

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης  
Σχολή Διοίκησης Και Οικονομίας  
Τμήμα Διοίκησης Επιχειρήσεων



Τίτλος: Περιγραφή Πληροφοριακού Συστήματος Εταιρίας  
Κινητής Τηλεφωνίας

Επιβλέπων καθηγητής: Παπαδάκης Νικόλαος

Φοιτητής: Λαμπαρδάκης Εμμανουήλ

Ηράκλειο, 2015

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τους ανθρώπους που μου στάθηκαν σε όλη τη φοιτητική μου πορεία και στη διαδικασία της πτυχιακής εργασίας, δηλαδή τους γονείς μου και τους φίλους μου, που σε στιγμές απογοήτευσης δεν με άφησαν να ξεχάσω το στόχο μου.

Επίσης τον καθηγητή μου, κύριο Παπαδάκη Νικόλαο, που με την αστείρευτη υπομονή του με βοήθησε να φτάσω στο τέλος των φοιτητικών μου χρόνων και στην ολοκλήρωση της πτυχιακής μου εργασίας.

## Σύνοψη

Στη συγκεκριμένη εργασία γίνεται η περιγραφή και η υλοποίηση ενός Πληροφοριακού Συστήματος. Πιο συγκεκριμένα, αναλύεται το Επιχειρησιακό Σύστημα μιας Εταιρείας Κινητής Τηλεφωνίας. Για τη διαχείριση της βάσης δεδομένων χρησιμοποιούνται οι κύριες τεχνολογίες HTML, PHP, MYSQL, CSS, JAVASCRIPT και XAMPP. Το πληροφοριακό σύστημα με το οποίο ασχοληθήκαμε μπορεί να γνωστοποιήσει ανά πάσα στιγμή πληροφορίες σχετικά με τα καταστήματα, τους εργαζόμενους, τους πελάτες, τους λογαριασμούς, τους τηλεφωνικούς αριθμούς, τις βλάβες, τις κεραιές και τα πακέτα σύνδεσης.

## Abstract:

In the present assignment we try to describe and implement the structure of an Information System. More specifically, we analyze the operational system of a mobile phone operator. For the database management, the main technologies that are utilized are HTML, PHP, MYSQL, CSS, JAVASCRIPT and XAMPP. The Information System we studied is able to provide information concerning the Operator's branches, employees, customers, statements, phone numbers, system breakdown, antennae status and connection packages at any time.

## Πίνακας Περιεχομένων

Ευχαριστίες.....	i
Σύνοψη.....	ii
Abstract.....	iii
Πίνακας Περιεχομένων.....	iv
Πίνακας Εικόνων.....	v
ΚΕΦΑΛΑΙΟ 1: Εισαγωγή.....	1
1.1 Περίληψη.....	1
ΚΕΦΑΛΑΙΟ 2: Μεθοδολογία Υλοποίησης – Τεχνολογίες.....	2
2.1 Η HTML.....	2
2.1.1 Τι είναι η HTML.....	2
2.1.2 Η ιστορία της HTML.....	2
2.1.3 HTML και η εξέλιξή της.....	3
2.1.4 XHTML.....	4
2.1.5 HTML 5.0.....	4
2.2 Javascript.....	6
2.3 PHP.....	6
2.4 CSS.....	6
2.5 XML (Extensive Markup Language).....	7
2.5.1 Τι είναι XML (Extensive Markup Language).....	7
2.5.2 Ορισμός της δομής ενός XML εγγράφου.....	8
2.5.3 Σχεδιαστικοί στόχοι της XML.....	8
2.5.5 XML έννοιες.....	9
2.5.6 Σύνοψη.....	9
2.6 MySql.....	10
2.7 Hosting Πλατφόρμα – Xampp.....	10
ΚΕΦΑΛΑΙΟ 3: Επιχειρησιακό Σύστημα – Εταιρία Κινητής Τηλεφωνίας.....	12
3.1 Πληροφοριακό Σύστημα για Εταιρία Κινητής Τηλεφωνίας.....	12
3.2 Οι διαδικασίες που θα υποστηρίξει το σύστημα.....	12
ΚΕΦΑΛΑΙΟ 4: Μοντελοποίηση του Πληροφοριακού Συστήματος.....	14
4.1 Διάγραμμα οντοτήτων – σχέσεων.....	14
4.2 Μετάφραση του μοντέλου στο σχεσιακό μοντέλο.....	15

4.3	Περιορισμοί	ακεραιότητας	και	συναρτησιακές	εξαρτήσεις.....	15
4.4	Μετατροπή του μοντέλου σε τρίτη κανονική μορφή με διατήρηση των συναρτησιακών εξαρτήσεων και χωρίς απώλεια πληροφορίας.....					16
ΚΕΦΑΛΑΙΟ 5:	DLL	εντολές	για	τη δημιουργία	του Data Model.....	17
ΚΕΦΑΛΑΙΟ 6:	Τεχνολογίες	για	την	υλοποίηση	των διαδικασιών.....	24
6.1	Για	την	υλοποίηση	του	συστήματος χρησιμοποιήθηκαν.....	24
6.2	CSS	Code		(Βασικό Interface).....		24
6.3	RHP					25
6.3.1	RHP		(Φόρμα	καταχώρησης).....		25
6.3.2	RHP		(Φόρμα	παρουσίασης).....		28
6.3.3	RHP (Φόρμα report).....					30
6.3.4	RHP (Φόρμα πληρωμής).....					32
6.3.5	RHP (Λογαριασμός).....					33
ΚΕΦΑΛΑΙΟ 7:	Εγκατάσταση	Web		Εφαρμογής.....		34
ΚΕΦΑΛΑΙΟ 8:	Λειτουργικότητα	των	διαδικασιών	της	εφαρμογής.....	40
ΚΕΦΑΛΑΙΟ 9:					Επίλογος.....	44
ΚΕΦΑΛΑΙΟ 10:		Βιβλιογραφικές		Αναφορές/Πηγές.....		46

#### ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1:	Εργαλείο	εκτέλεσης	δυναμικών	ιστοσελίδων.....		10
Εικόνα 2:	Μοντέλο		σύνδεσης	οντοτήτων.....		14
Εικόνα 3:	Μοντέλο		σύνδεσης	οντοτήτων.....		15
Εικόνα 4:	Μοντέλο		σύνδεσης	οντοτήτων.....		16
Εικόνα 5:	Χρήση	CSS Code	σε	HTML	αρχείο.....	24
Εικόνα 6:	Χρήση	CSS Code	σε	HTML	αρχείο.....	25
Εικόνα 7:	RHP	φόρμα	καταχώρησης	λογαριασμού.....		25
Εικόνα 8:	RHP	φόρμα	καταχώρησης	κεραίας.....		26
Εικόνα 9:	RHP	φόρμα	καταχώρησης	του εξαρτώμενου μέλους.....		26
Εικόνα 10:	RHP	φόρμα	καταχώρησης	στοιχείων	υπαλλήλου.....	27
Εικόνα 11:	RHP	φόρμα	καταχώρησης	στοιχεία	αριθμού.....	27

Εικόνα 12:	PHP	φόρμα	καταχώρησης	στοιχεία	αριθμο- ύ.....	28		
Εικόνα 13:	PHP	φόρμα	παρουσίασης	προϊστάμενου	καταστήμα- τος.....	28		
Εικόνα 14:	PHP	φόρμα	παρουσίασης	στοιχείων	κεραί- ας.....	29		
Εικόνα 15:	PHP	φόρμα	παρουσίασης	στοιχείων	πελάτη.....	29		
Εικόνα 16:	PHP	φόρμα	παρουσίασης	στοιχείων	αριθμού	τηλεφώνο- υ.....	30	
Εικόνα 17:	PHP	φόρμα	αναφοράς	συμβολαίου	και	προ	πλήρω- σης.....	30
Εικόνα 18:	PHP	φόρμα	αναφορών	αποδείξεων	καταστήμα- τος.....	31		
Εικόνα 19:	PHP	φόρμα	αναφοράς	νέων	συμβολαί- ων.....	31		
Εικόνα 20:	PHP	φόρμα	αναφοράς	ανανέωσης	προπληρωμένων	συμβολαίων.....	32	
Εικόνα 21:	PHP	φόρμα	πληρω- μής.....			32		
Εικόνα 22:	PHP	φόρμα	λογαριασμού.....			33		
Εικόνα 23:	Πίνακας	Ελέγχου	XAMPP.	Διεργασία	εφαρμογών	Apache, MySQL.....	34	
Εικόνα 24:	Τοπικός	δίσκος,	φάκελος	htdocs.....		35		
Εικόνα 25:	Τοπικός	δίσκος,	φάκελος	cell company.....		36		
Εικόνα 26:	Αρχική	σελίδα	XAMPP.....			37		
Εικόνα 27:	Αρχική	σελίδα	phpMyAd.....			38		
Εικόνα 28:	Δημιουργία	βάσης	δεδομένων.....			38		
Εικόνα 29:	Εισαγωγή	στη	Βάση	Δεδομένων.....		39		
Εικόνα 30:	Υποκατάστημα	της	εταιρείας.....			40		
Εικόνα 31:	Διαγραφή	υποκαταστήματος.....				40		
Εικόνα 32:	Υπάλληλοι.....					41		
Εικόνα 33:	Πελάτες.....					41		
Εικόνα 34:	Κεραίες	Εταιρείας.....				42		
Εικόνα 35:	Καταχωρημένες	και	επιδιορθωμένες	κεραίες.....		42		
Εικόνα 36:	Πληροφορίες	αριθμού	τηλεφώνου.....			42		
Εικόνα 37:	Πληρωμές.....					43		

# ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

Η πτυχιακή εργασία είναι ένα αναπόσπαστο κομμάτι των σπουδών, ένα ταξίδι γνώσης για τον εκάστοτε σπουδαστή αφού του δίνεται η δυνατότητα να γνωρίσει, να αναλύσει και να εντυφώσει σε διάφορα θέματα υπό τη συστηματική καθοδήγηση του επιτηρητή της πτυχιακής. Δίνοντάς του ταυτόχρονα βάσεις και «πολεμοφόδια» για την επικείμενη έξοδό του στην αγορά εργασίας.

## 1.1 Περίληψη

Το αντικείμενο της εργασίας είναι η ανάπτυξη ενός Πληροφοριακού Συστήματος που συνδέεται άμεσα με την Κινητή Τηλεφωνία. Βασικές διαδικασίες που χρησιμοποιούνται είναι η μοντελοποίηση του Πληροφοριακού Συστήματος και η ανάπτυξη του λογισμικού σε Web Platform. Για να υλοποιηθούν όλα τα παραπάνω χρησιμοποιήθηκαν οι κύριες τεχνολογίες HTML, PHP, MYSQL, CSS, JAVASCRIPT και XAMPP.



## ΚΕΦΑΛΑΙΟ 2: Μεθοδολογία Υλοποίησης - Τεχνολογίες

### 2.1 Η HTML

#### 2.1.1 Τι είναι η HTML

Τα αρχικά HTML προέρχονται από τις λέξεις Hyper Text Markup Language. Η HTML δεν είναι γλώσσα προγραμματισμού. Είναι μια markup language (γλώσσα σήμανσης), δηλαδή ένας συγκεκριμένος, ειδικά δομημένος τρόπος με τον οποίο γράφουμε κείμενα. Ο κάθε χρήστης ηλεκτρονικού υπολογιστή, μπορεί να δημιουργήσει ένα αρχείο HTML. Αυτό μπορεί να το πετύχει χρησιμοποιώντας έναν επεξεργαστή κειμένου και στην συνέχεια να δημιουργήσει ένα έγγραφο με τη δομή που επιτρέπει η συγκεκριμένη γλώσσα. Η HTML αποτελεί υποσύνολο της SGML (Standard Generalized Markup Language). Κατασκευάστηκε από την IBM προκειμένου να δώσουν λύση στο πρόβλημα της ανόμοιας απεικόνισης σε διαφορετικά υπολογιστικά συστήματα. Ένας περιηγητής αναγνωρίζει αυτό το συγκεκριμένο τρόπο γραφής, ο οποίος δομεί/αποτελεί ένα HTML έγγραφο και εκτελεί τις εντολές που περιέχεται σε αυτό. Σε αυτό το σημείο πρέπει να τονιστεί ότι η HTML, είναι η πιο ευρέως διαδεδομένη γλώσσα, η οποία χρησιμοποιείται για να περιγραφεί μια ιστοσελίδα στο διαδίκτυο. Παράλληλα είναι και η πρώτη γλώσσα, η οποία χρησιμοποιήθηκε και είχε τη δυνατότητα για κάτι τέτοιο. Ένα HTML έγγραφο, αφού αποτελείται από markup περιεχόμενο, περιέχει διάφορες ετικέτες (tags) για τις οποίες υπάρχουν κανόνες, για το πως θα τις αξιοποιήσει ένας περιηγητής. Οι ετικέτες είναι εντολές, οι οποίες έχουν την δυνατότητα να ορίζουν την αρχή και το τέλος μιας συγκεκριμένης λειτουργίας. Το tag πρέπει να βρίσκεται πάντα ανάμεσα από τα σύμβολα < και >. Οι εντολές είναι case insensitive, δηλαδή δεν έχει σημασία εάν είναι γραμμένες με πεζά ή κεφαλαία. Τέλος, τα HTML έγγραφα έχουν κατάληξη “.htm” ή “html” χωρίς αυτό να σημαίνει ότι δεν μπορούμε να ενσωματώσουμε HTML σε αρχεία τα οποία έχουν διαφορετική κατάληξη. [1]-[11]

#### 2.1.2 Ιστορία της HTML

Το 1980, ο φυσικός Tim Berners-Lee, ο οποίος εργαζόταν στο CERN, πρότεινε το ENQUIRE. Ένα σύστημα για τους ερευνητές του CERN, οι οποίοι θα το χρησιμοποιούσαν προκειμένου να διαχειρίζονται και να μοιράζονται έγγραφα. Το 1989, ο Berners-Lee έγραψε μια σημείωση με την οποία σύστησε ένα σύστημα υπερκειμένου που θα βασιζόταν στο διαδίκτυο. Έτσι, δημιούργησε τη γλώσσα HTML, έναν περιηγητή ο οποίος αναγνώριζε τέτοιου είδους έγγραφα. Επίσης δημιούργησε και ένα σύστημα εξυπηρετητή προς το τέλος του 1990. Εκείνη τη χρονιά ο Berners-Lee σε συνεργασία με τον Βέλγο συνάδελφό του, Robert Cailliau ζήτησαν επιχορήγηση, αλλά το project αυτό, δεν υιοθετήθηκε τελικά από το CERN.

Η πρώτη δημόσια περιγραφή της HTML ήταν ένα έγγραφο το οποίο ονομάστηκε HTML Tags και αναφέρθηκε πρώτη φορά στο διαδίκτυο από τον Berners-Lee στα τέλη του 1991. Σε αυτό το έγγραφο, περιγράφει 20 στοιχεία γύρω από το σχεδιασμό της HTML. Εκτός από την ετικέτα (tag) που αφορούσε υπερσυνδέσμους, η γλώσσα αυτή επηρεάστηκε περισσότερο από την SGML. Γλώσσα η οποία ήταν ήδη γνωστό πρότυπο συγγραφής τεκμηρίων στο CERN. Δεκατρείς από αυτές τις ετικέτες υπάρχουν μέχρι και σήμερα στην HTML 4.0.

Η HTML είναι μια γλώσσα, η οποία χρησιμοποιείται κατά κόρον από τους περιηγητές (browsers), για να διαμορφώνει δυναμικά ιστοσελίδες που περιέχουν κείμενο και εικόνες. Πολλά από τα στοιχεία κειμένου (text elements) της HTML, βρέθηκαν από την τεχνική αναφορά της οργάνωσης ISO με ονομασία «Τεχνικές για τη χρήση της SGML» η οποία περιγράφει διάφορες από τις πρώτες γλώσσες που μορφοποιούσαν κείμενο όπως και η HTML.

Ο Berners-Lee μέχρι και τα μέσα του 1993, θεωρούσε την HTML ως μια εφαρμογή της SGML. Στην συνέχεια όμως δημοσιεύτηκε μια πρόταση για τις προδιαγραφές της HTML η οποία έγινε από τον Berners-Lee και τον Dan Connolly και άλλαξε αυτή την εικόνα. Σε αυτή

την πρόταση συμπεριλαμβανόταν και ο ορισμός μιας γραμματικής της HTML σε SGML Document Type Definition. Στη συνέχεια προτάθηκε η HTML+, από τον Dave Raggett, η οποία περιείχε στοιχεία όπως φόρμες και πίνακες.

Μετά την HTML και την HTML+ ο οργανισμός IETF, αποφάσισε να δημιουργήσει μια ομάδα, η οποία θα ασχολούνταν αποκλειστικά και μόνο με την ανάπτυξη της HTML. Το 1995 ολοκληρώθηκε με επιτυχία η HTML 2.0. Η έκδοση αυτή ήταν η βάση με την οποία θα πορεύονταν οι επόμενες εκδόσεις της γλώσσας, και για το λόγο αυτό ορίστηκε ως πρότυπο. Από το 1996, ο οργανισμός W3C (World Wide Web Consortium) διατήρησε και διαφύλαξε τις προδιαγραφές της HTML, ενώ το 2000 η γλώσσα έγινε διεθνές πρότυπο. Οι τελευταίες προδιαγραφές της γλώσσας, δημοσιεύτηκαν από τον W3C δημιουργώντας την έκδοση 4.0 της γλώσσας. Η δημιουργία της έκδοσης αυτής, προτάθηκε το 1999. Τέλος, η ανάπτυξη της HTML σταμάτησε και στην θέση της γεννήθηκε η γλώσσα XHTML. [1]-[11]

### 2.1.3 HTML και η εξέλιξή της

Παρακάτω συνοψίζεται η εξέλιξη της HTML:

⇒ Έκδοση 2.0

24 Νοεμβρίου 1995: Η HTML 2.0 δημοσιεύεται σαν πρότυπο. (RFC 1866)

25 Νοεμβρίου 1995: Προστίθεται σαν χαρακτηριστικό η φόρμα για ανέβασμα αρχείων. (RFC 1867)

Μάιος 1996: Προστίθεται σαν χαρακτηριστικό οι πίνακες (tables). (RFC 1942)

Αύγουστος 1996: Προστίθεται η δυνατότητα για χάρτες εικόνων στην πλευρά του client. (RFC 1980)

Ιανουάριος 1997: Προστίθενται δυνατότητες internationalization. (RFC 2070)

Μετά τον Ιούνιο του 2000, όλα αυτά τα χαρακτηριστικά δηλώθηκαν ως ξεπερασμένα και ιστορικά. (RFC 2854)

⇒ Έκδοση 3.0

Ιανουάριος 1997: Η HTML 3.2 δημοσιεύτηκε μετά από πρόταση του οργανισμού W3C. Ήταν η πρώτη φορά που αυτός ο οργανισμός ανέπτυξε και έθεσε ως πρότυπο τη γλώσσα. Η ομάδα εργασίας που ανέπτυξε και σχεδίασε την HTML, έκλεισε τον Σεπτέμβριο του 1996, η οποία και ανήκε στον οργανισμό IETF. Στη συνέχεια στην έκδοση HTML 3.2 εγκαταλείφθηκαν τελείως η δυνατότητα για μαθηματικές φόρμουλες, ενώ αντίθετα υιοθετήθηκαν οι περισσότερες από τις οπτικές markup ετικέτες του περιηγητή Netscape. Τα στοιχεία blink της Netscape και marquee της Microsoft εγκαταλείφθηκαν μετά από συμφωνία μεταξύ των δύο εταιριών.

⇒ Έκδοση 4.0

Δεκέμβριος 1997: Η HTML 4.0 προτάθηκε από τον οργανισμό W3C. Στην πρόταση αυτή υπήρχαν τρεις παραλλαγές: 1) Αυστηρή, στην οποία τα αποδοκιμασμένα στοιχεία δεν επιτρέπονταν, 2) Μεταβατική, στην οποία τα αποδοκιμασμένα στοιχεία δεν ήταν απαγορευμένα, 3) Frameset, στην οποία μόνο τα σχετικά με το πλαίσιο (frame), στοιχεία επιτρέπονταν.

Η HTML 4.0 υιοθέτησε πολλούς τύπους στοιχείων και χαρακτηριστικών που συνέλλεξε από διάφορους περιηγητές. Παράλληλα όμως, απέκλεισε τα οπτικά markup χαρακτηριστικά του περιηγητή Netscape για χάρη της CSS (αναλύεται στο επόμενο κεφάλαιο). Τέλος, η HTML 4.0 είναι μια εφαρμογή της SGML σύμφωνα με το πρότυπο ISO 8879 – SGML.

Απρίλιος 1998: Τροποποιήθηκε η γλώσσα χωρίς να αυξηθεί ο αριθμός της έκδοσης.

Δεκέμβριος 1999: Η HTML 4.01 που δημοσιεύτηκε σαν μια πρόταση της W3C, προσέφερε τις τρεις παραλλαγές της HTML 4.0 έχοντας και τα τελευταία τυπογραφικά λάθη που προϋπήρχαν στις προηγούμενες προτάσεις και δημοσιεύτηκαν στις 12 Μαΐου 2001.

Μάιος 2000: Ο οργανισμός ISO αναγνωρίζει την HTML σαν διεθνές πρότυπο. Στη συνέχεια η ομάδα εργασίας του W3C αποφασίζει να σταματήσει την ανάπτυξη της HTML και να

επικεντρωθεί στην ανάπτυξη μιας γλώσσας, που βασίζεται στην XML, και θα αποτελούσε τη συνέχεια της HTML. Αυτή η γλώσσα ονομάστηκε XHTML (eXtensible Hypertext Markup Language) και αναπτύχθηκε από την αρχή μέχρι τα μέσα του 2000. [1]-[11]

#### 2.1.4 XHTML

Η XHTML αποτελείται από XML markup γλώσσες οι οποίες αναπτύσσουν ή αντικαθιστούν εκδόσεις της HTML γλώσσας. Οι γλώσσες αυτές έχουν στόχο τη δημιουργία ιστοσελίδων. Ενώ η HTML είναι υποσύνολο της SGML (Standard Generalized Markup Language), η XHTML αποτελεί μια εφαρμογή της XML η οποία είναι μια πιο περιορισμένη εφαρμογή της SGML.

Τα XHTML έγγραφα όταν γράφονται πρέπει να ικανοποιούν ένα σύνολο κανόνων. Ως αποτέλεσμα, τα έγγραφα αυτά μπορούν να διαβαστούν χρησιμοποιώντας τυποποιημένους συντακτικούς αναλυτές XML (XMLparsers) οι οποίοι είναι αυστηροί ως προς τη σύνταξη. Αντίθετα, για την ανάγνωση των HTML εγγράφων, απαιτείται ένας πιο ελαστικός parser.

Μετά από πρόταση της οργάνωσης W3C δημιουργήθηκε η XHTML έκδοση 1.0 τον Ιανουάριο του 2000. Είναι μια αναδημιουργία των τριών τύπων εγγράφων που είχε η HTML 4.0 αλλά σαν εφαρμογές XML πλέον. Στη συνέχεια πρότεινε την έκδοση XHTML 1.1 η οποία υλοποιήθηκε στις 31 Μαΐου 2001. Τέλος, η XHTML 5 βρίσκεται υπό κατασκευή από το Σεπτέμβριο του 2009 και αποτελεί μέρος της HTML 5. Για την ώρα η πρόταση για την XHTML 5.0 ακόμη αναπτύσσεται.

Ως επανάσταση του διαδικτύου ονόμασε η οργάνωση W3C τον Αύγουστο του 2002 την οικογένεια γλωσσών XHTML. Με την μεταφορά σε XHTML, οι προγραμματιστές έχουν τη δυνατότητα να μπου στον κόσμο της XML, έχοντας πρόσβαση σε όλα τα πλεονεκτήματά της, και ταυτόχρονα μπορούν να νιώθουν σιγουριά για τη συμβατότητα στο μέλλον.

Η ανάπτυξη της XHTML, έγινε κυρίως για να κάνει την HTML περισσότερο επεκτάσιμη και διαλειτουργική με άλλους τύπους δεδομένων. Η HTML 4 υπήρχε ως ένα υποσύνολο της SGML. Η SGML είναι μια γλώσσα πολύπλοκη με την οποία ούτε οι περιηγητές, αλλά ούτε και η HTML 4, ήταν πλήρως συνυφασμένοι. Το 1998 έγινε αποδεκτό το πρότυπο XML, προσφέροντας έτσι στους προγραμματιστές, έναν πολύ απλό τύπο δεδομένων, παρόμοιο με την HTML 4. Η ιδέα ήταν ξεκάθαρη: Για να είναι η HTML συμβατή με όλα τα εργαλεία που υπάρχουν για την XML, χρειάστηκε να αλλάξει η φιλοσοφία της πρώτης σε ένα πρότυπο που βασίζεται η δεύτερη. Αυτό είχε ως αποτέλεσμα, οι (διακομιστές) servers και οι μεσολαβητές (proxies) να έχουν πλέον τη δυνατότητα, να μετατρέπουν περιεχόμενο σε τύπους, που διαβάζονται από περιορισμένες συσκευές, όπως είναι τα κινητά. Επίσης, έγγραφα γραμμένα σε XHTML, χρησιμοποιώντας XML namespaces, παρέχουν επεκτασιμότητα σε άλλες γλώσσες που βασίζονται στην XML. Η Scalable Vector Graphics και η MathML είναι κάποια χαρακτηριστικά παραδείγματα.

Τέλος, η XHTML ως ανανεωμένη μορφή της HTML, δίνει την δυνατότητα στο διαχωρισμό της HTML σε επαναχρησιμοποιήσιμα συστατικά (Components). Η διαδικασία έχει ονομαστεί XHTML Modularization και συνέβαλε στην απομόνωση των μη αποδοτικών χαρακτηριστικών της HTML. [1]-[11]

#### 2.1.5 HTML 5.0

Η X/HTML 5.0 είναι μια υπό-ανάπτυξη γλώσσα η οποία αποτελεί την εξέλιξη του HTML προτύπου. Θα χρησιμοποιείται για την αναπαράσταση και δόμηση περιεχομένου στον παγκόσμιο ιστό και προσφέρει ένα σύνολο σημαντικών χαρακτηριστικών σε σχέση με τον προκάτοχό της. Αναμένεται, οι προδιαγραφές της να έχουν καθοριστεί πλήρως μέσα στα επόμενα χρόνια.

Αυτό που κάνει ξεχωριστή την X/HTML 5.0 είναι α) η λύση-αντιμετώπιση των προβλημάτων σε θέματα τα οποία είχαν οδηγήσει σε ασυμβατότητες στο παρελθόν αλλά και β) η δημιουργία νέων στοιχείων για την υλοποίηση μοντέρνων ιστοσελίδων.

Ο τρόπος με τον οποίο αντιμετωπίζονται τα πράγματα τα οποία οδήγησαν σε λάθη, στο παρελθόν είναι ο εξής:

Για να μην προκύψουν προβλήματα ασυμβατότητας στο μέλλον, αν και η γλώσσα ήδη χρησιμοποιείται αφού ακόμα δεν έχει ολοκληρωθεί, οι δημιουργοί των προδιαγραφών έχουν θέσει ένα βαθμό ωριμότητας σε κάθε στοιχείο της γλώσσας. Σε μια πιο απλή ανάλυση, αυτό σημαίνει πως στοιχεία με μεγάλη ωριμότητα, είναι σίγουρο ότι θα περιέχονται στο επίσημο πρότυπο της γλώσσας. Αντίθετα, τα στοιχεία που δεν έχουν μεγάλη ωριμότητα, σημαίνει πως είναι ακόμη σε πειραματικό στάδιο. Οπότε δεν είναι σίγουρο ότι οι εταιρίες που κατασκευάζουν περιηγητές, θα συνεχίσουν στο μέλλον να υποστηρίζουν σε επόμενες εκδόσεις των εφαρμογών τους.

Από τη σκοπιά των προγραμματιστών, αυτό σημαίνει πως οι ιστοσελίδες που χρησιμοποιούν πειραματικά στοιχεία της X/HTML 5.0 πιθανόν θα πρέπει να ανανεωθούν αφού στο μέλλον υπάρχει το ενδεχόμενο να μην λειτουργούν σωστά. Ακόμα για κάθε στοιχείο των προδιαγραφών της γλώσσας, πρέπει να διαθέτουν δύο άμεσα και εύκολα προσβάσιμα από το κοινό, παραδείγματα που θα το υλοποιούν. Χωρίς να μπορούν νέα χαρακτηριστικά για τη γλώσσα να προταθούν, δίχως τουλάχιστον μια υλοποίηση αυτών. Αυτή η αρχή θεσπίστηκε ώστε να μην εισάγονται, στο πρόχειρο του προτύπου, χαρακτηριστικά χωρίς αιτιολογία όπως είδαμε στην περίπτωση της XHTML 2.

Ιδιαίτερη έμφαση δίνεται στη συμβατότητα καθώς, η γλώσσα σχεδιάστηκε με τέτοιο τρόπο ώστε να επιτρέπει τη συμβατότητα με παλαιούς περιηγητές μέσω επεκτάσεων (scripts) και προσθηκών (plugins). Επίσης, η X/HTML 5.0 πλέον αντιμετωπίζεται σαν μία νέα γλώσσα απεικόνισης client-server εφαρμογών, όχι όπως στο παρελθόν ως μία γλώσσα απεικόνισης εγγράφων. Χωρίς φυσικά να εννοείται πως τα έγγραφα αγνοούνται εντελώς, αλλά μετατρέπεται η οπτική γωνία με την οποία αντιμετωπίζονται κάποια θέματα. Στην ουσία πλησιάζει περισσότερο την ιδέα της δημιουργίας μιας γλώσσας η οποία πρόκειται να χρησιμεύσει για την απεικόνιση δεδομένων στο διαδίκτυο.

Ένα από τα χαρακτηριστικά που κάνουν την X/HTML 5 κατάλληλη γλώσσα για εφαρμογές του διαδικτύου είναι η ικανότητα για αποθήκευση εκτός σύνδεσης. Είναι γνωστό ότι τα προβλήματα για τις εφαρμογές διαδικτύου προκύπτουν όταν δεν υπάρχει δίκτυο. Οι περισσότερες εφαρμογές είναι client-side, προγραμματισμένες σε HTML και Javascript, και χρησιμοποιούν τον web-server όταν χρειαστεί η αποθήκευση κάποιων εγγράφων. Αν προστεθεί και η δυνατότητα για αποθήκευση δεδομένων τοπικά (στον client δηλαδή), τότε ο περιηγητής μετατρέπεται σε μια πλατφόρμα τοπικής εκτέλεσης εφαρμογών. Αυτό αποτελεί θέμα σοβαρής συζήτησης, μεταξύ των ομάδων ανάπτυξης της γλώσσας για να αποφασιστεί, αν θα προκύψουν περισσότερα πλεονεκτήματα απ' ότι μειονεκτήματα.

Η αποθήκευση δεδομένων στον client είναι μια διαδικασία πολυδιάστατη. Η διατήρηση των δεδομένων είναι εφικτή, όσο η σελίδα παραμένει ανοιχτή, όπως εφικτή παραμένει και η δυναμική ανανέωσή της. Τι γίνεται όμως όταν ο χρήστης κλείσει το παράθυρο της σελίδας; Μια πρώτη λύση γι' αυτό, είναι η διατήρηση δεδομένων από το server, για κάποιο συγκεκριμένο χρονικό διάστημα, έτσι ώστε η ανάκτηση αυτών να γίνει αυτόματα όταν ανοιχθεί πάλι η σελίδα. Σαν δεύτερη λύση, είναι η διατήρηση μικρών ποσοτήτων δεδομένων σε cookies. Τα προβλήματα που προκύπτουν με τη δεύτερη λύση, είναι αρχικά το γεγονός ότι τα cookies στέλνονται σε κάθε επικοινωνία με το server, αναγκάζοντας το μέγεθος της πληροφορίας που διατηρείται σε αυτά, να είναι πολύ μικρό. Δευτερεύον πρόβλημα είναι ότι τα cookies περιέχουν μοναδικά ID, δηλαδή ότι η προσπέλαση της πληροφορίας από το server δεν είναι ανά πάσα στιγμή επιτρεπτή, καθώς το ID αυτό αλλάζει.

Τέλος, η X/HTML 5.0 αφήνει πίσω της το μονοσέλιδο μοντέλο και στοχεύει σε έναν διαφορετικό τρόπο συμπεριφοράς, μιας διαδικτυακής συμπεριφοράς. Χρησιμοποιώντας sessions για την αλληλεπίδραση διαφορετικών παραθύρων της ίδιας διαδικτυακής εφαρμογής. Με αποτέλεσμα, την δυνατότητα για επικοινωνία μεταξύ των παραθύρων του ίδιου session. [1]-[11]

## 2.2 Javascript

Η javascript είναι μια γλώσσα σεναρίων (scripting language). Η λειτουργία της είναι να βοηθά τον προγραμματιστή στην κατασκευή διαδραστικών και λειτουργικών εγγράφων στο διαδίκτυο. Σενάρια που έχουν χρησιμοποιήσει τη javascript ενσωματώνονται στον HTML κώδικα και εκτελούνται από τον περιηγητή του χρήστη (client-side processing). Για να καταφέρει ένας περιηγητής να εκτελέσει ένα σενάριο γραμμένο σε javascript, είναι απαραίτητο να διαθέτει έναν διερμηνέα javascript (interpreter). Γνωστοί περιηγητές όπως οι Chrome, Firefox, Opera και Internet Explorer έχουν ενσωματωμένο το διερμηνέα της javascript.

Μεταξύ των δυνατοτήτων που δίνει η javascript είναι:

1. Έλεγχος στο περιεχόμενο και στην εμφάνιση των ιστοσελίδων
2. Αλληλεπίδραση με το χρήστη με τη βοήθεια γεγονότων
3. Ανάγνωση ή καταγραφή του Η/Υ του χρήστη μέσω cookies
4. Δυνατότητα για εναλλαγή εικόνων
5. Έλεγχος της συμπεριφοράς και των λειτουργιών του browser
6. Αλληλεπίδραση με φόρμες HTML

Οι περιορισμοί που διαθέτει η javascript είναι ότι δεν επιτρέπει την προσπέλαση ή χειρισμό αρχείων στον Η/Υ, για λόγους ασφάλειας. Επίσης τα σενάρια που εκτελούνται στον περιηγητή του χρήστη, δεν μπορούν να ενημερώσουν αρχεία που βρίσκονται στο server. [1]-[11]

## 2.3 PHP

Η PHP είναι γλώσσα προγραμματισμού για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο. Μια PHP σελίδα περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache). Με αυτό τον τρόπο θα παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (\*.php, \*.php4, \*.phtml κ.α.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον περιηγητή τον κώδικα χωρίς καμιά επεξεργασία, εκτός αν έχει γίνει κατάλληλη ρύθμιση στα MIME types του server. Ακόμα και όταν ένα αρχείο έχει επέκταση .php θα πρέπει ο server, να είναι ρυθμισμένος για να επεξεργάζεται τον κώδικα PHP. Ο διακομιστής Apache, που χρησιμοποιείται ευρέως σήμερα, σε συστήματα με τα λειτουργικά συστήματα Linux και Windows, υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP.[1]-[10], [13]

## 2.4 CSS

Η CSS (*Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ*) είναι γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στιλ. Η συγκεκριμένη γλώσσα χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου, που έχει γραφτεί με μια γλώσσα σήμανσης. Πιο συγκεκριμένα χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου σε γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα του υπολογιστή με σκοπό να αναπτύξει στιλιστικά μια ιστοσελίδα. Να διαμορφώσει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση ενώ δίνει πάρα πολλές δυνατότητες σε σχέση με την HTML. Η CSS κρίνεται απαραίτητη για την ύπαρξη μιας στιλιστικά όμορφης και καλοσχεδιασμένης ιστοσελίδας.

Για ένα έγγραφο XHTML θα υπάρχουν παραπάνω από ένα φύλλα στυλ, τα οποία περιέχουν δηλώσεις για την εμφάνιση ενός συγκεκριμένου στοιχείου. Το φύλλο στυλ που εφαρμόζεται σε ένα έγγραφο μπορεί να προέρχεται από:

- i. Το συγγραφέα μιας ιστοσελίδας
- ii. Το χρήστη του πλοηγού
- iii. Τον ίδιο τον πλοηγό, αν έχει το δικό του [[προκαθορισμένο|προκαθορισμένη ρύθμιση (επιστήμη υπολογιστών)] φύλλο στυλ.

Οπότε καταλαβαίνουμε πως για ένα XHTML στοιχείο, θα υπάρχουν παραπάνω από μια δηλώσεις που πολύ πιθανόν θα συγκρούονται. Το πρότυπο CSS σε μια προσπάθεια να επιλύσει παρόμοιες συγκρούσεις, έχει καθορίσει μια αλληλουχία-σειρά με την οποία θα μπουν αυτές οι δηλώσεις και με βάση την οποία θα επιλεγεί πχ. η δήλωση που είναι πρώτη στη σειρά.

Ο αλγόριθμος δημιουργίας αυτής της σειράς-αλληλουχίας είναι ο ακόλουθος:

1. Βρες όλες τις δηλώσεις που εφαρμόζονται στο στοιχείο που μας ενδιαφέρει. Οι δηλώσεις εφαρμόζονται στο στοιχείο αν ο επιλογέας του το επιλέγει (ταιριάζει με αυτό).
2. Ταξινόμησε με βάση τη σημασία (κανονική ή σημαντική) και προέλευση ( συγγραφέας , χρήστη ή πλοηγός χρήστη). Σε αυξανούσα σειρά προτεραιότητας:
  - a. Δηλώσεις πλοηγού χρήστη
  - b. Κανονικές δηλώσεις χρήστη
  - c. Κανονικές δηλώσεις συγγραφέα
  - d. Σημαντικές δηλώσεις συγγραφέα
  - e. Σημαντικές δηλώσεις χρήστη
3. Ταξινόμησε δηλώσεις της ίδιας σημασίας και προέλευσης με κριτήριο την εξειδίκευση του επιλογέα: οι πιο εξειδικευμένοι επιλογείς υπερισχύουν τον πιο γενικών. Τα ψευδό-στοιχεία και οι ψευδό-κλάσεις λογαριάζονται σαν κανονικά στοιχεία και κλάσεις αντίστοιχα.
4. Τέλος ταξινόμησε ανάλογα με σειρά καθορισμού: αν δύο δηλώσεις έχουν το ίδιο βάρος , προέλευση και εξειδίκευση , αυτή που προσδιορίστηκε τελευταία επικρατεί. Οι δηλώσεις σε εισαγόμενα φύλλα στυλ θεωρούνται ότι δηλώνονται πριν από τις δηλώσεις στο ίδιο το φύλλο στυλ.

Με την εμφάνιση μιας σειράς-αλληλουχίας κανόνων, που αφορούν το ίδιο στοιχείο, θα επιλεγεί προς εφαρμογή (για την αποφυγή συγκρούσεων), η δήλωση που θα έχει έρθει τελευταία στην σειρά που αναλύσαμε μόλις. [1]-[11]

## 2.5 XML (eXtensive Markup Language)

### 2.5.1 Τί είναι XML (eXtensive Markup Language)

Η XML είναι markup γλώσσα για έγγραφα στα οποία περιέχεται δομημένη πληροφορία, ενώ αποτελεί σημαντικό στοιχείο του semantic web. Η δομημένη πληροφορία περιέχει περιεχόμενα (εικόνες, λέξεις κτλ) και κάποια στοιχεία για τι ακριβώς αντιπροσωπεύει, και ποιο ρόλο έχει. Τα περισσότερα έγγραφα έχουν κάποια δομή. Μια markup γλώσσα είναι ένας μηχανισμός, ώστε να υποδηλώνεται η δομή ενός εγγράφου. Οι προδιαγραφές της XML ορίζουν ένα συγκεκριμένο τρόπο με τον οποίο μπορεί κάποιος να προσθέτει markup στα έγγραφα. Ένα χαρακτηριστικό παράδειγμα στο οποίο μπορούμε να δούμε μια πληροφορία και το είδος το οποίο αντιπροσωπεύει είναι το εξής: <author>Berners-Lee</author>.

Πλεονέκτημα της XML, είναι το λογισμικό της που μπορεί να αναγνωρίσει συγκεκριμένες ετικέτες και να εφαρμόσει χρήσιμες διαδικασίες, όπως είναι η εξόρυξη βιβλιογραφικής πληροφορίας, αλλά και άλλες ακόμα. Οι σχεδιαστικοί στόχοι της συγκεκριμένης γλώσσας δίνουν σημασία στην απλότητα, τη γενικότητα και την χρήση της στο Internet. Η XML υποστηρίζεται από όλες τις σύγχρονες γλώσσες προγραμματισμού του κόσμου. Στοχεύει στη δόμηση εγγράφων, παρόλα αυτά υπάρχουν πολλές περιπτώσεις στις οποίες χρησιμοποιείται για την παρουσίαση πολύπλοκων δομών δεδομένων όπως για παράδειγμα είναι τα XML Web

Services. Τέλος, πρέπει να αναφερθεί ότι η XML, είναι η βάση για ένα σύνολο άλλων γλωσσών όπως οι: RSS, Atom, SOAP, XHTML και άλλες.[1]-[11]

### 2.5.2 Ορισμός της δομής ενός XML εγγράφου

Το XML είναι έγγραφο με καθορισμένη δομή όπως αναφέραμε στην αρχή της ανάλυσης μας. Για να δώσουμε περισσότερες πληροφορίες για το πώς ορίζεται αυτή η δομή, θα χρειαστεί να αναλύσουμε το XML Schema, με το οποίο περιγράφουμε τη δομή ενός αρχείου XML

Το XML Schema λοιπόν, εκφράζει με ένα κοινό λεξιλόγιο το περιεχόμενο του XML εγγράφου. Πιο ειδικά, παρέχει τα μέσα ώστε να δηλωθεί η δομή, το περιεχόμενο και τα tags ενός XML αρχείου. Αυτή η διαδικασία επικυρώνει τη σύνταξη και τη φόρμα που επιτρέπεται, επίσης διευκολύνει το διαμοιρασμό πληροφοριών μέσω διαδικτύου, σε κάθε τέτοιο έγγραφο. Το XML Schema επιτρέπει την επικύρωση, την ανάγνωση και την επεξεργασία XML εγγράφων, από εφαρμογές λογισμικού. Αυτό παρέχει την βάση για τη σύλληψη, αναπαράσταση, ανταλλαγή και αποθήκευση πληροφοριών στις οποίες μπορούν εύκολα να έχουν πρόσβαση ευφυείς πράκτορες. [1]-[11]

### 2.5.3 Σχεδιαστικοί στόχοι της XML

Οι προδιαγραφές της XML ορίζουν κάποιους στόχους που πρέπει να πετύχει η γλώσσα. Καθορίστηκαν μετά από πρόταση της W3C. Οι στόχοι περιγράφονται παρακάτω:

Πρέπει να γίνεται χρήση της XML στο Internet. Οι χρήστες θα πρέπει να μπορούν να βλέπουν XML έγγραφα τόσο εύκολα, όσο βλέπουν και HTML σελίδες. Πρακτικά, αυτό είναι δυνατό μόνο όταν οι XML περιηγητές γίνουν παντού διαθέσιμοι, όπως και οι HTML περιηγητές. Αυτό μπορεί μεν να είναι δύσκολο, αλλά ο στόχος παραμένει.

Η XML πρέπει να υποστηρίζει μια ποικιλία προγραμμάτων. Να χρησιμοποιείται προς όφελος των χρηστών σε εφαρμογές συγγραφής, περιήγησης και ανάλυσης περιεχομένου κτλ. Παρ' όλο που αρχικά δημιουργήθηκε για τον καθορισμό της δομής των εγγράφων στο διαδίκτυο, δε σημαίνει ότι δεν μπορεί να χρησιμοποιηθεί για ένα πλήθος άλλων διεργασιών.

Η XML θα πρέπει να είναι συμβατή με την SGML. Κατά κύριο λόγο άνθρωποι που έχουν σχέση με τη δημιουργία της γλώσσας, έρχονται από οργανισμούς που έχουν ένα μεγάλο ποσό υλικού σε SGML. Η XML όντως, σχεδιάστηκε ώστε να είναι συμβατή με ήδη υπάρχοντα πρότυπα, τα οποία δίνουν λύση στο σύγχρονο πρόβλημα της αποστολής πλούσιων δομημένων εγγράφων στο διαδίκτυο.

Θα πρέπει να είναι εύκολη η συγγραφή προγραμμάτων τα οποία επεξεργάζονται XML έγγραφα. Αυτό σημαίνει πρακτικά ότι ένας μαθητής της επιστήμης των υπολογιστών μέσα σε 2 εβδομάδες θα μπορεί να δημιουργήσει ένα πρόγραμμα το οποίο θα επεξεργάζεται XML έγγραφα.

Ο αριθμός των προαιρετικών χαρακτηριστικών της XML θα πρέπει να είναι απολύτως μικρός, ιδανικά μηδενικός. Τα προαιρετικά χαρακτηριστικά αυξάνουν τα προβλήματα συμβατότητας, όταν οι χρήστες επιθυμούν να μοιραστούν έγγραφα, οδηγούν σε σύγχυση και απογοήτευση.

Τα XML έγγραφα πρέπει να είναι αναγνώσιμα από ανθρώπους και απολύτως σαφή. Με απλά λόγια, ακόμη και αν δεν υπάρχει διαθέσιμος κάποιος XML περιηγητής, θα πρέπει τα έγγραφα να είναι κατανοητά ως προς τη περιγραφή και το περιεχόμενο, με ένα από πρόγραμμα ανάγνωσης κειμένου.

Ο σχεδιασμός της XML πρέπει να προετοιμαστεί ταχύτατα. Η χρήση της γλώσσας αυτής ήταν αναγκαία και θα έπρεπε να αναπτυχθεί όσο το δυνατόν πιο γρήγορα.

Ο σχεδιασμός της XML θα πρέπει να είναι επίσημος και συνοπτικός. Αυτό πρακτικά σημαίνει ότι η XML θα πρέπει να εκφράζεται σε EBNF και θα πρέπει να είναι η συνοφασμένη η

λογική της με αυτή των μοντέρνων μεταγλωττιστικών εργαλείων και τεχνικών. Μια από τις ανάγκες για τη δημιουργία της XML ήταν και αυτή, καθώς η γραμματική της γλώσσας SGML, λόγω κάποιων τεχνικών προβλημάτων, δε μπορεί να εκφραστεί σε EBNF. Η συγγραφή ενός SGML parser απαιτεί τον χειρισμό μιας πλειάδας δύσκολων και σπάνιων χαρακτηριστικών ενώ η XML όχι.

Η δημιουργία των XML εγγράφων πρέπει να είναι εύκολη. Παρά το ότι μπορεί να υπάρχουν περίπλοκες εφαρμογές επεξεργασίας τέτοιων εγγράφων, η δημιουργία και η επεξεργασία XML περιεχομένου θα πρέπει να γίνεται εύκολα και από έναν απλό επεξεργαστή κειμένου, από Perl scripts κλπ. [1]-[11]

### 2.5.5 XML έννοιες

(Unicode) Character: Είναι γνωστό ότι το έγγραφο XML, από τον ορισμό της γλώσσας, είναι ένα αλφαριθμητικό το οποίο αποτελείται από χαρακτήρες. Οι περισσότεροι Unicode χαρακτήρες μπορεί να εμφανιστούν σε ένα XML έγγραφο.

Processor and Application: Ο επεξεργαστής αναλύει το markup κομμάτι του εγγράφου και παραχωρεί σε μια εφαρμογή τη δομημένη πληροφορία που λαμβάνει από αυτό. Το τι επιτρέπεται να κάνει ένας επεξεργαστής και τι όχι, ορίζεται από τις προδιαγραφές της γλώσσας. Από τη σκοπιά του προγράμματος, δεν υπάρχουν προδιαγραφές για το πως θα διαχειριστούν τα δεδομένα. Ο επεξεργαστής συχνά αναφέρεται και ως XML parser.

Markup and Content: Οι χαρακτήρες οι οποίοι αποτελούν ένα XML έγγραφο είναι διαμοιρασμένοι σε markup και περιεχόμενο. Τα οποία διαχωρίζονται από την εφαρμογή, με κάποιους απλούς συντακτικούς κανόνες. Παραδείγματος χάριν, όσα αλφαριθμητικά ανήκουν στη κατηγορία markup ξεκινάν ή τελειώνουν με τα σύμβολα “<” και “>” αντίστοιχα. Επίσης είναι δυνατό να ξεκινούν με το χαρακτήρα “&” και να τελειώνουν με το χαρακτήρα “;”. Ενώ τα αλφαριθμητικά τα οποία δεν αποτελούν markup είναι το περιεχόμενο.

Ετικέτες (Tags): Ετικέτα ονομάζεται το markup αλφαριθμητικό το οποίο ξεκινά με τον χαρακτήρα “<” και τελειώνει με το χαρακτήρα “>”. Τα tags αποτελούνται από τρεις κατηγορίες. Αρχικά υπάρχουν τα start-tags (ετικέτες αρχής), πχ. <section>, στη συνέχεια έχουμε τα end-tags (ετικέτες τέλους), πχ. </section> και τέλος έχουμε τα empty-element tags, πχ. <line-break/>.

Στοιχείο (Element): Το στοιχείο είναι κάθε λογικό συστατικό του εγγράφου, το οποίο είτε ξεκινά με ένα start-tag και τελειώνει με ένα end-tag, είτε αποτελείται από ένα empty-element tag. Οι χαρακτήρες ανάμεσα από τα start-tags και end-tags, αν υπάρχουν, είναι το περιεχόμενο του στοιχείου, και πιθανώς να περιέχουν markup στοιχεία αλλά και άλλα στοιχεία τα οποία καλούνται elements-παιδιά. Για παράδειγμα το <Greeting>Hello, world.</Greeting>.

Ιδιότητες (Attribute): Ιδιότητα ονομάζεται μια markup δομή η οποία αποτελείται από ένα ζευγάρι ονόματος-τιμής το οποίο βρίσκεται μέσα σε ένα start-tag ή empty-element tag. Για παράδειγμα, παρακάτω το στοιχείο img έχει δύο ιδιότητες,src και alt: <img src =”ceid.jpg” alt=’patras University ’/>. Άλλο ένα παράδειγμα θα ήταν το <step number=”3”>Connect A to B.</step> όπου το όνομα της ιδιότητας είναι το “number” και ο αριθμός είναι το “3”.

XML δήλωση (XML Declaration): Στα XML έγγραφα πρέπει να δηλώνονται από την αρχή κάποιες πληροφορίες, γύρω από αυτά. Για παράδειγμα ποια έκδοση από το πρότυπο XML χρησιμοποιούν και σε τι κωδικοποίηση είναι γραμμένα αυτά τα έγγραφα. [1]-[11]

### 2.5.6 Σύνοψη

Η XML είναι μια markup γλώσσα, η οποία έχει δημιουργηθεί για την ανταλλαγή πληροφοριών στο διαδίκτυο, εύκολα και αποδοτικά. Το μεγάλο πλεονέκτημά της είναι, η ικανότητα προσπέλασης των XML εγγράφων από οποιαδήποτε πλατφόρμα λογισμικού η υλικού. Παράλληλα είναι μια γλώσσα που την κατανοείς εύκολα και σημαντικό μπορεί να βοηθήσει στην προσπέλαση δεδομένων από άτομα με ειδικές ανάγκες. Τέλος, η XML αποτελεί τη βά-



ση δημιουργίας μιας πλειάδας γλωσσών που χρησιμοποιούνται στο Internet και χρησιμοποιείται κατά κόρον στα Web Services.

## 2.6 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System ή RDBMS).

Χρησιμοποιώντας την MySQL είναι εύκολη η πρόσβαση σ' αυτές τις πληροφορίες, κάνοντας χρήση μιας γλώσσας συγγραφής σεναρίων, στην πλευρά του εξυπηρετητή (server-side scripting language), όπως είναι η PHP. Ο κώδικας που θα χρησιμοποιήσουμε μπορεί να εφαρμοσθεί σε έναν εξυπηρετητή που βασίζεται στα Windows ή στα Unix. Η MySQL είναι ένα πολύ γρήγορο και δυνατό σύστημα διαχείρισης βάσεων δεδομένων.

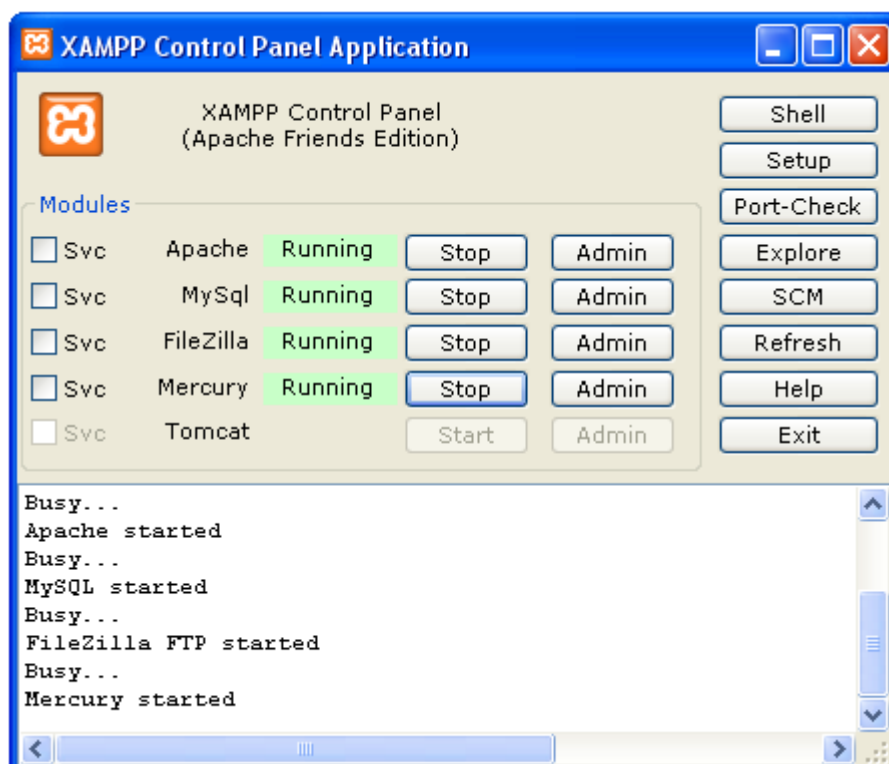
Μια βάση δεδομένων μας δίνει την ευκαιρία να αποθηκεύουμε, να αναζητάμε, να ταξινομούμε και να ανακαλούμε τα δεδομένα αποτελεσματικά. Ο MySQL εξυπηρετητής ελέγχει την πρόσβαση στα δεδομένα. Με αυτό τον τρόπο μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα, ενώ παρέχει γρήγορη πρόσβαση και διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Συνεπώς η MySQL είναι ένας πολυνηματικός εξυπηρετητής πολλαπλών χρηστών. Χρησιμοποιεί SQL (Structured Query Language) την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων, σε όλο τον κόσμο.

Η MySQL είναι πιο κατάλληλη για χρήση στο Internet γιατί:

1. Παρέχει ευκολίες στο backup
2. Είναι ιδιαίτερα βελτιωμένη σε ταχύτητα για την ανάκτηση δεδομένων.
3. Είναι συμβατή και μεταφέρσιμη σε διάφορες πλατφόρμες και για διάφορα εργαλεία ανάπτυξης.
4. Είναι οικονομική
5. Είναι λογισμικό ανοιχτού κώδικα.[1]-[11]

## 2.7 Hosting Πλατφόρμα - Xampp

Εκτός από τις παραπάνω τεχνολογίες χρησιμοποιήθηκε και το πρόγραμμα Xampp.



### Εικόνα 1. «Εργαλείο εκτέλεσης δυναμικών ιστοσελίδων»

Για την ανάπτυξη και διανομή διαδικτυακών εφαρμογών είναι απαραίτητο κάποιο εργαλείο, το οποίο να υποστηρίζει την εκτέλεση δυναμικών ιστοσελίδων. Δηλαδή, να διαθέτει ένα web server που να μπορεί να εκτελέσει server-side scripts καθώς και να υποστηρίζει τη χρήση βάσεων δεδομένων.

## ΚΕΦΑΛΑΙΟ 3: Επιχειρησιακό Σύστημα - Εταιρία Κινητής Τηλεφωνίας

### 3.1 Πληροφοριακό Σύστημα για εταιρία Κινητής Τηλεφωνίας

Πρέπει να αποθηκεύουμε πληροφορίες για μια εταιρία κινητής τηλεφωνίας. Για κάθε υποκατάστημα της εταιρίας, το όνομα και τη διεύθυνσή της.

Για κάθε υπάλληλο της εταιρίας θέλουμε να αποθηκεύουμε το όνομα, το επώνυμο, τον αριθμό ταυτότητας, το μισθό, τα προστατευόμενα μέλη της οικογένειας τους (ονοματεπώνυμο, φύλο, και ηλικία). Οι υπάλληλοι χωρίζονται σε δύο κατηγορίες α) σε διοικητικούς για τους οποίους θέλουμε να αποθηκεύουμε πληροφορία για το πτυχίο τους (τίτλο και πανεπιστήμιο που το έδωσε και ημερομηνία κτήσης) β) σε τεχνικούς για τους οποίους θέλουμε να αποθηκεύουμε πληροφορία για την ειδικότητα και τα έτη προϋπηρεσίας.

Για τις κεραιές που έχει η εταιρία. Θέλουμε το όνομα, την διεύθυνση και τον τύπο της κεραιάς.

Για κάθε πελάτη το όνομα, την διεύθυνση, τον αριθμό ταυτότητας, το επάγγελμα.

Για κάθε αριθμό τηλεφώνου που έχει η εταιρία θέλουμε να αποθηκεύουμε τον αριθμό κλήσης, τον κωδικό αριθμό (ruk1 και 2) καθώς και αν είναι σύνδεση ή καρτοκινητό.

Θέλουμε να αποθηκεύουμε πληροφορία για τα διάφορα πακέτα χρέωσης που υπάρχουν. Το όνομα, τα δωρεάν λεπτά ομιλίας και τα μηνύματα, το πάγιο και την τιμή χρέωσης για τον παραπάνω χρόνο ομιλίας και το ίδιο για τα μηνύματα.

Για κάθε βλάβη/επιδιόρθωση που λαμβάνει χώρα στο δίκτυο της εταιρίας. Πότε έγινε, σε ποια κεραία, τι είδους ήταν και ποιοι υπάλληλοι την επιδιορθώσανε.

Για τους λογαριασμούς πρέπει να αποθηκεύονται το ποσό χρέωσης, η ημερομηνία έκδοσης, η ημερομηνία πληρωμής και σε ποιον αριθμό και πελάτη αντιστοιχεί.

Επίσης θέλουμε να αποθηκεύουμε τις εξής πληροφορίες:

- Ένας πελάτης ανήκει σε ένα και μόνο ένα υποκατάστημα.
- Ένα υποκατάστημα έχει τουλάχιστον δύο υπαλλήλους.
- Ένα υποκατάστημα έχει ένα και μόνο ένα υπάλληλο ως διευθυντή και ένας υπάλληλος είναι το πολύ σε ένα υποκατάστημα διευθυντής.
- Ένας αριθμός συνδέεται με ένα το πολύ πελάτη αλλά ένας πελάτης μπορεί να έχει πολλούς αριθμούς.
- Ένας υπάλληλος μπορεί να επιδιορθώνει πολλές κεραιές και μια κεραία μπορεί να επιδιορθώνεται από πολλούς υπαλλήλους.
- Μια κεραία «ανήκει» σε ένα και μόνο ένα υποκατάστημα και ένα υποκατάστημα μπορεί να έχει πολλές κεραιές.
- Ένας πελάτης που έχει σύνδεση, έχει υπογράψει ένα και μόνο ένα συμβόλαιο με κάποιον υπάλληλο της εταιρίας και αυτό θα ανήκει σε ένα από τα πακέτα σύνδεσης.

### 3.2 Οι διαδικασίες που θα υποστηρίζει το σύστημα

Οι υποστηριζόμενες από το σύστημα διαδικασίες είναι:

1. Εισαγωγή και διαγραφή από τις πιο πάνω κατηγορίες.
2. Interface για την εισαγωγή του χρόνου ομιλίας και των μηνυμάτων που έγινε με την χρήση κάποιου αριθμού οπότε μετά θα υπολογίζεται το ποσόν χρέωσης ως  $\text{Ποσόν} = \text{πάγιο} + [(\text{λεπτά\_ομιλίας} - \text{δωρεάν\_λεπτά}) / 60] * \text{extra} + [\text{αριθμός\_μηνυμάτων} - \text{αριθμός\_δωρεάν\_μηνυμάτων}] * \text{χρέωση\_μηνύματος}$ .
3. Αυτόματη μείωση του παγίου κατά 10% όταν συμπληρωθεί ένας χρόνος σύνδεσης.

4. Για κάθε ανανέωση συμβολαίου θα υπολογίζεται το ποσόν της επιδότησης και θα το εμφανίζει στην οθόνη. Το ποσό της επιδότησης προκύπτει ως εξής Ποσό\_επιδότησης = άθροισμα\_λογαριασμών\_τελευταίου\_χρόνου\*0,12.

5. Συγκεντρωτική αναφορά για τα νέα συμβόλαια και καρτοκινητά που κάνει κάθε υποκατάστημα σε ένα χρονικό διάστημα.

6. Συγκεντρωτική αναφορά για τις εισπράξεις κάθε καταστήματος σε ένα χρονικό διάστημα.

7. Συγκεντρωτική αναφορά για τις ανανεώσεις κάθε καταστήματος σε ένα χρονικό διάστημα.

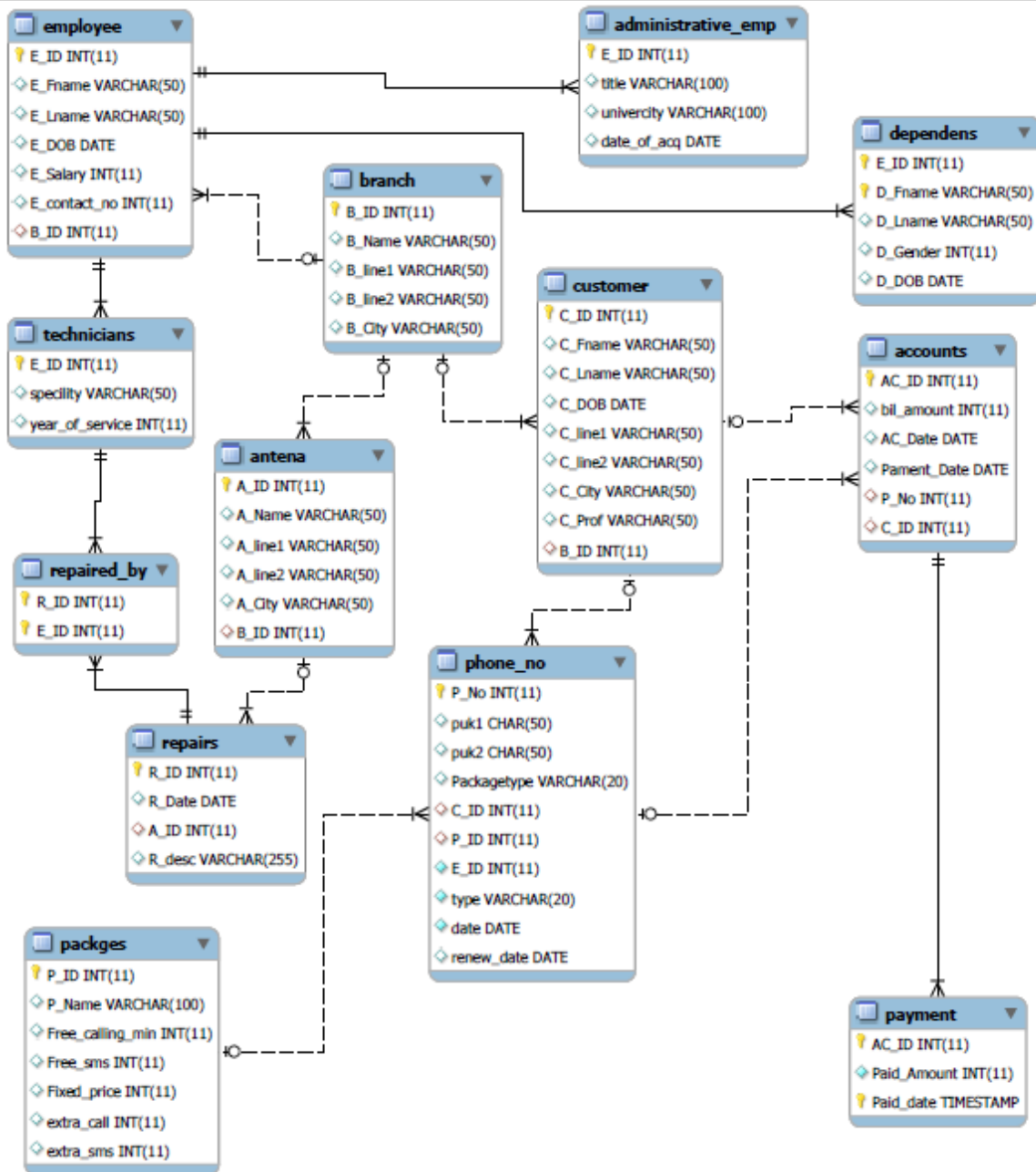
8. Συγκεντρωτική αναφορά για κάθε υπάλληλο όσον αφορά

α) τα νέα συμβόλαια που κάνει και

β) τις ανανεώσεις

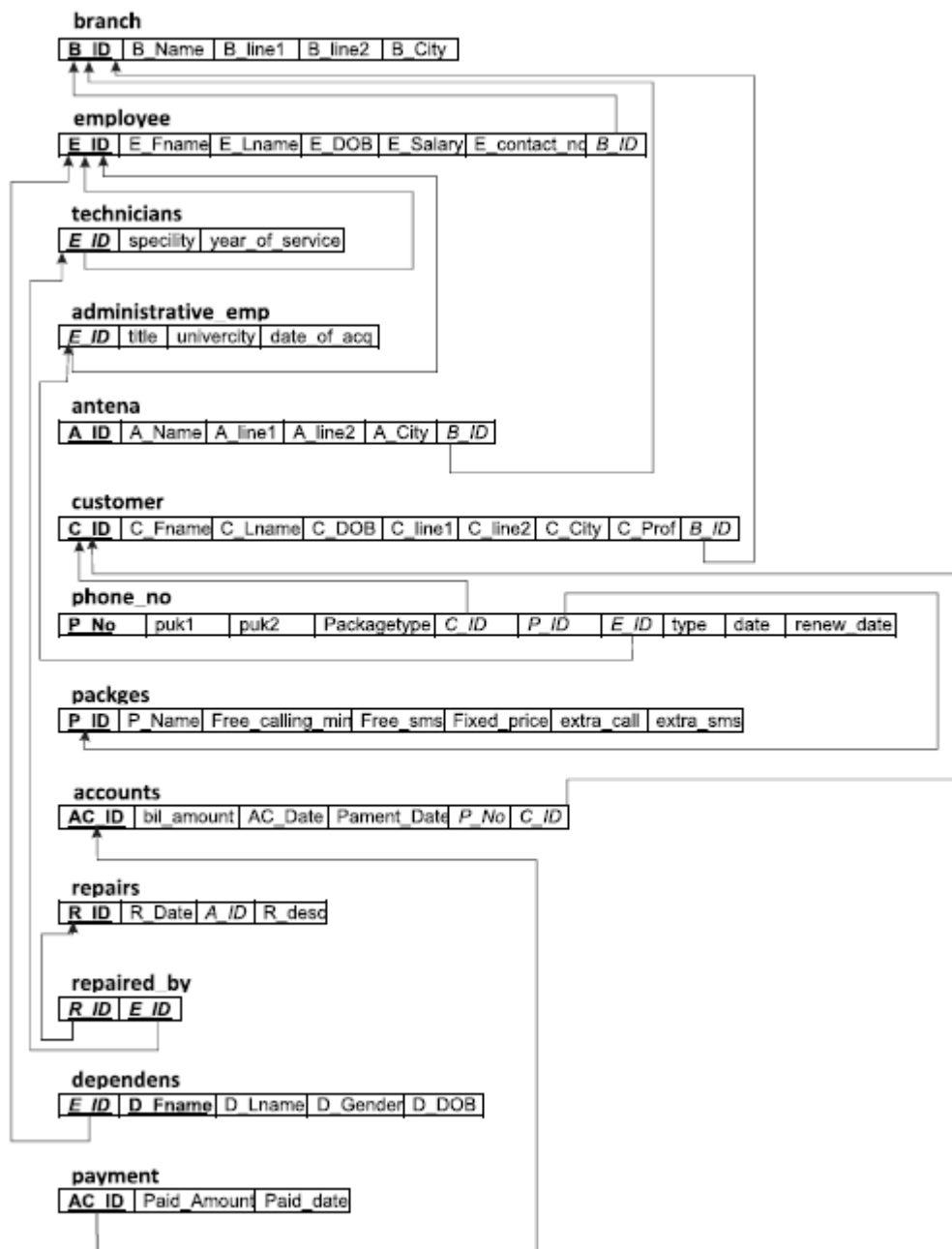
# ΚΕΦΑΛΑΙΟ 4: Μοντελοποίηση του Πληροφοριακού Συστήματος

## 4.1 Διάγραμμα οντοτήτων-σχέσεων.



Εικόνα 2. «Μοντέλο σύνδεσης οντοτήτων»

## 4.2 Μετάφραση του μοντέλου στο σχεσιακό μοντέλο



Εικόνα 3. « Μοντέλο σύνδεσης οντοτήτων»

## 4.3 Περιορισμοί ακεραιότητας και συναρτησιακές εξαρτήσεις

Η λογική ακεραιότητα της βάσης δίνεται στο ER model, ενώ οι συναρτησιακές εξαρτήσεις, βάση αυτού, προκύπτουν ως εξής:

1. E\_ID INT -> E\_Fname E\_Lname E\_DOB E\_Salary E\_contact\_no B\_ID
2. E\_ID -> title university date\_of\_acq
3. E\_ID D\_Fname -> D\_Lname D\_Gender D\_DOB
4. B\_ID -> B\_Name B\_line1 B\_line2 B\_City
5. A\_ID -> A\_Name A\_line1 A\_line2 A\_City B\_ID
6. C\_ID -> C\_Fname C\_Lname C\_DOB C\_line1 C\_line2 C\_City C\_Prof B\_ID
7. AC\_ID -> bil\_amount AC\_Date Pament\_Date P\_No C\_ID

- 8. AC\_ID -> Paid\_Amount Paid\_date
- 9. P\_No -> puk1 puk2 Packagetype C\_ID P\_ID E\_ID type date renew\_date
- 10. P\_ID -> P\_Name Free\_calling\_min Free\_sms Fixed\_price extra\_call extra\_sms
- 11. E\_ID -> specility year\_of\_service
- 12. R\_ID E\_ID
- 13. R\_ID -> R\_Date A\_ID R\_desc

#### 4.4 Μετατροπή του μοντέλου σε τρίτη κανονική μορφή με διατήρηση των συναρτησιακών εξαρτήσεων και χωρίς απώλεια πληροφορίας

Η Τρίτη κανονική μορφή του μοντέλου (από σχήμα ΟΣ και σε 3NF) δίνεται από τον παρακάτω πίνακα:

employee	E_ID	E_Fnam	E_Lna	E_DOB	E_Sala	E_contact_no			
administrative_emp	E_ID	title	university	date_of_acq					
dependens	E_ID	D_Fname	D_Lnam	D_Gende	D_DOB				
branch	B_ID	B_Name	B_line1	B_line2	B_City				
antenna	A_ID	A_Name	A_line	A_line2	A_City	B_ID			
customer	C_ID	C_Fname	C_Lname	C_DOB	C_line1	C_line2	C_City	C_Prof	B_ID
accounts	AC_ID	bil_amount	AC_Date	Payment_Date	P_No	C_ID			
payment	AC_ID	Paid_Amount	Paid_date						
packages	P_ID	P_Name	Free_calling_min	Free_sms	Fixed_price	extra_call	extra_sms		
technicians	E_ID	specility	year_of_service						
repaired_by	R_ID	E_ID							
repairs	R_ID	R_Date	A_ID	R_desc					

Εικόνα 4. «Μοντέλο σύνδεσης οντοτήτων»

## ΚΕΦΑΛΑΙΟ 5: DDL εντολές για την δημιουργία του Data Model

```
CREATE TABLE `accounts` (  
  `AC_ID` int(11) NOT NULL,  
  `bil_amount` int(11) DEFAULT NULL,  
  `AC_Date` date DEFAULT NULL,  
  `Pament_Date` date DEFAULT NULL,  
  `P_No` int(11) DEFAULT NULL,  
  `C_ID` int(11) DEFAULT NULL,  
  PRIMARY KEY (`AC_ID`),  
  KEY `C_ID` (`C_ID`),  
  KEY `P_No` (`P_No`)  
);  
--  
-- Dumping data for table `accounts`  
--  
INSERT INTO `accounts` (`AC_ID`, `bil_amount`, `AC_Date`, `Pament_Date`, `P_No`,  
`C_ID`) VALUES  
(1, 20000, '2013-02-02', '2013-02-06', 2147483647, 1111),  
(3, 20000, '2013-02-02', '2013-02-06', 1234567890, 2),  
(5, 5000, '2013-02-02', '2013-02-06', 1234567897, 3);  
-----  
--  
-- Table structure for table `administrative_emp`  
--  
CREATE TABLE `administrative_emp` (  
  `E_ID` int(11) NOT NULL,  
  `title` varchar(100) DEFAULT NULL,  
  `univercity` varchar(100) DEFAULT NULL,  
  `date_of_acq` date DEFAULT NULL,  
  PRIMARY KEY (`E_ID`)  
);  
--  
-- Dumping data for table `administrative_emp`  
--  
INSERT INTO `administrative_emp` (`E_ID`, `title`, `univercity`, `date_of_acq`) VAL-  
UES  
(5, 'eeeeeee', 'www', '0000-00-00');  
-----  
--  
-- Table structure for table `antena`  
--  
CREATE TABLE `antena` (  
  `A_ID` int(11) NOT NULL,  
  `A_Name` varchar(50) DEFAULT NULL,  
  `A_line1` varchar(50) DEFAULT NULL,  
  `A_line2` varchar(50) DEFAULT NULL,  
  `A_City` varchar(50) DEFAULT NULL,  
  `B_ID` int(11) DEFAULT NULL,  
  PRIMARY KEY (`A_ID`),  
  KEY `B_ID` (`B_ID`)  
);  
--
```



```

-- Dumping data for table `antena`
--
INSERT INTO `antena` (`A_ID`, `A_Name`, `A_line1`, `A_line2`, `A_City`, `B_ID`)
VALUES
(1, 'dddd', 'sd', 'saa', 'xxx', 1);
-----
--
-- Table structure for table `branch`
--
CREATE TABLE `branch` (
  `B_ID` int(11) NOT NULL,
  `B_Name` varchar(50) DEFAULT NULL,
  `B_line1` varchar(50) DEFAULT NULL,
  `B_line2` varchar(50) DEFAULT NULL,
  `B_City` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`B_ID`)
);

--
-- Dumping data for table `branch`
--

INSERT INTO `branch` (`B_ID`, `B_Name`, `B_line1`, `B_line2`, `B_City`) VALUES
(1, 'uuu', 'uu', 'u', 'j'),
(2, 'hhh', 'bbb', 'gg', 'ff');
-----
--
-- Table structure for table `customer`
--
CREATE TABLE `customer` (
  `C_ID` int(11) NOT NULL,
  `C_Fname` varchar(50) DEFAULT NULL,
  `C_Lname` varchar(50) DEFAULT NULL,
  `C_DOB` date DEFAULT NULL,
  `C_line1` varchar(50) DEFAULT NULL,
  `C_line2` varchar(50) DEFAULT NULL,
  `C_City` varchar(50) DEFAULT NULL,
  `C_Prof` varchar(50) DEFAULT NULL,
  `B_ID` int(11) DEFAULT NULL,
  PRIMARY KEY (`C_ID`),
  KEY `B_ID` (`B_ID`)
);

--
-- Dumping data for table `customer`
--

INSERT INTO `customer` (`C_ID`, `C_Fname`, `C_Lname`, `C_DOB`, `C_line1`,
`C_line2`, `C_City`, `C_Prof`, `B_ID`) VALUES
(2, 'ddddddddddd', 'ddddddd', '2014-02-25', 'wqwqwq', 'dddd', 'ssss', 'ccc', 1),
(3, 'vvvvvvvvvvvvvv', 'ssssssssssss', '2014-02-25', 'wqwqwq', 'saa', 'ssss', 'ccc', 2),
(1111, 'www', 'ww', '0000-00-00', 'rer', 'er', 'ff', 'ccc', 1);
-----
--
-- Table structure for table `dependens`
--
CREATE TABLE `dependens` (

```

```

`E_ID` int(11) NOT NULL DEFAULT '0',
`D_Fname` varchar(50) NOT NULL DEFAULT "",
`D_Lname` varchar(50) DEFAULT NULL,
`D_Gender` int(11) DEFAULT NULL,
`D_DOB` date DEFAULT NULL,
PRIMARY KEY (`E_ID`,`D_Fname`)
);
--
-- Dumping data for table `dependens`
--
INSERT INTO `dependens` (`E_ID`, `D_Fname`, `D_Lname`, `D_Gender`, `D_DOB`)
VALUES
(1, 'dddsd', 'dddddddd', 0, '2014-02-25'),
(1, 'vvvvvvvvvvvvvvvv', 'dddddddd', 1, '2014-02-25');
-----
--
-- Table structure for table `employee`
--
CREATE TABLE `employee` (
  `E_ID` int(11) NOT NULL,
  `E_Fname` varchar(50) DEFAULT NULL,
  `E_Lname` varchar(50) DEFAULT NULL,
  `E_DOB` date DEFAULT NULL,
  `E_Salary` int(11) DEFAULT NULL,
  `E_contact_no` int(11) DEFAULT NULL,
  `B_ID` int(11) DEFAULT NULL,
  PRIMARY KEY (`E_ID`),
  KEY `B_ID` (`B_ID`)
);
--
-- Dumping data for table `employee`
--
INSERT INTO `employee` (`E_ID`, `E_Fname`, `E_Lname`, `E_DOB`, `E_Salary`,
`E_contact_no`, `B_ID`) VALUES
/*examples*/
(0, 'www', 'ww', '0000-00-00', 0, 0, 1),
(1, 'ssssss', 'sssss', '0000-00-00', 1000000, 1234567890, 1),
(2, 'ddddddddddd', 'ddddddd', '2014-02-25', 2333, 1234567890, 1),
(5, 'ddddddddddd', 'ddddddd', '2014-02-25', 1000000, 1234567890, 1);
-----
--
-- Table structure for table `packges`
--
CREATE TABLE `packges` (
  `P_ID` int(11) NOT NULL,
  `P_Name` varchar(100) DEFAULT NULL,
  `Free_calling_min` int(11) DEFAULT NULL,
  `Free_sms` int(11) DEFAULT NULL,
  `Fixed_price` int(11) DEFAULT NULL,
  `extra_call` int(11) DEFAULT NULL,
  `extra_sms` int(11) DEFAULT NULL,
  PRIMARY KEY (`P_ID`)
);
--
-- Dumping data for table `packges`

```

```

--
INSERT INTO `packges` (`P_ID`, `P_Name`, `Free_calling_min`, `Free_sms`,
`Fixed_price`, `extra_call`, `extra_sms`) VALUES
(1, 'ddd', 12, 3, 34, 2, 1);
-----
--
-- Table structure for table `payment`
--
CREATE TABLE `payment` (
  `AC_ID` int(11) NOT NULL,
  `Paid_Amount` int(11) NOT NULL,
  `Paid_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`AC_ID`, `Paid_date`)
);
--
-- Dumping data for table `payment`
--
INSERT INTO `payment` (`AC_ID`, `Paid_Amount`, `Paid_date`) VALUES
(1, 50, '2014-02-16 17:46:40'),
(1, 50, '2014-03-16 17:46:20'),
(1, 50, '2014-04-16 16:46:38'),
(1, 50, '2014-04-16 17:46:42'),
(1, 50, '2014-04-16 17:46:43'),
(1, 50, '2014-04-16 17:47:44'),
(1, 50, '2014-04-16 17:48:12'),
(3, 500, '2014-04-17 02:03:05'),
(5, 2100, '2014-04-17 02:01:19');
-----
--
-- Table structure for table `phone_no`
--
CREATE TABLE `phone_no` (
  `P_No` int(11) NOT NULL,
  `puk1` char(50) DEFAULT NULL,
  `puk2` char(50) DEFAULT NULL,
  `Packagetype` varchar(20) DEFAULT NULL,
  `C_ID` int(11) DEFAULT NULL,
  `P_ID` int(11) DEFAULT NULL,
  `E_ID` int(11) NOT NULL,
  `type` varchar(20) NOT NULL,
  `date` date NOT NULL,
  `renew_date` date DEFAULT NULL,
  PRIMARY KEY (`P_No`),
  KEY `C_ID` (`C_ID`),
  KEY `P_ID` (`P_ID`)
);
--
-- Dumping data for table `phone_no`
--
INSERT INTO `phone_no` (`P_No`, `puk1`, `puk2`, `Packagetype`, `C_ID`, `P_ID`,
`E_ID`, `type`, `date`, `renew_date`) VALUES
(1234567890, '121221', '2212', 'Contract', 1111, 1, 5, 'New', '2014-04-16', NULL),
(1234567897, '121221', '2212', 'Prepaid', 3, 1, 5, 'Renew', '2014-04-17', '2014-03-04'),
(1234567899, '121221', '2212', 'Prepaid', 1111, 1, 5, 'Renew', '2014-04-16', '2014-05-06'),
(2147483647, '121221', '2212', 'Contract', 1111, 1, 5, 'New', '2013-02-02', NULL);

```

```

-----
--
-- Table structure for table `repaired_by`
--
CREATE TABLE `repaired_by` (
  `R_ID` int(11) NOT NULL DEFAULT '0',
  `E_ID` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`R_ID`,`E_ID`),
  KEY `E_ID` (`E_ID`)
);
--
-- Dumping data for table `repaired_by`
--
INSERT INTO `repaired_by` (`R_ID`,`E_ID`) VALUES
(1, 2);
-----
--
-- Table structure for table `repairs`
--
CREATE TABLE `repairs` (
  `R_ID` int(11) NOT NULL,
  `R_Date` date DEFAULT NULL,
  `A_ID` int(11) DEFAULT NULL,
  `R_desc` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`R_ID`),
  KEY `A_ID` (`A_ID`)
);
--
-- Dumping data for table `repairs`
--
INSERT INTO `repairs` (`R_ID`,`R_Date`,`A_ID`,`R_desc`) VALUES
(0, '0000-00-00', 1, ''),
(1, '2014-03-15', 1, ' ');
-----
--
-- Table structure for table `technicians`
--
CREATE TABLE `technicians` (
  `E_ID` int(11) NOT NULL,
  `specility` varchar(50) DEFAULT NULL,
  `year_of_service` int(11) DEFAULT NULL,
  PRIMARY KEY (`E_ID`)
);
--
-- Dumping data for table `technicians`
--
INSERT INTO `technicians` (`E_ID`,`specility`,`year_of_service`) VALUES
(2, 'sss', 4);
--
-- Constraints for dumped tables
--
--
-- Constraints for table `accounts`

```

```

--
ALTER TABLE `accounts`
  ADD CONSTRAINT `accounts_ibfk_1` FOREIGN KEY (`C_ID`) REFERENCES `cus-
tomer` (`C_ID`),
  ADD CONSTRAINT `accounts_ibfk_2` FOREIGN KEY (`P_No`) REFERENCES
`phone_no` (`P_No`);
--
-- Constraints for table `administrative_emp`
--
ALTER TABLE `administrative_emp`
  ADD CONSTRAINT `administrative_emp_ibfk_1` FOREIGN KEY (`E_ID`) REFER-
ENCES `employee` (`E_ID`);
--
-- Constraints for table `antena`
--
ALTER TABLE `antena`
  ADD CONSTRAINT `antena_ibfk_1` FOREIGN KEY (`B_ID`) REFERENCES
`branch` (`B_ID`);
--
-- Constraints for table `customer`
--
ALTER TABLE `customer`
  ADD CONSTRAINT `customer_ibfk_1` FOREIGN KEY (`B_ID`) REFERENCES
`branch` (`B_ID`);
--
-- Constraints for table `dependens`
--
ALTER TABLE `dependens`
  ADD CONSTRAINT `dependens_ibfk_1` FOREIGN KEY (`E_ID`) REFERENCES
`employee` (`E_ID`);
--
-- Constraints for table `employee`
--
ALTER TABLE `employee`
  ADD CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`B_ID`) REFERENCES
`branch` (`B_ID`);
--
-- Constraints for table `payment`
--
ALTER TABLE `payment`
  ADD CONSTRAINT `payment_ibfk_1` FOREIGN KEY (`AC_ID`) REFERENCES
`accounts` (`AC_ID`);
--
-- Constraints for table `phone_no`
--
ALTER TABLE `phone_no`
  ADD CONSTRAINT `phone_no_ibfk_1` FOREIGN KEY (`C_ID`) REFERENCES
`customer` (`C_ID`),
  ADD CONSTRAINT `phone_no_ibfk_2` FOREIGN KEY (`P_ID`) REFERENCES
`packges` (`P_ID`);
--
-- Constraints for table `repaired_by`
--
ALTER TABLE `repaired_by`

```

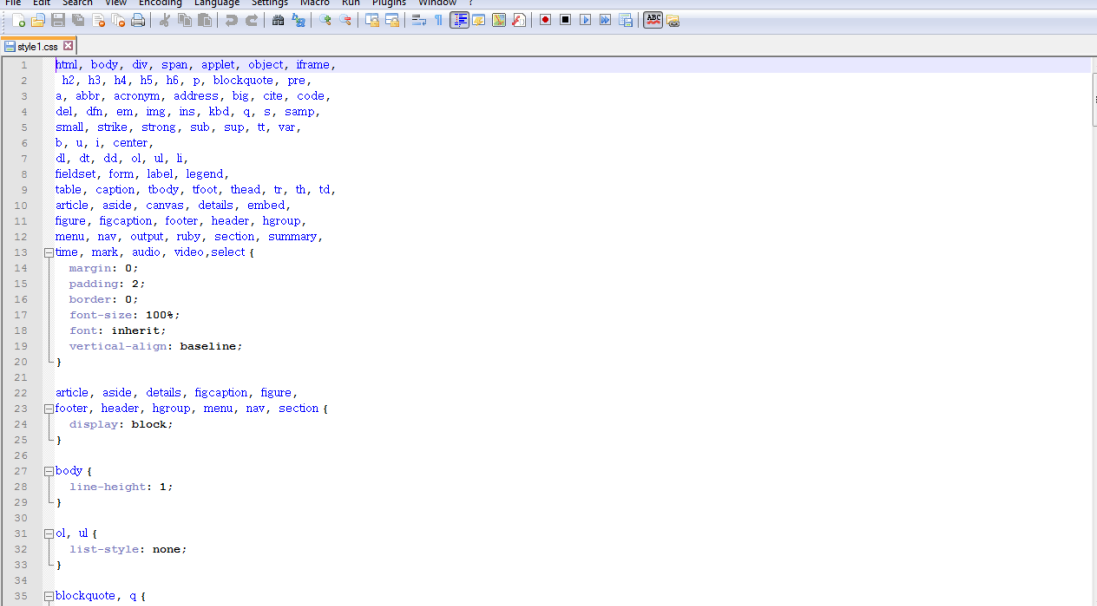
```
    ADD CONSTRAINT `repaired_by_ibfk_1` FOREIGN KEY (`R_ID`) REFERENCES
`repairs` (`R_ID`),
    ADD CONSTRAINT `repaired_by_ibfk_2` FOREIGN KEY (`E_ID`) REFERENCES
`technicians` (`E_ID`);
--
-- Constraints for table `repairs`
--
ALTER TABLE `repairs`
    ADD CONSTRAINT `repairs_ibfk_1` FOREIGN KEY (`A_ID`) REFERENCES `ante-
na` (`A_ID`);
--
-- Constraints for table `technicians`
--
ALTER TABLE `technicians`
    ADD CONSTRAINT `technicians_ibfk_1` FOREIGN KEY (`E_ID`) REFERENCES
`employee` (`E_ID`);
```

## ΚΕΦΑΛΑΙΟ 6: Τεχνολογίες για την υλοποίηση των διαδικασιών

### 6.1 Για την υλοποίηση του συστήματος χρησιμοποιήθηκαν

- SQL, (Structured Query Language), μία γλώσσα υπολογιστών στις βάσεις δεδομένων, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων.
- PHP, μια γλώσσα προγραμματισμού για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο.
- Η CSS (*Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ*) είναι γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ. Η συγκεκριμένη γλώσσα χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου, που έχει γραφτεί με μια γλώσσα σήμανσης. Πιο συγκεκριμένα χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου σε γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστότοπου. Η CSS είναι μια γλώσσα του υπολογιστή με σκοπό να αναπτύξει στιλιστικά μια ιστοσελίδα. Να διαμορφώσει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση ενώ δίνει πάρα πολλές δυνατότητες σε σχέση με την HTML. Η CSS κρίνεται απαραίτητη για την ύπαρξη μιας στιλιστικά όμορφης και καλοσχεδιασμένης ιστοσελίδας.
- HTML, (ακρωνύμιο του αγγλικού HyperText Markup Language ή αλλιώς γλώσσα σήμανσης υπερκειμένου). Είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

### 6.2 CSS Code (Βασικό Interface)



```
1 html, body, div, span, applet, object, iframe,
2 h2, h3, h4, h5, h6, p, blockquote, pre,
3 a, abbr, acronym, address, big, cite, code,
4 del, dfn, em, img, ins, kbd, q, s, samp,
5 small, strike, strong, sub, sup, tt, var,
6 b, u, i, center,
7 dl, dt, dd, ol, ul, li,
8 fieldset, form, label, legend,
9 table, caption, tbody, tfoot, thead, tr, th, td,
10 article, aside, canvas, details, embed,
11 figure, figcaption, footer, header, hgroup,
12 menu, nav, output, ruby, section, summary,
13 time, mark, audio, video, select {
14     margin: 0;
15     padding: 2;
16     border: 0;
17     font-size: 100%;
18     font: inherit;
19     vertical-align: baseline;
20 }
21
22 article, aside, details, figcaption, figure,
23 footer, header, hgroup, menu, nav, section {
24     display: block;
25 }
26
27 body {
28     line-height: 1;
29 }
30
31 ol, ul {
32     list-style: none;
33 }
34
35 blockquote, q {
```

Εικόνα 5. « Χρήση CSS Code σε HTML αρχείο»

```

295 .btn:after {
296   content: '';
297   position: absolute;
298   top: 15px;
299   left: 12px;
300   width: 25px;
301   height: 19px;
302   background: url("../img/arrow.png") 0 0 no-repeat;
303 }
304
305 ::-moz-focus-inner {
306   border: 0;
307   padding: 0;
308 }
309
310 .lt-ie9 input[type=text], .lt-ie9 input[type=password] {
311   line-height: 40px;
312   background: #282828;
313 }
314
315 .lt-ie9 .login-submit {
316   position: absolute;
317   top: 12px;
318   right: -28px;
319   padding: 4px;
320 }
321
322 .lt-ie9 .login-submit:before, .lt-ie9 .login-submit:after {
323   display: none;
324 }
325
326 .lt-ie9 .login-button {
327   line-height: 48px;
328 }
329
330 .lt-ie9 .about {
331   background: #313131;
332 }
333

```

Cascade Style Sheets File length: 7499 lines: 329 Ln: 6 Col: 1 Sel: 0 | 0 UNIX ANSI as UTF-8 INS

Εικόνα 6. «Χρήση CSS Code σε HTML αρχείο»

## 6.3 PHP

### 6.3.1 PHP (Φόρμα καταχώρησης)

```

1 <?php
2 require_once('config.php');
3
4 $rDB1 = mysql_select_db($database_test) or die("Could not select database");
5 if(isset($_POST['submit']))
6 {
7
8   $v1=$_POST['add'];
9   $v2=$_POST['aid'];
10  $v3=$_POST['last'];
11  $v4=$_POST['last'];
12  $v5=$_POST['last'];
13  $v6=$_POST['last'];
14
15  $sql = "INSERT INTO accounts VALUES ('$v3','$v4','$v5','$v6','$v2','$v1')";
16  //echo $sql;
17  $result1 = mysql_query($sql) or die("Could not Updated to po");
18
19
20 }
21
22
23 <!doctype html>
24 <html lang="en">
25 <!--[if lt IE 7]> <html class="lt-ie9 lt-ie8 lt-ie7" lang="en"> <![endif]-->
26 <!--[if IE 7]> <html class="lt-ie9 lt-ie8" lang="en"> <![endif]-->
27 <!--[if IE 8]> <html class="lt-ie9" lang="en"> <![endif]-->
28 <!--[if gt IE 8]><!--> <html lang="en"> <!--<![endif]-->
29 </html>
30
31 <head>
32 <meta charset="utf-8">
33 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
34 <title></title>
35 <link rel="stylesheet" href="css/style1.css">
36 <link href="load.css" rel="stylesheet" type="text/css" >
37 <script src="load.js"></script>
38

```

PHP Hypertext Preprocessor file length: 2735 lines: 105 Ln: 1 Col: 1 Sel: 0 | 0 UNIX ANSI as UTF-8 INS

Εικόνα 7. «PHP φόρμα καταχώρησης λογαριασμού»



```
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
add_antena.php
20 }
21 <?>
22 <!doctype html>
23 <html lang="en">
24 <!--[if lt IE 7]> <html class="lt-ie9 lt-ie8 lt-ie7" lang="en"> <![endif]-->
25 <!--[if IE 7]> <html class="lt-ie9 lt-ie8" lang="en"> <![endif]-->
26 <!--[if IE 8]> <html class="lt-ie9" lang="en"> <![endif]-->
27 <!--[if gt IE 8]><!--> <html lang="en"> <!--<![endif]-->
28 <head>
29 <meta charset="utf-8">
30 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
31 <title></title>
32 <link rel="stylesheet" href="css/style1.css">
33 <link href="load.css" rel="stylesheet" type="text/css" >
34 <script src="load.js"></script>
35
36 </script>
37 <!--[if lt IE 9]><script src="//html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
38 </head>
39 <body>
40 <form method="post" action="add_antena.php" class="login">
41 <fieldset>
42 <legend>Add New Branch</legend>
43
44 <p>
45 <label for="login">Branch ID:</label>
46 <select name="b_id" id="login">
47 <?php
48
49 $sql2="SELECT B_ID, B_Name FROM branch";
50 $result3 = mysql_query($sql2) or die("Could not connect to branch");
51 while($row = mysql_fetch_array($result3))
52 {
53     echo<option value="'. @$row['B_ID'] .'". @$row['B_Name'] ."/option">;
54 }
55 </select>
56 </p>
57 </body>
58 </html>
59 </?>
```

PHP Hypertext Preprocessor file length: 2514 lines: 99 Ln: 1 Col: 1 Sel: 0|0 UNIX ANSI as UTF-8 INS

Εικόνα 8. «PHP φόρμα καταχώρησης κεραίας»

```
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
add_dependent.php
1 <?php
2 require_once('config.php');
3
4 $rdb1 = mysql_select_db($database_test) or die("Could not select database");
5 if(isset($_POST['submit']))
6 {
7
8 $v1=$_POST['cid'];
9 $v2=$_POST['pid'];
10 $v3=$_POST['aid'];
11 $v4=$_POST['ba'];
12 $v5=$_POST['ca'];
13 $v6=$_POST['pd'];
14
15 $sql = "INSERT INTO accounts VALUES ('$v3','$v4','$v5','$v6','$v2','$v1')";
16 //echo $sql;
17 $result1 = mysql_query($sql) or die("Could not Updated to po");
18
19 }
20 <?>
21 <?>
22 <!doctype html>
23 <html lang="en">
24 <!--[if lt IE 7]> <html class="lt-ie9 lt-ie8 lt-ie7" lang="en"> <![endif]-->
25 <!--[if IE 7]> <html class="lt-ie9 lt-ie8" lang="en"> <![endif]-->
26 <!--[if IE 8]> <html class="lt-ie9" lang="en"> <![endif]-->
27 <!--[if gt IE 8]><!--> <html lang="en"> <!--<![endif]-->
28 <head>
29 <meta charset="utf-8">
30 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
31 <title></title>
32 <link rel="stylesheet" href="css/style1.css">
33 <link href="load.css" rel="stylesheet" type="text/css" >
34 <script src="load.js"></script>
35
36 </script>
37 </head>
38 <body>
39 </body>
40 </html>
41 </?>
```

PHP Hypertext Preprocessor file length: 2320 lines: 91 Ln: 1 Col: 1 Sel: 0|0 UNIX ANSI as UTF-8 INS

Εικόνα 9. « PHP φόρμα καταχώρησης του εξαρτώμενου μέλους »

```

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
add_emp.php
23 <?php
24 <?>
25 <doctype html>
26 <html lang="en">
27 <!--[if !IE 7]> <html class="lx-ie9 lx-ie8 lx-ie7" lang="en"> <![endif]-->
28 <!--[if IE 7]> <html class="lx-ie9 lx-ie8" lang="en"> <![endif]-->
29 <!--[if IE 8]> <html class="lx-ie9" lang="en"> <![endif]-->
30 <!--[if !IE 8]><!--> <html lang="en"> <!--<![endif]-->
31 <head>
32 <meta charset="utf-8">
33 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
34 <title></title>
35 <link rel="stylesheet" href="css/style1.css">
36 <link href="load.css" rel="stylesheet" type="text/css" >
37 <script src="load.js"></script>
38
39 </script>
40 <!--[if !IE 9]><script src="//html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
41 </head>
42 <body>
43 <form method="post" action="add_branch.html" class="login">
44 <fieldset>
45 <legend>Add New Branch</legend>
46 <p>
47 <label for="login">Customer:</label>
48 <select name="eid" id="login">
49 <?php
50
51 $sql2="SELECT E_ID FROM technicians";
52 $result3 = mysql_query($sql2) or die("Could not connect to branch");
53 while($row = mysql_fetch_array($result3))
54 {
55     echo<option value="'. @$row['E_ID'] .' ">". @$row['E_ID'] .'</option>;
56 }
57 </select>
58 </p>
59 </form>
60 </body>
61 </html>
62 </?>
PHP Hypertext Preprocessor file length: 2609 lines:103 Ln:1 Col:1 Sel:0|0 UNIX ANSI as UTF-8 INS

```

Εικόνα 10. « PHP φόρμα καταχώρησης στοιχείων υπαλλήλου»

```

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
add_number.php
1 <?php
2 require_once('config.php');
3
4 $rDB1 = mysql_select_db($database_test) or die("Could not select database");
5 if(isset($_POST['submit']))
6 {
7
8 $v1=$_POST['cid'];
9 $v2=$_POST['cid'];
10 $v3=$_POST['num'];
11 $v4=$_POST['puk1'];
12 $v5=$_POST['puk2'];
13 $v6=$_POST['pt'];
14 $v7=$_POST['eid'];
15 $date=date("Y-m-d");
16 $sql = "INSERT INTO phone_no VALUES ('$v3','$v4','$v5','$v6','$v1','$v2','$v7','New','$date',Null)";
17 //echo $sql;
18 $result1 = mysql_query($sql) or die("Could not Updated to phone_no");
19
20 }
21 <?>
22 <doctype html>
23 <html lang="en">
24 <!--[if !IE 7]> <html class="lx-ie9 lx-ie8 lx-ie7" lang="en"> <![endif]-->
25 <!--[if IE 7]> <html class="lx-ie9 lx-ie8" lang="en"> <![endif]-->
26 <!--[if IE 8]> <html class="lx-ie9" lang="en"> <![endif]-->
27 <!--[if !IE 8]><!--> <html lang="en"> <!--<![endif]-->
28 <head>
29 <meta charset="utf-8">
30 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
31 <title></title>
32 <link rel="stylesheet" href="css/style1.css">
33 <link href="load.css" rel="stylesheet" type="text/css" >
34 <script src="load.js"></script>
35

```

Εικόνα 11. «PHP φόρμα καταχώρησης στοιχεία αριθμού»

```

62
63 </p>
64 <p>
65 <label for="login">Customer ID:</label>
66 <input type="text" name="cid" id="login" required>
67 </b>
68 </p>
69 <p>
70 <label for="login">First Name:</label>
71 <input type="text" name="f_name" id="login" required>
72 </p>
73 <p>
74 <label for="login">Last Name:</label>
75 <input type="text" name="l_name" id="login" required>
76 </p>
77 <p>
78 <label for="login">Date of Birth:</label>
79 <input type="text" name="dob" id="login" required>
80 </p>
81 <p>
82 <label for="login">Address:</label>
83 <input type="text" name="add1" id="login" placeholder="Address Line 1">
84 </p>
85 <p>
86 <label for="login">&nbsp;&nbsp;&nbsp;</label>
87 <input type="text" name="add2" id="login" placeholder="Address Line 2" >
88 </p>
89 <p>
90 <label for="login">City:</label>
91 <input type="text" name="city" id="login" required>
92 </p>
93 </p>
94 </p>
95 <input type="text" name="city" id="login" required>
96

```

PHP Hypertext Preprocessor file length: 2956 lines: 116 Ln: 1 Col: 1 Sel: 0 | 0 UNIX ANSI as UTF-8 INS

Εικόνα 12. « PHP φόρμα καταχώρησης στοιχείων πελάτη»

### 6.3.2 PHP (Φόρμα παρουσίασης)

```

48
49
50
51
52 </script>
53
54 <div id="main-content">
55
56
57 <div class="show-grid" style="margin-bottom:20px;">
58 <table class="table-grid" id="table-grid">
59 <thead>
60 <tr>
61
62 <th>ID</th>
63 <th>First Name</th>
64 <th>Last Name</th>
65 <th>DOB</th>
66 <th>Salary</th>
67 <th>Contact_no</th>
68 <th>Branch ID</th>
69 <th>title</th>
70 <th>university</th>
71 <th>date_of_acq</th>
72 <th>Delete</th>
73 </tr>
74 </thead>
75 <tbody>
76 <?php //include("showband.php") ?>
77 </tbody>
78 </table>
79 </div>
80 </div><!--End of main-content-->
81
82

```

PHP Hypertext Preprocessor file length: 1683 lines: 85 Ln: 1 Col: 1 Sel: 0 | 0 Dos/Windows ANSI as UTF-8 INS

Εικόνα 13. «PHP φόρμα παρουσίασης προϊστάμενου καταστήματος»

```

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
artennashow.php
46      error:function(){
47          alert('Unable to read .');
48      }
49      });
50  }
51  });
52  </script>
53
54  <div id="main-content">
55
56
57      <div class="show-grid" style="margin-bottom:20px;">
58          <table class="table-grid" id="table-grid">
59              <thead>
60                  <tr>
61
62                      <th>ID</th>
63                      <th>Name</th>
64                      <th>Address</th>
65
66                      <th>B_ID</th>
67                      <th>Delete</th>
68                  </tr>
69              </thead>
70              <tbody>
71                  <?php //include("showband.php")?>
72              </tbody>
73          </table>
74      </div>
75  </div><!--End of main-content-->
76
77
78
79
80
PHP Hypertext Preprocessor file      length : 1535   lines : 80      Ln : 1   Col : 1   Sel : 0 | 0      Dos\Windows   ANSI as UTF-8   INS

```

Εικόνα 14. «PHP φόρμα παρουσίασης στοιχείων κεραίας»

```

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
customershow.php
1
2
3  <script type="text/javascript" src="jquery-1.9.1.js"></script>
4  <script type="text/javascript">
5      $(document).ready(function(){
6          loadBand();
7
8
9
10
11      $(document).on('click','table tr td .delete',function(e){
12          e.preventDefault();
13          var C_ID = $(this).closest('tr').find('td:eq(0)').text();
14          $.ajax({
15              url: 'customer.php',
16              type: "POST",
17              data: ({action: 'Delete',id:C_ID}),
18              success: function(data){
19
20                  alert('has been deleted successfully');
21                  loadBand();
22              },
23              error:function(){
24                  alert('Unable to delete band.');
```

Εικόνα 15. «PHP φόρμα παρουσίασης στοιχείων πελάτη»

```

47         alert('Unable to read .');
48     }
49     });
50 }
51 });
52 </script>
53
54 <div id="main-content">
55
56
57     <div class="show-grid" style="margin-bottom:20px;">
58         <table class="table-grid" id="table-grid">
59             <thead>
60                 <tr>
61
62                     <th>Phone No</th>
63                     <th>pk1</th>
64                     <th>pk2</th>
65                     <th>PackageType</th>
66                     <th>C_ID</th>
67                     <th>P_ID</th>
68                     <th>Delete</th>
69                 </tr>
70             </thead>
71             <tbody>
72                 <?php //include("showband.php") ?>
73             </tbody>
74         </table>
75     </div>
76 </div><!--End of main-content-->
77
78
79
80
81

```

PHP Hypertext Preprocessor file      length: 1575   lines: 81      Ln: 1   Col: 1   Sel: 0 | 0      Dos/Windows      ANSI as UTF-8      INS

Εικόνα 16. «PHP φόρμα παρουσίασης στοιχείων αριθμού τηλεφώνου»

### 6.3.3 PHP (Φόρμα report)

```

1 <?php
2     require_once('config.php');
3
4     $rDB1 = mysql_select_db($database_test) or die("Could not select database");
5     $sql2="SELECT MONTHNAME( date ) as MONTH , b.B_ID, B_Name, PackageType, count( P_No)as total
6     FROM branch AS b, customer AS c, phone_no AS p
7     WHERE b.B_ID = c.B_ID
8     AND c.C_ID = p.C_ID
9     AND TYPE = 'New'
10    GROUP BY PackageType, Month( date ) , b.B_ID";
11    $result3 = mysql_query($sql2) or die("Could not connect to branch");
12    $month="";
13    $branch="";
14    while($row = mysql_fetch_array($result3))
15    {
16        if($month!=$row['MONTH'])
17        {
18            echo "<h1/><H1>".$row['MONTH']. "</H1>";
19        }
20        if($branch!=$row['B_ID'] || ($branch==$row['B_ID'] && $month!=$row['MONTH']))
21        {
22            echo "<h2/><H2>".$row['B_ID']. " " . $row['B_Name']. "</H2><h2/>";
23            $branch=$row['B_ID'];
24        }
25        echo " " . $row['PackageType']. " " . $row['total'];
26        //echo "<hr/>";
27        $month=$row['MONTH'];
28    }
29
30
31
32 <?>

```

PHP Hypertext Preprocessor file      length: 1012   lines: 32      Ln: 1   Col: 1   Sel: 0 | 0      UNIX      ANSI as UTF-8      INS

Εικόνα 17. «PHP φόρμα αναφοράς συμβολαίου και προ πλήρωσης»

```

1 <?php
2     require_once('config.php');
3
4     $rDB1 = mysql_select_db($database_test) or die("Could not select database");
5     $sql2="SELECT MONTHNAME( Paid_date )as MONTH, b.B_ID, B_Name, sum( Paid_Amount )as total
6 FROM payment AS p, accounts AS a, customer AS c, branch AS b
7 WHERE p.AC_ID = a.AC_ID
8 AND c.C_ID = a.C_ID
9 AND c.B_ID = b.B_ID
10 GROUP BY b.B_ID,Month( Paid_date )";
11     $result3 = mysql_query($sql2) or die("Could not connect to branch");
12     $month="";
13     $branch="";
14     while($row = mysql_fetch_array($result3))
15     {
16         if($month!=$row['MONTH'])
17         {
18             echo "<h1>".$row['MONTH']. "</h1>";
19         }
20         if($branch!=$row['B_ID'] || ($branch==$row['B_ID'] && $month!=$row['MONTH']))
21         {
22             echo "<h2>".$row['B_ID']. " ".$row['B_Name']. "</h2><br/>";
23             $branch=$row['B_ID'];
24         }
25         echo " Monthly income " . $row['total'];
26         //echo "<br/>";
27         $month=$row['MONTH'];
28     }
29 }
30 }
31 }
32 ?>

```

PHP Hypertext Preprocessor file      length: 1007   lines: 32      Ln: 1   Col: 1   Sel: 0 | 0      UNIX      ANSI as UTF-8      INS

Εικόνα 18. «PHP φόρμα αναφορών αποδείξεων καταστήματος»

```

1 <?php
2     require_once('config.php');
3
4     $rDB1 = mysql_select_db($database_test) or die("Could not select database");
5     $sql2="SELECT e.E_ID, E_Fname, E_Lname, count( P_No ) as total
6 FROM phone_no AS p, employee AS e
7 WHERE TYPE = 'New'
8 AND e.E_ID = p.E_ID
9 AND P.contract_type = 'Contract'
10 GROUP BY E_ID";
11     $result3 = mysql_query($sql2) or die("Could not connect to branch");
12     $month="";
13     $branch="";
14     while($row = mysql_fetch_array($result3))
15     {
16
17
18
19
20         echo "<h2>".$row['E_ID']. " ".$row['E_Fname']. " ".$row['E_Lname']. " New Contract " . $row['total']. "</h2><br/>";
21         $branch=$row['B_ID'];
22
23         echo "<br/>";
24     }
25 }
26 }
27 }
28 ?>

```

PHP Hypertext Preprocessor file      length: 740   lines: 28      Ln: 1   Col: 1   Sel: 0 | 0      UNIX      ANSI as UTF-8      INS

Εικόνα 19. «PHP φόρμα αναφοράς νέων συμβολαίων»

```

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
prepaid_renewals_report.php
1 <?php
2     require_once('config.php');
3
4     $rDB1 = mysql_select_db($database_test) or die("Could not select database");
5     $sql2="SELECT MONTHNAME( renew_date ) as MONTH , b.B_ID, B_Name, count( P_No ) as total
6 FROM phone_no AS p, customer AS c, branch AS b
7 WHERE TYPE = 'Renew'
8 AND c.C_ID = p.C_ID
9 AND b.B_ID = c.B_ID
10 AND P.no_of_payments = 'Prepaid'
11 GROUP BY Month( renew_date ) , b.B_ID";
12     $result3 = mysql_query($sql2) or die("Could not connect to branch");
13     $month="";
14     $branch="";
15     while($row = mysql_fetch_array($result3))
16     {
17         if($month!=$row['MONTH'])
18         {
19             echo "<br/><h1>".@$row['MONTH']. "</h1>";
20         }
21         if($branch!=$row['B_ID'] || ($branch==$row['B_ID'] && $month!=$row['MONTH']))
22         {
23             echo "<br/><h2>".@$row['B_ID']. " ".@$row['B_Name']. "</h2><br/>";
24             $branch=$row['B_ID'];
25         }
26         echo " Monthly Prepaid renew " .@$row['total'];
27         //echo "<br/>";
28         $month=$row['MONTH'];
29     }
30 }
31 }
32 }
33 ?>

```

PHP Hypertext Preprocessor file length: 1027 lines: 33 Ln: 1 Col: 1 Sel: 0|0 UNIX ANSI as UTF-8 INS

Εικόνα 20. « PHP φόρμα αναφοράς ανανέωσης προπληρωμένων συμβολαίων»

### 6.3.4 PHP (Φόρμα πληρωμής)

```

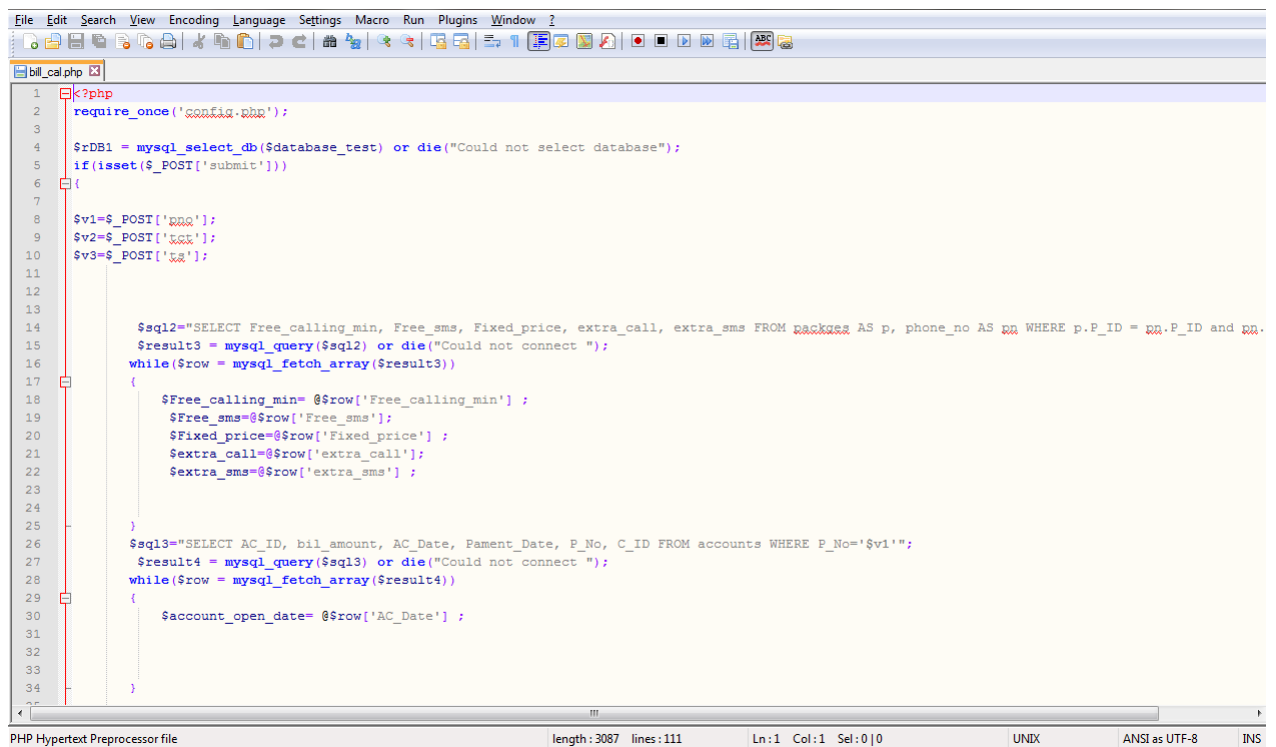
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
payment.php
1 <?php
2     require_once('config.php');
3
4     $rDB1 = mysql_select_db($database_test) or die("Could not select database");
5     if(isset($_POST['submit']))
6     {
7
8         $v1=$_POST['pano'];
9         $v2=$_POST['amo'];
10
11
12
13
14
15         $sql3="SELECT AC_ID, bil_amount, AC_Date, Pament_Date, P_No, C_ID FROM accounts WHERE P_No='$v1'";
16         $result4 = mysql_query($sql3) or die("Could not connect ");
17         while($row = mysql_fetch_array($result4))
18         {
19             $said= @$row['AC_ID'] ;
20
21
22
23         }
24         $sql = "INSERT INTO payment VALUES ('$said','$v2',Null)";
25         //echo $sql;
26         $result1 = mysql_query($sql) or die("Could not Updated ");
27
28
29     }
30 }
31 <?>
32 <!doctype html>
33 <html lang="en">
34 <!--[if !IE 7]> <html class="lt-ie9 lt-ie8 lt-ie7" lang="en"> <![endif]-->
35 <!--[if IE 7]> <html class="lt-ie9 lt-ie8" lang="en"> <![endif]-->
36 <!--[if IE 8]> <html class="lt-ie9" lang="en"> <![endif]-->

```

PHP Hypertext Preprocessor file length: 2196 lines: 90 Ln: 1 Col: 1 Sel: 0|0 UNIX ANSI as UTF-8 INS

Εικόνα 21. «PHP φόρμα πληρωμής»

### 6.3.5 PHP (Λογαριασμός)



```
1 <?php
2 require_once('config.php');
3
4 $rDB1 = mysql_select_db($database_test) or die("Could not select database.");
5 if(isset($_POST['submit']))
6 {
7
8     $v1=$_POST['pno'];
9     $v2=$_POST['acc'];
10    $v3=$_POST['acc'];
11
12
13
14    $sql2="SELECT Free_calling_min, Free_sms, Fixed_price, extra_call, extra_sms FROM packages AS p, phone_no AS pn WHERE p.P_ID = pn.P_ID and pn.
15 $result3 = mysql_query($sql2) or die("Could not connect ");
16 while($row = mysql_fetch_array($result3))
17 {
18     $Free_calling_min= @$row['Free_calling_min'] ;
19     $Free_sms=@$row['Free_sms'];
20     $Fixed_price=@$row['Fixed_price'] ;
21     $extra_call=@$row['extra_call'];
22     $extra_sms=@$row['extra_sms'] ;
23
24 }
25
26 $sql3="SELECT AC_ID, bil_amount, AC_Date, Pament_Date, P_No, C_ID FROM accounts WHERE P_No='$v1'";
27 $result4 = mysql_query($sql3) or die("Could not connect ");
28 while($row = mysql_fetch_array($result4))
29 {
30     $account_open_date= @$row['AC_Date'] ;
31
32
33
34 }
```

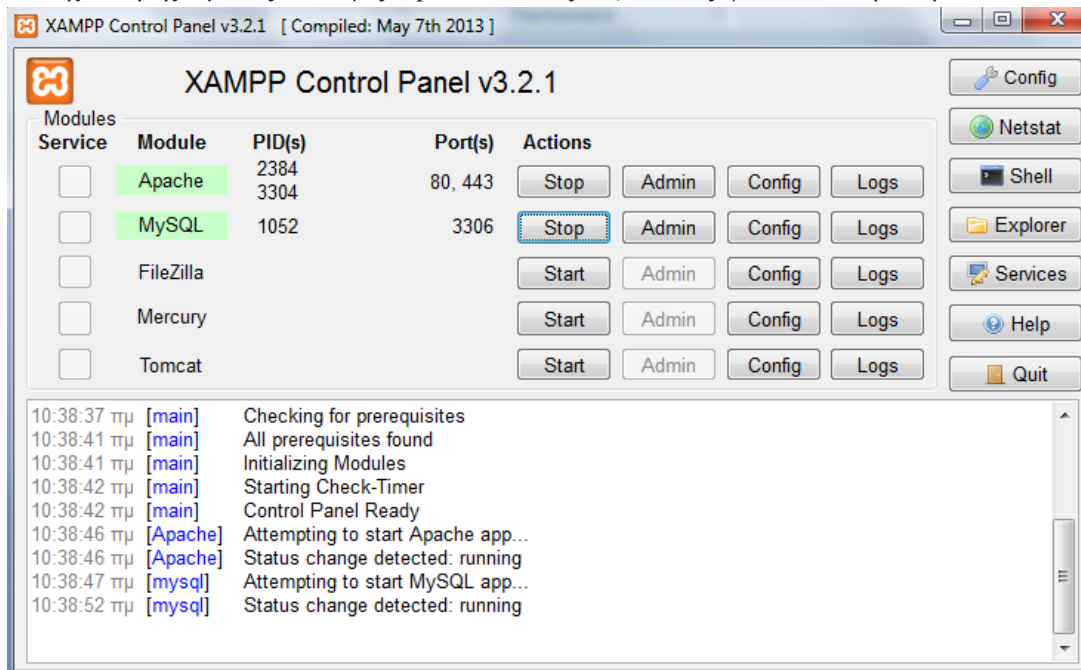
PHP Hypertext Preprocessor file      length: 3087    lines: 111      Ln:1    Col:1    Sel:0|0      UNIX      ANSI as UTF-8      INS

Εικόνα 22. « PHP φόρμα λογαριασμού»



## Κεφάλαιο 7: Εγκατάσταση Web Εφαρμογής

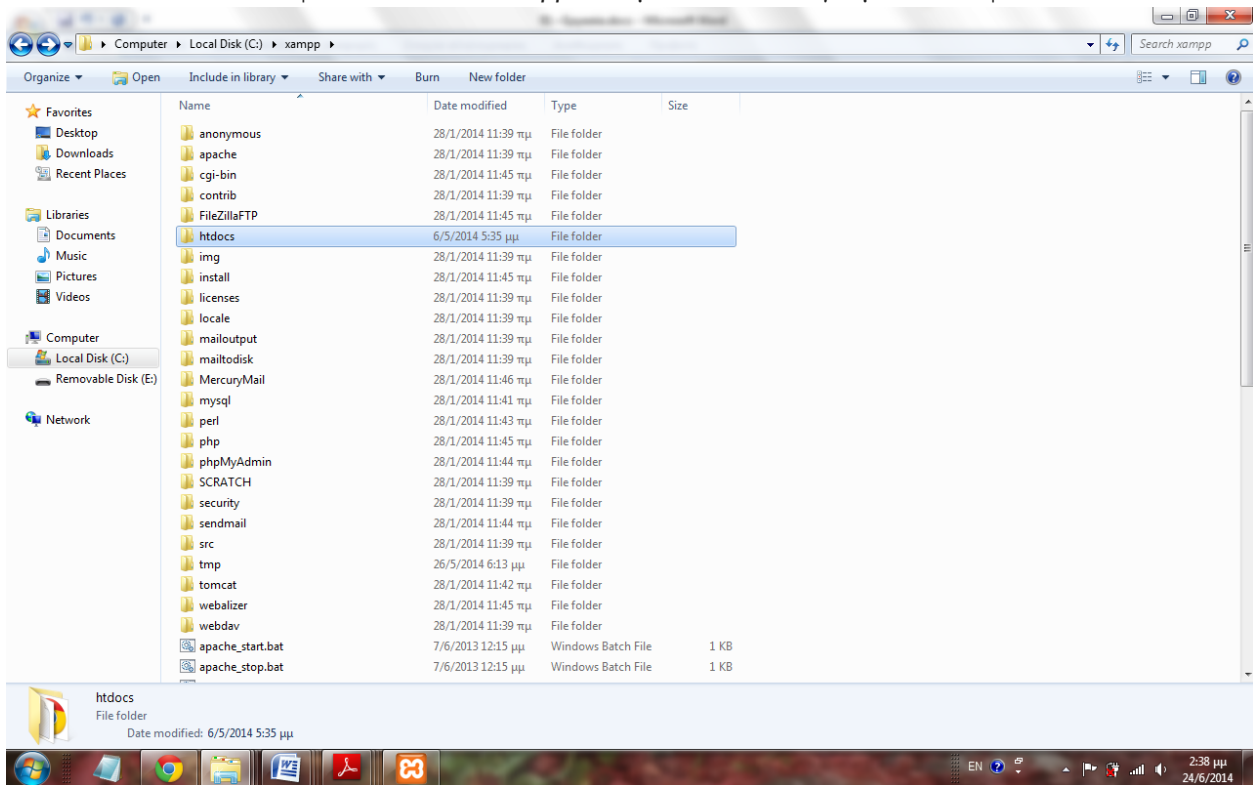
Πηγαίνουμε στην ιστοσελίδα: <http://xampp.joydownload.com/?c=18&gclid=CLvvu5vO-L0CFSQFwwod14gAcw>. Επιλέγουμε την έκδοση του XAMPP που ταιριάζει στο λογισμικό μας και την κατεβάζουμε. Αφού κατέβει το XAMPP, προχωράμε στην εγκατάσταση. Στη συνέχεια τρέχουμε τις επιλογές Apache και MySQL, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 23. « Πίνακας Ελέγχου XAMPP. Διεργασία εφαρμογών Apache, MySQL»

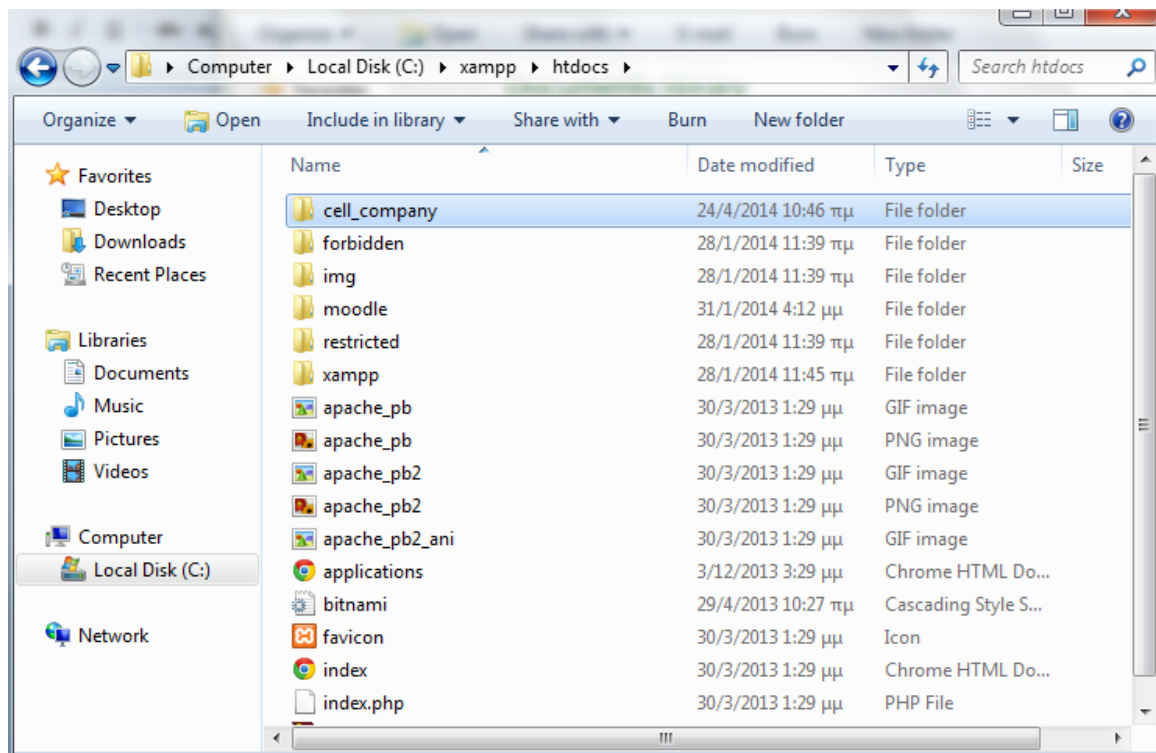
Μετά πάμε στον φάκελο C: (local disk) του υπολογιστή μας, όπου θα δούμε ότι μέσα του έχει δημιουργηθεί ένας φάκελος με την ονομασία XAMPP.

Μέσα στον φάκελο XAMPP βρίσκουμε και επιλέγουμε τον φάκελο htdocs.



Εικόνα 24. « Τοπικός δίσκος, φάκελος htdocs».

Μέσα σε αυτόν τον φάκελο τοποθετούμε τον φάκελο του συστήματος που έχει δημιουργηθεί στα πλαίσια αυτής της εργασίας



Εικόνα 25. «Τοπικός δίσκος, φάκελος cell company»

Στην συνέχεια ανοίγουμε τον browser μας και πάμε στην διεύθυνση 127.0.0.1/XAMPP και επιλέγουμε για γλώσσα τα αγγλικά. Θα μας εμφανιστεί η παρακάτω εικόνα:

**XAMPP 1.8.3**  
[PHP: 5.5.6]

**Welcome**  
Status  
Security  
Documentation  
Components  
Applications

**Php**  
phpinfo()  
CD Collection  
Biorhythm  
Instant Art  
Phone Book

**Perl**  
perlinfo()  
Guest Book

**J2ee**  
Info  
omcat examples

**Tools**  
phpMyAdmin  
FileZilla FTP  
Webalizer  
Mail

©2002-2014  
...**APACHE**  
**FRIENDS...**

### Welcome to XAMPP for Windows!

**Congratulations:**  
**You have successfully installed XAMPP on this system!**

Now you can start using Apache and Co. You should first try »Status« on the left navigation to make sure everything works fine.

For OpenSSL support please use the test certificate with <https://127.0.0.1> or <https://localhost>

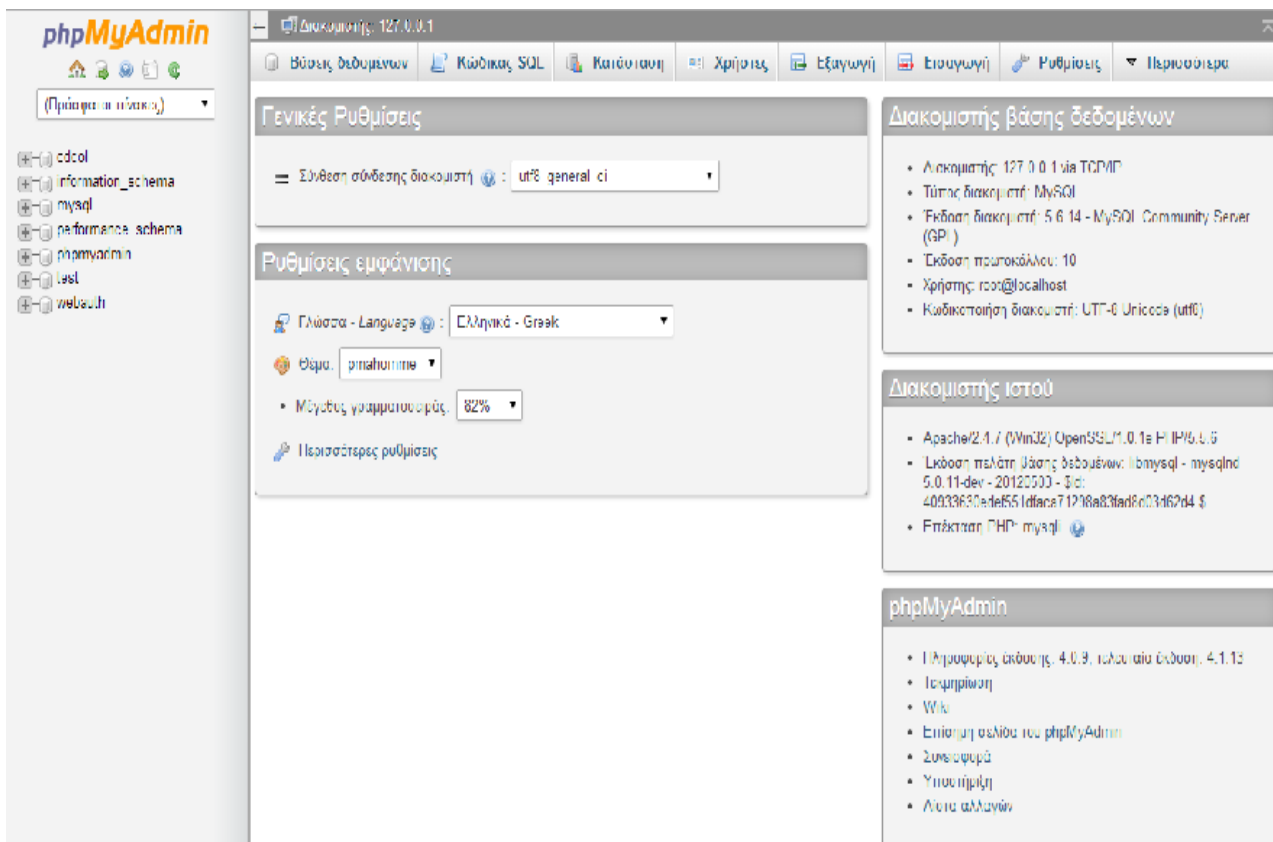
Good luck, Kay Vogelgesang + Kai 'Oswald' Seidler

### Install applications on XAMPP using BitNami

Apache Friends and BitNami are cooperating to make dozens of open source applications available on XAMPP, for free. BitNami-packaged applications include Wordpress, Drupal, Joomla! and dozens of others and can be deployed with one-click installers. Visit the [BitNami XAMPP page](#) for details on the currently available apps.

Εικόνα 26. « Αρχική σελίδα XAMPP»

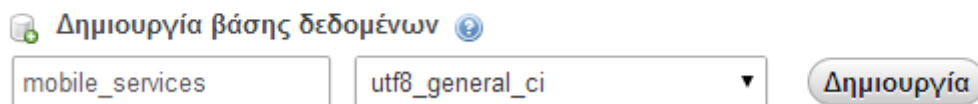
Από τον πίνακα στα αριστερά βρίσκουμε εκεί που λέει Tools και επιλέγουμε phpMyAd.  
Τότε ανοίγουμε την παρακάτω εικόνα :



Εικόνα 27. « Αρχική σελίδα phpMyAd»

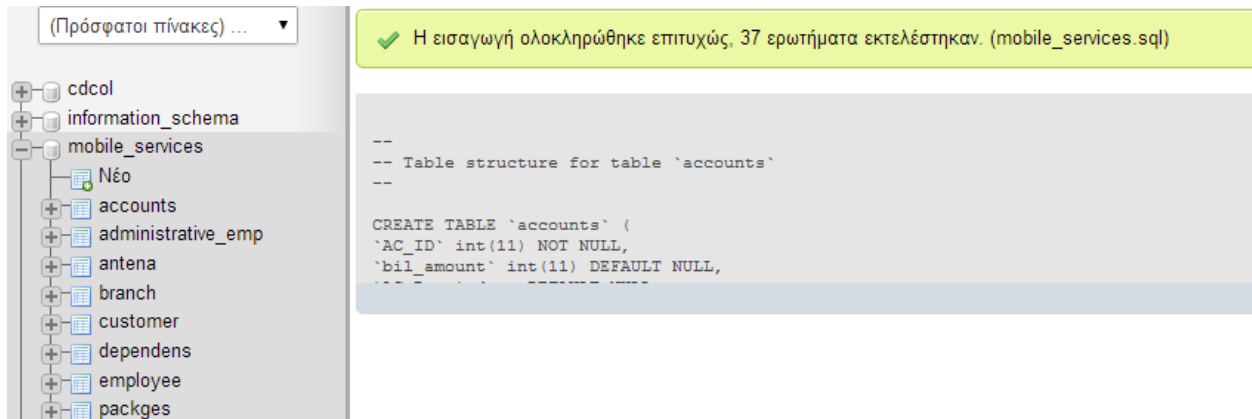
Επιλέγουμε «βάσεις δεδομένων» και δημιουργούμε μια νέα βάση, την οποία ονομάζουμε με το ίδιο όνομα που είναι και η βάση δεδομένων μας στο σύστημά μας, δηλαδή mobile\_services. Στην Σύνθεση επιλέγουμε utf8\_general\_ci . Μετά επιλέγουμε Δημιουργία.

## Βάσεις δεδομένων



Εικόνα 28. «Δημιουργία βάσης δεδομένων»

Η βάση που δημιουργήσαμε θα εμφανιστεί στην στήλη αριστερά μας. Αφού την επιλέξουμε, βρίσκουμε στο πάνω μενού εκεί που λέει Εισαγωγή και εισάγουμε την βάση του συστήματος που έχει δημιουργηθεί για αυτήν την εργασία. Στην συνέχεια πατάμε Εκτέλεση. Θα πρέπει να μας ενημερώσει πως η εισαγωγή ολοκληρώθηκε επιτυχώς.



Εικόνα 29. «Εισαγωγή στη Βάση Δεδομένων»

Ανοίγω την κεντρική μου σελίδα και το σύστημα είναι έτοιμο να λειτουργήσει  
[http://127.0.0.1/cell\\_company/mobi/index.html](http://127.0.0.1/cell_company/mobi/index.html)



## Κεφάλαιο 8: Λειτουργικότητα των διαδικασιών της εφαρμογής

Με βάση τα ζητούμενα της εργασίας, το σύστημα εκτελεί τις παρακάτω λειτουργίες. Για το υποκατάστημα υπάρχουν οι εξής επιλογές:

Branch
Add
View

Όπου στο view μπορούμε να εκτελέσουμε και τη διαδικασία της διαγραφής.

Για κάθε υποκατάστημα της εταιρίας, το σύστημα αποθηκεύει την ταυτότητα, το όνομα και τη διεύθυνση.

Branch ID:	MOBI-01-AA7
Branch Name:	Cell Company Athens
Address:	Sarafi 11
	Elliniko
City:	Athens
Reset	Submit

Εικόνα 30. «Υποκατάστημα της εταιρείας»

Αν θελήσουμε να δούμε ή να διαγράψουμε κάποιο κατάστημα που έχει καταχωρηθεί στο σύστημα:

The screenshot shows a web browser window with the address bar displaying `127.0.0.1/cell_company/mobi/branchshow.php`. Below the address bar is a table with the following content:

ID	Name	Address	Delete
0	Cell Company Athens	Sarafi 11, Elliniko, Athens	<a href="#">Delete</a>

Overlaid on the right side of the browser window is a confirmation dialog box with the title "Ειδοποίηση από τη σελίδα στη διεύθυνση 127.0.0.1:". The message inside the dialog box reads "Band has been deleted successfully". There is an "OK" button at the bottom right of the dialog box.

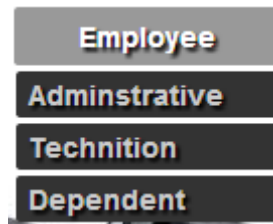
Εικόνα 31. «Διαγραφή υποκαταστήματος»

Για τους υπαλλήλους έχουμε δύο κατηγορίες:

α) τους διοικητικούς, για τους οποίους θέλουμε να αποθηκεύουμε πληροφορίες για το πτυχίο τους (τίτλο, πανεπιστήμιο που το εξέδωσε και ημερομηνία κτήσης),

β) τους τεχνικούς, για τους οποίους θέλουμε να αποθηκεύουμε πληροφορίες για την ειδικότητα και τα έτη προϋπηρεσίας.

Στο σύστημα βλέπουμε ότι, στην επιλογή employee εμφανίζονται, πέρα από τις δύο αυτές κατηγορίες και η επιλογή «dependent», που αφορά το εξαρτώμενο μέλος του υπαλλήλου.



Εικόνα 32. «Υπάλληλου»

Για τους διοικητικούς η επιλογή «customer» προκύπτει δεδομένου ότι ένας πελάτη που έχει σύνδεση, έχει υπογράψει ένα και μόνο ένα συμβόλαιο με κάποιον υπάλληλο της εταιρίας.

Αντίστοιχα η φόρμα για τους τεχνικούς είναι:

Εικόνα 33. «Πελάτες»

Για τις κεραίες που έχει η εταιρία καταχωρούμε το όνομα, την διεύθυνση και τον τύπο της κεραίας.

Ακόμα στην κεραία μπορεί κανείς να επιλέξει από τα υποκαταστήματα σε ποιο «ανήκει».



Branch ID:

Antenna ID:   
19003

Antenna Name:

Address:

City:

Εικόνα 34. «Κεραίες Εταιρείας»

Επιπλέον, εκτός από την καταχώρηση, μπορούμε να δούμε τις κεραίες που έχουν ήδη καταχωρηθεί και που έχουν επισκευαστεί.

Antena	ID	Name	Address	B_ID	Delete
<b>Add</b>	19003	Ant001	Kifisias 13 Argiroupoli	Athina 0	<a href="#">Delete</a>
<b>View</b>	19005	Ant002	Aneksartisias 36 Kipseli	Athina 1	<a href="#">Delete</a>
<b>Repair</b>					

Εικόνα 35. «Καταχωρημένες και επιδιορθωμένες κεραίες»

Για κάθε πελάτη το σύστημα αποθηκεύει το όνομα, την διεύθυνση, τον αριθμό ταυτότητας, το επάγγελμα.

**Account**

**Add**

Για κάθε αριθμό τηλεφώνου που έχει η εταιρία θέλουμε να αποθηκεύουμε τον αριθμό κλήσης, τον κωδικό αριθμό (ruk1 και 2), αν είναι σύνδεση (special) ή καρτοκινητό (normal).

Package ID:

Package Name:

Free Calling Min:

Free SMS:

Fixed Price:

Extra Charge for Call:

Extra Charge for SMS:

Εικόνα 36. «Πληροφορίες αριθμού τηλεφώνου»

Για την πληρωμή έχουμε:

Add New Payment

Phone NO:

Paid Amount:

[Click here to return to index](#)

Και

Add New Branch

Phone NO:

Total Call time:

Total SMS :

Εικόνα 37. «Πληρωμές»

Όπου ο αριθμός τηλεφώνου επιλέγεται από την λίστα των εγγεγραμμένων πελατών.

## Κεφάλαιο 9: Επίλογος

Στη συγκεκριμένη εργασία έγινε η ανάλυση και η ανάπτυξη ενός Πληροφοριακού Συστήματος, που έχει σχέση με την Κινητή Τηλεφωνία. Βασικές διαδικασίες που χρησιμοποιήθηκαν ήταν η μοντελοποίηση του Πληροφοριακού Συστήματος και η ανάπτυξη του λογισμικού σε Web Platform. Χρησιμοποιήθηκαν όλες οι διαδικασίες του Επιχειρησιακού Συστήματος που είχαν προδιαγραφεί για την ηλεκτρονική του χρήση.

Συμπέρασμα:

Κατά την υλοποίηση φάνηκε πως ο σχεδιασμός του συστήματος, δεν είχε την πλέον δυνατή αποδοτικότητα, όσον αφορά την εκτέλεση των διεργασιών, από θέμα ροής των λειτουργιών. Για παράδειγμα, θα πρέπει πρώτα να καταχωρείται το τηλέφωνο του πελάτη και στην συνέχεια να γίνεται η εγγραφή των υπόλοιπων στοιχείων του στο σύστημα, με το να ανοίγεται για αυτόν ένας λογαριασμός (account). Αυτό θα μπορούσε να αποφευχθεί είτε με έναν διαφορετικό σχεδιασμό, είτε αν θεωρήσουμε ως δεδομένο το ότι η επιχείρηση θα φροντίσει για την σωστή κατάρτιση των υπαλλήλων της σχετικά με την χρήση του συστήματος που πρόκειται να χρησιμοποιήσει.

## Κεφάλαιο 10: Βιβλιογραφικές Αναφορές / Πηγές

### Βιβλιογραφία

1. Tenenbaum, J. M. AI Meets Web 2.0: Building the Web of Tomorrow, Today. (2006). AI Magazine 27(4), 47-68.
2. Deshpande, Y., Murugesan, S., Ginige, A., Hansen, S., Schwbe, D., Gaedke, M., & White, B. Web Engineering. Rinton Press Journal of Web Engineering. (2002).
3. Information Systems Today: Why IS Matters, 2008, Leonard M. Jessup, Joseph S.Valacich, Michael Wade, 0131740393, 9780131740396, Pearson Education Canada, 2008
4. Efraim Turban, Lee Jae, King David, Chung H. Michael, “Electronic commerce – A managerial perspective”, Prentice Hall 2000
5. TU Shuli, School of Business Administration, Jiangxi University of Finance and Economics, “On Courier Service Deregulation and Sustainable Development”, Schach, S., Object-oriented and classical software engineering. McGraw Hill. (2005).
6. Subramanian, N., & Whitson, G. Augmented WebHelix: A Practical Process for Web Engineering
7. In Daniel M. Brandon (ed.), Software Engineering for Modern Web Applications: Methodologies and Technologies (pp. 25-52). IGI Global.
8. Tenenbaum, J. M. AI Meets Web 2.0: Building the Web of Tomorrow, Today. (2006).
9. Luke Welling. Laura Tomson, ‘Php and MySQL Web Development’, 2008
10. Laura Lemay, ‘Εγχειρίδιο της Html 3.2’, 3<sup>η</sup> Αμερικανική Έκδοση, Μ. Γκιούρδας, 1997
11. Σημειώσεις καθηγητή κύριου Παπαδάκη Νικολάου.

### Ηλεκτρονικές Πηγές

<https://www.apachefriends.org/index.html>  
<http://www.w3schools.com/css/>  
<http://www.oracle.com/gr/index.html>  
<http://www.php.net>  
<http://www.daniweb.com>  
<http://www.plus2net.com>  
<http://www.jonasjohn.de/snippets/php/headers.htm>  
<http://www.freestuff.gr>  
<http://www.webmasterworld.com>  
<http://www.codeproject.com>  
<http://www.the-art-of-web.com>  
<http://www.stackoverflow.com>  
<http://www.developertutorials.com>  
<http://www.mysql.com/>  
<http://www.apache.org/>  
<http://en.wikipedia.org>