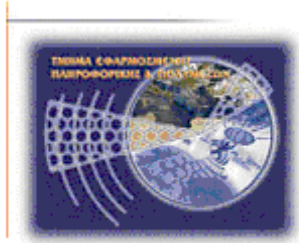




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



Πτυχιακή εργασία

Κατασκευή ηλεκτρονικού καταστήματος

Ντιάνα Τσιλίδη (ΑΜ: 1411)

Ηράκλειο – Δεκέμβριος 2013

Επόπτης Καθηγητής: Δρ. Παχουλάκης Ιωάννης

Υπεύθυνη Δήλωση: Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Πληροφορικής και Πολυμέσων του Τ.Ε.Ι. Κρήτης.

Ευχαριστίες

Θέλω να ευχαριστήσω τον κ. Ιωάννη Παχουλάκη για την στήριξη και την πολύτιμη βοήθειά του στην παρούσα πτυχιακή.

Επίσης, ένα μεγάλο ευχαριστώ στους γονείς μου και για την στήριξη και την υπομονή τους σε ολόκληρη την φοιτητική μου πορεία.

Περίληψη

Η εργασία αυτή αποτελεί το θεωρητικό μέρος της πτυχιακής, το αντικείμενο της οποίας είναι η ανάλυση, σχεδίαση και η υλοποίηση μίας εφαρμογής ηλεκτρονικού καταστήματος (e-shop). Το ηλεκτρονικό αυτό κατάστημα εμπορεύεται επιτραπέζια παιχνίδια, συλλεκτικές κάρτες και μινιατούρες, βιβλία και περιοδικά. Περιλαμβάνει καλάθι αγορών, λειτουργίες διαχειριστή, κατηγορίες χρηστών και επικοινωνία με το κατάστημα.

Το πρώτο κεφάλαιο περιγράφει την έννοια του ηλεκτρονικού εμπορίου, τις μορφές που μπορεί να πάρει και περιλαμβάνει μια σύντομη ιστορική αναδρομή στην εξέλιξή του. Επιπλέον, αναφέρεται σε παραδείγματα γνωστών και πετυχημένων ηλεκτρονικών καταστημάτων παγκοσμίως, αλλά και στην Ελλάδα.

Στη συνέχεια υπάρχει αναφορά στο λογισμικό που χρησιμοποιήθηκε για την υλοποίηση του έργου, στην τεχνολογία ASP.NET, το περιβάλλον Visual Studio, τη γλώσσα προγραμματισμού VB.NET (Visual Basic), καθώς και τη διαχείριση της βάσης δεδομένων σε SQL Server.

Στο τρίτο κεφάλαιο περιγράφεται αναλυτικά η λειτουργία της βάσης δεδομένων και η χρήση του SQL Server Management Studio για τη διαχείρισή της.

Στο τέταρτο κεφάλαιο περιγράφεται η όλη λειτουργία του καταστήματος, από την πλευρά του επισκέπτη (guest), του χρήστη (user) και του διαχειριστή (administrator). Αναφέρονται οι πίνακες που χρειάστηκαν για την κατασκευή και λειτουργία του ηλεκτρονικού καταστήματος και η σχέσεις μεταξύ τους.

Στο τέλος της εργασίας αναφέρεται η βιβλιογραφία που χρησιμοποιήθηκε.

Abstract

This work constitutes the theoretical part of the thesis, the object of which is the analysis, design and implementation of an electronic store (e-shop). This electronic store sells board games, collectible card and miniatures, books and magazines. It includes basket, operations manager, user groups and communication with the store. The first chapter describes the concept of e-commerce, the forms it can take and includes a brief historic overview of it. Moreover, referring to examples of known and successful online shops worldwide, but also in Greece.

Then there is a reference to the software used for the project implementation, the ASP.NET technology, the Visual Studio environment, VB.NET (Visual Basic) programming language, as well as managing the database in SQL Server.

The third chapter details the operation of the database and the use of SQL Server Management Studio to manage it .

The fourth chapter describes the overall operation of the store, on behalf of the visitor (guest), user (user) and manager (administrator). Specifies the tables needed for the construction and operation of the shop and the relationship between them.

At the end of the work there are references to the literature which was used.

Περιεχόμενα

Περίληψη.....	iv
Περιεχόμενα.....	v
Πίνακας εικόνων	vii
Κεφάλαιο 1 Εισαγωγή	1
1.1 Ηλεκτρονικό εμπόριο	1
1.1.1 Είδη ηλεκτρονικού εμπορίου	2
1.1.2 Τεχνολογίες ηλεκτρονικού εμπορίου	2
1.1.3 Πλεονεκτήματα και μειονεκτήματα του ηλεκτρονικού εμπορίου	3
1.1.4 Ιστορική Αναδρομή.....	5
1.1.5 Πεδία δράσης του ηλεκτρονικού εμπορίου.....	5
1.1.6 Στρατηγικές Επιχειρήσεων στο ηλεκτρονικό Εμπόριο.....	7
1.1.7 Επίπεδα του Ηλεκτρονικού Εμπορίου	7
1.1.8 Ασφάλεια Συναλλαγών	7
1.1.9 Γνωστά ηλεκτρονικά καταστήματα	8
Κεφάλαιο 2 Λογισμικό.....	11
2.1 Ιστοσελίδες.....	11
2.1.1 Είδη ιστοσελίδων.....	11
2.2 Web Server (Εξυπηρετητής δικτύου).....	12
2.3 ASP.NET	14
2.3.1 Ιστορική Αναδρομή.....	14
2.3.2 ASP.NET σε σύγκριση με την κλασική ASP	15
2.4 CSS.....	16
2.4.1 Ιστορική αναδρομή.....	17
2.4.2. Εκδόσεις.....	19
2.5 Visual Basic.....	23
2.4.1 Χαρακτηριστικά της Γλώσσας.....	23
2.6 Visual studio	25
Κεφάλαιο 3 Βάση Δεδομένων	32
3.1 Ιστορική αναδρομή	33
3.2 Sql Server.....	34
3.3 SQL Server Management Studio	37
3.4 Γλώσσα SQL.....	38
3.4.1 Ιστορική αναδρομή.....	38
Κεφάλαιο 4 Εφαρμογή	40
4.1 Περιγραφή της εφαρμογής	40
4.2 Οι κλάσεις	47
4.3 Η βάση δεδομένων.....	53
4.3.1. Stored Procedures	54
Βιβλιογραφία	57

ΠΑΡΑΡΤΗΜΑ Α	59
Κύρια μέρη της εφαρμογής.....	59

Πίνακας εικόνων

1.0.1 - Το ηλεκτρονικό κατάστημα ebay.com.....	9
1.0.2 - Το ηλεκτρονικό κατάστημα amazon.com.....	9
1.0.3 - Το ηλεκτρονικό κατάστημα e-shop.gr.....	10
1.0.4 - Το ηλεκτρονικό κατάστημα kaissagames.gr.....	10
2.0.1 - Το περιβάλλον ανάπτυξης του Visual Studio.....	25
2.0.2 - Designer και Source View του Visual Studio.....	26
2.0.3 - Τα μενού του Visual Studio.....	29
2.0.4 - Διαχείριση βάσης δεδομένων μέσω Visual Studio.....	30
3.0.1 - Περιβάλλον εργασίας SQL Server Management Studio.....	37
4.1 - Αρχική σελίδα.....	40
4.2 - Κατηγορίες προϊόντων.....	41
4.3 - Φωτογραφίες (Flash book).....	41
4.4 - Εγγραφή χρήστη.....	42
4.5 - Αναζήτηση στη σελίδα.....	42
4.6 - Καλάθι αγορών.....	43
4.7 - Επιβεβαίωση αγοράς.....	43
4.8 - Λογαριασμός χρήστη.....	44
4.9 - Μενού διαχειριστή.....	44
4.10 - Προσθήκη προϊόντων.....	45
4.11 - Επεξεργασία κατηγοριών.....	45
4.12 - Επεξεργασία προϊόντων.....	46
4.13 - Επεξεργασία παραγγελιών.....	46
4.14 - Επεξεργασία στοιχείων χρηστών.....	47
4.15 - Βάση δεδομένων.....	53

Κεφάλαιο 1 Εισαγωγή

1.1 Ηλεκτρονικό εμπόριο

Υπάρχουν πολλοί ορισμοί για την έννοια του ηλεκτρονικού εμπορίου (e-commerce), αλλά στην ουσία είναι η δυνατότητα των καταναλωτών και των επιχειρήσεων να διεξάγουν εμπορικές συναλλαγές μέσω του διαδικτύου. Εννοείται κάθε εμπορική συναλλαγή, η οποία εκτελείται αποκλειστικά σε ηλεκτρονικό επίπεδο. Οι συναλλαγές αυτές είναι πολύπλευρες, καθώς δίνουν τη δυνατότητα για ηλεκτρονική διανομή πληροφοριών μέσω δικτυακού τόπου, αυτοματοποιημένες εμπορικές συναλλαγές, βελτιωμένη παροχή υπηρεσιών με χαμηλότερο κόστος συναλλαγής, αλλά και μηχανισμούς πρόσβασης πραγματικού χρόνου για την αγορά και πώληση αγαθών και υπηρεσιών. Για την πραγματοποίηση μιας τέτοιας συναλλαγής χρησιμοποιούνται προγραμματιστικοί μηχανισμοί και το κατάλληλο λογισμικό το οποίο επιτρέπει την Ηλεκτρονική Ανταλλαγή Δεδομένων (Electronic Data Interchange- EDI) ανάμεσα στις δύο πλευρές (μεταξύ επιχειρήσεων αλλά και μεταξύ επιχειρήσεων και καταναλωτών) που εμπλέκονται στη συγκεκριμένη συναλλαγή. Με άλλα λόγια, η συγκεκριμένη μορφή συναλλαγής πραγματοποιείται μόνο μέσω υπολογιστών, παρακάμπτοντας τον ανθρώπινο παράγοντα και ελαχιστοποιώντας ταυτόχρονα την πιθανότητα λάθους και την κακόβουλη χρήση στοιχείων.

Ήδη, πλειάδα επιχειρήσεων, τόσο στην Ευρώπη όσο και στις Ηνωμένες Πολιτείες διαθέτουν τα προϊόντα τους μέσω του διαδικτύου. Στην Ελλάδα, αν και υπάρχει μια σχετική καθυστέρηση σε αυτό τον τομέα, οι εξελίξεις είναι σημαντικές και υπάρχουν ήδη αρκετές εταιρίες και επιχειρήσεις που δραστηριοποιούνται στο ηλεκτρονικό εμπόριο. Σε κάθε περίπτωση, ο κύριος λόγος που μια επιχείρηση δραστηριοποιείται σε ηλεκτρονικό επίπεδο είναι για να προσελκύσει αγοραστικό κοινό πέρα από τα στενά όρια της γεωγραφικής της έδρας, αυξάνοντας έτσι τις πωλήσεις των προϊόντων της και κατά συνέπεια τα έσοδά της. Οι επιχειρήσεις προσφέρουν στις σελίδες του ηλεκτρονικού τους καταστήματος ένα είδος on-line καταλόγου με όλα τα προϊόντα τους, τα λεπτομερή χαρακτηριστικά τους και βέβαια τις αντίστοιχες τιμές τους.

Από την άλλη πλευρά, οι ηλεκτρονικοί πελάτες έχουν στη διάθεσή τους ένα «καλάθι» στο οποίο στοιβάζουν τα προϊόντα που επιθυμούν να αγοράσουν και έχοντας ολοκληρώσει τα ψώνια τους από το συγκεκριμένο site, δίνουν την τελική παραγγελία τους, η οποία και εκτελείται αυτόματα. Οι πληρωμές, όπως και οι παραγγελίες, γίνονται και αυτές ηλεκτρονικά, μέσω της πιστωτικής κάρτας του καταναλωτή η οποία χρεώνεται με το αντίστοιχο ποσό και σε μερικές περιπτώσεις και μέσω αντικαταβολής.

Θα λέγαμε πως το ηλεκτρονικό εμπόριο ενδείκνυται για επιχειρήσεις που παράγουν βιομηχανοποιημένα προϊόντα τα οποία δεν διακρίνονται για τη μοναδικότητά τους και δεν χρειάζεται ο αγοραστής κατά κάποιο τρόπο να τα «δοκιμάσει» πριν τα αγοράσει, όπως βιβλία, εισιτήρια, CD, διάφορα είδη δώρων, ηλεκτρικές και ηλεκτρονικές συσκευές που δεν απαιτούν κάποιου είδους test drive, προγράμματα software & hardware, καινούρια αυτοκίνητα κλπ.

Το ηλεκτρονικό εμπόριο διακρίνεται σε έμμεσο και άμεσο. Ο πρώτος όρος χρησιμοποιείται όταν πρόκειται για την ηλεκτρονική παραγγελία υλικών αγαθών που μπορούν να παραδοθούν μόνο με παραδοσιακούς τρόπους όπως είναι το ταχυδρομείο. Άμεσο είναι το ηλεκτρονικό εμπόριο που περιλαμβάνει παραγγελία, πληρωμή και παράδοση αυτών των αγαθών και υπηρεσιών. Η πληρωμή των υπηρεσιών αυτών γίνεται είτε με πιστωτικές κάρτες είτε με ηλεκτρονικό χρήμα με την αρωγή πάντα και τη σύμπραξη των τραπεζών.

1.1.1 Είδη ηλεκτρονικού εμπορίου

Το ηλεκτρονικό εμπόριο μπορεί να πάρει τις εξής μορφές:

- **Επιχείρηση με επιχείρηση (Business to business)**

Είναι μια επιχείρηση που χρησιμοποιεί ένα δίκτυο για τις παραγγελίες της από προμηθευτές, που λαμβάνει τιμολόγια και κάνει πληρωμές. Θα μπορούσαμε να το ονομάσουμε χονδρικό εμπόριο. Χαρακτηριστικό παράδειγμα είναι οι παραγγελίες και οι αγορές ενός εργοστασίου. Η διαδικασία είναι αμφίδρομη και περιλαμβάνει την αγορά και την πώληση, αλλά και στοιχεία που έχουν σημασία για τη λειτουργία της επιχείρησης, όπως είναι η διαθεσιμότητα, οι όροι πληρωμής ή ο χρόνος παράδοσης. Αυτή η κατηγορία έχει κατοχυρωθεί αρκετά χρόνια, ειδικά με την χρησιμοποίηση του EDI σε κλειστά ή διεθνή δίκτυα.

- **Επιχείρηση με καταναλωτή (Business to consumer)**

Εξομοιώνεται με την ηλεκτρονική λιανική πώληση. Αυτή η κατηγορία έχει αναπτυχθεί με την εκτόξευση του World Wide Web. Οι καταναλωτές μαθαίνουν για τα προϊόντα μέσα από ηλεκτρονικές εκδόσεις, αγοράζουν προϊόντα με "ψηφιακό" χρήμα και άλλα ασφαλή συστήματα πληρωμής. Υπάρχουν τώρα "καταστήματα" σε όλο το Internet, που προσφέρουν κάθε είδος προϊόντων, από βιβλία και CD, μέχρι Η/Υ και αυτοκίνητα.

- **Επιχείρηση με δημόσια διοίκηση (Business to government)**

Καλύπτει όλες τις συναλλαγές μεταξύ επιχειρήσεων και δημόσιων οργανισμών. Προς το παρόν, αυτή η κατηγορία είναι σε νηπιακό στάδιο, αλλά μπορεί να αναπτυχθεί ραγδαία όσο οι κυβερνήσεις χρησιμοποιούν τις δικές τους λειτουργίες για να προωθήσουν την αντίληψη τους για το Ηλεκτρονικό Εμπόριο. Επιπροσθέτως, οι διοικήσεις πρέπει να παρέχουν την ευκαιρία ηλεκτρονικών συναλλαγών για καταστάσεις όπως επιστροφές ΦΠΑ και δασμών.

- **Καταναλωτή με καταναλωτή (Consumer to consumer)**

Δίνει την δυνατότητα δημιουργίας ηλεκτρονικού εμπορίου μεταξύ καταναλωτή με καταναλωτή μέσω ενός τρίτου παράγοντα. Ένα παράδειγμα είναι οι online δημοπρασίες που γίνονται σε ορισμένες ιστοσελίδες όπως είναι οι : eBay , Amazon κτλ. Ο πρώτος καταναλωτής δίνει ένα αντικείμενο για πώληση, ο δεύτερος πλειοδοτεί για την απόκτηση του και ο τρίτος , ο οποίος είναι η ιστοσελίδα η οποία παρέχει αυτήν την λειτουργία, πληρώνεται με ένα ποσοστό επί τις αγοράς.

1.1.2 Τεχνολογίες ηλεκτρονικού εμπορίου

- **Ηλεκτρονική Ανταλλαγή Δεδομένων (EDI - Electronic Data Interchange)**

Δημιουργήθηκε στις αρχές της δεκαετίας του '70. Η EDI είναι μια κοινή δομή αρχείων που σχεδιάστηκε ώστε να επιτρέψει σε μεγάλους οργανισμούς να μεταδίδουν πληροφορίες μέσα από μεγάλα ιδιωτικά δίκτυα. Πρόκειται για την ηλεκτρονική ανταλλαγή εμπορικών και διοικητικών δεδομένων από υπολογιστή σε υπολογιστή, με την ελάχιστη παρέμβαση χειρόγραφων διαδικασιών. Τα δεδομένα αυτά είναι οργανωμένα σε αυτοτελή μηνύματα (τιμολόγια, παραγγελίες,

τιμοκατάλογοι, φορτωτικές κλπ.), το περιεχόμενο και η δομή των οποίων καθορίζονται από κάποιο κοινώς αποδεκτό πρότυπο. Τα πρότυπα που χρησιμοποιούνται σε παγκόσμιο επίπεδο προέρχονται από τον Οργανισμό Ηνωμένων Εθνών και καλύπτουν ένα ευρύ φάσμα επικοινωνιακών αναγκών των εμπορικών εταιριών. Το πρότυπο αυτό είναι το EDIFACT (EDI For Administration, Commerce and Transportation).

- **Επίπεδο Ασφαλών Συνδέσεων (SSL - Secure Sockets Layer)**

Το πρωτόκολλο αυτό σχεδιάστηκε προκειμένου να πραγματοποιεί ασφαλή σύνδεση με τον εξυπηρετητή (server). Το SSL χρησιμοποιεί "κλειδί" δημόσιας κρυπτογράφησης, με σκοπό να προστατεύει τα δεδομένα καθώς "ταξιδεύουν" μέσα στο Internet.

- **Ασφαλείς Ηλεκτρονικές Συναλλαγές (SET - Secure Electronic Transactions)**

Το SET κωδικοποιεί τους αριθμούς της πιστωτικής κάρτας που αποθηκεύονται στον εξυπηρετητή του εμπόρου. Το πρότυπο αυτό, που δημιουργήθηκε από τη Visa και τη MasterCard, απολαμβάνει μεγάλης αποδοχής από την τραπεζική κοινότητα.

- **Πιστοποίηση και ασφάλεια**

Για την ασφάλεια των ηλεκτρονικών συναλλαγών χρησιμοποιούνται ευρέως τα firewalls. Το firewall αποτελεί λογισμικό ή υλικό, που επιτρέπει μόνο στους εξωτερικούς χρήστες που έχουν τα κατάλληλα δικαιώματα, να προσπελάσουν το προστατευόμενο δίκτυο. Ένα firewall επιτρέπει στους εσωτερικούς χρήστες να έχουν πλήρη πρόσβαση στις παρεχόμενες υπηρεσίες, ενώ οι εξωτερικοί χρήστες πρέπει να πιστοποιηθούν. Υπάρχουν πολλοί τύποι firewalls, καθένας από τους οποίους παρέχει διαφορετικά επίπεδα προστασίας. Ο συνηθέστερος τρόπος χρησιμοποίησης ενός firewall είναι η τοποθέτηση ενός υπολογιστή ή δρομολογητή μεταξύ συγκεκριμένου δικτύου και του Internet, και η παρακολούθηση όλης της κυκλοφορίας μεταξύ του εξωτερικού και του τοπικού δικτύου.

Η εμπιστευτική πληροφορία που διακινείται στο δίκτυο μπορεί να προστατευθεί με κρυπτογράφηση και χρήση μυστικών κωδικών. Η ασφάλεια του ηλεκτρονικού εμπορίου βασίζεται κατεξοχήν στην κρυπτογράφηση, δηλαδή στην κωδικοποίηση του μεταδιδόμενου κειμένου κατά τέτοιο τρόπο ώστε να μπορεί να αποκρυπτογραφηθεί μόνο με τη χρήση του ειδικού κλειδιού αποκρυπτογράφησης. Η κρυπτογράφηση συνοδεύεται πολλές φορές και από την ψηφιακή υπογραφή του αποστολέα, έτσι ώστε ο παραλήπτης να μπορεί να βεβαιωθεί για την ταυτότητα του πρώτου.

1.1.3 Πλεονεκτήματα και μειονεκτήματα του ηλεκτρονικού εμπορίου

Πλεονεκτήματα:

- εισαγωγή σε νέες αγορές
- απόκτηση νέων πελατών
- αύξηση παραγωγικότητας
- ασφαλείς συναλλαγές τοις μετρητοίς
- ανταγωνιστικά πλεονεκτήματα

Κατασκευή ηλεκτρονικού καταστήματος

Μειονεκτήματα

- Δεν υπάρχει εμπιστευτικότητα και ασφάλεια όσον αφορά το περιεχόμενο κάποιων πληροφοριών.
- Δεν υπάρχει ακεραιότητα, ώστε να προφυλάσσεται το υποκείμενο των πληροφοριών που διακινούνται.
- Μερικά είδη επιχειρήσεων είναι πολύ δύσκολο να ελεγχθούν ως προς την ποιότητα των προϊόντων τους. Αυτό αφορά κυρίως τα καταστήματα που εμπορεύονται είδη διατροφής, φαρμακευτικά προϊόντα κ.α.

Πλεονεκτήματα του ηλεκτρονικού εμπορίου για τον καταναλωτή

- Τα ηλεκτρονικά καταστήματα είναι ανοιχτά 24 ώρες το 24ωρο. Με άλλα λόγια οποιαδήποτε στιγμή το επιθυμείτε, μπορείτε να αγοράσετε π.χ. ένα CD, ή ένα αεροπορικό εισιτήριο.
- Το κόστος των προϊόντων που πωλούνται μέσω Internet είναι κατά γενικό κανόνα πολύ χαμηλότερο από τις τιμές του εμπορίου, αφού ένα ηλεκτρονικό κατάστημα είναι απαλλαγμένο από μεγάλο μέρος του λειτουργικού κόστους ενός πραγματικού καταστήματος (ενοικίαση χώρου και «αέρα», ηλεκτρικό, νερό κλπ) και γενικά απαιτεί πολύ λιγότερο υπαλληλικό προσωπικό.
- Η αγορά είναι πραγματικά παγκόσμια. Με άλλα λόγια, μπορείτε μέσω του υπολογιστή σας να αγοράσετε ακόμα και κάτι το οποίο δεν κυκλοφορεί στην χώρα σας.
- Η συναλλαγή είναι γρήγορη και άμεση. Από τη στιγμή που ολοκληρώνετε την παραγγελία σας, το αργότερο σε 30-40 ημέρες την έχετε λάβει, ακόμα και αν εκείνη τη στιγμή το προϊόν βρισκόταν στην άλλη άκρη του πλανήτη.
- Ο καθένας βρίσκει αυτό που θέλει, όποτε το θέλει, χωρίς κόπο και χωρίς καμία σπατάλη χρόνου. Με άλλα λόγια απλά και εύκολα ψώνια από το σπίτι ή το γραφείο.

Πλεονεκτήματα του ηλεκτρονικού εμπορίου για την εταιρία

- Όπως προαναφέραμε, κάθε εταιρία που έχει ηλεκτρονική παρουσία μπορεί να διευρύνει τον κύκλο εργασιών της επεκτείνοντας τα γεωγραφικά όρια των συναλλαγών της. Αυτό σημαίνει πως κάθε επιχείρηση που διαθέτει τα προϊόντα της online μπορεί και αποκτά πελάτες σε περιοχές που βρίσκονται μακριά από την έδρα της, ακόμα και στο εξωτερικό. Με άλλα λόγια, κάθε επιχείρηση που έχει ένα ηλεκτρονικό κατάστημα, είναι σαν να έχει υποκαταστήματα σε πολλές περιοχές και μάλιστα με ελάχιστο λειτουργικό κόστος.
- Κάθε εταιρία που χρησιμοποιεί τις νέες τεχνολογίες- όπως το Internet- γίνεται εξ ορισμού πιο ανταγωνιστική, αφού μπορεί να ενημερώνεται πιο εύκολα για τις τρέχουσες εξελίξεις στο χώρο της. Με άλλα λόγια και με δεδομένο το ότι σε λίγα χρόνια όλες οι εμπορικές δραστηριότητες θα γίνονται μέσω Internet, το ηλεκτρονικό εμπόριο είναι η νέα μεγάλη πρόκληση για κάθε εταιρία που θέλει να είναι ανταγωνιστική.
- Οι ηλεκτρονικές συναλλαγές επιτρέπουν την αμφίδρομη σχέση μεταξύ επιχείρησης και καταναλωτή (interaction). Αυτό σημαίνει πως κάθε εταιρία μέσω των ηλεκτρονικών συναλλαγών μπορεί να συλλέξει πολλά στοιχεία για τις συνήθειες, τις ανάγκες και τα γούστα των καταναλωτών και σύμφωνα με αυτά να αναπροσαρμόσει την πολιτική της προς το θετικότερο.
- Τέλος, γνωρίζοντας τις συγκεκριμένες ανάγκες των πελατών τους, οι εταιρίες μπορούν να προχωρήσουν στη δημιουργία συγκεκριμένων προϊόντων είτε

ανταποκρινόμενων σε έναν καταναλωτή, είτε σε μια ομάδα καταναλωτών που χρειάζονται ένα νέο προϊόν το οποίο δεν υπάρχει ακόμα στην αγορά.

1.1.4 Ιστορική Αναδρομή

Αν και οι ηλεκτρονικές αγορές έγιναν ευρέως γνωστές τα τελευταία χρόνια, το ηλεκτρονικό εμπόριο προϋπάρχει εδώ και πολύ καιρό. Για δεκαετίες τώρα οι τράπεζες χρησιμοποιούν τις ηλεκτρονικές μεταφορές χρημάτων και άλλων πληροφοριών συναλλαγών. Τη δεκαετία του '70 εμφανίζονται τα πρώτα συστήματα μεταφοράς χρηματικών πόρων (EFT) μεταξύ των τραπεζών, που χρησιμοποιούν ασφαλή ιδιωτικά δίκτυα.

Τη δεκαετία του '80 αναπτύσσονται οι τεχνολογίες ηλεκτρονικής επικοινωνίας που βασίζονται στην αρχιτεκτονική της ανταλλαγής μηνυμάτων (συστήματα EDI και ηλεκτρονικό ταχυδρομείο). Οι συναλλαγές που παλιότερα απαιτούσαν τη χρήση εγγράφων, όπως οι παραγγελίες και οι επιταγές πληρωμής, μπορούν να γίνουν ηλεκτρονικά.

Στις αρχές της δεκαετίας του '90 τα ηλεκτρονικά δίκτυα προσφέρουν μια νέα μορφή επικοινωνίας με πολλές δυνατότητες, την ηλεκτρονική διάσκεψη (conferencing), ηλεκτρονική συνομιλία, ομάδες συζήτησης, μεταφορά αρχείων κ.λπ.

Στα μέσα της δεκαετίας γίνεται η εμφάνιση του Παγκόσμιου Ιστού (World Wide Web) και η επικράτηση των προσωπικών ηλεκτρονικών υπολογιστών που προσφέρουν μεγάλη ευκολία όσο αναφορά την δημοσίευση και την αναζήτηση των πληροφοριών. Το ηλεκτρονικό εμπόριο γίνεται ένας φτηνότερος τρόπος για την πραγματοποίηση μεγάλου όγκου συναλλαγών.

Στο τέλος της δεκαετίας του '90 καθιερώνονται οι μέθοδοι κρυπτογράφησης του περιεχομένου και εξακρίβωσης της ταυτότητας του αποστολέα ηλεκτρονικών μηνυμάτων. Έτσι οι ηλεκτρονικές συναλλαγές γίνονται ακόμα πιο εύκολες και ασφαλείς.

1.1.5 Πεδία δράσης του ηλεκτρονικού εμπορίου

Το Ηλεκτρονικό Εμπόριο σαν μια γενική ιδέα καλύπτει όλες τις μορφές επιχειρηματικών συναλλαγών που διεξάγονται ηλεκτρονικά και χρησιμοποιούν τα τηλεπικοινωνιακά δίκτυα. Συμπεριλαμβάνει ένα μεγάλο εύρος δραστηριοτήτων και ο βασικός στόχος είναι φυσικά ο κύκλος των εμπορικών συναλλαγών. Δηλαδή το να "εμπορεύεσαι" ηλεκτρονικά φυσικά αγαθά και υπηρεσίες δεν είναι μόνο ηλεκτρονικό θέμα. Μεγάλη προσοχή πρέπει να δοθεί στη διαφήμιση και στην προώθηση του προϊόντος, στις διευκολύνσεις που παρέχονται μεταξύ των εμπόρων, στην προμήθεια και στους όρους αγοράς. Ακόμα στην υποστήριξη πριν αλλά και μετά την πώληση καθώς επίσης και στην υποστήριξη για διαμοιραζόμενες επιχειρηματικές εργασίες. Το ηλεκτρονικό εμπόριο επιδρά σε ένα μεγάλο αριθμό επιχειρηματικών Δραστηριοτήτων.

- Μάρκετινγκ, πωλήσεις και προώθηση πωλήσεων προσφορές πριν την πώληση
- Χρηματοδότηση και ασφάλιση εμπορικές συναλλαγές: παραγγελία, μεταφορά και πληρωμή, σέρβις προϊόντος και συντήρηση-υποστήριξη ανάπτυξη προϊόντος, κατανεμημένη εργασία
- Χρήση δημοσίων και ιδιωτικών υπηρεσιών επιχείρηση-δημόσια διοίκηση (παραχωρήσεις, άδειες, φόροι, κτλ.)
- Μεταφορές και λογιστική προσωπικού και υλικών

Κατασκευή ηλεκτρονικού καταστήματος

- Προμήθειες δημοσίου αυτόματο εμπόριο ψηφιακών αγαθών λογιστικά

Η όλη εμπορική συναλλαγή μπορεί να υποστηριχθεί ηλεκτρονικά, συμπεριλαμβανομένων και της μεταφοράς και της πληρωμής. Θεωρητικά ακόμα υπάρχει και η δυνατότητα να γίνεται η συνδιαλλαγή με τις δημόσιες υπηρεσίες ηλεκτρονικά, δηλαδή για πληρωμή δασμών και φόρων. Παρόλα αυτά όμως ένας αριθμός ζητημάτων όπως η προστασία και η ασφάλεια, η νομική κάλυψη δεν έχουν διευθετηθεί ακόμα ώστε να αποτελέσουν αναπόσπαστο κομμάτι του κεφαλαίου αυτού που λέγεται Ηλεκτρονικό Εμπόριο.

Θα πρέπει να γίνεται όμως ένας σαφής διαχωρισμός μεταξύ της ηλεκτρονικής μεταφοράς φυσικών αγαθών και υπηρεσιών και ανάμεσα στην ηλεκτρονική μεταφορά περιεχομένων βασισμένα αποκλειστικά σε ψηφιακή μορφή (εικόνες, ήχος, κείμενο, software).

Ο Η.Ε. φυσικών αγαθών και υπηρεσιών αναπαριστά θα λέγαμε την εξέλιξη της μορφής του εμπορίου γενικότερα στη σημερινή εποχή, κεφαλαιοποιώντας τις νέες δυνατότητες που προσφέρει η τεχνολογία για να επιτευχθεί η μέγιστη αποδοτικότητα των πόρων της επιχείρησης. Παράλληλα, προσφέρει το άνοιγμα της αγοράς για νέα προϊόντα και αναβαθμισμένες υπηρεσίες μέσα από μια πρωτοποριακή άμεση συναλλαγή πελάτη-προμηθευτή. Αναμένεται να έχει μεγάλη επίδραση στον ανταγωνισμό και λιγότερη στην απασχόληση.

Ειδικότερα, το εμπόριο ηλεκτρονικού υλικού (εικόνες, ήχος, κείμενο, video, software, games, multimedia works) αναπαριστά μια επαναστατική νέα μορφή εμπορίου, στην οποία ο κύκλος των εμπορικών συναλλαγών δεν κλείνει ποτέ, μια και βρίσκεται συνέχεια μέσα στο δίκτυο. Τα εμπορευόμενα "ηλεκτρονικά αγαθά" μπορούν να δημιουργήσουν ολοκληρωτικά καινούργιες αγορές, βασιζόμενα βέβαια σε επιτυχείς λύσεις, αλλά και να φέρουν επανάσταση σε μερικές βιομηχανίες (π.χ. εκδοτικούς οίκους). Αυτή καθαυτή η καινοτόμος μορφή εμπορίου αναμένεται να έχει μια σημαντική επίδραση στην ανταγωνιστικότητα και στη δημιουργία απασχόλησης.

Παραδείγματα από Ορισμένα Επιχειρηματικά Οφέλη

- Μειωμένα έξοδα διαφήμισης
- Μειωμένα έξοδα μεταφοράς, ιδιαίτερα για αγαθά που μπορούν να μεταφερθούν ηλεκτρονικά
- Μειωμένα έξοδα στο σχεδιασμό και στην παραγωγή
- Ανεπτυγμένος στρατηγικός σχεδιασμός
- Περισσότερες ευκαιρίες για nice marketing
- Ίση πρόσβαση στις αγορές από όλους
- Πρόσβαση σε νέες αγορές
- Ανάμειξη του πελάτη στο προϊόν και καινοτομία υπηρεσιών

Πληροφορίες για οποιαδήποτε αγορά και χώρα μπορούν πλέον να προωθηθούν ηλεκτρονικά και να αναπτυχθούν στο εμπορικό περιβάλλον, δημιουργώντας όμως έτσι και έναν αριθμό από προβλήματα όπως η μυστικότητα (privacy) που είναι ανάγκη να διευθετηθεί.

Επαφές μεταξύ εταιριών μπορούν να διευκολυνθούν με on-line επιχειρησιακούς καταλόγους και να βελτιωθούν με εθνικά ή τοπικά πληροφοριακά κέντρα. Επαφές μεταξύ εταιριών και καταναλωτών μπορούν να υποστηριχθούν με σημαντικά μέσα, συμπεριλαμβανομένων on-line διαφημίσεων και κλειστών αγορών. Εταιρίες μπορούν να προωθήσουν αναλυτικές πληροφορίες για τα προϊόντα τους και τις υπηρεσίες τους καθώς επίσης και να απαντήσουν σε ερωτήσεις που υποστηρίζονται από κατανοητές ευκολίες πλοήγησης και αναζήτησης.

Τα τελευταία χρόνια έγιναν προσπάθειες για να βελτιωθεί η επιχειρησιακή αποδοτικότητα και αυτό το γεγονός επέδρασε καταλυτικά και ξεπέρασε έτσι τα "σύνορα" εταιριών και πελατών. Ένα τέτοιο παράδειγμα συμβαίνει με την λεγόμενη "virtual

enterprise", εκεί όπου κάθε τέτοιου είδους εταιρία παίζει τον δικό της ρόλο σε ένα πολύ κοντινό δίκτυο εταιριών απευθύνοντας μια πολύ συγκεκριμένη ευκαιρία αγοράς.

1.1.6 Στρατηγικές Επιχειρήσεων στο ηλεκτρονικό Εμπόριο

- Ηλεκτρονική παρουσία στο χώρο αγοράς: προώθηση πωλήσεων
- Interactive TV / Internet αγορές
- Διοίκηση αποτελεσματικής ανταπόκρισης πελατών
- Διοίκηση αλυσίδας προσφοράς
- Καταγραφή πωλήσεων σε επίπεδο πωλητών

Το ηλεκτρονικό εμπόριο επιτρέπει επίσης ικανοποιητική υποστήριξη για μοιραζόμενες επιχειρηματικές εργασίες άσχετα με τη φύση τους και δίχως να λαμβάνεται υπόψη η γεωγραφική τους θέση ή η χρονολογική τους τοποθέτηση. Η αιτιολόγηση και τα παραδείγματα που δώσαμε παραπάνω, ενδυναμώνουν το επιχείρημα ότι το ηλεκτρονικό εμπόριο είναι ένα φαινόμενο το οποίο θα πρέπει να ληφθεί σοβαρά υπόψη από όλους μας κάτω από το πρίσμα των διαφορετικών πολιτικών που ακολουθούνται και των δεκάδων υπαρχόντων τομέων αγοράς.

1.1.7 Επίπεδα του Ηλεκτρονικού Εμπορίου

Υπάρχουν διάφορα και σημαντικά επίπεδα στα οποία μπορεί να συναντήσουμε το ηλεκτρονικό εμπόριο. Ποικίλλουν από μια απλή ηλεκτρονική παρουσία στο δίκτυο έως μια πλήρης ηλεκτρονική υποστήριξη εργασιών που ανήκουν σε δύο ή και παραπάνω εταιρίες και οι οποίες έχουν θεσπιστεί απ' αυτές.

Τα σημαντικότερα επίπεδα του ηλεκτρονικού εμπορίου φαίνονται και στο σχήμα. Βλέπουμε ότι γίνεται ένας διαχωρισμός μεταξύ εθνικών και διεθνών συναλλαγών και η διάκριση αυτή δεν γίνεται τόσο από τεχνικής άποψης όσο από νομικής. Το ηλεκτρονικό εμπόριο είναι πιο περίπλοκο στο διεθνές επίπεδο επειδή ακριβώς εμπλέκονται παράγοντες όπως η φορολόγηση, οι δασμοί, οι πληρωμές και οι διαφορές που υπάρχουν στις τραπεζικές πρακτικές.

Τα κατώτερα επίπεδα του ηλεκτρονικού εμπορίου ασχολούνται βασικά με μια απλή παρουσία δικτύου, προώθηση της εταιρίας και υποστήριξη πριν και μετά την πώληση. Χρησιμοποιώντας τεχνολογία "off the selves" μπορούν να έχουν ένα φτηνό αλλά ικανό εργαλείο στα χέρια τους. Σε αντίθεση σε πιο αναπτυγμένες μορφές συναντάμε προβλήματα τόσο νομικά όσο και τεχνολογικά. Οι εταιρίες αυτές είναι αναγκασμένες να αναπτύξουν μόνες τους τα συστήματα.

1.1.8 Ασφάλεια Συναλλαγών

Για τους εμπόρους

Τον Ιούλιο του '97, κάποιος στις ΗΠΑ κατόρθωσε με κάποιο τρόπο να σπάσει τους κωδικούς μιας επιχείρησης και να στείλει ηλεκτρονικά μηνύματα σε εκατοντάδες πελάτες που είχαν πραγματοποιήσει αγορές μέσω των sites ESPN SportsZone and NBA.com, τα οποία δημιουργήθηκαν από την Starwave. Ο "εισβολέας" ενημέρωσε τους πελάτες ότι οι πιστωτικές τους κάρτες είχαν παραβιαστεί από τους υπολογιστές της επιχείρησης και περιλάμβανε τα τελευταία νούμερα του αριθμού τους ως απόδειξη.

Κατασκευή ηλεκτρονικού καταστήματος

Αυτό το γεγονός αποδεικνύει ότι οι συναλλαγές δεν είναι απόλυτα ασφαλείς στο Internet. Όσο όμως η διατάραξη των διαδικτυακών συναλλαγών βρίσκουν το φως της δημοσιότητας, τόσο και περισσότεροι ερευνητές διχάζονται, αν και επικρατεί ότι οι συναλλαγές είναι πιο ασφαλείς στον Κυβερνοχώρο παρά στον φυσικό κόσμο. Αυτό συμβαίνει γιατί έχουν πραγματοποιηθεί πολλές απάτες πιστωτικών καρτών από υπαλλήλους σε καταστήματα λιανικών πωλήσεων που κρατούσαν τους αριθμούς των καρτών. Τα συστήματα Η.Ε. απομάκρυναν τον πειρασμό, κρυπτογραφώντας τους αριθμούς στους servers της εταιρίας.

Για τους εμπόρους το Η.Ε. είναι πραγματικά ασφαλέστερο, μιας και ένα κατάστημα μπορεί να διαρριχθεί, να καεί ή να πλημμυρίσει. Η μόνη δυσκολία είναι να αποδείξει στους πελάτες ότι το Η.Ε. είναι πραγματικά ασφαλές και γι' αυτούς.

Για τους πελάτες

Οι πελάτες δεν το έχουν πιστέψει ακόμα, αλλά οι ειδικοί λένε ότι οι ηλεκτρονικές συναλλαγές είναι ασφαλέστερες από τις συνηθισμένες αγορές μέσω πιστωτικής κάρτας. Κάθε φορά που πληρώνει κάποιος σε ένα κατάστημα, σε ένα restaurant, και πετάει την απόδειξη της πιστωτικής κάρτας γίνεται ελαστικός σε μια απάτη. Όμως στην έκδοση 2.0 (ή αργότερη) του Netscape Navigator ή του Microsoft Internet Explorer, οι συναλλαγές μπορούν να κρυπτογραφηθούν χρησιμοποιώντας Secure Sockets Layer (SSL), ένα πρωτόκολλο που δημιουργεί μια ασφαλή σύνδεση σε ένα server, προστατεύοντας την διακινούμενη πληροφορία στο Internet. Το SSL χρησιμοποιεί κρυπτογράφηση με κοινό κλειδί, μια από τις ισχυρότερες μεθόδους κρυπτογράφησης, που κυκλοφορούν. Στον Navigator, μπορείς να προτιμήσεις τον ασφαλή τρόπο όταν ένα άθραυστο κλειδί εμφανίζεται στην κάτω αριστερή γωνία του παραθύρου. Ο Internet Explorer εμφανίζει ένα λουκέτο στην κάτω δεξιά γωνία της οθόνης. Άλλος ένας τρόπος υποστήριξης από το SSL είναι όταν η γραμμή διευθύνσεων αρχίζει με https αντί για http:

Οι κατασκευαστές των browsers και οι εταιρίες πιστωτικών καρτών προωθούν ένα επιπλέον πρωτόκολλο ασφάλειας το Secure Electronic Transactions (SET). Το SET κωδικοποιεί τους αριθμούς πιστωτικών καρτών που βρίσκονται σε servers, έτσι ώστε μόνο οι τράπεζες και οι εταιρίες πιστωτικών καρτών να μπορούν να διαβάσουν τους αριθμούς αυτούς.

Κανένα σύστημα Η.Ε. δεν μπορεί να εγγυηθεί 100% προστασία της πιστωτικής κάρτας, αλλά είναι σίγουρα ασφαλέστερο από μια πιθανή διάρρηξη των χρημάτων σου σε ένα κατάστημα.

1.1.9 Γνωστά ηλεκτρονικά καταστήματα

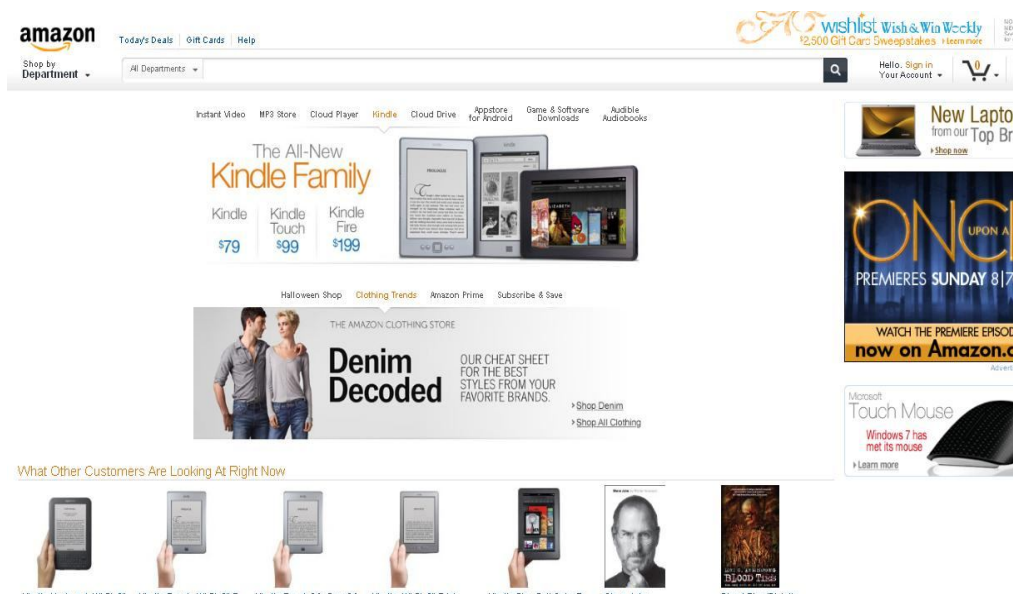
Καθημερινά δημιουργούνται όλο και πεισσότερα ηλεκτρονικά καταστήματα. Πλέον κάθε πραγματική επιχείρηση προσπαθεί να διευρύνει τους ορίζοντές της και στο διαδίκτυο. Από την άλλη υπάρχουν και καταστήματα που έχουν μόνο ηλεκτρονική μορφή. Κάποια από αυτά έχουν καθιερωθεί και είναι ευρέως γνωστά και πετυχημένα. Μερικά παραδείγματα τέτοιων ηλεκτρονικών καταστημάτων είναι τα εξής:

- eBay – Αμερικάνικο ηλεκτρονικό κατάστημα τύπου «καταναλωτής προς καταναλωτή» (Consumer to consumer), που δημιουργήθηκε το 1995 και έχει πάνω από 500 εκατομμύρια εγγεγραμμένους χρήστες παγκοσμίως. Προσφέρει τη δυνατότητα για online δημοπρασίες, αγοραπωλησίες μεταξύ χρηστών ατομικά, αλλά και μεταξύ επιχειρήσεων και καταναλωτών. Προσφέρει όλων των ειδών τα προϊόντα, από ενδύματα μέχρι και έπιπλα και πολλά άλλα.



1.0.1 - Το ηλεκτρονικό κατάστημα ebay.com

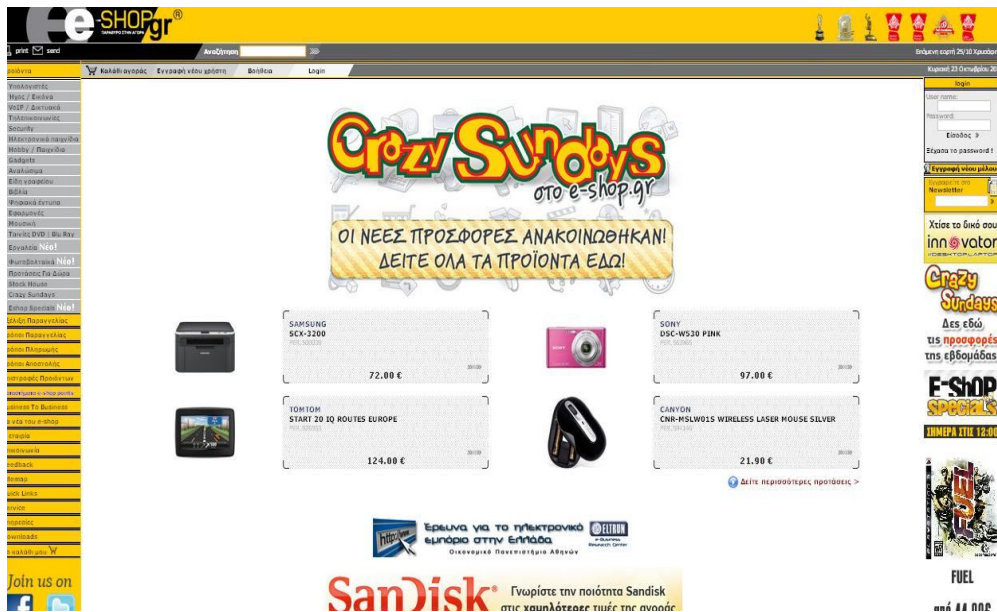
- Amazon – Αμερικάνικη πολυεθνική εταιρία ηλεκτρονικού εμπορίου με έδρα το Σιάτλ. Ιδρύθηκε το 1994 και σήμερα είναι μία από τις μεγαλύτερες, αν όχι η μεγαλύτερη εταιρία online λιανικής πώλησης. Ξεκίνησε ως ηλεκτρονικό βιβλιοπωλείο, αλλά στη συνέχεια επεκτάθηκε σε πώληση cd, dvd, mp3 downloads, λογισμικού ηλεκτρονικών υπολογιστών, video games, ηλεκτρονικών ειδών, ειδών ένδυσης, τροφίμων και παιχνιδιών.



1. 0.2 - Το ηλεκτρονικό κατάστημα amazon.com

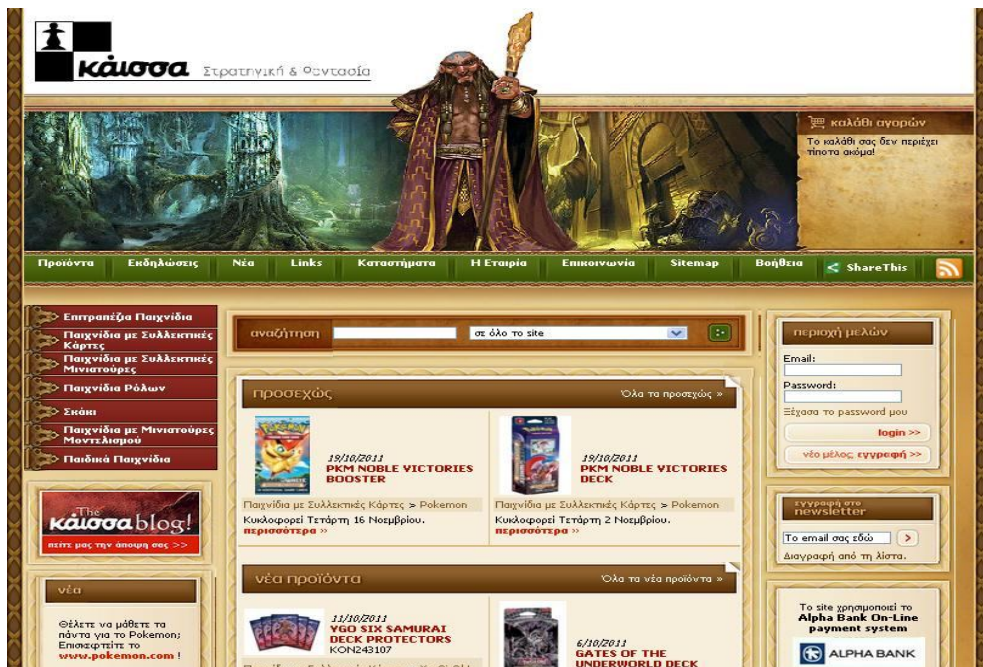
- E-shop.gr - Ελληνική ανώνυμη εταιρία ηλεκτρονικού εμπορίου και παροχής υπηρεσιών διαδικτύου με έτος ίδρυσης το 1998. Είναι κυριάρχη εταιρία στην Ελλάδα όσο αναφορά την πώληση λογισμικού ηλεκτρονικών υπολογιστών, hardware, περιφερειακών, κινητής τηλεφωνίας, βιβλίων κ.α.

Κατασκευή ηλεκτρονικού καταστήματος



1.0.3 - Το ηλεκτρονικό κατάστημα e-shop.gr

- Kaissagames.gr – Ελληνική εταιρία που ιδρύθηκε το 1985 ως σκακιστικό κατάστημα, αλλά σήμερα θεωρείται η κορυφαία ελληνική εταιρία στη διανομή επιτραπέζιων παιχνιδιών στρατηγικής και φαντασίας, συλλεκτικών καρτών, παιχνιδιών ρόλων και παιχνιδιών με συλλεκτικές μινιατούρες. Μέσα από το ηλεκτρονικό της κατάστημα μπορεί κανείς να προμηθευτεί τα παραπάνω προϊόντα. Έχει ως έδρα την Αθήνα, αλλά έχει πολλά καταστήματα σε όλη την Ελλάδα.



1.0.4 – Το ηλεκτρονικό κατάστημα kaissagames.gr

Κεφάλαιο 2 Λογισμικό

2.1 Ιστοσελίδες

Ιστοσελίδα ονομάζεται ένα έγγραφο συμβατό με το WWW (World Wide Web) και μπορεί να προσπελαστεί με έναν περιηγητή ιστού (web browser) και να εμφανιστεί σε οθόνη ή κινητή συσκευή. Μπορεί να περιέχει κείμενα, εικόνες, video, κινούμενα γραφικά και υπερσυνδέσμους που οδηγούν τον χρήστη σε κάποια άλλη ιστοσελίδα, αλλά μπορεί να αντιδρά πιο άμεσα στις ενέργειες του χρήστη, προσαρμόζοντας το περιεχόμενό τους, ανάλογα αν η ιστοσελίδα είναι στατική ή δυναμική. Οι ιστοσελίδες μπορούν να ανακτηθούν από τοπικό υπολογιστή ή από κάποιο απομακρυσμένο web server.

2.1.1 Είδη ιστοσελίδων

- Στατικές Ιστοσελίδες είναι οι ιστοσελίδες που παραδίδονται στο χρήστη όπως ακριβώς είναι αποθηκευμένες. Δείχνουν ακριβώς τις ίδιες πληροφορίες σε όλους τους χρήστες και δεν προσαρμόζονται στις ανάγκες τους και δεν ανανεώνονται παρά μόνο αν κάποιος τροποποιήσει τον κώδικά τους. Είναι, όμως, εύκολες στην κατασκευή τους.
- Δυναμικές Ιστοσελίδες είναι οι ιστοσελίδες που μπορούν να παρέχουν «φρέσκιες» πληροφορίες σε κάθε προβολή τους. Μπορούν να αλλάζουν με το χρόνο, ανάλογα με το χρήστη, ανάλογα με την αλληλεπίδραση με το χρήστη και εξαιτίας πολλών άλλων παραγόντων.
 - Client-side scripting, δηλαδή εκτέλεση προγραμμάτων στον υπολογιστή του επισκέπτη της ιστοσελίδας. Κάθε φορά που κάποιος επισκέπτεται έναν ιστοχώρο, τα προγράμματα φορτώνονται και εκτελούνται από τον περιηγητή ιστού του.
 - Server-side scripting, δηλαδή εκτέλεση προγραμμάτων στον εξυπηρετητή (web server) του ιστοχώρου. Πολλές φορές τα προγράμματα αυτά δεν γίνονται αντιληπτά από τους χρήστες, γιατί εκτελούν εσωτερικές λειτουργίες. Επεξεργάζονται πληροφορίες που εισάγουμε με διάφορους τρόπους, όπως φόρμες επικοινωνίας, μενού επιλογών, κ.α, ή χρησιμοποιούν πληροφορίες από βάσεις δεδομένων (πχ. SQL database). Μπορούν να αυτοματοποιήσουν πολλές ενέργειες διαχείρισης περιεχομένου και δεδομένων. Με αυτά δημιουργούνται δυναμικά sites, που παρουσιάζουν διαφορετικό-μεταβλητό περιεχόμενο ανάλογα με τον χρήστη, την ώρα ή άλλες παραμέτρους. Τα συστήματα διαχείρισης περιεχομένου (content management system ή CMS) εφαρμόζουν τις τεχνικές αυτές, μαζί με μια βάση δεδομένων για την άντληση και αποθήκευση των περιεχομένων. Οι επισκέπτες του ιστοχώρου δεν επιβαρύνονται με την εκτέλεση των προγραμμάτων αυτών και δεν υπάρχουν γι' αυτό το λόγο ασυμβατότητες. Το βάρος της εκτέλεσής τους, επωμίζεται ο server του site.

Πρέπει να συντηρείται το υποσύστημα του server, με συνεχείς αναβαθμίσεις, για την αποφυγή προβλημάτων ασφαλείας και σταθερότητας. Όμως, ακόμα και με την σωστή συντήρηση που κλείνει τις γνωστές τρύπες, είναι πιθανό σημείο εισόδου για απόκτηση ελέγχου του site, από κακόβουλους χρήστες ή κώδικα που εκμεταλλεύονται νεότερες αδυναμίες. Ένα site που εφαρμόζει server-side scripting, έχει διαφορετικές απαιτήσεις στην επεξεργαστική ισχύ του server, ανάλογα με τα προγράμματα που θα

εκτελούνται, τον αριθμό των επισκεπτών του site και τον φόρτο του υπολογιστή αυτού συνολικά.

- Τέλος, συναντάμε αρκετές φορές συνδυασμό των παραπάνω. Client-side scripting για πρόσβαση στα δεδομένα του browser και γρήγορη εκτέλεση τοπικά και server-side scripting για δυναμικά περιεχόμενα στις σελίδες και χρήση βάσεων δεδομένων. Κλασικό παράδειγμα είναι η AJAX. Με αυτόν τον συνδυασμό, μπορεί να επιτευχθεί δυναμική αλλαγή περιεχομένων στις ιστοσελίδες, χωρίς ξαναφόρτωμα αυτών.

2.2 Web Server (Εξυπηρετητής δικτύου)

Εξυπηρετητής ή διακομιστής (αγγλ.: server) είναι υλικό ή / και λογισμικό που αναλαμβάνει την παροχή διάφορων υπηρεσιών, «εξυπηρετώντας» αιτήσεις από άλλους υπολογιστές γνωστούς ως πελάτες (clients). Όταν ένας υπολογιστής εκτελεί κυρίως τέτοια προγράμματα εξυπηρετητές συνεχόμενα, 24 ώρες την ημέρα, τότε μπορούμε να αναφερθούμε σε όλον τον υπολογιστή ως εξυπηρετητή, αφού αυτή είναι η κύρια λειτουργία του. Παρομοίως, ως πελάτη μπορούμε να θεωρήσουμε είτε κάποιο λογισμικό που επικοινωνεί και υποβάλει αιτήματα στον εξυπηρετητή, είτε τον υπολογιστή στον οποίο εκτελείται το λογισμικό πελάτης.

Η επικοινωνία μεταξύ πελάτη και εξυπηρετητή γίνεται μέσω ενός τοπικού δικτύου, ή ακόμα και μέσω του Διαδικτύου. Ο εξυπηρετητής είναι συνήθως ένας υπολογιστής που διαφέρει ως προς τη σύνθεσή του από άλλους κοινούς υπολογιστές, μιας και οι δυνατότητες του είναι σαφώς αναβαθμισμένες. Κύρια χαρακτηριστικά ενός εξυπηρετητή είναι οι επεξεργαστές που υποστηρίζει και χρησιμοποιεί για την επεξεργασία των δεδομένων που δέχεται, οι γρήγοροι και μεγάλης χωρητικότητας σκληροί δίσκοι αλλά και οι ταχύτητες μνήμης που υποστηρίζει. Συνήθως συνοδεύεται από σύστημα διπλής τροφοδοσίας (dual power supply) και από συσκευή παροχής αδιάλειπτης ενέργειας (UPS), για μεγαλύτερη αξιοπιστία και σιγουριά στις παρεχόμενες υπηρεσίες του.

Οι υπολογιστές που χρησιμοποιούνται ως εξυπηρετητές πρέπει να μπορούν να δουλεύουν όλη την ημέρα και συνεχώς χωρίς διακοπές. Για το λόγο αυτό οι εταιρείες κατασκευής υπολογιστών και τμημάτων υπολογιστών προσφέρουν ξεχωριστές κατασκευές για εξυπηρετητές. Φυσικά αυτές οι κατασκευές μπορούν να χρησιμοποιηθούν και για προσωπικό υπολογιστή όμως ανεβάζουν το κόστος.

Συνηθισμένα χαρακτηριστικά για το υλικό είναι:

- μητρικές πλακέτες με μεγάλη αντοχή στις θερμοκρασίες
- ενσωματωμένα συστήματα RAID
- ενσωματωμένη κάρτα οθόνης μικρής μνήμης και ταχύτητας
- πιο άνετα καταναμημένα τα στοιχεία της μνήμης και του επεξεργαστή ώστε να είναι δυνατή η ψύξη και των δύο από ξεχωριστά δυνατά ανεμιστηράκια
- λιγότερες θύρες επέκτασης
- περισσότερα ενδεικτικά στοιχεία λειτουργίας
- δυνατότητα χρήσης μνήμης ECC
- κουτιά βαριά με ειδικό κλείδωμα
- θέσεις για περισσότερα ανεμιστηράκια
- ειδικά διαμορφωμένους εσωτερικούς χώρους ώστε να γίνεται πιο καλή κυκλοφορία του αέρα
- ειδικά κουτιά που μπορούν να ενσωματωθούν σε rack
- δίσκοι με προδιαγραφές για μεγαλύτερο συνεχόμενο χρόνο χρήσης που μπορούν να αποσυνδέονται την ώρα που είναι σε λειτουργία

Τα περισσότερα λειτουργικά συστήματα έχουν ειδική έκδοση για χρήση ως εξυπηρετητή. Τα χαρακτηριστικά που την διαφοροποιούν είναι πολλά μερικά από τα οποία είναι: δυνατότητα λειτουργίας χωρίς την ύπαρξη:

- πληκτρολογίου
- οθόνης
- γραφικού περιβάλλοντος
- κάρτας ήχου
- δυνατότητες επιλογής χρόνου επεξεργασίας των διάφορων προγραμμάτων
- πρόγραμμα επικοινωνίας με το UPS
- δυνατότητα χρήσης περισσότερων του ενός επεξεργαστή
- δυνατότητα συνεπεξεργασίας με άλλους υπολογιστές- εξυπηρετητές
- περισσότερες δυνατότητες αλλαγής του υλικού και αναβάθμισης του λειτουργικού χωρίς την ανάγκη επανεκκίνησης του συστήματος
- περισσότερα συστήματα ασφαλείας

Προγράμματα

Πολλά σύγχρονα προγράμματα δουλεύουν με τη λογική πελάτη - εξυπηρετητή. Ακόμα και το ίδιο το λειτουργικό σύστημα δουλεύει με αυτήν τη λογική. Τα προγράμματα ζητάνε κάποια ενέργεια και το λειτουργικό σύστημα αναλαμβάνει να τα εξυπηρετήσει εκτελώντας τις λειτουργίες που του ζητήθηκαν.

Συνήθως τα περισσότερα προγράμματα εξυπηρετητών απαιτούν και ξεχωριστό υπολογιστή-εξυπηρετητή χωρίς αυτό να είναι πάντα απαραίτητο.

Συνηθισμένοι εξυπηρετητές-προγράμματα σε περιβάλλον γραφείου που μπορεί να βρίσκονται ή όχι σε ξεχωριστούς υπολογιστές είναι:

- Εξυπηρετητής αρχείων (file server)
- Εξυπηρετητής εκτυπωτών (printer server)
- Εξυπηρετητής αντιγράφων ασφαλείας (backup server)
- Εξυπηρετητής βάσεων δεδομένων (database server)
- Εξυπηρετητής φαξ (fax server)
- Εξυπηρετητής διαμεσολαβητή (proxy server)
- Εξυπηρετητής ηλεκτρονικού ταχυδρομείου IMAP

Συνηθισμένοι εξυπηρετητές-προγράμματα στο Ίντερνετ είναι:

- Παγκόσμιου Ιστού (World Wide Web, web server)
- Domain Name System (DNS server)
- Ηλεκτρονικού ταχυδρομείου (mail server)
- Μεταφοράς αρχείων με το πρωτόκολλο FTP (ftp server)
- irc και instant messaging (irc server)
- Επικοινωνίες φωνής
- streaming audio και video (streaming server)
- Online παιχνίδια

Κάθε εξυπηρετητής έχει διαφορετικές ανάγκες για υλικό. Συνήθως έχει ανάγκη από γρήγορη πρόσβαση στο δίκτυο όμως ένας εξυπηρετητής φαξ δεν θα υποφέρει τόσο αν δεν έχει γρήγορη κάρτα δικτύου ή αρκετή μνήμη όσο το να έχει προβληματικό modem.

Ένας εξυπηρετητής αρχείων και ένας εξυπηρετητής αντιγράφων ασφαλείας είναι προτιμότερο να έχουν όσο γίνεται πιο αξιόπιστους και πιο ταχείς δίσκους από το να έχουν μεγαλύτερο επεξεργαστή ή περισσότερη και ακριβότερη μνήμη.

Ένας εξυπηρετητής DNS χρειάζεται όσο το δυνατό μεγαλύτερη ασφάλεια και γι αυτό το λόγο εκείνο που χρειάζεται περισσότερο είναι να τρέχουν όσο το δυνατόν λιγότερα προγράμματα.

2.3 ASP.NET

Η ASP.NET είναι μια τεχνολογία διαδικτυακών εφαρμογών που έχει αναπτυχθεί και διατίθεται στο εμπόριο από τη Microsoft για να επιτρέπει στους προγραμματιστές την δημιουργία δυναμικών ιστοσελίδων, διαδικτυακές εφαρμογές και υπηρεσίες. Κυκλοφόρησε για πρώτη φορά τον Ιανουάριο του 2002 με την έκδοση 1.0 του .NET Framework, και είναι ο διάδοχος της παλιότερης ASP τεχνολογίας της Microsoft. Η ASP.NET είναι χτισμένη στο Common Language Runtime (CLR), επιτρέποντας στους προγραμματιστές να γράφουν κώδικα ASP.NET χρησιμοποιώντας οποιαδήποτε υποστήριξη της πλατφόρμας .NET.

2.3.1 Ιστορική Αναδρομή

Μετά την κυκλοφορία του Internet Information Services 4.0 (IIS), το 1997, η Microsoft άρχισε να ερευνά τις δυνατότητες για ένα νέο μοντέλο διαδικτυακών εφαρμογών που θα λύσουν συνηθισμένα προβλήματα που αφορούν την ASP, ιδίως σε σχέση με το διαχωρισμό της παρουσίασης και του περιεχομένου και να είναι σε θέση να γράψει "καθαρό" κώδικα. Ο Mark Anders, ο διευθυντής στην ομάδα IIS, και ο Scott Guthrie, ο οποίος είχε προσχωρήσει στη Microsoft το 1997, μετά την αποφοίτησή του από το Πανεπιστήμιο Duke, ήταν υπεύθυνοι για τον καθορισμό του μοντέλου. Ο αρχικός σχεδιασμός αναπτύχθηκε κατά τη διάρκεια των δύο μηνών από τον Anders και Guthrie.

Το αρχικό πρωτότυπο ονομάζεται "XSP". Ο Guthrie εξήγησε σε μια συνέντευξή του το 2007 ότι, «Οι άνθρωποι θα ρωτάνε πάντα τι είναι το X, ενώ στην πραγματικότητα δεν είχε κάποια σημασία. Το XML ξεκίνησε με αυτό, το XSLT ξεκίνησε με αυτό. Όλα τα «cool» πράγματα φαίνεται να ξεκινάνε με ένα X, γι' αυτό και ονομάστηκε έτσι αρχικά ».

Το αρχικό πρωτότυπο του XSP είχε γίνει με τη χρήση Java, αλλά σύντομα αποφάσισε να χτίσει τη νέα πλατφόρμα που στηρίζεται στην Κοινή Γλώσσα Runtime (CLR), δεδομένου ότι προσφέρει ένα αντικειμενοστραφές περιβάλλον προγραμματισμού, συλλογή σκουπιδιών και άλλα χαρακτηριστικά που θεωρήθηκαν ως επιθυμητά χαρακτηριστικά που το μοντέλο αντικειμενοστραφούς πλατφόρμας περιεχομένου της Microsoft δεν υποστήριζε. Ο Guthrie αναφέρει ότι η απόφαση αυτή είναι ένας "τεράστιος κίνδυνος", καθώς η επιτυχία της νέας πλατφόρμας ανάπτυξης Ιστού θα πρέπει να συνδέεται με την επιτυχία της CLR, η οποία, όπως και η XSP, ήταν ακόμα στα πρώτα στάδια της ανάπτυξης.

Με την κίνηση προς το CLR, η XSP εφαρμόστηκε στη C # και το όνομά της άλλαξε σε ASP +, και από αυτό το σημείο η νέα πλατφόρμα θεωρήθηκε ως ο διάδοχος του Active Server Pages, και η πρόθεση ήταν να παρέχει έναν εύκολο δρόμο για την ανάπτυξη της ASP.

Ο Mark Anders παρουσίασε για πρώτη φορά την ASP + στο συνέδριο ASP Συνδέσεις στο Φοίνιξ της Αριζόνα στις 2 Μαΐου, 2000. Οι επιδείξεις στο ευρύ κοινό και η αρχική έκδοση beta του ASP + (και το υπόλοιπο του .NET Framework) έγιναν στη Διάσκεψη Επαγγελματιών Developers του 2000, στις 11 Ιουλίου 2000 στο Ορλάντο, Φλόριντα. Κατά τη διάρκεια της κύριας ομιλία του Bill Gates, η Fujitsu παρουσίασε την ASP + να χρησιμοποιείται σε συνδυασμό με COBOL. Επίσης, ανακοινώθηκε η υποστήριξη για μια ποικιλία άλλων γλωσσών, συμπεριλαμβανομένων των νέων Visual της Microsoft Basic. NET

και τη γλώσσα C #, καθώς επίσης και την Python και την Perl υποστήριξη από εργαλεία διαλειτουργικότητας που δημιουργήθηκαν από την ActiveState.

Μετά από τέσσερα χρόνια ανάπτυξης, καθώς και μια σειρά από εκδόσεις beta, το 2000 και το 2001, κυκλοφόρησε η ASP.NET 1.0 στις 5 Ιανουαρίου 2002 ως μέρος της έκδοσης 1.0 του .NET Framework. Ακόμη και πριν από την κυκλοφορία, είχαν γραφτεί δεκάδες βιβλία για ASP.NET, και η Microsoft την προώθησε σε μεγάλο βαθμό ως μέρος της πλατφόρμας της για τις υπηρεσίες Web. Η ανάπτυξη συνεχίστηκε με εντατικούς ρυθμούς, με την έκδοση 1.1 που κυκλοφόρησε στις 24 Απριλίου 2003 ως μέρος του Windows Server 2003. Αυτή η έκδοση εστιάζεται στη βελτιωμένη υποστήριξη της ASP.NET για κινητές συσκευές.

2.3.2 ASP.NET σε σύγκριση με την κλασική ASP

Η ASP.NET απλοποιεί τη μετάβαση των προγραμματιστών από την ανάπτυξη εφαρμογών των Windows στην ανάπτυξη Web εφαρμογών, προσφέροντας τη δυνατότητα να φτιάξουν ιστοσελίδες με στοιχεία ελέγχου παρόμοια με το περιβάλλον εργασίας χρήστη των Windows. Ένα στοιχείο ελέγχου Web, όπως ένα κουμπί ή μια ετικέτα, λειτουργεί σε μεγάλο βαθμό με τον ίδιο τρόπο με τους ομολόγους του των Windows: οι ιδιότητές του μπορούν να εκχωρηθούν με κώδικα και να ανταποκριθούν στα συμβάντα. Η ASP.NET ενθαρρύνει τον προγραμματιστή για την ανάπτυξη εφαρμογών χρησιμοποιώντας ένα event-driven μοντέλο GUI, και όχι τα συμβατικά Web-scripting περιβάλλοντα όπως στην ASP και την PHP. Το πλαίσιο συνδυάζει τις υπάρχουσες τεχνολογίες, όπως η JavaScript με τα εσωτερικά μέρη όπως το "ViewState".

Άλλες διαφορές σε σύγκριση με το κλασικό ASP είναι τα εξής:

- Ο μεταγλωττισμένος κώδικας συμβάλει στο να τρέχουν γρηγορότερα οι εφαρμογές και τα περισσότερα σφάλματα να παγιδεύονται στο στάδιο της ανάπτυξης.
- Σημαντικά βελτιωμένη αντιμετώπιση των λαθών, κάνοντας χρήση των εξαιρέσεων (exception handling) και μπλοκ try-catch.
- Παρόμοια στοιχεία ελέγχου και συμβάντα (events) με τις εφαρμογές των Windows.
- Ένα εκτεταμένο σύνολο ελέγχων και βιβλιοθηκών επιτρέπει την ταχεία ανάπτυξη των εφαρμογών. Διάταξη αυτών των ελέγχων σε μια σελίδα είναι πιο εύκολο, διότι το μεγαλύτερο μέρος του μπορεί να γίνει οπτικά στους περισσότερους συντάκτες.
- Η ASP.NET χρησιμοποιεί την πολυ-γλωσσική ικανότητα της .NET CLR, που επιτρέπει στις ιστοσελίδες να κωδικοποιηθούν σε VB.NET, C#, J#, Delphi.NET, Chrome, κλπ.
- Δυνατότητα αποθήκευσης (cache) ολόκληρης σελίδας ή μόνο τμημάτων της για τη βελτίωση των επιδόσεων.
- Δυνατότητα χρήσης του μοντέλου ανάπτυξης code-behind για το διαχωρισμό της επιχειρηματικής λογικής από την παρουσίαση.
- Δυνατότητα χρήσης πραγματικού αντικειμενοστραφούς σχεδιασμού για τον προγραμματισμό των σελίδων και των ελέγχων.
- Αν μια εφαρμογή ASP.NET έχει διαρροές μνήμης, το ASP.NET runtime ξεφορτώνει το AppDomain που τη φιλοξενεί και φορτώνει την εφαρμογή σε ένα νέο AppDomain.
- Η session state της ASP.NET μπορεί να αποθηκευτεί σε μια βάση δεδομένων Microsoft SQL Server ή σε μια ξεχωριστή διαδικασία που τρέχει στον ίδιο υπολογιστή με το διακομιστή Web ή σε μια διαφορετική μηχανή. Με αυτόν τον τρόπο δεν χάνονται οι τιμές της συνεδρίας όταν ο διακομιστής Web κάνει επαναφορά ή η διαδικασία εργασίας ASP.NET ανανεώνεται.

2.4 CSS

Η Cascading Style Sheets (CSS) είναι μια γλώσσα που χρησιμοποιείται για να περιγράψει την παρουσίαση (την εμφάνιση και τη μορφοποίηση) ενός εγγράφου γραμμένου σε μια γλώσσα σήμανσης. Πιο κοινή εφαρμογή του είναι να προσθέτει στυλ σε ιστοσελίδες γραμμένες σε HTML και XHTML, αλλά η μπορεί επίσης να εφαρμοστεί σε οποιοδήποτε είδος εγγράφου XML, συμπεριλαμβανομένου και του απλού XML, SVG και XUL. Η CSS έχει σχεδιαστεί για να επιτρέπει κυρίως το διαχωρισμό του περιεχομένου του εγγράφου (γραμμένο σε HTML ή παρόμοια γλώσσα σήμανσης) από την παρουσίαση του εγγράφου, συμπεριλαμβανομένων των στοιχείων όπως η διάρθρωση, τα χρώματα και τις γραμματοσειρές. Αυτός ο διαχωρισμός μπορεί να βελτιώσει την προσβασιμότητα του περιεχομένου, να παρέχει μεγαλύτερη ευελιξία και έλεγχο των προδιαγραφών των χαρακτηριστικών παρουσίασης, ώστε πολλές σελίδες να μετέχουν στη μορφοποίηση, καθώς και να μειώσουν την πολυπλοκότητα και την επανάληψη του διαρθρωτικού περιεχομένου (π.χ. παρέχοντας τη δυνατότητα για tableless web design).

Η CSS μπορεί, επίσης, να επιτρέπει στην ίδια σελίδα σημειώσεων να παρουσιαστεί σε διαφορετικά στυλ για τις διάφορες μεθόδους επεξεργασίας, όπως η επί της οθόνης, σε έντυπη μορφή, με τη φωνή (όταν διαβάζονται από μια ομιλία με βάση το πρόγραμμα περιήγησης ή πρόγραμμα ανάγνωσης οθόνης) και σε Braille με βάση, απτικές συσκευές. Μπορεί επίσης να χρησιμοποιηθεί για να επιτρέψει στην ιστοσελίδα να εμφανιστεί με διαφορετικό τρόπο ανάλογα με το μέγεθος της οθόνης ή της συσκευής, στην οποία το παρακολουθείτε. Η CSS καθορίζει ένα σύστημα προτεραιότητας για να καθοριστεί ποιο κανόνες στυλ εφαρμόζονται σε περίπτωση που περισσότεροι του ενός κανόνα αναφέρονται στο ίδιο συγκεκριμένο στοιχείο. Σε αυτήν την περίπτωση, οι προτεραιότητες ή συντελεστές στάθμισης υπολογίζονται και ανατίθενται σε κανόνες, έτσι ώστε τα αποτελέσματα να είναι προβλεπόμενα.

Η CSS έχει μια απλή σύνταξη και χρησιμοποιεί μια σειρά από αγγλικές λέξεις-κλειδιά για να καθορίσετε τα ονόματα των διαφόρων ιδιοτήτων μορφοποίησης. Ένα φύλλο μορφοποίησης αποτελείται από μια λίστα με τους κανόνες. Κάθε κανόνας ή ομάδα κανόνων αποτελείται από μία ή περισσότερες επιλογές και ένα τμήμα της δήλωσης. Μια δήλωση-μπλοκ αποτελείται από μια λίστα με δηλώσεις σε αγκύλες. Κάθε δήλωση αποτελείται από μια ιδιότητα, μια άνω και κάτω τελεία (:) και μια τιμή. Εάν υπάρχουν πολλαπλές δηλώσεις σε ένα μπλοκ, ένα ερωτηματικό (?) πρέπει να εισάγεται σε κάθε ξεχωριστή δήλωση.

Στην CSS, επιλογείς χρησιμοποιούνται για να δηλώσουν ποια από τα στοιχεία σήμανσης της μορφοποίησης εφαρμόζονται. Οι επιλογείς μπορούν να εφαρμόζονται σε όλα τα στοιχεία ενός συγκεκριμένου τύπου, ή μόνο εκείνα τα στοιχεία που ταιριάζουν με ένα συγκεκριμένο χαρακτηριστικό. Τα στοιχεία μπορούν να συνδυαστούν ανάλογα με το πώς είναι τοποθετημένα σε σχέση με κάθε άλλο στοιχείο στον κώδικα σήμανσης, ή για το πώς είναι ένθετα εντός του Μοντέλου Αντικειμένου του Εγγράφου.

Ψευδο-κλάση είναι μια άλλη μορφή των προδιαγραφών που χρησιμοποιούνται στη CSS για τον προσδιορισμό στοιχείων σήμανσης, καθώς και σε ορισμένες περιπτώσεις, συγκεκριμένες ενέργειες του χρήστη για τον οποίο ισχύει ένα συγκεκριμένο τμήμα της δήλωσης. Ένα συχνά χρησιμοποιούμενο παράδειγμα είναι η `:hover` ψευδο-κλάση που εφαρμόζει ένα στυλ μόνο όταν τα «σημεία», ο χρήστης δείχνει ένα ορατό στοιχείο, συνήθως κρατώντας τον κέρσορα του ποντικιού πάνω από αυτό. Πρόκειται να προστεθούν σε έναν επιλογέα όπως σε ένα `:hover` ή `# elementid: hover`. Άλλες ψευδο-κλάσεις και ψευδο-στοιχεία είναι, για παράδειγμα, «`:first-line`», «`:visited`» ή «`:before`».

Πριν από την CSS, σχεδόν όλα τα χαρακτηριστικά παρουσίασης των HTML εγγράφων περιέχονταν στο κείμενο HTML. Όλα τα χρώματα γραμματοσειράς, στυλ φόντου, ευθυγραμμίσεις στοιχείων, τα περιθώρια και τα μεγέθη επρεπε να περιγράφεται ρητά, συχνά κατ'επανάληψη, στο πλαίσιο της HTML. Η CSS επιτρέπει στους συγγραφείς να μετακινήσουν μεγάλο μέρος των πληροφοριών αυτών σε ένα ξεχωριστό φύλλο στυλ με αποτέλεσμα την σημαντικά απλούστερη σήμανση HTML. Οι τίτλοι (`h1` στοιχεία), οι υποτομείς (`h2`), οι υποκατηγορίες (`h3`), κλπ., ορίζονται δομικά με τη χρήση HTML.

Πριν από την CSS, οι συντάκτες του εγγράφου που ήθελαν να αναθέσουν τέτοια τυπογραφικά χαρακτηριστικά, για παράδειγμα, το σύνολο των επικεφαλίδων h2 έπρεπε να χρησιμοποιεί την HTML font εντολή και άλλα στοιχεία παρουσίασης για κάθε εμφάνιση της κλάσης τύπου. Η πρόσθετη σήμανση παρουσίασης στην HTML έκανε πιο πολύπλοκα τα έγγραφα, και γενικά πιο δύσκολο να συντηρηθούν. Στην CSS, η παρουσίαση χωρίζεται από τη δομή. Στην εκτύπωση, η CSS μπορεί να καθορίσει το χρώμα, τη γραμματοσειρά, τη στοίχιση του κειμένου, το μέγεθος, τα σύνορα, την απόσταση, τη διάταξη και πολλά άλλα τυπογραφικά χαρακτηριστικά. Αυτό μπορεί να το κάνει ανεξάρτητα αν είναι για προβολή στην οθόνη ή εκτυπωμένη προβολή. Η CSS ορίζει επίσης τα μη-οπτικά στυλ, όπως η ταχύτητα και η έμφαση με την οποία διαβάζουν το κείμενο οι ακουστικοί αναγνώστες κειμένου. Το W3C θεωρεί τώρα τα πλεονεκτήματα των CSS για τον ορισμό όλων των πτυχών της παρουσίασης της HTML σελίδας ανώτερη από άλλες μεθόδους. Έχει αποδοκιμάσει κατά συνέπεια τη χρήση όλης της αρχικής παρουσίασης σήμανσης HTML.

Τα CSS αρχεία εισάγονται στα έγγραφα HTML χρησιμοποιώντας την ακόλουθη σύνταξη:

```
<link rel="stylesheet" href="http://example.com/css/style.css" type="text/css" />
```

Οι CSS πληροφορίες μπορούν να παρέχονται από διάφορες πηγές. Οι πληροφορίες στυλ είτε επισυνάπτονται ως χωριστό έγγραφο ή ενσωματώνονται στο έγγραφο HTML. Μπορούν να εισαχθούν πολλαπλά φύλλα στυλ. Μπορούν να εφαρμοστούν διαφορετικά στυλ ανάλογα με τη συσκευή εξόδου που χρησιμοποιείται. Για παράδειγμα, η έκδοση της οθόνης μπορεί να είναι αρκετά διαφορετική από την έντυπη έκδοση, έτσι ώστε οι συγγραφείς να μπορούν να προσαρμόσουν την παρουσίαση κατάλληλα για κάθε μέσο.

Ένας από τους στόχους της CSS είναι, επίσης, να επιτρέψει στους χρήστες μεγαλύτερο έλεγχο της παρουσίασης. Κάποιος που βρίσκει τους κόκκινους πλάγιους τίτλους δύσκολο να διαβαστούν μπορεί να εφαρμόσει ένα διαφορετικό στυλ παρουσίασης. Ανάλογα με το πρόγραμμα περιήγησης και το web site, ο χρήστης μπορεί να επιλέξει από διάφορα στυλ που παρέχονται από τους σχεδιαστές, μπορεί να αφαιρέσει όλες τις μορφοποιήσεις και να δει την ιστοσελίδα χρησιμοποιώντας το προεπιλεγμένο στυλ του browser, ή μπορεί να αντικαταστήσει μόνο το κόκκινο πλάγιο στυλ επικεφαλίδας χωρίς να αλλοιώνει τα υπόλοιπα χαρακτηριστικά.

2.4.1 Ιστορική αναδρομή

Τα διαδοχικά φύλλα στυλ έχουν υπάρξει με τη μία μορφή ή την άλλη από τις αρχές του SGML στη δεκαετία του 1970. Τα επικαλυπτόμενα φύλλα στυλ αναπτύχθηκαν ως μέσο για τη δημιουργία μιας συνεκτικής προσέγγισης για την παροχή πληροφοριών για το στυλ των εγγράφων διαδικτύου.

Καθώς αναπτυσσόταν η HTML, έπρεπε να συμπεριλάβει μια ευρύτερη ποικιλία των στυλιστικών δυνατοτήτων για να ανταποκριθεί στις απαιτήσεις των σχεδιαστών. Αυτή η εξέλιξη έδωσε στους σχεδιαστές περισσότερο έλεγχο πάνω στην εμφάνιση των ιστότοπων αλλά με το κόστος την όλο και πιο περίπλοκη για να γραφτεί και να επεξεργαστεί HTML. Παραλλαγές σε πρόγραμμα περιήγησης, όπως το ViolaWWW και το WorldWideWeb, έκανε δύσκολη την ομοιόμορφη εμφάνιση των ιστοσελίδων, και οι χρήστες είχαν λιγότερο έλεγχο για το πώς εμφανίζεται το περιεχόμενο. Έτσι, ο Robert Cailliau θέλησε να διαχωρίσει τη δομή από την παρουσίαση. Ο ιδανικός τρόπος θα ήταν να δώσει στο χρήστη διάφορες επιλογές και να διαθέτει τρία διαφορετικά είδη φύλλων στυλ: Ένα για την εκτύπωση, ένα για την παρουσίαση στην οθόνη και ένα για την επεξεργασία.

Για να βελτιωθεί η παρουσίαση των δυνατοτήτων του διαδικτύου, εννέα διαφορετικές γλώσσες φύλλων στυλ προτάθηκαν στην αλληλογραφία της Κοινοπραξίας του Παγκοσμίου Ιστού. Από τις εννέα προτάσεις, δύο επελέγησαν ως τα θεμέλια για αυτό που έγινε αργότερα η CSS, τα επικαλυπτόμενα φύλλα στυλ της HTML (CHSS) και η πρόταση

φύλλων στυλ που βασίζεται σε ροή δεδομένων (SSP). Η CHSS, μια γλώσσα που έχει κάποια ομοιότητα με τη CSS σήμερα, προτάθηκε από τον Håkon Wium Lie τον Οκτώβριο του 1994. Ο Bert Bos εργαζόταν σε ένα πρόγραμμα περιήγησης που ονομάζεται Argo, το οποίο χρησιμοποιούσε δική του γλώσσα φύλλων στυλ που ονομάζεται SSP. Ο Lie και ο Yves Lafon συνεργάστηκαν με τον Dave Raggett για να επεκτείνουν το πρόγραμμα περιήγησης Arena για την υποστήριξη της CSS ως πλατφόρμα δοκιμών για την εφαρμογή του W3C. Ο Lie και ο Bos εργάστηκαν μαζί για να αναπτύξουν το πρότυπο CSS (το «H» αφαιρέθηκε από το όνομα, επειδή αυτά τα φύλλα στυλ θα μπορούσαν επίσης να εφαρμοστούν και σε άλλες γλώσσες σήμανσης, εκτός από την HTML).

Σε αντίθεση με τις υπάρχουσες γλώσσες στυλ όπως DSSSL και FOSI, η CSS επιτρέπεται το στυλ ενός εγγράφου να επηρεάζεται από πολλαπλά φύλλα στυλ. Ένα φύλλο στυλ θα μπορούσε να κληρονομήσει από κάποιο άλλο, επιτρέποντας έτσι ένα μείγμα από στυλιστικές προτιμήσεις που ελέγχονται επίσης από το σχεδιαστή της ιστοσελίδας, αλλά και των χρηστών.

Η πρόταση του Lie παρουσιάστηκε στο συνέδριο "Mosaic and the Web" (αργότερα ονομάστηκε www2) στο Σικάγο, Ιλλινόις το 1994, και πάλι με τον Bert Bos το 1995. Εκείνο τον καιρό το W3C είχε ήδη συσταθεί, και υπήρχε ενδιαφέρον για την ανάπτυξη της CSS. Οργανώθηκε ένα εργαστήριο για το σκοπό αυτό υπό την προεδρία του Steven Pemberton. Αυτό είχε ως αποτέλεσμα την προσθήκη εργασιών του W3C. Ο Lie και ο Bos ήταν το κύριο τεχνικό προσωπικό σε αυτή την πτυχή του έργου, με πρόσθετα μέλη, συμπεριλαμβανομένου του Thomas Reardon της Microsoft, που επίσης συμμετείχαν. Τον Αύγουστο του 1996 ανακοίνωση της Netscape Corporation παρουσίασε μια εναλλακτική γλώσσα φύλλων στυλ που ονομάζεται JavaScript Style Sheets (JSSS). Οι προδιαγραφές δεν είχαν ποτέ ολοκληρωθεί και καταργήθηκαν. Μέχρι το τέλος του 1996, η CSS ήταν έτοιμη να γίνει επίσημη, και η σύσταση επιπέδου 1 της CSS δημοσιεύθηκε το Δεκέμβριο.

Η ανάπτυξη της HTML, της CSS, και του DOM είχε γίνει από μία ομάδα, την HTML Editorial Review Board (ERB). Στις αρχές του 1997, το ERB χωρίστηκε σε τρεις ομάδες εργασίας: Ομάδα εργασίας HTML, υπό την προεδρία του Dan Connolly του W3C, ομάδα εργασίας του DOM, υπό την προεδρία του Lauren Wood της SoftQuad και ομάδα εργασίας της CSS, υπό την προεδρία του Chris Lilley του W3C.

Η Ομάδα Εργασίας της CSS ξεκίνησε την αντιμετώπιση θεμάτων που δεν είχαν αντιμετωπιστεί με στο επίπεδο 1 της CSS, με αποτέλεσμα τη δημιουργία του επιπέδου 2 της CSS στις 4 Νοεμβρίου 1997. Η δημοσίευση έγινε ως Σύσταση του W3C στις 12 Μαΐου, 1998. Το επίπεδο 3 της CSS, το οποίο ξεκίνησε το 1998, εξακολουθεί να βρίσκεται υπό ανάπτυξη από το 2009.

Το 2005, η CSS Ομάδα Εργασίας αποφάσισε να επιβάλλει πιο αυστηρά τις απαιτήσεις για τα πρότυπα. Αυτό σήμαινε ότι τα ήδη υπάρχοντα πρότυπα όπως το CSS 2.1, CSS 3 Selectors και το κείμενο της CSS 3 είχαν αποσυρθεί από την Υποψήφια Σύσταση σε επίπεδο επεξεργασίας σχεδίου.

Αν και οι προδιαγραφές CSS 1 ολοκληρώθηκαν το 1996 και ο Microsoft Internet Explorer 3 εκδόθηκε εκείνο το έτος είχε περιορισμένη υποστήριξη για τη CSS, πέρασαν περισσότερα από τρία έτη πριν οποιοσδήποτε web browser καταφέρει να έχει σχεδόν πλήρη εφαρμογή των προδιαγραφών. Ο Internet Explorer 5.0 για Macintosh, τα οποία εκδόθηκαν το Μάρτιο του 2000, ήταν ο πρώτος browser που είχε πλήρη (καλύτερη από το 99 τοις εκατό) υποστήριξη CSS 1, ξεπερνώντας το Opera, που ήταν το καλύτερο στην υποστήριξη CSS 15 μήνες νωρίτερα. Άλλοι browsers ακολούθησαν αμέσως μετά, και πολλοί από αυτούς επιπλέον υλοποιούσαν τμήματα του CSS 2. Μέχρι τον Αύγουστο του 2010, κανένας (τελικά) browser δεν είχε εφαρμόσει πλήρως την CSS 2, με διαφορετικά επίπεδα εφαρμογής. Αν και νωρίς browsers όπως ο Internet Explorer 3 και 4, και ο Netscape 4.x είχαν υποστήριξη για CSS, ήταν συνήθως ελλιπής και με σοβαρά σφάλματα. Αυτό ήταν ένα σοβαρό εμπόδιο για την υιοθέτηση του CSS.

Όταν αργότερα οι browsers "5ης έκδοσης" άρχισαν να προσφέρουν μια αρκετά πλήρη εφαρμογή του CSS, ήταν ακόμα λάθος σε ορισμένα σημεία και ήταν γεμάτα σφάλματα, ασυνέπειες, και άλλες ιδιορρυθμίες. Η πληθώρα αυτών των σφαλμάτων που σχετίζονται με

τις ασυνέπειες και ακόμη και η διακύμανση της υποστήριξης της CSS είχε καταστήσει δύσκολο για τους σχεδιαστές να επιτευχθεί μια ομοιόμορφη εμφάνιση σε όλες τις πλατφόρμες. Μερικοί συντάκτες κατέφυγαν σε λύσεις όπως CSS αμυχές και CSS φίλτρα για να αποκτήσουν σταθερά αποτελέσματα σε όλους τους web browsers και πλατφόρμες.

Προβλήματα με την αποσπασματική υιοθέτηση της CSS από τους browsers μαζί με διορθώσεις στις αρχικές προδιαγραφές οδήγησαν το W3C στην αναθεώρηση του CSS 2 προτύπου σε CSS 2.1, το οποίο κινήθηκε πιο κοντά σε ένα λειτουργικό στιγμιότυπο της τρέχουσας στήριξης για τη CSS σε HTML browsers. Μερικές CSS 2 ιδιότητες που δεν υλοποιούνταν με επιτυχία αποσύρθηκαν, και σε λίγες περιπτώσεις, συμπεριφορές άλλαξαν για να αναβαθμιστεί η συνοχή στις κυρίαρχες από τις υπάρχουσες υλοποιήσεις.

Στο παρελθόν, ορισμένοι διακομιστές διαδικτύου είχαν ρυθμιστεί ώστε να εξυπηρετούν όλα τα έγγραφα με την επέκταση ονόματος αρχείου ".css" ως mime τύπου εφαρμογές και όχι text / css. Εκείνη την εποχή, υπήρχε λογισμικό που μετέτρεπε αρχεία PowerPoint σε Compact αρχεία παρουσίασης (χρησιμοποιώντας μια επέκταση .css).

2.4.2. Εκδόσεις

Η CSS έχει διάφορα επίπεδα και προφίλ. Κάθε επίπεδο του CSS βασίζεται στο τελευταίο, συνήθως προσθέτοντας νέα χαρακτηριστικά και συμβολίζεται ως CSS1, CSS2, CSS3, και CSS4. Τα προφίλ είναι συνήθως ένα υποσύνολο από ένα ή περισσότερα επίπεδα CSS που έχουν κατασκευαστεί για μια συγκεκριμένη συσκευή ή διεπαφή χρήστη. Αυτήν την περίοδο υπάρχουν προφίλ για κινητές συσκευές, εκτυπωτές, και τηλεοράσεις. Το προφίλ δεν πρέπει να συγχέεται με τους τύπους μέσων, τα οποία προστέθηκαν στο CSS2.

CSS1

Η πρώτη προδιαγραφή CSS που έγινε μια επίσημη σύσταση του W3C είναι η CSS επιπέδου 1, που δημοσιεύθηκε τον Δεκέμβριο του 1996. Μεταξύ των δυνατοτήτων της είναι η στήριξη για:

- Ιδιότητες γραμματοσειράς, όπως η γραμματοσειρά και η έμφαση
- Χρώμα του φόντου, του κειμένου, και άλλα στοιχεία
- Χαρακτηριστικά του κειμένου, όπως η απόσταση μεταξύ των λέξεων, γραμμάτων, και τις γραμμές του κειμένου
- Η ευθυγράμμιση των εικόνων κειμένου, πίνακες και άλλα στοιχεία
- Περιθώρια και τη θέση για τα περισσότερα στοιχεία
- Μοναδική ταυτοποίηση και την ταξινόμηση των ομάδων των ιδιοτήτων

Το W3C δεν διατηρεί την σύσταση CSS1.

CSS2

Το επίπεδο 2 του CSS προδιαγραφών αναπτύχθηκε από το W3C και δημοσιεύεται ως σύσταση το Μάιο του 1998. Ένα υπερσύνολο του CSS1, CSS2 περιλαμβάνει μια σειρά από νέες δυνατότητες, όπως απόλυτη, σχετική και σταθερή τοποθέτηση των στοιχείων και z-index, η έννοια των τύπων μέσων, η στήριξη για τα ακουστικά φύλλα στυλ και αμφίδρομο κείμενο, και νέες ιδιότητες γραμματοσειράς, όπως σκιές.

Το W3C δεν διατηρεί την σύσταση CSS2.

CSS 2.1

Το επίπεδο 2.1 του CSS, που συχνά αναφέρεται ως "CSS 2.1", διορθώνει λάθη του CSS2, αφαιρεί κακώς υποστηριζόμενα ή όχι πλήρως λειτουργικά χαρακτηριστικά και προσθέτει ήδη εφαρμοσμένα με τις προδιαγραφές χαρακτηριστικά σε επεκτάσεις του προγράμματος περιήγησης. Προκειμένου να συμμορφωθεί με τη Διαδικασία του W3C για την τυποποίηση των τεχνικών προδιαγραφών, η έκδοση CSS 2.1 ταλαντεύθηκε μεταξύ

Κατασκευή ηλεκτρονικού καταστήματος

κατάστασης επεξεργασίας σχεδίου και κατάστασης υποψήφιας σύστασης για πολλά χρόνια. Η CSS 2.1 έγινε για πρώτη φορά υποψήφια σύσταση στις 25 Φεβρουαρίου του 2004, αλλά επανήλθε σε υπό διαμόρφωση προσχέδιο στις 13 Ιουνίου 2005, για περαιτέρω εξέταση. Επέστρεψε ως υποψήφια σύσταση στις 19 Ιουλίου 2007 και στη συνέχεια ενημερώθηκε δύο φορές το 2009. Ωστόσο, παρόλες τις αλλαγές και τις διευκρινίσεις που έγιναν, και πάλι επέστρεψε στη διαμόρφωση προσχέδιου στις 7 Δεκεμβρίου 2010 και τελικά δημοσιεύθηκε στις 7 Ιουνίου 2011.

CSS3

Σε αντίθεση με τη CSS2, όπου μια μεγάλη ενιαία προδιαγραφή καθορίζει τα διάφορα χαρακτηριστικά γνωρίσματα, η CSS3 χωρίζεται σε αρκετά χωριστά έγγραφα που ονομάζονται «ενότητες». Κάθε ενότητα προσθέτει νέες δυνατότητες ή επεκτείνει τα χαρακτηριστικά που ορίζονται στο CSS2, διατηρώντας τη συμβατότητα. Οι εργασίες σχετικά με τη CSS επιπέδου 3 ξεκίνησαν το χρόνο δημοσίευσης της αρχικής σύστασης CSS2. Τα πρώτα σχέδια CSS3 δημοσιεύθηκαν τον Ιούνιο του 1999.

Λόγω της αρθρωτής σύνδεσης, διαφορετικές ενότητες έχουν διαφορετική σταθερότητα και βρίσκονται σε διαφορετική κατάσταση. Από τον Νοέμβριο του 2011, υπάρχουν πάνω από πενήντα CSS ενότητες που δημοσιεύθηκαν από την Ομάδα Εργασίας CSS. Τρεις από αυτές-επιλογείς, χώροι ονομάτων και χρώμα- έγιναν Σύσταση του W3C το 2011.

Μερικές ενότητες, όπως το φόντο και τα χρώματα, πολυμεσικά ερωτήματα, πολλαπλή-στήλη διάταξης είναι σε κατάσταση υποψήφιας σύστασης και θεωρούνται μετρίως σταθερά.

Υποστήριξη του προγράμματος περιήγησης

Επειδή δεν αναλύουν όλα τα προγράμματα περιήγησης σωστά τον CSS κώδικα, αναπτύχθηκαν κωδικοποιήσεις γνωστά ως CSS αμυχές για να φιλτράρουν είτε συγκεκριμένα προγράμματα περιήγησης ή να στοχεύουν σε συγκεκριμένα προγράμματα περιήγησης (γενικά και τα δύο είναι γνωστά ως CSS φίλτρα). Το πρώτο μπορεί να οριστεί ως CSS φιλτράρισμα και το τελευταίος μπορεί να οριστεί ως στοχευμένες αμυχές CSS και τα δύο από τα οποία μπορούν να χρησιμοποιηθούν για να αποκρύψετε ή να εμφανίσετε τμήματα του CSS σε διαφορετικά προγράμματα περιήγησης. Αυτό επιτυγχάνεται είτε με την αξιοποίηση CSS-χειρισμού ιδιοτροπίες ή σφάλματα στο πρόγραμμα περιήγησης, είτε με την αξιοποίηση της έλλειψης υποστήριξης για τμήματα των προδιαγραφών CSS. Χρησιμοποιώντας CSS φίλτρα, μερικοί σχεδιαστές παρέχουν διαφορετικά CSS σε ορισμένα browsers για να εξασφαλίσουν ότι το αποτέλεσμα θα είναι όπως αναμένεται. Επειδή πολύ παλιά προγράμματα περιήγησης διαδικτύου είτε ήταν εντελώς ανίκανα για χειρισμό κώδικα CSS, ή τον επεξεργάζονταν ελλιπώς, οι σχεδιαστές σήμερα συχνά χρησιμοποιούν CSS φίλτρα που εμποδίζουν εντελώς αυτά τα προγράμματα περιήγησης από την πρόσβαση σε οποιοδήποτε από τα CSS. Η υποστήριξη Internet Explorer για CSS ξεκίνησε με IE 3.0 και αυξάνεται σταδιακά με την κάθε έκδοση. Μέχρι το 2008, η πρώτη beta έκδοση του Internet Explorer 8 προσφέρει υποστήριξη για CSS 2.1 .

Ένα παράδειγμα ενός γνωστού bug του προγράμματος περιήγησης CSS είναι το σφάλμα μοντέλου του Internet Explorer, όπου τα πλάτη του πλαισίου ερμηνεύονται λανθασμένα σε διάφορες εκδόσεις του προγράμματος περιήγησης, με αποτέλεσμα τα μπλοκ να είναι πάρα πολύ στενά όταν προβάλλονται στον Internet Explorer, ενώ είναι συμβατά με τα πρότυπα των browsers. Το bug μπορεί να αποφευχθεί στον Internet Explorer 6 χρησιμοποιώντας το σωστό DOCTYPE σε (X) HTML έγγραφα. Τα CSS hacks και τα CSS φίλτρα χρησιμοποιούνται για την αντιστάθμιση των σφαλμάτων, όπως αυτό, που είναι μόνο ένα από τις εκατοντάδες των CSS σφαλμάτων που έχουν καταγραφεί σε διάφορες εκδόσεις του Netscape, Mozilla Firefox, Opera και Internet Explorer (συμπεριλαμβανομένου του Internet Explorer 7).

Ακόμα και όταν η διαθεσιμότητα των ικανών για το CSS browsers την έκανε μια βιώσιμη τεχνολογία, ο αγώνας των σχεδιαστών για την υιοθέτηση του CSS εξακολουθούσε

να εμποδίζεται από την εσφαλμένη εφαρμογή της CSS και αποσπασματική υποστήριξή της. Ακόμα και σήμερα, τα προβλήματα αυτά συνεχίζουν να κάνουν την επιχείρηση του σχεδιασμού CSS πιο πολύπλοκη και δαπανηρή από ό, τι επρόκειτο να είναι, και η cross-browser δοκιμή παραμένει μια αναγκαιότητα. Άλλοι λόγοι για τη συνέχιση της μη έκδοσης της CSS είναι: αντιληπτή η πολυπλοκότητα της, η έλλειψη των συγγραφέων εξοικειωμένων με τη σύνταξη CSS και τις απαιτούμενες τεχνικές, η κακή υποστήριξη από εργαλεία συγγραφής, των κινδύνων που παρουσιάζουν ασυνέπεια μεταξύ browsers και το αυξημένο κόστος των δοκιμών.

Αυτή τη στιγμή υπάρχει έντονος ανταγωνισμός μεταξύ Gecko μηχανής διάταξης του Mozilla που χρησιμοποιείται στον Firefox, τον κινητήρα διάταξης WebKit που χρησιμοποιείται στην Apple Safari και Google Chrome, την παρόμοια μηχανή KHTML που χρησιμοποιείται στο πρόγραμμα περιήγησης Konqueror του KDE, και του Presto, διάταξη της Opera - το καθένα από αυτά οδηγεί σε διαφορετικές πτυχές του CSS. Από τον Αύγουστο του 2009, ο Internet Explorer 8, Firefox 2 και 3 έχουν εύλογα πλήρη επίπεδα εφαρμογής του CSS 2.1.

Περιορισμοί

Μερικοί περιορισμοί που αξίζει να σημειωθούν για τις υπάρχουσες δυνατότητες του CSS περιλαμβάνουν:

- Κακός έλεγχος για ευέλικτες διατάξεις. Ενώ οι νέες προσθήκες στο CSS 3 παρέχουν μια πιο ισχυρή, με πιο ισχυρά χαρακτηριστικά, διάταξη, η CSS είναι ακόμα στην καρδιά μια γλώσσα styling (για τις γραμματοσειρές, τα χρώματα, τα σύνορα και άλλα διακοσμητικά) και όχι μια γλώσσα διάταξης (για συγκροτήματα με τις θέσεις, τα μεγέθη, περιθώρια, και ούτω καθεξής). Αυτοί οι περιορισμοί σημαίνουν ότι η δημιουργία μακέτας απαιτεί γενικά κωδικοποίηση των CSS χειροκίνητα, και έχει καθυστερήσει την ανάπτυξη προτύπων με βάση το πρόγραμμα επεξεργασίας WYSIWYG.
- Οι επιλογείς δεν είναι σε θέση να ανέβουν. Η CSS δεν προσφέρει κανένα τρόπο να επιλέξετε ένα γονέα ή πρόγονο ενός στοιχείου που πληρεί ορισμένα κριτήρια. Ένα πιο προηγμένο σύστημα επιλογής (όπως το XPath) επιτρέπουν πιο εξελιγμένα φύλλα στυλ. Ωστόσο, οι βασικοί λόγοι για την απόρριψη των προτάσεων για επιλογείς γονέα από την Ομάδα Εργασίας της CSS έχουν σχέση με την απόδοση του προγράμματος περιήγησης και τα στοιχειώδη προβλήματα στην απόδοση.
- Κάθετος περιορισμός ελέγχου. Ενώ η οριζόντια τοποθέτηση των στοιχείων είναι γενικά εύκολο να ελεγχθεί, η κάθετη τοποθέτηση είναι συχνά μη-διαισθητική, πολύπλοκη, ή αδύνατη. Απλές εργασίες, όπως το κεντράρισμα ενός στοιχείου κάθετα ή το να είναι ένα υποσέλιδο που θα τοποθετηθεί ψηλότερα από ό, τι το κάτω μέρος της viewport, είτε απαιτούν περίπλοκους και μη-διαισθητικούς κανόνες στυλ, ή απλά δεν υποστηρίζει ευρέως κανόνες.
- Απουσία εκφράσεων. Σήμερα δεν υπάρχει δυνατότητα να καθορίσετε τιμές ιδιοτήτων ως απλές εκφράσεις (όπως το περιθώριο κέρδους-αριστερά: 10% - 3em + 4PX?). Αυτό θα ήταν χρήσιμο σε μια ποικιλία περιπτώσεων, όπως τον υπολογισμό του μεγέθους των στηλών που υπόκεινται σε περιορισμούς σχετικά με το άθροισμα όλων των στηλών.
- Έλλειψη δήλωσης στήλης. Ενώ είναι δυνατόν στη σημερινή CSS 3 (χρησιμοποιώντας το μετρητή στηλών), σχεδιαγράμματα με πολλαπλές στήλες μπορεί να είναι περίπλοκο να εφαρμοστούν στη CSS 2.1. Με τη CSS 2.1, η διαδικασία γίνεται συχνά με τη χρήση πλωτών στοιχείων, οι οποίες συχνά αποδίδονται με διαφορετικό τρόπο από διαφορετικούς browsers, διαφορετικά σχήματα στην οθόνη του υπολογιστή, και διαφορετικές αναλογίες οθόνης.

Κατασκευή ηλεκτρονικού καταστήματος

- Δεν μπορεί να δηλώσει ρητά νέο πεδίο εφαρμογής ανεξάρτητα από τη θέση. Η οριοθέτηση του πεδίου εφαρμογής των κανόνων για τις ιδιότητες, όπως η z-index αναζητάει το πλησιέστερο γονικό στοιχείο με μια θέση: απόλυτη ή θέση: σχετική η ιδιότητα. Αυτή η περιέργη σύζευξη έχει ανεπιθύμητα αποτελέσματα, καθώς είναι αδύνατο να αποφύγει κανείς την κήρυξη ενός νέου πεδίου, όταν είναι αναγκασμένος να προσαρμόζει τη θέση ενός στοιχείου, εμποδίζοντας τον να χρησιμοποιήσει το επιθυμητό πεδίο εφαρμογής ενός γονικού στοιχείου.
- Η δυναμική συμπεριφορά της ψευδο-κλάσης δεν ελέγχεται. Η CSS υλοποιεί ψευδο-κλάσεις που επιτρέπουν σε έναν βαθμό τα σχόλια των χρηστών μέσω της υπό όρους εφαρμογής των εναλλακτικών μορφών. Μία CSS ψευδο-κλάση, ":hover", είναι δυναμική (ισοδύναμη της javascript "onmouseover") και έχει δυνατότητες για δυναμική συμπεριφορά (π.χ., την εφαρμογή δείκτη άμεσης εγγύτητας), αλλά η CSS δεν παρέχει καμία δυνατότητα για έναν πελάτη να την απενεργοποιήσει (δεν υπάρχει "ακύρωση" σαν ιδιότητα) ή τον περιορισμό των συνεπειών του (δεν υπάρχει "nochange» ως τιμές για κάθε ιδιότητα).

Πλεονεκτήματα

- Διαχωρισμός του περιεχομένου από την παρουσίαση. Η CSS διευκολύνει τη δημοσίευση του περιεχομένου σε πολλές μορφές παρουσίασης με βάση τις ονομαστικές παραμέτρους. Ονομαστικοί παράμετροι περιλαμβάνουν σαφείς προτιμήσεις των χρηστών, διαφορετικά προγράμματα περιήγησης στο διαδίκτυο, το είδος της συσκευής που χρησιμοποιείται για να δείτε το περιεχόμενο (επιτραπέζιο υπολογιστή ή κινητή συσκευή διαδικτύου), η γεωγραφική θέση του χρήστη και πολλές άλλες μεταβλητές.
- Όταν η CSS χρησιμοποιείται αποτελεσματικά, από την άποψη της κληρονομικότητας, ένα καθολικό φύλλο στυλ μπορεί να χρησιμοποιηθεί για να επηρεάσει και το στυλ των στοιχείων σε όλη την ιστοσελίδα. Εάν προκύπτει η κατάσταση όπου η σχεδίαση των στοιχείων θα πρέπει να αλλάξει ή να προσαρμοστεί, αυτές οι αλλαγές μπορούν να γίνουν με την επεξεργασία των κανόνων στο καθολικό φύλλο στυλ. Πριν τη CSS, αυτό το είδος της συντήρησης ήταν πιο δύσκολο, δαπανηρό και χρονοβόρο.
- Ένα φύλλο στυλ, είτε είναι εσωτερικής φύσεως προς το έγγραφο προέλευσης ή ξεχωριστό, θα προσδιορίζει το στυλ μία φορά και για μια σειρά από στοιχεία HTML επιλέγει ανά κατηγορία, τον τύπο ή τη σχέση με τους άλλους. Αυτό είναι πολύ πιο αποδοτικό από ό, τι επαναλαμβάνοντας πληροφορίες στυλ γραμμή γραμμή για κάθε εμφάνιση του στοιχείου. Ένα εξωτερικό φύλλο στυλ είναι συνήθως αποθηκευμένο στο cache του browser, και μπορούν συνεπώς να χρησιμοποιηθούν σε πολλές σελίδες, χωρίς να επαναγεμίζονται, μειώνοντας περαιτέρω τη μεταφορά δεδομένων μέσω δικτύου.
- Με μια απλή αλλαγή μια γραμμή, ένα διαφορετικό φύλλο στυλ μπορεί να χρησιμοποιηθεί για την ίδια σελίδα. Αυτό έχει πλεονεκτήματα για την προσβασιμότητα, καθώς και την παροχή της δυνατότητας προσαρμογής μιας σελίδας για διαφορετικές συσκευές. Επιπλέον, οι συσκευές που δεν είναι σε θέση να κατανοήσουν το στυλ εμφάνισης, εμφανίζουν το περιεχόμενο.
- Χωρίς CSS, οι web designers πρέπει συνήθως να επεξεργάζονται τις σελίδες τους με τις τεχνικές που εμποδίζουν την πρόσβαση για χρήστες με προβλήματα όρασης, όπως οι πίνακες HTML.

Τα CSS πλαίσια είναι έτοιμες βιβλιοθήκες οι οποίες έχουν σκοπό να επιτρέψουν την ευκολότερη, πιο συμβατή με τα πρότυπα στυλ των σελίδων του Διαδικτύου που χρησιμοποιούν την γλώσσα CSS. Όπως όλες οι βιβλιοθήκες προγραμματισμού και κωδικοποίησης, το CSS πλαίσιο συνήθως ενσωματώνεται σε εξωτερικά .css φύλλα και αναφέρεται στο τμήμα <head> στην HTML. Παρέχουν μια σειρά έτοιμων επιλογών για τον

σχεδιασμό και για τον καθορισμό της ιστοσελίδας. Αν και πολλά από αυτά τα πλαίσια έχουν δημοσιευθεί, μερικοί συγγραφείς τα χρησιμοποιούν ως επί το πλείστον για την ταχεία εκπόνηση πρωτοτύπων, ή για την εκμάθηση από αυτά, και προτιμούν να συντάξουν κώδικα CSS από την αρχή, χωρίς το σχεδιασμό, τη συντήρηση και την λήψη των γενικών αποτελεσμάτων που έχουν πολλά χαρακτηριστικά που δεν έχουν χρησιμοποιηθεί στο στυλ πολλών ιστοτόπων.

2.5 Visual Basic

Η Visual Basic (VB) είναι γλώσσα προγραμματισμού τρίτης γενιάς, οδηγούμενη από συμβάντα (event driven) και έχει ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft για το μοντέλο προγραμματισμού COM. Η VB θεωρείται επίσης μία σχετικά εύκολη γλώσσα προγραμματισμού στην εκμάθηση και τη χρήση, λόγω των χαρακτηριστικών της, καθώς έχει Γραφικό Περιβάλλον Χρήστη και συγγένεια με την γλώσσα προγραμματισμού BASIC.

Η Visual Basic προέρχεται από τη BASIC και επιτρέπει την ταχεία ανάπτυξη εφαρμογών (RAD) με Γραφικό Περιβάλλον Χρήστη (GUI), πρόσβαση σε βάσεις δεδομένων χρησιμοποιώντας αντικείμενα (Data Access Objects, Remote Data Objects, ή ActiveX Data Objects), και τη δημιουργία στοιχείων ελέγχου ActiveX και αντικειμένων. Οι γλώσσες προγραμματισμού τύπου "scripting", όπως η VBA και VBScript συντακτικά είναι παρόμοιες με τη Visual Basic, αλλά έχουν διαφορετικές επιδόσεις.

Ένας προγραμματιστής μπορεί να ολοκληρώσει μια εφαρμογή χρησιμοποιώντας τα στοιχεία που παρέχονται με την Visual Basic. Προγράμματα γραμμένα σε Visual Basic μπορούν, επίσης, να χρησιμοποιήσουν το Windows API, αλλά κάτι τέτοιο απαιτεί δηλώσεις εξωτερικών συναρτήσεων. Η τελική έκδοση 6 βγήκε το 1998. Η εκτεταμένη υποστήριξη της Microsoft έληξε το Μάρτιο του 2008 και ορίστηκε διάδοχος της η Visual Basic.NET (γνωστή απλά ως Visual Basic).

2.4.1 Χαρακτηριστικά της Γλώσσας

Όπως και η γλώσσα προγραμματισμού BASIC, η Visual Basic έχει σχεδιαστεί για να είναι εύκολη στην εκμάθηση και το χειρισμό. Η γλώσσα δεν επιτρέπει στους προγραμματιστές να δημιουργήσουν μόνο απλές εφαρμογές GUI, αλλά μπορούν, επίσης, να αναπτύξουν πολύπλοκες εφαρμογές. Ο προγραμματισμός σε VB συνίσταται από τον οπτικό συνδυασμό στοιχείων ή ελέγχων σε μια φόρμα, τον προσδιορισμό χαρακτηριστικών και ενεργειών αυτών των στοιχείων και την σύνταξη επιπλέον γραμμών κώδικα για αυξημένη λειτουργικότητα. Καθώς υπάρχουν προεπιλεγμένα χαρακτηριστικά και ενέργειες για τα επιμέρους στοιχεία, μπορεί να δημιουργηθεί ένα απλό πρόγραμμα χωρίς ο προγραμματιστής να γράψει πολλές γραμμές κώδικα. Στις προηγούμενες εκδόσεις υπήρχαν προβλήματα επιδόσεων, αλλά με τους ταχύτερους υπολογιστές και τη μεταγλώττιση εγγενούς κώδικα αυτό παύει να είναι ένα τόσο σημαντικό ζήτημα.

Αν και τα προγράμματα μπορούν να μετατραπούν σε εγγενή εκτελέσιμο κώδικα από την έκδοση 5 και μετά, αυτά εξακολουθούν να απαιτούν την παρουσία των βιβλιοθηκών χρόνου εκτέλεσης (runtime) με μέγεθος περίπου 1 MB. Οι βιβλιοθήκες runtime υπάρχουν στα Windows 2000 και αργότερα, αλλά στις παλαιότερες εκδόσεις των Windows όπως τα 95/98/NT πρέπει να διανέμονται μαζί με το εκτελέσιμο αρχείο.

Οι φόρμες δημιουργούνται χρησιμοποιώντας τεχνικές "σύρε κι άσε" (drag-and-drop). Χρησιμοποιείται ένα εργαλείο για την τοποθέτηση στοιχείων ελέγχου (π.χ. πλαίσια κειμένου, κουμπιά, κλπ.) στη φόρμα (παράθυρο). Τα στοιχεία ελέγχου έχουν χαρακτηριστικά και χειριστές συμβάντων συνδεδεμένους με αυτά. Οι προεπιλεγμένες τιμές παρέχονται όταν δημιουργείται το στοιχείο ελέγχου, αλλά μπορούν να τροποποιηθούν από τον προγραμματιστή. Πολλές τιμές χαρακτηριστικών είναι δυνατό να τροποποιηθούν κατά το

Κατασκευή ηλεκτρονικού καταστήματος

χρόνο εκτέλεσης από ενέργειες του χρήστη ή αλλαγές του περιβάλλοντος, παρέχοντας έτσι μια δυναμική εφαρμογή. Για παράδειγμα, μπορεί να εισαχθεί κώδικας στον χειριστή συμβάντων αλλαγής διαστάσεων της φόρμας, ώστε ένα στοιχείο ελέγχου να παραμένει πάντα στο κέντρο της φόρμας ή να μεγαλώσει ώστε να την γεμίσει, κλπ. Με την προσθήκη κώδικα μέσα σε ένα χειριστή συμβάντων για το πάτημα των πλήκτρων σε ένα πλαίσιο κειμένου, το πρόγραμμα μπορεί αυτόματα να μετατρέψει το εισαγόμενο κείμενο σε κεφαλαία ή πεζά ή ακόμα και να εμποδίσει ορισμένους από τους χαρακτήρες να εμφανιστούν.

Με τη Visual Basic είναι δυνατή η δημιουργία εκτελέσιμων (EXE) αρχείων, στοιχείων ελέγχου ActiveX ή αρχείων DLL, αλλά χρησιμοποιείται κυρίως για την ανάπτυξη εφαρμογών για τα Windows και τη διασύνδεση συστημάτων βάσεων δεδομένων. Πλαίσια διαλόγου με λιγότερες λειτουργίες μπορούν να χρησιμοποιηθούν για pop-up δυνατότητες. Τα στοιχεία ελέγχου παρέχουν τις βασικές λειτουργίες της εφαρμογής, ενώ οι προγραμματιστές μπορούν να εισαγάγουν επιπλέον λογική μέσα στο κατάλληλο χειριστή γεγονότων. Για παράδειγμα, ένα πτυσσόμενο πλαίσιο θα εμφανίζει αυτόματα μια λίστα που θα επιτρέπει στο χρήστη να επιλέξει οποιοδήποτε στοιχείο. Ένας χειριστής γεγονότων καλείται όταν ένα αντικείμενο είναι επιλεγμένο, και στη συνέχεια μπορεί να εκτελεστεί πρόσθετος κώδικας που δημιουργείται από τον προγραμματιστή για να εκτελεστεί κάποια ενέργεια που βασίζεται στο στοιχείο που έχει επιλεγεί.

Εναλλακτικά, ένα συστατικό της Visual Basic μπορεί να μην έχει Γραφικό Περιβάλλον Χρήστη, αλλά, αντ' αυτού, να παρέχει αντικείμενα ActiveX σε άλλα προγράμματα μέσω Component Object Model (COM). Αυτό επιτρέπει επεξεργασία στην πλευρά του διακομιστή (server-side processing) ή τη δημιουργία πρόσθετων μορφωμάτων (add-in module).

Η γλώσσα έχει αυτόματη διαχείριση μνήμης με την τεχνική της συλλογής σκουπιδιών (garbage collection) χρησιμοποιώντας υπολογισμό αναφορών και έχει μια μεγάλη βιβλιοθήκη με βοηθητικά αντικείμενα καθώς και βασική αντικειμενοστραφή υποστήριξη. Από τα πιο κοινά στοιχεία που περιλαμβάνονται στο προεπιλεγμένο πρότυπο έργου, ο προγραμματιστής σπάνια χρειάζεται να καθορίσει πρόσθετες βιβλιοθήκες. Αντίθετα με πολλές άλλες γλώσσες προγραμματισμού η Visual Basic γενικά δεν διαχωρίζει τους πεζούς από τους κεφαλαίους χαρακτήρες, αν και θα μετατρέψει τις λέξεις-κλειδιά σε μία τυπική διαμόρφωση. Οι συγκρίσεις συμβολοσειρών διαχωρίζουν τα πεζά από τα κεφαλαία από προεπιλογή, αλλά μπορεί να αλλάξει αυτό, εφόσον το επιθυμείτε.

Ο μεταγλωττιστής της Visual Basic είναι κοινός με τις άλλες γλώσσες του Visual Studio (C#, C++), αλλά οι περιορισμοί στον IDE δεν επιτρέπουν τη δημιουργία ορισμένων στόχων (μοντέλα Windows DLL) και σε μοντέλα νημάτων.

Χαρακτηριστικά της Visual Basic

Visual Basic έχει τα εξής χαρακτηριστικά που διαφέρουν από τις γλώσσες που προέρχονται από τη C:

Η πολλαπλή ανάθεση που διατίθεται στη C δεν είναι δυνατή. Το $A = B = C$, δεν σημαίνει ότι οι τιμές των A, B και C είναι ίδιες. Το αποτέλεσμα της boolean "B = C;" αποθηκεύεται στην μεταβλητή A. Το αποτέλεσμα που αποθηκεύεται στην A θα μπορούσε συνεπώς να είναι ψευδές (0) ή αληθές (-1).

Οι λογικοί και δυαδικοί τελεστές είναι ενοποιημένοι. Αυτό έρχεται σε αντίθεση με ορισμένες γλώσσες προερχόμενες από τη C (όπως η γλώσσα Perl), οι οποίες έχουν ξεχωριστούς λογικούς και δυαδικούς τελεστές. Αυτό είναι ένα παραδοσιακό χαρακτηριστικό της γλώσσας BASIC.

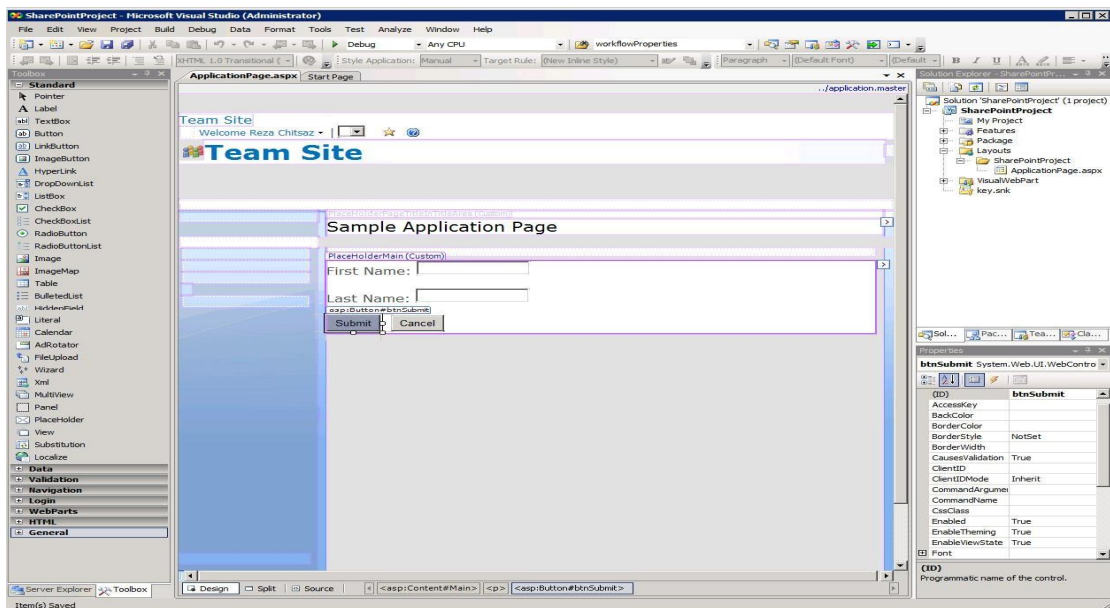
2.6 Visual studio

Το Microsoft Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) της Microsoft. Χρησιμοποιείται για την ανάπτυξη εφαρμογών κονσόλας αλλά και γραφικές εφαρμογές με διεπαφή χρήστη, καθώς και Windows Forms εφαρμογές, ιστοσελίδες, διαδικτυακές εφαρμογές και υπηρεσίες για όλες τις πλατφόρμες που υποστηρίζονται από τα Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework και το Microsoft Silverlight.

Το Visual Studio περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα που υποστηρίζει IntelliSense καθώς και Refactoring του κώδικα. Το ολοκληρωμένο πρόγραμμα εντοπισμού σφαλμάτων λειτουργεί τόσο ως ένα πρόγραμμα εντοπισμού σφαλμάτων σε επίπεδο πηγής και σε επίπεδο μηχανής. Αλλά ενσωματωμένα εργαλεία που περιλαμβάνει είναι ο σχεδιαστής εντύπων για τη δημιουργία GUI εφαρμογών, web designer, σχεδιαστής κλάσης και το σχεδιαστή διαγράμματος βάσης δεδομένων. Δέχεται plug-ins που βελτιώνουν τη λειτουργικότητα σε σχεδόν κάθε επίπεδο-όπως η προσθήκη υποστήριξης για την πηγή των συστημάτων ελέγχου (όπως το Subversion και Visual SourceSafe) και την προσθήκη νέων toolsets όπως εκδότες και οπτικοί σχεδιαστές για συγκεκριμένες γλώσσες ή toolsets για άλλες πτυχές του κύκλου ανάπτυξης λογισμικού.

Το Visual Studio υποστηρίζει διαφορετικές γλώσσες προγραμματισμού, μέσω των γλωσσικών υπηρεσιών, οι οποίες επιτρέπουν στον επεξεργαστή κώδικα και στο πρόγραμμα εντοπισμού σφαλμάτων την υποστήριξη (σε διαφορετικό βαθμό) σχεδόν όλων των γλωσσών προγραμματισμού, υπό την προϋπόθεση να υπάρχει η υπηρεσία domain-specific languages. Οι ενσωματωμένες γλώσσες που περιλαμβάνει είναι οι C/C++ (μέσω Visual C++), VB.NET (μέσω της Visual Basic .NET), C# (μέσω Visual C#), και F# (όπως του Visual Studio 2010). Υποστήριξη για άλλες γλώσσες, όπως η M, Python, Ruby και άλλων είναι διαθέσιμη μέσω της υπηρεσίας γλωσσών, που πρέπει να εγκατασταθεί ξεχωριστά. Υποστηρίζει, επίσης, XML / XSLT, HTML / XHTML, CSS και JavaScript.

Η Microsoft παρέχει "Express" εκδόσεις του Visual Studio 2010 που περιλαμβάνουν Visual Basic, Visual C #, Visual C + +, και Visual Web Developer. Το Visual Studio 2010, 2008 και 2005 Professional Εκδόσεις, μαζί με τις εκδόσεις για συγκεκριμένες γλώσσες (Visual Basic, C++, C#, J#) του Visual Studio Express 2010 είναι διαθέσιμο δωρεάν για τους φοιτητές.

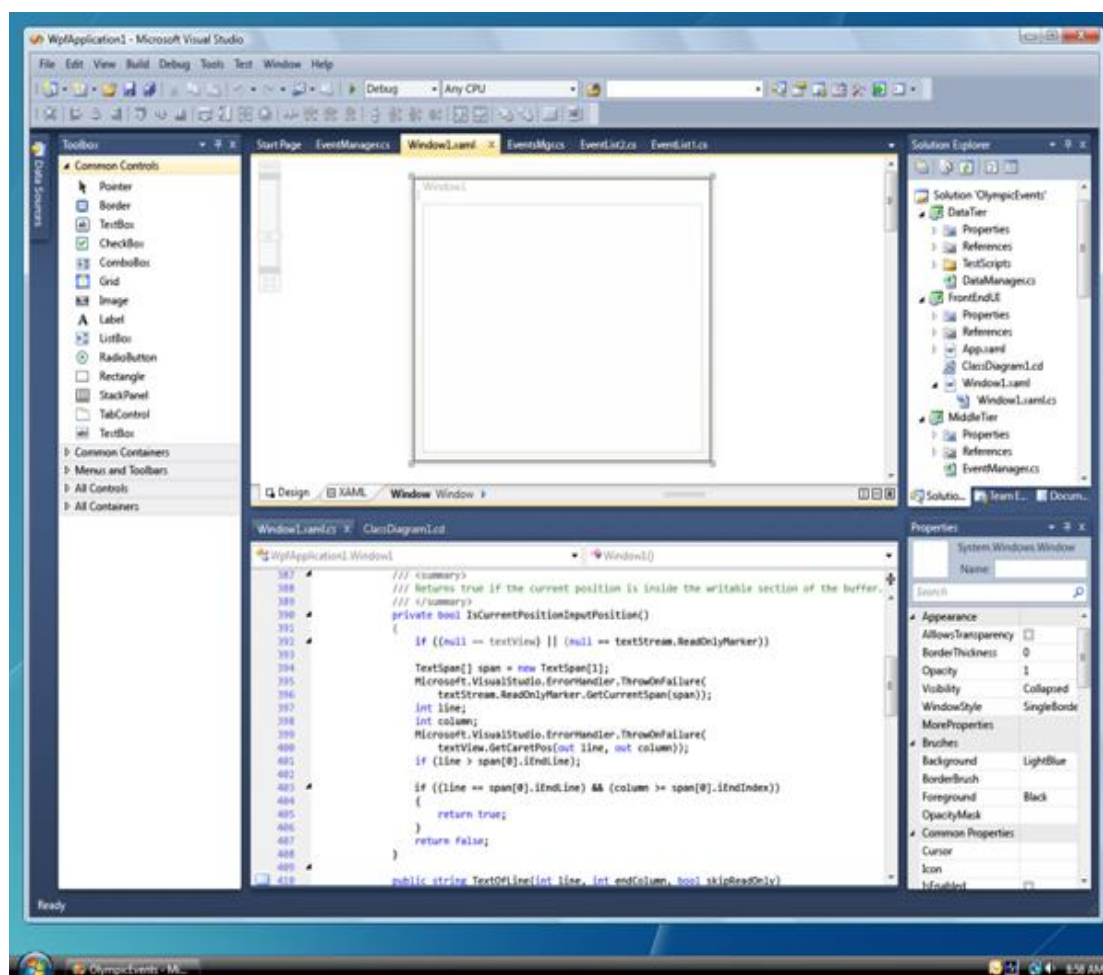


2.0.1 - Το περιβάλλον ανάπτυξης του Visual Studio

Κατασκευή ηλεκτρονικού καταστήματος

Το Visual Studio δεν υποστηρίζει όλες τις γλώσσες προγραμματισμού εγγενώς, αλλά επιτρέπει την ενσωμάτωση της λειτουργικότητας, που κωδικοποιείται ως VSPackage. Όταν εγκατασταθεί, η λειτουργικότητα είναι διαθέσιμη ως υπηρεσία. Το IDE παρέχει τρεις υπηρεσίες: η SVsSolution, η οποία παρέχει τη δυνατότητα να απαριθμεί τα έργα και τις λύσεις, η SVsUIshell, η οποία παρέχει λειτουργικότητα των παραθύρων και της UI (συμπεριλαμβανομένων των καρτελών, των γραμμών εργαλείων και εργαλείων παραθύρων), και η SVsShell, η οποία ασχολείται με την καταγραφή των VSPackages. Επιπλέον, το IDE είναι υπεύθυνο για το συντονισμό και τη δυνατότητα επικοινωνίας μεταξύ των υπηρεσιών. Όλοι οι συντάκτες, οι σχεδιαστές, οι τύποι έργων και άλλα εργαλεία εφαρμόζονται ως VSPackages. Το Visual Studio χρησιμοποιεί COM (Component Object Model) για την πρόσβαση στα VSPackages.

Το Visual Studio SDK περιλαμβάνει επίσης το πλαίσιο για τη διαχείριση πακέτου MPF, το οποίο είναι ένα σύνολο των υπό διαχείριση wrappers γύρω από τις COM-διασυνδέσεις που επιτρέπουν στα πακέτα να γραφτούν σε οποιαδήποτε γλώσσα συμβατή με το CLI. Ωστόσο, το MPF δεν παρέχει όλες τις λειτουργικότητα που περιλαμβάνονται από τις διασυνδέσεις COM του Visual Studio. Οι υπηρεσίες μπορούν στη συνέχεια να χρησιμοποιηθούν για τη δημιουργία άλλων πακέτων, που προσθέτουν λειτουργικότητα στο Visual Studio IDE.



2.0.2 - Designer και Source View του Visual Studio

Υποστήριξη για γλώσσες προγραμματισμού, προστίθεται με τη χρήση συγκεκριμένης VSPackage που ονομάζεται μεταφραστική υπηρεσία. Μια υπηρεσία γλωσσών που ορίζει

διάφορες διασυνδέσεις τις οποίες η εφαρμογή VSPackage μπορεί να εφαρμόσει για να προσθέσετε υποστήριξη για διάφορες λειτουργίες. Λειτουργίες που μπορούν να προστεθούν με αυτόν τον τρόπο περιλαμβάνουν χρωματισμό σύνταξης, ολοκλήρωση δηλώσεων, αντιστοίχιση παρενθέσεων, βοηθητικές πληροφορίες παραμέτρων, λίστες μελών και δείκτες σφάλματος. Αν το interface είναι σε εφαρμογή, η λειτουργικότητα θα είναι διαθέσιμη για τη γλώσσα. Γλωσσικές υπηρεσίες πρόκειται να εφαρμοστούν για κάθε γλώσσα.

Οι εφαρμογές μπορούν να επαναχρησιμοποιήσουν κώδικα από το πρόγραμμα ανάλυσης ή ο μεταγλωττιστής για τη γλώσσα. Γλωσσικές υπηρεσίες μπορούν να εφαρμοστούν ή σε εγγενή κώδικα ή σε διαχειριζόμενο κώδικα. Για τον εγγενή κώδικα, μπορούν να χρησιμοποιηθούν είτε οι ενσωματωμένες διασυνδέσεις COM ή το πλαίσιο Βαβέλ (μέρος του Visual Studio SDK). Για το διαχειριζόμενο κώδικα, το MPF περιλαμβάνει περιτυλίγματα για το γράψιμο και τη διαχείριση των γλωσσικών υπηρεσιών.

Το Visual Studio δεν περιλαμβάνει ενσωματωμένη υποστήριξη ελέγχου πηγής, αλλά ορίζει δύο εναλλακτικούς τρόπους για συστήματα ελέγχου πηγής και την ενσωμάτωση με το IDE. Τα VSPackage ελέγχου προέλευσης μπορούν να παρέχουν το δικό σας προσαρμοσμένο περιβάλλον εργασίας του χρήστη. Αντίθετα, ένα plugin ελέγχου πηγής παρέχει ένα σύνολο λειτουργιών που χρησιμοποιούνται για την εφαρμογή των διαφόρων λειτουργιών ελέγχου πηγής χρησιμοποιώντας το MSSCCI (έλεγχος διεπαφής πηγαίου κώδικα της Microsoft), σε ένα τυπικό Visual Studio περιβάλλον χρήστη. Το MSSCCI χρησιμοποιήθηκε για πρώτη φορά για να ενσωματώσει το Visual SourceSafe με το Visual Studio 6.0, αλλά αργότερα αναπτύχθηκε μέσω του Visual Studio SDK.

Το Visual Studio, όπως και κάθε άλλο IDE, περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα που υποστηρίζει την συντακτική επισήμανση και την ολοκλήρωση κώδικα χρησιμοποιώντας IntelliSense όχι μόνο για τις μεταβλητές, λειτουργίες και μεθόδους, αλλά και γλωσσικές δομές όπως οι βρόχοι και τα ερωτήματα.

Το πρόγραμμα επεξεργασίας κώδικα του Visual Studio υποστηρίζει επίσης τον καθορισμό των σελιδοδεικτών σε κώδικα για γρήγορη πλοήγηση. Άλλα βοηθήματα πλοήγησης περιλαμβάνουν κατάρρευση μπλοκ κώδικα και στοιχειώδη έρευνα, πέραν της συνήθους αναζήτησης κειμένου και regex αναζήτησης. Το πρόγραμμα επεξεργασίας κώδικα περιλαμβάνει επίσης ένα πρόχειρο και μια λίστα εργασιών.

Καθώς γράφεται ο κώδικας, το Visual Studio συγκεντρώνει στο παρασκήνιο, προκειμένου να παράσχει ανάδραση σχετικά με τη σύνταξη και τα λάθη, τα οποία επισημαίνονται με μια κόκκινη κυματιστή υπογράμμιση. Οι προειδοποιήσεις επισημαίνονται με μια πράσινη υπογράμμιση. Η συλλογή Ιστορικό δεν παράγει εκτελέσιμο κώδικα, δεδομένου ότι απαιτεί ένα διαφορετικό compiler από αυτό που χρησιμοποιείται για να παράγει εκτελέσιμο κώδικα. Η συλλογή αυτή, που αρχικά εισήχθη με τη Microsoft Visual Basic, τώρα έχει επεκταθεί για όλες τις γλώσσες που περιλαμβάνονται.

Το Visual Studio περιλαμβάνει ένα πρόγραμμα εντοπισμού σφαλμάτων που λειτουργεί τόσο ως ένα πρόγραμμα εντοπισμού σφαλμάτων σε επίπεδο πηγής και ως μια μηχανή επιπέδου εντοπισμού σφαλμάτων. Λειτουργεί τόσο με διαχειριζόμενο κώδικα, καθώς και με τοπικό κώδικα και μπορεί να χρησιμοποιηθεί για τον εντοπισμό σφαλμάτων εφαρμογών γραμμένων σε οποιαδήποτε γλώσσα που υποστηρίζεται από το Visual Studio. Επιπλέον, μπορεί να συνοδεύσει την εκτέλεση των διαδικασιών και την παρακολούθηση και τον εντοπισμό σφαλμάτων αυτών των διαδικασιών. Αν ο πηγαίος κώδικας για τη λειτουργία της διαδικασίας είναι διαθέσιμος, εμφανίζει τον κώδικα όπως τρέχει. Αν ο πηγαίος κώδικας δεν είναι διαθέσιμος, μπορεί να δείξει την αποσυναρμολόγηση. Το πρόγραμμα εντοπισμού σφαλμάτων Visual Studio μπορεί, επίσης, να δημιουργήσει χώρους μνήμης, καθώς και να τους φορτώσει αργότερα για τον εντοπισμό σφαλμάτων. Υποστηρίζονται επίσης Multi-threaded προγράμματα. Το πρόγραμμα εντοπισμού σφαλμάτων μπορεί να ρυθμιστεί ώστε να ξεκινήσει όταν μια εφαρμογή τρέχει έξω από το Visual Studio περιβάλλον.

Το πρόγραμμα εντοπισμού σφαλμάτων επιτρέπει ρύθμιση οριακών σημείων (η οποία, για να επιτρέψει την εκτέλεση, πρέπει να διακοπεί προσωρινά σε μια συγκεκριμένη θέση) και τα ρολόγια (που παρακολουθούν τις τιμές των μεταβλητών, όπως η εκτέλεση προχωρεί). Σημεία διακοπής μπορεί να είναι υπό όρους, που σημαίνει ότι ενεργοποιούνται όταν πληρούνται οι προϋποθέσεις.

Κατασκευή ηλεκτρονικού καταστήματος

Windows Forms Designer

Ο σχεδιαστής των Windows Forms χρησιμοποιείται για τη δημιουργία GUI εφαρμογών με τη χρήση των Windows Forms. Ελέγχει ότι τα δεδομένα της οθόνης (όπως πλαίσιο κειμένου, πλαίσιο λίστας, πλέγμα, κλπ.) μπορούν να συνδεθούν με πηγές δεδομένων, όπως βάσεις δεδομένων ή ερωτήματα. Έλεγχοι δεδομένων μπορούν να δημιουργηθούν με μεταφορά στοιχείων από τις διάφορες πηγές δεδομένων επάνω σε μια επιφάνεια σχεδιασμού. Η διεπαφή χρήστη συνδέεται με τον κώδικα χρησιμοποιώντας ένα event-driven μοντέλο προγραμματισμού. Ο σχεδιαστής δημιουργεί είτε C# ή VB.NET κώδικα για την εφαρμογή.

WPF Designer

Ο σχεδιαστής WPF, με κωδικό όνομα Cider, θεσπίστηκε με το Visual Studio 2008. Όπως και ο σχεδιαστής Windows Forms υποστηρίζει την drag and drop μεταφορά. Υποστηρίζει όλες τις λειτουργίες WPF συμπεριλαμβανομένων και της δέσμωσης δεδομένων και αυτόματη διαχείριση διάταξης. Παράγει XAML κώδικα για το UI. Το παραγόμενο αρχείο XAML είναι συμβατό με το Expression Design της Microsoft.

Web designer

Το Visual Studio περιλαμβάνει επίσης ένα web-site συντάκτη και σχεδιαστή που επιτρέπει σε ιστοσελίδες να συνταχθούν με μεταφορά και απόθεση widgets. Χρησιμοποιείται για την ανάπτυξη εφαρμογών ASP.NET και υποστηρίζει HTML, CSS και JavaScript. Χρησιμοποιεί ένα πίσω από τον κώδικα πρότυπο για τη σύνδεση με τον ASP.NET κώδικα. Από το Visual Studio 2008 και μετά, ο κινητήρας διάταξης χρησιμοποιείται από το web designer από κοινού με το Expression Web της Microsoft. Υπάρχει επίσης ASP.NET MVC υποστήριξη για την τεχνολογία MVC ως ξεχωριστό στοιχείο λήψης και ASP.NET Dynamic Data έργο διαθέσιμο από τη Microsoft.

Σχεδιαστής Κλάσης

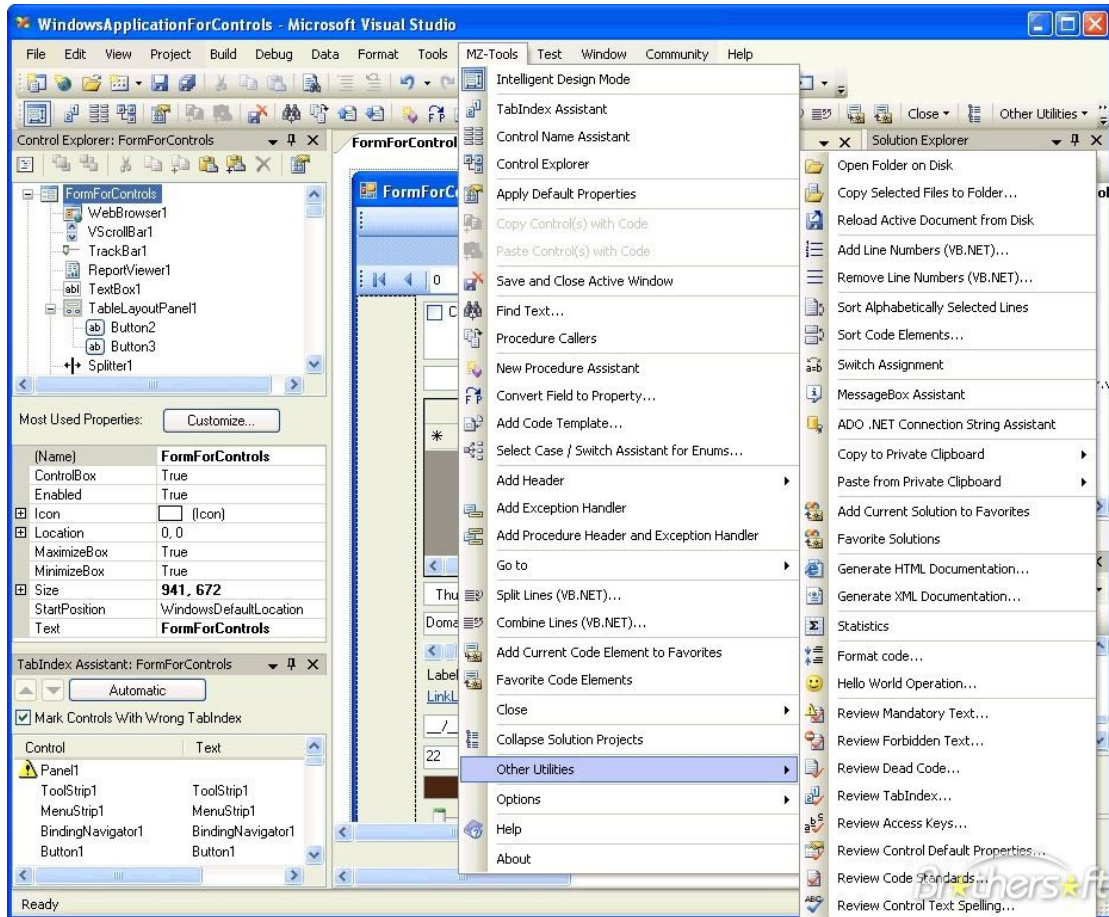
Η Designer Class χρησιμοποιείται στον συντάκτη για επεξεργασία των κλάσεων (συμπεριλαμβανομένων των μελών τους και την πρόσβασή τους) με τη χρήση UML μοντέλων. Η Designer Class μπορεί να δημιουργήσει C# και VB.NET κώδικα περιγράφοντας τις τάξεις και τις μεθόδους. Μπορεί επίσης να δημιουργήσει διαγράμματα κλάσεων από χειρόγραφη κλάση.

Σχεδιαστής δεδομένων

Ο σχεδιαστής των δεδομένων μπορεί να χρησιμοποιηθεί για να επεξεργαστείτε γραφικά σχήματα βάσης δεδομένων, συμπεριλαμβανομένων των δακτυλογραφημένων πινάκων, πρωτεύοντα και ξένα κλειδιά και τους περιορισμούς. Μπορεί επίσης να χρησιμοποιηθεί για το σχεδιασμό ερωτημάτων από το design view.

Ευρετήριο των Αντικειμένων

Μπορεί να χρησιμοποιηθεί για να περιηγηθείτε στο χώρους ονομάτων (τα οποία είναι διατεταγμένα ιεραρχικά). Η ιεραρχία μπορεί να αντικατοπτρίζει την οργάνωση του συστήματος αρχείων.



2.0.3 - Τα μενού του Visual Studio

Επεκτασιμότητα

Το Visual Studio επιτρέπει στους προγραμματιστές να γράψουν επεκτάσεις για το Visual Studio για να επεκτείνει τις δυνατότητές του. Οι επεκτάσεις μπορεί να είναι σε μορφή μακροεντολών ή πακέτων. Οι μακροεντολές αποτελούν επαναλαμβανόμενες εργασίες και δράσεις που οι προγραμματιστές μπορούν να καταγράφουν προγραμματιστικά για την αποθήκευση, αναπαραγωγή, και τη διανομή. Οι μακροεντολές, ωστόσο, δεν μπορούν να εφαρμόσουν νέες εντολές ή να δημιουργήσουν παράθυρα εργαλείων. Πρόκειται για γραπτή χρήση της Visual Basic και δεν γίνεται.

Οι προσθήκες (add-ins) παρέχουν πρόσβαση στο μοντέλο αντικειμένου Visual Studio και μπορούν να αλληλεπιδράσουν με τα IDE εργαλεία. Μπορούν να χρησιμοποιηθούν για την εφαρμογή νέων λειτουργιών και μπορούν να προσθέσουν νέα παράθυρα εργαλείων. Τα πρόσθετα συνδέονται με το IDE μέσω COM και μπορούν να δημιουργηθούν σε κάθε γλώσσα συμβατή με το COM. Τα πακέτα δημιουργούνται με τη χρήση του Visual Studio SDK και παρέχουν το υψηλότερο επίπεδο επεκτασιμότητας.

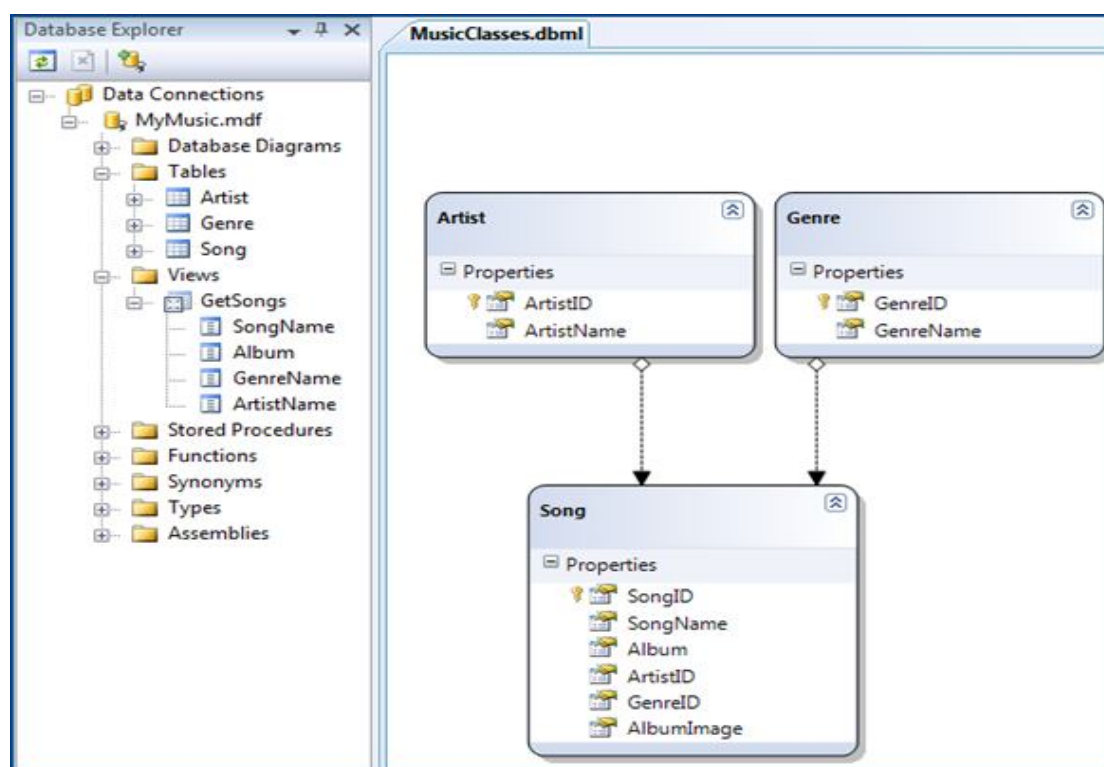
Οι σχεδιαστές μπορούν να δημιουργήσουν και άλλα εργαλεία, καθώς και την ενσωμάτωση με άλλες γλώσσες προγραμματισμού. Το Visual Studio SDK παρέχει μη διαχειριζόμενο API, καθώς και διαχειριζόμενο API για την εκτέλεση των καθηκόντων αυτών. Ωστόσο, το διαχειριζόμενο API δεν είναι τόσο πλήρες όσο το μη διαχειριζόμενο. Οι επεκτάσεις υποστηρίζονται από τις πρότυπες (και μεταγενέστερες) εκδόσεις του Visual Studio 2005. Οι Express εκδόσεις δεν υποστηρίζουν τη φιλοξενία των επεκτάσεων.

Το Visual Studio 2008 εισήγαγε το Visual Studio Shell που επιτρέπει την ανάπτυξη μιας προσαρμοσμένης έκδοσης του IDE. Το Visual Studio Shell ορίζει ένα σύνολο VSPackages που παρέχουν τη λειτουργικότητα που απαιτείται σε κάθε IDE. Σύν τοις άλλους, άλλα πακέτα μπορούν να προστεθούν για να προσαρμόσετε την εγκατάσταση. Η

Κατασκευή ηλεκτρονικού καταστήματος

απομονωμένη λειτουργία του κελύφους δημιουργεί μια νέα AppID όπου είναι εγκατεστημένα τα πακέτα. Αυτή πρέπει να ξεκινήσει με ένα διαφορετικό εκτελέσιμο. Έχει ως στόχο την ανάπτυξη προσαρμοσμένου περιβάλλοντος ανάπτυξης, είτε για μια συγκεκριμένη γλώσσα ή ένα συγκεκριμένο σενάριο.

Η ενοποιημένη λειτουργία εγκαθιστά τα πακέτα στο AppID της Επαγγελματικής / Standard εκδόσεις του συστήματος, έτσι ώστε τα εργαλεία να ενταχθούν σε αυτές τις εκδόσεις. Το Visual Studio Shell είναι διαθέσιμο δωρεάν. Μετά την κυκλοφορία του Visual Studio 2008, η Microsoft δημιούργησε το Visual Studio Gallery. Χρησιμοποιεί ως η κεντρική τοποθεσία για τη δημοσίευση πληροφοριών σχετικά με τις επεκτάσεις για το Visual Studio. Προγραμματιστές της Κοινότητας καθώς και άλλοι μπορούν να φορτώσουν τις πληροφορίες σχετικά με τις επεκτάσεις τους στο Visual Studio. NET 2002 μέσω Visual Studio 2010. Οι χρήστες της ιστοσελίδας μπορούν να αξιολογούν και να σχολιάζουν τις επεκτάσεις για να βοηθήσουν την αξιολόγηση της ποιότητας των επεκτάσεων που δημοσιεύτηκαν.



2.0.4 - Διαχείριση βάσης δεδομένων μέσω Visual Studio

Εκδόσεις

Το Microsoft Visual Studio είναι διαθέσιμο στις ακόλουθες εκδόσεις:

- Visual Studio Express Editions - είναι ένα σύνολο δωρεάν ελαφρύ μεμονωμένων IDEs τα οποία παρέχονται ως απογυμνωμένες εκδόσεις του Visual Studio IDE με βάση την πλατφόρμα ή τη γλώσσα, δηλαδή, εγκαθιστά τα εργαλεία ανάπτυξης για τις υποστηριζόμενες πλατφόρμες (web, Windows, τηλέφωνο) ή υποστηριζόμενες γλώσσες ανάπτυξης (VB, C #). Περιλαμβάνει μόνο ένα μικρό σύνολο εργαλείων σε σύγκριση με τα άλλα συστήματα. Δεν περιλαμβάνει υποστήριξη για plug-ins. Compilers δεν περιλαμβάνονται στην Visual Studio Express έκδοση, αλλά είναι διαθέσιμα ως μέρος ενός κιτ ανάπτυξης λογισμικού των Windows που μπορεί να εγκατασταθεί ξεχωριστά. Η Microsoft στοχεύει στην Express έκδοση σε μαθητές και χομπίστες. Express εκδόσεις δεν χρησιμοποιούν την πλήρη βιβλιοθήκη MSDN,

αλλά η χρήση της βιβλιοθήκης του MSDN Essentials. Οι γλώσσες που διατίθεται ως μέρος της Express έκδοσης είναι τα εξής:

- Visual Basic Express
 - Visual C ++ Express
 - Visual C # Express
 - Visual Web Developer Express
 - Express για τα Windows Phone
-
- Visual Studio LightSwitch - είναι μια έκδοση, ειδικά σχεδιασμένη για τη δημιουργία line-of-business εφαρμογών που στηρίζονται σε υπάρχουσες .NET τεχνολογίες και πλατφόρμες της Microsoft. Η επιχειρηματική λογική μπορεί να γραφτεί είτε σε Visual Basic ή Visual C #. Το εργαλείο μπορεί να εγκατασταθεί ως stand-alone SKU ή ως ένα ολοκληρωμένο πρόσθετο για το Visual Studio 2010 Professional και άνω.
 - Visual Studio Professional - παρέχει μια έκδοση για όλες τις υποστηριζόμενες γλώσσες ανάπτυξης. Είναι διαθέσιμη MSDN υποστήριξη ως MSDN Essentials ή πλήρης βιβλιοθήκη MSDN ανάλογα με την έκδοση αδειών. Υποστηρίζει XML και XSLT επεξεργασία. Περιλαμβάνει εργαλεία όπως Explorer Server και ενοποίηση με τον Microsoft SQL Server. Υποστήριξη της ανάπτυξης των Windows Mobile συμπεριλήφθηκε στο Visual Studio 2005 Standard, ωστόσο, με το Visual Studio 2008, είναι διαθέσιμο μόνο στην επαγγελματική και υψηλότερη εκδόσεις. Η υποστήριξη της ανάπτυξης Windows Phone 7 προστέθηκε για όλες τις εκδόσεις του Visual Studio 2010. Η ανάπτυξη για τα Windows Mobile δεν υποστηρίζεται πλέον στο Visual Studio 2010. Έχει αντικατασταθεί από υποστήριξη για Windows Phone 7.
 - Visual Studio Premium Edition - περιλαμβάνει όλα τα εργαλεία του Visual Studio Professional και προσθέτει επιπλέον λειτουργικότητα, όπως μετρήσεις κώδικα, προφίλ, στατική ανάλυση κώδικα, και βάση δεδομένων δοκιμών μονάδας.

Visual Studio Εργαλεία για το Office

Εργαλεία του Visual Studio για το Office είναι ένα SDK και μια προσθήκη για το Visual Studio το οποίο περιλαμβάνει εργαλεία για την ανάπτυξη για την οικογένεια προγραμμάτων του Microsoft Office. Στο παρελθόν (για το Visual Studio .NET 2003 και το Visual Studio 2005) ήταν μια ξεχωριστή SKU που υποστήριζε μόνο Visual C # και Visual Basic γλώσσες. Με το Visual Studio 2008, δεν είναι πλέον ένα ξεχωριστό SKU αλλά περιλαμβάνει Professional και υψηλότερες εκδόσεις.

Visual Studio Ultimate

Το Visual Studio Ultimate παρέχει σύνολο του λογισμικού και την ανάπτυξη βάσεων δεδομένων, τη συνεργασία, τις μετρήσεις, την αρχιτεκτονική, τον έλεγχο και την υποβολή εκθέσεων εργαλείων εκτός από τις δυνατότητες που παρέχονται από το Visual Studio Premium. Από το Visual Studio 2010, η Team Suite έκδοση μετονομάστηκε σε έκδοση Ultimate. Το Visual Studio Ultimate προσφέρει ένα υπερσύνολο εργαλείων που βασίζονται στην εφαρμογή Lifecycle Management (ALM).

Κεφάλαιο 3 Βάση Δεδομένων

Μια βάση δεδομένων είναι μια οργανωμένη συλλογή των δεδομένων για έναν ή περισσότερους λόγους, συνήθως σε ψηφιακή μορφή. Τα δεδομένα οργανώνονται συνήθως με μοντέλο τις σχετικές πτυχές της πραγματικότητας (για παράδειγμα, τη διαθεσιμότητα των δωματίων σε ξενοδοχεία), με τρόπο που να υποστηρίζει τις διαδικασίες που απαιτούν αυτές τις πληροφορίες (για παράδειγμα, η εύρεση ενός ξενοδοχείου με κενές θέσεις). Αυτός ο ορισμός είναι πολύ γενικός, και είναι ανεξάρτητος από τη χρησιμοποιούμενη τεχνολογία.

Ο όρος «βάση δεδομένων» μπορεί να περιοριστεί σε ειδικά ορισμένες πτυχές της οργανωμένης συλλογής των δεδομένων και μπορεί να αναφέρεται στη λογική βάση δεδομένων, στην φυσική βάση δεδομένων με τα δεδομένα περιεχομένου σε ηλεκτρονικά δεδομένα αποθήκευσης ή σε πολλούς άλλους υπο-βάση δεδομένων ορισμούς.

Ο όρος βάση δεδομένων εφαρμόζεται ορθά στα δεδομένα καθώς και των υποστηρικτικών δομών δεδομένων τους, και όχι με το σύστημα διαχείρισης βάσεων δεδομένων (που αναφέρεται με το ακρωνύμιο DBMS). Η συλλογή των στοιχείων βάσης δεδομένων με DBMS ονομάζεται ένα σύστημα βάσης δεδομένων.

Ο όρος σύστημα βάσης δεδομένων σημαίνει ότι τα δεδομένα είναι σε κάποιο επίπεδο ποιότητας (μετράται με βάση την ακρίβεια, διαθεσιμότητα, τη χρησιμότητα και προσαρμοστικότητα) και αυτή με τη σειρά του συνεπάγεται συχνά η χρήση μιας γενικής χρήσης του συστήματος διαχείρισης βάσεων δεδομένων (DBMS). Ένα γενικής χρήσης Σύστημα Διαχείρισης Βάσεων Δεδομένων είναι συνήθως ένα πολύπλοκο σύστημα λογισμικού που καλύπτει πολλές απαιτήσεις για χρήσεις, και οι βάσεις δεδομένων που διατηρεί είναι συχνά μεγάλες και πολύπλοκες. Η αξιοποίηση των βάσεων δεδομένων έχει πλέον εξαπλωθεί σε τέτοιο ευρύ βαθμό που σχεδόν σε κάθε τεχνολογία και τα προϊόντα βασίζονται σε βάσεις δεδομένων και Σύστημα Διαχείρισης Βάσεων Δεδομένων για την ανάπτυξη και την εμπορευματοποίηση της, ή ακόμα μπορεί να είναι ενσωματωμένη σε αυτά. Επίσης, οργανισμοί και εταιρείες, από μικρές μέχρι μεγάλες, εξαρτώνται σε μεγάλο βαθμό από τις βάσεις δεδομένων για τις δραστηριότητές τους.

Το γνωστό Σύστημα Διαχείρισης Βάσεων Δεδομένων περιλαμβάνει την Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, MySQL και SQLite. Μια βάση δεδομένων δεν είναι γενικά φορητή σε διαφορετικά ΣΔΒΔ, αλλά διαφορετικά ΣΔΒΔ μπορεί να δια-λειτουργούν σε κάποιο βαθμό με τη χρήση προτύπων όπως SQL και ODBC για να υποστηρίξουν μαζί μια ενιαία αίτηση. Ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων πρέπει επίσης να παρέχει αποτελεσματική run-time εκτέλεση για την ορθή υποστήριξη (π.χ., όσον αφορά τις επιδόσεις, τη διαθεσιμότητα και την ασφάλεια), για πολλούς τελικούς χρήστες, όπως απαιτείται.

Ο σχεδιασμός, η κατασκευή και η συντήρηση μιας σύνθετης βάσης δεδομένων απαιτεί ειδικές δεξιότητες: το προσωπικό που εκτελεί αυτές τις λειτουργίες αναφέρονται ως προγραμματιστές εφαρμογής της βάσης δεδομένων και διαχειριστές βάσεων δεδομένων. Τα καθήκοντά τους υποστηρίζονται από εργαλεία που παρέχονται, είτε ως μέρος της DBMS είτε ως αυτόνομα προϊόντα λογισμικού. Τα εργαλεία αυτά περιλαμβάνουν εξειδικευμένες γλώσσες βάσης δεδομένων που περιλαμβάνουν δεδομένες γλώσσες ορισμού (DDL), οι δεδομένες γλώσσες χειραγώγησης (ΟΘΔ), και οι γλώσσες ερωτημάτων. Αυτές μπορούν να θεωρηθούν ως ειδικού σκοπού γλώσσες προγραμματισμού, ειδικά προσαρμοσμένες για να χειραγωγήσουν τις βάσεις δεδομένων. Μερικές φορές παρέχονται ως επεκτάσεις των υφιστάμενων γλωσσών προγραμματισμού, με προσθήκη τις εντολές βάσης δεδομένων. Οι γλώσσες βάσης δεδομένων είναι γενικές σε ένα συγκεκριμένο μοντέλο δεδομένων, και σε πολλές περιπτώσεις είναι συγκεκριμένες σε έναν τύπο ΣΔΒΔ. Η πιο ευρέως υποστηριζόμενη γλώσσα βάση δεδομένων είναι η SQL, η οποία έχει αναπτυχθεί για το σχεσιακό μοντέλο δεδομένων και συνδυάζει τους ρόλους των δύο DDL, DML, και γλώσσας αναζήτησης.

Ένας τρόπος για την ταξινόμηση των βάσεων δεδομένων περιλαμβάνει το είδος του περιεχομένου τους, για παράδειγμα: Η βιβλιογραφική, εγγράφου-κειμένου, στατιστική, ή αντικείμενα πολυμέσων. Ένας άλλος τρόπος είναι ανάλογα την περιοχή εφαρμογής τους,

όπως για παράδειγμα: λογιστική, συνθέσεις μουσικής, ταινίες, τραπεζική, ανάλογα την κατασκευή, ή την ασφάλεια.

3.1 Ιστορική αναδρομή

Η έννοια βάση δεδομένων έχει εξελιχθεί από τη δεκαετία του 1960 για να διευκολύνει αυξανόμενες δυσκολίες στο σχεδιασμό, την κατασκευή, και συντηρώντας πολύπλοκα συστήματα πληροφοριών (συνήθως με πολλούς ταυτόχρονα τελικούς χρήστες, και με έναν διαφορετικά μεγάλο όγκο δεδομένων). Έχει εξελιχθεί από κοινού με τα συστήματα διαχείρισης βάσεων δεδομένων που επιτρέπουν την αποτελεσματική διαχείριση των βάσεων δεδομένων. Αν και ο όρος βάση δεδομένων και Σύστημα Διαχείρισης Βάσεων Δεδομένων καθορίζουν διαφορετικές οντότητες, που είναι άρρηκτα συνδεδεμένες: οι ιδιότητες μιας βάσης δεδομένων που καθορίζονται από την υποστήριξη DBMS της και αντίστροφα. Μια τεχνική έκθεση του 1962 ήταν η πρώτη που χρησιμοποίησε τον όρο «βάση δεδομένων». Με την πρόοδο της τεχνολογίας στους τομείς των επεξεργαστών, της μνήμης του υπολογιστή, την αποθήκευση στον υπολογιστή, και δικτύων υπολογιστών, τα μεγέθη, τις δυνατότητες και την απόδοση των βάσεων δεδομένων και των αντίστοιχων ΣΔΒΔ τους έχουν αυξηθεί σε τάξεις μεγεθών. Για δεκαετίες, είναι απίθανο ένα πολύπλοκο σύστημα πληροφοριών να μπορεί να χτιστεί αποτελεσματικά χωρίς μια σωστή βάση δεδομένων που υποστηρίζεται από ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων. Η αξιοποίηση των βάσεων δεδομένων έχουν πλέον εξαπλωθεί σε τέτοιο ευρύ βαθμό που σχεδόν σε κάθε τεχνολογία και τα προϊόντα της βασίζονται σε βάσεις δεδομένων και Σύστημα Διαχείρισης Βάσεων Δεδομένων για την ανάπτυξη και την εμπορευματοποίηση τους. Επίσης, οργανισμοί και εταιρείες, από μικρές έως μεγάλες, εξαρτώνται σε μεγάλο βαθμό από τις βάσεις δεδομένων για τις δραστηριότητές τους.

Δεν υπάρχει ευρέως αποδεκτός ακριβής ορισμός για το Σύστημα Διαχείρισης Βάσεων Δεδομένων. Ωστόσο, ένα σύστημα πρέπει να παρέχει σημαντικές λειτουργίες για να χαρακτηριστεί ως ένα DBMS. Ως εκ τούτου για να υποστηρίξει τη συλλογή των δεδομένων του πρέπει να πληροί τις αντίστοιχες απαιτήσεις ευχρηστίας (που σε μεγάλο βαθμό καθορίζονται από τις απαιτήσεις που αναφέρονται παρακάτω) για να χαρακτηριστεί ως μια βάση δεδομένων. Έτσι, μια βάση δεδομένων και την υποστήριξη του DBMS ορίζεται από ένα σύνολο γενικών απαιτήσεων. Σχεδόν όλα τα υπάρχοντα ώριμα προϊόντα ΣΔΒΔ πληρούν τις απαιτήσεις αυτές σε μεγάλο βαθμό, ενώ οι λιγότερο ώριμες συγκλίνουν για την αντιμετώπισή τους.

Η εισαγωγή του όρου βάση δεδομένων συνέπεσε με την δυνατότητα αποθήκευσης άμεσης πρόσβασης από τα μέσα της δεκαετίας του 1960 και μετά. Ο όρος αντιπροσώπευε μια αντίθεση με την τα συστήματα του παρελθόντος, επιτρέποντας κοινή διαδραστική χρήση και όχι καθημερινή μαζική επεξεργασία.

Στα πρώτα συστήματα βάσεων δεδομένων, η αποτελεσματικότητα ήταν ίσως το κύριο μέλημα, αλλά είχαν ήδη αναγνωρίσει ότι υπήρχαν και άλλοι σημαντικοί στόχοι. Ένας από τους βασικούς στόχους ήταν να κάνουν τα δεδομένα ανεξάρτητα από τη λογική των προγραμμάτων εφαρμογής, έτσι ώστε τα ίδια δεδομένα θα μπορούσαν να διατεθούν σε διαφορετικές εφαρμογές.

Η πρώτη γενιά των συστημάτων βάσεων δεδομένων ήταν πλοήγησης, πρόσβαση σε τυπικά δεδομένα της εφαρμογής, ακολουθώντας δείκτες από το μια καταγραφή στην άλλη. Τα δύο βασικά μοντέλα δεδομένων αυτή τη στιγμή ήταν το ιεραρχικό μοντέλο, που συνοψίζεται από το σύστημα IMS της IBM, και η Codasyl μοντέλο (μοντέλο Network), που εφαρμόζονται σε διάφορα προϊόντα, όπως IDMS.

Το Σχεσιακό μοντέλο, που προτάθηκε για πρώτη φορά το 1970 από τον Edgar F. Codd, απομακρύνθηκε από αυτή την παράδοση, επιμένοντας στο ότι οι αιτήσεις θα πρέπει να αναζητάνε δεδομένα από το περιεχόμενο, και όχι από τους συνδέσμους. Αυτό κατέστησε αναγκαίο να επιτραπεί το περιεχόμενο της βάσης δεδομένων να εξελιχθεί χωρίς συνεχή επανεγγραφή των εφαρμογών. Σχεσιακά συστήματα δημιουργούν σοβαρές απαιτήσεις όσον αφορά τους πόρους επεξεργασίας, και δεν ήταν μέχρι τα μέσα της δεκαετίας του 1980 ότι το

υλικό υπολογιστών έγινε αρκετά ισχυρό ώστε να τους επιτρέψει να αναπτυχθούν ευρέως. Στις αρχές της δεκαετίας του 1990, ωστόσο, τα σχεσιακά συστήματα ήταν κυρίαρχα για όλα τα μεγάλης κλίμακας εφαρμογές επεξεργασίας δεδομένων, και παραμένουν κυρίαρχα και σήμερα (2012), εκτός από εξειδικευμένες περιοχές. Η κυρίαρχη γλώσσα βάσης δεδομένων είναι το πρότυπο SQL για το Σχεσιακό μοντέλο, το οποίο έχει επηρεάσει γλώσσες βάσης δεδομένων και για άλλα μοντέλα δεδομένων.

Επειδή το σχεσιακό μοντέλο δίνει έμφαση στην αναζήτηση αντί πλοήγησης, δεν έχει κάνει τις σχέσεις μεταξύ διαφόρων οντοτήτων ρητά με τη μορφή δεικτών, αλλά αντιπροσωπεύει τη χρήση τους και όχι τα πρωτεύοντα κλειδιά και τα ξένα κλειδιά. Αν και αυτό είναι μια καλή βάση για ένα ερώτημα γλώσσας, είναι λιγότερο κατάλληλο ως γλώσσα μοντελοποίησης. Για το λόγο αυτό ένα διαφορετικό μοντέλο, το μοντέλο Οντοτήτων-Σχέσεων που προέκυψε λίγο αργότερα (1976), κέρδισε τη δημοτικότητα για το σχεδιασμό της βάσης δεδομένων.

Κατά την περίοδο μετά το 1970 η τεχνολογία βάσης δεδομένων είχε συμβαδίσει με την αύξηση των πόρων που υπήρχαν διαθέσιμοι από την υπολογιστική πλατφόρμα, ιδιαίτερα με την ταχεία αύξηση της δυναμικότητας και την ταχύτητα (και μείωση της τιμής) των συστημάτων αποθήκευσης, και την αύξηση της ικανότητας της κύριας μνήμης. Αυτό επέτρεψε μεγαλύτερες βάσεις δεδομένων και υψηλότερα διερχόμενα που πρέπει να επιτευχθούν.

Η ακαμψία του σχεσιακού μοντέλου, στο οποίο όλα τα δεδομένα τηρούνται σε πίνακες με μια σταθερή δομή γραμμών και στηλών, όλο και περισσότερο θεωρείται ως περιορισμός κατά το χειρισμό πληροφοριών που είναι πιο πλούσια ή μεγαλύτερη ποικιλία ως προς τη δομή του «βιβλίου προς βιβλίο» της παραδοσιακής παρουσίασης δεδομένων των εταιρικών πληροφοριακών συστημάτων: για παράδειγμα, οι βάσεις δεδομένων εγγράφου, βάσεις δεδομένων της μηχανικής, βάσεις δεδομένων πολυμέσων, ή βάσεις δεδομένων που χρησιμοποιούνται στη μοριακή επιστήμη.

Σε αυτό το μοντέλο μια ιεραρχική σχέση μεταξύ δύο ειδών καταγραφής (που αντιπροσωπεύουν πραγματικές οντότητες) αποδεικνύεται μέσω της κατασκευής σετ. Ένα σετ αποτελείται από κυκλικές συνδεδεμένες λίστες, όπου ένας τύπος εγγραφής, ο ιδιοκτήτης ή ο γονέας που, εμφανίζεται μία φορά σε κάθε κύκλο, και ένα δεύτερο τύπο εγγραφής, την υποδεέστερη ή παιδί, μπορεί να εμφανιστεί πολλές φορές σε κάθε κύκλο. Με αυτόν τον τρόπο μια ιεραρχία μπορεί να δημιουργηθεί ανάμεσα σε δύο τύπους αρχείων, π.χ., τύπου A είναι ο ιδιοκτήτης του B. Την ίδια στιγμή ένα άλλο που μπορεί να οριστεί, όπου B είναι ο ιδιοκτήτης της A. Έτσι, όλα τα σετ αποτελείται από ένα γενικό κατευθυνόμενος γράφος (ιδιοκτησία καθορίζει μια κατεύθυνση), ή την κατασκευή του δικτύου. Η πρόσβαση στα αρχεία είναι είτε διαδοχική (συνήθως σε κάθε τύπο εγγραφής), είτε με πλοήγηση στη συνδεδεμένη κυκλική λίστες.

Αυτό το μοντέλο είναι πιο γενικό και ισχυρό από το ιεραρχικό, και ήταν το πιο δημοφιλές πριν αντικατασταθεί από το Σχεσιακό μοντέλο. Έχει τυποποιηθεί από την CODASYL. Δημοφιλή προϊόντα ΣΔΒΔ που χρησιμοποιούνται ήταν το Σύνολο “Cincom Systems” και το IDMS Cullinet.

Ένα ανεστραμμένο αρχείο ή ένας ανεστραμμένος δείκτης του πρώτου αρχείου, από ένα πεδίο σε αυτό το αρχείο (στον τομέα αντιστροφή), είναι ένα δεύτερο αρχείο στο οποίο αυτό το πεδίο είναι το κλειδί.

3.2 Sql Server

Ο SQL Server είναι μια σχεσιακή βάση δεδομένων, η οποία αναπτύσσεται από τη Microsoft. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η T-SQL και η ANSI SQL. Ο SQL Server βγήκε για πρώτη φορά στην αγορά το 1989 σε συνεργασία με την Sybase. Η κύρια μονάδα αποθήκευσης στοιχείων είναι μια βάση δεδομένων, η οποία αποτελείται από μια συλλογή πινάκων και κώδικα.

Η κεντρική βάση δεδομένων του SQL υποστηρίζει διαφορετικούς τύπους, συμπεριλαμβανομένων των ακεραίων αριθμών, αριθμών κινητής υποδιαστολής, δεκαδικών, αλφαριθμητικών, Varchar (σειρές χαρακτήρων μεταβλητού μήκους), δυαδικών αριθμών (για τα μη δομημένα δεδομένα), κειμένων (για κείμενα). Επιτρέπει επίσης καθορισμένους από το χρήστη σύνθετους τύπους δεδομένων (UDTs), δηλαδή τύπους που βασίζονται στους βασικούς τύπους αλλά μπορούν να τροποποιηθούν. Τα στοιχεία στη βάση δεδομένων αποθηκεύονται σε ένα (ή περισσότερα) αρχεία με επέκταση .mdf.

Τα δευτεροβάθμια στοιχεία αποθηκεύονται στο αρχείο με επέκταση .ndf. Το αρχείο καταγραφής το οποίο περιέχει όλες τις πρόσφατες αλλαγές στη βάση δεδομένων αποθηκεύεται σε αρχείο με επέκταση .ldf. Ο χώρος αποθήκευσης που διατίθεται σε μια βάση δεδομένων διαιρείται σε διαδοχικά αριθμημένες σελίδες, κάθε μία από τις οποίες έχει μέγεθος 8 KB.

Οι σελίδες αποθηκεύονται στην ενδιάμεση (buffer) μνήμη RAM για να ελαχιστοποιηθεί η μεταφορά δεδομένων προς και από τον σκληρό δίσκο. Οποιαδήποτε σελίδα 8 KB μπορεί να είναι αποθηκευμένη στη μνήμη, και το σύνολο όλων των σελίδων που αποθηκεύονται σε μία περίοδο καλείται λανθάνουσα μνήμη (cache). Το ποσό μνήμης που είναι διαθέσιμο στον κεντρικό διακομιστή SQL αποφασίζει πόσες σελίδες θα εναποθηκευθούν στη λανθάνουσα μνήμη. Ο SQL Server έχει διαφορετικούς και έξυπνους αλγόριθμους για την καλύτερη απόδοση της λανθάνουσας μνήμης.

Ο κεντρικός διακομιστής SQL εξασφαλίζει ότι οποιαδήποτε αλλαγή στα στοιχεία είναι "ACID", δηλ. χρησιμοποιεί τις "συναλλαγές" (transactions) που εξασφαλίζουν ότι οποιαδήποτε λειτουργία είτε ολοκληρώνεται συνολικά είτε αποτυγχάνει συνολικά, αλλά δεν αφήνει ποτέ τη βάση δεδομένων σε μία ενδιάμεση κατάσταση. Χρησιμοποιώντας τις συναλλαγές αυτές, μια ακολουθία εντολών μπορεί να αποτελέσει μία ομάδα, με την προϋπόθεση ότι είτε όλες οι εντολές θα πετύχουν ή καμία δεν θα πετύχει. Ο κεντρικός διακομιστής SQL επιτρέπει σε πολλούς πελάτες να χρησιμοποιούν την ίδια βάση δεδομένων ταυτόχρονα.

Υπό αυτήν τη μορφή, θα πρέπει να ελέγξει την ταυτόχρονη πρόσβαση στα κοινά στοιχεία, για να εξασφαλίσει την ακεραιότητα των στοιχείων – Π.χ όταν πολλαπλάσιοι πελάτες ενημερώνουν τα ίδια στοιχεία, ή οι πελάτες προσπαθούν να διαβάσουν ένα στοιχείο το οποίο αλλάζεται από έναν άλλο πελάτη. Ο κεντρικός διακομιστής SQL παρέχει δύο τρόπους ελέγχου του ταυτοχρονισμού: απαισιόδοξος ταυτοχρονισμός και αισιόδοξος ταυτοχρονισμός.

Όταν ο απαισιόδοξος έλεγχος ταυτοχρονισμού χρησιμοποιείται, ο κεντρικός διακομιστής SQL ελέγχει την ταυτόχρονη πρόσβαση με τη χρησιμοποίηση κλειδώματος (locks). Τα κλειδώματα μπορούν είτε να διαμοιράζονται είτε να είναι αποκλειστικά. Το αποκλειστικό κλειδώμα χορηγεί στο χρήστη την αποκλειστική πρόσβαση στα στοιχεία (συνήθως κατά την αλλαγή στοιχείων) - κανένας άλλος χρήστης δεν μπορεί να έχει πρόσβαση στα στοιχεία εφόσον υπάρχει κλειδώμα. Το κοινό κλειδώμα χρησιμοποιείται όταν διαβάζεται κάποιο στοιχείο - πολλαπλοί χρήστες μπορούν να διαβάσουν από τα στοιχεία που κλειδώνονται με ένα κοινό κλειδώμα, αλλά να μην αποκτούν αποκλειστικό.

Το κλειδώμα μπορεί να εφαρμοστεί σε διαφορετικά επίπεδα - σε ολόκληρους πίνακες, σελίδες ή ακόμα και σε διαφορετικές γραμμές. Το επίπεδο που χρησιμοποιείται καθορίζεται σε μια βάση δεδομένων από περίπλοκους αλγόριθμους του SQL Server, οι οποίοι έχουν σκοπό την γρηγορότερη και καλύτερη λειτουργία από χιλιάδες χρήστες η ακόμη και εκατοντάδες χιλιάδες την ίδια στιγμή. Ένας SQL Server είναι δυνατόν να πραγματοποιήσει μέχρι και 1.000.000 transactions το δευτερόλεπτο με το ανάλογο υλικό. Ο SQL Server λειτουργεί σε 32 ή 64 bit και υποστηρίζει μέχρι και 256 πυρήνες (CPU) Τυπική χρήση είναι οι 8-16 επεξεργαστές.

Η ερώτηση (query) είναι ο κύριος τρόπος για την ανάκτηση στοιχείων από μια βάση δεδομένων. Η ερώτηση εκφράζεται χρησιμοποιώντας μια παραλλαγή της αποκαλούμενου SQL T-SQL, είναι μια διάλεκτος SQL που αναπτύχθηκε από την Microsoft και Sybase. Η T-SQL είναι πολύ κοντά στα ANSI standards που έχουν καθιερωθεί διεθνώς, σε αντιδιαστολή με άλλες διαλέκτους όπως η PL-SQL της Oracle που διαφέρουν περισσότερο από το ANSI standards.

Η ερώτηση διευκρινίζει επεξηγηματικά αυτό που είναι να ανακτηθεί από την βάση δεδομένων. Η ερώτηση υποβάλλεται σε επεξεργασία από τον επεξεργαστή ερώτησης, ο οποίος υπολογίζει τη σειρά των βημάτων που θα είναι απαραίτητα για να ανακτηθούν τα ζητούμενα στοιχεία. Η ακολουθία ενεργειών απαραίτητων να εκτελέσουν μια ερώτηση καλείται σχέδιο ερώτησης.

Υπάρχουν πολλοί τρόποι για να υποβληθεί σε επεξεργασία η ίδια ερώτηση. Π.χ, μια ερώτηση που ενώνει 2 πίνακες A & B μαζί θα μπορούσε να γίνει ή πηγαίνοντας πρώτα στο A βρισκοντας τα στοιχεία και μετά ενώνοντας με τα στοιχεία του πίνακα B, ή πρώτα από το B και μετά πίνακα A. Ο SQL Server και όχι ο Προγραμματιστής παίρνει την απόφαση μέσω στατιστικών στοιχείων και άλλων στοιχείων που διαθέτει για τους 2 πίνακες να ακολουθήσει την διαδρομή A-B γιατί αυτός είναι ο γρηγορότερος τρόπος.

Στις procedural γλώσσες προγραμματισμού πχ C, Pascal, .NET, ο προγραμματιστής θα πρέπει να κάνει την απόφαση το οποίο πολλές φορές σημαίνει λάθος απόφαση. Αυτό καλείται βελτιστοποίηση ερώτησης και εκτελείται από τον επεξεργαστή ερώτησης. Ο SQL παίρνει πολύ περίπλοκες αποφάσεις βασισμένος σε αλγόριθμους που έχουν αναπτυχθεί τα τελευταία 20-30 χρόνια πάνω στο αντικείμενο.

Μια παρόμοια ερώτηση σε μια προηγούμενη έκδοση του SQL Server θα μπορούσε να πάρει 10-20 φορές περισσότερο χρόνο λόγω του διαφορετικού αλγόριθμου. Γενικά κάθε καινούργια έκδοση βελτιώνει την αποτελεσματικότητα, και άρα τον χρόνο εκτέλεσης. Εδώ ακριβώς είναι ένα από τα πλεονεκτήματα προγραμματισμού ενός προβλήματος σε μια βάση δεδομένων σε σύγκριση με C, Pascal, C++ κτλ. Βέβαια όλα τα προβλήματα δεν είναι κατάλληλα για εφαρμογή σε βάση δεδομένων.

Ο SQL Server επιτρέπει επίσης αποθηκευμένες ερωτήσεις (stored procedures). Οι αποθηκευμένες ερωτήσεις είναι ερωτήσεις με παραμέτρους T-SQL, οι οποίες αποθηκεύονται στον SQL Server. Μια stored procedure τυπικά είναι από 100-1000 γραμμές κώδικα SQL. Οι αποθηκευμένες ερωτήσεις μπορούν να δεχτούν τιμές που στέλνονται από τον πελάτη ως παράμετροι εισαγωγής, και να στείλουν τα αποτελέσματα ως παραμέτρους παραγωγής. Μπορούν να καλέσουν functions, και άλλες stored procedures. Οι stored procedures είναι γρηγορότερες από κατασκευή σε σχέση με απλές ακολουθίες SQL.

Ο Microsoft SQL Server 2005 – 2008 περιλαμβάνει ένα module που λέγεται SQL CLR μέσω του οποίου ενσωματώνει το .NET μέσα στον SQL Server. Με το SQL CLR, οι αποθηκευμένες διαδικασίες μπορούν να γραφτούν σε οποιαδήποτε γλώσσα .NET συμπεριλαμβανομένου C# και VB.NET και να δημιουργήσουν μια stored procedure. Αυτό σημαίνει ότι ο SQL Server έχει όλες τις βιβλιοθήκες και πλεονεκτήματα του .NET, αυτόχθονα μέσα στο περιβάλλον του, τα οποία μπορεί να τα καλέσει οποιαδήποτε στιγμή.

Ο Μεσίτης Υπηρεσιών (Service Broker) που τρέχει ως μέρος της μηχανής βάσεων δεδομένων, παρέχει μια αξιόπιστη πλατφόρμα μηνυμάτων και αναμονής μηνυμάτων για τις εφαρμογές κεντρικών SQL Server. Ο Service Broker έχει ανώτερες δυνατότητες προγραμματισμού στο μέλλον (scheduling). Οι υπηρεσίες συγχρονισμού (Replication) του SQL Server χρησιμοποιούνται από τον SQL Server για και να συγχρονίσουν βάσεις δεδομένων, είτε ολοκληρωτικά είτε ένα υποσύνολο των αντικειμένων. Υπάρχουν τουλάχιστον 4 διαφορετικοί μέθοδοι (log shipping, mirroring, replication, back-ups, clustering) για τον συγχρονισμό. Αυτό βοηθάει επίσης και σε περιπτώσεις ολοκληρωτικής καταστροφής της μητρικής βάσης δεδομένων, δίνοντας την δυνατότητα ακόμη και για αυτόματη λειτουργία της δευτερεύουσας συγχρονισμένης βάσης δεδομένων.

Οι υπηρεσίες ανάλυσης (Analysis Services) SQL Server προσθέτουν τις ικανότητες OLAP και περίπλοκης ανάκτησης δεδομένων για τις βάσεις δεδομένων SQL, με περίπλοκη ανάλυση χωρίς την ανάγκη για δύσκολες SQL queries. Υποστηρίζεται το MOLAP, ROLAP και HOLAP για τα στοιχεία. Ο SQL Server έχει υπηρεσίες Reporting για στοιχεία που συγκεντρώνονται από τις βάσεις δεδομένων SQL Server. Με αυτόν τον τρόπο είναι δυνατό να παραχθεί μια έκθεση Ιστού (Web Report) σχεδόν αυτόματα χωρίς προγραμματισμό HTML, CSS ή JavaScript, μέσα από το Visual Studio.

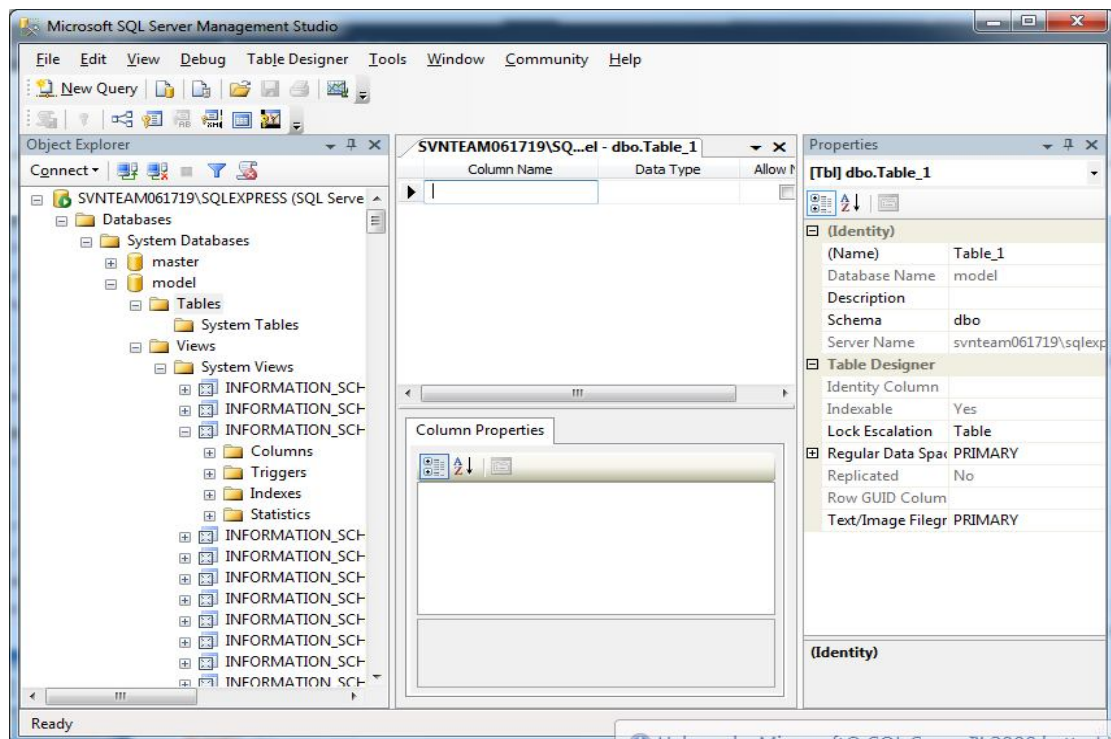
Οι υπηρεσίες ανακοίνωσης (Notification Services) δίνουν την δυνατότητα στον χρήστη να παράγει μηνύματα ηλεκτρονικού ταχυδρομείου με πολύ λίγο κόπο. Το Integration Services του SQL χρησιμοποιούνται για να ενσωματώσουν στοιχεία από διαφορετικές πηγές

δεδομένων. Χρησιμοποιεί τις ικανότητες ETL. Οι Integration Services περιλαμβάνουν εργαλεία GUI για να χτίσουν διάφορες λειτουργίες ροών δεδομένων, όπως η εξαγωγή στοιχείων από διάφορες πηγές, η συζήτηση στοιχείων, ο μετασχηματισμός των στοιχείων συμπεριλαμβανομένης της συνάθροισης, του διπλασιασμού και η συγχώνευσης. Ο SQL Server μπορεί επίσης να αναζητήσει μέσα σε κείμενα (Full Text Search Service). Ο δείκτης αναζήτησης μπορεί να δημιουργηθεί σε οποιαδήποτε στήλη και έτσι η αναζήτηση πραγματοποιείται άμεσα. Για παράδειγμα, κείμενα Word και έγγραφα pdf αποθηκευμένα στον SQL Server μπορούν αναζητηθούν πολύ γρήγορα.

3.3 SQL Server Management Studio

Το SQL Server Management Studio είναι ένα ολοκληρωμένο περιβάλλον για την πρόσβαση, τη διαμόρφωση, τη διαχείριση και την ανάπτυξη όλων των συνιστωσών του SQL Server. Συνδυάζει μια ευρεία ομάδα γραφικών εργαλείων με μια σειρά από πλούσια σενάρια για την παροχή πρόσβασης στον SQL Server για τους προγραμματιστές και τους διαχειριστές όλων των επιπέδων. Το SQL Server Management Studio συνδυάζει τα χαρακτηριστικά της διαχείρισης επιχείρησης, της ανάλυσης ερωτήματος βάσης δεδομένων, και διαχείρισης ανάλυσης, που περιλαμβάνονται σε προηγούμενες εκδόσεις του SQL Server, σε ένα ενιαίο περιβάλλον.

Επιπλέον, ο SQL Server Management Studio λειτουργεί με όλες τις συνιστώσες του SQL Server, όπως υπηρεσία αναφοράς, ολοκληρωμένες υπηρεσίες, και SQL Server Compact 3.5 SP2. Οι προγραμματιστές νιώθουν εξοικειωμένοι με το περιβάλλον και οι διαχειριστές βάσεων δεδομένων απολαμβάνουν μια ενιαία συνολική χρησιμότητα που συνδυάζει εύκολα στη χρήση γραφικά εργαλεία με πλούσιες δυνατότητες συγγραφής κώδικα.



3.0.1 - Περιβάλλον εργασίας SQL Server Management Studio

3.4 Γλώσσα SQL

Η SQL (Structured Query Language) είναι μία γλώσσα υπολογιστών στις βάσεις δεδομένων, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System, RDBMS) και η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα. Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα. Η SQL ήταν μία από τις πρώτες γλώσσες για το σχεσιακό μοντέλο του Edgar F. Codd, στο σημαντικό άρθρο του το 1970, και έγινε η πιο ευρέως χρησιμοποιούμενη γλώσσα για τις σχεσιακές βάσεις δεδομένων.

3.4.1 Ιστορική αναδρομή

Η SQL αναπτύχθηκε στην IBM από τους Andrew Richardson, Donald C. Messerly και Raymond F. Boyce, στις αρχές της δεκαετίας του 1970. Αυτή η έκδοση, αποκαλούμενη αρχικά SEQUEL, είχε ως σκοπό να χειριστεί και να ανακτήσει τα στοιχεία που αποθηκεύτηκαν στο πρώτο RDBMS της IBM, το System R.

Το πρώτο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) ήταν το RDMBS που αναπτύχθηκε στο MIT, στις αρχές της δεκαετίας του 1970 και η Ingres, που αναπτύχθηκε το 1974 στο Πανεπιστήμιο Μπέρκλεϋ. Η Ingres εφάρμοσε μία γλώσσα διατύπωσης ερωτήσεων γνωστή ως QUEL, το οποίο αντικαταστάθηκε αργότερα στην αγορά από την SQL.

Προς το τέλος της δεκαετίας του 70 η Relational Software (τόρα Oracle Corporation) είδε τη δυνατότητα αυτών που περιγράφηκαν από Codd, Chamberlin, και Boyce και αναπτύξε την SQL βασισμένο στο RDBMS, με τις φιλοδοξίες πώλησης του στο Αμερικανικό ναυτικό, την Κεντρική Υπηρεσία Πληροφοριών και άλλες Αμερικανικές Υπηρεσίες.

Το καλοκαίρι του 1979, η Relational Software εισήγαγε την πρώτη διαθέσιμη στο εμπόριο εφαρμογή του SQL και νίκησε την IBM με τη διάθεση του πρώτου εμπορικού RDBMS για μερικές εβδομάδες.

Η γλώσσα SQL υποδιαιρείται σε διάφορα γλωσσικά στοιχεία, που περιλαμβάνουν Clauses, οι οποίες είναι σε μερικές περιπτώσεις προαιρετικές, αλλά απαραίτητα συστατικά των δηλώσεων και ερωτήσεων.

- Expressions που μπορούν να παραγάγουν είτε τις κλιμακωτές τιμές είτε πίνακες που αποτελούνται από στήλες και σειρές στοιχείων.
- Predicates που διευκρινίζουν τους όρους που μπορούν να αξιολογηθούν σαν σωστό ή λάθος.
- Queries που ανακτούν τα στοιχεία βασισμένες σε ειδικά κριτήρια.
- Statements που μπορούν να έχουν μια επίδραση στα σχήματα και τα στοιχεία, ή που μπορούν να ελέγξουν τη ροή του προγράμματος και τις συνδέσεις από άλλα προγράμματα.
- Το κενό αγνοείται γενικά στις Statements και τις Queries SQL. Ένα κενό είναι όμως απαραίτητο για να ξεχωρίζει Statements όπως και στην κανονική γραφή κειμένων.

Παράδειγμα Επερωτήματος

Ένα επερώτημα (αγγλ. query) συνήθως αποτελείται από:

```
SELECT *  
FROM Sygrafeas  
WHERE epitheto = 'Solomos'
```

AND onoma = 'Dionysios'

αλλά μπορεί επίσης να περιλαμβάνει GROUP BY, ORDER BY και άλλα:

```
SELECT Book.title,  
COUNT(*) AS No_of_Authors  
FROM Book  
JOIN Book_author ON Book.isbn = Book_author.isbn  
GROUP BY Book.title  
HAVING COUNT(*) > 1  
ORDER BY No_of_Authors
```

Με αποτέλεσμα:

Title	No_of_Authors
SQL Examples and Guide	2
The Joy of SQL	3
An Introduction to SQL	5
Pitfalls of SQL	6

Κεφάλαιο 4 Εφαρμογή

4.1 Περιγραφή της εφαρμογής

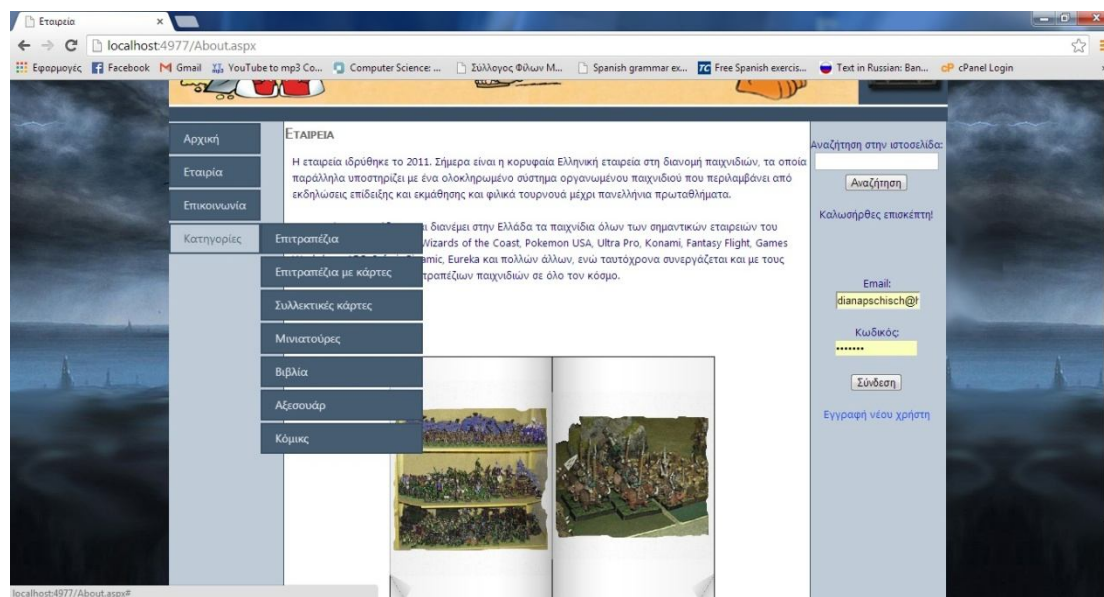
Η εφαρμογή αποτελεί μια πλατφόρμα ηλεκτρονικού εμπορίου, συγκεκριμένα ένα ηλεκτρονικό κατάστημα που εμπορεύεται διαφόρων ειδών επιτραπέζια παιχνίδια, βιβλία, κόμικς και αξεσουάρ. Υπάρχουν τρεις διαφορετικοί ρόλοι χρηστών που μπορούν να επισκεφτούν την ιστοσελίδα, οι επισκέπτες (guest), οι χρήστες (user) και οι διαχειριστές(administrator).

Ως επισκέπτης, μπορεί κανείς να δει τα προϊόντα που εμπορεύεται το κατάστημα, να δει τις λεπτομέρειες των προϊόντων, όπως η τιμή και η περιγραφή. Μπορεί, επίσης, να δει πληροφορίες σχετικά με την εταιρεία και τα στοιχεία επικοινωνίας της.



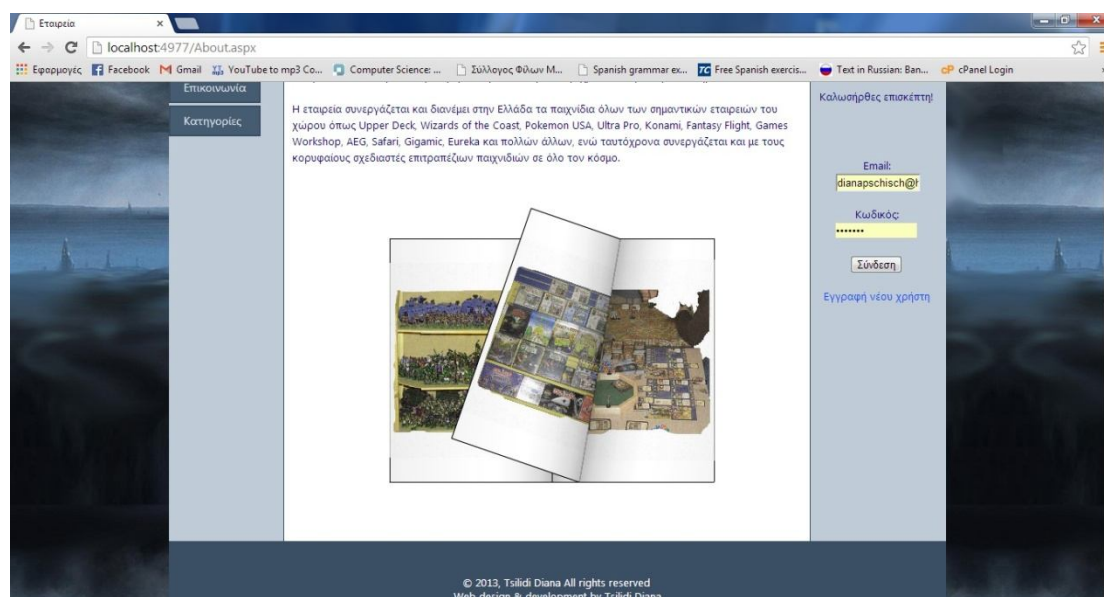
4.1 - Αρχική σελίδα

Αρχικά υπάρχουν επτά κατηγορίες προϊόντων, αυτές που συνηθίζεται να υπάρχουν σε αντίστοιχα καταστήματα που εμπορεύονται επιτραπέζια παιχνίδια. Μπορούν, όμως, να προστεθούν περισσότερες κατηγορίες από τον διαχειριστή της ιστοσελίδας, ανάλογα με τις ανάγκες του καταστήματος, αφού οι κατηγορίες εμφανίζονται στη σελίδα δυναμικά με τη βοήθεια της βάσης δεδομένων.



4.2 - Κατηγορίες προϊόντων

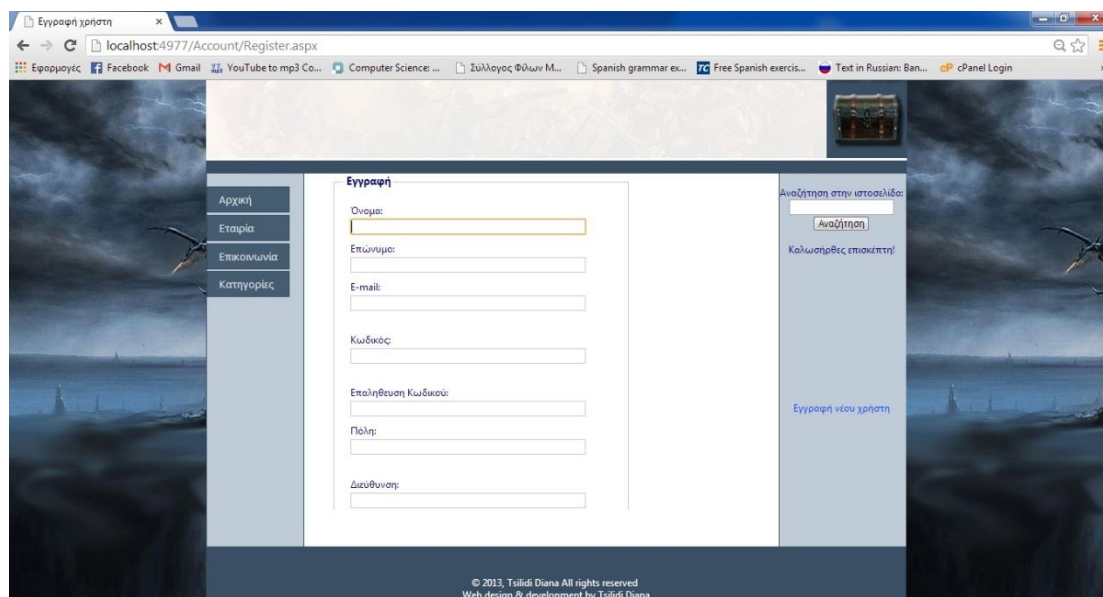
Στη σελίδα που αναφέρεται στην εταιρεία, ο επισκέπτης βλέπει πληροφορίες σχετικά με το κατάστημα, καθώς και φωτογραφίες από το χώρο του καταστήματος ή φωτογραφίες των προϊόντων.



4.3 - Φωτογραφίες (Flash book)

Ο επισκέπτης δεν έχει τη δυνατότητα να παραγγείλει κάποιο προϊόν χωρίς πρώτα να δημιουργήσει το λογαριασμό του στο κατάστημα. Παρακάτω φαίνεται η φόρμα που θα πρέπει να συμπληρώσει κανείς για να εγγραφεί στην ιστοσελίδα και να μπορέσει να παραγγείλει κάποιο προϊόν. Κάποια στοιχεία, όπως η διεύθυνση ηλεκτρονικού ταχυδρομείου ή η φυσική διεύθυνση, είναι απαραίτητα για τη δημιουργία λογαριασμού.

Κατασκευή ηλεκτρονικού καταστήματος



4.4 - Εγγραφή χρήστη

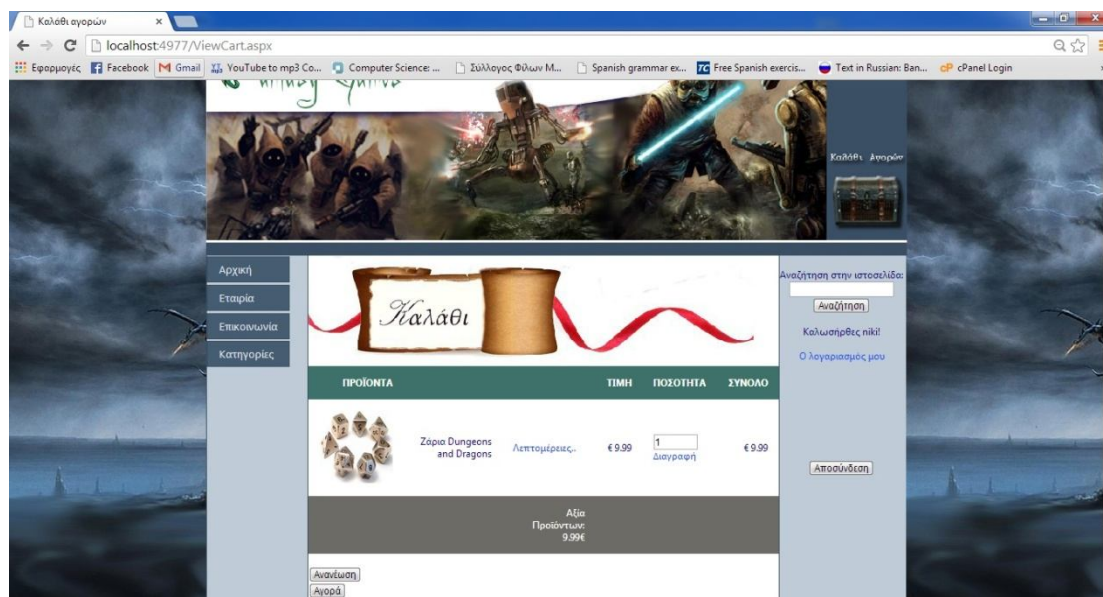
Επίσης, στη σελίδα υπάρχει η δυνατότητα για αναζήτηση συγκεκριμένου προϊόντος χρησιμοποιώντας το όνομά του ή κάποιες λέξεις κλειδιά, ακόμα και μερικά γράμματα από το όνομα του προϊόντος.



4.5 - Αναζήτηση στη σελίδα

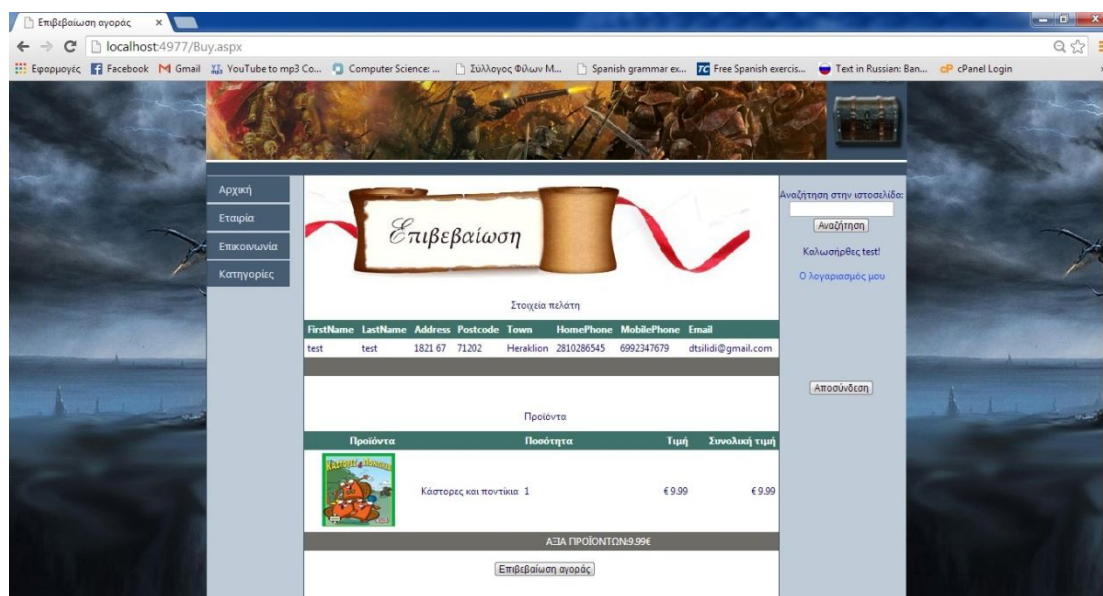
Αφού ο επισκέπτης δημιουργήσει λογαριασμό και συνδεθεί με τη διεύθυνση ηλεκτρονικού του ταχυδρομείου (e-mail) και τον κωδικό που έχει επιλέξει, αλλάζει ρόλο και γίνεται πλέον χρήστης. Ως χρήστης, λοιπόν, έχει ένα δικό του καλάθι αγορών και μπορεί να επιλέξει προϊόντα είτε από την αρχική σελίδα, που προβάλλει τις νέες παραλαβές, είτε από τις συγκεκριμένες κατηγορίες προϊόντων.

Μέσα στο καλάθι μπορεί κανείς να δει την εικόνα των προϊόντων που έχει επιλέξει, την ονομασία, την τιμή και την περιγραφή τους. Φυσικά, μπορεί να τροποποιήσει και την ποσότητα που θέλει να αγοράσει.



4.6 - Καλάθι αγορών

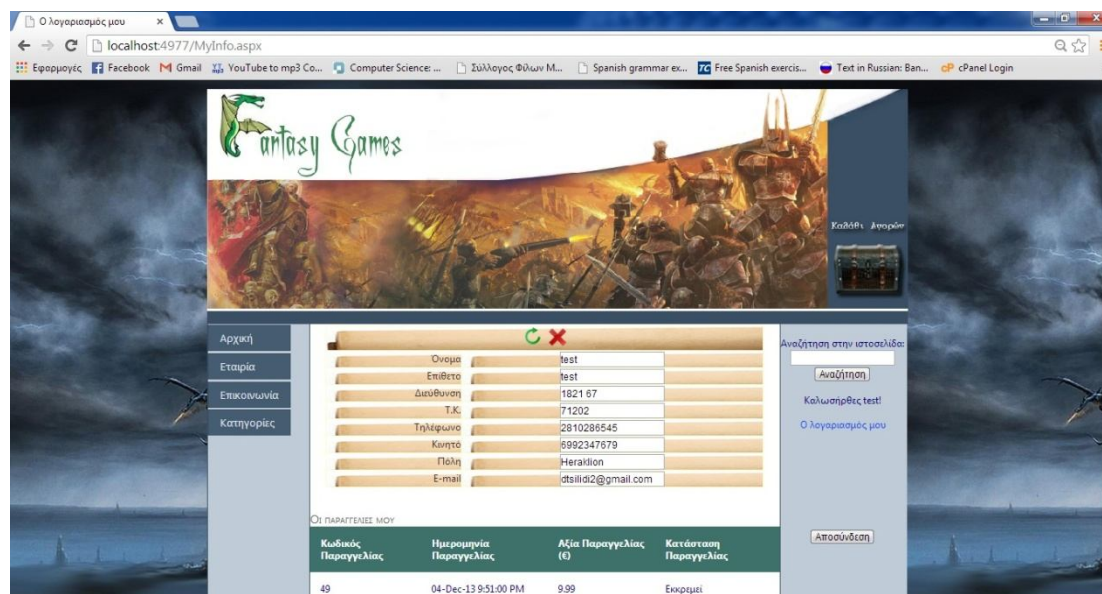
Όταν ο πελάτης είναι πια σίγουρος για το εμπόρευμα που έχει επιλέξει να αγοράσει, πατάει το κουμπί “Αγορά” και μεταφέρεται σε μια άλλη σελίδα, στην οποία καλείται να επιβεβαιώσει την αγορά. Εκεί μπορεί να ξαναδεί τα προϊόντα που επέλεξε και όλες τις πληροφορίες σχετικά με αυτά, καθώς και τα στοιχεία του λογαριασμού του ίδιου. Έτσι, αν όλα βαίνουν καλά, επιβεβαιώνει την αγορά και αυτομάτως δημιουργείται μια παραγγελία.



4.7 - Επιβεβαίωση αγοράς

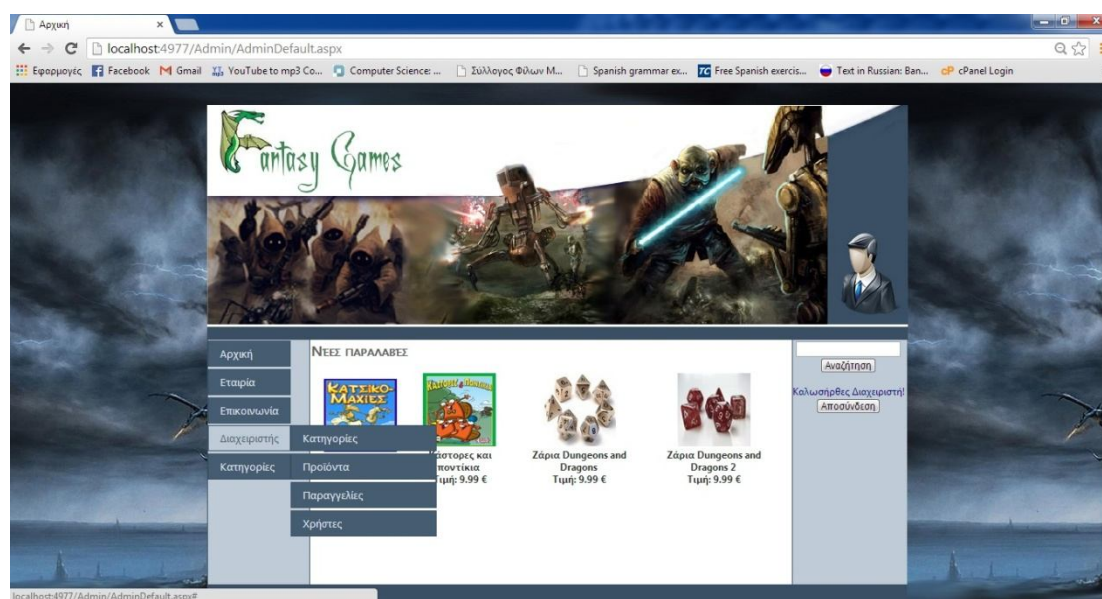
Ο χρήστης μπορεί να δει τις παραγγελίες που έχει κάνει στη σελίδα “Ο λογαριασμός μου”. Εκεί, επιπλέον, μπορεί να δει και να επεξεργαστεί τα στοιχεία του λογαριασμού του, δηλαδή το ονοματεπώνυμό του, τη διεύθυνσή του, το σταθερό και το κινητό του τηλέφωνο, την πόλη και τον ταχυδρομικό κώδικα.

Κατασκευή ηλεκτρονικού καταστήματος



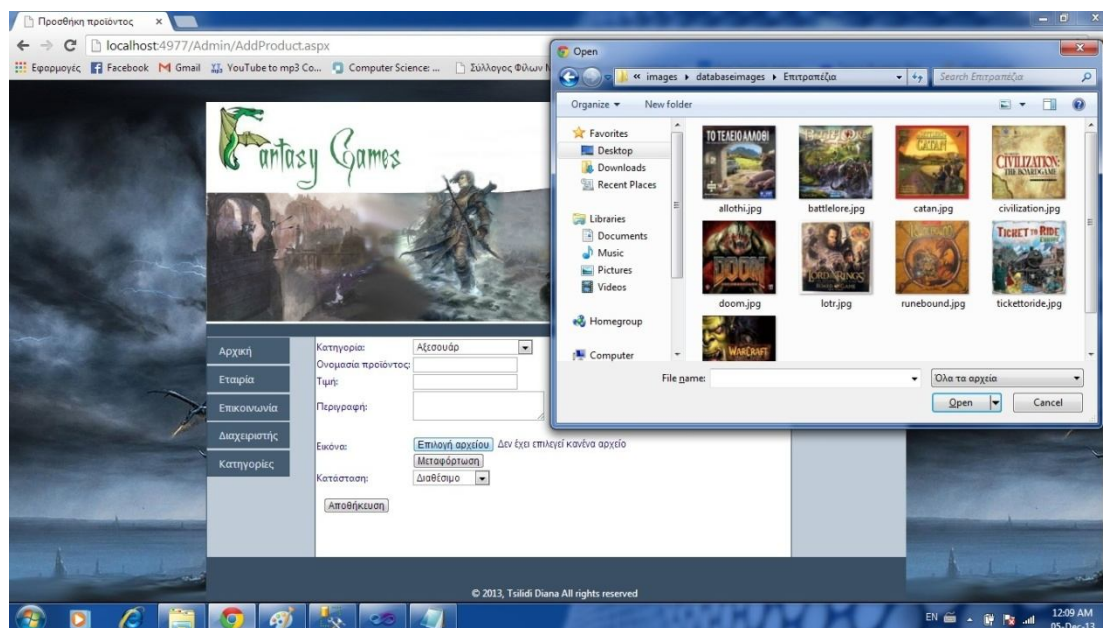
4.8 - Λογαριασμός χρήστη

Ο διαχειριστής, όταν συνδέεται στην ιστοσελίδα, βλέπει λίγο διαφορετικό μενού από ο χρήστη και τον επισκέπτη. Προστίθεται ένα πεδίο διαχειριστή που του προσφέρει επιπλέον δυνατότητες επεξεργασίας στοιχείων προϊόντων, κατηγοριών, χρηστών και παραγγελιών.



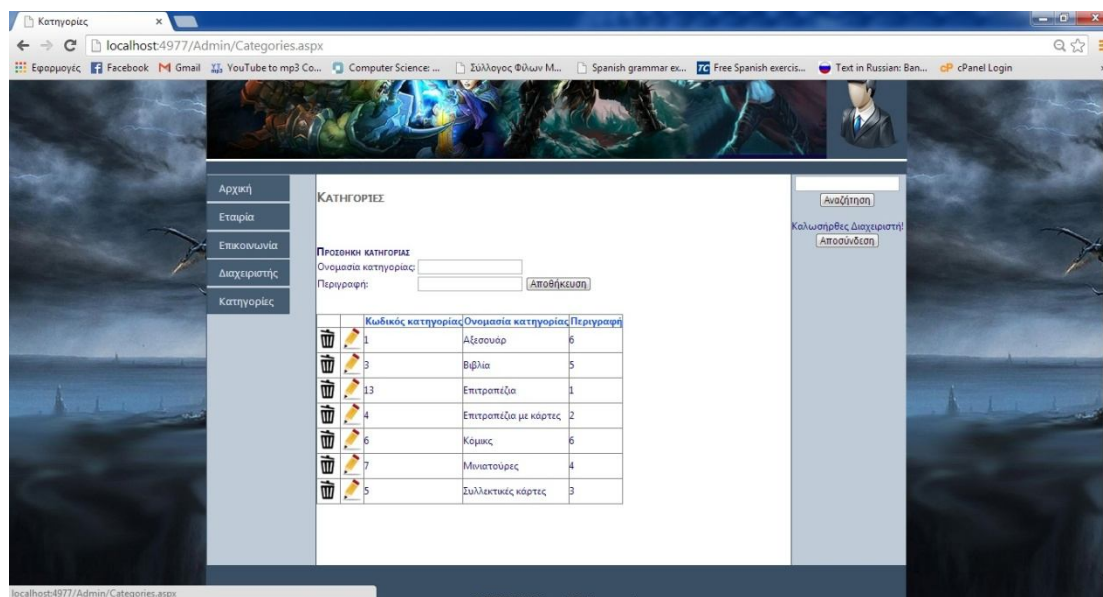
4.9 - Μενού διαχειριστή

Στη σελίδα με τα προϊόντα μπορεί να προσθέσει ένα καινούργιο προϊόν επιλέγοντας την κατηγορία στην οποία ανήκει, συμπληρώνοντας την ονομασία, την τιμή και την περιγραφή και επιλέγοντας κάποια εικόνα του προϊόντος απο το αρχείο, καθώς και την κατάστασή του. Η κατάσταση του προϊόντος ενημερώνει σχετικά με τη διαθεσιμότητά του και πληροφορεί αν είναι νέο προϊόν και αν είναι σε προσφορά.



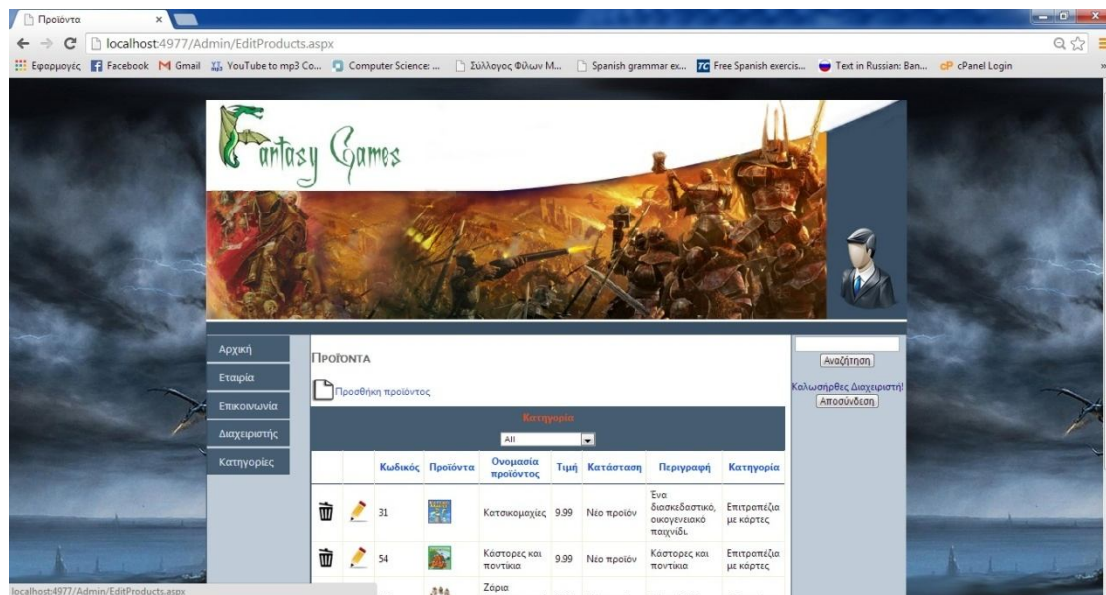
4.10 - Προσθήκη προϊόντων

Ο διαχειριστής μπορεί, επίσης, να επεξεργαστεί τις κατηγορίες των προϊόντων, να καθορίσει τη σειρά προτεραιότητας, με την οποία θα προβάλλονται στο μενού, καθώς και να προσθέσει ή να διαγράψει μια κατηγορία.



4.11 - Επεξεργασία κατηγοριών

Στις δυνατότητες του διαχειριστή υπάρχει επίσης η επεξεργασία των προϊόντων. Στην αντίστοιχη σελίδα ο διαχειριστής μπορεί να αλλάξει την ονομασία ενός προϊόντος, την τιμή του και την περιγραφή του. Επίσης, μπορεί να επιλέξει κάποια διαφορετική κατάσταση του προϊόντος, ανάλογα με τις προσφορές που έχει το κατάστημα ή τη διαθεσιμότητα του εμπορεύματος. Όπως φαίνεται και στην παρακάτω εικόνα, ο διαχειριστής βλέπει όλα τα προϊόντα στη σελίδα αυτή, όμως μπορεί να επιλέξει να δει κάποια συγκεκριμένη κατηγορία προϊόντων, αν θελήσει.

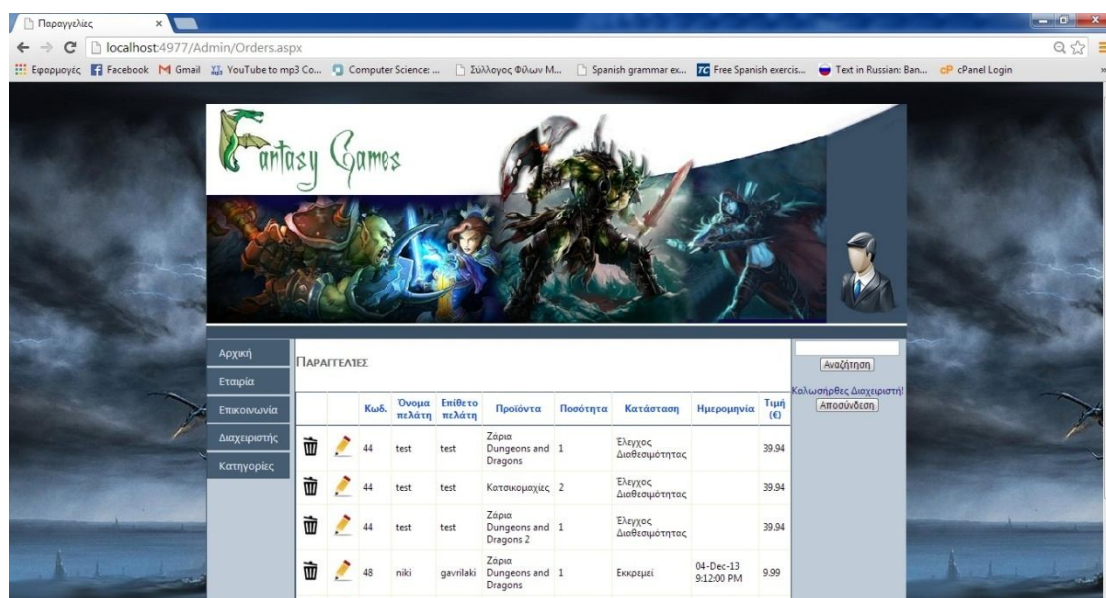


4.12 - Επεξεργασία προϊόντων

Το υπομενού “Παραγγελίες” οδηγεί τον διαχειριστή στη σελίδα στην οποία έχει τη δυνατότητα να επεξεργαστεί τις υπάρχουσες παραγγελίες των χρηστών. Μπορεί να αλλάξει την κατάσταση των παραγγελιών ανάλογα με το στάδιο στο οποίο βρίσκονται. Η κατάσταση μιας παραγγελίας μπορεί να είναι:

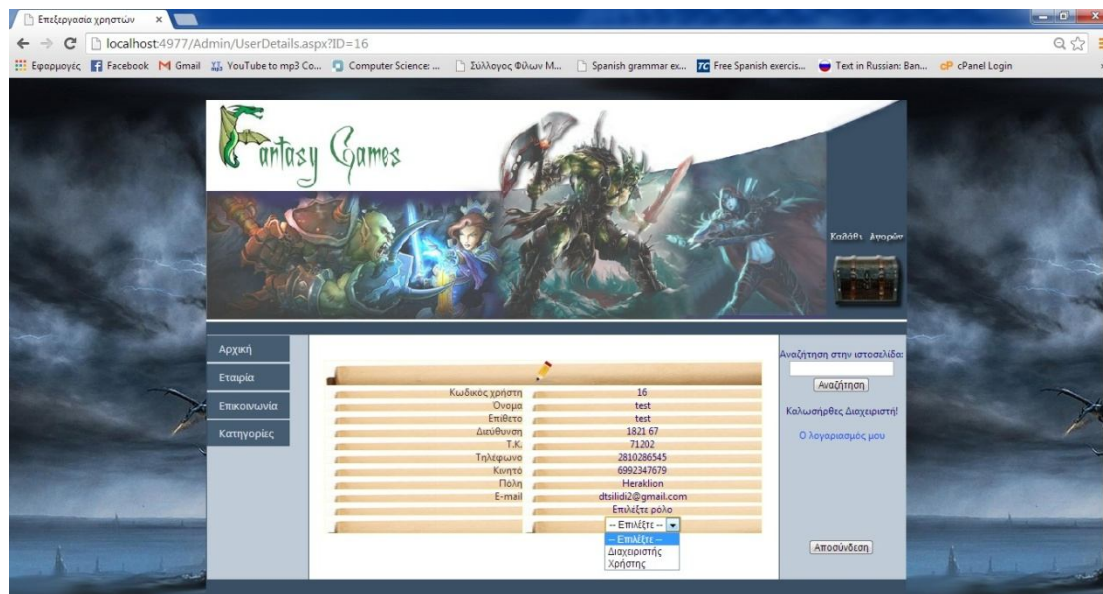
- Ακυρωμένη
- Αναμονή προϊόντων
- Έλεγχος Διαθεσιμότητας
- Έχει αποσταλεί
- Κλειστή
- Μη ελεγμένη
- Στο κατάστημα
- Συναρμολόγηση

Μπορεί, επίσης, να διαγράψει την παραγγελία μετά που θα ολοκληρωθεί η παραλαβή της.



4.13 - Επεξεργασία παραγγελιών

Ως διαχειριστής, μπορεί κανείς να επεξεργαστεί τα στοιχεία των χρηστών. Φυσικά, και ο διαχειριστής δεν έχει πρόσβαση στα στοιχεία σύνδεσης, διεύθυνση ηλεκτρονικού ταχυδρομείου (email) και κωδικό πρόσβασης (password), στοιχεία που χρησιμοποιεί ο χρήστης για να συνδεθεί στο λογαριασμό του. Μπορεί, όμως, σε περίπτωση που χρειάζεται κάτι τέτοιο, να αλλάξει τα υπόλοιπα στοιχεία του χρήστη, όπως η διεύθυνση και το τηλέφωνο.



4.14 - Επεξεργασία στοιχείων χρηστών

4.2 Οι κλάσεις

Η κλάση ShoppingCart

Η κλάση ShoppingCart υλοποιεί το καλάθι αγορών του ηλεκτρονικού καταστήματος. Διαθέτει κατάλληλες μεθόδους, ώστε να μπορεί να δημιουργεί το καλάθι, να προσθέτει προϊόντα, να υπολογίζει το σύνολο και να διαγράφει τα προϊόντα, όποτε χρειάζεται.

```
Imports Microsoft.VisualBasic
```

```
Public Class ShoppingCart
```

```
    '#Region "Properties"
```

```
    Private _items As List(Of CartItem)
```

```
    Public Property Items() As List(Of CartItem)
```

```
        Get
```

```
            Return _items
```

```
        End Get
```

```
        Private Set(ByVal value As List(Of CartItem))
```

```
            _items = value
```

```
        End Set
```

```
    End Property
```

Κατασκευή ηλεκτρονικού καταστήματος

```
'#End Region

'#Region "Singleton Implementation"

' Readonly variables can only be set in initialization or
in a constructor
Public Shared ReadOnly Instance As ShoppingCart
' The static constructor is called as soon as the class is
loaded into memory
Shared Sub New()
    ' If the cart is not in the session, create one and
put it there
    ' Otherwise, get it from the session
    If HttpContext.Current.Session("ShoppingCart") Is
Nothing Then
        Instance = New ShoppingCart()
        Instance.Items = New List(Of CartItem)
        HttpContext.Current.Session("ASPNETShoppingCart")
= Instance
    Else
        Instance =
CType(HttpContext.Current.Session("ASPNETShoppingCart"),
ShoppingCart)
    End If

End Sub

' A protected constructor ensures that an object can't be
created from outside
Protected Sub New()

End Sub

'#End Region

'#Region "Item Modification Methods"

' AddItem() - Adds an item to the shopping
Public Sub AddItem(ByVal product_id As Integer)
    ' Create a new item to add to the cart
    Dim newItem = New CartItem(product_id)

    ' If this item already exists in our list of items,
increase the quantity
    ' Otherwise, add the new item to the list
    If Items.Contains(newItem) Then
        For Each item As CartItem In Items
            If item.Equals(newItem) Then
                item.Quantity += 1
                Return
            End If
        Next
    Else
        newItem.Quantity = 1
        Items.Add(newItem)
    End If
```



```

End Sub

' SetItemQuantity() - Changes the quantity of an item in
the cart
Public Sub SetItemQuantity(ByVal product_id As Integer,
ByVal quantity As Integer)
    ' If we are setting the quantity to 0, remove the item
entirely
    If quantity = 0 Then
        RemoveItem(product_id)
        Return
    End If

    ' Find the item and update the quantity
    Dim updatedItem = New CartItem(product_id)
    For Each item As CartItem In Items
        If item.Equals(updatedItem) Then
            item.Quantity = quantity
            Return
        End If
    Next
End Sub

' RemoveItem() - Removes an item from the shopping cart
Public Sub RemoveItem(ByVal product_id As Integer)
    Dim removedItem = New CartItem(product_id)

    For Each item As CartItem In Items
        If item.Equals(removedItem) Then
            Items.Remove(item)
            Return
        End If
    Next
End Sub

'#End Region

'#Region "Reporting Methods"

' GetSubTotal() - returns the total price of all of the
items before tax, shipping, etc.
Public Function GetSubTotal() As Decimal
    Dim subTotal As Decimal = 0
    For Each item As CartItem In Items
        subTotal += item.TotalPrice
    Next

    Return subTotal
End Function
End Class

```

Η κλάση Product

Η κλάση Products περιγράφει τις ιδιότητες ενός προϊόντος. Ορίζει τον κωδικό του προϊόντος, την ονομασία του, την τιμή του και την εικόνα του. Αφού ορίσει τα στοιχεία αυτά, η κλάση Products επικοινωνεί με τη βάση δεδομένων και αποθηκεύει τα πεδία ονομασία, τιμή και εικόνα του προϊόντος σε προσωρινές μεταβλητές για να τα προβάλει στο καλάθι αγορών του χρήστη.

```
Imports Microsoft.VisualBasic
Imports System.Data.SqlClient

Public Class product
    '#Region "Properties"

    Private _id As Integer
    Public Property Id() As Integer
        Get
            Return _id
        End Get
        Set(ByVal value As Integer)
            _id = value
        End Set
    End Property

    Private _title As String
    Public Property Title() As String
        Get
            Return _title
        End Get
        Set(ByVal value As String)
            _title = value
        End Set
    End Property

    Private _image As String
    Public Property Image() As String
        Get
            Return _image
        End Get
        Set(ByVal value As String)
            _image = value
        End Set
    End Property

    Private _price As Decimal
    Public Property Price() As Decimal
        Get
            Return _price
        End Get
        Set(ByVal value As Decimal)
            _price = value
        End Set
    End Property

    '#End Region
```

```

    Public Sub New(ByVal id As Integer)
        Dim con As New
SqlConnection(ConfigurationManager.ConnectionStrings("BoardGamesConnectionString").ConnectionString)
        Dim cmd As New SqlCommand
        Dim dr As SqlDataReader
        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.CommandText = "Product"
        cmd.Parameters.AddWithValue("@product_id", id)
        cmd.Connection = con
        Try
            con.Open()
            dr = cmd.ExecuteReader
            If dr.Read() Then
                Me.Price = dr("product_price")
                Me.Title = dr("product_title")
                Me.Image = dr("product_image")
            End If
            dr.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
        Finally
            cmd.Dispose()
            con.Close()
        End Try
    End Sub
End Class

```

Η κλάση CartItem

Η κλάση CartItem επιστρέφει τα πεδία ονομασία, τιμή και εικόνα του προϊόντος, καθώς και την ποσότητα που έχει επιλέξει ο χρήστης.

```

Imports Microsoft.VisualBasic
Public Class CartItem
    Implements IEquatable(Of CartItem)

    '#Region "Properties"

    ' A place to store the quantity in the cart
    Private _quantity As Integer
    Public Property Quantity() As Integer
        Get
            Return _quantity
        End Get
        Set(ByVal value As Integer)
            _quantity = value
        End Set
    End Property

    Private _product_id As Integer
    Public Property product_id() As Integer
        Get
            Return _product_id
        End Get
    End Property

```

Κατασκευή ηλεκτρονικού καταστήματος

```
        End Get
        Set(ByVal value As Integer)
            ' To ensure that the Prod object will be re-
created
            _product = Nothing
            _product_id = value
        End Set
    End Property

    Private _product As product = Nothing
    Public ReadOnly Property product() As product
        Get
            ' Lazy initialization - the object won't be
created until it is needed
            If _product Is Nothing Then
                _product = New product(product_id)
            End If
            Return _product
        End Get
    End Property
    Public ReadOnly Property UnitTitle() As String
        Get
            Return product.Title
        End Get
    End Property

    Public ReadOnly Property UnitImage() As String
        Get
            Return product.Image
        End Get
    End Property

    Public ReadOnly Property UnitPrice() As Decimal
        Get
            Return product.Price
        End Get
    End Property

    Public ReadOnly Property TotalPrice() As Decimal
        Get
            Return UnitPrice * Quantity
        End Get
    End Property

'#End Region

' CartItem constructor just needs a productId
Public Sub New(ByVal product_id As Integer)
    Me.product_id = product_id
End Sub

' Equals() - Needed to implement the IEquatable interface
' Tests whether or not this item is equal to the parameter
' This method is called by the Contains() method in the
List class
```

```

' We used this Contains() method in the ShoppingCart
AddItem() method
Public Overloads Function Equals(ByVal item As CartItem)
As Boolean Implements IEquatable(Of CartItem).Equals
    Return item.product_id = Me.product_id
End Function
End Class

```

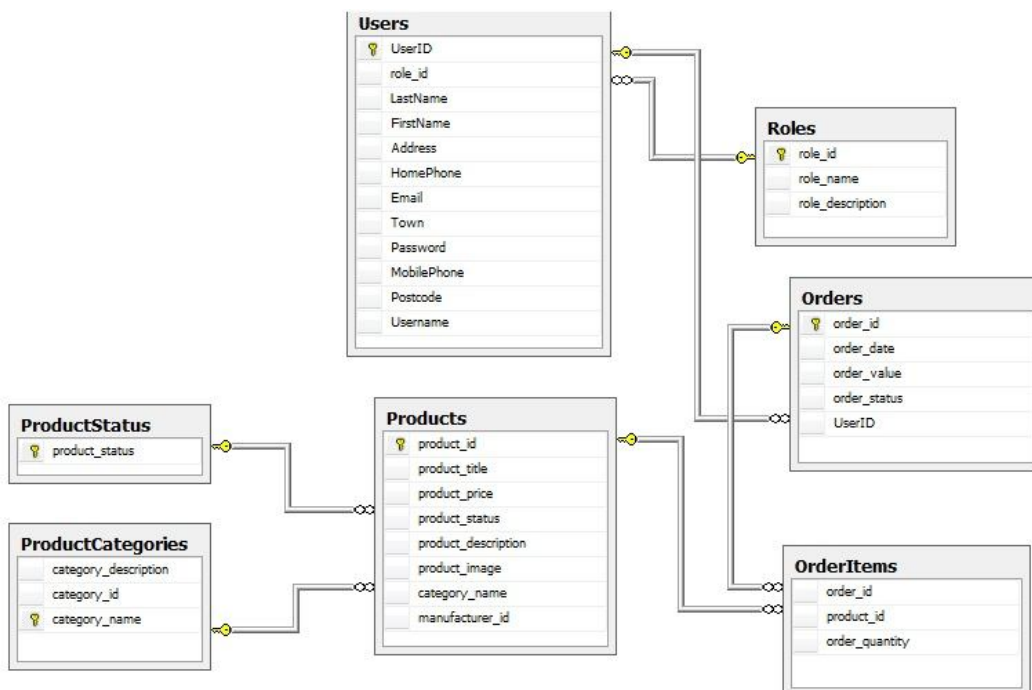
4.3 Η βάση δεδομένων

Στην παρακάτω εικόνα φαίνεται το σχεδιάγραμμα της βάσης δεδομένων Board Games με τους πίνακές της και τις σχέσεις μεταξύ των πινάκων.

Ο πίνακας Users περιλαμβάνει τα στοιχεία των χρηστών: κωδικός χρήστη, κωδικός ρόλου του χρήστη, όνομα, επίθετο, διεύθυνση, σταθερό τηλέφωνο, διεύθυνση ηλεκτρονικού ταχυδρομείου, πόλη διαμονής, κωδικός πρόσβασης, κινητό τηλέφωνο, ταχυδρομικός κώδικας και όνομα χρήστη. Οι ρόλοι των χρηστών φαίνονται στον πίνακα Roles που περιλαμβάνει τους κωδικούς των ρόλων, τις ονομασίες των ρόλων και την περιγραφή τους.

Ο πίνακας Products περιέχει τα στοιχεία των προϊόντων, δηλαδή τον κωδικό του προϊόντος, την ονομασία, την τιμή, την κατάσταση (διαθέσιμο, σε προσφορά κ.λπ.), την περιγραφή, την εικόνα, το όνομα της κατηγορία και τον κωδικό του κατασκευαστή. Οι καταστάσεις των προϊόντων καθορίζονται από τον πίνακα ProductStatus, που περιέχει την ονομασία των καταστάσεων. Ενώ η κατηγορία στην οποία ανήκει ένα προϊόν παίρνει στοιχεία από τον πίνακα ProductCategories, που περιλαμβάνει τον κωδικό της κατηγορίας, την ονομασία και την περιγραφή της.

Ο πίνακας Orders ορίζει τα στοιχεία της παραγγελίας που κάνει ο χρήστης. Έχει τα πεδία κωδικός παραγγελίας, ημερομηνία, συνολική αξία, κατάσταση (Ακυρωμένη, Μη ελεγμένη κ.λπ.) και κωδικός του χρήστη που έχει κάνει τη συγκεκριμένη παραγγελία. Αντίστοιχα, τα προϊόντα που ανήκουν στην παραγγελία αποθηκεύονται στον πίνακα OrderItems, που περιλαμβάνει τα στοιχεία κωδικός παραγγελίας, κωδικός προϊόντος και ποσότητα του συγκεκριμένου προϊόντος που έχει επιλεγεί από το χρήστη.



4.15 - Βάση δεδομένων

4.3.1. Stored Procedures

Οι αποθηκευμένες διαδικασίες (stored procedures) είναι τμήματα προγράμματος βάσης δεδομένων που αποθηκεύονται μόνιμα και εκτελούνται στον διακομιστή της βάσης δεδομένων. Αποθηκεύονται και τρέχουν στον διακομιστή της βάσης και καλούνται με το όνομα τους από οποιαδήποτε εφαρμογή συνδεθεί με τη βάση. Η εφαρμογή μπορεί να χωριστεί σε δύο μέρη, η εφαρμογή τρέχει στον πελάτη, ενώ η αποθηκευμένη διαδικασία στον διακομιστή. Περιέχει προγραμματιστικές εντολές (if, loops) αλλά ταυτόχρονα μπορεί να χειριστεί SQL κώδικα. Οι αποθηκευμένες διαδικασίες μπορούν τα έχουν την παρακάτω χρήση:

- Για έλεγχο πολύπλοκων περιορισμών.
- Για απόκρυψη του κώδικα της διαδικασίας.
- Για δημιουργία βιβλιοθήκης έτοιμων συναρτήσεων στον διακομιστή.

Πλεονεκτήματα των αποθηκευμένων διαδικασιών:

- Βελτιώνουν την ασφάλεια του DBMS. Υποστηρίζουν το «κρύψιμο» (encapsulation) του κώδικα τους, και με σωστό σχεδιασμό και συστηματική χρήση τους είναι δυνατόν να εξαφανιστεί η δομή της βάσης από τους χρήστες.
- Βελτιώνει την ταχύτητα απόκρισης του εξυπηρετητή της Βάσης. Αυτό οφείλεται στο ότι είναι προμεταγλωττισμένες (precompiled) και έτσι εκτελούνται γρηγορότερα.
- Επίσης, αντικαθιστώντας ένα σύνολο SQL εντολών με την κλήση μιας συνάρτησης που περιέχει όλες αυτές τις SQL εντολές, «ελαφρύνει» την κίνηση στο δίκτυο μεταξύ πελάτη και διακομιστή.

Η δήλωση της περιέχει κώδικα SQL και αποτελείται από το όνομα της αποθηκευμένης διαδικασίας, τις παραμέτρους που θα πάρει, τον τύπο της παραμέτρου, το είδος της παραμέτρου (IN, OUT, INOUT) και τη γλώσσα στην οποία είναι γραμμένη, π.χ. SQL.

Μερικά παραδείγματα Stored Procedure από τη βάση Board Games:

Η διαδικασία CreateUser χρησιμοποιείται για να δημιουργηθεί μια καινούργια εγγραφή στον πίνακα Users, που έχει αποθηκευμένα τα στοιχεία των εγγεγραμμένων χρηστών.

```
USE [Board Games]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[CreateUser]

    @FirstName          nvarchar (256) ,
    @LastName           nvarchar (256) ,
    @UserName           nvarchar (256) ,
    @Password           nvarchar (256) ,
    @Address            nvarchar (256) ,
    @Town              nvarchar (256) ,
    @PostCode          nvarchar (256) ,
    @HomePhone         nvarchar (256) ,
    @MobilePhone       nvarchar (256)
```

```
AS
BEGIN

    INSERT dbo.Users (FirstName, LastName, UserName, Password,
Address, Town, PostCode, HomePhone, MobilePhone, role_id)
    VALUES (@FirstName, @LastName, @UserName, @Password,
@Address, @Town, @PostCode, @HomePhone, @MobilePhone, 2)

END
```

Η διαδικασία UpdateUser χρησιμοποιείται για να ανανεωθεί μια εγγραφή στον πίνακα Users, που έχει αποθηκευμένα τα στοιχεία του επιλεγμένου χρήστη.

```
USE [Board Games]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[UpdateUser]

    @FirstName          nvarchar(256),
    @LastName           nvarchar(256),
    @UserName            nvarchar(256),
    @Password           nvarchar(256),
    @Address            nvarchar(256),
    @Town              nvarchar(256),
    @PostCode          nvarchar(256),
    @Email             nvarchar(256),
    @HomePhone         nvarchar(256),
    @MobilePhone       nvarchar(256),
    @UserID            int
```

```
AS

    UPDATE Users
SET FirstName = @FirstName, LastName = @LastName, [Password] =
@Password, Email =@Email, [Address] = @Address, Town = @Town,
PostCode = @PostCode, HomePhone = @HomePhone, MobilePhone =
@MobilePhone
WHERE UserID = @UserID
SELECT * FROM Users WHERE UserID = @UserID
```

Η διαδικασία Order χρησιμοποιείται για να δημιουργηθεί μια καινούργια εγγραφή στον πίνακα Orders, που έχει αποθηκευμένες τις παραγγελίες των επιλεγμένων χρηστών.

```
USE [Board Games]
GO
SET ANSI_NULLS ON
GO
```

Κατασκευή ηλεκτρονικού καταστήματος

```
SET QUOTED_IDENTIFIER ON
GO
```

```
ALTER PROCEDURE [dbo].[Order]
```

```
    @userid int,
    @ordervalue nvarchar(max),
    @date nvarchar(max),
    @orderstatus nvarchar(max)
```

```
AS
```

```
    insert into Orders
        ([UserID], [order_value], [order_date], [order_status] )
values
    (@userid, @ordervalue, @date, @orderstatus)
select Orders.order_id
from Orders
where UserID=@userid and order_value=@ordervalue
```

Η διαδικασία Order_Items χρησιμοποιείται για να δημιουργηθεί μια καινούργια εγγραφή στον πίνακα OrderItems, που έχει αποθηκευμένα τα προϊόντα των παραγγελιών, τους κωδικούς των παραγγελιών και την ποσότητα των συγκεκριμένων προϊόντων.

```
USE [Board Games]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
ALTER PROCEDURE [dbo].[Order_Items]
```

```
    @orderid int,
    @productid nvarchar(max),
    @quantity int
```

```
AS
```

```
    insert into OrderItems
        ([order_id], [product_id], [order_quantity]) values
    (@orderid, @productid, @quantity)
```


Βιβλιογραφία

- [1] Makoto Yokoo, Takayuki Ito, Minjie Zhang, Juhnyoung Lee, Tokuro Matsuo, “*Electronic Commerce: Theory and Practice*”
- [2] “*Visual Basic .net Προγραμματισμός Εισαγωγή στο .NET και τα web services Pro ASP.NET 2.0 Website Programming*”
- [3] Jason N. Gaylord, Christian Wenz, Pranav Rastogi, Todd Miranda, Scott Hanselman “*Professional ASP.NET 4.5 in C# and VB*”
- [4] Jeremy Kasher, Tim Kent, Martyn Bullerwell, “*Microsoft SQL Server 2008 R2 Master Data Services*”
- [6] McGraw Hill Professional, 14 Οκτ 2010, “*Practical SQL Queries for Microsoft SQL Server 2008 R2*”
- [7] Philo Janus, Stacia Misner, McGraw Hill Professional, 3 Φεβ 2011, “*Building Integrated Business Intelligence Solutions with SQL Server 2008 R2 & Office 2010*”
- [8] Matthew Shepker, “*Writing Stored Procedures with Microsoft SQL Server*”
- [9] Meyer, Eric A. (2006). “*Cascading Style Sheets: The Definitive Guide, Third Edition.*” O'Reilly Media, Inc. ISBN 0-596-52733-0.
- [10] Håkon Wium Lie and Bert Bos, “*Cascading Style Sheets: Designing for the Web* (2005)”
- [11] <http://en.wikipedia.org/wiki/E-commerce>
- [12] <http://msdn.microsoft.com/en-us/library/ms174173.aspx>
- [13] <http://msdn.microsoft.com/en-us/library/ms166104.aspx>
- [14] http://www.cs.uoi.gr/~pitoura/courses/db/MySQL_stored_procedures.html
- [15] http://en.wikipedia.org/wiki/SQL_Server_Management_Studio_Express
- [16] http://el.wikipedia.org/wiki/Sql_server
- [17] <http://support.microsoft.com/kb/306590/el>
- [18] http://el.wikipedia.org/wiki/HTTP_cookies
- [19] http://www.bloggertips.gr/2011/03/blog-post_05.html
- [20] <http://www.codeproject.com/KB/aspnet/aspnetintro.aspx>
- [21] <http://msdn.microsoft.com/en-us/library/wtxbf3hh.aspx>

Κατασκευή ηλεκτρονικού καταστήματος

[22] http://en.wikipedia.org/wiki/Cascading_Style_Sheets#Use

[23] http://el.wikipedia.org/wiki/Visual_Basic

[24] [http://msdn.microsoft.com/en-us/library/yedba920\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/yedba920(v=vs.71).aspx)

ΠΑΡΑΡΤΗΜΑ Α

Κύρια μέρη της εφαρμογής

Η σελίδα ViewCart.aspx υλοποιεί το καλάθι αγορών. Περιλαμβάνει έναν πίνακα, στον οποίο προβάλλει τα προϊόντα που έχει επιλέξει ο χρήστης να προστεθούν στο καλάθι. Εμφανίζεται η εικόνα του προϊόντος, η ονομασία, η τιμή και η περιγραφή του. Επίσης, ο χρήστης μπορεί να επιλέξει την ποσότητα του συγκεκριμένου προϊόντος και να δει το υποσύνολο μετά την αλλαγή της ποσότητας και τη συνολική αξία των προϊόντων. Επιπλέον, γίνεται έλεγχος αν είναι συνδεδεμένος ο χρήστης, ώστε αν αποσυνδεθεί ο χρήστης, να αποσυνδέεται από το καλάθι και να διαγράφονται τα προϊόντα που έχουν επιλεγεί, εφόσον ο χρήστης δεν είχε προχωρήσει σε αγορά και παραγγελία.

```
<%@ Page Title="Καλάθι αγορών" Language="vb"
AutoEventWireup="false" EnableEventValidation="false"
MasterPageFile="~/Site.Master" CodeBehind="ViewCart.aspx.vb"
Inherits="eshop.ViewCart" %>
<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent"
runat="server">

    <br /> <object
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=8,5,0,175"
    classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
        style="width: 610px; height: 150px;">
        <param value="images\header2.swf" name="movie"/>
        <param value="undefined" name="FlashVars"/>
        <param value="sameDomain"
name="allowScriptAccess"/>
        <param value="high" name="quality"/>
        <param value="false" name="menu"/>
        <param value="transparent" name="wmode"/>
        <embed width="610" height="150"
pluginspage="http://www.macromedia.com/go/getflashplayer"
type="application/x-shockwave-flash"
wmode="transparent" menu="false" quality="high"
allowsriptaccess="sameDomain"
flashvars="undefined" src="..\images\scroll.swf"
align="center"></object>

    <asp:SqlDataSource ID="sqlCart" runat="server"
ConnectionString="<%$
ConnectionString:BoardGamesConnectionString %>"
SelectCommand="Addproduct"
SelectCommandType="StoredProcedure">
    <SelectParameters>
        <asp:Parameter Name="product_id" Type="Int32" />
        <asp:Parameter Name="product_title" Type="String" />
        <asp:Parameter Name="product_description"
Type="String" />
```

Κατασκευή ηλεκτρονικού καταστήματος

```
<asp:Parameter Name="product_price" Type="Decimal" />
<asp:Parameter Name="product_image" Type="String" />
</SelectParameters>
</asp:SqlDataSource>
<div class="container">

    <asp:GridView runat="server"
ID="gvShoppingCart" AutoGenerateColumns="false"
EmptyDataText="Το καλάθι σας είναι άδειο." GridLines="None"
Width="100%" CellPadding="15" ShowFooter="true"
DataKeyNames="product_id"
OnRowDataBound="gvShoppingCart_RowDataBound"
OnRowCommand="gvShoppingCart_RowCommand">
    <HeaderStyle HorizontalAlign="Left"
BackColor="#3D7169" ForeColor="#FFFFFF" />
    <FooterStyle HorizontalAlign="Right"
BackColor="#6C6B66" ForeColor="#FFFFFF" />
    <AlternatingRowStyle BackColor="#F8F8F8"
/>

    <Columns>

        <asp:ImageField
DataImageUrlField="UnitImage" HeaderText="ΠΡΟΪΟΝΤΑ" ItemStyle-
HorizontalAlign="Right" HeaderStyle-HorizontalAlign="Right"
/>

        <asp:BoundField DataField="UnitTitle"
HeaderText="" ItemStyle-HorizontalAlign="Right" HeaderStyle-
HorizontalAlign="Right" DataFormatString="{0}" />
        <asp:TemplateField HeaderText=" ">
            <ItemTemplate>

                <asp:HyperLink
ID="HyperLink1" runat="server" NavigateUrl='<%#
Eval("product_id", "~/ProductidDetails.aspx?ID={0}") %>'
Text='<%# Eval("product_id", "Λεπτομέρειες..")
%>'></asp:HyperLink><br />

            </ItemTemplate>
        </asp:TemplateField>
        <asp:BoundField DataField="UnitPrice"
HeaderText="ΤΙΜΗ" ItemStyle-HorizontalAlign="Right"
HeaderStyle-HorizontalAlign="Right" DataFormatString="&#8364;
{0:N}" />
        <asp:TemplateField
HeaderText="ΠΟΣΟΤΗΤΑ">
            <ItemTemplate>
                <asp:TextBox
runat="server" ID="txtQuantity" Columns="5" Text='<%#
Eval("Quantity") %>'></asp:TextBox><br />
                <asp:LinkButton runat="server"
ID="btnRemove" Text="Διαγραφή" CommandName="Remove"
CommandArgument='<%# Eval("product_id") %>' style="font-
size:12px;"></asp:LinkButton>
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
</div>
```

```

                </ItemTemplate>
            </asp:TemplateField>
            <asp:BoundField DataField="TotalPrice"
HeaderText="ΣΥΝΟΛΟ" ItemStyle-HorizontalAlign="Right"
HeaderStyle-HorizontalAlign="Right" DataFormatString="{0:N}";
{0:N}" />
        </Columns>
    </asp:GridView>
    <br />
    <asp:Button runat="server" ID="btnUpdateCart"
Text="Ανανέωση" OnClick="btnUpdateCart_Click" />
    <br />
    <asp:Button runat="server" ID="btnbuy"
Text="Αγορά" />

</div>

</asp:Content>

```

ViewCart.vb

```

Public Class ViewCart
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        If Session("UserID") <= 0 Then
            Response.Redirect("~/Account/login.aspx")
        End If

        ' ASP.NET keeps track of the state of all server
controls
        ' (like the GridView). Because of this, we don't need
to supply it with
        ' the data every time the page loads. (No more re-
checking checkboxes
        ' that were checked or re-selecting the right item in
the drop-down when
        ' the page is submitted. It's all done for you!)

        If Not IsPostBack Then
            BindData()
        End If
    End Sub

    Protected Sub BindData()
        ' Let's give the data to the GridView and let it work!
        ' The GridView will take our cart items one by one and
use the properties
        ' that we declared as column names (DataFields)
        gvShoppingCart.DataSource =
ShoppingCart.Instance.Items
        gvShoppingCart.DataBind()
    End Sub
End Class

```

Κατασκευή ηλεκτρονικού καταστήματος

```
End Sub

Protected Sub gvShoppingCart_RowDataBound(ByVal sender As Object, ByVal e As GridViewRowEventArgs) Handles gvShoppingCart.RowDataBound
    If e.Row.RowType = DataControlRowType.Footer Then
        e.Row.Cells(2).Text = "Αξία Προϊόντων: " &
            ShoppingCart.Instance.GetSubTotal().ToString("N")
        & "€"
    End If
End Sub

' This is the method that responds to the Remove button's click event
Protected Sub gvShoppingCart_RowCommand(ByVal sender As Object, ByVal e As GridViewCommandEventArgs) Handles gvShoppingCart.RowCommand
    If e.CommandName = "Remove" Then
        Dim product_id =
            Convert.ToInt32(e.CommandArgument)
        ShoppingCart.Instance.RemoveItem(product_id)
    End If

    ' We now have to re-setup the data so that the GridView doesn't keep displaying the old data
    BindData()
End Sub

Protected Sub btnUpdateCart_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnUpdateCart.Click
    For Each row As GridViewRow In gvShoppingCart.Rows
        If row.RowType = DataControlRowType.DataRow Then
            ' We'll use a try catch block in case something other than a number is typed in. If so, we'll just ignore it.
            Try
                ' Get the productId from the GridView's datakeys
                Dim product_id =
                    Convert.ToInt32(gvShoppingCart.DataKeys(row.RowIndex).Value)
                ' Find the quantity TextBox and retrieve the value
                Dim quantity =
                    Integer.Parse(CType(row.Cells(1).FindControl("txtQuantity"),
                    TextBox).Text)

                ShoppingCart.Instance.SetItemQuantity(product_id, quantity)
            Catch ex As FormatException
            End Try
        End If
    Next

    BindData()
End Sub
```

```

        Protected Sub btnbuy_Click(ByVal sender As Object, ByVal e
As EventArgs) Handles btnbuy.Click
            For Each row As GridViewRow In gvShoppingCart.Rows
                If gvShoppingCart.DataKeys(row.RowIndex).Value > 0
Then
                    Response.Redirect("~/Buy.aspx", True)
                Else
                End If
            Next
        End Sub
    End Class

```

Η σελίδα Buy.aspx προβάλλει τα προϊόντα του καλαθιού που έχουν επιλεγεί για αγορά. Τα πεδία που εμφανίζονται στον πίνακα είναι η εικόνα του προϊόντος, η ονομασία του και η τιμή του. Υπάρχουν, ακόμα, ένας τα στοιχεία του χρήστη, καθώς και το κουμπί επαλήθευσης αγοράς που οδηγεί στην παραγγελία.

```

<%@ Page Title="Επιβεβαίωση αγοράς" Language="vb"
AutoEventWireup="false" MasterPageFile="~/Site.Master"
CodeBehind="Buy.aspx.vb" Inherits="eshop.Buy" %>
<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent"
runat="server">
<br /> <object
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=8,5,0,175"
classid="clsid:d27cdb6e-ae6d-11cf-96b8-
444553540000"
style="width: 610px; height: 150px;">
    <param value="images\header2.swf" name="movie"/>
    <param value="undefined" name="FlashVars"/>
    <param value="sameDomain"
name="allowScriptAccess"/>
    <param value="high" name="quality"/>
    <param value="false" name="menu"/>
    <param value="transparent" name="wmode"/>
    <embed width="610" height="150"
pluginspage="http://www.macromedia.com/go/getflashplayer"
type="application/x-shockwave-flash"
wmode="transparent" menu="false" quality="high"
allowscriptaccess="sameDomain"
flashvars="undefined" src="..\images\scroll2.swf"></object>

<p>
    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%"$
ConnectionString:connection_userInfo %>"
SelectCommand="UserInfo"
SelectCommandType="StoredProcedure">
        <SelectParameters>
            <asp:Parameter Name="UserID" Type="String" />
        </SelectParameters>

```

```

</asp:SqlDataSource>

<asp:SqlDataSource ID="sqlCart" runat="server"
ConnectionString="<%$
ConnectionStrings:BoardGamesConnectionString %>"
SelectCommand="Addproduct"
SelectCommandType="StoredProcedure">
    <SelectParameters>
        <asp:Parameter Name="product_id" Type="Int32" />
        <asp:Parameter Name="product_title" Type="String" />
        <asp:Parameter Name="product_description"
Type="String" />
        <asp:Parameter Name="product_price" Type="Decimal" />
        <asp:Parameter Name="product_image" Type="String" />
    </SelectParameters>
</asp:SqlDataSource>

<p align = "center"> Στοιχεία πελάτη</p>

    <asp:GridView ID="GridView1" runat="server"
GridLines="None" Width="100%" CellPadding="5"
ShowFooter="true" ShowHeader="true" >
        <HeaderStyle HorizontalAlign="Left"
BackColor="#3D7169" ForeColor="#FFFFFF" />
        <FooterStyle HorizontalAlign="Right"
BackColor="#6C6B66" ForeColor="#FFFFFF" />
        <AlternatingRowStyle BackColor="#F8F8F8"
/>
        <Columns>
        </Columns>
    </asp:GridView>

    <br />
    <br />
    <p align = "center"> Προϊόντα</p>
    <asp:GridView ID="GridView2" runat="server"
AutoGenerateColumns="false" GridLines="None" Width="100%"
CellPadding="5" ShowFooter="true" DataKeyNames="product_id" >
        <HeaderStyle HorizontalAlign="Left"
BackColor="#3D7169" ForeColor="#FFFFFF" />
        <FooterStyle HorizontalAlign="Right"
BackColor="#6C6B66" ForeColor="#FFFFFF" />
        <AlternatingRowStyle BackColor="#F8F8F8"
/>
        <Columns>
            <asp:ImageField
DataImageUrlField="UnitImage" HeaderText="Προϊόντα" ItemStyle-
HorizontalAlign="Right" HeaderStyle-HorizontalAlign="Right"
/>
            <asp:BoundField DataField="UnitTitle"
HeaderText="" ItemStyle-HorizontalAlign="Right" HeaderStyle-
HorizontalAlign="Right" DataFormatString="{0}" />

```



```

                                <asp:TemplateField
HeaderText="Ποσότητα">
                                <ItemTemplate>
                                    <asp:Label
runat="server" ID="txtQuantity" Columns="5" Text='<%#
Eval("Quantity") %>'></asp:Label><br />
                                </ItemTemplate>
                                </asp:TemplateField>
                                <asp:BoundField DataField="UnitPrice"
HeaderText="Τιμή" ItemStyle-HorizontalAlign="Right"
HeaderStyle-HorizontalAlign="Right" DataFormatString="&#8364;
{0:N}" />
                                <asp:BoundField
DataField="TotalPrice" HeaderText="Συνολική τιμή" ItemStyle-
HorizontalAlign="Right" HeaderStyle-HorizontalAlign="Right"
DataFormatString="&#8364; {0:N}" />
                                </Columns>
                                </asp:GridView>
        </p>
        <p align="center">
            <asp:Button ID="Button2" Text ="Επιβεβαίωση αγοράς"
runat="server" />
        </p>
</asp:Content>

```

Buy.vb

```

Imports System.Data.SqlClient

Public Class Buy
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        ' tblinvatbuy.Visible = False
        Try
            If Session("UserID") > 0 Then

                Dim reader As SqlDataReader
                Dim conn As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STR
ING"))

                Dim command As New SqlCommand
                Dim cmd As New SqlCommand
                command.CommandType =
CommandType.StoredProcedure
                cmd.CommandType = CommandType.StoredProcedure
                command.CommandText = "UserInfo"
                command.Connection = conn
                command.Parameters.AddWithValue("@UserID",
Session("UserID"))
                conn.Open()

```

Κατασκευή ηλεκτρονικού καταστήματος

```
        reader = command.ExecuteReader()
        GridView1.DataSource = reader
        GridView1.DataBind()
        reader.Close()
        conn.Close()
        conn.Dispose()
    Else
        Session("MESSAGE") = " Δεν εχετε συνδεθεί."
        Response.Redirect("~/Message.aspx", True)
    End If

    GridView2.DataSource = ShoppingCart.Instance.Items
    GridView2.DataBind()
Catch ex As Exception
    Response.Redirect("~/Account/Login.aspx", True)
End Try

End Sub

Protected Sub GridView2_RowDataBound(ByVal sender As
Object, ByVal e As GridViewRowEventArgs) Handles
GridView2.RowDataBound
    If e.Row.RowType = DataControlRowType.Footer Then
        e.Row.Cells(2).Text = "ΑΕΙΑ ΠΡΟΪΟΝΤΩΝ:" &
        ShoppingCart.Instance.GetSubTotal().ToString("N")
& "€"
    End If
End Sub

Protected Sub Button2_Click(ByVal sender As Object, ByVal
e As EventArgs) Handles Button2.Click
    Dim reader As SqlDataReader
    Dim conn As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STR
ING"))

    Dim status As String
    status = "Εκκρεμεί"
    Dim command As New SqlCommand
    Dim cmd As New SqlCommand
    command.CommandType = CommandType.StoredProcedure
    cmd.CommandType = CommandType.StoredProcedure
    command.CommandText = "Order"
    command.Connection = conn
    command.Parameters.AddWithValue("@userid",
Session("UserID"))
    command.Parameters.AddWithValue("@ordervalue",
ShoppingCart.Instance.GetSubTotal().ToString("N"))
    command.Parameters.AddWithValue("@date", Date.UtcNow)
    command.Parameters.AddWithValue("@orderstatus",
status)

    conn.Open()
    reader = command.ExecuteReader()
    While (reader.Read)
        Dim conn2 As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STR
ING"))
```

```

        For Each row As GridViewRow In GridView2.Rows
            If row.RowType = DataControlRowType.DataRow
Then
                Try
                    Dim productid =
Convert.ToInt32(GridView2.DataKeys(row.RowIndex).Value)
                    Dim quantity =
Integer.Parse(CType(row.Cells(2).FindControl("txtQuantity"),
Label).Text)
                    Dim comm As New SqlCommand
                    comm.CommandType =
CommandType.StoredProcedure
                    comm.CommandText = "Order_Items"
                    comm.Connection = conn2

                    comm.Parameters.AddWithValue("@orderid", reader("order_id"))

                    comm.Parameters.AddWithValue("@productid", productid)

                    comm.Parameters.AddWithValue("@quantity", quantity)
                    conn2.Open()
                    comm.ExecuteNonQuery()
                    conn2.Close()
                    comm.Dispose()
                Catch ex As Exception
                    Session("MESSAGE") = ex.ToString
                    Response.Redirect("~/Message.aspx",
True)
                End Try
            End If
        Next
    End While
    reader.Close()

    conn.Close()
    conn.Dispose()
    Dim i As Integer
    For i = 1 To 100000
        ShoppingCart.Instance.RemoveItem(i)
    Next

    Response.Redirect("~/Order.aspx", True)
End Sub

End Class

```

Η σελίδα Register.aspx επιτρέπει την εγγραφή καινούργιου χρήστη. Ζητάει απο τον χρήστη να συμπληρώσει τα στοιχεία του, τα ελέγχει και τα καταχωρεί, δημιουργώντας έναν νέο λογαριασμό χρήστη.

Κατασκευή ηλεκτρονικού καταστήματος

```
<%@ Page Title="Εγγραφή χρήστη" Language="vb"
AutoEventWireup="false" MasterPageFile="~/Site.Master"
CodeBehind="Register.aspx.vb" Inherits="eshop.Register" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent"
runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent"
runat="server">
<div style="width: 600px; padding-left:40px">
<asp:Panel ID="Panell1" runat="server" Width="680px"
ScrollBars="Vertical" Height="475px"
DefaultButton="btnSignUp">
<div style="width: 405px;">
<fieldset class="login" style="float:left;">
<legend><b>Εγγραφή</b></legend>
<p>
<asp:Label ID="lblFirstname" runat="server">Όνομα:</asp:Label>
<asp:TextBox ID="txtFirstname" runat="server"
CssClass="textEntry"></asp:TextBox>
<asp:RequiredFieldValidator ID="Firstname" runat="server"
CssClass="failureNotification" ErrorMessage="Συμπληρώστε το
όνομα"
ControlToValidate="txtFirstname">*</asp:RequiredFieldValidator
>
</p>
<p>
<asp:Label ID="lblLastname"
runat="server">Επώνυμο:</asp:Label>
<asp:TextBox ID="txtLastname" runat="server"
CssClass="textEntry"></asp:TextBox>
<asp:RequiredFieldValidator ID="Lastname" runat="server"
CssClass="failureNotification" ErrorMessage="Συμπληρώστε το
επίθετο"
ControlToValidate="txtLastname">*</asp:RequiredFieldValidator>
</p>
<p>
<asp:Label ID="lblUsername" runat="server">E-mail:</asp:Label>
<asp:TextBox ID="txtUsername" runat="server"
CssClass="textEntry"></asp:TextBox>
<asp:RequiredFieldValidator ID="Email" runat="server"
CssClass="failureNotification" ErrorMessage="Συμπληρώστε το e-
mail"
ControlToValidate="txtUsername">*</asp:RequiredFieldValidator>
<asp:RegularExpressionValidator Id="vldRejex" RunAt="server"
ControlToValidate="txtUsername" ErrorMessage="Εισάγετε μια
έγκυρη διεύθυνση e-mail"
ValidationExpression="^([a-zA-Z0-9_-\.\.])@[a-z0-9-]+(\.[a-z0-
9-]+)*(\.[a-z]{2,3})$" ></asp:RegularExpressionValidator>
</p>
<p>
<asp:Label ID="lblpassword"
runat="server">Κωδικός:</asp:Label>
<asp:TextBox ID="txtPassword" runat="server"
CssClass="passwordEntry" TextMode="Password"></asp:TextBox>
```

```

<asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" CssClass="failureNotification"
ErrorMessage="Συμπληρώστε τον κωδικό"
ControlToValidate="txtPassword">*</asp:RequiredFieldValidator>
<br />
<asp:RegularExpressionValidator Id="passwordVal"
RunAt="server" ControlToValidate="txtPassword" ErrorMessage="Ο
κωδικός πρέπει να έχει 6-15 ψηφία"
ValidationExpression="^{6,15}$">
</asp:RegularExpressionValidator>
</p>
<p>
<asp:Label ID="lblPasswordConf" runat="server">Επαληθευση
Κωδικού:</asp:Label>
<asp:TextBox ID="ConfirmPassword" runat="server"
CssClass="passwordEntry" TextMode="Password"></asp:TextBox><br
/>
<asp:RequiredFieldValidator
ControlToValidate="ConfirmPassword"
CssClass="failureNotification" Display="Dynamic"
ErrorMessage="Συμπληρώστε τον κωδικό"
ID="ConfirmPasswordRequired"
runat="server">*</asp:RequiredFieldValidator>
<asp:CompareValidator ID="CompareValidator1" runat="server"
ControlToCompare="txtPassword"
ControlToValidate="ConfirmPassword"
CssClass="failureNotification" Display="Dynamic"
ErrorMessage="Ο κωδικός επαλήθευσης δεν αντιστοιχεί στον
κωδικό που έχετε επιλέξει!">*</asp:CompareValidator>
</p>
<p>
<asp:Label ID="lbltown" runat="server">Πόλη:</asp:Label>
<asp:TextBox ID="txtTown" runat="server"
CssClass="textEntry"></asp:TextBox>
<br />
<asp:RequiredFieldValidator ID="town" runat="server"
CssClass="failureNotification" ErrorMessage="Συμπληρώστε την
πόλη"
ControlToValidate="txtTown">*</asp:RequiredFieldValidator>

</p>
<p>
<asp:Label ID="lblAddress"
runat="server">Διεύθυνση:</asp:Label>
<asp:TextBox ID="txtAddress" runat="server"
CssClass="textEntry"></asp:TextBox>
<br />
<asp:RequiredFieldValidator ID="address" runat="server"
CssClass="failureNotification" ErrorMessage="Συμπληρώστε τη
διεύθυνση"
ControlToValidate="txtAddress">*</asp:RequiredFieldValidator>
</p>
<p>
<asp:Label ID="lblPostCode" runat="server">Ταχυδρομικός
Κώδικας:</asp:Label>

```

Κατασκευή ηλεκτρονικού καταστήματος

```
<asp:TextBox ID="txtPostCode" runat="server"
CssClass="textEntry"></asp:TextBox>
<br />
<asp:RequiredFieldValidator ID="pcode" runat="server"
CssClass="failureNotification" ErrorMessage="Συμπληρώστε τον
T.K."
ControlToValidate="txtPostCode">*</asp:RequiredFieldValidator>
<asp:RegularExpressionValidator Id="postVal" RunAt="server"
ControlToValidate="txtPostCode"
ErrorMessage="Εισάγετε έγκυρο TK" ValidationExpression="^[0-
9]{5}$"></asp:RegularExpressionValidator>
</p>
<p>
<asp:Label ID="lblHomephone" runat="server">Σταθερό
Τηλέφωνο:</asp:Label>
<asp:TextBox ID="txtHomephone" runat="server"
CssClass="textEntry"></asp:TextBox>
<asp:RequiredFieldValidator ID="Phone" runat="server"
CssClass="failureNotification" ErrorMessage="Συμπληρώστε το
τηλέφωνο"
ControlToValidate="txtHomephone">*</asp:RequiredFieldValidator
>
<asp:RegularExpressionValidator Id="phoneVal" RunAt="server"
ControlToValidate="txtHomephone" ErrorMessage="Εισάγετε ένα
έγκυρο αριθμό"
ValidationExpression="^[0-
9]{10}$"></asp:RegularExpressionValidator>
</p>
<p>
<asp:Label ID="lblMobilePhone" runat="server">Κινητό
Τηλέφωνο:</asp:Label>
<asp:TextBox ID="txtMobilephone" runat="server"
CssClass="textEntry"></asp:TextBox>
<asp:RegularExpressionValidator Id="mobileVal" RunAt="server"
ControlToValidate="txtMobilephone" ErrorMessage="Εισάγετε ένα
έγκυρο αριθμό"
ValidationExpression="^[0-9]{10}$">
</asp:RegularExpressionValidator>
</p>
<p class="btnSubmit">
<asp:Button ID="btnSignUp" runat="server" Text="Υποβολή" /><br
/>
<asp:ValidationSummary ID="Summary" style="float:left;"
runat="server" CssClass="failureNotification" /><br />
<asp:Label ID="Label1" runat="server"></asp:Label>
</p>
</fieldset>
</div>

</asp:Panel>
</div>
</asp:Content>
```

Register.vb

```

Imports System.Data.SqlClient
Imports System.Web.Configuration
Imports System.Web.UI
Public Class Register
    Inherits System.Web.UI.Page
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load

        Page.Master.FindControl("Uname").Visible = False
        Page.Master.FindControl("Username").Visible = False
        Page.Master.FindControl("Pass").Visible = False
        Page.Master.FindControl("Password").Visible = False
        Page.Master.FindControl("LoginBtn").Visible = False

    End Sub
    Protected Sub btnSignUp_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles btnSignUp.Click
        Dim Con As New
SqlConnection(WebConfigurationManager.ConnectionStrings("CONNE
CTION_STRING").ConnectionString)
        Dim cmd As New SqlCommand
        Dim cmd2 As New SqlCommand
        cmd.CommandType = CommandType.StoredProcedure

        If Session("UserID") > 0 Then
            cmd.CommandText = "UpdateUser"
            Try
                Con.Open()
            Catch ex As Exception
                MsgBox(ex.ToString)
            Finally
                Con.Close()
            End Try
        Else
            cmd.CommandText = "CreateUser"
            Dim conn As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STR
ING"))
            Dim command As New SqlCommand

            command.CommandType = CommandType.StoredProcedure
            command.CommandText = "CheckUsername"
            command.Connection = conn
            command.Parameters.AddWithValue("@Username",
txtUsername.Text)
            conn.Open()
            Session("Uname") = command.ExecuteScalar()
            If Session("Uname") > 0 Then
                Labell.Text = "Το όνομα χρήστη που επιλέξατε
χρησιμοποιείται ήδη."
            Else
                cmd.Connection = Con

                With cmd
                    If Session("UserID") > 0 Then

```

Κατασκευή ηλεκτρονικού καταστήματος

```
                .Parameters.AddWithValue("@UserID",
Session("UserID"))
            End If
                .Parameters.AddWithValue("@FirstName",
txtFirstname.Text.Trim)
                .Parameters.AddWithValue("@LastName",
txtLastname.Text.Trim)
                .Parameters.AddWithValue("@UserName",
txtUsername.Text.Trim)
                .Parameters.AddWithValue("@Password",
txtPassword.Text.Trim)
                .Parameters.AddWithValue("@Address",
txtAddress.Text.Trim)
                .Parameters.AddWithValue("@Town",
txtTown.Text.Trim)
                .Parameters.AddWithValue("@PostCode",
txtPostCode.Text.Trim)
                .Parameters.AddWithValue("@HomePhone",
txtHomephone.Text.Trim)
                .Parameters.AddWithValue("@MobilePhone",
txtMobilephone.Text.Trim)
            End With

            Try
                Con.Open()
                Dim dr As SqlDataReader
                dr = cmd.ExecuteReader
                While dr.Read()
                    Session("UserID") = dr("UserID")
                End While
                dr.Close()
                Labell1.Text = "Η εγγραφή ολοκληρώθηκε με
επιτυχία"

                Catch ex As Exception
                    Session("MESSAGE") = ex.ToString
                    Response.Redirect("~/Message.aspx", True)
                Finally
                    cmd.Dispose()
                    Con.Close()
                    Session("MESSAGE") = "Η εγγραφή
ολοκληρώθηκε με επιτυχία!"
                    Response.Redirect("~/Message.aspx", True)
                End Try
            End If
        End If
    End Sub
End Class
```

Η σελίδα Login.aspx επιτρέπει στο χρήστη να συνδεθεί με τα στοιχεία του, email και password, στο λογαριασμό του, ώστε να μπορεί να προσθέσει προϊόντα στο καλάθι του και να δει και να αλλάξει τα στοιχεία του λογαριασμού του.

```
<%@ Page Title="Σύνδεση" Language="vb"
MasterPageFile="~/Site.Master" AutoEventWireup="false"
```



```

        CodeBehind="Login.aspx.vb" Inherits="eshop.Login" %>

<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server"
ContentPlaceHolderID="MainContent">
    <h2>
        Log In
    </h2>
    <p>
        Εισάγετε το όνομα χρήστη και τον κωδικό. Πατήστε
        <asp:HyperLink ID="RegisterHyperLink" runat="server"
EnableViewState="false" NavigateUrl="Register.aspx"
>Εγγραφή</asp:HyperLink>, αν δεν έχετε λογαριασμό.
    </p>
    <p>
        <asp:Label ID="Label1" runat="server">Όνομα
χρήστη:</asp:Label><br/>
        <asp:TextBox ID="UserName" runat="server"
CssClass="textEntry" Width="100px" TabIndex="0"
></asp:TextBox>
<asp:RequiredFieldValidator ID="UserNameRequired"
runat="server" ControlToValidate="UserName"
CssClass="failureNotification" ErrorMessage="User Name is
required.">*</asp:RequiredFieldValidator>
    </p>
<p>
<asp:Label ID="lblPass"
runat="server">Κωδικός:</asp:Label><br/>
<asp:TextBox ID="Password" runat="server"
CssClass="passwordEntry" TextMode="Password" Width="100px"
TabIndex="1"></asp:TextBox>
<asp:RequiredFieldValidator ID="PasswordRequired"
runat="server" ControlToValidate="Password"
CssClass="failureNotification" ErrorMessage="Password is
required.">*</asp:RequiredFieldValidator>
    </p>
<p class="btnSubmit">
<asp:Button ID="LoginButton" runat="server"
CommandName="Login"
Text="Log In" TabIndex="2"/>
    </p>

<asp:ValidationSummary ID="LoginSummary" runat="server"
CssClass="failureNotification" />

</asp:Content>

```

Login.vb

```

Imports System.Data.SqlClient
Imports System.Web.Configuration
Public Class Login

```

Κατασκευή ηλεκτρονικού καταστήματος

```
Inherits System.Web.UI.Page

Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    Page.Master.FindControl("Uname").Visible = False
    Page.Master.FindControl("Username").Visible = False
    Page.Master.FindControl("Pass").Visible = False
    Page.Master.FindControl("Password").Visible = False
    Page.Master.FindControl("LoginBtn").Visible = False
End Sub

Protected Sub LoginButton_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles LoginButton.Click

    Dim con As New SqlConnection
    Dim cmd As New SqlCommand
    Dim dr As SqlDataReader
    Dim cmd2 As New SqlCommand
    Dim roleid As Integer
    Dim userid As Integer

    ' Try
    con.ConnectionString =
ConfigurationManager.AppSettings("CONNECTION_STRING")
    con.Open()
    cmd = New SqlCommand("SELECT UserID FROM Users " _
& "WHERE UserName = '" & UserName.Text & "' " _
& "AND Password = '" & Password.Text & "'", con)

    cmd2 = New SqlCommand("SELECT role_id FROM Users " _
& "WHERE UserName = '" & UserName.Text & "' " _
& "AND Password = '" & Password.Text & "'", con)

    roleid = cmd2.ExecuteScalar()
    userid = cmd.ExecuteScalar()

    If roleid = 1 Then
        Session("RoleID") = 1
        Session("Username") = "Καλωσήρθες
Διαχειριστή!"

        Session("UserID") = userid
        Response.Redirect("~/Admin/AdminDefault.aspx",
False)

    Else

        dr =
cmd.ExecuteReader(Data.CommandBehavior.SequentialAccess)
```

```
        If dr.Read Then

            Session("UserID") =
dr.GetValue(0).ToString.Trim
            Session("Username") = "Καλωσήρθες " &
UserName.Text & "!"
            Response.Redirect("~/Default.aspx")

        Else
            Session("MESSAGE") = "Παρακαλώ εισάγετε
έγκυρο όνομα χρήστη και κωδικό."
            Response.Redirect("~/Message.aspx", False)
        End If
        dr.Close()

        cmd.Dispose()
        con.Close()

    End If

End Sub
End Class
```

Η σελίδα Products.aspx προβάλλει τα προϊόντα της αντίστοιχης κατηγορίας που έχει επιλεγεί από το μενού σε έναν πίνακα. Τα πεδία που εμφανίζονται στον πίνακα είναι η εικόνα του προϊόντος, η ονομασία του και η τιμή του. Υπάρχει, ακόμα, ένας σύνδεσμος που οδηγεί το χρήστη στην περιγραφή του προϊόντος, καθώς και το κουμπί επιλογής του προϊόντος για το καλάθι αγορών.

```
<%@ Page Title="Προϊόντα" Language="vb"
AutoEventWireup="false" MasterPageFile="~/Site.Master"
CodeBehind="Products.aspx.vb" Inherits="eshop.Products" %>
<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server"
ContentPlaceHolderID="MainContent">

    <asp:Table ID="Table1" runat="server"
BackImageUrl="images/scroll17.jpg" width="657" Height="300">
<asp:TableRow >
<asp:TableCell VerticalAlign="top">
<h2>
    <asp:Label ID="CategoryTitle" runat
="server"></asp:Label>
</h2>
<p>
    <asp:SqlDataSource ID="SqlDataSource1"
runat="server"
ConnectionString="<%$
ConnectionStrings:BoardGamesConnectionString %>"
```

Κατασκευή ηλεκτρονικού καταστήματος

```
SelectCommand="SELECT product_id, product_title,
product_image, product_price, product_description FROM
[Products] WHERE category_name=@category_name">
  <SelectParameters>
    <asp:QueryStringParameter Name="category_name"
DefaultValue="213-46-8915" QueryStringField="ID" />
  </SelectParameters>
</asp:SqlDataSource>

  <asp:DataList ID="DataList1" runat="server"
  Font-Bold="True" Font-Italic="False" Font-Overline="False"
  Font-Strikeout="False" Font-Underline="False"
ForeColor="#333333"
  HorizontalAlign="Left" RepeatColumns="4" ShowFooter="False"
  ShowHeader="True" CellPadding="4" CellSpacing="15" Font-
Size="small"
  DataKeyField="product_id" >
<AlternatingItemStyle BackColor="" />
  <FooterStyle BackColor="#507CD1" Font-Bold="True"
ForeColor="White" />
  <HeaderStyle BackColor="#507CD1" Font-Bold="True"
ForeColor="White" />
  <ItemStyle BackColor="" Font-Size="small"
HorizontalAlign="Center"
  VerticalAlign="Middle" />
  <ItemTemplate>
    <asp:Image ID="product_image" runat="server" ImageUrl = '<%#
Eval("product_image") %>' /><br />
    <asp:Label ID="product_title" runat="server" Text='<%#
Eval("product_title") %>' /> <br />
    Τιμή:
    <asp:Label ID="product_price" runat="server" Text='<%#
Eval("product_price") %>' /> €<br />

    <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl='<%# Eval("product_title",
"~/ProductDetails.aspx?ID={0}") %>' Text='<%#
Eval("product_title", "Λεπτομέρειες..") %>'></asp:HyperLink>
  <br />

  <asp:ImageButton ID="Button2" runat="server" ImageUrl
= "../images/chest.gif" CommandName="AddtoCart" alt="Προσθηκη
στο καλάθι" />
  <br />
<br />
</ItemTemplate>
  <SelectedItemStyle BackColor="#D1DDF1" Font-Bold="True"
ForeColor="#333333" />
</asp:DataList>
</p>
</asp:TableCell>
</asp:TableRow>
</asp:Table>
</asp:Content>
```

Products.vb

```
Imports System
Imports System.Data
Imports System.Data.SqlClient

Public Class Products
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        If Not IsPostBack Then
            CategoryTitle.Text = Session("Category")
            Dim reader As SqlDataReader
            Dim conn As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STR
ING"))

            Dim command As New SqlCommand
            Dim cmd As New SqlCommand
            command.CommandType = CommandType.StoredProcedure
            cmd.CommandType = CommandType.StoredProcedure
            command.CommandText = "Category"
            command.Connection = conn
            command.Parameters.AddWithValue("@category_name",
Session("Category"))
            conn.Open()
            reader = command.ExecuteReader()
            DataList1.DataSource = reader
            DataList1.DataBind()
            reader.Close()

            conn.Close()
            conn.Dispose()
        End If
    End Sub

    Protected Sub DataList1_ItemCommand(ByVal sender As
Object, ByVal e As
System.Web.UI.WebControls.DataListCommandEventArgs) Handles
DataList1.ItemCommand

        Dim currentCommand As String = e.CommandName
        Dim currentRowIndex As Integer =
Int32.Parse(e.Item.ItemIndex)
        If currentCommand = "ProductID" Then
            Session("Product") =
DataList1.DataKeys(currentRowIndex)
            Response.Redirect("Products.aspx", False)
        ElseIf currentCommand = "AddtoCart" Then
            Session("Product") =
DataList1.DataKeys(currentRowIndex)
```

Κατασκευή ηλεκτρονικού καταστήματος

```
        ShoppingCart.Instance.AddItem(Session("Product"))
        Response.Redirect("ViewCart.aspx", False)
    End If
End Sub

End Class
```

Η σελίδα Search.aspx προβάλλει έναν πίνακα με τα αποτελέσματα της αντίστοιχης αναζήτησης που έχει πραγματοποιήσει ο χρήστης. Τα πεδία που εμφανίζονται στον πίνακα είναι η εικόνα του προϊόντος, η ονομασία του και η τιμή του. Υπάρχει, ακόμα, ένας σύνδεσμος που οδηγεί το χρήστη στην περιγραφή του προϊόντος, καθώς και το κουμπί επιλογής του προϊόντος για το καλάθι αγορών.

```
<%@ Page Title="Αποτελέσματα αναζήτησης" Language="vb"
AutoEventWireup="false" MasterPageFile="~/Site.Master"
CodeBehind="Search.aspx.vb" Inherits="eshop.Search" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent"
runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent"
runat="server">
    <br />
    <h2>
        Αποτελεσματα αναζητησης:
    </h2>

    <br />

    <br />
    <asp:Label ID="Label1" runat="server" Text="Search
results" Visible="false" />
    <br />
    <asp:DataList ID="DataList1" runat="server"
Font-Bold="True" Font-Italic="False" Font-Overline="False"
Font-Strikeout="False" Font-Underline="False"
ForeColor="#333333"
HorizontalAlign="Left" RepeatColumns="4" ShowFooter="False"
ShowHeader="True" CellPadding="4" CellSpacing="15" Font-
Size="small" DataKeyField="product_title"
>
<AlternatingItemStyle BackColor="" />
<FooterStyle BackColor="#507CD1" Font-Bold="True"
ForeColor="White" />
<HeaderStyle BackColor="#507CD1" Font-Bold="True"
ForeColor="White" />
<ItemStyle BackColor="" Font-Size="small"
HorizontalAlign="Center"
VerticalAlign="Middle" />
<ItemTemplate>
    <asp:Image ID="product_image" runat="server" ImageUrl = '<%#
Eval("product_image") %>' /><br />
    <asp:Label ID="product_title" runat="server" Text='<%#
Eval("product_title") %>' /> <br />
    Τιμή:
```

```
<asp:Label ID="product_price" runat="server" Text='<%#
Eval("product_price") %>' /> €<br />

<asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl='<%# Eval("product_title",
"~/ProductDetails.aspx?ID={0}") %>' Text='<%#
Eval("product_title", "Λεπτομέρειες..") %>'></asp:HyperLink>
<br />

<asp:ImageButton ID="Button2" runat="server" ImageUrl
= "../images/chest.gif" CommandName="AddtoCart" alt="Προσθήκη
στο καλάθι" />
<br />
<br />
</ItemTemplate>
<SelectedItemStyle BackColor="#D1DDF1" Font-Bold="True"
ForeColor="#333333" />
</asp:DataList>
</asp:Content>
```

Search.vb

```
Imports System.Data.SqlClient
```

```
Public Class Search
```

```
    Inherits System.Web.UI.Page
```

```
    Protected Sub Page_PreRender(ByVal sender As Object, ByVal
e As System.EventArgs) Handles Me.PreRender
        Dim searchcriteria As String = ""
        Dim reader As SqlDataReader
        Dim conn As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STR
ING"))
        Dim command As New SqlCommand
        Dim cmd As New SqlCommand

        searchcriteria = Session("Search")
        command.CommandType = CommandType.StoredProcedure
        cmd.CommandType = CommandType.StoredProcedure
        command.CommandText = "Search"
        command.Connection = conn
        command.Parameters.AddWithValue("@searchcriteria",
searchcriteria)
        conn.Open()
        reader = command.ExecuteReader()

        DataList1.DataSource = reader
        DataList1.DataBind()
        reader.Close()

        conn.Close()
```

Κατασκευή ηλεκτρονικού καταστήματος

```
        conn.Dispose ()
    End Sub

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load

    End Sub

    Protected Sub DataList1_ItemCommand(ByVal sender As
Object, ByVal e As
System.Web.UI.WebControls.DataListCommandEventArgs) Handles
DataList1.ItemCommand

        Dim currentCommand As String = e.CommandName
        Dim currentRowIndex As Integer =
Int32.Parse(e.Item.ItemIndex)
        If currentRowIndex < 0 Then
            Session("Message") = "row index=" &
currentRowIndex & "count=" & DataList1.DataKeys.Count
            Response.Redirect("Message.aspx", False)
        ElseIf currentRowIndex >= DataList1.DataKeys.Count
Then
            Session("Message") = "row index=" &
currentRowIndex & "count=" & DataList1.DataKeys.Count
            Response.Redirect("Message.aspx", False)
        End If
        If currentCommand = "ProductID" Then
            Session("Product") =
DataList1.DataKeys(currentRowIndex)
            Response.Redirect("Search.aspx", False)
        ElseIf currentCommand = "AddtoCart" Then
            Dim title = DataList1.DataKeys(currentRowIndex)
            Dim conn As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STR
ING"))

            Dim command As New SqlCommand

            command.CommandType = CommandType.StoredProcedure
            command.CommandText = "GetID"
            command.Connection = conn
            command.Parameters.AddWithValue("@product_title",
title)

            conn.Open ()
            Session("Product") = command.ExecuteScalar()
            ShoppingCart.Instance.AddItem(Session("Product"))
            Response.Redirect("ViewCart.aspx", False)

        End If
    End Sub
End Class
```