-Government Citizens Portals», Μεταπτυχιακή εργασία , Αριστοτέλειο Πανεπιστήμιο,Θεσσαλονίκη
2. Bruce Eckel : «Thinking in Java 3rd edition» , Prentice-Hall
3. Darwin Ian (2001): Java Cookbook, O'Reilly
4. David Chappell (2002) : «Java Web Services» ,O'Reilly
5. Dion Almaer (2003) : «Creating Web Services with Apache Axis» , O'Reilly
6. Doug Tidwell , James , Kulchenko (2001) : «Programming Web Services with Soap» , O'Reilly
7. Drigas, Koukianakis, Domoxoudis (2004) : «E-Government Structure for e-Protocol, e-Application Submission and Internal Organizational and Operational Support - A.S», Dimokritos , Ag. Paraskevi
8. Duane K Fields, Mark A. Kolb (2002): «Web Development with JavaServer Pages», Manning Publications
9. Dumitri Kononov (2006): «Developing and deploying Web service EJB endpoints on a JBoss Application Server September», JBoss institute
10. Interchange of Data between Administrations, European Commission, <http://www.europa.eu.int/idabc>
11. John Wiley (2005) : «Master EJB and J2EE» , O'Reilly
12. Laxxuss : «Servlets and JavaServer Pages - The J2EE Web Tier [JSP] , Sams
13. Laxxuss I. (2002) : «Building Web Services with Java - Making Sense of XML, SOAP, WSDL and UDDI», Sams
14. Miliband David «e-Government Interoperability Framework» , Cabinet Office-Office,London
15. Miliband David, «e-Government Schema Guidelines for XML», Cabinet Office-Office of the e-Envoy, London
16. Setting up Apache Tomcat and a simple Apache SOAP client for SSL communication, Apache ,April 2001, http://ws.apache.org/soap/docs/install/FAO_Tomcat_SOAP_SSL.html
17. Tambouris Efthimios (2006) : «An Integrated Platform for Realising Online One-Stop Government» , TEE

18. Zhiyuan Fang (2001) : «E-Government in Digital Era: Concept, Practice, and Developmen», chung Yuan Cristian University
19. Γκελμπέσης Αλέξανδρος (2003) : «Web Services για e government», Διπλωματική εργασία , Εθνικό Μετσόβιο Πολυτεχνείο , Ζωγράφου
20. Δήμος Βόλου , <http://www.volos-city.gr>
21. Ελληνικό πλαίσιο διαλειτουργικότητας, «Δημόσια διαβούλευση για τη διαλειτουργικότητα», <http://www.infosoc.gr>
22. Κακαρόντζας Γιώργος , «Προγραμματισμός internet με την τεχνολογία των java Server Pages» , ΤΕΙ Λαρίσας, Λάρισα
23. Κέντρο εξυπηρέτησης Πολιτών , Υπουργείο Εσωτερικών, <http://www.kep.gov.gr>

Παράρτημα Α

EJBs

Δίνεται το αρχείο ejb-jar (που αφορά την Αστυνομία) το οποίο είναι απαραίτητο για την δημιουργία και πρόσβαση των EJBs στην βάση δεδομένων

```
<?xml version="1.0" encoding="UTF-8"?>
<ejb-jar version="2.1" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/ejb-jar_2_1.xsd">
  <display-name>Astinomiko_tmima</display-name>
  <enterprise-beans>
    <session>
      <display-name>AstinomikoSessionSB</display-name>
      <ejb-name>AstinomikoSessionBean</ejb-name>
      <local-home>astinomiko.AstinomikoSessionLocalHome</local-home>
      <local>astinomiko.AstinomikoSessionLocal</local>
      <ejb-class>astinomiko.AstinomikoSessionBean</ejb-class>
      <session-type>Stateless</session-type>
      <transaction-type>Container</transaction-type>
      <ejb-local-ref>
        <ejb-ref-name>ejb/AstinomikoTmimaBean</ejb-ref-name>
        <ejb-ref-type>Entity</ejb-ref-type>
        <local-home>astinomiko.AstinomikoTmimaLocalHome</local-home>
        <local>astinomiko.AstinomikoTmimaLocal</local>
        <ejb-link>AstinomikoTmimaBean</ejb-link>
      </ejb-local-ref>
    </session>
    <entity>
      <description>jdbc:mysql://localhost/mydata [root on Default
schema]</description>
      <display-name>AstinomikoTmimaEB</display-name>
      <ejb-name>AstinomikoTmimaBean</ejb-name>
      <local-home>astinomiko.AstinomikoTmimaLocalHome</local-home>
      <local>astinomiko.AstinomikoTmimaLocal</local>
      <ejb-class>astinomiko.AstinomikoTmimaBean</ejb-class>
      <persistence-type>Container</persistence-type>
      <prim-key-class>java.lang.String</prim-key-class>
      <reentrant>false</reentrant>
      <abstract-schema-name>AstinomikoTmima</abstract-schema-name>
      <cmp-field>
        <field-name>arithtautotitas</field-name>
      </cmp-field>
      <cmp-field>
        <field-name>eponimo</field-name>
      </cmp-field>
```



```

    <cmp-field>
      <field-name>onoma</field-name>
    </cmp-field>
    <primkey-field>arithtautotitas</primkey-field>
    <query>
      <description>auto generated method</description>
      <query-method>
        <method-name>findByArithtautotitas</method-name>
        <method-params>
          <method-param>java.lang.String</method-param>
        </method-params>
      </query-method>
      <ejb-ql>SELECT OBJECT(a) FROM AstinomikoTmima AS a WHERE
a.arithtautotitas = ?1</ejb-ql>
    </query>
    <query>
      <description>auto generated method</description>
      <query-method>
        <method-name>findByEponimo</method-name>
        <method-params>
          <method-param>java.lang.String</method-param>
        </method-params>
      </query-method>
      <ejb-ql>SELECT OBJECT(a) FROM AstinomikoTmima AS a WHERE
a.eponimo = ?1</ejb-ql>
    </query>
    <query>
      <description>auto generated method</description>
      <query-method>
        <method-name>fmdByOnoma</method-name>
        <method-params>
          <method-param>java.lang.String</method-param>
        </method-params>
      </query-method>
      <ejb-ql>SELECT OBJECT(a) FROM AstinomikoTmima AS a WHERE
a.onoma = ?1</ejb-ql>
    </query>
  </entity>
</enterprise-beans>
<assembly-descriptor>
  <container-transaction>
    <method>
      <ejb-name>AstinomikoTmimaBean</ejb-name>
      <method-name>*</method-name>
    </method>
    <trans-attribute>Required</trans-attribute>
  </container-transaction>
  <container-transaction>
    <method>
      <ejb-name>AstinomikoSessionBean</ejb-name>

```



```
<method-name>*</method-name>
</method>
<trans-attribute>Required</trans-attribute>
</container-transaction>
</assembly-descriptor>
</ejb-jar>
```

Απαραίτητες συναρτήσεις στα EJBs που αφορούν την sql (περίπτωση Δήμου Βόλου)

- **Finder methods :**
 - **findByArithMerida : SELECT OBJECT(s) FROM StoixeiaDimoti AS s WHERE s.arithMerida = ?1**
 - **ffindByOnomaPatros : SELECT OBJECT(s) FROM StoixeiaDimoti AS s WHERE s.onomaPatros = ?1**
 - **findByOnomaMitros : SELECT OBJECT(s) FROM StoixeiaDimoti AS s WHERE s.onomaMitros = ?1**
 - **findByOnoma : SELECT OBJECT(s) FROM StoixeiaDimoti AS s WHERE s.onoma = ?1**
 - **findByEponimo : SELECT OBJECT(s) FROM StoixeiaDimoti AS s WHERE s.eponimo = ?1**
 - **findByToposGennisis : SELECT OBJECT(s) FROM StoixeiaDimoti AS s WHERE s.toposGennisis = ?1**
- **Select methods :**
 - **SelectMax : SELECT MAX(a.arithMerida) FROM StoixeiaDimoti AS a**

Πρότυπα WSDL

• WSDL πρότυπο Δήμου Βόλου

```
<?xml version="1.0" encoding="UTF-8" ?>
<wsl:definitions targetNamespace="http://localhost:8080/axis/services/DhmosBolou"
  xmlns:apache="http://xml.apache.org/xml-soap"
  xmlns:impl="http://localhost:8080/axis/services/DhmosBolou"
  xmlns:intf="http://localhost:8080/axis/services/DhmosBolou"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsl="http://schemas.xmlsoap.org/wsl/"
  xmlns:wslsoap="http://schemas.xmlsoap.org/wsl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsl:message name="add_marriageRequest">
    <wsl:part name="in0" type="xsd:string" />
    <wsl:part name="in1" type="xsd:string" />
    <wsl:part name="in2" type="xsd:int" />
    <wsl:part name="in3" type="xsd:date" />
    <wsl:part name="in4" type="xsd:string" />
    <wsl:part name="in5" type="xsd:int" />
  </wsl:message>
  <wsl:message name="del_from_municipalityResponse">
    <wsl:part name="del_from_municipalityReturn" type="xsd:int" />
  </wsl:message>
  <wsl:message name="add_to_municipalityRequest">
    <wsl:part name="in0" type="xsd:string" />
    <wsl:part name="in1" type="xsd:string" />
    <wsl:part name="in2" type="xsd:string" />
    <wsl:part name="in3" type="xsd:string" />
    <wsl:part name="in4" type="xsd:date" />
    <wsl:part name="in5" type="xsd:string" />
    <wsl:part name="in6" type="xsd:int" />
  </wsl:message>
  <wsl:message name="add_marriageResponse">
    <wsl:part name="add_marriageReturn" type="xsd:int" />
  </wsl:message>
  <wsl:message name="del_from_municipalityRequest">
    <wsl:part name="in0" type="xsd:string" />
    <wsl:part name="in1" type="xsd:string" />
    <wsl:part name="in2" type="xsd:string" />
    <wsl:part name="in3" type="xsd:string" />
    <wsl:part name="in4" type="xsd:date" />
    <wsl:part name="in5" type="xsd:string" />
    <wsl:part name="in6" type="xsd:int" />
    <wsl:part name="in7" type="xsd:string" />
  </wsl:message>
  <wsl:message name="add_to_municipalityResponse">
    <wsl:part name="add_to_municipalityReturn" type="xsd:int" />
  </wsl:message>
  <wsl:portType name="IpiresiesLocal">
    <wsl:operation name="add_to_municipality" parameterOrder="in0 in1 in2 in3 in4 in5 in6">
      <wsl:input message="impl:add_to_municipalityRequest" name="add_to_municipalityRequest" />
      <wsl:output message="impl:add_to_municipalityResponse" name="add_to_municipalityResponse" />
    </wsl:operation>
    <wsl:operation name="del_from_municipality" parameterOrder="in0 in1 in2 in3 in4 in5 in6 in7">
      <wsl:input message="impl:del_from_municipalityRequest" name="del_from_municipalityRequest" />
    </wsl:operation>
  </wsl:portType>
</wsl:definitions>
```



```

<wsdl:output message="impl:del_from_municipalityResponse" name="del_from_municipalityResponse" />
</wsdl:operation>
<wsdl:operation name="add_marriage" parameterOrder="in0 in1 in2 in3 in4 in5">
  <wsdl:input message="impl:add_marriageRequest" name="add_marriageRequest" />
  <wsdl:output message="impl:add_marriageResponse" name="add_marriageResponse" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="DhmosBolouSoapBinding" type="impl:IpiresiesLocal">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="add_to_municipality">
    <wsdlsoap:operation soapAction="" />
    <wsdl:input name="add_to_municipalityRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://Ipiresies"
        use="encoded" />
    </wsdl:input>
    <wsdl:output name="add_to_municipalityResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://localhost:8080/axis/services/DhmosBolou" use="encoded" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="del_from_municipality">
    <wsdlsoap:operation soapAction="" />
    <wsdl:input name="del_from_municipalityRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://Ipiresies"
        use="encoded" />
    </wsdl:input>
    <wsdl:output name="del_from_municipalityResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://localhost:8080/axis/services/DhmosBolou" use="eucoded" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="add_marriage">
    <wsdlsoap:operation soapAction="" />
    <wsdl:input name="add_marriageRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://Ipiresies"
        use="encoded" />
    </wsdl:input>
    <wsdl:output name="add_marriageResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://localhost:8080/axis/services/DhmosBolou" use="encoded" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="IpiresiesLocalService">
  <wsdl:port binding="impl:DhmosBolouSoapBinding" name="DhmosBolou">
    <wsdlsoap:address location="http://localhost:8080/axis/services/DhmosBolou" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

• WSDL πρότυπο Δήμου Λάρισας

```

<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions targetNamespace="http://localhost:8080/axis/services/DhmosLarissas"
  xmlns:apache="http://xml.apache.org/xml-soap"

```



```

xmlns:impl="http://localhost:8080/axis/services/DhmosLarissas"
xmlns:intl="http://localhost:8080/axis/services/DhmosLarissas"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<wsdl:message name="add_to_municipalityRequest">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:string" />
  <wsdl:part name="in3" type="xsd:string" />
  <wsdl:part name="in4" type="xsd:date" />
  <wsdl:part name="in5" type="xsd:string" />
  <wsdl:part name="in6" type="xsd:int" />
</wsdl:message>
<wsdl:message name="add_to_municipalityResponse">
  <wsdl:part name="add_to_municipalityReturn" type="xsd:int" />
</wsdl:message>
<wsdl:message name="add_marriageResponse">
  <wsdl:part name="add_marriageReturn" type="xsd:int" />
</wsdl:message>
<wsdl:message name="del_from_municipalityRequest">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:string" />
  <wsdl:part name="in3" type="xsd:string" />
  <wsdl:part name="in4" type="xsd:date" />
  <wsdl:part name="in5" type="xsd:string" />
  <wsdl:part name="in6" type="xsd:int" />
  <wsdl:part name="in7" type="xsd:string" />
</wsdl:message>
<wsdl:message name="del_from_municipalityResponse">
  <wsdl:part name="del_from_municipalityReturn" type="xsd:int" />
</wsdl:message>
<wsdl:message name="add_marriageRequest">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:int" />
  <wsdl:part name="in3" type="xsd:date" />
  <wsdl:part name="in4" type="xsd:string" />
  <wsdl:part name="in5" type="xsd:int" />
</wsdl:message>
<wsdl:portType name="IpiresiesLarissaLocal">
  <wsdl:operation name="add_to_municipality" parameterOrder="in0 in1 in2 in3 in4 in5 in6">
    <wsdl:input message="impl:add_to_municipalityRequest" name="add_to_municipalityRequest" />
    <wsdl:output message="impl:add_to_municipalityResponse" name="add_to_municipalityResponse" />
  </wsdl:operation>
  <wsdl:operation name="del_from_municipality" parameterOrder="in0 in1 in2 in3 in4 in5 in6 in7">
    <wsdl:input message="impl:del_from_municipalityRequest" name="del_from_municipalityRequest" />
    <wsdl:output message="impl:del_from_municipalityResponse" name="del_from_municipalityResponse" />
  </wsdl:operation>
  <wsdl:operation name="add_marriage" parameterOrder="in0 in1 in2 in3 in4 in5">
    <wsdl:input message="impl:add_marriageRequest" name="add_marriageRequest" />
    <wsdl:output message="impl:add_marriageResponse" name="add_marriageResponse" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="DhmosLarissasSoapBinding" type="impl:IpiresiesLarissaLocal">
  <wsdl:soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
</wsdl:binding>
<wsdl:operation name="add_to_municipality">

```

```

<wsdl:operation soapAction="" />
<wsdl:input name="add_to_municipalityRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    namespace="http://Ipiresies" use="encoded" />
</wsdl:input>
<wsdl:output name="add_to_municipalityResponse">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    namespace="http://localhost:8080/axis/services/DhmosLarissas" use="encoded" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="del_from_municipality">
  <wsdlsoap:operation soapAction="" />
<wsdl:input name="del_from_municipalityRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    namespace="http://Ipiresies" use="encoded" />
</wsdl:input>
<wsdl:output name="del_from_municipalityResponse">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    namespace="http://localhost:8080/axis/services/DhmosLarissas" use="encoded" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="add_marriage">
  <wsdlsoap:operation soapAction="" />
<wsdl:input name="add_marriageRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    namespace="http://Ipiresies" use="encoded" />
</wsdl:input>
<wsdl:output name="add_marriageResponse">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    namespace="http://localhost:8080/axis/services/DhmosLarissas" use="encoded" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="IpiresiesLarissaLocalService">
  <wsdl:port binding="impl:DhmosLarissasSoapBinding" name="DhmosLarissas">
    <wsdlsoap:address location="http://localhost:8080/axis/services/DhmosLarissas" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

• WSDL πρότυπο Οργανισμού ΔΕΗ

```

<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions targetNamespace="http://localhost:8080/axis/services/Deh"
  xmlns:apache="http://xml.apache.org/xml-soap"
  xmlns:impl="http://localhost:8080/axis/services/Deh"
  xmlns:intf="http://localhost:8080/axis/services/Deh"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:message name="check_time_passedRequest">
    <wsdl:part name="in0" type="xsd:int" />
    <wsdl:part name="in1" type="xsd:string" />
    <wsdl:part name="in2" type="xsd:string" />
    <wsdl:part name="in3" type="xsd:string" />
    <wsdl:part name="in4" type="xsd:int" />
  </wsdl:message>

```



```

: <wsdl:message name="check_time_passedResponse">
  <wsdl:part name="check_time_passedReturn" type="xsd:int" />
</wsdl:message>
: <wsdl:message name="match_dataResponse">
  <wsdl:part name="match_dataReturn" type="xsd:int" />
</wsdl:message>
: <wsdl:message name="match_dataRequest">
  <wsdl:part name="in0" type="xsd:int" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:string" />
  <wsdl:part name="in3" type="xsd:string" />
</wsdl:message>
: <wsdl:portType name="DehSessionLocal">
  <wsdl:operation name="match_data" parameterOrder="in0 in1 in2 in3">
    <wsdl:input message="impl:match_dataRequest" name="match_dataRequest" />
    <wsdl:output message="impl:match_dataResponse" name="match_dataResponse" />
  </wsdl:operation>
  <wsdl:operation name="check_time_passed" parameterOrder="in0 in1 in2 in3 in4">
    <wsdl:input message="impl:check_time_passedRequest" name="check_time_passedRequest" />
    <wsdl:output message="impl:check_time_passedResponse" name="check_time_passedResponse" />
  </wsdl:operation>
</wsdl:portType>
: <wsdl:binding name="DehSoapBinding" type="impl:DehSessionLocal">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="match_data">
    <wsdlsoap:operation soapAction="" />
    <wsdl:input name="match_dataRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://deh"
        use="encoded" />
    </wsdl:input>
    <wsdl:output name="match_dataResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://localhost:8080/axis/services/Deh" use="encoded" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="check_time_passed">
    <wsdlsoap:operation soapAction="" />
    <wsdl:input name="check_time_passedRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://deh"
        use="encoded" />
    </wsdl:input>
    <wsdl:output name="check_time_passedResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://localhost:8080/axis/services/Deh" use="encoded" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
: <wsdl:service name="DehSessionLocalService">
  <wsdl:port binding="impl:DehSoapBinding" name="Deh">
    <wsdlsoap:address location="http://localhost:8080/axis/services/Deh" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

• WSDL πρότυπο Αστυνομίας

```

<?xml version="1.0" encoding="UTF-8" ?>
<wsl:definitions targetNamespace="http://localhost:8080/axis/services/AstinomikoTmima"
  xmlns:apache="http://xml.apache.org/xml-soap"
  xmlns:impl="http://localhost:8080/axis/services/AstinomikoTmima"
  xmlns:intf="http://localhost:8080/axis/services/AstinomikoTmima"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsl="http://schemas.xmlsoap.org/wsl/"
  xmlns:wslsoap="http://schemas.xmlsoap.org/wsl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsl:message name="match_dataRequest">
    <wsl:part name="in0" type="xsd:string" />
    <wsl:part name="in1" type="xsd:string" />
    <wsl:part name="in2" type="xsd:string" />
  </wsl:message>
  <wsl:message name="match_dataResponse">
    <wsl:part name="match_dataReturn" type="xsd:int" />
  </wsl:message>
  <wsl:portType name="AstinomikoSessionLocal">
    <wsl:operation name="match_data" parameterOrder="in0 in1 in2">
      <wsl:input message="impl:match_dataRequest" name="match_dataRequest" />
      <wsl:output message="impl:match_dataResponse" name="match_dataResponse" />
    </wsl:operation>
  </wsl:portType>
  <wsl:binding name="AstinomikoTmimaSoapBinding" type="impl:AstinomikoSessionLocal">
    <wslsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsl:operation name="match_data">
      <wslsoap:operation soapAction="" />
      <wsl:input name="match_dataRequest">
        <wslsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://astinomiko" use="encoded" />
      </wsl:input>
      <wsl:output name="match_dataResponse">
        <wslsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://localhost:8080/axis/services/AstinomikoTmima" use="encoded" />
      </wsl:output>
    </wsl:operation>
  </wsl:binding>
  <wsl:service name="AstinomikoSessionLocalService">
    <wsl:port binding="impl:AstinomikoTmimaSoapBinding" name="AstinomikoTmima">
      <wslsoap:address location="http://localhost:8080/axis/services/AstinomikoTmima" />
    </wsl:port>
  </wsl:service>
</wsl:definitions>

```


Παράδειγμα SOAP request – response

• Request :

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns1:hmerominia_apostolis
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0"                               xsi:type="xsd:dateTime"
xmlns:ns1="http://my.name.space/headers">2007-04-
21T14:40:29.577Z</ns1:hmerominia_apostolis>
    <ns2:ipiresia      soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0"                               xsi:type="soapenc:string"
xmlns:ns2="http://my.name.space/headers"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">Astinomia</ns2:ipi
resia>
  </soapenv:Header>
  <soapenv:Body>
    <ns3:match_data
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns3="http://astinomiko">
      <in0                               xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">asdasdasd</in0>
      <in1                               xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">asdad</in1>
      <in2                               xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">adad</in2>
    </ns3:match_data>
  </soapenv:Body>
</soapenv:Envelope>
```

• Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:match_dataResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://astinomiko">
      <match_dataReturn xsi:type="xsd:int">440</match_dataReturn>
    </ns1:match_dataResponse>
```

```
</soapenv:Body>
</soapenv:Envelope>
```

XML πρότυπα

- Αρχείο XmlEmailValues.xml που αφορά την δομή του e mail configuration

```
<?xml version="1.0" encoding="iso-8859-7"?>
<!DOCTYPE email SYSTEM "EmailDTD.dtd">
<email>
  <subject>για σας</subject>
  <from>mkatsar@tee.gr</from>
  <host>mail.uth.gr</host>
  <message>Ολα καλα πανε</message>
</email>
```

- Το Dtd αρχείο EmailDTD.dtd

```
?xml version="1.0" encoding="iso-8859-7"?>
```

```
<!--
Document : EmailDTD.dtd
Created on : 23 Ιανουάριος 2007, 9:18 μμ
Author : kyriakos
Description:
  Purpose of the document follows.

TODO define vocabulary identification data
PUBLIC ID : -//vendor//vocabulary//EN
SYSTEM ID : http://server/path/EmailDTD.dtd
-->
```

```
<!-- TODO define your own vocabulary/syntax. Example follows: -->
```

```
<!ELEMENT email (subject,from,host,message)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT host (#PCDATA)>
<!ELEMENT message (#PCDATA)>
```

- Αρχείο XmlLogValues.xml που αφορά την δομή του log configuration

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

Document : XmlEmailValues.xml
Created on : 23 Ιανουάριος 2007, 9:18 μμ
Author : kyriakos
Description:
Purpose of the document follows.

-->

```
<!DOCTYPE ipiresies SYSTEM "logDTD.dtd">
<ipiresies>
```

```
    <catalog>C://log</catalog>
    <metadimoteusi>Ipiresia_metadimoteusi</metadimoteusi>
    <meridaGamou>Ipiresia_Gamou</meridaGamou>
```

```
</ipiresies>
```

- To DtD αρχείο logDTD.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

```
Document : logDTD.dtd
Created on : 23 Ιανουάριος 2007, 9:18 μμ
Author : kyriakos
Description:
Purpose of the document follows.
```

```
TODO define vocabulary identification data
PUBLIC ID : -//vendor//vocabulary//EN
SYSTEM ID : http://server/path/EmailDTD.dtd
```

```
-->
```

```
<!-- TODO define your own vocabulary/syntax. Example follows: -->
```

```
<!ELEMENT ipiresies (catalog ,metadimoteusi,meridaGamou)>
<!ELEMENT catalog (#PCDATA)>
<!ELEMENT metadimoteusi (#PCDATA)>
<!ELEMENT meridaGamou (#PCDATA)>
```

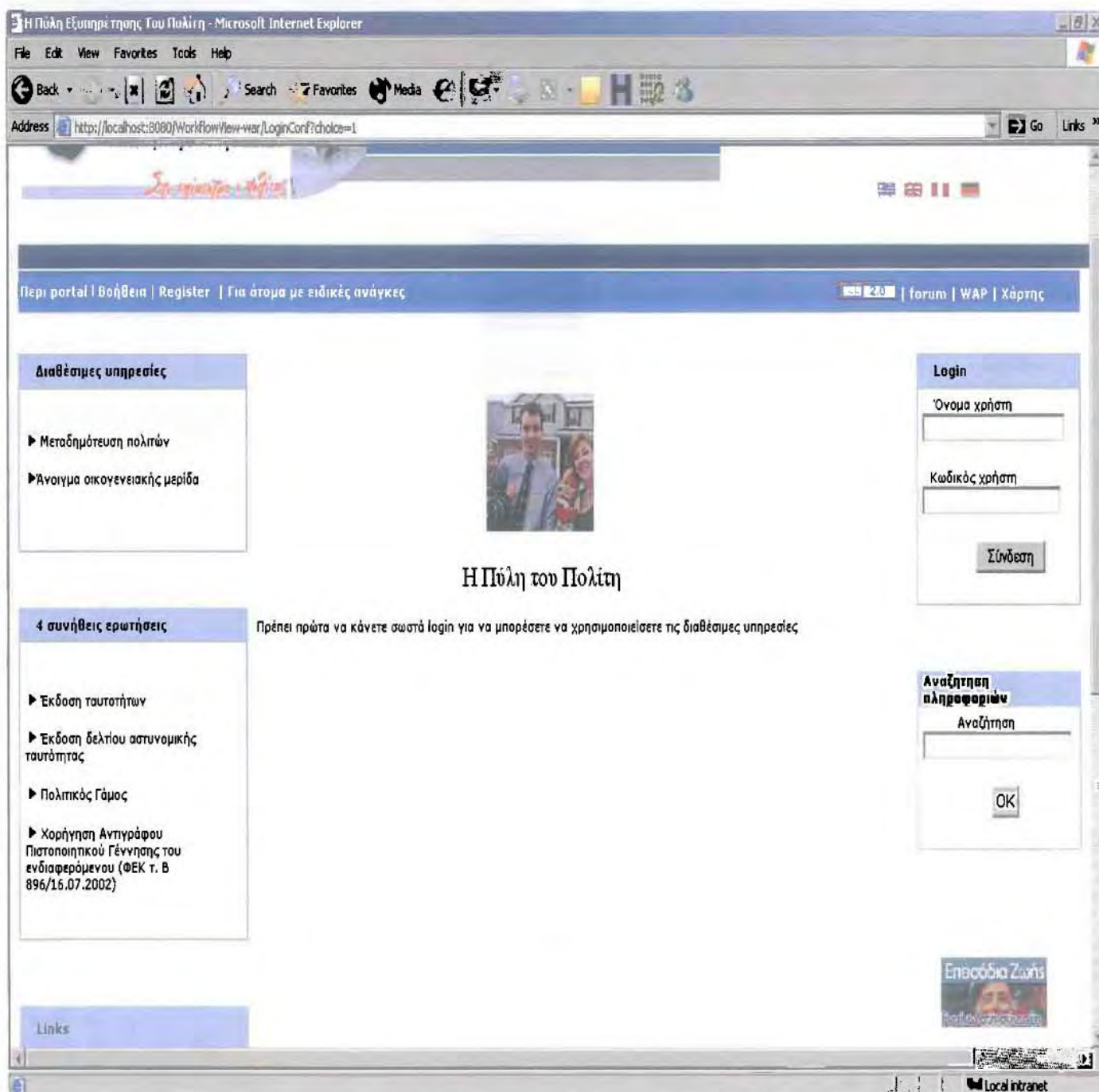

Παράρτημα Β

Οθόνες

Εδώ παρουσιάζονται screenshots για διάφορες περιπτώσεις χρήσης τόσο για τον απλό χρήστη όσο και για τον διαχειριστή.

Περίπτωση απλού χρήστη

Αρχικά θα εμφανιστεί το μήνυμα λάθους που εμφανίζεται όταν γίνεται προσπάθεια εισαγωγής σε μία υπηρεσία χωρίς πρώτα να γίνει login ο πολίτης.



Τώρα θα δειχθεί η διαδικασία login ενός πολίτη στην Πύλη.

Η Πύλη Εξυπηρέτησης Του Πολίτη - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media

Address http://localhost:8080/WorkflowView-war/login.jsp

Go Links

Περι portal | Βοήθεια | Register | Για άτομα με ειδικές ανάγκες

RSS 2.0 | forum | WAP | Χάρτης

Εκτυπώσιμη Έκδοση

E mail

Διαθέσιμες υπηρεσίες

Αρχική Σελίδα: Νέα

- Μεταδημότευση πολιτών
- Ανοιγμα οικογενειακής μερίδας

4 συνήθεις ερωτήσεις

- Έκδοση ταυτοτήτων
- Έκδοση δελτίου αστυνομικής ταυτότητας
- Πολιτικός Γάμος
- Χορήγηση Απηνγράφου Πιστοποιητικού Γέννησης του ενδιαφερόμενου (ΦΕΚ τ. Β 896/16.07.2002)
- *επιπλέον*

Links

- ΠΡΟΕΔΡΟΣ ΔΗΜΟΚΡΑΤΙΑΣ

Η Πύλη του Πολίτη

Νέες υπηρεσίες προστίθενται για την εξυπηρέτηση των πολιτών

Είναι ήδη έτοιμη η νέα πλατφόρμα που εξυπηρετεί αρχικά τους δήμους βόλου και Λάρισσας. Οι υπηρεσίες που προσφέρονται είναι η μεταδημότευση μεταξύ πολιτών αυτών των δύο πόλεων καθώς και η δυνατότητα καταχώρησης της πράξης γαμού και την έκδοση οικογενειακής μερίδας.

Σε λίγο καιρό θα ενσωματωθούν και νέες πόλεις που θα διαθέτουν αυτές τις υπηρεσίες και πολύ σύντομα θα έχουμε και νέες διαθέσιμες υπηρεσίες προς όφελος του πολίτη

Τα τελευταία Νέα

16/01/07 - Ανακοινώθηκαν τα ονόματα των επιτυχόντων για το Κρατικό Πιστοποιητικό Γλωσσικών Δεξιοτήτων. Τα ονόματα των επιτυχόντων στις εξετάσεις του Κρατικού Πιστοποιητικού Γλωσσικών Δεξιοτήτων.

Login

Όνομα χρήστη

Κωδικός χρήστη

Σύνδεση

Είστε login : user

Logout Χρήστη

Αλλαγή των στοιχείων μου

Αναζήτηση πληροφοριών

Αναζήτηση

OK

υποτίστε

Done Local intranet

Όνομα χρήστη : user

Κωδικός χρήστη : user

Τώρα θα γίνει πρόσβαση στην υπηρεσία «Μεταδημότευση Πολιτών». Αξίζει να προσεχθεί η ασφάλεια SSL. Πριν εισαχθούν τα δεδομένα θα δειχθεί ο έλεγχος φόρμας και στην συνέχεια θα δοθούν λανθασμένα δεδομένα κειμένου ώστε να δειχθεί το μήνυμα λάθους που εμφανίζεται

Η Πύλη Εξυπηρέτησης Του Πολίτη - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media

Address <https://localhost:8443/WorkflowView-war/metadmoteusi> Go Links

Εκδοση ταυτοτήτων

Εκδοση δελτίου αστυνομικής ταυτότητας

Πολιτικός Γάμος

Χορήγηση Αντιγράφου Πιστοποιητικού Γέννησης του ενδιαφερόμενου (ΦΕΚ τ. Β 896/16.07.2002)

Links

- ΠΡΟΕΔΡΟΣ ΔΗΜΟΚΡΑΤΙΑΣ
- ΒΟΥΛΗ ΤΩΝ ΕΛΛΗΝΩΝ
- ΥΠΟΥΡΓΕΙΑ
- ΔΙΚΑΣΤΗΡΙΑ
- ΑΝΕΞΑΡΤΗΤΕΣ ΑΡΧΕΣ

forum

Local intranet

Αριθμός Ταυτότητας	<input type="text"/>	
Επώνυμο Δημότη	<input type="text"/>	Δεν επιτρέπονται γούμερα
Όνομα Δημότη	<input type="text"/>	
Όνομα Πατρός	<input type="text"/>	
Όνομα Μητρός	<input type="text"/>	
Ημερομηνία Γέννησης(Έτος-Μήνας-Μέρα)	<input type="text"/>	Βάλτε σε μορφή Χρονας-Μήνας-Μέρα
Τόπος Γέννησης	<input type="text"/>	
A.Φ.Μ	<input type="text"/>	
Δήμος που ανήκετε τώρα	<input type="text" value="Βόλος"/>	
Αριθμός μερίδας που είχατε στην δήμο	<input type="text"/>	Πρέπει να βάλετε αριθμό
Δήμος Προορισμού μεταδημότευσης	<input type="text" value="Λάρισα"/>	
*Αριθμός πελάτη ΔΕΗ	<input type="text"/>	
E mail	<input type="text"/>	Πρέπει να δώσετε μια έγκυρη μορφή διεύθυνσης
<input checked="" type="checkbox"/> Δηλώνω ότι δεν έχω τελέσει θρησκευτικό ή πολιτικό γάμο στην Ελλάδα ή το εξωτερικό.		
<input checked="" type="checkbox"/> Δηλώνω γνωρίζοντας τις επιπτώσεις του νόμου ότι τα στοιχεία που δίνω δεν είναι ψευδή		

Αναζήτηση πληροφοριών

Αναζήτηση

OK

Επισκόπια Ζωής

Οι τιμές ασφαλιστικής ONLINE

Κάνε την αίτησή σου Online Τώρα >>

Done

Η Πύλη Εξυπηρέτησης Του Πολίτη - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

Address <https://localhost:8443/WorkflowView-war/metakadimoteusi.jsp> Go Links

Η Πύλη του Πολίτη

4 συνηθεις ερωτήσεις

- Έκδοση ταυτοτήτων
- Έκδοση δελτίου αστυνομικής ταυτότητας
- Πολιτικός Γάμος
- Χορήγηση Ανηγράφου Πιστοποιητικού Γέννησης του ενδιαφερόμενου (ΦΕΚ τ. Β 896/16.07.2002)

"Μεταδημότευση άγαμου ενήλικου σε δημοτολόγιο δήμου ή κοινότητας εκτός του τόπου καταγωγής του ίδιου ή των γονέων του"

Αριθμός Ταυτότητας	<input type="text" value="Σ123"/>
Επώνυμο Δημότη	<input type="text" value="ΔΗΜΗΤΡΙΟΥ"/>
Όνομα Δημότη	<input type="text" value="ΠΕΤΡΟΣ"/>
Όνομα Πατρός	<input type="text" value="ΚΩΣΤΑΣ"/>
Όνομα Μητρός	<input type="text" value="ΜΑΡΙΑ"/>
Ημερομηνία Γέννησης(Έτος-Μήνας-Μέρα)	<input type="text" value="1970-10-10"/>
Τόπος Γέννησης	<input type="text" value="ΑΘΗΝΑ"/>
Α.Φ.Μ	<input type="text" value="1234566"/>
Δήμος που ανήκετε τώρα	<input type="text" value="Βόλος"/>
Αριθμός μερίδας που είχατε στην δήμο	<input type="text" value="1"/>
Δήμος Προορισμού μεταδημότευσης	<input type="text" value="Λάρισα"/>
*Αριθμός πελάτη ΔΕΗ	<input type="text" value="1"/>
E mail	<input type="text" value="mikatsar@tee.gr"/>

☒ Δηλώνω ότι δεν έχω τελέσει θρησκευτικό ή πολιτικό γάμο στην Ελλάδα ή το εξωτερικό.

Σύνδεση

Είστε login

Logout Χρήστη

Links

- ΠΡΟΕΔΡΟΣ ΔΗΜΟΚΡΑΤΙΑΣ
- ΒΟΥΛΗ ΤΩΝ ΕΛΛΗΝΩΝ
- ΥΠΟΥΡΓΕΙΑ
- ΔΙΚΑΣΤΗΡΙΑ
- ΑΝΕΞΑΡΤΗΤΕΣ ΑΡΧΕΣ

Αναζήτηση πληροποποιών

Αναζήτηση

OK

Επικοινωνία Ζωής

Οι τ.μέ.ς αμ.λυ.β.δ.ς στο Νομό

Κάνε την αίτησή σου Online Τώρα>>>

forum

Local intranet

Το αποτέλεσμα που θα δοθεί από το ΠΕΠ είναι :

Η Πύλη Εξυπηρέτησης Του Πολίτη - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media

Address <https://localhost:8443/WorkflowView-war/metadimoteusi> Go Links »

Η Πύλη του Πολίτη

Αναφορά Διαδικασίας

2/3/2007 : Δυστυχώς κύριε/α ΔΗΜΗΤΡΙΟΥ παρουσιάστηκε σφάλμα κατά την προσπάθεια διεκπεραίωσης της υπηρεσίας. Το λάθος που προέκυψε είναι το εξής Κωδικός 440 : Μη ύπαρξη φυσικού προσώπου στην Αστυνομία

Βεβαιωθείτε ότι έχετε εισάγει ορθά τα στοιχεία που απαιτούνται στην φόρμα συμπλήρωσης. Σε περίπτωση που το λάθος δεν οφείλεται σε εσάς αλλά σε τεχνικούς λόγους, σας ζητούμε εκ των προτέρων συγγνώμη για την ταλαιπωρία

Ευχαριστούμε,
Πύλη Εξυπηρέτησης Πολιτών

*Για περισσότερες πληροφορίες επικοινωνήστε στο τηλέφωνο 210-xxxxxx και ρωτήστε για τον κωδικό λάθους που σας εμφανίστηκε προηγουμένως

4 συνήθειες ερωτήσεις

- ▶ Έκδοση ταυτοτήτων
- ▶ Έκδοση δελτίου αστυνομικής ταυτότητας
- ▶ Πολιτικός Γάμος
- ▶ Χορήγηση Απηνγράφου Πιστοποιητικού Γέννησης του ενδιαφερόμενου (ΦΕΚ τ. Β 896/16.07.2002)

Links

- ▶ ΠΡΟΕΔΡΟΣ ΔΗΜΟΚΡΑΤΙΑΣ
- ▶ ΒΟΥΛΗ ΤΩΝ ΕΛΛΗΝΩΝ
- ▶ ΥΠΟΥΡΓΕΙΑ
- ▶ ΔΙΚΑΣΤΗΡΙΑ
- ▶ ΑΝΕΞΑΡΤΗΤΕΣ ΑΡΧΕΣ

Σύνδεση

Είστε login

Logout Χρήστη

Αναζήτηση πληροφοριών

Αναζήτηση

OK

Επεσόδια Ζωής

Οι τιμές αμύλου

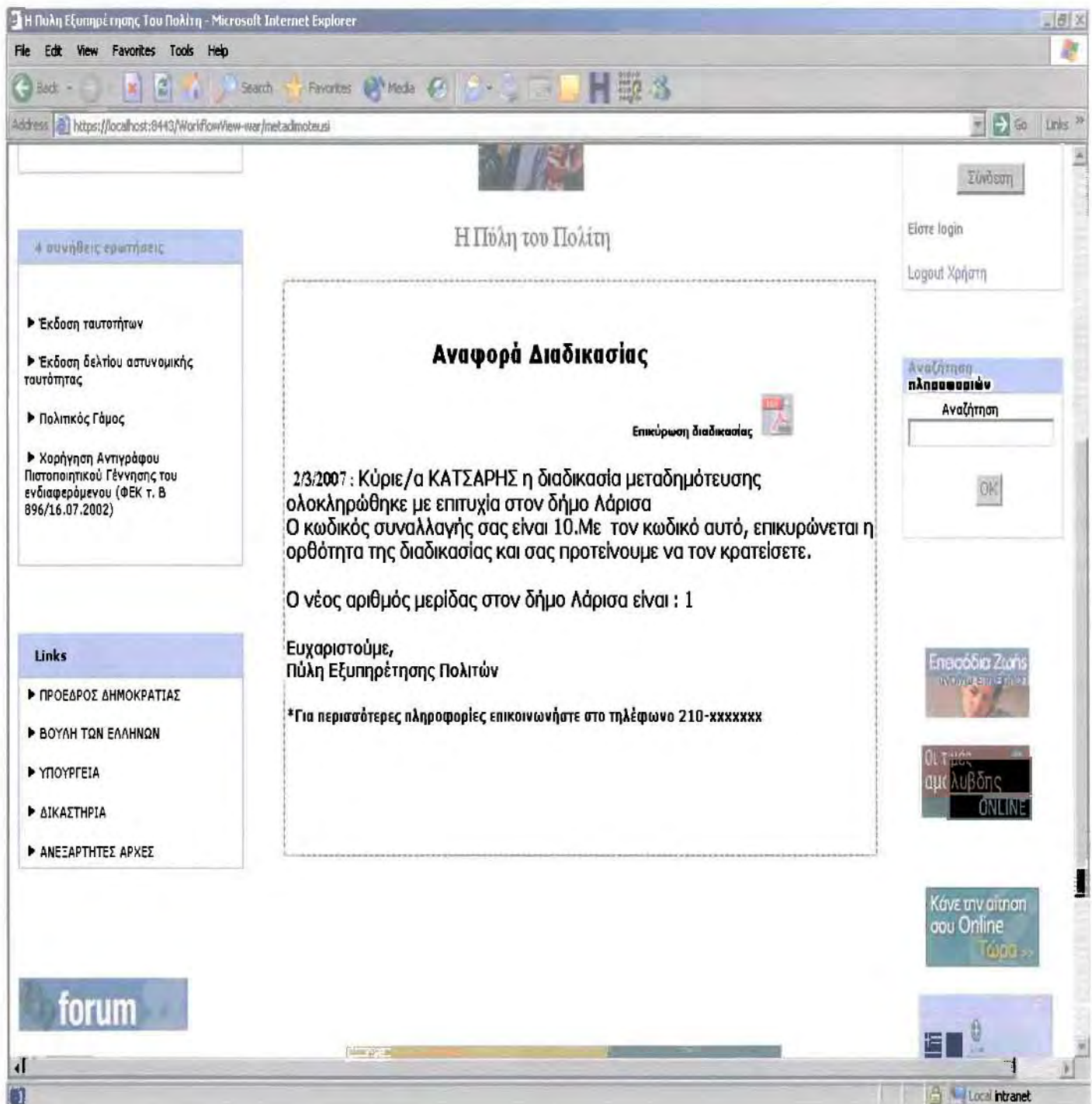
Κάνε την αίτηση σου Online

forum

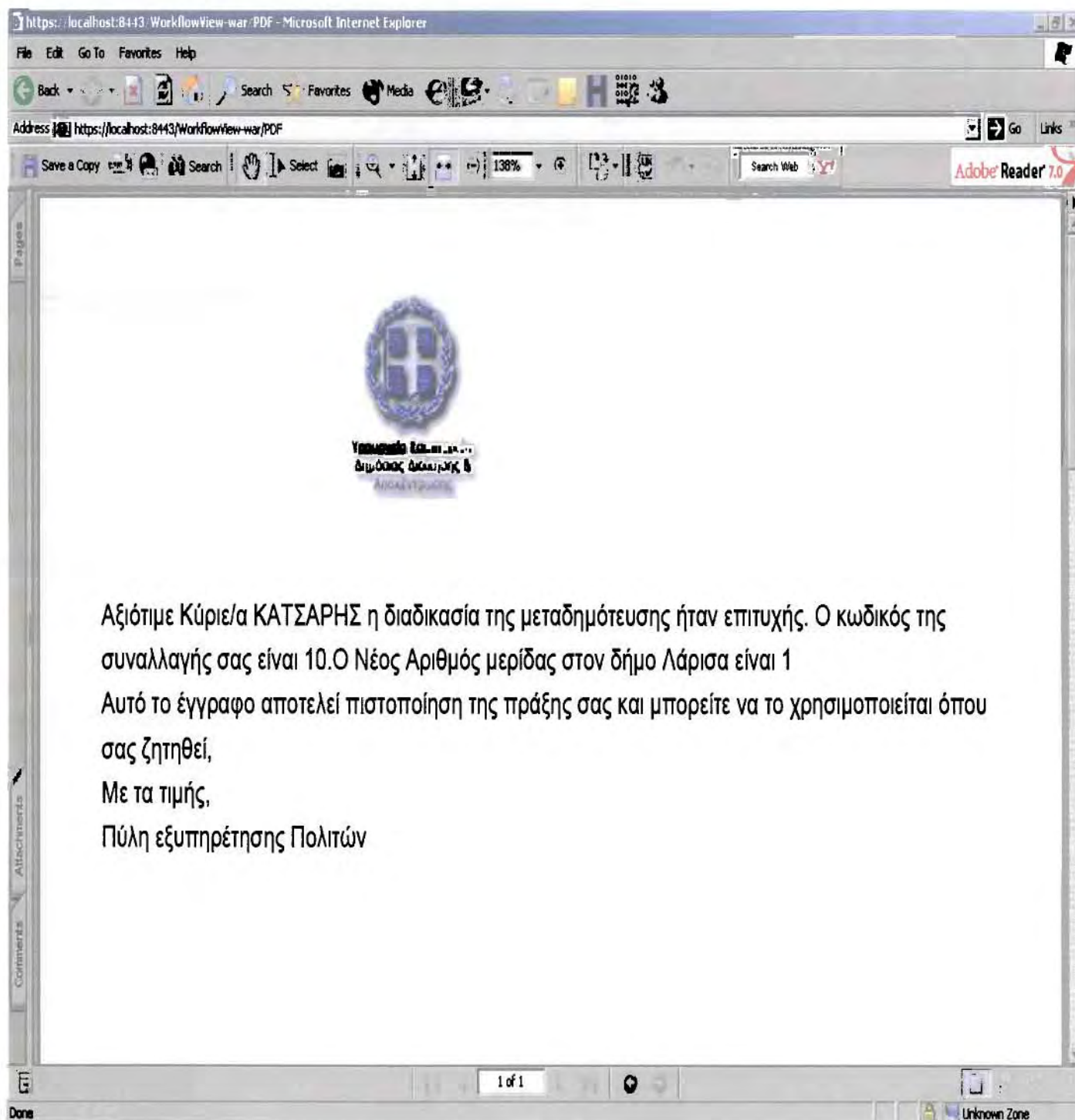
Done Local intranet

Φαίνεται αναλυτικά στο μήνυμα σε ποιο «κομμάτι» της διαδικασίας εμφανίστηκε το λάθος.

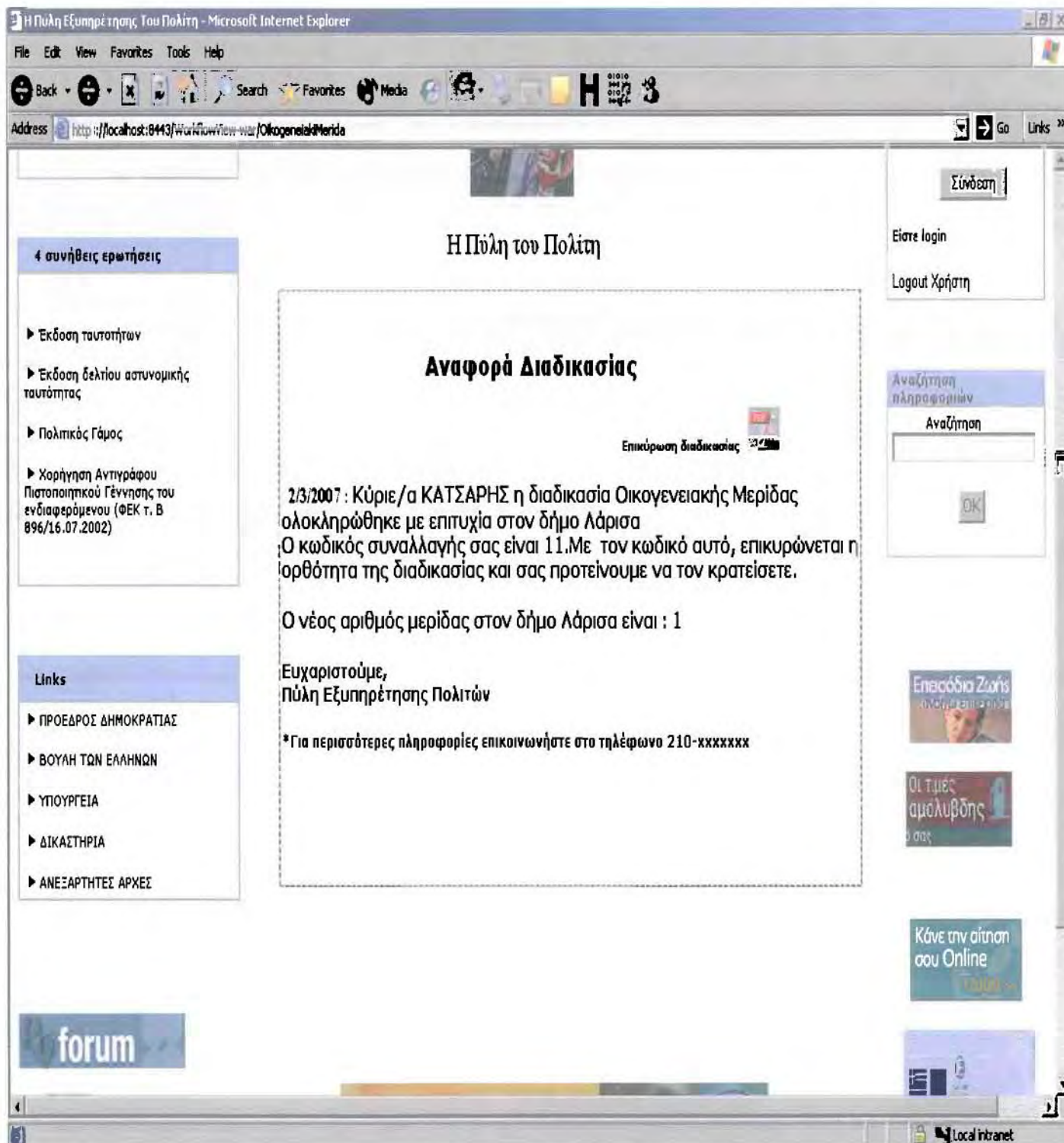
Τώρα θα δειχθεί μια επιτυχημένη διαδικασία της υπηρεσίας. Στο παράδειγμα θα γίνει μεταδημότευση από τον δήμο Βόλου στον δήμο Λάρισας.



Και το PDF που δημιουργείται :



Για την υπηρεσία «Άνοιγμα οικογενειακής Μερίδας» θα δοθεί μια εικόνα όταν η διαδικασία είναι επιτυχής :



Οθόνες Περίπτωσης διαδικασιών απλού χρήστη

Περίπτωση διαχειριστή

Τώρα θα δειχθούν κάποιες από τις ενέργειες που μπορεί να κάνει ο διαχειριστής. Παρακολούθηση SOAP μηνυμάτων που ανταλλάσσονται μεταξύ του ΠΕΠ και των δικτυακών υπηρεσιών :

SOAP Monitor - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost:8080/axis/SOAPMonitor> Go Links

localhost

Time	Target Service	Status
Most Recent		
6:55:01 μμ		Complete

Remove Remove All Filter ...

Time: 6:55:01 μμ Target Service: Status: Complete

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  <soapenv:Header>
    <ns1:timeFromina_apostolis soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next" soapenv:mustUnderstand="0" xsi:type="
    <ns2:ipiresia soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next" soapenv:mustUnderstand="0" xsi:type="soapenv:st
  </soapenv:Header>
  <soapenv:Body>
    <ns3:match_data soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:ns3="http://astinomiko">
      <in0 xsi:type="soapenc:string" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">8#x3A3;8#x3A7;12</in0>
      <in1 xsi:type="soapenc:string" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">8#x39A;8#x391;8#x3A4;8#x3A
      <in2 xsi:type="soapenc:string" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">8#x39A;8#x3A5;8#x3A1;8#x39
    </ns3:match_data>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  <soapenv:Header>
    <ns1:match_dataResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:ns1="http://astinomiko"
      <match_dataReturn xsi:type="xsd:int">440</match_dataReturn>
    </ns1:match_dataResponse>
  </soapenv:Header>
  <soapenv:Body>
    </soapenv:Body>
</soapenv:Envelope>
```

☒ Refresh XML text Switch Layout

Start Stop The SOAP Monitor is started.

Applet: SOAPMonitorApplet started Local intranet

Εδώ φαίνονται τα links με τις διαδικτυακές υπηρεσίες των Οργανισμών

Η Πύλη Εξυπηρέτησης Του Πολίτη - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media

Address https://localhost:8443/WorkflowView-war/web_services.jsp Go Links »

Διευθυντής - Τμήμα

Περι portal | Βοήθεια | Logout

Διαχείριση των Web Services

- Deploy / unDeploy Web services
- Παρακολούθηση soap μηνυμάτων
- Επισκόπηση των Web Services

Διαχείριση των Μελών

- Διαγραφή Account μέλους
- Προσθήκη Διαχειριστή Υπηρεσιών
- Προβολή στοιχείων κάποιου μέλους

Διαχείριση κωδικών συναλλαγών

- Διαγραφή μιας συναλλαγής
- Προβολή στοιχείων κάποιας συναλλαγής

Δυναμική επισκόπηση των Web services κάθε οργανισμού

Αστυνομία : <http://localhost:8080/axis/servlet/AxisServlet>

Δήμος Λάρισας: <http://localhost:8080/axis/servlet/AxisServlet>

Δήμος Βόλου: <http://localhost:8080/axis/servlet/AxisServlet>

ΔΕΗ: <http://localhost:8080/axis/servlet/AxisServlet>

Πληροφορίες Διαχειριστή

Username : admin

Όνομα : admin

e-mail : admin@uth.gr

Μέρα : 2 Μάρπος 2007

Logout

Ρύθμιση του e mail

- Ιδιότητες των email

Διαχείριση των Logs

- Ιδιότητες των logs

Done Local intranet

Εδώ βλέπουμε τα στοιχεία ενός μέλους της Πύλης δίνοντας το Username

The screenshot shows a web browser window titled "Η Πύλη Εξυπηρέτησης Του Πολίτη - Microsoft Internet Explorer". The address bar shows the URL "https://localhost:8443/WorkflowView-war/ShowAccount.jsp". The page has a blue header with navigation links: "Περι portal", "Βοήθεια", and "Logout".

The main content area is divided into three columns:

- Left Column:**
 - Διαχείριση των Web Services**
 - ▶ Deploy / unDeploy Web servicescs
 - ▶ Παρακολούθηση soap μηνυμάτων
 - ▶ Επισκόπηση των Web Services
 - Διαχείριση των Μελών**
 - ▶ Διαγραφή Account μέλους
 - ▶ Προσθήκη Διαχειριστή Υπηρεσιών
 - ▶ Προβολή στοιχείων κάποιου μέλους
 - Διαχείριση κωδικών συναλλαγών**
 - ▶ Διαγραφή μιας συναλλαγής
 - ▶ Προβολή στοιχείων κάποιας συναλλαγής
- Center Column:**
 - Προβολή των στοιχείων ενός μέλους/πολίτη**
 - Username μέλους:
 -
 - Username: user
 - Όνομα: Δημήτρης
 - Επίθετο: Δημητρίου
 - E-mail: user@uth.gr
- Right Column:**
 - Πληροφορίες Διαχειριστή**
 - Username : admin
 - Όνομα : admin
 - e-mail : admin@uth.gr
 - Μέρα : 2 Μάρτιος 2007
 -
 - Ρύθμιση του e mail**
 - ▶ Ιδιότητες των email
 - Διαχείριση των Logs**
 - ▶ Ιδιότητες των logs

The status bar at the bottom shows "ocal intranet".

Εδώ εμφανίζονται τα στοιχεία που αντιστοιχούν σε μία συναλλαγή

Η Πύλη Εξυπηρέτησης Του Πολίτη - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media

Address <https://localhost:8443/WorkflowView-war/ShowSinal.jsp> Go Links

Περι portal | Βοήθεια | Logout

Διαχείριση των Web Services

- Deploy / unDeploy Web services
- Παρακολούθηση soap μηνυμάτων
- Επισκόπηση των Web Services

Διαχείριση των Μελών

- Διαγραφή Account μέλους
- Προσθήκη Διαχειριστή Υπηρεσιών
- Προβολή στοιχείων κάποιου μέλους

Διαχείριση κωδικών συναλλαγών

- Διαγραφή μιας συναλλαγής
- Προβολή στοιχείων κάποιας συναλλαγής

Πληροφορίες Διαχειριστή

Username : admin

Όνομα : admin

e-mail : admin@uth.gr

Μέρα : 2 Μάρτιος 2007

Logout

Ρύθμιση του e mail

- Ιδιότητες των email

Διαχείριση των Logs

- Ιδιότητες των logs

Προβολή των στοιχείων ενός μέλους/πολίτη με βάση μια συναλλαγή

Αριθμός συναλλαγής

Όνομα: ΚΥΡΙΑΚΟΣ

Επίθετο: ΚΑΤΣΑΡΗΣ

E-mail: mkatsar@tee.gr

Περιγραφή Υπηρεσίας: Οικογενειακή Μεριδα

Εδώ παρουσιάζεται η φόρμα με την οποία επιτυγχάνεται η παραμετροποίηση των Email

Η Πύλη Εξυπηρέτησης Του Πολίτη - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media

Address <https://localhost:8443/WorkflowView-war/EmailConf> Go Links

Διαχείριση των Web Services

- ▶ Deploy / unDeploy Web services
- ▶ Παρακολούθηση soap μηνυμάτων
- ▶ Επισκόπηση των Web Services

Διαχείριση των Μελών

- ▶ Διαγραφή Account μέλους
- ▶ Προσθήκη Διαχειριστή Υπηρεσιών
- ▶ Προβολή στοιχείων κάποιου μέλους

Διαχείριση κωδικών συναλλαγών

- ▶ Διαγραφή μιας συναλλαγής
- ▶ Προβολή στοιχείων κάποιας συναλλαγής

Παραμετροποίηση του e mail που στέλνεται σε πολίτη μετά απο μια επιτυχημένη έκβαση κάποιας διαδικασίας

Αποστολέας email Πρέπει να βάλετε τιμή

Host server Πρέπει να βάλετε τιμή

Επικεφαλίδα Πρέπει να βάλετε τιμή

Κυρίως μήνυμα

Ευχαριστούμε που χρησιμοποιήσατε τις υπηρεσίες μας

 Πρέπει να βάλετε τιμή

Πληροφορίες Διαχειριστή

Username : admin

Όνομα : admin

e-mail : admin@uth.gr

Μέρα : 2 Μάρτιος 2007

Ρύθμιση του e mail

- ▶ Ιδιότητες των email

Διαχείριση των Logs

- ▶ Ιδιότητες των logs

Done local intranet

Οθόνες Περίπτωσης διαδικασιών Διαχειριστή συστήματος