



**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ**  
**Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΦΟΡΗΤΟ ΣΥΣΤΗΜΑ ΑΥΤΟΜΑΤΗΣ ΠΡΟΣΒΑΣΗΣ ΣΕ  
INTRANET ΠΛΗΡΟΦΟΡΙΑ**

Πατσέας Φώτης

**Εισηγητής : Μαλάμος Αθανάσιος**

## ΠΡΟΛΟΓΟΣ

Στις μέρες μας, η τεχνολογία εισχωρεί όλο και περισσότερο στη ζωή του καθενός από εμάς με αποτέλεσμα ο άνθρωπος γίνεται πιο εξοικειωμένος με τη χρήση του ηλεκτρονικού υπολογιστή. Η τάση αυτή οδηγεί στην αυτοματοποίηση και τη συμμετοχή του ανθρώπου σε όλο ένα και λιγότερες διαδικασίες που μπορούν να εκτελεστούν από υπολογιστές με σκοπό τη μείωση του κόστους και του χρόνου υλοποίησης. Η ενσωμάτωση του υπολογιστή στην καθημερινότητά του ανθρώπου, όχι απλά σαν μια ηλεκτρονική συσκευή αλλά σαν ένα εργαλείο, ανοίγει το φάσμα των δυνατοτήτων πληροφόρησης και επικοινωνίας. Ένας αυξανόμενος αριθμός δικτύων θέτει τις βάσεις για την επικοινωνία, δίνοντας τη δυνατότητα ανάπτυξης μιας παγκόσμιας κουλτούρας.

Η πτυχιακή αυτή είναι ένα παράδειγμα στο πώς η τεχνολογία μπορεί να εκμεταλλευτεί για τον σκοπό για τον οποίο αναπτύχθηκε και δεν είναι άλλος από το να κάνει τη ζωή του ανθρώπου ευκολότερη.

Για αυτήν την προσπάθειά μου θα ήθελα να ευχαριστήσω την οικογένειά μου για την οικονομική και ηθική, προπάντων, συμπαράσταση που μου έδειξαν.

Οκτώβριος 2007  
Πατσέας Φώτης

## ΠΕΡΙΛΗΨΗ

Στη παρούσα διπλωματική εργασία θα ασχοληθούμε με τη χρήση των τεχνολογιών asp.net 2.0 και wi-fi. Σκοπός μας είναι η δημιουργία ενός πλήρους λειτουργικού intranet based ιστοχώρου, ανοιχτού σε όποιον διαθέτει τα κατάλληλα wi-fi ready μέσα για τη προσπέλασή του.

Οι τεχνολογίες .NET Framework και Mobile Framework δίνουν τη δυνατότητα μιας πολυδιάστατης εφαρμογής με τη χρήση της ίδιας βάσης δεδομένων για το σύνολο των χρηστών. Η διαδικασία της αναγνώρισης και ταυτοποίησης αν και τυπική, καθώς η ασφάλεια της εφαρμογής βασίζεται στη μοναδικότητα κάθε κάρτας δικτύου, είναι ένας ακόμα κοινός παράγοντας.

Ένας αυτοπονημένος τρόπος εισαγωγής στο σύστημα στηριζόμενος στη δρομολόγηση από το router – access point του δικτύου προς τον κεντρικό εξυπηρετητή, απλοποιεί τη χρήση της εφαρμογής ακόμα και για τον πιο τεχνολογικά αδαή χρήστη. Από τη στιγμή αυτή η βάση δεδομένων έχει πρωταρχικό ρόλο στο σύστημα και ένας κώδικας βασιζόμενος σε VBScript αναλαμβάνει να φέρει εις πέρας όλες τις απαραίτητες διαδικασίες.

Πρόθεσή μας ήταν το αποτέλεσμα αυτής της διπλωματικής να ήταν μια ιδιαίτερα εύχρηστη εφαρμογή με αρκετές δυνατότητες τροποποίησης, αναβάθμισης και χρήσης.

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

<b>ΠΡΟΛΟΓΟΣ</b>	1
<b>ΠΕΡΙΛΗΨΗ</b>	2
<b>ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ</b>	3
<b>1. ΕΙΣΑΓΩΓΗ</b>	4
1.1 Αντικείμενο της Διπλωματικής	5
1.2 Τεχνολογίες και Λογισμικό	5
1.2.1 ASP.NET	5
1.2.2 VISUAL BASIC .NET	6
1.2.3 INTRANET	7
1.2.4 Wi-Fi	8
1.2.5 Λογισμικό	8
1.3 Οργάνωση του Τόμου	9
<b>2. ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ</b>	10
2.1 Δομή	10
2.1.1 Υποστηριζόμενες Συσκευές	10
2.1.2 Κεντρική Σελίδα	11
2.1.3 Μενού Πλοήγησης	13
2.2 Λειτουργία	14
2.2.1 Δημόσιες Σελίδες	14
2.2.2 Εγγραφή και Διαδικασία Εισαγωγής	15
2.2.3 Προσωπικός Λογαριασμός	16
2.2.4 Παραμετροποίηση Πληροφοριών	18
2.2.5 Προβολή Λεπτομερειών	19
2.2.6 Χειρισμός Παραμετροποίησης	20
2.2.7 Διαδικασία Εξόδου	22
<b>3. ΕΓΧΕΙΡΙΔΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ</b>	23
3.1 Ανάλυση και Σχεδίαση Βάσης Δεδομένων	24
3.2 Υλοποίηση και Κώδικας	40
<b>4. ΕΠΙΛΟΓΟΣ</b>	58
<b>5. ΒΙΒΛΙΟΓΡΑΦΙΑ</b>	59

# 1

## ΕΙΣΑΓΩΓΗ

Ένα ασύρματο δίκτυο είναι ένα σημαντικό εργαλείο για μια εταιρεία. Η δυνατότητα επικοινωνίας στα φυσικά όριά της αλλά και στην εμβέλεια του δικτύου της, το καθιστά από τους βασικότερους συντελεστές ανάπτυξης και λειτουργικότητας.

Όταν τώρα αυτό το δίκτυο αντί να περιοριστεί στο ρόλο του διαμεσολαβητή μεταξύ κυρίως υπαλλήλων, προσφέρεται και στο κοινό, υπό κάποιους περιορισμούς πάντα, τότε μιλάμε για μια δικτυακή “βιτρίνα”, για ένα σύμμαχο στο πλευρό κάθε πωλητή αλλά και κάθε καταναλωτή, εξοικειωμένου έστω και σε μικρό βαθμό με τη τεχνολογία.

Η αναζήτηση νέων τρόπων επικοινωνίας με το πελάτη, μιας πιο άμεσης επαφής η οποία θα προσέγγιζε τα πραγματικά “θέλω” του πριν ακόμα δημιουργηθεί η ανάγκη πραγματοποίησής τους, θα έδινε ένα προβάδισμα σε κάθε επιχείρηση μέσα στο ανταγωνιστικό περιβάλλον κάθε αναπτυσσόμενης αγοράς.

Από την άλλη πλευρά ο πελάτης έχει να αντιμετωπίσει τον καταιγισμό από πλασματικά “θέλω” που εσκεμμένα χρησιμοποιεί κάθε πάροδος για τη προώθηση υπηρεσιών και προϊόντων. Πολύχρωμες συσκευασίες, ποσοτικές ή ποιοτικές προσφορές και σλόγκαν που χαρακτηρίζουν προϊόντα είναι τα μέχρι τώρα μέσα που χρησιμοποιούνται για το δελεασμό του καταναλωτή.

## 1.1 Αντικείμενο της Διπλωματικής

Το σύστημα λοιπόν της πτυχιακής βασίζεται σε όλα τα παραπάνω προκειμένου να δικαιολογήσει τη αναγκαιότητά του. Η αρχική σκέψη ήταν η προσέγγιση της φοριτότητας της τεχνολογίας και ο συνδυασμός της με το “σερφάρισμα” στο internet, τη πιο συνηθισμένη δραστηριότητα μετά το περιήγηση στα Windows για κάποιον που διαθέτει υπολογιστή. Έτσι το σύστημα θα είναι λειτουργικό αλλά και παράλληλα όσο το δυνατόν πιο οικείο στον χρήστη, βασικός παράγοντας για τη διεκδίκηση της χρήσης του από τους πελάτες.

Το σύστημα δίνει τη δυνατότητα πρόσβασης μέσω δύο οντοτήτων , η πρώτη απευθύνεται σε χρήστες που χρησιμοποιούν Laptop και οι δεύτερη σε εκείνους που χρησιμοποιούν PDA. Οι διαφορά των δύο αυτών μορφών είναι μόνο στη ανάλυση με την οποία προβάλλονται οι πληροφορίες. Οποιοδήποτε άλλο χαρακτηριστικό του συστήματος παραμένει το ίδιο είτε στη μια, είτε στην άλλη μορφή του.

Η ασφάλεια του συστήματος βασίζεται κυρίως στο χαρακτηριστικό αποτύπωμα της κάρτας δικτύου κάθε χρήστη, κατά τα άλλα η είσοδος στο ασύρματο δίκτυο είναι ελεύθερη αλλά και η χρήση cookies δρα επικουρικά σε αναγνώριση και ταυτοποίηση.

Μετά τα διακριτά βήματα της εισόδου και εν συνεχεία της ταυτοποίησης ο χρήστης καλείται να μοιραστεί προσωπικές πληροφορίες και λεπτομέρειες προκειμένου να εξαχθούν κάποια συμπεράσματα για τις ανάγκες του και να προβληθούν στη συνέχεια οι αντίστοιχες προσφορές. Αυτά τα προσωπικά δεδομένα μπορούν να επεξεργαστούν από το χρήστη οποιαδήποτε στιγμή κριθεί απαραίτητο από εκείνον.

Από τη στιγμή αυτή και μετά ο χρήστης αποκτά το δικό του χώρο στο σύστημα.

## 1.2 Τεχνολογίες και Λογισμικό

Πριν περάσω στην ανάλυση του θέματος της διπλωματικής θα ήθελα να αναφέρω μερικές πληροφορίες για τις τεχνολογίες και το λογισμικό που χρησιμοποιήθηκε.

### 1.2.1 ASP.NET

Οι νέες τεχνολογίες, οι έξυπνες και λειτουργικές εφαρμογές αλλά και η μεγάλη διείσδυση του Internet στην καθημερινή ζωή όλων ανέβασαν τον πήχη ακόμα πιο ψηλά. Οι απαιτήσεις των εφαρμογών είναι πλέον μεγαλύτερες, χαρακτηριστικό που προϋποθέτει διαφορετική πλατφόρμα ανάπτυξης, η οποία θα επιτρέπει στους κατασκευαστές λογισμικού να χτίζουν πιο σύνθετες εφαρμογές αλλά και να τις διαθέτουν με πιο γρήγορο και άμεσο τρόπο. Επιπλέον, η ανάγκη για άμεση ανταλλαγή των πληροφοριών μεταξύ διαφορετικών εφαρμογών οδήγησε στη δημιουργία της γλώσσας XML, ενός προτύπου που αναδεικνύεται στο ASCII του Internet. Δεν είναι άλλωστε τυχαίο το γεγονός ότι, κατ' εντολή του ίδιου του Γκέιτς, όλες οι εφαρμογές της Microsoft υποστηρίζουν πλέον εγγενώς την XML. Σε αυτό το νέο περιβάλλον που διαμορφώνεται, η Microsoft έρχεται να δημιουργήσει μια νέα πλατφόρμα, την οποία ονομάζει .NET προκειμένου να δώσει ώθηση στις κατανεμημένες εφαρμογές της επόμενης γενιάς.

Με το .NET η μεγάλη αμερικανική εταιρεία θα προσπαθήσει να ξανασχεδιάσει το Internet ή, σύμφωνα με κάποιους, να επιβάλει έναν άλλο τρόπο δημιουργίας και υποστήριξης εφαρμογών σε αυτό. Η εν λόγω πλατφόρμα αποτελείται από δύο βασικές συνιστώσες, την υποδομή που πρέπει να ενσωματώνει ένας διακομιστής, και τις εφαρμογές που εκτελούνται και προσδιορίζουν τις λειτουργίες που έχει ο χρήστης στη διάθεσή του.

Η υποδομή (*.NET Framework*) περιλαμβάνει τις απαραίτητες υπηρεσίες που θα διευκολύνουν τις εφαρμογές να επικοινωνήσουν με το χρήστη, να αποθηκεύσουν και να εμφανίσουν στοιχεία από τις βάσεις δεδομένων από το συγκεκριμένο διακομιστή ή ακόμα και από άλλους απομακρυσμένους αποθηκευτικούς χώρους. Απαιτεί ένα διακομιστή Windows 2000 ή μεταγενέστερο και τις επεκτάσεις του .NET Framework.

Η δεύτερη συνιστώσα της πλατφόρμας απαιτεί την εγκατάσταση και τη χρήση του Visual Studio .NET, το οποίο επιτρέπει τη δημιουργία εφαρμογών και υπηρεσιών που εκτελούνται σε ένα διακομιστή. Ένα παράδειγμα είναι η νέα μορφή των σελίδων ASP (Active Server Pages), που ονομάζονται ASP .NET (ή ASP+, όπως είχαν αρχικά ονομαστεί) και έχουν κατάληξη .aspx (αντί για .asp). Η νέα αυτή έκδοση στοχεύει, όπως πάντα, στον προγραμματιστή, αλλά απαιτεί πολύ λιγότερο κώδικα. Αντί για κώδικα VBScript μέσα σε εντολές HTML, μπορεί κανείς να χρησιμοποιήσει έναν πιο "καλό" κώδικα Basic, που περιλαμβάνει όλα τα χαρακτηριστικά της

Visual Basic, καθώς και τις νέες επεκτάσεις της γλώσσας (αντικειμενοστραφής προγραμματισμός).

Η πλατφόρμα .NET προσφέρει όλη την αναγκαία υποδομή για την ανάπτυξη και την εκτέλεση n-tier εφαρμογών στο Internet. Όπως πάντα, τόσο μεγάλες αλλαγές έχουν σημαντικό κόστος: τη συμβατότητα με τις προηγούμενες εκδόσεις. Εργαζόμενοι με την πλατφόρμα .NET, θα χρειαστεί πολλές φορές να ξαναγράψετε μεγάλα κομμάτια κώδικα ή ακόμα και ολόκληρες εφαρμογές από την αρχή. Για αυτό και τις περισσότερες φορές δεν συνιστάται η μετατροπή από τη VB 6 στο περιβάλλον του .NET, καθώς οι αλλαγές είναι πολλές και κρύβουν αρκετές παγίδες ακόμα και στη ίδια τη λογική της εφαρμογής.

## 1.2.2 VISUAL BASIC .NET

Η Visual Basic ακολούθησε εντυπωσιακή εξέλιξη την τελευταία δεκαετία, από την έκδοση 1.0 στην σημερινή έκδοση .NET, ή 7.0, όπως είναι η εσωτερική αρίθμηση του προϊόντος. Οι νέες εντολές, οι βελτιώσεις στον τρόπο ανίχνευσης των λαθών αλλά και όσα εργαλεία προστέθηκαν μέχρι τη σημερινή έκδοση, δεν άλλαξαν τον αντικειμενικό σκοπό, που είναι η εύκολη και γρήγορη ανάπτυξη εφαρμογών.

Έτσι, η Visual Basic, μέσα από το Visual Studio, παραμένει ένα εύκολο και ιδιαίτερα γρήγορο εργαλείο για την ανάπτυξη αυτόνομων εφαρμογών αλλά και υπηρεσιών στο Web. Επιλέγοντας μια νέα λύση (Solution), μέσα από το περιβάλλον της Visual Basic, μπορείτε να δημιουργήσετε μια αυτόνομη εφαρμογή (Windows Application), μια εφαρμογή στο Internet (Web Application) ή μια υπηρεσία στο Web (Web Service). Και στις τρεις περιπτώσεις όλα γίνονται εύκολα, με το σύστημα να κάνει για εσάς την περισσότερη δουλειά.

Ωστόσο, η νέα έκδοση Visual Basic.NET φέρνει αρκετές αλλαγές στην ίδια τη γλώσσα, στο περιβάλλον προγραμματισμού και κατ'επέκταση στη μεταφορά μιας υπάρχουσας εφαρμογής. Οι βελτιώσεις στη γλώσσα περιλαμβάνουν νέες εντολές και σχήματα που φέρνουν πιο κοντά τη VB στις υπόλοιπες αντικειμενοστραφείς γλώσσες, όπως η C++ και η Java.

## 1.2.3 INTRANET



Το Intranet είναι ένα δίκτυο υπολογιστών που βρίσκεται εγκατεστημένο σε μια επιχείρηση, προκειμένου να εξυπηρετήσει κυρίως τις ανάγκες της για εσωτερική πληροφόρηση και οργάνωση. Αποτελείται από ηλεκτρονικούς υπολογιστές (εκ των οποίων τουλάχιστον ο ένας είναι ο κεντρικός, ο server), οι οποίοι συνδέονται μεταξύ τους ενσύρματα ή, σπανιότερα, ασύρματα.

Τη δικτύωση αυτή πλαισιώνουν εξειδικευμένες εφαρμογές λογισμικού, οι περισσότερες από τις οποίες είναι ίδιες με εκείνες που χρησιμοποιούνται στο Internet. Ενδεικτικά, χρησιμοποιούνται τα πρωτόκολλα επικοινωνίας HTTP, TCP/IP, οι γλώσσες προγραμματισμού HTML, XML, ενώ για την πλοήγηση (στο Intranet) χρησιμοποιούνται φυλλομετρητές (browsers), όπως λ.χ. ο Internet Explorer ή ο Netscape Navigator.

Λόγω των ομοιοτήτων αυτών, το Intranet αποκαλείται και "Internet της επιχείρησης". Στα ελληνικά, ο όρος Intranet μπορεί να αποδοθεί ως "ενδοδίκτυο" ή "εσωτερικό δίκτυο", ενώ περισσότερο περιγραφικός είναι ο αγγλικός όρος "Enterprise Information Portal", που μεταφράζεται ως "πληροφοριακή πύλη της επιχείρησης".

Δομικό χαρακτηριστικό του Intranet είναι η ιδιωτικότητα, σύμφωνα με την οποία δικαίωμα εισόδου στο δίκτυο έχουν μόνο όσοι διαθέτουν κωδικό πρόσβασης. Τα δικαιώματα πρόσβασης μπορεί να είναι διαβαθμισμένα, δηλαδή η πρόσβαση να μην επιτρέπεται σε όλους και σε όλο το περιεχόμενο του Intranet, αλλά οι εργαζόμενοι να έχουν πρόσβαση ανάλογα με τη θέση και τα καθήκοντά τους. Σημειώνεται ότι παρά την ιδιωτικότητα και τον εσωτερικό τους χαρακτήρα, τα Intranets έχουν διεξόδους πρόσβασης στο Διαδίκτυο.

Η συνηθέστερη μορφή που λαμβάνει το Intranet είναι αυτή του μικρού τοπικού δικτύου, αποτελούμενου από έναν αριθμό υπολογιστών, οι οποίοι στεγάζονται στα γραφεία της επιχείρησης. Μπορεί όμως να αποτελείται και από πολλά μικρά ή μεγαλύτερα τοπικά δίκτυα, τα οποία έχουν ενοποιηθεί μέσω μισθωμένων γραμμών (οι οποίες παρέχονται από τους ISP). Με αυτό τον τρόπο, το Intranet μπορεί να συμπεριλάβει μια ολόκληρη επιχείρηση, από τα κεντρικά της γραφεία μέχρι τα απομακρυσμένα υποκαταστήματα.

Πρακτικά, η πρόσβαση στο Intranet πραγματοποιείται μέσω ενός φυλλομετρητή (browser), που μόλις ενεργοποιηθεί, ανοίγει την αρχική σελίδα του Enterprise Information Portal.

Παρενθετικά αναφέρεται ότι ο υπολογιστής μέσω του οποίου θα πραγματοποιηθεί η πρόσβαση στο Intranet δεν είναι απαραίτητο να είναι συνδεδεμένος στο τοπικό δίκτυο. Μπορεί να είναι συνδεδεμένος μόνο στο Internet, και η πρόσβαση στο Intranet να γίνεται μέσω Διαδικτύου.

## 1.2.4 Wi-Fi

Το Wi-Fi προέρχεται από τα αρχικά των «Wireless Fidelity» (Ψηφιακή Πιστότητα) και έχει επικρατήσει σαν όρος για το υψηλής συχνότητας ασύρματο τοπικό δίκτυο (WLAN). Βασικά αποτελεί ένα ασύρματο τρόπο διασύνδεσης, ενώ δίνει την δυνατότητα σύνδεσης και με το Internet.

Οι ασύρματες τεχνολογίες πρόσβασης χρησιμοποιούνται για να αντικαταστήσουν ή να επεκτείνουν ένα κοινό ενσύρματο δίκτυο (Ethernet) και επιτρέπουν στον κινητό χρήστη την ασύρματη μετάδοση και λήψη δεδομένων.

Τα Ασύρματα Τοπικά Δίκτυα (WLANs) ακολουθούν το πρότυπο IEEE 802.11, το πρώτο πρότυπο για ασύρματη δικτύωση το οποίο αναπτύχθηκε. Τα ασύρματα τοπικά δίκτυα τα οποία είναι συμβατά με το πρότυπο IEEE 802.11 ονομάζονται και δίκτυα Wi-Fi.

## 1.2.5 Λογισμικό

- **ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ**  
Microsoft Windows Sever 2003 Enterprise R1  
Microsoft Windows XP Professional
- **ΦΥΛΛΟΜΕΤΡΗΤΕΣ (BROWSERS)**  
Microsoft Internet Explorer 7.0
- **ΠΡΟΣΟΜΟΙΩΤΕΣ**  
Microsoft Device Emulator 1.0
- **ΕΠΕΞΕΡΓΑΣΙΑ ΓΡΑΦΙΚΩΝ**  
Adobe Photoshop CS2
- **ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ – ΣΧΕΔΙΑΣΜΟΣ ΙΣΤΟΣΕΛΙΔΩΝ**  
Microsoft Visual Studio 2005  
Adobe Macromedia Dreamweaver 8

- **ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ**  
Microsoft SQL Server Express 2005
- **ΕΠΕΞΕΡΓΑΣΙΑ ΚΕΙΜΕΜΟΥ**  
Microsoft Word Enterprise 12

## 1.3 Οργάνωση του Τόμου

Στην ενότητα αυτή παραθέτω την οργάνωση του τόμου ο οποίος χωρίζεται σε πέντε κεφάλαια.

Στο πρώτο κεφάλαιο όπως είδαμε κάναμε μια σύντομη αναφορά στο θέμα της διπλωματικής, ενώ αναπτύχθηκαν με λίγα λόγια οι τεχνολογίες που χρησιμοποιήθηκαν για την αποπεράτωση της πτυχιακής αλλά και τα εργαλεία που υπηρέτησαν το σκοπό αυτό.

Το δεύτερο κεφάλαιο απευθύνεται στους εν δυνάμει χρήστες της εφαρμογής. Γίνεται καταρχάς μια προσπάθεια αποσαφήνισης του περιβάλλοντος στο οποίο καλείται ο χρήστης να κινηθεί, με αρκετές φωτογραφίες για καλύτερη κατανόησή του και στη συνέχεια περιγράφονται λεπτομερώς οι υποστηριζόμενες διαδικασίες που ο χρήστης έχει στη διάθεσή του.

Στο τρίτο κεφάλαιο πηγαίνουμε πιο βαθιά αγγίζοντας τον κώδικα της εφαρμογής καθώς γίνεται και μία ανάλυση της βάσης δεδομένων και των στοιχείων της. Οι αναφορές σε ανάλυση και σχεδίαση των απαιτήσεων της εφαρμογής απευθύνονται σε προγραμματιστές τέτοιου είδους εφαρμογών ή σε χρήστες με κάποιες γνώσεις στις συγκεκριμένες τεχνολογίες.

Στο τέταρτο κεφάλαιο παραθέτονται τα συμπεράσματα στα οποία μπορεί να καταλήξει κάποιος μπαίνοντας στη διαδικασία χρήσης αλλά και προγραμματισμού της εφαρμογής. Επίσης γίνεται μια αναφορά στη περαιτέρω ανάπτυξη ή τροποποίηση της για την εξυπηρέτηση διαφορετικών αναγκών σε κάθε περίπτωση. Κλείνοντας το κεφάλαιο προτείνω προτάσεις όσον αφορά στο φάσμα δυνατών χρήσεων μιας τέτοιας εφαρμογής αν λάβουμε υπόψιν μας τις δυνατότητες που μας δίνει η χρήση των συγκεκριμένων τεχνολογιών.

Στο πέμπτο και τελευταίο κεφάλαιο καταγράφεται η σχετική βιβλιογραφία αλλά και μια συλλογή από διαδικτυακές τοποθεσίες που έδωσαν τις απαραίτητες ιδέες και πληροφορίες για τη συγγραφή του παρόντος κειμένου.

# 2

## ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ

### 2.1 Δομή

#### 2.1.1 Υποστηριζόμενες Συσκευές

Το σύστημα έχει δομηθεί έτσι ώστε οποιαδήποτε συσκευή με δυνατότητα σύνδεσης wi-fi να έχει πρόσβαση σε αυτό. Τεχνολογίες .NET Framework και .NET Mobile Framework παρέχουν τις απαραίτητες πληροφορίες για τη διαφορά απεικόνισης για τη μία και την άλλη περίπτωση σύνδεσης. Έτσι ο χρήστης κάτοχος PDA έχει τα ίδια δικαιώματα και δυνατότητες με οποιοδήποτε άλλο χρήστη με πρόσβαση μέσω Laptop ή κάποιο άλλο υπολογιστή γραφείου.



Η χρήση των Laptop έχει αυξηθεί αρκετά τα τελευταία χρόνια. Πριν λίγο καιρό το να είχε κάποιος μαζί μια τέτοια συσκευή ήταν αδιανόητο. Η αύξηση των ασύρματων δικτύων όμως έκανε εμφανές ότι τα Laptop θα μπορούσαν να έχουν και μια άλλη χρήση από αυτή που είχαμε συνηθίσει μέχρι τότε. Ο χρήστης κατάλαβε ότι μπορεί να απολαμβάνει τις χρήσεις του internet κάνοντας τη δουλειά του ακριβώς με τον ίδιο τρόπο όπως στο σπίτι ή στο γραφείο του. Με αυτή τη λογική

υπολογίστηκαν και αυτού του είδους οι συσκευές στο σχεδιασμό της εφαρμογής. Αυτοί οι χρήστες έχουν ένα πλεονέκτημα όσον αφορά στο θέμα της ανάλυσης της οθόνης. Το συγκεκριμένο χαρακτηριστικό τους δίνει τη δυνατότητα μίας άνετης περιήγησης στους χώρους του ιστοχώρου και μια πιο σφαιρική άποψη της εφαρμογής.

Από την άλλη πλευρά η ταυτόχρονη αύξηση της χρήσης των κινητών τηλεφώνων υποβοηθούμε και από τεχνολογίες μεταφοράς ήχου και εικόνας αλλά και η δυνατότητα συνεργασίας με υπολογιστές, άνοιξαν



το δρόμο σε μία καινούρια αγορά εκείνη των υπολογιστών χειρός. Αν και στα πρώτα στάδια της αγοράς αυτής φαινόταν ότι απευθύνεται σε εξειδικευμένους χρήστες η αγορά αυτή έχει αρχίσει να απευθύνεται σε όλο και μεγαλύτερο κομμάτι χρηστών. Γιαυτό το λόγο δε θα μπορούσαν να μην υπολογιστούν και αυτοί οι χρήστες σαν εν δυνάμει χρήστες του συστήματος. Όσον αφορά τώρα στη πρόσβαση στο σύστημα μέσω αυτού του είδους τις συσκευές θα μπορούσαμε να πούμε ότι δεν αλλάζουν και πολλά πράγματα σε

σχέση με τα Laptop. Το γεγονός ότι έχει μικρότερη ανάλυση είναι σίγουρο ότι περιορίζει τη λειτουργικότητα περιήγησης αλλά για ένα χρήστη εξοικειωμένου σε αυτές τις συσκευές και με μία ιστοσελίδα που δε κάνει διακρίσεις όσων αφορά τις συσκευές που έχουν πρόσβαση σε αυτή, τέτοιου είδους προβλήματα εξαλείφονται.

## 2.1.2 Κεντρική Σελίδα



Η κεντρική σελίδα της εφαρμογής (όπως παρουσιάζεται και στη διπλανή φωτογραφία) αποτελείται από τα εξής βασικά στοιχεία. Καταρχάς στο πάνω μέρος της σελίδας υπάρχει το λογότυπο της εταιρείας. Η συγκεκριμένη εικόνα είναι και υπερσύνδεση στη κεντρική σελίδα της εφαρμογής και είναι προσβάσιμη από οποιαδήποτε σελίδα της εφαρμογής. Ακριβώς από κάτω μπορούμε να δούμε τη κατάσταση στην οποία βρισκόμαστε, αν είμαστε εγγεγραμμένοι και συνδεδεμένοι σαν επώνυμοι χρήστες του συστήματος. Αυτή η διαδικασία θα επεξηγηθεί περαιτέρω στη συνέχεια του κειμένου. Στη προκειμένη περίπτωση εμφανιζόμαστε συνδεδεμένοι και αυτό

το καταλαβαίνουμε από την έκφραση “LoggedIn as,” ακολουθούμενη από το όνομα με το οποίο έχουμε επιλέξει η εφαρμογή να απευθύνεται σε εμάς. Η αναγνώριση του κάθε χρήστη γίνεται μέσω της κάρτας δικτύου του εκάστοτε χρήστη αλλά και του cookie που δημιουργείται με τη πρώτη είσοδο του χρήστη στο σύστημα και αποθηκεύεται στο μηχάνημά του. Η διαδικασία της αναγνώρισης και ταυτοποίησης θα περιγραφεί αναλυτικότερα στη συνέχεια και έννοιες όπως cookies και MAC Address ή Physical Address θα επαρκώς έτσι ώστε να κατανοηθούν τα χαρακτηριστικά τους και η βασική λειτουργία τους. Προχωρώντας στη κεντρική σελίδα τώρα μπορούμε να δούμε (όπως φαίνεται και στην εικόνα από Laptop παρακάτω) τα προϊόντα που βρίσκονται σε προσφορά για τη συγκεκριμένη μέρα και ανταποκρίνονται στα προσωπικά δεδομένα που έχουμε εισάγει προηγουμένως. Σε αυτό το σημείο θα ήθελα να πω ότι μπαίνοντας για πρώτη φορά στο σύστημα δε θα συναντήσουμε τη συγκεκριμένη σελίδα, αλλά το σύστημα θα μας μεταφέρει καταρχάς στη σελίδα εγγραφής και στη συνέχεια στη σελίδα συμπλήρωσης των προσωπικών μας δεδομένων και προτιμήσεων. Ο χρήστης είναι καταγεγραμμένος και όσο και οι προτιμήσεις του και σαν κεντρική σελίδα από κει και πέρα θα του εμφανίζεται η παρακάτω (αν μιλάμε για laptop) ή η παραπάνω αν μιλάμε για PDA.

Βέβαια καθόλη τη διάρκεια της περιήγησής σου ο χρήστης δε χάνει το δικαίωμα να αλλάξει τα προσωπικά του δεδομένα τα οποία ενημερώνονται άμεσα και γίνεται η προβολή των αντίστοιχων προσφορών της ημέρας. Όσον αφορά στα Laptop η συγκεκριμένη επιλογή βρίσκεται στο πάνω μέρος της λίστας προσφορών δίπλα στο “Οι Προσφορές της Ημέρας” και υποδεικνύεται με το βελάκι δίπλα από

τη φράση το οποίο εξελίσσεται σε υπερσύνδεση με το που περάσει ο κέρσορας από πάνω του. Στα PDA τώρα η συγκεκριμένη υπερσύνδεση βρίσκεται στο κάτω μέρος της λίστας προσφορών και λειτουργεί ακριβώς με το ίδιο τρόπο όπως και στα Laptop.

Αν μπορούσαμε να πούμε για διαφορές στις δύο προβολές θα μπορούσαμε να πούμε ότι στη μεν προβολή για Laptop στο δεξί μέρος της οθόνης υπάρχει ένα κείμενο το οποίο αφορά κάποια στοιχεία της εταιρείας. Το συγκεκριμένο μέρος χρησιμοποιείται και για τη προβολή περαιτέρω πληροφοριών για κάθε προσφορά όταν επιλεγεί η αντιστοιχεί διαδικασία από τον χρήστη. Σε αυτό το θέμα θα μπορούσαμε να πούμε ότι η προβολή του ιστοχώρου έχει κάποιο μειονέκτημα λόγω του ότι η έλλειψη χώρου δημιουργεί την ανάγκη προβολής των πληροφοριών αυτών σε διαφορετική σελίδα.

### 2.1.3 Μενού Πλοήγησης

Το μενού της εφαρμογής είναι από τα πιο απλά κομμάτια του συστήματος ο χρήστης έχει να επιλέξει μονάχα μία υπερσύνδεση με την οποία μάλιστα μεταφέρεται στη σελίδα εισαγωγής – μετατροπής των προσωπικών του στοιχείων και χαρακτηριστικών. Στην προβολή σε Laptop τώρα το μενού είναι τοποθετημένο στο πάνω μέρος της λίστας προσφορών και βρίσκεται εκεί από την πρώτη στιγμή που ο χρήστης έχει πρόσβαση στη σελίδα. Παρόλαυτα η πρόσβαση στη συγκεκριμένη υπερσύνδεση επιτρέπεται μόνο εφόσον ο χρήστης έχει εγγράφει επιτυχώς και η MAC Address του έχει αποθηκευτεί στη βάση μαζί με τις προτιμήσεις του.

Στη προβολή σε PDA τώρα όπως αναφέραμε και πιο πάνω το μενού είναι τοποθετημένο στο κάτω μέρος της λίστας προσφορών και λειτουργεί ακριβώς με τον ίδιο τρόπο όπως και στη μεγαλύτερη προβολή, έχοντας ακόμα και του ίδιους περιορισμούς πρόσβασης.

Ο λόγος που το μενού πλοήγησης δεν είναι τόσο πολύπλοκο είναι γιατί καταρχάς η εφαρμογή έχει συγκεκριμένες υπηρεσίες που παρέχει, έτσι λοιπόν δεν υπήρχε λόγος να προστεθούν περισσότεροι σύνδεσμοι. Ένας άλλος λόγος που το μενού είναι τόσο λιτό είναι γιατί στο σχεδιασμό της εφαρμογής ο βαθμός λειτουργικότητας της εφαρμογής το οποίο συνεπάγεται ότι θα έπρεπε να είχε και ξεκάθαρο τρόπο λειτουργίας και τρόπο με τον οποίο προβάλλει τις πληροφορίες που θέλει να προσφέρει στο κοινό και να διεκδικήσει από αυτούς.

Σαυτό το σημείο θα ήθελα να πω ότι αν και η εφαρμογή χαρακτηρίζεται σαν ιστοσελίδα αλλά και αντιμετωπίζεται μαυτό το τρόπο έστω και αν είναι εσωτερική παρολαυτά η δομή της σε κάποια σημεία δε συμβαδίζει

με τα κοινά αποδεκτά στάνταρ σχεδίασης και κατασκευής μιας ιστοσελίδας. Το χαρακτηριστικό αυτό μπορεί να μπερδέψει προς στιγμή τον χρήστη ιδιαίτερα τους χρήστες που μπαίνουν για πρώτη φορά σ αυτή.

Θα μπορούσε φυσικά να υπάρχουν περισσότερες δυνατότητες στην εφαρμογή όπως η διαδικασία αποστολής mail αλλά και εμφάνιση πληροφοριών (πράγμα το οποίο γίνεται στη προβολή για Laptop σε κάποιο βαθμό). Τέτοιου είδους υπηρεσίες κρίθηκαν ανούσιες από τη στιγμή που το σύστημα προοριζόταν για τη συγκεκριμένη χρήση.

Για τις δυνατότητες εμπλουτισμού της εφαρμογής θα γίνει λόγος στο τέταρτο κεφάλαιο πιο αναλυτικά μαζί με μια προσπάθεια απαρίθμησης και των πιθανών εφαρμογών τις έτσι ώστε να καλύπτει διαφορετικές ανάγκες σε κάθε περίπτωση. Προς το παρόν επικεντρώνουμε το ενδιαφέρον μας στη κατανόηση της εφαρμογής με τις συγκεκριμένες υποστηριζόμενες διαδικασίες.

## 2.2 Λειτουργία

Μπαίνουμε τώρα στη περαιτέρω επεξήγηση της εφαρμογής και πιο συγκεκριμένα στο κομμάτι της λειτουργίας της. Αφού είδαμε την εμφάνισή της και στις δύο διαφορετικές μορφές προβολής της ανάλογα με τη συσκευή του χρήστη και πήραμε μια γεύση για το τι θα συναντήσει ο χρήστης αφού συνδεθεί είναι ώρα εξηγήσουμε τις διαδικασίες αυτές πιο λεπτομερώς κάνοντας μια προσπάθεια μάλιστα να τις χωρίσουμε και να τις απορρυθμίσουμε σε διακριτά κομμάτια – βήματα υποθέτοντας ότι είμαστε χρήστες που μπαίνουμε για πρώτη φορά μέσα στο σύστημα.

### 2.2.1 Δημόσιες Σελίδες

Στην εφαρμογή μας δεν υπάρχουν δημόσιες σελίδες (public pages) όπως έχουμε συνηθίσει σε τέτοιου είδους εφαρμογές. Σε αντίθεση με άλλα συστήματα δεν υπάρχει μια κεντρική σελίδα όπου ο χρήστης έχει στη διάθεση του κάποια εργαλεία ή πληροφορίες και από κει και πέρα αν θέλει να αγοράσει κάτι ,ας πούμε να κάνει κάποια εγγραφή και να προχωράει σε μια πιο προσωποποιημένη σελίδα ανάλογα με τον λογαριασμό και τα χαρακτηριστικά του.



Στη προκειμένη περίπτωση αν ο χρήστης δεν είναι εγγεγραμμένος δεν μπορεί να έχει πρόσβαση στην εφαρμογή. Όπου εγγραφή είναι μια διαδικασία αρκετά απλοποιημένη διαδικασία χωρίς password, security question και answer ούτε verification mail και activation. Η διαδικασία της εγγραφής θα επεξηγηθεί περισσότερο στο επόμενο βήμα πάντως αυτό που θα πρέπει να ξέρουμε και να κρατήσουμε είναι ότι αν δεν είμαστε εγγεγραμμένοι δεν έχουμε πρόσβαση στην εφαρμογή η οποία



δεν περιέχει δημόσιες σελίδες.

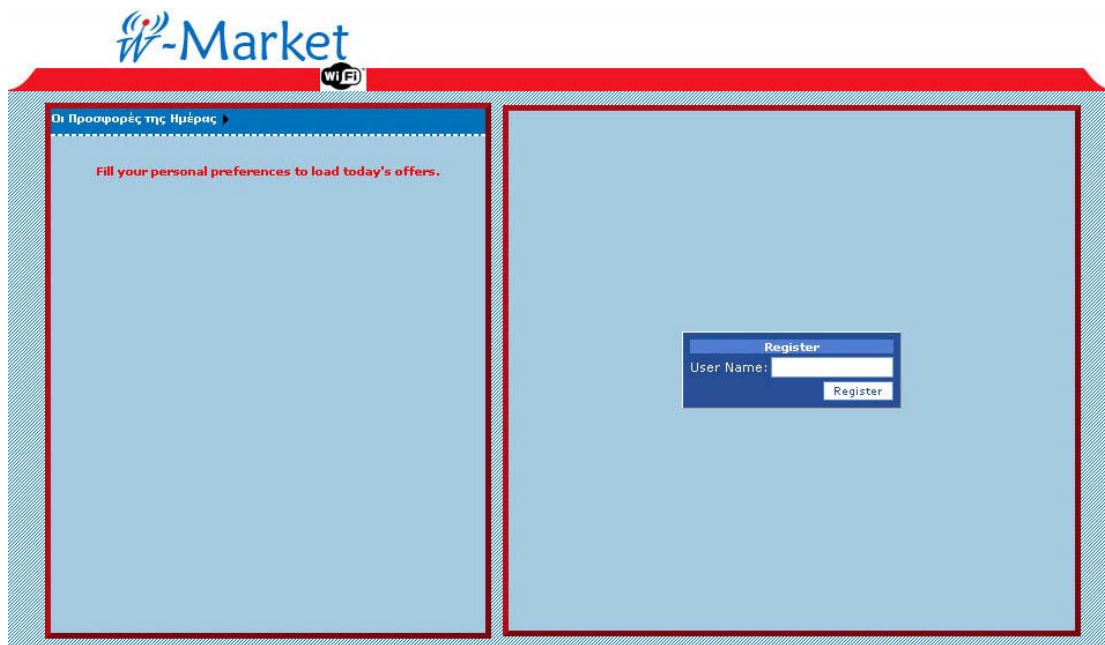
## 2.2.2 Εγγραφή και Διαδικασία Εισαγωγής

Περνάμε τώρα στη διαδικασία εγγραφής και εισόδου στο σύστημά μας. Ας υποθέσουμε ότι είμαστε ένας χρήστης ο οποίος είναι η πρώτη φορά που μπαίνει στην εφαρμογή. Σε αυτό το σημείο θα ήταν καλό να πούμε ότι κάνουμε τη συγκεκριμένη υπόθεση γιατί κάθε χρήστης καλείται να εγγραφεί μόνο τη πρώτη φορά τη οποία εισέρχεται στο σύστημα, από κει και πέρα το σύστημα τον αποθηκεύει στη βάση δεδομένων, αυτόν όπως και τις επιλογές που έχει κάνει σε προτιμήσεις και χαρακτηριστικά.



Τώρα όπως μπορούμε να δούμε και στις εικόνες μετά την αναγνώριση της συσκευής που χρησιμοποιείται από το σύστημα και η προώθηση στην αντίστοιχη προβολή, ο χρήστης βρίσκεται μπροστά σε μία από τις δύο αυτές σελίδες. Σ αυτές τις σελίδες ο χρήστης καλείται να συμπληρώσει μόνο ένα Username δηλαδή ένα όνομα

χρήστη με το οποίο η εφαρμογή θα απευθύνεται σαυτόν. Ο λόγος για τον οποίο για την εγγραφή όπως έχουμε πει και πιο πάνω είναι γιατί η εγγραφή του χρήστη έχει περισσότερο τυπικό ρόλο ύπαρξης και αυτό γιατί η συσκευή του χρήστη και κατά συνέπεια και ο ίδιος ο χρήστης έχουν ταυτοποιηθεί και αναγνωρισθεί από τον μοναδικό αριθμό της κάρτας δικτύου της συσκευής, ένα αριθμό για τον οποίο έχει γίνει και γιαυτόν λόγω πιο πάνω. Τότε ποιος ο λόγος ύπαρξης της διαδικασίας εγγραφής θα αναρωτηθείτε. Ο λόγος είναι απλός ,θέλουμε να δώσουμε στην εφαρμογή ένα χαρακτηριστικό που να αποπέμπει στη μετάβαση σε ένα πιο προσωποποιημένου τρόπου περιήγησης από αυτό το σημείο και μετά στο χρήστη.



Μετά από αυτή τη διαδικασία λοιπόν ο χρήστης αποκτά ένα δικό του μέρος στη βάση δεδομένων της εφαρμογής, τον οποίο εν μέρη έχει δικαίωμα να διαχειρίζεται αυτός αφού εκεί αποθηκεύονται τα προσωπικά του στοιχεία κάθε φορά αλλά και τα αμετάβλητα για τον χρήστη χαρακτηριστικά που χρησιμοποιεί το σύστημα για την αναγνώρισή του. Στο επόμενο βήμα θα αναλύσουμε περαιτέρω τη έννοια και τη λειτουργία του προσωπικού λογαριασμού.



## 2.2.3 Προσωπικός Λογαριασμός

Όπως αναφέραμε και παραπάνω από τη στιγμή που ο χρήστης μπει για πρώτη φορά στο σύστημα καλείται να εισάγει ένα όνομα χρήστη με τον θα έχει πρόσβαση

στην εφαρμογή καταρχάς , θα αναγνωρίζεται από το σύστημα αλλά θα δημιουργείται και μια περιοχή προκειμένου να αποθηκεύονται στοιχεία του αλλά και θα προωθείται από κει και πέρα σε μια πιο προσωποποιημένη περιήγηση. Όλα καλά μέχρι εδώ αλλά ποια ακριβώς είναι η λειτουργία του προσωπικού χώρου, πως ο χρήστης καταλαβαίνει αν ο χρήστης βρίσκεται κάτω από αυτό το περιβάλλον έτσι ώστε να γίνει λάβει όλα τα δικαιώματα που έχει με τη δημιουργία ενός τέτοιου λογαριασμού;

Ας αρχίσουμε λοιπόν από την αναγνώριση της κατάστασης του χρήστη από τον χρήστη. Με το πέρας της διαδικασίας της εγγραφής ο χρήστης προωθείται στη σελίδα

The screenshot shows the W-Market website interface. At the top, there is a logo for W-Market and a 'WiFi' icon. The user is logged in as 'fotis'. The main content is divided into two columns. The left column, titled 'Οι Προσφορές της Ημέρας', lists several products with images and links to view more details: 'ΜΠΑΡΕΣ ΔΗΜΗΤΡΙΑΚΩΝ ΙΔΑΝΙΚΑ ΠΡΟΪΟΝΤΑ ΓΙΑ ΦΥΤΟΦΑΓΟΥΣ', 'ΑΒΓΑ ΤΕΝΕΚΕΤΖΗ ΒΙΟΛΟΓΙΚΑ ΑΥΓΑ', 'ΚΝΟΥΣΙΕΣ ΔΙΧΟΝΔΙΑ ΓΙΑ ΤΟΝ ΣΚΥΛΟ ΣΑΣ 1.5kg', 'DOG LOTION Η ΦΡΟΝΤΙΔΑ ΤΟΥ ΣΚΥΛΟΥ ΣΑΣ', and 'ΦΟΥΡΝΑΚΙ SIEMENS ΙΔΑΝΙΚΗ ΛΥΣΗ ΓΙΑ ΦΟΙΤΗΤΕΣ ΚΑΙ ΕΛΕΥΘΕΡΟΥΣ'. The right column contains text under the heading 'Σημεία Παρουσίας', 'Εγκαταστάσεις', 'Οι άνθρωποι του Ομίλου', and 'Τα προϊόντα'. The 'Σημεία Παρουσίας' section states that the group has 204 stores in Attica and 82 in 40 other major centers in the region. The 'Εγκαταστάσεις' section describes the central warehouse in Mandra, Attica. The 'Οι άνθρωποι του Ομίλου' section mentions 6,000 employees. The 'Τα προϊόντα' section highlights the quality and variety of products offered.

συμπλήρωσης των προσωπικών του προτιμήσεων και στοιχείων , αυτό το βήμα θα επεξηγηθεί στη συνέχεια προς το παρόν ας υποθέσουμε ότι ο χρήστης έχει συμπληρώσει επιτυχώς τα στοιχεία του και αυτά έχουν αποθηκευτεί στο σύστημα και στο προσωπικό του λογαριασμό. Από εκεί ο χρήστης μεταφέρεται στη κεντρική σελίδα της εφαρμογής, η οποία εξορισμού της μεταβάλλεται ανάλογα με τα στοιχεία που ο χρήστης έχει επιλέξει να μοιραστεί.

Ένα από τα πρώτα πράγματα που μπορούμε εύκολα να διακρίνουμε είναι μια φράση στο πάνω δεξιά μέρος της οθόνης (όπως φαίνεται και στις παραπάνω εικόνες) η οποία πληροφορεί τον χρήστη για δύο πράγματα. Καταρχάς με τη φράση "LoggedIn" πληροφορεί τον χρήστη ότι η κατάσταση του είναι συνδεδεμένος και ταυτοποιημένος από το σύστημα και δεύτερον αναγνωρισμένος με το συγκεκριμένο όνομα χρήστη , το οποίο έχουμε εισάγει στη διαδικασία εγγραφής.

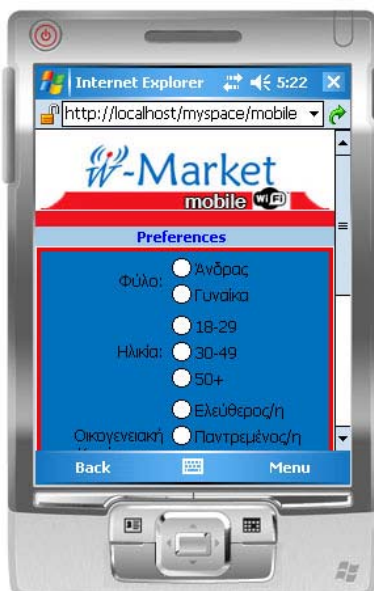


Ένα άλλο χαρακτηριστικό που μπορούμε να διακρίνουμε είναι η προβολή κάποιων προϊόντων – προσφορών στο αριστερό μέρος της σελίδας. Πράγμα το οποίο σημαίνει ότι το σύστημα όχι μόνο έχει όντως λάβει και αποθηκεύσει τις πληροφορίες που του έχουμε εισάγει αλλά τα έχει συγκρίνει και στη συνέχεια φιλτράρει με τα χαρακτηριστικά των προϊόντων που βρίσκονται τη συγκεκριμένη στιγμή σε προσφορά με αποτέλεσμα τη προβολή αυτών των προϊόντων.

## 2.2.4 Παραμετροποίησηση Πληροφοριών

Στο προηγούμενο βήμα είδαμε το πώς κατανοούμε τη κατάσταση στην οποία βρισκόμαστε και το πώς καταλαβαίνουμε αν τα στοιχεία μας έχουν ληφθεί υπόψιν ώστε το σύστημα να μας προβάλλει τα αντίστοιχα προϊόντα σε προσφορά.

Όπως είπαμε όμως πιο πάνω μετά την εγγραφή του χρήστη η εφαρμογή μας προωθεί στη σελίδα εισαγωγής των προσωπικών μας πληροφοριών. Στην παραπάνω εικόνα (φορμα συμπλήρωσης προτιμήσεων σε Laptop) καλούμαστε να συμπληρώσουμε στοιχεία όπως φύλο, ηλικία, οικογενειακή κατάσταση κ.α. ώστε το σύστημα να μπορέσει να φιλτράρει τις προσφορές τις ημέρας με τις οποίες έχει τροφοδοτηθεί.



Σε αυτό το σημείο θα ήθελα να τονίσω ότι τα στοιχεία της φόρμας συμπλήρωσης προσωπικών πληροφοριών είναι εντελώς ενδεικτικά και μπορούν να επεξεργαστούν ποιοτικά και ποσοτικά έτσι ώστε να αντιπροσωπεύουν όσο το δυνατόν πιο καλά τα προϊόντα ή τις υπηρεσίες που προωθεί η εταιρεία. Στη προκειμένη περίπτωση τα συγκεκριμένα στοιχεία κρίνονται επαρκή για το σκοπό αυτής της διπλωματικής λαμβάνοντας βέβαια υπόψιν και τα χαρακτηριστικά των προϊόντων του παραδείγματος , μέχρι κάποιο βαθμό.

Εν προκειμένω ο χρήστης καλείται να συμπληρώσει όλα τα ζητούμενα από το σύστημα στοιχεία για να αρχίσει το φιλτράρισμα. Η διαδικασία αυτή τελειώνει με τη συμπλήρωση όλων των απαιτούμενων στοιχείων και στη συνέχεια το πάτημα του κουμπιού Save με το οποίο γίνεται αυτόματα η αποθήκευση αυτών των δεδομένων στο σύστημα, το φιλτράρισμα των προσφορών και η προώθηση του χρήστη στη προσωποποιημένη κεντρική σελίδα όπου γίνεται και η προβολή των προϊόντων που αντιστοιχούν όσο το δυνατόν στα στοιχεία του χρήστη, μια σελίδα που είδαμε και στο προηγούμενο βήμα.

## 2.2.5 Προβολή Λεπτομερειών

Είδαμε πως μπορούμε να εγγραφούμε στο σύστημα και να αποκτήσουμε πρόσβαση σε αυτό , ακόμα είδαμε πως μπορούμε να συμπληρώσουμε τα στοιχεία μας προκειμένου να μας προβληθούν οι αντίστοιχες πληροφορίες. Τώρα θα δούμε

The screenshot shows the W-Market website interface. At the top, there is a logo for 'W-Market' and a 'WiFi' icon. The user is logged in as 'fotis'. The main content area is divided into two columns. The left column, titled 'Οι Προσφορές της Ημέρας', lists several products with images and brief descriptions: 'ΜΠΑΡΕΣ ΔΗΜΗΤΡΙΑΚΩΝ', 'ΑΒΓΑ ΤΕΝΕΚΕΤΖΗ', 'ΚΝΟΥΣΙΕΣ', 'DOG LOTION', and 'ΦΟΥΡΝΑΚΙ SIEMENS'. The right column displays the details for the selected product, 'ΑΒΓΑ ΤΕΝΕΚΕΤΖΗ'. It includes a product image, the 'Product ID : 54', the product name 'ΑΒΓΑ ΤΕΝΕΚΕΤΖΗ', and the price '3,40€ 2,20€'. Below the image, there is a short description in Greek: 'Τα Αβγά Τενεκετζή, είναι βιολογικά αβγά από άρνιθες αγροκτήματος, που τρέφονται με υψηλής ποιότητας δημητριακά βιολογικής γεωργίας, πιστοποιημένα από διεθνείς και εθνικούς φορείς πιστοποίησης ποιότητας'.

από τη στιγμή που το σύστημα έχει προσωποποιηθεί και προσαρμοστεί στα δικά μας δεδομένα πως αντλούμε πληροφορίες από αυτό. Καταρχάς έχουμε αναφέρει, αλλά μπορούμε να δούμε και στις εικόνες ότι τα προϊόντα που επαληθεύουν τα στοιχεία μας προβάλλονται στα αριστερά της σελίδας μας.

Με μια πρώτη ματιά μπορούμε να διακρίνουμε τη φωτογραφία του προϊόντος, ακριβώς δίπλα της η επικεφαλίδα της προσφοράς η οποία μπορεί να περιέχει απλά το όνομα του προϊόντος ,από κάτω από τον τίτλο της προφοράς βλέπουμε μια σύντομη περιγραφή για τη προσφορά και τέλος μια υπερσύνδεση η οποία αποτι μπορούμε να διακρίνουμε είναι τονισμένη και με κόκκινη γραμματοσειρά και η οποία μας προωθεί σε μία άλλη σελίδα όπου περιέχονται περισσότερες πληροφορίες σχετικά με το προϊόν που έχουμε διαλέξει να προβάλουμε. Το ίδιο μοτίβο ακολουθούν όλο τα προϊόντα στη λίστα, αλλά και τα προϊόντα τα οποία ενδεχόμενος είναι σε προσφορά αλλά στο φιλτράρισμα έχουν αποκλειστεί σαν υποψήφιες προσφορές για το προσωπό μας.

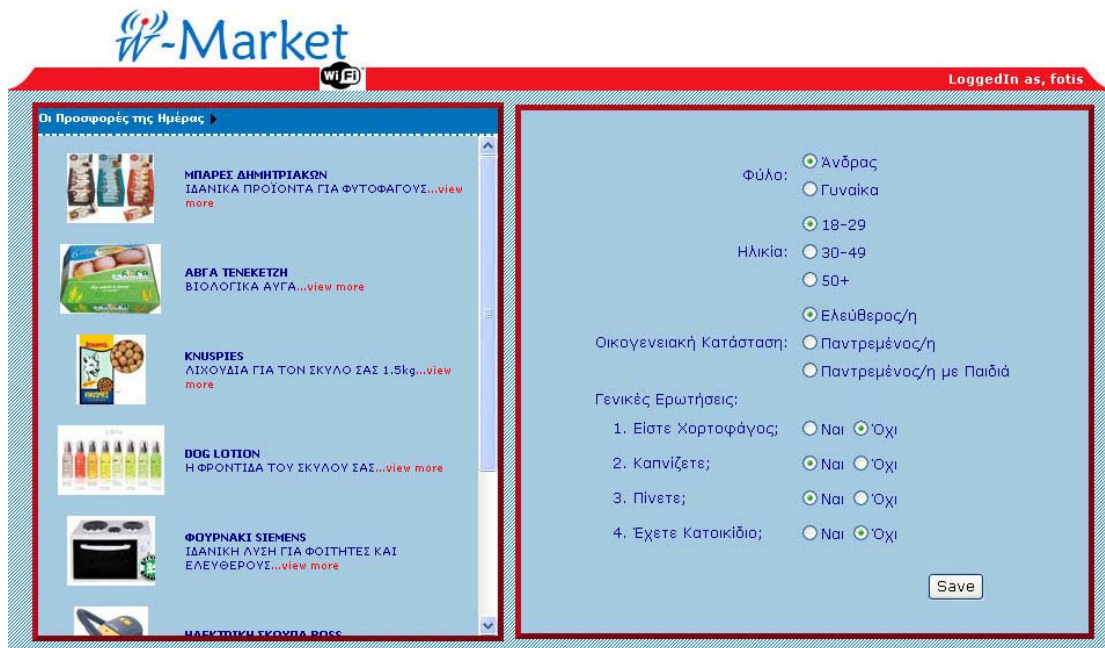


Στη σελίδα των λεπτομερειών τώρα μπορούμε να δούμε την προηγούμενη εικόνα του προϊόντος, σε προσφορά που έχουμε επιλέξει, σε μεγαλύτερη ανάλυση. Δίπλα από την εικόνα του προϊόντος μπορούμε να δούμε τον κωδικό του προϊόντος, ο οποίος έχει οριστεί να είναι ίδιος με τον κωδικό της προσφοράς. Από κάτω τώρα υπάρχει ο τίτλος της προσφοράς ,που μπορεί να είναι απλά το όνομα του προϊόντος. Κάτω από τον τίτλο τώρα βλέπουμε δύο τιμές η μία έχει μαύρο χρώμα γραμματοσειράς και η δεύτερη κόκκινο. Η μεν πρώτη αντιπροσωπεύει τη τιμή του προϊόντος κανονικά , όταν αυτό δε βρίσκεται σε προσφορά, γιαυτό είναι και διαγραμμένη, ενώ η δεύτερη τιμή

αντιπροσωπεύει την τιμή της προσφοράς, την τιμή που ισχύει δηλαδή γιαυτό και είναι τονισμένη χρωματικά.

Τέλος στην ίδια σελίδα πάντα στο κάτω μέρος της δεξιάς πλευράς της σελίδας οι πληροφορίες για το προϊόν κλείνουν με μία εκτεταμένη περιγραφή του προϊόντος, η οποία μπορεί να περιέχει κάποιες επιπλέον πληροφορίες όσον αφορά στη προσφορά είτε είναι αυτό τιμή, ποιότητα και ποσότητα.

## 2.2.6 Χειρισμός Παραμετροποίησης



Μπορεί κανείς να σκεφτεί, ωραία συμπληρώνουμε τα στοιχεία μας, αποθηκεύονται, φιλτράρει το σύστημα τις προσφορές και προβάλλει εκείνες που πιστεύει ότι είναι πιο πιθανό να ενδιαφερθούμε για την αγορά τους ουσιαστικά. Τι γίνεται όμως στη περίπτωση που κάποια στοιχεία από αυτά που έχουμε δώσει στο σύστημα πάψουν πια να μας αντιπροσωπεύουν ή απλά θέλουμε να δούμε και άλλες προσφορές για προϊόντα τα οποία σκοπεύουμε να τα δωρίσουμε σε ένα άλλο πρόσωπο, για παράδειγμα. Αυτή η περίπτωση έχει προβλεφθεί κατά το σχεδιασμό και τη κατασκευή της εφαρμογής, μάλιστα επιτυγχάνεται με πολύ εύκολο τρόπο από τη πλευρά του χρήστη.

Ας υποθέσουμε λοιπόν ότι είμαστε εγγεγραμμένοι έχουμε συμπληρώσει τα στοιχεία μας στη σχετική φόρμα και βρισκόμαστε στη κεντρική σελίδα, όπου το σύστημα προβάλλει τις προσφορές για λογαριασμό μας. Πάνω από τη λίστα των προσφορών υπάρχει η υπερσύνδεση με την ονομασία "Αλλαγή Προσωπικών Προτιμήσεων". Είναι μια υπερσύνδεση που έχουμε κάνει λόγο για αυτή και σε προηγούμενες παραγράφους. Θα ήθελα επίσης να προσθέσω σαυτό το σημείο ότι η πρόσβαση στη συγκεκριμένη υπερσύνδεση δεν είναι σαυτή μόνο από τη κεντρική σελίδα της εφαρμογής μας. Ο χρήστης έχει τη δυνατότητα να αλλάξει τις προτιμήσεις του σε οποιοδήποτε σημείο της εφαρμογής αρκεί να είναι συνδεδεμένος και εγγεγραμμένος.

Πατώντας λοιπόν στην υπερσύνδεση το σύστημα μας προωθεί στη σελίδα αλλαγής – εισαγωγής των προσωπικών μας στοιχείων και για την οποία έχει γίνει πάλι λόγος. Εκεί λοιπόν και έχοντας πάντα στην



αριστερή μεριά τα προϊόντα που αντιστοιχούν στα προηγούμενα στοιχεία που έχουμε εισάγει, μπορούμε να συμπληρώσουμε τα καινούρια στοιχεία που θέλουμε να εισάγουμε. Αφού συμπληρώσουμε τη φόρμα για μια ακόμα φορά με τα καινούρια στοιχεία και πατήσουμε το κουμπί Save ,μεταφερόμαστε ξανά στη κεντρική σελίδα με τη διαφορά όμως ότι αυτή τη φορά στη λίστα των προσφορών περιέχονται οι ανανεωμένες προσφορές , που έχουν προκύψει από τη γνωστή διαδικασία.

The screenshot shows the W-Market website interface. At the top, there is a navigation bar with the W-Market logo, a Wi-Fi icon, and a user login status 'Logged In as, fotis'. Below the navigation bar, the page is divided into two main sections. The left section, titled 'Οι Προσφορές της Ημέρας', contains a vertical list of five promotional items, each with a small image and a 'view more' link. The items are: 1. ΠΡΟΦΥΛΑΚΤΙΚΑ DUO (H ΑΠΟΛΥΤΗ ΑΣΦΑΛΕΙΑ ΣΤΟ ΣΕΧ...), 2. ΚΑΠΝΟΣ OLDHOLBORN 3τημ. ΠΡΟΣΦΟΡΑ ΤΑ 3 ΤΕΜΑΧΙΑ..., 3. ΑΝΑΠΗΡΑΣ PIERRE CARDIN ΚΑΝΤΕ DVRO ΣΤΟΝ ΕΑΥΤΟ ΣΑΣ ΕΝΑ ΕΡΓΟ ΤΕΧΝΗΣ..., 4. ΦΟΥΡΝΑΚΙ SIEMENS ΙΔΑΝΙΚΗ ΛΥΣΗ ΓΙΑ ΦΟΙΤΗΤΕΣ ΚΑΙ ΕΛΕΥΘΕΡΟΥΣ..., and 5. ΑΝΔΡΙΚΟ ΑΡΩΜΑ DAVIDOFF ΕΚΦΡΑΣΤΕ ΤΟ ΔΥΝΑΜΙΣΜΟ ΣΑΣ ΧΩΡΙΣ ΛΟΓΙΑ... The right section contains a detailed article with the following sections: 'Σημεία Παρουσίας' (describing 204 stores in Attica and Crete), 'Εγκαταστάσεις' (describing the central warehouse in Mandra, Attica), 'Οι άνθρωποι του Ομίλου' (describing 6,000 employees), and 'Τα προϊόντα' (describing the product range).

## 2.2.7 Διαδικασία Εξόδου

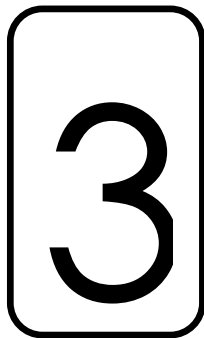
Συνδεθήκαμε με το σύστημα εισάγαμε ότι στοιχεία μας ζητήθηκαν να εισάγουμε , περιηγηθήκαμε στην εφαρμογή είδαμε τα προϊόντα που βρισκόντουσαν σε προσφορά και καταλήξαμε αν ενδιαφερόμαστε να αγοράσουμε κάτι ή όχι. Πώς βγαίνουμε τώρα από την εφαρμογή, πώς αποσυνδεόμαστε;

Η συγκεκριμένη διαδικασία είναι η πιο απλή διαδικασία στο σύστημα και λειτουργεί με έναν ευκολονόητο τρόπο. Όταν έρθει η στιγμή να αποσυνδεθούμε από την εφαρμογή απλά κλείνουμε το παράθυρο του φυλλομετρητή (browser) στο οποίο έχει φορτωθεί η εφαρμογή και είχε ανοίξει αυτόματα με την είσοδό μας στο δίκτυο.

Θα περιμένετε να υπάρχει μια διαδικασία log off όπως και στα περισσότερα παρόμοια συστήματα που λειτουργούν με αυτό το τρόπο. Όπως έχουμε πει όμως η εφαρμογή αυτή δε συμβαδίζει με τα κοινά αποδεκτά στάνταρ κατασκευής τέτοιου είδους εφαρμογών. Έτσι λοιπόν το κλείσιμο του browser κάνει ακριβώς την ίδια δουλειά με το log off.



Όσον αφορά στα στοιχεία μας τώρα αυτά δεν διατρέχουν κανένα κίνδυνο καθώς αποθηκεύονται με την συμπλήρωσή τους, μάλιστα το σύστημα καθιστά δυνατή την αυτόματη χρήση και προβολή τους την επόμενη φορά που θελήσουμε να συνδεθούμε σε αυτό.



## ΕΓΧΕΙΡΙΔΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ

Στο προηγούμενο κεφάλαιο κάναμε λόγο για το τρόπο με τον οποίο ο απλός χρήστης μπορεί να χρησιμοποιήσει το σύστημα. Δε δώσαμε έμφαση στις λεπτομέρειες, προσπαθήσαμε απλά να περιγράψουμε με λίγα λόγια πως μπορεί κάποιος χωρίς μεγάλη εμπειρία σε τέτοιου είδους εφαρμογές να μπορέσει να καταλάβει τη έκταση του, τη δομή του και κυρίως τη λειτουργία του.

Σε αυτό το κεφάλαιο, το οποίο απευθύνεται σε πιο έμπειρους όχι απλώς χρήστες αλλά και κατασκευαστές τέτοιων εφαρμογών, θα εμβαθύνουμε στην εφαρμογή. Δε θα μείνουμε μόνο στο οπτικό αποτέλεσμα, αλλά θα ανατρέξουμε στη ρίζα του και θα πάρουμε μια γεύση από τη σχεδίαση καταρχάς της εφαρμογής, η οποία θα περιέχει και τη σύσταση της βάσης δεδομένων.

Στη συνέχεια θα περάσουμε στο προγραμματιστικό κομμάτι της εφαρμογής θα επεξηγήσουμε βήμα προς βήμα τις κλάσεις, τις μεταβλητές, τα sessions, τα cookies και γενικά όλα στοιχεία του προγραμματισμού που μπορούν να κάνουν πραγματικότητα ένα τέτοιο σύστημα.

Ένα άλλο κομμάτι της κατασκευής της εφαρμογής με το οποίο θα ασχοληθούμε είναι ο έλεγχος, ο οποίος θα απεικονιστεί σε διαγράμματα χρήσης κυρίως για ευκολότερη κατανόηση. Το τελευταίο κομμάτι αυτού του κεφαλαίου έχει να κάνει με τη διαχείριση του συστήματος το administration δηλαδή. Θα δούμε πώς επεξεργαζόμαστε τους χρήστες,

τα προϊόντα και θα πούμε και μερικά πράγματα για την αλλαγή εμφάνισης της εφαρμογής.

## 3.1 Ανάλυση και Σχεδίαση Βάσης Δεδομένων

Ξεκινώντας το μέρος που απευθύνεται στους προγραμματιστές τέτοιων εφαρμογών θα αναλύσουμε καταρχάς το σχεδιασμό και τη κατασκευή της βάσης δεδομένων της εφαρμογής. Ένα από τα βασικότερα κομμάτια τα οποία είναι βαρόμετρο για τη σωστή λειτουργία όλου του συστήματος.

Ας αρχίσουμε λοιπόν να αναλύσουμε το πώς δημιουργήσαμε τους πίνακες για την εφαρμογή του intranet. Οι πίνακες μπορούν να θεωρηθούν ως συρτάρια σε ένα ντουλάπι αρχειοθέτησης, ακριβώς όπως θα μπορούσαμε να χωρίσουμε τις διαφορετικές πληροφορίες στα διαφορετικά συρτάρια μέσα στο ντουλάπι αρχειοθέτησής μας, μπορούμε να σπάσουμε τις πληροφορίες για πελάτες, προϊόντα, προσφορές κατηγορίες και υποκατηγορίες προϊόντων στους διαφορετικούς πίνακες. Οι πίνακες μπορούν επίσης συγκριθούν με τους υπολογισμούς σε λογιστικό φύλλο (spreadsheet), δεδομένου μιας και έχουν σειρές και τις στήλες, αλλά έχουν και πολλά άλλα βασικά χαρακτηριστικά γνωρίσματα. Γνωρίζουν ποια είδη στοιχείων επιτρέπονται να αποθηκευτούν, και μπορούν να αφορούν τα στοιχεία που περιλαμβάνονται σε άλλους πίνακες, αλλά επίσης να αναζητηθούν και να χειριστούν με μια πολύ ισχυρή γλώσσα όπως η SQL.

Μπορούμε να οργανώσουμε τους πίνακες στη βάση δεδομένων μας χρησιμοποιώντας είτε τον Visual Web Developer είτε το SQL Server Management Studio, ανάλογα με την προτίμησή μας. Ενώ το SQL Server Management Studio είναι ισχυρότερο, και τα δύο εργαλεία μπορούν να χρησιμοποιηθούν για το βασικό στόχο όπως η δημιουργία των πινάκων βάσεων δεδομένων. Επίσης ένας άλλος τρόπος δημιουργίας των πινάκων της βάσης είναι και μέσα από το Visual Studio, μιας και η εφαρμογή μας προσφέρει αυτή τη δυνατότητα.

Εμβαθίνοτας τώρα στο πώς δημιουργούμε τον πίνακά μας. Προτού όμως, αξίζει να κάνουμε μια σκέψη για το πόσους πίνακες η εφαρμογή μας θα χρειαστεί, και τι ακριβώς θα περιέχουν. Μπορούμε να σκεφτούμε τους πίνακες ως καταλόγους οντοτήτων. Οι οντότητες είναι οι σειρές ή τα αρχεία στον πίνακά μας. Το στρέψιμο της προσοχής των πινάκων μας και των οντοτήτων τους σε χαρτί είναι ένας καλός τρόπος για να προγραμματιστεί το λογικό σχέδιο της βάσης δεδομένων. Το λογικό σχέδιο παρουσιάζει ποια είδη από τα στοιχεία η βάση δεδομένων μας

θα πρέπει να αποθηκεύσει, και περιγράφει τις σχέσεις που εμείς θέλουμε να υπάρξουν μεταξύ των συγκεκριμένων κομματιών των στοιχείων.

Εντούτοις, αντίθετα από ένα χαρακτηριστικό αρχείο υπολογισμών με λογιστικό φύλλο (spreadsheet), οι πίνακες που καθορίζονται στο λογικό σχέδιο δεν αντιπροσωπεύουν συνήθως τον τρόπο που θα αποθηκεύσουμε τα στοιχεία στη βάση δεδομένων. Αυτό λαμβάνεται προσοχή στη φυσική σχεδίαση, στην οποία δημιουργούμε ένα πρακτικό σχεδιάγραμμα που μας επιτρέπει να βελτιώσουμε την ταχύτητα των βάσεων δεδομένων, να επιτρέψει τη βελτιστοποίηση των σχέσεων μεταξύ διαφορετικών οντοτήτων, ή την εφαρμογή άλλων εξελιγμένων γνωρισμάτων, για να βελτιστοποιήσει τη βάση δεδομένων μας με τους διάφορους τρόπους.

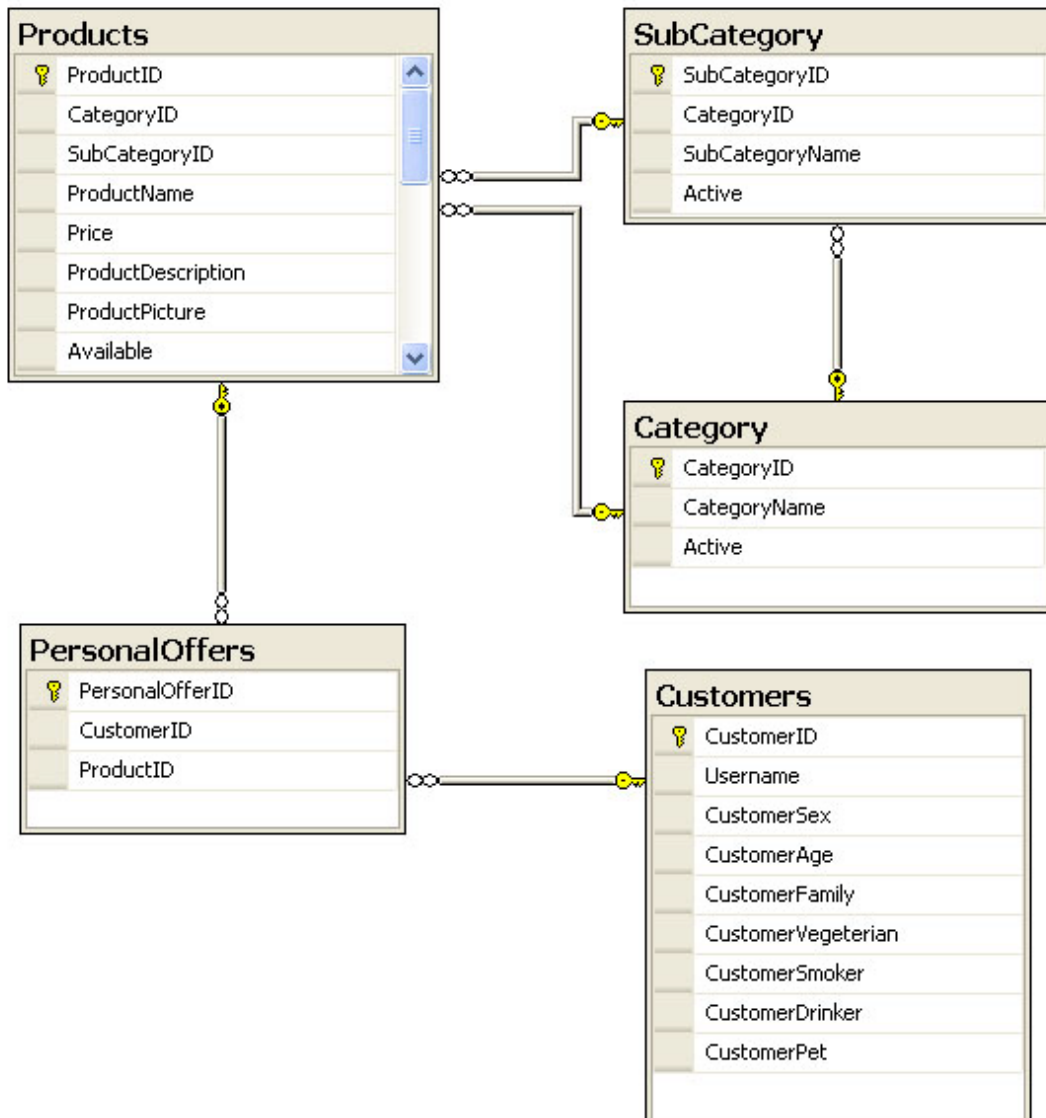
Το σχέδιο της βάσης δεδομένων μας έχει σημαντικές συνέπειες στον τρόπο που η εφαρμογή λειτουργεί, και στην ευκολία να επεκταθεί, έτσι είναι σημαντικό να δώσουμε στον λογικό και φυσικό σχεδιασμό προσοχή.

Ο τρόπος που θα σχεδιάζατε φυσικά τη βάση δεδομένων σας σε χαρτί, και ο καλύτερος τρόπος να εφαρμοστεί το σχέδιο στην πράξη, μπορούν να είστε δύο διαφορετικά πράγματα. Εντούτοις, λάβετε υπόψη ότι δεν υπάρχει κανένας κανόνας στο σχέδιο βάσεων δεδομένων, και οι ειδικοί σχεδιαστές βάσεων δεδομένων κάμπτουν μερικές φορές ή προσπερνούν κανόνες για να καλυφθούν οι απαιτήσεις σε ιδιαίτερες περιπτώσεις.

Εντούτοις, το σημαντικότερο χαρακτηριστικό αυτού του σχεδίου βάσεων δεδομένων είναι ότι, αφού αποθηκεύουμε τις πληροφορίες για δύο διαφορετικούς τύπους οντοτήτων (πελάτες και προϊόντα), χρησιμοποιούμε δύο πίνακες. Αυτή η προσέγγιση επεξηγεί έναν σημαντικό που πρέπει να λάβουμε υπόψη κατά το σχεδιασμό των βάσεων δεδομένων:

*Σε κάθε τύπο οντότητας για τον οποίο θέλουμε να είμαστε σε θέση να αποθηκεύσουμε τις πληροφορίες πρέπει να δοθεί ένας πίνακας.*

Με αυτόν τον κανόνα στο μυαλό, μπορούμε να σκεφτούμε για την εφαρμογή που θέλουμε να χτίσουμε. Πρέπει να σκεφτούμε το σχέδιο από άποψη οντοτήτων που θέλουμε να ακολουθήσουμε, και να βρούμε έναν προκαταρκτικό κατάλογο πινάκων. Όπως μπορούμε να δούμε και στη παρακάτω εικόνα του διαγράμματος της βάση δεδομένων, έχει γίνει ο βασικός σχεδιασμός της βάσης όπως αυτός περιγράφεται από τους πίνακες και τις σχεσιακές τους αλληλεξαρτήσεις.



Τα βασικά κομμάτια της βάσης μας λοιπόν αποτελούνται από τους παρακάτω πίνακες:

### Προϊόντα

Αυτός ο πίνακας περιέχει τα προϊόντα της επιχείρησης, όπως επίσης και περαιτέρω πληροφορίες για αυτά σχετικά με το αν είναι σε προσφορά ή όχι, με το γκρουπ των προτιμήσεων των πελατών που απευθύνονται, τη τιμή τους και άλλες σχετικές πληροφορίες.

### Κατηγορίες Προϊόντων

Οι κατηγορίες που είναι διαθέσιμες για τα προϊόντα της εταιρείας (οπωροπωλείο, κρεοπωλείο, κ.λ.π.) αποθηκεύονται σε αυτόν το πίνακα και συνδέονται με τους πίνακες των προϊόντων και των υποκατηγοριών.

### Υποκατηγορίες Προϊόντων

Αυτός ο πίνακας αποθηκεύει τις υποκατηγορίες των προϊόντων της εταιρείας (φρούτα , λαχανικά , ψάρια, κ.λ.π.) που είναι διαθέσιμες και συνδέεται με τους πίνακες των κατηγοριών και των προϊόντων.

### **Πελάτες**

Αυτός ο πίνακας παρακολουθεί τους πελάτες της επιχείρησής μας, κάθε φορά που κάποιος από αυτούς εισέρχεται στο σύστημά μας. Περιέχει πληροφορίες για την αναγνώριση και ταυτοποίησή τους αλλά και τα στοιχεία που έχουν εισάγει όσον αφορά στα προσωπικά τους στοιχεία και χαρακτηριστικά.

### **Προσωπικές Προσφορές**

Αυτός ο πίνακας περιέχει τα προϊόντα που βρίσκονται σε προσφορά κάτω από το χαρακτηριστικό κάθε πελάτη και χρησιμεύει στο να θυμάται το σύστημα τις προσφορές που έχουν προβληθεί σε κάθε πελάτη και να τις εμφανίζει με το που ξανασυνδεθεί ο χρήστης σε αυτό.

Ο ορισμός και η ανάλυση των στοιχείων των πληροφοριών που πρέπει να σωθούν είναι το πρώτο βήμα στον καθορισμό του σχεδίου - αυτή η διαδικασία της βάσης δεδομένων αντιπροσωπεύει τη λογική φάση που σχεδίου που αναφέραμε νωρίτερα. Μέσω αυτής της διαδικασίας, εργαζόμαστε ώστε να κατασκευάσουμε υψηλού επιπέδου στοιχεία που πρέπει να σωθούν. Με αυτό το καθορισμό θα μπορέσετε έπειτα να μετασχηματιστείτε τη βάση σε μια φυσική δομή του σχεδίου, η οποία θα περιέχει τις απαιτούμενες λεπτομέρειες που θα εφαρμόσει η βάση δεδομένων.

Δεδομένου ότι αναλύσαμε το στοιχείο που πρέπει να αποθηκευτεί ,όπως το στοιχείο προϊόντα, πελάτες οι οποίες δεν είναι προφανείς οντότητες της αίτησής μας στο τρέχον σχέδιο. Εντούτοις, θυμηθείτε ότι όποτε προβλέπετε ότι η βάση δεδομένων σας να περιέχει έναν τομέα που πρέπει μόνο να δεχτεί τις τιμές από έναν συγκεκριμένο κατάλογο, καλό θα είναι να δημιουργήσει έναν πίνακα για να κρατήσει εκείνο τον κατάλογο. Αυτή η προσέγγιση το καθιστά εύκολο να εκτελέσει αλλαγές στον κατάλογο στο μέλλον και μειώνει επίσης στο χώρο δίσκου που απαιτείται στη βάση δεδομένων σας, και βοηθάει εσάς να αποφύγει τον πλεονασμό, δεδομένου ότι αποθηκεύετε μόνο τις ενιαίες περιπτώσεις από τα ονόματα κατηγοριών.

Αυτή η διαδικασία έξω από τις οντότητες, τους πίνακες, και τις σχέσεις μεταξύ πινάκων για να περιορίσουν τα προβλήματα συντήρησης και τα περιττά στοιχεία καλείται κανονικοποίηση βάση δεδομένων. Αν και θα μιλήσω λίγο περισσότερο για την κανονικοποίηση πριν από τέλος αυτού του κεφαλαίου. Όπως και οποιοσδήποτε ειδικός πληροφορικής θα σας πει, ο σχεδιασμός βάσεων δεδομένων είναι ένας πολύ σοβαρός

τομέας της έρευνας, με δοκιμασμένες και μαθηματικά αποδεδειγμένες αρχές.

Έτσι, έχουμε τον κατάλογο πινάκων μας. Στο επόμενο τμήμα, θα εξετάσουμε τις στήλες μέσα σε εκείνους τους πίνακες, και θα συζητήσουμε πώς μπορούμε να εξακριβώσουμε τα χαρακτηριστικά τους.

Το επόμενο βήμα είναι να αποφασίσουμε ποια κομμάτια της πληροφορίας θα περιληφθούν μέσα σε εκείνους τους πίνακες. Για παράδειγμα, μπορείτε να θελήσετε περιλάβετε ένα όνομα, επώνυμο, τηλέφωνο, διεύθυνση, πόλη, κράτος, για όλους τους πελάτες στον πίνακα πελατών.

Μια από τις διαφορές μεταξύ του λογικού σχεδίου και του φυσικού σχεδίου είναι ότι όταν προγραμματίζουμε το φυσικό σχέδιο της βάσης δεδομένων, πρέπει να εξετάσουμε λεπτομέρειες σαν τον τύπο των στοιχείων. Έτσι όπως με τα στοιχεία που αποθηκεύουμε σε VB.NET έτσι και τα στοιχεία που αποθηκεύουμε σε κάθε μια από τις στήλες των πινάκων μας έχουν έναν ιδιαίτερο τύπο στοιχείων. Ο SQL Server ξέρει πολλούς τύπους στοιχείων.

Στη συνέχεια θα αναφερθούμε στα είδη πληροφορίας τα οποία χρησιμοποιήσαμε για να αποθηκεύσουμε τα δεδομένα μας στη βάση.

#### **int**

Χρησιμοποιήστε τον τύπο στοιχείων `int` όταν πρέπει να αποθηκεύσετε ολόκληρους τους ακέραιους αριθμούς. Αυτός ο τύπος στοιχείων μπορέστε να αποθηκεύσετε τους αριθμούς από -2.147.483.648 έως 2.147.483.647.

#### **money**

Ο τύπος στοιχείων `money` πρέπει να χρησιμοποιηθεί για να αποθηκεύσει τα νομισματικά στοιχεία, όπως οι τιμές για έναν κατάλογο προϊόντων. Αυτός ο τύπος στοιχείων είναι στενά συνδεδεμένος στον τύπο στοιχείων `int`.

#### **bit**

Χρησιμοποιήστε τον τύπο στοιχείων κομματιών όταν ένας όρος είναι είτε αληθινός (αντιπροσωπευόμενος ως 1) είτε ψεύτικος (αντιπροσωπευόμενος ως 0).

#### **datetime**

Όπως είναι δυνατόν να υποθέσετε, ο `datetime` τύπος στοιχείων χρησιμοποιείται για να αποθηκεύσει τις ημερομηνίες και χρόνους. Είναι πολύ χρήσιμο όταν θέλετε να ταξινομήσετε τα στοιχεία στον πίνακά σας χρονολογικά.

**nvarchar (v)**

Ο nvarchar τύπος στοιχείων αποθηκεύει τις σειρές του κειμένου. Συνηθέστερα χρησιμοποιείται για τα στοιχεία που δακτυλογραφούνται επειδή αποθηκεύουν τα ονόματα και τις περιγραφές. Κατά την καθορισμό μιας στήλης αυτού του τύπου, πρέπει επίσης να διευκρινίσουμε ένα μέγιστο μέγεθος σε παρένθεση οι μακρύτερες σειρές θα τακτοποιηθούν για να εγκαταστήσουν το καθορισμένο μέγεθος. Παραδείγματος χάριν, nvarchar (50) διευκρινίζει έναν τομέα που μπορεί να κρατήσει μέχρι 50 χαρακτήρες. Το VAR μέρος του nvarchar ονόματος προέρχεται από το γεγονός που αυτός ο τύπος στοιχείων μπορεί να αποθηκεύσει σειρές του μεταβλητού μήκους μέχρι το διευκρινισμένο μέγιστο.

Μερικές φορές, μπορείτε να δείτε στις κακοσχεδιασμένες βάσεις δεδομένων να χρησιμοποιείτε float για να αποθηκεύονται νομισματικά στοιχεία. Δεδομένου ότι το επιπλέον σώμα είναι ευαίσθητο στη στρογγυλοποίηση των λαθών, αυτό είναι μια κακή ιδέα. Το στοιχείο money , από την άλλη δεν είναι ευαίσθητο σε αυτά τα λάθη και είναι μια πολύ καλύτερη επιλογή.

Οι τύποι στοιχείων των SQL Servers, όπως με τις άλλες λέξεις κλειδιά των SQL Servers, δεν διαχωρίζουν κεφαλαία και μικρά (case sensitive). Τα nvarchar και nchar ονομάζονται ξαδέλφια του μη-Unicode varchar και char, τα οποία μπορείτε να χρησιμοποιήσετε εάν είστε βέβαιοι δεν θα πρέπει να αποθηκεύσετε στοιχεία Unicode. Μπορείτε εν ανάγκη να χρησιμοποιήσετε Unicode (ή μια συγκεκριμένη γλωσσικά μορφή κωδικοποίησης) κατά την αποθήκευση μη Αγγλικού κειμένου, όπως τα κινέζικα, Αραβικά, και άλλα. Το Unicode είναι πολύ ευρέως υποστηριζόμενο πρότυπο, έτσι αυτό συστήνεται έντονα ότι κολλάτε με nvarchar και nchar.

Ο τύπος μιας στήλης καθορίζει πώς εκείνη η στήλη συμπεριφέρεται. Παραδείγματος χάριν, η ταξινόμηση των στοιχείων από μια datetime στήλη θα αναγκάσουν τα αρχεία να ταξινομηθούν χρονολογικά, αντί αλφαβητικά ή αριθμητικά.

Επίσης θα ήθελα να αναφέρω μια ιδιότητα στήλης που έχει χρησιμοποιηθεί ευρέως στην εφαρμογή.

**IDENTITY**

Οι στήλες identity είναι αριθμημένες αυτόματα. Εάν θέτετε μια στήλη ως στήλη identity, ο SQL Server θα παραγάγει τους αριθμούς αυτόματα για αυτήν τη στήλη δεδομένου ότι προσθέτετε νέες σειρές σε αυτήν. Ο πρώτος αριθμός στη στήλη καλείται ο σπόρος ταυτότητας. Για να παραγάγει τους επόμενους αριθμούς, η στήλη ταυτότητας προσθέτει

μια δεδομένη αξία στο σπόρο η αξία που προστίθεται καλείται *identity increment*. Εξ ορισμού, ο σπόρος και η αύξηση έχουν μια αξία 1, όπου σε κάθε περίπτωση οι παραγόμενες τιμές είναι 1, 2, 3, και τα λοιπά. Εάν ο σπόρος ταυτότητας ήταν 5 και η αύξηση ταυτότητας ήταν 10, οι παραγόμενοι αριθμοί θα ήταν 5, 15, 25, και τα λοιπά. Η *IDENTITY* είναι χρήσιμη για τις στήλες ταυτότητας, όπως η ταυτότητα προϊόντος, για την οποία εσείς δε χρειάζεται να φροντίστε τις τιμές, εφ' όσον είναι μοναδικές. Όταν χρησιμοποιείτε *identity*, οι παραγόμενες τιμές θα είναι πάντα μοναδικές. Εξ ορισμού, δεν μπορείτε διευκρινίστε τις τιμές για μια στήλη *IDENTITY*. Σημειώστε επίσης ότι η στήλη δεν μπορεί ποτέ να είναι άδεια.

Τα πρωτεύοντα κλειδιά είναι η τελευταία θεμελιώδης έννοια για την οποία πρέπει να κάνουμε λόγο πριν περάσουμε στην ανάλυση των πινάκων μας. Στον κόσμο των σχεσιακών βάσεων δεδομένων, κάθε σειρά σε έναν πίνακα πρέπει να προσδιοριστεί μεμονωμένα από μια στήλη αποκαλούμενη κλειδί, στην οποία όλες οι διαδικασίες βάσεων δεδομένων είναι βασισμένες.

Οι πίνακες στις βάσεις δεδομένων μας θα μπορούσαν να περιέχουν εκατοντάδες ή ακόμα και χιλιάδες σειρές από παρόμοια στοιχεία - θα μπορούσαμε να έχουμε αρκετούς πελάτες μόνο σε ένα πίνακα. Φανταστείτε ότι το πρόγραμμά μας πρέπει να ενημερώσει ή να διαγράψει κάποιο αρχείο, αν ας πούμε υπάρχουν διάφοροι άνθρωποι με εκείνο το όνομα στο πελατολόγιο μας. Δεν θα μπορούσαμε να στηριχθούμε στη βάση δεδομένων για να βρούμε την εγγραφή για το συγκεκριμένο αρχείο και θα υπήρχε πιθανότητα να καταλήξουμε να ενημερώσουμε ή να διαγράψουμε λανθασμένο αρχείο.

Μπορούμε να αποφύγουμε αυτά τα προβλήματα μόνο με τη χρησιμοποίηση ενός συστήματος που μεμονωμένα προσδιορίζει κάθε σειρά στον πίνακα. Το πρώτο βήμα προς την επίτευξη αυτού του στόχου είναι προσθέσουμε στον πίνακα μια στήλη ταυτότητας που παρέχει έναν μοναδικό στοιχείο για κάθε πελάτη, όπως η στήλη *CustomerID*.

Θυμηθείτε ότι όταν συζητήσαμε για τον πίνακα πελατών, σημειώσαμε ότι μπορεί να μπείτε στον πειρασμό να χρησιμοποιήσετε το όνομα χρήστη κάθε πελάτη για να προσδιορίσετε μεμονωμένα κάθε πελάτη. Είναι αλήθεια ότι αυτή η στήλη προσδιορίζει μεμονωμένα κάθε σειρά στον πίνακα, και εμείς καλούμε μια τέτοια στήλη υποψηφίων κλειδιών. Εντούτοις, δεν θα ήταν καλή ιδέα να χρησιμοποιήσετε αυτή η στήλη σε διαδικασίες βάσεων δεδομένων για διάφορους λόγους. Αρχικά, στο δίκτυο τα ονόματα χρήστη ήταν γνωστά πως αλλάζουν, και μια τέτοια αλλαγή θα προκαλούσε τον όλεθρο σε οποιαδήποτε βάση. Δεδομένου ότι θα το δούμε αργότερα, τα κλειδιά είναι θεμελιώδη στην καθιέρωση των σχέσεων μεταξύ των πινάκων, και αυτές οι σχέσεις στηρίζονται στο



γεγονός ότι τα κλειδιά δεν θα αλλάξουν ποτέ. Αφετέρου, τα μη αριθμητικά κλειδιά απαιτούν πολύ περισσότερη δύναμη επεξεργασίας από τα αριθμημένα. Η χρησιμοποίηση ενός `nvarchar` τομέα για να προσδιορίσει μεμονωμένα τις σειρές στον πίνακά μας θα κάνει τον SQL Server μας πιο αργό από ότι να επιλέγαμε ένα απλό, αριθμητικό κλειδί.

Η στήλη που επιλέγουμε να προσδιορίσουμε μεμονωμένα μια σειρά σε έναν πίνακα είναι στην πράξη το πρωτεύον κλειδί. Στην περίπτωση του πίνακα πελατών μας, η `CustomerID` πάντα θα να είστε μοναδική, έτσι είναι ένα κατάλληλο πρωτεύον κλειδί.

Αν και αναφερόμαστε συνήθως στα πρωτεύοντα κλειδιά σαν να ήταν στήλες, τεχνικά αυτά είναι περιορισμοί που εφαρμόζουμε στις υπάρχουσες στήλες ενός πίνακα. Οι περιορισμοί επιβάλλουν τους περιορισμούς στα στοιχεία που μπορούμε να εισαγάγουμε στους πίνακές μας, και το αρχικό κλειδί είναι ένα ιδιαίτερο είδος περιορισμού. Όταν ο αρχικός βασικός περιορισμός τίθεται στη στήλη, η βάση δεδομένων θα αρνηθεί να αποθηκεύσει τις διπλές τιμές σε εκείνη την στήλη.

Οι περιορισμοί γενικά, και τα πρωτεύοντα κλειδιά συγκεκριμένα, αντιπροσωπεύουν κατά πόσον η βάση δεδομένων μπορεί να διατηρήσει την ακεραιότητα και τη συνέπεια των στοιχείων.

Τα αρχικά κλειδιά που αποτελούνται από μια ενιαία στήλη, όπως η `CustomerID`, συχνά χρησιμοποιούνται από κοινού με την ιδιοκτησία `IDENTITY` που κάναμε λόγο πιο πάνω. Ο αρχικός βασικός περιορισμός εγγυάται ότι οι διπλές τιμές δεν μπορούν να παρεμβληθούν στον πίνακα. Η `IDENTITY` μας βοηθά με το να παραγάγει πάντα μια νέα αξία που δεν έχει χρησιμοποιηθεί ήδη στο πρωτεύον κλειδί.

### **Πρωτεύοντα κλειδιά και η ιδιότητα `IDENTITY`**

Η χρησιμοποίηση της ιδιότητας `IDENTITY` για μια στήλη δεν σημαίνει ότι μπορούμε να αποφύγουμε τη διευκρίνιση ενός πρωτεύοντος κλειδιού. Είναι αλήθεια ότι η ιδιότητα `IDENTITY` παράγει πάντα μοναδικές τιμές, αλλά αυτό δεν σημαίνει ότι τους επιβάλλουν απαραίτητως. Παραδείγματος χάριν, έχουμε έναν πίνακα με διάφορες στήλες, μια από τις οποίες θέτει την ιδιότητα `IDENTITY`. Αυτός ο πίνακας περιέχει τρία αρχεία που είναι πιθανός να περιέχουν τις αυτόματα παραγόμενες τιμές 1, 2, και 3 Στήλη `IDENTITY`. Παρεχόμενη από την ιδιότητα `IDENTITY_INSERT` για αυτόν τον πίνακα επιτρέπεται να είναι αρκετά απλό να παρεμβάλει ένα άλλο αρχείο με την αξία 2. Η στήλη `IDENTITY` θα συνέχιζε να παράγει τις μοναδικές τιμές (4, 5, 6, και τα λοιπά), αλλά δεν εγγυάται η στήλη παραμένει μοναδική.

Σε αυτό το σημείο θα ήταν πρέπει να μιλήσουμε λίγο για τις σχέσεις μεταξύ πινάκων. Επειδή όπως είδαμε και στην εικόνα του διαγράμματος πιο πάνω οι πίνακες συνδέονται και αλληλοεξαρτώνται από αυτές τις σχέσεις, τις κάνει πολύ σημαντικό κομμάτι των βάσεων δεδομένων και γιαυτό το λόγο παρακάτω θα πούμε μερικά πράγματα περισσότερα για αυτές.

Οι σχέσεις λοιπόν αυτές περιγράφουν πώς το στοιχείο σε έναν πίνακα συνδέεται με τα στοιχεία σε άλλους πίνακες. Το γεγονός, αυτό γίνεται επειδή οι σχέσεις είναι τόσο κρίσιμες που αυτοί οι τύποι βάσεων δεδομένων είναι καλούνται με το όνομα «σχεσιακές βάσεις δεδομένων.» Οι σχέσεις υπάρχουν για το σκοπό της ένωσης ενός πίνακα με έναν ή περισσότερους άλλους πίνακες που χρησιμοποιούν τα πρωτεύοντα και δευτερεύοντα κλειδιά.

Υπάρχουν τρεις τύποι σχέσεων που μπορούν να εμφανιστούν μεταξύ των πινάκων στη βάση δεδομένων:

- ❑ ένα προς ένα σχέσεις
- ❑ ένα προς πολλά σχέσεις
- ❑ πολλά προς πολλά σχέσεις

Μια ένα προς ένα σχέση σημαίνει ότι για κάθε αρχείο σε έναν πίνακα, μόνο ένα άλλο σχετικό αρχείο μπορεί να υπάρξει σε έναν άλλο πίνακα.

Οι ένα προς ένα σχέσεις χρησιμοποιούνται σπάνια, δεδομένου ότι είναι συνήθως αποδοτικότερο να συνδυάζουμε τα δύο αρχεία και τα αποθηκεύστε μαζί ως στήλες σε έναν ενιαίο πίνακα. Παραδείγματος χάριν, κάθε πελάτης στη βάση δεδομένων μας θα αποθηκεύσει ένα όνομα χρήστη στη στήλη Username του πίνακα πελατών. Θεωρητικά, θα μπορούσαμε να αποθηκεύσουμε τα ονόματα χρηστών σε ένα άλλο πίνακα και μια σύνδεση μέσω ενός δευτερεύοντος κλειδιού στον πίνακα πελάτες, αλλά αυτό θα ωφελούσε κανένα την εφαρμογή μας, δεδομένου ότι υποθέτουμε ότι εκείνο το όνομα χρήστη μπορεί να ανήκει μόνο σε έναν πελάτη. Υπό αυτήν τη μορφή, μπορούμε παραλείψουμε αυτή την ένα προς ένα σχέση από το σχέδιο βάσεων δεδομένων μας.

Η ένα προς πολλά σχέση είναι κατά πολύ ο πιο κοινός τύπος σχέσης. Η σύνδεση αυτή είναι και η σύνδεση που έχουμε χρησιμοποιήσει και στην εφαρμογή μας. Μέσα σε μια ένα προς πολλά σχέση, κάθε αρχείο σε έναν πίνακα μπορεί να συνδεθεί με πολλαπλά αρχεία από έναν δεύτερο πίνακα. Αυτά τα αρχεία συσχετίζονται συνήθως βάση του πρωτεύοντος κλειδιού από τον πρώτο πίνακα. Στο παράδειγμα προϊόντων/υποκατηγοριών, μια ένα προς πολλά σχέση υπάρχει μεταξύ των προϊόντων και των υποκατηγοριών οι πίνακες, ως μία υποκατηγορία μπορούν να συνδεθούν με πολλά προϊόντα.

Όταν ένα δευτερεύον κλειδί χρησιμοποιείται για να συνδέσει δύο πίνακες, ο πίνακας που περιέχει το δευτερεύον κλειδί είναι «στη πολλά» πλευρά της σχέσης, και ο πίνακας που περιέχει το πρωτεύον κλειδί είναι στη «ένα» πλευρά της σχέσης. Στα διαγράμματα βάσεων δεδομένων, ένα προς πολλά σχέσεις δηλώνονται από μια γραμμή μεταξύ των δύο πινάκων και ένα χρυσό κλειδί εμφανίζεται δίπλα στον πίνακα στη «ένα» πλευρά της σχέσης, και το σημάδι απείρου ( $\infty$ ) επιδεικνύεται δίπλα στον πίνακα που θα μπορούσε να αντιστοιχήσει πολλά στοιχεία σε κάθε ένα από τα αρχεία του.

Οι πολλαπλές σχέσεις εμφανίζονται μεταξύ δύο πινάκων, όταν αρχεία από καθένα πίνακας μπορεί να συνδεθεί με τα πολλαπλά αρχεία στον άλλο πίνακα.

Φανταστείτε ότι θέλετε έναν προϊόν να είναι σε θέση να ανήκει σε περισσότερο από ένα υποκατηγορίες. Ένα προϊόν μπορεί να ανήκει σε πολλές υποκατηγορίες, και μία υποκατηγορία μπορεί να περιέχει πολλά προϊόντα, έτσι αυτό είναι μια πολλαπλή σχέση. Πώς το αντιπροσωπεύουμε στη βάση δεδομένων μας; Αντιμέτωπος με αυτήν την ερώτηση, κάποιοι λιγότερο έμπειροι προγραμματιστές θα άρχιζαν να σκέφτονται τρόπους να αποθηκευτούν διάφορες τιμές σε μία ενιαία στήλη, επειδή η προφανής λύση είναι να αλλαχτεί η στήλη SubCategoryID έτσι ώστε να περιέχει έναν κατάλογο του IDs εκείνων των υποκατηγοριών στις οποίες κάθε προϊόν ανήκει.

Ο σωστός τρόπος να αντιπροσωπευθεί μια πολλαπλή σχέση είναι να προστεθεί ένας τρίτος πίνακας, ονόμασε έναν πίνακα χαρτογράφησης, στη βάση δεδομένων. Ένας πίνακας χαρτογράφησης είναι ένας πίνακας που δε περιέχει κανένα άλλο στοιχείο εκτός από τους ορισμούς των ζευγαριών των καταχωρήσεων που συσχετίζονται.


Ένας πίνακας χαρτογράφησης δημιουργείται με τον ίδιο σχεδόν τρόπο όπως οποιοσδήποτε άλλος πίνακας. Η μόνη διαφορά βρίσκεται στην επιλογή του αρχικού κλειδιού. Κάθε πίνακας που έχουμε δημιουργήσει μέχρι τώρα έχει ονομάσει μια στήλη somethingID που σχεδιάστηκε για να είναι εκείνη στον πίνακα σαν πρωτεύον κλειδί. Η υπόδειξη μιας στήλης ως πρωτεύον κλειδί λέει στη βάση δεδομένων να μην επιτρέψει δύο καταχωρήσεις σε εκείνη την στήλη για να έχει την ίδια αξία. Επιταχύνει επίσης τις αναζητήσεις βάσεων δεδομένων με βάση εκείνη την στήλη.

Στην περίπτωση ενός πίνακα χαρτογράφησης, δεν υπάρχει καμία στήλη στην οποία να θέλουμε να αναγκάσουμε να έχει μοναδικές τιμές. Κάθε προϊόν μπορεί να εμφανιστεί περισσότερο από μία φορά, ως προϊόν θα μπορεί να ανήκει σε περισσότερα από μία υποκατηγορίες, ταυτότητα και κάθε υποκατηγορία μπορεί να εμφανιστεί περισσότερο από μία φορά, δεδομένου ότι μία υποκατηγορία μπορεί να περιέχει πολλά προϊόντα. Αυτό που δε θέλουμε να επιτρέψουμε είναι το ίδιο

ζευγάρι των τιμών να εμφανιστεί στον πίνακα δύο φορές. Για αυτόν τον λόγο, δημιουργούμε συνήθως τους πίνακες χαρτογράφησης με multi-column πρωτεύον κλειδί.

Πιο συγκεκριμένα τώρα στους πίνακες τις εφαρμογής όπως τους είδαμε και στο διάγραμμα της βάσης δεδομένων.

### Πίνακας Κατηγοριών (Categories)

	Column Name	Data Type	Allow Nulls
	CategoryID	int	<input type="checkbox"/>
	CategoryName	nvarchar(50)	<input checked="" type="checkbox"/>
	Active	bit	<input type="checkbox"/>
			<input type="checkbox"/>

Στον πίνακα της εικόνας μπορούμε να δούμε τρεις στήλες ορισμένες, ενώ μπορούμε να διακρίνουμε και ότι έχει οριστεί μία από αυτές σαν πρωτεύον κλειδί.


Στη πρώτη στήλη λοιπόν με το όνομα CategoryID έχουμε προσθέσει στο πίνακά μας προκειμένου να αποθηκεύει τον αύξοντα και μοναδικό αριθμό που χαρακτηρίζει κάθε κατηγορία. Η στήλη αυτή έχει επίσης την ιδιότητα IDENTITY που έχουμε κάνει λόγο πιο πάνω. Πράγμα το οποίο σημαίνει ότι με την εισαγωγή μιας καινούρια εγγραφής στο πίνακα αναπαράγεται μια καινούρια τιμή για τη συγκεκριμένη εγγραφή. Η ιδιότητα IDENTITY έχει οριστεί έτσι ώστε να αυξάνεται κατά μία μονάδα κάθε φορά. Έτσι λοιπόν οι εγγραφές μας θα έχουν πρωτεύον κλειδί με μορφή 1,2,3 και τα λοιπά. Χαρακτηριστικό που καταλαβαίνουμε και από τη δήλωση της μορφής δεδομένων που μπορούν να αποθηκευτούν στη στήλη αυτή.

Μία τελευταία λεπτομέρεια είναι ότι δεν επιτρέπονται κενές εγγραφές στη στήλη και είναι απόλυτα λογικό αν αναλογιστούμε ότι αυτή η στήλη είναι το μοναδικό χαρακτηριστικό κάθε καταχώρησης.

Η επόμενη στήλη αναφέρεται στο όνομα της κατηγορίας και έχει τίτλο CategoryName, το είδος των δεδομένων που δέχεται είναι nvarchar και από το νούμερο στη παρένθεση καταλαβαίνουμε ότι το μέγιστο μέγεθος του ονόματος της κατηγορίας δε μπορεί να ξεπεράσει τους 50 χαρακτήρες. Θα πρέπει να τονίσουμε εδώ ότι σε αυτή τη στήλη επιτρέπονται οι κενές εγγραφές, σε σχέση με τη προηγούμενη στήλη και αυτό γιατί δεν είναι τόσο σημαντικό δεδομένο για μία εγγραφή στη λίστα αυτή.

Η τρίτη και τελευταία στήλη του πίνακα αναφέρεται στη κατάσταση της εγγραφής, της κατηγορίας δηλαδή και μπορεί να πάρει δύο τιμές, μιας και έχει οριστεί σαν bit και αυτές είναι 0 για False και 1 για True. Ο τίτλος της είναι Active και επιπλέον δε μπορεί ούτε αυτή να είναι κενή καθώς θεωρείτε σημαντική πληροφορία για κάποια εγγραφή στο πίνακα αυτό.

### Πίνακας Υποκατηγοριών (SubCategories)

	Column Name	Data Type	Allow Nulls
	SubCategoryID	int	<input type="checkbox"/>
	CategoryID	int	<input type="checkbox"/>
	SubCategoryName	nvarchar(50)	<input checked="" type="checkbox"/>
	Active	bit	<input type="checkbox"/>
			<input type="checkbox"/>

Στον πίνακα της εικόνας μπορούμε να δούμε τρεις στήλες ορισμένες, ενώ μπορούμε να διακρίνουμε και ότι έχει οριστεί μία από αυτές σαν πρωτεύον κλειδί.

Στη πρώτη στήλη λοιπόν με το όνομα SubCategoryID έχουμε προσθέσει στο πίνακά μας προκειμένου να αποθηκεύει τον αύξοντα και μοναδικό αριθμό που χαρακτηρίζει κάθε υποκατηγορία. Η στήλη αυτή έχει επίσης την ιδιότητα IDENTITY που έχουμε κάνει λόγο πιο πάνω. Πράγμα το οποίο σημαίνει ότι με την εισαγωγή μιας καινούρια εγγραφής στο πίνακα αναπαράγεται μια καινούρια τιμή για τη συγκεκριμένη εγγραφή. Η ιδιότητα IDENTITY έχει οριστεί έτσι ώστε να αυξάνεται κατά μία μονάδα κάθε φορά. Έτσι λοιπόν οι εγγραφές μας θα έχουν πρωτεύον κλειδί με μορφή 1,2,3 και τα λοιπά. Χαρακτηριστικό που καταλαβαίνουμε και από τη δήλωση της μορφής δεδομένων που μπορούν να αποθηκευτούν στη στήλη αυτή.

Μία τελευταία λεπτομέρεια είναι ότι δεν επιτρέπονται κενές εγγραφές στη στήλη και είναι απόλυτα λογικό αν αναλογιστούμε ότι αυτή η στήλη είναι το μοναδικό χαρακτηριστικό κάθε καταχώρησης.


Η στήλη με τίτλο CategoryID είναι ένα κοινό στοιχείο του πίνακα αυτού και του προηγούμενου πίνακα των κατηγοριών. Ο λόγος για τον οποίο έχουμε προσθέσει τη στήλη αυτή και σαυτόν τον πίνακα είναι για να υπάρχει μία σύνδεση μεταξύ των κατηγοριών και των υποκατηγοριών τους.

Η επόμενη στήλη αναφέρεται στο όνομα της κατηγορίας και έχει τίτλο SubCategoryName, το είδος των δεδομένων που δέχεται είναι nvarchar

και από το νούμερο στη παρένθεση καταλαβαίνουμε ότι το μέγιστο μέγεθος του ονόματος της κατηγορίας δε μπορεί να ξεπεράσει τους 50 χαρακτήρες. Θα πρέπει να τονίσουμε εδώ ότι σε αυτή τη στήλη επιτρέπονται οι κενές εγγραφές, σε σχέση με τη προηγούμενη στήλη και αυτό γιατί δεν είναι τόσο σημαντικό δεδομένο για μία εγγραφή στη λίστα αυτή.

Η τρίτη και τελευταία στήλη του πίνακα αναφέρεται στη κατάσταση της εγγραφής, της κατηγορίας δηλαδή και μπορεί να πάρει δύο τιμές, μιας και έχει οριστεί σαν bit και αυτές είναι 0 για False και 1 για True. Ο τίτλος της είναι Active και επιπλέον δε μπορεί ούτε αυτή να είναι κενή καθώς θεωρείτε σημαντική πληροφορία για κάποια εγγραφή στο πίνακα αυτό.

### Πίνακας Πελατών (Customers)

	Column Name	Data Type	Allow Nulls
	CustomerID	nvarchar(50)	<input type="checkbox"/>
	Username	nvarchar(50)	<input checked="" type="checkbox"/>
	CustomerSex	int	<input checked="" type="checkbox"/>
	CustomerAge	int	<input checked="" type="checkbox"/>
	CustomerFamily	int	<input checked="" type="checkbox"/>
	CustomerVegeterian	bit	<input checked="" type="checkbox"/>
	CustomerSmoker	bit	<input checked="" type="checkbox"/>
	CustomerDrinker	bit	<input checked="" type="checkbox"/>
	CustomerPet	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Στον πίνακα της εικόνας μπορούμε να δούμε εννέα στήλες οι οποίες χωρίζονται σε δύο κατηγορίες. Η πρώτη κατηγορία περιέχει τις δύο πρώτες στήλες στις οποίες αποθηκεύονται τα χαρακτηριστικά του κάθε χρήστη, με τα οποία το σύστημα έχει τη δυνατότητα να τον ξεχωρίσει από τους υπόλοιπους χρήστες. Στη δεύτερη κατηγορία περιέχονται οι υπόλοιπες επτά στήλες, στις οποίες αποθηκεύονται τα προσωπικά στοιχεία και προτιμήσεις του χρήστη και τα οποία χρησιμοποιεί το σύστημα για να του προβάλει τις κατάλληλες πληροφορίες όσον αφορά στις προσφορές.

Πιο αναλυτικά τώρα η πρώτη στήλη έχει να κάνει με το στοιχείο που χρησιμοποιούμε για να ξεχωρίσουμε τους πελάτες στη συγκεκριμένη περίπτωση αυτό είναι η φυσική τους διεύθυνση ή αλλιώς MAC Address. Γιαυτό και το έχουμε ορίσει και σαν nvarchar , επειδή η διεύθυνση αυτή

αποτελείται από αλφαριθμητικούς χαρακτήρες, συν το ότι ποτέ δε ξεπερνά τους 50 χαρακτήρες.

Στην αμέσως επόμενη στήλη έχουμε το όνομα του χρήστη (Username). Εδώ αποθηκεύεται το όνομα το οποίο ο χρήστης διαλέγει στη πρώτη του σύνδεση με το σύστημα και είναι αυτό τον οποίο χρησιμοποιεί η εφαρμογή για να απευθυνθεί σ' αυτόν αφού γίνει ο αναγκαίος συσχετισμός του με τη φυσική διεύθυνση του που αναφέραμε πιο πριν. Η μορφή των δεδομένων τα οποία δέχεται η συγκεκριμένη στήλη είναι , όπως πολύ σωστά φαντάζομαι ότι μαντέψατε, χαρακτήρες μιας και πρόκειται για όνομα χρήστη.

Περνάμε τώρα στη δεύτερη κατηγορία δεδομένων που αποθηκεύονται σ' αυτόν τον πίνακα και έχουν να κάνουν πάντα με τον χρήστη.

Οι στήλες από δω και κάτω είναι οι περισσότερες ίδιο τύπου, bit δηλαδή και αυτό σημαίνει ότι μπορούν πάρουν δύο τιμές 0 για False και 1 για True. Στις στήλες αυτές αποθηκεύουμε όπως είπαμε πιο πάνω τα προσωπικά στοιχεία του πελάτη με τη βοήθεια των οποίων φιλτράρουμε στη συνέχεια τις προσφορές. Τον τρόπο αποθήκευσης αυτών των δεδομένων στη βάση θα τον δούμε αργότερα πιο αναλυτικά. Στις στήλες μας προς το παρόν χρησιμοποιούμε επτά για να αποθηκεύσουμε αντίστοιχα στοιχεία. Ποια είναι αυτά;

Καταρχάς το φύλο του πελάτη αποθηκεύεται στη στήλη με τίτλο CustomerSex όπου παίρνουμε τις τιμές 0 αν ο πελάτης μας είναι Άνδρας και 1 αν είναι Γυναίκα. Ο τύπος εδώ είναι int για τον εξής απλό λόγο. Υπάρχουν προϊόντα τα οποία απευθύνονται και στα δύο φύλα και τα οποία είναι χαρακτηρισμένα με τη τιμή 2 στην αντίστοιχη στήλη τους.


Στην επόμενη στήλη τώρα, εδώ αποθηκεύεται η ηλικία του πελάτη. Η τιμή στη στήλη CustomerAge θα μπορεί να έχει τις εξής τιμές 0 αν η ηλικία του είναι από 18-29, 1 αν είναι από 30 – 49 και 2 αν η ηλικία του είναι από 50 και πάνω. Εδώ πάλι είναι περισσότερο προφανείς ο λόγος που χαρακτηρίζουμε τα δεδομένα αυτά σαν int αλλά πρέπει να πούμε επίσης ότι υπάρχουν και προϊόντα με τιμή όσον αφορά στο φύλλο που απευθύνονται 3, αυτό σημαίνει ότι το προϊόν απευθύνεται σε όλες τις ηλικίες.

Αμέσως μετά έχουμε τη στήλη που αποθηκεύουμε την οικογενειακή κατάσταση, με τίτλο CustomerFamily. Η στήλη αυτή έχει το ίδιο εύρος τιμών με την από πάνω της, την στήλη με την ηλικία δηλαδή, με τις τιμές να αντιπροσωπεύουν 0 τον ελεύθερο/ή πελάτη ή πελάτισσα, 1 τον παντρεμένο, 2 τον παντρεμένο με παιδιά.

Προχωράμε τώρα στις στήλες χαρακτηρισμένες σαν bit, όπου έχουμε και σειρά τις CustomerVegeterian με τιμές 0 για μη χορτοφάγους και 1 για χορτοφάγους, CustomerSmoker με 0 για μη καπνιστές και 1 για

καπνιστές, CustomerDrinker με 0 για πελάτες που δε πίνουν και 1 για αυτούς που πίνουν και τέλος CustomerPet με 0 για πελάτες που έχουν δεν έχουν κατοικίδιο και 1 για αυτούς που έχουν.

### Πίνακας Προϊόντων (Products)

	Column Name	Data Type	Allow Nulls
	ProductID	int	<input type="checkbox"/>
	CategoryID	int	<input type="checkbox"/>
	SubCategoryID	int	<input type="checkbox"/>
	ProductName	nvarchar(50)	<input checked="" type="checkbox"/>
	Price	money	<input checked="" type="checkbox"/>
	ProductDescription	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ProductPicture	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Available	bit	<input type="checkbox"/>
	CustomerSex	int	<input checked="" type="checkbox"/>
	CustomerAge	int	<input checked="" type="checkbox"/>
	CustomerFamily	int	<input checked="" type="checkbox"/>
	CustomerVegeterian	bit	<input checked="" type="checkbox"/>
	CustomerSmoker	bit	<input checked="" type="checkbox"/>
	CustomerDrinker	bit	<input checked="" type="checkbox"/>
	CustomerPet	bit	<input checked="" type="checkbox"/>
	InOffer	bit	<input checked="" type="checkbox"/>
	OfferPrice	money	<input checked="" type="checkbox"/>
	OfferTitle	nvarchar(100)	<input checked="" type="checkbox"/>
	OfferSmallDesc	nvarchar(MAX)	<input checked="" type="checkbox"/>
	OfferLargeDesc	nvarchar(MAX)	<input checked="" type="checkbox"/>
	OfferPicture	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Στον παραπάνω πίνακα μπορούμε να δούμε τον βασικότερο πίνακα της εφαρμογής. Ο πίνακας αυτός περιέχει τα προϊόντα της εταιρείας και χωρίζεται και αυτός σε δύο βασικές κατηγορίες. Η πρώτη κατηγορία περιέχει τις πληροφορίες για το προϊόν γενικά και η δεύτερη περιέχει τις πληροφορίες για το προϊόν όταν αυτό βρίσκεται σε προσφορά.

Αναλυτικά τώρα για τις στήλες του πίνακά μας, η πρώτη μας στήλη με τίτλο ProductID περιέχει το μοναδικό αυτό αριθμό (identity) που μας



βοηθάει να ξεχωρίσουμε το ένα προϊόν από τα άλλα. Πρόκειται για το πρωτεύον κλειδί του πίνακα αυτού και είναι φυσικά τύπου `int`. Ενώ δεν επιτρέπονται κενές εγγραφές σ αυτή τη στήλη για προφανείς λόγους.

Στην επόμενη , όπως και στην αμέσως από κάτω της στήλη μας περιέχονται οι πληροφορίες για την κατηγορία του προϊόντος `CategoryID`, όπως και η υποκατηγορία του `SubCategoryID`. Τα δεδομένα που μπορούμε να εισάγουμε και στις δύο στήλες είναι τύπου `int` , ενώ είναι και ο συνδεδεικός κρίκος με τους πίνακες `Category` και `SubCategory` αντίστοιχα. Τέλος μπορούμε να πούμε ότι δεν επιτρέπονται κενές εγγραφές ούτε εδώ.

Η αμέσως από κάτω στήλη τώρα με τίτλο `ProductName` περιέχει το όνομα του προϊόντος και είναι τίτλου `nvarchar`. Επίσης το μέγιστο μήκος του ονόματος έχει οριστεί στους 50 χαρακτήρες, ενώ σ αυτή τη στήλη επιτρέπονται οι κενές εγγραφές.

Η επόμενη στήλη τώρα , με τίτλο `Price`. Όπως μπορείτε εύκολα να καταλάβετε περιέχει την τιμή του προϊόντος , όταν αυτό δε βρίσκεται σε προσφορά. Η στήλη αυτή λοιπόν δέχεται τιμές τύπου `money` , όπως έχουμε αναφέρει και πιο πάνω όταν κάναμε λόγο για τους τύπους των μεταβλητών που μπορούν να δεχθούν οι στήλες. Εδώ μπορούμε να πούμε ότι θα μπορούσαμε να χρησιμοποιήσουμε και `float` τύπο για τη συγκεκριμένη στήλη, αλλά όπως έχουμε αναφέρει και πιο πάνω, ο τύπος `float` είναι επιρρεπής στις στρογγυλοποιήσεις, επειδή λοιπόν στην εφαρμογή μας χρησιμοποιούμε προϊόντα που έχουν αρκετά δεκαδικά ψηφία και οι προσφορές βασίζονται σε άλλοτε μικρές και άλλοτε μεγάλες διαφορές στις τιμές αυτές διαλέγουμε να ορίσουμε τις τιμές αυτές σαν `money`. Αν είχαμε ένα κατάστημα που εμπορευόταν αυτοκίνητα για παράδειγμα οι τιμές αυτές θα μπορούσαν να είναι και `float` , γιατί η στρογγυλοποίηση σε αυτή τη περίπτωση θα ήταν μηδαμινή σε σχέση με τη τιμή του αυτοκινήτου.

Κάτω από τη στήλη της τιμής του προϊόντος, υπάρχει η στήλη της περιγραφής του, η οποία όπως φαίνεται και στην εικόνα του πίνακα έχει τίτλο `ProductDescription`. Η στήλη αυτή δέχεται τιμές χαρακτήρες και δεν έχει περιορισμούς όσον αφορά στο μέγεθός τους λόγω `nvarchar(MAX)`.

Στην από κάτω στήλη αποθηκεύουμε τη φωτογραφία του προϊόντος και έχουμε δώσει τίτλο `ProductPicture`. Εδώ μπορούμε να πούμε όσον αφορά τον τύπο των δεδομένων που δέχεται η στήλη αυτή ότι η φωτογραφία του προϊόντος περιγράφεται με το `path` της εικόνας στην εφαρμογή, δηλαδή όπως θα δούμε και παρακάτω όλες οι εικόνες είναι αποθηκευμένες σε ένα φάκελο με όνομα `images` μέσα στην εφαρμογή, ο οποίος περιέχει όλες τις εικόνες των προϊόντων και όχι μόνο. Έτσι λοιπόν σε αυτή τη στήλη αποθηκεύουμε τη διεύθυνση στην οποία

βρίσκεται η εικόνα και το όνομά της φυσικά. Για τη συγκεκριμένη στήλη θα μπορούσαμε να χρησιμοποιήσουμε `picture` δεδομένα με τα οποία ο `SQL Server` θα αποθήκευε τις φωτογραφίες στη βάση και όχι το `path` τους. Αυτό θα έκανε το μέγεθος της βάσης αρκετά μεγάλο, χωρίς λόγο από τη στιγμή που και ο αριθμός των προϊόντων στη συγκεκριμένη περίπτωση δεν είναι αρκετά μεγάλος έτσι ώστε να μας δημιουργήσει προβλήματα η διαχείρισή τους.

Τέλος και κλείνοντας τη κατηγορία των πληροφοριών που αφορούν το προϊόν, έχουμε προσθέσει μία στήλη με τίτλο `Available` και τύπο δεδομένων `bit`, για να αποθηκεύουμε τη κατάσταση ενός προϊόντος, αν δηλαδή το προϊόν είναι σε διάθεση ή όχι και αντιπροσωπεύουμε αυτές τις δύο καταστάσεις με τις τιμές 1 και 0 αντίστοιχα.

Περνάμε τώρα στη κατηγορία όπου αποθηκεύουμε τις πληροφορίες της προσφοράς. Αρχίζουμε λοιπόν με τις τα χαρακτηριστικά του προϊόντος σχετικά με τα οποία θα γίνεται και το φιλτράρισμα τους σε σχέση με τα δεδομένα που έχει εισάγει ο πελάτης.

Στη πρώτη στήλη μας λοιπόν, όπως έχουμε αναφέρει και πιο πάνω στο πίνακα των πελατών, με τίτλο `CustomerSex`, αποθηκεύουμε το φύλο στο οποίο απευθύνεται το προϊόν. Οι τιμές που μπορεί να πάρει είναι 0 για άνδρες, 1 για γυναίκες και 2 και για τους δύο. Επειδή μπορεί να δεχθεί τρεις τιμές ορίζεται και σαν `int`.

Στην επόμενη στήλη τώρα, εδώ αποθηκεύεται η ηλικία του πελάτη. Η τιμή στη στήλη `CustomerAge` θα μπορεί να έχει τις εξής τιμές 0 αν η ηλικία του είναι από 18-29, 1 αν είναι από 30 – 49 και 2 αν η ηλικία του είναι από 50 και πάνω. Εδώ πάλι είναι περισσότερο προφανείς ο λόγος που χαρακτηρίζουμε τα δεδομένα αυτά σαν `int` αλλά πρέπει να πούμε επίσης ότι υπάρχουν και προϊόντα με τιμή όσον αφορά στο φύλλο που απευθύνονται 3, αυτό σημαίνει ότι το προϊόν απευθύνεται σε όλες τις ηλικίες.

Αμέσως μετά έχουμε τη στήλη που αποθηκεύουμε την οικογενειακή κατάσταση, με τίτλο `CustomerFamily`. Η στήλη αυτή έχει το ίδιο εύρος τιμών με την από πάνω της, την στήλη με την ηλικία δηλαδή, με τις τιμές να αντιπροσωπεύουν 0 τον ελεύθερο/ή πελάτη ή πελάτισσα, 1 τον παντρεμένο, 2 τον παντρεμένο με παιδιά.

Προχωράμε τώρα στις στήλες χαρακτηρισμένες σαν `bit`, όπου έχουμε και σειρά τις `CustomerVegeterian` με τιμές 0 για μη χορτοφάγους και 1 για χορτοφάγους, `CustomerSmoker` με 0 για μη καπνιστές και 1 για καπνιστές, `CustomerDrinker` με 0 για πελάτες που δε πίνουν και 1 για αυτούς που πίνουν και τέλος `CustomerPet` με 0 για πελάτες που έχουν δεν έχουν κατοικίδιο και 1 για αυτούς που έχουν.


Στην αμέσως επόμενη σειρά έχουμε μια στήλη που καθορίζει και αυτή κατάσταση προϊόντος. Πιο συγκεκριμένα η στήλη InOffer καθορίζει αν το προϊόν βρίσκεται σε προσφορά ή όχι. Με τιμές True και False αντίστοιχα ή 1 και 0 αν προτιμάτε. Στη περίπτωση που το προϊόν δε βρίσκεται σε προσφορά οι πληροφορίες προσφοράς δε υπολογίζονται ούτε προσμετρώνται στο φιλτράρισμα.

Στις επόμενες στήλες προϋποθέτουν ότι το προϊόν είναι σε προσφορά για να ληφθούν υπόψη. Η στήλη με τίτλο OfferPrise περιέχει τη τιμή της προσφοράς είναι τύπου money και υπερισχύει της ProductPrise αν το προϊόν είναι σε προσφορά.

Από κάτω έχουμε τον τίτλο της προσφοράς χαρακτηριμένο σαν OfferTitle και είναι τύπου nvarchar , ενώ δέχεται μέχρι 100 χαρακτήρες. Στην επόμενη και μεθεπόμενη στήλη αποθηκεύουμε τις περιγραφές της προσφοράς με OfferSmallDesc τη περιγραφή που εμφανίζεται για κάθε προϊόν στην αριστερή λίστα των προσφορών και OfferLargeDesc τη περιγραφή που εμφανίζεται όταν ο χρήστης ζητήσει περισσότερες πληροφορίες για τη προσφορά. Αυτές οι στήλες δέχονται δεδομένα τύπου nvarchar ενώ δεν έχουν περιορισμό στο μέγεθος.

Τέλος για τον πίνακα των προϊόντων έχουμε τη στήλη με την εικόνα του προϊόντος σε προσφορά η οποία λειτουργεί ακριβώς με το ίδιο τρόπο με τη στήλη ProductPicture για την αποθήκευση της εικόνας, ενώ υπερισχύει της εικόνας του προϊόντος όταν αυτό είναι σε προσφορά.

### Πίνακας Προσωπικών Προσφορών (PersonalOffers)

	Column Name	Data Type	Allow Nulls
	PersonalOfferID	int	<input type="checkbox"/>
	CustomerID	nvarchar(50)	<input checked="" type="checkbox"/>
	ProductID	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ο τελευταίος πίνακας της βάσης μας περιέχει τις φιλτραρισμένες προσφορές του κάθε πελάτη και χρησιμεύει στο να θυμάται το σύστημα της προσφορές που έχει προβάλλει σε κάθε πελάτη και στη συνέχεια να τις ξαναπροβάλλει όταν αυτός ξανασυνδεθεί στο σύστημα.

Στον πίνακά μας τώρα , αρχικά έχουμε μια στήλη με τίτλο PersonalOfferID, η οποία είναι και πρωτεύον κλειδί στον πίνακα αυτό , ενώ έχει και την ιδιότητα identity. Για να πούμε την αλήθεια η στήλη αυτή θα μπορούσε να μην υπάρχει αφού όπως θα δούμε παρακάτω η

ταυτότητα του κάθε χρήστη είναι αρκετή ώστε να διαχωριστούν οι προσφορές.

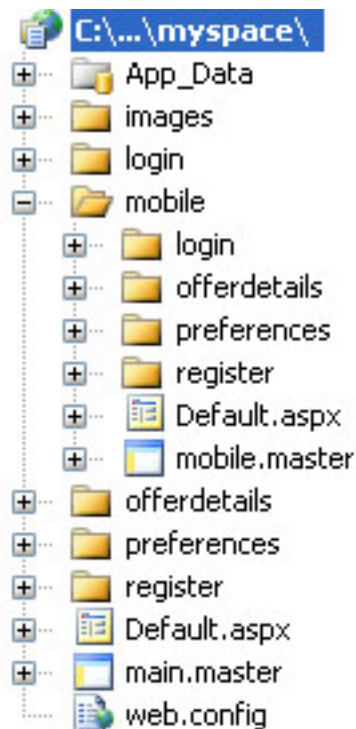
Η επόμενη στήλη είναι η στήλη που αποθηκεύεται η ταυτότητα του χρήστη , η φυσική διεύθυνση του δηλαδή. Η στήλη αυτή είναι επίσης το σημείο σύνδεσης αυτού του πίνακα με τον πίνακα των πελατών.

Τέλος έχουμε τη στήλη που αποθηκεύεται η ταυτότητα του προϊόντος, που βρίσκεται σε προσφορά εννοείται και εδώ αυτή η στήλη είναι ο συνδετικός κρίκος μεταξύ αυτού του πίνακα και του πίνακα των προϊόντων.

## 3.2 Υλοποίηση και Κώδικας

Στο προηγούμενο κεφάλαιο αναλύσαμε τη βάση του συστήματος, είδαμε πώς το σύστημά μας δέχεται και αποθηκεύει τις πληροφορίες των χρηστών αλλά και πώς αυτές συνδέονται με τις ήδη υπάρχουσες πληροφορίες της εφαρμογής. Σε αυτό το κομμάτι θα μιλήσουμε πιο συγκεκριμένα για το πώς λειτουργεί το σύστημα, για το πώς δηλαδή μέσω του κώδικα λαμβάνει, φιλτράρει, προβάλλει και γενικότερα αλληλεπιδρά.

Στη παρακάτω εικόνα μπορούμε να διακρίνουμε τη βασική δομή της εφαρμογής, η οποία όπως είναι λογικό χωρίζεται σε δύο κομμάτια , το ένα είναι η εφαρμογή σε μέγεθος για προβολή σε Laptop και το δεύτερο (η δομή που φαίνεται και μέσα στο φάκελο mobile) η εφαρμογή σε μέγεθος για προβολή σε PDA.



Παρά τον χωρισμό της εφαρμογής στις δύο αυτές μορφές υπάρχουν κάποια στοιχεία τα οποία είναι κοινά και για τις δύο. Τα στοιχεία λοιπόν αυτά είναι τα παρακάτω:

❑ **App\_Data** είναι ένας πρόσθετος φάκελος που η ASP.NET χρησιμοποιεί για να αποθηκεύσει τα αρχεία βάσεων δεδομένων. Σε αυτόν τον φάκελο βρίσκεται και η βάση δεδομένων της εφαρμογής μας την οποία χρησιμοποιεί οποιαδήποτε συσκευή ανεξαρτήτου ανάλυσης.

❑ **Default.aspx** είναι η προεπιλεγμένη φόρμα την οποία ο Visual Web Developer δημιουργεί για εμάς. Εάν κοιτάξετε πιο προσεκτικά, θα δείτε ότι μπορείτε να επεκτείνετε τον Default.aspx κόμβο πατώντας το σύμβολο + στα αριστερά του. Εάν επεκτείνετε τον κόμβο, θα βρείτε ένα αρχείο κώδικα από πίσω με όνομα Default.aspx.vb. Ο Visual Web Developer μπορεί να εργαστεί με τις φόρμες που χρησιμοποιούν τέτοια αρχεία, καθώς επίσης και με εκείνες δεν χρησιμοποιούν. Αν και η σελίδα Default.aspx είναι μέρος της προβολής για laptop έχουν πρόσβαση και οι δύο προβολές και μετά την αναγνώριση της συσκευής αν κριθεί αναγκαίο τότε δρομολογείται ο χρήστης στην αντίστοιχη προβολή. Μια διαδικασία που θα δούμε αργότερα πιο αναλυτικά όταν αναλύσουμε τον κώδικα της σελίδας.

❑ **Web.config** είναι αρχείο διαμόρφωσης της εφαρμογής μας. Με την επεξεργασία του Web.config, μπορούμε να θέσουμε πολυάριθμες προκαθορισμένες επιλογές διαμόρφωσης για την εφαρμογή μας. Μπορούμε επίσης να καθορίσουμε τις δικές μας

ρυθμίσεις για όλη την εφαρμογή που μπορούν έπειτα να διαβαστούν από τον κώδικά μας (όπως η διεύθυνση ηλεκτρονικού ταχυδρομείου του διαχειριστή, το όνομα του προγράμματος, το αγαπημένο μας χρώμα, ή οποιοδήποτε άλλη απλή αξία που θα επιθυμούσαμε να αποθηκεύσουμε σε μια κεντρική θέση).

Πάμε τώρα να τα δούμε όλα αυτά πιο αναλυτικά στο κώδικα της εφαρμογής.

## Web.config

```
<?xml version="1.0"?>

<configuration>
  <appSettings/>
  <connectionStrings>

<add name="eshopdbfileConnectionString"
  connectionString="Data Source = .\SQLEXPRESS;
  AttachDbFilename =
C:\inetpub\wwwroot\myspace\App_Data\eshopdbfile.mdf;
  Integrated Security=True;
  Connect Timeout=30;
  User Instance=True"
  providerName="System.Data.SqlClient"/>

  </connectionStrings>

  <system.web>

  <authentication mode="Forms">
    <forms name=".LoginCookie"
      loginUrl="login/Default.aspx"
      protection="All"
      timeout="40"
      path="/"
      cookieless="UseUri"/>
  </authentication>
  <compilation debug="true" strict="false" explicit="true"/>
  <pages>
    <namespaces>
      <clear/>
      <add namespace="System"/>
      <add namespace="System.Collections"/>
      <add
namespace="System.Collections.Specialized"/>
      <add namespace="System.Configuration"/>
    </namespaces>
  </pages>
</system.web>
</configuration>
```

```
<add namespace="System.Text"/>
<add
namespace="System.Text.RegularExpressions"/>
<add namespace="System.Web"/>
<add namespace="System.Web.Caching"/>
<add namespace="System.Web.SessionState"/>
<add namespace="System.Web.Security"/>
<add namespace="System.Web.Profile"/>
<add namespace="System.Web.UI"/>
<add namespace="System.Web.UI.WebControls"/>
<add
namespace="System.Web.UI.WebControls.WebParts"/>
<add namespace="System.Web.UI.HtmlControls"/>
</namespaces>
</pages>
</system.web>
</configuration>
```

Σχεδόν κάθε εφαρμογή ASP.NET περιέχει ένα αρχείο Web.config, το οποίο αποθηκεύει διάφορες ρυθμίσεις της εφαρμογής. Εξ ορισμού, όλες οι εφαρμογές ASP.NET είναι διαμορφωμένες στο αρχείο Machine.config, που περιέχει τις γενικές ρυθμίσεις της μηχανής, και είναι αποθηκευμένο στον κατάλογο C:\WINDOWS\Microsoft.NET\Framework\version\CONFIG.

Για το μεγαλύτερο μέρος, δεν θα χρειαστεί να κάνουμε οποιεσδήποτε τροποποιήσεις σε αυτό το αρχείο. Εντούτοις, μπορούμε να προσπεράσουμε κάποιες ρυθμίσεις του αρχείου Machine.config με την προσθήκη του Web.config αρχείου στον κεντρικό κατάλογο της εφαρμογής μας.

Το αρχείο Web.config είναι ένα αρχείο XML που μπορεί να κρατήσει τις ρυθμίσεις της εφαρμογής στην οποία το αρχείο κατοικεί. Μια από τις πιο χρήσιμες ρυθμίσεις που το Web.config ελέγχει είναι το debug mode της ASP.NET. Μπορούμε δηλαδή να επιτρέψουμε το debug μέσω Web.config, όπως φαίνεται και στη παρακάτω γραμμή κώδικα:

```
<compilation debug="true" strict="false" explicit="true"/>
```

Η αλλαγή του mode του debug είναι πολύ απλή όπως μπορείτε να δείτε και εσείς και έχει επιτευχθεί απλά και μόνο με την αλλαγή του debug σε true. Οι άλλες ιδιότητες που υπάρχουν εδώ προστέθηκαν από τον Visual Web Developer για να προσφέρουν βοήθεια στην ανάπτυξη της VB εφαρμογής σε θέματα συμβατότητα προς τα πίσω. Παραδείγματος χάριν, το strict="false" κάνει το μεταγλωττιστή να μας συγχωρήσει σε μερικά από τα λάθη που μπορεί να έχουμε κάνει, όπως η χρησιμοποίηση της λανθασμένης χρήσης κεφαλαίων και μικρών στα ονόματα των μεταβλητών.

Τα αρχεία Web.config μπορούν επίσης να χρησιμοποιηθούν για να αποθηκεύσουν συνηθισμένες πληροφορίες για την εφαρμογή μας σε μια κεντρική θέση που είναι προσιπή από όλες τις σελίδες της περιοχής μας. Παραδείγματος χάριν, αν θέλετε να αποθηκεύσετε τη διεύθυνση ηλεκτρονικού ταχυδρομείου κάποιου έτσι ώστε αυτή να μπορεί να αλλάξει εύκολα, μπορούμε να την αποθηκεύσουμε εδώ.

Μια άλλη βοήθεια για τους χρήστες, είναι ότι το αρχείο αρχικό Web.config που παράγεται από τον Visual Web Developer περιέχει διάφορες αναφορές namespace όπως μπορούμε να δούμε και στο κώδικά μας.

Σε αυτό το σημείο καλό είναι να αναφερόμαστε λίγα πράγματα για τα namespaces. Δεδομένου λοιπόν ότι η ASP.NET είναι μέρος του .NET Framework, έχουμε πρόσβαση σε όλα τα εργαλεία που χτίζουμε σε αυτό κάτω από τη μορφή .NET Framework Class Library. Αυτή η βιβλιοθήκη αντιπροσωπεύει έναν τεράστιο πόρο εργαλείων και χαρακτηριστικών γνωρισμάτων υπό μορφή κλάσεων, αυτές οι κλάσεις οργανώνονται σε μια ιεραρχία από namespaces. Όταν θέλουμε να χρησιμοποιήσουμε ορισμένα χαρακτηριστικά γνωρίσματα που το .NET παρέχει, πρέπει να βρούμε μόνο το namespace που περιέχει τη επιθυμητή λειτουργία, και να το εισάγουμε στη ASP.NET σελίδα μας. Μόλις το κάνουμε αυτό, μπορούμε να χρησιμοποιήσουμε τις .NET κλάσεις.

Για παράδειγμα, εάν θελήσουμε να έχουμε πρόσβαση σε μια βάση δεδομένων από μια σελίδα, θα εισάγαμε το namespace που περιέχει τις κατηγορίες για αυτόν το λόγο, οι οποίες θα μπορούσαν να είναι η System.Data.SqlClient. Επίσης η κλάση του ελέγχου κουμπιών μπορεί να βρεθεί στο System.Web.UI.WebControls.

Για να χρησιμοποιήσετε μια κλάση που είναι μέρος ενός namespace που δεν είναι διαθέσιμο σε σας εξ ορισμού, είτε πρέπει να εισαγάγετε το namespace, είτε να παραπέμψετε στη κλάση χρησιμοποιώντας το πλήρες όνομα, όπως System.Web.UI.WebControls.

Μπορούμε λοιπόν να χρησιμοποιήσουμε τις κλάσεις από αυτά τα namespaces στον κώδικά μας χωρίς να είμαστε αναγκασμένοι να τα αναφέρουμε σε κάθε αρχείο στο οποίο χρησιμοποιούνται.

Το βασικό στοιχείο του αρχείου Web.config είναι πάντα το configuration, το οποίο μπορεί να περιέχει τρεις διαφορετικούς τύπους στοιχείων: **configuration section groups**, **configuration sections** και **configuration section handler declarations**. Από αυτά εμείς θα αναλύσουμε τα **configuration sections** που είναι και το στοιχείο του web.config που χρησιμοποιήσαμε.



Τα **configuration sections** είναι οι πραγματικές ετικέτες ρυθμίσεων στο αρχείο διαμόρφωσής μας. Από ένα στοιχείο μπορεί να περιέχετε διάφορες ρυθμίσεις. Η Microsoft καλεί κάθε μια από αυτές τις ετικέτες «configuration sections.» Η ASP.NET παρέχει ένα ευρύ φάσμα των ενσωματωμένων configuration sections για να ελέγξουν τις διάφορες πτυχές της εφαρμογής μας. Ο ακόλουθος κατάλογος περιγράφει μερικές από τα συνήθως χρησιμοποιούμενα από την ASP.NET configuration sections, τα οποία εμφανίζονται μέσα στο τμήμα system.web στοιχείο ομαδοποίησης:

**Authentication** περιγράφει τα configuration sections για την επικύρωση χρηστών.

**Authorization** διευκρινίζει τους χρήστες και τους ρόλους, και ελέγχει την πρόσβασή τους στα ιδιαίτερα αρχεία μέσα σε μια εφαρμογή.

**Compilation** περιέχει τις ρυθμίσεις που συσχετίζονται με τη σύνταξη σελίδων, και μας αφήνει να διευκρινίσουμε τη γλώσσα προεπιλογής που χρησιμοποιείται για τη σύνταξη των σελίδων.

**CustomErrors** χρησιμοποιείται για να προσαρμόσουμε τον τρόπο με τον οποίο προβάλλονται τα λάθη.

**Globalization** χρησιμοποιείται για να προσαρμόσουμε την κωδικοποίηση χαρακτήρων για τα αιτήματα και τις απαντήσεις.

**Pages** χειρίζεται τις configuration options για τις συγκεκριμένες σελίδες ASP.NET και μας επιτρέπει να θέσουμε εκτός λειτουργίας το session state.

**SessionState** περιέχει τις πληροφορίες διαμόρφωσης για την τροποποίηση του session state (δηλ. μεταβλητές συνδεδεμένες με την επίσκεψη ενός ιδιαίτερου χρήστη στην περιοχή μας).

**Trace** περιέχει τις πληροφορίες σχετικές με την επισήμανση των σελίδων και της εφαρμογής.

## Main.aspx.vb

```
Imports System.Data
Imports System.Data.SqlClient
Imports System.Net
Imports System.Net.NetworkInformation

Partial Class main
    Inherits System.Web.UI.MasterPage
```

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

```
Dim bc As HttpBrowserCapabilities = Request.Browser
If bc.IsMobileDevice = True Then
    Response.Redirect("/myspace/mobile/Default.aspx")
End If
```

```
Dim nics() As NetworkInterface = NetworkInterface.GetAllNetworkInterfaces()
Dim nic As String = nics(0).GetPhysicalAddress().ToString()
Session("CustomerMac") = nic
```

```
Dim userCookie As HttpCookie
userCookie = Request.Cookies("UserID")
```

```
If userCookie Is Nothing Then
    userCookie = New HttpCookie("UserID", "Anonymous")
    Response.Cookies.Add(userCookie)
End If
```

```
Dim connectionString As String = "Data Source=.\SQLEXPRESS;
```

```
AttachDbFilename=C:\inetpub\wwwroot\myspace\App_Data\eshopdbfile.mdf;
Integrated Security=True;
Connect Timeout=30;
User Instance=True"
```

```
Dim con As SqlConnection = New SqlConnection(connectionString)
```

```
Dim dadPref As SqlCommand
Dim dadCust As SqlCommand
Dim SQLString As String
Dim SQLString2 As String
```

```
SQLString = "SELECT Count(*) FROM Customers WHERE CustomerID = '" &
Session("CustomerMac") & "'"
dadCust = New SqlCommand(SQLString, con)
dadCust.Connection.Open()
If dadCust.ExecuteScalar() = 0 Then
```

```
    SQLString2 = "INSERT INTO Customers (CustomerID) VALUES ('" &
Session("CustomerMac") & "'"
    dadPref = New SqlCommand(SQLString2, con)
    dadPref.ExecuteNonQuery()
```

```
If userCookie.Value <> "Anonymous" Then
```

```
UserStatus.Text = "LoggedIn as, " & userCookie.Value
Session("Username") = userCookie.Value

Dim connectionString2 As String = "Data Source=.\SQLEXPRESS;
AttachDbFilename=C:\inetpub\wwwroot\myspace\App_Data\eshopdbfile.mdf;
Integrated Security=True;
Connect Timeout=30;
User Instance=True"
Dim con2 As SqlConnection = New SqlConnection(connectionString2)

Dim dadPref2 As SqlCommand
Dim SQLString3 As String

SQLString3 = "UPDATE Customers SET Username = " & Session("Username")
& " WHERE CustomerID = " & Session("CustomerMac") & ""
dadPref2 = New SqlCommand(SQLString3, con2)
dadPref2.Connection.Open()
dadPref2.ExecuteNonQuery()
con2.Close()

End If
Response.Redirect("register/Default.aspx")
End If

con.Close()

If userCookie.Value <> "Anonymous" Then
UserStatus.Text = "LoggedIn as, " & userCookie.Value
Session("Username") = userCookie.Value

Dim connectionString2 As String = "Data Source=.\SQLEXPRESS;
AttachDbFilename=C:\inetpub\wwwroot\myspace\App_Data\eshopdbfile.mdf;
Integrated Security=True;
Connect Timeout=30;
User Instance=True"

Dim con2 As SqlConnection = New SqlConnection(connectionString2)

Dim dadPref2 As SqlCommand
Dim SQLString3 As String

SQLString3 = "UPDATE Customers SET Username = " & Session("Username") &
" WHERE CustomerID = " & Session("CustomerMac") & ""
dadPref2 = New SqlCommand(SQLString3, con2)
```

```
dadPref2.Connection.Open()  
dadPref2.ExecuteNonQuery()  
con2.Close()  
End If  
  
End Sub  
  
End Class
```

Οι master pages είναι ένα νέο χαρακτηριστικό γνώρισμα στην ASP.NET 2.0 που μπορεί να έχει σημαντική διαφορά στον τρόπο που συνθέτουμε τις web forms. Οι master pages είναι παρόμοιες με τον web user control δεδομένο ότι αποτελούνται επίσης από HTML και άλλους ελέγχους, μπορούν να επεκταθούν με την προσθήκη γεγονότων, μεθόδων, ή ιδιοτήτων και δε μπορούν να φορτωθούν άμεσα από τον χρήστη, αντ' αυτού χρησιμοποιούνται ως δομικές μονάδες στο σχεδιασμό της δομής των web forms.

Μια κύρια σελίδα είναι ένα πρότυπο σελίδας που μπορεί να εφαρμοστεί για να δώσει σε πολλές σελίδες συνεπή εμφάνιση. Παραδείγματος χάριν, μια master σελίδα μπορεί να καθορίσει μια τυποποιημένη δομή περιλαμβάνοντας την επιγραφή, την υποσημείωση, και άλλα στοιχεία που αναμένετε να επιδείξουμε μέσα σε πολλαπλές σελίδες μέσα σε μια εφαρμογή.

Τα master pages αρχεία έχουν την επέκταση .master, και ακριβώς όπως τις σελίδες και τα web user controls, υποστηρίζουν κώδικας πίσω από τα αρχεία. Όλες οι κύριες σελίδες κληρονομούνται από την κλάση System.Web.UI.MasterPage.

Ο σχεδιασμός της δομής της εφαρμογής χρησιμοποιώντας master page και web user controls μας δίνει τη δυνατότητα να τροποποιήσουμε και να επεκτείνουμε εύκολα την εφαρμογή. Εάν η εφαρμογή μας χρησιμοποιεί αυτά τα χαρακτηριστικά γνώρισμα με ένα καλά σχεδιασμένο τρόπο, μπορεί να είναι πολύ εύκολο να τροποποιηθούν ορισμένες λεπτομέρειες στο σχεδιασμό ή τη λειτουργία της εφαρμογής μας, επειδή ενημερώνοντας μια master page ή web user control έχει άμεσο αποτέλεσμα σε όλες τις σελίδες που χρησιμοποιούν το αρχείο.

Όπως έχουμε αναφέρει ήδη, μια master page χτίζεται χρησιμοποιώντας HTML και τους ελέγχους, συμπεριλαμβανομένου του πρόσθετου ελέγχου ContentPlaceHolder. Όπως το όνομά του προτείνει, Το ContentPlaceHolder είναι placeholder που μπορεί να γεμίσει με το περιεχόμενο σχετικά με τις ανάγκες κάθε σελίδας που χρησιμοποιεί την master page. Στη δημιουργία μιας master page, περιλαμβάνουμε όλη την βασική δομή των μελλοντικών σελίδων στην ίδια την master page, συμπεριλαμβανομένου των <html>, <head>, <body> ετικετών, και

αφήνουν τις σελίδες να διευκρινίσουν το περιεχόμενο που θα εμφανίζεται στα placeholders.

Εδώ καλό θα ήταν να πούμε μερικά πράγματα για τα cookies και τον τρόπο που τα χρησιμοποιούμε. Δεδομένου ότι στην εφαρμογή μας τα cookies χρησιμοποιούνται αρκετά για την αναγνώριση και ταυτοποίηση του κάθε χρήστη, όπως φαίνεται και στο κώδικα της master page μας.

Εάν θέλουμε να αποθηκεύσουμε στοιχεία σχετικά με έναν ιδιαίτερο χρήστη, θα μπορούσαμε να χρησιμοποιήσουμε το Session Object, αλλά αυτή η προσέγγιση έχει ένα σημαντικό μειονέκτημα: το περιεχόμενό του χάνεται όταν ο χρήστης κλείνει το παράθυρο του φυλλομετρητή. Για να αποθηκεύσουμε τα στοιχεία χρηστών για μακρύτερες χρονικές περιόδους, πρέπει να χρησιμοποιήσουμε cookies. Τα cookies είναι κομμάτια των στοιχείων που η ASP.NET εφαρμογή μας μπορεί να σώσει στον browser του χρήστη, για να διαβαστεί αργότερα από το σύστημα μας. Τα cookies δεν χάνονται όταν ο browser κλείσει (εκτός αν ο χρήστης τα διαγράψει), έτσι μπορείτε να σώσετε τα στοιχεία που προσδιορίζουν τον χρήστη μας σε ένα cookie.

Στην ASP.NET, ένα cookie αντιπροσωπεύεται από την κατηγορία HttpCookie. Διαβάζουμε τα cookie του χρήστη μέσω της ιδιότητας Request, και θέτουμε τα cookies με την ιδιότητα των cookies Response. Τα μπισκότα εξ ορισμού λήγουν όταν το παράθυρο του browser κλείσει (σαν το session state), αλλά το σημείο λήξης τους μπορεί να τεθεί σε μελλοντικές ημερομηνίες σε τέτοιες περιπτώσεις, τα cookies καλούνται persistent cookies.

### Default.aspx.vb

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

Partial Class _Default
    Inherits System.Web.UI.Page

    Sub View_More(ByVal Src As Object, ByVal Args As CommandEventArgs)

        Response.Redirect("~/offerdetails/Default.aspx?OfferID=" &
        Args.CommandName)

    End Sub

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

        Dim bc As HttpBrowserCapabilities = Request.Browser
```

```
If bc.IsMobileDevice = True Then
    Response.Redirect("/myspace/mobile/Default.aspx")
End If

Dim connectionString As String = "Data Source=.\SQLEXPRESS;
AttachDbFilename=C:\inetpub\wwwroot\myspace\App_Data\eshopdbfile.mdf;
Integrated Security=True;
Connect Timeout=30;
User Instance=True"

Dim con As SqlConnection = New SqlConnection(connectionString)

Dim dadCust As SqlCommand
Dim SQLString2 As String

SQLString2 = "SELECT Count(*) FROM PersonalOffers WHERE CustomerID = '" &
Session("CustomerMac") & "'"
dadCust = New SqlCommand(SQLString2, con)
dadCust.Connection.Open()
If dadCust.ExecuteScalar() = 0 Then
    Label1.Visible = True
Else

    Dim SQLString As String = "SELECT * FROM Products WHERE ProductID IN
(SELECT ProductID FROM PersonalOffers WHERE CustomerID = '" &
Session("CustomerMac") & "') ORDER BY ProductID DESC"
    SqlDataAdapter1.SelectCommand = SQLString

End If
con.Close()
End Sub
End Class
```

Στη Default σελίδα της εφαρμογής μας τώρα, καταρχάς δηλώνουμε τα namespaces που χρειαζόμαστε για τη χρησιμοποίηση της βάσης δεδομένων μέσω αυτού του κώδικα.

Στη συνέχεια ορίζουμε εμείς μια κλάση με όνομα View More με τη βοήθεια της οποίας για κάθε προϊόν – προφορά στη λίστα των προσφορών θα μεταφερόμαστε στην αντίστοιχη σελίδα λεπτομερειών του. Η έκφραση Response.Redirect ακολουθούμενη από τη διεύθυνση μεταφοράς και του κωδικού προϊόντος επαληθεύει τη παραπάνω περιγραφή.

Αμέσως μετά μπαίνουμε σε ένα κομμάτι κώδικα το οποίο εκτελείται κάθε φορά που η σελίδα μας φορτώνεται (Page Load). Κατά τη διαδικασία λοιπόν αυτή γίνονται οι ακόλουθες κινήσεις.

Καταρχάς γίνεται ο έλεγχος του browser, στη περίπτωση που ο browser του χρήστη είναι browser PDA ή mobile όπως χαρακτηρίζεται και από κώδικα, ο χρήστης μεταφέρεται στην αντίστοιχη σελίδα της PDA προβολής.

Αμέσως μετά ανοίγουμε μια σύνδεση με τη βάση δεδομένων όπου και μεταβιβάζουμε και ένα query με το οποίο γίνεται καταμέτρηση των καταγραφών στη βάση και πιο συγκεκριμένα στον πίνακα PersonalOffers με το χαρακτηριστικό του χρήστη, της MAC Address του δηλαδή. Αν κατά την καταμέτρηση των καταγραφών δε βρεθούν δεδομένα (dadCust.ExecuteScalar() = 0), τότε εμφανίζεται μια συμβουλή προς τον χρήστη ,στη λίστα προσφορών όπου τον ενημερώνει ότι για να του προβληθούν οι προσφορές θα πρέπει να συμπληρώσει τα προσωπικά του στοιχεία. Αν τώρα πάλι βρεθούν εγγραφές στη βάση δεδομένων στέλνεται ένα καινούριο query με το οποίο για κάθε κωδικό προϊόντος στον πίνακα PersonalOffers που αντιστοιχεί στον κωδικό πελάτη παίρνουμε και τα αντίστοιχα στοιχεία από τον πίνακα των προϊόντων για τη προβολή των προσφορών.

### Register.aspx.vb

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

Partial Class register_Default
    Inherits System.Web.UI.Page

    Sub RegisterUser(ByVal sender As Object, ByVal e As
System.EventArgs) Handles submitButton.Click

        Session("Username") = username.Text
        Dim userCookie As HttpCookie
        userCookie = Request.Cookies("UserID")
        If userCookie.Value = "Anonymous" Then
            userCookie = New HttpCookie("UserID", Session("Username"))
            userCookie.Expires = DateTime.Now.AddMonths(1)
            Response.Cookies.Add(userCookie)
        End If

        Dim connectionString As String = "Data
Source=. \SQLEXPRESS;AttachDbFilename=C:\Inetpub\wwwroot\mysp
ace\App_Data\eshopdbfile.mdf;Integrated Security=True;Connect
Timeout=30;User Instance=True"
        Dim con As SqlConnection = New
```

```
SqlConnection(connectionString)

    Dim dadPref As SqlCommand
    Dim dadCust As SqlCommand
    Dim SQLString As String
    Dim SQLString2 As String

    SQLString = "SELECT Count(*) FROM Customers WHERE CustomerID
= '" & Session("CustomerMac") & "' AND Username = '" &
Session("Username") & "'"
    dadCust = New SqlCommand(SQLString, con)
    dadCust.Connection.Open()
    If dadCust.ExecuteScalar() = 0 Then

        SQLString2 = "UPDATE Customers SET Username = '" &
Session("Username") & "' WHERE CustomerID = '" &
Session("CustomerMac") & "'"
        dadPref = New SqlCommand(SQLString2, con)
        dadPref.ExecuteNonQuery()
    End If
    con.Close()
    FormsAuthentication.RedirectToLoginPage(username.Text, True)
End Sub

End Class
```

Στη Register σελίδα της εφαρμογής μας τώρα δηλώνουμε πάλι τα απαραίτητα namespaces για τη χρησιμοποίηση της βάση δεδομένων.

Ενώ ορίζουμε και μία κλάση με το όνομα Register User η οποία εκτελείται με το που ο χρήστης πατήσει το κουμπί Register στη σελίδα μας.

Με το που πατηθεί λοιπόν το κουμπί Register από τον χρήστη το Session με όνομα Username παίρνει τη τιμή του TextBox όπου ο χρήστης έχει συμπληρώσει το όνομα του, στη συνέχεια γίνεται έλεγχος για ήδη υπάρχον cookie στον browser του χρήστη. Αν υπάρχει cookie και έχει UserID την τιμή Anonymous δημιουργείται ένα καινούριο Cookie με UserID την τιμή που έχει θέσει ο χρήστης, χρόνο λήξης ένα μήνα και αποθηκεύεται στον χρήστη.

Μετά τη διαδικασία αυτή ανοίγουμε μια σύνδεση με τη βάση δεδομένων όπου ψάχνουμε στον πίνακα Customers για χρήστες με εγγραφή πρώτον την MAC Address του χρήστη που γνωρίζουμε και δεύτερον το Username που έχει εισάγει ο χρήστης. Αν δεν υπάρχει τέτοια εγγραφή τότε αποθηκεύεται το Username που έχει διαλέξει ο χρήστης δίπλα στην αντίστοιχη εγγραφή της MAC Address του.



Τέλος κλείνουμε τη σύνδεση με τη βάση και μεταφέρουμε τα στοιχεία που έχει εισάγει ο χρήστης στη σελίδα Login για την είσοδό του στο σύστημα. Το όρισμα True μετά το Username.Text δηλώνει ότι το cookie παραμένει στον χρήστη ακόμα και μετά στο κλείσιμο του browser και έτσι ο χρήστης θα αναγνωρισθεί αυτόματα και την επόμενη φορά που θα εισέλθει στο σύστημα.

### Login.aspx.vb

```
Partial Class login_Default
    Inherits System.Web.UI.Page

    Sub LoginUser(ByVal s As Object, ByVal e As EventArgs)
        If (username.Text = Session("Username")) Then
            FormsAuthentication.RedirectFromLoginPage(username.Text,
True)
        End If
    End Sub

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

        Dim userCookie As HttpCookie
        userCookie = Request.Cookies("UserID")
        If userCookie Is Nothing Then
            Response.Redirect("../register/Default.aspx")
        Else
FormsAuthentication.RedirectFromLoginPage(Session("Username"),
True)
        End If
    End Sub

End Class
```

Στη Login σελίδα της εφαρμογής μας τώρα καταρχάς έχουμε ένα κομμάτι κώδικα το οποίο εκτελείται με την φόρτωση της σελίδας. Εδώ γίνεται ο έλεγχος ύπαρξης του cookie με το UserID του χρήστη. Αν δεν υπάρχει τέτοιο cookie ο χρήστης μεταφέρεται στη register σελίδα για να γίνει η εγγραφή του και να δημιουργηθεί και το cookie με το UserID του, αν υπάρχει τώρα το cookie ο χρήστης μεταφέρεται στη Default σελίδα της εφαρμογής μας.

### Preferences.aspx.vb

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

```
Partial Class preferences_Default
    Inherits System.Web.UI.Page

    Sub View_More(ByVal Src As Object, ByVal Args As
    CommandEventArgs)
        Response.Redirect("~/offerdetails/Default.aspx?OfferID=" &
        Args.CommandName)
    End Sub

    Sub Save_Preferences(ByVal sender As Object, ByVal e As
    System.EventArgs)

        Dim sexv As Integer = sex.SelectedItem.Value
        Dim agev As Integer = age.SelectedItem.Value
        Dim familyv As Integer = family.SelectedItem.Value
        Dim vegetarianv As Boolean = vegetarian.SelectedItem.Value
        Dim smokerv As Boolean = smoker.SelectedItem.Value
        Dim drinkerv As Boolean = drinker.SelectedItem.Value
        Dim petv As Boolean = pet.SelectedItem.Value

        Session("sex") = sexv
        Session("age") = agev
        Session("family") = familyv
        Session("vegetarian") = vegetarianv
        Session("smoker") = smokerv
        Session("drinker") = drinkerv
        Session("pet") = petv

        Dim connectionString As String = "Data
        Source=. \SQLEXPRESS;AttachDbFilename=C:\Inetpub\wwwroot\mys
        pace\App_Data\eshopdbfile.mdf;Integrated Security=True;Connect
        Timeout=30;User Instance=True"
        Dim con As SqlConnection = New
        SqlConnection(connectionString)

        Dim dadCust As SqlCommand
        Dim dadPref As SqlCommand
        Dim dadPref2 As SqlCommand
        Dim dadPref3 As SqlCommand
        Dim dadPref4 As SqlCommand

        Dim SQLString As String = "SELECT Count(*) FROM Customers
        WHERE CustomerID = " & Session("CustomerMac") & """"
        dadCust = New SqlCommand(SQLString, con)
        dadCust.Connection.Open()
        If dadCust.ExecuteScalar() = 0 Then
```

```

Dim SQLString1 As String = "INSERT INTO Customers (CustomerID
,Username ,CustomerSex ,CustomerAge ,CustomerFamily
,CustomerVegeterian ,CustomerSmoker ,CustomerDrinker
,CustomerPet) VALUES ( '" & Session("CustomerMAC") & "', '" &
Session("Username") & "', '" & sexv & "', '" & agev & "', '" & familyv & "', '"
& vegeterianv & "', '" & smokerv & "', '" & drinkerv & "', '" & petv & "'"
dadPref = New SqlCommand(SQLString1, con)
dadPref.ExecuteNonQuery()

Dim SQLString2 As String = "INSERT INTO PersonalOffers
(ProductID) SELECT ProductID FROM Products WHERE CustomerSex = '"
& Session("sex") & "' OR CustomerAge = '" & Session("age") & "' OR
CustomerFamily = '" & Session("family") & "' OR CustomerVegeterian =
'" & Session("vegeterian") & "' OR CustomerSmoker = '" &
Session("smoker") & "' OR CustomerDrinker = '" & Session("drinker") & "'
OR CustomerPet = '" & Session("pet") & "'"
dadPref2 = New SqlCommand(SQLString2, con)
dadPref2.ExecuteNonQuery()

Dim SQLString3 As String = "UPDATE PersonalOffers SET
CustomerID = '" & Session("CustomerMac") & "' WHERE CustomerID IS
NULL"
dadPref3 = New SqlCommand(SQLString3, con)
dadPref3.ExecuteNonQuery()

Else
Dim SQLString4 As String = "UPDATE Customers SET CustomerSex
= '" & sexv & "',CustomerAge = '" & agev & "', CustomerFamily = '" &
familyv & "', CustomerVegeterian = '" & vegeterianv & "',
CustomerSmoker = '" & smokerv & "', CustomerDrinker = '" & drinkerv &
"', CustomerPet = '" & petv & "' WHERE CustomerID = '" &
Session("CustomerMac") & "'"
dadPref = New SqlCommand(SQLString4, con)
dadPref.ExecuteNonQuery()

Dim SQLString5 As String = "DELETE FROM PersonalOffers WHERE
CustomerID = '" & Session("CustomerMac") & "'"
dadPref2 = New SqlCommand(SQLString5, con)
dadPref2.ExecuteNonQuery()

Dim SQLString6 As String = "INSERT INTO PersonalOffers
(ProductID) SELECT ProductID FROM Products WHERE CustomerSex = '"
& Session("sex") & "' OR CustomerAge = '" & Session("age") & "' OR
CustomerFamily = '" & Session("family") & "' OR CustomerVegeterian =
'" & Session("vegeterian") & "' OR CustomerSmoker = '" &
Session("smoker") & "' OR CustomerDrinker = '" & Session("drinker") & "'
OR CustomerPet = '" & Session("pet") & "'"

```

```
dadPref3 = New SqlCommand(SQLString6, con)
dadPref3.ExecuteNonQuery()

Dim SQLString7 As String = "UPDATE PersonalOffers SET
CustomerID = '' & Session("CustomerMac") & '' WHERE CustomerID IS
NULL"
dadPref4 = New SqlCommand(SQLString7, con)
dadPref4.ExecuteNonQuery()

End If
con.Close()
Response.Redirect("../Default.aspx")
End Sub

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

Dim connectionString As String = "Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\Inetpub\wwwroot\mys
pace\App_Data\eshopdbfile.mdf;Integrated Security=True;Connect
Timeout=30;User Instance=True"
Dim con As SqlConnection = New
SqlConnection(connectionString)

Dim dadCust As SqlCommand
Dim SQLString2 As String

SQLString2 = "SELECT Count(*) FROM PersonalOffers WHERE
CustomerID = '' & Session("CustomerMac") & ''"
dadCust = New SqlCommand(SQLString2, con)
dadCust.Connection.Open()
If dadCust.ExecuteScalar() = 0 Then
Label1.Visible = True
Else
Dim SQLString As String = "SELECT * FROM Products WHERE
ProductID IN (SELECT ProductID FROM PersonalOffers WHERE
CustomerID = '' & Session("CustomerMac") & '') ORDER BY ProductID
DESC"
SqlDataSource1.SelectCommand = SQLString
End If
con.Close()

End Sub
End Class
```

Στη σελίδα Preferences της εφαρμογής μας τώρα που είναι η σελίδα στην οποία ο χρήστης εισάγει τα προσωπικά του δεδομένα.

Καταρχάς χρησιμοποιούμε πάλι τα γνωστά namespaces για τη βάση δεδομένων. Επίσης έχουμε και εδώ τη κλάση View More που χρησιμοποιούμε για να μεταφερθούμε στη σελίδα με τις πληροφορίες των προσφορών.

Από κάτω ορίζουμε μία κλάση η οποία εκτελείται με το πάτημα του κουμπιού Save της φόρμας μας και η οποία εκτελεί τα ακόλουθα. Καταρχάς αποθηκεύει τις τιμές που έχουμε εισάγει σε αντίστοιχες μεταβλητές, στη συνέχεια δίνει τις τιμές αυτών των μεταβλητών σε αντίστοιχα sessions.

Εδώ ανοίγουμε μια καινούρια σύνδεση με τη βάση δεδομένων μας , όπου καταρχάς ψάχνουμε στον πίνακα Customers να βρούμε την εγγραφή με την γνωστή MAC Address. Αν δεν υπάρχει εγγραφή αποθηκεύουμε την MAC Address και τις επιλογές του χρήστη στον πίνακα και φιλτράροντας τις προσφορές από τον πίνακα Products αποθηκεύουμε ταυτόχρονα τα προϊόντα που επαληθεύουν τις επιλογές του χρήστη στον πίνακα PersonalOffers μαζί με την MAC Address του χρήστη.

Αν υπάρχει τώρα εγγραφή με την MAC Address του χρήστη κάνουμε update τον πίνακα Customers με καινούρια στοιχεία που ο χρήστης έχει εισάγει σβήνουμε τις εγγραφές την ταυτότητα του χρήστη από τον πίνακα PersonalOffers ξαναφιλτράρουμε τις προσφορές και αποθηκεύουμε τα καινούρια δεδομένα στον πίνακα.

Τέλος κλείνουμε τη σύνδεση με τη βάση μας και κάνουμε Redirect τον χρήστη στη Default σελίδα του συστήματός μας.

Στον κώδικα που εκτελείται κατά την φόρτωση της σελίδας μας τώρα , ανοίγουμε μία καινούρια σύνδεση με τη βάση όπου κάνουμε τον έλεγχο αν υπάρχει εγγραφή με την ταυτότητα του χρήστη στον πίνακα PersonalOffers. Αν δεν υπάρχει εμφανίζεται το μήνυμα που είχαμε πει και πιο πάνω για τη συμπλήρωση των προτιμήσεων. Αν υπάρχει λαμβάνονται οι πληροφορίες των προσφορών και προβάλλονται στη σχετική λίστα, αριστερά της σελίδας.

### OfferDetails.aspx.vb

```
Partial Class offerdetails_Default
    Inherits System.Web.UI.Page

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

        Dim SQLString As String
        SQLString = "SELECT * FROM Products WHERE ProductID = '" &
Request.QueryString("OfferID") & "'"
```

```
SqlDataSource2.SelectCommand = SQLString

    Dim SQLString2 As String = "SELECT * FROM Products WHERE
ProductID IN (SELECT ProductID FROM PersonalOffers WHERE
CustomerID = " & Session("CustomerMac") & ") ORDER BY ProductID
DESC"
    SqlDataSource1.SelectCommand = SQLString2

End Sub

Sub View_More(ByVal Src As Object, ByVal Args As
CommandEventArgs)

    Response.Redirect("~/offerdetails/Default.aspx?OfferID=" &
Args.CommandName)

End Sub

End Class
```

Στη σελίδα OfferDetails της εφαρμογής μας όπου προβάλλονται περαιτέρω πληροφορίες για κάθε προσφορά.

Με την φόρτωση της σελίδας μας λοιπόν τρέχουν δύο queries όπου με το πρώτο παίρνουμε από τη βάση δεδομένων της πληροφορίες για το προϊόν που θέλουμε να δούμε και με το δεύτερο τις πληροφορίες για όλες τις προσωπικές προσφορές.

Τα δύο αυτά queries τροφοδοτούν δύο συνδέσεις με τη βάση δεδομένων που βρίσκονται στον κώδικα της σελίδας μας γιαυτό και δεν δηλώσαμε όπως έχουμε κάνει και πιο πάνω τα αντίστοιχα namespaces.

# 4 ΕΠΙΛΟΓΟΣ

Στα κεφάλαια που πέρασαν προσπαθήσαμε να δώσουμε μια γενική εικόνα της εφαρμογής , στόχος μας ήταν ένας χρήστης ανεξαρτήτου γνώσεων πάνω στο θέμα της διπλωματικής και στα όσα γίνεται λόγος στο παρόν κείμενο να μπορεί να κατανοήσει τη λειτουργία και τη συμπεριφορά του συστήματος.

Παρόλαυτά η παραπάνω εφαρμογή είναι ένα μικρό παράδειγμα στα όσα η τεχνολογία ASP.NET μπορεί να προσφέρει. Η ανάπτυξη του θέματος σε βάθος που γίνεται κυρίως στο τρίτο κεφάλαιο , μπορεί να δώσει αρκετές γόνιμες ιδέες για περαιτέρω ανάπτυξη της εφαρμογής ή ακόμα και παραμετροποίησής της στα γούστα και τις ανάγκες της κάθε περίπτωσης.

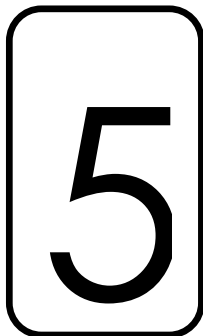
Μία από τις πολλές ιδέες για την ανάπτυξη της εφαρμογής που θα μπορούσαμε να πούμε ,ότι θα ήταν το ανέβασμα της στο διαδίκτυο και τη μετατροπή της από ένα απλό εργαλείο καταγραφής στοιχείων και προβολής προσφορών, σε μία διαδικτυακή βιτρίνα της επιχείρησης με αρκετά μεγαλύτερο πλήθος υποψήφιων πελατών. Θα ήταν επίσης ενδιαφέρον η μετατροπή της εφαρμογής σε online κατάσταση.

Το γεγονός βέβαια ότι η εφαρμογή στηρίζεται στο ασύρματο δίκτυο για τη πρόσβασή της δε θα έπρεπε να θεωρείται επουδενεί σαν μειονέκτημα, καθώς υπάρχει ένας μακρύς κατάλογος πλεονεκτημάτων που προσφέρει η συγκεκριμένη τεχνολογία.

Λαμβάνοντας υπόψιν όλα αυτά μπορούσαμε να πούμε και για υποθετικές χρήσης της εφαρμογής με άλλον τρόπο. Ένα κατάστημα μουσικής για παράδειγμα θα έβρισκε πολύ χρήσιμο ένα τέτοιο εργαλείο.

Ένας καλά σχεδιασμένος κατάλογος των μουσικών CD και DVD του καταστήματος θα ήταν στη διάθεση των πελατών μέσω μιας απλής διαδικασίας αναζήτησης επίσης η καταγραφή των προτιμήσεων των πελατών, όπως είδαμε και στην εφαρμογή μας ,θα ήταν μια πολύ καλή κίνηση από τη πλευρά του καταστήματος.

Πέραν αυτών μία τέτοια εφαρμογή θα μπορούσε να μη περιοριστεί κατανάγκη μόνο στα πλαίσια της προβολής και προώθησης προϊόντων. Μία ιδιωτική νοσοκομειακή δικτυακή εφαρμογή θα ήταν μια ακόμα ακόμα τρόπος λειτουργίας του.



## ΒΙΒΛΙΟΓΡΑΦΙΑ

### ● ΙΣΤΟΣΕΛΙΔΕΣ

- Προγραμματίζοντας το Internet με το Microsoft .NET  
[www.in.gr/computer/net/default.htm](http://www.in.gr/computer/net/default.htm)
- Το intranet.  
[http://www.go-online.gr/ebusiness/specials/article.html?article\\_id=1236](http://www.go-online.gr/ebusiness/specials/article.html?article_id=1236)
- Wi-Fi  
<http://www.omnia.ondsl.gr/active.aspx?mode=el%7Be313ab8e-a9f9-4f32-aba1-c9e16f143092%7DView>
- ASP.NET 2.0 Tutorial  
<http://msconline.maconstate.edu/tutorials/ASPNET20/default.htm>
- ASP.NET 2.0  
<http://www.asp.net>
- ASP.NET 2.0  
<http://msdn2.microsoft.com/en-us/asp.net/default.aspx>

### ● ΒΙΒΛΙΑ

- **Professional ASP.NET 2.0 Databases** Thiru Thangarathinam - Wrox Pub.
- **Building .NET Applications for Mobile Devices** Andy Wigley, Peter Roxburgh - Microsoft Press
- **Managing E-Commerce and Mobile Computing Technologies** Julie Mariga - Idea Group Publishing



- **Writing Mobile Code Essential Software Engineering for Building Mobile Applications** *Ivo Salmre - Addison Wesley Professional*
- **.NET Compact Framework Programming with Visual Basic .NET** *Paul Yao, David Durant - Addison Wesley*