

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης



Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή εργασία

**Τίτλος: Πληροφοριακό Δικτυοκεντρικό Σύστημα
Τραπεζικών Συναλλαγών**

Καλλιάρια Αικατερίνη (ΑΜ: 1885)

Επιβλέπων καθηγητής : Παπαδάκης Νικόλαος

Επιτροπή Αξιολόγησης :

Ημερομηνία παρουσίασης: 27-5-2014

The paper “**Electronic bank management system**” is written within an attempt to develop an online information system that will be used in all transactions.

Main aim of this paper is to design and upgrade a website of a bank. The primary goal was to create a database, in which all obligatory information suitable for a recording system as such, would be filled.

The website will provide the bank with the possibility of managing with the accounts and the cards of its clients.

Project structure: The paper consists of 5 chapters.

In the 1st chapter, some introductory information on the project are provided, such as how useful and necessary it is.

In the 2nd chapter, the theoretical background of the project is presented. Terms and notions, such as internet etc, are clarified.

The 3rd chapter discusses the new ICT methods and tools that are used in order to accomplish the development of the application.

In the 4th chapter, there is the analysis of the software design pattern starting with the presuppositions in demand.

In chapter 5, the main course of the project is presented; in specifics, how this database and the webpage are created.

Chapter 6 contains the most important parts of the code.

Chapter 7 includes the evaluation of how the system works.

In the end, there is added the bibliography supporting this paper.

Πίνακας περιεχομένων

1. Εισαγωγικά στοιχεία	6
1.1. Πρόλογος.....	6
1.2. Σκοπός και στόχοι της εργασίας.....	6
1.3. Διάρθρωση της εργασίας	6
2. Θεωρητικό υπόβαθρο	8
2.1. Τι είναι το Διαδίκτυο	8
2.2. Ιστορία και εξέλιξη του Διαδικτύου	9
2.3. Τρόποι πρόσβασης στο διαδίκτυο	9
2.4. Το προγραμματιστικό περιβάλλον.....	10
2.4.1. HTML	10
2.4.1.1. Η Ιστορία της HTML.....	10
2.4.1.2. Ο ορισμός της HTML	11
2.4.1.3. Η δομή της HTML	11
2.4.2. CSS	12
2.4.2.1. Η ιστορία του CSS.....	12
2.4.2.2. Ο ορισμός του CSS.....	13
2.4.3. Η γλώσσα PHP	14
2.4.4. PHP Sessions.....	16
2.4.5. MySQL.....	19
2.5. PHPMysqlAdmin.....	20
2.5.1. Δυνατότητες του PhpMyAdmin.....	20
2.6. Javascript	21
2.6.1. Τι είναι Javascript	21
2.6.2. Δυνατότητες της Javascript	23
2.6.3. Τρόπος Λειτουργίας της JavaScript	24
3. Εργαλεία Ανάπτυξης Ιστοσελίδας.....	26
3.1. Εγκατάσταση <i>EasyPHP</i>	26
3.2. Adobe Dreamweaver CS6.....	31
4. Ανάλυση Λογισμικού.....	33
4.1. Καταγραφή απαιτήσεων	34

4.2. Περιπτώσεις χρήσης (Use Cases)	36
5. Σχεδιασμός Βάσεων Δεδομένων.....	39
5.1. Μοντέλο Οντοτήτων - Συσχετίσεων (E-R Model).....	39
5.2. Σχεσιακό Μοντέλο.....	42
5.3. Ο σχεδιασμός της βάσης δεδομένων.....	44
5.3.1. Ζωγραφίζοντας το E-R διάγραμμα.....	44
5.3.2. Μετατροπή E-R σε πίνακες βάσης δεδομένων.....	45
5.3.3. Η εφαρμογή της βάσης δεδομένων.....	46
6. Εγχειρίδιο χρήσης και ανάλυση κώδικα	49
6.1. Ανάλυση του κώδικα.....	49
6.1.1. Εγγραφή πελάτη.....	49
6.1.2. Σύνδεση με την βάση δεδομένων.....	52
6.1.3. Διαγραφή λογαριασμού.....	53
6.1.4. Εμφάνιση ιστοσελίδας.....	54
7. Έλεγχος Συστήματος.....	58
7.1. Τα στάδια του ελέγχου.....	58
7.2. Έλεγχος λειτουργίας εγγραφής πελάτη.....	59
7.3. Έλεγχος λειτουργίας ανοίγματος λογαριασμού	61
7.4. Έλεγχος λειτουργίας Ανάληψης Χρημάτων	62
8. Επίλογος – Συμπεράσματα – Μελλοντικές Επεκτάσεις.....	64
8.1. Υλοποίηση απαιτήσεων	64
8.2. Προβλήματα	65
8.3. Γνώση που αποκτήθηκε	65
8.4. Μελλοντικές επεκτάσεις	65
Αναφορές.....	67

1. Εισαγωγικά στοιχεία

1.1. Πρόλογος

Η εργασία «Ηλεκτρονικό σύστημα διαχείρισης τραπεζής» αναπτύχθηκε στα πλαίσια μιας προσπάθειας για την ανάπτυξη ενός διαδικτυακού πληροφοριακού συστήματος που θα δίνει τη δυνατότητα για συναλλαγές..

1.2. Σκοπός και στόχοι της εργασίας

Κύριος σκοπός της παρούσας εργασίας είναι ο σχεδιασμός και ανάπτυξη μιας ιστοσελίδας διαχείρισης τράπεζας. Πρωταρχικός σκοπός αποτέλεσε η δημιουργία και κανονικοποίηση μιας βάσης δεδομένων, στην οποία θα καταχωρούνται όλα τα απαιτούμενα στοιχεία για ένα τέτοιο σύστημα.

Η ιστοσελίδα θα δίνει τη δυνατότητα διαχείριση λογαριασμών και καρτών των πελατών της τράπεζας.

1.3. Διάρθρωση της εργασίας

Η εργασία αποτελείται από 5 κεφάλαια.

Στο 1^ο κεφάλαιο παρατίθενται κάποια εισαγωγικά στοιχεία της εργασίας όπως η αναγκαιότητα της.

Στο 2^ο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο της εργασίας. Παρουσιάζονται όροι όπως το διαδίκτυο κτλ.

Στο 3^ο κεφάλαιο γίνεται λόγος για τις οι τεχνολογίες και τα εργαλεία με τα οποία αναπτύχθηκε η εφαρμογή.

Το 4^ο κεφάλαιο αναλύει την σχεδίαση ενός λογισμικού, ξεκινώντας από τις απαιτήσεις που τίθενται.

Στο 5^ο κεφάλαιο παρουσιάζεται το κύριο μέρος της εργασίας δηλαδή το πώς δημιουργήθηκε η βάση δεδομένων και η ιστοσελίδα.

Στο 6^ο κεφάλαιο παρατίθενται τα σημαντικότερα κομμάτια του κώδικα.

Στο 7^ο κεφάλαιο γίνεται ο έλεγχος λειτουργίας του συστήματος.

Στο τέλος παρατίθεται η βιβλιογραφία στην οποία στηρίχθηκε η εργασία.

2. Θεωρητικό υπόβαθρο

2.1. Τι είναι το Διαδίκτυο

Το Διαδίκτυο ή Internet όπως ορίζεται το παγκόσμιο δίκτυο ηλεκτρονικών υπολογιστών, οι οποίοι επικοινωνούν μεταξύ τους χρησιμοποιώντας ένα κοινό πρωτόκολλο επικοινωνίας, το TCP/IP (Transmission Control Protocol/Internet Protocol). Ουσιαστικά, το διαδίκτυο είναι ένα σύνολο πολλών υποδικτύων διαφορετικού μεγέθους και περιοχής. Οι χρήστες του Διαδικτύου έχουν τη δυνατότητα να περιηγηθούν πολύ γρήγορα σε μια τεράστια αποθήκη πληροφοριών, να αποστείλουν και να λάβουν αρχεία, να κάνουν χρήση της ηλεκτρονικής αλληλογραφίας, και να κάνουν χρήση ενός μεγάλου εύρους διαφορετικών εφαρμογών.

Για να μεταφερθεί η πληροφορία χωρίς καμία αλλοίωση υφίστανται τα κατάλληλα πρωτόκολλα μεταφοράς (transfer protocol). Το πρωτόκολλο ορίζει την αποστολή και λήψη πληροφοριών. Τα πιο διαδεδομένα πρωτόκολλα από αυτά είναι τα εξής:

Hypertext Transfer Protocol (HTTP): Για την περιήγηση στον Παγκόσμιο Ιστό.

Simple Mail Transfer Protocol (SMTP): Για την υπηρεσία του ηλεκτρονικού ταχυδρομείου.

File Transfer Protocol (FTP): Για την υπηρεσία της μεταφοράς αρχείων.

Network News Transfer Protocol (NNTP): Για τη χρήση του Usenet και τη συμμετοχή σε ομάδες ειδήσεων – συζητήσεων (Newsgroups).

2.2. Ιστορία και εξέλιξη του Διαδικτύου

Η απαρχή του διαδικτύου συναντάται το 1969 όταν το Υπουργείο Εθνικής Αμύνης των Η.Π.Α. χορήγησε κονδύλια για τη δημιουργία δικτύου μεταξύ τεσσάρων υπερυπολογιστών. Το πρόγραμμα λεγόταν ARPA (Advanced Research Projects Agency). Η γεωγραφική θέση των υπολογιστών ήταν στο πανεπιστήμιο UCLA στο Los Angeles, στο πανεπιστήμιο της Santa Barbara, στο πανεπιστήμιο της Utah και στο ινστιτούτο ερευνών Stanford. Μέσα σε δύο έτη συνδέθηκαν στο δίκτυο αυτό ακόμα είκοσι υπολογιστές, ενώ σταδιακά το νούμερο αυτό μεγάλωνε με υψηλούς ρυθμούς. Το 1974 ήταν χρονιά ορόσημο, καθώς δόθηκε το σημερινό όνομα του διαδικτύου και έχασε την στρατιωτική του υπόσταση. Με εκρηκτικό ρυθμό αύξησης στο νέο αιώνα οι συνδεδεμένοι υπολογιστές έφτασαν τα 180.000.000 με πλήθος επιχειρήσεων να μπαίνουν δυναμικά στο χώρο αυτό.

Το διαδίκτυο χαρακτηρίζεται και ως παγκόσμιο ηλεκτρονικό χωριό, όπου, οι κάτοικοι του χωρίς περιορισμούς από υπηκοότητα, ηλικία, θρήσκευμα και χρώμα, διαμοιράζονται πληροφορίες και εκθέτουν ελεύθερα απόψεις πέρα από γεωγραφικά και κοινωνικά σύνορα. Σύμφωνα με μελέτες, αυτός ο παγκόσμιος ιστός υπολογιστών και χρηστών αριθμεί την τρέχουσα περίοδο πάνω από δέκα εκατομμύρια υπολογιστές και εκατό εκατομμύρια χρήστες, ενώ απλώνεται διαρκώς με εκθετικούς ρυθμούς με πάνω από ένα δισεκατομμύριο χρήστες.

2.3. Τρόποι πρόσβασης στο διαδίκτυο

Πλέον, οι τρόποι σύνδεσης στο διαδίκτυο έχουν αυξηθεί και ο τρόπος σύνδεσης αποτελεί συνάρτηση πολλών παραγόντων. Για την σύνδεση του υπολογιστή με το Internet, απαιτείται η εγκατάσταση του λογισμικού TCP/IP. Υπάρχουν αρκετοί τρόποι σύνδεσης στο Internet όπου βάσει των κυριότερων ευρυζωνικών τεχνολογιών μπορούν να ταξινομηθούν σε δύο βασικές κατηγορίες: τους ενσύρματους και τους ασύρματους. Οι σημαντικότεροι και πιο σύγχρονοι από τους αυτούς είναι:

- DSL
- Δορυφορικό σήμα (3/4G)

2.4. Το προγραμματιστικό περιβάλλον

Η εργασία θα βασιστεί στις γλώσσες προγραμματισμού HTML, PHP, Javascript και MySQL. Η κάθε μία θεωρείται ιδανική στο πεδίο εφαρμογών της, αλλά και συνδυαστικά αποτελούν κορυφαία λύση στην κατασκευή δυναμικών ιστοσελίδων.

2.4.1. HTML

2.4.1.1. Η Ιστορία της HTML

Η HTML είναι ένα μέλος μιας οικογένειας που ονομάζεται γλώσσα σήμανσης SGML, το οποίο σημαίνει πρότυπο Generalized Markup Language. Η SGML αναπτύχθηκε από τον Διεθνή Οργανισμό Προτύπων το 1986 για να καθορίσει τη γλώσσα σήμανσης σχεδιαστεί για διάφορους σκοπούς. (Mack & Platt 1997, 29).

Η HTML και το Web για πρώτη φορά σχεδιάστηκαν το 1989 από έναν ερευνητή που ονομάζεται Tim Berners-Lee, που εργαζόταν για το CERN, το Ευρωπαϊκό Εργαστήριο Φυσικής Σωματιδίων στη Γενεύη, Ελβετία. Οι ερευνητές εκεί ανέπτυξαν το πρώτο World Wide Web προγραμμάτων το 1990. Τον Δεκέμβριο του 1994, το World Wide Web Consortium (W3C), το οποίο είναι πλέον υπεύθυνο για τα πρότυπα του HTTP, HTML, και άλλες τεχνολογίες Web, αναδιοργανώθηκε.

Το πρώτο όνομα της γλώσσας ήταν το HTML. Ήταν μια εξαιρετική πρώτη προσπάθεια, και το θεμέλιο για τις επόμενες εκδόσεις. Ο διάδοχος του HTML ονομαζόταν HTML + το 1993. (Mack & Platt 1997, 30). Το 1995, η HTML 2.0 δημοσιεύτηκε ως το πρώτο πρότυπο και ως η πιο δημοφιλής έκδοση του HTML. Έφερε κάποιες πρόσθετες δυνατότητες, όπως το ανέβασμα αρχείων και οι πίνακες. Η επόμενη βασική έκδοση ήταν η HTML 3.2, η οποία δημοσιεύθηκε τον Ιανουάριο του 1997 ως η πρώτη έκδοση που αναπτύχθηκε και τυποποιήθηκε αποκλειστικά από το W3C. Μεταξύ HTML 2.0 και 3.2, υπήρξε μία έκδοση που ονομάζεται HTML 3.0. Λόγω της πολύ μεγάλης αλλαγής από την προηγούμενη έκδοση, αντικαταστάθηκε από την HTML 3.2. (Mack & Platt 1997, 30). Τον Δεκέμβριο του 1997, η HTML 4.0 δημοσιεύτηκε ως σύσταση του W3C. Αρχικά με την κωδική ονομασία Cougar-l. Τον Δεκέμβριο του 1999, η HTML 4.01, δόθηκε στη δημοσιότητα. Και από τα μέσα του 2008, η HTML 4.01, είναι

η πιο πρόσφατη έκδοση της HTML, μέχρι να σταθεροποιηθεί η HTML5. Η HTML5 είναι μια υπό ανάπτυξη γλώσσα σήμανσης για τον Παγκόσμιο Ιστό που όταν ετοιμαστεί θα είναι η επόμενη μεγάλη έκδοση της HTML. Η ομάδα Web Hypertext Application Technology Working Group (WHATWG) άρχισε δουλειά σε αυτή την έκδοση τον Ιούνιο του 2004 με το όνομα Web Applications 1.0. Η HTML5 προορίζεται για αντικατάσταση της HTML 4.01, της XHTML 1.0, και της DOM Level 2 HTML. Ο σκοπός είναι η μείωση της ανάγκης για ιδιόκτητα plug-in και πλούσιες διαδικτυακές εφαρμογές (RIA) όπως το Adobe Flash, το Microsoft Silverlight, το Apache Pivot, και η Sun JavaFX.

2.4.1.2. Ο ορισμός της HTML

HTML σημαίνει Hypertext Markup Language. Η κατάληξη ενός τέτοιου αρχείου είναι html ή htm. Αποτελεί τη γλώσσα σήμανσης με την οποία γράφονται τα έγγραφα του παγκόσμιου ιστού και επιτρέπει τη δημιουργία υπερσυνδέσμων, την συμπλήρωση φορμών κτλ. (Graham 1998, xix).

Ως υπερκείμενο ορίζουμε ένα συνηθισμένο κείμενο το οποίο έχει εμπλουτιστεί με επιπλέον χαρακτηριστικά όπως εικόνες, πολυμέσα και σύνδεσμους σε άλλα έγγραφα. (Mack & Platt 1997, 21).

Σήμανση είναι η διαδικασία κατά την οποία προστίθονται επιπλέον σύμβολα στο κείμενο. Κάθε τέτοιο σύμβολο είναι στην HTML μία εντολή που υπαγορεύει στο φυλλομετρητή πως θα παρουσιάσει το κείμενο. Τα σύμβολα αυτά ονομάζονται elements ή tags. (Mack & Platt 1997, 22)

Αναμφισβήτητα η HTML είναι μια γλώσσα, καθώς έχει το δικό της συντακτικό και κανόνες επικοινωνίας. Στην πραγματικότητα, όμως, δεν αποτελεί μια γλώσσα προγραμματισμού. Ο σκοπός της δεν είναι η δημιουργία κάποιου προγράμματος. (Mack & Platt 1997, 22).

2.4.1.3. Η δομή της HTML

Η HTML ως δομημένη γλώσσα έχει κανόνες σχετικά με την σύνταξη των στοιχείων της. Αυτοί οι κανόνες οδηγούν στη λογική δομή πάνω στο έγγραφο. (Graham 1998, 67).

Η δομή της HTML είναι αυστηρά ιεραρχική με εμφώλευση (Bowers 2007, 37).

Τα έγγραφα της γλώσσας αποτελούνται αποκλειστικά από τα στοιχεία της και η γενική μορφή αποτελείται από τρία στοιχεία:

- Το σύμβολο ανοίγματος < και το σύμβολο κλεισίματος >
- Επιπλέον χαρακτηριστικά μετά το σύμβολο ανοίγματος
- Οποιοδήποτε περιεχόμενο κειμένου ή/και εικόνας ή άλλες φωλιασμένες ετικέτες μεταξύ των συμβόλων ανοίγματος και κλεισίματος (HTML Wikipedia 2011).



Οι κύριες δομικές ετικέτες είναι οι <html>, <head>, and <body>. Οι πληροφορίες σχετικά με το έγγραφο, όπως ο τίτλος, εισάγονται στο <head> όπως και το περιεχόμενο του εγγράφου περιέχεται στο <body> (Bowers 2007, 37).

2.4.2. CSS

2.4.2.1. Η ιστορία της CSS

Η γλώσσα Cascading Style Sheets (CSS) αναπτύχθηκε για να παρέχει καλύτερη εμφάνιση στα HTML έγγραφα (Cascading Style Sheets Wikipedia 2011). Μέχρι την εμφάνιση της CSS δεν υπήρχε ένα συγκεκριμένο στυλ εμφάνισης των εγγράφων με αποτέλεσμα αυτό να επαφίεται στους φυλλομετρητές. (York 2005, 13).

Όταν, όμως, η HTML επεκτάθηκε, η CSS ήρθε να καλύψει ένα ευρύτερο φάσμα στιλιστικών κενών και να ανταποκριθεί στις απαιτήσεις της ανάπτυξης και σχεδιασμού ιστοσελίδων. Αυτή η εξέλιξη έδωσε στον σχεδιαστή περισσότερο έλεγχο πάνω από την εμφάνιση της τοποθεσίας.

```
body {
  margin: 4px;
  border: 3px dotted #;
  font-family: sans-serif;
  color: #000000;
  background-color: #FFFFFF;
}

h1 {
  padding: 5px;
  margin: 10px;
  border: 1px solid #C0C0C0;
  color: #FF0000;
  background-color: #0000FF;
}
```



Το 1994 ο Håkon Wium Lie δημοσίευσε το πρώτο σχέδιο του Cascading Style Sheet HTML. Το σχέδιο αυτό ήταν μια πρόταση για το πώς τα HTML έγγραφα θα μπορούσαν να οριστούν χρησιμοποιώντας απλές δηλώσεις. Ο Bert Bos, υποστήριξε ότι η CSS θα ήταν καλύτερη εφαρμόσιμη ως ένα γενική γλώσσα φύλλο στυλ, που εφαρμόζεται σε περισσότερους από έναν τύπο του εγγράφου. (York 2005, 13).

Επιπλέον, η CSS έρχεται σε τέσσερις διαφορετικές εκδόσεις, κάθε νεότερη έκδοση συνέχεια στο έργο της προηγούμενης. Η πρώτη έκδοση ονομάζεται CSS1 και συστήθηκε από το W3C το 1996. Η δεύτερη εκδοχή, CSS2, συστήθηκε από το W3C το 1998. Η τρίτη έκδοση ονομάστηκε CSS2.1. (York 2005, 11). Η τέταρτη έκδοση ονομάζεται CSS3 και τα πρώτα σχέδια της δημοσιεύθηκαν τον Ιούνιο του 1999 (Cascading Style Sheets Wikipedia 2011).

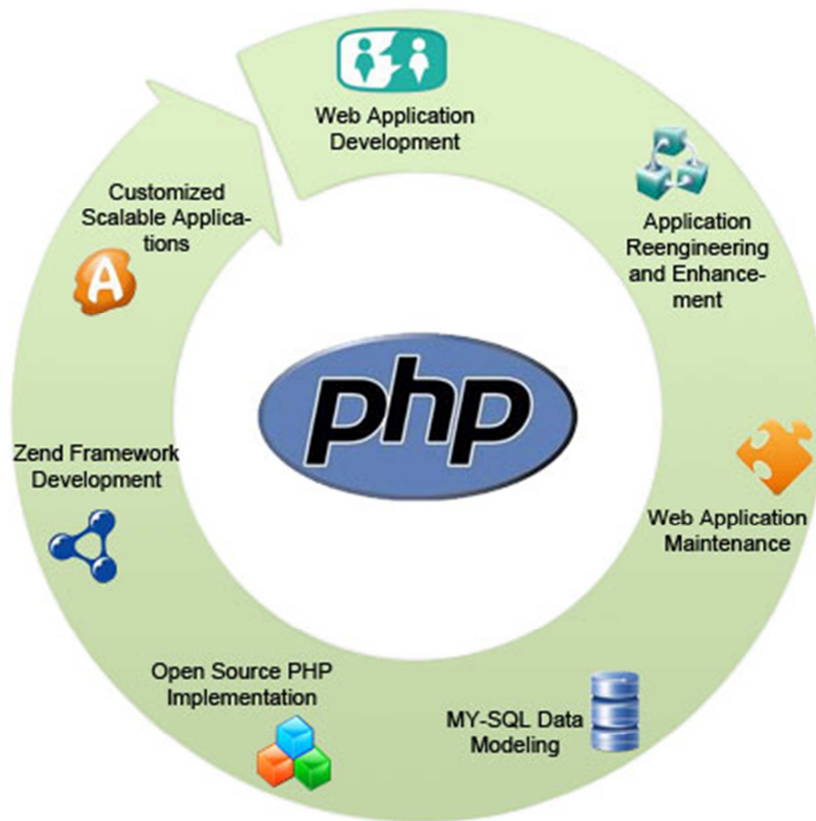
2.4.2.2. Ο ορισμός του CSS

Το Cascading Style Sheet, αναφέρεται ως CSS, είναι μια γλώσσα στυλ που χρησιμοποιείται για να περιγράψει την εμφάνιση και μορφοποίηση ενός εγγράφου γραμμένο σε μια γλώσσα σήμανσης (Cascading Style Sheets Wikipedia 2011). Έτσι, η CSS είναι ένας τρόπος για να ελέγχετε τον τρόπο εμφάνισης των HTML στοιχείων στους φυλλομετρητές (Lemay & Colburn 2006). Αποτελεί μια μορφή διαχωρισμού της παρουσίασης και του περιεχομένου για το σχεδιασμό ιστοσελίδων.

Πιο συχνά, η CSS συνδυάζεται με τη σήμανση γλώσσα HTML ή XHTML. Οι προαναφερθείσες περιέχουν δεδομένα της ιστοσελίδας της, ενώ το έγγραφο CSS που περιέχει πληροφορίες σχετικά με το πως η ιστοσελίδα πρέπει να μοιάσει. (York 2005)

2.4.3. Η γλώσσα PHP

Η PHP είναι μια γλώσσα δέσμης ενεργειών που είναι γραμμένη μέσα σε κώδικα HTML και εκτελείται στην πλευρά του διακομιστή. Έχει πολλά στοιχεία από την γλώσσα C, Java και την Perl. Η κύρια χρήση της PHP είναι η δημιουργία των ιστοσελίδων δυναμικά. Μια ιστοσελίδα που έχει γραφεί σε php ανοίγει από το πρόγραμμα περιήγησης σε πραγματικό χρόνο έχει για επεξεργασία από ένα διακομιστή όπως ο apache. Το πλεονέκτημα του apache είναι όταν ο χρήστης επισκεφθεί την ιστοσελίδα, το πρόγραμμα περιήγησης επικοινωνεί με το διακομιστή μέσω του πρωτοκόλλου http και ως εκ τούτου οι ιστοσελίδες εμφανίζονται στην οθόνη του χρήστη ως html κώδικας. Η ιστορία της PHP ξεκινά από το 1994 όταν ο Rasmus Lerdorf χρησιμοποίησε τη γλώσσα Perl για να δημιουργήσει μια απλή δέσμη ενεργειών με όνομα php.cgi. Ο λόγος για τη δημιουργία της δέσμης ενεργειών ήταν για να περιέχει μια λίστα με στατιστικά στοιχεία για τα πρόσωπα που βλέπουν διαδικτυακά το βιογραφικό του. Στη συνέχεια έδωσε σε φίλους τη δέσμη ενεργειών ώστε να προστεθούν περισσότερες στοιχεία. Στην αρχή το όνομα της γλώσσας ήταν PHP/FI (προσωπική κεντρική σελίδα / μορφή διερμηνείας). Το 1997 πάνω από 50.000 σελίδες web χρησιμοποιούσαν τη γλώσσα αυτή. Η γλώσσα ανανεώθηκε στην έκδοση 2.0 έχοντας ως βάση της και τη γλώσσα C. Οι Andi Gutmans και Zeev Suraski ξαναέγραψαν την γλώσσα PHP, με βάση την PHP / FI 2.0. Έτσι, ήρθε η PHP 3.0. Το 1998 δημιουργήθηκε η 4η έκδοση της PHP και τον Ιούλιο του 2004 γεννήθηκε η έκδοση 5.0. Στις μέρες μας χρησιμοποιείται η 6η έκδοση.



Πλεονεκτήματα:

- Η PHP είναι προσβάσιμη
- Διατίθεται δωρεάν
- Διατίθεται με τεκμηρίωση σε πολλές γλώσσες
- Υπάρχουν πολλές ομάδες υποστήριξης, φόρουμ και ομάδες που υποστηρίζουν PHP
- Υπάρχει πληθώρα διαδικτυακών πληροφοριών όσον αφορά την PHP
- είναι γρήγορο να αναπτυχθεί κώδικας σε PHP
- ένα βασικό PHP script μπορεί να δημιουργηθεί χωρίς μια βαθειά κατανόηση του προγραμματισμού
- Οι προγραμματιστές σε Java, PERL, BASIC και άλλες δημοφιλείς γλώσσες βρίσκουν πολλούς παραλληλισμούς με τη PHP άρα και διευκόλυνση στη μετάβαση σε PHP
- PHP είναι ευέλικτη.
- Εκτελείται σε πολλά διαφορετικά λειτουργικά συστήματα

2.4.4. PHP Sessions

Όταν μια ιστοσελίδα γίνεται όλο και πιο πολύπλοκη, γίνεται πολύπλοκος και ο κώδικας που το υποστηρίζει. Όταν φτάσουμε σε ένα στάδιο όπου η ιστοσελίδα πρέπει να περάσει τα δεδομένα του χρήστη από τη μία σελίδα στην άλλη, ίσως ήρθε η ώρα να αρχίσουμε να σκεφτόμαστε σχετικά με τη χρήση των PHP Sessions.

Μια κανονική ιστοσελίδα HTML δεν θα περάσει τα δεδομένα από τη μία σελίδα στην άλλη. Με άλλα λόγια, όλες οι πληροφορίες θα έχουν ξεχαστεί όταν μια νέα σελίδα φορτωθεί. Αυτό δημιουργεί πρόβλημα για εργασίες όπως ένα καλάθι αγορών, το οποίο απαιτεί τα δεδομένα (επιλεγμένο προϊόν του χρήστη) που πρέπει να θυμόμαστε από τη μία σελίδα στην άλλη. Στην περίπτωση της παρούσας εργασίας όταν το Latitude και το Longitude πρέπει να περάσει στην επόμενη σελίδα, ώστε να φορτωθεί ο χάρτης με αυτά τα δεδομένα ή το username και το password όταν ο χρήστης κάνει login. Ένα PHP Session λύνει αυτό το πρόβλημα επιτρέποντάς μας να αποθηκεύσουμε πληροφορίες του χρήστη στον διακομιστή για να χρησιμοποιηθούν αργότερα (δηλαδή το Latitude και το Longitude κ.λπ.).

Σύνταξη των PHP Sessions

Πριν ξεκινήσουμε την αποθήκευση των στοιχείων του χρήστη, πρέπει να ξεκινήσουμε για πρώτη φορά τα Sessions. Αυτό γίνεται στην αρχή του κώδικα PHP, πριν να σταλεί κάποιο Html ή άλλο κείμενο. Παρακάτω φαίνεται μια απλή δέσμη ενεργειών που θα πρέπει να τοποθετηθεί στην αρχή του κώδικα PHP μας για να ξεκινήσει ένα PHP Session.

```
<?php  
session_start(); // start up your PHP session!  
?>
```

Αυτό το μικροσκοπικό κομμάτι του κώδικα καταχωρεί το Session του χρήστη με τον server και μας επιτρέπει να ξεκινήσουμε την αποθήκευση πληροφοριών του χρήστη και να εκχωρεί μια UID (μοναδικό αναγνωριστικό αριθμό) για τη συνεδρία του συγκεκριμένου χρήστη.

Δήλωση ενός Session

Όταν θέλουμε να αποθηκεύσουμε τα δεδομένα ενός χρήστη σε μια σύνοδο χρησιμοποιούμε το `$_SESSION`. Με αυτό γίνεται η αποθήκευση και η ανάκτηση των δεδομένων του Session.

```
<?php session_start();
```

```
$_SESSION['views'] = 1; // store session data
```

```
echo "Pageviews = ". $_SESSION['views']; //retrieve data ?>
```

Χρήση της λειτουργίας isset της PHP

Τώρα που είμαστε σε θέση να αποθηκεύσουμε και να ανακτήσουμε δεδομένα από το `$_SESSION`, μπορούμε να διερευνήσουμε ορισμένες από τις πραγματικές λειτουργίες των Sessions. Όταν δημιουργείται μια μεταβλητή και αποθηκεύεται σε μια περίοδο, πιθανόν να θέλουμε να τη χρησιμοποιήσουμε στο μέλλον. Ωστόσο, προτού τη χρησιμοποιήσουμε είναι απαραίτητο να ελέγξουμε αν υπάρχει.

Σε αυτό το σημείο είναι όπου η λειτουργία `isset` της PHP θα μας βοηθήσει. Η `isset` είναι μια λειτουργία που θα λαμβάνει κάθε μεταβλητή που θέλουμε να χρησιμοποιήσουμε και ελέγχει για να δει αν έχει τεθεί. Δηλαδή, αν της έχει ήδη ανατεθεί μια τιμή.

Με το προηγούμενο παράδειγμα, μπορούμε να δημιουργήσουμε μια πολύ απλή προβολή σελίδας με τη χρήση της `isset` για να ελέγξουμε αν η μεταβλητή έχει ήδη δημιουργηθεί. Αν έχει μπορούμε να προσάξουμε τον counter μας. Αν δεν υπάρχει μπορούμε να δημιουργήσουμε έναν μετρητή προβολής σελίδας και να του δώσουμε την τιμή ένα. Εδώ είναι ο κώδικας για να γίνει αυτό:

```
<?php session_start();  
if(isset($_SESSION['views'])) $_SESSION['views'] = $_SESSION['views']+ 1;  
else  
$_SESSION['views'] = 1;  
echo "views = ". $_SESSION['views']; ?>
```

Η πρώτη φορά που εκτελείτε αυτήν τη δέσμη ενεργειών σε ένα πρόσφατα ανοιγμένο browser, δήλωση if θα αποτύχει διότι καμία μεταβλητή Session δεν θα έχει αποθηκευτεί ακόμα. 4στόσο, εάν ανανεώσουμε τη σελίδα, η δήλωση if θα ήταν true και ο μετρητής θα αυξάνονται κατά ένα. Κάθε φορά που ξανατρέχουμε αυτό το σενάριο θα βλέπουμε μια αύξηση κατά ένα του counter.

Καθαρισμός και Καταστροφή της Session

Αν και τα δεδομένα Session είναι προσωρινά και δεν χρειάζεται να γίνει διαγραφή, ίσως θελήσουμε να διαγράψουμε μερικά δεδομένα για τις διάφορες εργασίες μας. Όπως για παράδειγμα μετά από μια συναλλαγή, όπου ο χρήστης έχει δώσει τα στοιχεία του, για παράδειγμα τον αριθμό της πιστωτικής του κάρτας.

```
<?php session_start()  
if(isset($_SESSION['cart'])) unset($_SESSION['cart']);  
?>
```

Μπορούμε επίσης να καταστρέψουμε εντελώς το Session εξ ολοκλήρου με την κλήση της λειτουργίας session_destroy.

```
<?php session_start(); session_destroy(); ?>
```

Η καταστροφή του Session θα επαναφέρει τη συνεδρία μας, και όλα τα αποθηκευμένα δεδομένα μας θα χαθούν.

2.4.5. MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που έχει εγκατασταθεί περισσότερες από 11,000,000 φορές. Το όνομα προέρχεται από την κόρη του Modi Videnious. Το πρόγραμμα εκτελείται σε ένα διακομιστή που παρέχει πρόσβαση σε πολλούς χρήστες σε ένα σύνολο βάσεων δεδομένων. Ο πηγαίος κώδικας του προγράμματος είναι διαθέσιμος μέσω του GNU General Public Licence.



Η MySQL παρέχει μια υλοποίηση μιας βάσης δεδομένων SQL κατάλληλη για μικρές και μεσαίες ιστοσελίδες. Η βάση δεδομένων είναι άνευ κόστους και ανοικτού κώδικα με μία εμπορική άδεια διαθέσιμη.

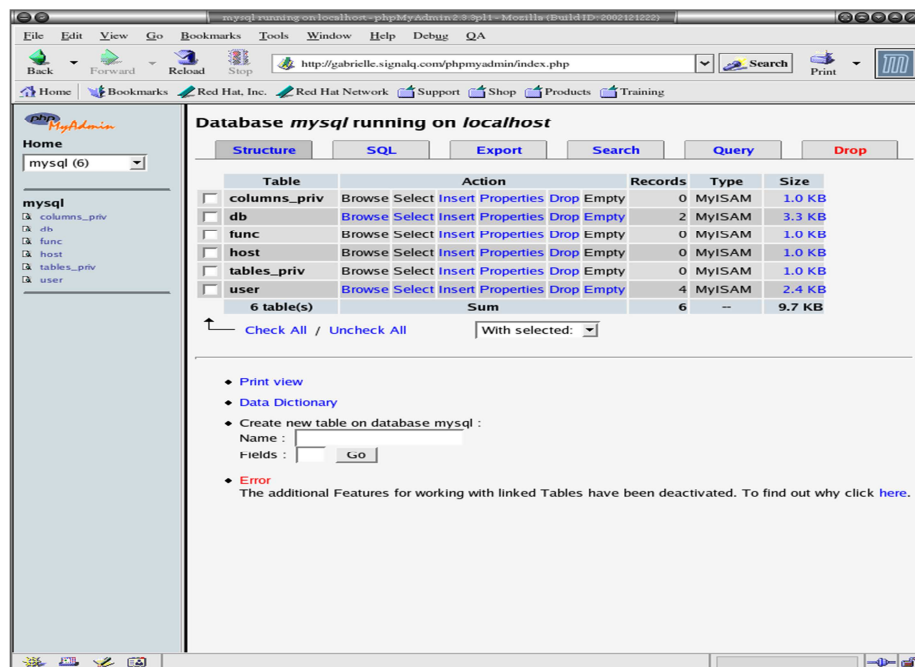
Οι πιο κοινές εφαρμογές για την MySQL συμπεριλαμβάνουν web εφαρμογές βασισμένες σε php και java που επικοινωνούν με μια βάση δεδομένων στο παρασκήνιο.

Η MySQL χρησιμοποιείται συνήθως με δύο διαφορετικές αποθηκευτικές μηχανές. Η μία με όνομα MyISAM, η οποία δεν υποστηρίζει συναλλαγές (transactions) και αποθηκεύει κάθε πίνακα σε ένα σετ τριών αρχείων. Η άλλη ονομάζεται InnoDB και υποστηρίζει συναλλαγές και αποθηκεύει όλα τα δεδομένα σε ένα ενιαίο σύνολο από bytes.

Η MySQL έχει ένα βασικό πλεονέκτημα, δεδομένου ότι είναι δωρεάν: είναι συνήθως διαθέσιμη σε κοινόχρηστο πακέτα φιλοξενίας και μπορεί να ρυθμίζεται εύκολα σε περιβάλλον Linux, Unix και Windows. Σε περίπτωση που μια εφαρμογή web απαιτεί περισσότερο από μία βάση δεδομένων ή απαιτεί εξισορρόπηση φορτίου, δεν είναι δύσκολο να οριστούν παρουσίες της βάσης δεδομένων που απαιτούν μόνο το κόστος του υλικού, σε αντίθεση με εμπορικές βάσεις δεδομένων που θα απαιτούσαν μια ενιαία άδεια για κάθε περίπτωση.

2.5 PHPMYAdmin

Το PhpMyAdmin είναι ένα εργαλείο γραμμένο σε php με το οποίο γίνεται η διαχείριση βάσεων δεδομένων που διατίθενται στο web. Το phpMyAdmin μπορεί να χειρίζεται πλήρως βάσεις δεδομένων, πίνακες, πεδία πινάκων αλλά και ένα ολόκληρο MySQL Server. Υποστηρίζει 54 γλώσσες, μεταξύ των οποίων και τα ελληνικά και είναι λογισμικό ανοιχτού κώδικα.



2.4.6. Δυνατότητες του PhpMyAdmin

Το PhpMyAdmin μπορεί να :

- Δημιουργεί και να διαγράφει βάσεις δεδομένων
- Δημιουργεί, τροποποιεί, διαγράφει, αντιγράφει και μετονομάζει πίνακες
- Κάνει συντήρηση της βάσης
- Προσθέτει, διαγράφει και τροποποιεί πεδία πινάκων
- Εκτελεί ερωτήματα SQL ακόμα και ομαδικά (batch)

- Διαχειρίζεται κλειδιά σε πεδία
- Φορτώνει αρχεία κειμένου σε πίνακες
- Δημιουργεί και διαβάζει πίνακες (που προέρχονται από dump βάσης)
- Εξάγει δεδομένα σε μορφή CVS, XML και LATEX
- Διαχειρίζεται πολλούς διακομιστές
- Διαχειρίζεται τους χρήστες MySQL και τα δικαιώματά τους
- Ελέγχει την αναφορική δραστηριότητα των δεδομένων των MyISAM πινάκων
- Δημιουργεί PDF γραφικών του layout της βάσης δεδομένων
- Εκτελεί αναζητήσεις σε όλη τη βάση δεδομένων ή μέρος αυτής
- Υποστηρίζει πίνακες InnoDB και ξένα κλειδιά
- Υποστηρίζει MySQLi, μια βελτιωμένη επέκταση του MySQL

2.5. Javascript

Η JavaScript αναπτύχθηκε αρχικά από τον Brendan Eich του Netscape με το όνομα Μόκα, η οποία αργότερα μετονομάστηκε σε LiveScript, και τελικά σε JavaScript. LiveScript ήταν το επίσημο όνομα για τη γλώσσα, όταν εισήχθη για πρώτη φορά σε beta εκδόσεις του Netscape Navigator 2,0 τον Σεπτέμβριο του 1995, αλλά μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την Sun Microsystems τον Δεκέμβριο του 1995, όταν προοριζόταν να αναπτυχθεί στη νέα έκδοση του Netscape Navigator 2.0B3.

2.5.1. Τι είναι Javascript

JavaScript, επίσης γνωστή ως ECMAScript, είναι μία πρότυπη object-oriented scripting γλώσσα η οποία είναι δυναμική και έχει πρώτης τάξεως λειτουργίες. Θεωρείται επίσης μια λειτουργική γλώσσα προγραμματισμού, όπως τις Scheme και OCaml.

JavaScript είναι μια εφαρμογή του προτύπου γλώσσας ECMAScript και χρησιμοποιείται κυρίως με τη μορφή client-side Javascript, όπου υλοποιείται ως μέρος ενός web browser, ώστε να παρέχονται ενισχυμένη διεπαφή χρηστών και δυναμικές ιστοσελίδες.

Αυτό επιτρέπει την πρόσβαση μέσω προγραμματισμού στα υπολογιστικά αντικείμενα μέσα σε ένα περιβάλλον υποδοχής.

Η χρήση JavaScript σε εφαρμογές έξω από ιστοσελίδες, όπως για παράδειγμα σε έγγραφα PDF, site-specific προγράμματα περιήγησης και desktop widgets, είναι επίσης σημαντική. Νεότερα και ταχύτερα VMs Javascript και πλαίσια βασισμένα πάνω σε αυτές (κυρίως Node.js), παρουσίασαν επίσης αύξηση της δημοτικότητας του Javascript για server-side web εφαρμογές.

JavaScript χρησιμοποιεί σύνταξη επηρεασμένη από αυτή της C. Πολλά ονόματα και συμβάσεις ονοματολογίας έχουν αντιγραφεί από τη Java, αλλά οι δύο γλώσσες είναι διαφορετικές και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού στο πλαίσιο της JavaScript, λαμβάνονται από Self και Scheme γλώσσες προγραμματισμού.

Είναι μια γλώσσα συγγραφής σεναρίων (scripting language) για την προσθήκη διαδραστικότητας (interactivity) σε ιστοσελίδες. Δημιουργήθηκε από την εταιρεία Netscape με αρχικό όνομα LiveScript και είναι πολύ δημοφιλής στους δημιουργούς ιστοσελίδων καθώς είναι απλή στη σύνταξή της και υποστηρίζεται απ' όλους τους δημοφιλείς φυλλομετρητές. Ανταγωνιστής της είναι η γλώσσα συγγραφής σεναρίων VBScript της εταιρείας Microsoft. Μια άλλη προσέγγιση στον προγραμματισμό από την πλευρά του πελάτη είναι η γλώσσα προγραμματισμού JavaScript. Για τον προγραμματισμό σε JavaScript πρέπει να εισάγετε δηλώσεις προγραμματισμού απευθείας στον κώδικα HTML. Αυτός ο κώδικας αλληλεπιδρά με τον browser, για παράδειγμα παράγει οπτικά εφέ, όπως rollover. Αυτό βρίσκεται σε αντίθεση με τα applet όπου ο κώδικας βρίσκεται στον διακομιστή, υπάρχει μόνο αναφορά για αυτόν στον κώδικα HTML, φορτώνεται στον πελάτη χρησιμοποιώντας την αναφορά και εκτελείται.

2.5.2. Δυνατότητες της Javascript

Με τη JavaScript μπορούμε να φτιάξουμε σενάρια που να εκτελούν αυτόματες εργασίες, για παράδειγμα όταν μια σελίδα του Web ανοίγει ή κλείνει. Επίσης μπορούμε να κάνουμε την JavaScript να εκτελεί ενέργειες ανταποκρινόμενη σε ένα συγκεκριμένο γεγονός. Για παράδειγμα όταν ο χρήστης επιλέγει ένα κουμπί ή ένα σύνδεσμο, όταν εστιάζει από ένα στοιχείο μιας φόρμας σε ένα άλλο στοιχείο της . Τα σενάρια μπορεί να ανοίγουν νέα παράθυρα στον browser και να εμφανίζουν συγκεκριμένα HTML έγγραφα ή να παρουσιάζουν μια σελίδα επιλεγμένη από τον κατάλογο ιστορικού του browser. Μπορεί επίσης να είναι και περίπλοκες δηλαδή ένα σενάριο μπορεί να ελέγχει τα περιεχόμενα μιας φόρμας που θέλει να υποβάλει ένας χρήστης και στη συνέχεια να προειδοποιεί τον χρήστη αν τα δεδομένα είναι λάθος. Το σενάριο μπορεί να ψάξει για πληροφορίες σε μια μικρή βάση δεδομένων ή να κάνει πολύπλοκους υπολογισμούς οικονομικών στοιχείων.

Με την βοήθεια της Java Script μπορούμε να πετύχουμε:

- Πολυμερή έγγραφα με πλαίσια
- Επαναφόρτωση μέρους του παραθύρου
- Δημιουργία έγγραφων με αλληλεπίδραση
- Περισσότερος έλεγχος στην αλληλεπίδραση με το χρήστη
- Έγγραφα με μνήμη
- Ζωντανά έγγραφα
- Μηνύματα που ολισθαίνουν
- Ρολόγια
- Χρονικός μηχανισμός αντίστροφης μέτρησης

- Έγγραφα με αυτόματη ενημέρωση κ.α.

2.5.3. Τρόπος Λειτουργίας της JavaScript

Η JavaScript είναι ενσωματωμένη σ' ένα αρχείο HTML ανάμεσα σε tags SCRIPT. Παρακάτω δίνεται ένα παράδειγμα ενός πολύ απλού προγράμματος:

```
<SCRIPT LANGUAGE = "JavaScript"> document.writeln( "<H3> Hello there");  
</SCRIPT>
```

Ο κώδικας βρίσκεται μέσα στα <SCRIPT> </SCRIPT> και αποτελείται από μία απλή δήλωση που εμφανίζει σε μια ιστοσελίδα το μήνυμα 'Hello there' στο σημείο όπου υπάρχει δήλωση μέσα στο HTML αρχείο. Το μήνυμα εμφανίζεται σε επικεφαλίδα μεγέθους 3, χρησιμοποιώντας το κατάλληλο tag. Το αντικείμενο document αντιπροσωπεύει το αρχείο HTML στο οποίο περικλείεται ο κώδικας. Η μέθοδος writeln του αντικειμένου document γράφει ένα string στον HTML κώδικα που θα εμφανιστεί.

JavaScript μοιάζει πολύ με μια κανονική γλώσσα προγραμματισμού καθώς περιλαμβάνει:

1. Αριθμητικούς τύπους δεδομένων, όπως ακέραιους, καθώς και strings.
2. Ένα μεγάλο αριθμό τελεστών που μπορούν να εφαρμοστούν στους διάφορους τύπους δεδομένων.
3. Δομές ελέγχου, όπως οι if και while.
- *
4. Συναρτήσεις: κομμάτια κώδικα που μπορούν να κληθούν επανειλημμένα και ανεξάρτητα όποτε χρειάζεται να εκτελεστεί η συγκεκριμένη λειτουργία που μέθοδος υλοποιεί.
5. Πίνακες που περιέχουν συλλογές δεδομένων.

Αυτό που την ξεχωρίζει πάντως είναι η ικανότητα, χρησιμοποιώντας την τεχνολογία της δυναμικής HTML (DHTML), να αλληλεπιδρά με τα στοιχεία μίας ιστοσελίδας, για παράδειγμα μπορεί να έχει πρόσβαση στα στοιχεία μιας φόρμας HTML ή ν' αλλάξει την εμφάνιση ενός στοιχείου κειμένου.

3. Εργαλεία Ανάπτυξης Ιστοσελίδας

Για την ανάπτυξη της ιστοσελίδας βασιστήκαμε σε 2 βασικά εργαλεία: Την εφαρμογή στην οποία γράφεται ο κώδικας και σχεδιάζεται η ιστοσελίδα και ένα αυτοματοποιημένο πακέτο εκτέλεσης των δομικών στοιχείων μιας δυναμικής σελίδας, δηλαδή των apache,mysql και phpz

3.1. Εγκατάσταση *EasyPHP*

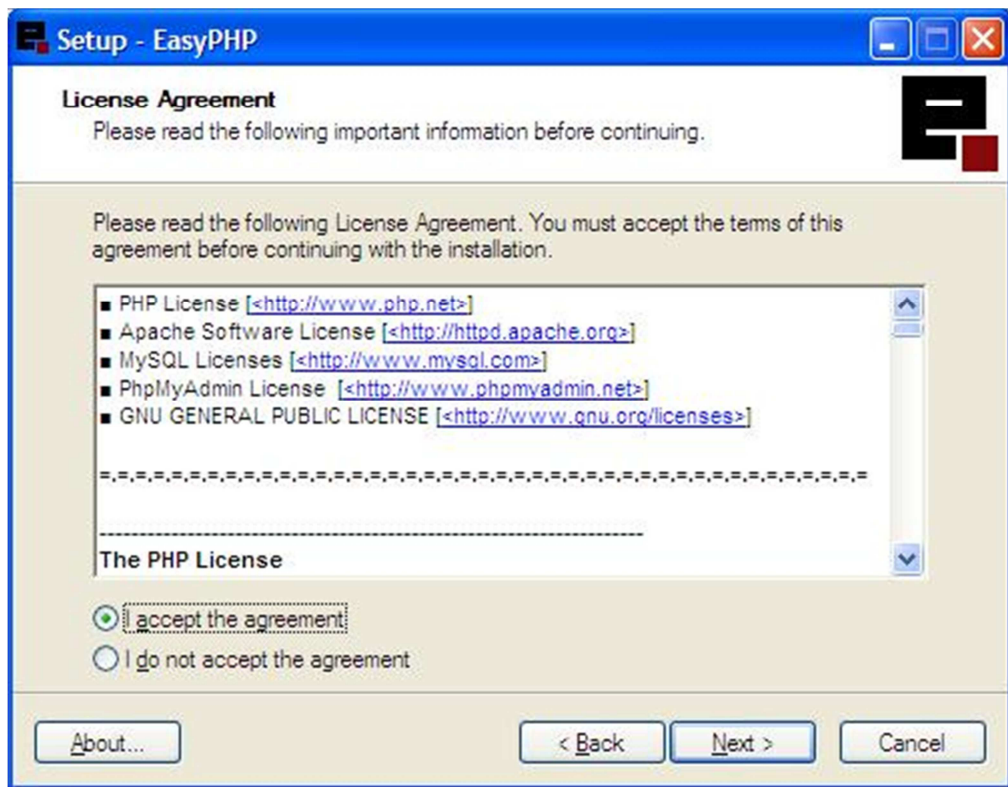
Το EasyPHP είναι ένα πλήρες πακέτο λογισμικού για τα Windows που επιτρέπει την ανάπτυξη PHP και MySQL στον προσωπικό υπολογιστή του χρήστη. Ξεκίνησε το 1999 και το πακέτο περιλαμβάνει ένα διακομιστή Apache, μια βάση δεδομένων MySQL, την επέκταση της PHP καθώς και εργαλεία εύκολης ανάπτυξης για το web site ή τις εφαρμογές μας.

Τρέχουμε το αρχείο easyphp 5.3.0 το οποίο έχουμε κατεβάσει από το site <http://www.easyphp.org/download.php>.

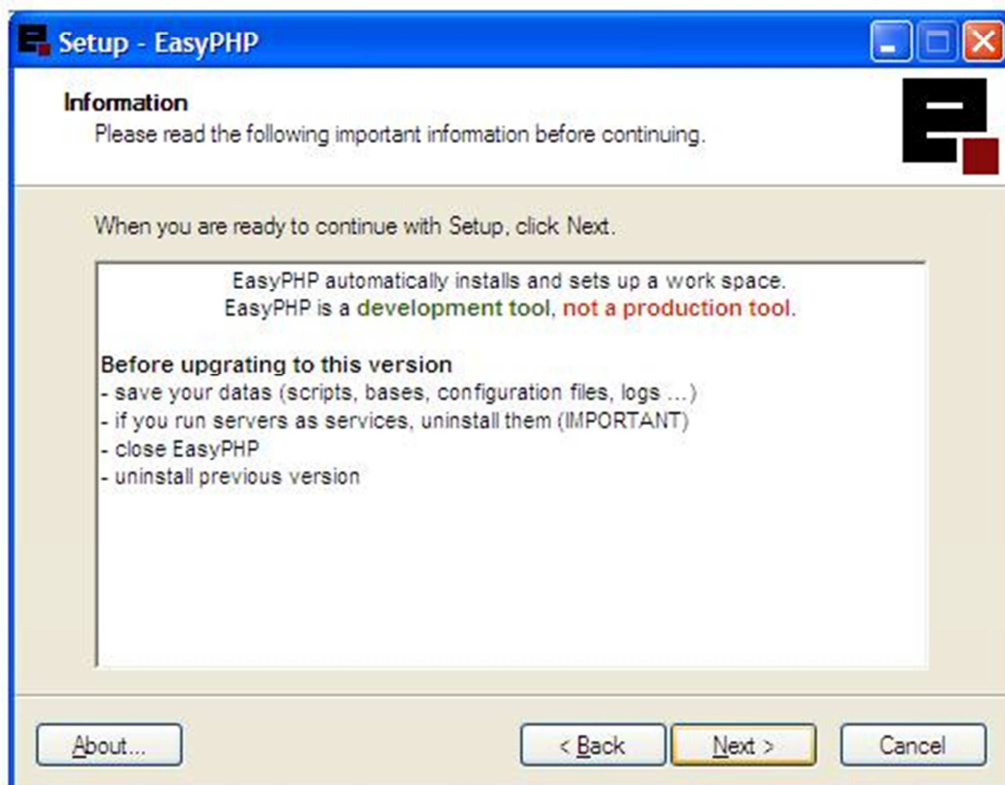
Στο παράθυρο που εμφανίζεται πατάμε το κουμπί next.



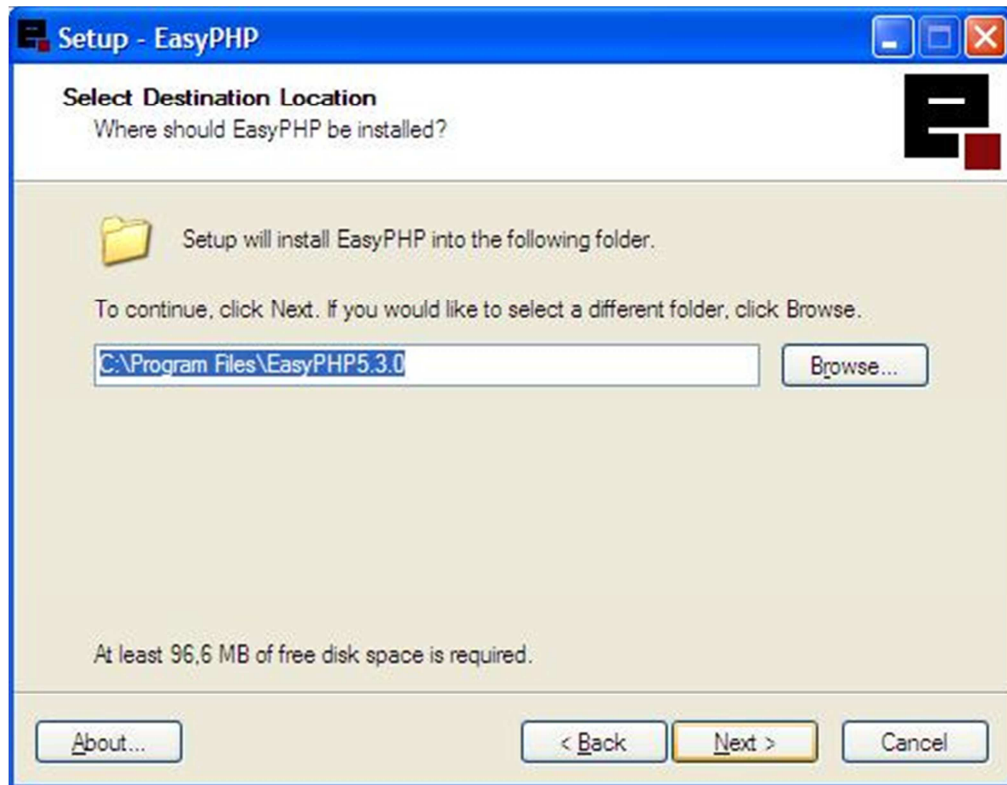
Έπειτα επιλέγουμε την επιλογή “I accept the agreement”, που σημαίνει ότι αποδεχόμαστε τους όρους χρήσης και πατάμε next.



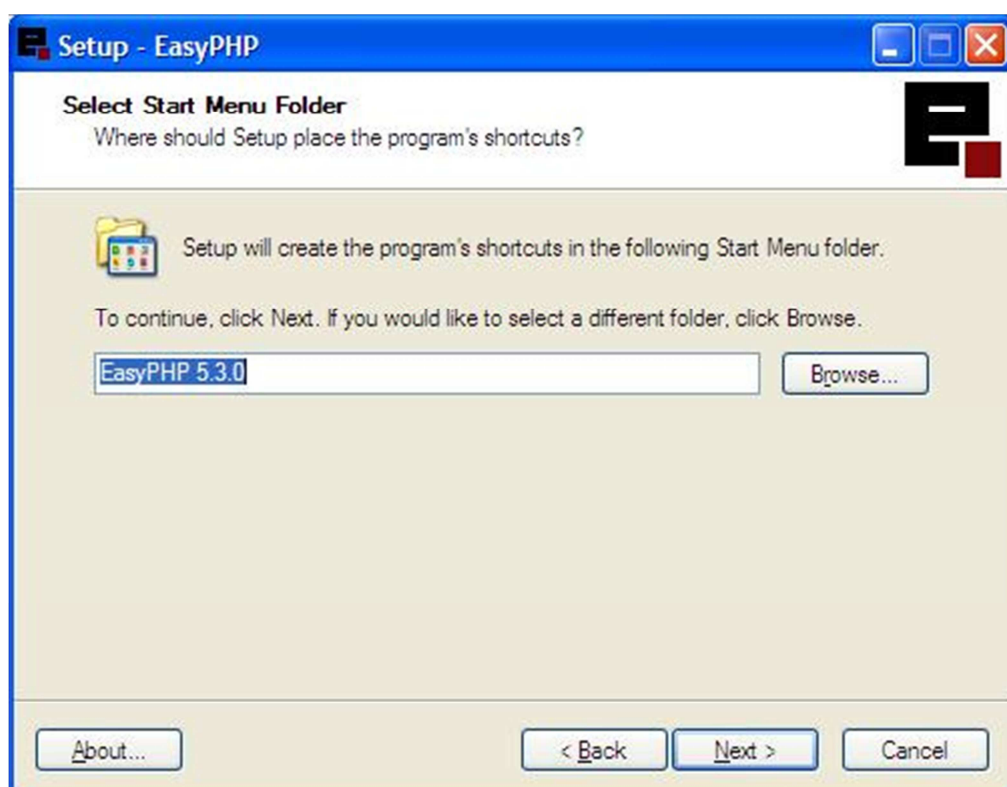
Πατάμε next και στο επόμενο παράθυρο, το οποίο μας ενημερώνει ότι πρέπει να απεγκαταστήσουμε τυχόν άλλους servers πριν την εγκατάσταση.



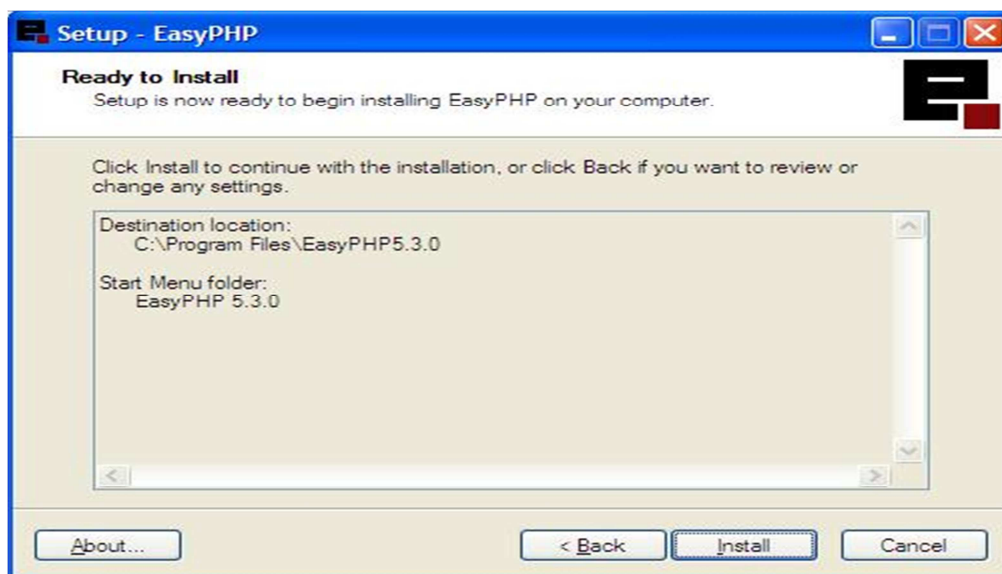
Για να συνεχίσουμε πατάμε next, αφήνοντας τη διαδρομή που έχει επιλέξει η εφαρμογή.



Πατάμε ξανά next και αφήνουμε το όνομα του φακέλου που θα δημιουργηθεί ως έχει.



Στη συνέχεια πατάμε install.



Τέλος πατάμε το κουμπί finish και αφήνουμε τσεκαρισμένες τις επιλογές “Open Help” και “Launch EasyPHP”, τα οποία θα φορτώσουν την εφαρμογή μας και θα ανοίξουν ένα παράθυρο με πληροφορίες που θα μας βοηθήσουν.



Εφόσον έχουμε ολοκληρώσει την εγκατάσταση ανοίγουμε την EasyPHP.



Πατώντας δεξί κλικ στο εικονίδιο της EasyPHP που θα εμφανιστεί στην κάτω δεξιά μπάρα και μετά administrator ανοίγουμε την αρχική σελίδα της EasyPHP. Το ίδιο μπορεί να γίνει πληκτρολογώντας τη διεύθυνση <http://127.0.0.1/home/>

APACHE 2.2.13 LOCAL WEB

Document root : C:\Program Files\EasyPHP5.3.0\www\ [change] [set to default]

- Root
 - joomla_1.5.23-Stable-Full_Package
 - plyxaki

"ALIASES"

- [add]

MYSQL 5.1.37 PHPMYADMIN

MySQL parameters (default MySQL account)

- Host : localhost
- Username : root
- Password : (no password)
- Path to the database root (datadir) : C:\Program Files\EasyPHP5.3.0\mysql\data\

How to change these parameters ?

- Open the MySQL Manager (PhpMyAdmin)
- Select the tab 'Privileges'
- Click on the 'modify' link of the user you want to change the password [edit]
- Follow the instructions

WARNING

If you set a new password for the 'root' user, you will not be able to access PhpMyAdmin anymore. You will need to re-configure PhpMyAdmin :

- Edit "C:\Program Files\EasyPHP5.3.0\phpmyadmin\config.inc.php"
- Set your new password (\$cfgServers[\$i]['password'] = 'newpassword')

PHP 5.3.0 PHPINFO

Paris Tuesday 12th of April 2011 10:01:00 PM

EXTENSIONS

You have 40 extensions loaded [show]

3.2. Adobe Dreamweaver CS6

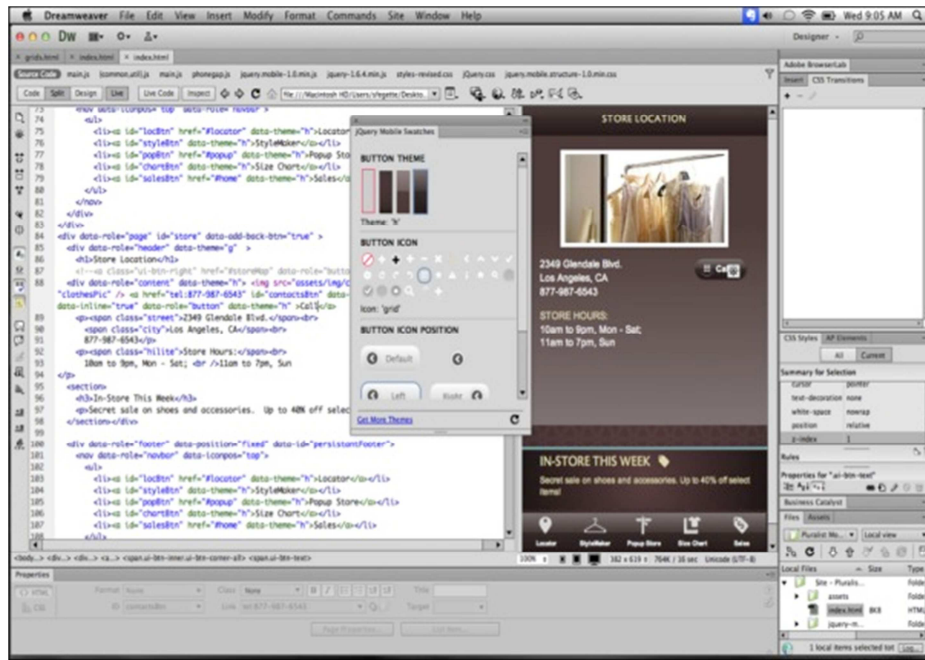
Ένα από τα κορυφαία προγράμματα σχεδίασης ιστοσελίδων αποτελεί το Dreamweaver. Η έκδοση CS6 αποτελεί την πιο πρόσφατη της σειράς. Υποστηρίζει πλέον HTML5 και CSS3. Επίσης, δίνεται η δυνατότητα κατά τη διάρκεια συγγραφής HTML και PHP εμφάνισης τυχόν συντακτικών λαθών και πρότασης διόρθωσης.

.

Απαιτήσεις

Από την στιγμή που το Dreamweaver υλοποιείται σε περιβάλλον Windows υπάρχουν κάποιες ελάχιστες απαιτήσεις για την σταθερή εκτέλεση του και είναι:

- Intel® Pentium® 4 ή AMD Athlon® 64 επεξεργαστής
- Microsoft Windows 7
- 512MB RAM
- 1GB ελεύθερου χώρου στον σκληρό δίσκο
- 1280x800 ανάλυση με 16-bit κάρτα γραφικών
- DVD-ROM δίσκος
- Σύνδεση στο διαδίκτυο για διαδικτυακές υπηρεσίες.



4. Ανάλυση Λογισμικού

Η Ανάλυση λογισμικού αποτελείται από ένα σύνολο δραστηριοτήτων που παράγουν λογισμικό. Υπάρχουν τέσσερις κοινές δραστηριότητες σε όλες αυτές τις διαδικασίες:

- **Προδιαγραφές του λογισμικού.** Οι πελάτες και οι μηχανικοί καθορίζουν όχι μόνο του λογισμικού, αλλά και τους περιορισμούς της λειτουργικότητάς του.
- **Ανάπτυξη λογισμικού.** Σε αυτό το στάδιο, το λογισμικό έχει σχεδιαστεί και αναπτυχθεί με τα κατάλληλα προγράμματα.
- **Επικύρωση λογισμικού.** Οι δοκιμές γίνονται με σκοπό να διασφαλίσει ότι το λογισμικό πληροί τις απαιτήσεις του πελάτη.
- **Εξέλιξη λογισμικού.** Αλλαγές γίνονται σε περίπτωση που νέες απαιτήσεις καταλήξει.

Για το χειρισμό αυτών των δραστηριοτήτων διάφορα μοντέλα productionhave λογισμικού έχουν αναπτυχθεί, τα οποία είναι απλουστευμένη περιγραφή της διαδικασίας παραγωγής λογισμικού. Παραδείγματα τέτοιων μοντέλων είναι το μοντέλο του καταρράκτη, το μοντέλο της εξελικτικής ανάπτυξης και η επαναχρησιμοποιηθεί με βάση το μοντέλο.

Οι απαιτήσεις του συστήματος είναι οι περιγραφές των υπηρεσιών που παρέχονται από το σύστημα και τους περιορισμούς υπό τους οποίους λειτουργεί. Αυτές οι απαιτήσεις να ικανοποιήσετε τις ανάγκες των πελατών για ένα σύστημα που βοηθά να επιλυθούν ορισμένα προβλήματα, όπως ο έλεγχος μιας συσκευής, τοποθετώντας μια παραγγελία ή εύρεση πληροφοριών. Η διαδικασία της αναγνώρισης, ανάλυσης και τεκμηρίωσης, καθώς και τον έλεγχο των υπηρεσιών αυτών και περιορισμούς ονομάζεται μηχανικές απαιτήσεις:

- i. **Λειτουργικές απαιτήσεις:** οι δηλώσεις που ορίζουν ποιες υπηρεσίες θα πρέπει να παρέχονται από το σύστημα, τον τρόπο με τον οποίο θα πρέπει να αντιδράσουμε σε συγκεκριμένες εισόδους και πώς πρέπει να συμπεριφέρονται σε ιδιαίτερες καταστάσεις.
 - 4.1. **Μη λειτουργικές απαιτήσεις:** οι προδιαγραφές, οι οποίες δεν σχετίζονται άμεσα με τις λειτουργίες που παρέχονται από το σύστημα.

Περιλαμβάνουν χρονικούς περιορισμούς, τους περιορισμούς και τους κανόνες διαδικασίας ανάπτυξης. Η χρηστικότητα απαιτήσεις, τις επιδόσεις και την αξιοπιστία αποτελούν παραδείγματα της μη λειτουργικές απαιτήσεις.

4.1. Καταγραφή απαιτήσεων

Σε αυτή την ενότητα θα παρουσιάσουμε τις λειτουργικές απαιτήσεις του συστήματος μας.

Κωδικό όνομα της απαίτησης
R1
Τίτλος απαίτησης
Δημιουργία πελάτη
Περιγραφή απαίτησης
<ul style="list-style-type: none">• Ο χρήστης εισάγει τα στοιχεία του πελάτη• Το πρόγραμμα ελέγχει αν δόθηκαν όλα τα απαιτούμενα στοιχεία• Εμφανίζεται μήνυμα επιτυχίας

Κωδικό όνομα της απαίτησης
R2
Τίτλος απαίτησης
Άνοιγμα λογαριασμού
Περιγραφή απαίτησης
<ul style="list-style-type: none">• Ο χρήστης επιλέγει από τη διαθέσιμη λίστα πελατών το όνομα ενός πελάτη και πατώντας το κουμπί δημιουργείται νέος λογαριασμός με αυτόματη δημιουργία του αριθμού του.

Κωδικό όνομα της απαίτησης
R3
Τίτλος απαίτησης
Κλείσιμο λογαριασμού
Περιγραφή απαίτησης
<ul style="list-style-type: none"> • Ο χρήστης επιλέγει από τη λίστα των διαθέσιμων λογαριασμών το λογαριασμό που επιθυμεί και πατάει το αντίστοιχο κουμπί. • Η επιτυχής διαγραφή του λογαριασμού πιστοποιείται με κατάλληλο μήνυμα.

Κωδικό όνομα της απαίτησης
R4
Τίτλος απαίτησης
Κατάθεση χρημάτων
Περιγραφή απαίτησης
<ul style="list-style-type: none"> • Ο χρήστης επιλέγει το λογαριασμό στον οποίο επιθυμείται να πιστωθούν χρήματα. • Εισάγει το κατάλληλο ποσό. • Ο λογαριασμός πιστώνεται και εμφανίζεται το νέο του υπόλοιπο.

Κωδικό όνομα της απαίτησης
R5
Τίτλος απαίτησης
Ανάληψη χρημάτων

Περιγραφή απαίτησης

- Ο χρήστης επιλέγει το λογαριασμό στον οποίο επιθυμείται να αναληφθούν χρήματα.
- Εισάγει το κατάλληλο ποσό.
- Αν τα χρήματα που ζητούνται υπερβαίνουν το διαθέσιμο υπόλοιπο, τότε εμφανίζεται κατάλληλο μήνυμα αδυναμίας εκτέλεσης της ανάληψης.
- Διαφορετικά, εμφανίζεται το νέο υπόλοιπο.

4.2. Περιπτώσεις χρήσης (Use Cases)

Μία περίπτωση χρήσης είναι μια περιγραφή ενός σεναρίου ή σύνολο σεναρίων που συνδέονται στενά μεταξύ τους. Μία περίπτωση χρήσης προσδιορίζει και περιγράφει την αλληλεπίδραση μεταξύ των παραγόντων και ειδικών λειτουργικών τμημάτων της εφαρμογής (system services). Αυτό είναι ένα παράδειγμα χρήσης που περιγράφει ένα υψηλό επίπεδο, κατά τρόπο συστηματικό και μεθοδικό τρόπο πώς η εφαρμογή χρησιμοποιείται για την εφαρμογή των στόχων αυτών. Πολλές περιπτώσεις τελικών χρήσεων καλύπτουν όλες τις απαιτήσεις και την επιθυμητή συμπεριφορά της εφαρμογής. Του συστήματος χωρίς να μπούμε σε λεπτομέρειες, όπως το πώς η εφαρμογή. Μια περίπτωση χρήσης συνήθως περιλαμβάνει πολλά σενάρια και ενεργοποιεί αποφυγή περιπτώσεων πληροφοριών και περιγραφές των λειτουργιών του συστήματος. Αυτό που προσπαθούμε να επιτύχουμε με τη χρήση περιπτώσεις είναι μία περιεκτική προβολή των λειτουργιών και της συμπεριφοράς του συστήματος λογισμικού.

Η χρήση περιπτώσεις είναι κατασκευασμένη με τυπικό τρόπο και αυτό γίνεται μέσω μιας δομημένης περιγραφής. Τα συστατικά στοιχεία της δομής αυτής είναι:

- όνομα / σύντομη περιγραφή
- χαρακτήρες (παράγοντες): η επικοινωνία με την χρήση.
- Προϋποθέσεις: Τι προϋποθέσεις πρέπει να πληρούνται από την αρχή της υπόθεσης δήλωση.

- MetaTreaty : Ποιες προϋποθέσεις πρέπει να πληρούνται κατά το τέλος της υπόθεσης δήλωση.
- Διέγερση: ένα γεγονός που προκαλεί η χρήση υπόθεση.
- Οι Σχέσεις: σχετίζεται, περιλαμβάνει, επέκταση και γενίκευση.
- Ομαλή ροή Από συμβάντα
- Συστατικά της ροής γεγονότα
- Εναλλακτικές ροές - Εξαιρέσεις: αποκλίσεις από το κύριο σενάριο για την επιτυχία.

Χρησιμοποιήστε σενάρια πρέπει να πληρούν συγκεκριμένα ποιοτικά χαρακτηριστικά που καθορίζουν την πληρότητα του σεναρίου.

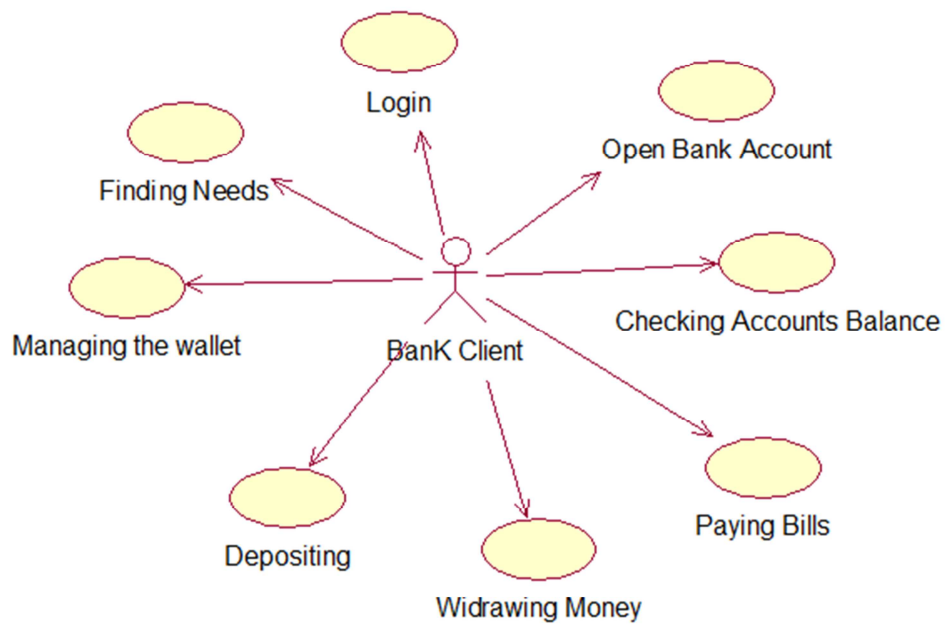
- Ένα σενάριο για να είναι πλήρης θα πρέπει να καταγράφονται σε μορφή που επιτρέπει την περαιτέρω ανάλυση.
- Το σενάριο πρέπει να καθορίσετε το στόχο ή σκοπό εξυπηρετούνται και μια σαφή αναφορά στους ενδιαφερόμενους φορείς (χρήστες και τα συστατικά του συστήματος).
- Το σενάριο θα αποτελείται από επιμέρους δραστηριότητες (βήματα) που όταν εκτελούνται επιτευχθεί το επιθυμητό αποτέλεσμα.
- Το σενάριο θα πρέπει να προσδιορίζει τη ροή των δραστηριοτήτων (μέτρα) που πρέπει να εκτελεστούν.
- Για κάθε δραστηριότητα ή βήμα θα πρέπει να είναι ή μπορεί να διαπιστωθεί όρια που εξασφαλίζουν την έναρξη και το τέλος της δραστηριότητας, και σε γενικές γραμμές το σενάριο.
- Ένα σενάριο θα πρέπει να προσδιορίζει τις προϋποθέσεις ότι η πλήρης και τυχόν εξαιρέσεις είναι συνήθως εναλλακτικά σενάρια.

Όσον αφορά την τεχνική εγγραφής ένα σενάριο ανάλογα με το βαθμό ωριμότητας μπορεί να καταγραφεί με πολλούς διαφορετικούς τρόπους, όπως οι εξής:

- Κείμενο / αφήγηση γεγονότων.
- Διατεταγμένη ακολουθία των βημάτων.
- Μορφή δομημένη περιγραφή.
- Διαχωρισμένη αφήγηση.
- Περιπτώσεις χρήσης Model.

- Διάγραμμα script Vord.
- Ακολουθία μοντέλο της UML, κλπ.

Επίσης, είναι δυνατόν το σενάριο αρχικά καταγράφεται από τον γενικό τύπο (π.χ. κείμενο) πρέπει να χαρακτηριστεί με τη χρήση άλλης μορφής (π.χ. διαχωρίζεται αλληλουχία των σταδίων) προκειμένου να καταστεί δυνατή η καλύτερη κατανόηση των επιμέρους στοιχείων.



5. Σχεδιασμός Βάσεων Δεδομένων

5.1. Μοντέλο Οντοτήτων - Συσχετίσεων (E-R Model)

Η οντότητα-σχέση μοντέλο που αναπτύχθηκε από τον P. Chen (1976), ως ένας τρόπος επικοινωνίας μεταξύ των χρηστών και σχεδιαστών και χρησιμοποιείται στο πρώτο στάδιο του σχεδιασμού ενός συστήματος πληροφοριών για την ανάλυση των απαιτήσεων. Σκοπός του είναι να περιγράψει τις απαραίτητες πληροφορίες που θα αποθηκεύονται στη βάση δεδομένων ή με τον απαιτούμενο τύπο. Η μοντελοποίηση των δεδομένων είναι η περιγραφή των όρων και βωτων σχέσεων τους σε μια συγκεκριμένη περιοχή ενδιαφέροντος. Στην περίπτωση του σχεδιασμού ενός συστήματος πληροφοριών βασίζεται σε μια βάση δεδομένων, το εννοιολογικό μοντέλο δεδομένων χαρτογραφείται σε προχωρημένο στάδιο σε ένα λογικό μοντέλο δεδομένων, όπως το σχεσιακό μοντέλο δεδομένων. Αυτό το στάδιο είναι συνήθως ονομάζεται λογικό στάδιο του σχεδιασμού. Στη συνέχεια, κατά τη διάρκεια του φυσικού σχεδιασμού το λογικό μοντέλο αντιστοιχίζεται σε ένα φυσικό μοντέλο.

Τα Διαγράμματα οντοτήτων-συσχετίσεων έχουν ως κύριο χαρακτηριστικό τους την οντότητα.. Η οικονομική οντότητα είναι μια αναπαράσταση ενός αυτόνομη ύπαρξη με σωματική ή θεωρητική ύπαρξη, κάθε έννοια αντικείμενο, κατάσταση ή αφηρημένη που υπάρχει. Σε αυτή την περίπτωση, η οικονομική οντότητα είναι ένας χρήστης του δικτυακού τύπου (ένας άνθρωπος με ένα όνομα, τα χαρακτηριστικά, κλπ.), αλλά και ένα προϊόν του καταστήματος.

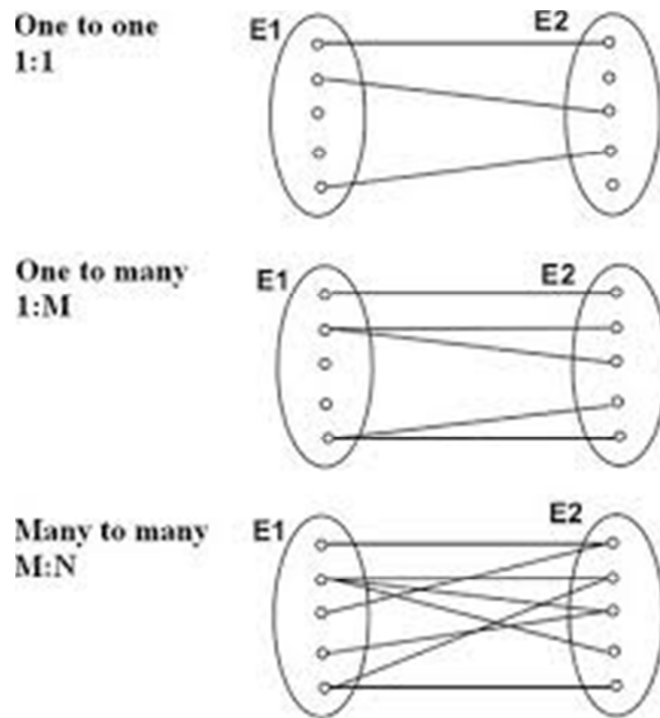
Τα δεδομένα θα αποθηκεύονται σε μία οντότητα μοντέλο οντοτήτων-συσχετίσεων, που αντιστοιχούν στις καταχωρήσεις στο φυσικό επίπεδο μιας βάσης δεδομένων. Έτσι, η οικονομική οντότητα θα πρέπει να έχει ένα όνομα, και κάποια στοιχεία που καθορίζουν τα χαρακτηριστικά της οντότητας. Κάθε οντότητα περιγράφει μια ειδική κατηγορία των εγγραφών που μπορούν να ομαδοποιηθούν. Έτσι, όταν γίνεται αναφορά σε στοιχεία που αποτελούν μέρος της οντότητας "Χρήστης", για παράδειγμα, θα πρέπει να υπάρχουν συγκεκριμένες περιοχές που αντιστοιχούν στα χαρακτηριστικά της εν λόγω οντότητας.

Χαρακτηριστικά μπορούν να είναι απλές ή σύνθετες. Οι απλές ιδιότητες δέχονται τιμές από μια σειρά. Για παράδειγμα, η τιμή του χαρακτηριστικού είναι μόνο χαρακτηριστικό της οντότητας "προϊόν" με τιμές στο σύνολο των θετικών πραγματικών αριθμών. Τα σύνθετα χαρακτηριστικά αποτελούνται από μια σειρά από χαρακτηριστικά που περιγράφουν ένα σύνολο το καθεστώ. Όπως για παράδειγμα, η διεύθυνση χαρακτηριστικό αποτελείται από τον αριθμό χαρακτηριστικών, οδός, πόλη, ταχυδρομικό κώδικα. Χαρακτηριστικά χωρίζονται σε μονότιμες και πολλαπλών τιμών, απλή ή σύνθετη, κενό (null), ή δημιουργούνται. Βασικά για μια οντότητα του διαγράμματος οντοτήτων-συσχετίσεων είναι το σύνολο των κατηγορημάτων της οντότητας που είναι μοναδικά για κάθε εγγραφή και πλήρως διακρίνει από τους άλλους. Κάθε οντότητα έχει ένα μοναδικό χαρακτηριστικό και είναι το πρωτεύον κλειδί για την οντότητα. Με λίγα λόγια, τα βασικά είναι το σύνολο των χαρακτηριστικών μιας οντότητας που προσδιορίζει μοναδικά την οντότητα, η οποία προσδιορίζει κάθε ρεκόρ και τον διαφοροποιεί από τους άλλους.

Οι φορείς που συνδέονται με τις σχέσεις. Η σχέση τους φαίνεται να Database. Η σχέση μεταξύ των οντοτήτων ορίζει ένα σύνολο συσχετίσεων μεταξύ των καταχωρίσεων αυτών των οντοτήτων. Η συνολική συσχέτιση (set σχέση) είναι μια συλλογή των συσχετίσεων του ίδιου τύπου και τον αριθμό των φορέων που εμπλέκονται σε μια σχέση είναι ο βαθμός.

Οι σχέσεις χωρίζονται στις ακόλουθες κατηγορίες

- Η αναλογία του 1 προς 1, πράγμα που σημαίνει ότι οι φορείς που εμπλέκονται στη σχέση είναι μια αναλογία ένα προς ένα. Κάθε εγγραφή που σχετίζεται με την πρώτη από τη δεύτερη και αντιστρόφως.
- Η ISA σχέση, είναι απλά μια εξειδικευμένη μορφή μιας σχέσης 1 προς 1, πράγμα που σημαίνει ότι ο φορέας μπορεί να ταξινομηθεί σε δύο (ή περισσότερα) υπο-φορείς.
- Η αναλογία 1 προς N σημαίνει ότι κάθε εγγραφή της πρώτης οντότητας σχετίζεται με πολλές εγγραφές στον δεύτερο φορέα, αλλά κάθε εγγραφή της δεύτερης οντότητας συνδέεται με μία μόνο εγγραφή της πρώτης οντότητας.
- Η αναλογία του N στο M σημαίνει ότι πολλές εγγραφές της κάθε οντότητας που σχετίζεται με πολλές εγγραφές στον άλλο.



Ο πιο συνηθισμένος τρόπος για να περιγράψει την οντότητα-σχέση του μοντέλου είναι η χρήση των Διαγράμματα οντοτήτων-συσχετίσεων (ER διαγράμματα). Τα διαγράμματα αυτά παρέχουν ένα κατανοητό και απλό τρόπο για να περιγράψει τη δομή των δεδομένων της βάσης δεδομένων, οντότητες, τις σχέσεις, και τα χαρακτηριστικά. Τα βασικά γεωμετρικά σχήματα που συνθέτουν μια οντότητα-σχέση διάγραμμα είναι: ορθογώνιο που αντιπροσωπεύει ένα σύνολο οντοτήτων, το διαμάντι αντιπροσωπεύει ένα σύνολο σχέσεων? Έλλειψης αντιπροσωπεύει ένα χαρακτηριστικό, ενώ το τμήμα δείχνει τα χαρακτηριστικά ενός συνόλου των φορέων και τη συμμετοχή ενός σύνολο οντοτήτων ένα σύνολο συσχετισμών.

Ασθενής οντότητες είναι οντότητες που δεν έχουν καμία εγγενή ιδιότητα, δηλαδή εξαρτάται από άλλους φορείς να είναι. Μια οικονομική οντότητα E1 ορίζεται ως αδύναμη, εάν η ύπαρξη του άλλου εξαρτάται από το E2 οντότητας μέσω μιας σχέσης R. Η E2 χαρακτηρίζεται ως ισχυρή οντότητα. Οι ασθενείς οντότητες έχουν τα δικά τους χαρακτηριστικά τον εντοπισμό τους, αλλά μπορεί να έχουν μερικά αναγνωριστικά (μερική αναγνώρισης), για παράδειγμα, ένα σύνολο χαρακτηριστικών που καθορίζουν μοναδικά στιγμιότυπα αδύναμη οντότητα που συνδέεται με την ίδιο στιγμιότυπο με αυτό της ισχυρής οντότητας. Τα αναγνωριστικά χαρακτηριστικά για τις αδύναμες οντότητες που δημιουργούνται με τον εντοπισμό χαρακτηριστικά των ισχυρών φορέων και μερικά αναγνωριστικά της αδύναμης οικονομικής οντότητας.

Κάθε ασθενής ή ασθενής οντότητα είναι υπαρξιακά υποτελής, αλλά όχι το αντίστροφο.

Η βάση οντότητας και του σχεσιακού μοντέλου εμπλουτίζεται με την προσθήκη συμπληρωματικών στοιχείων (J. Hoffer, κλπ. 2011:125):

- *Ειδίκευση-γενίκευση*: όταν ένα σύνολο οντοτήτων (κλάση) μπορεί να περιέχουν υποσύνολα των φορέων (υποκατηγορίες), ενδεχομένως με διαφορετικά χαρακτηριστικά μεταξύ τους. Οι κατηγορίες και υποκατηγορίες που συνδέονται με την ιεραρχία ISA. Η εξειδίκευση και η γενίκευση είναι διπλή έννοιες.
- *Characteristics when Κληρονομικότητα τα χαρακτηριστικά όλων των φορέων* ότι ένα επίπεδο κληρονομεί από την οντότητα που το επόμενο χαμηλότερο επίπεδο του ISA ιεραρχία.
- *Συνυπολογισμός, μια ένωση που εμπλέκονται σε μια άλλη σχέση*

Εν συντομία ER μοντέλο που βασίζεται στην αντίληψη ότι ο πραγματικός κόσμος αποτελείται από πρόσωπα με τα χαρακτηριστικά και ενώσεις (σχέσεις) μεταξύ των οντοτήτων. Οι συσχετίσεις με τα ίδια χαρακτηριστικά που δημιουργούν ένα σύνολο συσχετισμών. Οι φορείς που διακρίνονται από το πρωτεύον κλειδί. Χαρακτηρίζεται μια συσχέτιση ο λόγος πληθικότητας 1:1, 1: N, N: 1, N: M. Η συμμετοχή μιας οντότητας σε ένα σύνολο συσχετίσεων μπορεί να είναι υποχρεωτική ή προαιρετική. Αναπτύχθηκε για να διευκολύνει τη σχεδίαση μιας βάσης δεδομένων, επιτρέποντας τον ορισμό ενός σχήματος που αντιπροσωπεύει τη συνολική λογική δομή της βάσης δεδομένων.

5.2. Σχεσιακό Μοντέλο

Ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS) αποτελείται από ένα σύνολο δεδομένων και προγραμμάτων για την πρόσβαση σε αυτά τα δεδομένα. Το σύνολο δεδομένων ονομάζεται βάση δεδομένων. Στόχος του ΣΔΒΔ είναι η εύκολη και γρήγορη χρήση και την ανάκτηση των δεδομένων Η διαχείριση των δεδομένων περιλαμβάνει.:

- Ορίστε δομές για την αποθήκευση δεδομένων
- Καθορισμός μεθόδων για τη διαχείριση των δεδομένων

Ο ορισμός της δομής της βάσης δεδομένων βασίζεται σε ένα μοντέλο δεδομένων που καθορίζει το πώς τα δεδομένα που περιγράφονται, τις σχέσεις τους, τη σημασία τους και τους περιορισμούς σε αυτά δεδομένα. Το σχεσιακό μοντέλο περιγράφει τη βάση δεδομένων και να οργανώνει τις εγγραφές με βάση τις σχέσεις. Για το λόγο αυτό, μια βάση δεδομένων σχεδιάστηκε με βάση το σχεσιακό μοντέλο, μπορεί εύκολα να πραγματοποιηθεί με ένα μοντέλο οντότητα-σχέση. Το σχεσιακό μοντέλο δεδομένων αντιπροσωπεύει τα δεδομένα και τις σχέσεις τους ως ένα σύνολο πινάκων. Κάθε πίνακας σε μια σχεσιακή βάση δεδομένων αποτελείται από στήλες με μοναδικά ονόματα που αντιστοιχούν σε τιμές (ή τα χαρακτηριστικά για την οντότητα - μοντέλο σχέση) και από σειρές ή πλειάδες του πίνακα αντιπροσωπεύει μια σχέση μεταξύ ενός συνόλου τιμών.

Η SQL (Structured Query Language) είναι σήμερα το πιο ευρέως χρησιμοποιούμενη γλώσσα για τη διαχείριση σχεσιακή βάση δεδομένων και παρέχει δυνατότητες για:

- ορισμό, τη διαγραφή και την αλλαγή στους πίνακες και τα κλειδιά
- ερωτήσεις γραπτώς (ερωτήματα),
- εισάγετε, να διαγράψετε και να αλλάξει τα δεδομένα
- ορισμό απόψεων σχετικά με τα δεδομένα,
- καθορισμός των δικαιωμάτων πρόσβασης,
- τον έλεγχο της ακεραιότητας των δεδομένων,
- συναλλαγές κοινού ελέγχου

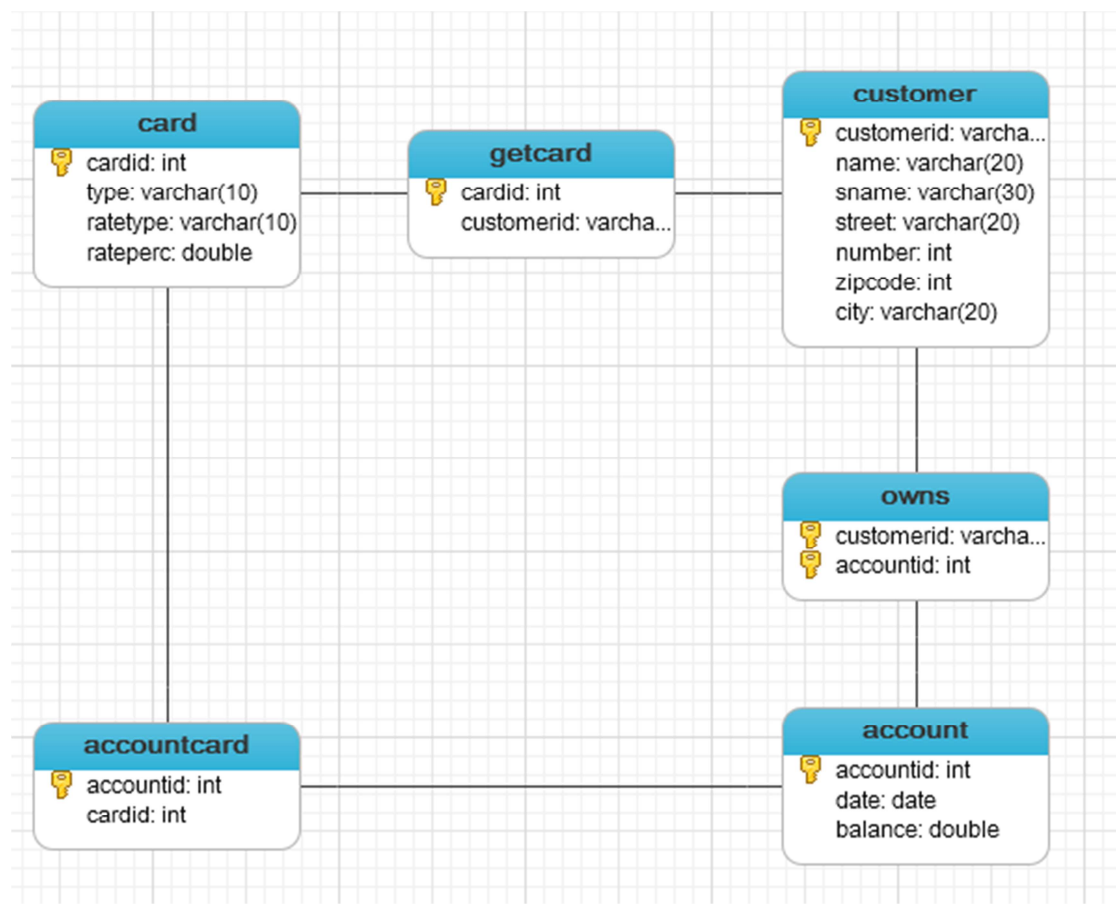
Το σχεσιακό μοντέλο προτάθηκε από τον J. Codd το 1970 που ήταν η βάση των βιομηχανικών συστημάτων στην IBM, DB2 και SQL / DS. Το σχεσιακό μοντέλο βασίζεται στην μαθηματική έννοια της σχέσης. Ο Codd, και άλλοι διεύρυνε την έννοια να ισχύουν για το σχεδιασμό μιας βάσης δεδομένων, εκμεταλλεύτηκε τη δύναμη της μαθηματικής αφαίρεσης και την εκφραστικότητα του μαθηματικού συμβολισμού και δημιούργησε ένα απλό αλλά δυναμικό μοντέλο για τις βάσεις δεδομένων. Η αφαίρεση αυτής της προσέγγισης έχει το πλεονέκτημα ότι η επίδραση είναι γενική. Το μοντέλο χρησιμοποιεί απλές δομές δεδομένων και ένα μικρό σύνολο

των τελεστών πολύ ισχυρό για να επιτευχθεί επεξεργασία των δεδομένων που κυμαίνονται από απλό στο σύνθετο.

5.3. Ο σχεδιασμός της βάσης δεδομένων

5.3.1. Ζωγραφίζοντας το e-r διάγραμμα

Λαμβάνοντας υπόψη όλες τις απαιτήσεις και τη γενική ανάλυση του συστήματος το διάγραμμα σχέσης οντοτήτων κάτω έχει συνταχθεί. Σε αυτό το διάγραμμα το πρωτεύον κλειδί έχει επισημανθεί, όπως υπογραμμίζεται.



5.3.2. Μετατροπή ER σε πίνακες βάσης δεδομένων

Μετατροπή του ER διαγράμματος σε σχήμα σχέσης καταλήγουμε στους παρακάτω πίνακες:

Customer	Τα δεδομένα για όλα τα προϊόντα
Card	Τα στοιχεία κάθε κατηγορία προϊόντων
Account	Τα δεδομένα για κάθε πελάτη
Owns	Τα δεδομένα για κάθε σχόλιο
getCard	Τα δεδομένα για κάθε δημοπρασία.
AccountCard	Όλες οι λεπτομέρειες για τις ολοκληρωμένες δημοπρασίες.

Πίνακας Customer

Το πρωτεύον κλειδί σε αυτό το τραπέζι είναι ο αριθμός ταυτότητας γιατί κάθε φυσικό πρόσωπο έχει μια διαφορετική ταυτότητα. Τα άλλα πεδία είναι το όνομα του πελάτη, το επίθετο, η διεύθυνση και το τηλέφωνο του.

Πίνακας Card

Κάθε πιστωτική/χρεωστική κάρτα έχει έναν μοναδικό αριθμό που χρησιμοποιείται ως πρωτεύον κλειδί. Εν συνεχεία, άλλα πεδία αποτελούν ο τύπος της κάρτας, ο τύπος του επιτοκίου και το ποσοστό επιτοκίου.

Πίνακας Account

Ο πίνακας αυτός περιέχει τον αριθμό λογαριασμού, το υπόλοιπό και την ημερομηνία ανοίγματος του.

Πίνακας Owns

Τα πεδία αυτού του πίνακα είναι η ταυτότητα του πελάτη και ο αριθμός λογαριασμού. Ο πίνακας αυτός υπάρχει διότι η σχέση πελάτης – λογαριασμός είναι σχέση πολλά προς πολλά.

Πίνακας GetCard

Ο πίνακας αυτό συνδέει την κάρτα με τον πελάτη και περιέχει τα πρωτεύοντα κλειδιά των δύο πινάκων.

Πίνακας AccountCard

Ο πίνακας αυτό συνδέει την κάρτα με το λογαριασμό και περιέχει τα πρωτεύοντα κλειδιά των δύο πινάκων.

5.3.3. Η εφαρμογή της βάσης δεδομένων

Η εφαρμογή της βάσης δεδομένων λαμβάνει χώρα απευθείας στην phpMyAdmin.

Παρακάτω δίνεται η SQL κώδικα για τη δημιουργία του πελάτη πίνακα:

```
CREATE TABLE `account` (  
  `accountid` int(20) NOT NULL,  
  `date` date NOT NULL,  
  `balance` double(20,0) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`accountid`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `accountcard` (  
  `accountid` int(20) NOT NULL,  
  `cardid` int(20) NOT NULL,  
  PRIMARY KEY (`accountid`),  
  KEY `a2` (`cardid`),  
  CONSTRAINT `a1` FOREIGN KEY (`accountid`) REFERENCES `account`  
  (`accountid`) ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `a2` FOREIGN KEY (`cardid`) REFERENCES `card` (`cardid`) ON  
  DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `card` (  
  `cardid` int(20) NOT NULL,  
  `type` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT  
  NULL,
```

```

`ratetype` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci
DEFAULT NULL,
`rateperc` double DEFAULT NULL,
PRIMARY KEY (`cardid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `customer` (
`customerid` varchar(10) NOT NULL,
`name` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
`sname` varchar(30) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
`street` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
`number` int(3) NOT NULL,
`zipcode` int(5) NOT NULL,
`city` varchar(20) NOT NULL,
PRIMARY KEY (`customerid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `getcard` (
`cardid` int(20) NOT NULL,
`customerid` varchar(10) NOT NULL,
PRIMARY KEY (`cardid`),
KEY `c2` (`customerid`),
CONSTRAINT `c1` FOREIGN KEY (`cardid`) REFERENCES `card` (`cardid`) ON
DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `c2` FOREIGN KEY (`customerid`) REFERENCES `customer`
(`customerid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `owns` (
`customerid` varchar(10) NOT NULL,

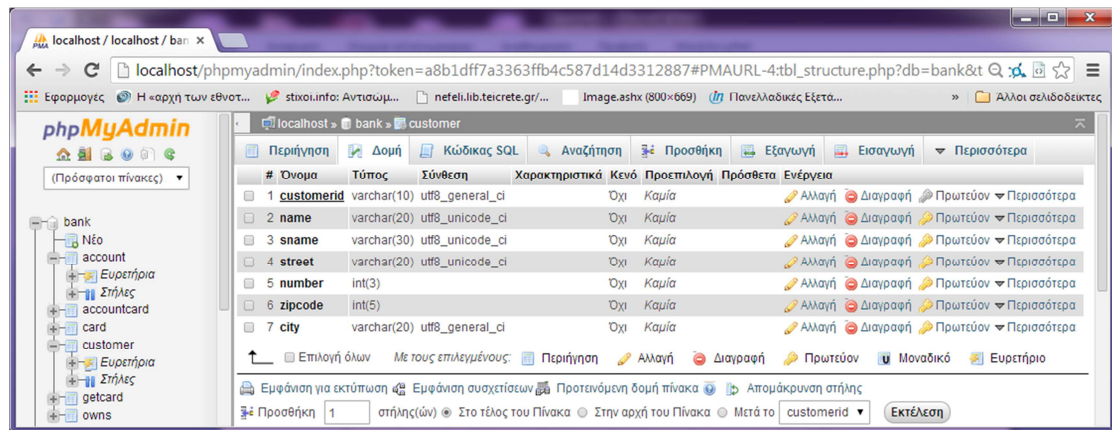
```

```

`accountid` int(20) NOT NULL,
PRIMARY KEY (`customerid`,`accountid`),
KEY `a` (`accountid`),
CONSTRAINT `a` FOREIGN KEY (`accountid`) REFERENCES `account`
(`accountid`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `c` FOREIGN KEY (`customerid`) REFERENCES `customer`
(`customerid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Η παρακάτω εικόνα δείχνει τη δομή του νέου πίνακα. Αντί της σύνταξης SQL κώδικα, phpmyadmin έχει το χαρακτηριστικό της τη δημιουργία ενός πίνακα αυτόματα.



6. Εγχειρίδιο χρήσης και ανάλυση κώδικα

Σε αυτό το κεφάλαιο υπάρχει αναλυτική περιγραφή των βημάτων που ακολουθούνται από τους χρήστες αυτής της ιστοσελίδας.

6.1. Ανάλυση του κώδικα

Το παρόν κεφάλαιο υπογραμμίζει τα πιο σημαντικά τμήματα του κώδικα. Πολλές λειτουργίες της τοποθεσίας απαιτούν βαθιά γνώση της εκμάθησης δεξιοτήτων. Είναι πολύ χρήσιμο να υπάρχει ο πηγαίος κώδικας σε περισσότερα από μία ενότητες, ειδικά όταν κώδικα δεν βασίζεται σε ένα σύστημα διαχείρισης περιεχομένου σαν το Joomla.

6.1.1. Εγγραφή πελάτη

Η εγγραφή επιτυγχάνεται μέσα από ένα αρχείο με όνομα index.html Αυτό το αρχείο χρησιμοποιείται για την εγγραφή ενός νέου πελάτη. Αρχικά, ο χρήστης έχει γεμίσει όλα τα πεδία και πατάει το κουμπί Εγγραφή. Αν δεν έχει γεμίσει τα πεδία τότε εμφανίζεται ένα μήνυμα λάθους. Ο έλεγχος για το λάθος γίνεται από ένα σενάριο javascript.

```
function formvalidation() {  
    var id = document.getElementById('cid').value;  
    var name = document.getElementById('name').value;  
    var sname = document.getElementById('sname').value;  
    var zip = document.getElementById('zip').value;  
    var street = document.getElementById('street').value;  
    var city = document.getElementById('city').value;  
    var number = document.getElementById('number').value;
```

```
var flag = true;

if (cid=="") {
    alert("Enter identity number");
    document.form.cid.focus();
    flag = false;
}
else if (name=="") {
    alert("Enter first name");
    document.form.name.focus();
    flag = false;
}
else if (sname=="") {
    alert("Enter last name");
    document.form.sname.focus();
    flag = false;
}
else if (street=="") {
    alert("Enter street");
    document.form.street.focus();
    flag = false;
}
else if (number=="") {
    alert("Enter street number");
    document.form.street.focus();
    flag = false;
}
else if (zip=="") {
    alert("Enter zip code");
    document.form.zip.focus();
    flag = false;
}
else if (city=="") {
```

```

        alert("Enter city name");
        document.form.city.focus();
        flag = false;
    }

return flag;

```

Όταν τα πεδία είναι συμπληρωμένα διαβιβάζονται σε ένα αρχείο. Τότε το πρόγραμμα συνδέεται με τη βάση (μέσω αρχείου db_con.php) και εκτελεί το επερώτημα που έχει τεθεί. Αν το όνομα χρήστη που παρέχεται από το χρήστη υπάρχει ήδη στη βάση δεδομένων τότε εμφανίζεται εκ νέου μήνυμα λάθους.

```

include("db_con.php");
mysql_query("SET NAMES 'utf8'");
mysql_query("SET CHARACTER SET 'utf8_unicode_ci'");
mysql_set_charset('utf8', $con);

$cid = $_POST['cid'];
$name = $_POST['name'];
$sname = $_POST['sname'];
$zip = $_POST['zip'];
$street = $_POST['street'];
$city = $_POST['city'];
$number = $_POST['number'];
$q = mysql_query("SELECT customerid FROM customer where customerid =
'".$cid."'") or die("MySQL ERROR: ".mysql_error());
$num = mysql_num_rows($q);
if($num > 0)
{

```

```

        echo 'Customer has already been recorded!<br />Please <a
href="index.html">try again</a>...';
    }
    else
    {
        mysql_query("insert          into          customer
(customerid,name,sname,street,number,zipcode,city)          VALUES
('.$cid.','.$name.','.$sname.','.$street.','.$number.','.$zip.','.$city.')") or
die("MySQL ERROR: ".mysql_error());
        echo 'Customer registration is complete!<br />Please <a
href="index.html">Back to main menu</a>...';
    }
    mysql_close();

```

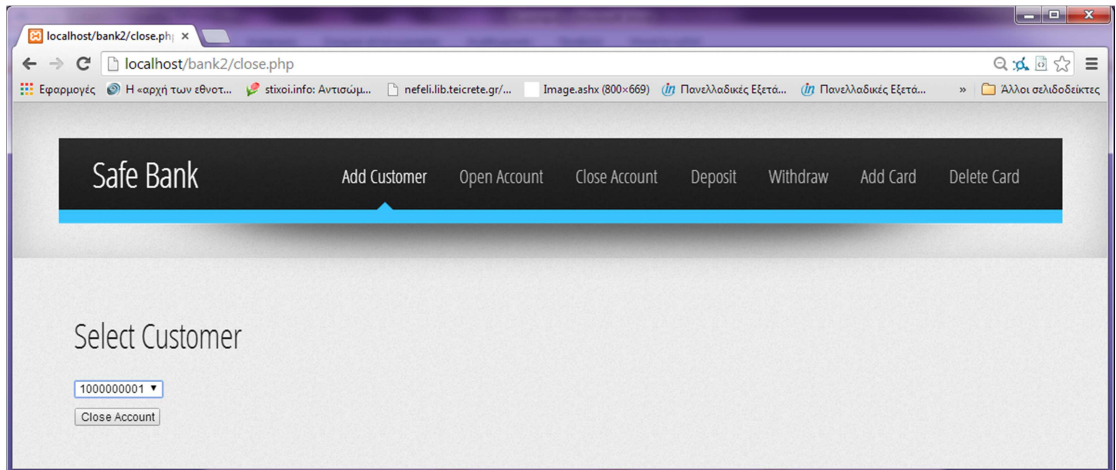
Αυτό επιτυγχάνεται καταμετρώντας τις εγγραφές που ικανοποιούν το ερώτημα. Αν βρείτε έστω και ένα, τότε εμφανίζεται το μήνυμα λάθους μέσω ενός συνδέσμου που επιστρέφει το χρήστη πίσω στην αρχική σελίδα. Διαφορετικά, ο νέος χρήστης εγγράφεται στον πίνακα.

6.1.2. Σύνδεση με τη βάση δεδομένων

Σε αρκετά τμήματα του κώδικα και αρχεία, το αρχείο db_con.php είναι εισαγόμενα (περιλαμβάνεται). Στο αρχείο αυτό γίνεται η σύνδεση με την βάση δεδομένων. Η εντολή \$con = mysql_connect (localhost,"root",""); ορίζει τις παραμέτρους (server, username, password) με το οποίο θα επιτευχθεί η σύνδεση. Εάν η σύνδεση είναι επιτυχής, τότε υπάρχει επικοινωνία με την επιλεγόμενη βάση δεδομένων. Η εντολή για την επιλογή της βάσης είναι mysql_select_db ("bank", \$con).

6.1.3. Διαγραφή λογαριασμού

Για τη διαγραφή ενός λογαριασμού απαιτείται πρώτα απ' όλα η εμφάνιση όλων των διαθέσιμων λογαριασμών που υπάρχουν



Ο κώδικας για να επιτευχθεί η εμφάνιση των διαθέσιμων λογαριασμών είναι

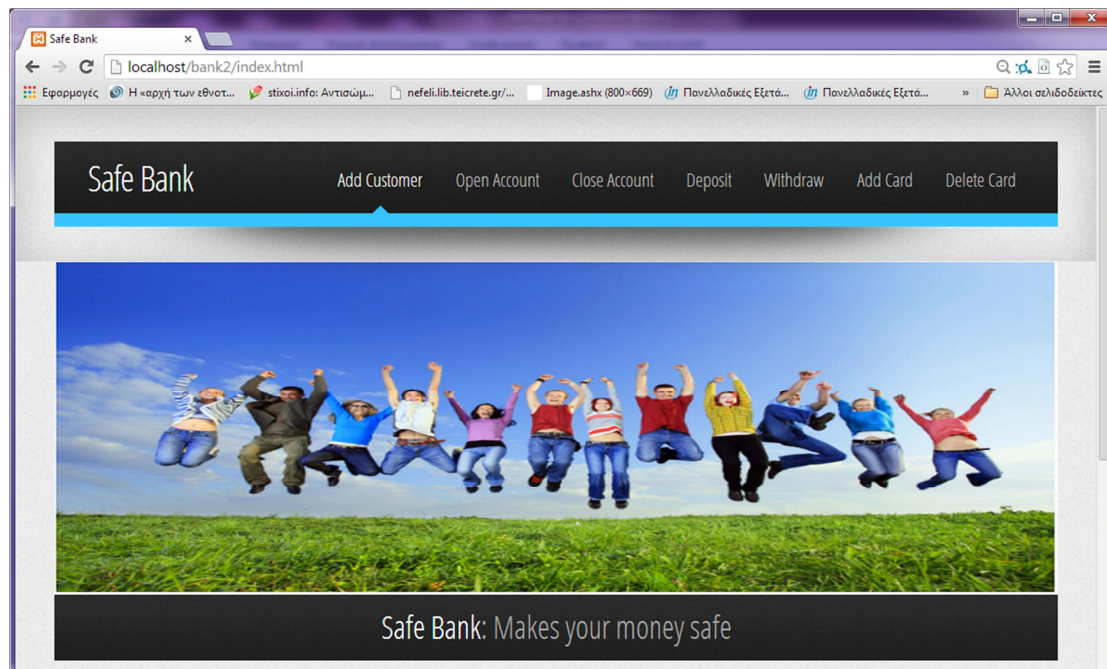
```
<h2>Select Customer</h2>
<form method="post" action="closeaccount.php">
<?php
    include("db_con.php");
    mysql_query("SET NAMES 'utf8'");
    mysql_query("SET CHARACTER SET 'utf8_unicode_ci'");
    mysql_set_charset('utf8', $con);
    $result = mysql_query("select * FROM account") or die("A problem
occured");
    echo "<select name='account'>\n";
    while($row = mysql_fetch_array($result))
    {
        $res = $row[accountid];
        echo "<option value='$row[accountid]'>$res</option>\n";
        echo "</br>";
    }?>
</select><br />
    <input type="submit" value="Close Account"><br>
```

Όταν επιλεγθεί ο διαθέσιμος λογαριασμός τότε τα στοιχεία του αποστέλλονται στο αρχείο closeaccount.php όπου και γίνεται διαγραφή από τους πίνακες account και owns.

```
$value = $_POST['account'];  
  
$q = mysql_query("DELETE FROM account WHERE accountid='".$value."'") or  
die("MySQL ERROR: ".mysql_error());  
  
$q = mysql_query("DELETE FROM owns WHERE accountid='".$value."'") or  
die("MySQL ERROR: ".mysql_error());  
  
echo "Account has been deleted";
```

6.1.4. Εμφάνιση ιστοσελίδας

Για την εμφάνιση της ιστοσελίδας χρησιμοποιήθηκε η γλώσσα CSS. Η αρχική σελίδα αποφασίστηκε να είναι η πιο ελκυστική σε εμφάνιση.



Μέρος του κώδικα για να εμφανιστεί έτσι η σελίδα είναι το εξής:

```
/* Basic */

*,*:before,*:after{-moz-box-sizing:content-box;-webkit-box-sizing:content-
box;-o-box-sizing:content-box;-ms-box-sizing:content-box;box-sizing:content-box}

body
{
    background: #f0f0f0 url('images/bg1.png');
    font-family: 'Open Sans Condensed', sans-serif;
    font-size: 14pt;
    line-height: 1.75em;
    color: #4c4c4c;
}

h1,h2,h3,h4
{
    font-family: 'Open Sans Condensed', sans-serif;
    color: #1d1d1d;
}

h2, h3
{
    font-size: 1.6em;
    letter-spacing: -1px;
    margin: 0 0 0.75em 0;
}

strong
{
    color: #1d1d1d;
}

a
```

```

    {
        color: #478903;
        text-decoration: underline;
    }

a:hover
{
    text-decoration: none;
}

/* Reusable */

.button
{
    display: inline-block;
    background: #0893cf;
    box-shadow: inset 0px 0px 0px 1px #48d3FF, 0px 2px 3px 0px
    rgba(0,0,0,0.25);
    border: solid 1px #0893cf;
    border-radius: 6px;
    padding: 0.45em 1.1em 0.5em 1.1em;
    color: #fff;
    font-size: 0.8em;
    text-decoration: none;
    text-shadow: -1px -1px 0 rgba(0,0,0,0.5);
    background-image: -moz-linear-gradient(top, #38C3FF, #0893cf);
    background-image: -webkit-linear-gradient(top, #38C3FF, #0893cf);
    background-image: -webkit-gradient(linear, 0% 0%, 0% 100%,
    from(#38C3FF), to(#0893cf));
    background-image: -ms-linear-gradient(top, #38C3FF, #0893cf);
    background-image: -o-linear-gradient(top, #38C3FF, #0893cf);
    background-image: linear-gradient(top, #38C3FF, #0893cf);
    outline: 0;
}

```



```
        white-space: nowrap;
    }

    .button:hover
    {
        background: #28a3df;
        background-image: -moz-linear-gradient(top, #58d3FF, #28a3df);
        background-image: -webkit-linear-gradient(top, #58d3FF, #28a3df);
        background-image: -webkit-gradient(linear, 0% 0%, 0% 100%,
from(#58d3FF), to(#28a3df));
        background-image: -ms-linear-gradient(top, #58d3FF, #28a3df);
        background-image: -o-linear-gradient(top, #58d3FF, #28a3df);
        background-image: linear-gradient(top, #58d3FF, #28a3df);
    }

    .button:active
    {
        background: #38C3FF;
        background-image: -moz-linear-gradient(top, #0893cf, #38C3FF);
        background-image: -webkit-linear-gradient(top, #0893cf, #38C3FF);
        background-image: -webkit-gradient(linear, 0% 0%, 0% 100%,
from(#0893cf), to(#38C3FF));
        background-image: -ms-linear-gradient(top, #0893cf, #38C3FF);
        background-image: -o-linear-gradient(top, #0893cf, #38C3FF);
        background-image: linear-gradient(top, #0893cf, #38C3FF);
    }
}
```

7. Έλεγχος Συστήματος

Στο κεφάλαιο αυτό εξετάζεται η λειτουργικότητα του συστήματος μέσω διαφόρων τεστ που γίνονται.

7.1. Τα στάδια του ελέγχου

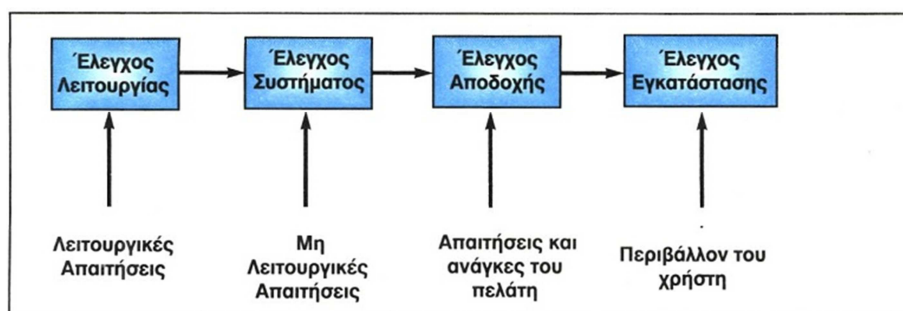
Ο τελικός έλεγχος αποτελείται από τέσσερα στάδια τα οποία περιγράφονται αναλυτικά στις ενότητες που ακολουθούν:

Έλεγχος λειτουργίας (function testing): Εξετάζει αν το ολοκληρωμένο σύστημα πληρεί τις λειτουργικές απαιτήσεις, όπως αυτές προσδιορίζονται στο έγγραφο απαιτήσεων λογισμικού.

Έλεγχος συστήματος (system testing): Συγκρίνει το σύστημα με τις μη λειτουργικές του απαιτήσεις. Ο έλεγχος αυτός περιλαμβάνει θέματα ασφάλειας, συμβατότητας, όγκου, χρηστικότητας, ακρίβειας και γενικά περιορισμούς κάτω από τους οποίους το σύστημα πρέπει να εκτελεί τις λειτουργίες του.

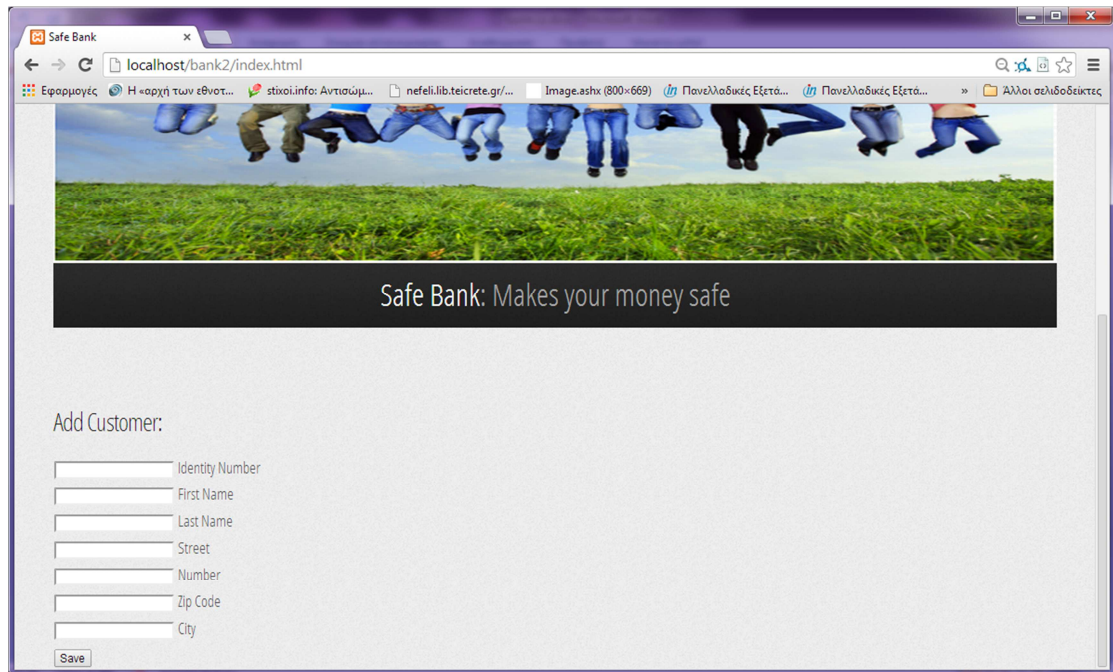
Έλεγχος αποδοχής (acceptance testing): Συγκρίνει το λογισμικό με τις αρχικές απαιτήσεις και τις τρέχουσες ανάγκες του τελικού χρήστη.

Έλεγχος εγκατάστασης (installation testing): Εντοπίζει τα σφάλματα που γίνονται κατά τη διάρκεια της εγκατάστασης του λογισμικού.

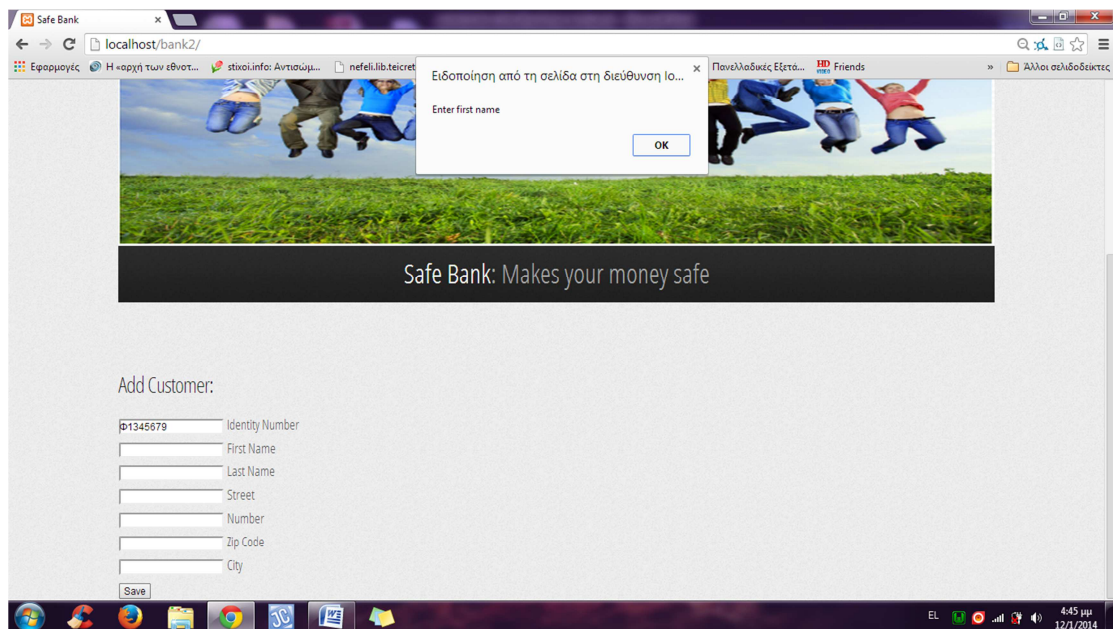


7.2. Έλεγχος λειτουργίας εγγραφής πελάτη

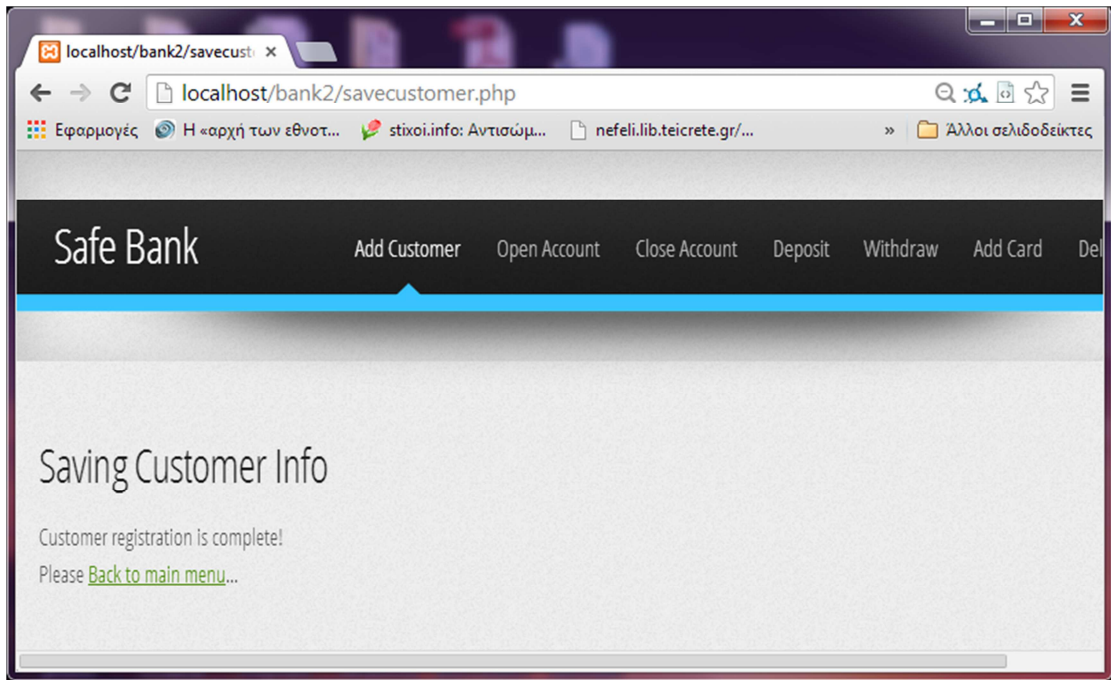
Ο χρήστης για να ξεκινήσει τη διαδικασία εγγραφής επιλέγει από το κεντρικό μενού την επιλογή Add Customer.



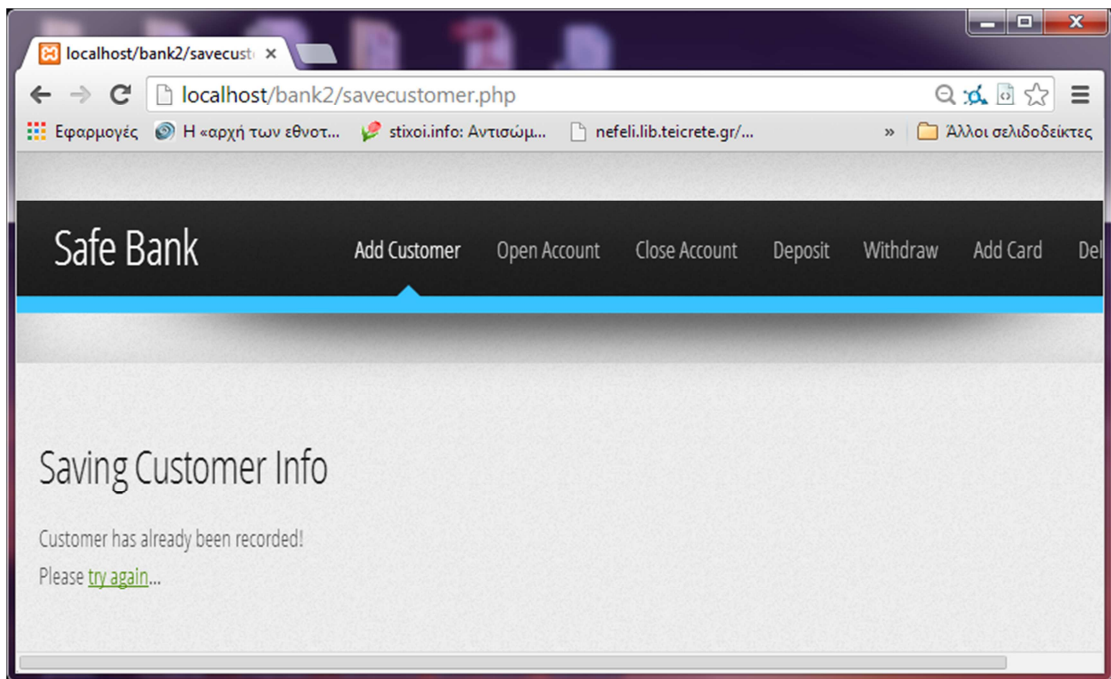
Δοκιμάζουμε να μην συμπληρώσουμε όλα τα πεδία και έχουμε το εξής αποτέλεσμα:



Αντίθετα, αν τα συμπληρώσουμε όλα

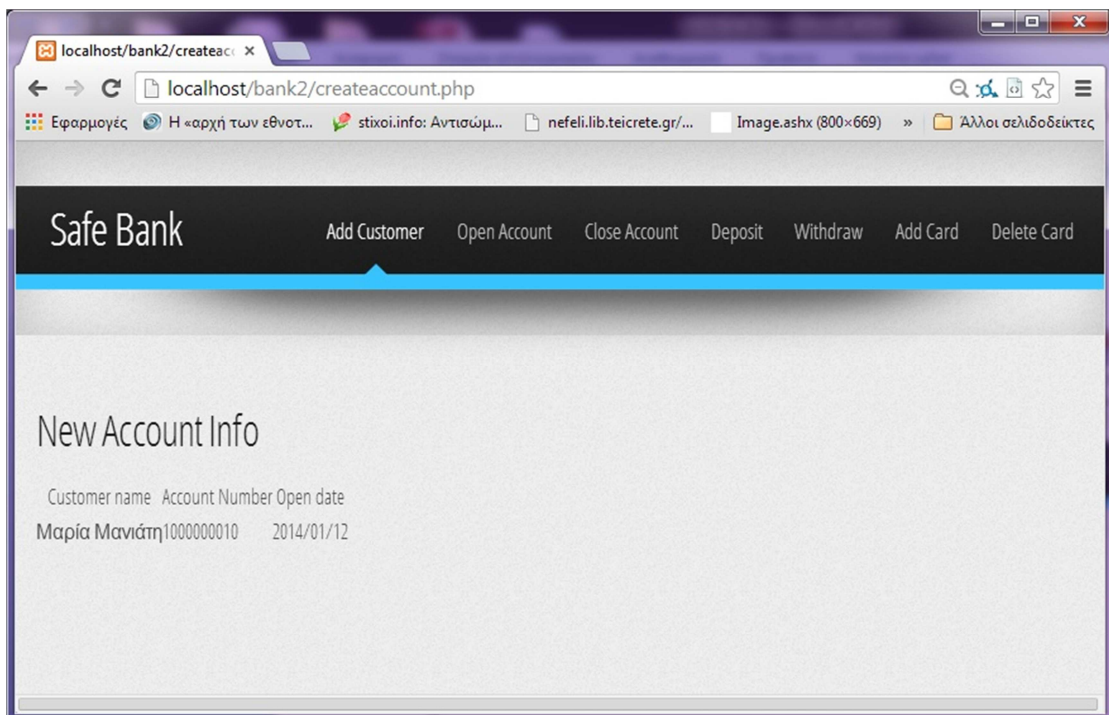
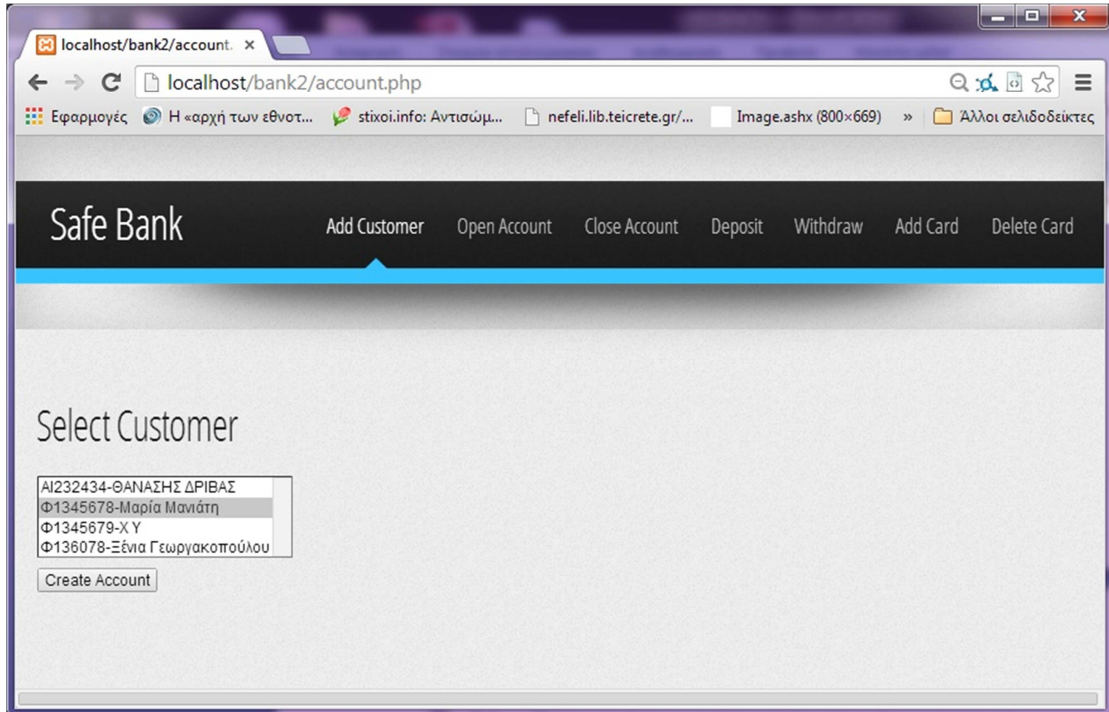


Αν, όμως, δώσουμε στοιχεία ενός υπάρχοντος πελάτη:



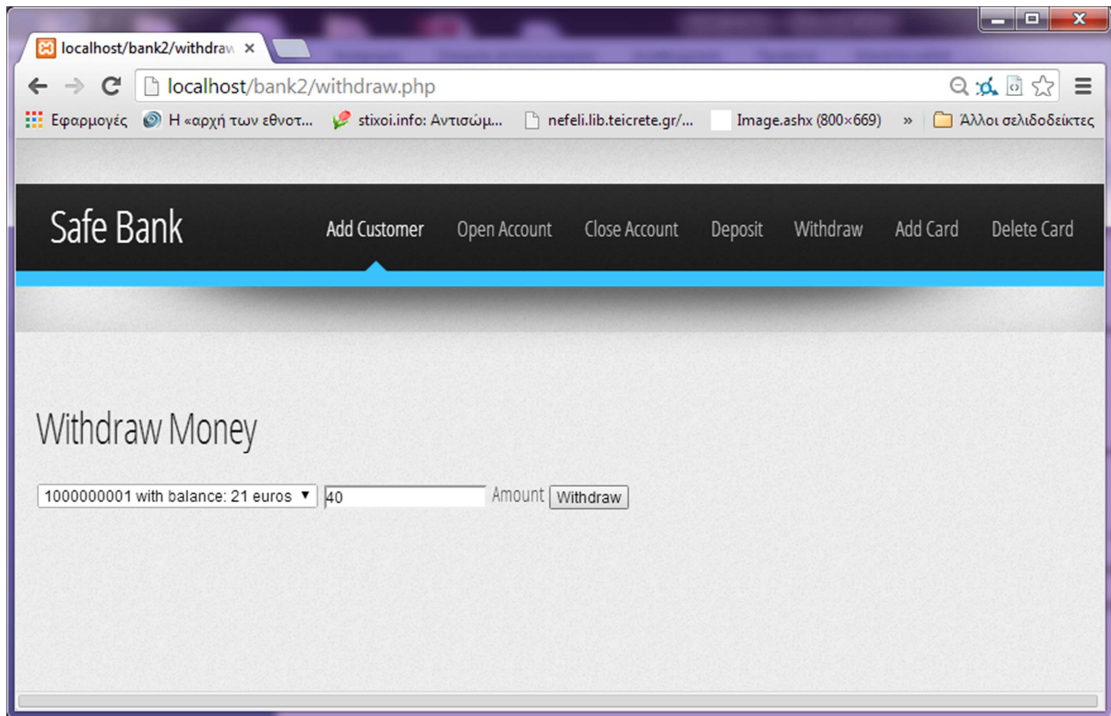
7.3. Έλεγχος λειτουργίας ανοίγματος λογαριασμού

Ο χρήστης επιλέγει Open Account και εν συνεχεία το όνομα ή τα ονόματα των πελατών για τους οποίους θέλει να ανοίξει λογαριασμό.

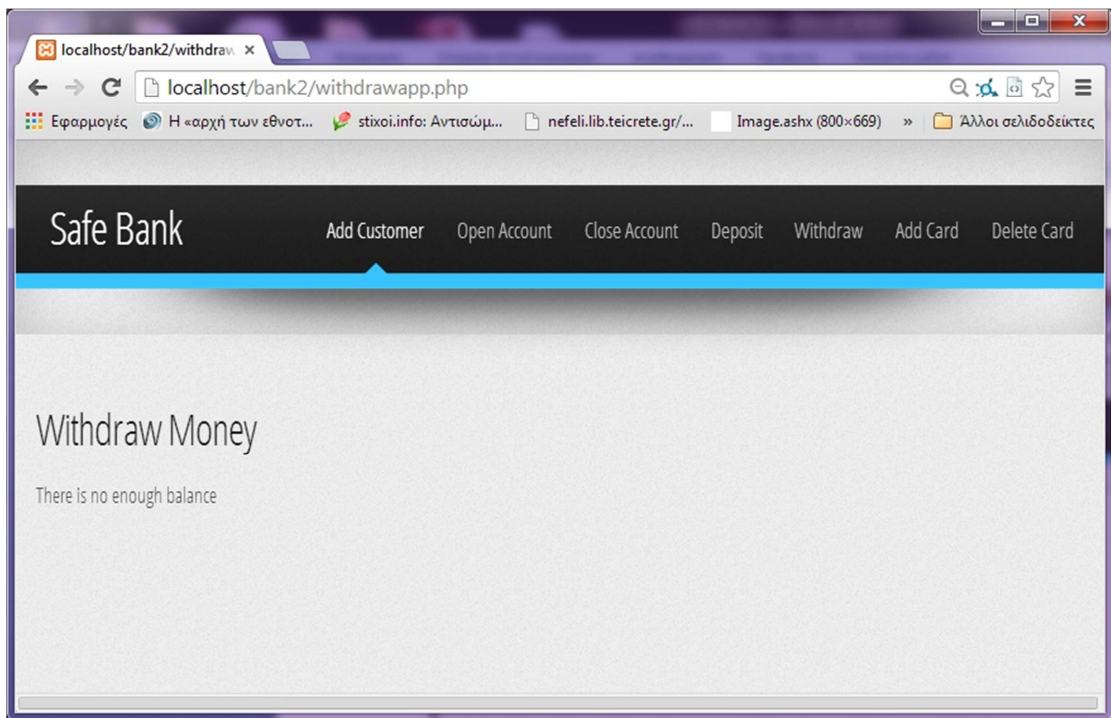


7.4. Έλεγχος λειτουργίας Ανάληψης Χρημάτων

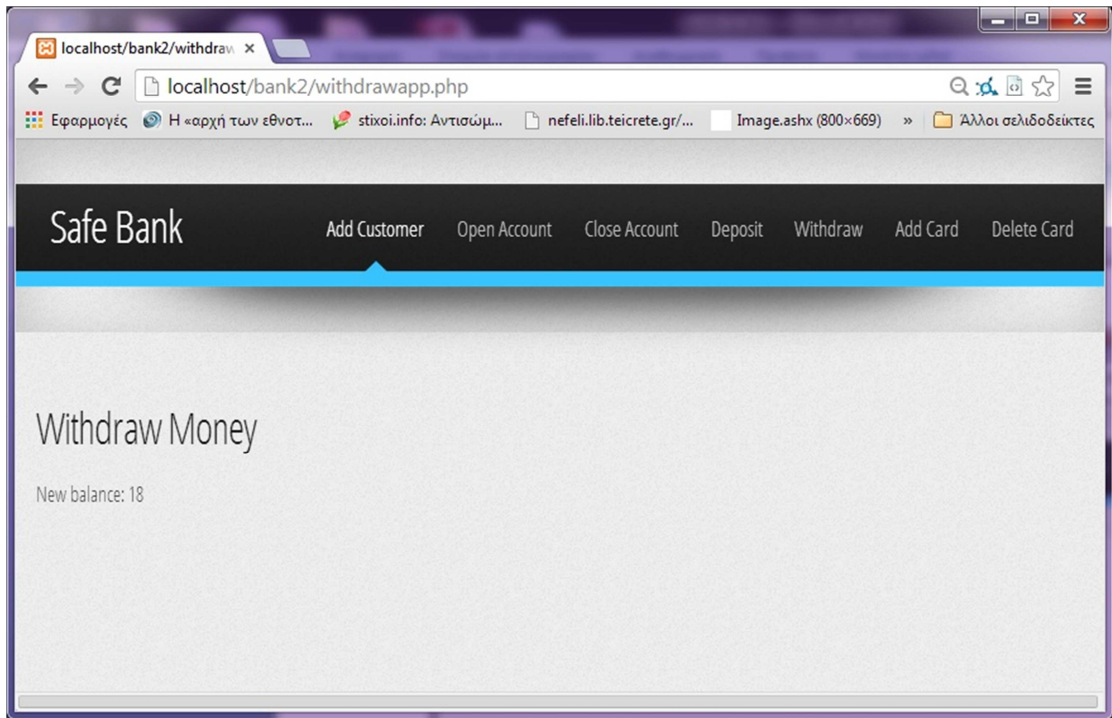
Ο χρήστης επιλέγει Withdraw. Ακολουθώντας το αριθμό λογαριασμού και το ποσό ανάληψης. Αν αυτό υπερβαίνει το διαθέσιμο υπόλοιπο, εμφανίζει κατάλληλο μήνυμα.



Η αδυναμία εκτέλεσης της συναλλαγής φαίνεται παρακάτω:



Αντιθέτως, αν επιλεγθεί κατάλληλο ποσό:



8. Επίλογος – Συμπεράσματα – Μελλοντικές Επεκτάσεις

Στο σημείο αυτό θα πρέπει να εξηγηθεί τι έχουμε αντιληφθεί σε σχέση με τις ανάγκες, τα προβλήματα που διαπιστώθηκαν κατά την περίοδο της ανάπτυξης του συστήματος, τι μάθαμε από την υλοποίηση του έργου και τις πιθανές βελτιώσεις του διαδικτυακού συστήματος δημοπρασιών.

8.1. Υλοποίηση απαιτήσεων

Παρακάτω φαίνεται ο πίνακας υλοποίησης των απαιτήσεων που τέθηκαν εξ αρχής και μερικές από αυτές αναλύθηκαν στο κεφάλαιο 4.

Απαιτήσεις	Υλοποίηση
Αρχική σελίδα	ΝΑΙ
Εγγραφή Πελάτη	ΝΑΙ
Σύνδεση	ΟΧΙ
Μενού πλοήγησης	ΝΑΙ
Διαχείριση Λογαριασμού	ΝΑΙ
Διαχείριση Καρτών	ΝΑΙ
Εκτύπωση στοιχείων	ΟΧΙ
Βοήθεια	ΟΧΙ
Αλλαγή Γλώσσας	ΟΧΙ
Συνομιλία σε πραγματικό χρόνο	ΟΧΙ

Είναι εύκολα παρατηρήσιμο πως οι περισσότερες απαιτήσεις υλοποιήθηκαν με εξαίρεση κάποιες δευτερεύουσες. Η μη υλοποίηση τους συνδέεται με την έλλειψη χρόνου και στη χαμηλή προτεραιότητα τους σε σχέση με τις υπόλοιπες.

8.2. Προβλήματα

Τα περισσότερα προβλήματα που ανέκυψαν είχαν να κάνουν με τη χρήση της γλώσσας Javascript. Η αρχική σκέψη ήταν κάνει μία συναλλαγή ένας χρήστης, να μην μεταφέρεται σε άλλη σελίδα, αλλά με τη βοήθεια της προαναφερθείσας γλώσσας και της τεχνολογίας AJAX να επαναφορτώνονται τα δεδομένα στην ίδια σελίδα. Θεωρώ πως η Javascript αποτελεί σπουδαίο εργαλείο. Ταυτόχρονα, όμως, απαιτεί αρκετή εμπειρία στο χειρισμό της.

8.3. Γνώση που αποκτήθηκε

Οι γνώσεις που απέκτησα κατά τη διάρκεια της ανάπτυξης του έργου έχουν να κάνουν κυρίως με το δυναμικό προγραμματισμό στο διαδίκτυο. Θέματα που αφορούν την σχεδίαση βάσεων δεδομένων και το χειρισμό τους μέσω της γλώσσας PHP. Ταυτόχρονα έγινε μια αρκετά ικανοποιητική προσέγγιση της γλώσσας σεναρίων Javascript. Με αυτό το έργο είχα επίσης την ευκαιρία να δω στην πράξη τεχνικές μοντελοποίησης (περιπτώσεων χρήσης, UML διαγράμματα, ER διαγράμματα, διαγράμματα ροής κτλ.) που είχα δει περισσότερο θεωρητικά σε διάφορα μαθήματα του πανεπιστημίου.

8.4. Μελλοντικές επεκτάσεις

Το σύστημα διαχείρισης τράπεζας λειτουργεί καλά, ωστόσο ορισμένα μελλοντικά έργα μπορεί να γίνουν στο υπάρχον σύστημα:

- Προσθήκη ενός SSL συστήματος ασφάλειας. Δεδομένου ότι ένας εγγεγραμμένος χρήστης μπορεί να δημοσιεύσει νέες δημοπρασίες, προσφορές, αποστολή μηνυμάτων κλπ.θα μπορούσε να είναι χρήσιμο για την προστασία αυτών των δεδομένων από υποκλοπή από τρίτους.
- Προσθήκη ενός δωματίου συνομιλίας με την πύλη. Αυτή η συνομιλία μπορεί να υλοποιηθεί χρησιμοποιώντας την Java Applet τεχνολογία.
- Πιο ελκυστικά γραφικά στο web σελίδες του portal.

- Προσθήκη αρχικής σελίδας με σύνδεση του χρήστη του συστήματος και εφαρμογή sessions.

Αναφορές

Διαδίκτυο

- PHP <http://el.wikipedia.org/wiki/PHP>
- MySQL <http://en.wikipedia.org/wiki/MySQL>
- CSS, <https://el.wikipedia.org/wiki/CSS>
- CSS, <http://www.xmluk.org/css-history-information.htm>
- Διαδίκτυο, <https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CE%B4%CE%AF%CE%BA%CF%84%CF%85%CE%BF>
- WampServer, <http://www.wampserver.com/en/>

Βιβλιογραφία

- Larry Ullman, *Peachpit Press PHP5Advanced*, 2007
- Timothy Boronczyk, Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. GlassPractical, *Beginning PHP 6 Apache MySQL 6 Web Development*, 2009
- Craig Cannell, Dionysios Synodinos, *The Essential Guide to HTML5 and CSS3 Web Design*, 2012